LAMPIRAN

Surat keputusan (SK) pembimbing

DARMAJAYA

SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK. 0320/DMJ/DFIK/BAAK/V-22
Tentang
Dosen Pembimbing Skripsi
Semester Genap TA.2021/2022
Program Studi S1 Sistem Komputer

REKTOR IIB DARMAJAYA

Memperhatikan :
1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam **Skripsi.**
2. Laporan dan usulan Ketua Program Studi **S1 Sistem Komputer.**

Menimbang :
1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi.**
2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.

Mengingat :
1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
6. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
7. STATUTA IBI Darmajaya
8. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.

Menetapkan

Pertama : Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Sistem Komputer.

Kedua : Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.

Ketiga : Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.

Keempat : Surat Keputusan ini berlaku sejak tanggal ditetap kan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
Pada tanggal : 09 Mei 2022
a.n. Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer

Dr. Sutedi, S.Kom., M.T.I
NIK. 00590203

1. Kepala Program Studi S1 Sistem Komputer
2. Yang bersangkutan
3. Arsip

Lampiran : Surat Keputusan Rektor IIB Darmajaya
Nomor : SK. 0320/DMJ/DFIK/BAAK/V-22
Tanggal : 09 Mei 2022
Perihal : Pembimbing Penulisan Skripsi Semester Genap TA. 2021/2022
Program Studi Strata Satu (S1) Sistem Komputer

Judul Skripsi Dan Dosen Pembimbing Skripsi Semester Genap TA. 2021/2022
Program Studi Strata Satu (S1) Sistem Komputer

No	NAMA	NPM	JUDUL	PEMBIMBING
17	Singgih Remiii Dharma	1911068005P	Implementasi Sistem Monitoring Serta Kontrol Suhu Dan Kelembaban Pada Ruang Green House Tanaman Sawi Berbasis Internet of Things (IoT)	Dodi Yudo Setiawan.S.Si.,M.Ti
18	Rino Cahya Prasetia	1911068011P	Perancangan Sistem Kontrol Dan Monitoring Internet of Things (IoT) Pada Kandang Burung Dara	Dodi Yudo Setiawan.S.Si.,M.Ti
19	M Renansyah Anderha	1811060041	Penerapan (IoT) Internet of Things Green House Untuk Kontrol Dan Monitoring Kadar pH, Suhu dan Kelembaban Tanah Pada Tanaman Melon	Ari Widiantoko, S.Kom.,M.Tech
20	Muhammad Saifudin Mahfudz	1811060042	Sistem Pendeteksi kendaraan Bermotor Berbasis Image Processing Menggunakan Raspberry Pi Pada alat Pantau pajak Parkir Kendaraan Bermotor	Bayu Nugroho, S.Kom.,M.Eng
21	Faqsy Cahya Syahbana	1811060003	Rancang Bangun Smart Farming Untuk Kontrol dan Monitoring Kadar Pupuk NPK Pada Tanaman Terung Berbasis IoT	Iia RosmaliaS.T., M.Kom
22	Galang Tirtoaji Pranata	1811060001	Rancang Bangun Sistem Antrian Berbasis RFID	Bayu Nugroho, S.Kom.,M.Eng
23	Suhendro	1811060040	Perancangan Sistem Monitoring Suhu Udara Dan Penyemprot Pestisida Secara Otomatis Berbasis ESP32	Zaidir Jamal, S.T.,M.Eng
24	Ramadani	1811060019	Sistem Smart Farming pada Tanaman Hidroponik Berbasis IoT di Screen House Balai Pelatihan Pertanian (BPP) Lampung	Dodi Yudo Setiawan.S.Si.,M.Ti
25	Wayan Aditya Pranata	1811060026	Rancang Bangun Alat Pengemasan Otomatis	Novi Herawadi Sudibyo, S.Kom., M.Ti
26	Gwin Cyril Vertido Somera	1811060043	Perancangan Sistem Monitoring Kualitas air (Kekeruhan Dan PH) Pada Tanaman Hidroponik Berbasis NodeMCU ESP32	Abdi Darmawan, S.T.,M.Ti
27	Satiro Dero Mahendra	1811060025	Rancang Bangun Kontrol dan Monitoring Akuarium Berbasis Internet of Things (IoT)	Novi Herawadi Sudibyo, S.Kom., M.Ti
28	Fadli Zaman	1811060017	Implementasi IOT Pada Vertikal Farming Di Green House	Nurfiana, S.Kom.,M.Kom
29	Aditya Pangestu	1811060035	Sistem Pengisian Madu Klengkeng Otomatis Menggunakan Mikrokontroller ESP 32	Lia Rosmalia, S.T.,M.Kom
30	Cindy Imas Pratiwi Sugiono	1811060031	Rancang Bangun Penyiram Taman Otomatis Berbasis Internet Of Things	Melia Gripin S, S.Kom.,M.T
31	Wulandari Eka Saputri	1811060002	Rancang Bangun Sistem Pemupukan Otomatis Untuk Tanaman Padi Sawah	Nurfiana, S.Kom.,M.Kom
32	Lintang Ayu Zahiroh	1811060022	Rancang Bangun Mesin Dan Penyiraman Otomatis Larutan Jeruk Nipis Pada Penetas Telur Berbasis Internet of Things (IoT)	Melia Gripin S, S.Kom.,M.T

Datasheet Node MCU

ESP8266EX

Datasheet

Version 6.0

Espressif Systems

About This Guide

This document introduces the specifications of ESP8266EX.

Release Notes

Date	Version	Release Notes
2015.12	V4.6	Updated Chapter 3.
2016.02	V4.7	Updated Section 3.6 and Section 4.1.
2016.04	V4.8	Updated Chapter 1.
2016.08	V4.9	Updated Chapter 1.
2016.11	V5.0	Added Appendix II "Learning Resources".
2016.11	V5.1	Changed the power consumption during Deep-sleep from 10 μA to 20 μA in Table 5-2.
2016.11	V5.2	Changed the crystal frequency range from "26 MHz to 52 MHz" to "24 MHz to 52 MHz" in Section 3.3.
2016.12	V5.3	Changed the minimum working voltage from 3.0V to 2.5V.
2017.04	V5.4	Changed chip input and output impedance from 50Ω to 39+j6 Ω.
2017.10	V5.5	Updated Chapter 3 regarding the range of clock amplitude to 0.8 ~ 1.5V.
2017.11	V5.6	Updated VDDPST from 1.8V ~ 3.3V to 1.8V ~ 3.6V.
2017.11	V5.7	• Corrected a typo in the description of SDIO_DATA_0 in Table 2-1; • Added the testing conditions for the data in Table 5-2.
2018.02	V5.8	• Updated Wi-Fi protocols in Section 1.1; • Updated description of the integrated Tensilica processor in 3.1.
2018.09	V5.9	• Update document cover; • Added a note for Table 1-1; • Updated Wi-Fi key features in Section 1.1; • Updated description of the Wi-Fi function in 3.5; • Updated pin layout diagram; • Fixed a typo in Table 2-1; • Removed Section AHB and AHB module; • Restructured Section Power Management; • Fixed a typo in Section UART; • Removed description of transmission angle in Section IR Remote Control; • Other optimization (wording).
2018.11	V6.0	• Added an SPI pin in Table 4-2; • Updated the diagram of packing information.

Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation. Please subscribe at *https://www.espressif.com/en/subscribe*.

Certification

Download certificates for Espressif products from *https://www.espressif.com/en/certificates*.

Table of Contents

1. Overview

Espressif's ESP8266EX delivers highly integrated Wi-Fi SoC solution to meet users' continuous demands for efficient power usage, compact design and reliable performance in the Internet of Things industry.

With the complete and self-contained Wi-Fi networking capabilities, ESP8266EX can perform either as a standalone application or as the slave to a host MCU. When ESP8266EX hosts the application, it promptly boots up from the flash. The integrated high-speed cache helps to increase the system performance and optimize the system memory. Also, ESP8266EX can be applied to any microcontroller design as a Wi-Fi adaptor through SPI/SDIO or UART interfaces.

ESP8266EX integrates antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules. The compact design minimizes the PCB size and requires minimal external circuitries.

Besides the Wi-Fi functionalities, ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor and on-chip SRAM. It can be interfaced with external sensors and other devices through the GPIOs. Software Development Kit (SDK) provides sample codes for various applications.

Espressif Systems' Smart Connectivity Platform (ESCP) enables sophisticated features including:

- Fast switch between sleep and wakeup mode for energy-efficient purpose;
- Adaptive radio biasing for low-power operation
- Advance signal processing
- Spur cancellation and RF co-existence mechanisms for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation

1.1. Wi-Fi Key Features

- 802.11 b/g/n support
- 802.11n support (2.4 GHz), up to 72.2 Mbps
- Defragmentation
- 2 x virtual Wi-Fi interface
- Automatic beacon monitoring (hardware TSF)
- Support Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- Antenna diversity

1.2. Specifications

Table 1-1. Specifications

Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n (HT20)
	Frequency Range	2.4G ~ 2.5G (2400M ~ 2483.5M)
	TX Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: –91 dbm (11 Mbps)
		802.11 g: –75 dbm (54 Mbps)
		802.11 n: –72 dbm (MCS7)
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5V ~ 3.6V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	–40°C ~ 125°C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

Note:

The TX power can be configured based on the actual user scenarios.

1.3. Applications

- Home appliances
- Home automation
- Smart plugs and lights
- Industrial wireless control
- Baby monitors
- IP cameras
- Sensor networks
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Wi-Fi position system beacons

2. Pin Definitions

Figure 2-1 shows the pin layout for 32-pin QFN package.

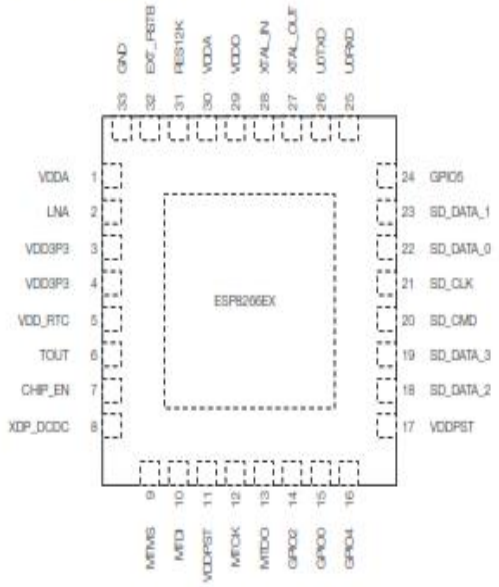


Figure 2-1. Pin Layout (Top View)

Table 2-1 lists the definitions and functions of each pin.

Table 2-1. ESP8266EX Pin Definitions

Pin	Name	Type	Function
1	VDDA	P	Analog Power 2.5V ~ 3.6V
2	LNA	I/O	RF antenna interface Chip output impedance=39+j6 Ω. It is suggested to retain the π-type matching network to match the antenna.
3	VDD3P3	P	Amplifier Power 2.5V ~ 3.6V
4	VDD3P3	P	Amplifier Power 2.5V ~ 3.6V
5	VDD_RTC	P	NC (1.1V)
6	TOUT	I	ADC pin. It can be used to test the power-supply voltage of VDD3P3 (Pin3 and Pin4) and the input power voltage of TOUT (Pin 6). However, these two functions cannot be used simultaneously.

Pin	Name	Type	Function
7	CHIP_EN	I	Chip Enable High: On, chip works properly Low: Off, small current consumed
8	XPD_DCDC	I/O	Deep-sleep wakeup (need to be connected to EXT_RSTB); GPIO16
9	MTMS	I/O	GPIO 14; HSPI_CLK
10	MTDI	I/O	GPIO 12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V ~ 3.6V)
12	MTCK	I/O	GPIO 13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO 15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART TX during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V ~ 3.6V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPI_MISO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART TX during flash programming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 2.5V ~ 3.6V
30	VDDA	P	Analog Power 2.5V ~ 3.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: active)

Note:

1. *GPIO2, GPIO0, and MTDO are used to select booting mode and the SDIO mode;*
2. *U0TXD should not be pulled externally to a low logic level during the powering-up.*

3. Functional Description

The functional diagram of ESP8266EX is shown as in Figure 3-1.

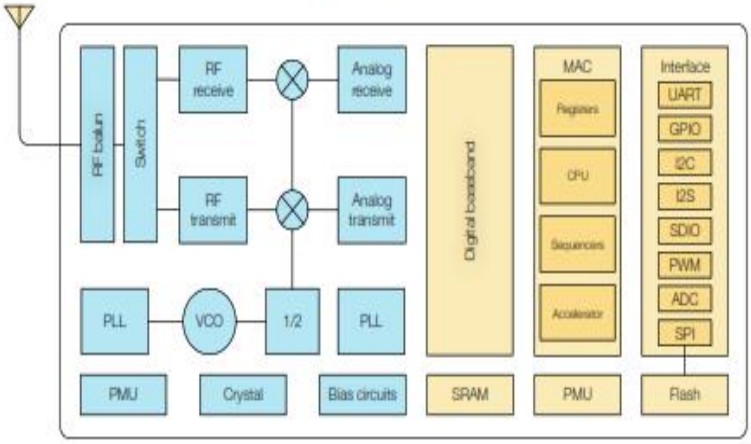


Figure 3-1. Functional Block Diagram

3.1. CPU, Memory, and Flash

3.1.1. CPU

The ESP8266EX integrates a Tensilica L106 32-bit RISC processor, which achieves extra-low power consumption and reaches a maximum clock speed of 160 MHz. The Real-Time Operating System (RTOS) and Wi-Fi stack allow 80% of the processing power to be available for user application programming and development. The CPU includes the interfaces as below:

- Programmable RAM/ROM interfaces (iBus), which can be connected with memory controller, and can also be used to visit flash.
- Data RAM interface (dBus), which can connected with memory controller.
- AHB interface which can be used to visit the register.

3.1.2. Memory

ESP8266EX Wi-Fi SoC integrates memory controller and memory units including SRAM and ROM. MCU can access the memory units through iBus, dBus, and AHB interfaces. All memory units can be accessed upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor.

According to our current version of SDK, SRAM space available to users is assigned as below.

- RAM size < 50 kB, that is, when ESP8266EX is working under the Station mode and connects to the router, the maximum programmable space accessible in Heap + Data section is around 50 kB.
- There is no programmable ROM in the SoC. Therefore, user program must be stored in an external SPI flash.

3.1.3. External Flash

ESP8266EX uses external SPI flash to store user programs, and supports up to 16 MB memory capacity theoretically.

The minimum flash memory of ESP8266EX is shown below:

- OTA disabled: 512 kB at least
- OTA enabled: 1 MB at least

> ⚠ **Notice:**
>
> *SPI mode supported: Standard SPI, Dual SPI and Quad SPI. The correct SPI mode should be selected when flashing bin files to ESP8266. Otherwise, the downloaded firmware/program may not be working properly.*

3.2. Clock

3.2.1. High Frequency Clock

The high frequency clock on ESP8266EX is used to drive both transmit and receive mixers. This clock is generated from internal crystal oscillator and external crystal. The crystal frequency ranges from 24 MHz to 52 MHz.

The internal calibration inside the crystal oscillator ensures that a wide range of crystals can be used, nevertheless the quality of the crystal is still a factor to consider to have reasonable phase noise and good Wi-Fi sensitivity. Refer to Table 3-1 to measure the frequency offset.

Table 3-1. High Frequency Clock Specifications

Parameter	Symbol	Min	Max	Unit
Frequency	FXO	24	52	MHz
Loading capacitance	CL	-	32	pF
Motional capacitance	CM	2	5	pF
Series resistance	RS	0	65	Ω
Frequency tolerance	ΔFXO	−15	15	ppm
Frequency vs temperature (−25°C ~ 75°C)	ΔFXO,Temp	−15	15	ppm

3.2.2. External Clock Requirements

An externally generated clock is available with the frequency ranging from 24 MHz to 52 MHz. The following characteristics are expected to achieve good performance of radio.

Table 3-2. External Clock Reference

Parameter	Symbol	Min	Max	Unit
Clock amplitude	VXO	0.8	1.5	Vpp
External clock accuracy	ΔFXO,EXT	−15	15	ppm
Phase noise @1-kHz offset, 40-MHz clock	-	-	−120	dBc/Hz
Phase noise @10-kHz offset, 40-MHz clock	-	-	−130	dBc/Hz
Phase noise @100-kHz offset, 40-MHz clock	-	-	−138	dBc/Hz

3.3. Radio

ESP8266EX radio consists of the following blocks.

- 2.4 GHz receiver
- 2.4 GHz transmitter
- High speed clock generators and crystal oscillator
- Bias and regulators
- Power management

3.3.1. Channel Frequencies

The RF transceiver supports the following channels according to IEEE802.11b/g/n standards.

Table 3-3. Frequency Channel

Channel No.	Frequency (MHz)	Channel No.	Frequency (MHz)
1	2412	8	2447
2	2417	9	2452
3	2422	10	2457
4	2427	11	2462
5	2432	12	2467
6	2437	13	2472
7	2442	14	2484

3.3.2. 2.4 GHz Receiver

The 2.4 GHz receiver down-converts the RF signals to quadrature baseband signals and converts them to the digital domain with 2 high resolution high speed ADCs. To adapt to varying signal channel conditions, RF filters, automatic gain control (AGC), DC offset cancelation circuits and baseband filters are integrated within ESP8266EX.

3.3.3. 2.4 GHz Transmitter

The 2.4 GHz transmitter up-converts the quadrature baseband signals to 2.4 GHz, and drives the antenna with a high-power CMOS power amplifier. The function of digital calibration further improves the linearity of the power amplifier, enabling a state of art performance of delivering +19.5 dBm average TX power for 802.11b transmission and +18 dBm for 802.11n (MSC0) transmission.

Additional calibrations are integrated to offset any imperfections of the radio, such as:

- Carrier leakage
- I/Q phase matching
- Baseband nonlinearities

These built-in calibration functions reduce the product test time and make the test equipment unnecessary.

3.3.4. Clock Generator

The clock generator generates quadrature 2.4 GHz clock signals for the receiver and transmitter. All components of the clock generator are integrated on the chip, including all inductors, varactors, loop filters, linear voltage regulators and dividers.

The clock generator has built-in calibration and self test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms to ensure the best performance of the receiver and transmitter.

3.4. Wi-Fi

ESP8266EX implements TCP/IP and full 802.11 b/g/n WLAN MAC protocol. It supports Basic Service Set (BSS) STA and SoftAP operations under the Distributed Control Function (DCF). Power management is handled with minimum host interaction to minimize active-duty period.

3.4.1. Wi-Fi Radio and Baseband

The ESP8266EX Wi-Fi Radio and Baseband support the following features:

- 802.11b and 802.11g
- 802.11n MCS0-7 in 20 MHz bandwidth
- 802.11n 0.4 μs guard-interval
- up to 72.2 Mbps of data rate

- Receiving STBC 2x1
- Up to 20.5 dBm of transmitting power
- Adjustable transmitting power
- Antenna diversity

3.4.2. Wi-Fi MAC

The ESP8266EX Wi-Fi MAC applies low-level protocol functions automatically, as follows:

- 2 x virtual Wi-Fi interfaces
- Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- Request To Send (RTS), Clear To Send (CTS) and Immediate Block ACK
- Defragmentation
- CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WEP (RC4) and CRC
- Automatic beacon monitoring (hardware TSF)
- Dual and single antenna Bluetooth co-existence support with optional simultaneous receive (Wi-Fi/Bluetooth) capability

3.5. Power Management

ESP8266EX is designed with advanced power management technologies and intended for mobile devices, wearable electronics and the Internet of Things applications.

The low-power architecture operates in the following modes:

- Active mode: The chip radio is powered on. The chip can receive, transmit, or listen.
- Modem-sleep mode: The CPU is operational. The Wi-Fi and radio are disabled.
- Light-sleep mode: The CPU and all peripherals are paused. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
- Deep-sleep mode: Only the RTC is operational and all other part of the chip are powered off.

Table 3-4. Power Consumption by Power Modes

Power Mode	Description	Power Consumption
Active (RF working)	Wi-Fi TX packet	Please refer to 5-2.
	Wi-Fi RX packet	
Modem-sleep①	CPU is working	15 mA
Light-sleep②	-	0.9 mA
Deep-sleep③	Only RTC is working	20 uA
Shut down	-	0.5 uA

Notes:

① ***Modem-sleep*** *mode is used in the applications that require the CPU to be working, as in PWM or I2S applications. According to 802.11 standards (like U-APSD), it shuts down the Wi-Fi Modem circuit while maintaining a Wi-Fi connection with no data transmission to optimize power consumption. E.g. in DTIM3, maintaining a sleep of 300 ms with a wakeup of 3 ms cycle to receive AP's Beacon packages at interval requires about 15 mA current.*

② *During* ***Light-sleep*** *mode, the CPU may be suspended in applications like Wi-Fi switch. Without data transmission, the Wi-Fi Modem circuit can be turned off and CPU suspended to save power consumption according to the 802.11 standards (U-APSD). E.g. in DTIM3, maintaining a sleep of 300 ms with a wakeup of 3ms to receive AP's Beacon packages at interval requires about 0.9 mA current.*

③ *During* ***Deep-sleep*** *mode, Wi-Fi is turned off. For applications with long time lags between data transmission, e.g. a temperature sensor that detects the temperature every 100s, sleeps for 300s and wakes up to connect to the AP (taking about 0.3 ~ 1s), the overall average current is less than 1mA. The current of 20 μA is acquired at the voltage of 2.5V.*

4. Peripheral Interface

4.1. General Purpose Input/Output Interface (GPIO)

ESP8266EX has 17 GPIO pins which can be assigned to various functions by programming the appropriate registers.

Each GPIO PAD can be configured with internal pull-up or pull-down (XPD_DCDC can only be configured with internal pull-down, other GPIO PAD can only be configured with internal pull-up), or set to high impedance. When configured as an input, the data are stored in software registers; the input can also be set to edge-trigger or level trigger CPU interrupts. In short, the IO pads are bi-directional, non-inverting and tristate, which includes input and output buffer with tristate control inputs.

These pins, when working as GPIOs, can be multiplexed with other functions such as I2C, I2S, UART, PWM, and IR Remote Control, etc.

For low power operations, the GPIOs can also be set to hold their state. For instance, when the IOs are not driven by internal and external circuits, all outputs will hold their states before the chip entered the low power modes.

The required drive strength is small— 5 μA or more is enough to pull apart the latch.

4.2. Secure Digital Input/Output Interface (SDIO)

ESP8266EX has one Slave SDIO, the definitions of which are described as Table 4-1, which supports 25 MHz SDIO v1.1 and 50 MHz SDIO v2.0, and 1 bit/4 bit SD mode and SPI mode.

Table 4-1. Pin Definitions of SDIOs

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SDIO_CLK
SDIO_DATA0	22	IO7	SDIO_DATA0
SDIO_DATA1	23	IO8	SDIO_DATA1
SDIO_DATA_2	18	IO9	SDIO_DATA_2
SDIO_DATA_3	19	IO10	SDIO_DATA_3
SDIO_CMD	20	IO11	SDIO_CMD

4.3. Serial Peripheral Interface (SPI/HSPI)

ESP8266EX has two SPIs.

- One general Slave/Master SPI
- One general Slave HSPI

Functions of all these pins can be implemented via hardware.

4.3.1. General SPI (Master/Slave)

Table 4-2. Pin Definitions of SPIs

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SPICLK
SDIO_DATA0	22	IO7	SPIQ/MISO
SDIO_DATA1	23	IO8	SPID/MOSI
SDIO_DATA_2	18	IO9	SPIHD
SDIO_DATA_3	19	IO10	SPIWP
U0TXD	26	IO1	SPICS1
GPIO0	15	IO0	SPICS2
SDIO_CMD	20	IO11	SPICS0

Note:

SPI mode can be implemented via software programming. The clock frequency is 80 MHz at maximum when working as a master, 20 MHz at maximum when working as a slave.

4.3.2. HSPI (Slave)

Table 4-3. Pin Definitions of HSPI (Slave)

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	HSPICLK
MTDI	10	IO12	HSPIQ/MISO
MTCK	12	IO13	HSPID/MOSI
MTDO	13	IO15	HPSICS

Note:

SPI mode can be implemented via software programming. The clock frequency is 20 MHz at maximum.

4.4. I2C Interface

ESP8266EX has one I2C, which is realized via software programming, used to connect with other microcontrollers and other peripheral equipments such as sensors. The pin definition of I2C is as below.

Table 4-4. Pin Definitions of I2C

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	I2C_SCL
GPIO2	14	IO2	I2C_SDA

Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized via software programming, and the clock frequency is 100 kHz at maximum.

4.5. I2S Interface

ESP8266EX has one I2S data input interface and one I2S data output interface, and supports the linked list DMA. I2S interfaces are mainly used in applications such as data collection, processing, and transmission of audio data, as well as the input and output of serial data. For example, LED lights (WS2812 series) are supported. The pin definition of I2S is shown in Table 4-5.

Table 4-5. Pin Definitions of I2S

I2S Data Input			
Pin Name	Pin Num	IO	Function Name
MTDI	10	IO12	I2SI_DATA
MTCK	12	IO13	I2SI_BCK
MTMS	9	IO14	I2SI_WS
MTDO	13	IO15	I2SO_BCK
U0RxD	25	IO3	I2SO_DATA
GPIO2	14	IO2	I2SO_WS

4.6. Universal Asynchronous Receiver Transmitter (UART)

ESP8266EX has two UART interfaces UART0 and UART1, the definitions are shown in Table 4-6.

Table 4-6. Pin Definitions of UART

Pin Type	Pin Name	Pin Num	IO	Function Name
UART0	U0RXD	25	IO3	U0RXD
	U0TXD	26	IO1	U0TXD
	MTDO	13	IO15	U0RTS
	MTCK	12	IO13	U0CTS
UART1	GPIO2	14	IO2	U1TXD
	SD_D1	23	IO8	U1RXD

Data transfers to/from UART interfaces can be implemented via hardware. The data transmission speed via UART interfaces reaches 115200 x 40 (4.5 Mbps).

UART0 can be used for communication. It supports flow control. Since UART1 features only data transmit signal (TX), it is usually used for printing log.

> **Note:**
>
> *By default, UART0 outputs some printed information when the device is powered on and booting up. The baud rate of the printed information is relevant to the frequency of the external crystal oscillator. If the frequency of the crystal oscillator is 40 MHz, then the baud rate for printing is 115200; if the frequency of the crystal oscillator is 26 MHz, then the baud rate for printing is 74880. If the printed information exerts any influence on the functionality of the device, it is suggested to block the printing during the power-on period by changing (U0TXD, U0RXD) to (MTDO, MTCK).*

4.7. Pulse-Width Modulation (PWM)

ESP8266EX has four PWM output interfaces. They can be extended by users themselves. The pin definitions of the PWM interfaces are defined as below.

Table 4-7. Pin Definitions of PWM

Pin Name	Pin Num	IO	Function Name
MTDI	10	IO12	PWM0
MTDO	13	IO15	PWM1
MTMS	9	IO14	PWM2
GPIO4	16	IO4	PWM3

The functionality of PWM interfaces can be implemented via software programming. For example, in the LED smart light demo, the function of PWM is realized by interruption of the timer, the minimum resolution reaches as high as 44 ns. PWM frequency range is adjustable from 1000 μs to 10000 μs, i.e., between 100 Hz and 1 kHz. When the PWM frequency is 1 kHz, the duty ratio will be 1/22727, and a resolution of over 14 bits will be achieved at 1 kHz refresh rate.

4.8. IR Remote Control

ESP8266EX currently supports one infrared remote control interface. For detailed pin definitions, please see Table 4-8 below.

Table 4-8. Pin Definitions of IR Remote Control

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	IR TX
GPIO5	24	IO 5	IR Rx

The functionality of Infrared remote control interface can be implemented via software programming. NEC coding, modulation, and demodulation are supported by this interface. The frequency of modulated carrier signal is 38 kHz, while the duty ratio of the square wave is 1/3. The transmission range is around 1m which is determined by two factors: one is the maximum current drive output, the other is internal current-limiting resistance value in the infrared receiver. The larger the resistance value, the lower the current, so is the power, and vice versa.

4.9. ADC (Analog-to-Digital Converter)

ESP8266EX is embedded with a 10-bit precision SAR ADC. TOUT (Pin6) is defined as below:

Table 4-9. Pin Definition of ADC

Pin Name	Pin Num	Function Name
TOUT	6	ADC Interface

The following two measurements can be implemented using ADC (Pin6). However, they cannot be implemented at the same time.

- Measure the power supply voltage of VDD3P3 (Pin3 and Pin4).

Hardware Design	TOUT must be floating.
RF Initialization Parameter	The 107th byte of ***esp_init_data_default.bin*** (0 ~ 127 bytes), `vdd33_const` must be set to `0xFF`.
RF Calibration Process	Optimize the RF circuit conditions based on the testing results of VDD3P3 (Pin3 and Pin4).
User Programming	Use `system_get_vdd33` instead of `system_adc_read`.

- Measure the input voltage of TOUT (Pin6).

Hardware Design	The input voltage range is 0 to 1.0V when TOUT is connected to external circuit.

RF Initialization Parameter	The value of the 107th byte of ***esp_init_data_default.bin*** (0 ~ 127 bytes), `vdd33_const` must be set to the real power supply voltage of Pin3 and Pin4. The unit and effective value range of `vdd33_const` is 0.1V and 18 to 36, respectively, thus making the working power voltage range of ESP8266EX between 1.8V and 3.6V.
RF Calibration Process	Optimize the RF circuit conditions based on the value of `vdd33_const`. The permissible error is ±0.2V.
User Programming	Use `system_adc_read` instead of `system_get_vdd33`.

📖 ***Notes:***

esp_init_data_default.bin *is provided in SDK package which contains RF initialization parameters (0 ~ 127 bytes). The name of the 107th byte in* ***esp_init_data_default.bin*** *is* `vdd33_const`*, which is defined as below:*

- *When vdd33_const = 0xff, the power voltage of Pin3 and Pin4 will be tested by the internal self-calibration process of ESP8266EX itself. RF circuit conditions should be optimized according to the testing results.*
- *When 18 =< vdd33_const =< 36, ESP8266EX RF Calibration and optimization process is implemented via (vdd33_const/10).*
- *When vdd33_const < 18 or 36 < vdd33_const < 255, vdd33_const is invalid. ESP8266EX RF Calibration and optimization process is implemented via the default value 3.3V.*

5. Electrical Specifications

5.1. Electrical Characteristics

Table 5-1. Electrical Characteristics

Parameters		Conditions	Min	Typical	Max	Unit
Operating Temperature Range		-	–40	Normal	125	°C
Maximum Soldering Temperature		IPC/JEDEC J-STD-020	-	-	260	°C
Working Voltage Value		-	2.5	3.3	3.6	V
I/O	V_{IL}	-	–0.3	-	$0.25V_{IO}$	V
	V_{IH}		$0.75V_{IO}$		3.6	
	V_{OL}	-	-	-	$0.1V_{IO}$	
	V_{OH}		$0.8V_{IO}$		-	
	I_{MAX}	-	-	-	12	mA
Electrostatic Discharge (HBM)		TAMB=25°C	-	-	2	KV
Electrostatic Discharge (CDM)		TAMB=25°C	-	-	0.5	KV

5.2. RF Power Consumption

Unless otherwise specified, the power consumption measurements are taken with a 3.0V supply at 25°C of ambient temperature. All transmitters' measurements are based on a 50% duty cycle.

Table 5-2. Power Consumption

Parameters	Min	Typical	Max	Unit
TX 802.11b, CCK 11Mbps, P_{OUT}=+17 dBm	-	170	-	mA
TX 802.11g, OFDM 54Mbps, P_{OUT}=+15 dBm	-	140	-	mA
TX 802.11n, MCS7, P_{OUT}=+13dBm	-	120	-	mA
Rx 802.11b, 1024 bytes packet length , –80 dBm	-	50	-	mA
Rx 802.11g, 1024 bytes packet length, –70 dBm	-	56	-	mA
Rx 802.11n, 1024 bytes packet length, –65 dBm	-	56	-	mA

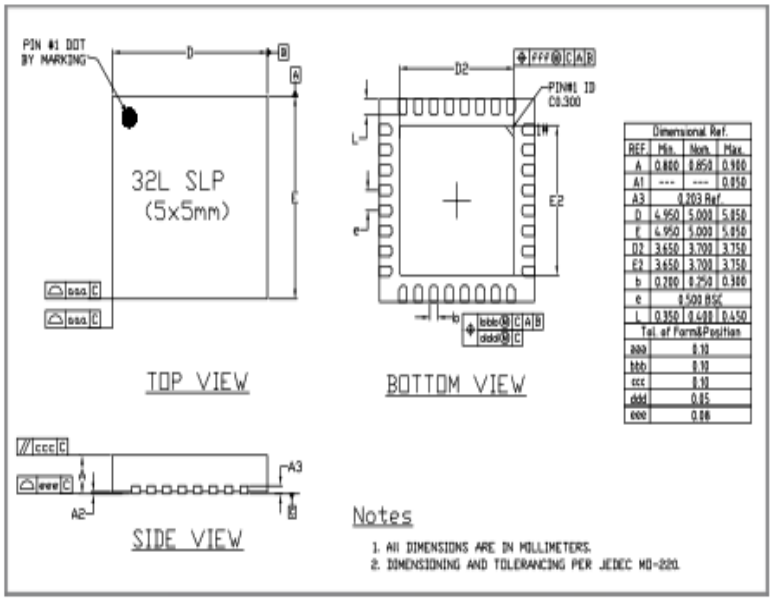
5.3. Wi-Fi Radio Characteristics

The following data are from tests conducted at room temperature, with a 3.3V power supply.

Table 5-3. Wi-Fi Radio Characteristics

Parameters	Min	Typical	Max	Unit
Input frequency	2412	-	2484	MHz
Output impedance	-	39+j6	-	Ω
Output power of PA for 72.2 Mbps	15.5	16.5	17.5	dBm
Output power of PA for 11b mode	19.5	20.5	21.5	dBm
Sensitivity				
DSSS, 1 Mbps	-	–98	-	dBm
CCK, 11 Mbps	-	–91	-	dBm
6 Mbps (1/2 BPSK)	-	–93	-	dBm
54 Mbps (3/4 64-QAM)	-	–75	-	dBm
HT20, MCS7 (65 Mbps, 72.2 Mbps)	-	–72	-	dBm
Adjacent Channel Rejection				
OFDM, 6 Mbps	-	37	-	dB
OFDM, 54 Mbps	-	21	-	dB
HT20, MCS0	-	37	-	dB
HT20, MCS7	-	20	-	dB

6. Package Information



Dimensional Ref.			
REF.	Min.	Nom.	Max.
A	0.800	0.850	0.900
A1	---	---	0.050
A3	0.203 Ref.		
D	4.950	5.000	5.050
E	4.950	5.000	5.050
D2	3.650	3.700	3.750
E2	3.650	3.700	3.750
b	0.200	0.250	0.300
e	0.500 BSC		
L	0.350	0.400	0.450
Tol. of Form&Position			
aaa	0.10		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		

Figure 6-1. ESP8266EX Package

I. Appendix - Pin List

For detailed pin information, please see *ESP8266 Pin List*.

- Digital Die Pin List
- Buffer Sheet
- Register List
- Strapping List

> **Notes:**
>
> - *INST_NAME refers to the IO_MUX REGISTER defined in* **eagle_soc.h**, *for example MTDI_U refers to PERIPHS_IO_MUX_MTDI_U.*
> - *Net Name refers to the pin name in schematic.*
> - *Function refers to the multifunction of each pin pad.*
> - *Function number 1 ~ 5 correspond to FUNCTION 0 ~ 4 in SDK. For example, set MTDI to GPIO12 as follows.*
> - `#define FUNC_GPIO12  3 //defined in eagle_soc.h`
> - `PIN_FUNC_SELECT(PERIPHS_IO_MUX_MTDI_U,FUNC_GPIO12)`

II. Appendix - Learning Resources

II.1. Must-Read Documents

- *ESP8266 Quick Start Guide*

  Description: This document is a quick user guide to getting started with ESP8266. It includes an introduction to the ESP-LAUNCHER, instructions on how to download firmware to the board and run it, how to compile the AT application, as well as the structure and debugging method of RTOS SDK. Basic documentation and other related resources for the ESP8266 are also provided.

- *ESP8266 SDK Getting Started Guide*

  Description: This document takes ESP-LAUNCHER and ESP-WROOM-02 as examples of how to use the ESP8266 SDK. The contents include preparations before compilation, SDK compilation and firmware download.

- *ESP8266 Pin List*

  Description: This link directs you to a list containing the type and function of every ESP8266 pin.

- *ESP8266 Hardware Design Guideline*

  Description: This document provides a technical description of the ESP8266 series of products, including ESP8266EX, ESP-LAUNCHER and ESP-WROOM.

- *ESP8266 Hardware Matching Guide*

  Description: This document introduces the frequency offset tuning and antenna impedance matching for ESP8266 in order to achieve optimal RF performance.

- *ESP8266 Technical Reference*

  Description: This document provides an introduction to the interfaces integrated on ESP8266. Functional overview, parameter configuration, function description, application demos and other pieces of information are included.

- *ESP8266 Hardware Resources*

  Description: This zip package includes manufacturing BOMs, schematics and PCB layouts of ESP8266 boards and modules.

- *FAQ*

II.2. Must-Have Resources

- *ESP8266 SDKs*

Description: This webpage provides links both to the latest version of the ESP8266 SDK and the older ones.

- *ESP8266 Tools*

  Description: This webpage provides links to both the ESP8266 flash download tools and the ESP8266 performance evaluation tools.

- *ESP8266 Apps*
- *ESP8266 Certification and Test Guide*
- *ESP8266 BBS*
- *ESP8266 Resources*

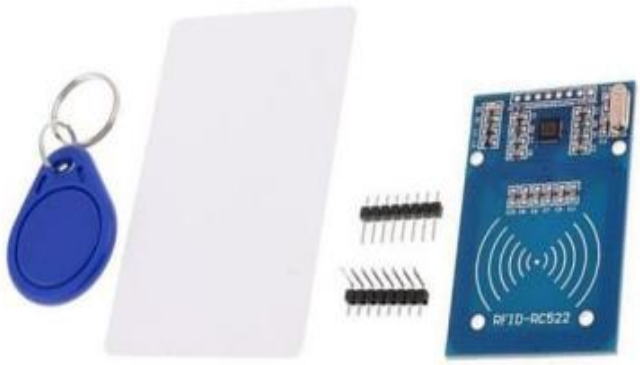
Datasheet RFID RC522

Handson Technology

Data Specs

RC522 RFID Development Kit

This RC522 RFID Development kit is based on NXP's a highly integrated reader/writer IC MFRC522 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 14443A cards and transponders without additional active circuitry. The receiver module provides a robust and efficient implementation for demodulating and decoding signals from ISO/IEC 14443A compatible cards and transponders.

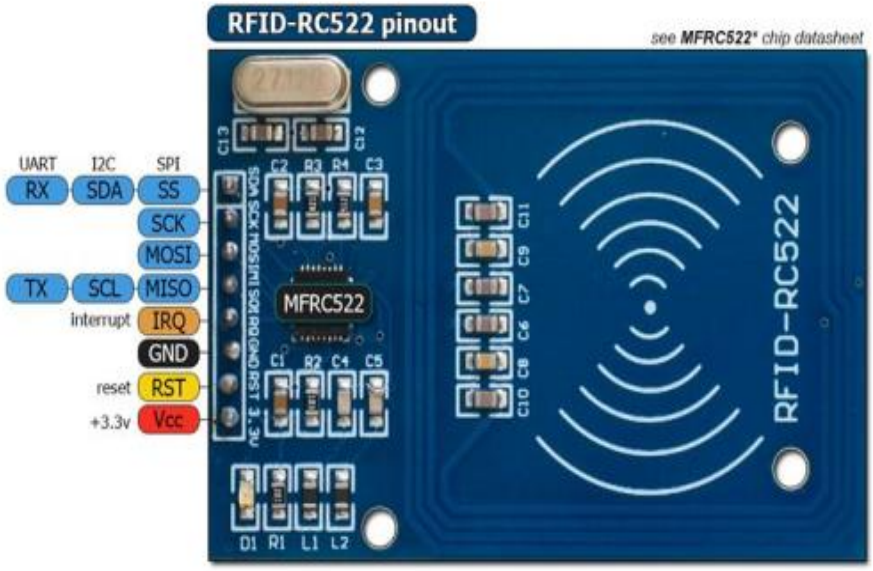


SKU: MDU1040

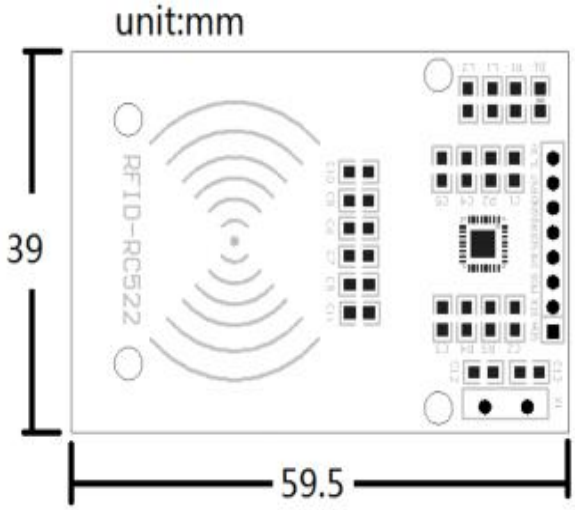
Brief Data:

- Operating Voltage: 2.5V~3.3V.
- Operating/Standby current: 13~26mA/10~13mA.
- Operating Frequency: 13.56MHz.
- Supports ISO/IEC 14443A higher transfer speed communication up to 848 KBd.
- SPI bus speed up to 10Mbit/s.
- I2C-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.
- Compatible with MIFARE and ISO 14443A cards.
- Typical operating distance in Read/Write mode up to 50 mm depending on the antenna size and tuning.

Interface Pins Function:



Mechanical Dimension:



P.S: This module does not support RFID cards which operate at 125KHz frequency range. It supports only the cards which operate at 13.56MHz frequency range.

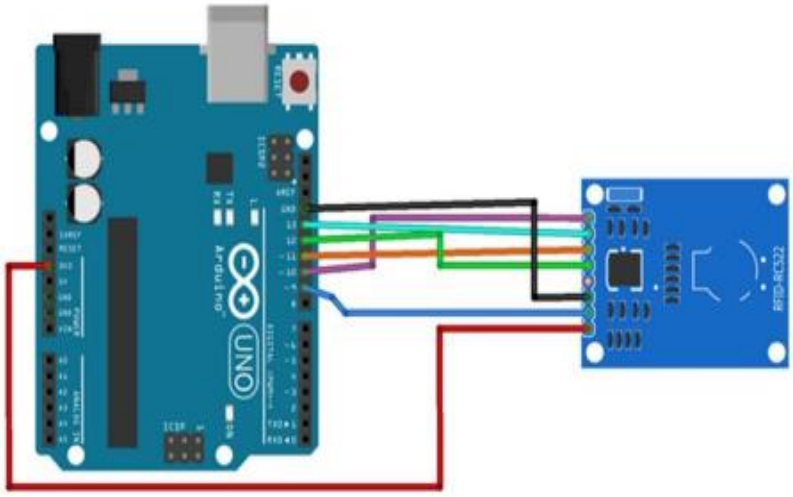
Application Example with Arduino:

Library download:

Here's the library you need for this project:

1. Download the RFID library here created by miguelbalboa
2. Unzip the RFID library
3. Install the RFID library in your Arduino IDE
4. Restart your Arduino IDE

Arduino Circuit Connection:



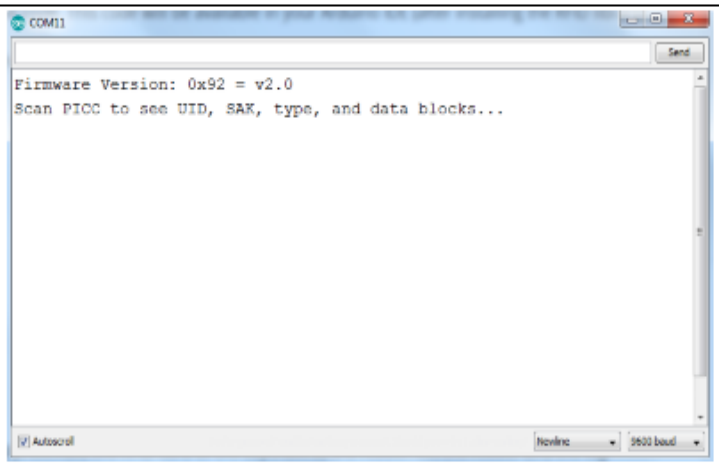
RC522 Pin	Wiring to Arduino Un
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
MISO	Digital 12
IRQ	unconnected
GND	GND
RST	Digital 9
3.3V Supply	3.3V

Pin Wiring

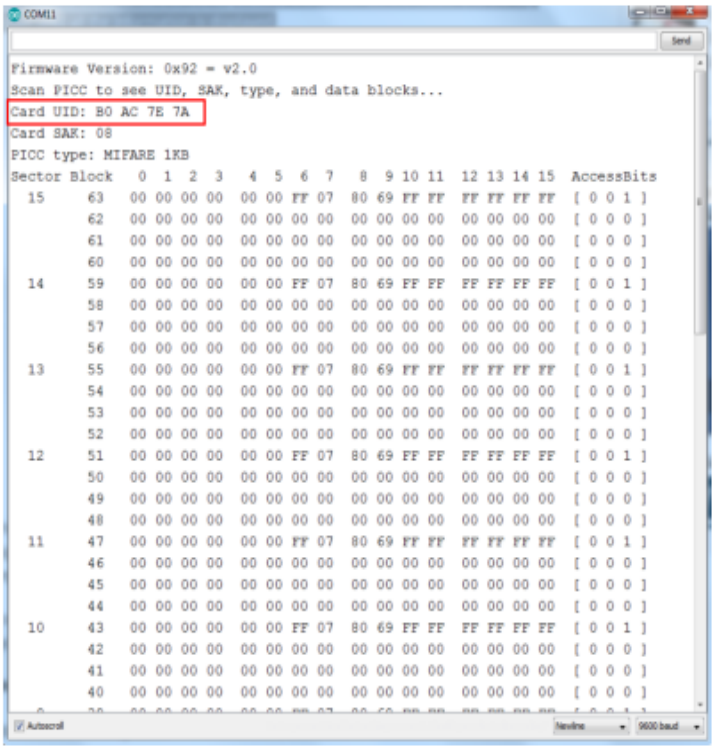
Reading Data from a RFID Tag:

After having the circuit ready, go to File > Examples > MFRC522 > DumpInfo and upload the code. This code will be available in your Arduino IDE (after installing the RFID library).

Then, open the serial monitor with 9600 baud. You should see something like the figure below:



Put the RFID card or the keychain to the reader. Let the reader and the tag closer until all the information is displayed.



This is the information that you can read from the card, including the card UID that is highlighted in red. The information is stored in the memory that is divided into segments and blocks as you can see in the previous picture.

You have 1024 bytes of data storage divided into 16 sectors and each sector is protected by two different keys, A and B.

Write down your UID card because you'll need it later. In this case, Card UID: B0 AC 7E 7A.

Upload the following code to the Arduino Board:

==========

```
/*
 *
 * All the resources for this project:
 * Modified by Handson Technology
 * www.handsontec.com
 * Created by Handsontec Tech team
 *
 */

#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);     // Create MFRC522 instance.

void setup()
{
 Serial.begin(9600);        // Initiate a serial communication
 SPI.begin();               // Initiate  SPI bus
 mfrc522.PCD_Init();        // Initiate MFRC522
 Serial.println("Put close your card to the reader...");
 Serial.println();

}

void loop()
{
 // Look for new cards
 if ( ! mfrc522.PICC_IsNewCardPresent())
 {
  return;
 }
 // Select one of the cards
 if ( ! mfrc522.PICC_ReadCardSerial())
 {
  return;
 }

 //Show UID on serial monitor
 Serial.print("UID tag :");
 String content= "";
 byte letter;
 for (byte i = 0; i < mfrc522.uid.size; i++)
 {
   Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
   Serial.print(mfrc522.uid.uidByte[i], HEX);
   content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
   content.concat(String(mfrc522.uid.uidByte[i], HEX));
 }

 Serial.println();
 Serial.print("Message : ");
 content.toUpperCase();
```

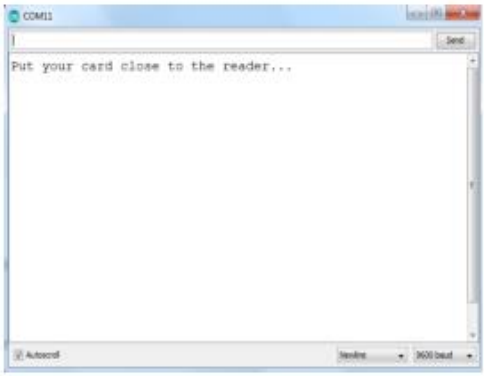
```
 if (content.substring(1) == "B0 AC 7E 7A")          //change here the UID of the card/cards that you want to give access
 {
  Serial.println("Authorized access");
  Serial.println();
  delay(3000);
 }

 else  {
  Serial.println(" Access denied");
  delay(3000);
 }
}
```

==========

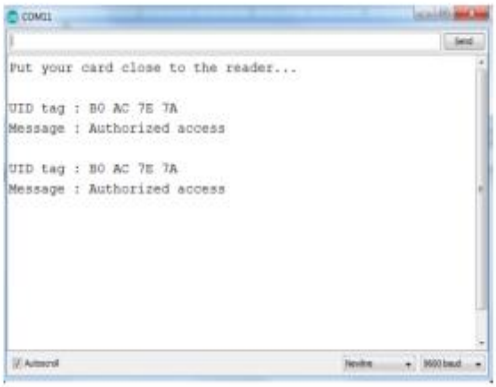
In the piece of code above you need to change the if (content.substring(1) == "REPLACE WITH YOUR UID") and type the UID card you've written previously.

Demonstration:

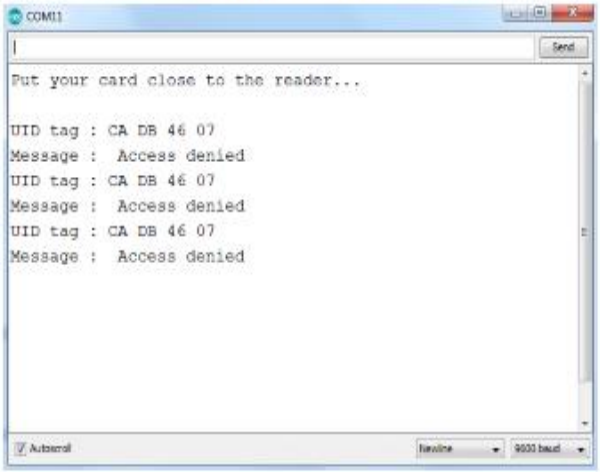
Now, upload the code to your Arduino and open the serial monitor. You will see the screen as below:



Put the card you've chosen to give access (in this case with UID: B0 AC 7E 7A) near to the reader module and you'll see:



Now if you put close another tag with another UID, the denial message will show up:



This completed our initial testing & setup for RC522 RFID reader module, hope you found this tutorial useful.

This RFID Development Package include:

- 1x RFID RC522 Reader Module
- 1x Credit card size RFID Card (M1 S50 IC Card)
- 1x RFID Keyring (M1 S50 Key tag)
- 1x 8-pin strait header connector
- 1x 8-pin right angle header connector

Web Resource:

- MDU1040 RC522
- Course Using the MF522 RFID Reader with the Arduino.
- MF1 S50 ISO/IEC14443A Contactless RFID Card
- MF1 S50 ISO/IEC14443A Contactless RFID Key Chain

Handsontec.com

We have the parts for your ideas

HandsOn Technology provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analog and digital, practical and theoretical; software and hardware.

HandsOn Technology support Open Source Hardware (OSHW) Development Platform.

Learn : Design : Share

www.handsontec.com

The Face behind our product quality…

In a world of constant change and continuous technological development, a new or replacement product is never far away – and they all need to be tested.

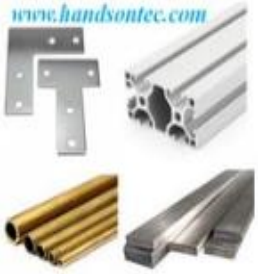
Many vendors simply import and sell wihtout checks and this cannot be the ultimate interests of anyone, particularly the customer. Every part sell on Handsotec is fully tested. So when buying from Handsontec products range, you can be confident you're getting outstanding quality and value.

We keep adding the new parts so that you can get rolling on your next project.

Breakout Boards & Modules

Connectors

Electro-Mechanical Parts



Engineering Material

Mechanical Hardware

Electronics Components

P

Power Supply

ARDUINO™

Arduino Board & Shield

Tools & Accessory

Kode program RFID

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN D4
#define RST_PIN D3
#define LED D2
MFRC522 mfrc522(SS_PIN, RST_PIN);
const char* ssid = "Sinau Tech";
const char* password = "HURUFKECIL@321";
//Your Domain name with URL path or IP address with path
const char* serverName = "http://192.168.18.182/skripsiku2022/public/node/add";
void rfid() {
SPI.begin();
mfrc522.PCD_Init();
}
void wifiinit() {
WiFi.begin(ssid, password);
Serial.println("Connecting");
while (WiFi.status() != WL_CONNECTED) {
digitalWrite(LED, !digitalRead(LED));
delay(100);
}
digitalWrite(LED, LOW);
}
void setup() {
Serial.begin(115200);
```

```
pinMode(LED, OUTPUT);
wifiinit();
rfid();
Serial.println("");
Serial.print("Connected to WiFi network with IP Address: ");
Serial.println(WiFi.localIP());
Serial.println("Timer set to 5 seconds (timerDelay variable), it will take 5 seconds
before publishing the first reading.");
}
void loop() {
if (!mfrc522.PICC_IsNewCardPresent()) return;
// Select one of the cards
if (!mfrc522.PICC_ReadCardSerial()) return;
//Show UID on serial monitor
//Send an HTTP POST request every 10 minutes
String data = "{\"api_key\":\"lulus\",\"rfid\":\"" + getrfid() + "\"}";
Serial.println(data);
if (getrfid() != "") {
//Check WiFi connection status
if (WiFi.status() == WL_CONNECTED) {
WiFiClient client;
HTTPClient http;
// Your Domain name with URL path or IP address with path
http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(data);
String payload = http.getString();
Serial.print("Response : ");
Serial.println(payload);
Serial.print("HTTP Response code: ");
```

```
Serial.println(httpResponseCode);
http.end();
} else {
Serial.println("WiFi Disconnected");
}
}
digitalWrite(LED, !digitalRead(LED));
delay(500);
}
String getrfid() {
Serial.print("UID tag :");
String content = "";
for (byte i = 0; i < mfrc522.uid.size; i++) {
Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
content.toUpperCase();
return content;
}
```

Kode program tampilan Web

1. Pendaftaran

```
<?php
// Path to the front controller (this file)
define('FCPATH', __DIR__ . DIRECTORY_SEPARATOR);
// Ensure the current directory is pointing to the front controller's directory
chdir(FCPATH);
/*
 *---------------------------------------------------------------
 * BOOTSTRAP THE APPLICATION
 *---------------------------------------------------------------
 * This process sets up the path constants, loads and registers
 * our autoloader, along with Composer's, loads our constants
 * and fires up an environment-specific bootstrapping.
 */
// Load our paths config file
// This is the line that might need to be changed, depending on your folder structure.
require FCPATH . '../app/Config/Paths.php';
// ^^^ Change this line if you move your application folder
$paths = new Config\Paths();
// Location of the framework bootstrap file.
require rtrim($paths->systemDirectory, '\/ ') . DIRECTORY_SEPARATOR . 'bootstrap.php';
// Load environment settings from .env files into $_SERVER and $_ENV
require_once SYSTEMPATH . 'Config/DotEnv.php';
(new CodeIgniter\Config\DotEnv(ROOTPATH))->load();
/*
 * ---------------------------------------------------------------
 * GRAB OUR CODEIGNITER INSTANCE
 * ---------------------------------------------------------------
```

```
 *
 * The CodeIgniter class contains the core functionality to make
 * the application run, and does all of the dirty work to get
 * the pieces all working together.
 */
$app = Config\Services::codeigniter();
$app->initialize();
$context = is_cli() ? 'php-cli' : 'web';
$app->setContext($context);
/*
 *---------------------------------------------------------------
 * LAUNCH THE APPLICATION
 *---------------------------------------------------------------
 * Now that everything is setup, it's time to actually fire
 * up the engines and make this app do its thang.
 */
$app->run();
```

2. User

```
<!doctype html>
<html lang="en" class="h-100">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Aplikasi Antrian Berbasis Web">
    <meta name="author" content="Indra Styawantoro">
    <!-- Title -->
    <title>Aplikasi Antrian Berbasis Web</title>
```

```
    <!-- Favicon icon -->

    <link rel="shortcut icon" href="../../assets/img/favicon.ico" type="image/x-icon">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x" crossorigin="anonymous">

<!-- Bootstrap Icons -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">

 <!-- Font -->

<link href="https://fonts.googleapis.com/css?family=Raleway:100,200,300,400,500,600,700,800,900&amp;display=swap" rel="stylesheet">

<!-- DataTables -->

<link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/v/bs5/dt-1.10.25/datatables.min.css" />

    <!-- Custom Style -->

    <link rel="stylesheet" href="../assets/css/style.css">

</head>

<body class="d-flex flex-column h-100">

 <main class="flex-shrink-0">

<div class="container pt-4">

<div class="d-flex flex-column flex-md-row px-4 py-3 mb-4 bg-white rounded-2 shadow-sm">

<!-- judul halaman -->

<div class="d-flex align-items-center me-md-auto">

<i class="bi-mic-fill text-success me-3 fs-3"></i>

<h1 class="h5 pt-2">Panggilan Antrian</h1>

</div>

<!-- breadcrumbs -->
```

```
<div class="ms-5 ms-md-0 pt-md-3 pb-md-0">
<nav style="--bs-breadcrumb-divider: '>';" aria-label="breadcrumb">
<ol class="breadcrumb">
<li class="breadcrumb-item"><a href="https://pustakakoding.com/"><i class="bi-house-fill text-success"></i></a></li>
<li class="breadcrumb-item" aria-current="page">Dashboard</li>
<li class="breadcrumb-item" aria-current="page">Antrian</li>
</ol>
</nav>
</div>
</div>
<div class="row">
<!-- menampilkan informasi jumlah antrian -->
<div class="col-md-3 mb-4">
<div class="card border-0 shadow-sm">
<div class="card-body p-4">
<div class="d-flex justify-content-start">
// auto reload data antrian setiap 1 detik untuk menampilkan data secara realtime
setInterval(function() {
$('#jumlah-antrian').load('../panggilan/get_jumlah_antrian.php').fadeIn("slow");
$('#antrian-sekarang').load('../panggilan/get_antrian_sekarang.php').fadeIn("slow");
$('#antrian-selanjutnya').load('../panggilan/get_antrian_selanjutnya.php').fadeIn("slow");
$('#sisa-antrian').load('../panggilan/get_sisa_antrian.php').fadeIn("slow");
table.ajax.reload(null, false);
}, 1000);
});
</script>
</body>
```

```
</html>
```

3. Panggilan

```
<!doctype html>
<html lang="en" class="h-100">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Aplikasi Antrian Berbasis Web">
    <meta name="author" content="Indra Styawantoro">
    <!-- Title -->
    <title>Aplikasi Antrian Berbasis Web</title>
    <!-- Favicon icon -->
    <link rel="shortcut icon" href="../../assets/img/favicon.ico" type="image/x-icon">
    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x" crossorigin="anonymous">
    <!-- Bootstrap Icons -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
    <!-- Font -->
    <link
href="https://fonts.googleapis.com/css?family=Raleway:100,200,300,400,500,600,700,800,900&amp;display=swap" rel="stylesheet">
    <!-- DataTables -->
    <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/v/bs5/dt-1.10.25/datatables.min.css" />
```

```
    <!-- Custom Style -->
    <link rel="stylesheet" href="../assets/css/style.css">
</head>
<body class="d-flex flex-column h-100">
<main class="flex-shrink-0">
<div class="container pt-4">
<div class="d-flex flex-column flex-md-row px-4 py-3 mb-4 bg-white rounded-2 shadow-sm">
<!-- judul halaman -->
<div class="d-flex align-items-center me-md-auto">
<i class="bi-mic-fill text-success me-3 fs-3"></i>
<h1 class="h5 pt-2" id="panggilan_antrian">Panggilan Antrian</h1>
</div>
<!-- breadcrumbs -->
<div class="ms-5 ms-md-0 pt-md-3 pb-md-0">
<nav style="--bs-breadcrumb-divider: '>';" aria-label="breadcrumb">
<ol class="breadcrumb">
<li class="breadcrumb-item"><a href="https://pustakakoding.com/"><i class="bi-house-fill text-success"></i></a></li>
<li class="breadcrumb-item" aria-current="page">Dashboard</li>
<li class="breadcrumb-item" aria-current="page">Antrian</li>
</ol>
</nav>
</div>
</div>
<div class="row">
<!-- menampilkan informasi jumlah antrian -->
<div class="col-md-3 mb-4">
<div class="card border-0 shadow-sm">
<div class="card-body p-4">
```

```
<div class="d-flex justify-content-start">
<div class="feature-icon-3 me-4">
<i class="bi-people text-warning"></i>
</div>
<div>
<p id="jumlah-antrian" class="fs-3 text-warning mb-1"></p>
<p class="mb-0">Jumlah Antrian</p>
</div>
</div>
</div>
</div>
</div>
<!-- menampilkan informasi nomor antrian yang sedang dipanggil -->
<div class="col-md-3 mb-4">
<div class="card border-0 shadow-sm">
<div class="card-body p-4">
<div class="d-flex justify-content-start">
<div class="feature-icon-3 me-4">
<i class="bi-person-check text-success"></i>
</div>
<div>
<p id="antrian-sekarang" class="fs-3 text-success mb-1"></p>
<p class="mb-0">Antrian Sekarang</p>
</div>
</div>
</div>
</div>
</div>
<!-- menampilkan informasi nomor antrian yang akan dipanggil selanjutnya -->
```

```
<div class="col-md-3 mb-4">
<div class="card border-0 shadow-sm">
<div class="card-body p-4">
<div class="d-flex justify-content-start">
<div class="feature-icon-3 me-4">
<i class="bi-person-plus text-info"></i>
</div>
<div>
<p id="antrian-selanjutnya" class="fs-3 text-info mb-1"></p>
<p class="mb-0">Antrian Selanjutnya</p>
</div>
</div>
</div>
</div>
</div>
<!-- menampilkan informasi jumlah antrian yang belum dipanggil -->
<div class="col-md-3 mb-4">
<div class="card border-0 shadow-sm">
<div class="card-body p-4">
<div class="d-flex justify-content-start">
<div class="feature-icon-3 me-4">
<i class="bi-person text-danger"></i>
</div>
<div>
<p id="sisa-antrian" class="fs-3 text-danger mb-1"></p>
<p class="mb-0">Sisa Antrian</p>
</div>
</div>
</div>
```

```
</div>
</div>
</div>
<div class="card border-0 shadow-sm">
<div class="card-body p-4">
<div class="table-responsive">
<table id="tabel-antrian" class="table table-bordered table-striped table-hover" width="100%">
<thead>
<tr>
<th>Nomor Antrian</th>
<th>Nama</th>
<th>Panggil</th>
</tr>
 </thead>
 </table>
 </div>
  <button id="btn-reset" class="btn btn-warning">reset</button>
   </div>
   </div>
   </div>
    </main>
    <!-- Footer -->
    <footer class="footer mt-auto py-4">
<div class="container">
<hr class="my-4">
<!-- copyright -->
<div class="text-center py-4">
&copy; 2022 - <a href="Skripsi 2020" target="_blank" class="text-brand text-decoration-none">Skripsi</a>. Sistem Komputer Darmajaya.
```

```
</div>
</div>
</footer>
// mainkan suara nomor antrian
setTimeout(function() {
responsiveVoice.speak("pasien, " + data["nama"] + ", menuju, loket, 1", "Indonesian Male", {
rate: 0.9,
pitch: 1,
volume: 1
});
}, durasi_bell);
// proses update data
$.ajax({
type: "POST", // mengirim data dengan method POST
url: "update.php", // url file proses update data
// tentukan data yang dikirim
data: {
id: id
}
});
});
$("#btn-reset").click(function() {
$.ajax({
url: "<?= 'http://' . $_SERVER['HTTP_HOST'] . '/skripsiku2022/panggilan/reset.php' ?>",
 context: document.body
 }).done(function() {
 location.reload();
 });
```

```
 });
// auto reload data antrian setiap 1 detik untuk menampilkan data secara realtime
setInterval(function() {
$('#jumlah-antrian').load('get_jumlah_antrian.php').fadeIn("slow");
$('#antrian-sekarang').load('get_antrian_sekarang.php').fadeIn("slow");
$('#antrian-selanjutnya').load('get_antrian_selanjutnya.php').fadeIn("slow");
$('#sisa-antrian').load('get_sisa_antrian.php').fadeIn("slow");
table.ajax.reload(null, false);
}, 1000);
});
</script>
</body>
</html>
```