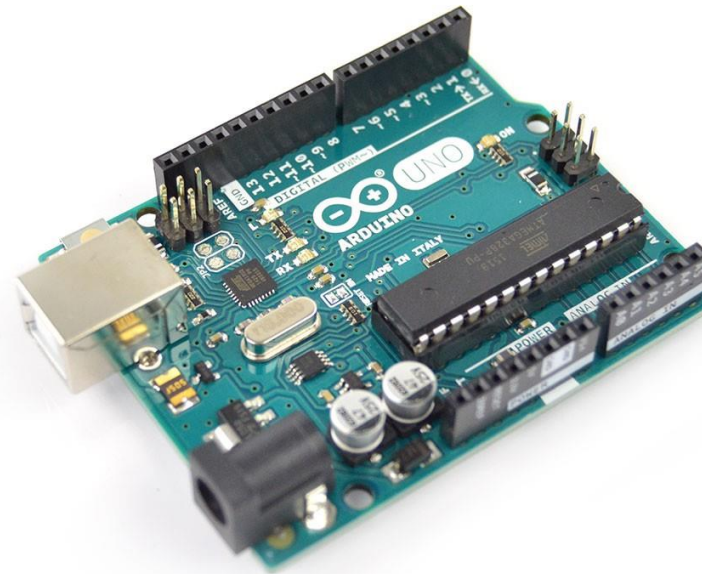


Microcontroller Unit

Lecturer : Putut Dewantoro (13214065)

Electrical Engineering, SEEI ITB

What is Microcontroller Unit ?



Microcontroller

Easy to learn

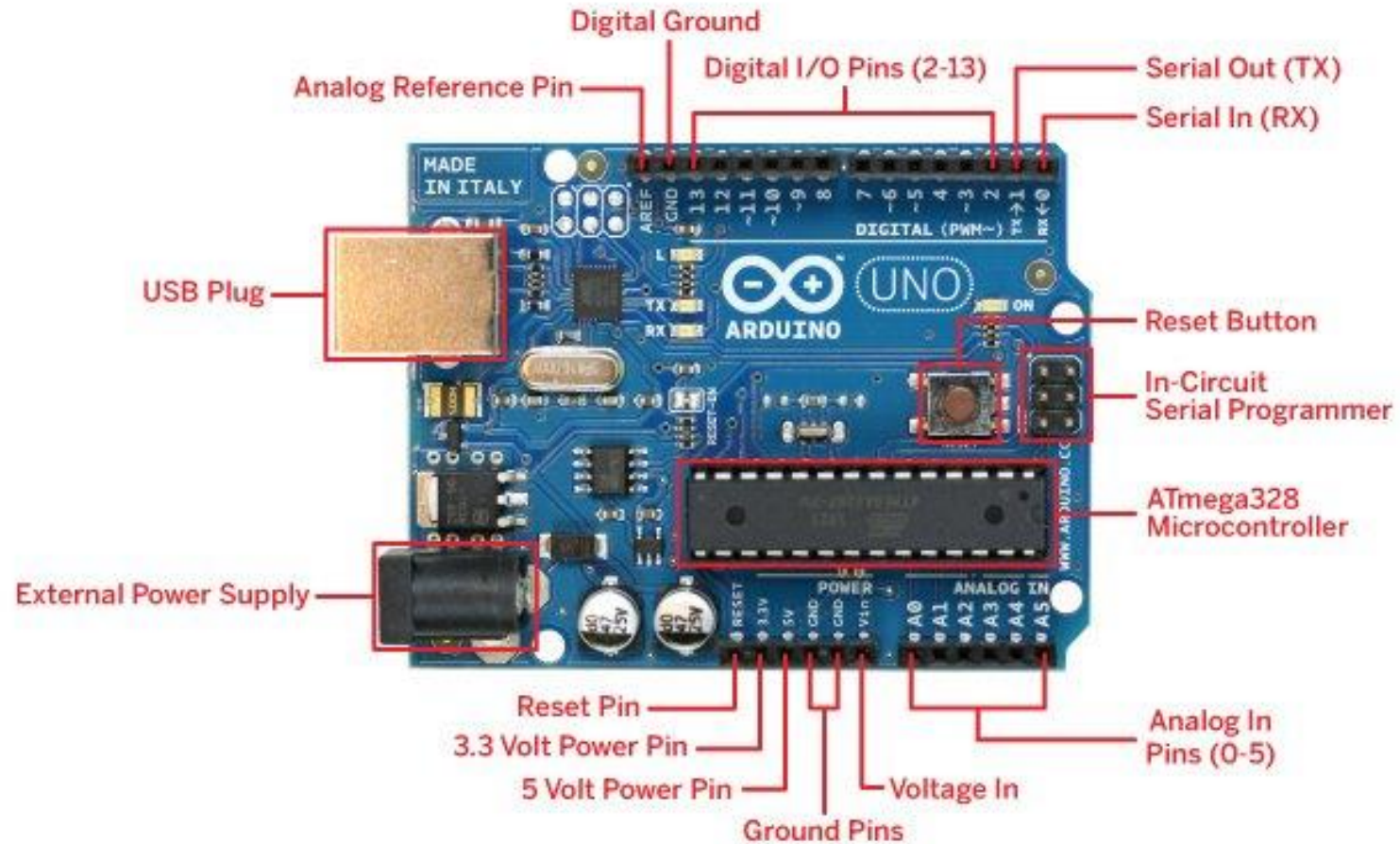
- Easy to use
- Cheap
- Time Limited
- Sequence Execution

Arduino

Semuanya berawal dari sebuah thesis yang dibuat oleh Hernando Barragan, di institute Ivrea, Italia pada tahun 2005, dikembangkan oleh Massimo Banzi dan David Cuartielles dan diberi nama Arduin of Ivrea. Lalu diganti nama menjadi Arduino yang dalam bahasa Italia berarti teman yang berani.

Tujuan awal dibuat Arduino adalah untuk membuat perangkat mudah dan murah, dari perangkat yang ada saat itu. Dan perangkat tersebut ditujukan untuk para siswa yang akan membuat perangkat desain dan interaksi. Visi awalnya aja udah mulia kan.

Arduino




Arduino



- Digital I/O Pin = untuk input dan output digital [jika ada tanda (~) bisa untuk output analog]
- Analog in = untuk input analog
- VCC, GND = as usual
- 5 / 3.3 volt power pin = Mengeluarkan output 5 / 3.3 volt
- Eksternal power supply = untuk supply dari eksternal (bukan dari PC/laptop)
- USB plug = untuk konek ke PC / laptop
- Serial out/in (Rx/Tx) = Serial Communication

Example

The image shows a simplified representation of the Arduino IDE interface. It has a dark teal header bar with five icons: a checkmark, a right-pointing arrow, a document with a grid, an upward-pointing arrow, and a downward-pointing arrow. Below the header is a teal bar with a white text box containing the filename "Blink_LED".

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(2, OUTPUT); //set pin 2 sebagai Output  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(2,HIGH); //set pin 2 bernilai 1 [LED Nyala]  
  delay(1000); //delay 1 detik  
  digitalWrite(2,LOW); // set pin 2 bernilai 0 [LED Mati]  
  delay(1000); //delay 1 detik  
}
```

Blink

```
void setup() { //set up code untuk run sekali  
  pinMode(2, OUTPUT); //set pin 2 sebagai output  
}
```

```
void loop() { // set up code untuk run repeatedly  
  digitalWrite(2, HIGH); // menulis pin 2 as HIGH  
  delay(1000); //Delay selama 1 detik  
  digitalWrite(2, LOW); // menulis pin 2 as LOW  
  delay(1000);  
}
```


Example

```
✓ → 📄 ⬆ ⬇ Verify
Switch_LED

int pushButton = 2;
int LED = 3;

void setup() {
  // put your setup code here, to run once:
  pinMode(pushButton, INPUT);
  pinMode(LED, OUTPUT);
}

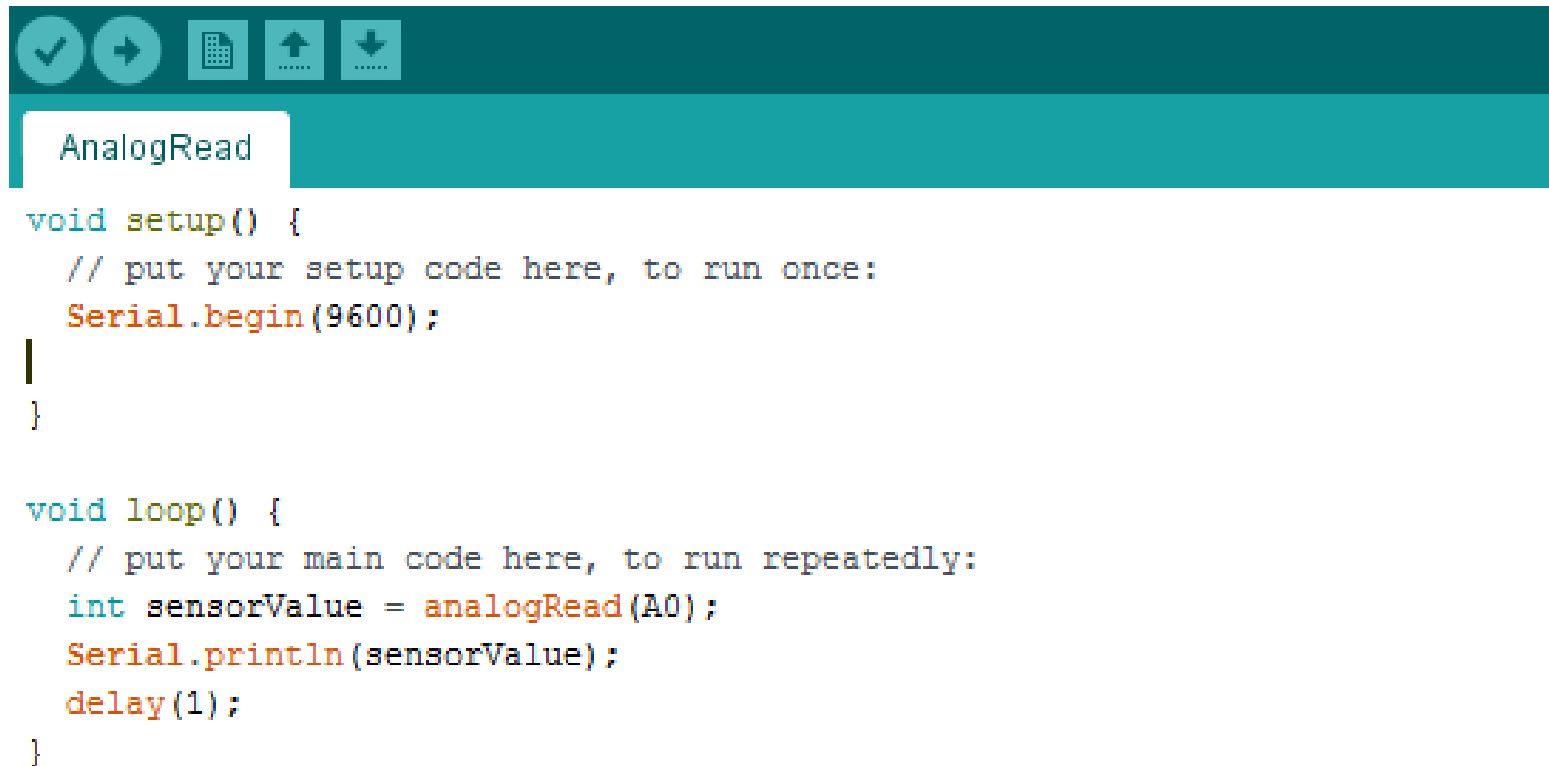
void loop() {
  // put your main code here, to run repeatedly:
  int buttonState = digitalRead(pushButton);
  if (buttonState == 1) {
    digitalWrite(2, HIGH);
    delay(1000);
    digitalWrite(2, LOW);
    delay(1000);
  }
}
```

Digital Read

```
int buttonState = digitalRead(pushButton); //Membaca Input dari button
```

```
if (buttonState == 1) {  
    digitalWrite(2,HIGH); // mengeluarkan output high  
    delay(1000);  
    digitalWrite(2,LOW); // mengeluarkan output low  
    delay(1000);  
}
```

Example

The image shows a screenshot of the Arduino IDE interface. At the top, there is a dark teal toolbar with icons for checking, running, serial monitor, and file operations. Below the toolbar, a tab labeled "AnalogRead" is active. The main workspace contains C++ code for an Arduino sketch. The code defines a setup function and a loop function. The setup function initializes the serial port at 9600 baud. The loop function reads an analog value from pin A0, prints it to the serial monitor, and delays for 1 second before repeating.

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  int sensorValue = analogRead(A0);  
  Serial.println(sensorValue);  
  delay(1);  
}
```

PWM (Fade)

```
int led = 10; int brightness = 0; int fadeAmount = 5;
```

```
void setup() {  
  pinMode(led, OUTPUT);  
}
```

```
void loop() {  
  analogWrite(led, brightness); //mengeluarkan output berupa analog  
  brightness = brightness + fadeAmount; //menambahkan brightnes sejumlah fade amount  
  if(brightness <= 0 || brightness >=255){ // kondisi merubah nyala-redup/ redup-nyala  
    fadeAmount = - fadeAmount;  
  }  
  delay(25); // akan menambah brightness setiap 25ms  
}
```

Example

The image shows a screenshot of the Arduino IDE interface. At the top, there is a dark teal toolbar with icons for a checkmark, a right arrow, a document, an upload arrow, and a download arrow. Below the toolbar is a teal header bar with a white tab labeled "Fade". The main area of the IDE is white and contains the following C++ code:

```
int led = 10;
int brightness = 0;
int fadeAmount = 5;

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

void loop() {
  // put your main code here, to run repeatedly:
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if(brightness <= 0 || brightness >=255){
    fadeAmount = - fadeAmount;
  }
  delay(25);
}
```

Analog Read

```
void setup() {  
  Serial.begin(9600); //set ke serial monitor  
}
```

```
void loop() {  
  int sensorValue = analogRead(A0); //Read analog dari pin 0 dan masuk ke variabel sensor value  
  Serial.println(sensorValue); //print hasil yang dibaca sensorValue ke monitor  
  delay(1);  
}
```

*lihat serial monitor di tab **tools -> serial monitor** atau **gambar kaca pembesar di kanan atas**

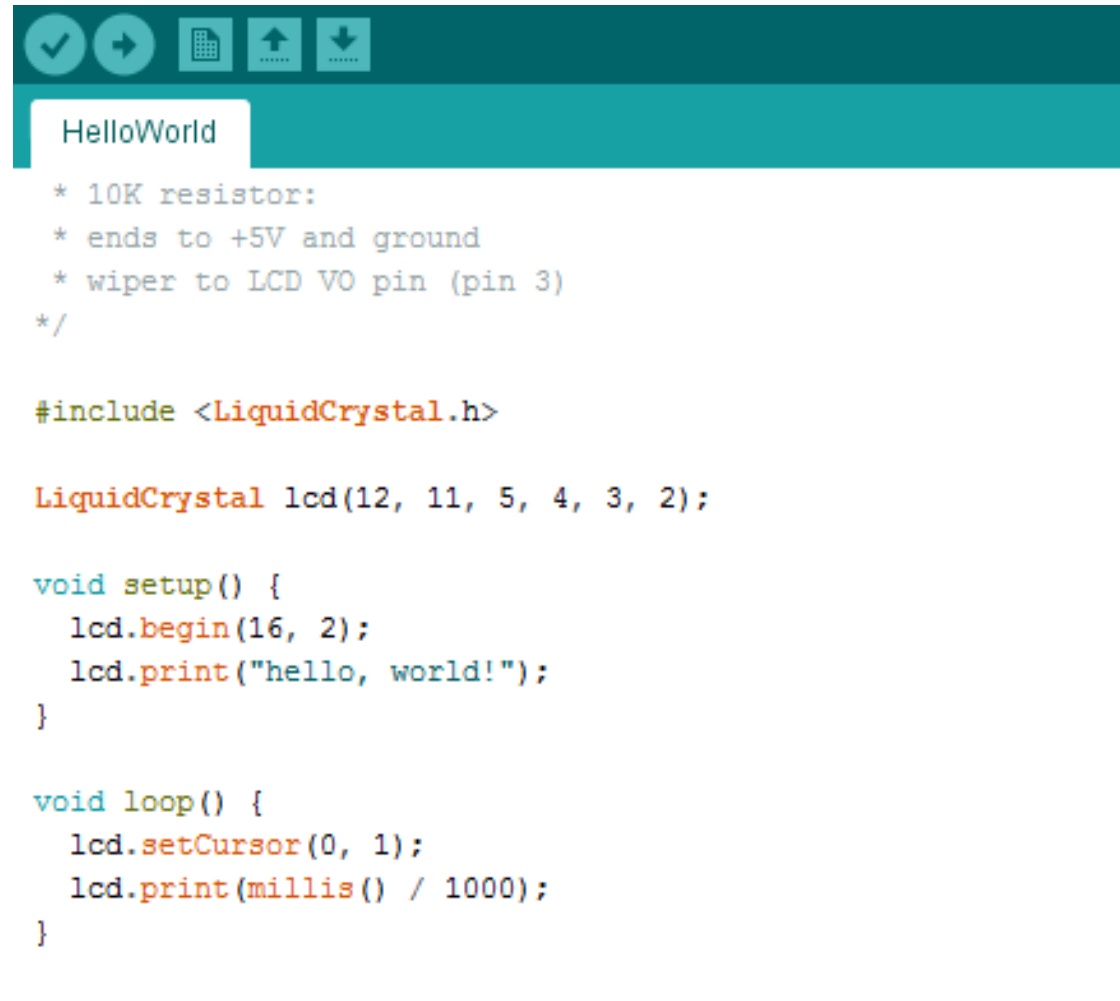
Libraries

- Tadi telah dipelajari beberapa fungsi *built-in* dari bahasa yang dipakai Arduino. Namun, bagaimana jika kita ingin meng-*extend* skill dari Arduino kita?
- Misal, mau pakai LCD? Mau mainin mp3 files? Mau memanfaatkan SD card?
- *One solution* adalah dengan membuat kode nya sendiri dari *scratch*
- *The other solution*, yang lebih preferable untuk seorang user Arduino ialah meng-equip Arduino mu dengan item yang dinamakan *Library*

What are Libraries?

- Intinya, *Library* adalah sekumpulan kode-kode yang dibuat oleh orang-orang yang jenius dan rajin yang membuat hidup pengguna Arduino lebih efisien dan harmonis
- Umumnya, *Library* berisi kumpulan kode yang mempermudah urusan *interface* dengan *display*, *sensor*, *actuator*, dsb.
- Contoh-contoh *Library* adalah *LiquidCrystal*, *SD*, *Servo*, *TMRpcm*, *AltSoftSerial*, dsb.
- Ok, *enough chit chat*, *lets write some code!*

LCD Library Hello World Example

The image shows a screenshot of the Arduino IDE interface. At the top, there is a dark teal toolbar with icons for checking, running, saving, and other functions. Below the toolbar, a tab labeled "HelloWorld" is active. The main area displays the following C++ code:

```
* 10K resistor:
* ends to +5V and ground
* wiper to LCD VO pin (pin 3)
*/

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}

void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis() / 1000);
}
```