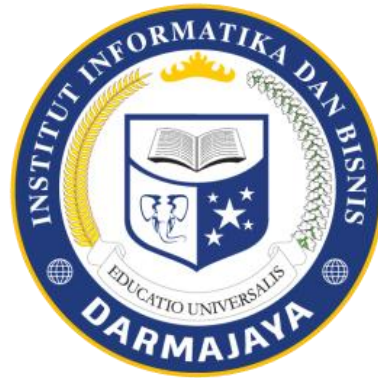


**SISTEM PENGAMAN KENDARAAN RODA EMPAT MENGGUNAKAN  
E-KTP BERBASIS MIKROKONTROLER**

**SKRIPSI**  
**Diajukan Sebagai Salah Satu Syarat untuk Mencapai Gelar**  
**SARJANA**  
**Pada Program Studi Sistem Komputer**  
**IIB Darmajaya Bandar Lampung**



**Oleh**  
**I GEDE PUTRA BUDIARTHA**  
**1511060002**

**FAKULTAS ILMU KOMPUTER**  
**PROGRAM STUDI SISTEM KOMPUTER**  
**INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA**  
**BANDAR LAMPUNG**  
**2019**

## PERNYATAAN ORISINILITAS PENELITIAN



### PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa Skripsi yang diajukan ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggungjawaban sepenuhnya berada dipundak saya.

Bandar Lampung, 19 Maret 2019

  
**METERAI  
TEMPEL**  
6000  
ENAM RIBU RUPIAH  
FAFF801387701  
**I GEDE PUTRA BUDIARTHA**  
1511060002

## HALAMAN PERSETUJUAN

Judul Skripsi : **SISTEM PENGAMAN KENDARAAN RODA  
EMPAT MENGGUNAKAN E-KTP BERBASIS  
MIKROKONTROLER**

Nama Mahasiswa : **I GEDE PUTRA BUDIARTHA**

No. Pokok Mahasiswa : 1511060006

Program Studi : **S 1 Sistem Komputer**

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam sidang Tugas Penutup Studi guna memperoleh gelar Sarjana Ilmu Komputer pada Program Studi Sistem Komputer IIB Darmajaya.



Disetujui oleh :

Dosen Pembimbing,



**Novi Herawadi Sudibyo, S.Kom., M.T.I**  
NIK 11690310

Ketua Program Studi,  
Sistem Komputer



**Bayu Nugroho, S.Kom., M.Eng**  
NIK 00200700

## HALAMAN PENGESAHAN

Telah diuji dan dipertahankan didepan Tim Penguji Skripsi  
Program Studi Sistem Komputer Institut Informatika dan Bisnis Darmajaya  
Bandar Lampung dan dinyatakan diterima untuk  
memenuhi syarat guna memperoleh  
Gelar Sarjana Komputer

### Mengesahkan

#### 1. Tim Penguji

Tanda Tangan

Ketua : **Abdi Darmawan, S.T., M.T.I**

Anggota : **Melia Gripin, S.Kom., M.T**

Dekan Fakultas Ilmu Komputer

**Sriyanto, S.Kom., M.M., Ph.D**

NIK 00210800

Tanggal Lulus Ujian Skripsi : 19 Maret 2019

## **HALAMAN PERSEMBAHAN**

Puji dan syukur Astungkare saya panjatkan kepada Ida Sang Hyang Widhi atas waranugraha dan inspirasi yang sangat melimpah sehingga saya dapat menyelesaikan skripsi ini ku persembahkan kepada :

1. Ayahanda tercinta I Gede Supartha yang telah memberikan saya semangat tanpa henti dan membawa saya sampai ke jenjang perkuliahan.
2. Ibuku tercinta Ni Made Yase terima kasih atas semua yang telah atas semua yang telah engkau berikan, kasih sayang yang begitu besar serta arahan dan motivasi dalam segala hal, tak lepas doamu untuk keberhasilan anakmu. You are my everything.
3. Kakak ku Putu Wiyanti yang selalu memberikan saya semangat serta dukungan untuk menyelesaikan tugas akhir (skripsi).
4. Adik –adikku Nyoman Ayu Widya Ningsih , I Ketut Andika Widya Parma, Putu Dina Kartika Putri, Made Dini Anggita Sari, Nyoman Gita Gayatri yang selalu memberikan doa serta dukungan.
5. Sahabat – sahabatku dan teman – teman yang tak bias ku sebutkan satu persatu yang telah memberikan semangat, kritikan yang membangun serta dukungan dalam menyusun karya ini.
6. Seluruh keluarga besarku yang selama ini mendukungku selamaku menuntut ilmu diperguruan tinggi IIB Darmajaya.
7. Seluruh dosen-dosen IIB Darmajaya terima kasih semua, khususnya dosen-dosen Program Studi Sistem Komputer dan Teknik Komputer.
8. Almamater tercinta ku IIB DARMAJAYA.

## MOTTO

*“kasih sayang orang tua ibarat bahan bakar yang membuat orang tua mampu melakukan hal-hal yang mustahil menjadi mungkin tuk dilakukannya ”*

*(Orang Tua Tercinta)*

*“perbuatan-perbuatan salah adalah biasa bagi manusia, tetapi perbuatan pura-pura itulah sebenarnya yang menimbulkan pengkhianatan”*

*(I GEDE PUTRA BUDIARTHA)*

## ABSTRAK

### SISTEM PENGAMAN KENDARAAN RODA EMPAT MENGGUNAKAN E-KTP BERBASIS MIKROKONTROLER

Oleh  
*I GEDE PUTRA BUDIARTHA*

Berdasarkan data Kepolisian Republik Indonesia, dari awal Tahun 2014 Statistik Kriminal terjadi kasus (pencurian Kendaraan) curanmor mencapai 294 kasus (Statistik, Badan Pusat, 2014). Kasus pencurian kendaraan terus meningkat setiap tahunnya. Hal ini sangat meresahkan pengguna kendaraan baik kendaraan roda dua ataupun kendaraan roda empat. Sistem pengamanan yang digunakan saat ini masih menggunakan analog, sehingga perlu diganti menggunakan sistem digital untuk meningkatkan sistem keamanan tersebut. Dari permasalahan diatas, maka peneliti ingin membuat sebuah “Sistem Pengaman Kendaraan Roda Empat Menggunakan E-Ktp Berbasis Mikrokontroler”. Peneliti menggunakan raider *RFID* sebagai inputan dalam membaca nomer ID pada E-KTP yang akan diproses oleh arduino nano sehingga akan menghasilkan output menyalakan kelitrikan, stater dan mematikan kendaraan. Jika E-KTP tidak terdeteksi sebanyak 3 kali maka *GSM Shield* akan mengirimkan *SMS (Short Massage Service)* kepada pemilik mobil. Dari hasil uji coba dapat diketahui yaitu jika hanya melakukan 1 kali *Scand* maka relay 1 akan *HIGH* untuk menyalakan kelitrikan sedangkan pada ujicoba ke 2 melakukan 2 kali scan yang berarti pemilik akan menyalakan kelitrikan dan stater (Engine) kendaraan yaitu relay 1 dan 2 akan *HIGH* dan pada ujicoba ke 3 melakukan 3 kali scan artinya kendaraan dimatikan. Serta jika kesalahan dalam melakukan scan sebanyak 3 kali maka *GSM Shield* akan mengirimkan *SMS* kepada pemilik kendaraan dan sistem akan diblokir selama 5 menit mengirimkan *SMS* apabila dalam melakukan scan E-KTP salah sebanyak 3 kali.

Kata Kunci : E-KTP, *RFID*, *Arduino*, Roda Empat, Pencurian

## ABSTRACT

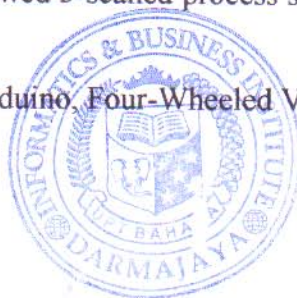
### THE MICROCONTROLLER-BASED E-ID CARD FOR THE FOUR-WHEEL VEHICLE SECURITY SYSTEM

By:

I GEDE PUTRA BUDIARTHA

According to the data of Indonesian National Police, there were 294 criminal cases on the vehicle theft (both two-wheeled and four-wheeled vehicles in the beginning of 2014 (Central Bureau of Statistics, 2014). This vehicle theft was increasing every year. The recent security system used on the vehicles was still analogous so that it needed to be replaced through the digital system to improve the security system. The researcher built "The Microcontroller-Based E-ID Card for the Four-Wheel Vehicle Safety System". The researcher used RFID reader as the input to read the ID numbers on the E-ID Card processed by Arduino nano so that it generated the output, turned on the electricity and the engine, and turned off the vehicle. If the E-ID Card was not detected 3 times, the GSM Shield sent an SMS (Short Message Service) to the vehicle owner and the system was blocked during 5 minutes. The result of this first test showed 1-scanned process, so that the relay 1 was HIGH to turn on the electricity. Moreover, the result of this second test showed that 2-scanned process so that the relay 1 and 2 were HIGH. Besides, the result of this third test showed 3-scanned process so that the vehicle was turned off.

**Keywords:** E-KTP, RFID, Arduino, Four-Wheeled Vehicle, Theft





## KATA PENGANTAR

Puji syukur penyusun panjatkan kepada Ida Sang Hyang Widhi Wasa atas Waranugrahanya sehingga saya dapat menyelesaikan suatu tugas dari jurusan dalam bentuk skripsi yang menjadi kewajiban bagi setiap mahasiswa Jurusan Sistem Komputer IIB Darmajaya Bandar Lampung dengan judul ***“SISTEM PENGAMAN KENDARAAN RODA EMPAT MENGGUNAKAN E-KTP BERBASIS MIKROKONTROLER”*** Skripsi ini disusun untuk memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer (S.Kom) Sistem Komputer, Institut Informatika dan Bisnis Darmajaya.

Dengan selesainya skripsi ini, saya mengucapkan terima kasih yang setulus-tulusnya kepada :

1. Bapak Dr. Hi., Andi Desfiandi, Se, Ma. Selaku ketua yayasan Institut Informatika dan Bisnis Darmajaya;
2. Bapak Ir. Firmansyah YA., MBA., M. Sc., Selaku Rektor IIB Darmajaya Bandar Lampung;
3. Bapak Sriyanto,S.Kom.,M.M Selaku Wakil Rektor I Institut Informatika dan Bisnis Darmajaya;
4. Bapak Bayu Nugroho,S.Kom., M.Eng Selaku Ketua Program Studi Sistem Komputer dan Teknik Komputer , terimakasih atas waktu dan saran yang telah bapak berikan kepada saya;
5. Bapak Novi Herawadi Sudibyo ,S.Kom., M.T.I selaku Sekretaris Program Studi Teknik Komputer dan Sistem Komputer, dan sekaligus sebagai pembimbing saya dalam menyelesaikan laporan skripsi ini. terimakasih atas waktu dan saran yang telah bapak berikan kepada saya.
6. Bapak dan Ibu ku tercinta yang selalu setia mendukungku baik dalam material maupun spiritual selama penyusunan skripsi ini.

7. Bapak dan Ibu Dosen Pengajar terutama jurusan Sistem Komputer yang telah membagi ilmu dan pengetahuan mereka yang bermanfaat kepada penyusunan dalam pembelajaran.
8. Kakak ku Putu Wiyanti yang selalu memberikan saya semangat serta dukungan untuk menyelesaikan tugas akhir (skripsi).
9. Adik –adikku Nyoman Ayu Widya Ningsih , I Ketut Andika Widya Parma, Putu Dina Kartika Putri, Made Dini Anggita Sari, Nyoman Gita Gayatri yang selalu memberikan doa serta dukungan;
10. Rekan – rekan seperjuangan jurusan Sistem Komputer Ilmu Komputer IIB Darmajaya serta mahasiswa/mahasiswi angkatan 2019, yang telah memotivasi dan membantu demi terwujudnya skripsi ini.
11. Terimakasih untuk seluruh keluarga besar atas dukungannya.
12. Seluruh pihak yang membantu dalam kelancaran dan kesuksesan Skripsi penulis.

Bandar Lampung,

**I GEDE PUTRA BUDIARTHA**  
**1511060002**



## DAFTAR ISI

PERNYATAAN ORISINILITAS PENELITIAN .....	<b>Error! Bookmark not defined.</b>
PERSETUJUAN.....	<b>Error! Bookmark not defined.</b>
PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
HALAMAN PERSEMBAHAN .....	<b>Error! Bookmark not defined.</b>
MOTTO .....	<b>Error! Bookmark not defined.</b>
ABSTRAK.....	<b>Error! Bookmark not defined.</b>
ABSTRACT.....	<b>Error! Bookmark not defined.</b>
KATA PENGANTAR .....	<b>Error! Bookmark not defined.</b>
DAFTAR ISI.....	1
DAFTAR TABEL.....	4
DAFTAR GAMBAR .....	5
DAFTAR LAMPIRAN.....	6
BAB I PENDAHULUAN.....	<b>Error! Bookmark not defined.</b>
1.1 Latar Belakang .....	<b>Error! Bookmark not defined.</b>
1.2 Ruang Lingkup Penelitian .....	<b>Error! Bookmark not defined.</b>
1.3 Rumusan Masalah .....	<b>Error! Bookmark not defined.</b>
1.4 Tujuan Penelitian.....	<b>Error! Bookmark not defined.</b>
1.5 Manfaat Penelitian.....	<b>Error! Bookmark not defined.</b>
1.6 Sistematika Penulisan.....	<b>Error! Bookmark not defined.</b>
BAB II TINJAUAN PUSTAKA .....	<b>Error! Bookmark not defined.</b>
1.1 <i>Studi Literatur</i> .....	<b>Error! Bookmark not defined.</b>
2.2 Dasar Teori .....	<b>Error! Bookmark not defined.</b>
2.2.1 <i>Sistem</i> .....	<b>Error! Bookmark not defined.</b>
2.2.2 Pengertian Keamanan.....	<b>Error! Bookmark not defined.</b>
2.2.3 Kelebihan e-KTP Sebagai Tag Pasif.	<b>Error! Bookmark not defined.</b>
2.2.4 Kendaraan .....	<b>Error! Bookmark not defined.</b>
2.3.1 <i>Aki (Accu)</i> .....	<b>Error! Bookmark not defined.</b>
2.3.2 <i>Radio Frequency Identification (RFID)</i>	<b>Error! Bookmark not defined.</b>
	<b>defined.</b>

2.3.3	Sistem <i>RFID</i> .....	<b>Error! Bookmark not defined.</b>
2.3.3.1	Pembaca <i>RFID</i> .....	<b>Error! Bookmark not defined.</b>
2.3.3.2	Tag <i>RFID</i> .....	<b>Error! Bookmark not defined.</b>
2.3.4	Relay .....	<b>Error! Bookmark not defined.</b>
2.3.4.1	Prinsip Kerja Relay .....	<b>Error! Bookmark not defined.</b>
2.3.4.2	Jenis-jenis Relay .....	<b>Error! Bookmark not defined.</b>
2.3.4.3	Fungsi-Fungsi <i>Relay</i> .....	<b>Error! Bookmark not defined.</b>
2.3.4.4	Driver Relay .....	<b>Error! Bookmark not defined.</b>
2.3.4.5	Interface Driver Relay .....	<b>Error! Bookmark not defined.</b>
2.3.5	<i>Internet Shield SIM900</i> .....	<b>Error! Bookmark not defined.</b>
2.3.6	<i>SMS (Short Massage Service)</i> .....	<b>Error! Bookmark not defined.</b>
2.3.7	Mikrokontroler .....	<b>Error! Bookmark not defined.</b>
2.3.7.1	Modul Arduino Nano .....	<b>Error! Bookmark not defined.</b>
2.3.7.2	Blok Arduino Nano .....	<b>Error! Bookmark not defined.</b>
2.4	Perangkat Lunak Yang Digunakan .....	<b>Error! Bookmark not defined.</b>
2.4.1	<i>Software</i> Mikrokontroler Arduino ..	<b>Error! Bookmark not defined.</b>
2.4.1.1	Program Arduino Ide .....	<b>Error! Bookmark not defined.</b>
2.4.2	<i>Software</i> ISIS & ARES Proteus 7.0 ..	<b>Error! Bookmark not defined.</b>
BAB III METODE PENELITIAN .....		<b>Error! Bookmark not defined.</b>
3.1	<i>Studi Literatur</i> .....	<b>Error! Bookmark not defined.</b>
3.2	Analisa Perancangan Sistem .....	<b>Error! Bookmark not defined.</b>
3.2.1	Perancangan Perangkat Keras .....	<b>Error! Bookmark not defined.</b>
3.2.1.1	Rangkaian <i>RFID (Radio Frequency Identification)</i> .....	<b>Error! Bookmark not defined.</b>
3.2.1.2	Rangkaian <i>Gsm Shield</i> .....	<b>Error! Bookmark not defined.</b>
3.2.1.3	Rangkaian <i>Relay</i> .....	<b>Error! Bookmark not defined.</b>
3.2.1.4	Rangkaian <i>Keseluruhan</i> .....	<b>Error! Bookmark not defined.</b>
3.2.2	Perancangan Perangkat Lunak .....	<b>Error! Bookmark not defined.</b>
3.3	Analisa Kebutuhan .....	<b>Error! Bookmark not defined.</b>
3.3.1	Alat .....	<b>Error! Bookmark not defined.</b>
3.3.2	Komponen .....	<b>Error! Bookmark not defined.</b>

3.3.3	Software .....	<b>Error! Bookmark not defined.</b>
3.4	Implementasi .....	<b>Error! Bookmark not defined.</b>
3.4.1	Implementasi Perangkat Keras.....	<b>Error! Bookmark not defined.</b>
3.4.2	Implementasi Perangkat Lunak.....	<b>Error! Bookmark not defined.</b>
3.5	Pengujian Sistem .....	<b>Error! Bookmark not defined.</b>
3.5.1	Pengujian Mikrokontroller .....	<b>Error! Bookmark not defined.</b>
3.5.2	Rancangan Pengujian RFID ( <i>Radio Frequency Identification</i> )	<b>Error!</b>
	<b>Bookmark not defined.</b>	
3.5.3	Rancangan Pengujian <i>Gsm Shield</i> .....	<b>Error! Bookmark not defined.</b>
3.5.4	Rancangan Pengujian <i>Relay</i> .....	<b>Error! Bookmark not defined.</b>
3.5.5	Pengujian Sistem Keseluruhan.....	<b>Error! Bookmark not defined.</b>
3.6	Analisis Kerja .....	<b>Error! Bookmark not defined.</b>
BAB IV HASIL DAN PEMBAHASAN .....		<b>Error! Bookmark not defined.</b>
4.1	Hasil.....	<b>Error! Bookmark not defined.</b>
4.1.1	Hasil Pengujian dan Pembahasan.....	<b>Error! Bookmark not defined.</b>
4.1.2	Pengujian Scan E-KTP.....	<b>Error! Bookmark not defined.</b>
4.1.3	Pengujian E-KTP Menyalakan Dan Mematikan Kendaraan ....	<b>Error!</b>
	<b>Bookmark not defined.</b>	
4.2	Pengujian Sistem Secara Keseluruhan ....	<b>Error! Bookmark not defined.</b>
BAB V KESIMPULAN DAN SARAN.....		<b>Error! Bookmark not defined.</b>
5.1	Kesimpulan.....	<b>Error! Bookmark not defined.</b>
5.2	Saran .....	<b>Error! Bookmark not defined.</b>
DAFTAR PUSTAKA .....		<b>Error! Bookmark not defined.</b>

## DAFTAR TABEL

Tabel 3.3	Alat Yang Dibutuhkan .....	<b>Error! Bookmark not defined.</b>
Tabel 3.4	Komponen Yang Dibutuhkan .....	<b>Error! Bookmark not defined.</b>
Tabel 3.5.	Daftar <i>Software</i> Yang Digunakan .....	<b>Error! Bookmark not defined.</b>
Tabel 4.2.	Hasil Pengujian RFID .....	<b>Error! Bookmark not defined.</b>
Tabel 4.3.	Hasil Pengujian <i>Relay</i> .....	<b>Error! Bookmark not defined.</b>
Tabel 4.4.	Hasil Pengujian <i>Gsm Shield</i> .....	<b>Error! Bookmark not defined.</b>
Tabel 4.5.	Hasil Pengujian Sistem Keseluruhan ....	<b>Error! Bookmark not defined.</b>

## DAFTAR GAMBAR

Gambar 2.1 Lapisan e-KTP .....	<b>Error! Bookmark not defined.</b>
Gambar 2.2 Simulasi dalam Aki dan Contoh Aki.....	<b>Error! Bookmark not defined.</b>
Gambar 2.3. Sistem <i>RFID</i> .....	<b>Error! Bookmark not defined.</b>
Gambar 2.4 Gambar dan Simbol Relay .....	<b>Error! Bookmark not defined.</b>
Gambar 2.5 Relay .....	<b>Error! Bookmark not defined.</b>
Gambar 2.6 Struktur Sederhana Relay.....	<b>Error! Bookmark not defined.</b>
Gambar 2.7 Jenis Relay berdasarkan <i>Pole</i> dan <i>Throw</i> .....	<b>Error! Bookmark not defined.</b>
Gambar 2.8 Rangkaian <i>Driver Relay</i> .....	<b>Error! Bookmark not defined.</b>
Gambar 2.9 Rangkaian Interface Driver Relay.....	<b>Error! Bookmark not defined.</b>
Gambar 2.10 <i>Internet Shield (GSM)</i> .....	<b>Error! Bookmark not defined.</b>
Gambar 2.11. Arduino Nano Atmega328 .....	<b>Error! Bookmark not defined.</b>
Gambar 2.12. Blok Diagram Arduino Nano .....	<b>Error! Bookmark not defined.</b>
Gambar 2.13. Tampilan Program <i>Arduino Nano</i> .....	<b>Error! Bookmark not defined.</b>
Gambar 2.14. Tampilan Software ISIS & ARES Proteus.	<b>Error! Bookmark not defined.</b>
Gambar 3.1. Alur Penelitian .....	<b>Error! Bookmark not defined.</b>
Gambar 3.2. Blok Diagram Sistem .....	<b>Error! Bookmark not defined.</b>
Gambar 3.4 Rangkaian RFID ( <i>Radio Frequency Identification</i> )	<b>Error! Bookmark not defined.</b>
<b>defined.</b>	
Gambar 3.5 Rangkaian <i>Gsm Shield</i> .....	<b>Error! Bookmark not defined.</b>
Gambar 3.6 Rangkaian <i>Relay</i> .....	<b>Error! Bookmark not defined.</b>
Gambar 3.7 Rancangan Rangkaian Keseluruhan.....	<b>Error! Bookmark not defined.</b>
Gambar 3.8 <i>Flowcart</i> Penyalaan Kendaraan Roda Empat.	<b>Error! Bookmark not defined.</b>
Gambar 3.9. Tampilan <i>Software</i> Arduino IDE .....	<b>Error! Bookmark not defined.</b>
Gambar 3.10. Pengaturan Port Arduino Uno .....	<b>Error! Bookmark not defined.</b>
Gambar 3.11. Hasil Compile Program.....	<b>Error! Bookmark not defined.</b>
Gambar 3.12. <i>Upload</i> Program.....	<b>Error! Bookmark not defined.</b>
Gambar. 4.1. Bentuk Fisik Alat .....	<b>Error! Bookmark not defined.</b>
Gambar 4.2 Pengujian Scan E-KTP.....	<b>Error! Bookmark not defined.</b>
Gambar 4.4 Peringatan SMS Jika Terjadi Kebakaran .....	<b>Error! Bookmark not defined.</b>



## DAFTAR LAMPIRAN

Lampiran 1. Program .....	48
Lampiran 2. Datasheet Arduino nano .....	52
Lampiran 3. Datasheet LCD .....	62
Lampiran 4. Datasheet RFID .....	65
Lampiran 5. Datasheet Relay 5 volt.....	68
Lampiran 6. Datasheet Sensor MQ2 .....	102
Lampiran 7. Datasheet Motor GSM Shield SIM900 .....	105

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Masyarakat di Indonesia sekarang khawatir dengan keamanan barang privasinya seperti Mobil sudah banyak pembobolan mobil yang terjadi pada bagian kontak mobil di daerah yang kurang aman atau kecerobohan pemilik kendaraan yang meninggalkan mobilnya dalam keadaan hidup. Berdasarkan data Kepolisian Republik Indonesia, dari awal Tahun 2014 Statistik Kriminal terjadi kasus (pencurian Kendaraan) curanmor mencapai 294 kasus (Statistik, Badan Pusat, 2014).

Kasus pencurian kendaraan terus meningkat setiap tahunnya. Hal ini sangat meresahkan pengguna kendaraan baik kendaraan roda dua ataupun kendaraan roda empat. Sistem pengamanan yang digunakan saat ini masih menggunakan analog, sehingga perlu diganti menggunakan sistem digital untuk meningkatkan sistem keamanan tersebut. Penggunaan sistem kendali elektronik hampir mencakup sebagian besar kehidupan sehari-hari manusia. Sistem kendali elektronik bersifat praktis dan efisien, sehingga banyak orang yang menyukainya. Sistem kendali elektronik digital dibuat untuk menggantikan sistem analog karena memiliki kelebihan, yaitu praktis, efisien, dan lebih futuristic.

Dalam membuat sistem pengaktifan dan keamanan kendaraan peneliti menggunakan beberapa komponen yaitu: peneliti menggunakan RFID yang digunakan sebagai inputan dalam untuk dapat membaca card E-KTP yang akan digunakan sebagai inputan dalam pengaktifan kendaraan yang ke dua peneliti menggunakan arduino yang akan digunakan sebagai proses dari keseluruhan sistem sehingga jika terjadinya pencurian atau pengaktifan kendaraan dapat memberikan informasi melalui pesan SMS (*Short Message Service*).

Berbagai upaya terus dilakukan untuk meningkatkan keamanan kendaraan roda empat yang sedang diparkir, baik di tempat parkir umum maupun di halaman rumah, agar tidak mudah dicuri. Salah satu upaya meningkatkan keamanan kendaraan roda empat dilakukan oleh peneliti (Rahman, 2015) dengan judul Rancang Bangun Stater Kendaraan Bermotor Menggunakan sistem kerja dari alat ini yaitu sistem yang dirancang menggunakan kartu RFID akan mendeteksi kartu yang telah memiliki id khusus. mikrokontroler yang digunakan adalah mikrokontroler atmega 16. dengan kartu RFID maka kendaraan bermotor bisa dijaga keamanannya, karena untuk menyalakan kendaraan bermotor harus menggunakan RFID.

Dari permasalahan diatas, maka peneliti ingin membuat sebuah **“SISTEM PENGAMAN KENDARAAN RODA EMPAT MENGGUNAKAN E-KTP BERBASIS MIKROKONTROLER”**. Dalam merancang sistem keamanan kendaraan roda empat menggunakan E-KTP peneliti menggunakan raider RFID sebagai inputan dalam membaca nomer ID pada E-KTP yang akan diproses oleh arduino nano sehingga akan menghasilkan outputan menyalakan kelitrikan , stater dan mematikan kendaraan. Jika E-KTP yang ditemplkan salah sebanyak 3 kali maka GSM Shield akan mengirimkan SMS (*Short Massage Service*) kepada pemilik mobil.

## **1.2 Ruang Lingkup Penelitian**

Berdasarkan dari hasil penelitian yang telah dilakukan, maka ruang lingkup dalam penelitian ini, yaitu;

1. Mikrokontroler yang digunakan adalah AT Mega 328.
2. Alat ini hanya menghidupkan kelistrikan dan stater kendaraan.
3. GSM shield SIM900A sebagai outputan dalam bentuk pengiriman pesan SMS (*Short Massage Service*) jika penggunaan telah mengaktifkan kendaraan.
4. RFID digunakan sebagai inputan dalam membaca E-KTP.

### **1.3 Rumusan Masalah**

Berdasarkan dari latar belakang yang telah dikemukakan, maka rumusan masalah dalam penelitian ini.

Bagaimana merancang dan membuat sistem yang dapat menghidupkan kelistrikan dan stater menggunakan E-KTP serta dapat mengirimkan perintah SMS (*Short Message Service*) dengan menggunakan arduino nano.

### **1.4 Tujuan Penelitian**

Tujuan dari pembuatan alat ini yaitu dapat membuat sistem pengaktifan dan keamanan kendaraan roda empat yang dikontrol menggunakan *E-KTP* dengan sistem relay pada kabel pengapian mesin serta dapat mengirimkan (SMS *Short Message Service*) jika E-KTP yang ditempelkan salah sebanyak 3 kali.

### **1.5 Manfaat Penelitian**

Manfaat dari penelitian ini adalah

1. Dapat meningkatkan keamanan kendaraan .
2. Dapat meminimalisir terjadinya kehilangan kendaraan .
3. Dapat memperlambat dalam percurian kendaraan.

### **1.6 Sistematika Penulisan**

Sistematika penulisan yang digunakan dalam tugas akhir ini terbagi dalam beberapa pokok bahasan, yaitu :

#### **BAB I PENDAHULUAN**

Dalam bab ini berisikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian dan manfaat penelitian.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini berisikan tentang teori – teori yang berkaitan dengan Sistem Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler”.

### BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan apa yang akan digunakan dalam uji coba pembuatan alat, tahapan perancangan dari alat, diagram blok dari alat, dan cara kerja alat tersebut.

### BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang implementasi alur, analisis dan pembahasan dari alur yang dirancang.

### BAB V SIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari pengujian sistem serta saran apakah rangkaian ini dapat digunakan secara tepat dan dikembangkan perakitannya.

### DAFTAR PUSTAKA

### LAMPIRAN



## **BAB II**

### **TINJAUAN PUSTAKA**

#### ***1.1 Studi Literatur***

Penelitian tentang menyalakan Kendaraan menggunakan *E-KTP* sudah pernah dilakukan oleh beberapa peneliti. Ringkasan *Studi Literatur* yang dilakukan untuk mengetahui sejauh mana penelitian yang sudah ada dapat dilihat sebagai berikut:

1. (Saputro, 2016) dengan judul Rancang Bangun Pengaman Pintu Otomatis Menggunakan E-KTP Berbasis Mikrokontroler Atmega328 sistem kerja dari alat ini yaitu RFID reader yang digunakan memiliki frekuensi 13,56 MHz yang diletakkan dalam box dengan tebal 2mm dapat membaca ID E-KTP dengan jarak maksimal 1,8 cm. Solenoid dapat membuka pengunci pintu apabila ID E-KTP sesuai dengan memori mikrokontroler ATmega328, solenoid akan mengunci kembali dalam waktu 10 detik Alat pengaman pintu otomatis menggunakan E-KTP ini mampu membaca ID E-KTP dengan jarak maksimal 1.8cm dengan sensor RFID *reader* MFRC522 yang memiliki frekuensi 13,56 MHz.
2. (Kurnia Dwi Artika, 2013) dengan judul Rancang Bangun Sistem Pengaman Pada Sepeda Motor Dengan Memanfaatkan Sensor Encoder Dan Sensor Ping Rancang Bangun Sistem Pengaman Pada Sepeda Motor Dengan memanfaatkan Sensor Encoder Dan Sensor Ping. Sensor encoder (jarak) dapat bekerja untuk membunyikan alarm pada saat jumlah ketukan 35 titik (lubang cakram) atau sejauh 4,92 meter. Sensor ping (tinggi) dapat bekerja untuk membunyikan alarm pada *counter* data 1040 atau ketinggian 0,4 meter dari permukaan/jalan.
3. (Nurdin Bagenda & Indra, 2014) dengan judul Prototipe sistem keamanan dan pengendalian Kendaraan Roda Empat berbasis mikrokontroler atmega8535. Sistem kerja dari alat ini yaitu Dengan penggunaan modem Wavecom fastrack

M1306B berfungsi mengirim informasi sepeda motor melalui SMS kepada pemilik kendaraan sehingga pemilik dapat dengan mudah mendapat informasi. Selain itu system kamanan ini menggunakan sensor ultrasonic sebagai pendeteksi stang kendaraan, sersor getar untuk mendeteksi getaran kendaraan. Mikrokontroler atmega8535 sebagai pusat pengontrol atau pengedali dari sensor- sensor yang digunakan. Code vision AVR sebagai perangkat lunak media penghubung antara program yang akan diisikan ke mikrokontroler atmega8535 dengan menggunakan bahasa pemograman C.

4. (Rizky Muhammad Syafii, 2017) dengan judul Desain Dan Implementasi Sistem Keamanan Locker Menggunakan E-Ktp Berbasis Arduino Pro Mini hasil pengujian dar alat ini yaitu sistem ini mampumerekam lebih dari 2 pengguna locker yaitu sebanyak 15 pengguna locker dengan jarak baca efektif untuk tag jenis key chain dengan tingkat keberhasilan 100% jarak bacanya= 2,5 cm dan e-KTP jarak baca efektif dengan tingkat keberhasilan 100% berjarak = 3,5 cm, untuk jenis tag RFID white card dengan tingkat keberhasilan 100% jarak baca = 5 cm, tag white card merupakan jarak baca paling terjauh diantara e-KTP dan key chain sesuai dengan data sheet reader RFID RC522.
5. (Ariesta Adhitama Satya Negara, Ufi Najib, Jenny Putri Hapsari, 2016) dengan judul Pemanfaatan E-Ktp Untuk Pengaktifan Sepeda Motor Berbasis Arduino Uno sistem kerja dari alat ini yaitu dengan memanfaatkan E-KTP untuk pengaktifan sepeda motor ini menggunakan Arduino UNO sebagai sistem kendali dan RFID untuk alat scanning kartu. Alat ini akan bekerja untuk menggantikan kunci konvensional yang digunakan sepeda motor selama ini. Pertama, alat akan dihidup dengan mengambil sumber tegangan dari akumulator sehingga lampu indikator menyala. Kedua, E-KTP yang sudah terkonfigurasi discan dan akan muncul suara dari buzzer yang mengidikasikan bahwa motor siap untuk diaktifkan. EKTP dapat discan dengan baik bila terdapat pada jarak kurang lebih 10 cm.

## **2.2 Dasar Teori**



### **2.2.1 Sistem**

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energy untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu setentitas yang berinteraksi, dimana suatu model matematika seringkali biasa dibuat. Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara yang berperan sebagai penggerakya yaitu rakyat yang berada dinegara tersebut. Kata "sistem" banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka (Sidarta, 2016).

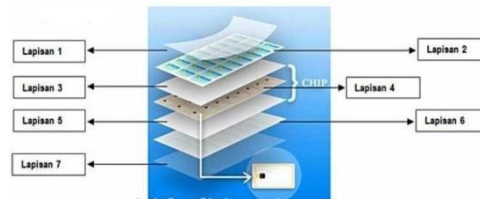
### **2.2.2 Pengertian Keamanan**

Keamanan adalah keadaan bebas dari bahaya. Istilah ini bisa digunakan dengan hubungan kepada kejahatan, segala bentuk kecelakaan, dan lain-lain. Keamanan merupakan topik yang luas termasuk keamanan nasional terhadap serangan teroris, keamanan komputer terhadap hacker atau *cracker*, keamanan rumah terhadap maling dan penyusup lainnya, keamanan finansial terhadap kehancuran ekonomi dan banyak situasi berhubungan lainnya. Kebutuhan dasar manusia prioritas kedua berdasarkan kebutuhan fisiologis dalam hirarki *Maslow* yang harus terpenuhi selama hidupnya, sebab dengan terpenuhinya rasa aman setiap individu dapat berkarya dengan optimal dalam hidupnya. Mencari lingkungan yang betul-betul aman memang sulit, maka konsekuensinya promosi keamanan berupa kesadaran dan penjagaan adalah hal yang penting. Keamanan fisik (*Biologic safety*) merupakan keadaan fisik

yang aman terbebas dari ancaman kecelakaan dan cedera (*injury*) baik secara mekanis, thermis, elektrik maupun bakteriologis. Kebutuhan keamanan fisik merupakan kebutuhan untuk melindungi diri dari bahaya yang mengancam kesehatan fisik, yang pada pembahasan ini akan difokuskan pada providing for safety atau memberikan lingkungan yang aman . (Sutris, 2015).

### 2.2.3 Kelebihan e-KTP Sebagai Tag Pasif

Bahan fisik *chip* yang tipis seperti kertas didominasi oleh silikon dan jenis plastik, tidak tahan panas, korosi, basah atau lembab. *Chip e-KTP* menggunakan antar muka nirsentuh (*contactless*) yang memenuhi standar ISO 14443 A/B. Transmisi data melalui gelombang radio. Blangko *e-KTP* terbuat dari bahan PETG, semacam polimer termoplastik, yang tersusun dalam 7 lapisan.



**Gambar 2.1 Lapisan e-KTP**

Bahan fisik *chip* yang tipis seperti kertas didominasi oleh silikon dan jenis plastik, tidak tahan panas, korosi, basah atau lembab serta dapat rusak akibat patah, sobek dan jenis pengrusakan fisik lainnya. *e-KTP* sendiri secara mekanisme teknis memiliki keuntungan:

1. *Chip e-KTP* dilindungi, salah satunya, dengan mekanisme autentikasi dua arah, yaitu suatu mekanisme untuk saling mengenali antara *chip e-KTP* dengan *reader* RFID, di mana *chip* harus dapat mengenali *reader* RFID (arah 1) dan *reader* RFID harus dapat mengenali *chip* (arah 2), setelah melalui mekanisme

otentikasi ini maka data yang tersimpan di dalam chip baru dapat dibaca oleh *reader* RFID.

2. *Reader* RFID harus menghasilkan medan radio frekuensi tinggi untuk memberikan pasokan daya yang sesuai dengan kebutuhan *chip e-KTP*, di mana medan radio tersebut akan dimodulasikan untuk keperluan komunikasi.

3. Kisaran dari besar medan magnet frekuensi radio yang dihasilkan oleh *reader* RFID adalah mengikuti ketentuan dalam ISO/IEC 14443, yaitu antara 1,5 A/m sampai dengan 7,5 A/m. Sedangkan besar frekuensi dari modulasi amplitudo medan magnet tersebut, yang digunakan untuk mengirimkan data ke *chip e-KTP*, adalah 13,56 MHz[6].

4. *Chip* yang tertanam dalam kartu ini memungkinkan melakukan berbagai proses komputasi yang tidak dapat dilakukan oleh kartu berbasis *magnetic stripe*. Dengan kemampuan ini, kartu *chip* dapat menjalankan berbagai algoritma dan protokol keamanan yang cukup kompleks.\

#### **2.2.4 Kendaraan**

Kendaraan adalah kendaraan yang digerakkan oleh peralatan teknik untuk pergerakannya, dan digunakan untuk transportasi darat. Umumnya kendaraan menggunakan mesin pembakaran dalam, namun mesin listrik dan mesin lainnya juga dapat digunakan. Kendaraan bermotor memiliki roda, dan biasanya berjalan diatas jalanan. Berdasarkan UU No. 14 tahun 1992 yang dimaksud dengan peralatan teknik dapat berupa kendaraan atau peralatan lainnya yang berfungsi untuk mengubah suatu sumber daya energi tertentu menjadi tenaga gerak kendaraan yang bersangkutan. Pengertian kata berada dalam ketentuan ini adalah terpasang pada tempat sesuai dengan fungsinya. Termasuk dalam pengertian kendaraan adalah kereta gandongan atau kereta tempelan yang dirangkaikan dengan kendaraan sebagai penariknya

#### **2.3 Perangkat Keras Yang Digunakan**

### 2.3.1 Aki (Accu)

Aki atau *accu* adalah suatu proses kimia listrik, dimana pada saat pengisian, cas, *charge* energi listrik diubah menjadi kimia dan saat pengeluaran (*discharge*) energi kimia diubah menjadi energi listrik. Aki terdiri dari sel-sel dimana tiap sel memiliki tegangan sebesar 2V, artinya aki mobil dan aki motor yang memiliki tegangan 12V terdiri dari 6 sel yang dipasang secara seri ( $12V = 6 \times 2V$ ) sedangkan aki yang memiliki teganga 6V memiliki 3 sel yang dipasang secara seri ( $6V = 3 \times 2V$ ).



**Gambar 2.2 Simulasi dalam Aki dan Contoh Aki**

(a) Aki 12Volt dan Aki 6Volt, (b) Aki Basah dan Kering

(<http://www.wartasaranamedia.com/2014/06/bagian-komponen-utama-aki-kering-dan-aki-basah-197.html>)

Antara satu sel dengan sel lainnya dipisahkan oleh dinding penyekat yang terdapat dalam bak aki, artinya tiap ruang pada sel tidak berhubungan karena itu cairan elektrolit pada tiap sel juga tidak berhubungan (dinding pemisah antar sel tidak boleh ada yang bocor atau merembes).

Sebuah aki kering 12V dan 18Ah mencantumkan nilai spesifikasi sebagai berikut

1.  $20\text{hr} \times 0,9\text{A} = 18\text{A}$

2.  $5\text{hr} \times 3,06\text{A} = 15,3\text{Ah}$  c.  $1\text{hr} \times 10,8\text{A} = 10,8\text{Ah}$  d.  $1/2\text{hr} \times 18\text{A} = 9\text{Ah}$

Jika dilihat dari spesifikasi maka aki ini memiliki kapasitas efektif sebesar 18Ah namun suplai dari aki sebenarnya hanya bisa dilakukan selama :

1. 20 jam jika kuat arus yang dipakai hanya sebesar 0,9A untuk tiap jam artinya hanya memakai daya sebesar 10,8Watt/jam ( $12\text{V} \times 0,9\text{A}$ ). Kapasitas = 18Ah ( $0,9\text{A} \times 20\text{hour}$ )
2. 5 jam jika kuat arus yang dipakai 3,06A atau berdaya 36,72Watt/jam ( $12\text{V} \times 3,06\text{A}$ ). Kapasitas = 15,3Ah ( $3,06\text{A} \times 5\text{hour}$ ).
3. 1 jam jika kuat arus yang dipakai 10,8A atau berdaya 129,6Watt/jam ( $12\text{V} \times 10,8\text{A}$ ). Kapasitas = 10,8Ah ( $10,8\text{A} \times 1\text{hour}$ )
4. 1/2 jam jika kuat arus yang dipakai sama dengan kapasitas efektifnya yang 18 Ah atau berdaya 216 Watt/jam ( $12\text{V} \times 18\text{A}$ ). Kapasitas = 9 Ah ( $18\text{A} \times 0,5\text{hour}$ ).

### **2.3.2 Radio Frequency Identification (RFID)**

*Radio Frequency Identification (RFID)* adalah teknologi identifikasi yang fleksibel, mudah digunakan, dan sangat cocok untuk operasi otomatis. *RFID* mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. *RFID* dapat disediakan dalam device yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka *RFID* dapat menyediakan tingkat keamanan yang tinggi.

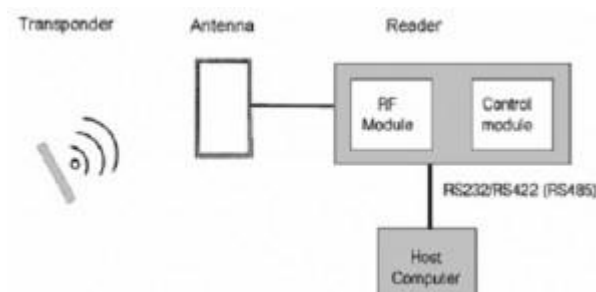
Pada sistem *RFID* umumnya, tag atau transponder ditempelkan pada suatu objek. Setiap tag dapat membawa informasi yang unik, di antaranya: serial number, model, warna, tempat perakitan, dan data lain dari objek tersebut. Ketika tag ini melalui

medan yang dihasilkan oleh pembaca *RFID* yang kompatibel, tag akan mentransmisikan informasi yang ada pada tag kepada pembaca *RFID*, sehingga proses identifikasi objek dapat dilakukan.

Sistem *RFID* terdiri dari empat komponen, di antaranya sebagai berikut :

1. Tag: Ini adalah *device* yang menyimpan informasi untuk identifikasi objek. Tag *RFID* sering juga disebut sebagai transponder.
2. Antena: untuk mentransmisikan sinyal frekuensi radio antara pembaca *RFID* dengan tag *RFID*. Pembaca *RFID*: adalah device yang kompatibel dengan tag *RFID* yang akan berkomunikasi secara wireless dengan tag.
3. Software Aplikasi: adalah aplikasi pada sebuah workstation atau PC yang dapat membaca data dari tag melalui pembaca *RFID*. Baik tag dan pembaca *RFID* dilengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

### 2.3.3 Sistem *RFID*



**Gambar 2.3. Sistem *RFID***

(Sumber <https://www.elektronikar,2014>)

### 2.3.3.1 Pembaca *RFID*

Sebuah pembaca *RFID* harus menyelesaikan dua buah tugas, yaitu:

1. Menerima perintah dari software aplikasi
2. Berkomunikasi dengan tag *RFID*

Pembaca *RFID* adalah merupakan penghubung antara *software* aplikasi dengan antena yang akan meradiasikan gelombang radio ke tag *RFID*. Gelombang radio yang diemisikan oleh antena berpropagasi pada ruangan di sekitarnya. Akibatnya data dapat berpindah secara wireless ke tag *RFID* yang berada berdekatan dengan antena.

### 2.3.3.2 Tag *RFID*

Tag *RFID* adalah device yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari tag *RFID* umumnya memiliki memori sehingga tag ini mempunyai kemampuan untuk menyimpan data. Memori pada tag secara dibagi menjadi sel-sel. Beberapa sel menyimpan data *Read Only*, misalnya serial number yang unik yang disimpan pada saat tag tersebut diproduksi. Sel lain pada *RFID* mungkin juga dapat ditulis dan dibaca secara berulang.

Berdasarkan catu daya tag, tag *RFID* dapat digolongkan menjadi:

1. **Tag Aktif:** yaitu tag yang catu dayanya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh pembaca *RFID* dan tag dapat mengirimkan informasi dalam jarak yang lebih jauh. Kelemahan dari tipe tag ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh tag *RFID* maka rangkaiannya akan semakin kompleks dan ukurannya akan semakin besar.
2. **Tag Pasif:** yaitu tag yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca *RFID*. Rangkaiannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah tag hanya dapat

mengirimkan informasi dalam jarak yang dekat dan pembaca *RFID* harus menyediakan daya tambahan untuk tag *RFID*.

Tag *RFID* telah sering dipertimbangkan untuk digunakan sebagai barcode pada masa yang akan datang. Pembacaan informasi pada tag *RFID* tidak memerlukan kontak sama sekali. Karena kemampuan rangkaian terintegrasi yang modern, maka tag *RFID* dapat menyimpan jauh lebih banyak informasi dibandingkan dengan *barcode*. Fitur pembacaan jamak pada teknologi *RFID* sering disebut sebagai *anti collision*.

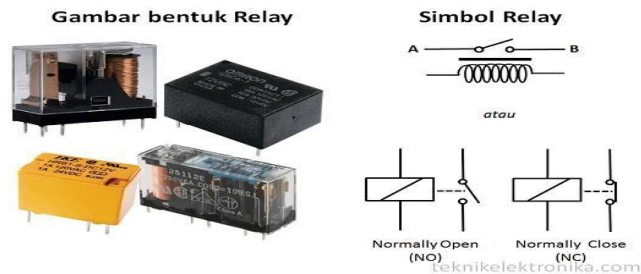
#### **2.3.4 Relay**

*Relay* merupakan bentuk hambatan terdiri atas titik-titik kontak bawah dengan gulungan *spool*-nya tidak bergerak dan titik kontak bagian atas yang bergerak. Prinsip kerja hambatan adalah menghubungkan titik-titik kontak bagian bawah dengan titik bagian atas yaitu terletak gulungan *spool* dialiri arus listrik yang timbul elektromagnet. (Handy Wicaksono,1996,1-12). *Relay* merupakan bentuk hambatan terdiri atas titik-titik kontak bawah dengan gulungan *spool*-nya tidak bergerak dan titik kontak bagian atas yang bergerak. Prinsip kerja hambatan adalah menghubungkan titik-titik kontak bagian bawah dengan titik bagian atas yaitu terletak gulungan *spool* dialiri arus listrik yang timbul elektromagnet. (Handy Wicaksono,1996,1-12). Secara sederhana relay elektromekanis ini didefinisikan sebagai berikut :

1. Alat yang menggunakan gaya elektromagnetik untuk menutup (atau membuka) kontak saklar.
2. Saklar yang digerakkan (secara mekanis) oleh daya/energi listrik.

Dibawah ini adalah gambar fisik, bentuk dan Simbol Relay yang sering ditemukan di Rangkaian Elektronika.





**Gambar 2.4 Gambar dan Simbol Relay**

(<http://teknikelektronika.com/pengertian-relay-fungsi-relay/>)



**Gambar 2.5 Relay**

(Sumber : Kilian, Christopher T, Modern Control Technology, (West Published Co : 1996)

Bagian titik kontak dibagi menjadi 2 bagian yaitu bagian kontak utama dan kontak bantu yaitu : Bagian kontak utama gunanya untuk menghubungkan dan memutuskan arus listrik bagian yang menuju beban/pemakai. Bagian kontak bantu gunanya untuk menghubungkan dan memutuskan arus listrik ke bagian yang menuju bagian pengendali. Kontak Bantu mempunyai 2 kontak yaitu kontak hubung (NC) dan kontak putus (NO) menandakan masing-masing kontak dan gulungan spool. Secara umum, relay digunakan untuk memenuhi fungsi–fungsi berikut :

1. Remote control : dapat menyalakan atau mematikan alat dari jarak jauh.
2. Penguatan daya : menguatkan arus atau tegangan.
3. Pengatur logika kontrol suatu sistem. Susunan kontak pada relay adalah:
4. Normally Open : Relay akan menutup bila dialiri arus listrik.
5. Normally Close : Relay akan membuka bila dialiri arus listrik.

6. Changeover : Relay ini memiliki kontak tengah yang akan melepaskan diri dan membuat kontak lainnya berhubungan.

### 2.3.4.1 Prinsip Kerja Relay

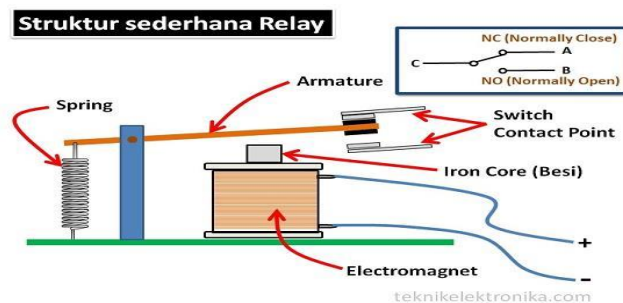
Pada dasarnya, Relay terdiri dari 4 komponen dasar yaitu :

1. Electromagnet (Coil)
2. Armature
3. Switch Contact Point (Saklar)
4. Spring

Seperti saklar, relay juga dibedakan berdasar pole dan throw yang dimilikinya.

1. Pole : banyaknya contact yang dimiliki oleh relay
2. Throw : banyaknya kondisi (state) yang mungkin dimiliki contact.

Berikut ini merupakan gambar dari bagian-bagian Relay :



**Gambar 2.6 Struktur Sederhana Relay**

(<http://teknikelektronika.com/pengertian-relay-fungsi-relay/>)

Kontak normally open akan membuka ketika tidak ada arus mengalir pada kumparan, tetapi tertutup secepatnya setelah kumparan menghantarkan arus atau diberi tenaga. Kontak normally close akan tertutup apabila kumparan tidak diberi tenaga dan membuka ketika kumparan diberi daya. Masing-masing kontak biasanya digambarkan sebagai kontak yang tampak dengan kumparan tidak diberi tenaga atau daya .

Relay terdiri dari 2 terminal trigger, 1 terminal input dan 1 terminal output.

1. Terminal trigger : yaitu terminal yang akan mengaktifkan relay, seperti alat elektronik lainnya relay akan aktif apabila di aliri arus + dan arus -. Pada contoh relay yang kita gunakan terminal trigger ini adalah 85 dan 86.
2. Terminal input : yaitu terminal tempat kita memberikan masukan, pada contoh adalah terminal 30.
3. Terminal output : yaitu tempat keluarnya output pada contoh adalah terminal 87.

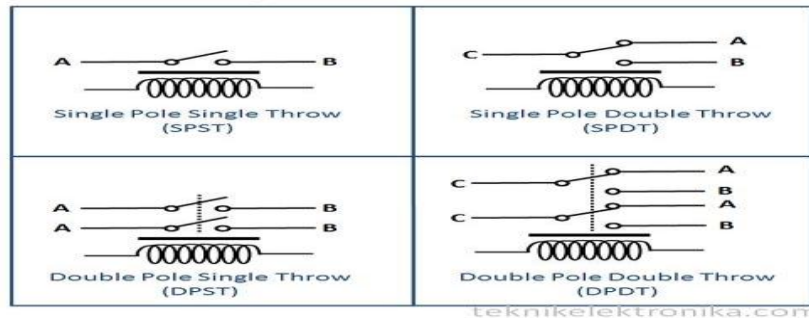
#### **2.3.4.2 Jenis-jenis Relay**

Berikut ini penggolongan relay berdasar jumlah pole dan throw :

1. DPST (Double Pole Single Throw), relay golongan ini memiliki 6 terminal, diantaranya 4 terminal yang terdiri dari 2 pasang terminal saklar sedangkan
2. terminal lainnya untuk *coil*. Relay DPST dapat dijadikan 2 saklar yang dikendalikan oleh 1 *coil*.
  - A. *SPST (Single Pole Single Throw)*, relay golongan ini memiliki 4 terminal, 2 terminal untuk saklar dan 2 terminalnya lagi untuk *coil*.
  - B. *SPDT (Single Pole Double Throw)*, relay golongan ini memiliki 5 terminal, 3 terminal untuk saklar dan 2 terminalnya lagi untuk *coil*.
  - C. *DPDT (Double Pole Double Throw)*, relay golongan ini memiliki terminal sebanyak 8 terminal, diantaranya terminal yang merupakan 2 pasang *relay SPDT* yang dikendalikan oleh 1 (*single*) *coil*. Sedangkan 2 terminal lainnya untuk *coil*.

Selain golongan *relay* diatas, terdapat juga *relay-relay* yang *Pole* dan *Throw*-nya melebihi dari dua. Misal-nya 3PDT (*Triple Pole Double Throw*) ataupun 4PDT (*Four Pole Double Throw*) dan lain sebagainya.

Berikut ini merupakan gambar dari jenis *Relay* berdasarkan *Pole* dan *Throw*-nya :



**Gambar 2.7** Jenis Relay berdasarkan *Pole* dan *Throw*

(<http://teknikelektronika.com/pengertian-relay-fungsi-relay/>)

### 2.3.4.3 Fungsi-Fungsi *Relay*

Beberapa fungsi *relay* yang telah umum diaplikasikan kedalam peralatan elektronika diantaranya adalah :

1. *Relay* digunakan untuk menjalankan fungsi logika (*logic function*).
2. *Relay* digunakan untuk memberikan fungsi penundaan waktu (*time delay function*).
3. *Relay* digunakan untuk mengendalikan sirkuit tegangan tinggi dengan bantuan dari signal tegangan rendah.
4. Ada juga *relay* yang berfungsi untuk melindungi motor ataupun komponen lainnya dari kelebihan tegangan ataupun hubung singkat (*short*).

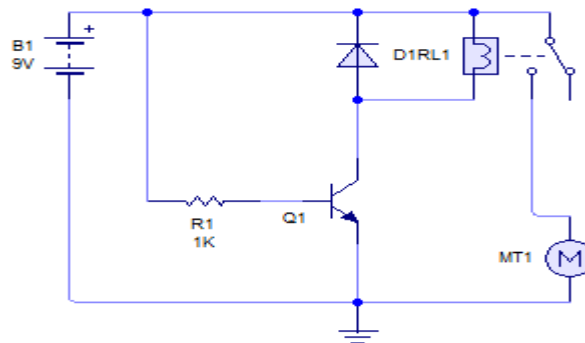
### 2.3.4.4 Driver *Relay*

Rangkaian *driver relay* berfungsi untuk mengendalikan motor arus searah (dc) yang dihasilkan dari *port* paralel I/O. Sinyal dari keluaran port biasanya berupa sinyal-sinyal yang kecil, sehingga tidak mampu untuk menggerakkan sistem daya berupa motor arus searah. Untuk dapat dimanfaatkan sinyal keluaran port, diperlukan suatu rangkaian *driver relay* agar sinyal yang kecil dapat dipergunakan untuk penggerak objek yang akan dikendalikan dari jarak jauh. Rangkaian *driver relay* ini dibangun oleh suatu komponen utama yaitu transistor dan *relay*. Transistor di

rangkaian *driver relay* difungsikan sebagai penguat sinyal dan *switching*, serta *relay* sebagai penggerak motor dc. *Driver relay* ini selain sebagai penguat dan *switching*, sekaligus difungsikan untuk mengendalikan motor dc dalam sistem pembalik putaran. Jadi, *driver relay* ini dapat mengatur arah putaran motor *forward* dan *reverse*. Semua *driver relay* pada sistem ini memiliki rangkaian dan karakteristik yang sama. Saat *relay 1* bekerja maka posisi positif motor akan mendapat sumber tegangan positif dan posisi negatif motor terhubung dengan kutub negatif sumber tegangan. Sehingga, motor akan berputar dengan arah putaran searah jarum jam (*clockwise*). Dengan cara yang sama untuk menggerakkan kontak *relay 2*, maka terjadi kondisi yang berkebalikan yaitu motor akan berputar dengan arah putaran yang berlawanan arah jarum jam (*counter clockwise*).

Penggunaan *driver relay* ini menjadi pilihan karena *driver relay* mudah dikontrol, dapat diberi beban yang besar baik beban AC maupun DC serta sebagai isolator yang baik antara rangkaian beban dengan rangkaian kendali. Rangkaian *driver relay* dapat dibangun menggunakan konsep transistor sebagai saklar. Teknik antara *relay* dengan rangkaian digital atau mikrokontroler adalah rangkaian *driver relay* dengan menggunakan transistor sebagai penguat.

Berikut merupakan contoh dari gambar rangkaian *Driver Relay* :



**Gambar 2.8 Rangkaian *Driver Relay***

(<http://teknikelektronika.com/pengertian-relay-fungsi-relay/>)

Pada rangkaian menyerupai sirkuit diatas, dapat dilihat untuk mengoperasikan

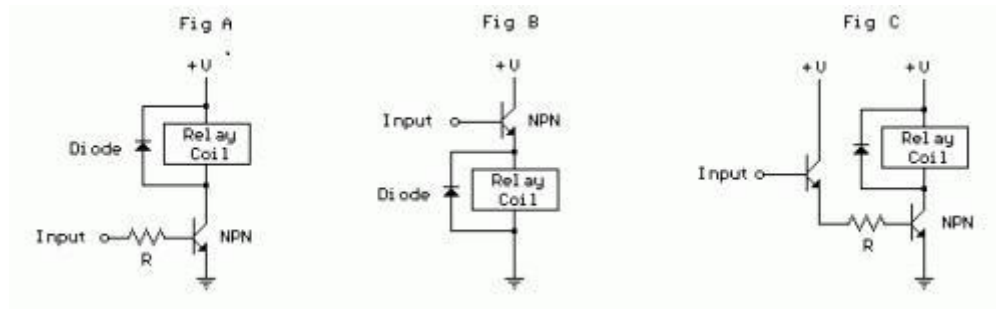
transistor sebagai saklar transistor dalam keadaan sepenuhnya "OFF" (*cut-off*) atau dalam keadaan "ON" (saturasi).. Namun, ketika dinyalakan dalam kondisi ON (saturasi) , maka aliran arus maksimum. Dalam prakteknya ketika transistor diaktifkan "OFF", arus kebocoran akan kecil ketika mengalir melalui transistor dan ketika diaktifkan "ON" maka rangkaian tersebut akan memiliki tegangan saturasi kecil ( $V_{CE}$ ) Meskipun transistor tidak dalam saklar yang sempurna, baik di *cut-off* dan daerah saturasi. Agar arus Basis mengalir, terminal input Basis harus dibuat lebih positif daripada Emitter dengan meningkatkan itu di atas 0,7 volt yang dibutuhkan untuk perangkat silikon. Dengan memvariasikan *Base-Emitter* ini tegangan  $V_{BE}$  arus basis juga mengontrol jumlah arus kolektor yang mengalir melalui transistor.

Ketika arus kolektor maksimum mengalir maka transistor dikatakan saturasi. Nilai dari resistor Basis menentukan berapa banyak masukan tegangan yang diperlukan dan sesuai saat Basis untuk beralih transistor sepenuhnya "ON".

Transistor BC108 adalah transistor umum NPN *bipolar junction* (BJT) digunakan untuk memperkuat daya rendah atau aplikasi *switching*. Hal ini dirancang untuk arus yang rendah sampai medium, daya yang rendah , tegangan medium, dan dapat beroperasi pada kecepatan yang cukup tinggi.

#### **2.3.4.5 Interface Driver Relay**

Penggunaan *relay* sering menjadi pilihan karena *relay* mudah dikontrol, *relay* dapat diberi beban yang besar baik beban AC maupun DC, dan sebagai isolator yang baik antara rangkaian beban dengan rangkaian kendali. Rangkaian *interface relay* dapat dibangun menggunakan konsep transistor sebagai saklar. Transistor yang digunakan untuk *driver relay* dapat dikonfigurasi dengan *common emitter*, *emitter follower* atau transistor *darlington*. Teknik *interface* antara *relay* dengan rangkaian digital atau rangkaian *microcontroller* dapat dilihat pada gambar dibawah ini :



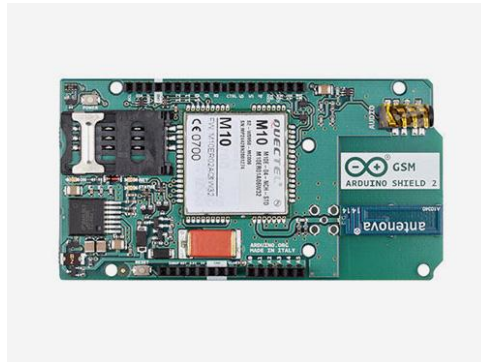
**Gambar 2.9 Rangkaian Interface Driver Relay**

(<http://elektronika-dasar.web.id/teori-elektronika/interface-relay-ke-rangkaian-digital>)

Rangkaian *inteface* antar *relay* dengan rangkaian digital pada gambar diatas ada 3 jenis interface yang dapat digunakan. Bagian dan fungsi komponen dari rangkaian *interface relay* diatas sebagai berikut :

1. Rangkaian pada gambar A, rangkaian pada gambar A tersebut menggunakan mode *common emitor*, apabila basis mendapat sinyal *input* logika 1 (sumber tegangan positif) maka transistor pada gambar A akan mendapat bias maju, sehingga transistor ON dan memberikan sumber tegangan ke *relay* dan *relay* menjadi ON.
2. Rangkaian pada gambar B adalah *interface relay* yang menggunakan transistor teknik *emior folower* dimana *relay* diletakan pada kaki emitor trnasistor. Fungsi dioda yang dipasangkan pada rangkaian *interface* tersebut digunakan untuk menyerap tegangan induksi yang dihasilkan oleh *relay*.
3. Rangkaian pada gambar C merupakan teknik *inteface relay* ke rangkaian digital menggunakan transistor yang dirangkai secara *darlington*.

### 2.3.5 *Internet Shield SIM900*



**Gambar 2.10 Internet Shield (GSM)**

(Sumber [https:// www.elektronika.cc.com](https://www.elektronika.cc.com))

Arduino *GSM Shield V2* menghubungkan Arduino Anda ke internet menggunakan jaringan nirkabel GPRS. Cukup colokkan modul ini ke papan Arduino Anda, pasang kartu SIM dari operator yang menawarkan jangkauan GPRS dan ikuti beberapa petunjuk sederhana untuk mulai mengendalikan dunia Anda melalui internet. Anda juga dapat membuat / menerima panggilan suara menggunakan jack audio / mikrofon on-board dan mengirim / menerima pesan SMS. Arduino *GSM Shield 2* memungkinkan dewan Arduino terhubung ke internet, membuat / menerima panggilan suara dan mengirim / menerima pesan SMS. Perisai menggunakan modem radio M10 oleh Quectel. Hal ini dimungkinkan untuk berkomunikasi dengan board menggunakan perintah AT. Perpustakaan *GSM* memiliki sejumlah besar metode untuk komunikasi dengan perisai.

Perisai menggunakan pin digital 2 dan 3 untuk komunikasi serial perangkat lunak dengan M10. Pin 2 terhubung ke pin TX M10 dan pin 3 ke pin RX-nya. Lihat catatan ini untuk bekerja dengan Arduino Uno, Uno ADK, atau Leonardo. Pin PWRKEY modem terhubung ke pin Arduino 7.

M10 adalah *modem Quad-band GSM / GPRS* yang bekerja pada frekuensi GSM850MHz, GSM900MHz, DCS1800MHz dan PCS1900MHz. Ini mendukung



protokol TCP / UDP dan HTTP melalui koneksi GPRS. Kecepatan downlink data GPRS dan kecepatan transfer uplink maksimal adalah 85,6 kbps.

Untuk antarmuka dengan jaringan selular, board memerlukan kartu SIM yang disediakan oleh operator jaringan. Lihat halaman awal untuk informasi tambahan tentang penggunaan SIM.

### **2.3.6 SMS (*Short Message Service*)**

Teknologi telekomunikasi pada saat ini semakin berkembang, salah satu teknologi telekomunikasi yang sedang berkembang yaitu *Short Message Service* atau biasa nya disebut dengan SMS. SMS adalah kemampuan untuk mengirim dan menerima pesan singkat dalam bentuk teks dari sebuah perangkat nirkabel, yaitu perangkat telekomunikasi telpon seluler, dalam hal ini perangkat nirkabel yang digunakan adalah telpon *seluler*. Teks tersebut bisa terdiri dari kata-kata atau nomor ataupun kombinasi *alphanumeric*. Pendapat lain mengenai pengertian SMS di utarakan oleh Romzi Imron (Romzi Imron 2004:1) yang mengungkapkan tentang pengertian SMS adalah sebagai berikut:

“Layanan yang banyak di aplikasikan pada jaringan komunikasi tanpa kabel yang memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric* antar terminal pelanggan (Ponsel) atau antara terminal pelanggan dengan sistem eksternal seperti *e-mail, paging, voice mail* dan sebagainya” (Imron,2004:1)

### **2.3.7 Mikrokontroller**

*Mikrocontroller* adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program pada umumnya terdiri dari CPU (*Central Processing Unit*), memori, I/O tertentu dan unit pendukung seperti *Analog-to-Digital Converter* (ADC) yang sudah terintegrasi di dalamnya. Kelebihan utama dari *Mikrokontroller* ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran board *Mikrokontroller* menjadi sangat ringkas. (Arduino, 2016)

#### **2.3.7.1 Modul Arduino Nano**

Arduino nano dapat berupa sebuah mikroprosesor single board yang bersifat open source yang dirancang untuk memudahkan sebuah aplikasi pengguna elektronik dalam merancang sebuah alat, arduino ini bertujuan agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan, Berikut ini gambar arduino dari tampak depan dan belakang, disebut sebagai papan pengembangan karena board ini memang berfungsi sebagai arena prototype sirkuit mikrokontroller. dengan menggunakan papan pengembangan akan lebih mudah dalam merangkai rangkaian elektronika mikrokontroller ATmega328 (Slamet Haryono, 2016). Mengapa prototype ini menggunakan Arduino nano? Apa bedanya dengan arduino Nano? terdapat perbedaan pada sebuah tegangan input pada kaki (VIN) yang dilengkapi dengan Jack (DC) dan sedangkan arduino nano tidak, karena arduino nano dapat menggunakan catu daya langsung dari *mini-USB port* dari luar. Adapun kelebihan dari arduino nano tidak perlu lagi menggunakan kabel USB ASP tapi cukup dengan menggunakan kabel USB Mini. adapun system kerja dari arduino nano ini adalah dengan menggunakan pin analog di papan arduino yang menggunakan besaran tegangan sebesar 5V. Gambar 2.1 arduino nano atmega328.

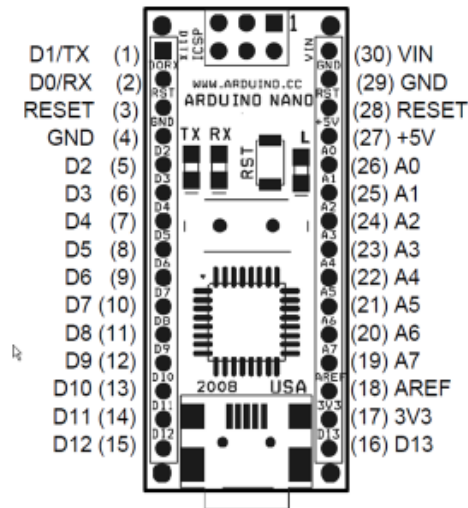


**Gambar 2.11. Arduino Nano Atmega328**

(Sumber <https://www.arduino.com>,2016)

### **2.3.7.2 Blok Arduino Nano**

Dengan mengambil sebuah contoh blok diagram arduino nano, dapat dijelaskan bagian bagian kaki pin arduino nano sebagai berikut. Dapat dilihat seperti gambar 2.12.



**Gambar 2.12. Blok Diagram Arduino Nano**

(Sumber <https://www.arduino.com>,2016)

1. **Serial 0 (RX) dan 1 (TX).** Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini terhubung ke pin yang sesuai dari chip FTDI USB-to-TTL Serial.
2. **External Interrupt (Interupsi Eksternal):** Pin 2 dan pin 3 ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.
3. **PWM :** Pin 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi `analogWrite()`.
4. **SPI :** Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI. Sebenarnya komunikasi SPI ini tersedia pada hardware, tapi untuk saat belum didukung dalam bahasa Arduino.

5. **LED** : Pin 13. Tersedia secara built-in pada papan Arduino Nano. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala, dan ketika pin diset bernilai LOW, maka LED padam.
6. **I2C** : Pin A4 (SDA) dan pin A5 (SCL). Yang mendukung komunikasi I2C (TWI) menggunakan perpustakaan Wire.
7. **AREF** : Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.
8. **RESET** : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino

## 2.4 Perangkat Lunak Yang Digunakan

Pengertian perangkat lunak atau biasa disebut software adalah sekumpulan data elektronik yang sengaja disimpan dan diatur oleh komputer berupa program ataupun instruksi yang akan menjalankan sebuah perintah. Perangkat lunak atau software disebut juga sebagai penerjemah perintah-perintah yang dijalankan oleh user untuk diteruskan dan diproses oleh perangkat keras (hardware). Dengan adanya perangkat lunak inilah sebuah sistem mampu menjalankan perintah.

### 2.4.1 *Software* Mikrokontroler Arduino

*Software arduino* yang digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan arduino. *Integrated Development Environment* (IDE), suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau *sketsa* program untuk papan *Arduino*. IDE *arduino* merupakan *software* yang sangat canggih ditulis dengan menggunakan *java*. IDE arduino terdiri dari :

#### 1. Editor Program

Sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

## 2. Compiler

Berfungsi untuk kompilasi *sketch* tanpa unggah ke *board* bisa dipakai untuk pengecekan kesalahan kode *sintaks sketch*. Sebuah modul yang mengubah kode program menjadi kode *biner* bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*.

## 3. Uploader

Berfungsi untuk mengunggah hasil kompilasi *sketch* ke *board* target. Pesan *error* akan terlihat jika *board* belum terpasang atau alamat *port* COM belum terkonfigurasi dengan benar. Sebuah modul yang memuat kode *biner* dari komputer ke dalam *memory* didalam papan *arduino*.(Sumber: B.Gustomo, 2015).

### 2.4.1.1 Program Arduino Ide

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino Boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialise the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

**Gambar 2.13. Tampilan Program Arduino Nano**

Kode Program *Arduino* biasa disebut *sketch* dan dibuat menggunakan bahasa pemrograman C. Program atau *sketch* yang sudah selesai ditulis di *Arduino* IDE bisa langsung *dicompile* dan *diupload* ke *Arduino Board*. Secara sederhana, *sketch* dalam *Arduino* dikelompokkan menjadi 3 blok (lihat gambar di atas):

1. Header
2. Setup
3. Loop

#### 2.4.1.2 Header

Pada bagian ini biasanya ditulis definisi-definisi penting yang akan digunakan selanjutnya dalam program, misalnya penggunaan *library* dan pendefinisian *variable*. *Code* dalam blok ini dijalankan hanya sekali pada waktu compile. Di bawah ini contoh *code* untuk mendeklarasikan *variable led* (integer) dan sekaligus di isi dengan angka 13

```
int led = 13;
```

#### 2.4.1.3 Setup

Di sinilah awal program *Arduino* berjalan, yaitu di saat awal, atau ketika *power on Arduino board*. Biasanya di blok ini diisi penentuan apakah suatu pin digunakan sebagai *input* atau *output*, menggunakan perintah *pinMode*. Inisialisasi *variable* juga bisa dilakukan di blok ini

```
// the setup routine runs once when you press reset: void setup() { // initialize the digital pin as an output. pinMode(led, OUTPUT); }
```

OUTPUT adalah suatu makro yang sudah didefinisikan *Arduino* yang berarti = 1. Jadi perintah di atas sama dengan `pinMode(led, 1);`

Suatu pin bisa difungsikan sebagai *OUTPUT* atau *INPUT*. Jika difungsikan sebagai output, dia siap mengirimkan arus listrik (maksimum 100 mA) kepada beban yang disambungkannya. Jika difungsikan sebagai *INPUT*, pin tersebut memiliki *impedance* yang tinggi dan siap menerima arus yang dikirimkan kepadanya.

#### 2.4.1.4 Loop

Blok ini akan dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok. Program akan berhenti apabila tombol *power Arduino* di matikan. Di sinilah fungsi utama program *Arduino* kita berada.

```
void loop() {
```

```
digitalWrite(led, HIGH); // nyalakan LED delay(1000); // tunggu 1000 milidetik  
digitalWrite(led, LOW); // matikan LED delay(1000); // tunggu 1000 milidetik }
```

Perintah *digitalWrite(pinNumber,nilai)* akan memerintahkan arduino untuk menyalakan atau mematikan tegangan di *pinNumber* tergantung nilainya. Jadi perintah di atas *digitalWrite(led,HIGH)* akan membuat pin nomor 13 (karena di header dideklarasikan led = 13) memiliki tegangan = 5V (HIGH). Hanya ada dua kemungkinan nilai *digitalWrite* yaitu *HIGH* atau *LOW* yang sebetulnya adalah nilai integer 1 atau 0. Kalau sudah dibuat program di atas, selanjutnya kita ambil kabel USB yang diikutsertakan pada saat membeli *Arduino*, pasang ke komputer dan *board arduino*, dan *upload* programnya. Lampu LED yg ada di *Arduino* board kita akan kelap-kelip. Sekedar informasi, sebuah LED telah disediakan di *board Arduino Nano* dan disambungkan ke pin 13.

Selain blok *setup()* dan *loop()* di atas kita bisa mendefinisikan sendiri blok fungsi sesuai kebutuhan. Kita akan jumpai nanti pada saat pembahasan proyek. (Sumber: Septa Ajjie, 2016)

#### **2.4.2 Software ISIS & ARES Proteus 7.0**

Proteus adalah sebuah software untuk mendesain PCB yang juga dilengkapi dengan simulasi PSpice pada level skematik sebelum rangkaian skematik di-upgrade ke PCB sehingga sebelum PCBnya di cetak kita akan tahu apakah PCB yang akan kita cetak apakah sudah benar atau tidak. Proteus mampu mengkombinasikan program ISIS untuk membuat skematik desain rangkaian dengan program ARES untuk membuat layout PCB dari skematik yang kita buat. Software Proteus ini bagus digunakan untuk desain rangkaian mikrokontroler.



**Gambar 2.14. Tampilan Software ISIS & ARES Proteus**

(<http://teknikelektronika.com/2016>)

Proteus juga bagus untuk belajar elektronika seperti dasar-dasar elektronika sampai pada aplikasi pada mikrokontroler. Software Proteus ini menyediakan banyak contoh aplikasi desain yang disertakan pada instalasinya. Sehingga memungkinkan bisa belajar dari contoh-contoh yang sudah ada. Fitur- fitur yang terdapat dalam Proteus adalah sebagai berikut :

1. Memiliki kemampuan untuk mensimulasikan hasil rancangan baik digital maupun analog maupun gabungan keduanya.
2. Mendukung simulasi yang menarik dan simulasi secara grafis.
3. Mendukung simulasi berbagai jenis microcontroller seperti PIC 8051 series
4. Memiliki model-model peripheral yang interactive seperti LED, tampilan LCD, RS232, dan berbagai jenis library lainnya.
5. Mendukung instrument-instrument virtual seperti voltmeter, ammeter, oscilloscope, logic analyser, dan lain-lainnya.



6. Memiliki kemampuan menampilkan berbagai jenis analisis secara grafis seperti transient, frekuensi, noise, distorsi, AC dan DC, dan lain-lainnya.
7. Mendukung berbagai jenis komponen-komponen analog.
8. Mendukung open architecture sehingga kita bisa memasukkan program seperti C++
9. untuk keperluan simulasi.
10. Mendukung pembuatan PCB yang di-update secara langsung dari program ISIS ke program pembuat PCB-ARES.

ISIS dipergunakan untuk keperluan pendidikan dan perancangan. Beberapa fitur umum dari ISIS adalah sebagai berikut :

1. Windows dapat dioperasikan pada Windows 98/Me/2k/XP dan Windows terbaru.
2. Routing secara otomatis dan memiliki fasilitas penempatan dan penghapusan dot.
3. Sangat powerful untuk pemilihan komponen dan pemberian properties-nya.
4. Mendukung untuk perancangan berbagai jenis bus dan komponen-komponen pin, port modul dan jalur.
5. Memiliki fasilitas report terhadap kesalahan-kesalahan perancangan dan simulasi elektrik.
6. Mendukung fasilitas interkoneksi dengan program pembuat PCB-ARES.
7. Memiliki fasilitas untuk menambahkan package dari komponen yang belum didukung.

ARES (Advanced Routing and Editing Software) digunakan untuk membuat modul layout PCB. Adapun fitur-fitur dari ARES adalah sebagai berikut :

1. Memiliki database dengan tingkat keakuratan 32-bit dan memberikan resolusi sampai
2. 10 nm, resolusi angular 0,1 derajat dan ukuran maksimum board sampai 10 m.
3. ARES mendukung sampai 16 layer.

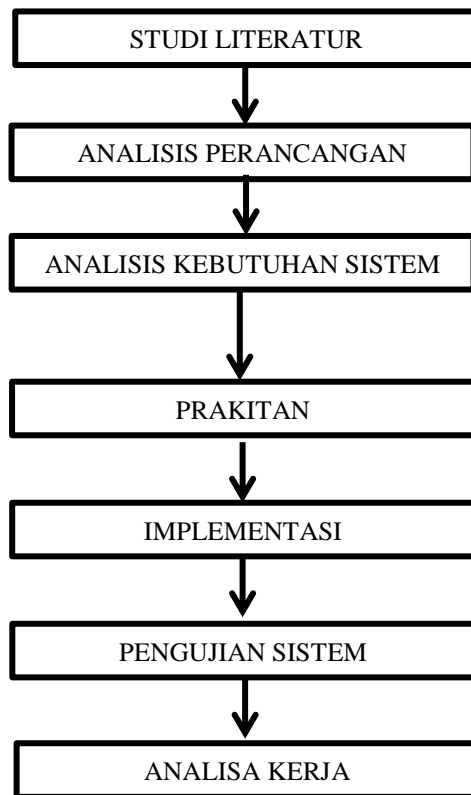
4. Terintegrasi dengan program pembuat skematik ISIS, dengan kemampuan untuk menentukan informasi routing pada skematik.
5. Visualisasi board 3-Dimensi.
6. Penggambaran 2-Dimensi dengan simbol library.

Proteus lebih memiliki kelebihan pada desainnya yang sederhana, sangat mudah dan bagus digunakan untuk perancangan rangkaian mikrokontroler yang akan sangat membantu digunakan oleh mahasiswa yang mengambil mata kuliah berhubungan dengan mikrokontroler. Kelebihannya yang lain adalah sebelum PCB dicetak skematiknya bisa disimulasikan dulu

## **BAB III**

### **METODE PENELITIAN**

Bab ini akan menjelaskan langkah-langkah penelitian yang akan dilakukan dalam Sistem Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler. Alur penelitian yang digunakan seperti pada gambar 3.1.



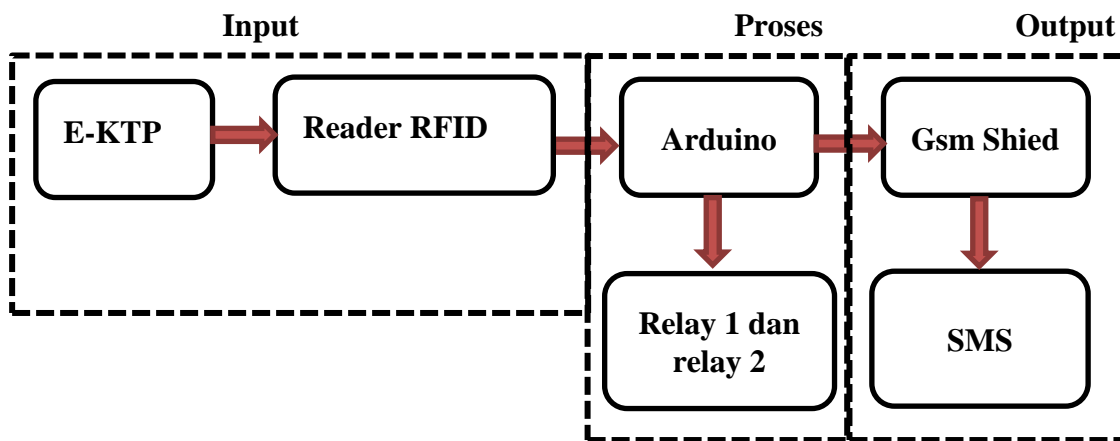
**Gambar 3.1. Alur Penelitian**

#### ***3.1 Studi Literatur***

Pada metode ini penulis mencari bahan penulisan tugas akhir yang diperoleh dari buku, jurnal dan website yang terkait dengan pembuatan Sistem Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler.

### 3.2 Analisa Perancangan Sistem

Perancangan sistem merupakan suatu hal yang dilakukan untuk mempermudah proses pembuatan alat. Konsep Sistem Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler digambarkan pada diagram blok dapat dilihat pada gambar 3.2 Blok diagram menjelaskan gambaran umum mengenai cara kerja dari sistem monitoring kebisingan yang akan dibahas.



Gambar 3.2. Blok Diagram Sistem

Dari gambar blok diagram sistem dapat diketahui sistem kerja alat Pemanfaatan E-Ktp Untuk Pengaktifan Dan Keamanan Kendaraan Roda Empat. E-KTP berfungsi sebagai *transponder* dan tergolong dalam *tag* pasif karena tidak memiliki catu daya sendiri, catu dayanya berasal dari pancaran gelombang RFID *reader*. RFID *reader* berfungsi untuk membaca nomor ID pada E-KTP. RFID *reader* yang digunakan memiliki frekuensi operasional 13,56 MHz. Data yang tersimpan dalam chip akan terkirim atau terbaca melalui gelombang radio setelah *tag*-antena menerima pancaran gelombang radio dari *reader*-antena kemudian data akan dikirim ke mikrokontroler. Kemudian mikrokontroler akan meritahkan Relay 1 HIGH (1) untuk menyalakan kelistrikan dan setelah kelistrikan menala maka relay 2 akan HIGH (1) digunakan sebagai stater (Engine) pada kendaraan roda empat. Jika E-KTP yang ditempelkan

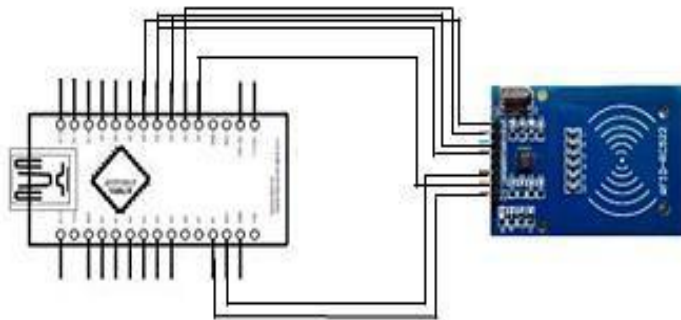
salah sebanyak 3 kali maka GSM Shield akan mengirimkan SMS kepada pemilik jika kendaraan dalam keadaan dalam keadaan tidak aman.

### 3.2.1 Perancangan Perangkat Keras

Perancangan menjadi bagian yang sangat penting dilakukan dalam pembuatan suatu alat karena dengan merancang terlebih dahulu dengan komponen yang tepat akan mengurangi berlebihnya pembelian komponen dan kerja alat sesuai dengan yang diinginkan. Untuk menghindari kerusakan komponen perlu dipahami juga akan karakteristik dari komponen-komponen tersebut.

#### 3.2.1.1 Rangkaian RFID (*Radio Frequency Identification*)

Rangkaian RFID digunakan sebagai inputan untuk mengaktifkan kelistrikan dan stater (Engine) kendaraan roda empat yang akan diproses oleh arduino gambar rangkaian RFID dan tata letak dapat dilihat seperti pada gambar 3.4



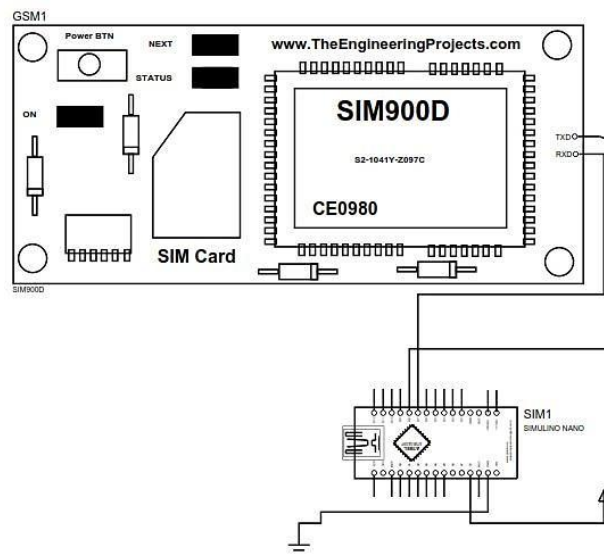
**Gambar 3.4 Rangkaian RFID (*Radio Frequency Identification*)**

Pada rangkaian RFID hanya beberapa kaki yang dihubungkan ke pin digital arduino nano agar hasil proses pada arduino dapat memberikan outputan pengaktifan kelistrikan dan stater (Engine) kendaraan roda empat. Penggunaan PIN arduino nano dan *RFID* dapat diketahui yaitu pin digital 2,3,4,5 dan 6 pada arduino akan dihubungkan ke pin SDA, SCK, MOSI, MISO dan RST pada pin RFID agar hasil

proses pada arduino dapat membaca nomer ID pada E-KTP yang akan digunakan untuk menghidupkan kelistrikan pada kendaraan roda empat.

### 3.2.1.2 Rangkaian *Gsm Shield*

Rangkaian *gsm shield* digunakan sebagai *output* untuk memberi informasi kepada pemilik kendaraan roda empat jika kendaraan telah menyala atau sebagai informasi jika terjadinya pencurian kendaraan roda empat. Gambar rangkaian *gsm shield* dan tata letak dapat dilihat seperti pada gambar 3.5.

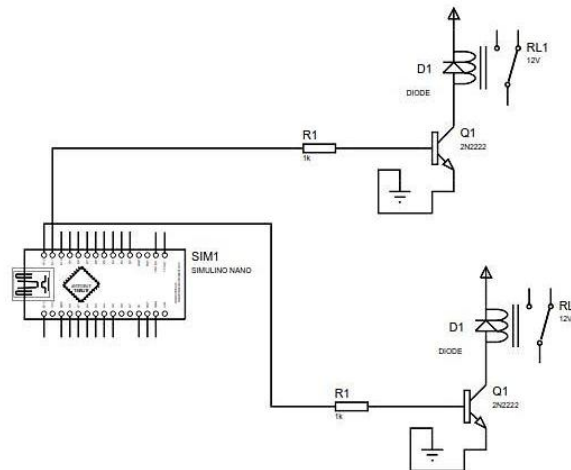


**Gambar 3.5 Rangkaian *Gsm Shield***

Pada rangkaian *gsm shield* hanya beberapa kaki yang dihubungkan ke pin digital arduino nano agar hasil proses pada arduino dapat mengirimkan pesan SMS kepada pemilik kendaraan. Penggunaan PIN arduino nano dan GSM Shield dapat diketahui yaitu pin digital 8 dan 7 pada arduino akan dihubungkan ke pin RXD dan TXD pada pin *gsm shield* agar hasil proses pada arduino dapat mengirimkan SMS (*Short Message Service*).

### 3.2.1.3 Rangkaian *Relay*

Rangkaian *relay* digunakan sebagai *output* untuk memberi menyalakan kelistrikan dan stater pada kendaraan roda empat. Gambar rangkaian *relay* dan tata letak dapat dilihat seperti pada gambar 3.6.

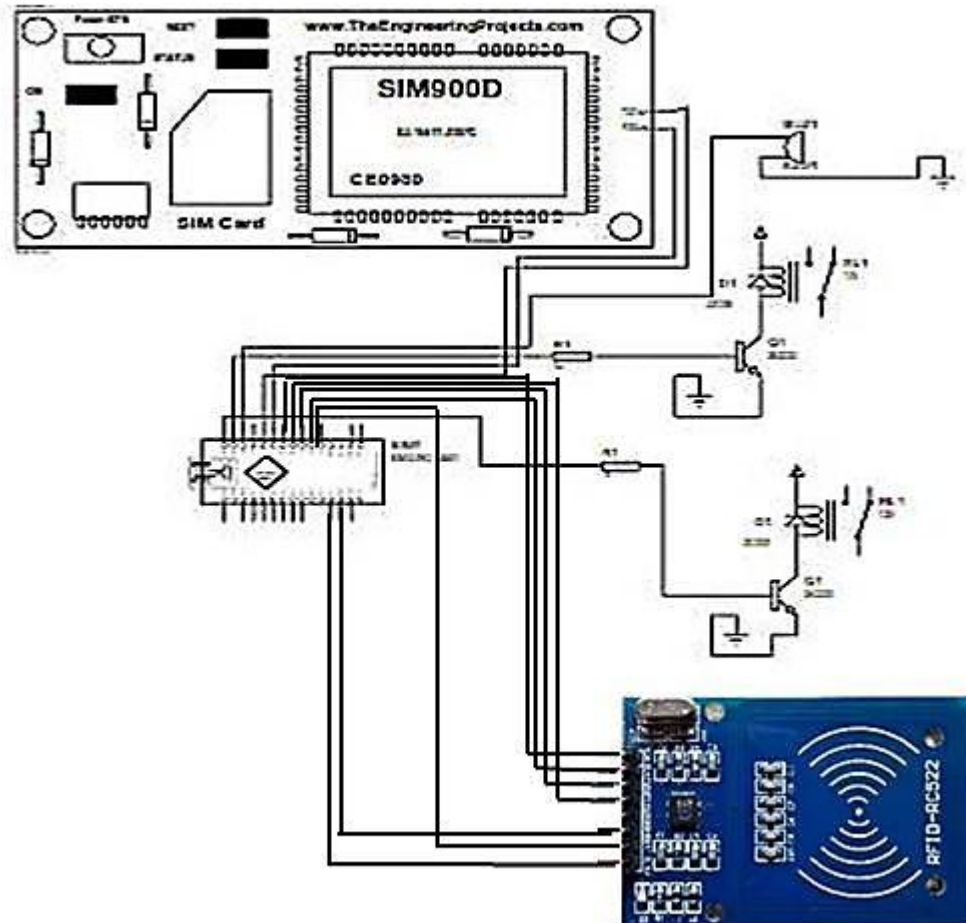


**Gambar 3.6 Rangkaian Relay**

Pada rangkaian *relay* hanya beberapa kaki yang dihubungkan ke pin digital arduino nano agar hasil proses pada arduino dapat menyalakan dan mematikan kendaraan roda empat. Penggunaan PIN arduino nano dan *relay* dapat diketahui yaitu pin digital 12 dan 13 pada arduino akan dihubungkan ke pin out 1 dan pin out 2 pada pin *relay* agar hasil proses pada arduino dapat menghidupkan relay berstatus HIGH dan LOW sehingga dapat digunakan sebagai otomatis pada kelistrikan kendaraan roda empat.

#### **3.2.1.4 Rangkaian Keseluruhan**

Rangkaian keseluruhan merupakan tahap terakhir dari perancangan yang telah dilakukan. Dalam tahap ini seluruh komponen dipasang sesuai dengan sistem yang telah dibuat, Adapun rangkaian keseluruhan dapat dilihat pada gambar 3.7



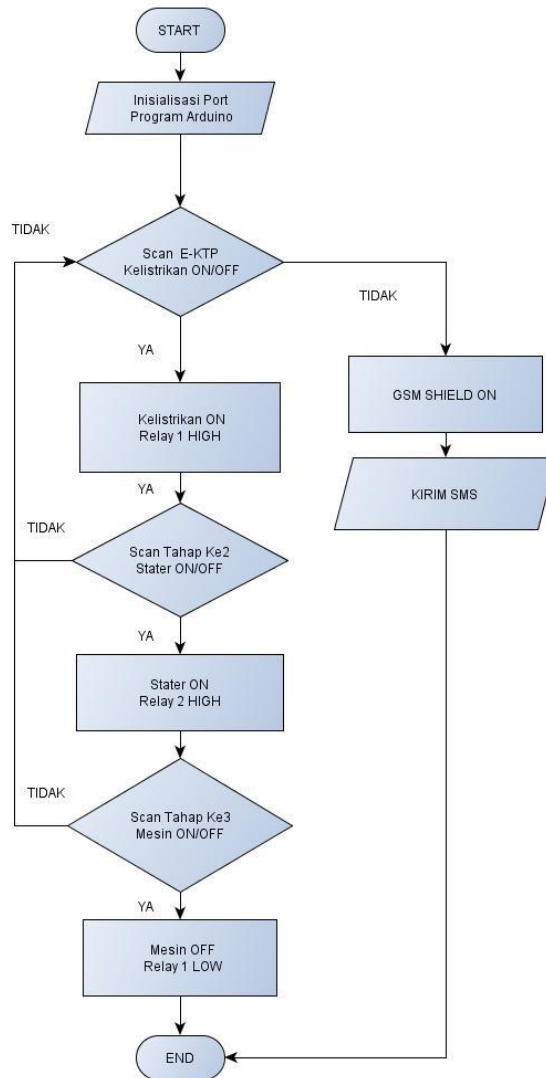
**Gambar 3.7 Rancangan Rangkaian Keseluruhan**

Dari rancangan rangkaian keseluruhan dapat diketahui sistem kerja dari alat yaitu jika RFID telah membaca nomer pada E-KTP dengan benar maka relay 1 dan relay 2 akan menyalakan kelistrikan dan stater (Engine) pada kendaraan roda empat. Jika E-KTP yang ditempelkan salah sebanyak 3 kali maka GSM Shield akan mengirimkan SMS (*Short Message Service*), kemudian Dengan adanya alat ini dapat membatu mencegah terjadinya pencurian kedaraan roda empat.

### 3.2.2 Perancangan Perangkat Lunak



Perancangan perangkat lunak dibuat dari pembuatan *flowchart* untuk pembuatan pada *hardware*. Pada gambar 3.8. akan ditampilkan *flowchart* dari program yang akan dibuat dalam penelitian ini.



**Gambar 3.8 Flowcart Penyalaan Kendaraan Roda Empat**

Di bawah ini merupakan penjelasan dari *flowchart* program pada gambar 3.8. :

1. Start adalah proses penyalaan alat.
2. Inisialisasi port adalah proses membaca port pada arduino Nano.

3. Proses Scan E-KTP adalah membaca ID pad E-KTP dan hasil scan akan ditampilkan pada LCD.
4. Jika hasil scan E-KTP salah sebanyak 3 kali maka GSM Shield akan mengirimkan SMS.
5. Jika Scan E-KTP tahap pertama benar maka akan kelistrikan ON kemudian scan E-KTP tahap kedua benar maka stater (Engine) kendaraan ON dan Jika scan E-KTP tahap ke3 maka mesin kendaraan akan OFF .
6. End

### 3.3 Analisa Kebutuhan

Tahapan selanjutnya setelah membuat rancangan perangkat keras dan perangkat lunak yaitu membuat analisa kebutuhan sistem. Analisa kebutuhan sistem dilakukan untuk mengetahui alat dan komponen serta perangkat lunak apa saja yang akan digunakan untuk mengimplementasikan sistem.

#### 3.3.1 Alat

Sebelum membuat Sistem Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler ada beberapa peralatan yang harus disiapkan. Daftar peralatan yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.3

**Tabel 3.3 Alat Yang Dibutuhkan**

No	Nama Alat	Spesifikasi	Fungsi	Jumlah
1	Komputer/ laptop	Window 7-10 32/64bit	Untuk membuat sebuah aplikasi yang akan di pakai di perangkat keras dan perngkat lunak	1 unit
2	Multitester	Analog/Digital	digunakan untuk mengukur tegangan (ACV-DCV), dan kuat arus (mA- $\mu$ A)	1 buah
3	Obeng	Obeng + dan -	Untuk merangkai alat	1 buah
4	Solder	-	Untuk menempelkan timah ke	1 buah

			komponen	
5	Bor pcb	-	Untuk membuat lobang baut atau komponen	1 buah
6	Tang Potong	-	Untuk memotong kabel dan kaki komponen	1 buah
7	Kit Arduino	-	Komponen Komplit arduino Nano	1 buah

### 3.3.2 Komponen

Sebelum membuat Sistem Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler ada beberapa peralatan yang harus disiapkan. Daftar komponen yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.4

**Tabel 3.4 Komponen Yang Dibutuhkan**

No	Nama Alat	Sepesifikasi	Fungsi	Jumlah
1	Kit Arduino Nano	Atmega328	Sebagai proses perintah yang akan di jalankan	1
2	RFID	-	Sebagai inputan untuk membaca E-KTP	1
3	<i>Gsm Shield</i>	-	Digunakan untuk membrikan inormasi berupa pesan SMS	1
4	Buzzer		Sebagai bunyi alarm	1
5	Relay		digunakan sebagai outputan untuk mengaktifkan dan mematikan kendaraan roda empat	1
6	Jumper		Digunakan sebagai penghubung/menjumper seluruh komponen	30

### 3.3.3 Software

Sebelum membuat Sistem Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler ada beberapa peralatan yang harus disiapkan. Daftar Software yang digunakan dalam penelitian ini akan dituliskan pada Tabel 3.5.

No	Nama	Spesifikasi	Fungsi
----	------	-------------	--------

1	IDE Arduino	Arduino 1.6.3	Membuat program yang akan di-download perangkat arduino
2	Proteus	7.1 Profesional	Merancang rangkaian yang akan digunakan untuk membuat alat

**Tabel 3.5. Daftar Software Yang Digunakan**

### **3.4 Implementasi**

Setelah mengumpulkan alat dan bahan, langkah selanjutnya adalah melakukan implementasi rancangan alat yang telah dibuat. Pada tahap ini hasil rancangan yang telah dibuat akan diimplementasikan untuk menjadi sistem yang sesungguhnya. Implementasi pada penelitian ini dibagi menjadi dua bagian, yaitu: Implementasi perangkat keras dan Implementasi perangkat lunak. Implementasi perangkat keras merupakan tahap terakhir dari perancangan sistem yang dilakukan dalam tahap ini seluruh komponen dipasang sesuai dengan sistem yang telah dibuat. Implementasi perangkat keras.

#### **3.4.1 Implementasi Perangkat Keras**

Realisasi perangkat keras merupakan tahap terakhir dari perancangan yang telah dilakukan. Dalam tahap ini seluruh komponen dipasang sesuai dengan sistem yang telah dibuat

#### **3.4.2 Implementasi Perangkat Lunak**

Penerapan perangkat lunak merupakan suatu tahap dimana program yang telah dirancang akan disimpan kedalam modul *mikrokontroler* melalui *downloader* dan menggunakan *software* tertentu sesuai dengan bahasa pemrograman yang akan digunakan. Disini peneliti menggunakan bahasa C dan menggunakan *software* Arduino. Pada *Software* Arduino program ditulis kemudian *dicompile*, tujuannya adalah untuk mengetahui apakah program yang dibuat sudah benar atau belum. Langkah terakhir yaitu meng-*upload* program kedalam modul *mikrokontroler*.

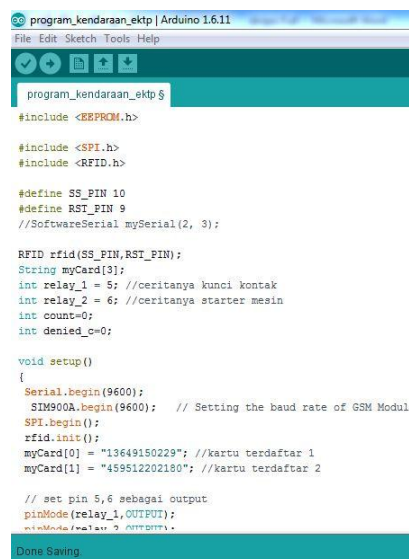
### **3.5 Pengujian Sistem**

Setelah perancangan *hardware* dan *software* selesai, maka yang dilakukan adalah *running* program, pengujian tiap-tiap rangkaian apakah sudah sesuai dengan yang diinginkan atau belum. Pengujian dilakukan pada bagian-bagian seperti pengujian respon, jangkauan sistem, catu daya dan rangkaian keseluruhan pada sistem ini.

### 3.5.1 Pengujian Mikrokontroler

Penerapan perangkat lunak merupakan suatu tahap dimana program yang telah dirancang akan disimpan kedalam modul *mikrokontroler* melalui *downloader* dan menggunakan *software* tertentu sesuai dengan bahasa pemrograman yang akan digunakan. Disini peneliti menggunakan bahasa C dan menggunakan *software* Arduino. Pada *Software* Arduino program ditulis kemudian *dcompile*, tujuannya adalah untuk mengetahui apakah program yang dibuat sudah benar atau belum. Langkah terakhir yaitu meng-*upload* program kedalam modul *mikrokontroler*.

Pada penelitian ini program yang dibuat untuk dapat menyalakan kendaraan roda empat menggunakan E-KTP dan dapat mengirimkan SMS. Berikut ini adalah tampilan *software* yang digunakan untuk menuliskan dan meng-*upload* program kedalam arduino seperti pada gambar 3.9.



```
program_kendaraan_ektp | Arduino 1.6.11
File Edit Sketch Tools Help
program_kendaraan_ektp $
#include <EEPROM.h>

#include <SPI.h>
#include <RFID.h>

#define SS_PIN 10
#define RST_PIN 9
//SoftwareSerial mySerial(2, 3);

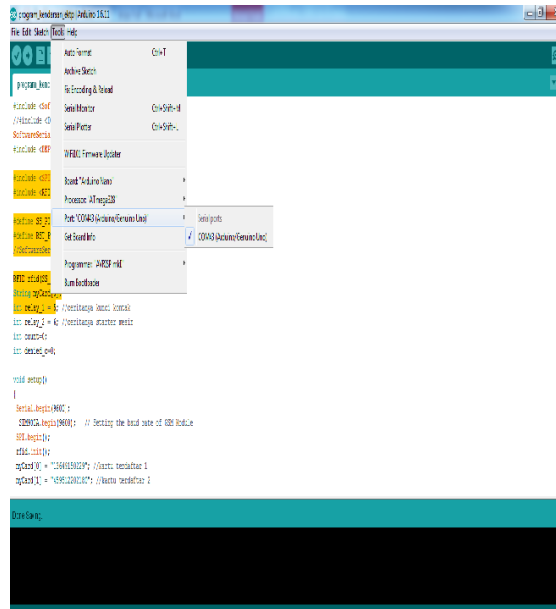
RFID rfid(SS_PIN,RST_PIN);
String myCard[3];
int relay_1 = 5; //ceritanya kunci kontak
int relay_2 = 6; //ceritanya starter mesin
int count=0;
int denied_c=0;

void setup()
{
  Serial.begin(9600);
  SIM900A.begin(9600); // Setting the baud rate of GSM Module
  SPI.begin();
  rfid.init();
  myCard[0] = "13649150229"; //kartu terdaftar 1
  myCard[1] = "459512202180"; //kartu terdaftar 2

  // set pin 5,6 sebagai output
  pinMode(relay_1,OUTPUT);
  pinMode(relay_2,OUTPUT);
}
Done Saving
```

### Gambar 3.9. Tampilan *Software* Arduino IDE

Untuk bisa meng-*upload* program ke Arduino Nano yang pertama harus mengatur port yang digunakan oleh Arduino. Pengaturan port Arduino dapat dilihat pada gambar 3.10.



Gambar 3.10. Pengaturan Port Arduino Uno

Pengaturan port Arduino diatas menggunakan port COM5. Setelah pengaturan port langkah selanjutnya yaitu meng-*compile* program. Berikut adalah hasil *compile* program pada gambar 3.11.

```
program_kendaraan_eksp
#include <SoftwareSerial.h>
// #include <FFlayer_mini_hg3.h>
SoftwareSerial SIM900A(7,8);
#include <MFRC522.h>

#include <SPI.h>
#include <RFID.h>

#define SS_PIN 10
#define RST_PIN 9
//SoftwareSerial mySerial(2, 3);

RFID rfid(SS_PIN,RST_PIN);
String myCard[3];
int relay_1 = 5; //certanya kunci kontak
int relay_2 = 6; //certanya starter mesin
int count=0;
int denied=0;

void setup()
{
  Serial.begin(9600);
  SIM900A.begin(9600); // Setting the baud rate of GSM Module
  SPI.begin();
  rfid.init();
  myCard[0] = "13649150229"; //kartu terdaftar 1
  myCard[1] = "45951202100"; //kartu terdaftar 2

  Done compiling.

Sketch uses 7,292 bytes (22%) of program storage space. Maximum is 30,720 bytes.
Global variables use 708 bytes (34%) of dynamic memory, leaving 1,340 bytes for local variables. Maximum is 2,048 bytes.
```

**Gambar 3.11. Hasil Compile Program**

Setelah program berhasil di *compile* selanjutnya yaitu meng-*upload* file ke Arduino Nano seperti pada gambar 3.12.

```
program_kendaraan_eksp
#include <SoftwareSerial.h>
// #include <FFlayer_mini_hg3.h>
SoftwareSerial SIM900A(7,8);
#include <MFRC522.h>

#include <SPI.h>
#include <RFID.h>

#define SS_PIN 10
#define RST_PIN 9
//SoftwareSerial mySerial(2, 3);

RFID rfid(SS_PIN,RST_PIN);
String myCard[3];
int relay_1 = 5; //certanya kunci kontak
int relay_2 = 6; //certanya starter mesin
int count=0;
int denied=0;

void setup()
{
  Serial.begin(9600);
  SIM900A.begin(9600); // Setting the baud rate of GSM Module
  SPI.begin();
  rfid.init();
  myCard[0] = "13649150229"; //kartu terdaftar 1
  myCard[1] = "45951202100"; //kartu terdaftar 2

  Done uploading.

Sketch uses 7,292 bytes (22%) of program storage space. Maximum is 30,720 bytes.
Global variables use 708 bytes (34%) of dynamic memory, leaving 1,340 bytes for local variables. Maximum is 2,048 bytes.
```

### Gambar 3.12. Upload Program

Berikut adalah potongan program yang telah di *download* oleh Arduino Uno beserta penjelasannya.

#### 3.5.2 Rancangan Pengujian RFID (*Radio Frequency Identification*)

Rancangan pengujian RFID bertujuan untuk mengukur akurat respon ketika RFID E-KTP ditempelkan pada reader RFID dan peneliti akan mengujicoba beberapa kali dengan menggunakan E-KTP yang berbeda apakah RFID dapat berkerja dengan baik. Dalam melakukan uji coba kali ini peneliti menggunakan timer yang digunakan sebagai pengukur lama respon RFID dalam membaca E-KTP yang digunakan sebagai pengganti ID card RFID. *Script* program arduino RFID dapat dilihat pada seperti berikut.

```
#include <SoftwareSerial.h>
#include <SPI.h>
#include <RFID.h>

#define SS_PIN 10
#define RST_PIN 9

RFID rfid(SS_PIN,RST_PIN);
String myCard[3];
int relay_1 = 12; //ceritanya kunci kontak
int relay_2 = 13; //ceritanya starter mesin
int count=0;
int denied_c=0;

void setup()
{
  Serial.begin(9600);
```



```
SPI.begin();
rfid.init();
myCard[0] = "13649150229"; //kartu terdaftar 1
myCard[1] = "459512202180"; //kartu terdaftar 2

// set pin 12,13 sebagai output
pinMode(relay_1,OUTPUT);
pinMode(relay_2,OUTPUT);

// set default relay mode off
digitalWrite(relay_1,HIGH);
digitalWrite(relay_2,HIGH);
}
void loop()
{
if(rfid.isCard())
{
if(rfid.readCardSerial())
{
String serial;
int member = 0;
// perulangan untuk membaca kartu rfid
for (int i=0; i<= 4; i++)
{
serial += String(rfid.serNum[i],DEC);
}
Serial.println("Card ID: " + serial); // untuk menampilkan serial kartu
```

```
//perulangan untuk memeriksa kartu rfid yang terdaftar
for (int i=0; i<2; i++)
{
  if (serial == myCard[i])
  {
    member = 1;
  }
}

// jika kartu terdaftar
if (member == 1)
{
  count++;
  if(count == 1){
    Serial.println("Hallo sir, silakan tap lagi untuk menyalakan mesin");
    // mp3_play (1);
    digitalWrite(relay_1,LOW);
    digitalWrite(relay_2,HIGH);
  }
  if(count == 2){
    // mp3_play (2);
    Serial.println("Menyalakan mesin");
    delay(1700);
    digitalWrite(relay_1,LOW);
    digitalWrite(relay_2,LOW);
    delay(800);
    digitalWrite(relay_2,HIGH);
  }if(count == 3){
```

```

Serial.println("Mematikan mesin dan kontak");
digitalWrite(relay_1,HIGH);
digitalWrite(relay_2,HIGH);
// delay(200);
// mp3_play (5);
}
if (count >=3){count=0;}

} else
{
denied_c++;
Serial.println("Maaf, kartu anda tidak terdaftar!");
// mp3_play (3);
digitalWrite(relay_1,HIGH);
digitalWrite(relay_2,HIGH);
count=0;
if(denied_c == 3){
delay(2000);
Serial.println("Akses gagal selama 3 kali, sistem di blokir dalam waktu 5
menit");
SendMessage1();
//sistem delay selama 5 menit
int longDelayInSeconds = 300;
int p = 0;
while(p < longDelayInSeconds){
delay(1000);
p++;
}
}

```

```

    //kembalikan denied count ke 0
    denied_c=0;
  }
}
}

}
rfid.halt();
delay(500);
if (SIM900A.available(>0)
  Serial.write(SIM900A.read());
}

```

### 3.5.3 Rancangan Pengujian *Gsm Shield*

Rancangan pengujian *gsm shield* bertujuan untuk mengetahui ketika *gsm shield* mendapat perintah dari inputan apakah dapat mengirimkan SMS kepada pemilik kendaraan dan mengukur berapa lama waktu respon GSM Shield dalam mengirimkan pesan SMS kepada pemilik kendaraan dalam melakukan ujicoba GSM Shield peneliti menggunakan timer untuk mengukur waktu respon. *Script* program arduino RFID dapat dilihat pada seperti berikut.

```

SoftwareSerial SIM900A(7,8);
#include <EEPROM.h>
void setup()
{
  Serial.begin(9600);
  SIM900A.begin(9600); // Setting the baud rate of GSM Module

```

```

}
void loop()
{
if (SIM900A.available(>0)
  Serial.write(SIM900A.read());
}
void SendMessage1()
{
  Serial.println ("SIM900A Mengirim SMS");
  SIM900A.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
  delay(100); // Delay of 1000 milli seconds or 1 second
  Serial.println ("Set SMS Number");
  SIM900A.println("AT+CMGS="+6285382932197+"\r"); // Replace with your
mobile number
  delay(100);
  Serial.println ("Set SMS Content");
  SIM900A.println("ADA MALING...");// The SMS text you want to send
  delay(100);
  Serial.println ("Finish");
  SIM900A.println((char)26);// ASCII code of CTRL+Z
  delay(100);
  Serial.println (" ->SMS Selesai dikirim");
}

void RecieveMessage()
{
  Serial.println ("SIM900A Membaca SMS");
  delay (100);

```

```
SIM900A.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live
SMS
delay(100);
Serial.write (" ->Unread SMS Selesai dibaca");
}
```

#### **3.5.4 Rancangan Pengujian Relay**

Rancangan pengujian *Relay* bertujuan untuk mengetahui ketika *relay* diberikan status *HIGH* apakah *relay* dapat mematikan dan menghidupkan kendaraan roda empat sesuai dengan program pada arduino. *Script* program arduino relay dapat dilihat pada seperti berikut.

```
int relay_1 = 5;
void setup()
{
  Serial.begin(9600);
  // set pin 5 sebagai output
  pinMode(relay_1,OUTPUT);
}
void loop()
{
  digitalWrite(relay_1,HIGH);
  delay(1000);
  digitalWrite(relay_1,LOW);
  delay (1000);
}
```

#### **3.5.5 Pengujian Sistem Keseluruhan**

Pengujian sistem secara keseluruhan bertujuan untuk memastikan semua komponen inputan dan outputan dapat berjalan dengan sempurna. Mulai dari RFID, *Gsm Shield*, *Relay*, blok sistem arduino Nano dan program yang mengatur jalannya sistem keseluruhan.

### **3.6 Analisis Kerja**

Untuk analisa kerja, dilakukan bersama pada saat melakukan uji coba alat yang bertujuan untuk mengetahui kerja alat tersebut. Selain itu yang akan dianalisa adalah jarak respon dalam untuk inputan pada sistem pemanfaatan E\_KTP sebagai pengaktifan kendaraan roda empat. Berdasarkan hasil pengujian sistem yang telah di dapat akan dianalisis untuk memastikan bahwa sistem yang telah dibuat sesuai dengan harapan.

## **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil uji coba dan analisis terhadap sistem. Pengujian dimulai dengan memastikan setiap komponen (*arduino, RFID, Relay dan GSM Shield Sim900*) apakah alat yang telah dibuat dalam kondisi bagus dapat bekerja dengan baik sesuai dengan program yang telah dibuat, kemudian mengecek setiap jalur yang terhubung dengan komponen yang digunakan telah terkoneksi, dimana rangkaiannya disesuaikan dengan gambar skematiknya. Pengujian yang dilakukan meliputi pengujian *RFID, Relay, GSM Shield Sim900* dan pengujian sistem keseluruhan.

### **4.1 Hasil**

Uji coba dilakukan untuk memastikan rangkaian yang dihasilkan mampu bekerja sesuai dengan yang diharapkan. maka terlebih dahulu dilakukan langkah pengujian dan mengamati langsung rangkaian serta komponen. Hasil pengukuran ini dapat diketahui rangkaian telah bekerja dengan baik atau tidak, sehingga apabila terdapat kesalahan dan kekurangan akan terdeteksi. Gambar 4.1 berikut ini merupakan gambar dari bentuk fisik alat yang telah dibuat.





**Gambar. 4.1. Bentuk Fisik Alat**

dari hasil perakitan peneliti dan uji sistem kerja dari alat yang telah berkerja dengan baik yaitu. Pertama harus menepelkan E-KTP jika E-KTP yang ditempelkan benar maka relay 1 akan HIGH yang digunakan sebagai penyalat kelistrikan kendaraa, kemudian pemilik harus menempelkan E-KTP kembali agar relay 2 bertatus HIGH yang digunakan sebagai stater kendaraan roda empat.

#### **4.1.1 Hasil Pengujian dan Pembahasan**

Pada pengujian ini meliputi pengujian E-KTP, relay, *GSM Shield Sim900* dan rangkaian keseluruhan. Pengujian ini dilakukan agar peneliti dapat mengetahui kelebihan dan kekurangan sistem yang telah di buat hasil pengujian sebagai berikut:

#### **4.1.2 Pengujian Scan E-KTP**

pengujian ini bertujuan untuk memastikan bahwa tidak ada kesalahan pada perintah terhadap keluaran yang didapatkan. Pada tahap ini peneliti melakukan pengujian dengan 2 kartu E-KTP yang berbeda agar dengan 1 E\_KTP yang sudah terdaftar dalam program dan 1 yang belum terdaftar pengujian ini dilakukan agar peneliti dapat mengetahui jika nomer ID pada setiap E-KTP adalah berbeda :



**Gambar 4.2 Pengujian Scan E-KTP**

**Tabel 4.2. Hasil Pengujian RFID**

NO	Nomer E_KTP	Keterangan
1	136497101136	Akses Ditrима
2	215220821312	Akses Ditolak

Dari hasil ujicoba *Sensor MQ2* dapat diketahui Pengambilan data dengan sensor Mq2 ini dapat diketahui jika jarak yang dapat terbaca oleh sensor MQ2 yaitu 30 cm sedangkan Jika lebih dari jarak 30 cm sensor sudah tidak dapat membaca dari inputan. Pada Tabel 2 menunjukkan data yang dihasilkan pada pengujian sensor Mq2.

#### **4.1.3 Pengujian E-KTP Menyalakan Dan Mematikan Kendaraan**

Pada pengujian ini dilakukan untuk mengetahui apakah E-KTP dapat dengan baik dalam melakukan inputan untuk menyalakan kendaraan, stater dan mematikan kendaraan roda empat hasil pengujian dapat dilihat seperti pada tabel 4.3.

**Tabel 4.3. Hasil Pengujian Relay**

Nomer E-KTP	Jumlah Scan			Status Relay		Keterangan
	Menyalakan kelistrikan	Stater	Mematikan	1	2	
136497101136	√			High	Low	Menyalakan Kelistrikan
		√		High	High	menyalakan Stater
			√	Low	Low	Mematikan Kendaraan

Dari hasil tabel diatas dapat diketahui yaitu pada ujicoba ke 1 sampai dengan ujicoba ke 3 dengan nomer E-KTP yang terdaftar maka dapat diketahui jika hanya melakukan 1 kali scan maka relay 1 akan HIGH untuk menyalakan kelistrikan sedangkan pada ujicoba ke 2 melakukan 2 kali scan yang berarti pemilik akan menyalakan kelistrikan dan stater kendaraan yaitu relay 1 dan 2 akan HIGH dan pada ujicoba ke 3 melakukan 3 kali scan ang artinya kendaraan dimatikan.

#### **4.1.4 Hasil Pengujian Gsm Shield**

pengujian ini bertujuan untuk memastikan bahwa jika scan E-KTP yang ditempelkan salah sebanyak 3 kali maka apakah GSM Shield dapat dengan baik salam mengirimkan SMS pengujian pada tahap ini akan ditampilkan pada gambar berikut :



**Gambar 4.4 Peringatan SMS Jika Terjadi Kebakaran**

**Tabel 4.4. Hasil Pengujian *Gsm Shield***

Nomer E=KTP	Kesalahan Scan			Status <i>Gsm Shield</i>	Waktu respon pengiriman SMS
	1	2	3		
215220821312	√			<i>Gsm Shield</i> tidak Mengirim SMS	-
	√	√		<i>Gsm Shield</i> tidak Mengirim SMS	-
	√	√	√	<i>Gsm Shield</i> Mengirim SMS	5 detik

Dari hasil tabel diatas dapat diketahui yaitu pada ujicoba ke 1 sampai dengan ujicoba ke 3 dengan nomer E-KTP yang tidak terdaftar maka dapat diketahui jika kesalahan dalam melakukan scan sebanyak 3 kali maka GSM Shield akan mengirimkan SMS kepada pemilik kendaraan dan sistem akan diblokir selama 5 menit.

#### 4.2 Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan dilakukan untuk menguji kinerja Sistem Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler. Dari hasil ujicoba sistem dapat diketahui bahwa sistem dapat berkerja dengan baik sesuai perintah pada program yang telah dibuat dapat dilihat seperti pada tabel 4.5. berikut hasil pengujian sistem keseluruhan.

**Tabel 4.5. Hasil Pengujian Sistem Keseluruhan**

Nomer E-KTP	Jumlah Scan			Relay		Gsm Shield	Tampilan LCD
	1	2	3	1	2		
36497101136	√			High	Low	<i>Gsm Shield</i> tidak Mengirim SMS	Menyalakan kelitrikan
	√	√		High	High	<i>Gsm Shield</i> tidak Mengirim SMS	Menyalakan mesin
	√	√	√	Low	low	<i>Gsm Shield</i> tidak Mengirim SMS	Mematikan kendaraan
2152208213 12	√			Low	Low	<i>Gsm Shield</i> Mengirim SMS	Akses ditolak
	√	√	√	Low	low	<i>Gsm Shield</i> Mengirim SMS	Akses ditolak

Telah berhasil membuat alat Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler. jika hanya melakukan 1 kali scan maka relay 1 akan HIGH untuk menyalakan kelistrikan sedangkan pada ujicoba ke 2 melakukan 2 kali scan yang berarti pemilik akan menyalakan kelistrikan dan stater kendaraan yaitu relay 1 dan 2 akan HIGH dan pada ujicoba ke 3 melakukan 3 kali scan yang artinya kendaraan dimatikan. Serta jika kesalahan dalam melakukan scan sebanyak 3 kali maka GSM Shield akan mengirimkan SMS kepada pemilik kendaraan dan sistem akan diblokir selama 5 menit.



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian dan pengukuran didapat kesimpulan sebagai berikut :

Telah berhasil membuat alat Pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler. jika hanya melakukan 1 kali scan maka relay 1 akan HIGH untuk menyalakan kelistrikan sedangkan pada ujicoba ke 2 melakukan 2 kali scan yang berarti pemilik akan menyalakan kelistrikan dan stater kendaraan yaitu relay 1 dan 2 akan HIGH dan pada ujicoba ke 3 melakukan 3 kali scan ang artinya kendaraan dimatikan. Serta jika kesalahan dalam melakukan scan sebanyak 3 kali maka GSM Shield akan mengirimkan SMS kepada pemilik kendaraan dan sistem akan diblokir selama 5 menitmengirimkan SMS apabila dalam melakukan scan E-KTP salah sebanyak 3kali.

#### **5.2 Saran**

Adapun saran yang dapat disampaikan penulis dari sistem pengaman Kendaraan Roda Empat Menggunakan E-KTP Berbasis Mikrokontroler yaitu :

1. Skripsi selanjutnya dapat mengembangkan dengan mengganti RFID reader dengan tipe RC522 yang berwarna merah agar pembacaan E-KTP dapat lebih baik.

Belum dilengkapi alat yang dapat memonitoring kendaraan jarak jauh. Diharapkan peneliti selanjutnya jika ingin mengembangkan alat ini dapat ditambahkan dengan GPS agar dapat memonitoring lokasi kendaraan berada.

## DAFTAR PUSTAKA

- Arduino. (2016). *Arduino Uno & Geniuno Uno*. Retrieved Mei 6, 2016, from Arduino Website: <https://www.arduino.cc/en/main/arduinoBoardUno>
- Ariesta Adhitama Satya Negara, Ufi Najib, Jenny Putri Hapsari. (2016). Pemanfaatan E-Ktp Untuk Pengaktifan Sepeda Motor Berbasis Arduino Uno. *Transistor Elektro*, Vol 2.
- Kurnia Dwi Artika. (2013). Rancang Bangun Sistem Pengaman Pada Sepeda Motor Denganmemanfaatkan Sensor Encoder Dan Sensor Ping. *Kurnia D.A, ROTOR*, Volume 6 Nomor 1.
- Nurdin Bagenda, D., & Indra, p. (2014). *Prototipe Sistem Keamanan Dan Pengendalian Sepeda Motor Berbasis Atmega 8535*. Bandung: STMIK LPKIA.
- Rahman, A. (2015). Rancang Bangun Stater Kendaraan Bermotor Menggunakan Kartu RFID . *Teknik Elektro*, Vol 3.
- Rizky Muhammad Syafii, M. I. (2017). Desain Dan Implementasi Sistem Keamanan Locker Menggunakan E-Ktp Berbasis Arduino Pro Mini. *Energi Elektrik*, Volume 7 .
- Saputro, E. (2016). Rancang Bangun Pengaman Pintu Otomatis Menggunakan E-KTP Berbasis Mikrokontroler Atmega328.



Sidarta. (2016). Rancang Bnagun Sistem Keamanan Kendaraan. *Teknik Elektro*.

Slamet Haryono, W. (2016). Pembuatan Trigger Gamelan dengan Memanfaatkan Mikrokontroler Arduino Nano At328. *Fakultas Bahasa dan Seni, Universitas Negeri Semarang*, 135.

Statistik, Badan Pusat. (2014). *Statistik Kriminal 2014*.

Sutris. (2015, januari 12). *Keamananan*. Retrieved Agustus 12, 2018, from Wikipedia.

Suyadhi, T. D. (2010). *Buku Pintar Robotika* (Nikodemus WK ed.). Yogyakarta: ANDI OFFSET.

# LAMPIRAN

```
#include <LiquidCrystal.h>

#include <SoftwareSerial.h>

SoftwareSerial SIM900A(3,2);

#include <EEPROM.h>

#include <SPI.h>

#include <RFID.h>

LiquidCrystal lcd (A0,A1,A2,A3,A4,A5);

/*

Sambungan Module RFID RC-522 ke Arduino

* MOSI: Pin 11

* MISO: Pin 12

* SCK : Pin 13

* SS/SDA: Pin 10

* RST: Pin 9

* RQ: Tidak digunakan

*/

#define SS_PIN 10

#define RST_PIN 9

RFID rfid(SS_PIN,RST_PIN);

String myCard[3];

int relay_1 = 5; //ceritanya kunci kontak

int relay_2 = 4; //ceritanya starter mesin
```

```
int count=0;

int denied_c=0;

void setup()
{
  Serial.begin(9600);
  SIM900A.begin(9600); // Setting the baud rate of GSM Module
  delay(1000);
  Serial.println ("SIM900A Ready");
  SPI.begin();
  rfid.init();
  myCard[0] = ""; //kartu E_KTP terdaftar 1
  myCard[1] = "136497101136"; //kartu E-KTP terdaftar 2

  // set pin 5,6 sebagai output
  pinMode(relay_1,OUTPUT);
  pinMode(relay_2,OUTPUT);

  // set default relay mode off
  digitalWrite(relay_1,HIGH);
  digitalWrite(relay_2,HIGH);

  lcd.begin(16,2);
```

```
lcd.setCursor(0,0);  
lcd.print("->Tugas Akhir<-");  
lcd.setCursor(2,1);  
lcd.print("-> GEDE <-");  
delay(2000);  
lcd.clear();  
lcd.setCursor (0,0);  
lcd.print(" -System Ready- ");  
}
```

```
void loop()  
{  
  if(rfid.isCard())  
  {  
    if(rfid.readCardSerial())  
    {  
      String serial;  
      int member = 0;  
  
      // perulangan untuk membaca kartu rfid  
      for (int i=0; i<= 4; i++)  
      {  
        serial += String(rfid.serNum[i],DEC);
```

```
}  
  
Serial.println("Card ID: " + serial); // untuk menampilkan serial kartu  
  
lcd.setCursor(0,1);  
  
lcd.print("ID= " +serial);  
  
  
//perulangan untuk memeriksa kartu rfid yang terdaftar  
for (int i=0; i<2; i++)  
{  
  if (serial == myCard[i])  
  {  
    member = 1;  
  }  
}  
  
  
// jika kartu terdaftar  
if (member == 1)  
{  
  count++;  
  if(count == 1){  
    Serial.println("Hallo , silakan tap lagi untuk menyalakan mesin");  
    lcd.setCursor (1,0);  
    lcd.print("Kelistrikan ON ");  
    digitalWrite(relay_1,LOW);
```

```
digitalWrite(relay_2,HIGH);

}

if(count == 2){

Serial.println("Menyalakan mesin");

lcd.setCursor (0,0);

lcd.print("Menyalakan Mesin");

digitalWrite(relay_1,LOW);

digitalWrite(relay_2,LOW);

delay(800);

digitalWrite(relay_2,HIGH);

}if(count == 3){

Serial.println("Mematikan mesin");

lcd.setCursor (0,0);

lcd.print("Mematikan mesin ");

delay(2000);

digitalWrite(relay_1,HIGH);

digitalWrite(relay_2,HIGH);

lcd.clear();

lcd.setCursor (0,1);

lcd.print("-Scan Your Card-");

}
```

```
if (count >=3){count=0;}

} else
{
denied_c++;

Serial.println("Maaf, kartu anda tidak terdaftar!");

lcd.setCursor (0,0);

lcd.print(" Akses ditolak ");

digitalWrite(relay_1,HIGH);

digitalWrite(relay_2,HIGH);

count=0;

if(denied_c == 3){

delay(2000);

Serial.println("Akses gagal selama 3 kali, sistem di blokir dalam waktu 5 menit");

//sistem delay selama 5 menit

int longDelayInSeconds = 10;

int p = 0;

while(p < longDelayInSeconds){

delay(1000);

p++;

SendMessage1();
```



```
Serial.print(p++);  
Serial.println("");  
  
}  
  
//kembalikan denied count ke 0  
denied_c=0;  
lcd.clear();  
lcd.setCursor (0,1);  
lcd.print("-Scan Your Card-");  
  
}  
}  
}  
}  
rfid.halt();  
delay(500);  
if (SIM900A.available(>0)  
Serial.write(SIM900A.read());  
}  
void SendMessage1()  
{  
Serial.println ("SIM900A Mengirim SMS");
```

```

SIM900A.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

Serial.println ("Set SMS Number");

SIM900A.println("AT+CMGS=\"+6282281109719\\r\"); // Replace with your mobile
number

delay(1000);

Serial.println ("Set SMS Content");

SIM900A.println("ADA YANG MENCoba AKSES SISTEM");// The SMS text you want to send

delay(100);

Serial.println ("Finish");

SIM900A.println((char)26);// ASCII code of CTRL+Z

delay(1000);

Serial.println (" ->SMS Selesai dikirim");
}

void RecieveMessage()

{

Serial.println ("SIM900A Membaca SMS");

delay (1000);

SIM900A.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS

delay(1000);

Serial.write (" ->Unread SMS Selesai dibaca");

}

```

# Arduino Nano 3.1

Item# ARMB-0022



## Overview:

Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a smallest, complete, and breadboard friendly. It has everything that Diecimila/Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano is automatically sense and switch to the higher potential source of power, there is no need for the power select jumper.

Nano's got the breadboard-ability of the Boarduino and the Mini+USB with smaller footprint than either, so users have more breadboard space. It's got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It is two layers. That make it easier to hack and more affordable.

---

Electronics Source Co.,Ltd  
7/129 Central Pinklao Bldg., 17FL., Unit 1702  
Baromrachonnee Rd., Bangkok-noi, Bangkok 10700

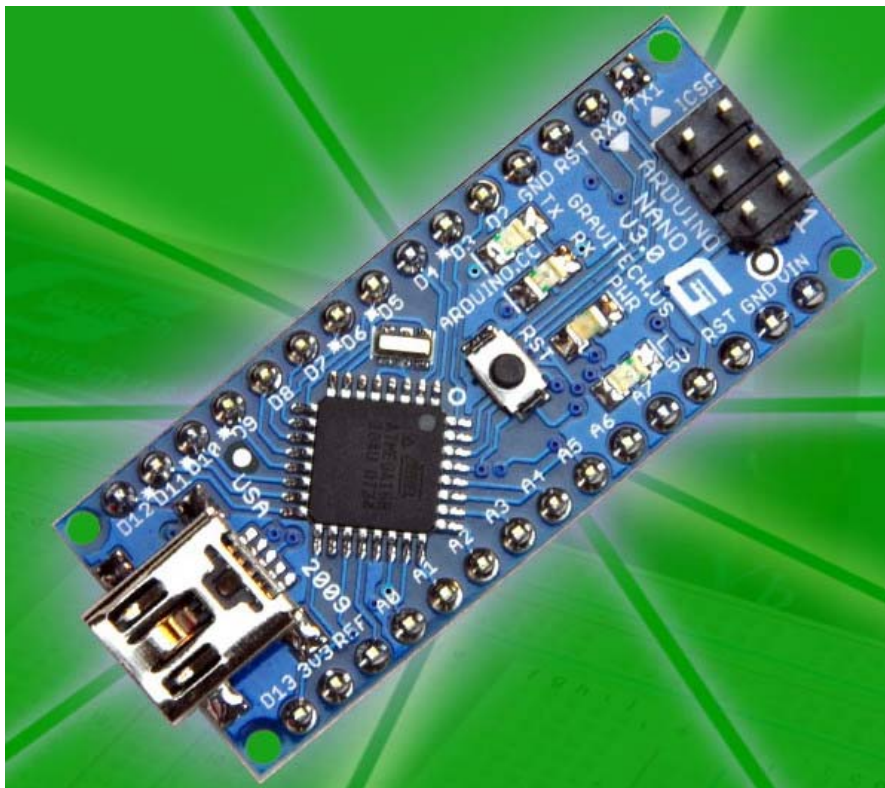
Website : <http://www.es.co.th>  
Email : [info@es.co.th](mailto:info@es.co.th)  
Tel : (662) 884-9210 (6 line)  
Fax : (662) 884-9213-4

**Specifications:**

Microcontroller	Atmel ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory bootloader)	32 KB (of which 2KB used by
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Dimensions	0.70" x 1.70"

# *Arduino Nano (V3.0)*

## *User Manual*



Released under the Creative Commons Attribution Share-Alike 2.5 License

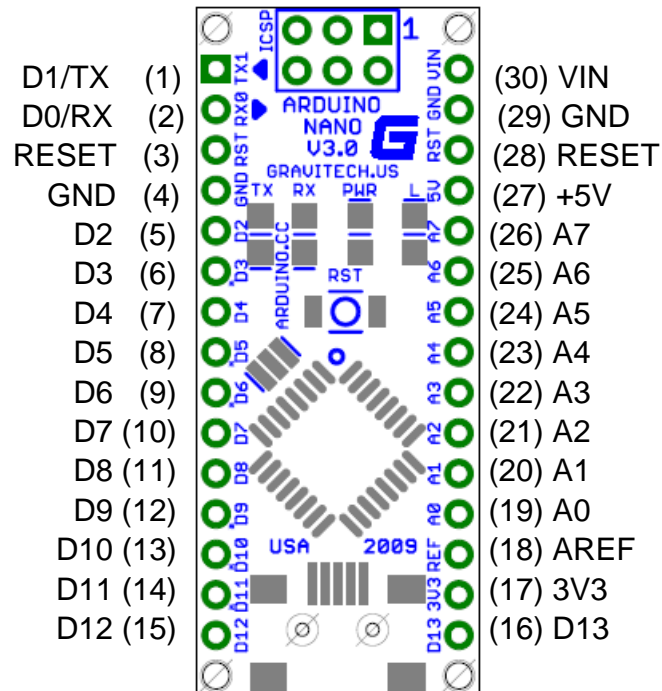
<http://creativecommons.org/licenses/by-sa/2.5/>

More information:

[www.arduino.cc](http://www.arduino.cc)

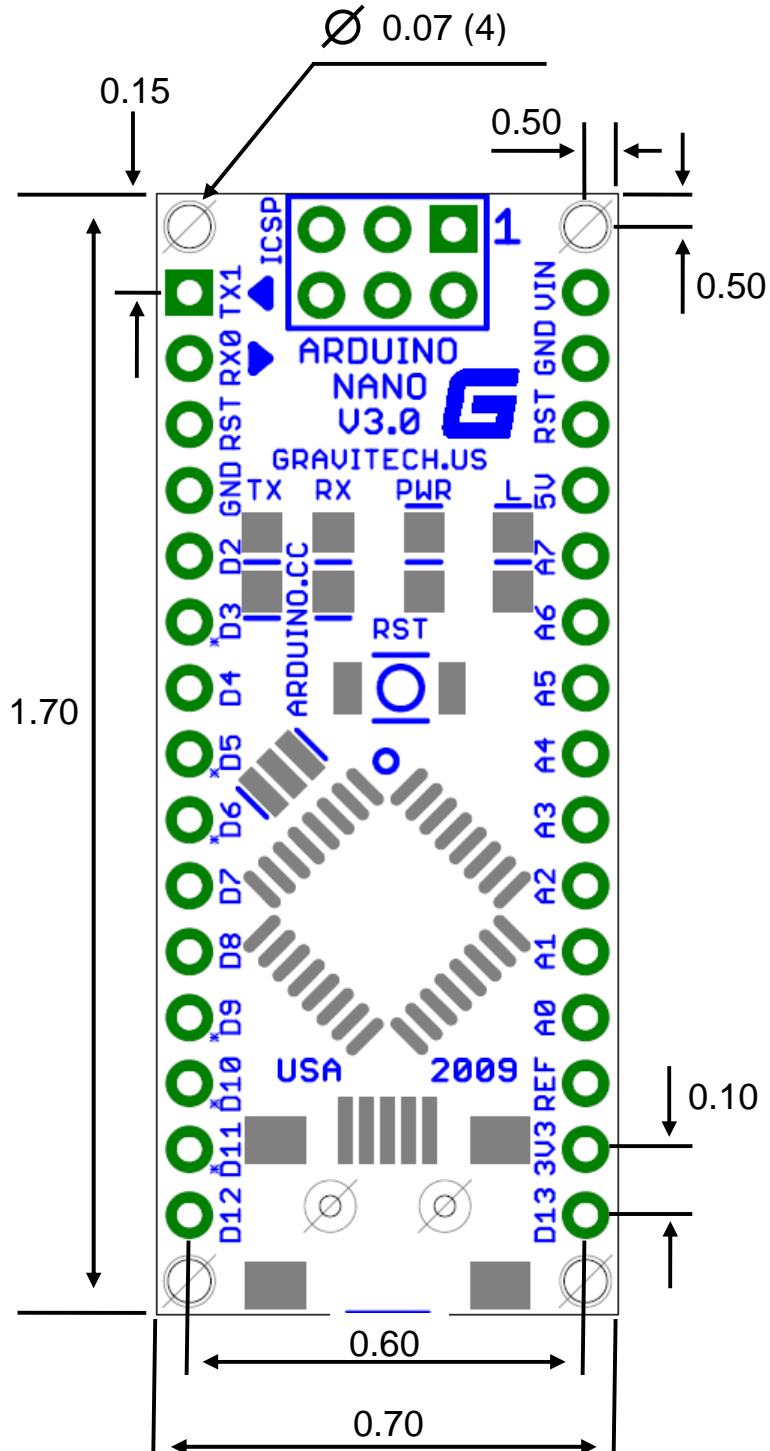
Rev 3.0

## Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A0-A7	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

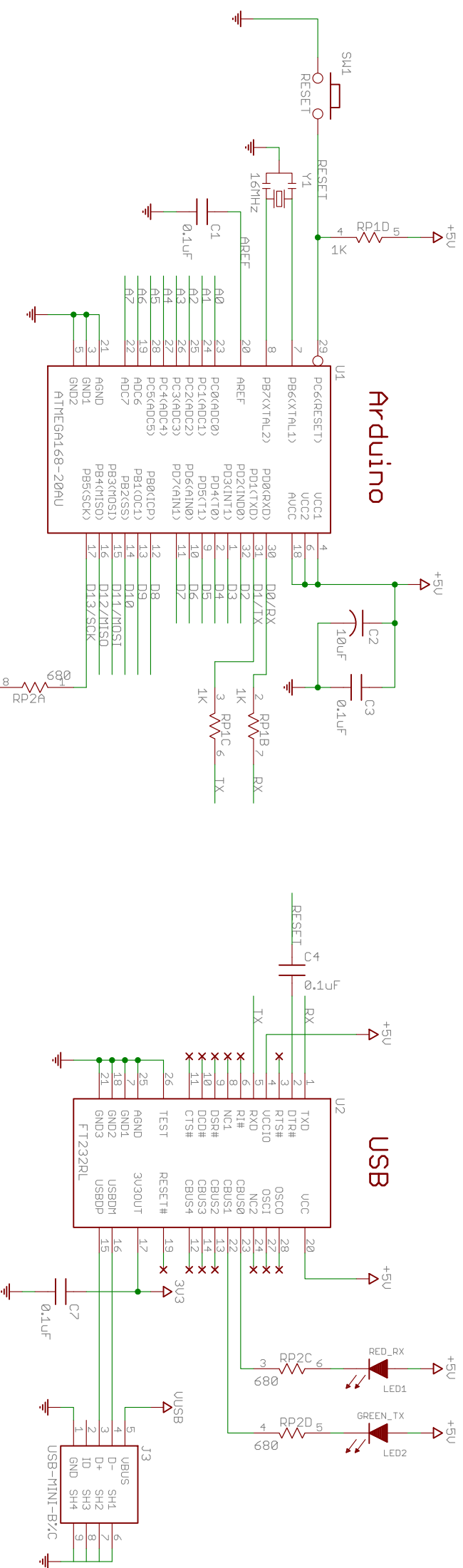
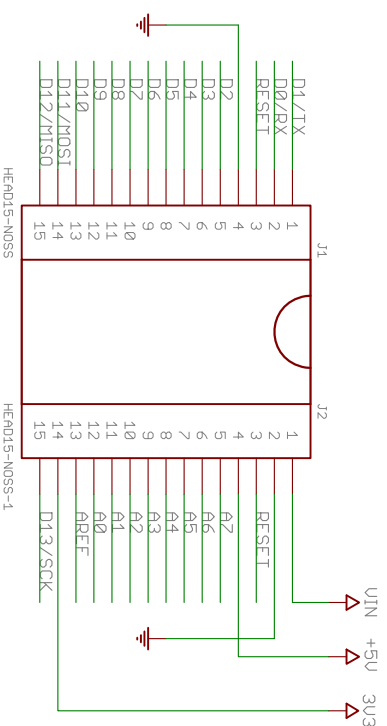
# Arduino Nano Mechanical Drawing



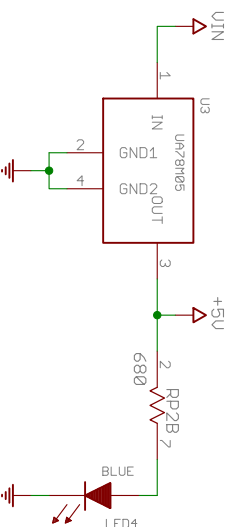
# Arduino Nano v3.0

Copyright 2010 under the Creative Commons Attribution Share-Alike 2.5 License

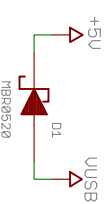
<http://creativecommons.org/licenses/by-sa/2.5/>



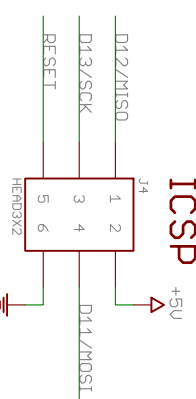
## +5V REG



## +5V AUTO SELECTOR



## ICSP



## NOT USED



TITLE: Arduino Nano30_2010	
Document Number:	
Date: 3/4/2010 7:01:53 PM	Sheet: 1/1
REV:	3.0



**EQ1 SERIES**

**FEATURES**

- Same pin-layout as conventional relay (MR301 Series)
- 70% less relay volume than conventional relay (MR301 Series)
- 80% less relay space than conventional relay (MR301 Series)
- 90% less relay height than conventional relay (MR301 Sereis)
- 60% less relay weight than conventional relay (MR301 Sereis)
- Contact switching current of 30A max.
- Flux tight housing
- Delivered in stick-tube for automatic insertion machine
- Washable type available

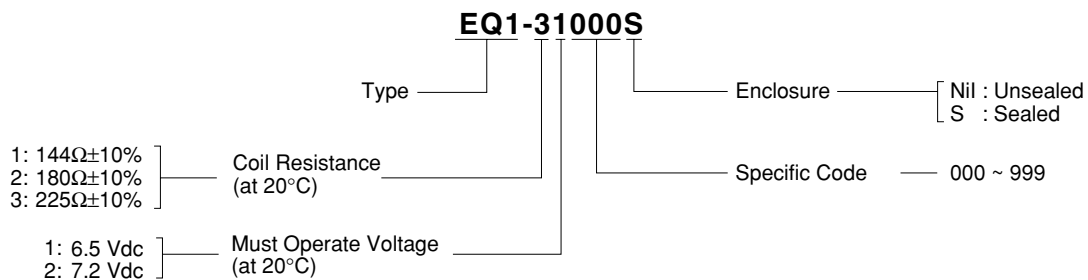


**PART NUMBERS AND COIL RATINGS**

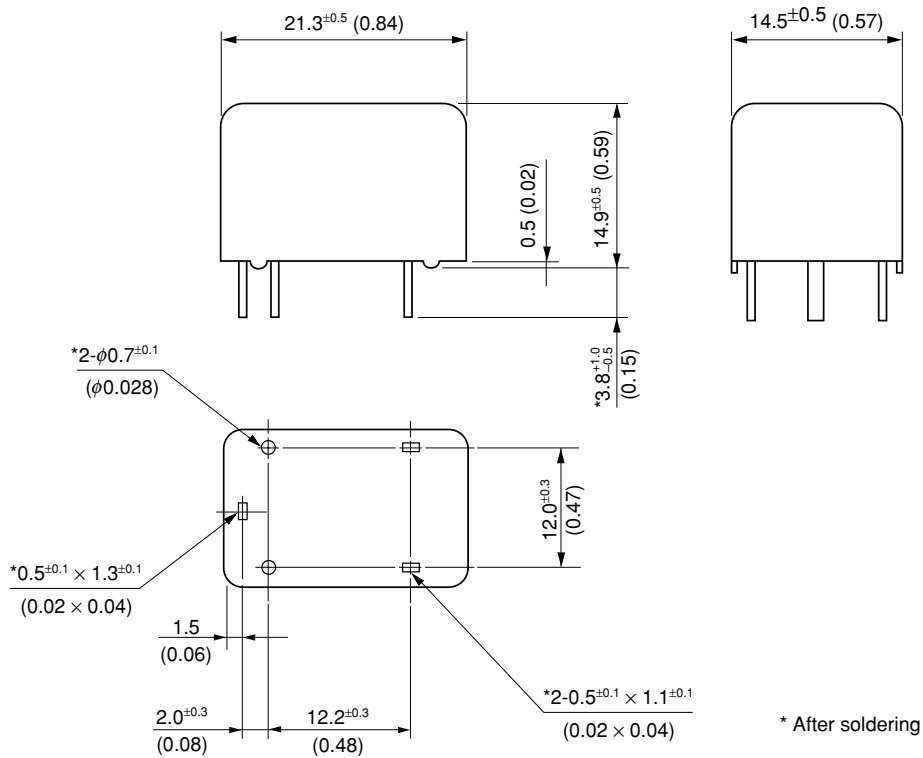
At 20°C (68°F)

Part Number	Nominal Voltage (Vdc)	Coil Resistance ( $\Omega \pm 10\%$ )	Nominal Current (mA)	Must Operate Voltage (Vdc)	Must Release Voltage (Vdc)	Nominal Operate Power (W)
EQ1-31000S	12	225	53.3	6.5	0.9	0.64
EQ1-11040S	12	144	83.3	6.5	0.6	1.00
EQ1-22111S	12	180	66.7	7.2	0.7	0.80
EQ1-11111S	12	144	83.3	6.5	0.6	1.00

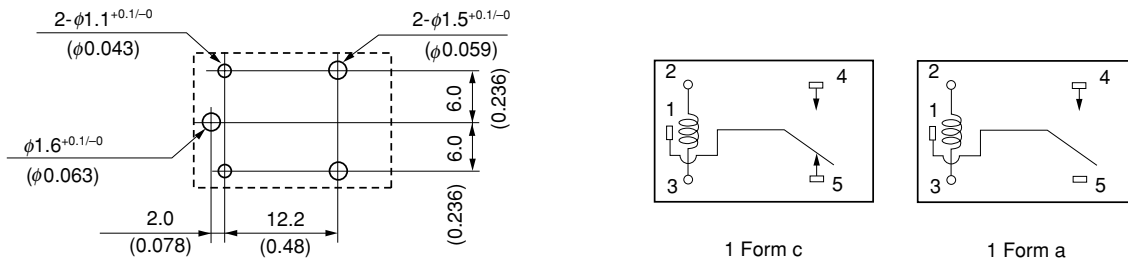
**PART NUMBER SYSTEM**



DIMENSIONS mm (inch)



PCB PAD LAYOUT and SCHEMATICS (bottom view) mm (inch)



SPECIFICATIONS

At 20°C (68°F)

Items	Specification			
	EQ1-31000S	EQ1-11040S	EQ1-11111S	EQ1-22111S
Contact Form	1 form c		1 form a	
Contact Material	Silver oxide complex alloy			
Contact Resistance	5 mΩ typical (measured at 1 A) initial			
Contact Switching Voltage	16 Vdc max. 5 Vdc min.			
Contact Switching Current	30 A max. (at 16 Vdc)			
Contact Carrying Current (2 minutes max.)	40 A (12 Vdc at 25°C)	35 A (12 Vdc at 25°C)	30 A (12 Vdc at 85°C)	
Operate Time	3 ms typical (at nominal voltage) initial			
Release Time	4 ms typical (at nominal voltage. with diode) initial			
Nominal Operate Power	640 mW	1000 mW		800 mW
Insulation Resistance	100 MΩ at 500 Vdc			
Breakdown Voltage	500 Vac min. for 1 minute			
Shock Resistance	98 m/s <sup>2</sup> min. [misoperating]			
Vibration Resistance	10 to 300 Hz, 43 m/s <sup>2</sup> min. [misoperating]			
Ambient Temperature	-40°C to +85°C (-40°F to +185°F)		-40°C to +125°C (-40°F to +257°F)	
Coil Temperature	70°C/W (contact carrying current 0 A)			
Life Expectancy	Mechanical	1 × 10 <sup>6</sup> operations		
	Electrical	1 × 10 <sup>5</sup> operations (at 14 Vdc, motor load 20 A/3 A)		1 × 10 <sup>5</sup> operations (at 14 Vdc. Lamp load 120 W)
Weight	Approx. 9 g			

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC/TOKIN Corporation. NEC/TOKIN Corporation assumes no responsibility for any errors which may appear in this document.

NEC/TOKIN Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC/TOKIN Corporation or others.

While NEC/TOKIN Corporation has been making continuous effort to enhance the reliability of its electronic components, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC/TOKIN electronic component, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features. NEC/TOKIN devices are classified into the following three quality grades:

"Standard," "Special," and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots

Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)

Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

The quality grade of NEC/TOKIN devices is "Standard" unless otherwise specified in NEC/TOKIN's Data Sheets or Data Books. If customers intend to use NEC/TOKIN devices for applications other than those specified for Standard quality grade, they should contact an NEC/TOKIN sales representative in advance.

(Note)

- (1) "NEC/TOKIN" as used in this statement means NEC/TOKIN Corporation and also includes its majority-owned subsidiaries.
- (2) "NEC/TOKIN electronic component products" means any electronic component product developed or manufactured by or for NEC/TOKIN (as defined above).

# Mouser Electronics

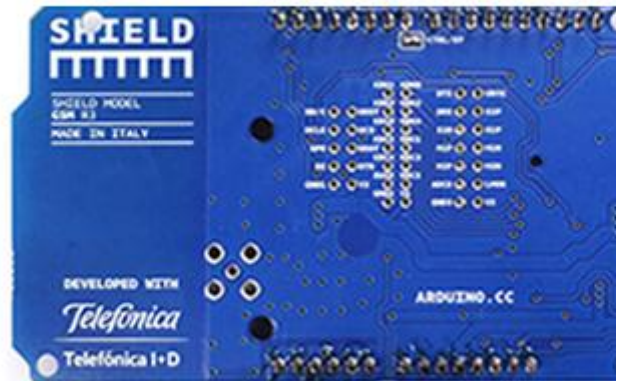
Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[NEC:](#)

[EQ1-31000S](#) [EQ1-22111S](#) [EQ1-11111S](#)

# Arduino GSM Shield



*Arduino GSM Shield Front*

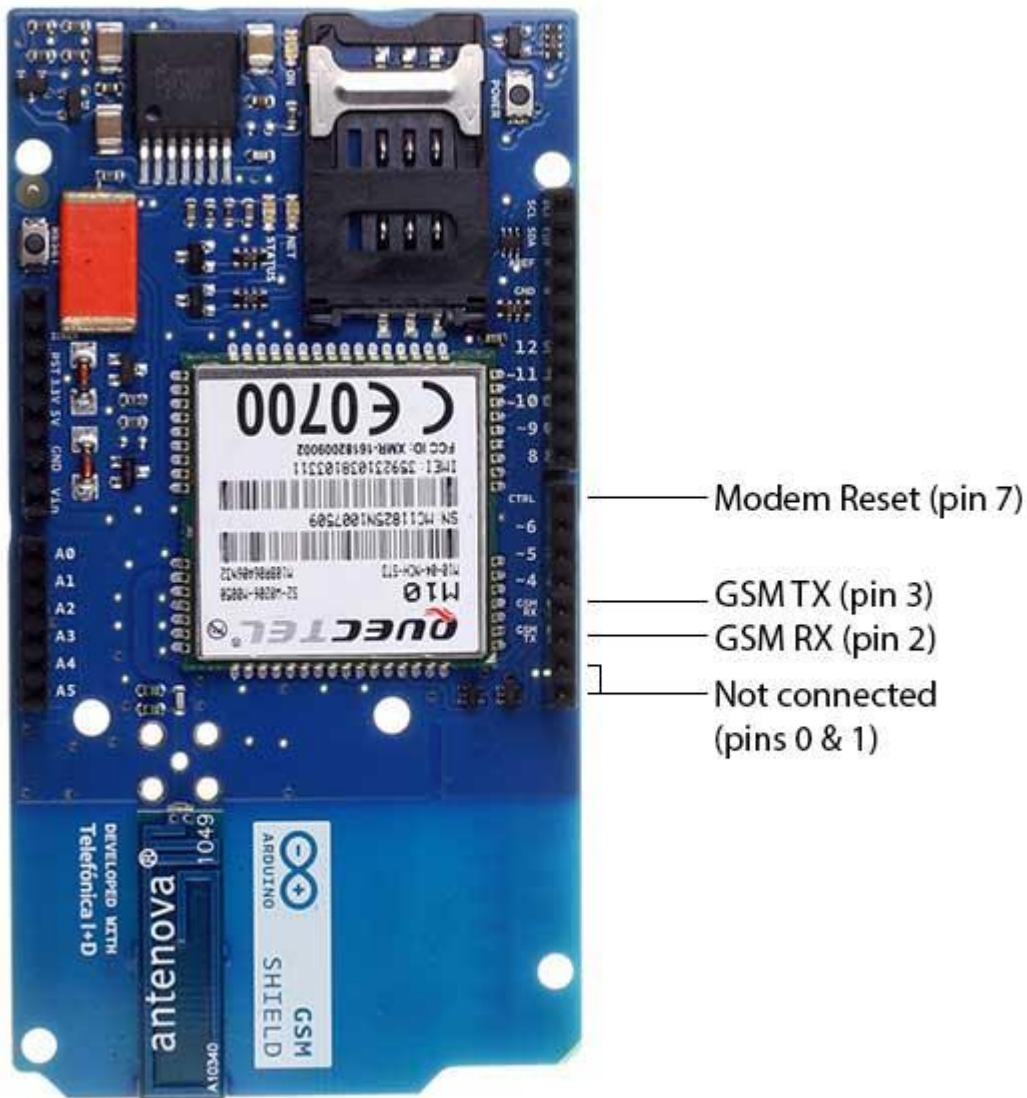
*Arduino GSM Shield Back*

**Download:** [PDF of GSM shield schematic](#), [Reference design](#)  
The [GSM library](#) is included with [Arduino IDE 1.0.4](#) and later.

## Overview

The Arduino GSM Shield connects your Arduino to the internet using the GPRS wireless network. Just plug this module onto your Arduino board, plug in a SIM card from an operator offering GPRS coverage and follow a few simple instructions to start controlling your world through the internet. You can also make/receive voice calls (you will need an external speaker and microphone circuit) and send/receive SMS messages.

As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Hundreds of thousands of Arduino boards are already fueling people's creativity all over the world, everyday. Join us now, Arduino is you!



- Requires an Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Connection with Arduino Uno on pins 2, 3 (Software Serial) and 7 (reset). [See these notes](#) for working with a Mega, Mega ADK, or Leonardo.

## Description

The Arduino GSM Shield allows an Arduino board to connect to the internet, make/receive voice calls and send/receive SMS messages. The shield uses a radio modem M10 by Quectel ([datasheet](#)). It is possible to communicate with the board using [AT commands](#). The [GSM library](#) has a large number of methods for communication with the shield.

The shield uses digital pins 2 and 3 for software serial communication with the M10. Pin 2 is connected to the M10's TX pin and pin 3 to its RX pin. [See these notes](#) for working with an Arduino Mega, Mega ADK, or Leonardo. The modem's PWRKEY pin is connected to Arduino pin 7.

The M10 is a Quad-band GSM/GPRS modem that works at frequencies GSM850MHz, GSM900MHz, DCS1800MHz and PCS1900MHz. It supports TCP/UDP and HTTP protocols through a GPRS connection. GPRS data downlink and uplink transfer speed maximum is 85.6 kbps.

To interface with the cellular network, the board requires a SIM card provided by a network operator. See the [getting started page](#) for additional information on SIM usage.

The most recent revision of the board uses the 1.0 pinout on rev 3 of the Arduino Uno board.

## Notes on the Telefonica/Bluevia SIM included with the shield

The GSM shield comes bundled with a SIM from Telefonica/Bluevia that will work well for developing machine to machine (M2M) applications. It is not necessary to use this specific card with the shield. You may use any SIM that works on a network in your area.

The Bluevia SIM card includes a roaming plan. It can be used on any supported GSM network. There is coverage throughout the Americas and Europe for this SIM, check the [Bluevia service availability page](#) for specific countries that have supported networks.

Activation of the SIM is handled by Bluevia. Detailed instructions on how to register and activate your SIM online and add credit are included on a [small pamphlet](#) that comes with your shield. The SIM must be inserted into a powered GSM shield that is mounted on an Arduino for activation.

These SIM card come without a PIN, but it is possible to set one using the GSM library's [GSMPIN class](#). You cannot use the included SIM to place or receive voice calls.

You can only place and receive SMS with other SIMs on the Bluevia network.

It's not possible to create a server that accepts incoming requests from the public internet. However, the Bluevia SIM will accept incoming requests from other SIM cards on the Bluevia network.

For using the voice, and other functions of the shield, you'll need to find a different network provider and SIM. Operators will have different policies for their SIM cards, check with them directly to determine what types of connections are supported.

## **Power requirements**

It is recommended that the board be powered with an external power supply that can provide between 700mA and 1000mA. Powering an Arduino and the GSM shield from a USB connection is not recommended, as USB cannot provide the required current for when the modem is in heavy use. The modem can pull up to 2A of current at peak usage, which can occur during data transmission. This current is provided through the large orange capacitor on the board's surface.

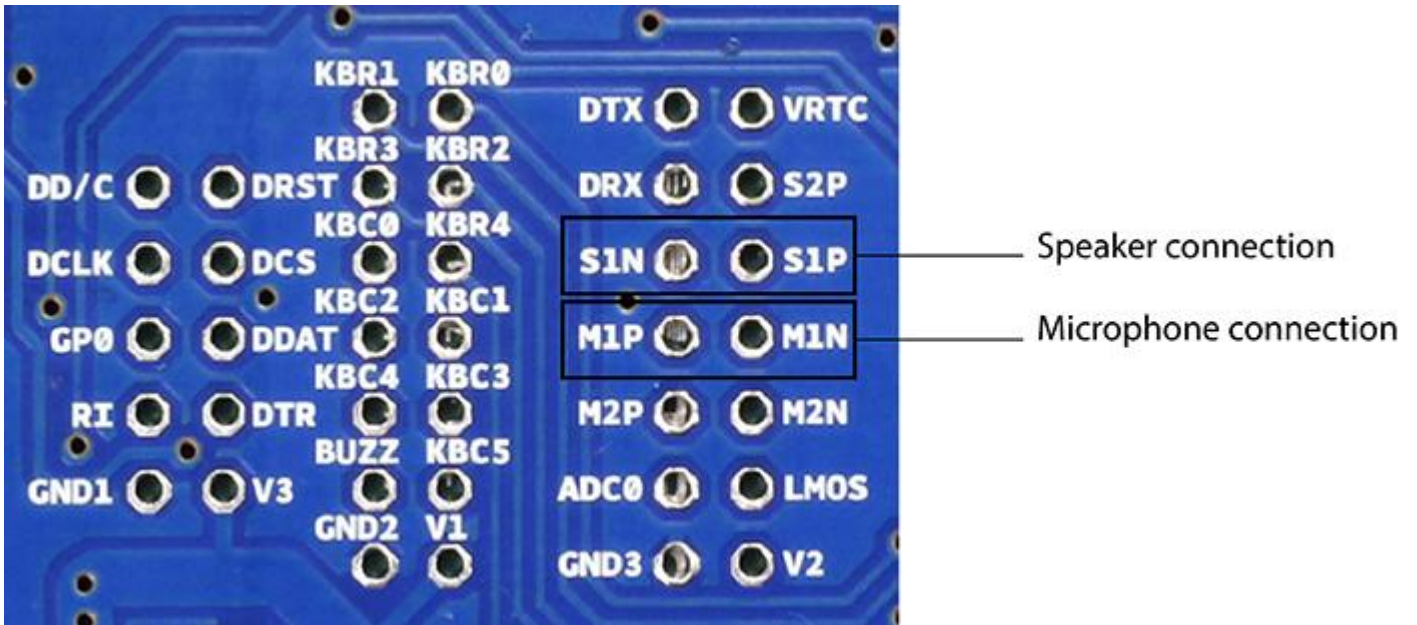
## **On board indicators**

The shield contains a number of status LEDs:

- On: shows the Shield gets power.
- Status: turns on to when the modem is powered and data is being transferred to/from the GSM/GPRS network.
- Net: blinks when the modem is communicating with the radio network.

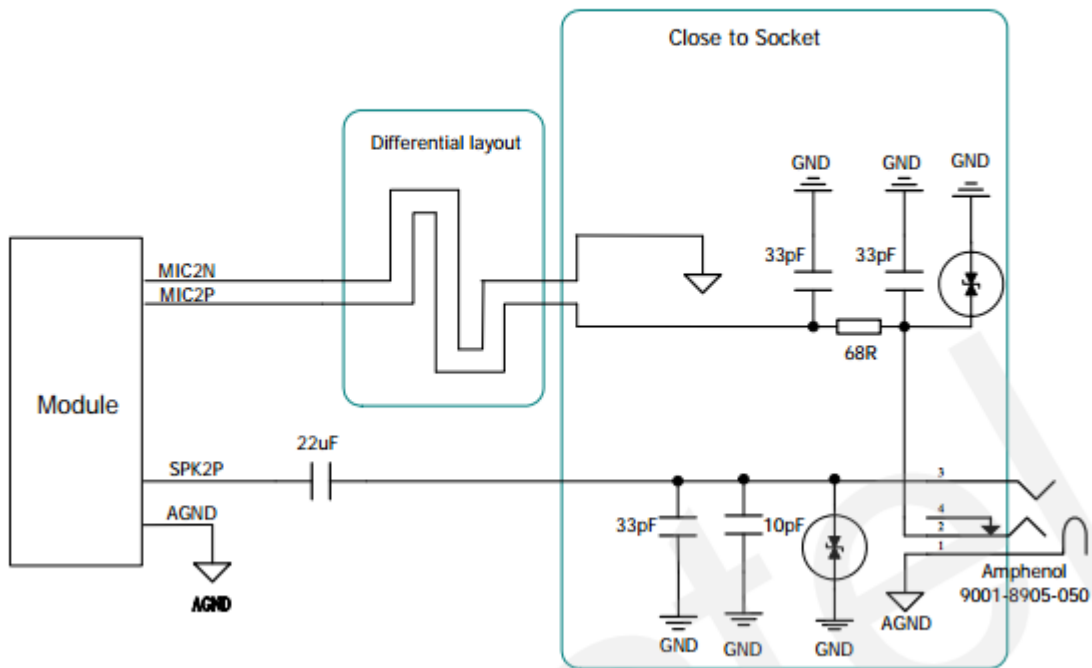
## **On board interfaces**

The shield supports AIN1 and AOUT1 as audio interfaces; an analog input channel and an analog output channel. The input, exposed on pins MIC1P/MIC1N, can be used for both microphone and line inputs. An electret microphone can be used for this interface. The output, exposed as lines SPK1P/SPK1N, can be used with either a receiver or speaker. Through the modem, it is possible to make voice calls. In order to speak to and hear the other party, you will need to add a speaker and microphone.

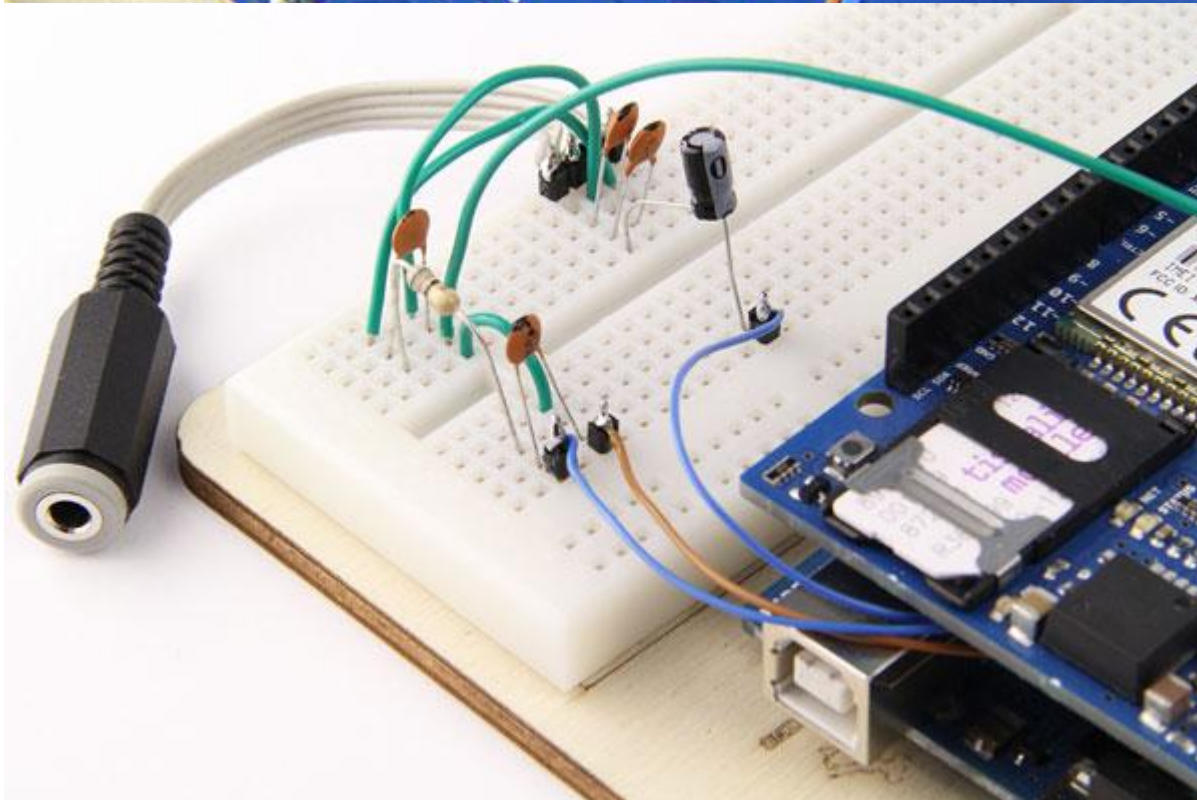
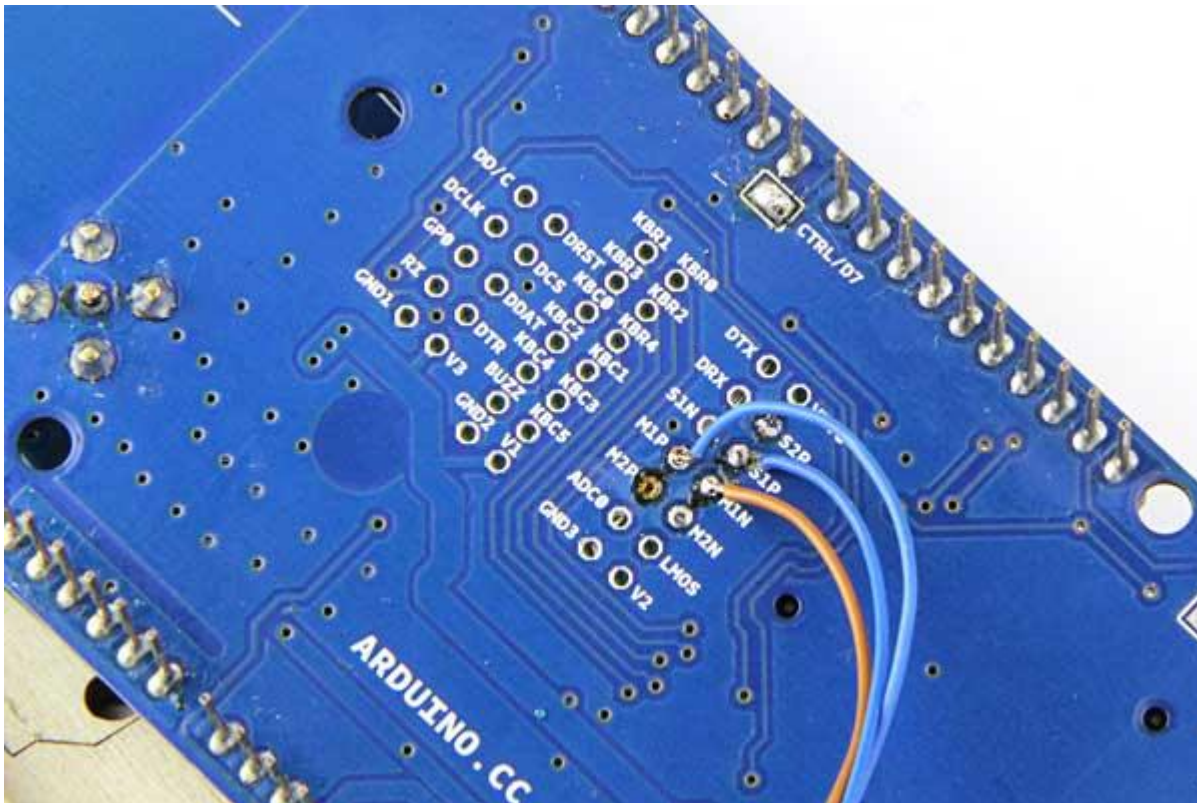


On page 43 of the [modem documentation](#), there is an example voice and sound circuit that will connect to an earphone:

### 3.9.3 Earphone interface configuration







There are two small buttons on the shield. The button labelled "Reset" is tied to the Arduino reset pin. When pressed, it will restart the sketch. The button labelled "Power" is connected to the modem and will power the modem on and off. For early versions of the shield, it was necessary to press the *power* button to turn on the modem. Newer versions of the board will turn the modem on automatically. If you have an early version of the shield, and it does not turn on automatically, you can solder a jumper to the CTRL/D7 pad on the reverse side of the board, and it will turn on when an attached Arduino receives power.

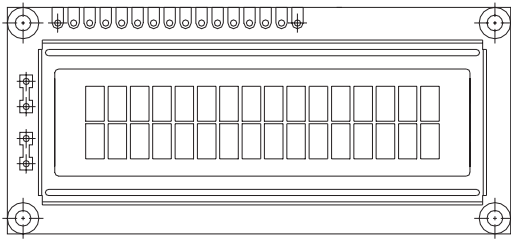
Bridged pin to automatically power the modem



Several of the modem pins are exposed on the underside of the board. These provide access to the modem for features like speaker output and microphone input. See the datasheet for complete information.



# 16 x 2 Character LCD



## FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	-	7.0	V
Input Voltage	VI	- 0.3	-	VDD	V

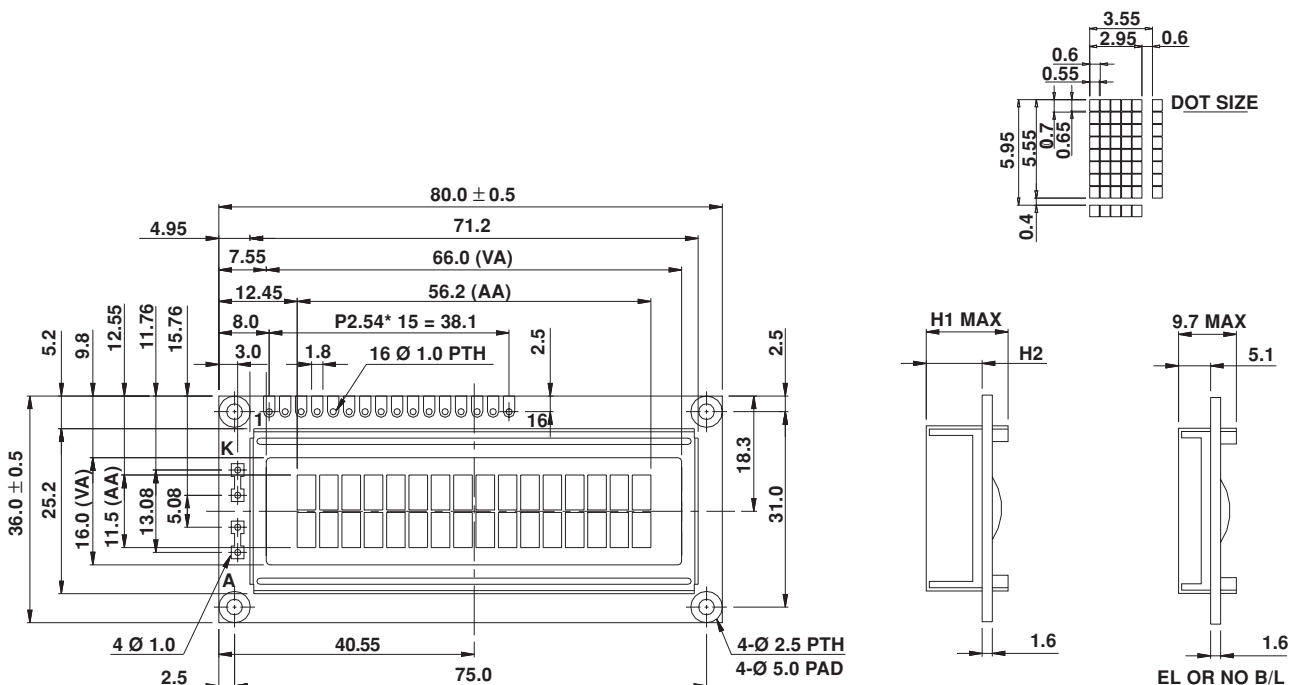
NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS							
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT	
			MIN.	TYP.	MAX.		
Input Voltage	VDD	VDD = + 5V	4.7	5.0	5.3	V	
		VDD = + 3V	2.7	3.0	5.3	V	
Supply Current	IDD	VDD = 5V	-	1.2	3.0	mA	
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	- 20 °C	-	-	-	V	
		0°C	4.2	4.8	5.1		
		25°C	3.8	4.2	4.6		
		50°C	3.6	4.0	4.4		
LED Forward Voltage	VF	25°C	-	4.2	4.6	V	
LED Forward Current	IF	25°C	Array	-	130	260	mA
			Edge	-	20	40	
EL Power Supply Current	IEL	Vel = 110VAC:400Hz	-	-	5.0	mA	

DISPLAY CHARACTER ADDRESS CODE:																
Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+ 3V or + 5V
3	Vo	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/W	H/L Read/Write Signal
6	E	H → L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/Vee	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (OV)

## DIMENSIONS in millimeters



LED - H/L B/L		
	HIGH	LOW
H1	13.2	12.1
H2	8.6	7.5

This datasheet has been download from:

[www.datasheetcatalog.com](http://www.datasheetcatalog.com)

Datasheets for electronics components.



# MFRC522

## Standard performance MIFARE and NTAG frontend

Rev. 3.9 — 27 April 2016  
112139

Product data sheet  
COMPANY PUBLIC

## 1. Introduction

---

This document describes the functionality and electrical specifications of the contactless reader/writer MFRC522.

**Remark:** The MFRC522 supports all variants of the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus RF identification protocols. To aid readability throughout this data sheet, the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus products and protocols have the generic name MIFARE.

### 1.1 Differences between version 1.0 and 2.0

The MFRC522 is available in two versions:

- MFRC52201HN1, hereafter referred to version 1.0 and
- MFRC52202HN1, hereafter referred to version 2.0.

The MFRC522 version 2.0 is fully compatible to version 1.0 and offers in addition the following features and improvements:

- Increased stability of the reader IC in rough conditions
- An additional timer prescaler, see [Section 8.5](#).
- A corrected CRC handling when RX Multiple is set to 1

This data sheet version covers both versions of the MFRC522 and describes the differences between the versions if applicable.

## 2. General description

---

The MFRC522 is a highly integrated reader/writer IC for contactless communication at 13.56 MHz. The MFRC522 reader supports ISO/IEC 14443 A/MIFARE and NTAG.

The MFRC522's internal transmitter is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443 A/MIFARE cards and transponders without additional active circuitry. The receiver module provides a robust and efficient implementation for demodulating and decoding signals from ISO/IEC 14443 A/MIFARE compatible cards and transponders. The digital module manages the complete ISO/IEC 14443 A framing and error detection (parity and CRC) functionality.

The MFRC522 supports MF1xxS20, MF1xxS70 and MF1xxS50 products. The MFRC522 supports contactless communication and uses MIFARE higher transfer speeds up to 848 kBd in both directions.



The following host interfaces are provided:

- Serial Peripheral Interface (SPI)
- Serial UART (similar to RS232 with voltage levels dependant on pin voltage supply)
- I<sup>2</sup>C-bus interface

### 3. Features and benefits

- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers for connecting an antenna with the minimum number of external components
- Supports ISO/IEC 14443 A/MIFARE and NTAG
- Typical operating distance in Read/Write mode up to 50 mm depending on the antenna size and tuning
- Supports MF1xxS20, MF1xxS70 and MF1xxS50 encryption in Read/Write mode
- Supports ISO/IEC 14443 A higher transfer speed communication up to 848 kBd
- Supports MFIN/MFOUT
- Additional internal power supply to the smart card IC connected via MFIN/MFOUT
- Supported host interfaces
  - ◆ SPI up to 10 Mbit/s
  - ◆ I<sup>2</sup>C-bus interface up to 400 kBd in Fast mode, up to 3400 kBd in High-speed mode
  - ◆ RS232 Serial UART up to 1228.8 kBd, with voltage levels dependant on pin voltage supply
- FIFO buffer handles 64 byte send and receive
- Flexible interrupt modes
- Hard reset with low power function
- Power-down by software mode
- Programmable timer
- Internal oscillator for connection to 27.12 MHz quartz crystal
- 2.5 V to 3.3 V power supply
- CRC coprocessor
- Programmable I/O pins
- Internal self-test

### 4. Quick reference data

Table 1. Quick reference data

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V <sub>DDA</sub>	analog supply voltage	V <sub>DD(PVDD)</sub> ≤ V <sub>DDA</sub> = V <sub>DDD</sub> = V <sub>DD(TVDD)</sub> ; V <sub>SSA</sub> = V <sub>SDD</sub> = V <sub>SS(PVSS)</sub> = V <sub>SS(TVSS)</sub> = 0 V	[1][2]	2.5	3.3	3.6	V
V <sub>DDD</sub>	digital supply voltage			2.5	3.3	3.6	V
V <sub>DD(TVDD)</sub>	TVDD supply voltage			2.5	3.3	3.6	V
V <sub>DD(PVDD)</sub>	PVDD supply voltage		[3]	1.6	1.8	3.6	V
V <sub>DD(SVDD)</sub>	SVDD supply voltage	V <sub>SSA</sub> = V <sub>SDD</sub> = V <sub>SS(PVSS)</sub> = V <sub>SS(TVSS)</sub> = 0 V		1.6	-	3.6	V

Table 1. Quick reference data ...continued

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	
I <sub>pd</sub>	power-down current	V <sub>D<sub>DA</sub></sub> = V <sub>D<sub>DD</sub></sub> = V <sub>D<sub>D(TVDD)</sub></sub> = V <sub>D<sub>D(PVDD)</sub></sub> = 3 V					
		hard power-down; pin NRSTPD set LOW	[4]	-	-	5	μA
		soft power-down; RF level detector on	[4]	-	-	10	μA
I <sub>DDD</sub>	digital supply current	pin DVDD; V <sub>D<sub>DD</sub></sub> = 3 V	-	6.5	9	mA	
I <sub>D<sub>DA</sub></sub>	analog supply current	pin AVDD; V <sub>D<sub>DA</sub></sub> = 3 V, CommandReg register's RcvOff bit = 0	-	7	10	mA	
		pin AVDD; receiver switched off; V <sub>D<sub>DA</sub></sub> = 3 V, CommandReg register's RcvOff bit = 1	-	3	5	mA	
I <sub>D<sub>D(PVDD)</sub></sub>	PVDD supply current	pin PVDD	[5]	-	40	mA	
I <sub>D<sub>D(TVDD)</sub></sub>	TVDD supply current	pin TVDD; continuous wave	[6][7][8]	-	60	100	mA
T <sub>amb</sub>	ambient temperature	HVQFN32	-25	-	+85	°C	

- [1] Supply voltages below 3 V reduce the performance in, for example, the achievable operating distance.
- [2] V<sub>D<sub>DA</sub></sub>, V<sub>D<sub>DD</sub></sub> and V<sub>D<sub>D(TVDD)</sub></sub> must always be the same voltage.
- [3] V<sub>D<sub>D(PVDD)</sub></sub> must always be the same or lower voltage than V<sub>D<sub>DD</sub></sub>.
- [4] I<sub>pd</sub> is the total current for all supplies.
- [5] I<sub>D<sub>D(PVDD)</sub></sub> depends on the overall load at the digital pins.
- [6] I<sub>D<sub>D(TVDD)</sub></sub> depends on V<sub>D<sub>D(TVDD)</sub></sub> and the external circuit connected to pins TX1 and TX2.
- [7] During typical circuit operation, the overall current is below 100 mA.
- [8] Typical value using a complementary driver configuration and an antenna matched to 40 Ω between pins TX1 and TX2 at 13.56 MHz.

## 5. Ordering information

Table 2. Ordering information

Type number	Package		
	Name	Description	Version
MFRC52201HN1/TRAYB[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1
MFRC52201HN1/TRAYBM[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1
MFRC52202HN1/TRAYB[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1
MFRC52202HN1/TRAYBM[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOT617-1

- [1] Delivered in one tray.
- [2] Delivered in five trays.

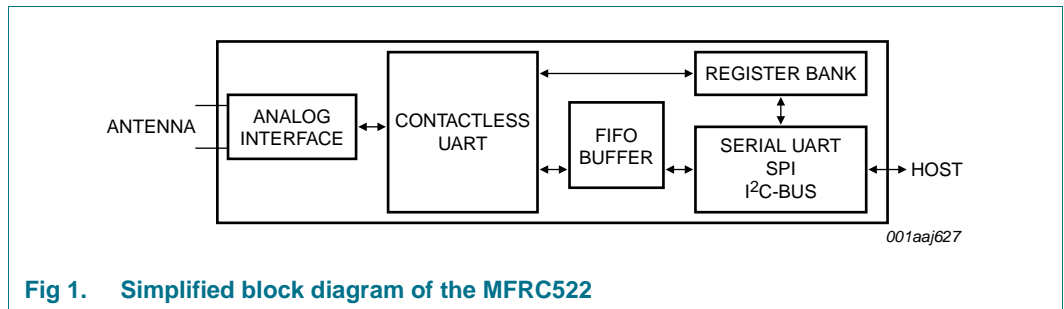


## 6. Block diagram

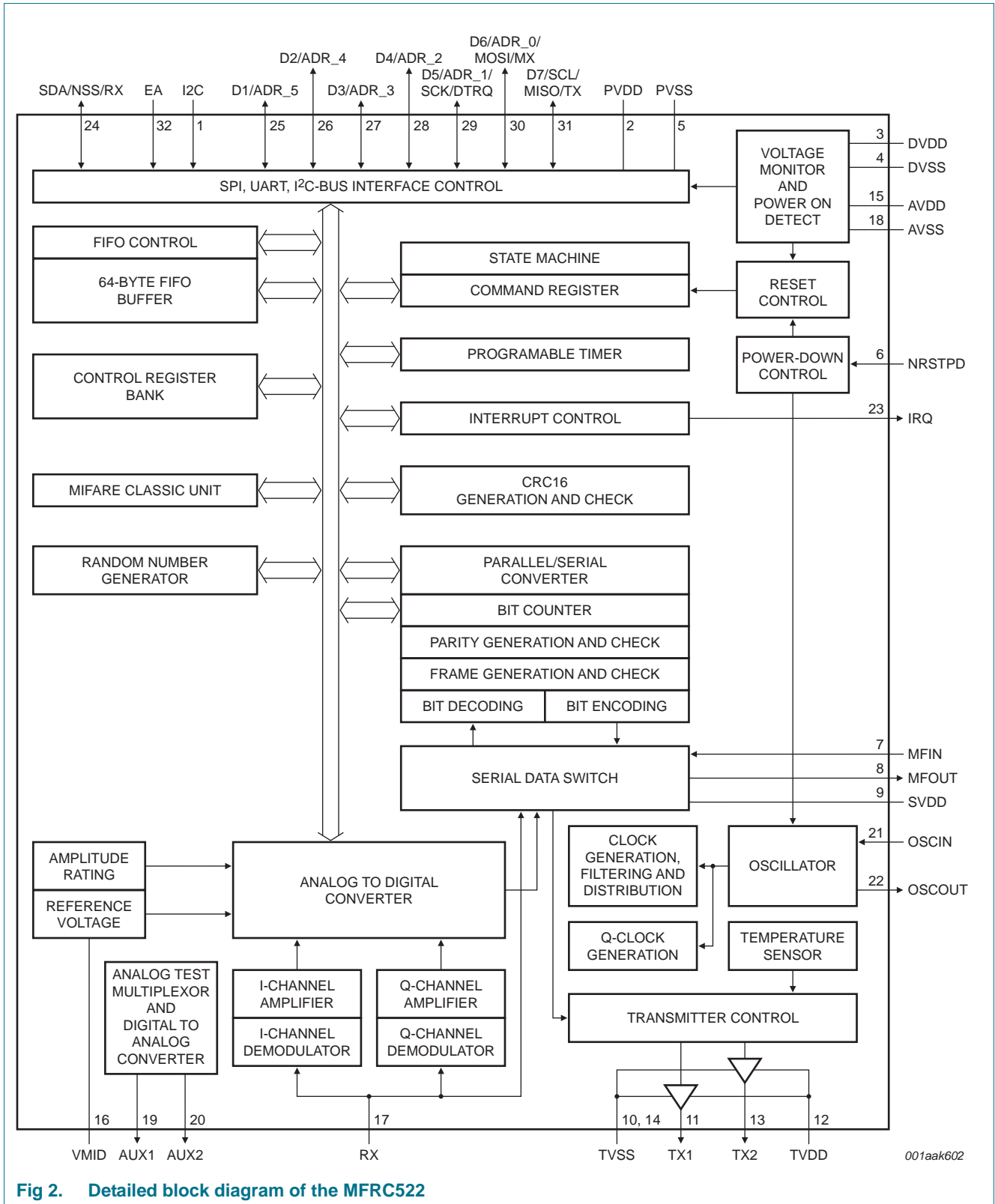
The analog interface handles the modulation and demodulation of the analog signals.

The contactless UART manages the protocol requirements for the communication protocols in cooperation with the host. The FIFO buffer ensures fast and convenient data transfer to and from the host and the contactless UART and vice versa.

Various host interfaces are implemented to meet different customer requirements.



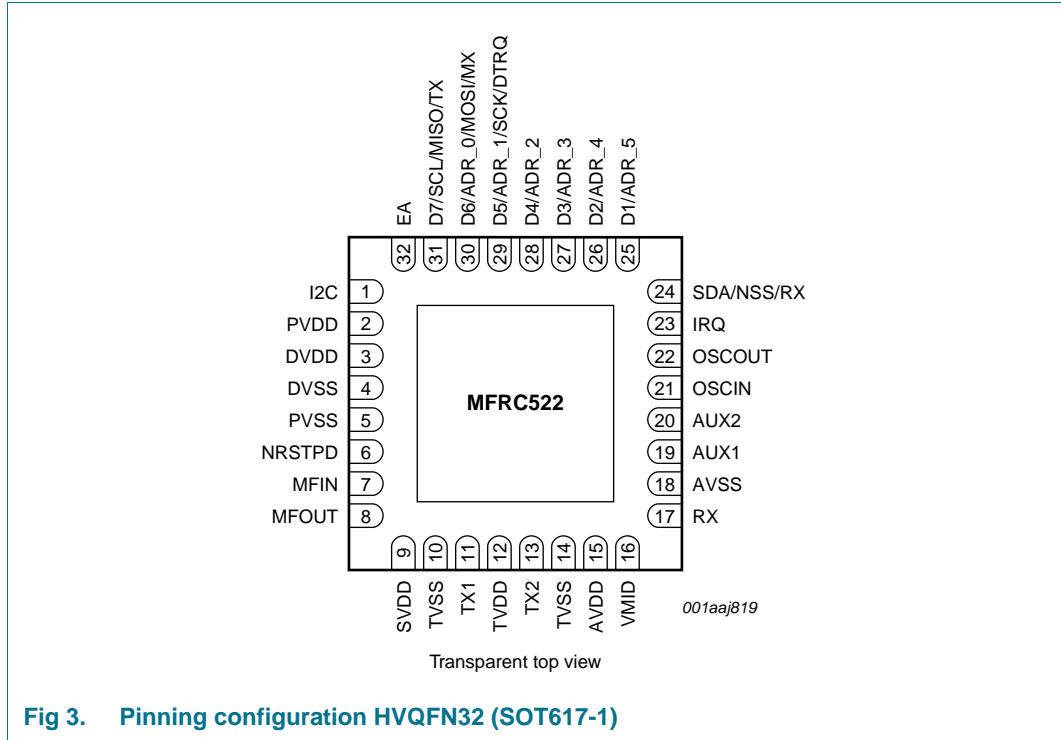
**Fig 1. Simplified block diagram of the MFRC522**



001aak602

**Fig 2. Detailed block diagram of the MFRC522**

## 7. Pinning information



**Fig 3. Pinning configuration HVQFN32 (SOT617-1)**

### 7.1 Pin description

**Table 3. Pin description**

Pin	Symbol	Type <sup>[1]</sup>	Description
1	I2C	I	I <sup>2</sup> C-bus enable input <sup>[2]</sup>
2	PVDD	P	pin power supply
3	DVDD	P	digital power supply
4	DVSS	G	digital ground <sup>[3]</sup>
5	PVSS	G	pin power supply ground
6	NRSTPD	I	reset and power-down input: power-down: enabled when LOW; internal current sinks are switched off, the oscillator is inhibited and the input pins are disconnected from the outside world reset: enabled by a positive edge
7	MFIN	I	MIFARE signal input
8	MFOUT	O	MIFARE signal output
9	SVDD	P	MFIN and MFOUT pin power supply
10	TVSS	G	transmitter output stage 1 ground
11	TX1	O	transmitter 1 modulated 13.56 MHz energy carrier output
12	TVDD	P	transmitter power supply: supplies the output stage of transmitters 1 and 2
13	TX2	O	transmitter 2 modulated 13.56 MHz energy carrier output
14	TVSS	G	transmitter output stage 2 ground
15	AVDD	P	analog power supply

Table 3. Pin description ...continued

Pin	Symbol	Type <sup>[1]</sup>	Description
16	VMID	P	internal reference voltage
17	RX	I	RF signal input
18	AVSS	G	analog ground
19	AUX1	O	auxiliary outputs for test purposes
20	AUX2	O	auxiliary outputs for test purposes
21	OSCIN	I	crystal oscillator inverting amplifier input; also the input for an externally generated clock ( $f_{clk} = 27.12$ MHz)
22	OSCOUT	O	crystal oscillator inverting amplifier output
23	IRQ	O	interrupt request output: indicates an interrupt event
24	SDA	I/O	I <sup>2</sup> C-bus serial data line input/output <sup>[2]</sup>
	NSS	I	SPI signal input <sup>[2]</sup>
	RX	I	UART address input <sup>[2]</sup>
25	D1	I/O	test port <sup>[2]</sup>
	ADR_5	I/O	I <sup>2</sup> C-bus address 5 input <sup>[2]</sup>
26	D2	I/O	test port
	ADR_4	I	I <sup>2</sup> C-bus address 4 input <sup>[2]</sup>
27	D3	I/O	test port
	ADR_3	I	I <sup>2</sup> C-bus address 3 input <sup>[2]</sup>
28	D4	I/O	test port
	ADR_2	I	I <sup>2</sup> C-bus address 2 input <sup>[2]</sup>
29	D5	I/O	test port
	ADR_1	I	I <sup>2</sup> C-bus address 1 input <sup>[2]</sup>
	SCK	I	SPI serial clock input <sup>[2]</sup>
	DTRQ	O	UART request to send output to microcontroller <sup>[2]</sup>
30	D6	I/O	test port
	ADR_0	I	I <sup>2</sup> C-bus address 0 input <sup>[2]</sup>
	MOSI	I/O	SPI master out, slave in <sup>[2]</sup>
	MX	O	UART output to microcontroller <sup>[2]</sup>
31	D7	I/O	test port
	SCL	I/O	I <sup>2</sup> C-bus clock input/output <sup>[2]</sup>
	MISO	I/O	SPI master in, slave out <sup>[2]</sup>
	TX	O	UART data output to microcontroller <sup>[2]</sup>
32	EA	I	external address input for coding I <sup>2</sup> C-bus address <sup>[2]</sup>

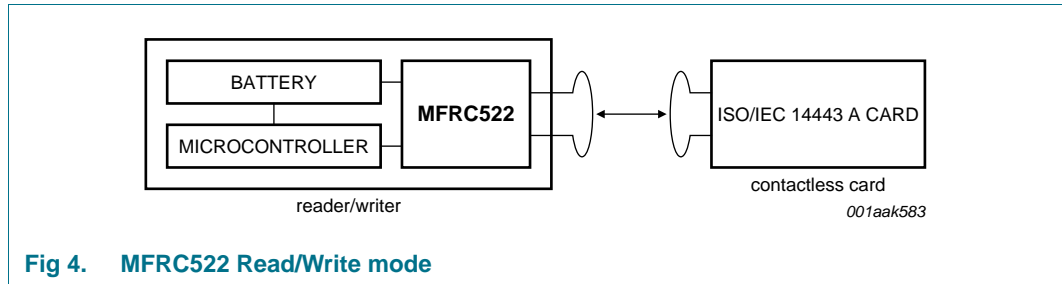
[1] Pin types: I = Input, O = Output, I/O = Input/Output, P = Power and G = Ground.

[2] The pin functionality of these pins is explained in [Section 8.1 "Digital interfaces"](#).

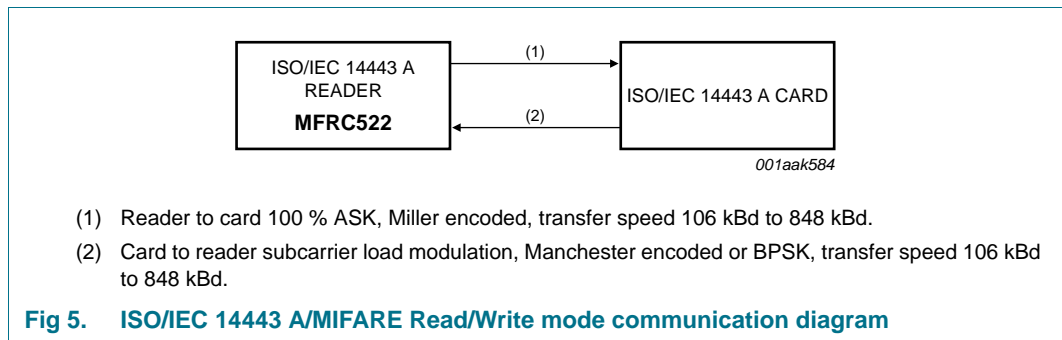
[3] Connection of heatsink pad on package bottom side is not necessary. Optional connection to pin DVSS is possible.

## 8. Functional description

The MFRC522 transmission module supports the Read/Write mode for ISO/IEC 14443 A/MIFARE using various transfer speeds and modulation protocols.



The physical level communication is shown in [Figure 5](#).

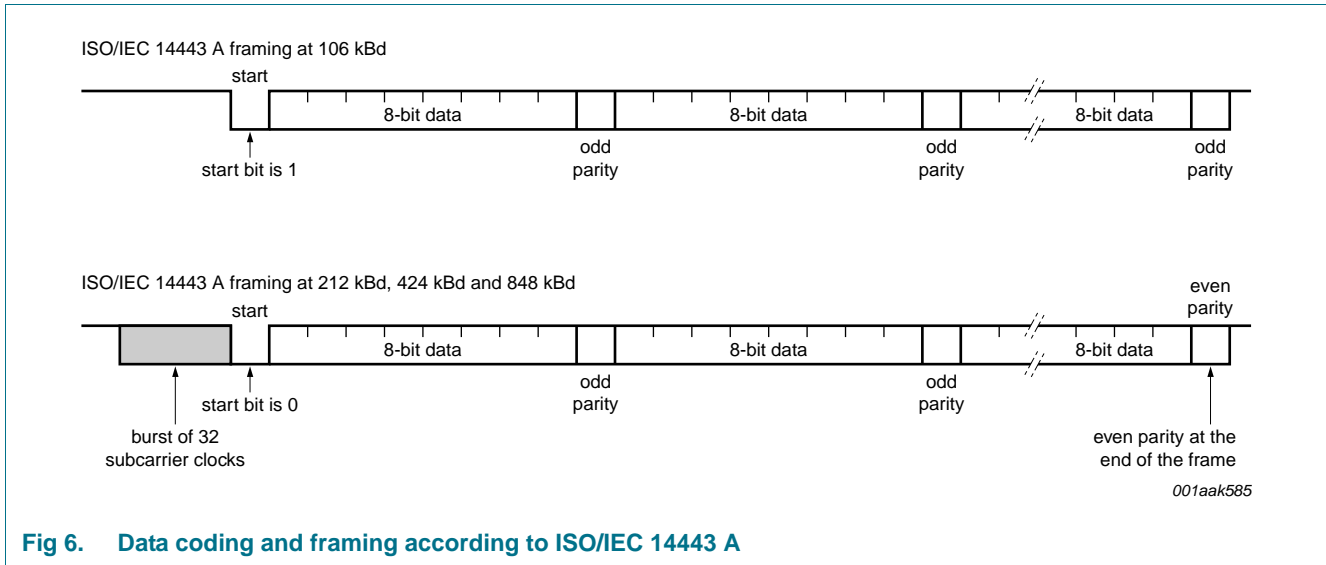


The physical parameters are described in [Table 4](#).

**Table 4. Communication overview for ISO/IEC 14443 A/MIFARE reader/writer**

Communication direction	Signal type	Transfer speed			
		106 kBd	212 kBd	424 kBd	848 kBd
Reader to card (send data from the MFRC522 to a card)	reader side modulation	100 % ASK	100 % ASK	100 % ASK	100 % ASK
	bit encoding	modified Miller encoding	modified Miller encoding	modified Miller encoding	modified Miller encoding
	bit length	128 (13.56 μs)	64 (13.56 μs)	32 (13.56 μs)	16 (13.56 μs)
Card to reader (MFRC522 receives data from a card)	card side modulation	subcarrier load modulation	subcarrier load modulation	subcarrier load modulation	subcarrier load modulation
	subcarrier frequency	13.56 MHz / 16	13.56 MHz / 16	13.56 MHz / 16	13.56 MHz / 16
	bit encoding	Manchester encoding	BPSK	BPSK	BPSK

The MFRC522's contactless UART and dedicated external host must manage the complete ISO/IEC 14443 A/MIFARE protocol. [Figure 6](#) shows the data coding and framing according to ISO/IEC 14443 A/MIFARE.



The internal CRC coprocessor calculates the CRC value based on ISO/IEC 14443 A part 3 and handles parity generation internally according to the transfer speed. Automatic parity generation can be switched off using the MfRxReg register's ParityDisable bit.

## 8.1 Digital interfaces

### 8.1.1 Automatic microcontroller interface detection

The MFRC522 supports direct interfacing of hosts using SPI, I<sup>2</sup>C-bus or serial UART interfaces. The MFRC522 resets its interface and checks the current host interface type automatically after performing a power-on or hard reset. The MFRC522 identifies the host interface by sensing the logic levels on the control pins after the reset phase. This is done using a combination of fixed pin connections. [Table 5](#) shows the different connection configurations.

**Table 5. Connection protocol for detecting different interface types**

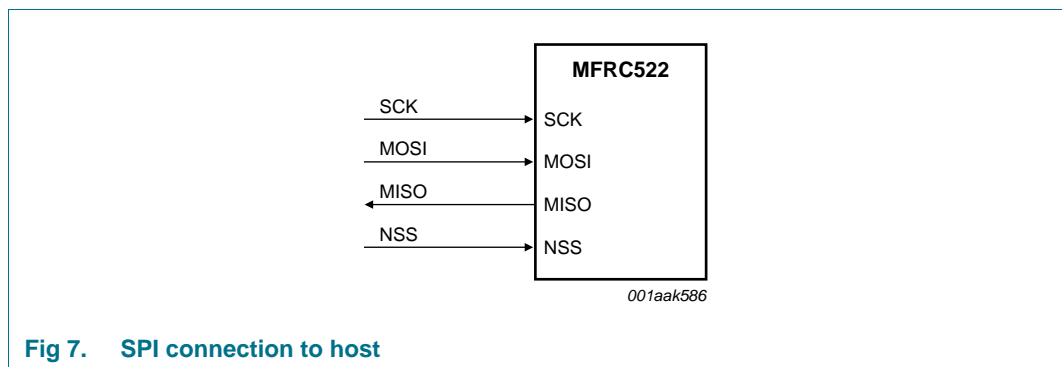
Pin	Interface type		
	UART (input)	SPI (output)	I <sup>2</sup> C-bus (I/O)
SDA	RX	NSS	SDA
I2C	0	0	1
EA	0	1	EA
D7	TX	MISO	SCL
D6	MX	MOSI	ADR_0
D5	DTRQ	SCK	ADR_1
D4	-	-	ADR_2
D3	-	-	ADR_3
D2	-	-	ADR_4
D1	-	-	ADR_5

**8.1.2 Serial Peripheral Interface**

A serial peripheral interface (SPI compatible) is supported to enable high-speed communication to the host. The interface can handle data speeds up to 10 Mbit/s. When communicating with a host, the MFRC522 acts as a slave, receiving data from the external host for register settings, sending and receiving data relevant for RF interface communication.

An interface compatible with SPI enables high-speed serial communication between the MFRC522 and a microcontroller. The implemented interface is in accordance with the SPI standard.

The timing specification is given in [Section 14.1 on page 78](#).



The MFRC522 acts as a slave during SPI communication. The SPI clock signal SCK must be generated by the master. Data communication from the master to the slave uses the MOSI line. The MISO line is used to send data from the MFRC522 to the master.

Data bytes on both MOSI and MISO lines are sent with the MSB first. Data on both MOSI and MISO lines must be stable on the rising edge of the clock and can be changed on the falling edge. Data is provided by the MFRC522 on the falling clock edge and is stable during the rising clock edge.

**8.1.2.1 SPI read data**

Reading data using SPI requires the byte order shown in [Table 6](#) to be used. It is possible to read out up to n-data bytes.

The first byte sent defines both the mode and the address.

**Table 6. MOSI and MISO byte order**

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	address 1	address 2	...	address n	00
MISO	X <sup>[1]</sup>	data 0	data 1	...	data n – 1	data n

[1] X = Do not care.

**Remark:** The MSB must be sent first.

8.1.2.2 SPI write data

To write data to the MFRC522 using SPI requires the byte order shown in [Table 7](#). It is possible to write up to n data bytes by only sending one address byte.

The first send byte defines both the mode and the address byte.

Table 7. MOSI and MISO byte order

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	data 0	data 1	...	data n – 1	data n
MISO	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	...	X <sup>[1]</sup>	X <sup>[1]</sup>

[1] X = Do not care.

**Remark:** The MSB must be sent first.

8.1.2.3 SPI address byte

The address byte must meet the following format.

The MSB of the first byte defines the mode used. To read data from the MFRC522 the MSB is set to logic 1. To write data to the MFRC522 the MSB must be set to logic 0. Bits 6 to 1 define the address and the LSB is set to logic 0.

Table 8. Address byte 0 register; address MOSI

7 (MSB)	6	5	4	3	2	1	0 (LSB)
1 = read 0 = write	address						0

8.1.3 UART interface

8.1.3.1 Connection to a host

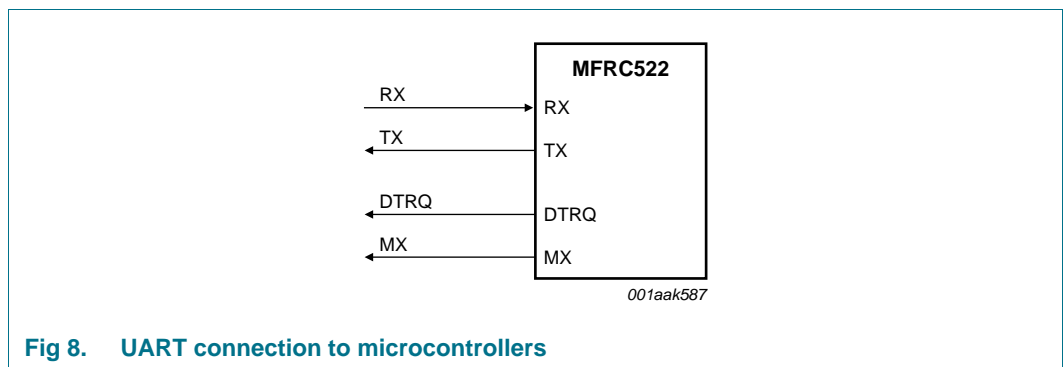


Fig 8. UART connection to microcontrollers

**Remark:** Signals DTRQ and MX can be disabled by clearing TestPinEnReg register's RS232LineEn bit.



### 8.1.3.2 Selectable UART transfer speeds

The internal UART interface is compatible with an RS232 serial interface.

The default transfer speed is 9.6 kBd. To change the transfer speed, the host controller must write a value for the new transfer speed to the SerialSpeedReg register. Bits BR\_T0[2:0] and BR\_T1[4:0] define the factors for setting the transfer speed in the SerialSpeedReg register.

The BR\_T0[2:0] and BR\_T1[4:0] settings are described in [Table 9](#). Examples of different transfer speeds and the relevant register settings are given in [Table 10](#).

**Table 9. BR\_T0 and BR\_T1 settings**

BR_Tn	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
BR_T0 factor	1	1	2	4	8	16	32	64
BR_T1 range	1 to 32	33 to 64	33 to 64	33 to 64	33 to 64	33 to 64	33 to 64	33 to 64

**Table 10. Selectable UART transfer speeds**

Transfer speed (kBd)	SerialSpeedReg value		Transfer speed accuracy (%) <sup>[1]</sup>
	Decimal	Hexadecimal	
7.2	250	FAh	-0.25
9.6	235	EBh	0.32
14.4	218	DAh	-0.25
19.2	203	CBh	0.32
38.4	171	ABh	0.32
57.6	154	9Ah	-0.25
115.2	122	7Ah	-0.25
128	116	74h	-0.06
230.4	90	5Ah	-0.25
460.8	58	3Ah	-0.25
921.6	28	1Ch	1.45
1228.8	21	15h	0.32

[1] The resulting transfer speed error is less than 1.5 % for all described transfer speeds.

The selectable transfer speeds shown in [Table 10](#) are calculated according to the following equations:

If BR\_T0[2:0] = 0:

$$transfer\ speed = \frac{27.12 \times 10^6}{(BR\_T0 + 1)} \tag{1}$$

If BR\_T0[2:0] > 0:

$$transfer\ speed = \left( \frac{27.12 \times 10^6}{\frac{(BR\_T1 + 33)}{2^{(BR\_T0 - 1)}}} \right) \tag{2}$$

**Remark:** Transfer speeds above 1228.8 kBd are not supported.

8.1.3.3 UART framing

Table 11. UART framing

Bit	Length	Value
Start	1-bit	0
Data	8 bits	data
Stop	1-bit	1

**Remark:** The LSB for data and address bytes must be sent first. No parity bit is used during transmission.

**Read data:** To read data using the UART interface, the flow shown in Table 12 must be used. The first byte sent defines both the mode and the address.

Table 12. Read data byte order

Pin	Byte 0	Byte 1
RX (pin 24)	address	-
TX (pin 31)	-	data 0

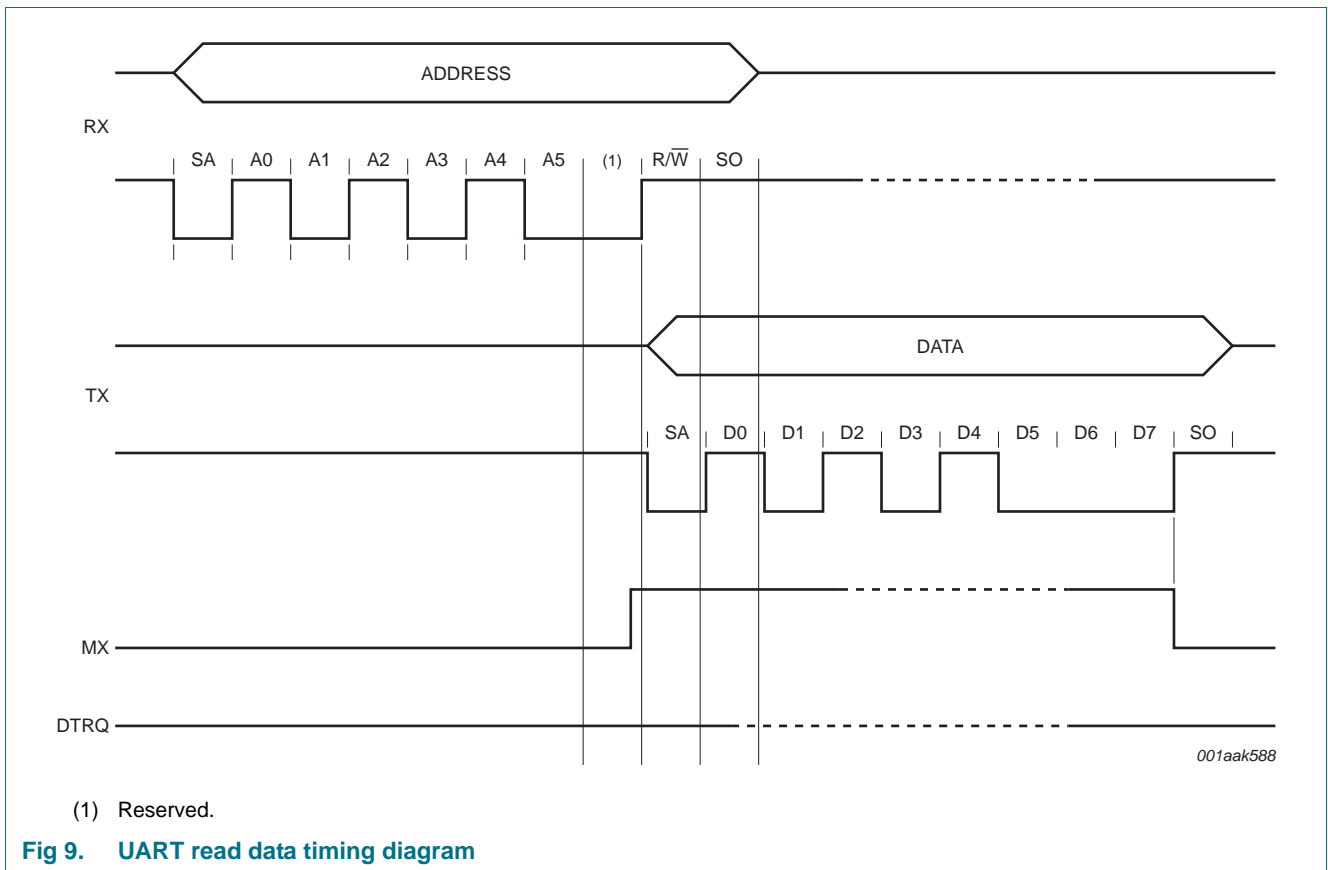


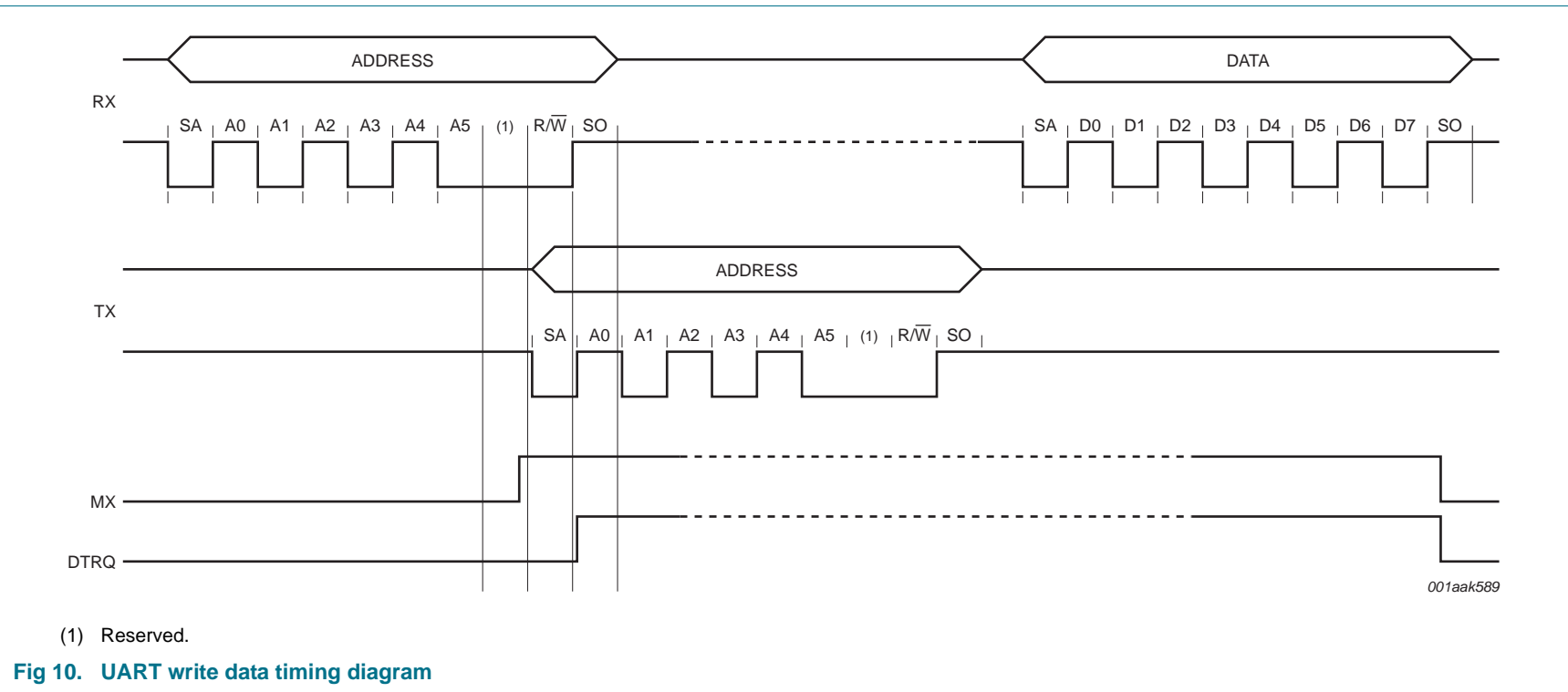
Fig 9. UART read data timing diagram

**Write data:** To write data to the MFRC522 using the UART interface, the structure shown in Table 13 must be used.

The first byte sent defines both the mode and the address.

Table 13. Write data byte order

Pin	Byte 0	Byte 1
RX (pin 24)	address 0	data 0
TX (pin 31)	-	address 0



**Fig 10. UART write data timing diagram**

**Remark:** The data byte can be sent directly after the address byte on pin RX.

**Address byte:** The address byte has to meet the following format:

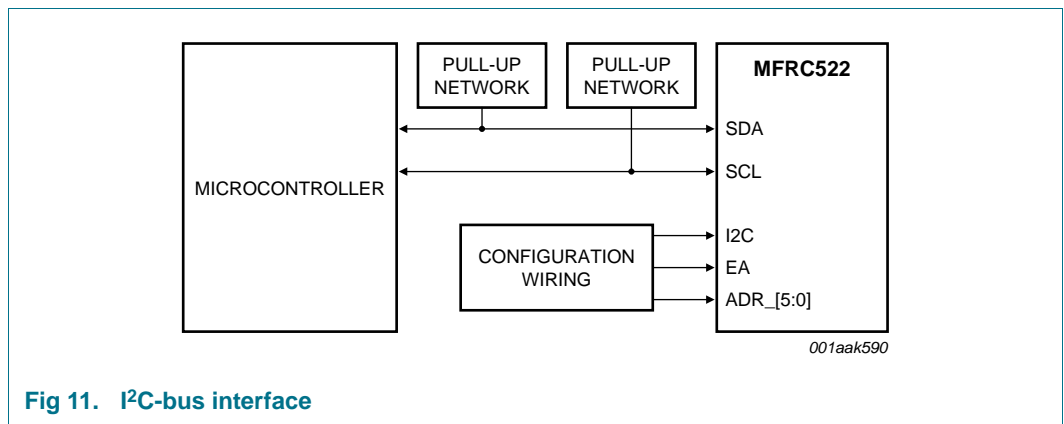
The MSB of the first byte sets the mode used. To read data from the MFRC522, the MSB is set to logic 1. To write data to the MFRC522 the MSB is set to logic 0. Bit 6 is reserved for future use, and bits 5 to 0 define the address; see [Table 14](#).

**Table 14. Address byte 0 register; address MOSI**

7 (MSB)	6	5	4	3	2	1	0 (LSB)
1 = read 0 = write	reserved	address					

**8.1.4 I<sup>2</sup>C-bus interface**

An I<sup>2</sup>C-bus (Inter-IC) interface is supported to enable a low-cost, low pin count serial bus interface to the host. The I<sup>2</sup>C-bus interface is implemented according to NXP Semiconductors' *I<sup>2</sup>C-bus interface specification, rev. 2.1, January 2000*. The interface can only act in Slave mode. Therefore the MFRC522 does not implement clock generation or access arbitration.



**Fig 11. I<sup>2</sup>C-bus interface**

The MFRC522 can act either as a slave receiver or slave transmitter in Standard mode, Fast mode and High-speed mode.

SDA is a bidirectional line connected to a positive supply voltage using a current source or a pull-up resistor. Both SDA and SCL lines are set HIGH when data is not transmitted. The MFRC522 has a 3-state output stage to perform the wired-AND function. Data on the I<sup>2</sup>C-bus can be transferred at data rates of up to 100 kBd in Standard mode, up to 400 kBd in Fast mode or up to 3.4 Mbit/s in High-speed mode.

If the I<sup>2</sup>C-bus interface is selected, spike suppression is activated on lines SCL and SDA as defined in the I<sup>2</sup>C-bus interface specification.

See [Table 155 on page 79](#) for timing requirements.

8.1.4.1 Data validity

Data on the SDA line must be stable during the HIGH clock period. The HIGH or LOW state of the data line must only change when the clock signal on SCL is LOW.

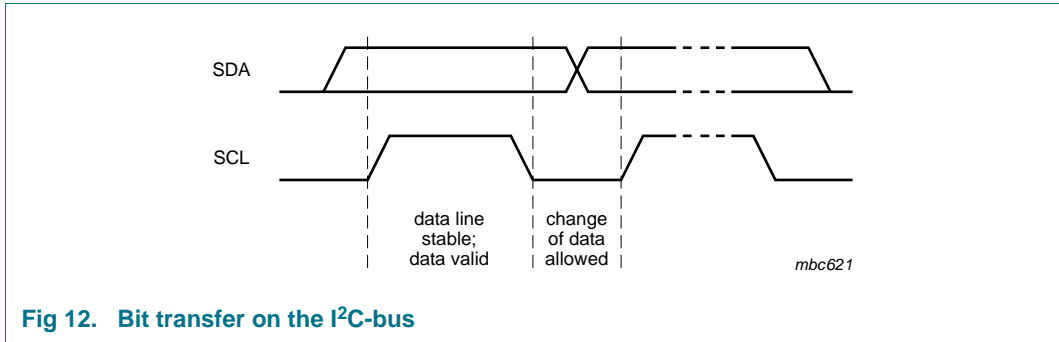


Fig 12. Bit transfer on the I<sup>2</sup>C-bus

8.1.4.2 START and STOP conditions

To manage the data transfer on the I<sup>2</sup>C-bus, unique START (S) and STOP (P) conditions are defined.

- A START condition is defined with a HIGH-to-LOW transition on the SDA line while SCL is HIGH.
- A STOP condition is defined with a LOW-to-HIGH transition on the SDA line while SCL is HIGH.

The I<sup>2</sup>C-bus master always generates the START and STOP conditions. The bus is busy after the START condition. The bus is free again a certain time after the STOP condition.

The bus stays busy if a repeated START (Sr) is generated instead of a STOP condition. The START (S) and repeated START (Sr) conditions are functionally identical. Therefore, S is used as a generic term to represent both the START (S) and repeated START (Sr) conditions.

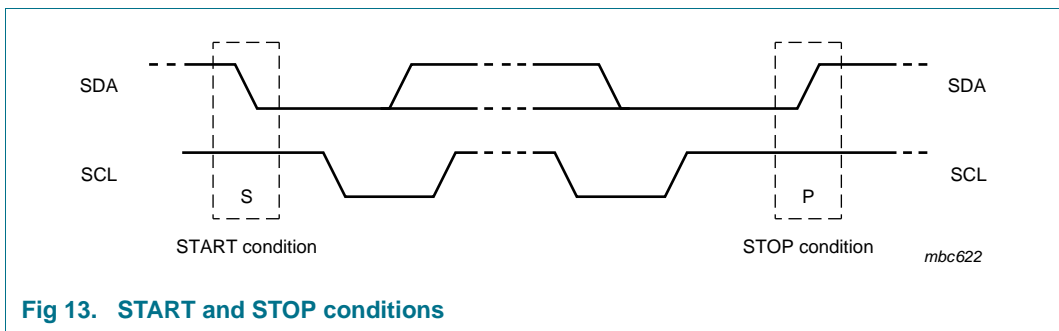


Fig 13. START and STOP conditions

8.1.4.3 Byte format

Each byte must be followed by an acknowledge bit. Data is transferred with the MSB first; see [Figure 16](#). The number of transmitted bytes during one data transfer is unrestricted but must meet the read/write cycle format.

8.1.4.4 Acknowledge

An acknowledge must be sent at the end of one data byte. The acknowledge-related clock pulse is generated by the master. The transmitter of data, either master or slave, releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver pulls down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse.

The master can then generate either a STOP (P) condition to stop the transfer or a repeated START (Sr) condition to start a new transfer.

A master-receiver indicates the end of data to the slave-transmitter by not generating an acknowledge on the last byte that was clocked out by the slave. The slave-transmitter releases the data line to allow the master to generate a STOP (P) or repeated START (Sr) condition.

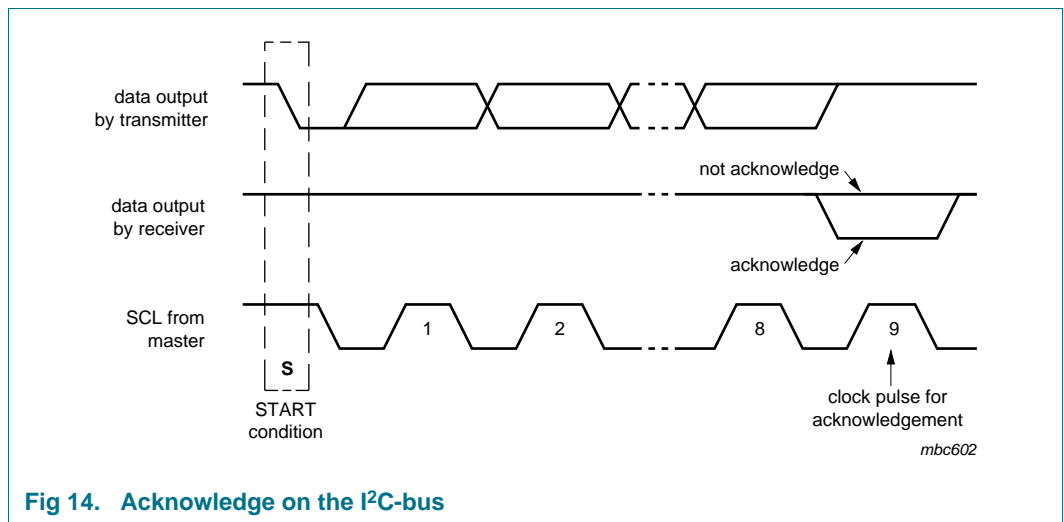


Fig 14. Acknowledge on the I<sup>2</sup>C-bus

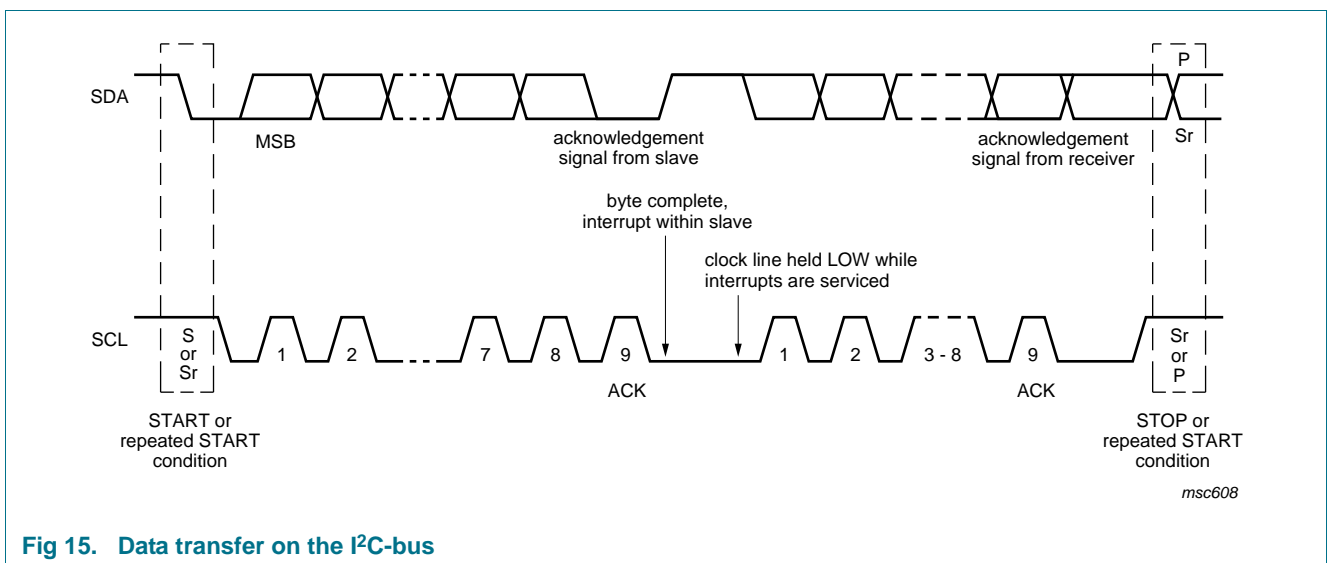


Fig 15. Data transfer on the I<sup>2</sup>C-bus

**8.1.4.5 7-Bit addressing**

During the I<sup>2</sup>C-bus address procedure, the first byte after the START condition is used to determine which slave will be selected by the master.

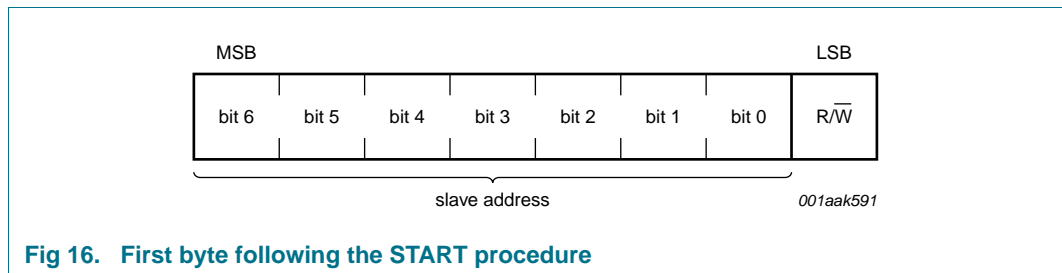
Several address numbers are reserved. During device configuration, the designer must ensure that collisions with these reserved addresses cannot occur. Check the *I<sup>2</sup>C-bus specification* for a complete list of reserved addresses.

The I<sup>2</sup>C-bus address specification is dependent on the definition of pin EA. Immediately after releasing pin NRSTPD or after a power-on reset, the device defines the I<sup>2</sup>C-bus address according to pin EA.

If pin EA is set LOW, the upper 4 bits of the device bus address are reserved by NXP Semiconductors and set to 0101b for all MFRC522 devices. The remaining 3 bits (ADR\_0, ADR\_1, ADR\_2) of the slave address can be freely configured by the customer to prevent collisions with other I<sup>2</sup>C-bus devices.

If pin EA is set HIGH, ADR\_0 to ADR\_5 can be completely specified at the external pins according to [Table 5 on page 9](#). ADR\_6 is always set to logic 0.

In both modes, the external address coding is latched immediately after releasing the reset condition. Further changes at the used pins are not taken into consideration. Depending on the external wiring, the I<sup>2</sup>C-bus address pins can be used for test signal outputs.



**Fig 16. First byte following the START procedure**

**8.1.4.6 Register write access**

To write data from the host controller using the I<sup>2</sup>C-bus to a specific register in the MFRC522 the following frame format must be used.

- The first byte of a frame indicates the device address according to the I<sup>2</sup>C-bus rules.
- The second byte indicates the register address followed by up to n-data bytes.

In one frame all data bytes are written to the same register address. This enables fast FIFO buffer access. The Read/Write (R/W) bit is set to logic 0.



8.1.4.7 Register read access

To read out data from a specific register address in the MFRC522, the host controller must use the following procedure:

- Firstly, a write access to the specific register address must be performed as indicated in the frame that follows
- The first byte of a frame indicates the device address according to the I<sup>2</sup>C-bus rules
- The second byte indicates the register address. No data bytes are added
- The Read/Write bit is 0

After the write access, read access can start. The host sends the device address of the MFRC522. In response, the MFRC522 sends the content of the read access register. In one frame all data bytes can be read from the same register address. This enables fast FIFO buffer access or register polling.

The Read/Write (R/W) bit is set to logic 1.

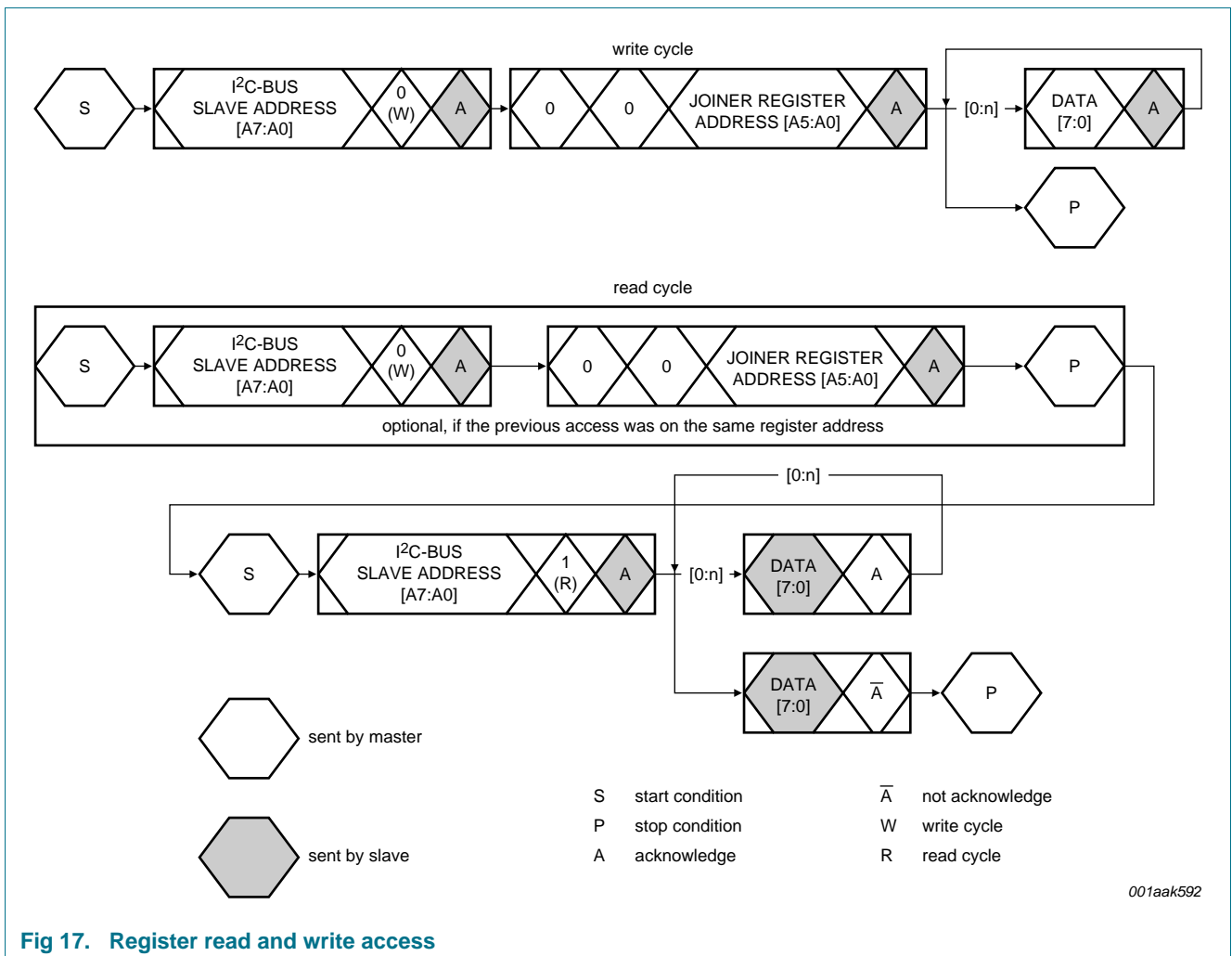


Fig 17. Register read and write access

**8.1.4.8 High-speed mode**

In High-speed mode (HS mode), the device can transfer information at data rates of up to 3.4 Mbit/s, while remaining fully downward-compatible with Fast or Standard mode (F/S mode) for bidirectional communication in a mixed-speed bus system.

**8.1.4.9 High-speed transfer**

To achieve data rates of up to 3.4 Mbit/s the following improvements have been made to I<sup>2</sup>C-bus operation.

- The inputs of the device in HS mode incorporate spike suppression, a Schmitt trigger on the SDA and SCL inputs and different timing constants when compared to F/S mode
- The output buffers of the device in HS mode incorporate slope control of the falling edges of the SDA and SCL signals with different fall times compared to F/S mode

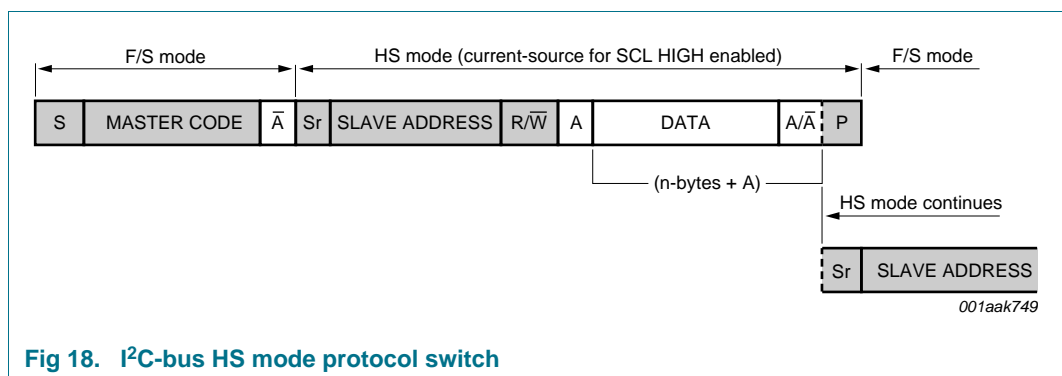
**8.1.4.10 Serial data transfer format in HS mode**

The HS mode serial data transfer format meets the Standard mode I<sup>2</sup>C-bus specification. HS mode can only start after all of the following conditions (all of which are in F/S mode):

1. START condition (S)
2. 8-bit master code (00001XXXb)
3. Not-acknowledge bit ( $\bar{A}$ )

When HS mode starts, the active master sends a repeated START condition (Sr) followed by a 7-bit slave address with a R/W bit address and receives an acknowledge bit (A) from the selected MFRC522.

Data transfer continues in HS mode after the next repeated START (Sr), only switching back to F/S mode after a STOP condition (P). To reduce the overhead of the master code, a master links a number of HS mode transfers, separated by repeated START conditions (Sr).



**Fig 18. I<sup>2</sup>C-bus HS mode protocol switch**

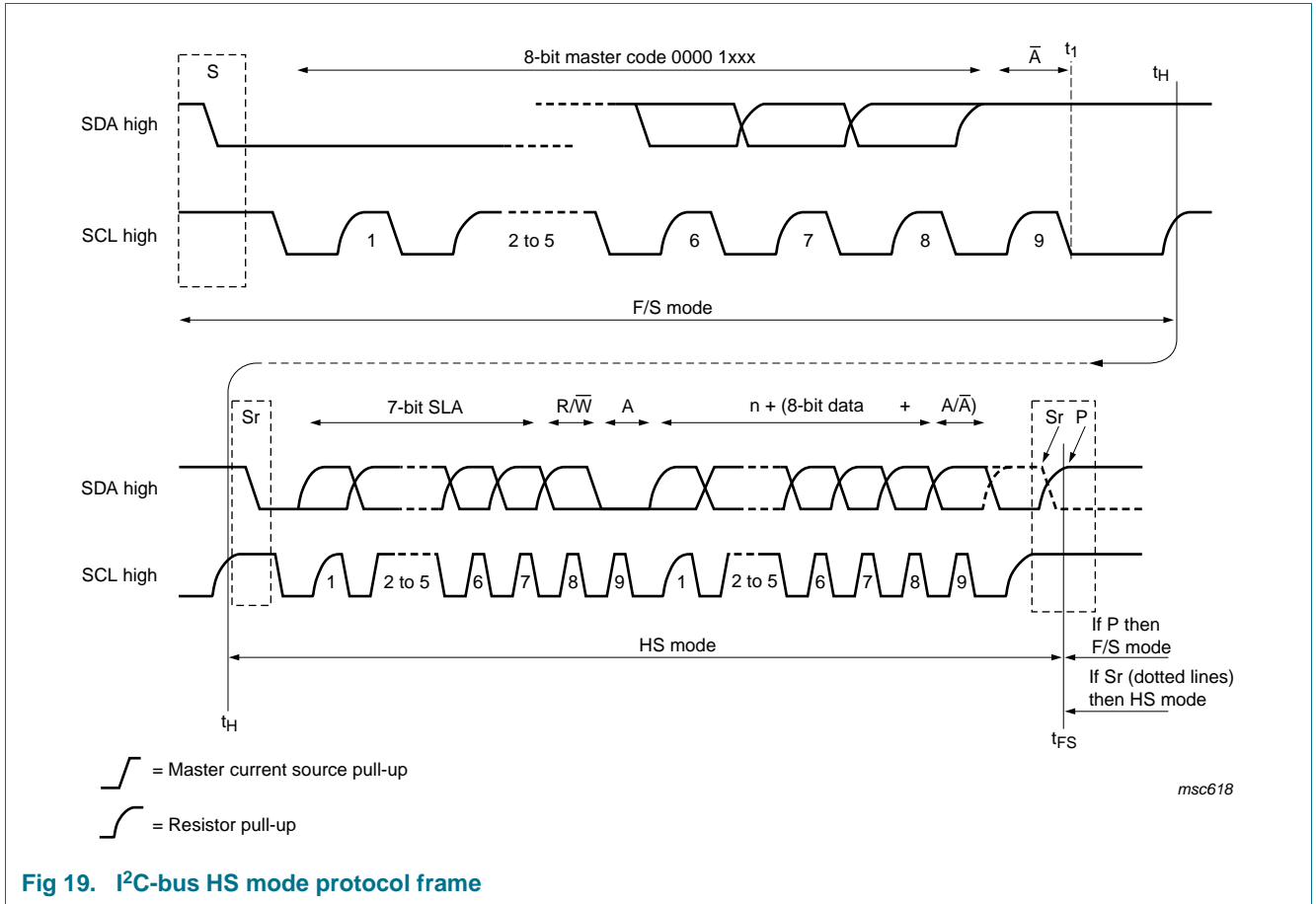


Fig 19. I<sup>2</sup>C-bus HS mode protocol frame

#### 8.1.4.11 Switching between F/S mode and HS mode

After reset and initialization, the MFRC522 is in Fast mode (which is in effect F/S mode as Fast mode is downward-compatible with Standard mode). The connected MFRC522 recognizes the "S 00001XXX A" sequence and switches its internal circuitry from the Fast mode setting to the HS mode setting.

The following actions are taken:

1. Adapt the SDA and SCL input filters according to the spike suppression requirement in HS mode.
2. Adapt the slope control of the SDA output stages.

It is possible for system configurations that do not have other I<sup>2</sup>C-bus devices involved in the communication to switch to HS mode permanently. This is implemented by setting Status2Reg register's I<sup>2</sup>CForceHS bit to logic 1. In permanent HS mode, the master code is not required to be sent. This is not defined in the specification and must only be used when no other devices are connected on the bus. In addition, spikes on the I<sup>2</sup>C-bus lines must be avoided because of the reduced spike suppression.

#### 8.1.4.12 MFRC522 at lower speed modes

MFRC522 is fully downward-compatible and can be connected to an F/S mode I<sup>2</sup>C-bus system. The device stays in F/S mode and communicates at F/S mode speeds because a master code is not transmitted in this configuration.

## 8.2 Analog interface and contactless UART

### 8.2.1 General

The integrated contactless UART supports the external host online with framing and error checking of the protocol requirements up to 848 kBd. An external circuit can be connected to the communication interface pins MFIN and MFOUT to modulate and demodulate the data.

The contactless UART handles the protocol requirements for the communication protocols in cooperation with the host. Protocol handling generates bit and byte-oriented framing. In addition, it handles error detection such as parity and CRC, based on the various supported contactless communication protocols.

**Remark:** The size and tuning of the antenna and the power supply voltage have an important impact on the achievable operating distance.

### 8.2.2 TX p-driver

The signal on pins TX1 and TX2 is the 13.56 MHz energy carrier modulated by an envelope signal. It can be used to drive an antenna directly using a few passive components for matching and filtering; see [Section 15 on page 81](#). The signal on pins TX1 and TX2 can be configured using the TxControlReg register; see [Section 9.3.2.5 on page 50](#).

The modulation index can be set by adjusting the impedance of the drivers. The impedance of the p-driver can be configured using registers CWGsPReg and ModGsPReg. The impedance of the n-driver can be configured using the GsNReg register. The modulation index also depends on the antenna design and tuning.

The TxModeReg and TxSelReg registers control the data rate and framing during transmission and the antenna driver setting to support the different requirements at the different modes and transfer speeds.

**Table 15. Register and bit settings controlling the signal on pin TX1**

Bit Tx1RFEn	Bit Force 100ASK	Bit InvTx1RFOn	Bit InvTx1RFOff	Envelope	Pin TX1	GSPMos	GSNMos	Remarks
0	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	not specified if RF is switched off
1	0	0	X <sup>[1]</sup>	0	RF	pMod	nMod	100 % ASK: pin TX1 pulled to logic 0, independent of the InvTx1RFOff bit
				1	RF	pCW	nCW	
	0	1	X <sup>[1]</sup>	0	RF	pMod	nMod	
				1	RF	pCW	nCW	
	1	1	X <sup>[1]</sup>	0	0	pMod	nMod	
				1	RF_n	pCW	nCW	

[1] X = Do not care.

Table 16. Register and bit settings controlling the signal on pin TX2

Bit Tx1RFEn	Bit Force 100ASK	Bit Tx2CW	Bit InvTx2RFOOn	Bit InvTx2RFOff	Envelope	Pin TX2	GSPMos	GSNMos	Remarks			
0	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	X <sup>[1]</sup>	not specified if RF is switched off			
1	0	0	0	X <sup>[1]</sup>	0	RF	pMod	nMod	-			
					1	RF	pCW	nCW				
			1	X <sup>[1]</sup>	0	RF_n	pMod	nMod				
					1	RF_n	pCW	nCW				
	1	0	X <sup>[1]</sup>	X <sup>[1]</sup>	RF	pCW	nCW	conductance always CW for the Tx2CW bit				
					RF_n	pCW	nCW					
	1	0	0	X <sup>[1]</sup>	X <sup>[1]</sup>	0	0	pMod	nMod	100 % ASK: pin TX2 pulled to logic 0 (independent of the InvTx2RFOOn/InvTx2RFOff bits)		
						1	RF	pCW	nCW			
						1	X <sup>[1]</sup>	0	0		pMod	nMod
								1	RF_n		pCW	nCW
1						0	X <sup>[1]</sup>	X <sup>[1]</sup>	RF		pCW	nCW
									RF_n		pCW	nCW

[1] X = Do not care.

The following abbreviations have been used in [Table 15](#) and [Table 16](#):

- RF: 13.56 MHz clock derived from 27.12 MHz quartz crystal oscillator divided by 2
- RF\_n: inverted 13.56 MHz clock
- GSPMos: conductance, configuration of the PMOS array
- GSNMos: conductance, configuration of the NMOS array
- pCW: PMOS conductance value for continuous wave defined by the CWGsPReg register
- pMod: PMOS conductance value for modulation defined by the ModGsPReg register
- nCW: NMOS conductance value for continuous wave defined by the GsNReg register's CWGsN[3:0] bits
- nMod: NMOS conductance value for modulation defined by the GsNReg register's ModGsN[3:0] bits
- X = do not care.

**Remark:** If only one driver is switched on, the values for CWGsPReg, ModGsPReg and GsNReg registers are used for both drivers.

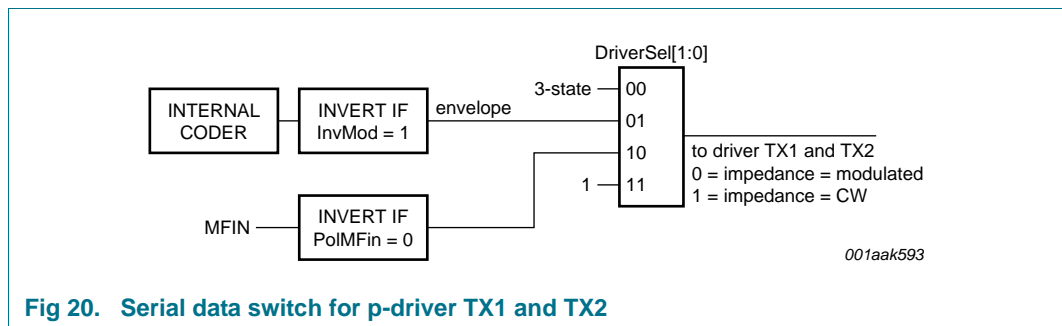
**8.2.3 Serial data switch**

Two main blocks are implemented in the MFRC522. The digital block comprises the state machines, encoder/decoder logic. The analog block comprises the modulator and antenna drivers, the receiver and amplifiers. It is possible for the interface between these two blocks to be configured so that the interfacing signals are routed to pins MFIN and MFOUT.

This topology allows the analog block of the MFRC522 to be connected to the digital block of another device.

The serial signal switch is controlled by the TxSelReg and RxSelReg registers.

Figure 20 shows the serial data switch for p-driver TX1 and TX2.



**Fig 20. Serial data switch for p-driver TX1 and TX2**

**8.2.4 MFIN and MFOUT interface support**

The MFRC522 is divided into a digital circuit block and an analog circuit block. The digital block contains state machines, encoder and decoder logic and so on. The analog block contains the modulator and antenna drivers, receiver and amplifiers. The interface between these two blocks can be configured so that the interfacing signals can be routed to pins MFIN and MFOUT; see Figure 21 on page 28. This configuration is implemented using TxSelReg register’s MFOutSel[3:0] and DriverSel[1:0] bits and RxSelReg register’s UARTSel[1:0] bits.

This topology allows some parts of the analog block to be connected to the digital block of another device.

Switch MFOutSel in the TxSelReg register can be used to measure MIFARE and ISO/IEC14443 A related signals. This is especially important during the design-in phase or for test purposes as it enables checking of the transmitted and received data.

The most important use of pins MFIN and MFOUT is found in the active antenna concept. An external active antenna circuit can be connected to the MFRC522’s digital block. Switch MFOutSel must be configured so that the internal Miller encoded signal is sent to pin MFOUT (MFOutSel = 100b). UARTSel[1:0] must be configured to receive a Manchester signal with subcarrier from pin MFIN (UARTSel[1:0] = 01).

It is possible to connect a passive antenna to pins TX1, TX2 and RX (using the appropriate filter and matching circuit) and an active antenna to pins MFOUT and MFIN at the same time. In this configuration, two RF circuits can be driven (one after another) by a single host processor.

**Remark:** Pins MFIN and MFOUT have a dedicated supply on pin SVDD with the ground on pin PVSS. If pin MFIN is not used it must be connected to either pin SVDD or pin PVSS. If pin SVDD is not used it must be connected to either pin DVDD, pin PVDD or any other voltage supply pin.



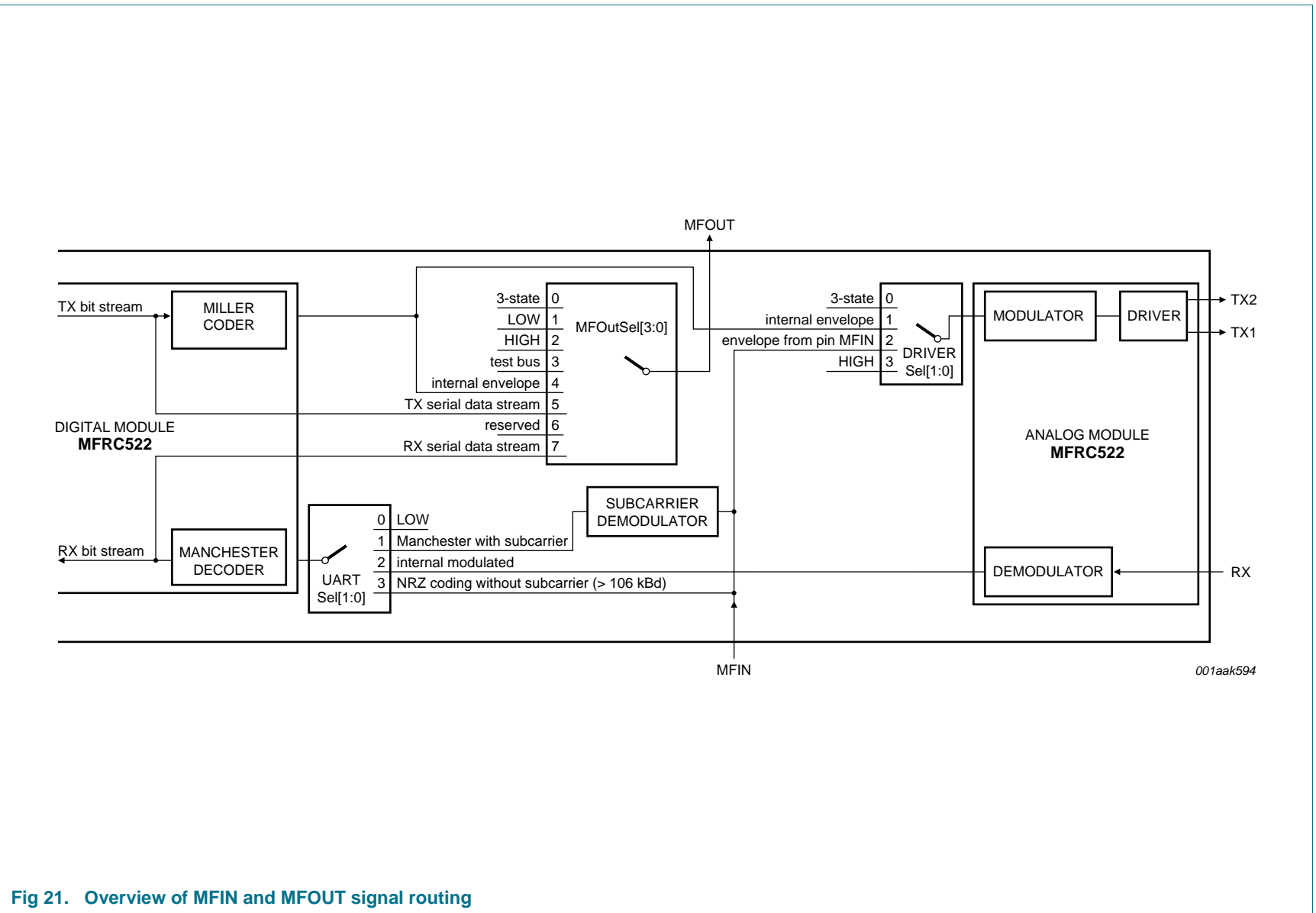


Fig 21. Overview of MFIN and MFOUT signal routing

### 8.2.5 CRC coprocessor

The following CRC coprocessor parameters can be configured:

- The CRC preset value can be either 0000h, 6363h, A671h or FFFFh depending on the ModeReg register's CRCPreset[1:0] bits setting
- The CRC polynomial for the 16-bit CRC is fixed to  $x^{16} + x^{12} + x^5 + 1$
- The CRCResultReg register indicates the result of the CRC calculation. This register is split into two 8-bit registers representing the higher and lower bytes.
- The ModeReg register's MSBFirst bit indicates that data will be loaded with the MSB first.

**Table 17. CRC coprocessor parameters**

Parameter	Value
CRC register length	16-bit CRC
CRC algorithm	algorithm according to ISO/IEC 14443 A and ITU-T
CRC preset value	0000h, 6363h, A671h or FFFFh depending on the setting of the ModeReg register's CRCPreset[1:0] bits

## 8.3 FIFO buffer

An 8 × 64 bit FIFO buffer is used in the MFRC522. It buffers the input and output data stream between the host and the MFRC522's internal state machine. This makes it possible to manage data streams up to 64 bytes long without the need to take timing constraints into account.

### 8.3.1 Accessing the FIFO buffer

The FIFO buffer input and output data bus is connected to the FIFODataReg register. Writing to this register stores one byte in the FIFO buffer and increments the internal FIFO buffer write pointer. Reading from this register shows the FIFO buffer contents stored in the FIFO buffer read pointer and decrements the FIFO buffer read pointer. The distance between the write and read pointer can be obtained by reading the FIFOLevelReg register.

When the microcontroller starts a command, the MFRC522 can, while the command is in progress, access the FIFO buffer according to that command. Only one FIFO buffer has been implemented which can be used for input and output. The microcontroller must ensure that there are not any unintentional FIFO buffer accesses.

### 8.3.2 Controlling the FIFO buffer

The FIFO buffer pointers can be reset by setting FIFOLevelReg register's FlushBuffer bit to logic 1. Consequently, the FIFOLevel[6:0] bits are all set to logic 0 and the ErrorReg register's BufferOvfl bit is cleared. The bytes stored in the FIFO buffer are no longer accessible allowing the FIFO buffer to be filled with another 64 bytes.

### 8.3.3 FIFO buffer status information

The host can get the following FIFO buffer status information:

- Number of bytes stored in the FIFO buffer: FIFOLevelReg register's FIFOLevel[6:0]
- FIFO buffer almost full warning: Status1Reg register's HiAlert bit

- FIFO buffer almost empty warning: Status1Reg register's LoAlert bit
- FIFO buffer overflow warning: ErrorReg register's BufferOvfl bit. The BufferOvfl bit can only be cleared by setting the FIFOLevelReg register's FlushBuffer bit.

The MFRC522 can generate an interrupt signal when:

- ComIEnReg register's LoAlertIEn bit is set to logic 1. It activates pin IRQ when Status1Reg register's LoAlert bit changes to logic 1.
- ComIEnReg register's HiAlertIEn bit is set to logic 1. It activates pin IRQ when Status1Reg register's HiAlert bit changes to logic 1.

If the maximum number of WaterLevel bytes (as set in the WaterLevelReg register) or less are stored in the FIFO buffer, the HiAlert bit is set to logic 1. It is generated according to [Equation 3](#):

$$HiAlert = (64 - FIFOLength) \leq WaterLevel \quad (3)$$

If the number of WaterLevel bytes (as set in the WaterLevelReg register) or less are stored in the FIFO buffer, the LoAlert bit is set to logic 1. It is generated according to [Equation 4](#):

$$LoAlert = FIFOLength \leq WaterLevel \quad (4)$$

## 8.4 Interrupt request system

The MFRC522 indicates certain events by setting the Status1Reg register's IRq bit and, if activated, by pin IRQ. The signal on pin IRQ can be used to interrupt the host using its interrupt handling capabilities. This allows the implementation of efficient host software.

### 8.4.1 Interrupt sources overview

[Table 18](#) shows the available interrupt bits, the corresponding source and the condition for its activation. The ComIrqReg register's TimerIrq interrupt bit indicates an interrupt set by the timer unit which is set when the timer decrements from 1 to 0.

The ComIrqReg register's TxIrq bit indicates that the transmitter has finished. If the state changes from sending data to transmitting the end of the frame pattern, the transmitter unit automatically sets the interrupt bit. The CRC coprocessor sets the DivIrqReg register's CRCIrq bit after processing all the FIFO buffer data which is indicated by CRCReady bit = 1.

The ComIrqReg register's RxIrq bit indicates an interrupt when the end of the received data is detected. The ComIrqReg register's IdleIrq bit is set if a command finishes and the Command[3:0] value in the CommandReg register changes to idle (see [Table 149 on page 70](#)).

The ComIrqReg register's HiAlertIrq bit is set to logic 1 when the Status1Reg register's HiAlert bit is set to logic 1 which means that the FIFO buffer has reached the level indicated by the WaterLevel[5:0] bits.

The ComIrqReg register's LoAlertIrq bit is set to logic 1 when the Status1Reg register's LoAlert bit is set to logic 1 which means that the FIFO buffer has reached the level indicated by the WaterLevel[5:0] bits.

The ComIrqReg register's ErrIRq bit indicates an error detected by the contactless UART during send or receive. This is indicated when any bit is set to logic 1 in register ErrorReg.

**Table 18. Interrupt sources**

Interrupt flag	Interrupt source	Trigger action
IRq	timer unit	the timer counts from 1 to 0
TxIRq	transmitter	a transmitted data stream ends
CRCIRq	CRC coprocessor	all data from the FIFO buffer has been processed
RxIRq	receiver	a received data stream ends
IdleIRq	ComIrqReg register	command execution finishes
HiAlertIRq	FIFO buffer	the FIFO buffer is almost full
LoAlertIRq	FIFO buffer	the FIFO buffer is almost empty
ErrIRq	contactless UART	an error is detected

## 8.5 Timer unit

The MFRC522A has a timer unit which the external host can use to manage timing tasks. The timer unit can be used in one of the following timer/counter configurations:

- Timeout counter
- Watchdog counter
- Stop watch
- Programmable one shot
- Periodical trigger

The timer unit can be used to measure the time interval between two events or to indicate that a specific event occurred after a specific time. The timer can be triggered by events explained in the paragraphs below. The timer does not influence any internal events, for example, a time-out during data reception does not automatically influence the reception process. Furthermore, several timer-related bits can be used to generate an interrupt.

The timer has an input clock of 13.56 MHz derived from the 27.12 MHz quartz crystal oscillator. The timer consists of two stages: prescaler and counter.

The prescaler (TPrescaler) is a 12-bit counter. The reload values (TReloadVal\_Hi[7:0] and TReloadVal\_Lo[7:0]) for TPrescaler can be set between 0 and 4095 in the TModeReg register's TPrescaler\_Hi[3:0] bits and TPrescalerReg register's TPrescaler\_Lo[7:0] bits.

The reload value for the counter is defined by 16 bits between 0 and 65535 in the TReloadReg register.

The current value of the timer is indicated in the TCounterValReg register.

When the counter reaches 0, an interrupt is automatically generated, indicated by the ComIrqReg register's TimerIRq bit setting. If enabled, this event can be indicated on pin IRQ. The TimerIRq bit can be set and reset by the host. Depending on the configuration, the timer will stop at 0 or restart with the value set in the TReloadReg register.

The timer status is indicated by the Status1Reg register's TRunning bit.

The timer can be started manually using the ControlReg register's TStartNow bit and stopped using the ControlReg register's TStopNow bit.

The timer can also be activated automatically to meet any dedicated protocol requirements by setting the TModeReg register's TAuto bit to logic 1.

The delay time of a timer stage is set by the reload value + 1. The total delay time ( $t_{d1}$ ) is calculated using [Equation 5](#):

$$t_{d1} = \frac{(TPrescaler \times 2 + 1) \times (TReloadVal + 1)}{13.56 \text{ MHz}} \quad (5)$$

An example of calculating total delay time ( $t_d$ ) is shown in [Equation 6](#), where the TPrescaler value = 4095 and TReloadVal = 65535:

$$39.59 \text{ s} = \frac{(4095 \times 2 + 1) \times (65535 + 1)}{13.56 \text{ MHz}} \quad (6)$$

**Example:** To give a delay time of 25  $\mu\text{s}$  requires 339 clock cycles to be counted and a TPrescaler value of 169. This configures the timer to count up to 65535 time-slots for every 25  $\mu\text{s}$  period.

The MFRC522 version 2.0 offers in addition a second prescaler timer. Due to the fact that the prescaler counts down to 0 the prescaler period always count an odd number of clocks (1, 3, 5, ..). This may lead to inaccuracy. The second available prescaler timer implements the possibility to change the prescaler reload value to odd numbers, which results in an even prescaler period. This new prescaler can be enabled only in version 2.0 using the register bit DemodeReg, see [Table 72](#). Within this option, the total delay time ( $t_{d2}$ ) is calculated using [Equation 5](#):

$$t_{d2} = \frac{(TPrescaler \times 2 + 2) \times (TReloadVal + 1)}{13.56 \text{ MHz}} \quad (7)$$

8.6 Power reduction modes

8.6.1 Hard power-down

Hard power-down is enabled when pin NRSTPD is LOW. This turns off all internal current sinks including the oscillator. All digital input buffers are separated from the input pins and clamped internally (except pin NRSTPD). The output pins are frozen at either a HIGH or LOW level.

8.6.2 Soft power-down mode

Soft Power-down mode is entered immediately after the CommandReg register's PowerDown bit is set to logic 1. All internal current sinks are switched off, including the oscillator buffer. However, the digital input buffers are not separated from the input pins and keep their functionality. The digital output pins do not change their state.

During soft power-down, all register values, the FIFO buffer content and the configuration keep their current contents.

After setting the PowerDown bit to logic 0, it takes 1024 clocks until the Soft power-down mode is exited indicated by the PowerDown bit. Setting it to logic 0 does not immediately clear it. It is cleared automatically by the MFRC522 when Soft power-down mode is exited.

**Remark:** If the internal oscillator is used, you must take into account that it is supplied by pin AVDD and it will take a certain time ( $t_{osc}$ ) until the oscillator is stable and the clock cycles can be detected by the internal logic. It is recommended for the serial UART, to first send the value 55h to the MFRC522. The oscillator must be stable for further access to the registers. To ensure this, perform a read access to address 0 until the MFRC522 answers to the last read command with the register content of address 0. This indicates that the MFRC522 is ready.

8.6.3 Transmitter power-down mode

The Transmitter Power-down mode switches off the internal antenna drivers thereby, turning off the RF field. Transmitter power-down mode is entered by setting either the TxControlReg register's Tx1RFEn bit or Tx2RFEn bit to logic 0.

8.7 Oscillator circuit

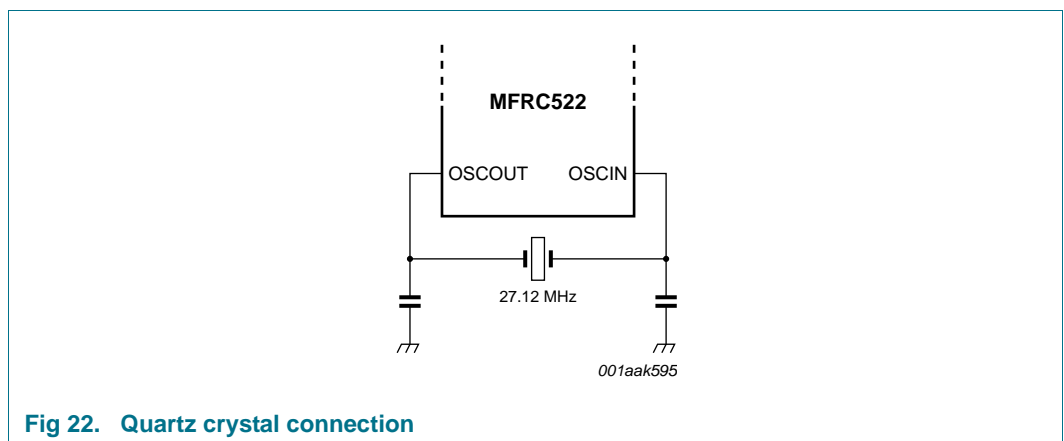


Fig 22. Quartz crystal connection

The clock applied to the MFRC522 provides a time basis for the synchronous system's encoder and decoder. The stability of the clock frequency, therefore, is an important factor for correct operation. To obtain optimum performance, clock jitter must be reduced as much as possible. This is best achieved using the internal oscillator buffer with the recommended circuitry.

If an external clock source is used, the clock signal must be applied to pin OSCIN. In this case, special care must be taken with the clock duty cycle and clock jitter and the clock quality must be verified.

## 8.8 Reset and oscillator start-up time

### 8.8.1 Reset timing requirements

The reset signal is filtered by a hysteresis circuit and a spike filter before it enters the digital circuit. The spike filter rejects signals shorter than 10 ns. In order to perform a reset, the signal must be LOW for at least 100 ns.

### 8.8.2 Oscillator start-up time

If the MFRC522 has been set to a Power-down mode or is powered by a  $V_{DDX}$  supply, the start-up time for the MFRC522 depends on the oscillator used and is shown in [Figure 23](#).

The time ( $t_{startup}$ ) is the start-up time of the crystal oscillator circuit. The crystal oscillator start-up time is defined by the crystal.

The time ( $t_d$ ) is the internal delay time of the MFRC522 when the clock signal is stable before the MFRC522 can be addressed.

The delay time is calculated by:

$$t_d = \frac{1024}{27 \mu s} = 37.74 \mu s \tag{8}$$

The time ( $t_{osc}$ ) is the sum of  $t_d$  and  $t_{startup}$ .

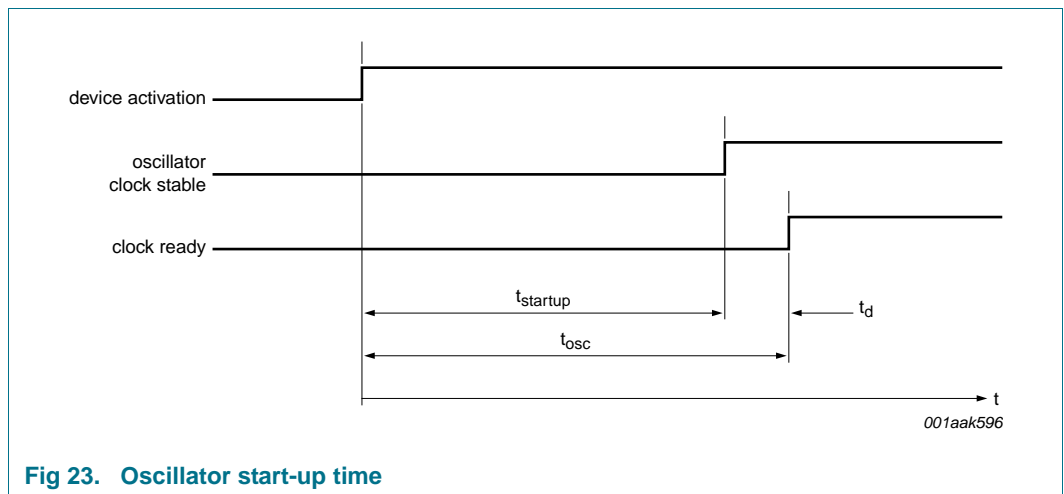


Fig 23. Oscillator start-up time

## 9. MFRC522 registers

### 9.1 Register bit behavior

Depending on the functionality of a register, the access conditions to the register can vary. In principle, bits with same behavior are grouped in common registers. The access conditions are described in [Table 19](#).

**Table 19. Behavior of register bits and their designation**

Abbreviation	Behavior	Description
R/W	read and write	These bits can be written and read by the microcontroller. Since they are used only for control purposes, their content is not influenced by internal state machines, for example the ComIEnReg register can be written and read by the microcontroller. It will also be read by internal state machines but never changed by them.
D	dynamic	These bits can be written and read by the microcontroller. Nevertheless, they can also be written automatically by internal state machines, for example the CommandReg register changes its value automatically after the execution of the command.
R	read only	These register bits hold values which are determined by internal states only, for example the CRCReady bit cannot be written externally but shows internal states.
W	write only	Reading these register bits always returns zero.
reserved	-	These registers are reserved for future use and must not be changed. In case of a write access, it is recommended to always write the value "0".
RFT	-	These register bits are reserved for future use or are for production tests and must not be changed.



## 9.2 Register overview

Table 20. MFRC522 register overview

Address (hex)	Register name	Function	Refer to
<b>Page 0: Command and status</b>			
00h	Reserved	reserved for future use	<a href="#">Table 21 on page 38</a>
01h	CommandReg	starts and stops command execution	<a href="#">Table 23 on page 38</a>
02h	ComlEnReg	enable and disable interrupt request control bits	<a href="#">Table 25 on page 38</a>
03h	DivlEnReg	enable and disable interrupt request control bits	<a href="#">Table 27 on page 39</a>
04h	ComlIrqReg	interrupt request bits	<a href="#">Table 29 on page 39</a>
05h	DivlIrqReg	interrupt request bits	<a href="#">Table 31 on page 40</a>
06h	ErrorReg	error bits showing the error status of the last command executed	<a href="#">Table 33 on page 41</a>
07h	Status1Reg	communication status bits	<a href="#">Table 35 on page 42</a>
08h	Status2Reg	receiver and transmitter status bits	<a href="#">Table 37 on page 43</a>
09h	FIFODataReg	input and output of 64 byte FIFO buffer	<a href="#">Table 39 on page 44</a>
0Ah	FIFOLevelReg	number of bytes stored in the FIFO buffer	<a href="#">Table 41 on page 44</a>
0Bh	WaterLevelReg	level for FIFO underflow and overflow warning	<a href="#">Table 43 on page 44</a>
0Ch	ControlReg	miscellaneous control registers	<a href="#">Table 45 on page 45</a>
0Dh	BitFramingReg	adjustments for bit-oriented frames	<a href="#">Table 47 on page 46</a>
0Eh	CollReg	bit position of the first bit-collision detected on the RF interface	<a href="#">Table 49 on page 46</a>
0Fh	Reserved	reserved for future use	<a href="#">Table 51 on page 47</a>
<b>Page 1: Command</b>			
10h	Reserved	reserved for future use	<a href="#">Table 53 on page 47</a>
11h	ModeReg	defines general modes for transmitting and receiving	<a href="#">Table 55 on page 48</a>
12h	TxModeReg	defines transmission data rate and framing	<a href="#">Table 57 on page 48</a>
13h	RxModeReg	defines reception data rate and framing	<a href="#">Table 59 on page 49</a>
14h	TxControlReg	controls the logical behavior of the antenna driver pins TX1 and TX2	<a href="#">Table 61 on page 50</a>
15h	TxASKReg	controls the setting of the transmission modulation	<a href="#">Table 63 on page 51</a>
16h	TxSelReg	selects the internal sources for the antenna driver	<a href="#">Table 65 on page 51</a>
17h	RxSelReg	selects internal receiver settings	<a href="#">Table 67 on page 52</a>
18h	RxThresholdReg	selects thresholds for the bit decoder	<a href="#">Table 69 on page 53</a>
19h	DemodReg	defines demodulator settings	<a href="#">Table 71 on page 53</a>
1Ah	Reserved	reserved for future use	<a href="#">Table 73 on page 54</a>
1Bh	Reserved	reserved for future use	<a href="#">Table 75 on page 54</a>
1Ch	MfTxReg	controls some MIFARE communication transmit parameters	<a href="#">Table 77 on page 55</a>
1Dh	MfRxReg	controls some MIFARE communication receive parameters	<a href="#">Table 79 on page 55</a>
1Eh	Reserved	reserved for future use	<a href="#">Table 81 on page 55</a>
1Fh	SerialSpeedReg	selects the speed of the serial UART interface	<a href="#">Table 83 on page 55</a>
<b>Page 2: Configuration</b>			
20h	Reserved	reserved for future use	<a href="#">Table 85 on page 57</a>

Table 20. MFRC522 register overview ...continued

Address (hex)	Register name	Function	Refer to
21h	CRCResultReg	shows the MSB and LSB values of the CRC calculation	<a href="#">Table 87 on page 57</a>
22h			<a href="#">Table 89 on page 57</a>
23h	Reserved	reserved for future use	<a href="#">Table 91 on page 58</a>
24h	ModWidthReg	controls the ModWidth setting	<a href="#">Table 93 on page 58</a>
25h	Reserved	reserved for future use	<a href="#">Table 95 on page 58</a>
26h	RFCfgReg	configures the receiver gain	<a href="#">Table 97 on page 59</a>
27h	GsNReg	selects the conductance of the antenna driver pins TX1 and TX2 for modulation	<a href="#">Table 99 on page 59</a>
28h	CWGsPReg	defines the conductance of the p-driver output during periods of no modulation	<a href="#">Table 101 on page 60</a>
29h	ModGsPReg	defines the conductance of the p-driver output during periods of modulation	<a href="#">Table 103 on page 60</a>
2Ah	TModeReg	defines settings for the internal timer	<a href="#">Table 105 on page 60</a>
2Bh	TPrescalerReg		<a href="#">Table 107 on page 61</a>
2Ch	TReloadReg	defines the 16-bit timer reload value	<a href="#">Table 109 on page 62</a>
2Dh			<a href="#">Table 111 on page 62</a>
2Eh	TCounterValReg	shows the 16-bit timer value	<a href="#">Table 113 on page 63</a>
2Fh			<a href="#">Table 115 on page 63</a>
<b>Page 3: Test register</b>			
30h	Reserved	reserved for future use	<a href="#">Table 117 on page 63</a>
31h	TestSel1Reg	general test signal configuration	<a href="#">Table 119 on page 63</a>
32h	TestSel2Reg	general test signal configuration and PRBS control	<a href="#">Table 121 on page 64</a>
33h	TestPinEnReg	enables pin output driver on pins D1 to D7	<a href="#">Table 123 on page 64</a>
34h	TestPinValueReg	defines the values for D1 to D7 when it is used as an I/O bus	<a href="#">Table 125 on page 65</a>
35h	TestBusReg	shows the status of the internal test bus	<a href="#">Table 127 on page 65</a>
36h	AutoTestReg	controls the digital self test	<a href="#">Table 129 on page 66</a>
37h	VersionReg	shows the software version	<a href="#">Table 131 on page 66</a>
38h	AnalogTestReg	controls the pins AUX1 and AUX2	<a href="#">Table 133 on page 67</a>
39h	TestDAC1Reg	defines the test value for TestDAC1	<a href="#">Table 135 on page 68</a>
3Ah	TestDAC2Reg	defines the test value for TestDAC2	<a href="#">Table 137 on page 68</a>
3Bh	TestADCReg	shows the value of ADC I and Q channels	<a href="#">Table 139 on page 68</a>
3Ch to 3Fh	Reserved	reserved for production tests	<a href="#">Table 141 to Table 147 on page 69</a>

## 9.3 Register descriptions

### 9.3.1 Page 0: Command and status

#### 9.3.1.1 Reserved register 00h

Functionality is reserved for future use.

**Table 21. Reserved register (address 00h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

**Table 22. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	-	reserved

#### 9.3.1.2 CommandReg register

Starts and stops command execution.

**Table 23. CommandReg register (address 01h); reset value: 20h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol:	reserved		RcvOff	PowerDown	Command[3:0]			
Access:	-		R/W	D	D			

**Table 24. CommandReg register bit descriptions**

Bit	Symbol	Value	Description
7 to 6	reserved	-	reserved for future use
5	RcvOff	1	analog part of the receiver is switched off
4	PowerDown	1	Soft power-down mode entered
		0	MFRC522 starts the wake up procedure during which this bit is read as a logic 1; it is read as a logic 0 when the MFRC522 is ready; see <a href="#">Section 8.6.2 on page 33</a> <b>Remark:</b> The PowerDown bit cannot be set when the SoftReset command is activated
3 to 0	Command[3:0]	-	activates a command based on the Command value; reading this register shows which command is executed; see <a href="#">Section 10.3 on page 70</a>

#### 9.3.1.3 ComIEnReg register

Control bits to enable and disable the passing of interrupt requests.

**Table 25. ComIEnReg register (address 02h); reset value: 80h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	IRqInv	TxIEn	RxIEn	IdleIEn	HiAlertIEn	LoAlertIEn	ErrIEn	TimerIEn
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26. ComIEnReg register bit descriptions

Bit	Symbol	Value	Description
7	IRqInv	1	signal on pin IRQ is inverted with respect to the Status1Reg register's IRq bit
		0	signal on pin IRQ is equal to the IRq bit; in combination with the DivIEnReg register's IRqPushPull bit, the default value of logic 1 ensures that the output level on pin IRQ is 3-state
6	TxIEn	-	allows the transmitter interrupt request (TxIRq bit) to be propagated to pin IRQ
5	RxIEn	-	allows the receiver interrupt request (RxIRq bit) to be propagated to pin IRQ
4	IdleIEn	-	allows the idle interrupt request (IdleIRq bit) to be propagated to pin IRQ
3	HiAlertIEn	-	allows the high alert interrupt request (HiAlertIRq bit) to be propagated to pin IRQ
2	LoAlertIEn	-	allows the low alert interrupt request (LoAlertIRq bit) to be propagated to pin IRQ
1	ErrIEn	-	allows the error interrupt request (ErrIRq bit) to be propagated to pin IRQ
0	TimerIEn	-	allows the timer interrupt request (TimerIRq bit) to be propagated to pin IRQ

#### 9.3.1.4 DivIEnReg register

Control bits to enable and disable the passing of interrupt requests.

Table 27. DivIEnReg register (address 03h); reset value: 00h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	IRQPushPull	reserved	MfinActIEn	reserved	CRCIEn	reserved		
Access	R/W	-	R/W	-	R/W	-		

Table 28. DivIEnReg register bit descriptions

Bit	Symbol	Value	Description
7	IRQPushPull	1	pin IRQ is a standard CMOS output pin
		0	pin IRQ is an open-drain output pin
6 to 5	reserved	-	reserved for future use
4	MfinActIEn	-	allows the MFIN active interrupt request to be propagated to pin IRQ
3	reserved	-	reserved for future use
2	CRCIEn	-	allows the CRC interrupt request, indicated by the DivIrqReg register's CRCIRq bit, to be propagated to pin IRQ
1 to 0	reserved	-	reserved for future use

#### 9.3.1.5 ComIrqReg register

Interrupt request bits.

Table 29. ComIrqReg register (address 04h); reset value: 14h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	Set1	TxIRq	RxIRq	IdleIRq	HiAlertIRq	LoAlertIRq	ErrIRq	TimerIRq
Access	W	D	D	D	D	D	D	D

**Table 30. ComlRqReg register bit descriptions**

All bits in the ComlRqReg register are cleared by software.

Bit	Symbol	Value	Description
7	Set1	1	indicates that the marked bits in the ComlRqReg register are set
		0	indicates that the marked bits in the ComlRqReg register are cleared
6	TxIRq	1	set immediately after the last bit of the transmitted data was sent out
5	RxIRq	1	receiver has detected the end of a valid data stream if the RxModeReg register's RxNoErr bit is set to logic 1, the RxIRq bit is only set to logic 1 when data bytes are available in the FIFO
4	IdleIRq	1	If a command terminates, for example, when the CommandReg changes its value from any command to the Idle command (see <a href="#">Table 149 on page 70</a> ) if an unknown command is started, the CommandReg register Command[3:0] value changes to the idle state and the IdleIRq bit is set The microcontroller starting the Idle command does not set the IdleIRq bit
3	HiAlertIRq	1	the Status1Reg register's HiAlert bit is set in opposition to the HiAlert bit, the HiAlertIRq bit stores this event and can only be reset as indicated by the Set1 bit in this register
2	LoAlertIRq	1	Status1Reg register's LoAlert bit is set in opposition to the LoAlert bit, the LoAlertIRq bit stores this event and can only be reset as indicated by the Set1 bit in this register
1	ErrIRq	1	any error bit in the ErrorReg register is set
0	TimerIRq	1	the timer decrements the timer value in register TCounterValReg to zero

### 9.3.1.6 DivlRqReg register

Interrupt request bits.

**Table 31. DivlRqReg register (address 05h); reset value: x0h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	Set2	reserved		MfinActIRq	reserved	CRCIRq	reserved	
Access	W	-		D	-	D	-	

**Table 32. DivlRqReg register bit descriptions**

All bits in the DivlRqReg register are cleared by software.

Bit	Symbol	Value	Description
7	Set2	1	indicates that the marked bits in the DivlRqReg register are set
		0	indicates that the marked bits in the DivlRqReg register are cleared
6 to 5	reserved	-	reserved for future use
4	MfinActIRq	1	MFIN is active this interrupt is set when either a rising or falling signal edge is detected
3	reserved	-	reserved for future use
2	CRCIRq	1	the CalcCRC command is active and all data is processed
1 to 0	reserved	-	reserved for future use

9.3.1.7 ErrorReg register

Error bit register showing the error status of the last command executed.

Table 33. ErrorReg register (address 06h); reset value: 00h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	WrErr	TempErr	reserved	BufferOvfl	CollErr	CRCErr	ParityErr	ProtocolErr
Access	R	R	-	R	R	R	R	R

Table 34. ErrorReg register bit descriptions

Bit	Symbol	Value	Description
7	WrErr	1	data is written into the FIFO buffer by the host during the MFAuthent command or if data is written into the FIFO buffer by the host during the time between sending the last bit on the RF interface and receiving the last bit on the RF interface
6	TempErr <sup>[1]</sup>	1	internal temperature sensor detects overheating, in which case the antenna drivers are automatically switched off
5	reserved	-	reserved for future use
4	BufferOvfl	1	the host or a MFRC522's internal state machine (e.g. receiver) tries to write data to the FIFO buffer even though it is already full
3	CollErr	1	a bit-collision is detected cleared automatically at receiver start-up phase only valid during the bitwise anticollision at 106 kBd always set to logic 0 during communication protocols at 212 kBd, 424 kBd and 848 kBd
2	CRCErr	1	the RxModeReg register's RxCRCEn bit is set and the CRC calculation fails automatically cleared to logic 0 during receiver start-up phase
1	ParityErr	1	parity check failed automatically cleared during receiver start-up phase only valid for ISO/IEC 14443 A/MIFARE communication at 106 kBd
0	ProtocolErr	1	set to logic 1 if the SOF is incorrect automatically cleared during receiver start-up phase bit is only valid for 106 kBd during the MFAuthent command, the ProtocolErr bit is set to logic 1 if the number of bytes received in one data stream is incorrect

[1] Command execution clears all error bits except the TempErr bit. Cannot be set by software.

### 9.3.1.8 Status1Reg register

Contains status bits of the CRC, interrupt and FIFO buffer.

**Table 35. Status1Reg register (address 07h); reset value: 21h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved	CRCOk	CRCReady	IRq	TRunning	reserved	HiAlert	LoAlert
Access	-	R	R	R	R	-	R	R

**Table 36. Status1Reg register bit descriptions**

Bit	Symbol	Value	Description
7	reserved	-	reserved for future use
6	CRCOk	1	the CRC result is zero for data transmission and reception, the CRCOk bit is undefined: use the ErrorReg register's CRCErr bit indicates the status of the CRC coprocessor, during calculation the value changes to logic 0, when the calculation is done correctly the value changes to logic 1
5	CRCReady	1	the CRC calculation has finished only valid for the CRC coprocessor calculation using the CalcCRC command
4	IRq	-	indicates if any interrupt source requests attention with respect to the setting of the interrupt enable bits: see the ComIEnReg and DivIEnReg registers
3	TRunning	1	MFRC522's timer unit is running, i.e. the timer will decrement the TCounterValReg register with the next timer clock <b>Remark:</b> in gated mode, the TRunning bit is set to logic 1 when the timer is enabled by TModeReg register's TGated[1:0] bits; this bit is not influenced by the gated signal
2	reserved	-	reserved for future use
1	HiAlert	1	the number of bytes stored in the FIFO buffer corresponds to equation: $HiAlert = (64 - FIFOLength) \leq WaterLevel$ example: FIFO length = 60, WaterLevel = 4 → HiAlert = 1 FIFO length = 59, WaterLevel = 4 → HiAlert = 0
0	LoAlert	1	the number of bytes stored in the FIFO buffer corresponds to equation: $LoAlert = FIFOLength \leq WaterLevel$ example: FIFO length = 4, WaterLevel = 4 → LoAlert = 1 FIFO length = 5, WaterLevel = 4 → LoAlert = 0

**9.3.1.9 Status2Reg register**

Contains status bits of the receiver, transmitter and data mode detector.

**Table 37. Status2Reg register (address 08h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TempSensClear	I2CForceHS	reserved		MFCrypto1On	ModemState[2:0]		
Access	R/W	R/W	-		D	R		

**Table 38. Status2Reg register bit descriptions**

Bit	Symbol	Value	Description
7	TempSensClear	1	clears the temperature error if the temperature is below the alarm limit of 125 °C
6	I2CForceHS		I2C-bus input filter settings:
		1	the I2C-bus input filter is set to the High-speed mode independent of the I2C-bus protocol
		0	the I2C-bus input filter is set to the I2C-bus protocol used
5 to 4	reserved	-	reserved
3	MFCrypto1On	-	indicates that the MIFARE Crypto1 unit is switched on and therefore all data communication with the card is encrypted can only be set to logic 1 by a successful execution of the MFAuthent command only valid in Read/Write mode for MIFARE standard cards this bit is cleared by software
2 to 0	ModemState[2:0]	-	shows the state of the transmitter and receiver state machines:
		000	idle
		001	wait for the BitFramingReg register's StartSend bit
		010	TxWait: wait until RF field is present if the TModeReg register's TxWaitRF bit is set to logic 1 the minimum time for TxWait is defined by the TxWaitReg register
		011	transmitting
		100	RxWait: wait until RF field is present if the TModeReg register's TxWaitRF bit is set to logic 1 the minimum time for RxWait is defined by the RxWaitReg register
		101	wait for data
		110	receiving



### 9.3.1.10 FIFODataReg register

Input and output of 64 byte FIFO buffer.

**Table 39. FIFODataReg register (address 09h); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	FIFOData[7:0]							
Access	D							

**Table 40. FIFODataReg register bit descriptions**

Bit	Symbol	Description
7 to 0	FIFOData[7:0]	data input and output port for the internal 64-byte FIFO buffer FIFO buffer acts as parallel in/parallel out converter for all serial data stream inputs and outputs

### 9.3.1.11 FIFOLevelReg register

Indicates the number of bytes stored in the FIFO.

**Table 41. FIFOLevelReg register (address 0Ah); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	FlushBuffer	FIFOLevel[6:0]						
Access	W	R						

**Table 42. FIFOLevelReg register bit descriptions**

Bit	Symbol	Value	Description
7	FlushBuffer	1	immediately clears the internal FIFO buffer's read and write pointer and ErrorReg register's BufferOvfl bit reading this bit always returns 0
6 to 0	FIFOLevel [6:0]	-	indicates the number of bytes stored in the FIFO buffer writing to the FIFODataReg register increments and reading decrements the FIFOLevel value

### 9.3.1.12 WaterLevelReg register

Defines the level for FIFO under- and overflow warning.

**Table 43. WaterLevelReg register (address 0Bh); reset value: 08h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved			WaterLevel[5:0]				
Access	-			R/W				

Table 44. WaterLevelReg register bit descriptions

Bit	Symbol	Description
7 to 6	reserved	reserved for future use
5 to 0	WaterLevel [5:0]	<p>defines a warning level to indicate a FIFO buffer overflow or underflow:</p> <p>Status1Reg register's HiAlert bit is set to logic 1 if the remaining number of bytes in the FIFO buffer space is equal to, or less than the defined number of WaterLevel bytes</p> <p>Status1Reg register's LoAlert bit is set to logic 1 if equal to, or less than the WaterLevel bytes in the FIFO buffer</p> <p><b>Remark:</b> to calculate values for HiAlert and LoAlert see <a href="#">Section 9.3.1.8 on page 42</a>.</p>

### 9.3.1.13 ControlReg register

Miscellaneous control bits.

Table 45. ControlReg register (address 0Ch); reset value: 10h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	TStopNow	TStartNow	reserved			RxLastBits[2:0]		
Access	W	W	-			R		

Table 46. ControlReg register bit descriptions

Bit	Symbol	Value	Description
7	TStopNow	1	timer stops immediately reading this bit always returns it to logic0
6	TStartNow	1	timer starts immediately reading this bit always returns it to logic 0
5 to 3	reserved	-	reserved for future use
2 to 0	RxLastBits[2:0]	-	indicates the number of valid bits in the last received byte if this value is 000b, the whole byte is valid

### 9.3.1.14 BitFramingReg register

Adjustments for bit-oriented frames.

**Table 47. BitFramingReg register (address 0Dh); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	StartSend	RxAlign[2:0]			reserved	TxLastBits[2:0]		
Access	W	R/W			-	R/W		

**Table 48. BitFramingReg register bit descriptions**

Bit	Symbol	Value	Description
7	StartSend	1	starts the transmission of data only valid in combination with the Transceive command
6 to 4	RxAlign[2:0]		used for reception of bit-oriented frames: defines the bit position for the first bit received to be stored in the FIFO buffer example:
		0	LSB of the received bit is stored at bit position 0, the second received bit is stored at bit position 1
		1	LSB of the received bit is stored at bit position 1, the second received bit is stored at bit position 2
		7	LSB of the received bit is stored at bit position 7, the second received bit is stored in the next byte that follows at bit position 0  These bits are only to be used for bitwise anticollision at 106 kBd, for all other modes they are set to 0
3	reserved	-	reserved for future use
2 to 0	TxLastBits[2:0]	-	used for transmission of bit oriented frames: defines the number of bits of the last byte that will be transmitted 000b indicates that all bits of the last byte will be transmitted

### 9.3.1.15 CollReg register

Defines the first bit-collision detected on the RF interface.

**Table 49. CollReg register (address 0Eh); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	ValuesAfterColl	reserved	CollPosNotValid	CollPos[4:0]				
Access	R/W	-	R	R				

**Table 50. CollReg register bit descriptions**

Bit	Symbol	Value	Description
7	ValuesAfterColl	0	all received bits will be cleared after a collision only used during bitwise anticollision at 106 kBd, otherwise it is set to logic 1
6	reserved	-	reserved for future use
5	CollPosNotValid	1	no collision detected or the position of the collision is out of the range of CollPos[4:0]

Table 50. CollReg register bit descriptions ...continued

Bit	Symbol	Value	Description
4 to 0	CollPos[4:0]	-	shows the bit position of the first detected collision in a received frame only data bits are interpreted example:
		00h	indicates a bit-collision in the 32 <sup>nd</sup> bit
		01h	indicates a bit-collision in the 1 <sup>st</sup> bit
		08h	indicates a bit-collision in the 8 <sup>th</sup> bit These bits will only be interpreted if the CollPosNotValid bit is set to logic 0

### 9.3.1.16 Reserved register 0Fh

Functionality is reserved for future use.

Table 51. Reserved register (address 0Fh); reset value: 00h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

Table 52. Reserved register bit descriptions

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

## 9.3.2 Page 1: Communication

### 9.3.2.1 Reserved register 10h

Functionality is reserved for future use.

Table 53. Reserved register (address 10h); reset value: 00h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

Table 54. Reserved register bit descriptions

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

9.3.2.2 ModeReg register

Defines general mode settings for transmitting and receiving.

Table 55. ModeReg register (address 11h); reset value: 3Fh bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	MSBFirst	reserved	TxWaitRF	reserved	PolMFin	reserved	CRCPreset[1:0]	
Access	R/W	-	R/W	-	R/W	-	R/W	

Table 56. ModeReg register bit descriptions

Bit	Symbol	Value	Description
7	MSBFirst	1	CRC coprocessor calculates the CRC with MSB first in the CRCResultReg register the values for the CRCResultMSB[7:0] bits and the CRCResultLSB[7:0] bits are bit reversed <b>Remark:</b> during RF communication this bit is ignored
6	reserved	-	reserved for future use
5	TxWaitRF	1	transmitter can only be started if an RF field is generated
4	reserved	-	reserved for future use
3	PolMFin		defines the polarity of pin MFIN <b>Remark:</b> the internal envelope signal is encoded active LOW, changing this bit generates a MFinActIRq event
		1	polarity of pin MFIN is active HIGH
		0	polarity of pin MFIN is active LOW
2	reserved	-	reserved for future use
1 to 0	CRCPreset [1:0]		defines the preset value for the CRC coprocessor for the CalcCRC command <b>Remark:</b> during any communication, the preset values are selected automatically according to the definition of bits in the RxModeReg and TxModeReg registers
		00	0000h
		01	6363h
		10	A671h
		11	FFFFh

9.3.2.3 TxModeReg register

Defines the data rate during transmission.

Table 57. TxModeReg register (address 12h); reset value: 00h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	TxCRCEn	TxSpeed[2:0]			InvMod	reserved		
Access	R/W	D			R/W	-		

**Table 58. TxModeReg register bit descriptions**

Bit	Symbol	Value	Description
7	TxCRCEn	1	enables CRC generation during data transmission <b>Remark:</b> can only be set to logic 0 at 106 kBd
6 to 4	TxSpeed[2:0]		defines the bit rate during data transmission the MFRC522 handles transfer speeds up to 848 kBd
		000	106 kBd
		001	212 kBd
		010	424 kBd
		011	848 kBd
		100	reserved
		101	reserved
		110	reserved
3	InvMod	1	modulation of transmitted data is inverted
2 to 0	reserved	-	reserved for future use

**9.3.2.4 RxModeReg register**

Defines the data rate during reception.

**Table 59. RxModeReg register (address 13h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	RxCRCEn	RxSpeed[2:0]			RxNoErr	RxMultiple	reserved	
Access	R/W	D			R/W	R/W	-	

**Table 60. RxModeReg register bit descriptions**

Bit	Symbol	Value	Description
7	RxCRCEn	1	enables the CRC calculation during reception <b>Remark:</b> can only be set to logic 0 at 106 kBd
6 to 4	RxSpeed[2:0]		defines the bit rate while receiving data the MFRC522 handles transfer speeds up to 848 kBd
		000	106 kBd
		001	212 kBd
		010	424 kBd
		011	848 kBd
		100	reserved
		101	reserved
		110	reserved
3	RxNoErr	1	an invalid received data stream (less than 4 bits received) will be ignored and the receiver remains active

Table 60. RxModeReg register bit descriptions ...continued

Bit	Symbol	Value	Description
2	RxMultiple	0	receiver is deactivated after receiving a data frame
		1	able to receive more than one data frame only valid for data rates above 106 kBd in order to handle the polling command after setting this bit the Receive and Transceive commands will not terminate automatically. Multiple reception can only be deactivated by writing any command (except the Receive command) to the CommandReg register, or by the host clearing the bit if set to logic 1, an error byte is added to the FIFO buffer at the end of a received data stream which is a copy of the ErrorReg register value. For the MFRC522 version 2.0 the CRC status is reflected in the signal CRCOK, which indicates the actual status of the CRC coprocessor. For the MFRC522 version 1.0 the CRC status is reflected in the signal CRCErr.
1 to 0	reserved	-	reserved for future use

### 9.3.2.5 TxControlReg register

Controls the logical behavior of the antenna driver pins TX1 and TX2.

Table 61. TxControlReg register (address 14h); reset value: 80h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	InvTx2RF On	InvTx1RF On	InvTx2RF Off	InvTx1RF Off	Tx2CW	reserved	Tx2RFEn	Tx1RFEn
Access	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W

Table 62. TxControlReg register bit descriptions

Bit	Symbol	Value	Description
7	InvTx2RFOn	1	output signal on pin TX2 inverted when driver TX2 is enabled
6	InvTx1RFOn	1	output signal on pin TX1 inverted when driver TX1 is enabled
5	InvTx2RFOff	1	output signal on pin TX2 inverted when driver TX2 is disabled
4	InvTx1RFOff	1	output signal on pin TX1 inverted when driver TX1 is disabled
3	Tx2CW	1	output signal on pin TX2 continuously delivers the unmodulated 13.56 MHz energy carrier
		0	Tx2CW bit is enabled to modulate the 13.56 MHz energy carrier
2	reserved	-	reserved for future use
1	Tx2RFEn	1	output signal on pin TX2 delivers the 13.56 MHz energy carrier modulated by the transmission data
0	Tx1RFEn	1	output signal on pin TX1 delivers the 13.56 MHz energy carrier modulated by the transmission data

9.3.2.6 TxASKReg register

Controls transmit modulation settings.

Table 63. TxASKReg register (address 15h); reset value: 00h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved	Force100ASK	reserved					
Access	-	R/W	-					

Table 64. TxASKReg register bit descriptions

Bit	Symbol	Value	Description
7	reserved	-	reserved for future use
6	Force100ASK	1	forces a 100 % ASK modulation independent of the ModGsPReg register setting
5 to 0	reserved	-	reserved for future use

9.3.2.7 TxSelReg register

Selects the internal sources for the analog module.

Table 65. TxSelReg register (address 16h); reset value: 10h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol:	reserved		DriverSel[1:0]		MFOutSel[3:0]			
Access:	-		R/W		R/W			

Table 66. TxSelReg register bit descriptions

Bit	Symbol	Value	Description
7 to 6	reserved	-	reserved for future use
5 to 4	DriverSel [1:0]	-	selects the input of drivers TX1 and TX2
		00	3-state; in soft power-down the drivers are only in 3-state mode if the DriverSel[1:0] value is set to 3-state mode
		01	modulation signal (envelope) from the internal encoder, Miller pulse encoded
		10	modulation signal (envelope) from pin MFIN
		11	HIGH; the HIGH level depends on the setting of bits InvTx1RFOn/InvTx1RFOff and InvTx2RFOn/InvTx2RFOff



**Table 66. TxSelReg register bit descriptions ...continued**

Bit	Symbol	Value	Description
3 to 0	MFOutSel [3:0]		selects the input for pin MFOUT
		0000	3-state
		0001	LOW
		0010	HIGH
		0011	test bus signal as defined by the TestSel1Reg register's TstBusBitSel[2:0] value
		0100	modulation signal (envelope) from the internal encoder, Miller pulse encoded
		0101	serial data stream to be transmitted, data stream before Miller encoder
		0110	reserved
		0111	serial data stream received, data stream after Manchester decoder
1000 to 1111		reserved	

**9.3.2.8 RxSelReg register**

Selects internal receiver settings.

**Table 67. RxSelReg register (address 17h); reset value: 84h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	UARTSel[1:0]			RxWait[5:0]				
Access	R/W			R/W				

**Table 68. RxSelReg register bit descriptions**

Bit	Symbol	Value	Description
7 to 6	UARTSel [1:0]	00	selects the input of the contactless UART constant LOW
		01	Manchester with subcarrier from pin MFIN
		10	modulated signal from the internal analog module, default
		11	NRZ coding without subcarrier from pin MFIN which is only valid for transfer speeds above 106 kBd
5 to 0	RxWait [5:0]	-	after data transmission the activation of the receiver is delayed for RxWait bit-clocks, during this 'frame guard time' any signal on pin RX is ignored  this parameter is ignored by the Receive command all other commands, such as Transceive, MFAuthent use this parameter  the counter starts immediately after the external RF field is switched on

### 9.3.2.9 RxThresholdReg register

Selects thresholds for the bit decoder.

**Table 69. RxThresholdReg register (address 18h); reset value: 84h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	MinLevel[3:0]				reserved	CollLevel[2:0]		
Access	R/W				-	R/W		

**Table 70. RxThresholdReg register bit descriptions**

Bit	Symbol	Description
7 to 4	MinLevel [3:0]	defines the minimum signal strength at the decoder input that will be accepted if the signal strength is below this level it is not evaluated
3	reserved	reserved for future use
2 to 0	CollLevel [2:0]	defines the minimum signal strength at the decoder input that must be reached by the weaker half-bit of the Manchester encoded signal to generate a bit-collision relative to the amplitude of the stronger half-bit

### 9.3.2.10 DemodReg register

Defines demodulator settings.

**Table 71. DemodReg register (address 19h); reset value: 4Dh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	AddIQ[1:0]		FixIQ	TPrescal Even	TauRcv[1:0]		TauSync[1:0]	
Access	R/W		R/W	R/W	R/W		R/W	

**Table 72. DemodReg register bit descriptions**

Bit	Symbol	Value	Description
7 to 6	AddIQ [1:0]	-	defines the use of I and Q channel during reception <b>Remark:</b> the FixIQ bit must be set to logic 0 to enable the following settings:
		00	selects the stronger channel
		01	selects the stronger channel and freezes the selected channel during communication
		10	reserved
		11	reserved
5	FixIQ	1	if AddIQ[1:0] are set to X0b, the reception is fixed to I channel if AddIQ[1:0] are set to X1b, the reception is fixed to Q channel

Table 72. DemodReg register bit descriptions ...continued

Bit	Symbol	Value	Description
4	TPrescalEven	R/W	Available on RC522 version 1.0 and version 2.0: If set to logic 0 the following formula is used to calculate the timer frequency of the prescaler: $f_{\text{timer}} = 13.56 \text{ MHz} / (2 * \text{TPreScaler} + 1)$ . Only available on version 2.0: If set to logic 1 the following formula is used to calculate the timer frequency of the prescaler: $f_{\text{timer}} = 13.56 \text{ MHz} / (2 * \text{TPreScaler} + 2)$ . Default TPrescalEven bit is logic 0, find more information on the prescaler in <a href="#">Section 8.5</a> .
3 to 2	TauRcv[1:0]	-	changes the time-constant of the internal PLL during data reception <b>Remark:</b> if set to 00b the PLL is frozen during data reception
1 to 0	TauSync[1:0]	-	changes the time-constant of the internal PLL during burst

### 9.3.2.11 Reserved register 1Ah

Functionality is reserved for future use.

Table 73. Reserved register (address 1Ah); reset value: 00h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

Table 74. Reserved register bit descriptions

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

### 9.3.2.12 Reserved register 1Bh

Functionality is reserved for future use.

Table 75. Reserved register (address 1Bh); reset value: 00h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

Table 76. Reserved register bit descriptions

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

### 9.3.2.13 MfTxReg register

Controls some MIFARE communication transmit parameters.

**Table 77. MfTxReg register (address 1Ch); reset value: 62h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved						TxWait[1:0]	
Access	-						R/W	

**Table 78. MfTxReg register bit descriptions**

Bit	Symbol	Description
7 to 2	reserved	reserved for future use
1 to 0	TxWait	defines the additional response time 7 bits are added to the value of the register bit by default

### 9.3.2.14 MfRxReg register

**Table 79. MfRxReg register (address 1Dh); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved			ParityDisable	reserved			
Access	-			R/W	-			

**Table 80. MfRxReg register bit descriptions**

Bit	Symbol	Value	Description
7 to 5	reserved	-	reserved for future use
4	ParityDisable	1	generation of the parity bit for transmission and the parity check for receiving is switched off the received parity bit is handled like a data bit
3 to 0	reserved	-	reserved for future use

### 9.3.2.15 Reserved register 1Eh

Functionality is reserved for future use.

**Table 81. Reserved register (address 1Eh); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

**Table 82. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

### 9.3.2.16 SerialSpeedReg register

Selects the speed of the serial UART interface.

**Table 83. SerialSpeedReg register (address 1Fh); reset value: EBh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	BR_T0[2:0]			BR_T1[4:0]				
Access	R/W			R/W				

**Table 84. SerialSpeedReg register bit descriptions**

Bit	Symbol	Description
7 to 5	BR_T0[2:0]	factor BR_T0 adjusts the transfer speed: for description, see <a href="#">Section 8.1.3.2 on page 12</a>
4 to 0	BR_T1[4:0]	factor BR_T1 adjusts the transfer speed: for description, see <a href="#">Section 8.1.3.2 on page 12</a>

### 9.3.3 Page 2: Configuration

#### 9.3.3.1 Reserved register 20h

Functionality is reserved for future use.

**Table 85. Reserved register (address 20h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	-							
Access	reserved							

**Table 86. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

#### 9.3.3.2 CRCResultReg registers

Shows the MSB and LSB values of the CRC calculation.

**Remark:** The CRC is split into two 8-bit registers.

**Table 87. CRCResultReg (higher bits) register (address 21h); reset value: FFh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	CRCResultMSB[7:0]							
Access	R							

**Table 88. CRCResultReg register higher bit descriptions**

Bit	Symbol	Description
7 to 0	CRCResultMSB [7:0]	shows the value of the CRCResultReg register's most significant byte only valid if Status1Reg register's CRCReady bit is set to logic 1

**Table 89. CRCResultReg (lower bits) register (address 22h); reset value: FFh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	CRCResultLSB[7:0]							
Access	R							

**Table 90. CRCResultReg register lower bit descriptions**

Bit	Symbol	Description
7 to 0	CRCResultLSB [7:0]	shows the value of the least significant byte of the CRCResultReg register only valid if Status1Reg register's CRCReady bit is set to logic 1

### 9.3.3.3 Reserved register 23h

Functionality is reserved for future use.

**Table 91. Reserved register (address 23h); reset value: 88h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

**Table 92. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

### 9.3.3.4 ModWidthReg register

Sets the modulation width.

**Table 93. ModWidthReg register (address 24h); reset value: 26h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	ModWidth[7:0]							
Access	R/W							

**Table 94. ModWidthReg register bit descriptions**

Bit	Symbol	Description
7 to 0	ModWidth[7:0]	defines the width of the Miller modulation as multiples of the carrier frequency ( $\text{ModWidth} + 1 / f_{\text{clk}}$ ) the maximum value is half the bit period

### 9.3.3.5 Reserved register 25h

Functionality is reserved for future use.

**Table 95. Reserved register (address 25h); reset value: 87h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

**Table 96. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

9.3.3.6 RFCfgReg register

Configures the receiver gain.

Table 97. RFCfgReg register (address 26h); reset value: 48h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved	RxGain[2:0]			reserved			
Access	-	R/W			-			

Table 98. RFCfgReg register bit descriptions

Bit	Symbol	Value	Description
7	reserved	-	reserved for future use
6 to 4	RxGain [2:0]		defines the receiver's signal voltage gain factor:
		000	18 dB
		001	23 dB
		010	18 dB
		011	23 dB
		100	33 dB
		101	38 dB
		110	43 dB
111	48 dB		
3 to 0	reserved	-	reserved for future use

9.3.3.7 GsNReg register

Defines the conductance of the antenna driver pins TX1 and TX2 for the n-driver when the driver is switched on.

Table 99. GsNReg register (address 27h); reset value: 88h bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	CWGsN[3:0]				ModGsN[3:0]			
Access	R/W				R/W			

Table 100. GsNReg register bit descriptions

Bit	Symbol	Description
7 to 4	CWGsN [3:0]	<p>defines the conductance of the output n-driver during periods without modulation which can be used to regulate the output power and subsequently current consumption and operating distance</p> <p><b>Remark:</b> the conductance value is binary-weighted during soft Power-down mode the highest bit is forced to logic 1 value is only used if driver TX1 or TX2 is switched on</p>
3 to 0	ModGsN [3:0]	<p>defines the conductance of the output n-driver during periods without modulation which can be used to regulate the modulation index</p> <p><b>Remark:</b> the conductance value is binary weighted during soft Power-down mode the highest bit is forced to logic 1 value is only used if driver TX1 or TX2 is switched on</p>



### 9.3.3.8 CWGsPReg register

Defines the conductance of the p-driver output during periods of no modulation.

**Table 101. CWGsPReg register (address 28h); reset value: 20h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved		CWGsP[5:0]					
Access	-		R/W					

**Table 102. CWGsPReg register bit descriptions**

Bit	Symbol	Description
7 to 6	reserved	reserved for future use
5 to 0	CWGsP[5:0]	defines the conductance of the p-driver output which can be used to regulate the output power and subsequently current consumption and operating distance <b>Remark:</b> the conductance value is binary weighted during soft Power-down mode the highest bit is forced to logic 1

### 9.3.3.9 ModGsPReg register

Defines the conductance of the p-driver output during modulation.

**Table 103. ModGsPReg register (address 29h); reset value: 20h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved		ModGsP[5:0]					
Access	-		R/W					

**Table 104. ModGsPReg register bit descriptions**

Bit	Symbol	Description
7 to 6	reserved	reserved for future use
5 to 0	ModGsP[5:0]	defines the conductance of the p-driver output during modulation which can be used to regulate the modulation index <b>Remark:</b> the conductance value is binary weighted during soft Power-down mode the highest bit is forced to logic 1 if the TxASKReg register's Force100ASK bit is set to logic 1 the value of ModGsP has no effect

### 9.3.3.10 TModeReg and TPrescalerReg registers

These registers define the timer settings.

**Remark:** The TPrescaler setting higher 4 bits are in the TModeReg register and the lower 8 bits are in the TPrescalerReg register.

**Table 105. TModeReg register (address 2Ah); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TAuto	TGated[1:0]		TAutoRestart	TPrescaler_Hi[3:0]			
Access	R/W	R/W		R/W	R/W			

**Table 106. TModeReg register bit descriptions**

Bit	Symbol	Value	Description
7	TAuto	1	timer starts automatically at the end of the transmission in all communication modes at all speeds  if the RxModeReg register's RxMultiple bit is not set, the timer stops immediately after receiving the 5th bit (1 start bit, 4 data bits)  if the RxMultiple bit is set to logic 1 the timer never stops, in which case the timer can be stopped by setting the ControlReg register's TStopNow bit to logic 1
		0	indicates that the timer is not influenced by the protocol
6 to 5	TGated[1:0]		internal timer is running in gated mode <b>Remark:</b> in gated mode, the Status1Reg register's TRunning bit is logic 1 when the timer is enabled by the TModeReg register's TGated[1:0] bits this bit does not influence the gating signal
		00	non-gated mode
		01	gated by pin MFIN
		10	gated by pin AUX1
4	TAutoRestart	1	timer automatically restarts its count-down from the 16-bit timer reload value instead of counting down to zero
		0	timer decrements to 0 and the ComlRqReg register's TimerlRq bit is set to logic 1
3 to 0	TPrescaler_Hi[3:0]	-	defines the higher 4 bits of the TPrescaler value The following formula is used to calculate the timer frequency if the DemodReg register's TPrescalEven bit in Demot Regis set to logic 0: $f_{\text{timer}} = 13.56 \text{ MHz} / (2 * \text{TPreScaler} + 1).$ Where TPreScaler = [TPrescaler_Hi:TPrescaler_Lo] (TPrescaler value on 12 bits) (Default TPrescalEven bit is logic 0) The following formula is used to calculate the timer frequency if the DemodReg register's TPrescalEven bit is set to logic 1: $f_{\text{timer}} = 13.56 \text{ MHz} / (2 * \text{TPreScaler} + 2).$ See <a href="#">Section 8.5 "Timer unit"</a> .

**Table 107. TPrescalerReg register (address 2Bh); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TPrescaler_Lo[7:0]							
Access	R/W							

**Table 108. TPrescalerReg register bit descriptions**

Bit	Symbol	Description
7 to 0	TPrescaler_Lo[7:0]	<p>defines the lower 8 bits of the TPrescaler value</p> <p>The following formula is used to calculate the timer frequency if the DemodReg register's TPrescalEven bit is set to logic 0:</p> $f_{\text{timer}} = 13.56 \text{ MHz} / (2 * \text{TPreScaler} + 1).$ <p>Where TPreScaler = [TPrescaler_Hi:TPrescaler_Lo] (TPrescaler value on 12 bits) (Default TPrescalEven bit is logic 0)</p> <p>The following formula is used to calculate the timer frequency if the DemodReg register's TPrescalEven bit in DemoReg is set to logic 1:</p> $f_{\text{timer}} = 13.56 \text{ MHz} / (2 * \text{TPreScaler} + 2).$ <p>See <a href="#">Section 8.5 "Timer unit"</a>.</p>

**9.3.3.11 TReloadReg register**

Defines the 16-bit timer reload value.

**Remark:** The reload value bits are contained in two 8-bit registers.

**Table 109. TReloadReg (higher bits) register (address 2Ch); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TReloadVal_Hi[7:0]							
Access	R/W							

**Table 110. TReloadReg register higher bit descriptions**

Bit	Symbol	Description
7 to 0	TReloadVal_Hi[7:0]	<p>defines the higher 8 bits of the 16-bit timer reload value</p> <p>on a start event, the timer loads the timer reload value</p> <p>changing this register affects the timer only at the next start event</p>

**Table 111. TReloadReg (lower bits) register (address 2Dh); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TReloadVal_Lo[7:0]							
Access	R/W							

**Table 112. TReloadReg register lower bit descriptions**

Bit	Symbol	Description
7 to 0	TReloadVal_Lo[7:0]	<p>defines the lower 8 bits of the 16-bit timer reload value</p> <p>on a start event, the timer loads the timer reload value</p> <p>changing this register affects the timer only at the next start event</p>

**9.3.3.12 TCounterValReg register**

Contains the timer value.

**Remark:** The timer value bits are contained in two 8-bit registers.

**Table 113. TCounterValReg (higher bits) register (address 2Eh); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TCounterVal_Hi[7:0]							
Access	R							

**Table 114. TCounterValReg register higher bit descriptions**

Bit	Symbol	Description
7 to 0	TCounterVal_Hi [7:0]	timer value higher 8 bits

**Table 115. TCounterValReg (lower bits) register (address 2Fh); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TCounterVal_Lo[7:0]							
Access	R							

**Table 116. TCounterValReg register lower bit descriptions**

Bit	Symbol	Description
7 to 0	TCounterVal_Lo [7:0]	timer value lower 8 bits

### 9.3.4 Page 3: Test

#### 9.3.4.1 Reserved register 30h

Functionality is reserved for future use.

**Table 117. Reserved register (address 30h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

**Table 118. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for future use

#### 9.3.4.2 TestSel1Reg register

General test signal configuration.

**Table 119. TestSel1Reg register (address 31h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved					TstBusBitSel[2:0]		
Access	-					R/W		

**Table 120. TestSel1Reg register bit descriptions**

Bit	Symbol	Description
7 to 3	reserved	reserved for future use
2 to 0	TstBusBitSel [2:0]	selects a test bus signal which is output at pin MFOUT if AnalogSelAux2[3:0] = FFh in AnalogTestReg register, test bus signal is also output at pins AUX1 or AUX2

**9.3.4.3 TestSel2Reg register**

General test signal configuration and PRBS control.

**Table 121. TestSel2Reg register (address 32h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TstBusFlip	PRBS9	PRBS15	TestBusSel[4:0]				
Access	R/W	R/W	R/W	R/W				

**Table 122. TestSel2Reg register bit descriptions**

Bit	Symbol	Value	Description
7	TstBusFlip	1	test bus is mapped to the parallel port in the following order: TstBusBit4, TstBusBit3, TstBusBit2, TstBusBit6, TstBusBit5, TstBusBit0; see <a href="#">Section 16.1 on page 82</a>
6	PRBS9	-	starts and enables the PRBS9 sequence according to ITU-TO150 <b>Remark:</b> all relevant registers to transmit data must be configured before entering PRBS9 mode the data transmission of the defined sequence is started by the Transmit command
5	PRBS15	-	starts and enables the PRBS15 sequence according to ITU-TO150 <b>Remark:</b> all relevant registers to transmit data must be configured before entering PRBS15 mode the data transmission of the defined sequence is started by the Transmit command
4 to 0	TestBusSel[4:0]	-	selects the test bus; see <a href="#">Section 16.1 "Test signals"</a>

**9.3.4.4 TestPinEnReg register**

Enables the test bus pin output driver.

**Table 123. TestPinEnReg register (address 33h); reset value: 80h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	RS232LineEn	TestPinEn[5:0]						reserved
Access	R/W	R/W						-

**Table 124. TestPinEnReg register bit descriptions**

Bit	Symbol	Value	Description
7	RS232LineEn	0	serial UART lines MX and DTRQ are disabled
6 to 1	TestPinEn [5:0]	-	enables the output driver on one of the data pins D1 to D7 which outputs a test signal <b>Example:</b> setting bit 1 to logic 1 enables pin D1 output setting bit 5 to logic 1 enables pin D5 output <b>Remark:</b> If the SPI is used, only pins D1 to D4 can be used. If the serial UART interface is used and the RS232LineEn bit is set to logic 1 only pins D1 to D4 can be used.
0	reserved	-	reserved for future use

**9.3.4.5 TestPinValueReg register**

Defines the HIGH and LOW values for the test port D1 to D7 when it is used as I/O.

**Table 125. TestPinValueReg register (address 34h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	UseIO	TestPinValue[5:0]						reserved
Access	R/W	R/W						-

**Table 126. TestPinValueReg register bit descriptions**

Bit	Symbol	Value	Description
7	UseIO	1	enables the I/O functionality for the test port when one of the serial interfaces is used the input/output behavior is defined by value TestPinEn[5:0] in the TestPinEnReg register the value for the output behavior is defined by TestPinValue[5:0]
6 to 1	TestPinValue [5:0]	-	defines the value of the test port when it is used as I/O and each output must be enabled by TestPinEn[5:0] in the TestPinEnReg register <b>Remark:</b> Reading the register indicates the status of pins D6 to D1 if the UseIO bit is set to logic 1. If the UseIO bit is set to logic 0, the value of the TestPinValueReg register is read back.
0	reserved	-	reserved for future use

**9.3.4.6 TestBusReg register**

Shows the status of the internal test bus.

**Table 127. TestBusReg register (address 35h); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	TestBus[7:0]							
Access	R							

**Table 128. TestBusReg register bit descriptions**

Bit	Symbol	Description
7 to 0	TestBus[7:0]	shows the status of the internal test bus the test bus is selected using the TestSel2Reg register; see <a href="#">Section 16.1 on page 82</a>

**9.3.4.7 AutoTestReg register**

Controls the digital self-test.

**Table 129. AutoTestReg register (address 36h); reset value: 40h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved	AmpRcv	RFT		SelfTest[3:0]			
Access	-	R/W	-		R/W			

**Table 130. AutoTestReg register bit descriptions**

Bit	Symbol	Value	Description
7	reserved	-	reserved for production tests
6	AmpRcv	1	internal signal processing in the receiver chain is performed non-linearly which increases the operating distance in communication modes at 106 kBd <b>Remark:</b> due to non-linearity, the effect of the RxThresholdReg register's MinLevel[3:0] and the CollLevel[2:0] values is also non-linear
5 to 4	RFT	-	reserved for production tests
3 to 0	SelfTest[3:0]	-	enables the digital self test the self test can also be started by the CalcCRC command; see <a href="#">Section 10.3.1.4 on page 71</a> the self test is enabled by value 1001b <b>Remark:</b> for default operation the self test must be disabled by value 0000b

**9.3.4.8 VersionReg register**

Shows the MFRC522 software version.

**Table 131. VersionReg register (address 37h); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	Version[7:0]							
Access	R							

**Table 132. VersionReg register bit descriptions**

Bit	Symbol	Description
7 to 4	Chiptype	'9' stands for MFRC522
3 to 0	Version	'1' stands for MFRC522 version 1.0 and '2' stands for MFRC522 version 2.0.

MFRC522 version 1.0 software version is: 91h.

MFRC522 version 2.0 software version is: 92h.

### 9.3.4.9 AnalogTestReg register

Determines the analog output test signal at, and status of, pins AUX1 and AUX2.

**Table 133. AnalogTestReg register (address 38h); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	AnalogSelAux1[3:0]				AnalogSelAux2[3:0]			
Access	R/W				R/W			

**Table 134. AnalogTestReg register bit descriptions**

Bit	Symbol	Value	Description
7 to 4	AnalogSelAux1 [3:0]		controls pin AUX1
		0000	3-state
		0001	output of TestDAC1 (AUX1), output of TestDAC2 (AUX2) <sup>[1]</sup>
		0010	test signal Corr1 <sup>[1]</sup>
		0011	reserved
		0100	DAC: test signal MinLevel <sup>[1]</sup>
		0101	DAC: test signal ADC_I <sup>[1]</sup>
		0110	DAC: test signal ADC_Q <sup>[1]</sup>
		0111	reserved
		1000	reserved, test signal for production test <sup>[1]</sup>
		1001	reserved
		1010	HIGH
		1011	LOW
		1100	TxActive: at 106 kBd: HIGH during Start bit, Data bit, Parity and CRC at 212 kBd: 424 kBd and 848 kBd: HIGH during data and CRC
1101	RxActive: at 106 kBd: HIGH during Data bit, Parity and CRC at 212 kBd: 424 kBd and 848 kBd: HIGH during data and CRC		
1110	subcarrier detected: 106 kBd: not applicable 212 kBd: 424 kBd and 848 kBd: HIGH during last part of data and CRC		
1111	test bus bit as defined by the TestSel1Reg register's TstBusBitSel[2:0] bits <b>Remark:</b> all test signals are described in <a href="#">Section 16.1 on page 82</a>		
3 to 0	AnalogSelAux2 [3:0]	-	controls pin AUX2 (see bit descriptions for AUX1)

[1] **Remark:** Current source output; the use of 1 kΩ pull-down resistor on AUXn is recommended.



### 9.3.4.10 TestDAC1Reg register

Defines the test value for TestDAC1.

**Table 135. TestDAC1Reg register (address 39h); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved		TestDAC1[5:0]					
Access	-		R/W					

**Table 136. TestDAC1Reg register bit descriptions**

Bit	Symbol	Description
7	reserved	reserved for production tests
6	reserved	reserved for future use
5 to 0	TestDAC1[5:0]	defines the test value for TestDAC1 output of DAC1 can be routed to AUX1 by setting value AnalogSelAux1[3:0] to 0001b in the AnalogTestReg register

### 9.3.4.11 TestDAC2Reg register

Defines the test value for TestDAC2.

**Table 137. TestDAC2Reg register (address 3Ah); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved		TestDAC2[5:0]					
Access	-		R/W					

**Table 138. TestDAC2Reg register bit descriptions**

Bit	Symbol	Description
7 to 6	reserved	reserved for future use
5 to 0	TestDAC2[5:0]	defines the test value for TestDAC2 output of DAC2 can be routed to AUX2 by setting value AnalogSelAux2[3:0] to 0001b in the AnalogTestReg register

### 9.3.4.12 TestADCReg register

Shows the values of ADC I and Q channels.

**Table 139. TestADCReg register (address 3Bh); reset value: xxh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	ADC_I[3:0]				ADC_Q[3:0]			
Access	R				R			

**Table 140. TestADCReg register bit descriptions**

Bit	Symbol	Description
7 to 4	ADC_I[3:0]	ADC I channel value
3 to 0	ADC_Q[3:0]	ADC Q channel value

### 9.3.4.13 Reserved register 3Ch

Functionality reserved for production test.

**Table 141. Reserved register (address 3Ch); reset value: FFh bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	RFT							
Access	-							

**Table 142. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for production tests

**Table 143. Reserved register (address 3Dh); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	RFT							
Access	-							

**Table 144. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for production tests

**Table 145. Reserved register (address 3Eh); reset value: 03h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	RFT							
Access	-							

**Table 146. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for production tests

**Table 147. Reserved register (address 3Fh); reset value: 00h bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved							
Access	-							

**Table 148. Reserved register bit descriptions**

Bit	Symbol	Description
7 to 0	reserved	reserved for production tests

## 10. MFRC522 command set

### 10.1 General description

The MFRC522 operation is determined by a state machine capable of performing a set of commands. A command is executed by writing a command code (see [Table 149](#)) to the CommandReg register.

Arguments and/or data necessary to process a command are exchanged via the FIFO buffer.

### 10.2 General behavior

- Each command that needs a data bit stream (or data byte stream) as an input immediately processes any data in the FIFO buffer. An exception to this rule is the Transceive command. Using this command, transmission is started with the BitFramingReg register's StartSend bit.
- Each command that needs a certain number of arguments, starts processing only when it has received the correct number of arguments from the FIFO buffer.
- The FIFO buffer is not automatically cleared when commands start. This makes it possible to write command arguments and/or the data bytes to the FIFO buffer and then start the command.
- Each command can be interrupted by the host writing a new command code to the CommandReg register, for example, the Idle command.

### 10.3 MFRC522 command overview

Table 149. Command overview

Command	Command code	Action
Idle	0000	no action, cancels current command execution
Mem	0001	stores 25 bytes into the internal buffer
Generate RandomID	0010	generates a 10-byte random ID number
CalcCRC	0011	activates the CRC coprocessor or performs a self test
Transmit	0100	transmits data from the FIFO buffer
NoCmdChange	0111	no command change, can be used to modify the CommandReg register bits without affecting the command, for example, the PowerDown bit
Receive	1000	activates the receiver circuits
Transceive	1100	transmits data from FIFO buffer to antenna and automatically activates the receiver after transmission
-	1101	reserved for future use
MFAuthent	1110	performs the MIFARE standard authentication as a reader
SoftReset	1111	resets the MFRC522

### 10.3.1 MFRC522 command descriptions

#### 10.3.1.1 Idle

Places the MFRC522 in Idle mode. The Idle command also terminates itself.

#### 10.3.1.2 Mem

Transfers 25 bytes from the FIFO buffer to the internal buffer.

To read out the 25 bytes from the internal buffer the Mem command must be started with an empty FIFO buffer. In this case, the 25 bytes are transferred from the internal buffer to the FIFO.

During a hard power-down (using pin NRSTPD), the 25 bytes in the internal buffer remain unchanged and are only lost if the power supply is removed from the MFRC522.

This command automatically terminates when finished and the Idle command becomes active.

#### 10.3.1.3 Generate RandomID

This command generates a 10-byte random number which is initially stored in the internal buffer. This then overwrites the 10 bytes in the internal 25-byte buffer. This command automatically terminates when finished and the MFRC522 returns to Idle mode.

#### 10.3.1.4 CalcCRC

The FIFO buffer content is transferred to the CRC coprocessor and the CRC calculation is started. The calculation result is stored in the CRCResultReg register. The CRC calculation is not limited to a dedicated number of bytes. The calculation is not stopped when the FIFO buffer is empty during the data stream. The next byte written to the FIFO buffer is added to the calculation.

The CRC preset value is controlled by the ModeReg register's CRCPreset[1:0] bits. The value is loaded in to the CRC coprocessor when the command starts.

This command must be terminated by writing a command to the CommandReg register, such as, the Idle command.

If the AutoTestReg register's SelfTest[3:0] bits are set correctly, the MFRC522 enters Self Test mode. Starting the CalcCRC command initiates a digital self test. The result of the self test is written to the FIFO buffer.

#### 10.3.1.5 Transmit

The FIFO buffer content is immediately transmitted after starting this command. Before transmitting the FIFO buffer content, all relevant registers must be set for data transmission.

This command automatically terminates when the FIFO buffer is empty. It can be terminated by another command written to the CommandReg register.

#### 10.3.1.6 NoCmdChange

This command does not influence any running command in the CommandReg register. It can be used to manipulate any bit except the CommandReg register Command[3:0] bits, for example, the RcvOff bit or the PowerDown bit.

### 10.3.1.7 Receive

The MFRC522 activates the receiver path and waits for a data stream to be received. The correct settings must be chosen before starting this command.

This command automatically terminates when the data stream ends. This is indicated either by the end of frame pattern or by the length byte depending on the selected frame type and speed.

**Remark:** If the RxModeReg register's RxMultiple bit is set to logic 1, the Receive command will not automatically terminate. It must be terminated by starting another command in the CommandReg register.

### 10.3.1.8 Transceive

This command continuously repeats the transmission of data from the FIFO buffer and the reception of data from the RF field. The first action is transmit and after transmission the command is changed to receive a data stream.

Each transmit process must be started by setting the BitFramingReg register's StartSend bit to logic 1. This command must be cleared by writing any command to the CommandReg register.

**Remark:** If the RxModeReg register's RxMultiple bit is set to logic 1, the Transceive command never leaves the receive state because this state cannot be cancelled automatically.

### 10.3.1.9 MFAuthent

This command manages MIFARE authentication to enable a secure communication to any MIFARE Mini, MIFARE 1K and MIFARE 4K card. The following data is written to the FIFO buffer before the command can be activated:

- Authentication command code (60h, 61h)
- Block address
- Sector key byte 0
- Sector key byte 1
- Sector key byte 2
- Sector key byte 3
- Sector key byte 4
- Sector key byte 5
- Card serial number byte 0
- Card serial number byte 1
- Card serial number byte 2
- Card serial number byte 3

In total 12 bytes are written to the FIFO.

**Remark:** When the MFAuthent command is active all access to the FIFO buffer is blocked. However, if there is access to the FIFO buffer, the ErrorReg register's WrErr bit is set.

This command automatically terminates when the MIFARE card is authenticated and the Status2Reg register's MFCrypto1On bit is set to logic 1.

This command does not terminate automatically if the card does not answer, so the timer must be initialized to automatic mode. In this case, in addition to the IdleIRq bit, the TimerIRq bit can be used as the termination criteria. During authentication processing, the RxIRq bit and TxIRq bit are blocked. The Crypto1On bit is only valid after termination of the MFAuthent command, either after processing the protocol or writing Idle to the CommandReg register.

If an error occurs during authentication, the ErrorReg register's ProtocolErr bit is set to logic 1 and the Status2Reg register's Crypto1On bit is set to logic 0.

#### 10.3.1.10 SoftReset

This command performs a reset of the device. The configuration data of the internal buffer remains unchanged. All registers are set to the reset values. This command automatically terminates when finished.

**Remark:** The SerialSpeedReg register is reset and therefore the serial data rate is set to 9.6 kBd.

## 11. Limiting values

**Table 150. Limiting values**

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions	Min	Max	Unit
V <sub>DDA</sub>	analog supply voltage		-0.5	+4.0	V
V <sub>DDD</sub>	digital supply voltage		-0.5	+4.0	V
V <sub>DD(PVDD)</sub>	PVDD supply voltage		-0.5	+4.0	V
V <sub>DD(TVDD)</sub>	TVDD supply voltage		-0.5	+4.0	V
V <sub>DD(SVDD)</sub>	SVDD supply voltage		-0.5	+4.0	V
V <sub>i</sub>	input voltage	all input pins except pins MFIN and RX	V <sub>SS(PVSS)</sub> - 0.5	V <sub>DD(PVDD)</sub> + 0.5	V
		pin MFIN	V <sub>SS(PVSS)</sub> - 0.5	V <sub>DD(SVDD)</sub> + 0.5	V
P <sub>tot</sub>	total power dissipation	per package; and V <sub>DDD</sub> in shortcut mode	-	200	mW
T <sub>j</sub>	junction temperature		-	100	°C
V <sub>ESD</sub>	electrostatic discharge voltage	HBM; 1500 Ω, 100 pF; JESD22-A114-B	-	2000	V
		MM; 0.75 μH, 200 pF; JESD22-A114-A	-	200	V
		Charged device model; JESD22-C101-A			
		on all pins	-	200	V
		on all pins except SVDD in TFBGA64 package	-	500	V

## 12. Recommended operating conditions

**Table 151. Operating conditions**

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>DDA</sub>	analog supply voltage	V <sub>DD(PVDD)</sub> ≤ V <sub>DDA</sub> = V <sub>DDD</sub> = V <sub>DD(TVDD)</sub> ; V <sub>SSA</sub> = V <sub>SSD</sub> = V <sub>SS(PVSS)</sub> = V <sub>SS(TVSS)</sub> = 0 V	[1][2] 2.5	3.3	3.6	V
V <sub>DDD</sub>	digital supply voltage	V <sub>DD(PVDD)</sub> ≤ V <sub>DDA</sub> = V <sub>DDD</sub> = V <sub>DD(TVDD)</sub> ; V <sub>SSA</sub> = V <sub>SSD</sub> = V <sub>SS(PVSS)</sub> = V <sub>SS(TVSS)</sub> = 0 V	[1][2] 2.5	3.3	3.6	V
V <sub>DD(TVDD)</sub>	TVDD supply voltage	V <sub>DD(PVDD)</sub> ≤ V <sub>DDA</sub> = V <sub>DDD</sub> = V <sub>DD(TVDD)</sub> ; V <sub>SSA</sub> = V <sub>SSD</sub> = V <sub>SS(PVSS)</sub> = V <sub>SS(TVSS)</sub> = 0 V	[1][2] 2.5	3.3	3.6	V
V <sub>DD(PVDD)</sub>	PVDD supply voltage	V <sub>DD(PVDD)</sub> ≤ V <sub>DDA</sub> = V <sub>DDD</sub> = V <sub>DD(TVDD)</sub> ; V <sub>SSA</sub> = V <sub>SSD</sub> = V <sub>SS(PVSS)</sub> = V <sub>SS(TVSS)</sub> = 0 V	[3] 1.6	1.8	3.6	V
V <sub>DD(SVDD)</sub>	SVDD supply voltage	V <sub>SSA</sub> = V <sub>SSD</sub> = V <sub>SS(PVSS)</sub> = V <sub>SS(TVSS)</sub> = 0 V	1.6	-	3.6	V
T <sub>amb</sub>	ambient temperature	HVQFN32	-25	-	+85	°C

[1] Supply voltages below 3 V reduce the performance (the achievable operating distance).

[2] V<sub>DDA</sub>, V<sub>DDD</sub> and V<sub>DD(TVDD)</sub> must always be the same voltage.

[3] V<sub>DD(PVDD)</sub> must always be the same or lower voltage than V<sub>DDD</sub>.

### 13. Thermal characteristics

Table 152. Thermal characteristics

Symbol	Parameter	Conditions	Package	Typ	Unit
$R_{th(j-a)}$	thermal resistance from junction to ambient	in still air with exposed pin soldered on a 4 layer JEDEC PCB	HVQFN32	40	K/W

### 14. Characteristics

Table 153. Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
<b>Input characteristics</b>						
Pins EA, I2C and NRSTPD						
$I_{LI}$	input leakage current		-1	-	+1	$\mu A$
$V_{IH}$	HIGH-level input voltage		$0.7V_{DD(PVDD)}$	-	-	V
$V_{IL}$	LOW-level input voltage		-	-	$0.3V_{DD(PVDD)}$	V
Pin MFIN						
$I_{LI}$	input leakage current		-1	-	+1	$\mu A$
$V_{IH}$	HIGH-level input voltage		$0.7V_{DD(SVDD)}$	-	-	V
$V_{IL}$	LOW-level input voltage		-	-	$0.3V_{DD(SVDD)}$	V
Pin SDA						
$I_{LI}$	input leakage current		-1	-	+1	$\mu A$
$V_{IH}$	HIGH-level input voltage		$0.7V_{DD(PVDD)}$	-	-	V
$V_{IL}$	LOW-level input voltage		-	-	$0.3V_{DD(PVDD)}$	V
Pin RX <sup>[1]</sup>						
$V_i$	input voltage		-1	-	$V_{DDA} + 1$	V
$C_i$	input capacitance	$V_{DDA} = 3 V$ ; receiver active; $V_{RX(p-p)} = 1 V$ ; 1.5 V (DC) offset	-	10	-	pF
$R_i$	input resistance	$V_{DDA} = 3 V$ ; receiver active; $V_{RX(p-p)} = 1 V$ ; 1.5 V (DC) offset	-	350	-	$\Omega$
<i>Input voltage range; see Figure 24</i>						
$V_{i(p-p)(min)}$	minimum peak-to-peak input voltage	Manchester encoded; $V_{DDA} = 3 V$	-	100	-	mV
$V_{i(p-p)(max)}$	maximum peak-to-peak input voltage	Manchester encoded; $V_{DDA} = 3 V$	-	4	-	V
<i>Input sensitivity; see Figure 24</i>						
$V_{mod}$	modulation voltage	minimum Manchester encoded; $V_{DDA} = 3 V$ ; $RxGain[2:0] = 111b$ (48 dB)	-	5	-	mV
Pin OSCIN						
$I_{LI}$	input leakage current		-1	-	+1	$\mu A$
$V_{IH}$	HIGH-level input voltage		$0.7V_{DDA}$	-	-	V
$V_{IL}$	LOW-level input voltage		-	-	$0.3V_{DDA}$	V



Table 153. Characteristics ...continued

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C <sub>i</sub>	input capacitance	V <sub>DDA</sub> = 2.8 V; DC = 0.65 V; AC = 1 V (p-p)	-	2	-	pF
<b>Input/output characteristics</b>						
pins D1, D2, D3, D4, D5, D6 and D7						
I <sub>LI</sub>	input leakage current		-1	-	+1	μA
V <sub>IH</sub>	HIGH-level input voltage		0.7V <sub>DD(PVDD)</sub>	-	-	V
V <sub>IL</sub>	LOW-level input voltage		-	-	0.3V <sub>DD(PVDD)</sub>	V
V <sub>OH</sub>	HIGH-level output voltage	V <sub>DD(PVDD)</sub> = 3 V; I <sub>O</sub> = 4 mA	V <sub>DD(PVDD)</sub> - 0.4	-	V <sub>DD(PVDD)</sub>	V
V <sub>OL</sub>	LOW-level output voltage	V <sub>DD(PVDD)</sub> = 3 V; I <sub>O</sub> = 4 mA	V <sub>SS(PVSS)</sub>	-	V <sub>SS(PVSS)</sub> + 0.4	V
I <sub>OH</sub>	HIGH-level output current	V <sub>DD(PVDD)</sub> = 3 V	-	-	4	mA
I <sub>OL</sub>	LOW-level output current	V <sub>DD(PVDD)</sub> = 3 V	-	-	4	mA
<b>Output characteristics</b>						
Pin MFOUT						
V <sub>OH</sub>	HIGH-level output voltage	V <sub>DD(SVDD)</sub> = 3 V; I <sub>O</sub> = 4 mA	V <sub>DD(SVDD)</sub> - 0.4	-	V <sub>DD(SVDD)</sub>	V
V <sub>OL</sub>	LOW-level output voltage	V <sub>DD(SVDD)</sub> = 3 V; I <sub>O</sub> = 4 mA	V <sub>SS(PVSS)</sub>	-	V <sub>SS(PVSS)</sub> + 0.4	V
I <sub>OL</sub>	LOW-level output current	V <sub>DD(SVDD)</sub> = 3 V	-	-	4	mA
I <sub>OH</sub>	HIGH-level output current	V <sub>DD(SVDD)</sub> = 3 V	-	-	4	mA
Pin IRQ						
V <sub>OH</sub>	HIGH-level output voltage	V <sub>DD(PVDD)</sub> = 3 V; I <sub>O</sub> = 4 mA	V <sub>DD(PVDD)</sub> - 0.4	-	V <sub>DD(PVDD)</sub>	V
V <sub>OL</sub>	LOW-level output voltage	V <sub>DD(PVDD)</sub> = 3 V; I <sub>O</sub> = 4 mA	V <sub>SS(PVSS)</sub>	-	V <sub>SS(PVSS)</sub> + 0.4	V
I <sub>OL</sub>	LOW-level output current	V <sub>DD(PVDD)</sub> = 3 V	-	-	4	mA
I <sub>OH</sub>	HIGH-level output current	V <sub>DD(PVDD)</sub> = 3 V	-	-	4	mA
Pins AUX1 and AUX2						
V <sub>OH</sub>	HIGH-level output voltage	V <sub>DDD</sub> = 3 V; I <sub>O</sub> = 4 mA	V <sub>DDD</sub> - 0.4	-	V <sub>DDD</sub>	V
V <sub>OL</sub>	LOW-level output voltage	V <sub>DDD</sub> = 3 V; I <sub>O</sub> = 4 mA	V <sub>SS(PVSS)</sub>	-	V <sub>SS(PVSS)</sub> + 0.4	V
I <sub>OL</sub>	LOW-level output current	V <sub>DDD</sub> = 3 V	-	-	4	mA
I <sub>OH</sub>	HIGH-level output current	V <sub>DDD</sub> = 3 V	-	-	4	mA
Pins TX1 and TX2						

Table 153. Characteristics ...continued

Symbol	Parameter	Conditions	Min	Typ	Max	Unit	
V <sub>OH</sub>	HIGH-level output voltage	V <sub>DD(TVDD)</sub> = 3 V; I <sub>DD(TVDD)</sub> = 32 mA; CWGsP[5:0] = 3Fh	V <sub>DD(TVDD)</sub> - 0.15	-	-	V	
		V <sub>DD(TVDD)</sub> = 3 V; I <sub>DD(TVDD)</sub> = 80 mA; CWGsP[5:0] = 3Fh	V <sub>DD(TVDD)</sub> - 0.4	-	-	V	
		V <sub>DD(TVDD)</sub> = 2.5 V; I <sub>DD(TVDD)</sub> = 32 mA; CWGsP[5:0] = 3Fh	V <sub>DD(TVDD)</sub> - 0.24	-	-	V	
		V <sub>DD(TVDD)</sub> = 2.5 V; I <sub>DD(TVDD)</sub> = 80 mA; CWGsP[5:0] = 3Fh	V <sub>DD(TVDD)</sub> - 0.64	-	-	V	
V <sub>OL</sub>	LOW-level output voltage	V <sub>DD(TVDD)</sub> = 3 V; I <sub>DD(TVDD)</sub> = 32 mA; CWGsP[5:0] = 0Fh	-	-	0.15	V	
		V <sub>DD(TVDD)</sub> = 3 V; I <sub>DD(TVDD)</sub> = 80 mA; CWGsP[5:0] = 0Fh	-	-	0.4	V	
		V <sub>DD(TVDD)</sub> = 2.5 V; I <sub>DD(TVDD)</sub> = 32 mA; CWGsP[5:0] = 0Fh	-	-	0.24	V	
		V <sub>DD(TVDD)</sub> = 2.5 V; I <sub>DD(TVDD)</sub> = 80 mA; CWGsP[5:0] = 0Fh	-	-	0.64	V	
<b>Current consumption</b>							
I <sub>pd</sub>	power-down current	V <sub>DDA</sub> = V <sub>DDD</sub> = V <sub>DD(TVDD)</sub> = V <sub>DD(PVDD)</sub> = 3 V					
		hard power-down; pin NRSTPD set LOW	[2]	-	5	μA	
		soft power-down; RF level detector on	[2]	-	10	μA	
I <sub>DDD</sub>	digital supply current	pin DVDD; V <sub>DDD</sub> = 3 V	-	6.5	9	mA	
I <sub>DDA</sub>	analog supply current	pin AVDD; V <sub>DDA</sub> = 3 V; CommandReg register's bit RcvOff = 0	-	7	10	mA	
		pin AVDD; receiver switched off; V <sub>DDA</sub> = 3 V; CommandReg register's bit RcvOff = 1	-	3	5	mA	
I <sub>DD(PVDD)</sub>	PVDD supply current	pin PVDD	[3]	-	40	mA	
I <sub>DD(TVDD)</sub>	TVDD supply current	pin TVDD; continuous wave	[4][5][6]	-	60	100	mA
I <sub>DD(SVDD)</sub>	SVDD supply current	pin SVDD	[7]	-	4	mA	
<b>Clock frequency</b>							
f <sub>clk</sub>	clock frequency		-	27.12	-	MHz	
δ <sub>clk</sub>	clock duty cycle		40	50	60	%	
t <sub>jit</sub>	jitter time	RMS	-	-	10	ps	
<b>Crystal oscillator</b>							

Table 153. Characteristics ...continued

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>OH</sub>	HIGH-level output voltage	pin OSCOUT	-	1.1	-	V
V <sub>OL</sub>	LOW-level output voltage	pin OSCOUT	-	0.2	-	V
C <sub>i</sub>	input capacitance	pin OSCOUT	-	2	-	pF
		pin OSCIN	-	2	-	pF
Typical input requirements						
f <sub>xtal</sub>	crystal frequency		-	27.12	-	MHz
ESR	equivalent series resistance		-	-	100	Ω
C <sub>L</sub>	load capacitance		-	10	-	pF
P <sub>xtal</sub>	crystal power dissipation		-	50	100	mW

- [1] The voltage on pin RX is clamped by internal diodes to pins AVSS and AVDD.
- [2] I<sub>pd</sub> is the total current for all supplies.
- [3] I<sub>DD(PVDD)</sub> depends on the overall load at the digital pins.
- [4] I<sub>DD(TVDD)</sub> depends on V<sub>DD(TVDD)</sub> and the external circuit connected to pins TX1 and TX2.
- [5] During typical circuit operation, the overall current is below 100 mA.
- [6] Typical value using a complementary driver configuration and an antenna matched to 40 Ω between pins TX1 and TX2 at 13.56 MHz.
- [7] I<sub>DD(SVDD)</sub> depends on the load at pin MFOUT.

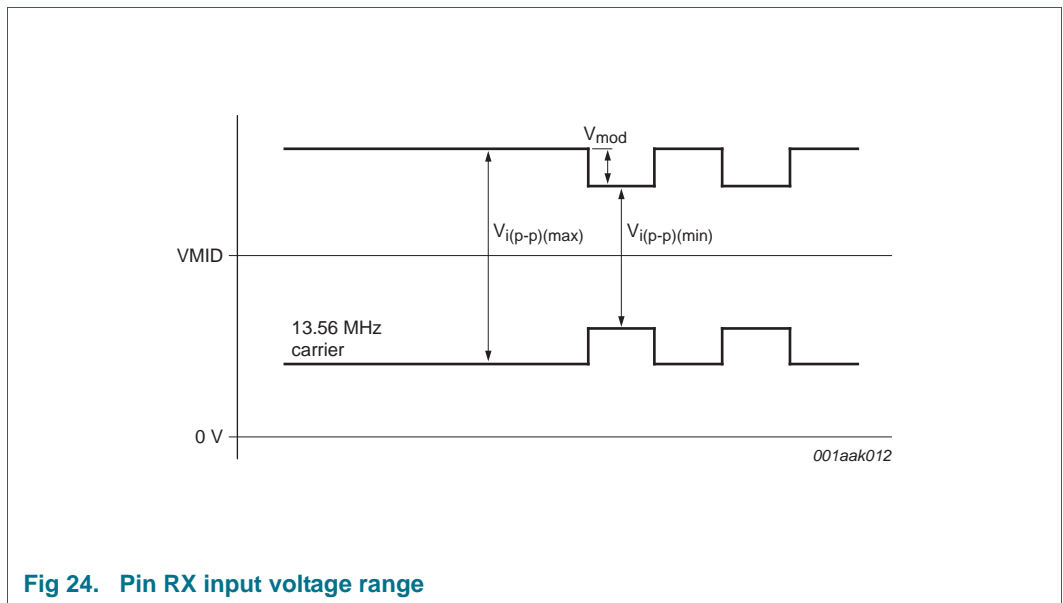


Fig 24. Pin RX input voltage range

## 14.1 Timing characteristics

Table 154. SPI timing characteristics

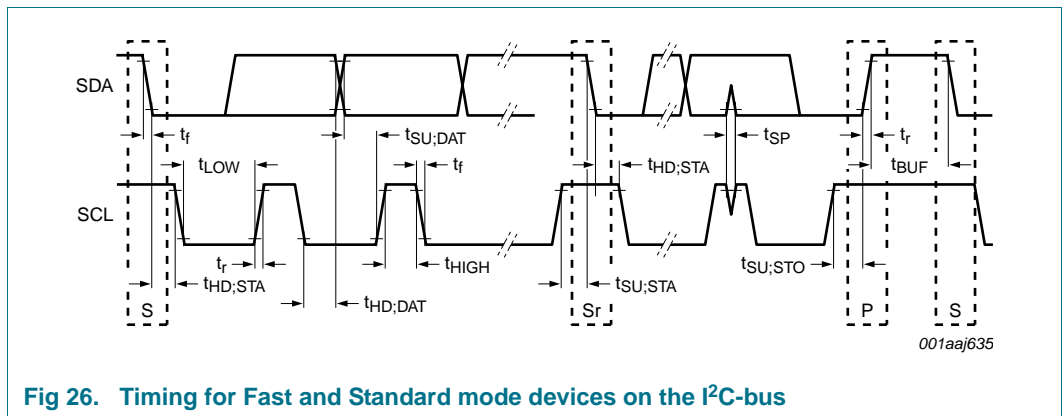
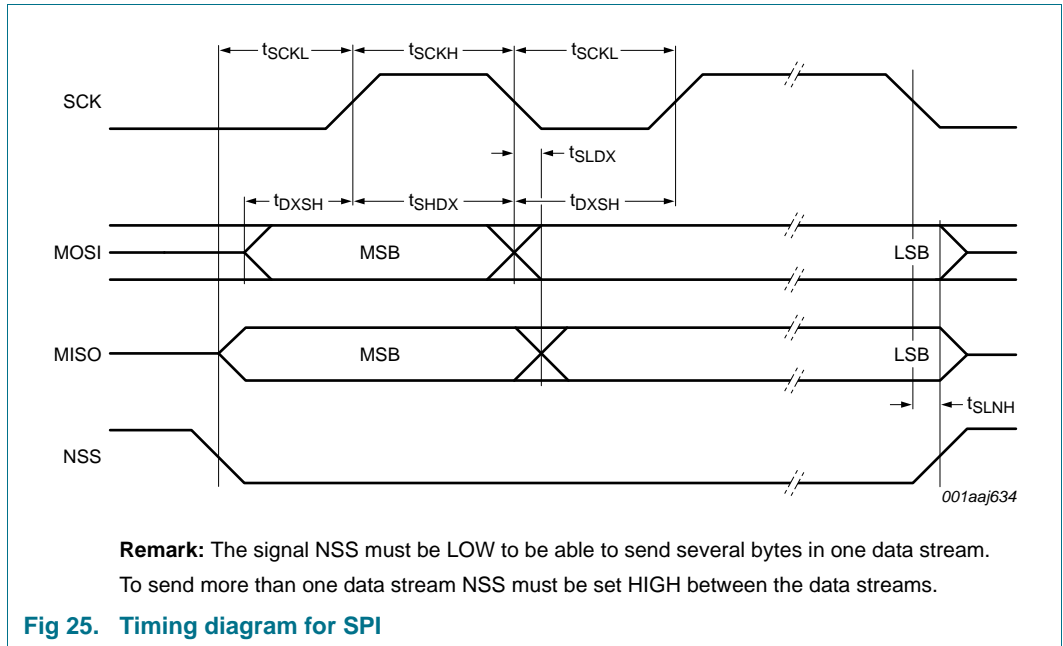
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t <sub>WL</sub>	pulse width LOW	line SCK	50	-	-	ns
t <sub>WH</sub>	pulse width HIGH	line SCK	50	-	-	ns
t <sub>h(SCKH-D)</sub>	SCK HIGH to data input hold time	SCK to changing MOSI	25	-	-	ns

Table 154. SPI timing characteristics ...continued

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{su(D-SCKH)}$	data input to SCK HIGH set-up time	changing MOSI to SCK	25	-	-	ns
$t_{h(SCKL-Q)}$	SCK LOW to data output hold time	SCK to changing MISO	-	-	25	ns
$t_{(SCKL-NSSH)}$	SCK LOW to NSS HIGH time		0	-	-	ns
$t_{NHNL}$	NSS high before communication		50	-	-	ns

Table 155. I<sup>2</sup>C-bus timing in Fast mode

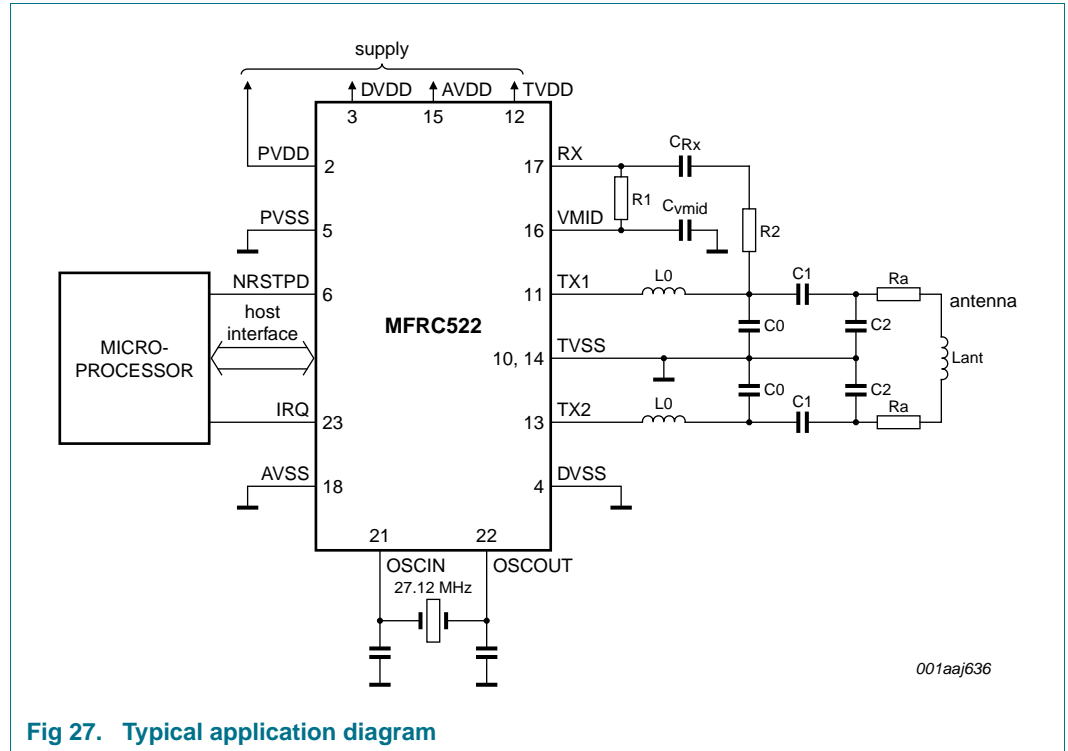
Symbol	Parameter	Conditions	Fast mode		High-speed mode		Unit
			Min	Max	Min	Max	
$f_{SCL}$	SCL clock frequency		0	400	0	3400	kHz
$t_{HD;STA}$	hold time (repeated) START condition	after this period, the first clock pulse is generated	600	-	160	-	ns
$t_{SU;STA}$	set-up time for a repeated START condition		600	-	160	-	ns
$t_{SU;STO}$	set-up time for STOP condition		600	-	160	-	ns
$t_{LOW}$	LOW period of the SCL clock		1300	-	160	-	ns
$t_{HIGH}$	HIGH period of the SCL clock		600	-	60	-	ns
$t_{HD;DAT}$	data hold time		0	900	0	70	ns
$t_{SU;DAT}$	data set-up time		100	-	10	-	ns
$t_r$	rise time	SCL signal	20	300	10	40	ns
$t_f$	fall time	SCL signal	20	300	10	40	ns
$t_r$	rise time	SDA and SCL signals	20	300	10	80	ns
$t_f$	fall time	SDA and SCL signals	20	300	10	80	ns
$t_{BUF}$	bus free time between a STOP and START condition		1.3	-	1.3	-	$\mu$ s



**15. Application information**

A typical application diagram using a complementary antenna connection to the MFRC522 is shown in [Figure 27](#).

The antenna tuning and RF part matching is described in the application note [Ref. 1](#) and [Ref. 2](#).



**Fig 27. Typical application diagram**

## 16. Test information

### 16.1 Test signals

#### 16.1.1 Self test

The MFRC522 has the capability to perform a digital self test. The self test is started by using the following procedure:

1. Perform a soft reset.
2. Clear the internal buffer by writing 25 bytes of 00h and implement the Config command.
3. Enable the self test by writing 09h to the AutoTestReg register.
4. Write 00h to the FIFO buffer.
5. Start the self test with the CalcCRC command.
6. The self test is initiated.
7. When the self test has completed, the FIFO buffer contains the following 64 bytes:

FIFO buffer byte values for MFRC522 version 1.0:

00h, C6h, 37h, D5h, 32h, B7h, 57h, 5Ch,  
C2h, D8h, 7Ch, 4Dh, D9h, 70h, C7h, 73h,  
10h, E6h, D2h, AAh, 5Eh, A1h, 3Eh, 5Ah,  
14h, AFh, 30h, 61h, C9h, 70h, DBh, 2Eh,  
64h, 22h, 72h, B5h, BDh, 65h, F4h, ECh,  
22h, BCh, D3h, 72h, 35h, CDh, AAh, 41h,  
1Fh, A7h, F3h, 53h, 14h, DEh, 7Eh, 02h,  
D9h, 0Fh, B5h, 5Eh, 25h, 1Dh, 29h, 79h

FIFO buffer byte values for MFRC522 version 2.0:

00h, EBh, 66h, BAh, 57h, BFh, 23h, 95h,  
D0h, E3h, 0Dh, 3Dh, 27h, 89h, 5Ch, DEh,  
9Dh, 3Bh, A7h, 00h, 21h, 5Bh, 89h, 82h,  
51h, 3Ah, EBh, 02h, 0Ch, A5h, 00h, 49h,  
7Ch, 84h, 4Dh, B3h, CCh, D2h, 1Bh, 81h,  
5Dh, 48h, 76h, D5h, 71h, 061h, 21h, A9h,  
86h, 96h, 83h, 38h, CFh, 9Dh, 5Bh, 6Dh,  
DCh, 15h, BAh, 3Eh, 7Dh, 95h, 03Bh, 2Fh

#### 16.1.2 Test bus

The test bus is used for production tests. The following configuration can be used to improve the design of a system using the MFRC522. The test bus allows internal signals to be routed to the digital interface. The test bus comprises two sets of test signals which are selected using their subaddress specified in the TestSel2Reg register's TestBusSel[4:0] bits. The test signals and their related digital output pins are described in [Table 156](#) and [Table 157](#).

**Table 156. Test bus signals: TestBusSel[4:0] = 07h**

Pins	Internal signal name	Description
D6	s_data	received data stream
D5	s_coll	bit-collision detected (106 kBd only)
D4	s_valid	s_data and s_coll signals are valid
D3	s_over	receiver has detected a stop condition
D2	RCV_reset	receiver is reset
D1	-	reserved

**Table 157. Test bus signals: TestBusSel[4:0] = 0Dh**

Pins	Internal test signal name	Description
D6	clkstable	oscillator output signal
D5	clk27/8	oscillator output signal divided by 8
D4 to D3	-	reserved
D2	clk27	oscillator output signal
D1	-	reserved

### 16.1.3 Test signals on pins AUX1 or AUX2

The MFRC522 allows the user to select internal signals for measurement on pins AUX1 or AUX2. These measurements can be helpful during the design-in phase to optimize the design or used for test purposes.

[Table 158](#) shows the signals that can be switched to pin AUX1 or AUX2 by setting AnalogSelAux1[3:0] or AnalogSelAux2[3:0] in the AnalogTestReg register.

**Remark:** The DAC has a current output, therefore it is recommended that a 1 k $\Omega$  pull-down resistor is connected to pin AUX1 or AUX2.

**Table 158. Test signal descriptions**

AnalogSelAux1[3:0] or AnalogSelAux2[3:0] value	Signal on pin AUX1 or pin AUX2
0000	3-state
0001	DAC: register TestDAC1 or TestDAC2
0010	DAC: test signal Corr1
0011	reserved
0100	DAC: test signal MinLevel
0101	DAC: test signal ADC_I
0110	DAC: test signal ADC_Q
0111 to 1001	reserved
1010	HIGH
1011	LOW
1100	TxActive



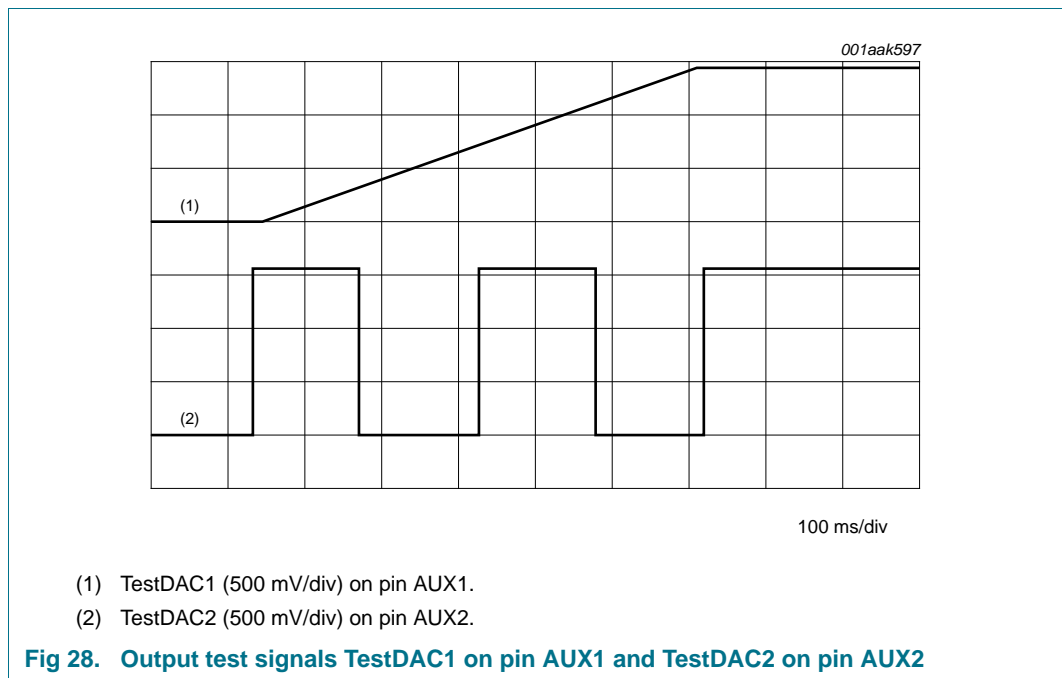
Table 158. Test signal descriptions ...continued

AnalogSelAux1[3:0] or AnalogSelAux2[3:0] value	Signal on pin AUX1 or pin AUX2
1101	RxActive
1110	subcarrier detected
1111	TstBusBit

16.1.3.1 Example: Output test signals TestDAC1 and TestDAC2

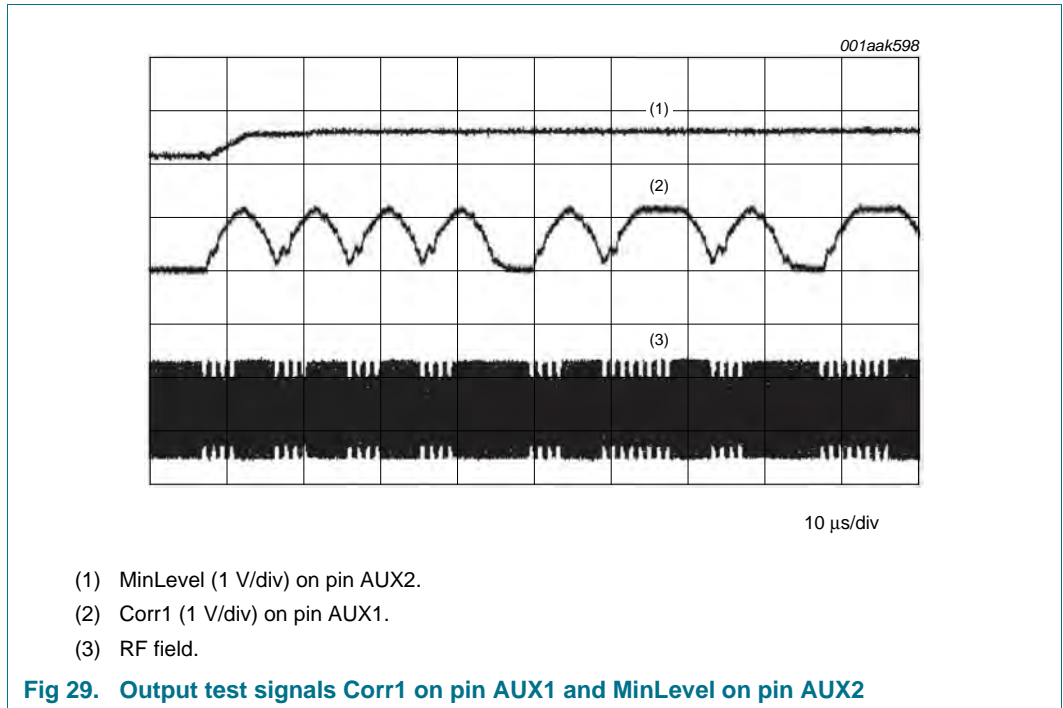
The AnalogTestReg register is set to 11h. The output on pin AUX1 has the test signal TestDAC1 and the output on pin AUX2 has the test signal TestDAC2. The signal values of TestDAC1 and TestDAC2 are controlled by the TestDAC1Reg and TestDAC2Reg registers.

Figure 28 shows test signal TestDAC1 on pin AUX1 and TestDAC2 on pin AUX2 when the TestDAC1Reg register is programmed with a slope defined by values 00h to 3Fh and the TestDAC2Reg register is programmed with a rectangular signal defined by values 00h and 3Fh.



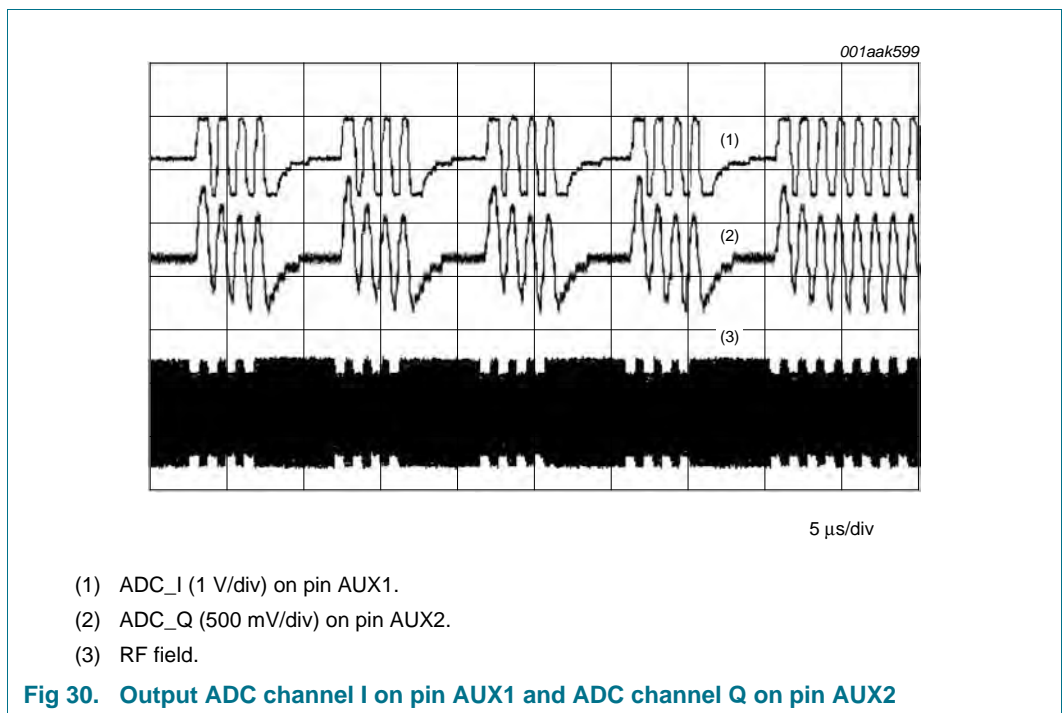
16.1.3.2 Example: Output test signals Corr1 and MinLevel

Figure 29 shows test signals Corr1 and MinLevel on pins AUX1 and AUX2, respectively. The AnalogTestReg register is set to 24h.



**16.1.3.3 Example: Output test signals ADC channel I and ADC channel Q**

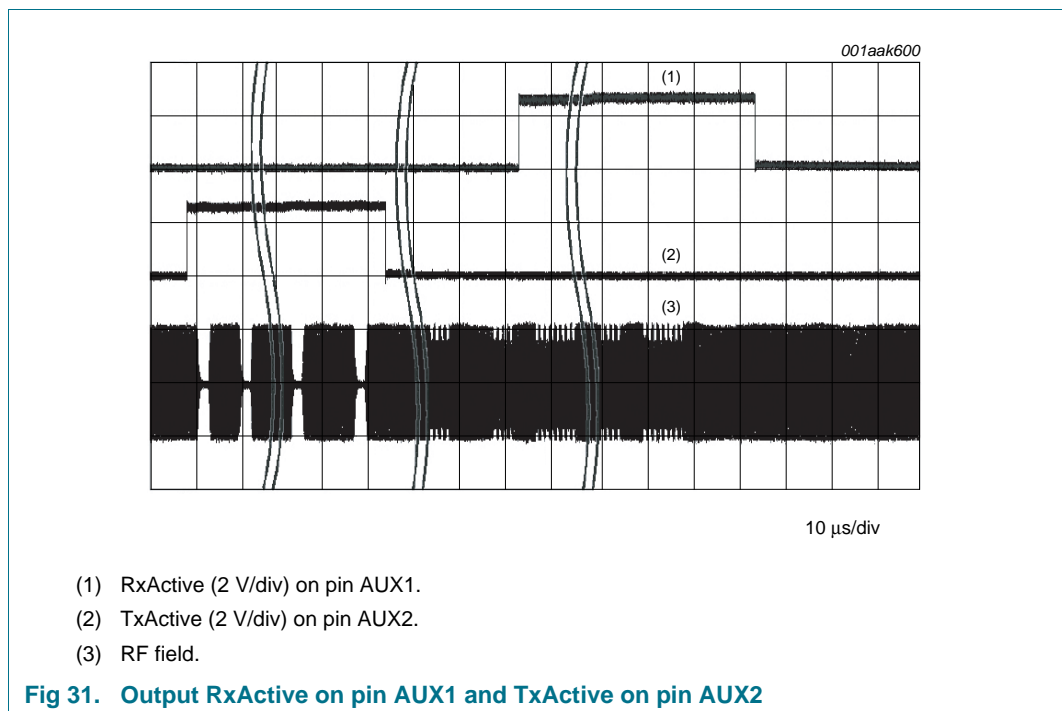
Figure 30 shows the channel behavior test signals ADC\_I and ADC\_Q on pins AUX1 and AUX2, respectively. The AnalogTestReg register is set to 56h.



16.1.3.4 Example: Output test signals RxActive and TxActive

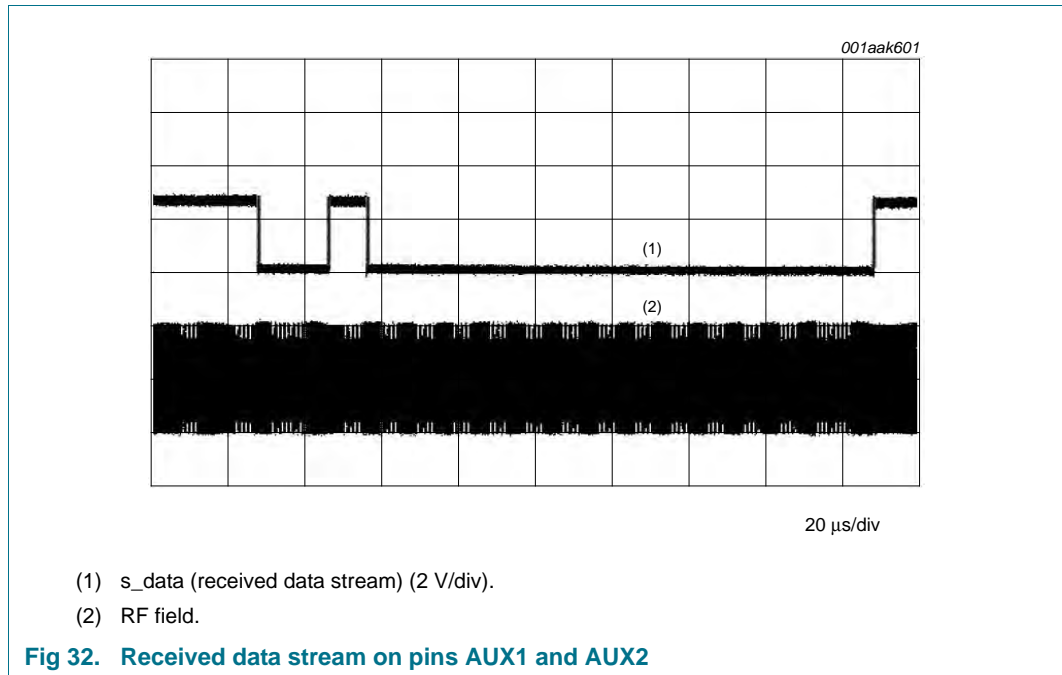
Figure 31 shows the RxActive and TxActive test signals relating to RF communication. The AnalogTestReg register is set to CDh.

- At 106 kBd, RxActive is HIGH during data bits, parity and CRC reception. Start bits are not included
- At 106 kBd, TxActive is HIGH during start bits, data bits, parity and CRC transmission
- At 212 kBd, 424 kBd and 848 kBd, RxActive is HIGH during data bits and CRC reception. Start bits are not included
- At 212 kBd, 424 kBd and 848 kBd, TxActive is HIGH during data bits and CRC transmission



16.1.3.5 Example: Output test signal RX data stream

Figure 32 shows the data stream that is currently being received. The TestSel2Reg register's TestBusSel[4:0] bits are set to 07h to enable test bus signals on pins D1 to D6; see Section 16.1.2 on page 82. The TestSel1Reg register's TstBusBitSel[2:0] bits are set to 06h (pin D6 = s\_data) and AnalogTestReg register is set to FFh (TstBusBit) which outputs the received data stream on pins AUX1 and AUX2.



16.1.3.6 PRBS

The pseudo-random binary sequences PRBS9 and PRBS15 are based on ITU-T0150 and are defined with the TestSel2Reg register. Transmission of either data stream is started by the Transmit command. The preamble/sync byte/start bit/parity bit are automatically generated depending on the mode selected.

**Remark:** All relevant registers for transmitting data must be configured in accordance with ITU-T0150 before selecting PRBS transmission.

17. Package outline

HVQFN32: plastic thermal enhanced very thin quad flat package; no leads; 32 terminals; body 5 x 5 x 0.85 mm

SOT617-1

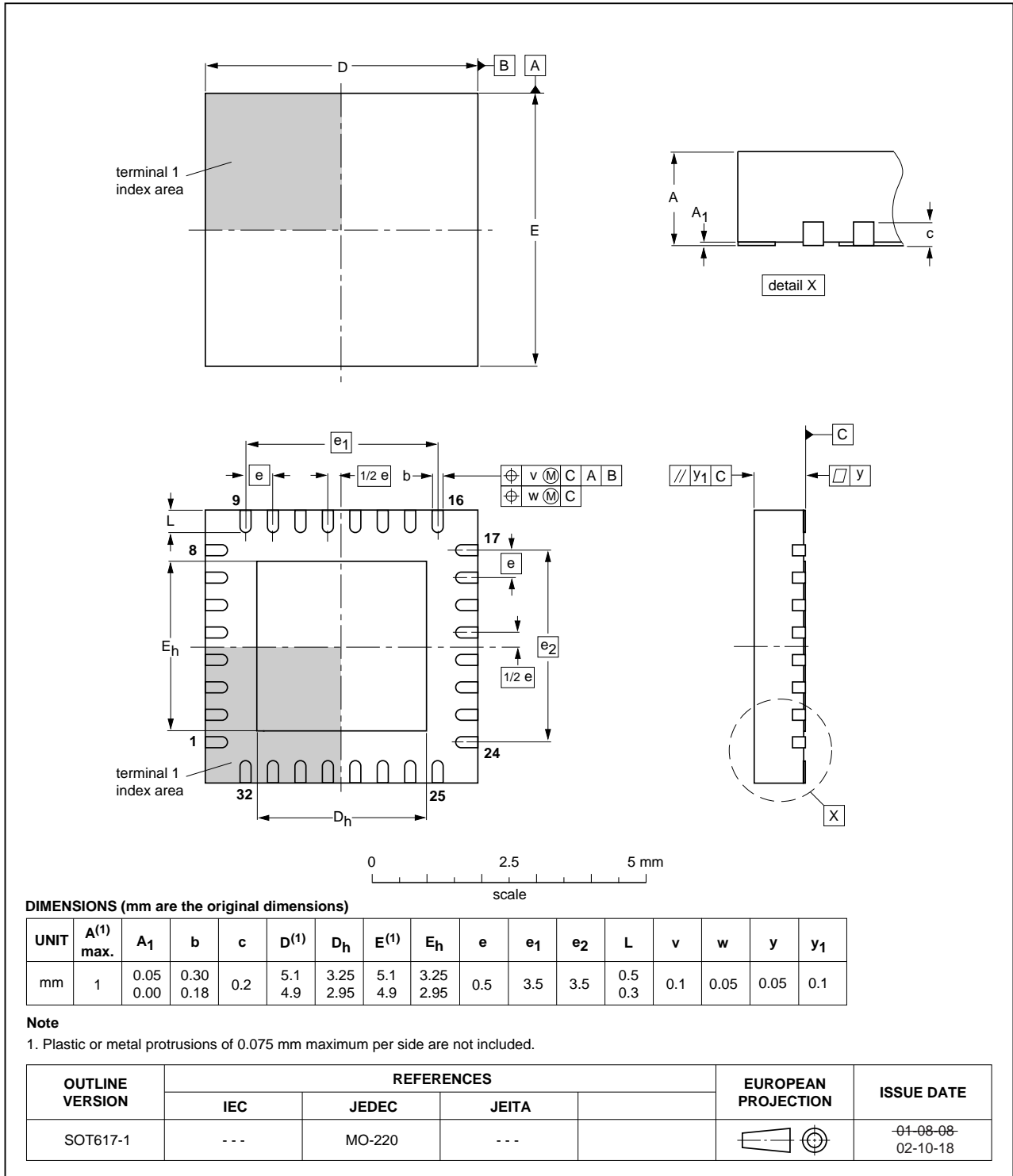


Fig 33. Package outline SOT617-1 (HVQFN32)

Detailed package information can be found at:  
<http://www.nxp.com/package/SOT617-1.html>.

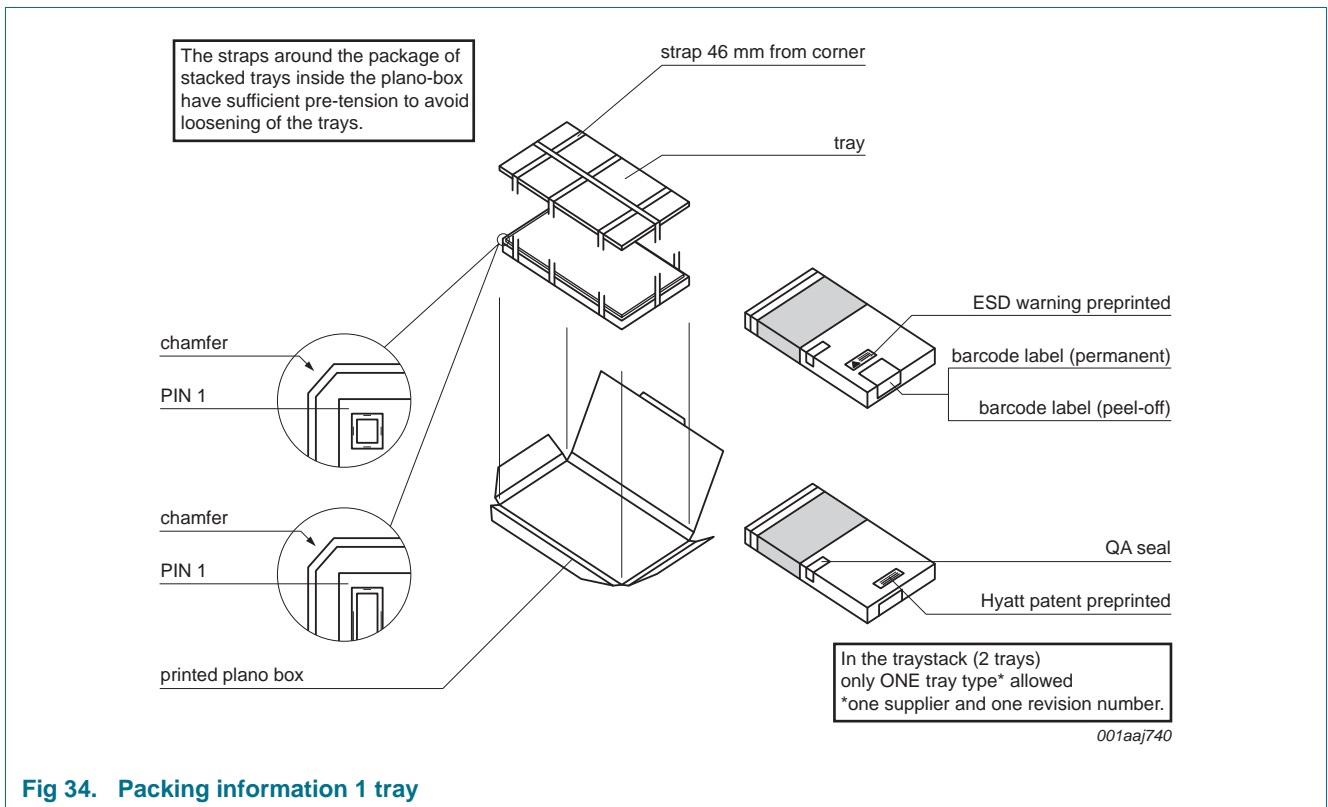
## 18. Handling information

Moisture Sensitivity Level (MSL) evaluation has been performed according to *SNW-FQ-225B rev.04/07/07 (JEDEC J-STD-020C)*. MSL for this package is level 1 which means 260 °C convection reflow temperature.

Dry pack is not required.

Unlimited out-of-pack floor life at maximum ambient 30 °C/85 % RH.

## 19. Packing information



## 20. Abbreviations

Table 159. Abbreviations

Acronym	Description
ADC	Analog-to-Digital Converter
BPSK	Binary Phase Shift Keying
CRC	Cyclic Redundancy Check
CW	Continuous Wave
DAC	Digital-to-Analog Converter
HBM	Human Body Model
I <sup>2</sup> C	Inter-integrated Circuit
LSB	Least Significant Bit
MISO	Master In Slave Out
MM	Machine Model
MOSI	Master Out Slave In
MSB	Most Significant Bit
NRZ	Not Return to Zero
NSS	Not Slave Select
PLL	Phase-Locked Loop
PRBS	Pseudo-Random Bit Sequence
RX	Receiver
SOF	Start Of Frame
SPI	Serial Peripheral Interface
TX	Transmitter
UART	Universal Asynchronous Receiver Transmitter

## 21. References

- [1] **Application note** — *MFRC52x Reader IC Family Directly Matched Antenna Design*
- [2] **Application note** — *MIFARE (ISO/IEC 14443 A) 13.56 MHz RFID Proximity Antennas*

## 22. Revision history

Table 160. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
MFRC522 v.3.9	20160427	Product data sheet	-	MFRC522 v.3.8
Modifications:	<ul style="list-style-type: none"> <li>• <a href="#">Section 1 "Introduction"</a> and <a href="#">Section 2 "General description"</a>: updated and NTAG functionality added</li> <li>• Descriptive title updated</li> </ul>			
MFRC522 v.3.8	20140917	Product data sheet	-	MFRC522 v.3.7
Modifications:	<ul style="list-style-type: none"> <li>• <a href="#">Table 150 "Limiting values"</a>: updated</li> </ul>			
MFRC522 v.3.7	20140326	Product data sheet	-	MFRC522 v.3.6
Modifications:	<ul style="list-style-type: none"> <li>• Change of descriptive title</li> <li>• Section 23.4 "Licenses" removed</li> </ul>			
MFRC522 v.3.6	20111214	Product data sheet	-	MFRC522_35
Modifications:	<ul style="list-style-type: none"> <li>• <a href="#">Section 1.1 "Differences between version 1.0 and 2.0" on page 1</a>: added</li> <li>• <a href="#">Table 2 "Ordering information" on page 3</a>: updated</li> <li>• <a href="#">Section 9.3.2.10 "DemodReg register" on page 53</a>: register updated and add reference to Timer unit</li> <li>• <a href="#">Section 8.5 "Timer unit" on page 31</a>: Pre Scaler Information for version 2.0 added</li> <li>• <a href="#">Section 9.3.4.8 "VersionReg register" on page 66</a>: version information structured in chip information and version information updated, including version 1.0 and 2.0</li> <li>• <a href="#">Section 16.1 "Test signals" on page 82</a>: selftest result including values for version 1.0 and 2.0</li> </ul>			
MFRC522_35	20100621	Product data sheet		MFRC522_34
Modifications:	<ul style="list-style-type: none"> <li>• <a href="#">Section 9.3.2.10 "DemodReg register" on page 53</a>: register updated</li> <li>• <a href="#">Section 9.3.3.10 "TModeReg and TPrescalerReg registers" on page 60</a>: register updated</li> <li>• <a href="#">Section 8.5 "Timer unit" on page 31</a>: timer calculation updated</li> <li>• <a href="#">Section 9.3.4.8 "VersionReg register" on page 66</a>: version B2h updated</li> <li>• <a href="#">Section 16.1 "Test signals" on page 82</a>: selftest result updated</li> </ul>			
MFRC522_34	20100305	Product data sheet		MFRC522_33
Modifications:	<ul style="list-style-type: none"> <li>• <a href="#">Section 8.5 "Timer unit"</a>: information added</li> <li>• <a href="#">Table 106 "TModeReg register bit descriptions"</a>: bit 7 updated</li> <li>• <a href="#">Table 154 "SPI timing characteristics"</a>: row added</li> </ul>			
MFRC522_33	20091026	Product data sheet	-	112132



## 23. Legal information

### 23.1 Data sheet status

Document status <sup>[1][2]</sup>	Product status <sup>[3]</sup>	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL <http://www.nxp.com>.

### 23.2 Definitions

**Draft** — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

**Short data sheet** — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

**Product specification** — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

### 23.3 Disclaimers

**Limited warranty and liability** — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

**Right to make changes** — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Suitability for use** — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

**Applications** — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Limiting values** — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

**Terms and conditions of commercial sale** — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

**No offer to sell or license** — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Quick reference data** — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

**Non-automotive qualified products** — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond

NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

**Translations** — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

## 23.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

**I<sup>2</sup>C-bus** — logo is a trademark of NXP B.V.

**MIFARE** — is a trademark of NXP B.V.

## 24. Contact information

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

## 25. Contents

<b>1</b>	<b>Introduction</b> .....	<b>1</b>	8.6	Power reduction modes .....	32
<b>2</b>	<b>General description</b> .....	<b>1</b>	8.6.1	Hard power-down .....	32
2.1	Differences between version 1.0 and 2.0 .....	1	8.6.2	Soft power-down mode .....	32
<b>3</b>	<b>Features and benefits</b> .....	<b>2</b>	8.6.3	Transmitter power-down mode .....	32
<b>4</b>	<b>Quick reference data</b> .....	<b>2</b>	8.7	Oscillator circuit .....	32
<b>5</b>	<b>Ordering information</b> .....	<b>3</b>	8.8	Reset and oscillator start-up time .....	33
<b>6</b>	<b>Block diagram</b> .....	<b>4</b>	8.8.1	Reset timing requirements .....	33
<b>7</b>	<b>Pinning information</b> .....	<b>6</b>	8.8.2	Oscillator start-up time .....	33
7.1	Pin description .....	6	<b>9</b>	<b>MFRC522 registers</b> .....	<b>34</b>
<b>8</b>	<b>Functional description</b> .....	<b>8</b>	9.1	Register bit behavior .....	34
8.1	Digital interfaces .....	9	9.2	Register overview .....	35
8.1.1	Automatic microcontroller interface detection ..	9	9.3	Register descriptions .....	37
8.1.2	Serial Peripheral Interface .....	10	9.3.1	Page 0: Command and status .....	37
8.1.2.1	SPI read data .....	10	9.3.1.1	Reserved register 00h .....	37
8.1.2.2	SPI write data .....	11	9.3.1.2	CommandReg register .....	37
8.1.2.3	SPI address byte .....	11	9.3.1.3	ComEnReg register .....	37
8.1.3	UART interface .....	11	9.3.1.4	DivlEnReg register .....	38
8.1.3.1	Connection to a host .....	11	9.3.1.5	ComlRqReg register .....	38
8.1.3.2	Selectable UART transfer speeds .....	12	9.3.1.6	DivlRqReg register .....	39
8.1.3.3	UART framing .....	13	9.3.1.7	ErrorReg register .....	40
8.1.4	I <sup>2</sup> C-bus interface .....	16	9.3.1.8	Status1Reg register .....	41
8.1.4.1	Data validity .....	17	9.3.1.9	Status2Reg register .....	42
8.1.4.2	START and STOP conditions .....	17	9.3.1.10	FIFODataReg register .....	43
8.1.4.3	Byte format .....	17	9.3.1.11	FIFOLevelReg register .....	43
8.1.4.4	Acknowledge .....	18	9.3.1.12	WaterLevelReg register .....	43
8.1.4.5	7-Bit addressing .....	19	9.3.1.13	ControlReg register .....	44
8.1.4.6	Register write access .....	19	9.3.1.14	BitFramingReg register .....	45
8.1.4.7	Register read access .....	20	9.3.1.15	CollReg register .....	45
8.1.4.8	High-speed mode .....	21	9.3.1.16	Reserved register 0Fh .....	46
8.1.4.9	High-speed transfer .....	21	9.3.2	Page 1: Communication .....	46
8.1.4.10	Serial data transfer format in HS mode .....	21	9.3.2.1	Reserved register 10h .....	46
8.1.4.11	Switching between F/S mode and HS mode ..	23	9.3.2.2	ModeReg register .....	47
8.1.4.12	MFRC522 at lower speed modes .....	23	9.3.2.3	TxModeReg register .....	47
8.2	Analog interface and contactless UART .....	24	9.3.2.4	RxModeReg register .....	48
8.2.1	General .....	24	9.3.2.5	TxControlReg register .....	49
8.2.2	TX p-driver .....	24	9.3.2.6	TxASKReg register .....	50
8.2.3	Serial data switch .....	26	9.3.2.7	TxSelReg register .....	50
8.2.4	MFIN and MFOUT interface support .....	26	9.3.2.8	RxSelReg register .....	51
8.2.5	CRC coprocessor .....	28	9.3.2.9	RxThresholdReg register .....	52
8.3	FIFO buffer .....	28	9.3.2.10	DemodReg register .....	52
8.3.1	Accessing the FIFO buffer .....	28	9.3.2.11	Reserved register 1Ah .....	53
8.3.2	Controlling the FIFO buffer .....	28	9.3.2.12	Reserved register 1Bh .....	53
8.3.3	FIFO buffer status information .....	28	9.3.2.13	MfTxReg register .....	53
8.4	Interrupt request system .....	29	9.3.2.14	MfRxReg register .....	54
8.4.1	Interrupt sources overview .....	29	9.3.2.15	Reserved register 1Eh .....	54
8.5	Timer unit .....	30	9.3.2.16	SerialSpeedReg register .....	54
			9.3.3	Page 2: Configuration .....	56
			9.3.3.1	Reserved register 20h .....	56

continued >>

9.3.3.2	CRCResultReg registers	56	16.1.2	Test bus	81
9.3.3.3	Reserved register 23h	57	16.1.3	Test signals on pins AUX1 or AUX2	82
9.3.3.4	ModWidthReg register	57	16.1.3.1	Example: Output test signals TestDAC1 and TestDAC2	83
9.3.3.5	Reserved register 25h	57	16.1.3.2	Example: Output test signals Corr1 and MinLevel	83
9.3.3.6	RFCfgReg register	58	16.1.3.3	Example: Output test signals ADC channel I and ADC channel Q	84
9.3.3.7	GsNReg register	58	16.1.3.4	Example: Output test signals RxActive and TxActive	85
9.3.3.8	CWGsPReg register	59	16.1.3.5	Example: Output test signal RX data stream	86
9.3.3.9	ModGsPReg register	59	16.1.3.6	PRBS	86
9.3.3.10	TModeReg and TPrescalerReg registers	59	<b>17</b>	<b>Package outline</b>	<b>87</b>
9.3.3.11	TReloadReg register	61	<b>18</b>	<b>Handling information</b>	<b>88</b>
9.3.3.12	TCounterValReg register	61	<b>19</b>	<b>Packing information</b>	<b>88</b>
9.3.4	Page 3: Test	62	<b>20</b>	<b>Abbreviations</b>	<b>89</b>
9.3.4.1	Reserved register 30h	62	<b>21</b>	<b>References</b>	<b>89</b>
9.3.4.2	TestSel1Reg register	62	<b>22</b>	<b>Revision history</b>	<b>90</b>
9.3.4.3	TestSel2Reg register	63	<b>23</b>	<b>Legal information</b>	<b>91</b>
9.3.4.4	TestPinEnReg register	63	23.1	Data sheet status	91
9.3.4.5	TestPinValueReg register	64	23.2	Definitions	91
9.3.4.6	TestBusReg register	64	23.3	Disclaimers	91
9.3.4.7	AutoTestReg register	65	23.4	Trademarks	92
9.3.4.8	VersionReg register	65	<b>24</b>	<b>Contact information</b>	<b>92</b>
9.3.4.9	AnalogTestReg register	66	<b>25</b>	<b>Contents</b>	<b>93</b>
9.3.4.10	TestDAC1Reg register	67			
9.3.4.11	TestDAC2Reg register	67			
9.3.4.12	TestADCReg register	67			
9.3.4.13	Reserved register 3Ch	67			
<b>10</b>	<b>MFRC522 command set</b>	<b>69</b>			
10.1	General description	69			
10.2	General behavior	69			
10.3	MFRC522 command overview	69			
10.3.1	MFRC522 command descriptions	70			
10.3.1.1	Idle	70			
10.3.1.2	Mem	70			
10.3.1.3	Generate RandomID	70			
10.3.1.4	CalcCRC	70			
10.3.1.5	Transmit	70			
10.3.1.6	NoCmdChange	70			
10.3.1.7	Receive	71			
10.3.1.8	Transceive	71			
10.3.1.9	MFAuthent	71			
10.3.1.10	SoftReset	72			
<b>11</b>	<b>Limiting values</b>	<b>73</b>			
<b>12</b>	<b>Recommended operating conditions</b>	<b>73</b>			
<b>13</b>	<b>Thermal characteristics</b>	<b>73</b>			
<b>14</b>	<b>Characteristics</b>	<b>74</b>			
14.1	Timing characteristics	77			
<b>15</b>	<b>Application information</b>	<b>80</b>			
<b>16</b>	<b>Test information</b>	<b>81</b>			
16.1	Test signals	81			
16.1.1	Self test	81			

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© NXP Semiconductors N.V. 2016. All rights reserved.

For more information, please visit: <http://www.nxp.com>  
 For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 27 April 2016  
 112139