

LAMPIRAN

```

#include <ESP8266WiFi.h>
#include <ESPAsyncWebServer.h>
#include <IRremoteESP8266.h>
#include <IRrecv.h>
#include <IRsend.h>
#include <IRutils.h>

const char* ssid = "V2029";
const char* password = "AAAABBBB";

const uint16_t kRecvPin = D3;
const uint16_t kIrLed = D4;
const int relayPin = D1;

IRrecv irrecv(kRecvPin);
IRsend irsend(kIrLed);
decode_results results;

AsyncWebServer server(80);

String irCodeBuffer = "0xC1AA09F6";

const char* htmlPage = R"rawliteral(
<!DOCTYPE HTML>
<html>
<head>
<title>Gedung G Lantai 3</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
html { font-family: Arial; display: inline-block; text-align: center; }
h2 { font-size: 2.0rem; }
.slider { width: 150px; }
.status { font-size: 1.2rem; margin-top: 20px; }
</style>
</head>
<body>
<h2>Lab Elektronik</h2>
<input type="range" min="0" max="1" value="0" class="slider" id="slider">
<p class="status" id="status"></p>
)rawliteral";

```

```

var sliderValue = this.value;
var xhr = new XMLHttpRequest();
xhr.open("GET", sliderValue == 1 ? "/on" : "/off", true);
xhr.onload = function() {
  if (xhr.status == 200) {
    document.getElementById("status").innerHTML = xhr.responseText;
  }
};
xhr.send();
});
</script>
</body>
</html>
)rawliteral";

void setup() {
  Serial.begin(115200);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Menghubungkan ke WiFi...");
  }
  Serial.println("Terhubung ke WiFi");
  Serial.print("Alamat IP: ");
  Serial.println(WiFi.localIP());

  irrecv.enableIRIn();
  irsend.begin();

  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);

  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", htmlPage);
  });

  server.on("/off", HTTP_GET, [](AsyncWebServerRequest *request){

```

```

Serial.println("Menghubungkan ke WiFi...");
}
Serial.println("Terhubung ke WiFi");
Serial.print("Alamat IP: ");
Serial.println(WiFi.localIP());

irrecv.enableIRIn();
irsend.begin();

pinMode(relayPin, OUTPUT);
digitalWrite(relayPin,HIGH);

server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/html", htmlPage);
});

server.on("/off", HTTP_GET, [](AsyncWebServerRequest *request){
  digitalWrite(relayPin, HIGH);
  request->send(200, "text/plain", "Proyektor dimatikan");
});

server.on("/on", HTTP_GET, [](AsyncWebServerRequest *request){
  digitalWrite(relayPin, LOW);
  delay(2000);
  uint16_t powerOnSignal[] = { 9000, 4500, 600, 1600, 600, 1600, 600, 1600, 600, 1600, 600, 500 };
  irsend.sendRaw(powerOnSignal, sizeof(powerOnSignal) / sizeof(powerOnSignal[0]), 38);
  request->send(200, "text/plain", "Proyektor dihidupkan");
});

server.begin();
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(resultToHumanReadableBasic(&results));
    irCodeBuffer = resultToHumanReadableBasic(&results);
    irrecv.resume();
  }
}

```

Dari gambar-gambar berikut adalah gambar dari kode program yang digunakan untuk memberikan perintah dari skripsi yang telah dibuat berikut penjelasan dari cara kerja kode program tersebut.

Pertama memastikan library sudah sesuai apa yang ingin kita gunakan sesuai program perintah nya contoh :

1. #include <ESPAsyncWebServer.h> : adalah library dari *Asyn Web Server* supaya untuk mendapatkan alamat ip dari esp8266 dan bisa mengakses web server dari *Asyn Web Server*.
2. #include <ESP8266WiFi.h> : adalah sebagai library untuk esp8266 bisa mengakses permintaan memasuki wifi.
3. #include <IRremoteESP8266.h> : adalah sebagai library untuk esp8266 bisa digunakan sebagai remote karna sesuai dengan instruksi dari alat yang dibuat.
4. #include <IRrecv.h> : adalah untuk digunakan sebagai penerima sinyal ir code.
5. #include <IRsend.h> : adalah untuk digunakan sebagai pengirim sinyal ir code ke alat elektronik yang diinginkan.`

Berikut adalah penjelasan dari kode program :

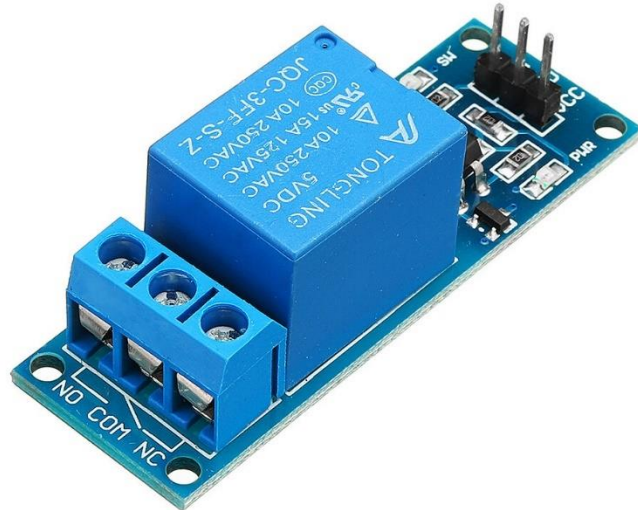
1. const char* ssid = "" : adalah sebagai ssid dari nama wifi yang digunakan.
const char* password = "" : adalah sebagai password dari wifi yang akan digunakan.
2. Ir code buffer adalah sebagai sarana menyimpan kode ir yang sudah di daftarkan.

3. Selanjutnya bagian html digunakan sebagai body web server seperti apa bentuknya bisa menggunakan html program seperti di foto program.
4. Serial.begin menggunakan 115200 sesuai baud rate yang dipakai saya sendiri.
5. `<input type="range" min="0" max="1" value="0" class="slider" id="slider">` : adalah sebagai tombol untuk on/off untuk yang ini saya menggunakan tombol slider.
6. `server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){ request->send_P(200, "text/html", htmlPage);` : untuk menampilkan halaman html yang akan diakses untuk tombol proyektor.
7. Selanjutnya ada `server.on` digunakan sebagai tombol on di *web server* untuk memberikan perintah on ke esp8266.
8. `Server.off` digunakan sebagai memberikan perintah of f ketika tombol di slide ke off.
9. `if (irrecv.decode(&results))` { adalah untuk membaca sinyal ir yang diterima ketika mendaftarkan ir code.
10. `resultToHumanReadableBasic(&results)` digunakan untuk mengonversi hasil decoding IR menjadi format yang lebih mudah dibaca di Serial Monitor.
11. `delay(2000)` : digunakan sebagai ketika alat yang dibuat menghidupkan arus listrik akan delay 2 detik untuk selanjutnya mengirim sinyal ir code.

Selanjutnya akan dibuatkan tahapan-tahapan untuk rancangan :

1. memastikan jaringan wifi sudah hidup.
2. Alat yang akan digunakan sudah terpasang pin nya dengan benar ke board esp8266.
3. Pastikan alat-alat hidup dengan normal.
4. Instal library terlebih dahulu jika belum ada di aplikasi arduino ide.
5. Buat kode program dan pastikan kode program tidak ada salah ketik.
6. Selanjutnya cek di monitor serial apakah sudah tersambung dengan wifi.
7. Cek monitor serial apakah alamat ip sudah ada.
8. Selanjutnya mendaftarkan ir code proyektor melalui remote diarah ke ir receiver dan akan memunculkan hasilnya di monitor serial.
9. Jika sudah ada salin ir code nya dan ditaruh di ir code buffer di kode program.
10. Selanjutnya buka *web server* menggunakan alamat ip yang sudah didapatkan.
11. Tekan tombol on untuk mentest apakah ir transmitter sudah berfungsi secara maksimal.
12. Dan dibawah tombol on/off akan muncul notif apakah proyektor hidup atau mati.

DATA SHEET
1. RELAY 5V



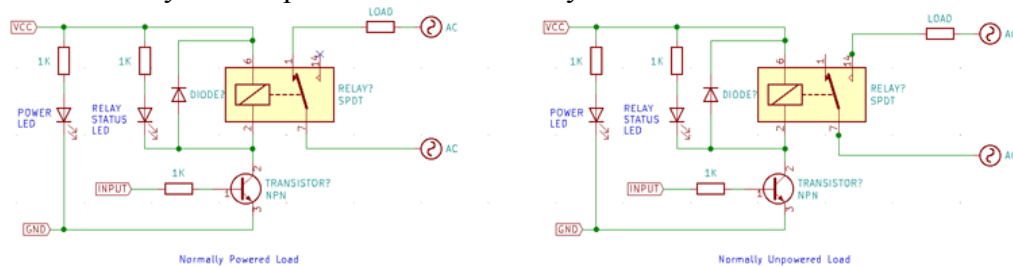
NO	NAMA PIN	keterangan
1	Relay trigger	input untuk mengaktifkan relay
2	ground	Referensi 0v
3	vcc	Pasokan masukan untuk memberi daya pada kumparan relay
4	Normal open	Terminal relay biasanya terbuka
5	Common	Terminal umum relay
6	Normally closed	Kontak relay yang biasanya tertutup

single-Channel Relay Module Specifications

- Supply voltage – 3.75V to 6V
- Quiescent current: 2mA
- Current when the relay is active: ~70mA
- Relay maximum contact voltage – 250VAC or 30VDC
- Relay maximum current – 10A

Modul relay saluran tunggal lebih dari sekedar relay biasa , modul ini berisi komponen-komponen yang memudahkan pengalihan dan penyambungan, serta berfungsi sebagai indikator untuk menunjukkan apakah modul diberi daya dan apakah relay aktif. Pertama adalah blok terminal sekrup. Ini adalah bagian modul yang bersentuhan dengan listrik sehingga diperlukan sambungan yang andal. Penambahan terminal sekrup memudahkan penyambungan kabel listrik tebal, yang

mungkin sulit disolder secara langsung. Ketiga sambungan pada blok terminal tersambung ke terminal relai yang biasanya terbuka, biasanya tertutup, dan umum. Yang kedua adalah relai itu sendiri, yang, dalam hal ini, adalah wadah plastik biru. Banyak informasi dapat diperoleh dari tanda pada relai itu sendiri. Nomor komponen relai di bagian bawah bertuliskan "05VDC", yang berarti bahwa koil relai diaktifkan pada minimum 5V - tegangan apa pun yang lebih rendah dari ini tidak akan dapat menutup kontak relai dengan andal. Ada juga tanda tegangan dan arus, yang mewakili tegangan dan arus maksimum, relai dapat beralih. Misalnya, tanda kiri atas bertuliskan "10A 250VAC", yang berarti relai dapat mengalihkan beban maksimum 10A saat terhubung ke sirkuit listrik 250V. Nilai kiri bawah bertuliskan "10A 30VDC", yang berarti relai dapat mengalihkan arus maksimum 10A DC sebelum kontak rusak. 'LED status relai' menyala setiap kali relai aktif dan memberikan indikasi arus yang mengalir melalui kumparan relai. Jumper input digunakan untuk memasok daya ke koil relai dan LED. Jumper juga memiliki pin input, yang jika ditarik ke atas akan mengaktifkan relai. Transistor switching mengambil input yang tidak dapat menyediakan arus yang cukup untuk menggerakkan koil relai secara langsung dan memperkuatnya menggunakan tegangan suplai untuk menggerakkan koil relai. Dengan cara ini, input dapat digerakkan dari mikrokontroler atau output sensor. Dioda penggerak bebas mencegah lonjakan tegangan saat relai dimatikan. LED daya dihubungkan ke V_{CC} dan menyala setiap kali modul diberi daya.

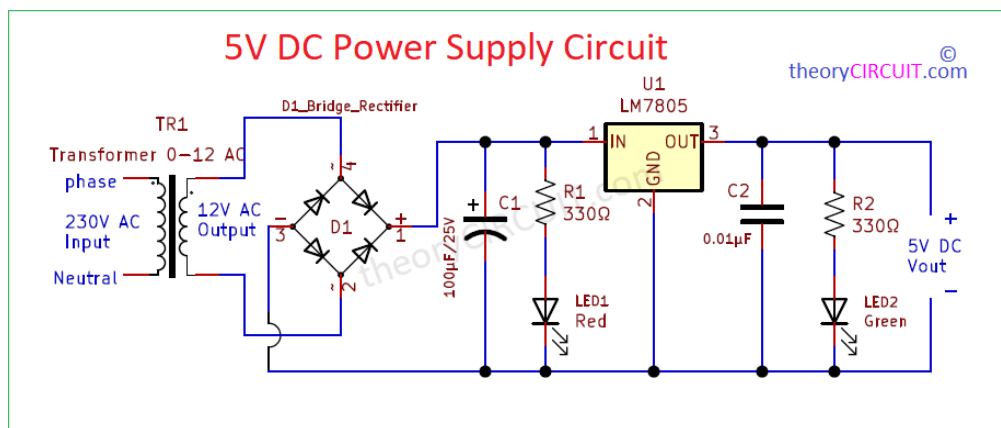


Untuk aplikasi on/off sederhana, relai dapat dihubungkan seperti yang ditunjukkan di atas. Satu terminal listrik dihubungkan ke common, dan yang lainnya dihubungkan ke NO atau NC tergantung pada apakah beban harus dihubungkan/diputus saat relai aktif.

2. POWER SUPPLY

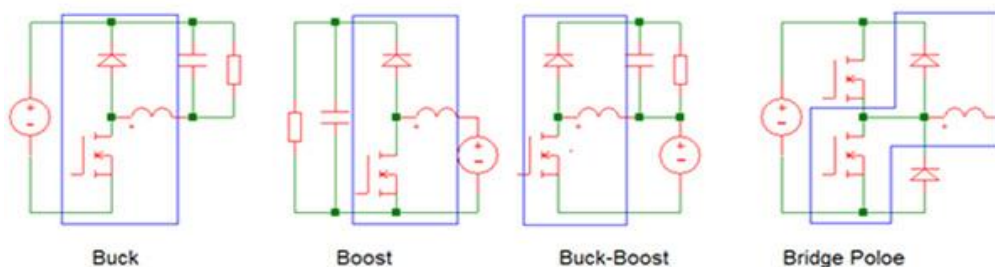


Rangkaian Catu Daya 5V merupakan rangkaian yang umum dan paling dibutuhkan untuk perangkat elektronik dan gerbang logika digital. Berikut ini adalah skema catu daya DC 5V untuk menyediakan tegangan keluaran DC yang andal dan stabil. Untuk merancang catu daya 5V, kita perlu mempelajari persyaratan dasar seperti sumber daya masukan yang tersedia, metode Stepdown catu daya dan rentang arus keluaran, kemudian metode penyaringan (Beban induktif memerlukan penyaringan yang tepat dan perlindungan terbalik). Jika Anda merancang adaptor 5V maka Anda harus menempatkan LED indikator untuk menunjukkan adanya pasokan keluar. Mengingat skema catu daya 5V dirancang untuk persyaratan dan aplikasi umum. Sebagian besar rangkaian dan perangkat elektronik atau mikrokontroler memerlukan rangkaian catu daya 5V DC. rangkaian catu daya 5V DC yang sederhana dan mudah dibuat dirancang dan diuji. Untuk menunjukkan keberadaan Tegangan Input dan Tegangan Output.



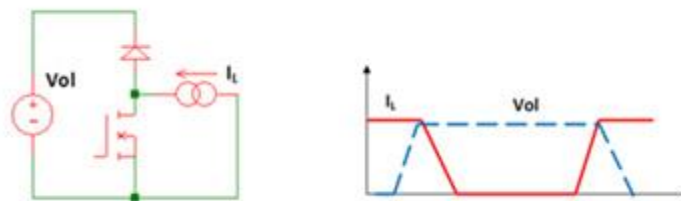
3. RC SNUBBER

Sakelar daya merupakan jantung dari setiap konverter daya. Pengoperasiannya akan secara langsung menentukan keandalan dan efisiensi produk. Untuk meningkatkan kinerja rangkaian sakelar konverter daya, peredam ditempatkan di seluruh sakelar daya untuk menekan lonjakan tegangan dan meredam bunyi dering yang disebabkan oleh induktansi rangkaian saat sakelar terbuka. Desain peredam yang tepat dapat menghasilkan keandalan yang lebih tinggi, efisiensi yang lebih tinggi, dan EMI yang lebih rendah. Di antara berbagai jenis peredam, peredam resistor-kapasitor (RC) merupakan rangkaian peredam yang paling populer. Artikel ini menjelaskan mengapa peredam diperlukan untuk sakelar daya. Beberapa kiat praktis untuk desain peredam yang optimal juga disediakan.



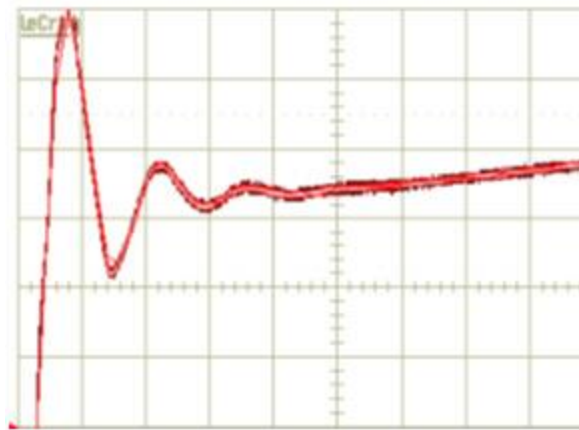
Gambar 1: Empat Rangkaian Pengalihan Daya

Ada banyak topologi berbeda yang digunakan dalam konverter daya, penggerak motor, dan ballast lampu. Gambar 1 menunjukkan empat rangkaian pengalihan daya dasar. Dalam keempat rangkaian dasar ini, dan dalam sebagian besar rangkaian pengalihan daya, jaringan sakelar-dioda-induktor yang sama ditunjukkan dalam garis biru. Perilaku jaringan ini sama di semua rangkaian ini. Oleh karena itu, rangkaian yang disederhanakan seperti yang ditunjukkan pada Gambar 2 dapat digunakan untuk analisis kinerja pengalihan untuk sakelar daya selama transien pengalihan. Karena arus dalam induktor hampir tidak berubah selama transien pengalihan, induktor diganti dengan sumber arus seperti yang ditunjukkan pada gambar. Bentuk gelombang tegangan dan pengalihan arus yang ideal dari rangkaian tersebut juga ditunjukkan pada Gambar 2



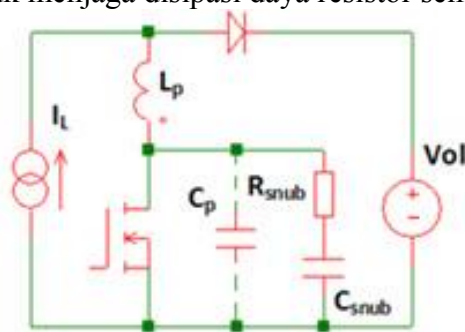
Gambar 2 : Rangkaian pengalihan daya yang disederhanakan dan bentuk gelombang pengalihan yang ideal.

Ketika sakelar MOSFET mati, tegangan yang melintasinya naik. Namun, arus I_L akan terus mengalir melalui MOSFET hingga tegangan sakelar mencapai V_{ol} . Arus I_L mulai turun setelah dioda menyala. Ketika sakelar MOSFET menyala, situasinya terbalik seperti yang ditunjukkan pada gambar. Jenis pengalihan ini disebut sebagai "pengalihan keras". Tegangan maksimum dan arus maksimum harus didukung secara bersamaan selama peralihan sementara. Oleh karena itu, "pengalihan keras" ini memaparkan sakelar MOSFET pada tekanan tinggi.



Gambar 3 : Tegangan berlebih pada sakelar MOSFET saat mematikan sementara.

Dalam sirkuit praktis, tegangan switching jauh lebih tinggi karena induktansi parasit (L_p) dan kapasitansi (C_p) seperti yang ditunjukkan pada Gambar 4. C_p mencakup kapasitansi keluaran sakelar dan kapasitansi liar karena tata letak dan pemasangan PCB. L_p mencakup induktansi parasit dari rute PCB dan induktansi kabel MOSFET. Induktansi parasit dan kapasitansi dari perangkat daya ini membentuk filter yang beresonansi tepat setelah transien mati-mati, dan karena itu menimpakan dering tegangan berlebihan ke perangkat seperti yang ditunjukkan pada Gambar 3. Untuk menekan tegangan puncak, peredam RC tipikal diterapkan di seluruh sakelar seperti yang ditunjukkan pada Gambar 4. Nilai resistor harus dekat dengan impedansi resonansi parasit yang dimaksudkan untuk diredam. Kapasitansi peredam harus lebih besar dari kapasitansi rangkaian resonansi, tetapi harus cukup kecil untuk menjaga disipasi daya resistor seminimal mungkin.



Gambar 4 : Konfigurasi snubber resistor-kapasitor.

Jika disipasi daya tidak kritis, ada pendekatan desain cepat untuk snubber RC. Secara empiris, pilih kapasitor snubber C_{snub} yang sama dengan dua kali jumlah kapasitansi keluaran sakelar dan kapasitansi pemasangan yang diestimasi.

Resistor snubber R_{snub} dipilih sehingga $R_{snub} = \frac{Vol}{I_L}$. Disipasi daya pada R_{snub} pada frekuensi pengalihan tertentu (f_s) dapat diestimasi sebagai:

$$P_{diss} = C_{snub} * Vol^2 * f_s$$

Snubber RC yang dioptimalkan : Dalam kasus-kasus di mana disipasi daya kritis, pendekatan desain yang lebih optimal harus digunakan. Pertama, ukur frekuensi dering (F_{ring}) pada simpul sakelar MOSFET (SW) saat mati. Solder kapasitor ESR rendah 100 pF jenis film melintasi MOSFET. Tingkatkan kapasitansi hingga frekuensi dering menjadi setengah dari nilai terukur asli. Sekarang total kapasitansi keluaran sakelar (kapasitansi tambahan plus kapasitansi parasit asli) ditingkatkan dengan faktor empat karena frekuensi dering berbanding terbalik dengan akar kuadrat dari produk kapasitansi induktansi rangkaian. Jadi kapasitansi parasit C_p adalah sepertiga dari nilai kapasitor yang ditambahkan secara eksternal. Induktansi parasit L_p sekarang dapat diperoleh dengan menggunakan persamaan berikut:

$$F_{ring} = \frac{1}{2\pi\sqrt{L_p * C_p}}$$

Setelah induktansi parasit L_p dan kapasitansi parasit C_p diketahui, resistor snubber

R_{snub} dan kapasitor C_{snub} dapat dipilih berdasarkan perhitungan berikut.

$$R_{snub} = \sqrt{L_p / C_p}$$

$$C_{snub} = 1 / (2\pi R_{snub} F_{ring})$$

Resistor snubber dapat diputar halus lebih lanjut untuk mengurangi dering jika ternyata tidak mencukupi. Disipasi daya pada R_{snub} pada frekuensi switching tertentu (f_s) adalah $C_{snub} Vol^2 * f_s$.

Dengan menggunakan semua nilai yang dihitung, desain untuk snubber sakelar catu daya selesai dan dapat diimplementasikan dalam aplikasi.

4. IR RECEIVER DAN IR TRANSMITTER



Komunikasi inframerah (IR) merupakan teknologi komunikasi yang banyak digunakan. Contoh umum komunikasi IR adalah remote control TV yang mengirimkan sinyal IR ke TV. Secara khusus, lampu IR pada remote tersebut dinyalakan dan dimatikan berulang kali (biasanya 38kHz / 38.000 kali per detik). Keunggulan teknologi IR adalah murah dan tidak terdeteksi oleh mata manusia. Kelemahannya adalah lampu IR harus diarahkan ke penerima IR. Akibatnya, komunikasi IR tidak boleh diterapkan jika pengirim dan penerima berada di ruangan yang berbeda (Bluetooth atau WiFi mungkin merupakan pilihan yang lebih baik).

IR TRANSMITTER

Modul pemancar IR memiliki LED IR untuk mengirim sinyal. Biasanya, LED beroperasi sekitar 940nm, yaitu LED sangat terang dalam spektrum IR. Selain LED, modul memiliki dua pin untuk memberi daya dan mengendalikan modul: GND, VCC, Dan itu pada dasarnya semua yang dibutuhkan untuk mengirim sinyal melalui komunikasi IR. Faktanya, inilah alasan mengapa komunikasi IR sangat murah. Anda hanya memerlukan LED (IR) untuk memulai.

Pin GND dan VCC digunakan untuk memberi daya pada modul dengan catu daya 5V (misalnya Arduino GND dan header pin 5V).

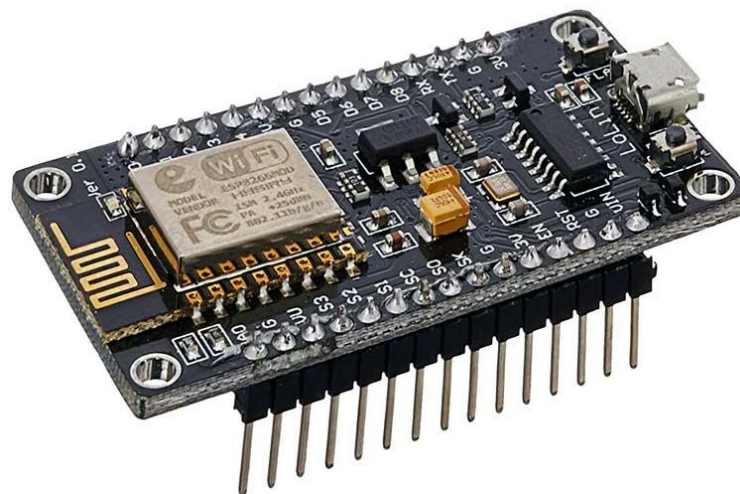
IR RECEIVER

Modul penerima *IR* biasanya memiliki unit penerima *IR* terintegrasi. Tata letak pin sangat mirip dengan *IR Transmitter*. Ada pin GND dan VCC untuk memberi daya pada modul dengan pasokan tegangan 5V. Data *IR* yang diterima dapat dibaca dari pin DAT. Tata letak pin pada dasarnya juga sama untuk modul penerima IR lainnya. Pin DAT sering kali memiliki nama yang berbeda.

PEMBAHASAN

Ide dari contoh aplikasi ini adalah memiliki modul penerima IR dan modul pemancar IR. Setiap modul dikontrol oleh Arduino Uno (=dua Arduino berbeda digunakan). Arduino Uno pengirim menggunakan "protokol NEC" untuk mengirim pesan dengan modul pemancar IR. Arduino Uno penerima menggunakan modul penerima IR untuk menerima pesan dan kemudian mencetak isi pesan. Kabel untuk memberi daya sama untuk kedua pasang Arduino dan modul. Kedua modul memiliki tegangan operasi 5V. Untuk setiap modul, gunakan kabel jumper dan hubungkan pin GND Arduino ke pin GND modul. Lakukan hal yang sama untuk setiap modul untuk menghubungkan pin 5V Arduino ke pin VCC modul.

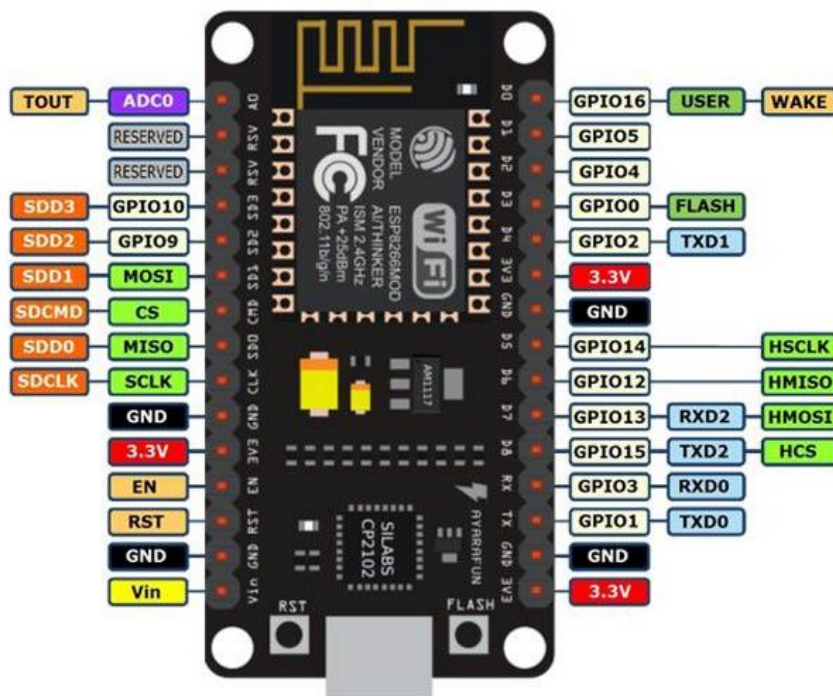
5. NODE MCU ESP8266



NodeMCU adalah **firmware dan papan pengembangan** berbasis Lua sumber terbuka yang secara khusus ditujukan untuk Aplikasi berbasis IoT. NodeMCU mencakup firmware yang berjalan pada ESP8266 Wi-Fi SoC dari Espressif Systems, dan perangkat keras yang berbasis pada modul ESP-12.

KATEGORI	SPEKIFIKASI
Tegangan Antarmuka Komunikasi	3.3V
Kecepatan Transfer	110-460800 bps
Ukuran Board	57mm x 30mm
Gpio	13 PIN
Flash Memory	4 MB
Wifi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 Ghz
Usb Port	Micro USB
USB To Serial Converter	CH340G

Papan pengembangan NodeMCU *ESP8266* dilengkapi dengan modul ESP-12E yang berisi chip *ESP8266* yang memiliki mikroprosesor Tensilica Xtensa 32-bit LX106 RISC. Mikroprosesor ini mendukung RTOS dan beroperasi pada frekuensi clock yang dapat disesuaikan dari 80MHz hingga 160 MHz. NodeMCU memiliki RAM 128 KB dan memori Flash 4MB untuk menyimpan data dan program. Daya pemrosesannya yang tinggi dengan Wi-Fi/Bluetooth internal dan fitur Deep Sleep Operating membuatnya ideal untuk proyek IoT. NodeMCU dapat diberi daya menggunakan jack Micro USB dan pin VIN (External Supply Pin). Mendukung antarmuka UART, SPI, dan I2C.



Keterangan:

1. Micro-USB : Pasti semuanya sudah tau bagian ini ya. Fungsinya sebagai power yang dapat terhubung dengan USB port. Selain itu, biasanya juga digunakan untuk melakukan pengiriman sketch atau memantau data serial dengan serial monitor di aplikasi Arduino IDE.
2. 3.3V : Digunakan sebagai tegangan untuk device lainnya. ada 3 tempat untuk 3.3v. Biasanya juga dituliskan hanya 3V (Sebenarnya tetap 3,3V)
3. GND : Ground. Sebagai tegangan 0 atau nilai negatif untuk mengalirkan arus.
4. Vin : Sebagai External Power yang akan mempengaruhi Output dari seluruh pin. Cara menggunakannya yaitu dengan menghubungkannya dengan tegangan 7 hingga 12volt.
5. EN, RST : Pin yang digunakan untuk reset program di mikrokontroler.
6. A0 : Analog pin, digunakan untuk membaca input secara analog.
7. GPIO 1 – GPIO 16 : Pin yang dapat digunakan sebagai input dan output. Pin ini dapat melakukan pembacaan dan pengiriman data secara analog juga.
8. SD1,CMD, SD0,CLK : SPI Pin untuk komunikasi SPI (Serial Peripheral Interface) dimana kita akan menggunakan clock untuk sinkronisasi deteksi bit pada receiver.
9. TXD0, RXD0,TXD2,RXD2 : Sebagai interface UART, Pasangannya adalah TXD0 dengan RXD0 dan TXD2 dengan RXD2. TXD1 digunakan untuk upload firmware/program.
10. SDA, SCL (I2C Pins) : Digunakan untuk device yang membutuhkan I2C.