Hardware II Seminar

Sensors and Data Analysis

Back To Basic session: Electronics

FAB LAB BCN

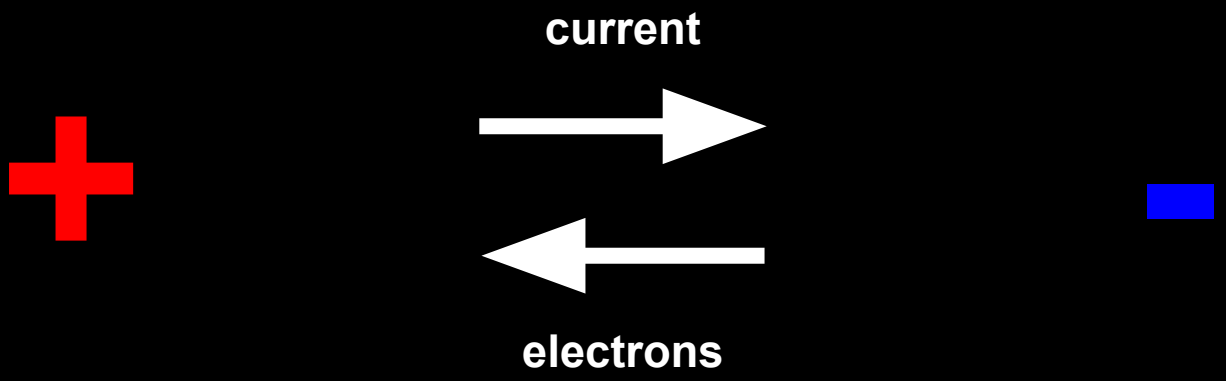
Iaac

Óscar González & Daniil Koshelyuk

Hardware

Electronics

Hardware: electronics



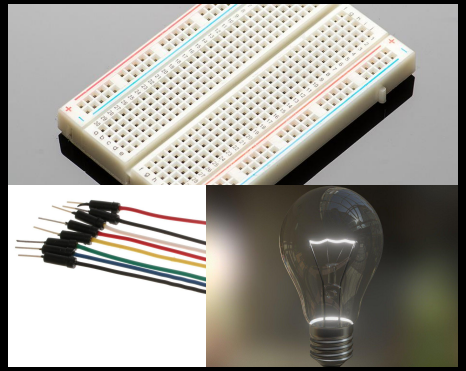
Hardware: electronics

$$V = I * R$$

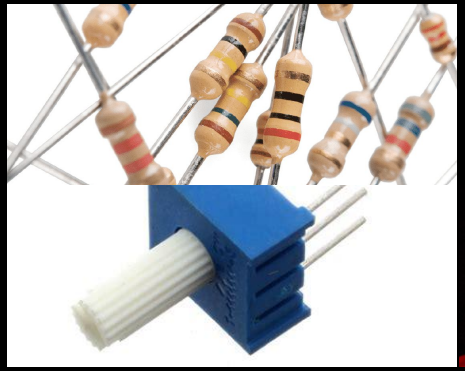
V	-	Voltage	in volts
I	-	Current	in amps
R	-	Resistance	in ohms

Hardware: electronics

Wire

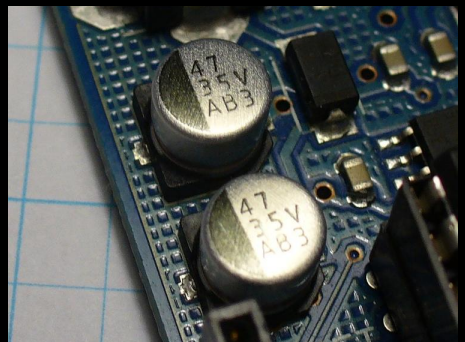


Resistor

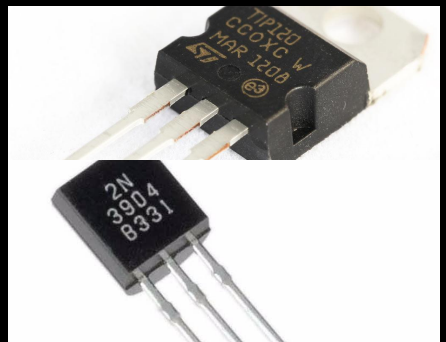


Diode

Capacitor



Transistor

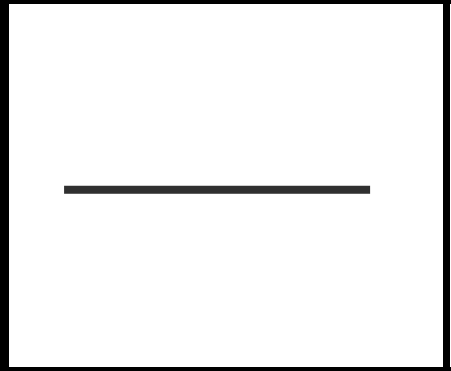


FAB LAB BCN

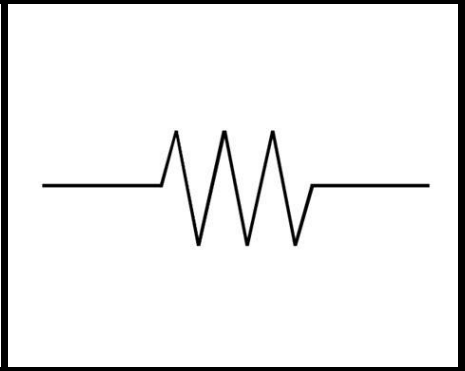
Iaac

Hardware: electronics

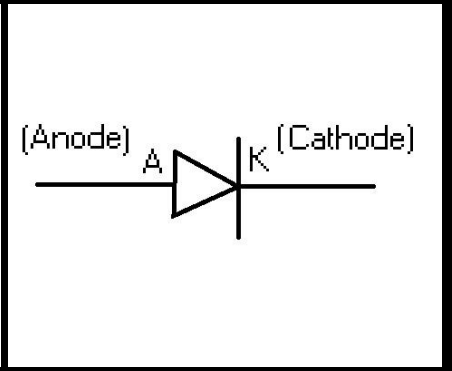
Wire



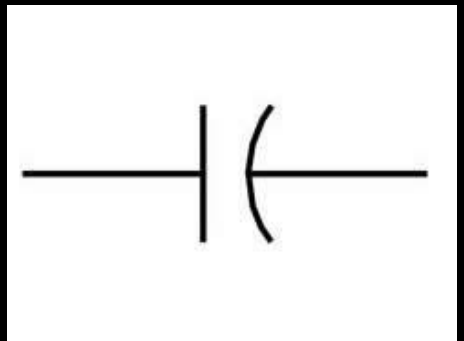
Resistor



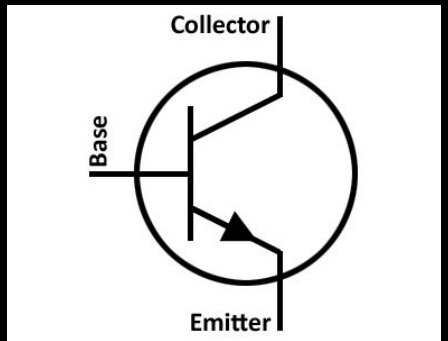
Diode



Capacitor



Transistor

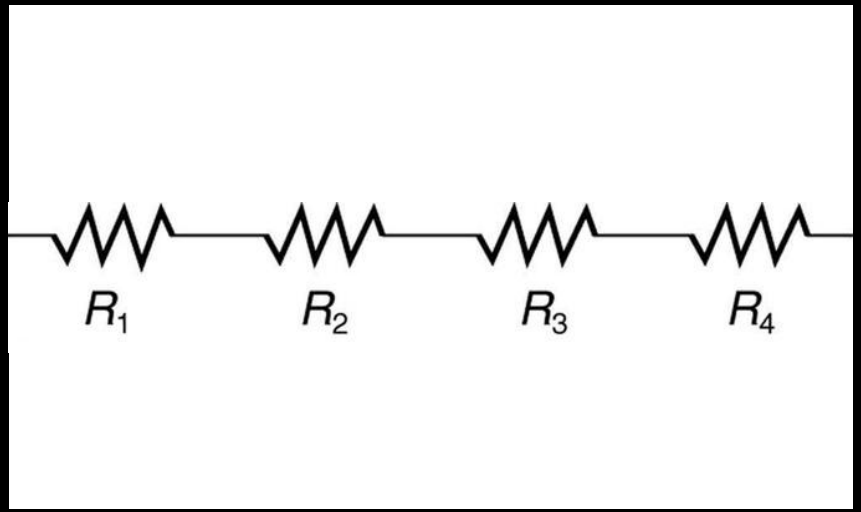


FAB LAB BCN

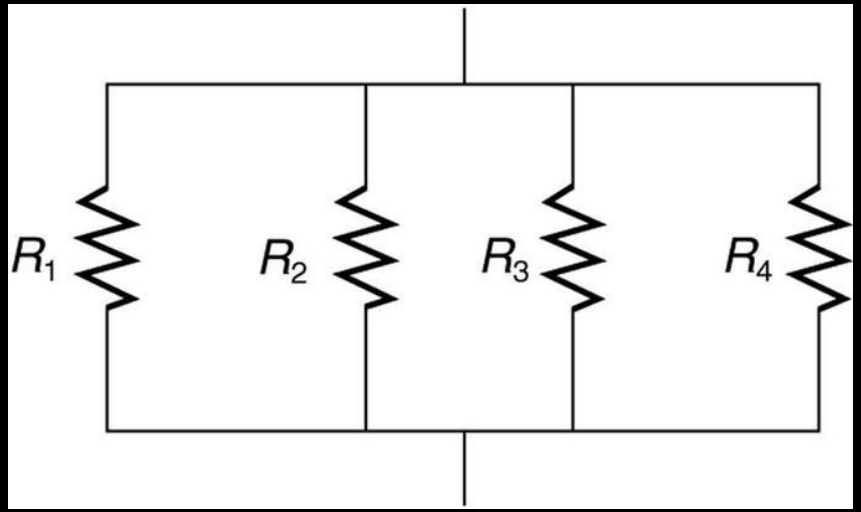
Iaac

Hardware: electronics

Series

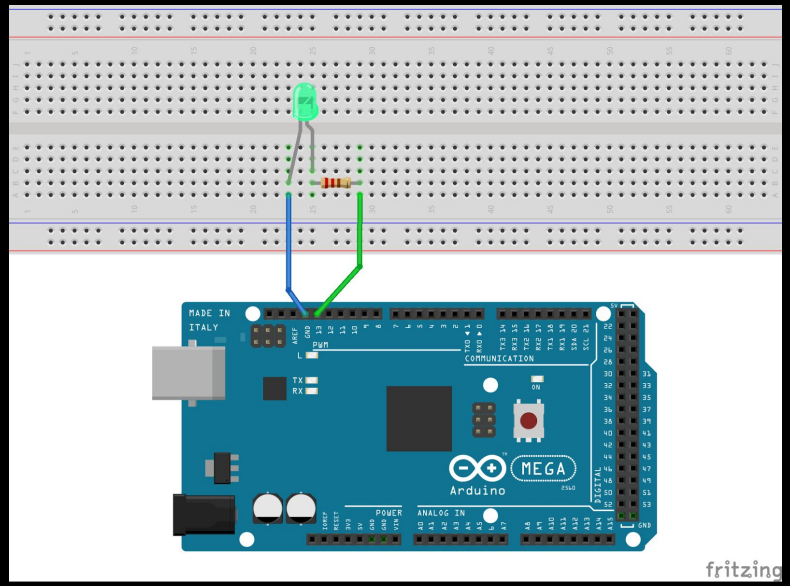


Parallel



Hardware: electronics

Limiting Forward Current of 20mA
Max Forward Voltage of 2.2 volts



$R = V / I$

$= (V_{sourve} - V_{led}) / I_{led}$

$= (5V - 2.2V) / 20mA$

$= 2.8V / 0.02A$

$= 140\ Ohm$

Source Voltage of 5V

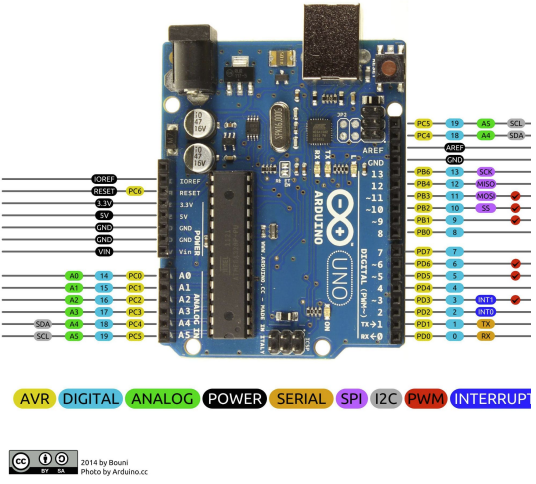
FAB LAB BCN

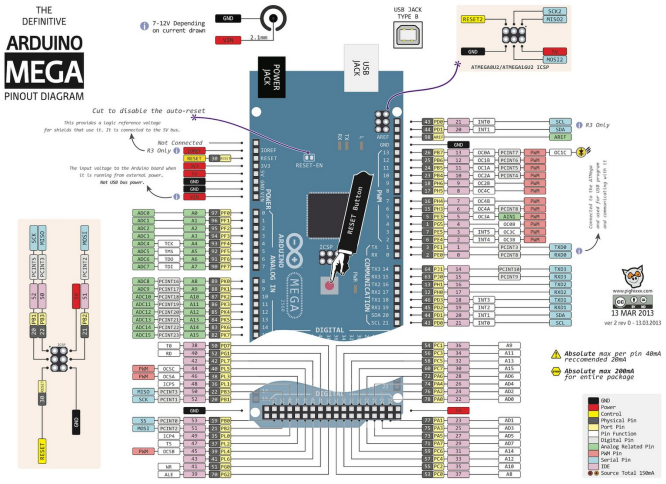
Iaac

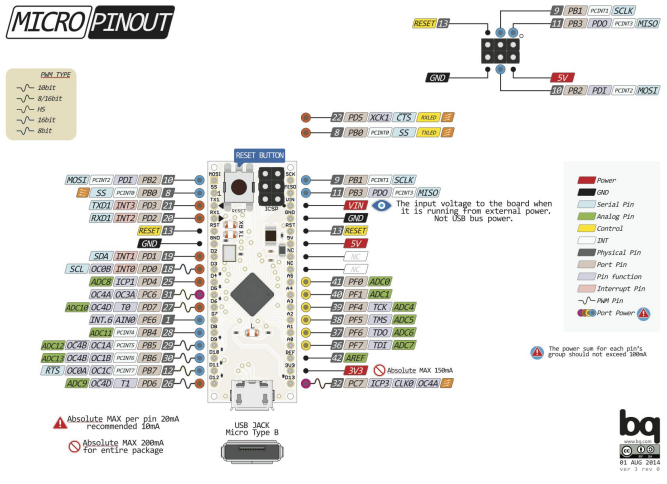
Hardware

Pins

Hardware: pins



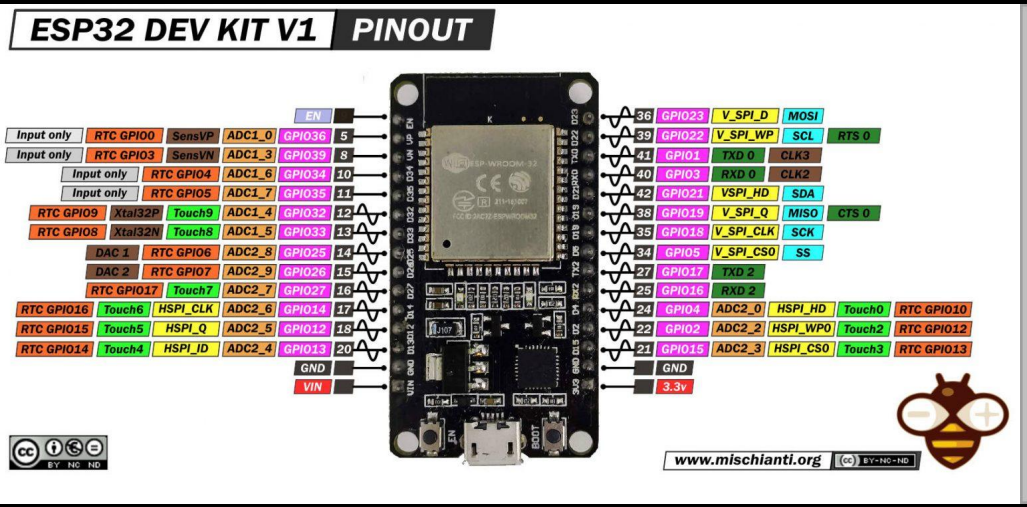


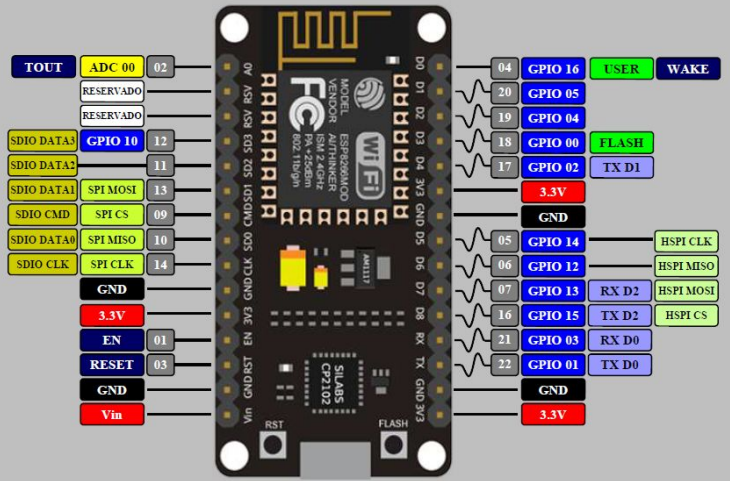


FAB LAB BCN

Iaac

Hardware: pins

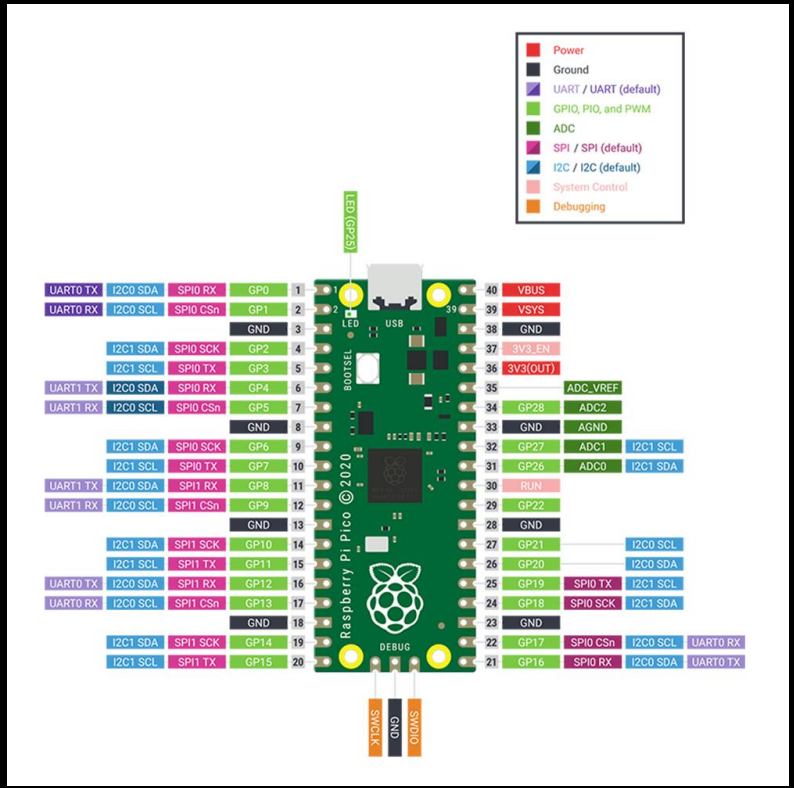




FAB LAB BCN

Iaac

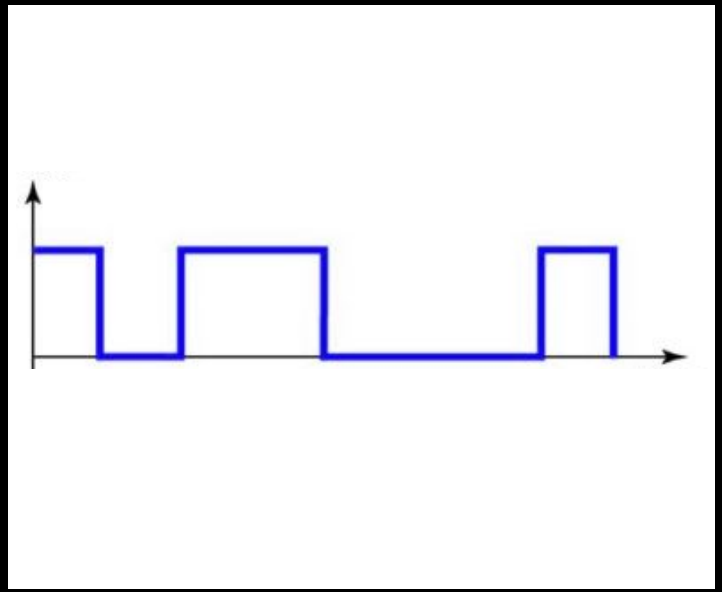
Hardware: pins

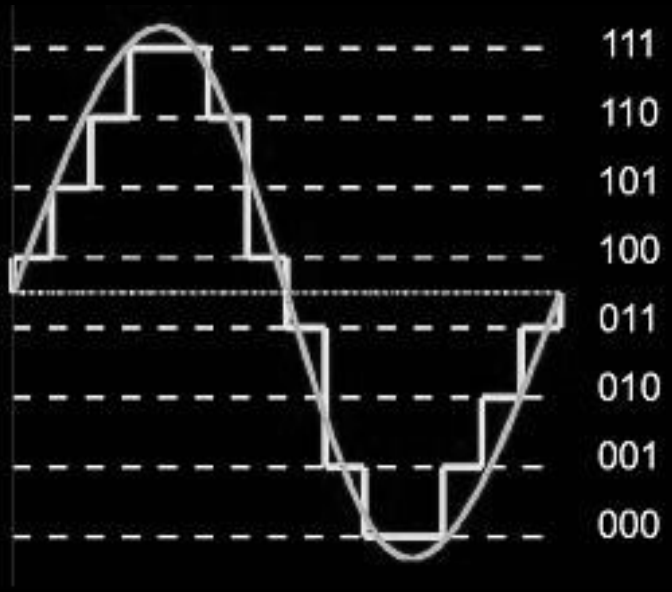


FAB LAB BCN

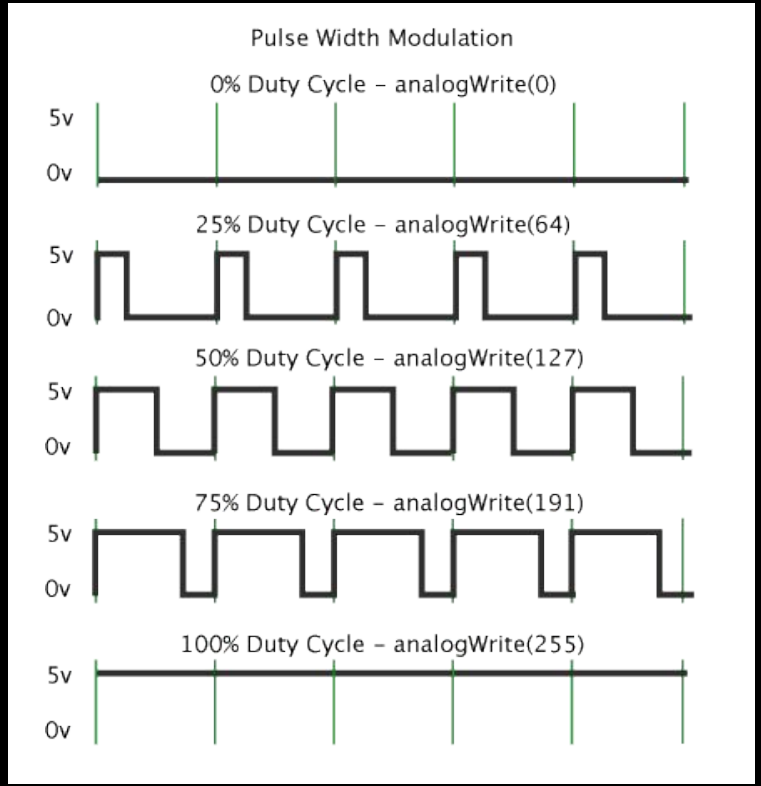
Iaac

Hardware: Simple Communication: digital vs analog



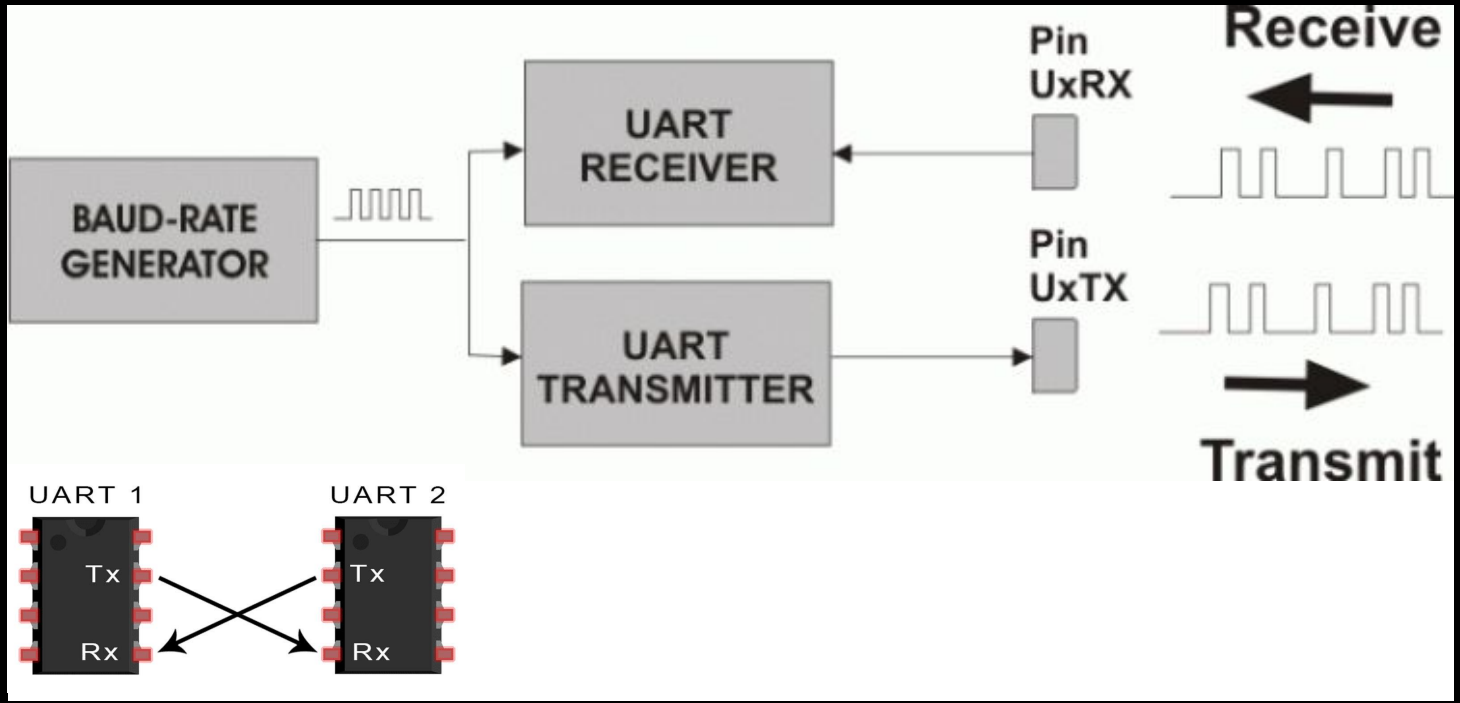


Hardware: Simple Communication: PWM

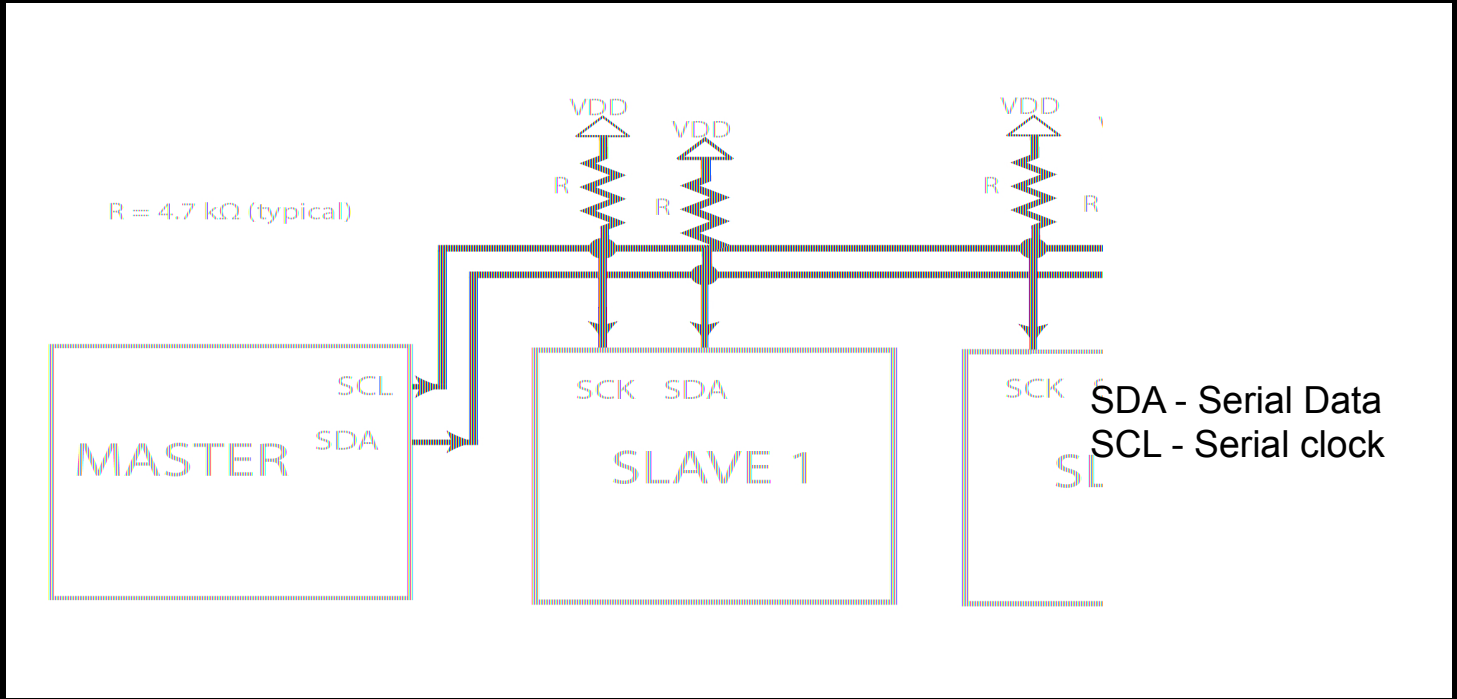


https://www.arduino.cc/en/pmwiki.php?n=Tutorial/PWM

Hardware: Advanced Communication: Serial (UART), I2C, SPI



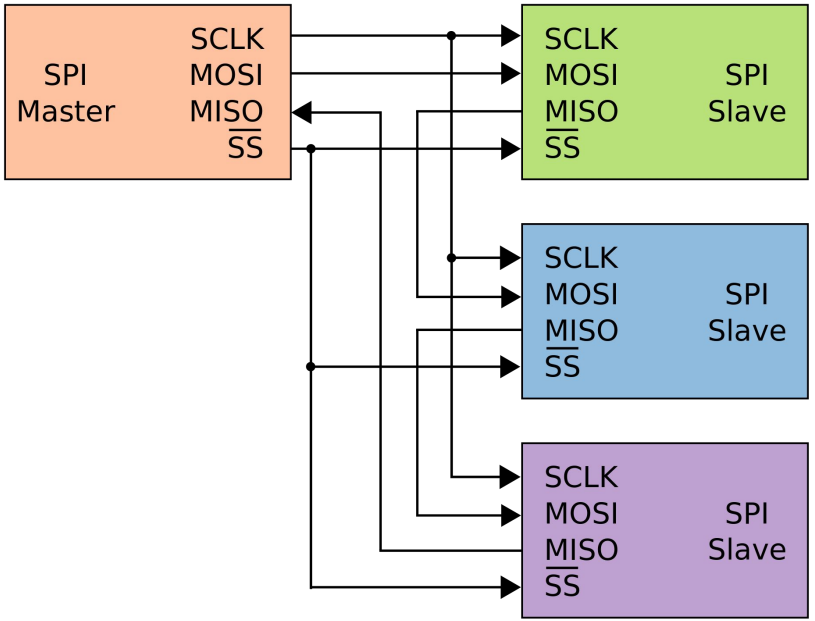
Hardware: Advanced Communication: Serial (UART), I2C, SPI



FAB LAB BCN

Iaac

Hardware: Advanced Communication: Serial (UART), I2C, SPI



SCLK – clock signal, generated by the master device

MOSI – Master Data Output, Slave Data Input

MISO – master data input, slave data output

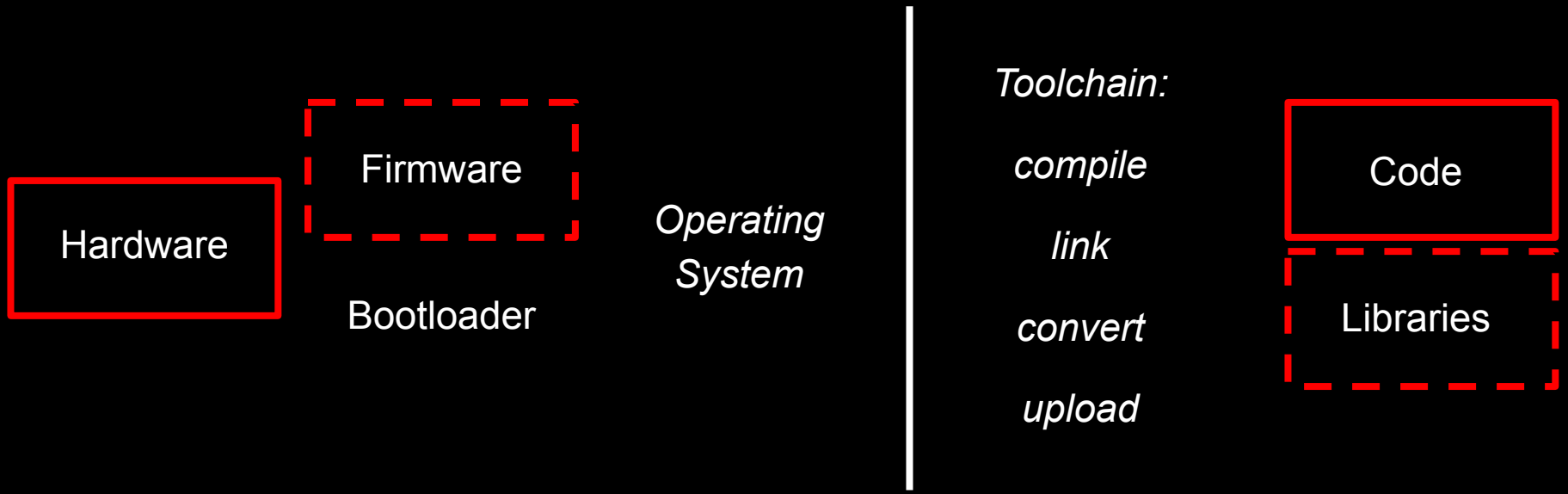
SS – Slave enabled signal, controlled by the master device

Hardware: pins

1. Do you need analog or digital?
2. Do you need PWM?
3. Is pin used for other tasks?
4. Do you need SPI, I2C? (read more on protocols here).

Software

Software: working with boards



Software: languages

C++

Arduino

Python

Micropython

Software: Pseudo-Code

Read the sensor value

Convert to units of measurement

Check if the value passes above threshold

    Send signal to other board to react

    Turn on an LED

Otherwise turn off the LED

Software: IDE

Main functions:

1. Write code
2. Manage libraries, data, files, etc.
3. Interface with target device (integrate toolchain, tools to customize an monitor, etc.)

Software: languages: comments and variables

C++

```c++
// This is not a code, but it helps humans to read.
/*
      This is a multiline comment.
*/

int pinNumber = 5;  // define a variable of type int
                    // and store a value.

float sensorValue = 20.4;  // numbers with floating point.

bool ledState = true;      // true or false for
                           // binary states.

int ledPins[] = {2,4,5,6}; // a grouped set of values.

char message[10] = “hello” // a text variable.
```

Python

```python
# This is not a code, but it helps humans to read.



pinNumber = 5    # create a variable and store a value.


sensorValue = 20.4  # numbers with floating point.

ledState = True     # True or False for
                    # binary states.

ledPins = [2,4,5,6] # a grouped set of values.

message = “hello”   # a text variable.
```

FAB LAB BCN

Iaac

Software: languages: functions

C++

```c++
// Define a function.
int ConvertSensorValue(int rawValue){ // signature.
      // Surround by curly braces anything related to
      // the function .

      // Do stuff with a sensor value.
      int newValue = rawValue * 5;
      // Pass the result back.
      return newValue;
}
void DebugSensorValue(char message[], int rawValue){
      Serial.print(message);
      Serial.println(rawValue);
}
int rawValue = 6;
// Call the function with no return.
DebugSensorValue("Current value: ", rawValue);
// Call the function and receive the result.
int newValue = CovertSensorValue(rawValue);
// Any variables defined in a scope stay there.
```

Python

```python
# Define the function before you use it.
def ConvertSensorValue(rawValue): # signature.
      # Indent after colon anything related to
      # the function.

      # Do stuff with a sensor value.
      newValue = rawValue * 5
      # Pass the result back.
      return newValue

def DebugSensorValue(message, rawValue):
      print(message)


rawValue = 6;
# Call the function with no return.
DebugSensorValue("Current value: ", rawValue)
# Call the function and receive result.
newValue = convertSensorValue(rawValue)
# Any variables defined in a scope stay there.
```

FAB LAB BCN

Iaac

Software: languages: conditions and boolean logic

C++

```c++
// Put condition in parentesis and follow
// with a block of code in curly braces.
if (sensor1Value > 5){ // One set of parentheses is required
      // Do stuff.
}

// Chain conditions - only first valid will execute.
if (sensor1Value == 5){
      // Do stuff if value is equal to 5.
} else if (sensor1Value <= 2){
      // Do stuff if value less than 2 inclusive.
} else{
      // Do stuff in all other cases.
}

// Boolean logic. && - and, || - or, ! - not.
if ((sernsor1Value > 5) && (sensor2Value < 3)){
      // Do stuff only if both evaluate to true.
} else if ((sensor1Value != 3) || !(sensor2Value > 5)){
      // Do stuff if either one evaluates to true.
}
```

Python

```python
# Put condition and follow with colon
# and indented block of code.
if sensor1Value > 5:       # Parentesis are optional.
      # Do stuff.


# Chain conditions - only first valid will execute.
if sensor1Value == 5:
      # Do stuff if value is equal to 5.
elif sensor1Value <= 2:
      # Do stuff if value less than 2 inclusive.
else
      # Do stuff in all other cases.


# Boolean logic. and - and, or - or, not - not.
if sernsor1Value > 5 and sensor2Value < 3:
      # Do stuff only if both evaluate to true.
elif sensor1Value != 3 or not(sensor2Value > 5):
      # Do stuff if either one evaluates to true.
```

Software: languages: loops

C++

```c++
int ledPins[] = {2, 3, 5, 6};
int pinCount = 4;   // In c++ it's harder to get
                    // length of array.

// Repeat a block of code:
for (int i = 0; i < pinCount; i++){
      // Create variable i - a counter.
      // Start it from 0.
      // Until the condition is met increment it by one.
      Serial.println(ledPins[i]);
}



// Better suited if unknown amount of iterations.
int sum = 0;
while (sum < 50){
      // Do stuff until the condition evaluates to false.
      sum += sensorValue; // equivalent of
                          // sum = sum + sensorValue;
}
```

Python

```python
ledPins = [2, 3, 5, 6]

# Repeat a block of code:
for i in range(len(ledPins)):
      # Generate an array of indices of same length and
      # iterate over it.
      print(ledPins[i])
# Same but better suited for arrays.
for ledPin in ledPins:
      # Go through and array and put item by item in
      # a temporary variable ledPin.
      print(ledPin)
}

// Better suited if unknown amount of iterations.
sum = 0
while sum < 50:
      # Do stuff until the condition evaluates to false.
      sum += sensorValue  # equivalent of
                          # sum = sum + sensorValue
```

FAB LAB BCN

Iaac

Software: languages: external code

C++

```c++
// Put imports in the beginning of the script.

// Import a non-local library.
#include <Arduino.h>
// Import a library from project folder.
#include "CameraInterface.h"



// Access things from the library. For example:
// - access function pinMode from any imported library.
pinMode(22, OUTPUT);
// - access function println from Serial class.
Serial.println(“hello world”);
```

Python

```python
# Put imports in the beginning of the script.

# Import libraries.
import time, math
# Import libraries with an alias.
import cv2 as cv
# Import a part of a library.
from machine import Pin

# Access things from the library. For example:
# - access class Pin and variable OUT from Pin class.
led = Pin(22, Pin.OUT)
```

Software: languages: hardware syntax

Arduino

arduino

```
int digitalSensorPin = 6;
int ledPin = 5;
int analogSensorPin = A0;

pinMode(digitalSensorPin , INPUT); // Setup pin as input
pinMode(ledPin, OUTPUT);           // Setup pin as output

// Read digital or analog values
int digitalSensorValue = digitalRead(digitalSensorPin);
int analogSensorValue = analogRead(analogSensorPin);

// Write to a digital pin
digitalWrite(ledPin, HIGH);
delay(1000);                // delay in milliseconds
digitalWrite(ledPin, LOW);

// Write a PWM to a digital pin.
// Range from 0 to 255 inclusive controls duty cycle.
analogWrite(ledPin, 100);
```

Micropython

micropython

```
digitalSensorPin = 6
ledPin = 5
analogSensorPin = 0

sensor = Pin(digitalSensorPin, Pin.IN) # Setup pin as input
led = Pin(ledPin, Pin.OUT)            # Setup pin as output
sound = machine.ADC(analogSensorPin)  # Setup pin as analog

# Read digital or analog values
digitalSensorValue = sensor.value()
analogSensorValue = sound.read_u16()

# Write to a digital pin
led.high()          # same as led.value(1)
time.sleep(1)       # delay in seconds
led.low()           # same as led.value(0)

# Write a PWM to a digital pin.
# Range from 0 to 65534 inclusive controls duty cycle.
led = machine.PWM(Pin(ledPin))
led.freq(200)
led.duty_u16(100)
```

FAB LAB BCN

Iaac

Software: languages: hardware syntax

Arduino

C++ documentation

Arduino documentation

Micropython

Python documentation (Python)

Python documentation (W3S)

Micropython documentation

Software: languages:

Arduino

```arduino
//
```

Micropython

```micropython
#
```

FAB LAB BCN

Iaac