

INTERACTIVE ROBOTICS

WITH TECHNOLOGY, GET FUTURE MORE BETTER

MODULE ELEKTRONIKA INTERMEDIATE

**Module ini dibuat Oleh Team Interactive Robotics untuk digunakan
pada pelatihan internal di PT. Delameta Bilano**

WORKSHOP

**Jl. Sholeh Iskandar No.3C RT.09/01
Kedung badak, Tanah Sareal ,
Kota Bogor 16164**



+6287775593898



interactive.robotics



roboticsinteractive@gmail.com



www.interactiverobotics.club

SILABUS ELEKTRONIKA INTERMEDIATE

1	Pengenalan Microcontroller <ul style="list-style-type: none"> ✓ Arduino ✓ ESP8266 / ESP-01 ✓ STM32 	Memahami macam-macam microcontroller yang dapat digunakan dalam proyek pembuatan alat-alat robotika dan IOT	Dapat menentukan microcontroller yang cocok untuk digunakan pada project yang akan diangun	Teori
2	ADC (Analog Digital Converter) Pembacaan Nilai Analog dan Digital pada Microcontroller <ul style="list-style-type: none"> ✓ Button (switch dan On/Off) ✓ Infrared Proximity (Button tanpa sentuh) ✓ Sensor LDR (Pendeteksi Cahaya) ✓ Sensor Flame (Pendeteksi Api) ✓ Sensor PIR (Pendeteksi Gerak) 	Mengerti penggunaan dari macam-macam sensor analog dan digital berikut cara merangkai dan memprogramnya	Dapat memanfaatkan sensor-sensor jenis analog dan digital dalam project elektronika yang ingin dibangun	Teori dan Praktikum
3	Mempelajari sensor-sensor Khusus <ul style="list-style-type: none"> ✓ Temperature ✓ Ultrasonic ✓ RFID ✓ GPS 	Mengerti penggunaan dari macam-macam sensor khusus dan berikut cara merangkai dan memprogramnya, serta memahami jenis protocol yang digunakan dalam mengakses sensor seperti (SPI dan UART)	Dapat memanfaatkan sensor-sensor jenis khusus dengan memanfaatkan protocol (SPI dan UART) dalam pembuatan project elektronika yang ingin dibangun	Teori dan Praktikum
4	Menghubungkan ke Aktuator dan Display <ul style="list-style-type: none"> ✓ Relay ✓ Servo ✓ Motor Driver + Motor DC ✓ LCD I2C 20x4 	Mengerti penggunaan macam-macam aktuator sebagai penggerak objek tertentu serta membuat output tampilan sehingga bisa digunakan untuk memantau data sensor dan mempelajari fungsi protocol I2C pada LCD	Dapat memanfaatkan macam-macam aktuator serta dapat menganalisa fungsionalitas dari jenis aktuator pada project yang akan dibangun.	Teori dan Praktikum
5	Otomatisasi Sensor dan Aktuator <ul style="list-style-type: none"> ✓ Temperature + Relay + LCD ✓ Ultrasonic + Servo + LCD ✓ RFID + Servo/Relay + LCD 	Mengerti penggunaan sensor dan aktuator serta display pada mini project otomatisasi	Dapat membangun system otomatisasi yang berjalan sendiri dalam mini project yang dibangun	Teori dan Praktikum
6	Test Evaluasi (Teori dan Praktikum)			

Pengenalan Microcontroller

1. Arduino
2. ESP8266 / ESP-01
3. STM32



Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu chip IC, sehingga sering disebut single chip microcomputer. Mikrokontroler merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC (Personal Computer) yang memiliki beragam fungsi. Perbedaan lainnya adalah perbandingan RAM dan ROM yang sangat berbeda antara komputer dengan mikrokontroler.

Mikrokontroler adalah sebuah system microprocessor dimana didalamnya sudah terdapat CPU, ROM, RAM, I/O, Clock dan peralatan internal lainnya yang sudah saling terhubung dan terorganisasi (teralamati) dengan baik oleh pabrik pembuatnya dan dikemas dalam satu chip yang siap pakai. Sehingga kita tinggal memprogram isi ROM sesuai aturan penggunaan oleh pabrik yang membuatnya menurut Winoto (2008:3).

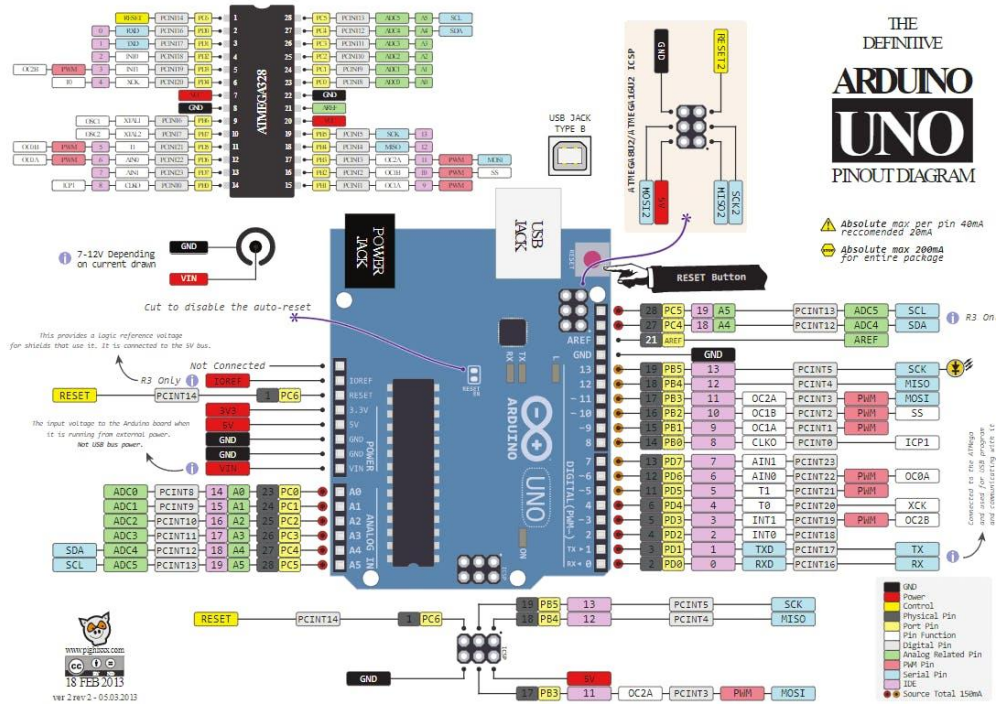
1. ARDUINO

Arduino adalah mikrokontroler / pengendali mikro papan tunggal (*single board*) yang bersifat **Open source** dan menjadi salah satu proyek **Open Source Hardware** yang paling populer. Dirancang untuk memudahkan pembuatan alat-alat elektronik interaktif berdasarkan hardware dan software yang fleksibel dan mudah digunakan dalam berbagai bidang. Perangkat **Hardware** memiliki prosesor **Atmel AVR**. Softwaranya terdiri dari beberapa *fitur* yakni **Integrated Development Environment (IDE)**, **Text-Editor**, **Compiler**, **Serial Monitor**, dan **Serial ISP Programmer**. Pada aplikasi Arduino, **Mikrokontroler** diprogram menggunakan bahasa pemrograman **C** atau **C++**, dengan **library** khas arduino. Karena sifatnya yang **Open Source** maka siapa saja dapat mengunduh skema hardware arduino dan membangunnya



Kelebihan Microcontroller Arduino:

- ✓ Harga terjangkau
- ✓ Dapat dijalankan diberbagai sistem operasi, Windows, Linux, Mac, dan sebagainya.
- ✓ Sederhana, dengan bahasa pemrograman yang mudah bisa dipelajari orang awam, bukan untuk orang teknik saja.
- ✓ Open Source, hardware maupun software.



2. ESP8266 / ESP-01

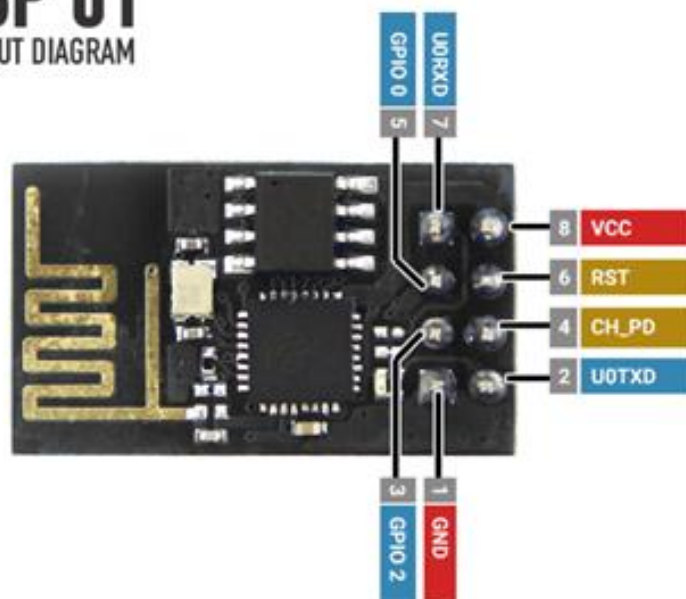
ESP8266 / ESP-01 merupakan modul Wi-Fi yang memungkinkan akses mikrokontroler ke jaringan Wi-Fi. merupakan tipe SOC mandiri (System On a Chip) yang tidak memerlukan mikrokontroler untuk memanipulasi input dan karena ESP-01 bertindak sebagai komputer kecil

Dengan demikian, kita bisa memberi akses internet mikrokontroler seperti perisai Wi-Fi ke Arduino, atau kita hanya bisa memprogram ESP8266 untuk tidak hanya memiliki akses ke jaringan Wi-Fi, tapi juga berfungsi sebagai mikrokontroler.



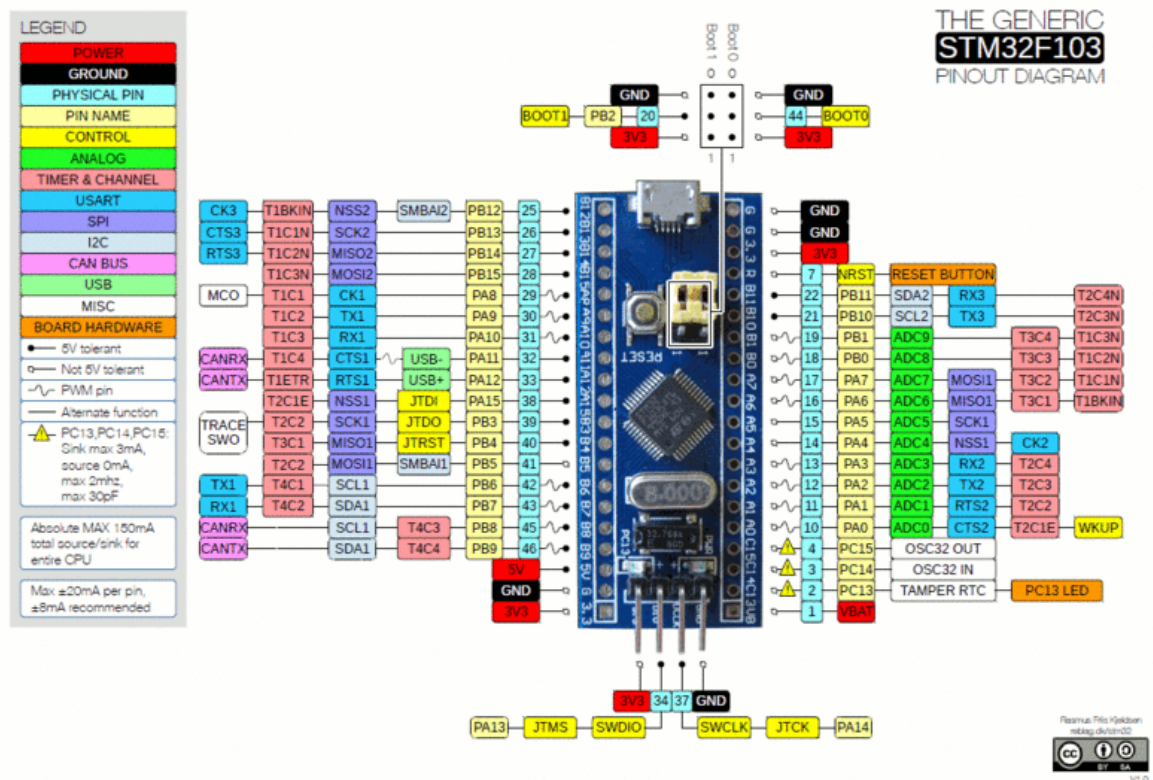
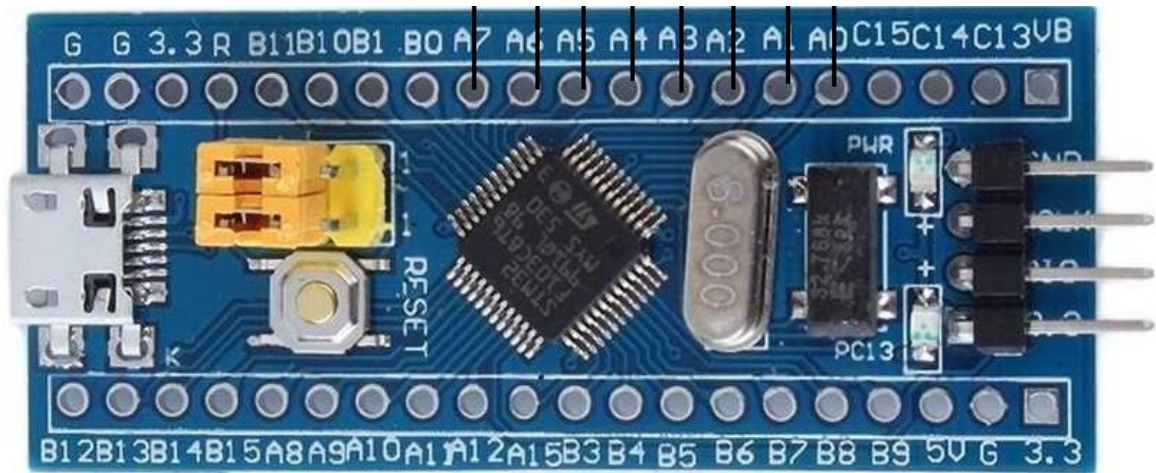
Modul ini
output

ESP 01 PINOUT DIAGRAM



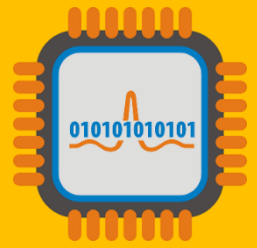
3. STM32

STM32 adalah mikrokontroler berbasis inti prosesor 32 bit RISC ARM Cortex-M7, Cortex-M4F, Cortex-M3, Cortex-M0+, dan Cortex-M0 dari STMicroelectronics. Mikrokontroler ini mempunyai frekuensi clock tinggi, umumnya berada pada kisaran 72MHz atau lebih.



ADC (Analog Digital Converter) Pembacaan Nilai Analog dan Digital pada Microcontroller

1. Button (switch dan On/Off)
2. Infrared Proximity (Button tanpa sentuh)
3. Sensor LDR (Pendeteksi Cahaya)
4. Sensor Flame (Pendeteksi Api)
5. Sensor PIR (Pendeteksi Gerak)



1. Button (Push Switch dan Toggle On/Off)

Push button switch (saklar tombol tekan) adalah perangkat / sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran listrik dengan sistem kerja tekan unlock (tidak mengunci). Sistem kerja disini berarti saklar akan bekerja sebagai device penghubung atau aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan maka saklar akan kembali pada kondisi normal.

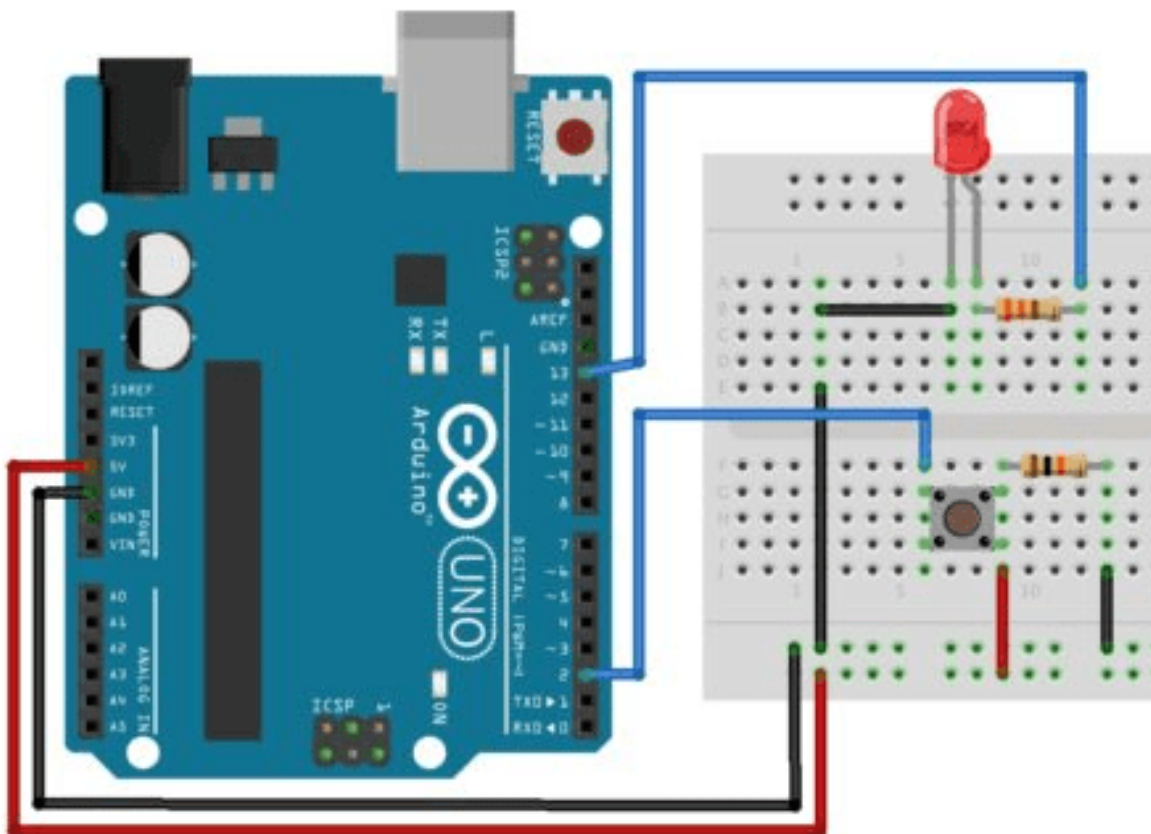


saklar
arus
unlock
pemutus
(dilepas),



Toggle ON / OFF adalah perantara untuk memutuskan atau menyambungkan tegangan pada sebuah rangkaian, sehingga rangkaian akan mulai bekerja bila saklar dalam posisi ON, sedangkan ketika OFF maka rangkaian akan berhenti terhubung.

Merakit Push Button dengan Arduino Uno



Source Code Push button

```
//Inisialisasi pembuatan variable
int tombol1 =2;
int led1     =13;

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  pinMode(tombol1,INPUT);
  pinMode(led1,OUTPUT);
}

void loop() {
  //Program yang akan dijalankan berulang-ulang
  int bacaTombol1=digitalRead(tombol1); //Membaca keadaan Tombol

  //Print ke Serial Monitor
  Serial.print("Kondisi tombol:");
  Serial.print(bacaTombol1);
  Serial.println();

  //Logika Program (Jika tombol membaca Nilai 1 maka led menyala, jika bukan 1 maka mati
  if(bacaTombol1==1){
    digitalWrite(led1,HIGH);
  }else{
    digitalWrite(led1,LOW);
  }
  delay(500); //Jeda waktu perulangan seama 500 mili detik
}
```

Chalange

1. **Rangkailah 3 Button dan 3 Led** selanjutnya program agar masing-masing led bisa menyala sesuai button yang tekan
2. Buatlah program untuk menyalakan led1 dan led2 dengan 1 Tombol
3. Buatlah program untuk menyalakan led1 dengan tombol1 **atau** tombol2 (fungsi or)
4. Buatlah program untuk menyalakan led1 dengan tombol1 **dan** tombol2 (fungsi and)

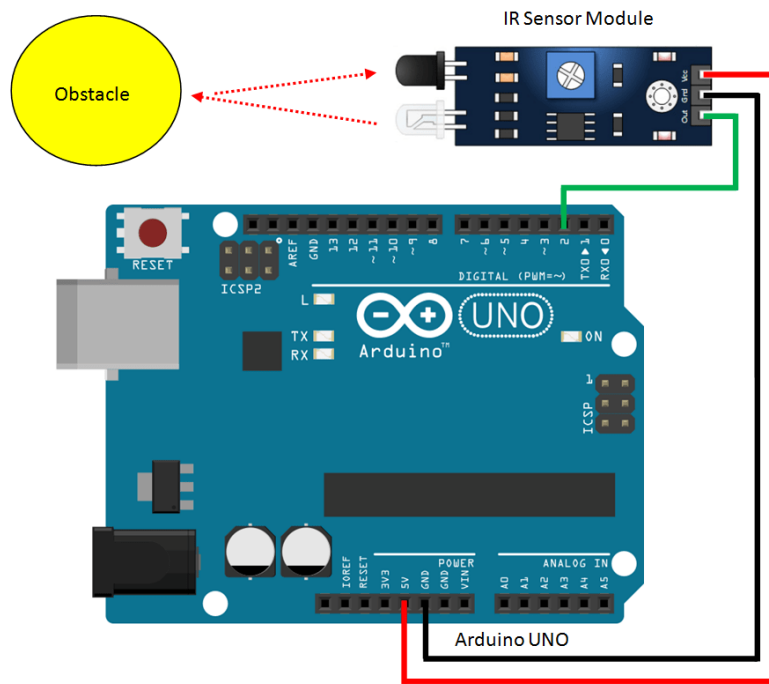
2. Infrared Proximity (Button tanpa sentuh)

Proximity Sensor (Sensor Proksimitas)

atau dalam bahasa Indonesia disebut dengan Sensor Jarak adalah sensor elektronik yang mampu mendeteksi keberadaan objek di sekitarnya tanpa adanya sentuhan fisik. Dapat juga dikatakan bahwa Sensor Proximity adalah perangkat yang dapat mengubah informasi tentang gerakan atau keberadaan objek menjadi sinyal listrik.



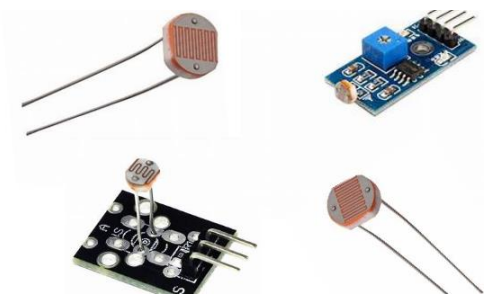
Merakit Infrared proximity pada arduino



Source Code

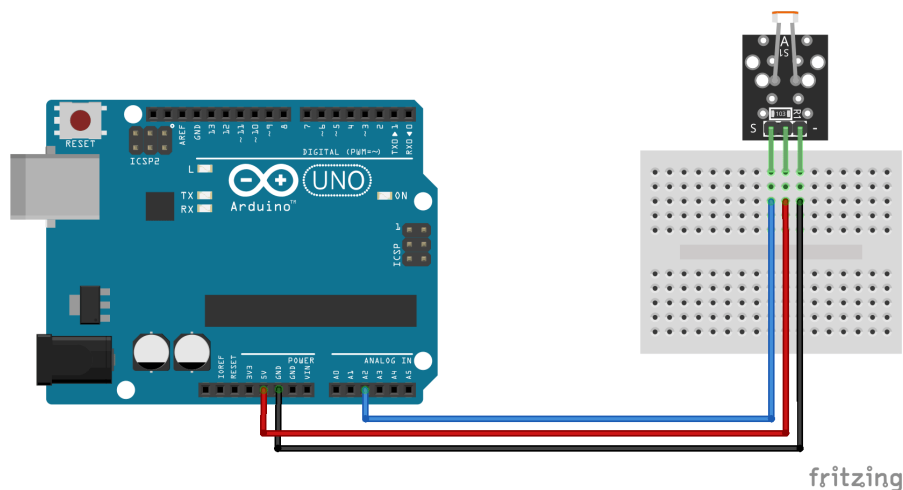
Gunakan source code yang sama dengan materi 1, karena sistem kerja dari proximity sensor ini sama persis dengan fungsi tombol push button pada materi 1.

3. Sensor LDR (Pendeteksi Cahaya)



LDR (Light Dependent Resistor) merupakan salah satu komponen resistor yang nilai resistansinya akan berubah-ubah sesuai dengan intensitas cahaya yang mengenai sensor ini. LDR juga dapat digunakan sebagai sensor cahaya. Perlu diketahui bahwa nilai resistansi dari sensor ini sangat bergantung pada intensitas cahaya. Semakin banyak cahaya yang mengenainya, maka akan semakin menurun nilai resistansinya. Sebaliknya jika semakin sedikit cahaya yang mengenai sensor (gelap), maka nilai hambatannya akan menjadi semakin besar sehingga arus listrik yang mengalir akan terhambat.

Merakit sensor LDR dengan Arduino



Source Code

```
//Inisialisasi pembuatan variable
int sensorLDR =A0;

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  pinMode(sensorLDR,INPUT);
}

void loop() {
  //Program yang akan dijalankan berulang-ulang
  int bacaLDR=analogRead(sensorLDR); //Membaca Nilai LDR

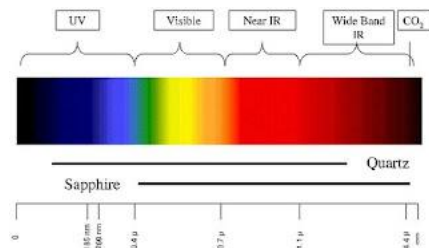
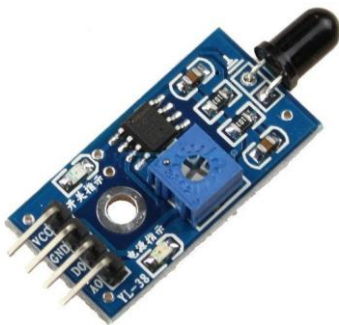
  //Print ke Serial Monitor
  Serial.print("Nilai LDR:");
  Serial.print(bacaLDR);
  Serial.println();
  delay(500); //Jeda waktu perulangan seama 500 mili detik
}
```

Chalange

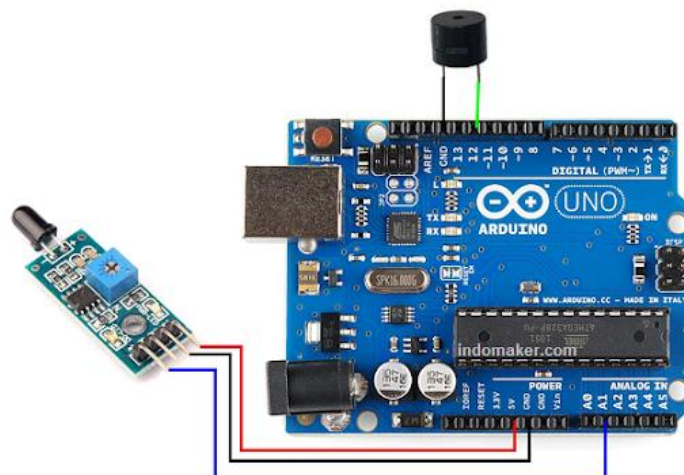
1. Buatlah sebuah rangkaian LED dan program dengan logika jika sensor LDR membaca cahaya terang maka LED mati dan jika gelap maka LED menyala
2. Buatlah program dengan memanfaatkan fungsi **PWM** untuk menentukan redup dan terangnya LED berdasarkan cahaya dari sensor LDR

4. Sensor Flame (Pendeteksi Api)

Flame detector merupakan salah satu alat instrument berupa sensor yang dapat mendeteksi nilai intensitas dan frekuensi api dengan panjang gelombang antara 760 nm ~ 1100 nm



Merakit sensor



Source Code

```
//Inisialisasi pembuatan variable
int sensorFlame =A1;

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  pinMode(sensorFlame,INPUT);
}

void loop() {
  //Program yang akan dijalankan berulang-ulang
  int bacaFlame=analogRead(sensorFlame); //Membaca Nilai Flame

  //Print ke Serial Monitor
  Serial.print("Nilai Flame:");
  Serial.print(bacaFlame);
  Serial.println();
  delay(500); //Jeda waktu perulangan seama 500 mili detik
}
```

Chalange

1. Tambahkan Buzzer pada rangkaian di atas dan buatlah program, Jika flame sensor mendeteksi adanya api degan nilai tertentu maka buzzer berbunyi
2. Buatlah 3 Logika pada program diatas dengan kondisi berikut:
 - ✓ Jika tidak ada api maka tampilkan diserial monitor “Aman” dan Buzzer mati
 - ✓ Jika api terdeteksi kecil tampilkan diserial monitor “Warning Silahkan periksa” dan Buzzer berbunyi lambat
 - ✓ Jika api terdeteksi besar tampilkan diserial monitor “Waspada harap mengungsi” dan Buzzer berbunyi cepat

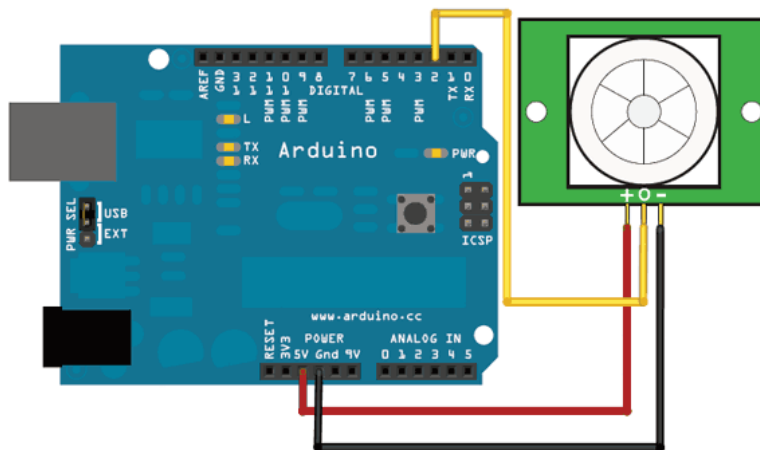
5. Sensor PIR (Pendeteksi Gerak)



Sensor PIR atau disebut juga dengan Passive Infra Red merupakan sensor yang digunakan untuk mendeteksi adanya pancaran sinar infra merah dari suatu object. Sesuai dengan namanya sensor PIR bersifat pasif, yang berarti sensor ini tidak memancarkan sinar infra merah melainkan hanya dapat menerima radiasi sinar infra merah dari luar. sebagai contoh ketika terdeteksi sebuah gerakan dari sumber infra merah dengan suhu tertentu yaitu manusia mencoba melewati sumber infra merah yang lain misal dinding, maka sensor akan membandingkan pancaran infra merah yang diterima setiap satuan waktu, sehingga jika ada pergerakan maka akan

terjadi perubahan pembacaan pada sensor.

Rangkaian Sensor PIR dengan Arduino



Source Code

```
//Inisialisasi pembuatan variable
int sensorPIR =2;

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  pinMode(sensorPIR,INPUT);
}

void loop() {
  //Program yang akan dijalankan berulang-ulang
  int bacaPIR=digitalRead(sensorPIR); //Membaca Nilai Sensor PIR

  //Print ke Serial Monitor
  Serial.print("Sensor PIR:");
  Serial.print(bacaPIR);
  Serial.println();
  delay(500); //Jeda waktu perulangan seama 500 mili detik
}
```

Chalange

1. Tambahkan Buzzer pada rangkaian di atas dan buatlah program, Jika PIR sensor mendeteksi adanya pergerakan maka buzzer berbunyi
2. Gabungkan ke 3 sensor di atas dan taampilkan semua nilai sensor pada serial monitor serta buatlah logika untuk masing-masing fungsi sensor dengan output Buzzer

Mempelajari sensor-sensor Khusus

1. Temperature
2. Ultrasonic
3. RFID
4. GPS

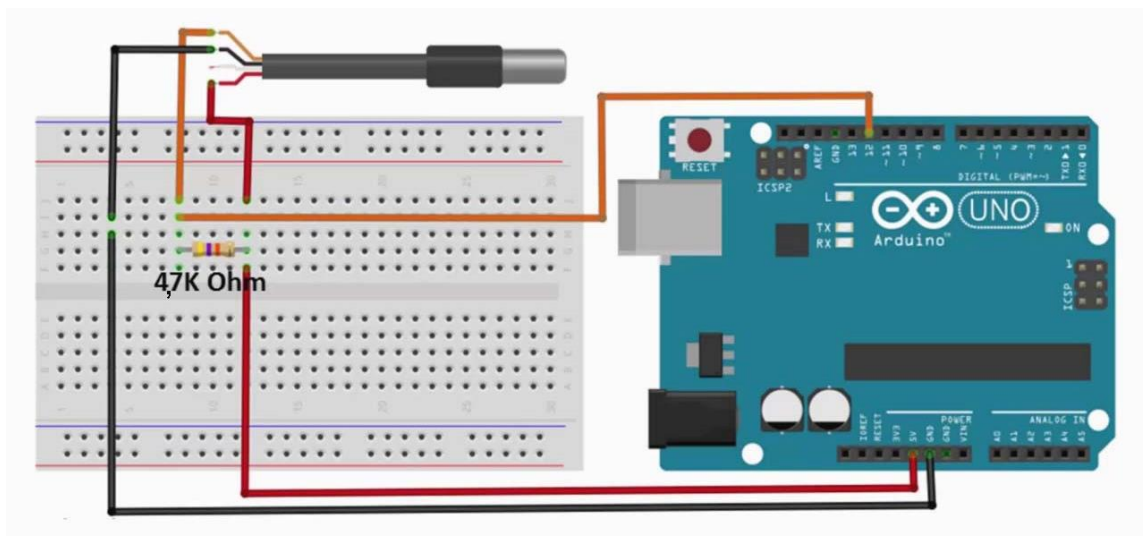


1. Temperature DS18B20

DS18B20 adalah sebuah sensor suhu digital *one wire* atau hanya membutuhkan 1 pin jalur data komunikasi. Setiap sensor DS18B20 memiliki nomor seri 64-bit yang unik yang berarti kita dapat menggunakan banyak sensor pada bus daya yang sama (banyak sensor terhubung ke GPIO yang sama). Hal tersebut sangat berguna untuk logging data pada proyek pengontrolan suhu. DS18B20 adalah sensor yang bagus karena murah, akurat, dan sangat mudah digunakan.



Rangkain Sensor DS18B20 arduino



Source Code

```
//Download dan masukan Library Onewire dan DallasTemperature
#include <OneWire.h>
#include <DallasTemperature.h>

//Inisialisasi pembuatan variable
int sensorSuhu =2;
OneWire oneWire(sensorSuhu);
DallasTemperature sensors(&oneWire);

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  sensors.begin(); //Memanggil fungsi
}

void loop() {
  //Program yang akan dijalankan berulang-ulang
  sensors.requestTemperatures();
  float bacaSuhu = sensors.getTempCByIndex(0); //Membaca data di index 0 pada one wire
}
```



```

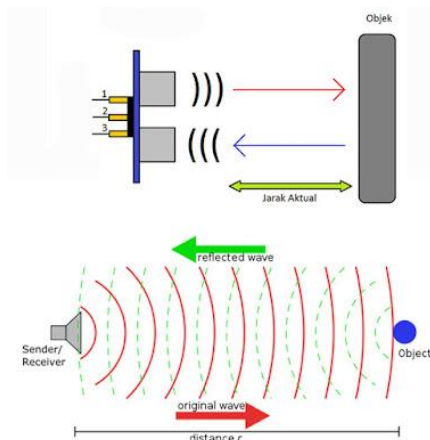
if(bacaSuhu != DEVICE_DISCONNECTED_C)
{
  Serial.print("Data suhu yang terbaca pada bus 1 (index 0): ");
  Serial.println(bacaSuhu);
}
else
{
  Serial.println("Error: sensor pada index 0 tidak ditemukan!");
}
}

```

Chalange

1. Buatlah rangkaian tambahan untuk buzzer dan buatlah logika, jika suhu melebihi batas tertentu maka buzzer berbunyi.

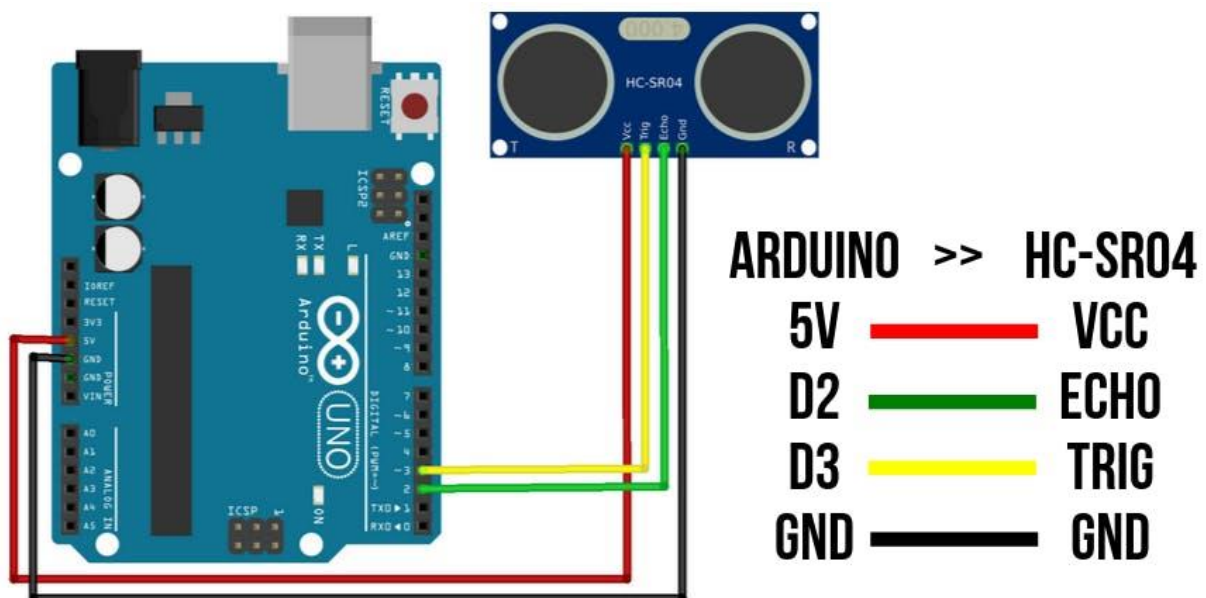
2. Ultrasonic



Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik).

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.

Rangkaian Sensor Ultrasonic pada arduino



Source Code

```
//Download dan masukan Library NewPing
#include <NewPing.h>

//Inisialisasi pembuatan variable
int triger =2;
int echo =3;
int batas =200; //Maksimal 400 cm

NewPing cm(triger,echo,batas);

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
}

void loop() {
  //Program yang akan dijalankan berulang-ulang
  int bacaJarak=cm.ping_cm();

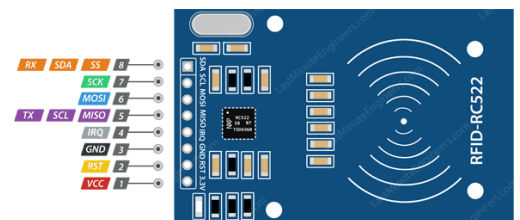
  //Print ke Serial Monitor
  Serial.print("Data jarak:");
  Serial.print(bacaJarak);
  Serial.print(" cm");
  Serial.println();
  delay(500); //Jeda waktu perulangan seama 500 mili detik
}
```

Challenge

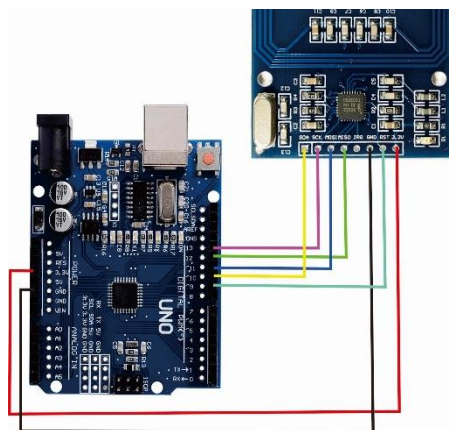
1. Tambahkan Buzzer pada project diatas dan buatlah logika jika jarak kurang dari 20 cm maka buzzer berbunyi lambat, dan selanjutnya ketika semakin dekat dengan object maka bunyi buzzer semakin cepat

3. RFID

RFID atau bisa disebut juga Radio Frequency Identification adalah sistem identifikasi berbasis wireless yang memungkinkan pengambilan data tanpa harus bersentuhan seperti barcode atau magnetic card. alat ini menggunakan sistem radiasi elektromagnetik untuk mengirimkan kode. RFID menggunakan sistem identifikasi dengan gelombang radio, karena itu minimal dibutuhkan dua buah perangkat agar alat ini dapat berfungsi, adapun perangkat yang dibutuhkan disebut TAG dan READER.



Rangkaian Sensor ultrasonic dan arduino



Arduino	RFID-RC522
SDA	10
SCK	13
MOSI	11
MISO	12
GND	GND
RST	9
3.3V	3.3V

Source Code

```
//Download dan masukan Library MFRC522
#include <SPI.h>
#include <MFRC522.h>

//Inisialisasi pembuatan variable
#define SS_PIN 10
#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  SPI.begin(); // Initiate SPI bus
  mfrc522.PCD_Init(); // Initiate MFRC522
}

void loop() {
  //Program yang akan dijalankan berulang-ulang
  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) == "6B A7 EF 0D") //Sesuaikan dengan ID kartu anda
  {
    Serial.println("Selamat datang iwan cilibur!!!");
    delay(2000);
  }else{
    Serial.println("Kartu Belum Terdaftar!");
  }
  Serial.println("Tempelkan Kartu!");
  delay(500); //Jeda waktu perulangan seama 500 mili detik
}
```

Chalange

1. Tambahkan Buzzer pada program diatas dan buatlah logika, jika kartu yang digunakan benar maka bunti buzzer beep 2 x dan jika salah maka bunyi beeb pajang 1 x
2. Tambahkan program untuk 2 kartu berhasil berhasil

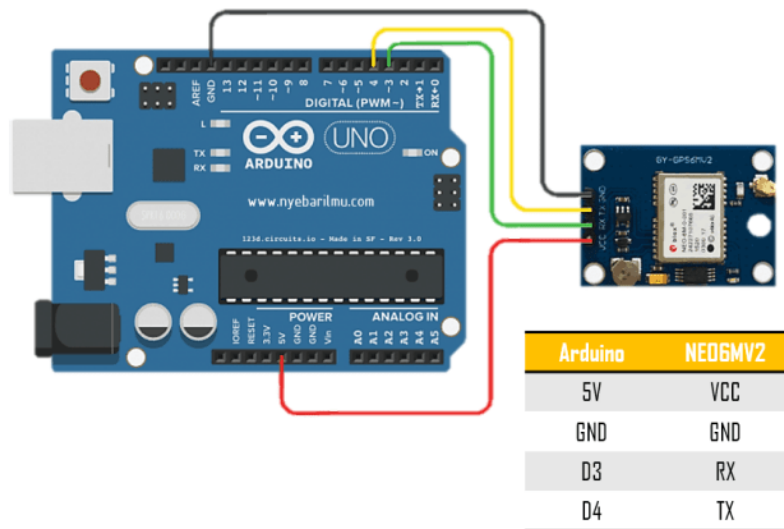
4. GPS



Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang terdiri dari setidaknya 24 satelit. GPS berfungsi dalam segala kondisi cuaca, di mana pun di dunia, 24 jam sehari, tanpa biaya berlangganan atau biaya penyiapan. Departemen Pertahanan AS (USDOD) awalnya menempatkan satelit ke orbit untuk penggunaan militer, tetapi mereka dibuat tersedia untuk digunakan sipil pada 1980-an.

Dasar pada sebuah sistem navigasi posisi dan perpindahan pada suatu objek, salah satu sensor yang dapat digunakan pada sistem navigasi adalah sensor GPS. Sensor GPS dapat menerima sinyal yang dipancarkan oleh satelit kemudian merubahnya menjadi suatu titik lokasi yang memiliki nilai longitude, latitude dan altitude

Rangkaian GPS dengan Arduino



Referensi Coding untuk mengambil data lain pada sensor GPS

```
1 Serial.println(gps.location.lat(), 6); // Latitude in degrees (double)
2 Serial.println(gps.location.lng(), 6); // Longitude in degrees (double)
3 Serial.print(gps.location.rawLat().negative ? "-" : "+");
4 Serial.println(gps.location.rawLat().deg); // Raw latitude in whole degrees
5 Serial.println(gps.location.rawLat().billionths); // ... and billionths (u16/u3:
6 Serial.print(gps.location.rawLng().negative ? "-" : "+");
7 Serial.println(gps.location.rawLng().deg); // Raw longitude in whole degrees
8 Serial.println(gps.location.rawLng().billionths); // ... and billionths (u16/u3:
9 Serial.println(gps.date.value()); // Raw date in DDMMYY format (u32)
10 Serial.println(gps.date.year()); // Year (2000+) (u16)
11 Serial.println(gps.date.month()); // Month (1-12) (u8)
12 Serial.println(gps.date.day()); // Day (1-31) (u8)
13 Serial.println(gps.time.value()); // Raw time in HHMMSSCC format (u32)
14 Serial.println(gps.time.hour()); // Hour (0-23) (u8)
15 Serial.println(gps.time.minute()); // Minute (0-59) (u8)
16 Serial.println(gps.time.second()); // Second (0-59) (u8)
17 Serial.println(gps.time.centisecond()); // 100ths of a second (0-99) (u8)
18 Serial.println(gps.speed.value()); // Raw speed in 100ths of a knot (i32)
19 Serial.println(gps.speed.knots()); // Speed in knots (double)
20 Serial.println(gps.speed.mph()); // Speed in miles per hour (double)
21 Serial.println(gps.speed.mps()); // Speed in meters per second (double)
22 Serial.println(gps.speed.kmph()); // Speed in kilometers per hour (double)
23 Serial.println(gps.course.value()); // Raw course in 100ths of a degree (i32)
24 Serial.println(gps.course.deg()); // Course in degrees (double)
25 Serial.println(gps.altitude.value()); // Raw altitude in centimeters (i32)
26 Serial.println(gps.altitude.meters()); // Altitude in meters (double)
27 Serial.println(gps.altitude.miles()); // Altitude in miles (double)
28 Serial.println(gps.altitude.kilometers()); // Altitude in kilometers (double)
29 Serial.println(gps.altitude.feet()); // Altitude in feet (double)
30 Serial.println(gps.satellites.value()); // Number of satellites in use (u32)
31 Serial.println(gps.hdop.value()); // Horizontal Dim. of Precision (100ths-i32)
```


Source Code

```
//Download dan masukan Library TinyGPS++
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

//Inisialisasi pembuatan variable
SoftwareSerial serial_gps(3, 4); //Komunikasi UART ke GPS
TinyGPSPlus gps;
String latitude, longitude;

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  serial_gps.begin(9600);
  Serial.println("GPS Mulai");
}

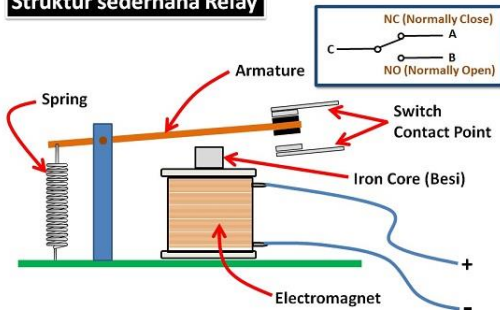
void loop() {
  //Program yang akan dijalankan berulang-ulang
  while(serial_gps.available()) {
    gps.encode(serial_gps.read());
  }
  if(gps.location.isUpdated()) {
    latitude = String(gps.location.lat(),6);
    longitude = String(gps.location.lng(),6);
    String link = "www.google.com/maps/place/" + String(latitude) + "," + String(longitude) ;
    Serial.println(link);
    // delay(2000);
  }
  delay(500); //Jeda waktu perulangan seama 500 mili detik
}
```

Menghubungkan ke Akuator dan Display

1. Relay
2. Servo
3. Motor Driver + Motor DC
4. LCD I2C 20x4

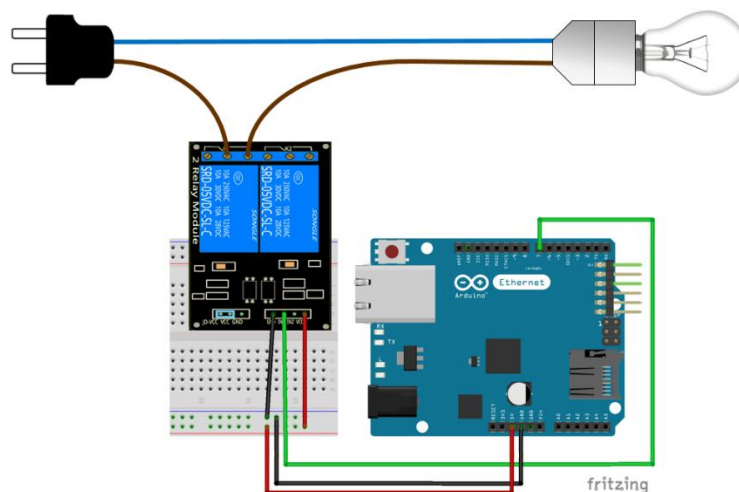
1. Relay

Struktur sederhana Relay



Relay adalah Saklar (Switch) yang dioperasikan secara listrik dan merupakan komponen Electromechanical (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/Switch). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (low power) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan Relay yang menggunakan Elektromagnet 5V dan 50 mA mampu menggerakkan Armature Relay (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.

Rangkaian Relay dan arduino



Source Code

```
//Inisialisasi pembuatan variable
int relay1=7;

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  pinMode(relay1,OUTPUT);
}

void loop() {
  //Program yang akan dijalankan berulang-ulang
  digitalWrite (relay1,HIGH);
  delay(500); //Jeda waktu perulagan seama 500 mili detik
  digitalWrite (relay1,LOW);
  delay(500); //Jeda waktu perulagan seama 500 mili detik
}
```

2. Servo

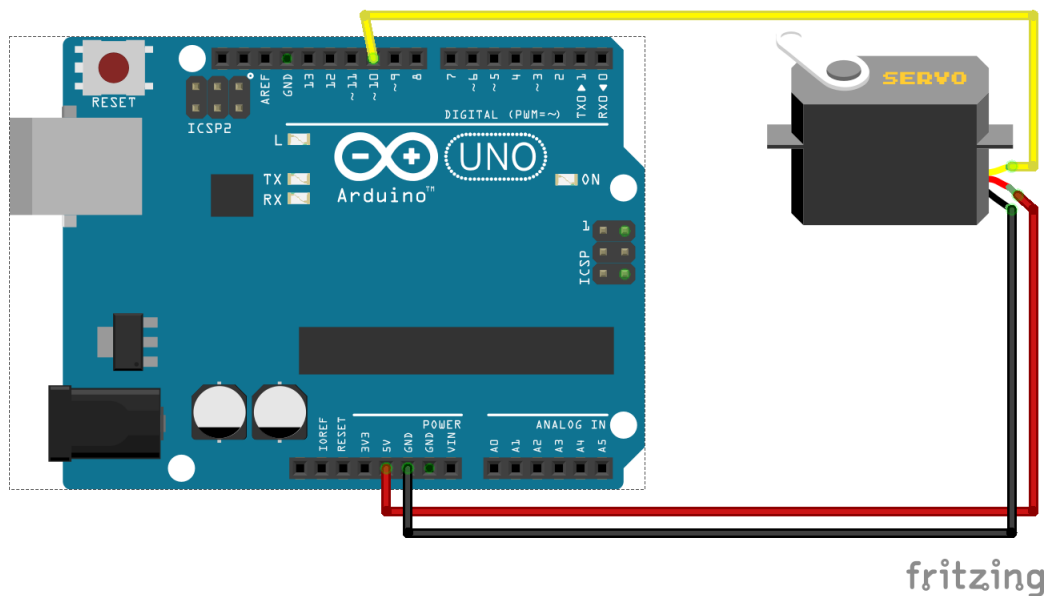


Motor Servo merupakan motor listrik dengan menggunakan sistem closed loop. Sistem tersebut digunakan untuk mengendalikan akselerasi dan kecepatan pada sebuah motor listrik dengan keakuratan yang tinggi. Selain itu, motor servo biasa digunakan untuk mengubah energi listrik menjadi mekanik melalui interaksi dari kedua medan magnet permanen.

Pada umumnya, motor servo terdiri dari tiga komponen utama yaitu (Motor, Sistem kontrol, Potensiometer atau encoder). Motor berfungsi sebagai penggerak roda gigi agar dapat memutar potensiometer dan poros output-nya secara bersamaan. Potensiometer atau encoder berfungsi sebagai sensor yang akan memberikan sinyal umpan balik ke sistem

kontrol untuk menentukan posisi targetnya.

Rangkaian Servo dengan arduino



Source code

```
//Download dan masukan Library
#include <Servo.h>

//Inisialisasi pembuatan variable
int outServo=10; //Pin untuk Servo
Servo myservo;

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  myservo.attach(outServo);
}

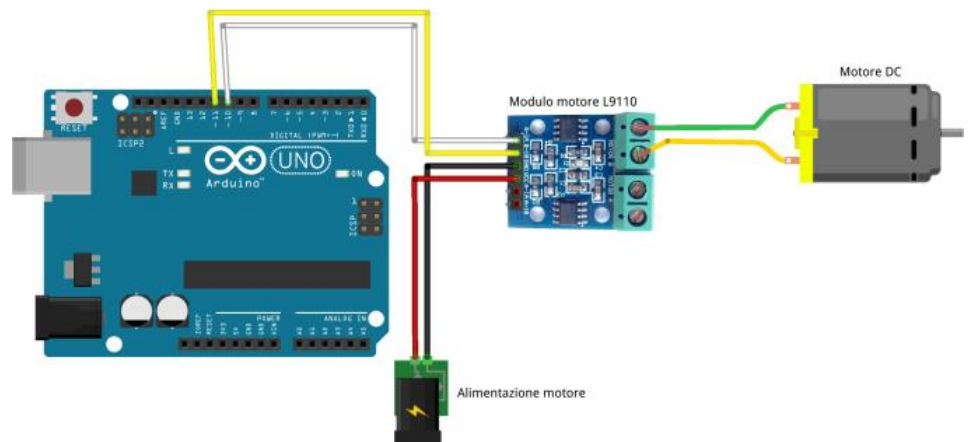
void loop() {
  //Program yang akan dijalankan berulang-ulang
  myservo.write(0); //0-180
  delay(1000); //Jeda waktu perulangan seama 500 mili detik
  myservo.write(180); //0-180
  delay(1000); //Jeda waktu perulangan seama 500 mili detik
}
```

3. Motor Driver + Motor DC



Module Driver motor merupakan rangkaian yang terdiri dari IC Driver salah satunya dengan IC (HG7881) Motor yang berbentuk module yang sudah siap digunakan untuk mengontrol motor DC.

Rangkaian Driver motor arduino



fritzing

	AIN1	AIN2
Forward Direction	HIGH	LOW
Reverse Direction	LOW	HIGH
Brake / Stopped	LOW	LOW
Brake / Stopped	HIGH	HIGH

Source Code

```
//Inisialisasi pembuatan variable
#define A1 5 // Motor A pins
#define A2 6
#define B1 10 // Motor B pins
#define B2 11

int incomingByte = 0; // for incoming serial data
int input = 0;

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT);
  pinMode(B1, OUTPUT);
  pinMode(B2, OUTPUT);
  digitalWrite(A1, LOW);
  digitalWrite(A2, LOW);
  digitalWrite(B1, LOW);
  digitalWrite(B2, LOW);

  Serial.println("Silahkan Pilih salah satu:");
  Serial.println("1.forward");
  Serial.println("2.backward");
  Serial.println("3.stop");
}
```



```

void loop() {
  //Program yang akan dijalankan berulang-ulang
  if (Serial.available() > 0) {
    // read the incoming byte:
    incomingByte = Serial.read();
    input = incomingByte - 48; //convert ASCII code of numbers to 1,2,3

    switch (input) {
      case 1:      // if input=1 ..... motors turn forward
        forward();
        break;
      case 2:      // if input=2 ..... motors turn backward
        backward();
        break;
      case 3:      // if input=1 ..... motors turn stop
        Stop();
        break;
    }
    delay(200);
    input=0;
  }
}

void forward() {      //function of forward
  analogWrite(A1, 255);
  analogWrite(A2, 0);
  analogWrite(B1, 255);
  analogWrite(B2, 0);
}

void backward() {      //function of backward
  analogWrite(A1, 0);
  analogWrite(A2, 210);
  analogWrite(B1, 0);
  analogWrite(B2, 210);
}

void Stop() {          //function of stop
  digitalWrite(A1, LOW);
  digitalWrite(A2, LOW);
  digitalWrite(B1, LOW);
  digitalWrite(B2, LOW);
}

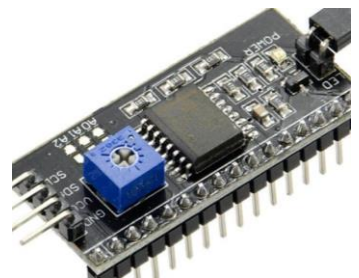
```

4. LCD I2C 16 x 2 atau 20x4

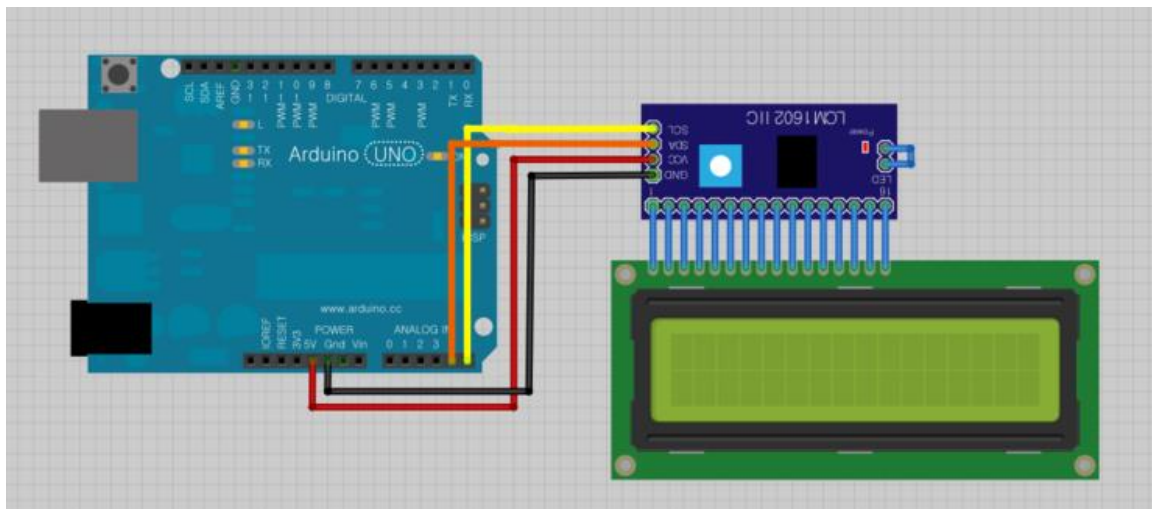


LCD adalah media tampilan yang paling mudah untuk diamati karena menghasilkan tampilan karakter yang baik dan cukup banyak. Pada LCD 20x4 dapat ditampilkan 80 karakter, masing-masing baris memiliki 20 karakter. LCD 20x4 pada umumnya menggunakan 16 pin sebagai kontrolnya, tentunya akan sangat boros apabila menggunakan 16 pin tersebut. Karena itu, digunakan driver khusus sehingga LCD dapat dikontrol dengan jalur I2C. melalui I2C maka LCD dapat dikontrol dengan menggunakan 2 pin saja yaitu SDA dan SCL.

Inter Integrated Circuit atau sering disebut **I2C** adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Slave adalah piranti yang dialamati master.



Rangkaian LCD 16x2 I2C Aduino



Source Code

```
//Download Library LiquidCrystal_I2C
#include <LiquidCrystal_I2C.h>

//Inisialisasi pembuatan variable
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  //Menentukan Fungsionalitas dari PIN pada Microcontroller
  Serial.begin(115200); //baudrate yang digunakan untuk komunikasi microcontroller dengan komputer
  lcd.begin();
}
```

```
void loop() {  
  //Program yang akan dijalankan berulang-ulang  
  lcd.clear(); //Untuk Menghapus karakter pada LCD  
  lcd.setCursor(0,0); lcd.print("  TUTORIAL BY");  
  lcd.setCursor(0,1); lcd.print("Interactive Robotics");  
  delay(1000); //Jeda waktu perulangan seama 1000 mili detik  
}
```

Chalage

Otomatisasi Sensor dan Akuator

1. LDR + Flame + Temperature + Relay + Buzzer + LCD
2. Ultrasonic + Servo + Buzzer + LCD
3. RFID + Servo/Relay/Motor Driver + Buzzer + LCD