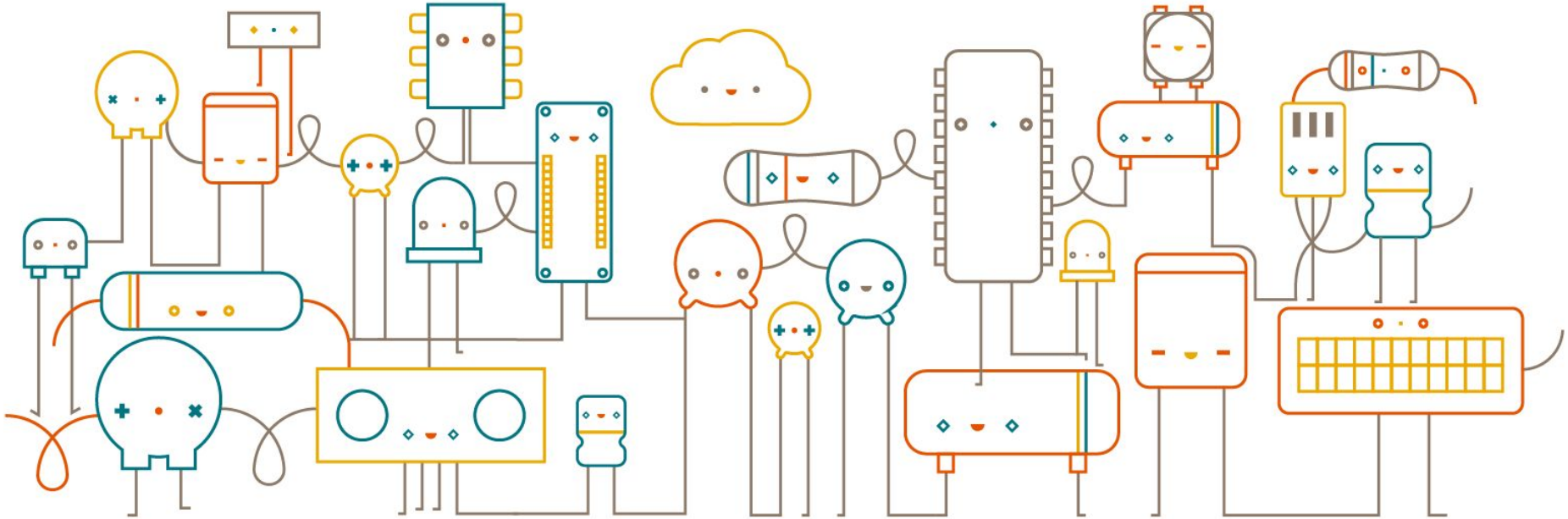


ARDUINO - FAST PROTOTYPING

Speaker - Ali Asgar I. Tashrifwala

ARDUINO DAY 2019

March 16TH
day.arduino.cc
#ArduinoD19

Ali Asgar I. Tashrifwala

About me:

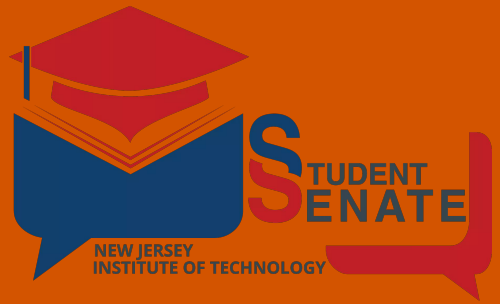
- Research Assistant in Robotics & Data Laboratory (RADLab)
- Event Coordinator, NJIT IEEE Student Branch
- VP of Education, University Heights Toastmasters
- Multiple Hackathon Winner!

Previous Arduino Day Events: 2015, 2016, and 2017

ORGANIZERS:

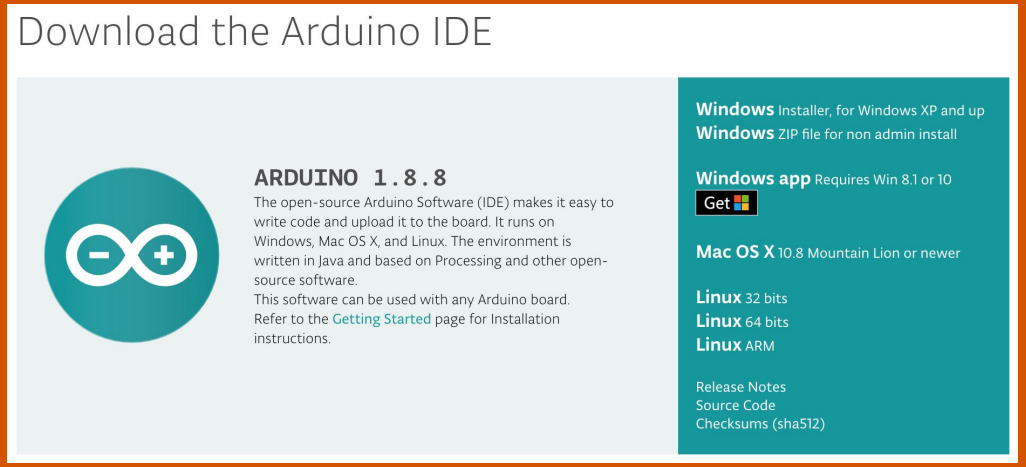
ARDUINO DAY 2019

IEEE North Jersey Section



Installation

- Go to - https://www.arduino.cc/en/Main/Software



- Reference - Installing Arduino IDE - Sparkfun

https://learn.sparkfun.com/tutorials/installing-arduino-ide/all

What is Embedded System?

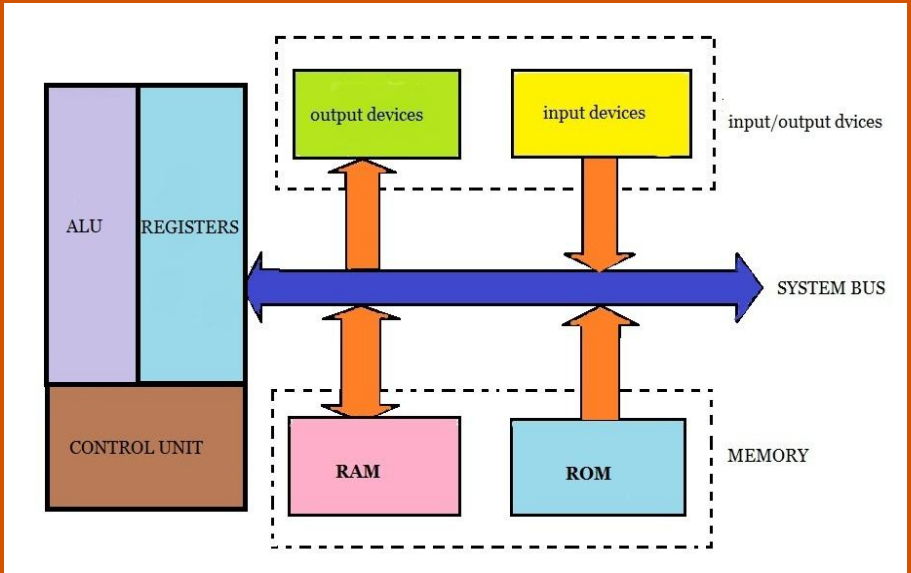
Embedded means "hidden inside". A branch of electronic systems where you see components (hardware) outside, but those components are controlled by set of instructions (software), which is hidden inside the components is called embedded systems.

SCOPE!


code size
performance
memory footprint
energy consumption
Home Appliances
Printer
AV Device
Control Panel

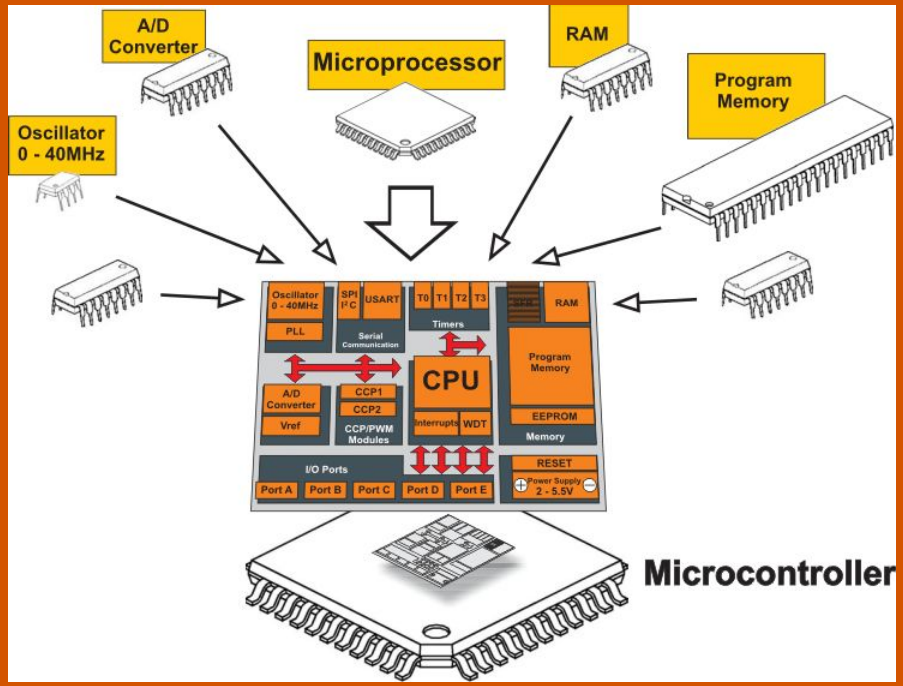
Microprocessors

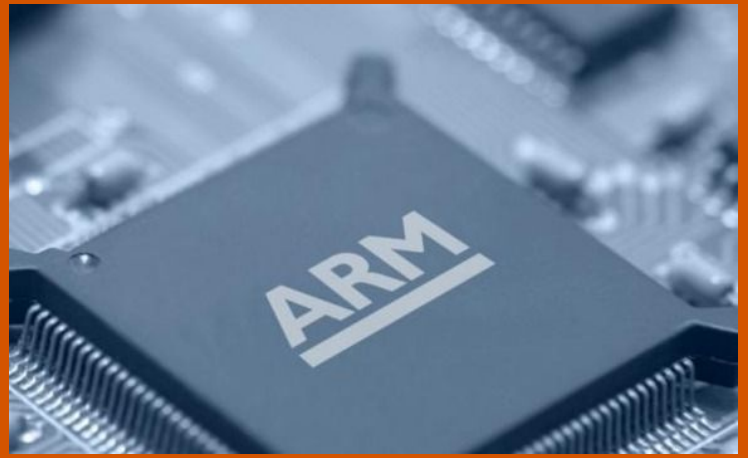
8085 & 8086



Microcontrollers

8051





Assembly Language V/S Embedded C

```
                         1  CSEG
                         2  PUBLIC main_
                         3  PUBLIC puts_
                         4  DSEG
0000 48656C6C6F20576F    5  __3 DB 'Hello World!',0
                         6  CSEG
                         7
0000 55                  8  main_: PUSH BP
0001 8BEC                9  MOV BP,SP
0003 B80000             10  MOV AX,OFFSET __3
0006 50                 11  PUSH AX
0007 E80000             12  CALL puts_
000A 8BE5               13  MOV SP,BP
000C 5D                 14  POP BP
000D C3                 15  RET
                        16  END
```

V/S

Link: http://codepad.org/QOmLUb6y [raw code | output | fork]

C, pasted 1 second ago:

```c
#include <stdio.h>

int main()
{
printf("Hello World");
return 0;
}
```

Output:

```
Hello World
```

Overview

What is Arduino ?

- What is it used for?
- How to get started
- Demonstration

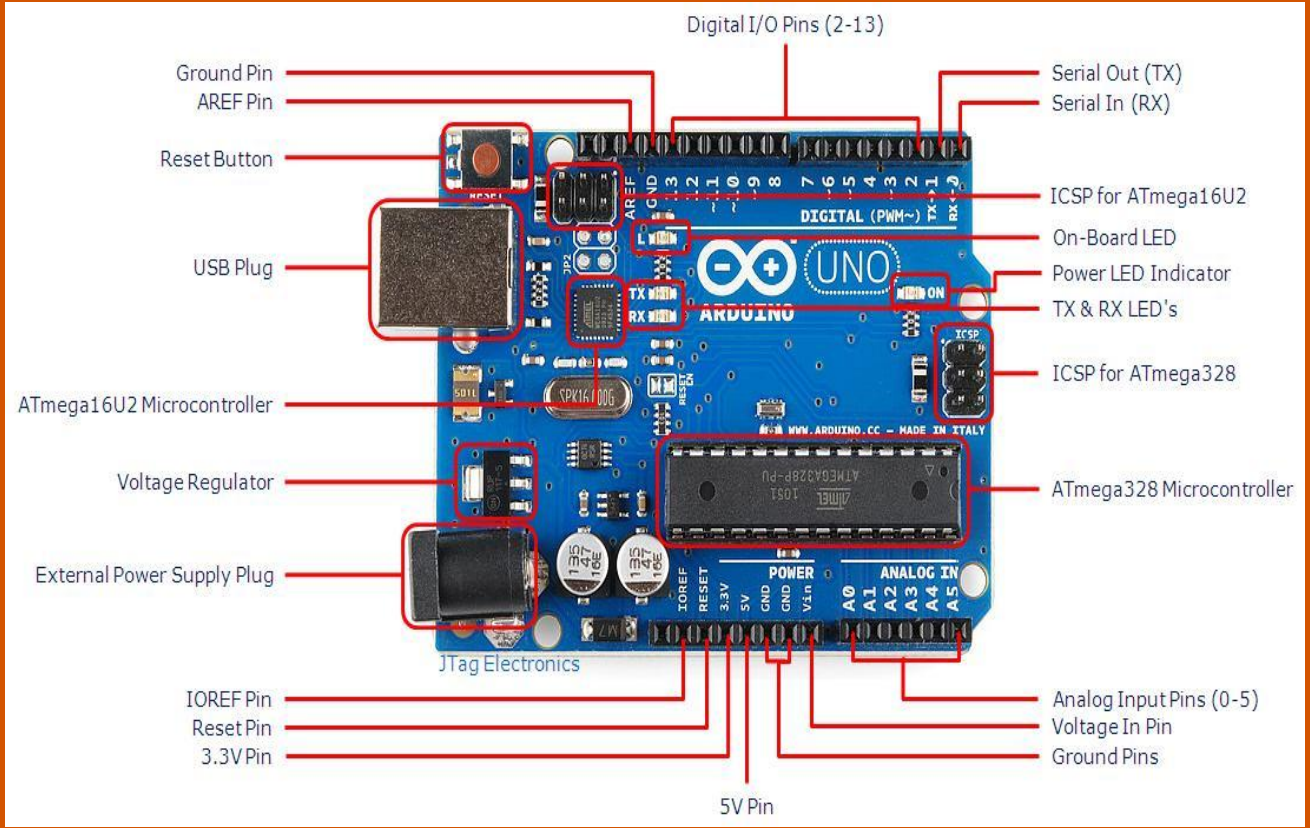
Questions are welcome at any time.

What is Arduino?

- *"Arduino is an open-source physical computing platform based on a simple i/o board and a development environment that implements the Processing / Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer."* (www.arduino.cc, 2006)

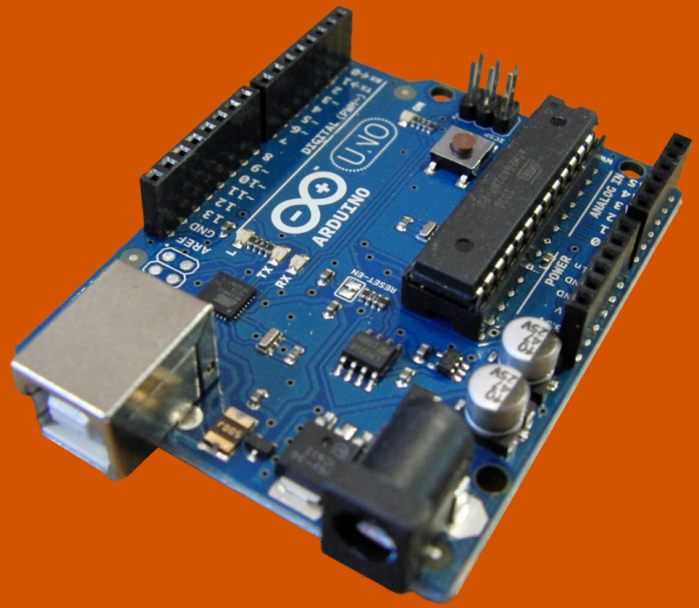
= FAST PROTOTYPING

Meet Arduino Uno R3



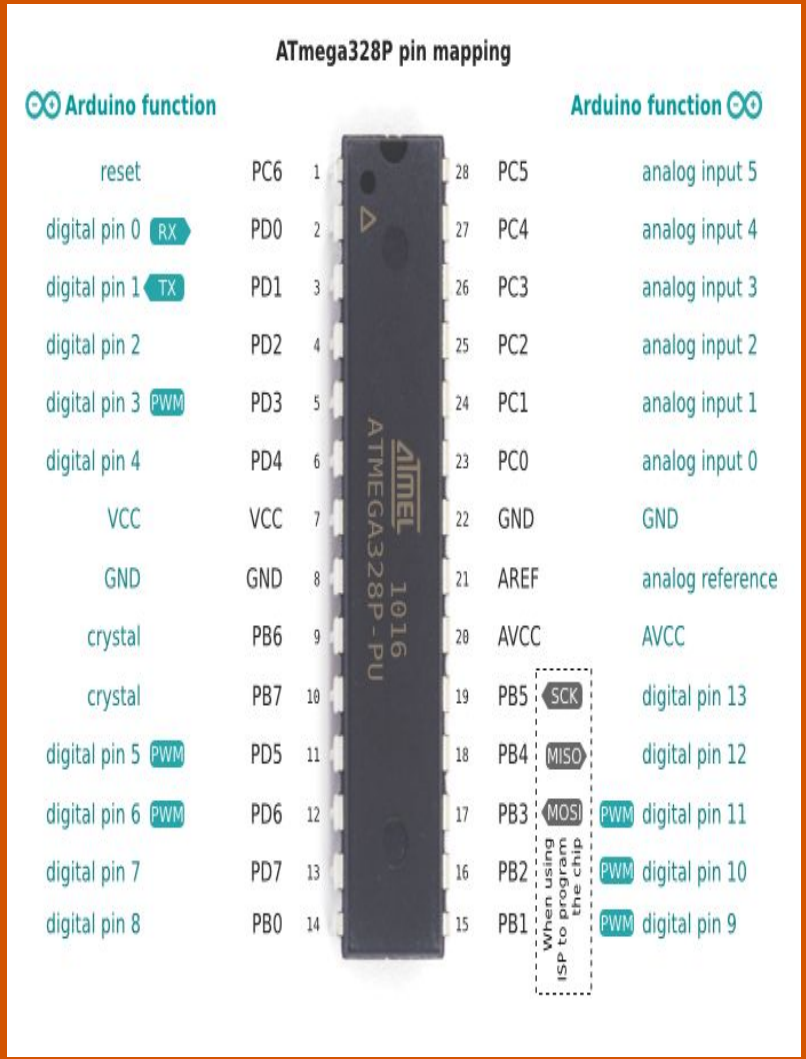
About Arduino Uno

- ATmega328 Microcontroller
- USB Plug & External Power Supply Plug
- Power Pins
- Input and Output Pins
- LED Indicators
- ICSP Headers
- ATmega16U2 Microcontroller Voltage regulator



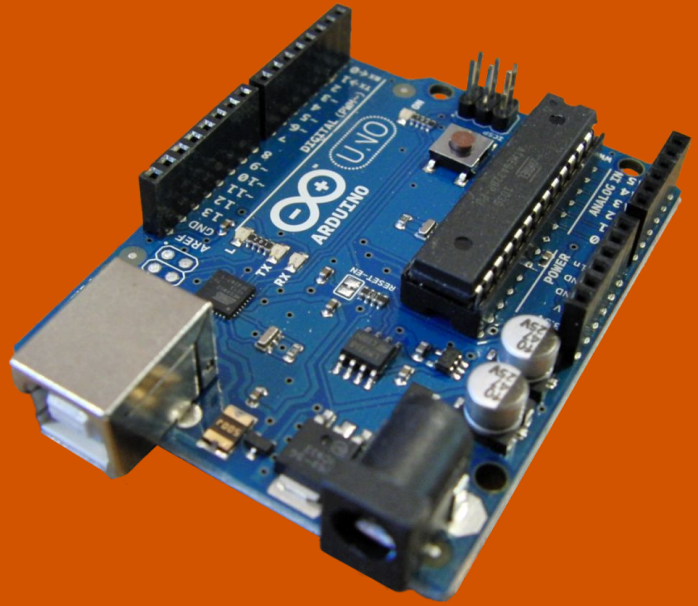
About Arduino Uno

- Operating Voltage 5V
- Input Voltage 7-12V
- Digital I/O Pins 14
- PWM output 6
- Analog Input Pins 6
- Flash Memory 32 KB
- SRAM 2 KB
- EEPROM 1 KB
- Clock Speed 16 MHz



The “Power” of Arduino is in:

- Standard board design
- Wiring language
- Open Source



Arduino is a platform

- Also including an Integrated Development Environment (IDE) for programming.

- The language itself is based in C but is largely modeled upon the www.processing.org language.

What is it used for?

- Physical Computing projects / research
- Interactive Installations
- Rapid prototyping

- When you wish to move beyond the traditional Mouse, Keyboard and Monitor to develop novel and custom interactions in your project work.

What can it do?

Sensors (to sense stuff)

- Push buttons, touch pads, tilt switches.
- Variable resistors (eg. volume knob / sliders)
- Photoresistors (sensing light levels)
- Thermistors (temperature)
- Ultrasound (proximity range finder)

Actuators (to do stuff)

- Lights, LED’s
- Motors
- Speakers
- Displays (LCD)

Why Arduino?

- It is Open Source, both in terms of Hardware and Software.
- It is cheap, the hardware can be built from components or a prefab board can be purchased for approx $25.
- It can communicate with a computer via serial connection over USB.
- It can be powered from USB or standalone DC power.

Why Arduino?

- It can run standalone from a computer (chip is programmable) and it has memory (a small amount).

- It can work with both Digital and Analogue electronic signals. Sensors and Actuators.

- You can make cool stuff! Some people are even making simple robots, and we all know robots are just cool. ☺

How to get started

- You'll need a board of course, Arduino UNO R3 , along with the USB cable and DC power supplies.
- Read about, understand what you are working with and download the IDE: http://www.arduino.cc
- Mac, Windows and Penguin friendly versions available
- Then you are ready to plug it in!

Not so fast!

- It’s important to note at this stage that Arduino’s are electronic devices.

- This means you MUST consider electrical safety and understand the basics before diving straight in.

- The board itself doesn’t operate at what would normally be considered dangerous Voltages or Amperage, but if in doubt at any stage of use you should seek more expert advice.

Basic Electrical knowledge

- A fantastic guide to electronics in theory, practice and of course safety is available as a PDF at:

  http://www.ibiblio.org/obp/electricCircuits/

- What you want is Volume 1, DC circuits. This will help you greatly in understanding how to wire circuits when using sensors and actuators.

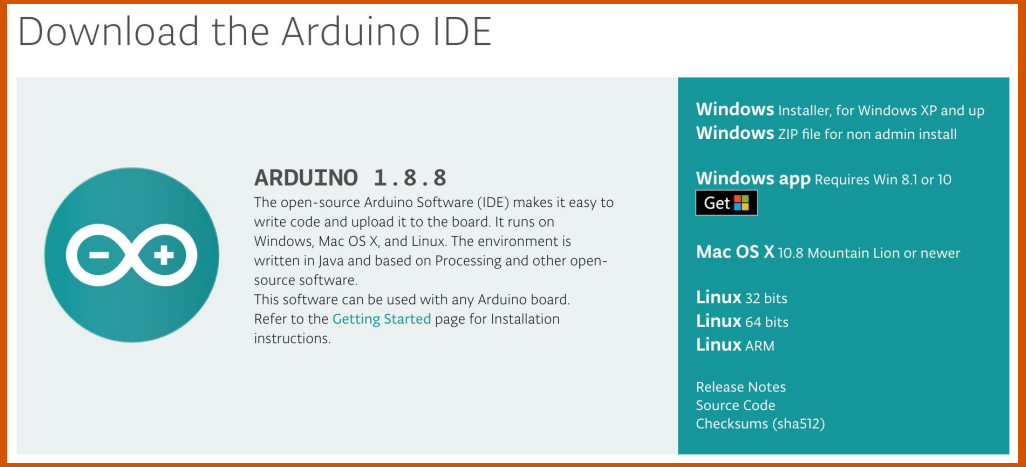
Getting up and running

- The power mode must be selected before you plug the board into anything.

- When powering from the USB cable (5 volts) the jumper should be closest to the USB input, for DC supply the jumper should be closest to the DC input.

Installation

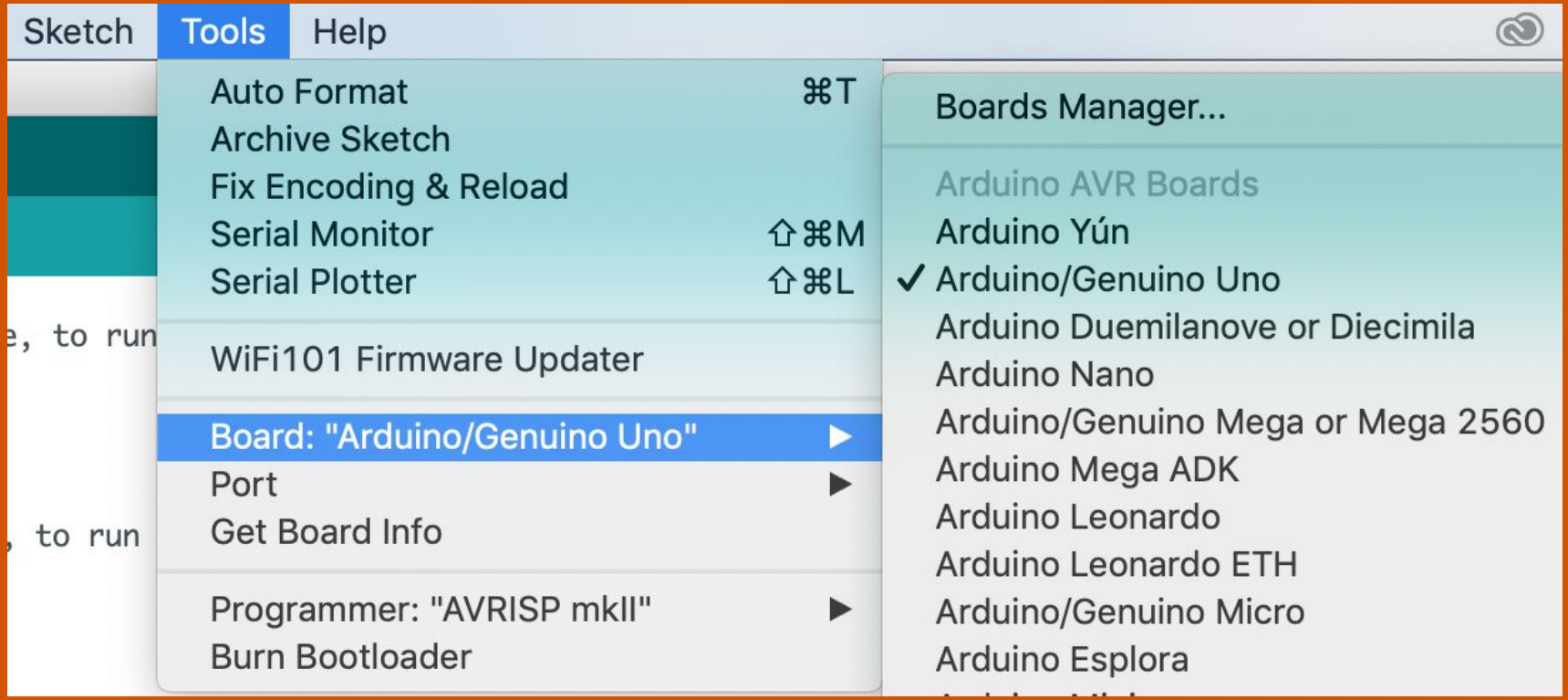
- Go to - https://www.arduino.cc/en/Main/Software



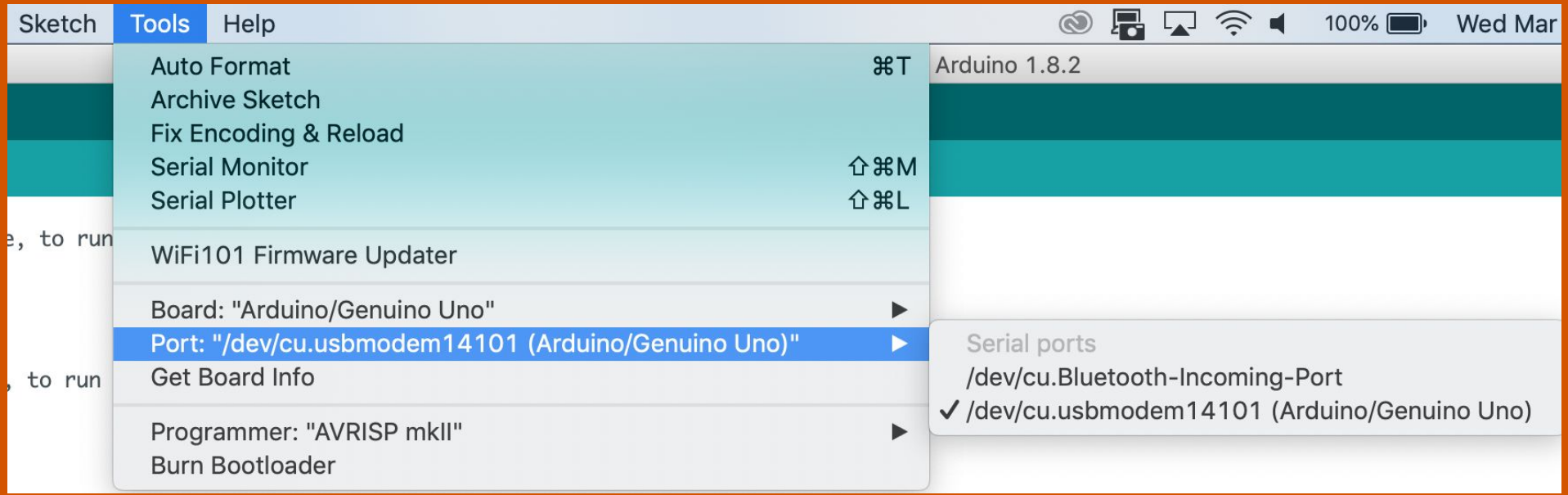
- Reference - Installing Arduino IDE - Sparkfun

https://learn.sparkfun.com/tutorials/installing-arduino-ide/all

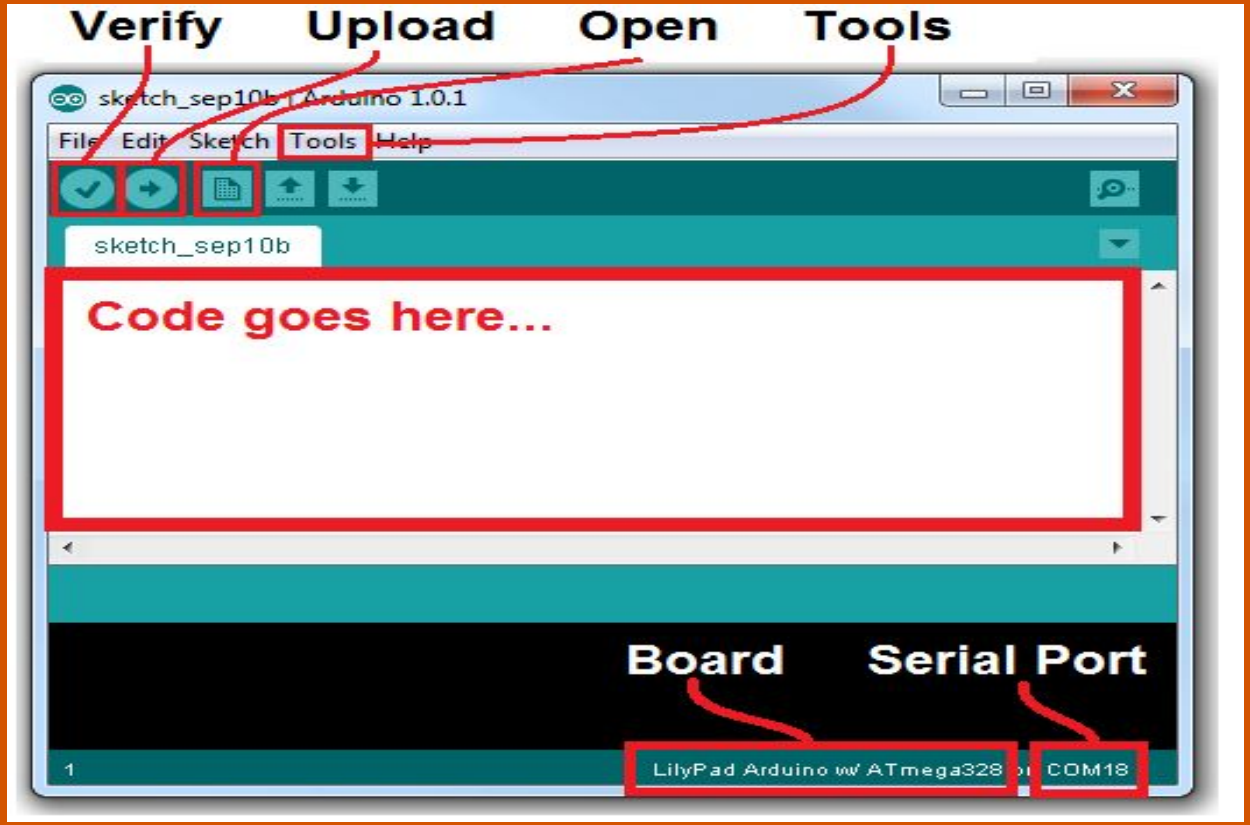
Board Type


Sketch
Tools
Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Serial Plotter ⇧⌘L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino/Genuino Micro
Arduino Esplora
e, to run
to run

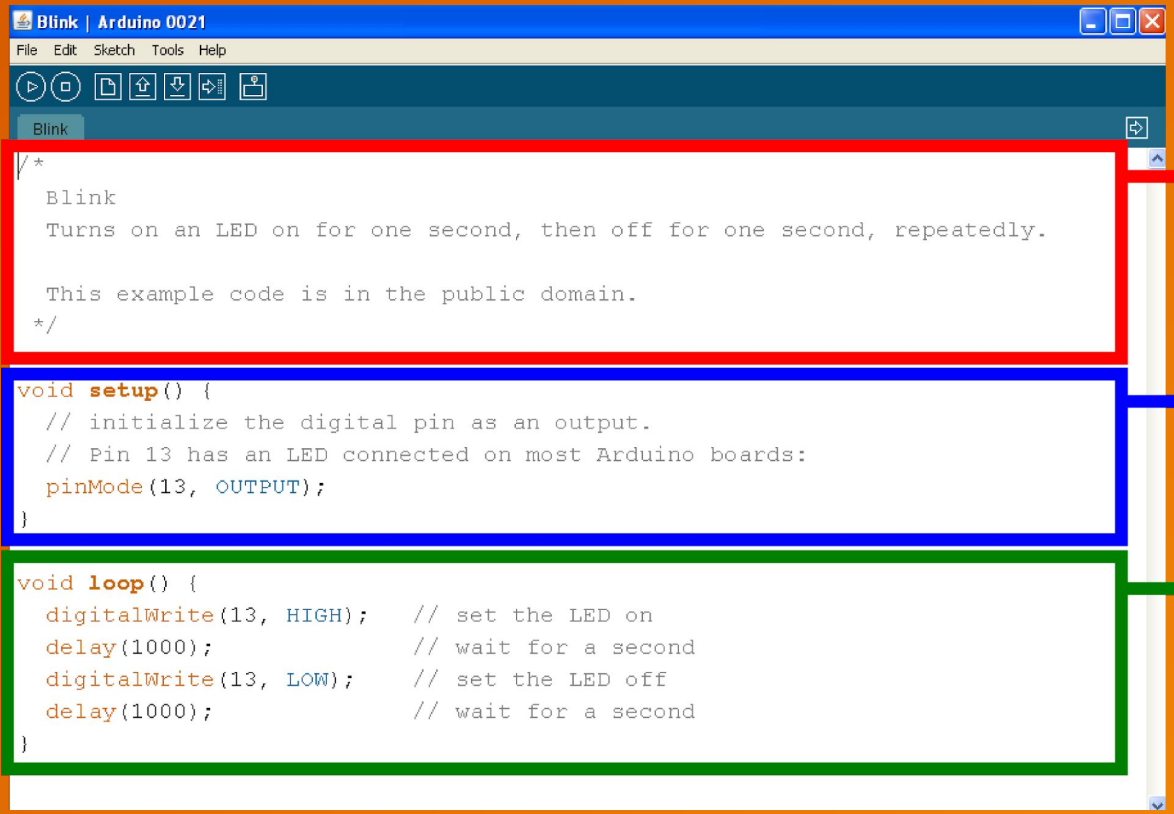
Serial Port / COM Port



The Environment



Parts of the Sketch



Comments / Explaining the game

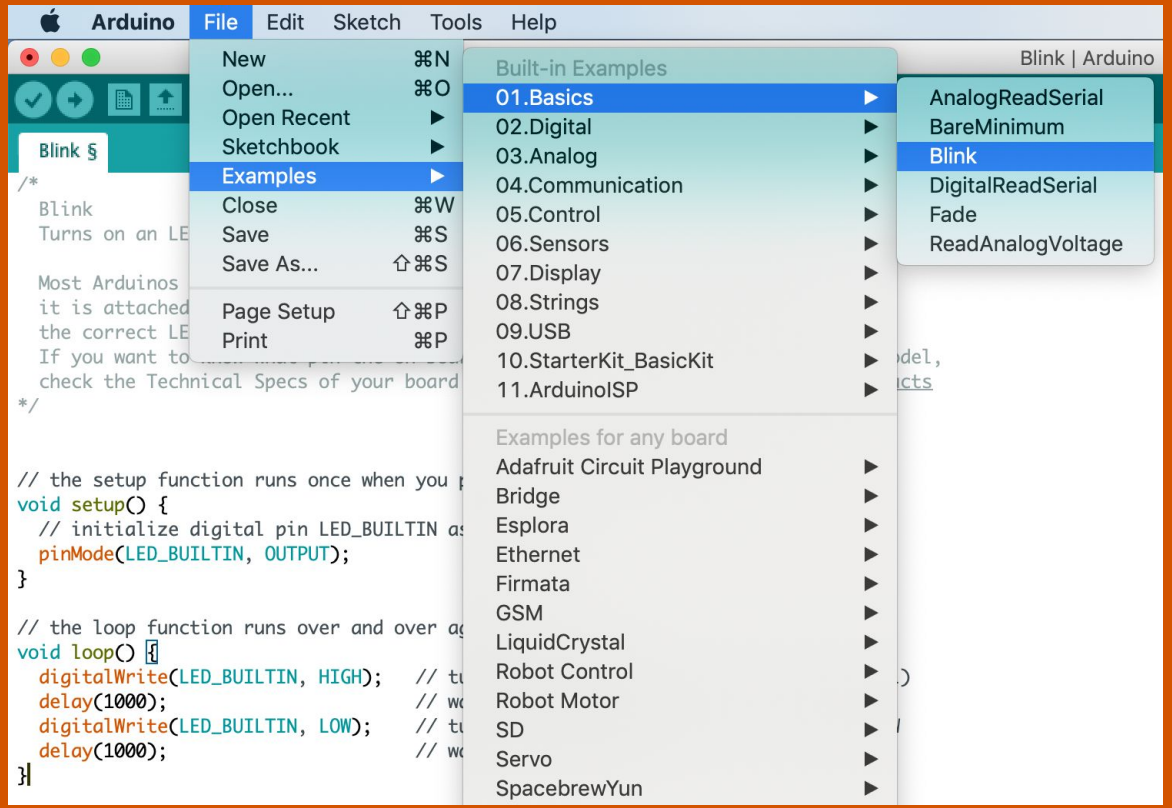
Setup / Stretching or tying shoes

Loop / Playing the game

Demonstration

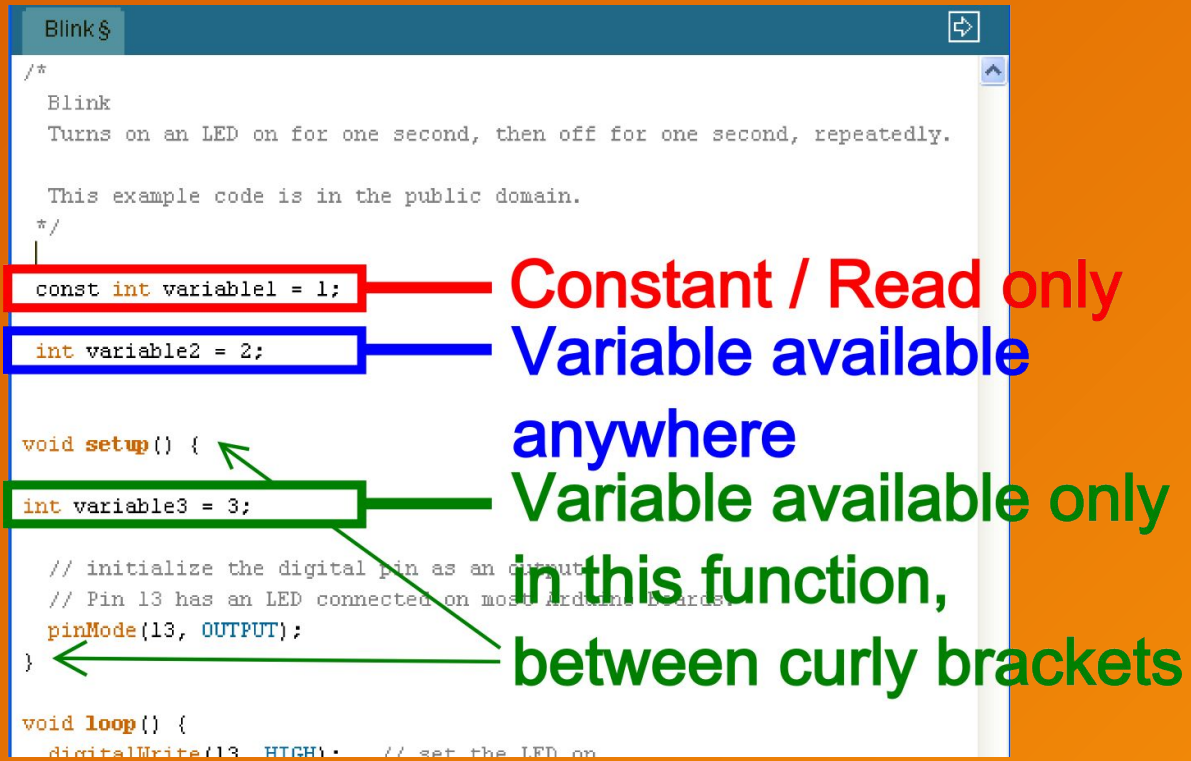
- It’s time for a simple demonstration.

BLINK PROGRAM



Variable Scope

Where you declare your variables matters



Setup void setup ( ) { }

- **The setup function comes before the loop function and is necessary for all Arduino sketches**

```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}
```

Setup void setup ( ) { pinMode (13, OUTPUT); }

Outputs are declare in setup, this is done by using the pinMode function

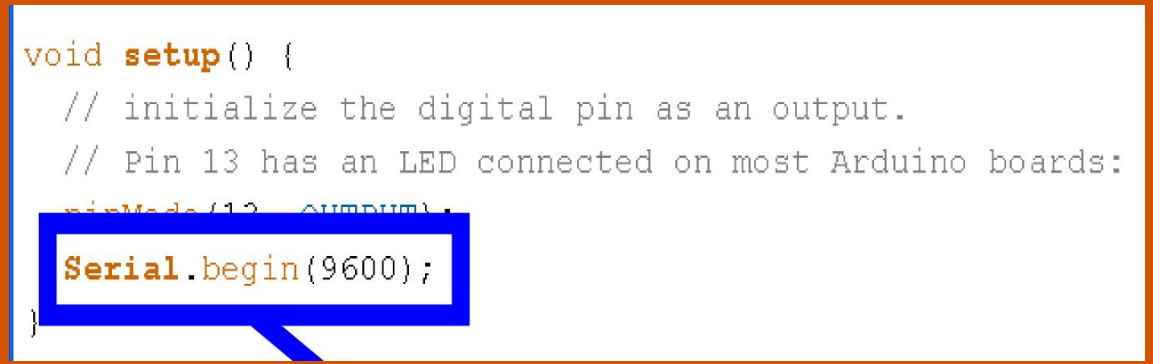
- **This particular example declares digital pin # 13 as an output, remember to use CAPS**

```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}
```

Setup void setup ( ) { Serial.begin;}

Serial communication also begins in setup

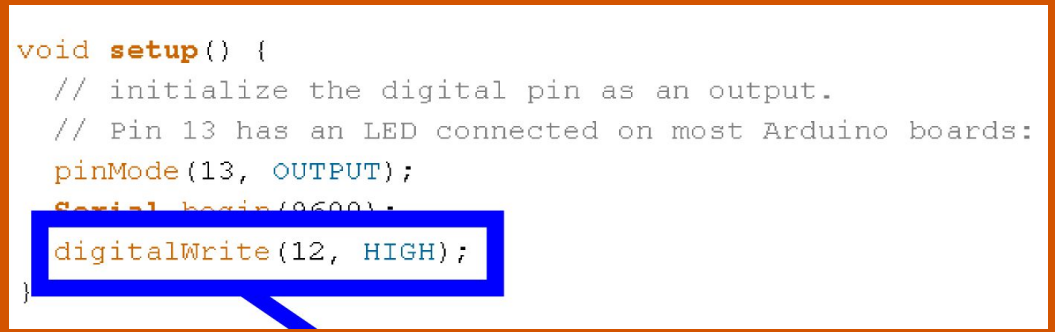
- **This particular example declares Serial communication at a baud rate of 9600. More on Serial later…**



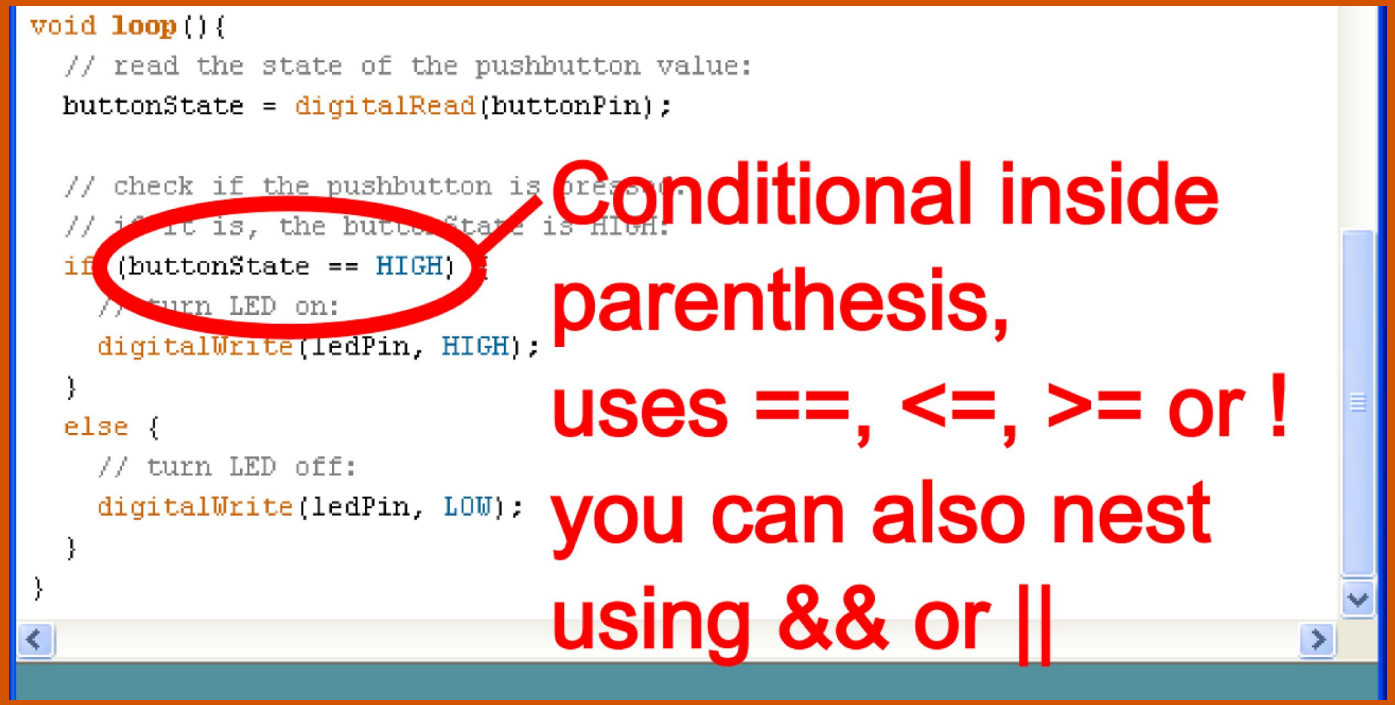
Setup, Internal Pullup Resistors
void setup ( ) { digitalWrite (12, HIGH); }

You can also create internal pullup resistors in setup, to do so digitalWrite the pin HIGH

- **This takes the place of the pullup resistors currently on your circuit 7 buttons**

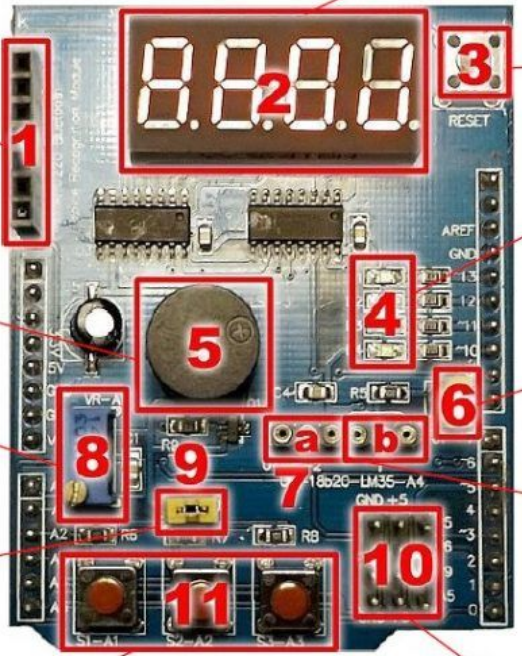


Conditional Statements
if (this is true) { do this; }



Arduino Multi-function Shield

(Funduino výukový multi-shield do škol pro Arduino)



Konektor pro modul Bluetooth APC220 (nebo Voice Recognification Module)

***Popis** (směrem shora)*
7 – nepoužito
6 – nepoužito
5 – pin 0
4 – pin 1
3 – nepoužito
2 – napětí +5 V
1 – GND

Piezo-bzučák pasivní – připojen k pinu 3 (PWM)

Odporový trimr 10 kΩ mezi referenční napětí (+5 V) a GND, jezdec vyveden na pin A0

Propojka (jumper) – připojuje Pull Up rezistory (10 kΩ) k tlačítkům S1, S2 a S3 *(viz 11)*

Vstupní tlačítka – připojena k pinům A1-3 *(stisknutím – hodnota LOW)*

***Popis** (zleva)*
Tlačítko S1 – pin A1
Tlačítko S2 – pin A2
Tlačítko S3 – pin A3

Čtveřice LED segmentů řízených obvodem 74HC595
Zápis do Latch – pin 4 (start zápisu LOW, konec HIGH)
Hodinový pulz (takt) – pin 7
Sériová data – pin 8 – zápis 2× 8 bitů (2× Byte)
(první Byte – data rozsvícených LED – rozsvícení LOW, vypnutí HIGH)
Bity MSB LSB odpovídají segmentu:
V druhém Byte se zapisuje adresa:
První *(zleva)* zobrazovač: 0000001
Druhý *(zleva)* zobrazovač: 0000010
Třetí *(zleva)* zobrazovač: 0000100
Čtvrtý *(zleva)* zobrazovač: 0001000

Reset – resetuje modul Arduino *(nikoliv Shield)*

Signalizační LED *(rozsvícení LOW, vypnutí HIGH)*

***Popis LED** (směrem shora)*
LED D1 – pin 13 (PWM)
LED D2 – pin 12
LED D3 – pin 11 (PWM)
LED D4 – pin 10 (PWM)

Propojka (jumper) – připojuje Pull Up rezistor 10 kΩ na vstup A4 *(vyvedeno na střední pin v konektoru 7b)*

Konektor pro připojení čidla IR *(7a)* a tepelných čidel Dallas 18B20 nebo LM35 *(7b)*

***Popis 7a** (zleva)*	***Popis 7b** (zleva)*
1 – pin 2	1 – GND
2 – GND	2 – pin A4
3 – napětí +5 V	3 – napětí +5 V

Konektor pro připojení čidel (např. sonar HCSR04, servo...)

***Popis** (zleva)*
První sloupec – GND (všechny 4 piny)
Druhý sloupec – napětí +5 V (všechny 4 piny)
Třetí sloupec *(shora)*
pin 5 (PWM),
pin 6 (PWM),
pin 9 (PWM)
pin A5

Vytvořil: RNDr. Miroslav Panoš, Ph.D.

Comments

- Comments can be anywhere Comments created with // or /* and */ Comments do not affect code

Operators

The equals sign

- = is used to assign a value
- == is used to compare values
- And & Or
- && is “and”
- || is “or”

Declaring Variables

- Boolean: **boolean variableName;**
- Integer: **int variableName;**
- Character: **char variableName;**
- String: **stringName [ ];**

Assigning Variables

- Boolean: **variableName = true;**
- or **variableName = false;**

- Integer: **variableName = 32767;**
- or **variableName = -32768;**

- Character: **variableName = ‘A’;**
- or **stringName = “BLAH!!”**

Basic Repetition

Loop

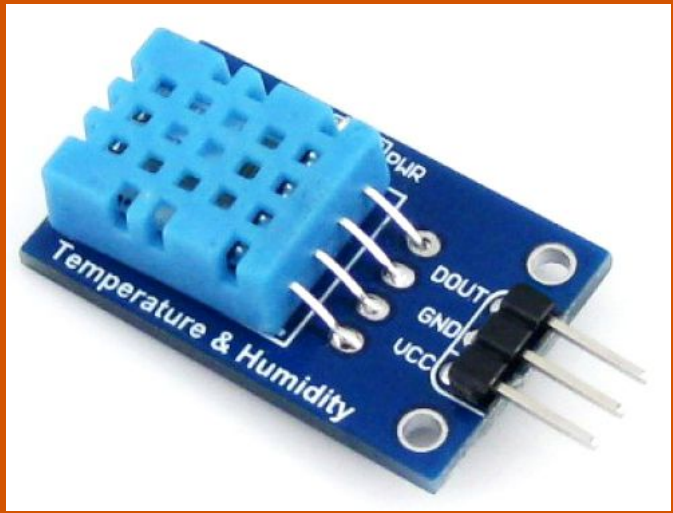
For

While

Temperature & Humidity Sensor (DHT11)

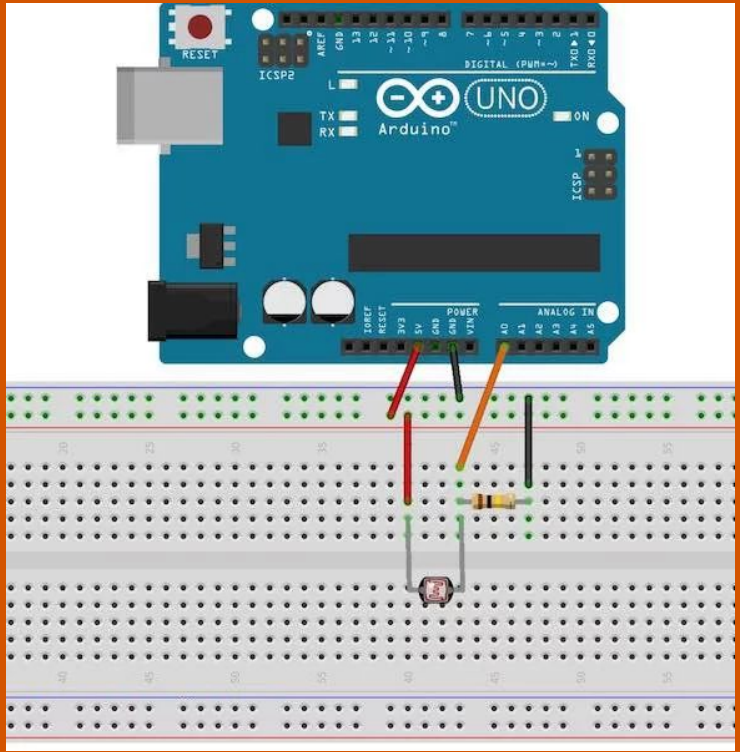
Specfications:

- **Temperature**
  - **Resolution : 1°C**
  - **Accuracy : ±2℃**
  - **Measuring range : 0°C ~ 50°C**
- **Humidity**
  - **Resolution : 1%RH**
  - **Accuracy : ±5%RH (0~50°C)**
  - **Measuring range : 20%RH ~ 90%RH (25°C)**
- **Operating voltage : 3.3V ~ 5.5 V**
- **Recommended storage condition**
  - **Temperature : 10°C ~40°C**
  - **Humidity : 60%RH or below**



Light Dependent Resistor (LDR)

High Resistance Photons reduce the resistance.



Buzzer

A buzzer or beeper is an audio signalling device,which may be mechanical, electromechanical, or piezoelectric (piezo for short).

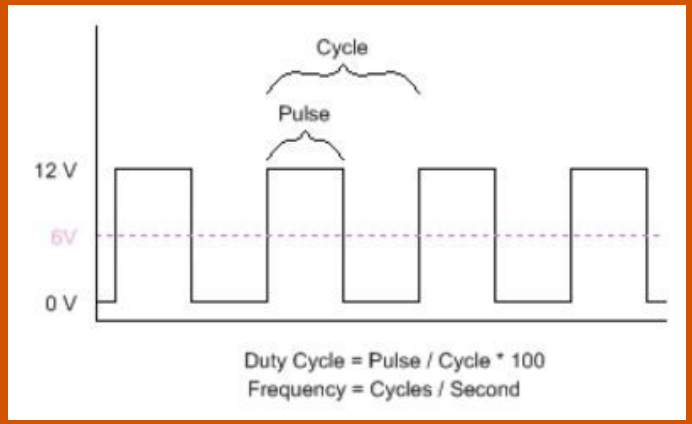
Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

PWM

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means.

The duration of "on time" is called the pulse width.

To get varying analog values, you change, or modulate, that pulse width. controlling the brightness of the LED.

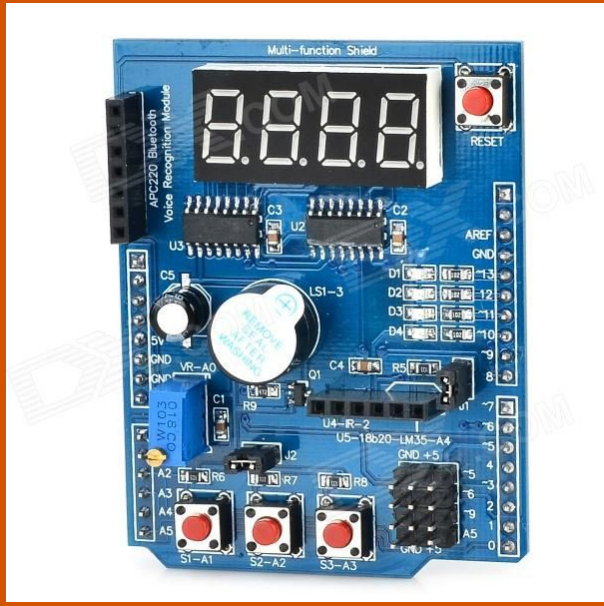


Ultranosic Sensor (HC-CR04)

This is the HC-SR04 ultrasonic distance sensor. This economical sensor provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm.

Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit.

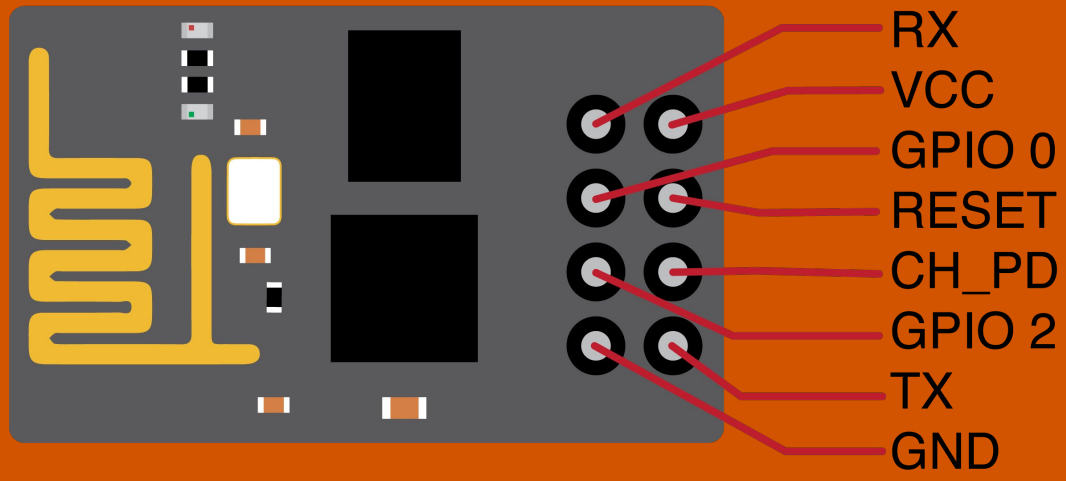
Project - Car Parking Assistant

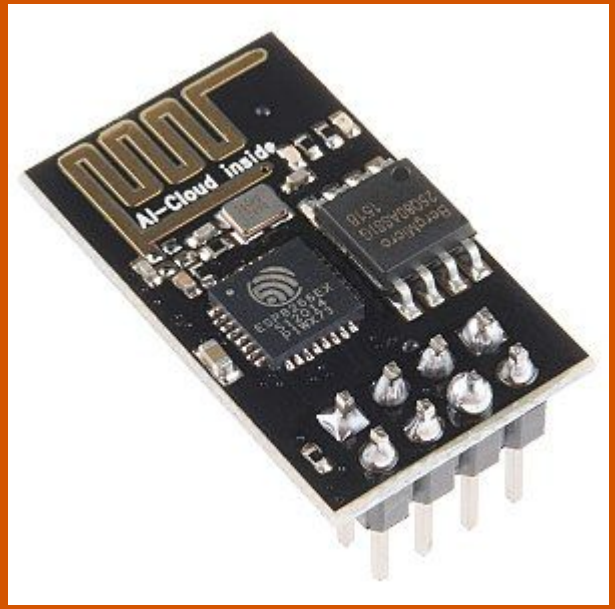


Internet of Things IoT

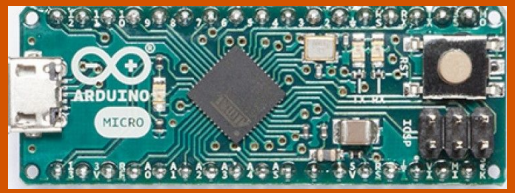
Internet of Things IoT

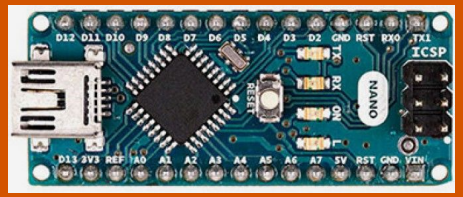
The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability.

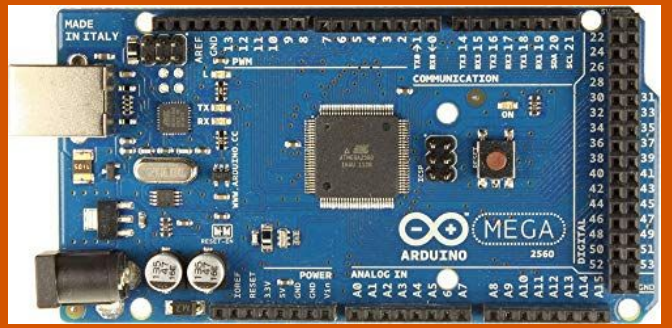


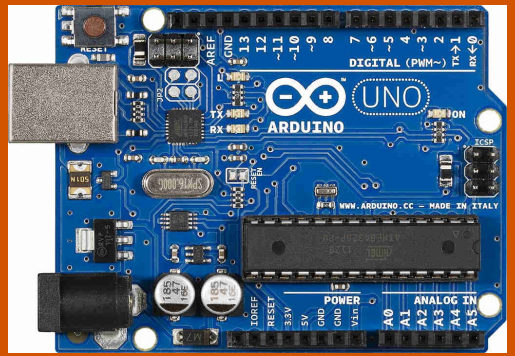


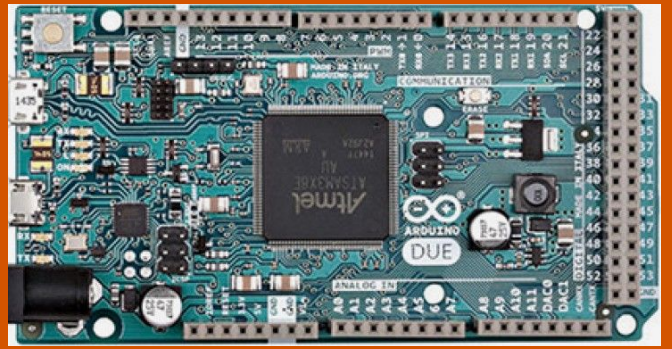
Arduino I/0 Boards











Shields

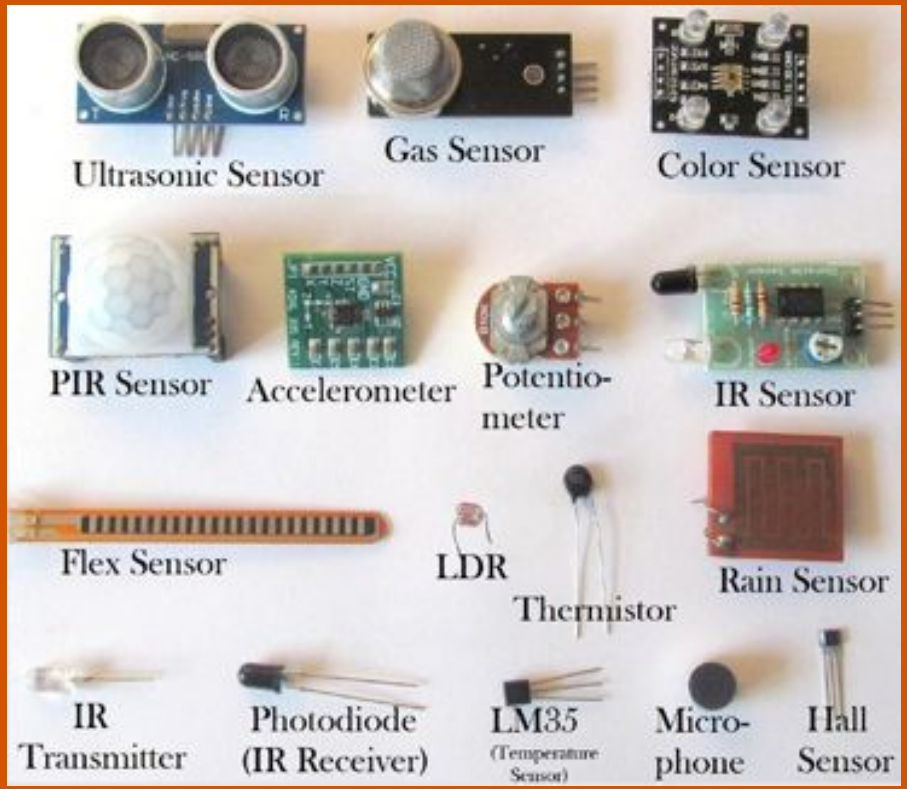
Touchscreen Shield

Datalogging Shield

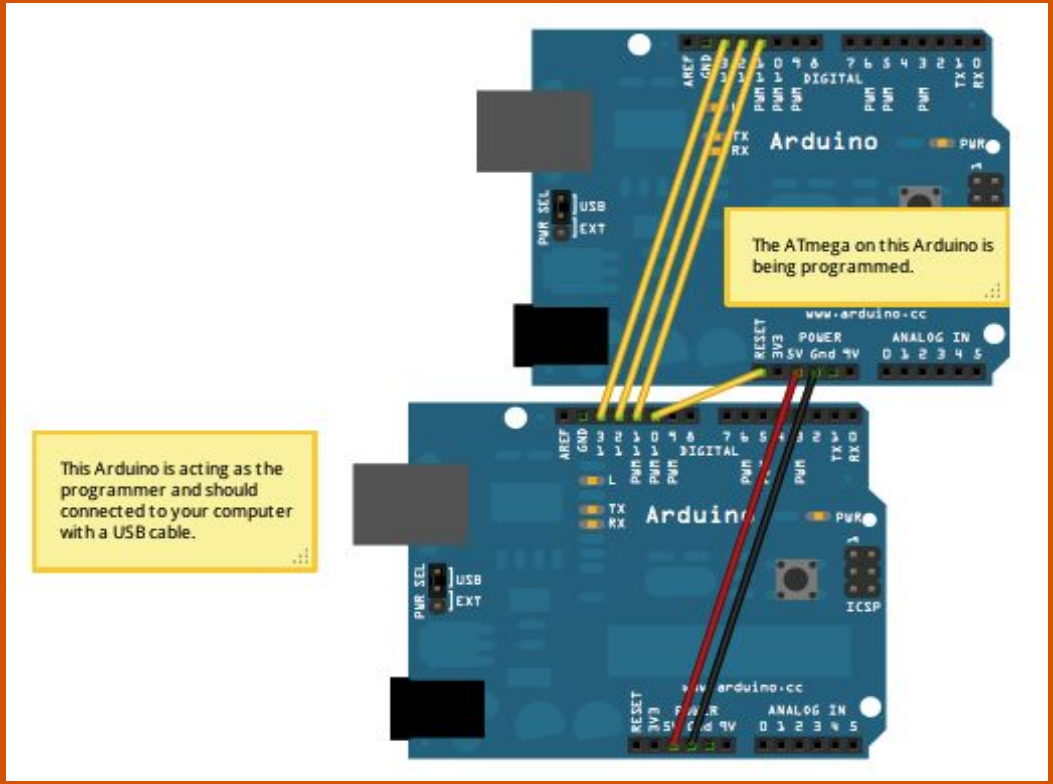
Wave Shield

Even more shields!

Sensors


Ultrasonic Sensor
Gas Sensor
Color Sensor
PIR Sensor
Accelerometer
Potentio-
meter
IR Sensor
Flex Sensor
LDR
Thermistor
Rain Sensor
IR
Transmitter
Photodiode
(IR Receiver)
LM35
(Temperature
Sensor)
Micro-
phone
Hall
Sensor

Bootloading



Arduino board to burn a bootloader onto an AVR

- Select the items in the Tools > Board and Serial Port menus that correspond to the board you are using as the programmer (not the board being programmed).

- Upload the ArduinoISP sketch.

- Wire your Arduino board to the target as shown in the diagram below. (Note for the Arduino Uno: you'll need to add a 10 uF capacitor between reset and ground.)

Arduino board to burn a bootloader onto an AVR

- Select the item in the Tools > Board menu that corresponds to the board on which you want to burn the bootloader (not the board that you're using as the programmer).
- Select the Arduino as ISP in the Tools>Programmer menu.
- Use the Burn Bootloader command.

Questions?

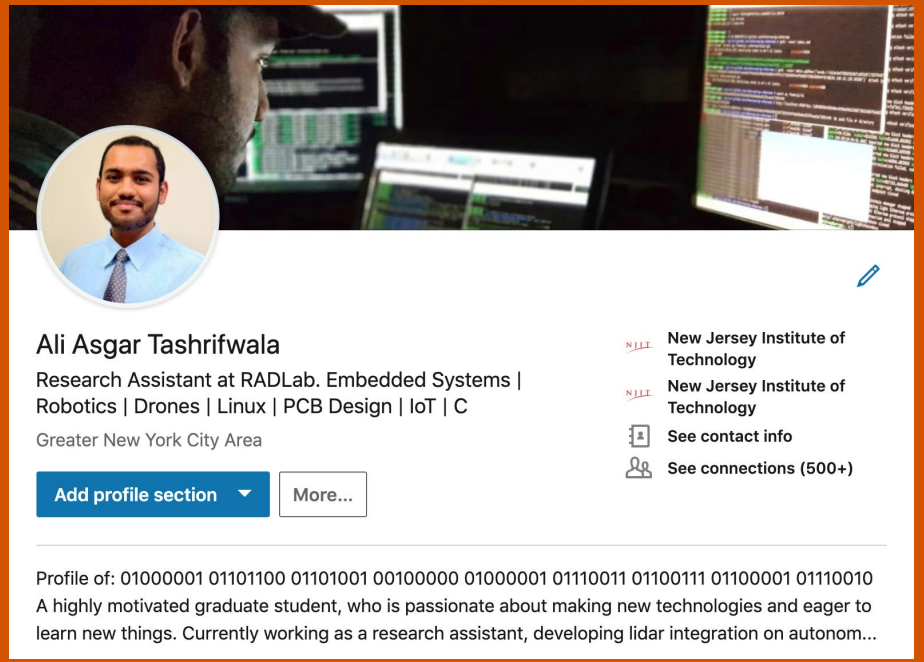
Connect with me on -
https://www.linkedin.com/in/ali-asgar/

Endorse me for -

- ***Arduino***
- ***Embedded Systems***
- ***Public Speaking***

How good? - Highly skilled

How do you know? - Worked together directly on the same team or project



Ali Asgar I. Tashrifwala
ai229@njit.edu
linkedin.com/in/ali-asgar/

Thank you!

March 16TH
day.arduino.cc
#ArduinoD19