Arduino

NodeMCU

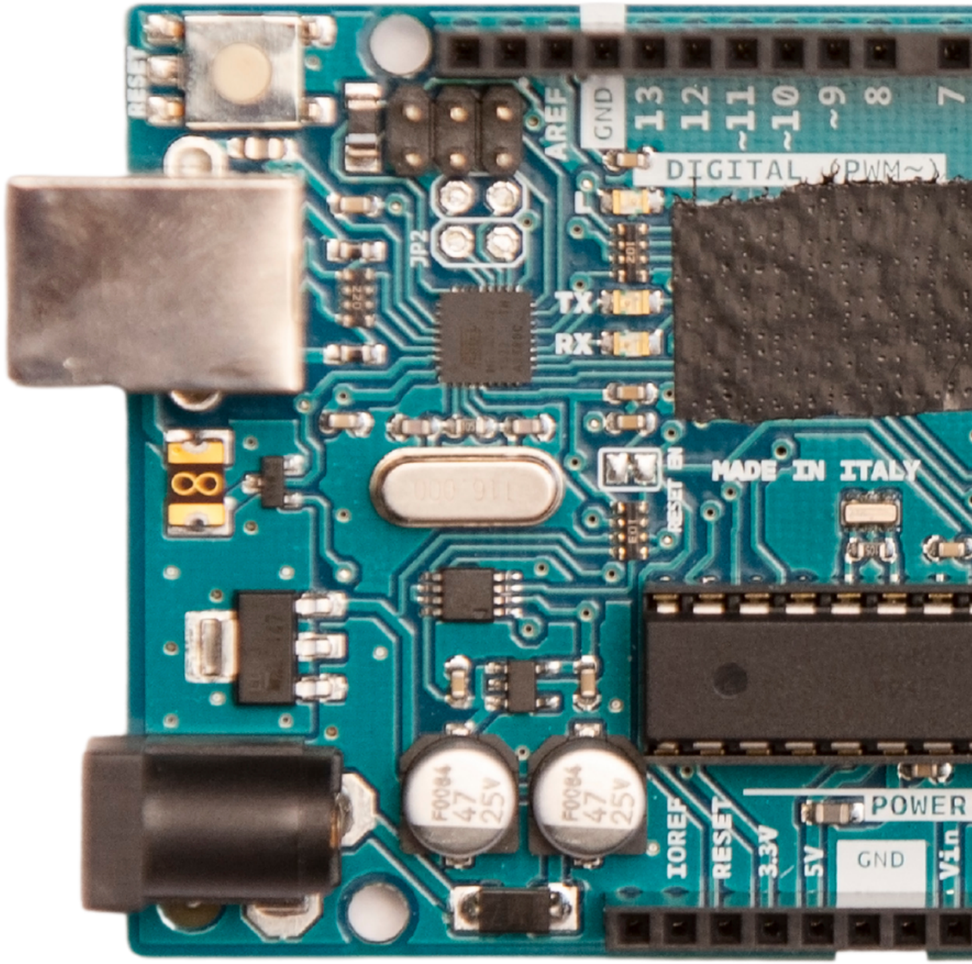
C++

M DIMAS EDITIYA | 582276

ARDUINO PROGRAMMING

Arduino programming with C++ on NodeMCU

Beyond Expectation



Message

Semoga semua hal yang didapatkan disini bisa menjadi manfaat bagi semuanya. Susah bukan berarti tak bisa, hanya butuh waktu lebih lama.

Tobe Connected

Coders Indonesia

Mochammad Dimas Editiya

582276

M Dimas Editiya

Idea Activator

Mochammad Dimas Editiya

Idea Activator and also as Content Creator for Programming content on youtube. I really interesting on programming and all things about automations. I have been playing arround computer, programming, and automations since 2006.

Cageur
Bener
Pinter

Skill

Virtualization

- Docker
- Kubernetes
- Openshit
- Augmented Reality

Embbed System

- Arduino
- Raspberry Pi
- NodeMCU

Aircraft

Saya seorang mechanic dengan license Avionic yang tercatat di DKUPPU. Dan sudah rating spesialisasi pesawat udara Boeing 737NG

582276

M Dimas Editiya

Idea Activator

Programming

- Java
- JavaScript
- TypeScript
- C++
- HTML
- CSS
- PHP
- Python
- Dart

Framework

- Laravel (PHP)
- Spring Boot (Java)
- Angular (TypeScript)
- React (JavaScript)
- Tailwind (CSS)
- Arduino (C++)
- Flutter (Dart)

Arduino Programming

Platform Review

Platform review

ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software.

---

60 Types+

Opensource

Opensource membuat arduino tumbuh lebih cepat dan termaintain dengan baik. Dan membuatnya semakin lebih mudah terintegrasi dengan software lainnya.

It's simple!

Arduino membuat proses development embed sistem menjadi lebih mudah. Karena semua aturan mainnya jelas antara hardware dan software yang digunakan.

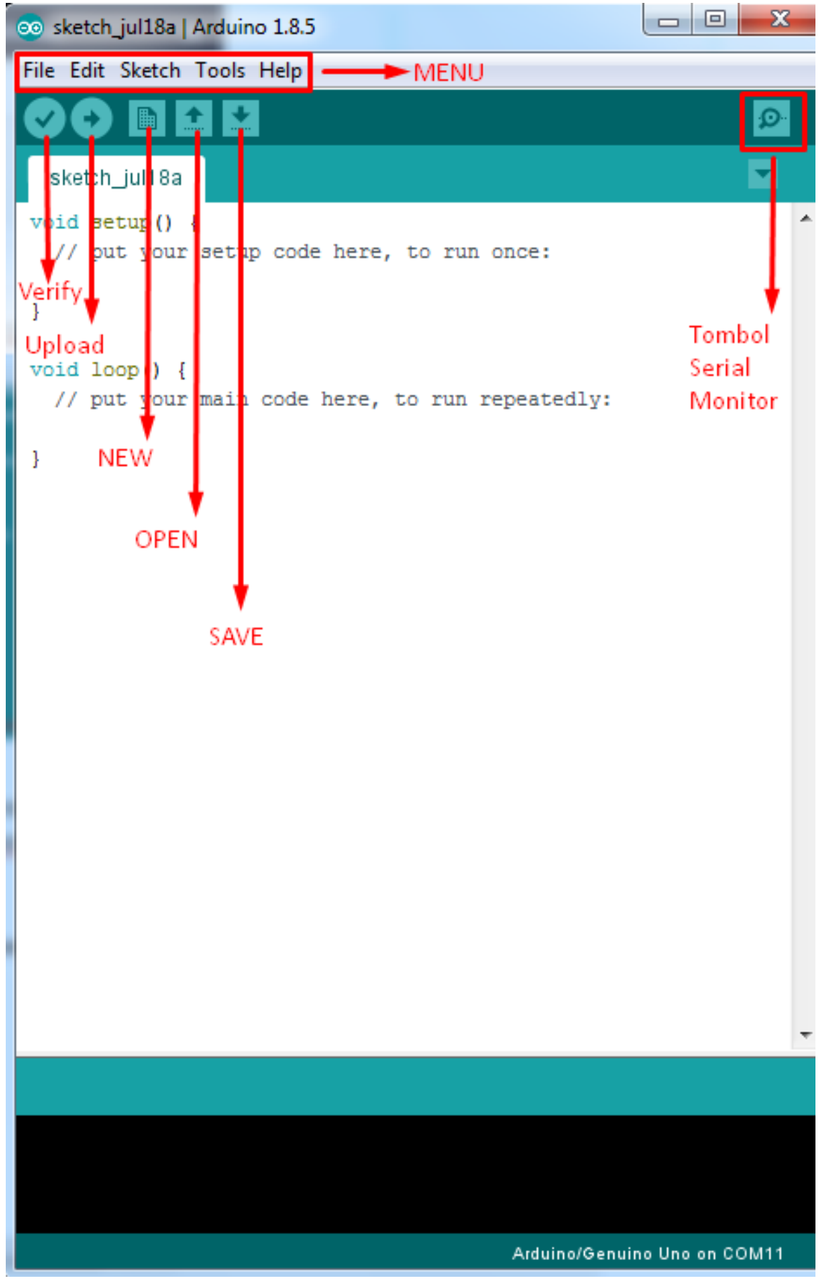
Arduino menggunakan C++ sebagai basic pemrogramannya.

ARDUINO IDE

C++ IDE

Integrated Development Editor (IDE) adalah sebuah tools untuk melakukan koding atau pemrograman. Arduino IDE adalah sebuah editor dengan support pemrograman C++ yang sudah disesuaikan dengan behaviour device. Dalam hal ini device yang memiliki bootloader seperti arduino.

Adapun dalam IDE ini, kelengkapan yang disediakan seperti Serial Monitor, compiler, verifier, library management dll.



Platform review

NODEMCU

An open-source firmware and development kit that helps you to prototype your IOT product within.

20 Features+

Arduino-like hardware IO

Dengan abilitas yang lengkap, bisa mengurangi penggunaan hardware pada project. Semua abilitas board arduino, bisa dilakukan di NodeMCU

Complete

NodeMCU dibangun menggunakan chip ESP8266 yang memiliki hampir seluruh abilitas mikrokontroller dengan tambahan kemudahan konektivitas.

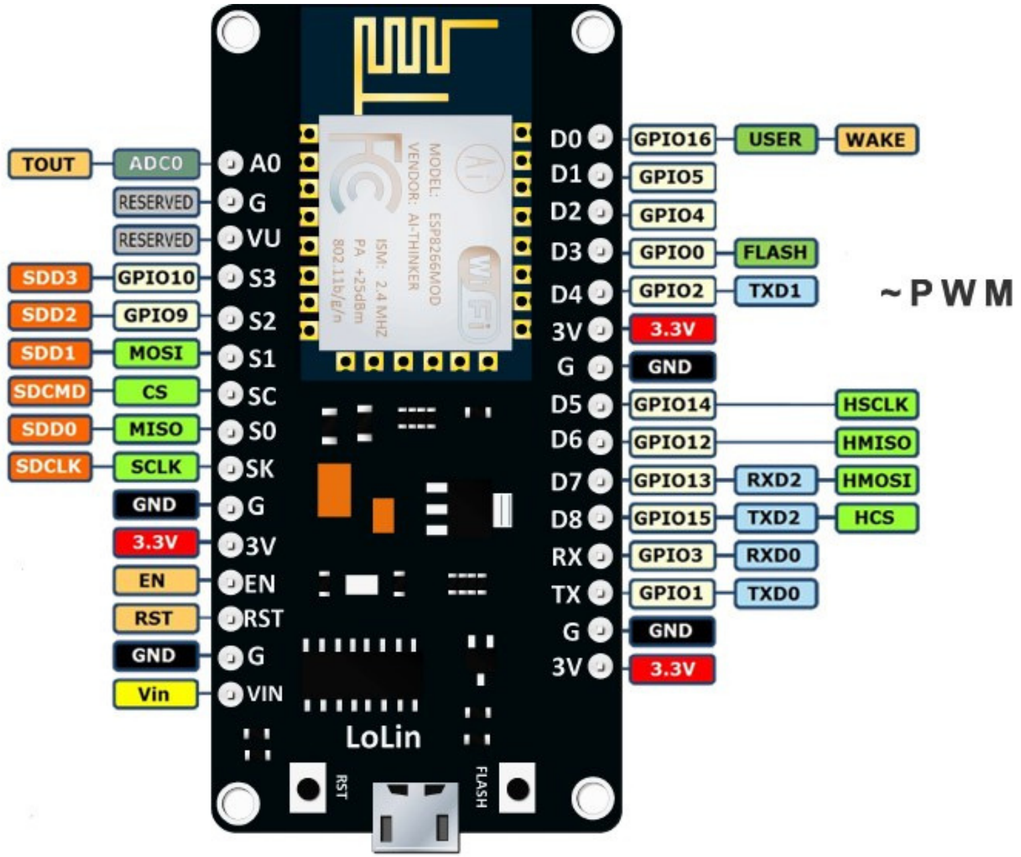
Dengan satu board, kamu dapat membuat project IoT dengan mudah.

NODEMCU BOARD

Built-in ESP8266

ESP8266 adalah sebuah chip yang memiliki abilitas arduino (I/O) dilengkapi dengan konektivitas yang lengkap. Salah satunya adalah konektivitas dengan WiFi. Tak hanya itu, ESP8266 mampu memproses beberapa protokol koneksi yang umum. Seperti HTTP, Websocket, MQTT dll.

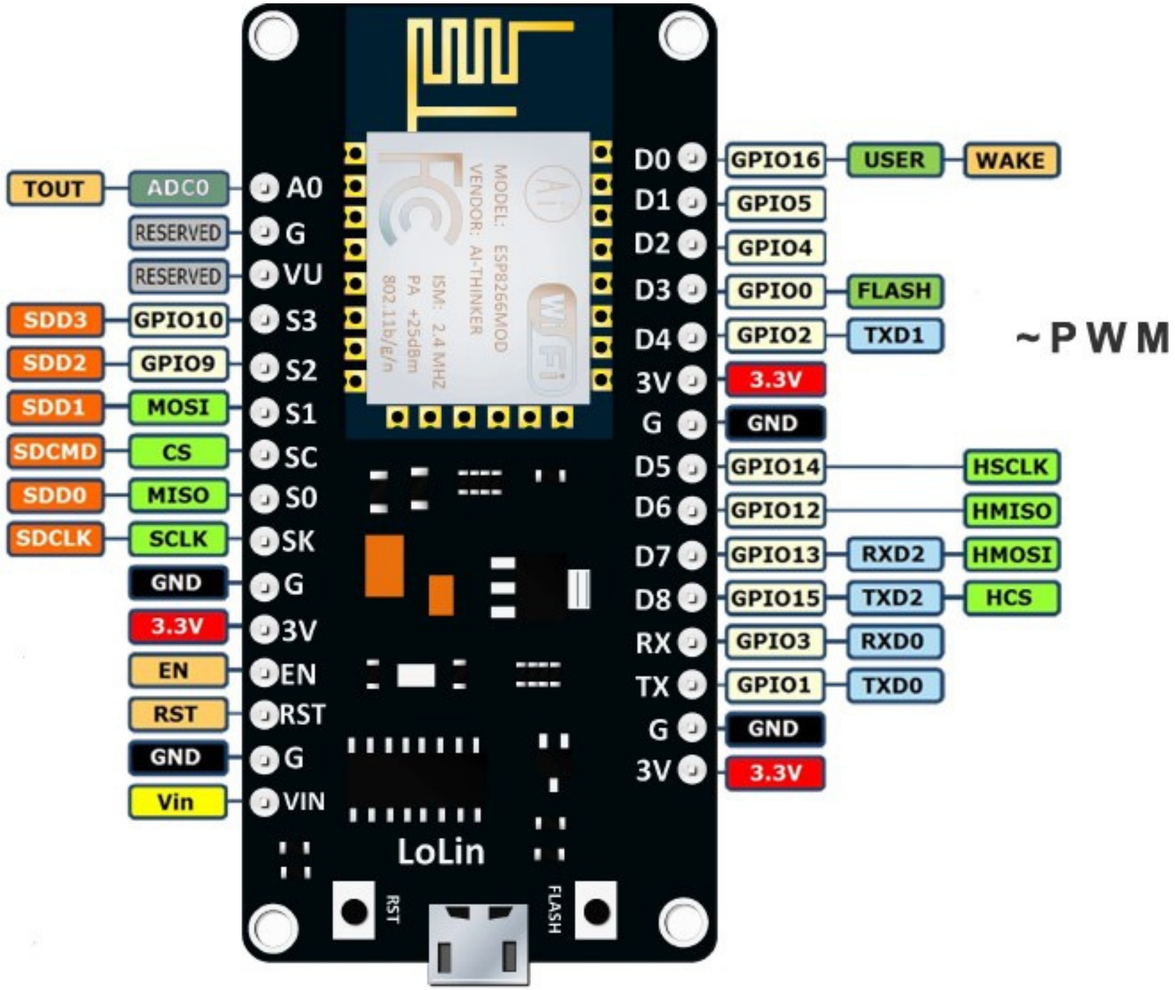
ESP8266 menjadi otak dari Board NodeMCU. NodeMCU memiliki tegangan referensi sebesar 3.3V. Berbeda dengan arduino yang memiliki tegangan referensi sebesar 5V.



Arduino Programming

Common Terms

COMMON TERMS



PWM

Pulse Width Modulation. Kemampuan untuk melakukan modulasi sinyal

SERIAL

Bentuk protokol komunikasi antara hardware to hardware

VIN

Pin untuk memberikan tegangan pada board

TX/RX

Transmit/Receive . Digunakan untuk komunikas

ADC

Analog Digital Converter. Proses konversi sinyal analog menjadi digital.

COMMON TERMS

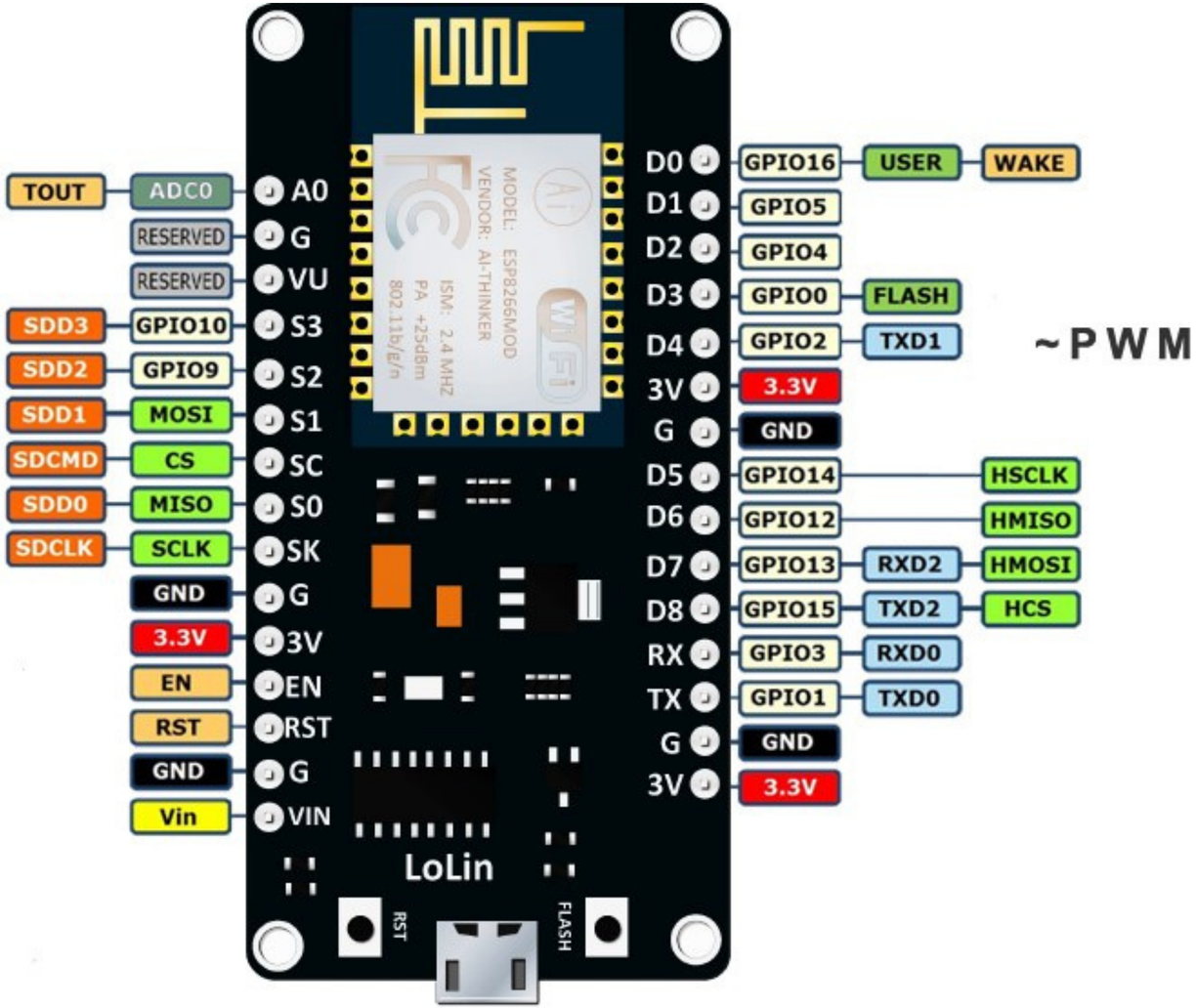
GPIO
General Purpose Input Output. Sebutan lain untuk pin

PIN
Kaki-kaki atau holes pada board yang digunakan untuk konektivitas dengan hardware

GND
Ground. sebagai titik nol atau kutub negatif

VCC
Tegangan sumber. Sebagai kutub positif.

RST
Reset. Pin reset yang digunakan untuk melakukan restart pada board.



Arduino

NodeMCU

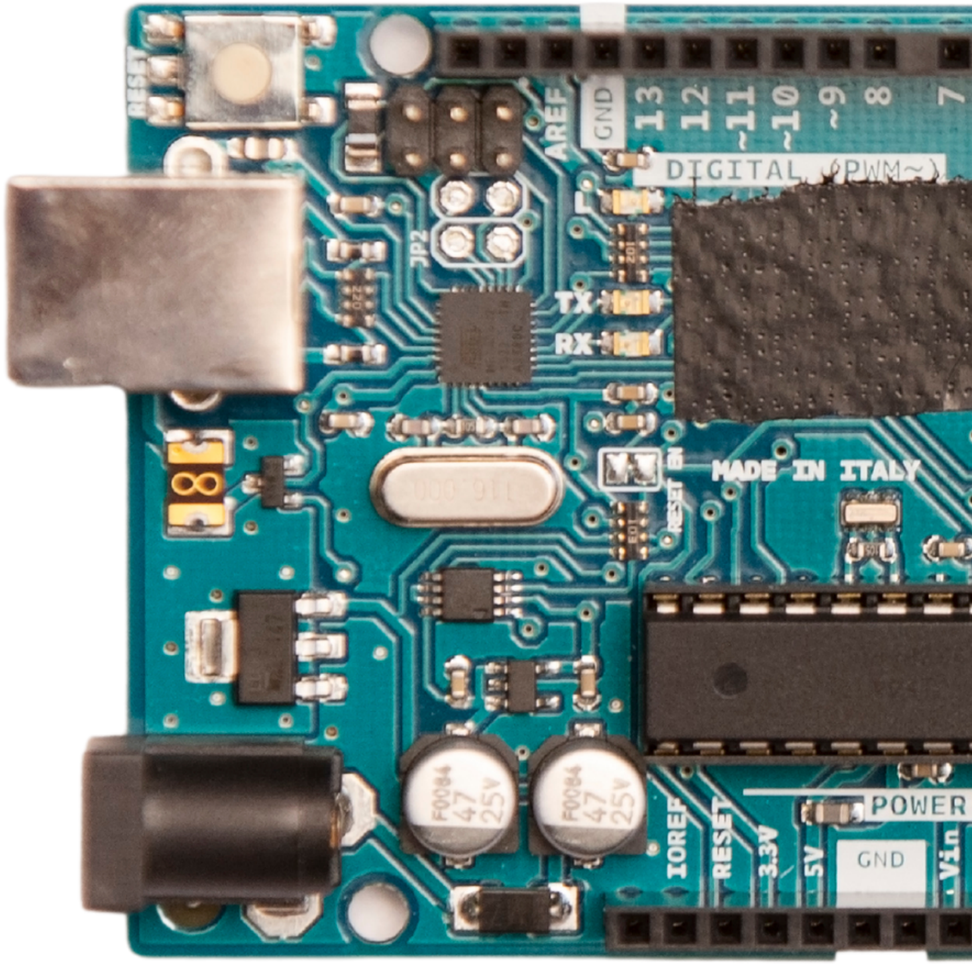
C++

ARDUINO PROGRAMMING

PROGRAMMING SECTION 1

Arduino programming with C++ on NodeMCU

Beyond Expectation



Arduino Programming

Commons Type

Pemrograman

Type Data

Type data menjadi dasar dalam penentuan alokasi memori di dalam pemrograman. Jika sebuah data tidak didefinisikan, program tidak akan berjalan.

Karena pada dasarnya sistem hardware tidak akan mengerti alias kebingungan untuk melakukan alokasi memori.

COMMON DATA TYPE

char
untuk menyimpan hanya satu karakter

```
char flag = 'F';
```

String
untuk menyimpan data berupa text

```
String nama = "Dimas";
```

bool
untuk menyimpan data berupa logic

```
bool isActive = true;
```

int
untuk menyimpan data berupa bilangan bulat

```
int usia = 17;
```

float
untuk menyimpan data berupa bilangan real

```
float humid = 25.1;
```

COMMON DATA TYPE

long
untuk menyimpan bilangan bulat 4 byte

```
long millisecond = 1662981269464;
```

void
untuk menyimpan data kosong

```
void getData() {}
```

unsigned
untuk menyimpan data hanya nilai positif

```
unsigned int totalDays = 25;
```

size_t
untuk menyimpan besaran kapasitas data

```
size_t arrayVol = sizeof(array);
```

COMMON DATA TYPE

Constants

HIGH | LOW
INPUT | OUTPUT | INPUT_PULLUP
LED_BUILTIN
true | false
Floating Point Constants
Integer Constants

Conversion

(unsigned int)
(unsigned long)
byte()
char()
float()
int()
long()
word()

Data Types

array
bool
boolean
byte
char
double
float
int
long
short
size_t
string
String()
unsigned char
unsigned int
unsigned long
void
word

Variable Scope & Qualifiers

const
scope
static
volatile

Utilities

PROGMEM
sizeof()

Full Docs
https://www.arduino.cc/reference/en/

Arduino Programming

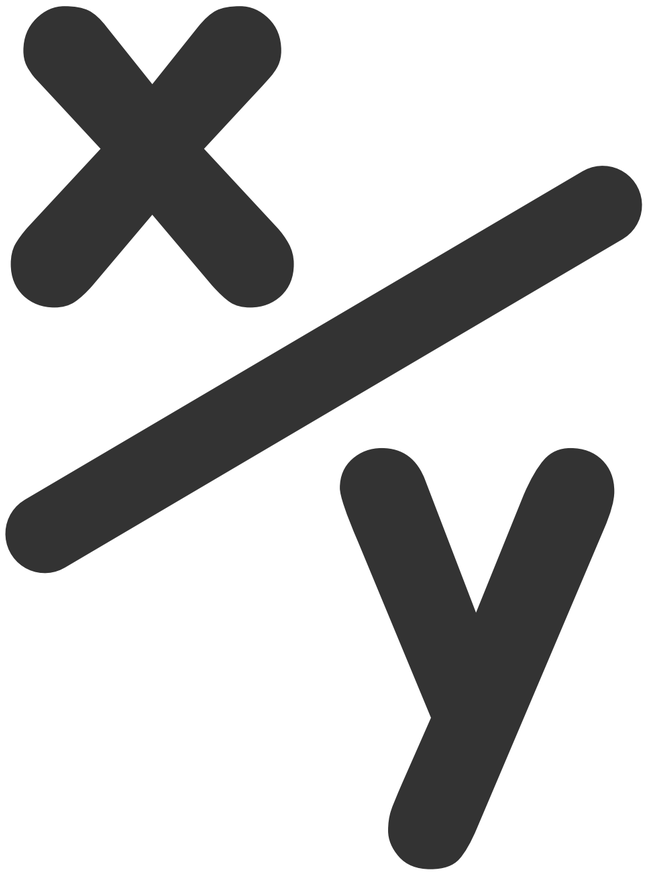
Variable

Pemrograman

Deklarasi Variable

Variable menjadi struktur utama dalam pemrograman. Variable dapat menyimpan sebuah nilai dan dapat dijadikan sebagai acuan untuk melakukan proses yang ada di dalam program yang berjalan.

Variable harus didekalarasikan terlebih dahulu sebelum digunakan. Deklarasi biasa ditulis dengan menyebutkan nama variable dan jenis data dari variable tersebut.



Pemrograman

Deklarasi Variable

Penulisan nama variable tidak boleh terpisah oleh spasi. Jika memang harus bisa menggunakan camelCase, atau underscore (pascal case)

```
long millisecond = 1662981269464;
```

```
void getData() {}
```

```
unsigned int totalDays = 25;
```

```
typedata nama_variable = value_variable;
```

```
typedata namaVariable = value_variable;
```

Arduino Programming

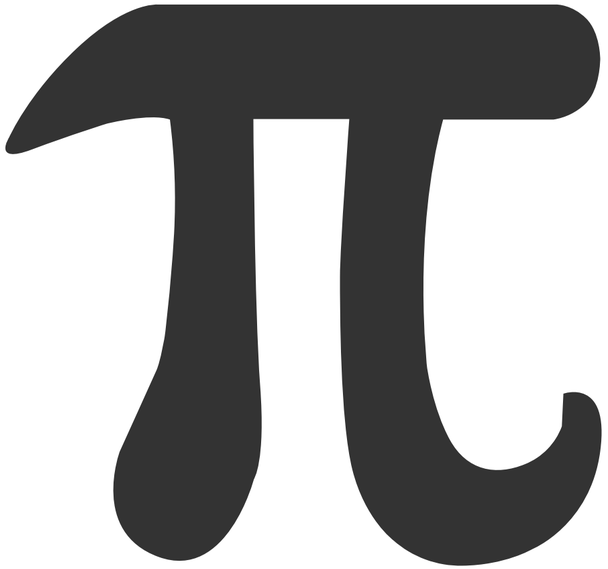
Constant

Pemrograman

Deklarasi Konstanta

Konstanta adalah sebuah nilai yang sudah didefinisikan secara tetap dan tidak berubah. Hal ini akan mempermudah pembacaan program secara keseluruhan.

Konstanta dapat berupa nilai yang bermacam-macam. Mulai dari string, integer, float atau boolean.



Pemrograman

Deklarasi Konstanta

```
const char isActive = 'y';
```

```
#define PIN_RELAY D1
```

Konstanta dapat didefinisikan dengan **const** atau **#** (**directive**).

```
const typedata nama_konst = value_const;
```

```
#define token value
```

COMMON CONSTANTS

HIGH
memberikan nilai digital 1 ~ (5V | 3.3V)

```
digitalWrite(11, HIGH);
```

LOW
memberikan nilai digital 0 ~ 0V

```
digitalWrite(11, LOW);
```

LED_BUILTIN
konstanta untuk pin dengan led builtin

```
digitalWrite(LED_BUILTIN, LOW);
```

INPUT
konstanta untu set pin menjadi input

```
pinMode(11, INPUT);
```

OUTPUT
konstanta untu set pin menjadi output

```
pinMode(11, OUTPUT);
```

Arduino Programming

function

Pemrograman

Deklarasi function

Function menjadi hal yang sangat berguna untuk melakukan enkapsulasi terhadap kode pemrograman. Function umumnya memiliki inputan yg disebut dengan **parameter** dan nilai return atau **nilai kembali**

Pemrograman

Deklarasi function

Function biasa kita temukan dalam matematika. Umumnya digunakan untuk melakukan operasional matematika dalam blok tertentu dan untuk kasus tertentu.

```
f(x) = 2x + 3

untuk:
x = 2 -> f(2) = 2.2 + 3 = 7

x = 2 adalah parameter
7 hasil fungsi disebut return value
```

```
int myFunction(int x) {
    return 2.x + 3;
}
```

```
typedata namaFunction (typedata params) {...}
```

COMMON FUNCTION

int digitalRead(pin)
Untuk membaca nilai digital pada pin

```
digitalRead(11);
```

void digitalWrite(pin, value)
untuk mengeluarkan nilai digital pada pin

```
digitalWrite(11, HIGH);
```

void pinMode(pin, mode)
untuk set pin GPIO

```
pinMode(11, OUTPUT);
```

int analogRead(pin)
untuk membaca nilai analog pada pin

```
analogRead(11);
```

void analogWrite(pin, value)
untuk menyimpan data berupa bilangan real

```
analogWrite(11, 100);
```

COMMON FUNCTION

void delay(value)
Memberikan delay dalam order millisecond

```
delay(1000);
```

unsigned long millis()
memberikan nilai milisecond dari saat board hidup

```
unsigned long time = millis();
```

int Serial.read()
untuk membaca data pada jalur komunikas serial

```
int data = Serial.read();
```

void Serial.write(val)
untuk menulis data di jalur komunikasi serial

```
Serial.write(11);
```

Buffer Serial.available()
Melakukan cek data pada komunikasi serial

```
if (Serial.available() > 0) {...}
```

void Serial.begin(baudrate)
Melakukan inisialisasi komunikasi serial

```
Serial.begin(9600);
```

Arduino Programming

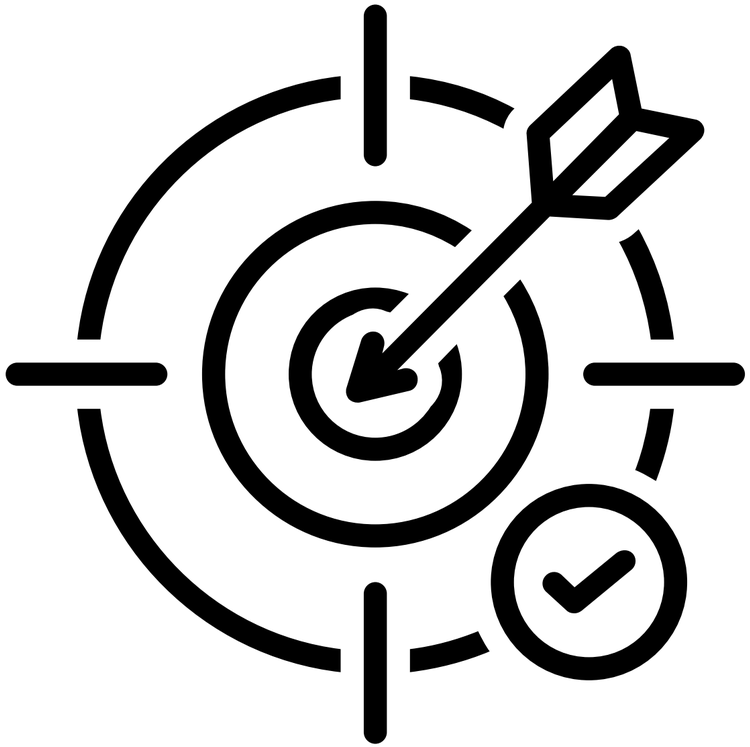
Property Access

Pemrograman

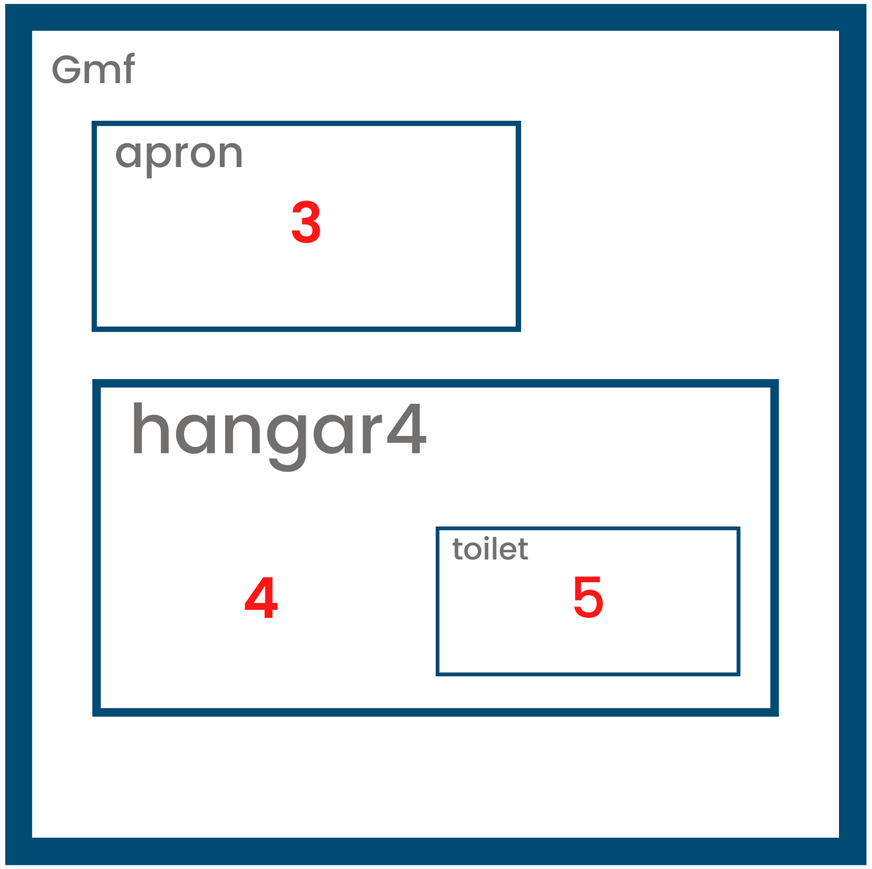
Property Access

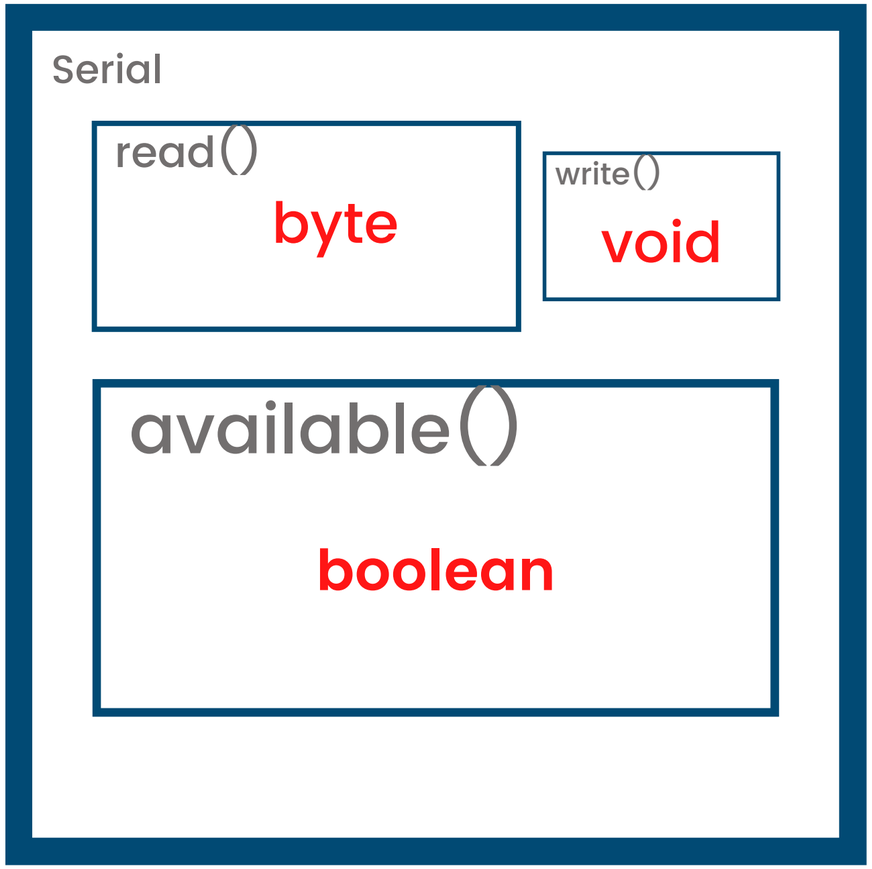
Adalah sebuah metode untuk mengakses suatu nilai atau service tertentu yang ada pada sebuah blok pemrograman.

Umumnya, kita menggunakan notasi titik [.] untuk melakukan akses ke sebuah properti milik object pemrograman.



Property Access






```
Gmf.apron -> 3
Gmf.hangar4 -> 4
Gmf.hangar4.toilet -> 5
```

```
Serial.read() -> byte
Serial.write() -> void
Serial.available() -> boolean
```

Arduino Programming

Main Structure

Main Structure

Save it, use later.

Arduino memiliki struktur dasar dalam pemrogramannya. Struktur ini berlaku secara universal ketika kita berurusan dengan pemrograman C++ nya arduino.

Adapun struktur tersebut terdiri dari

1. Declaration
2. Setup
3. Loop

```
#include <Arduino.h>

void setup() {

}

void loop() {

}
```

Main Structure

Declaration

Ini berada di posisi paling atas dari keseluruhan kode program. Ini menjadi penting, karena proses pembacaan program berjalan secara **up to bottom.** Jika urutannya tidak teratur, kemungkinan variable atau function yang akan digunakan tidak akan terbaca, karena belum dikenali oleh program

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
```

Deklarasi librari

```
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASS";
const char* mqtt_server = "test.mosquitto.org";

WiFiClient espClient;
PubSubClient client(espClient);
```

Deklarasi variabel, fungsi dan Object

Main Structure

Setup

Function dengan return data void ini akan dijalan **hanya sekali** saat device pertama dihidupkan. Umumnya, pada function setup ini akan berisi proses-proses inisialisasi, proses pengenalan librari, proses pembacaan peripheral yang terhubung dengan board, dan proses-proses lainnya.

```
void setup()
{
  // put your setup code here, to run once:
  pinMode(pin, OUTPUT);
}
```

Main Structure

Loop

Function dengan return data void ini akan dijalan **secara berulang** selama board hidup. Tentu, function loop akan dijalankan setelah proses setup selesai.

```
void loop() {
  /**
   * Update isi dari memori
   * update(alamat_memori, saving_data)
   */
  EEPROM.update(0, ledState);

  /**
   * Membaca isi dari alamat memori
   * read(alamat_memori)
   * masukin hasilnya ke variable ledstate
   */
  ledState = EEPROM.read(0);
  digitalWrite(13, ledState);
}
```

Blok ini akan berisi program utama dari project yang sedang dikerjakan. Semua logic, perulangan, proses utama, akan dilakukan secara berulang. Proses ini berulang dengan jeda hanya 1 microsecond.

Ini bergantung kepada nilai frekwensi yang digunakan pada board.

Arduino Programming Section 1

Coding Time

Coding Time

Code 1: Proses cetak nilai dengan koneksi serial

Buatlah sebuah program di Arduino IDE dengan kondisi sebagai berikut

- Terdapat deklarasi **variable dengan tipe data string dan integer**. Berikan nilai kepada dua variable tersebut.
- Pada blok **setup**, Initialisasi sebuah **koneksi serial** antara NodeMCU dengan Komputer. Untuk koneksi ini, menggunakan **baud rate** sebesar **9800**.
- Cetak setiap nilai variable yang telah dideklarasi dengan menggunakan **koneksi serial pada bagian loop**.
- berikan **delay sebesar 2000 ms**.

Coding Time

Code 2: Proses kendali LED Built-in

Buatlah sebuah program di Arduino IDE dengan kondisi sebagai berikut

- Pada blok setup, set pin **LED_BUILTIN** sebagai output. Berikan nilai awalan berupa **HIGH** pada pin tersebut.
- Pada blok loop, Rubah kondisi (state) pin **LED_BUILTIN** dari **HIGH** menjadi **LOW**. Berikan **delay sebesar 1000ms** setelahnya.
- Pada blok loop, Rubah kondisi (state) pin **LED_BUILTIN** dari **LOW** menjadi **HIGH**. Berikan **delay sebesar 1000ms** setelahnya

Coding Time

Code 3: Membuat fungsi kendali LED dan Serial

Buatlah sebuah program di Arduino IDE dengan kondisi sebagai berikut

- Deklarasikan sebuah fungsi dengan tipe data void pada blok deklarasi.
- Pada blok setup, initialisasi koneksi serial dengan baud rate 9800
- Pada blok setup, set pin LED_BUILTIN sebagai output. Berikan nilai awalan berupa HIGH pada pin tersebut.
- pada blok dibawah setup, implementasi fungsi yang sudah dibuat. Implementasi berisikan kendali kelap-kelip LED dengan delay 1000ms. Dan cetak kondisi LED yang sedang berjalan dengan koneksi serial.
- Pada blok loop, jalankan fungsi yang telah dibuat.

Arduino

NodeMCU

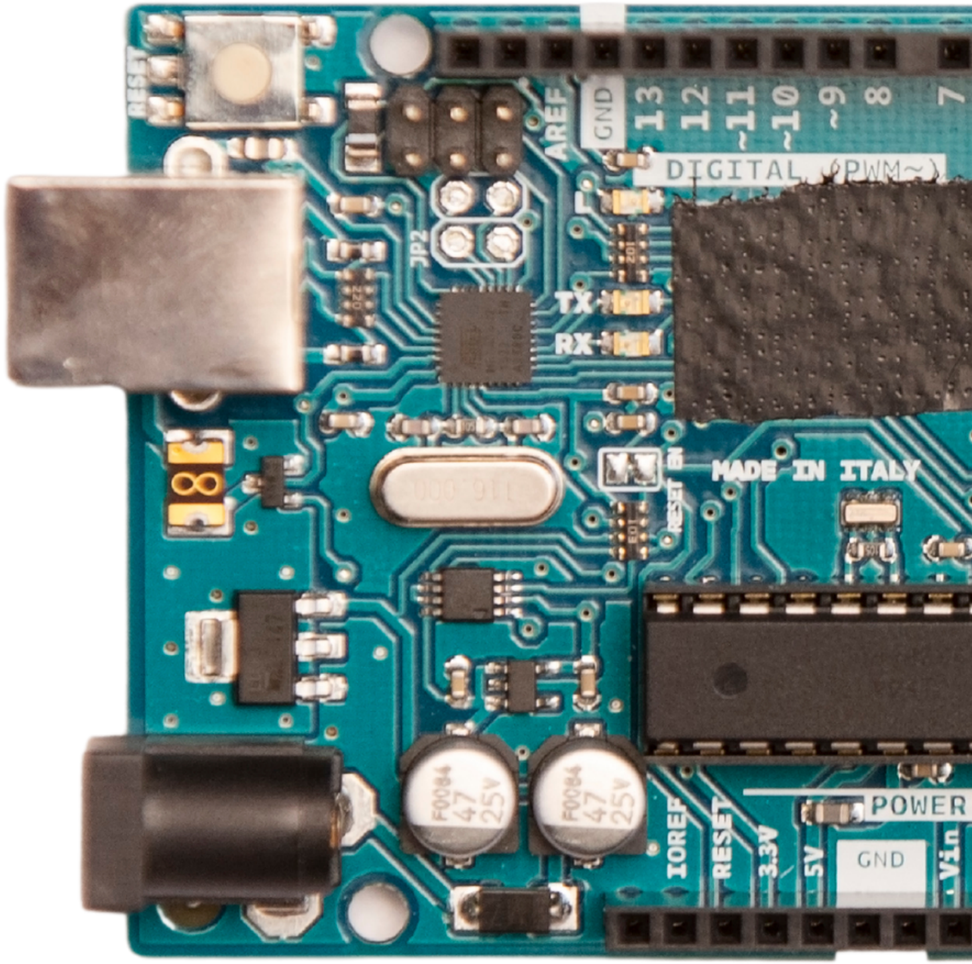
C++

ARDUINO PROGRAMMING

PROGRAMMING SECTION 2

Arduino programming with C++ on NodeMCU

Beyond Expectation



Arduino Programming Section 2

IF Structure

Pemrograman

Deklarasi IF...ELSE...

IF...ELSE... menjadi sebuah kontrol jika terdapat pilihan di dalam sebuah pemrograman. Ini cukup banyak digunakan ketika program kita mengharuskan untuk memilih.

Pada dasarnya, Statement IF...ELSE... akan saling berkaitan dengan operasi boolean.

OPERATION

Operator	Keterangan
==	Sama dengan (bukan pemberi nilai)
!=	Tidak sama dengan
>	Lebih dari
<	Kurang dari
>=	Lebih dari sama dengan
<=	Kurang dari sama dengan

OPERATION

Operator	Keterangan	Contoh
&&	Operator Logika AND	m && n
||	Operator Logika OR	m || n
!	Operator Logika NOT	!m

Pemrograman

Deklarasi IF...ELSE...

Secara singkat, IF...ELSE... terdiri dari dua blok. Blok pertama akan dijalankan ketika kondisi terpenuhi alias bernilai true.

Blok kedua, akan dijalankan ketika kondisi tidak terpenuhi alias bernilai false.

```
Jika temperatur lebih dari 25 maka
nyalakan kipas
atau
Matikan kipas
```

```
If temperatur > 25 then
nyalakan kipas
else
Matikan kipas
```

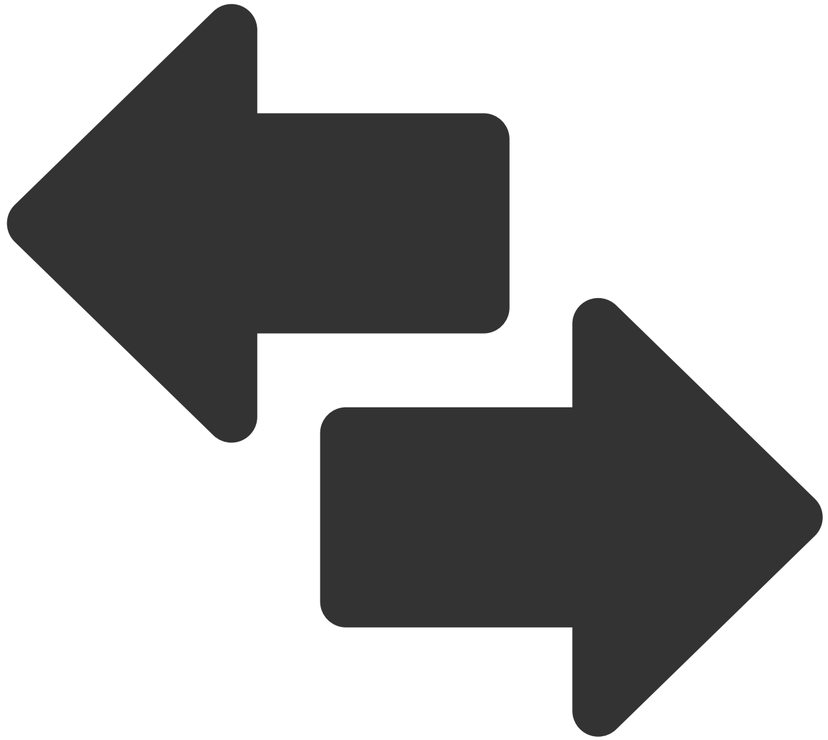
```
if (condition) {
   action when condition is true
} else {
   action when condition is false
}
```

Arduino Programming Section 2

SWITCH Structure

Pemrograman

Deklarasi Switch...Case



SWITCH...CASE... menjadi alternatif dari penggunaan IF...ELSE... dalam melakukan struktur kontrol percabangan.

Dalam operasinya, syntax break menjadi penanda batasan dari segment yang terpilih.

Pemrograman

Deklarasi SWITCH

```
Jika temperatur sama dengan 25 maka
nyalakan kipas
Jika temperatur sama dengan 30 maka
Guyur dengan air
Jika temperatur sama dengan 40 maka
Guyur dengan air & nyalakan kipas
atau
nyalakan kipas
```

```
switch (temperatur) {
    case 25:
      nyalakan kipas
      break;
    case 30:
      Guyur dengan air
      break;
    case 25:
      Guyur dengan air & nyalakan kipas
      break;
    default:
      nyalakan kipas
      break;
}
```

Arduino Programming Section 2

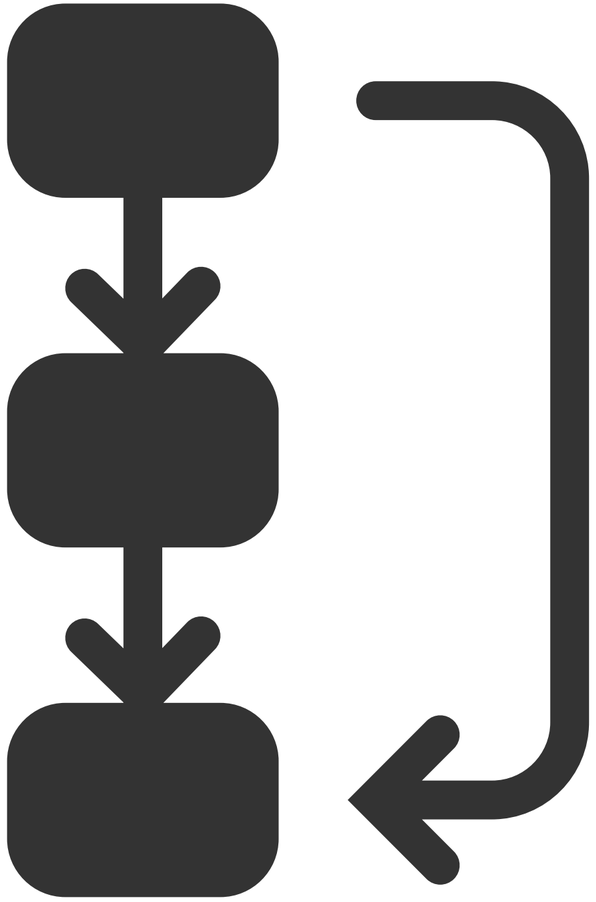
GOTO Structure

Pemrograman

Deklarasi Goto

Goto menjadi solusi ketika kita ingin melakukan skip atau melangkahi proses pada program yang dibuat.

Umumnya ini dilakukan ketika kita ingin mencoba untuk melakukan interupsi dari progress yang sedang berlanjut.



Pemrograman

Deklarasi Goto

```
for (byte r = 0; r < 255; r++) {
  for (byte g = 255; g > 0; g--) {
    for (byte b = 0; b < 255; b++) {
      if (analogRead(0) > 250) {
        goto bailout;
      }
      // more statements ...
    }
  }
}

bailout:
// more statements ...
```

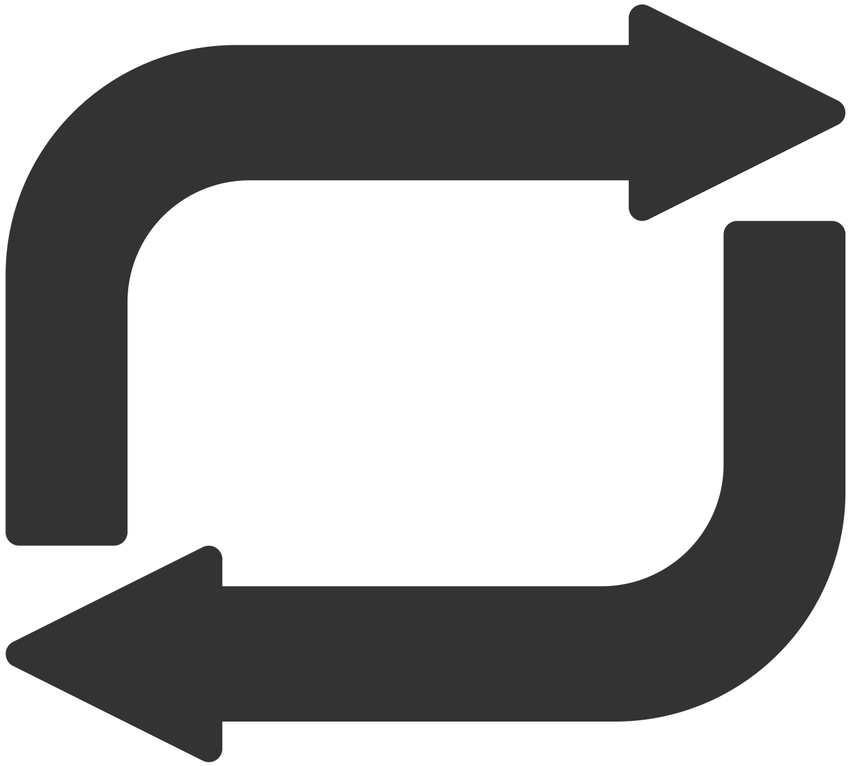
Arduino Programming Section 2

For Loop Structure

Pemrograman

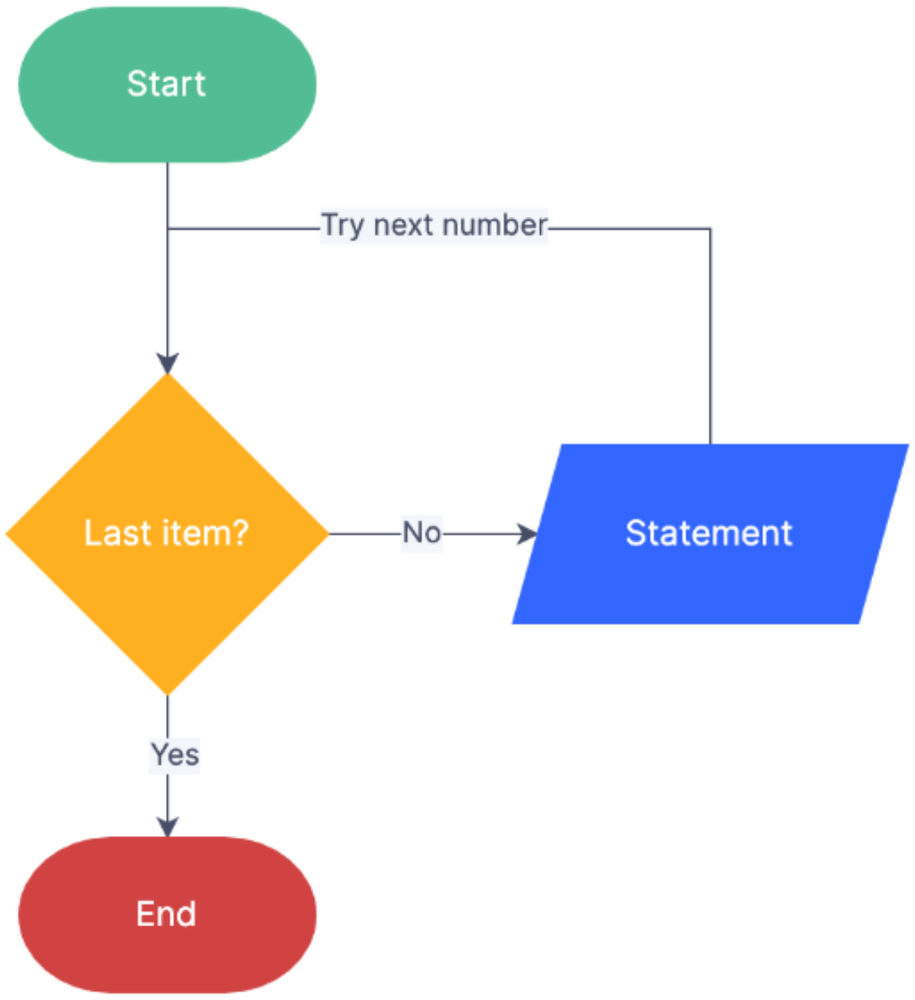
Deklarasi For Loop

For Loop menjadi sebuah struktur kontrol dalam melakukan perulangan. Perulangan ini bergantung pada kondisi yang ditentukan. Selama kondisi bernilai true, maka statement dalam blok akan dijalankan.



Pemrograman

Deklarasi For Loop



For Loop Flowchart

Pemrograman

Deklarasi For Loop

For loop akan **dijalankan secara terus menerus selama kondisi terpenuhi**, dan jika belum terpenuhi, **operasi setelah kondisi akan terus dijalankan**

```
Untuk
mangga sama dengan 0, dan
mangga kurang dari 1000, maka
ambil mangga,
dan tambah 1 mangga
```

```
For manggo = 0, manggo < 1000
take manggo
manggo = manggo + 1
```

```
for (init, condition, operation) {
  //...action
}
```

```
for (m=0, m < 1000, m++) {
  takeManggo();
}
```

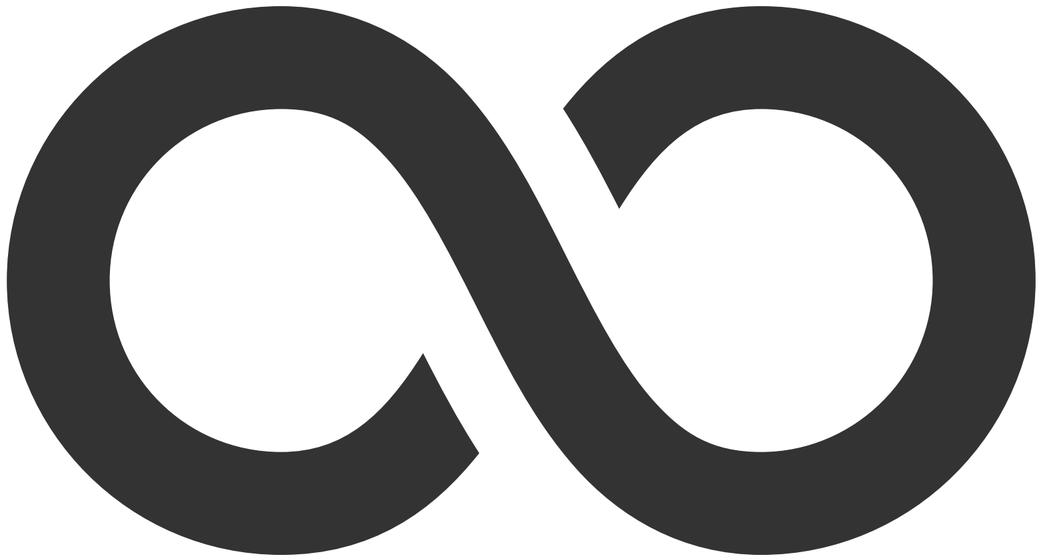
Arduino Programming Section 2

While Loop Structure

Pemrograman

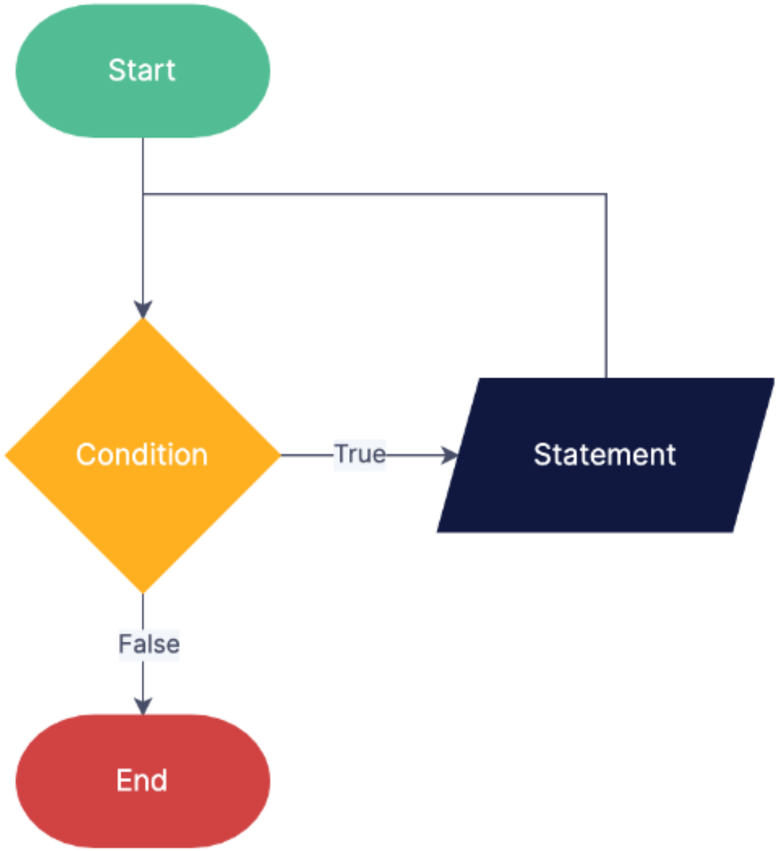
Deklarasi While loop

While Loop menjadi sebuah struktur kontrol dalam melakukan perulangan. Perulangan ini bergantung pada kondisi yang ditentukan. Selama kondisi bernilai true, maka statement dalam blok akan dijalankan.



Pemrograman

Deklarasi While Loop



While Loop Flowchart

Pemrograman

Deklarasi While Loop

While loop akan **dijalankan secara terus menerus selama kondisi terpenuhi**, dan jika belum terpenuhi, **operasi setelah kondisi akan terus dijalankan**

```
Ketika
mangga sama dengan 0, dan
mangga kurang dari 1000, maka
ambil mangga,
dan tambah 1 mangga
```

```
Let manggo = 0,
While manggo < 1000
take manggo
manggo = manggo + 1
```

```
while (condition) {
  //...action
}
```

```
int m = 0
while (m < 1000) {
  takeManggo();
  m++;
}
```

Arduino Programming Section 2

Coding Time

Coding Time

Code 4: Aktivasi led berdasarkan Serial

Buatlah sebuah program di Arduino IDE dengan kondisi sebagai berikut

- Pada blok setup, set pin LED_BUILTIN menjadi OUTPUT, dan inisialisasi komunikasi serial dengan baud rate 9800.
- Untuk membaca string, gunakan Serial.readString().
- Pada blok loop, baca data yang ada di dalam Serial.
  - Jika ada dan bernilai 'Y', maka berikan nilai LOW pada pin LED
  - Jika ada dan bernilai 'N', maka berikan nilai HIGH pada pin LED
- Berikan delay selama 500ms

Coding Time

Code 5: Cetak urutan angka pada Serial

Buatlah sebuah program di Arduino IDE dengan kondisi sebagai berikut

- Pada blok deklarasi, deklarasikan variable dengan tipe data int. Initialisasi dengan nilai 0
- Pada blok setup, inisialisasi komunikasi serial dengan nilai baud rate 9600
- Pada blok loop, buatlah sebuah loop hingga 10 kali. Setiap loopnya akan mencetak angka ke dalam serial.
- berikan delay sebesar 1000ms

Coding Time

Code 6: Cetak letter-index berdasarkan urutan angka

Buatlah sebuah program di Arduino IDE dengan kondisi sebagai berikut

- Pada blok deklarasi, deklarasi variable dengan tipe data integer dan beri nilai 0.
- pada blok setup, inisialisasi serial dengan baud rate 9600
- pada blok loop, buat sebuah loop hingga 10 kali.
  - Jika index loop kurang dari 3, cetak nilai C
  - Jika index loop kurang dari sama dengan 7 dan lebih dari 3, cetak nilai B
  - Jika index loop kurang dari sama dengan 10 dan lebih dari 7, cetak A
- berikan delay 1000ms

“ Code is like humor. When you have to explain it, it’s bad.”

Cory House

Arduino

NodeMCU

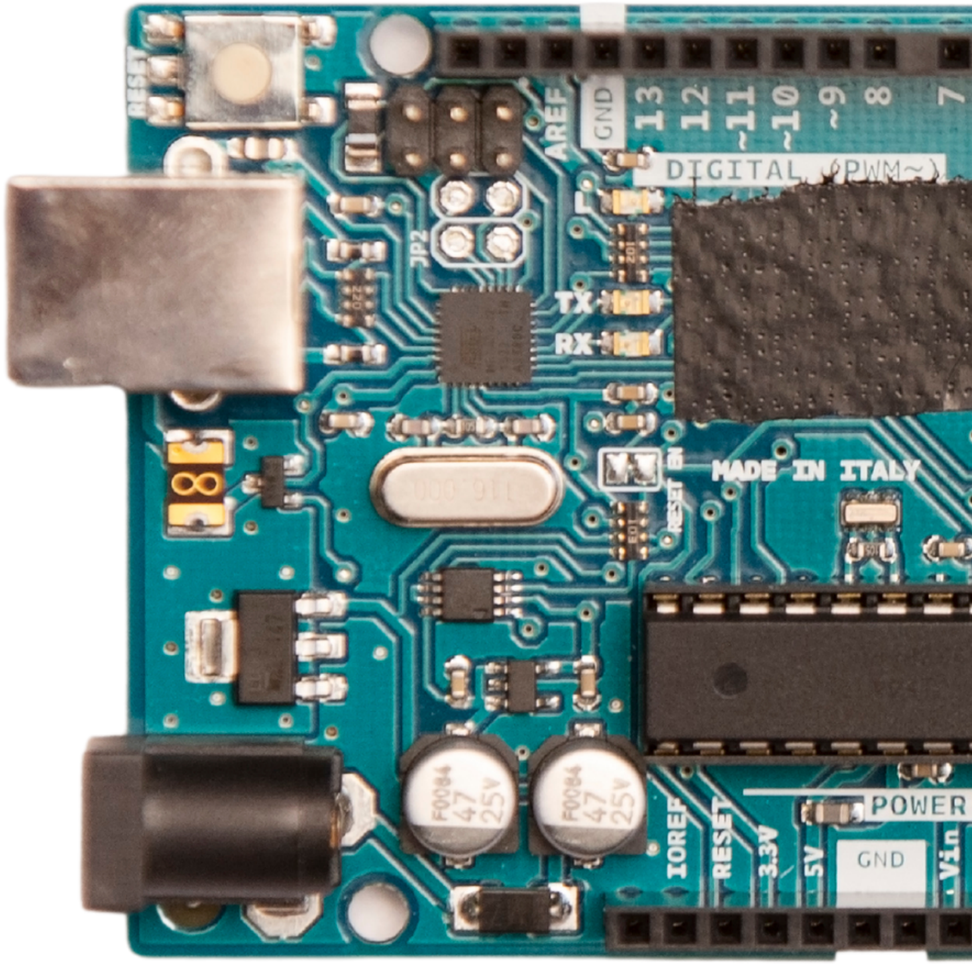
C++

M DIMAS EDITIYA | 582276

ARDUINO PROGRAMMING

Arduino programming with C++ on NodeMCU

Beyond Expectation



Arduino

NodeMCU

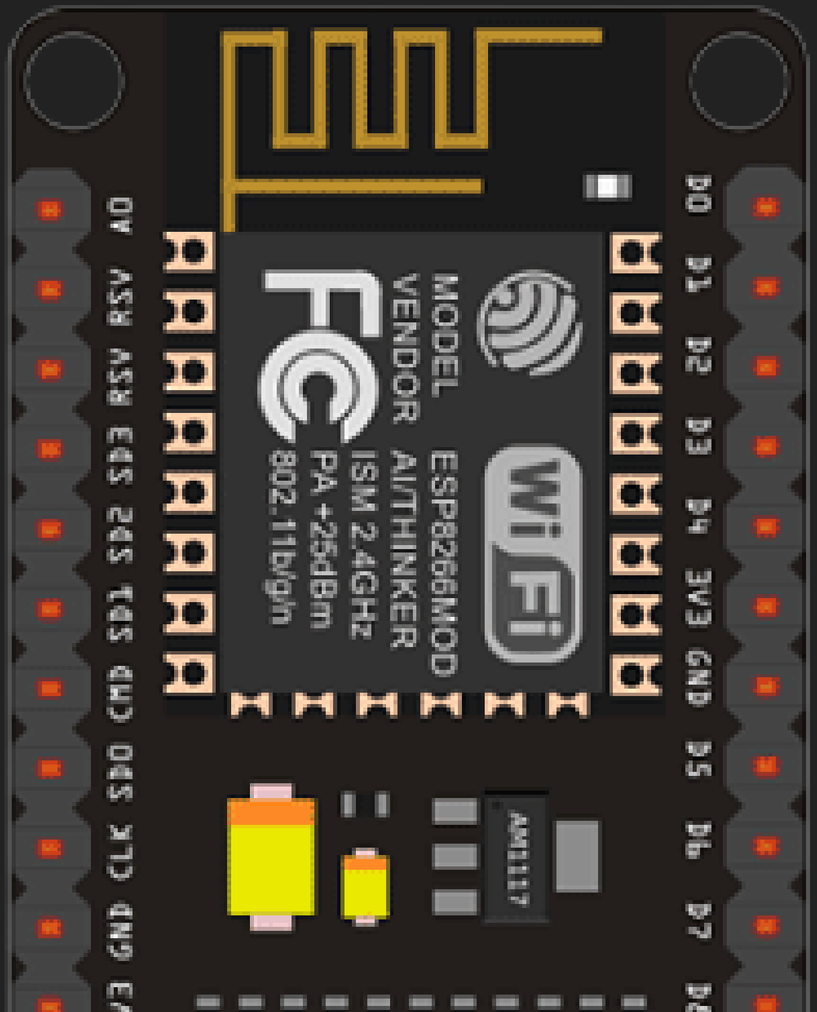
C++

ARDUINO PROGRAMMING

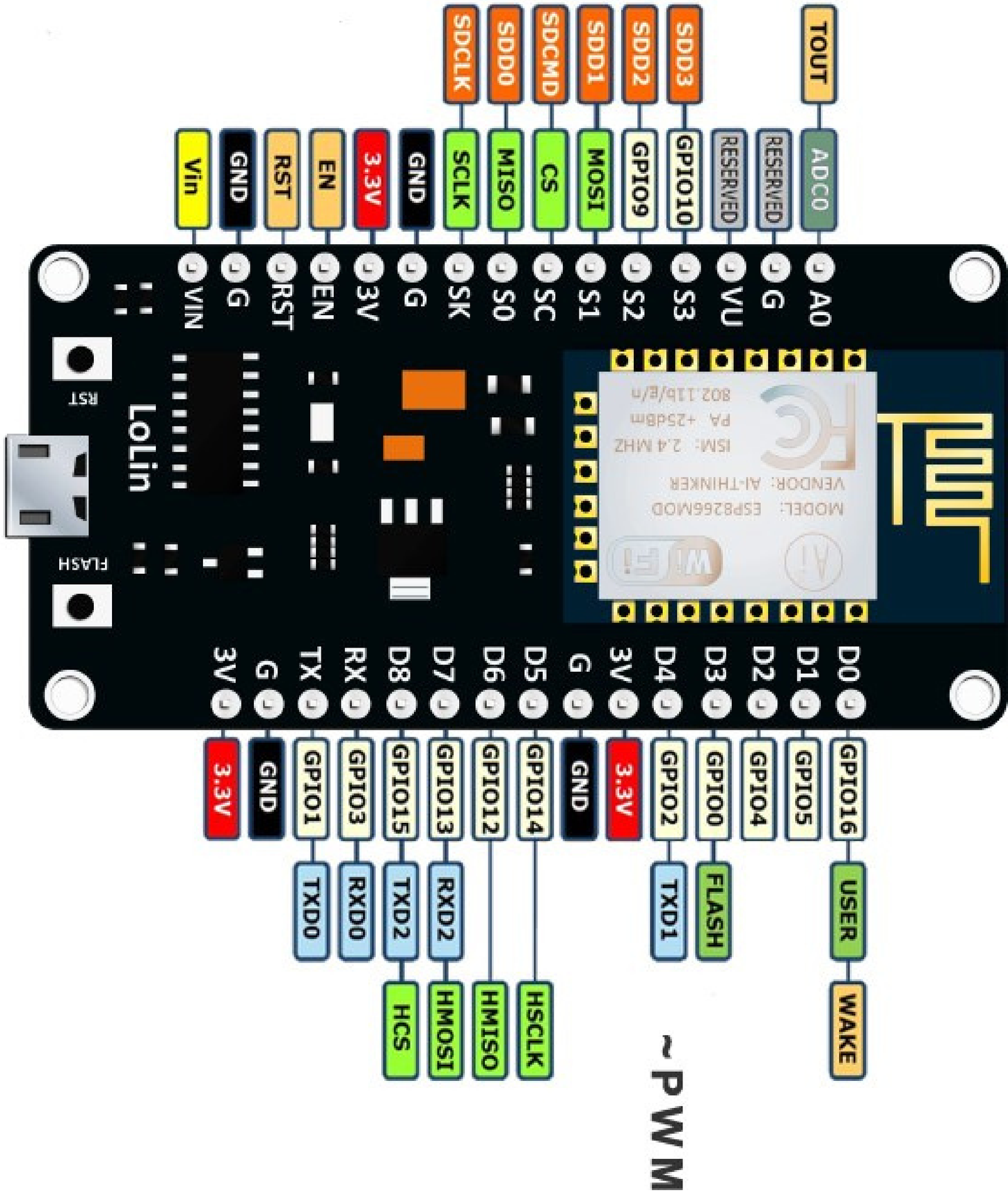
PROGRAMMING SECTION 3

Arduino programming with C++ on NodeMCU

Beyond Expectation



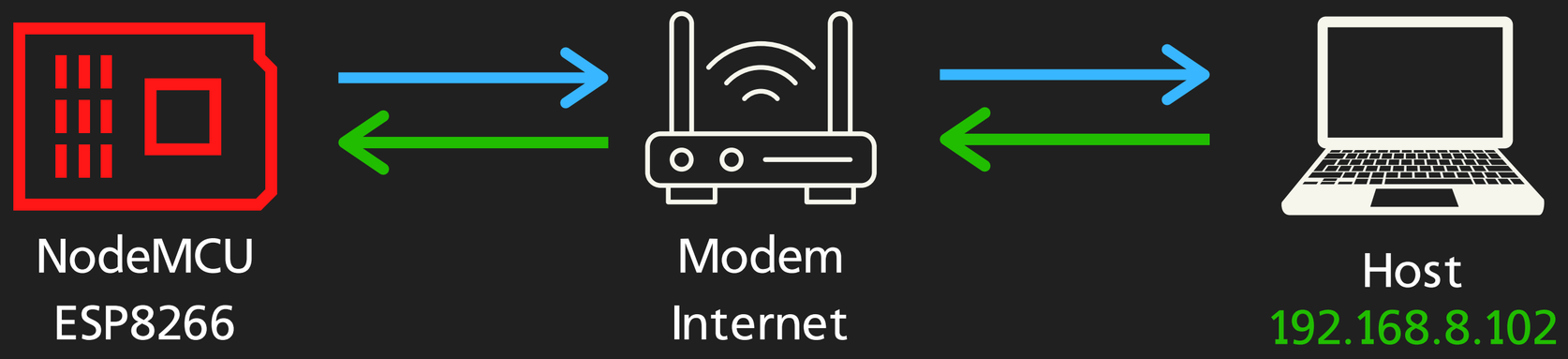
KNOW YOUR BOARD!



Arduino Programming Section 3

ESP8266 Wi-Fi

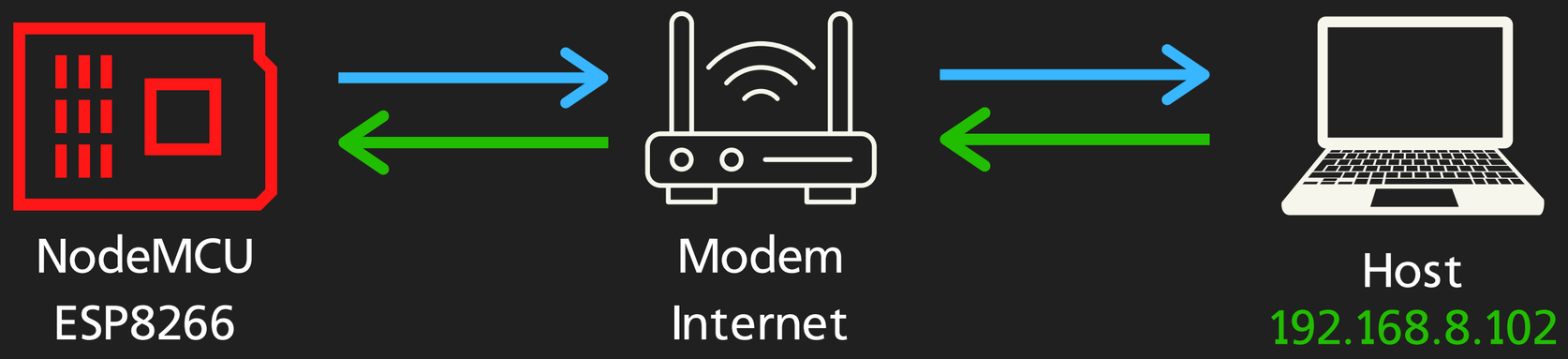
Connections



ESP8266 Wi-Fi

Salah satu kelebihan dari NodeMCU adalah memiliki kemampuan untuk melakukan koneksi dengan jaringan nirkabel (Wi-Fi). Dengan begitu, NodeMCU mampu menerima dan mengirim data melalui itu. Kemampuan ini juga bisa kita manfaatkan dalam hal yang berkaitan dengan kontrol jarak jauh, monitoring dll.

Connections



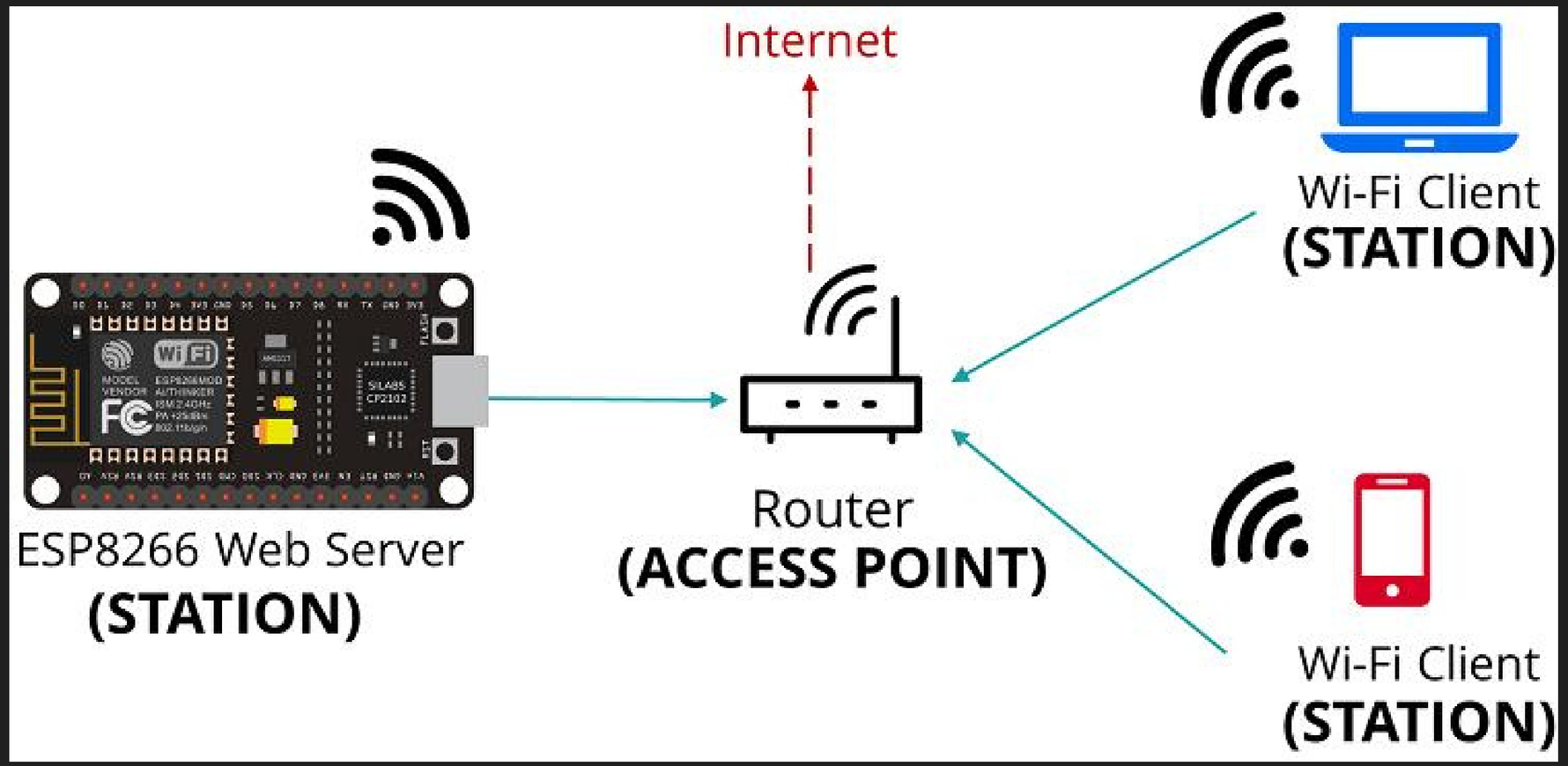
ESP8266 Wi-Fi

Untuk bisa menyambungkan NodeMCU dengan internet, kita akan menggunakan libray sebagai penyedia service atau layanan di dalam kode yang akan dibuat. NodeMCU dikatakan berhasil terkoneksi dengan jaringan, jika nodemcu sudah mendapatkan alamat IP (Index Protocol).

Connections

ESP8266 Wi-Fi

ESP8266 umumnya difungsikan sebagai station.
Maksudnya, board akan menjadi client dari sebuah access point (Router). Dan ESP8266 mendapatkan IP dari si router.

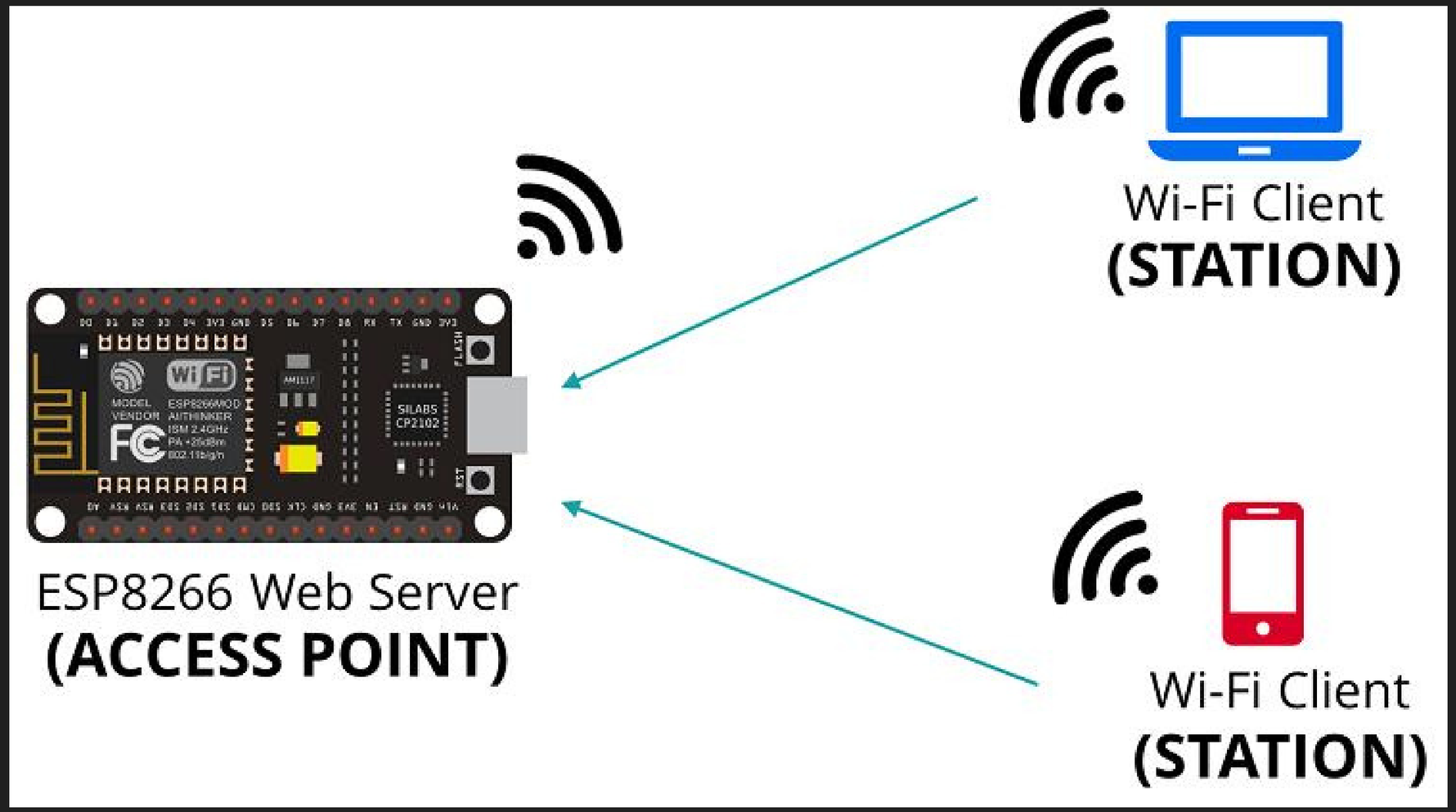


Connections

ESP8266 Wi-Fi

Adapun ESP8266 bisa difungsikan sebagai access point (Router) untuk device di sektiarnya.

ESP8266 bisa membuat sebuah SSID yang nantinya bisa kita gunakan untuk tersambung dengan device lainnya.



Arduino Programming Section 2

Coding Time

Coding Time

Code 7: Koneksi ESP8266 ke Access Point

Source code link:

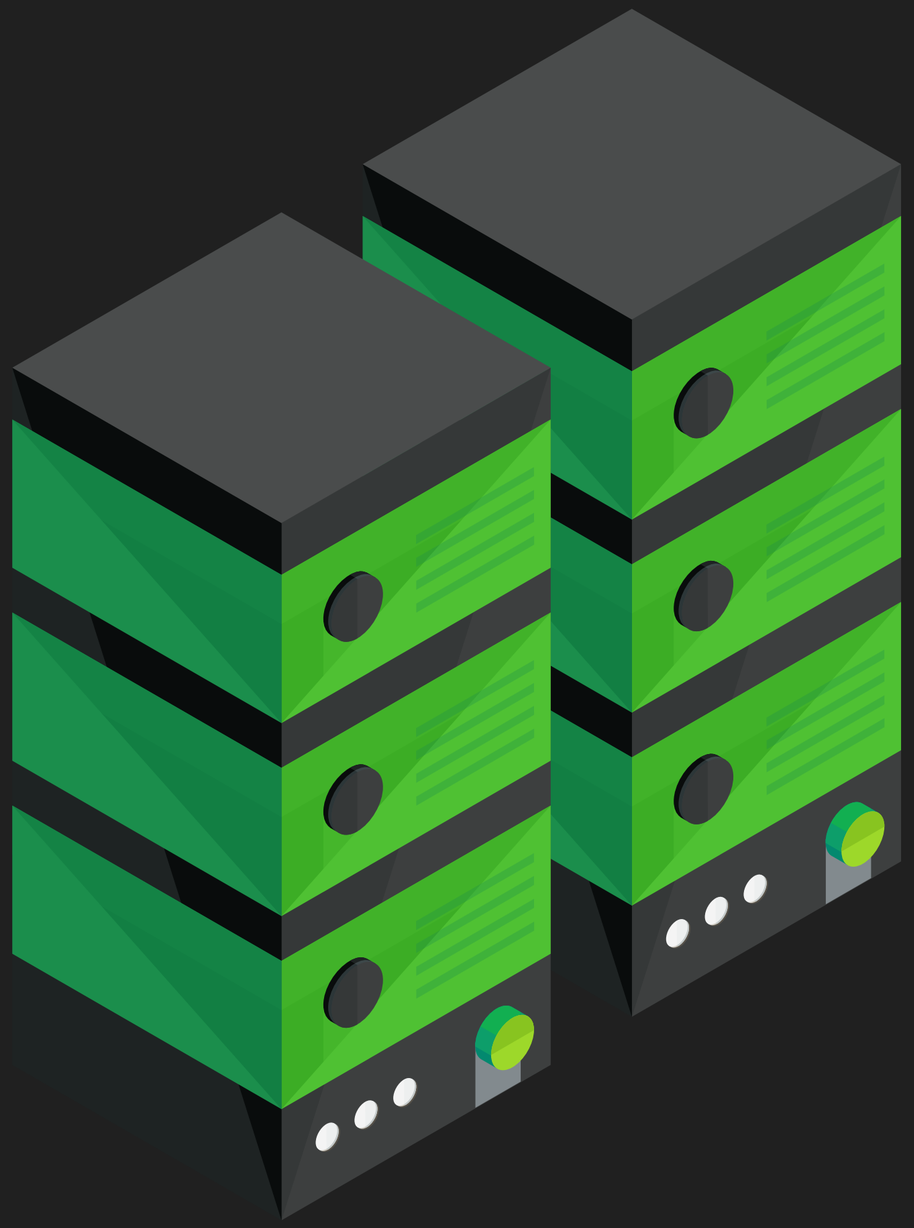
https://github.com/codersidprogrammer/xception/blob/master/ESP-WifiClient.cpp

Arduino Programming Section 3

ESP8266 Server

ESP8266 Server

ESP8266 dapat digunakan sebagai web server sederhana. Hal ini berguna ketika kita ingin membuat sebuah antarmuka sederhana dari hal sedang kita kontrol



ESP8266 Server

Dengan menggunakan Library yang sudah tersedia dari ESP, kita bisa menggunakan salah satu functionnya, yaitu softAP.

Jika dijalankan, ESP akan berfungsi sebagai Access Point dengan SSID dan password yang sudah ditentukan

```
const char* ssid     = "ESP8266-Access-Point";
const char* password = "123456789";

// Setting ESP8266 sebagai access point
WiFi.softAP(ssid, password);
```

ESP8266 Server

Dari cuplikan ini, kita akan mencari tau alamat IP yang difungsikan sebagai Access Point. Dan dicetak ke dalam layar serial

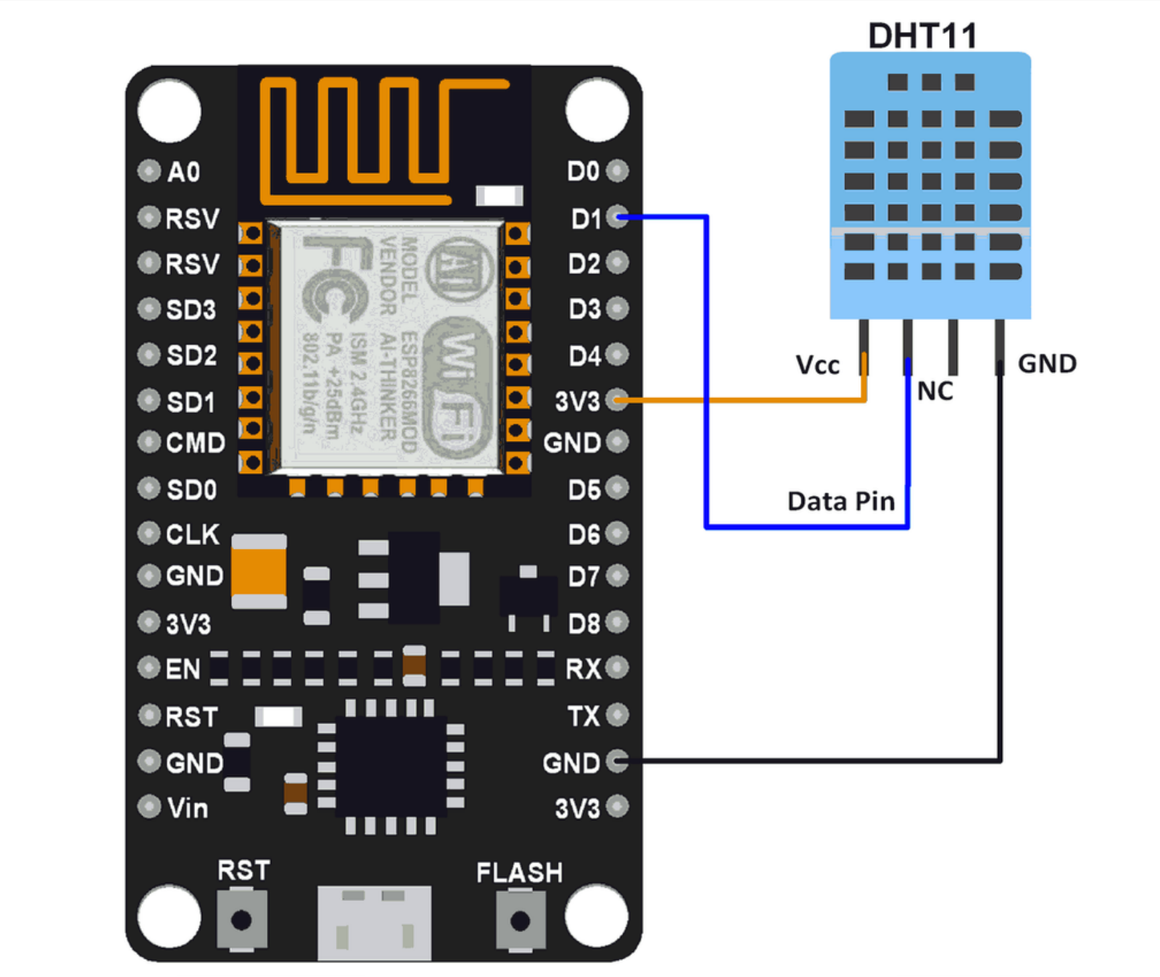
```
IPAddress IP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(IP);
```

Arduino Programming Section 2

Coding Time

Coding Time

Code 8: Membuat webserver ESP8266



Source code link:

https://github.com/codersidprogrammer/xception/blob/master/ESP-Server.cpp

Arduino Programming Section 3

ESP8266 & MQTT

Apa itu MQTT?

MQTT istilah dari Message Queuing Telemetry Transport. MQTT is a simple messaging protocol, didesai untuk koneksi yang low bandwidth. Ini menjadi pilihan yang tepat untuk multiple IoT devices.

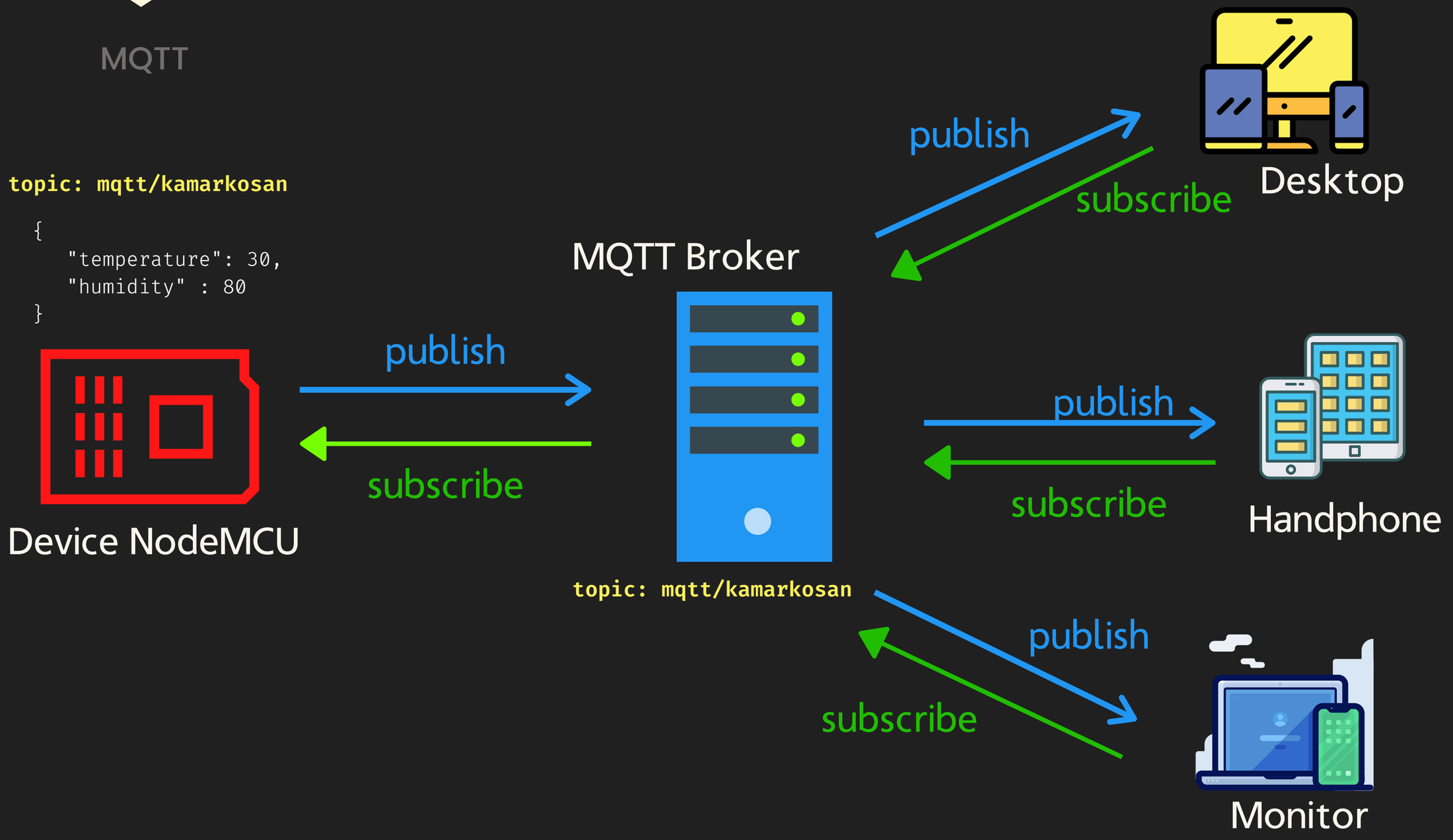
Kenapa MQTT?

Menekan ukuran paket data sekecil mungkin sehingga trafik bisa meningkat.

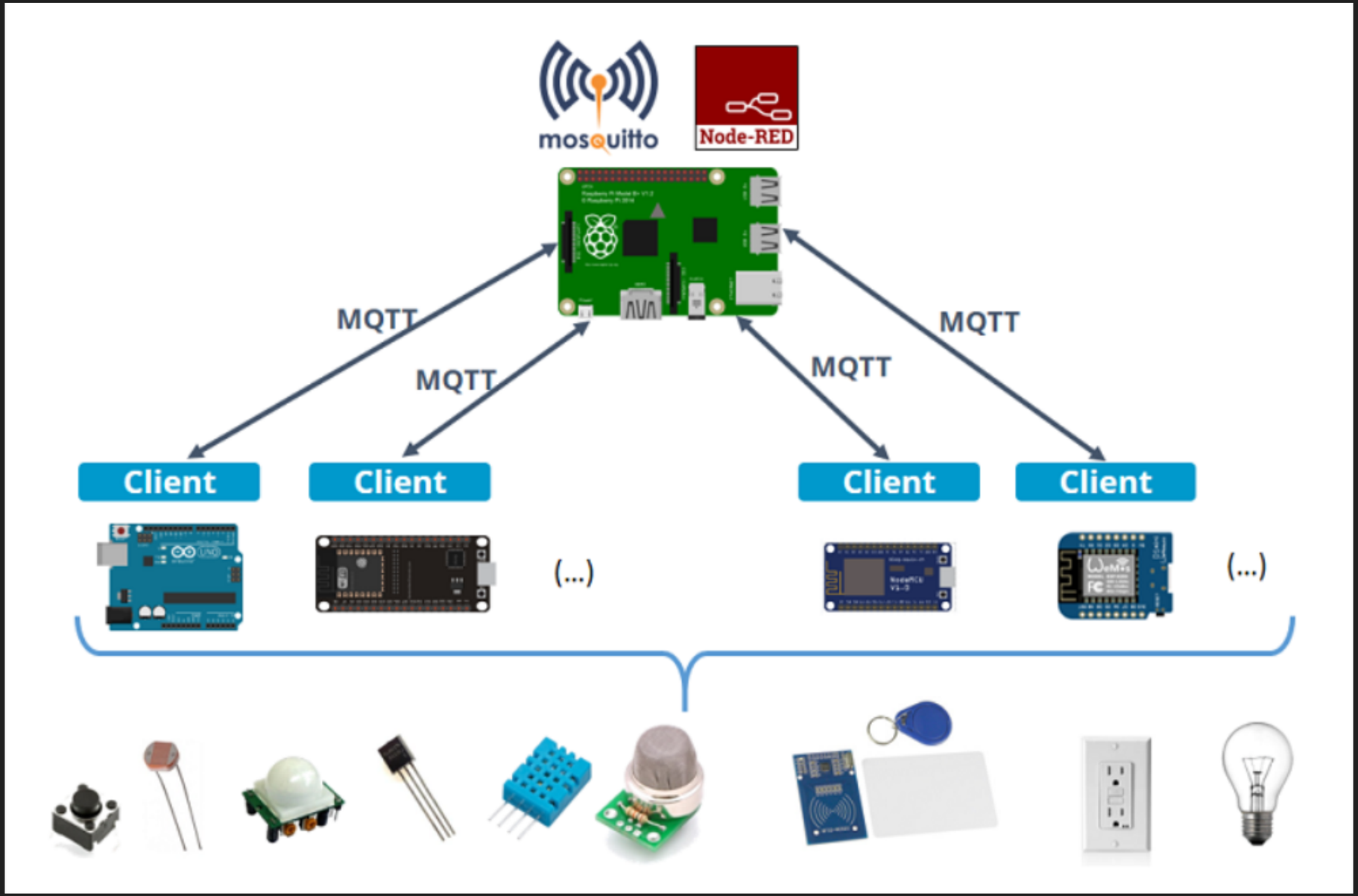
Meminimalisasi proses komputasi untuk encoding dan decoding dari paket data.

Data hanya menggunakan ruang **penyimpanan yang sekecil mungkin**.

MQTT


topic: mqtt/kamarkosan
{
"temperature": 30,
"humidity" : 80
}
publish
subscribe
Device NodeMCU
MQTT Broker
topic: mqtt/kamarkosan
publish
subscribe
Desktop
publish
subscribe
Handphone
publish
subscribe
Monitor

Connections



Arduino Programming Section 2

Coding Time

Coding Time

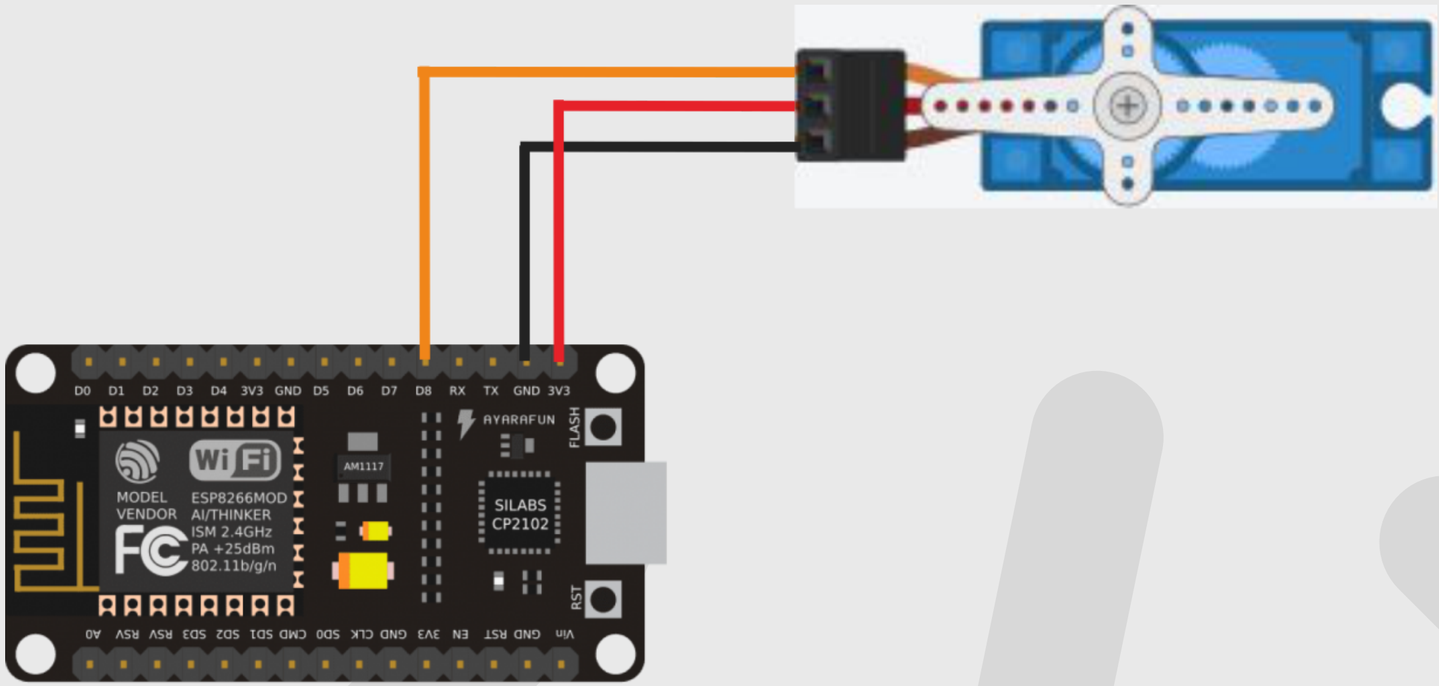
Code 9: Kendali LED Jarak Jauh dengan MQTT

Source code link:

https://github.com/codersidprogrammer/xception/blob/master/ESP-Mqtt.cpp

Coding Time

Code 9: Kendali SERVOJarak Jauh dengan MQTT



Source code link:

https://github.com/codersidprogrammer/xception/blob/master/ESP-Mqtt-2.cpp

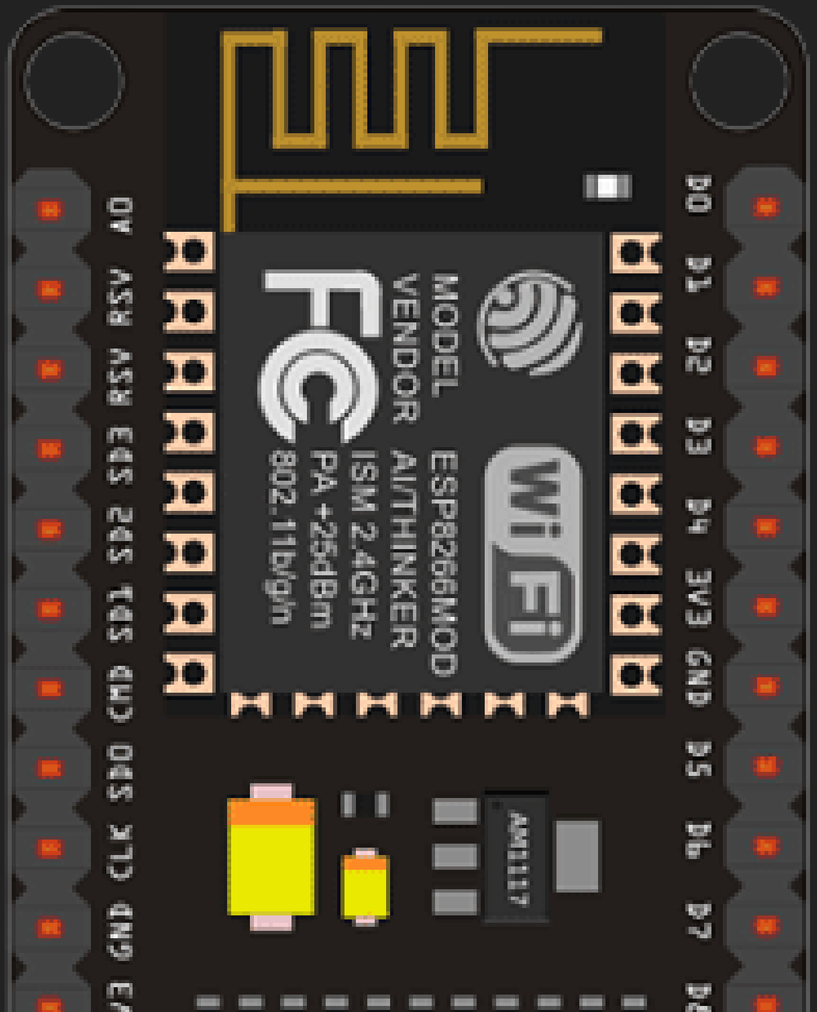
Arduino

NodeMCU

C++

ARDUINO PROGRAMMING

DANKE SCHÖN!

Arduino programming with C++ on NodeMCU

Beyond Expectation