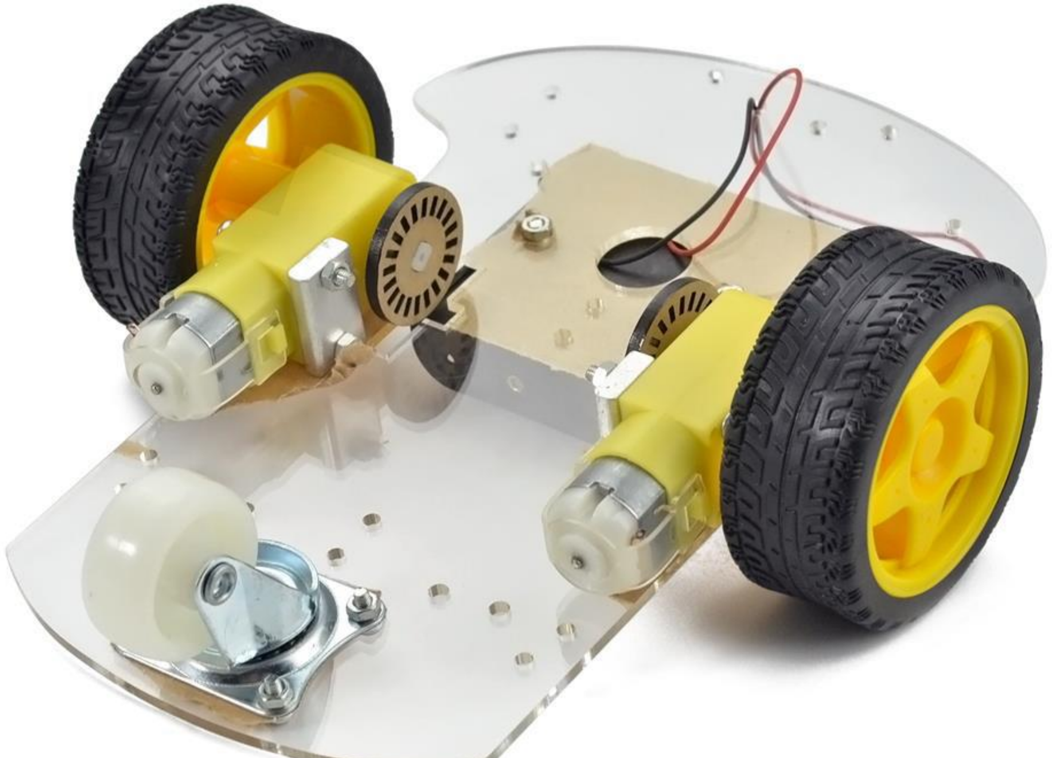
Week 12

Module 5: EE100 Course Project
Programming your first robot



Dr. –Ing. Ahmad Kamal Nasir

Office Hours: Room 9-245A

Tuesday (1000-1100)

Wednesday (1500-1600)

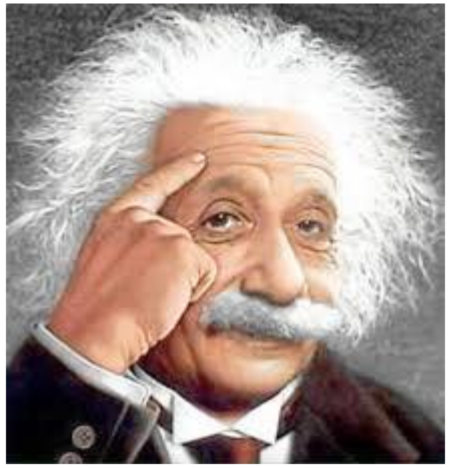
Objectives

- At the end of this session you are expected to have the following:
    - Know basics of Arduino microcontroller
    - Know basics of Arduino programming
    - Hands-on wheeled mobile robot programming experience
    - Control your robot’s ultra-sonic sensors and DC-motor

- There are five activities designed to teach you how to program your own robot.

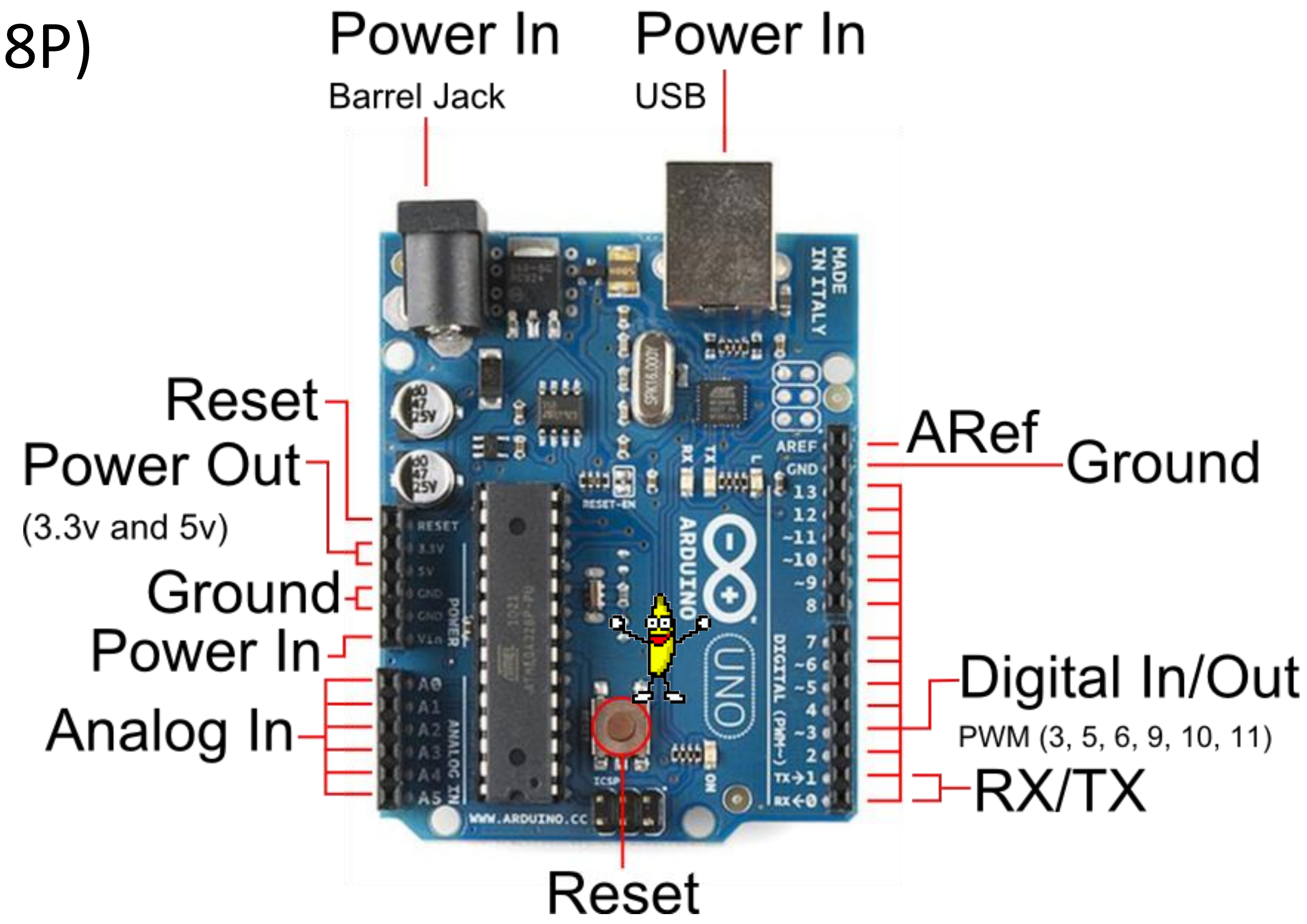
Microcontroller (Robot Brain)

- Capable of storing and executing your algorithm.
- Can timely execute actions through actuators.
- Can periodically acquire information through sensors.
- Can communicate with computer or other robots.
- Elements of a microcontroller:
  - Pins for digital inputs and outputs
  - Pins for analogue inputs and outputs
  - Timers for delays and task scheduling
  - Communication ports



Arduino Uno– Hardware Overview

- 8-Bit Microcontroller (ATmega328P)
  - Clock Speed: 16MHz
  - 32 KB Flash memory
  - 2 KB RAM
  - 1 KB EEPROM
- 14 Digital I/O
  - 6 can provide 8-bit PWM
  - 20mA per I/O pin
- 6 Analog Inputs
- 1 Serial Port (RX/TX)
- Operating Voltage, 5V

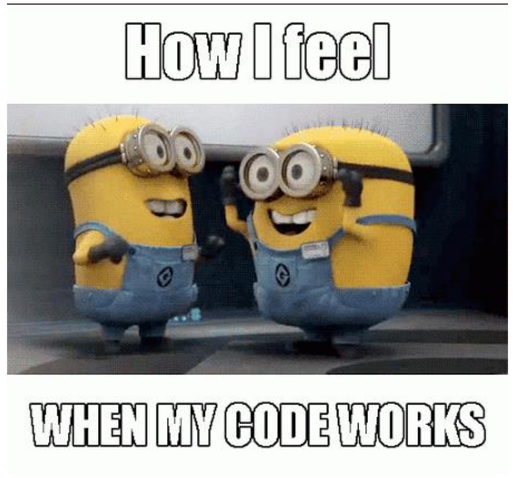


Mobile Robot: Programming

C Like Programming Language for Arduino Microcontrollers

Microcontroller Programming

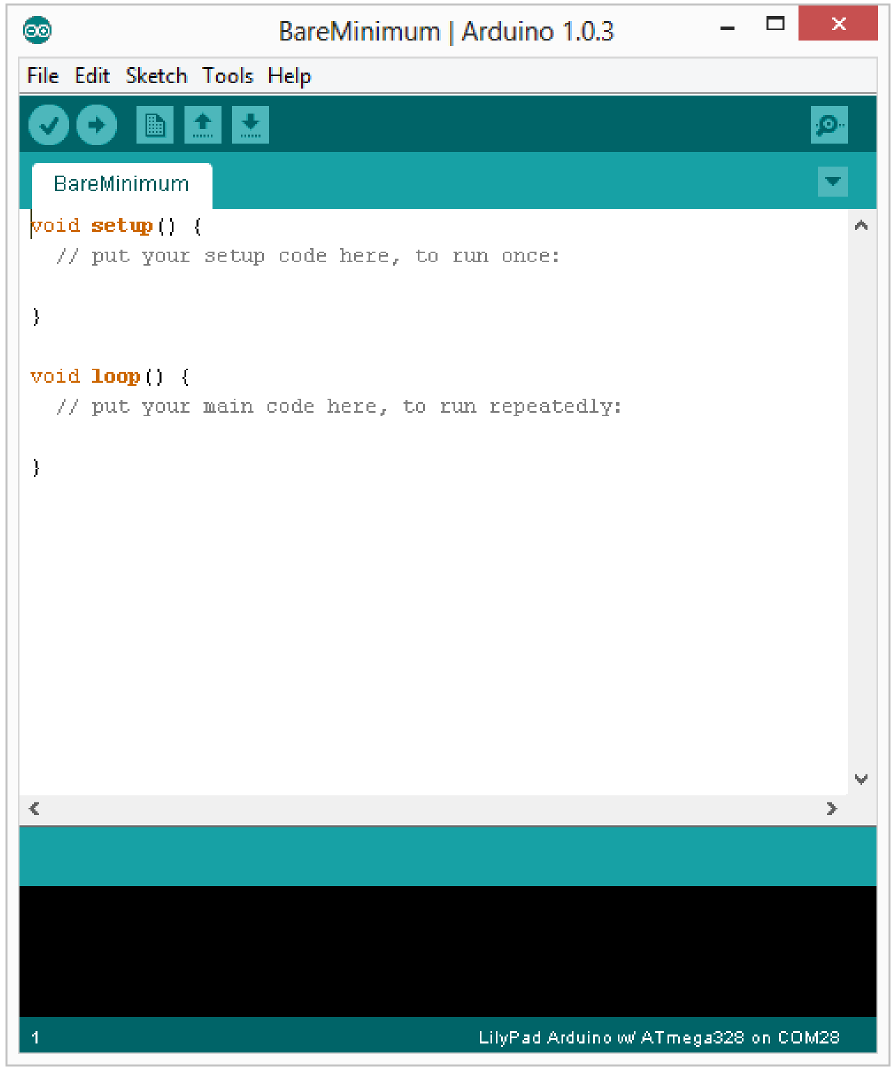
- To write instructions for the mobile robot
- High level **“C”** like language
  - After compilation translated into binary file which is downloaded in to microcontroller FLASH memory
- Program instructions are executed sequentially
- Program is written in a modular way
- Different thinking is required compared to computer programs because memory is limited
- Program runs forever until powered down



Arduino Program: Sketch

Two required functions

```
void setup()
{
        // runs once
}

void loop()
{
        // repeats
}
```


BareMinimum | Arduino 1.0.3
File Edit Sketch Tools Help
BareMinimum

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

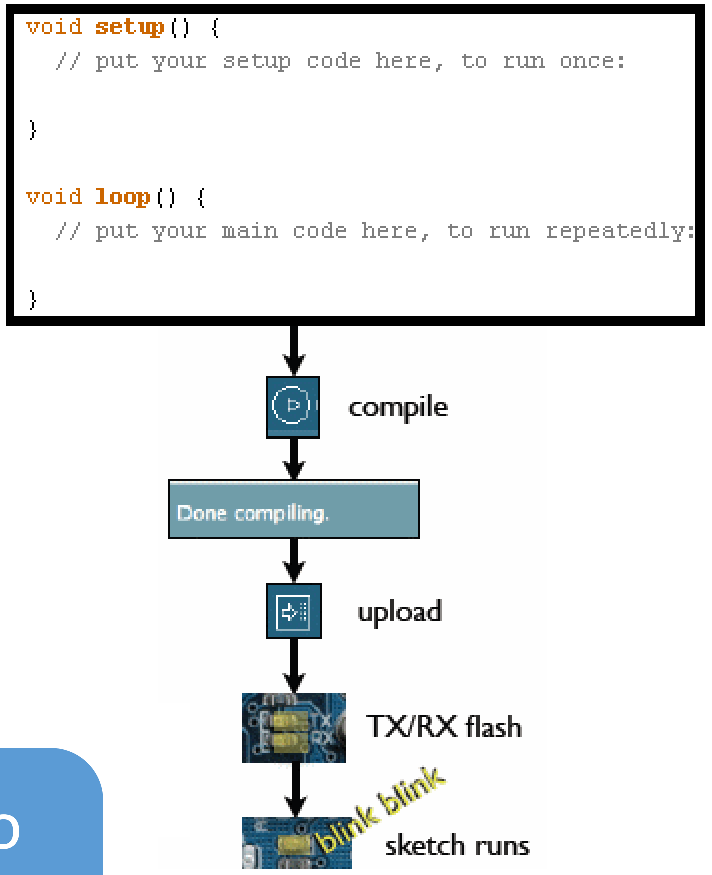
1
LilyPad Arduino w/ ATmega328 on COM28

Development Lifecycle

Write your sketch

Press compile button

Press upload button to download your sketch into the microcontroller



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Programming Reference

Digital I/O
- **pinMode** (pin, mode)
- **digitalWrite** (pin, value)
- **digitalRead** (pin)

Analog I/O
- analogReference (EXTERNAL)
- **analogRead** (pin)
- **analogWrite** (pin, value) - PWM

Time
- millis ()
- micros ()
- **delay** (ms)
- **delayMicroseconds** (us)

Math
- min()
- max()
- abs()
- constrain()
- map()
- pow()
- sqrt()

Trigonometry
- sin()
- cos()
- tan()

Random Numbers
- randomSeed()
- random()

Bits and Bytes
- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()

External Interrupts
- attachInterrupt ()
- detachInterrupt ()

Interrupts
- interrupts()
- noInterrupts()

Communication
- Serial.available ()
- **Serial.read** ()
- **Serial.print** ()
- **Serial.println** ()

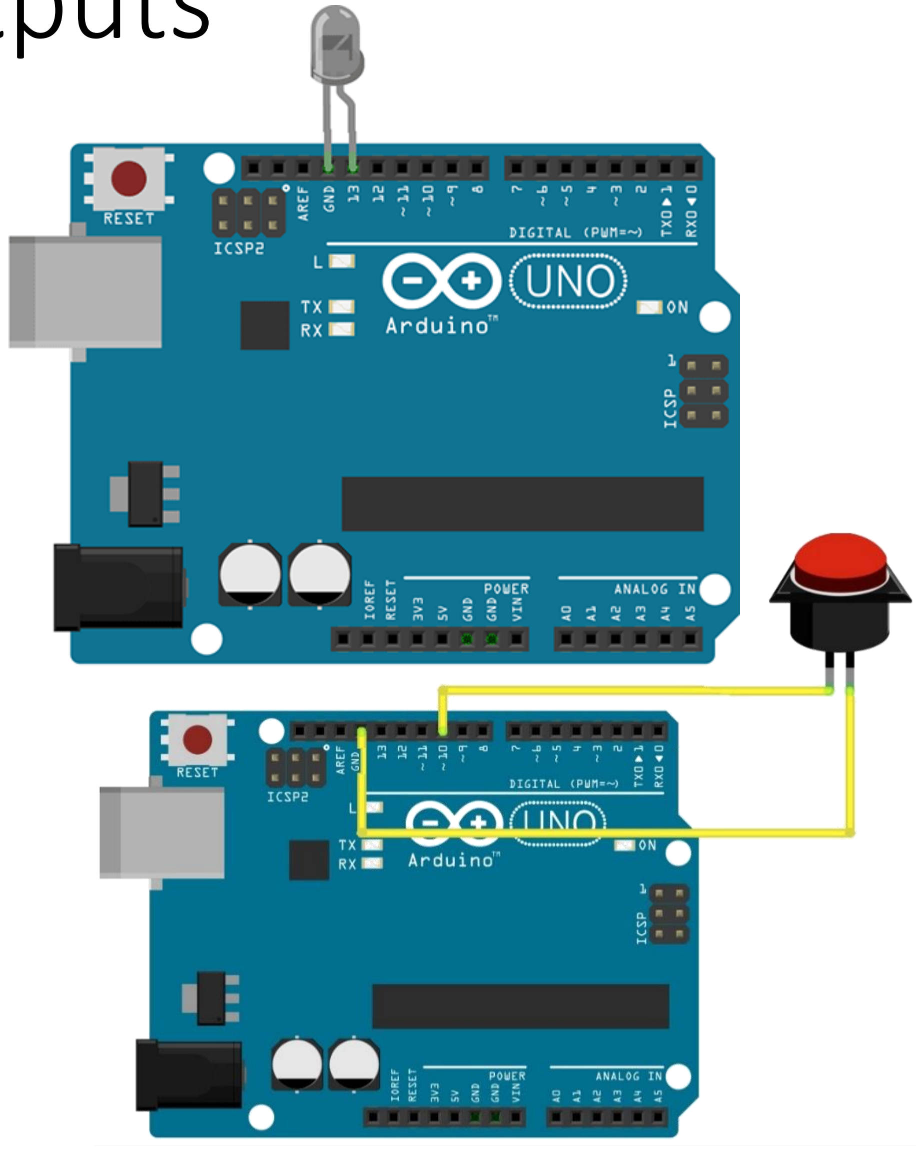
Preparing your development platform

- Follow the following steps in order to get started

1. **Copy** the folder containing Activities and Challenge Code
   - Folder include a sub-folder “Activities” containing source code for all activities.
   - Folder include a sub-folder “Challenge” containing source code for the challenge.
2. **Connect** the board to your computer via the UBS cable
3. Launch the Arduino IDE
4. Select your board (Arduino UNO): **Tools->Board**
5. Select the right Serial Port: **Tools->Port**

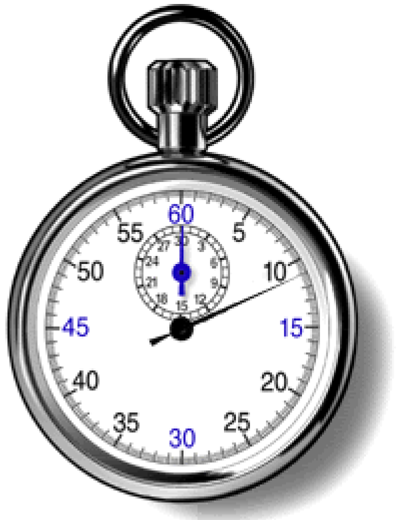
Knowing Digital Inputs and Outputs

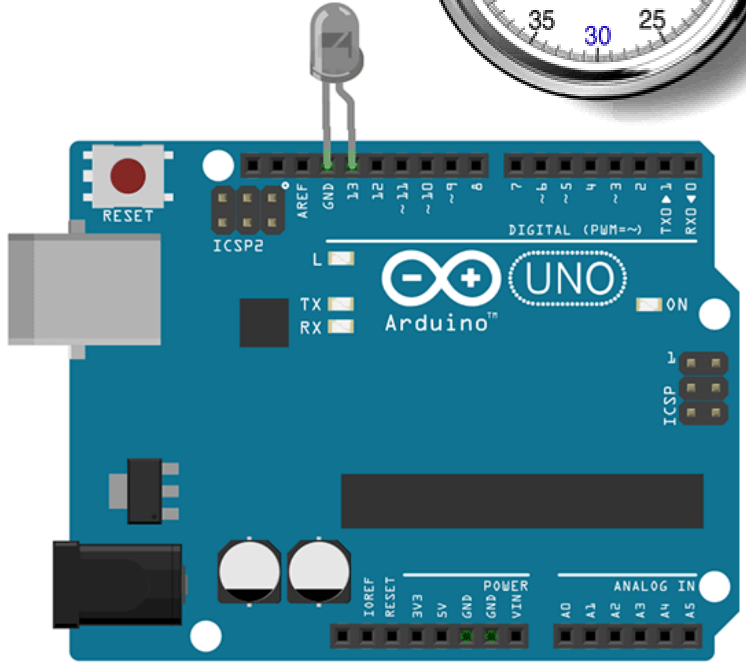
- 14 pins which can be configured as INPUT pin(s) or as OUTPUT pin(s) using following command:
  pinMode(PinNo, Direction)
- If a pin is configured as output then its state can be changed to HIGH/LOW by following command :
  digitalWrite(PinNo, State)
- If a pin is configured as input then its state, HIGH/LOW, can be read by the following command:
  digitalRead(PinNo)
- If a pin is configured as input then either an external pull-up resistor must be attached or the internal pull-up resistor must be enabled.



Timers

- Uses internal hardware counter to measure the pulses of clock source, oscillator.
- Use the following function to stop the execution of following instructions for specified time
  - **delayMicroseconds(uSec)**
  - **delay(mSec)**
- Use the following function to determine the number of milli/micro seconds elapsed since reset/startup
  - **millis()**
  - **micros()**

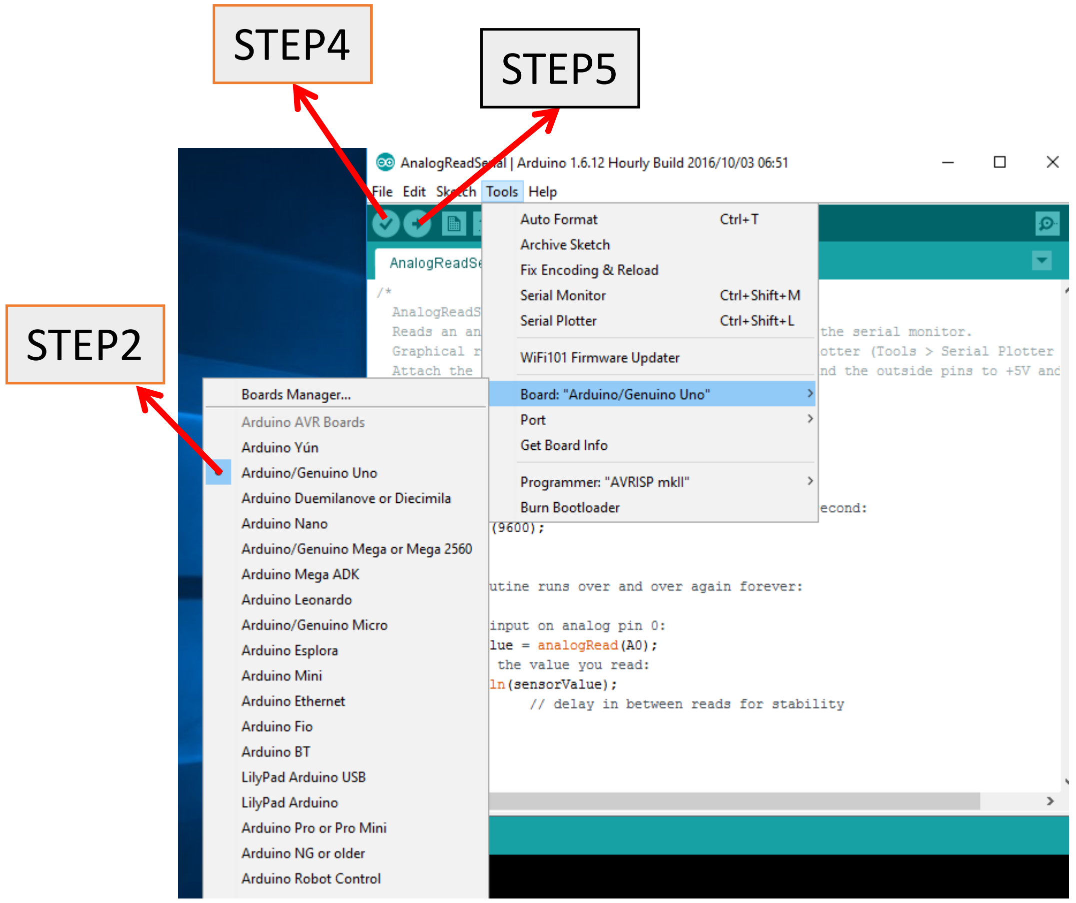




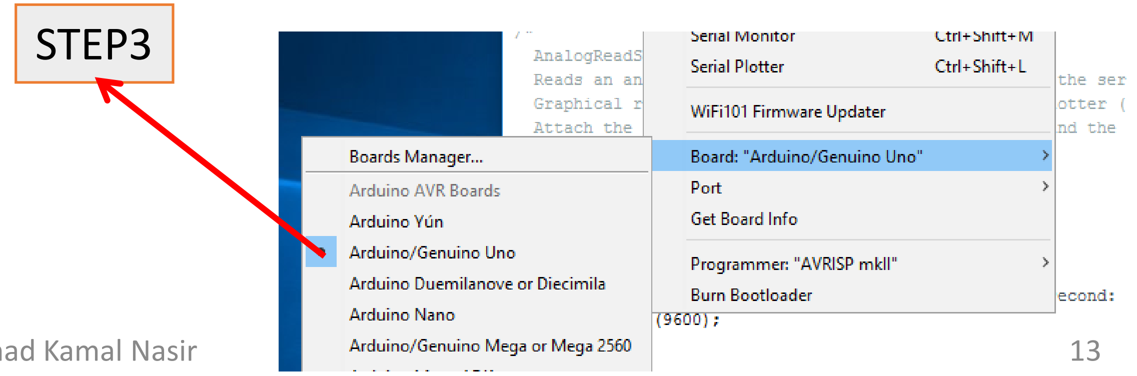
Activity 1: Controlling digital devices

- **Step 1**: Open **Activity 1** from Activities folder.
- **Step 2**: Select correct Arduino **Board**
- **Step 3**: Select correct COM **port**
- **Step 4:** Compile Program
- **Step 5:** Upload program in Arduino

✓Observe the blinking of LED on PIN 13 of Arduino Board



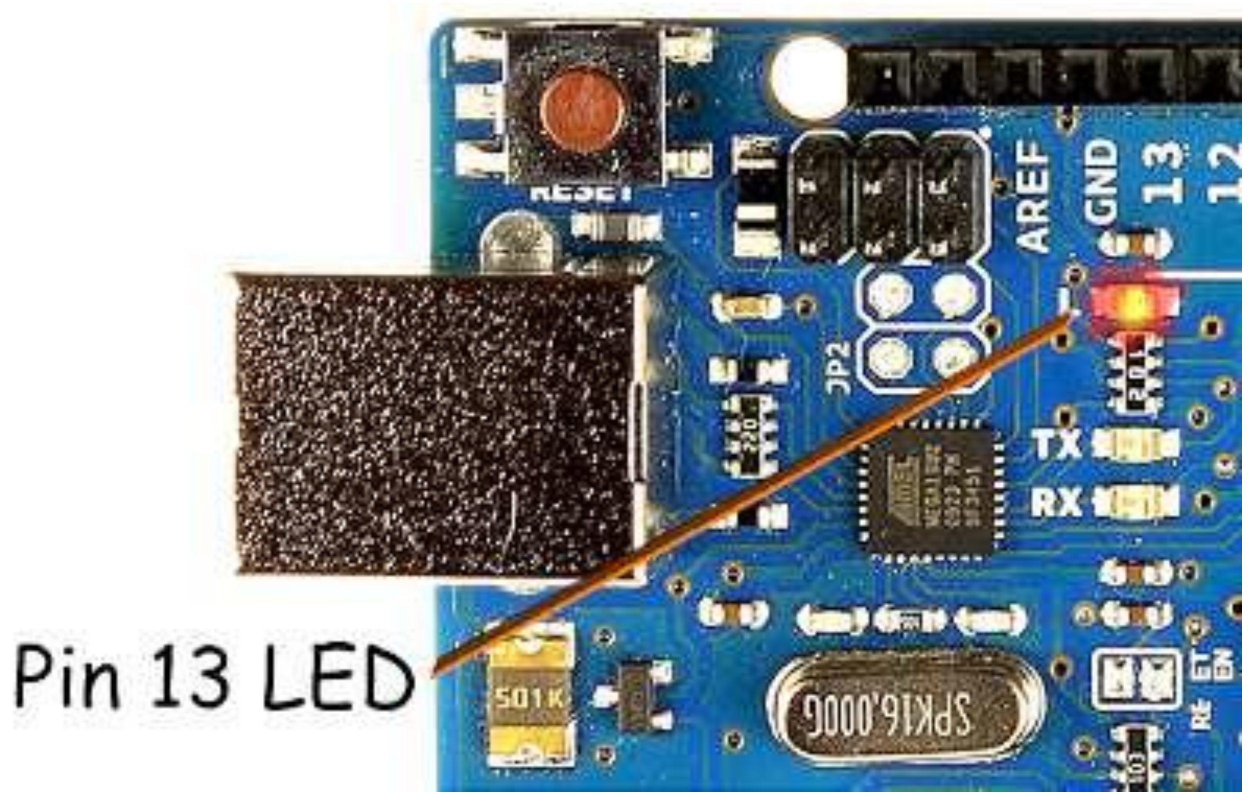
```
/*
  AnalogReadS
  Reads an an
  Graphical r
  Attach the
                the serial monitor.
                otter (Tools > Serial Plotter
                nd the outside pins to +5V anc
                econd:
(9600);

utine runs over and over again forever:

input on analog pin 0:
lue = analogRead(A0);
 the value you read:
ln(sensorValue);
     // delay in between reads for stability
```


STEP3
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+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

```
AnalogReadS
Reads an an
Graphical r
Attach the
                the ser
                otter (
                nd the
                econd:
(9600);
```

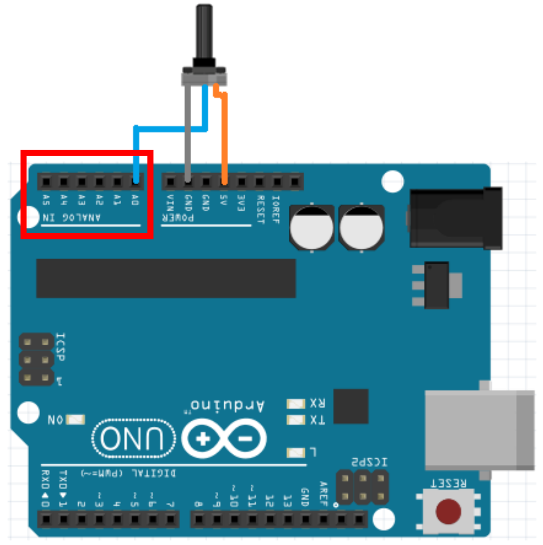
Activity 1: Controlling digital devices

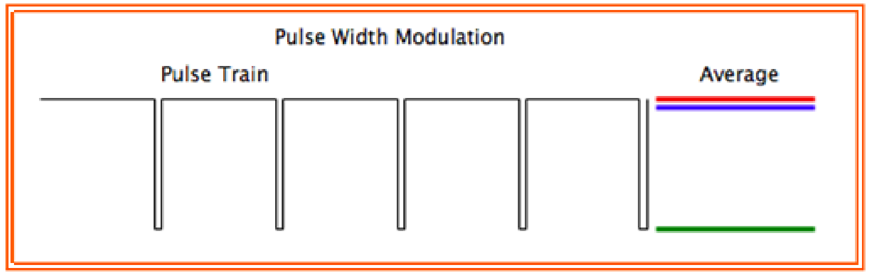
```
const int LED = 13;  // Pin number

void setup() {
  pinMode (LED, OUTPUT);
}

void loop() {
  digitalWrite (LED, HIGH);
  delay (500);
  digitalWrite (LED, LOW);
  delay (500);
}
```



Activity 2: Controlling analog devices

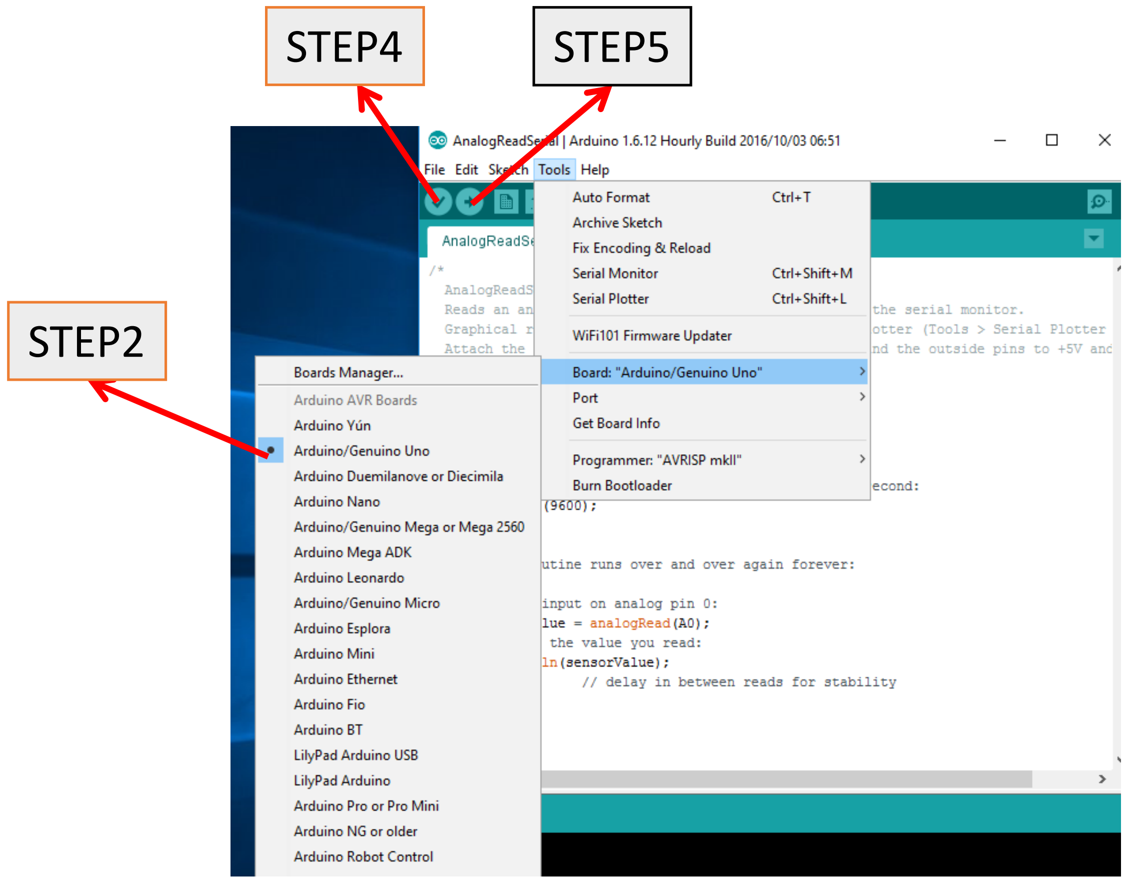
- 16 Analog inputs A0-A15, by default it measure voltage between Gnd (0V) to VCC (5V)
- Each provides 10 bit of resolution (0-1023) i.e. 4.9mV/Unit
- It takes 100uSec to read therefore maximum readings rate is 10KHz
- Use the following command to read analog input:
  analogRead(PinNo)
- No built-in DAC (Digital to Analog Converter)
  - Analog output can be created by connecting a low pass external RC filter to one of the fifteen PWM output pin using following command:
    analogWrite(PinNo, Value)





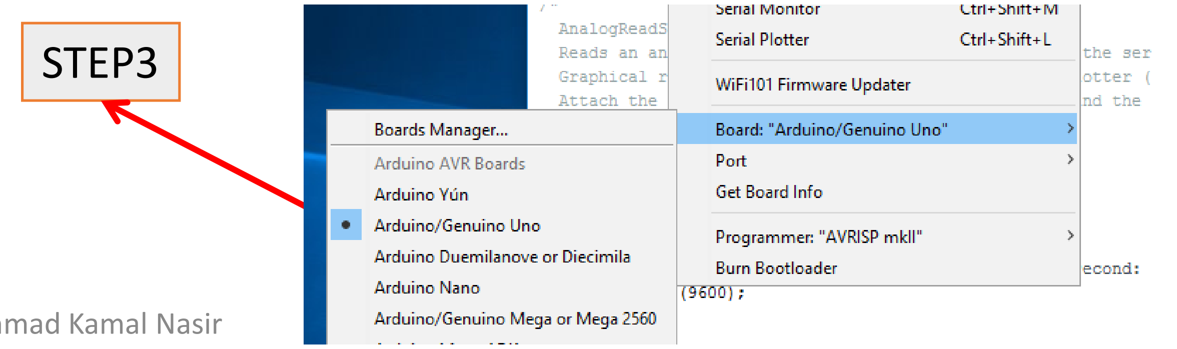
Activity 2: Controlling analog devices

- **Step 1**: Open **Activity 2** from Activities folder.
- **Step 2**: Select correct Arduino **Board**
- **Step 3**: Select correct COM **port**
- **Step 4:** Compile Program
- **Step 5:** Upload program in Arduino



```
/*
  AnalogReadS                                   
  Reads an an                                    the serial monitor.
  Graphical r                                    otter (Tools > Serial Plotter
  Attach the                                     nd the outside pins to +5V and

                                                 econd:
(9600);

utine runs over and over again forever:

input on analog pin 0:
lue = analogRead(A0);
 the value you read:
ln(sensorValue);
     // delay in between reads for stability
```

✓Observe the brightness of LED on PIN 13 of Arduino Board


STEP3
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+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

```
/*
  AnalogReadS
  Reads an an          the ser
  Graphical r          otter (
  Attach the           nd the
                       econd:
(9600);
```

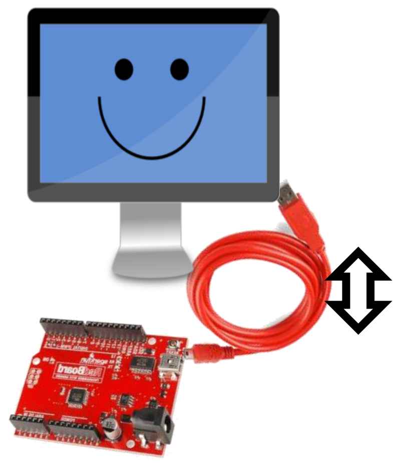
Activity 2: Controlling analog devices

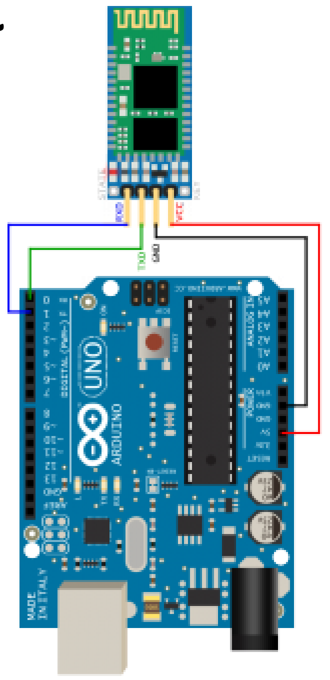
```
    const int LED = 13;         // The LED pin
    int brightness = 0;     // Brightness control
    int fadeAmount = 5; // Fading speed


void setup() {
    // declare pin 13 to be an output
      pinMode(LED, OUTPUT);
}
```

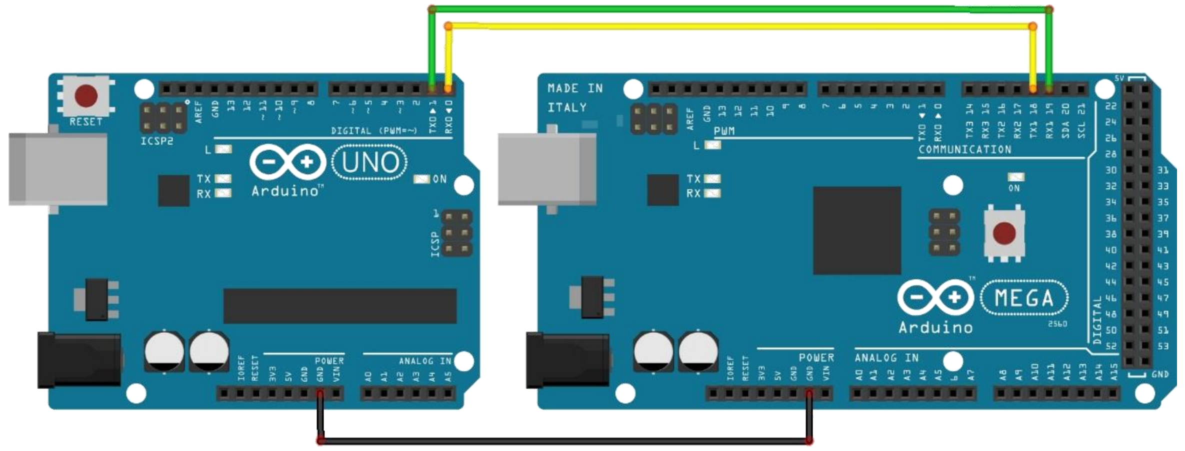
```
void loop() {
  // set the brightness of LED
  analogWrite(LED, brightness);

  // change the brightness for next iteration
  brightness = brightness + fadeAmount;

  // reverse the fade direction at the end of the cycle
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }

   delay(50);
}
```

Activity 3: Getting information from robot

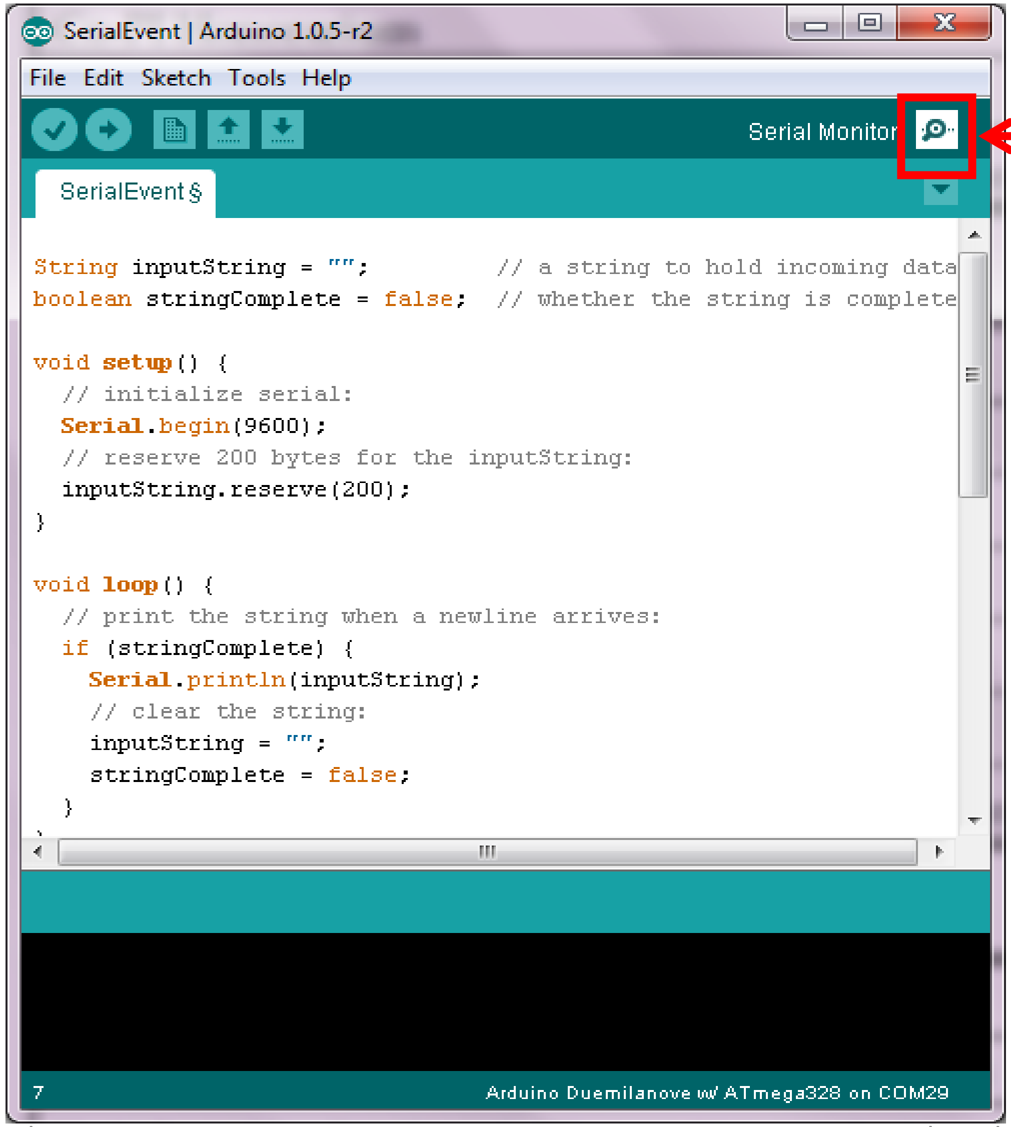
- Used to connect multiple Arduinos or an Arduino with a PC (Personal Computer) or an Arduino with your mobile phone via wireless Bluetooth device
- Data is transmitted as a **series** of zeros (‘0’) and ones (‘1’) sequentially using two pins simultaneously
  - RX is used to receive data bytes
  - TX is used to transmit data bytes
- Use the following function to configure the speed:
  Serial.begin(SpeedBPS)
- Use the following function to send data with newline character appended:
  Serial.println(StringData)
- Use the following function to read data byte by byte:
  Serial.read()







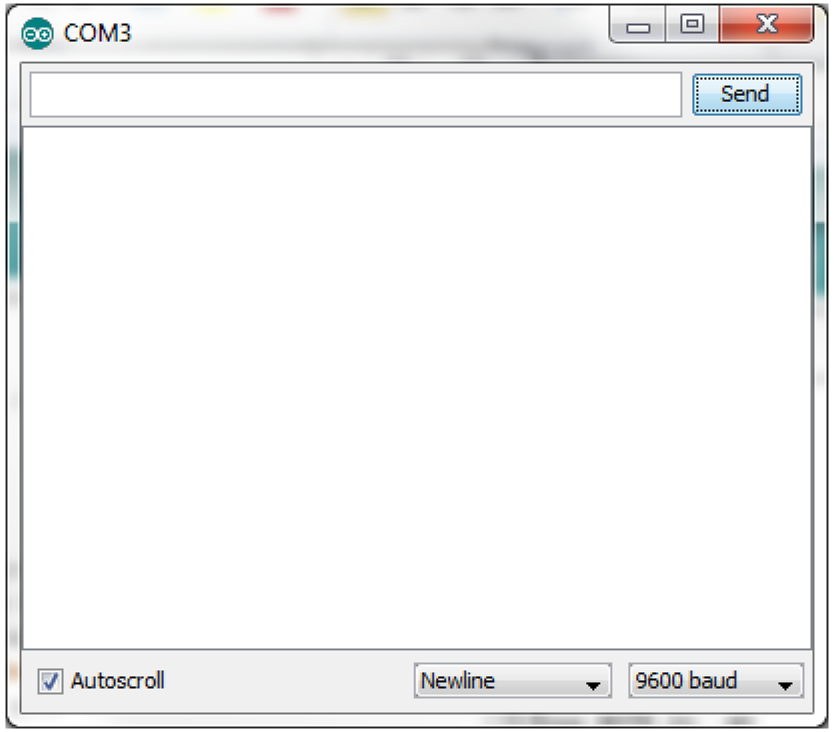
Activity 3: Getting information from robot



```
String inputString = "";         // a string to hold incoming data
boolean stringComplete = false;  // whether the string is complete

void setup() {
  // initialize serial:
  Serial.begin(9600);
  // reserve 200 bytes for the inputString:
  inputString.reserve(200);
}

void loop() {
  // print the string when a newline arrives:
  if (stringComplete) {
    Serial.println(inputString);
    // clear the string:
    inputString = "";
    stringComplete = false;
  }
```

```
7                Arduino Duemilanove w/ ATmega328 on COM29
```

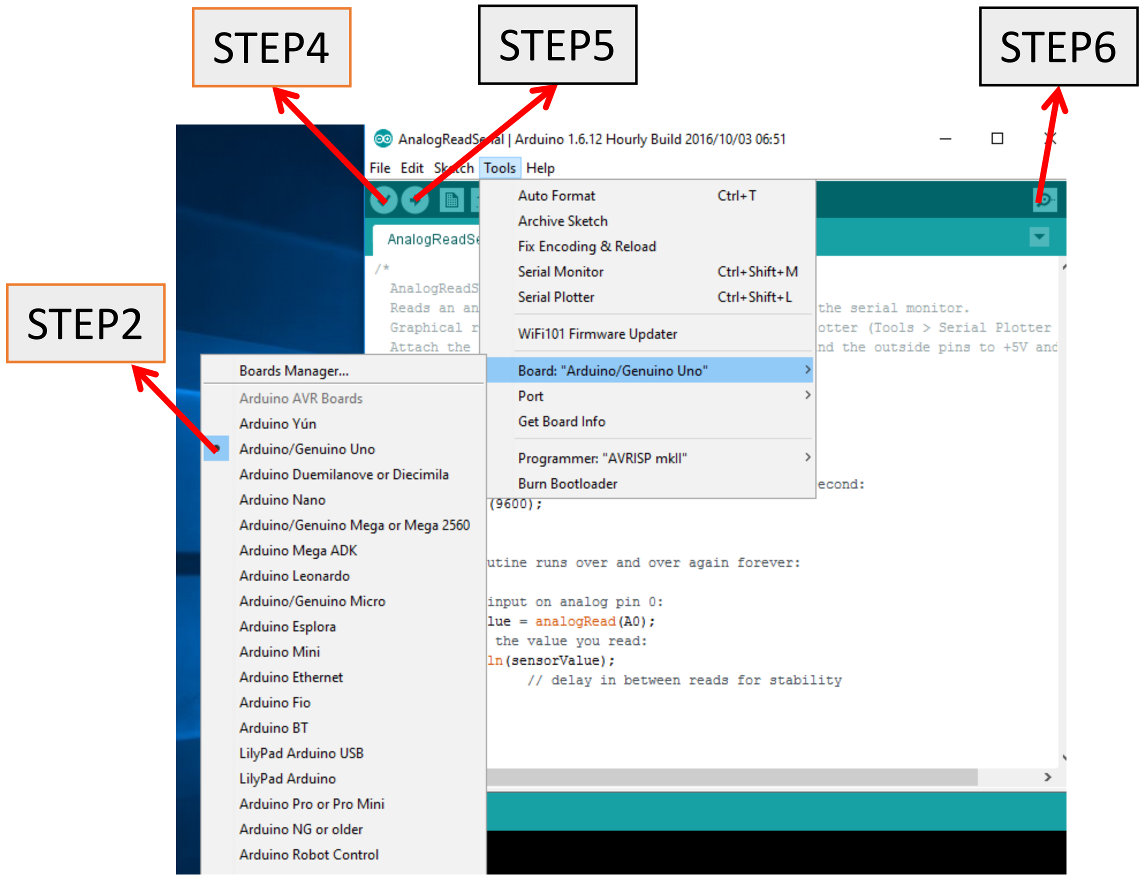
Opens up a Serial Terminal Window



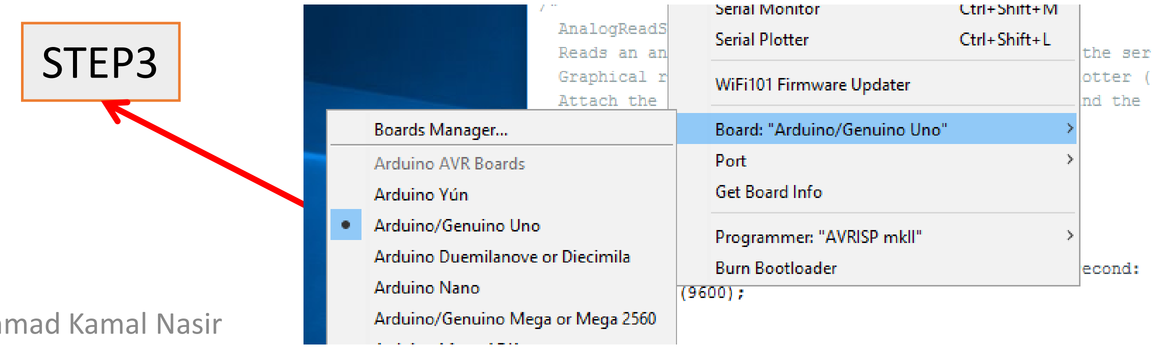
Activity 3: Getting information from robot

- **Step 1**: Open **Activity 3** from Activites folder.
- **Step 2**: Select Arduino **board**
- **Step 3**: Select COM **port**
- **Step 4:** Compile Program
- **Step 5:** Upload program in Arduino
- **Step 6:** Open Serial Monitor

✓ Change the information in **Serial.println()** instruction and upload program again.



```
/*
  AnalogReadS
  Reads an an                                  the serial monitor.
  Graphical r                                  otter (Tools > Serial Plotter
  Attach the                                   nd the outside pins to +5V and

                                               econd:
 (9600);

utine runs over and over again forever:

input on analog pin 0:
lue = analogRead(A0);
 the value you read:
ln(sensorValue);
     // delay in between reads for stability
```


STEP3
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+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

Activity 3: Getting information from robot

```
    const int LED = 13;         // The LED pin
    int brightness = 0;     // Brightness control
    int fadeAmount = 5; // Fading speed


void setup() {
    // declare pin 13 to be an output
      pinMode(LED, OUTPUT);

    // initialize serial port with specified speed

    Serial.begin(115200);

}
```

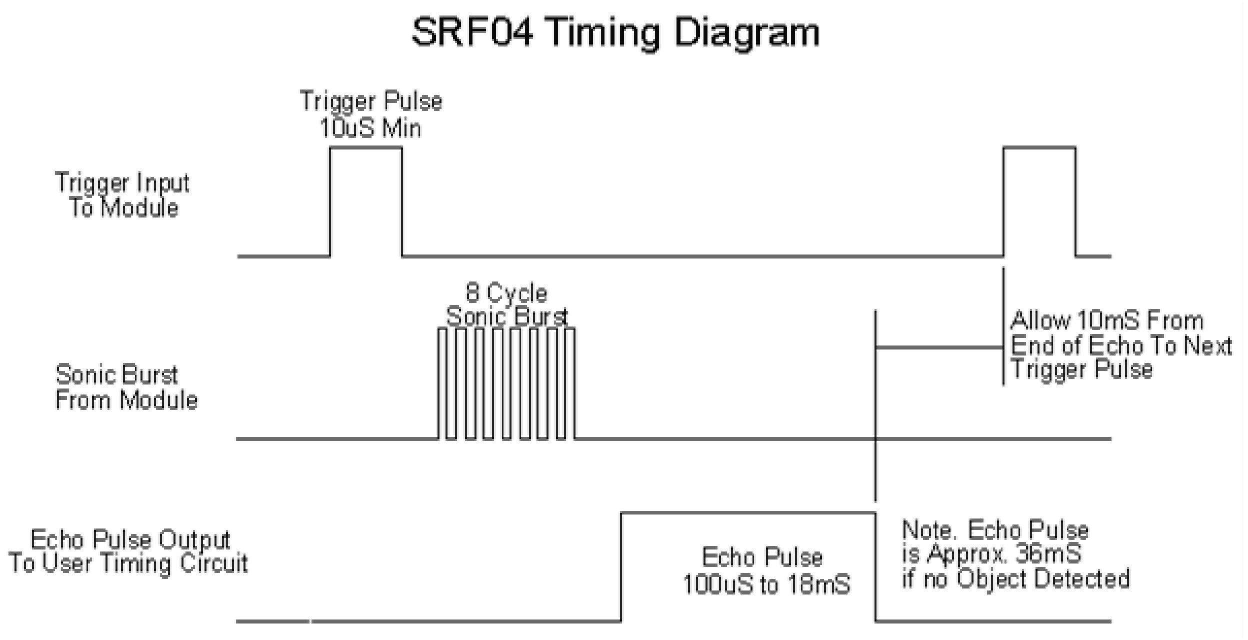
```
void loop() {
  // set the brightness of LED
  analogWrite(LED, brightness);

  // change the brightness for next iteration
  brightness = brightness + fadeAmount;

  // reverse the fade direction at the end of the cycle
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
   Serial.println (brightness);
   delay(50);
}
```

Mobile Robot: Sensors

Measuring robot’s internal or environmental parameters

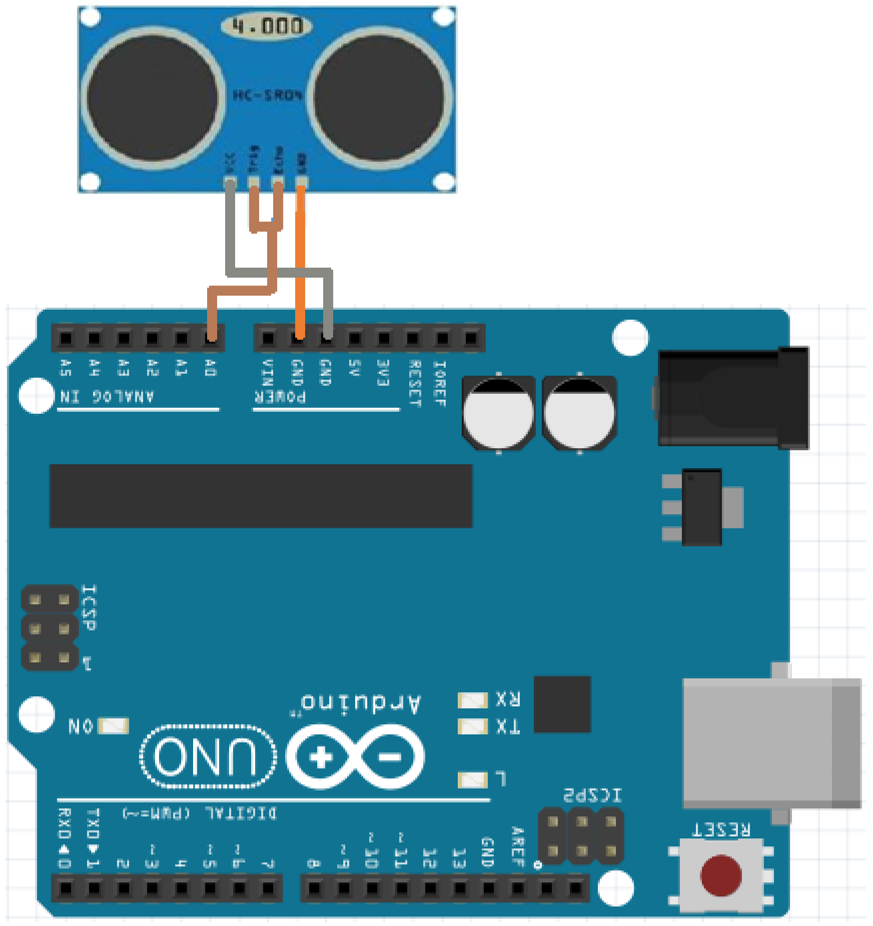
Activity 4: Obstacle Detection

- One digital input and one digital output signal is required to acquire the distance information from the sensor
- A pulse of 10uSec is generated on the Trigger pin of the sensor and the timer is started simultenously.
- When a pulse is received on the Echo pin, timer is stopped and distance is determined from the elapsed time.


SRF04 Timing Diagram
Trigger Pulse 10uS Min
Trigger Input To Module
8 Cycle Sonic Burst
Allow 10mS From End of Echo To Next Trigger Pulse
Sonic Burst From Module
Echo Pulse Output To User Timing Circuit
Echo Pulse 100uS to 18mS
Note. Echo Pulse is Approx. 36mS if no Object Detected

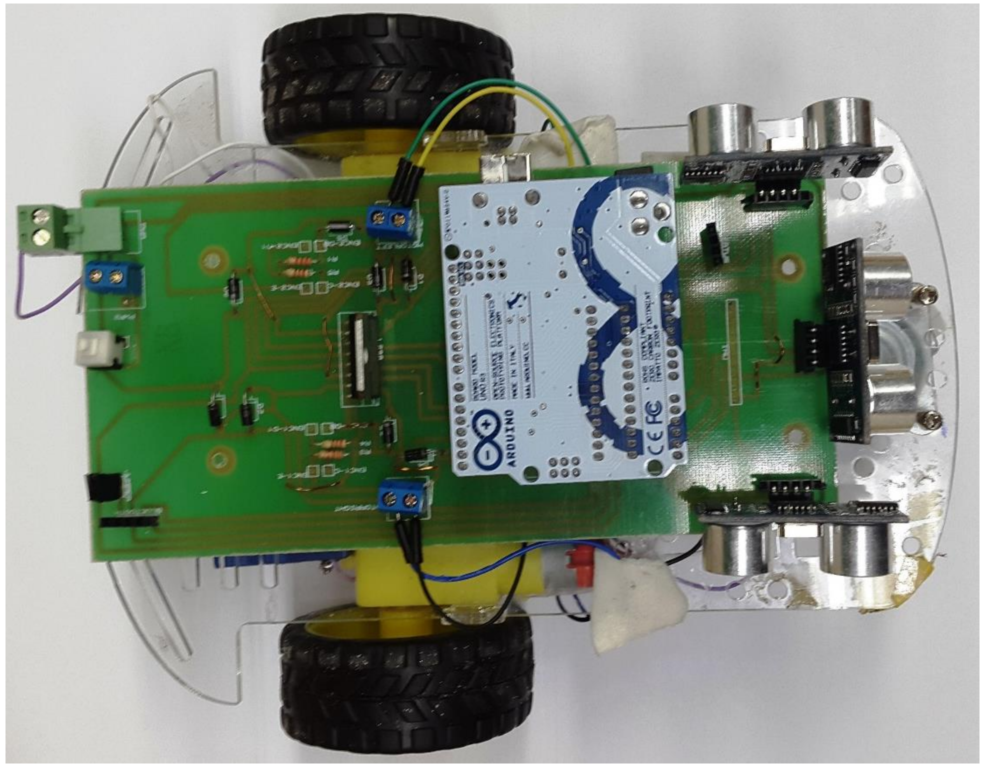
Activity 4: Obstacle Detection

- Copy the NewPing library into **Libraries** folder of Arduino IDE
- Insert all three ultra-sonic sensors on the robot PCB
- To reduce the number of pins required for three sensors, Trigger and Echo Signal are Generated by a single digital Input/Output pin of Arduino UNO.

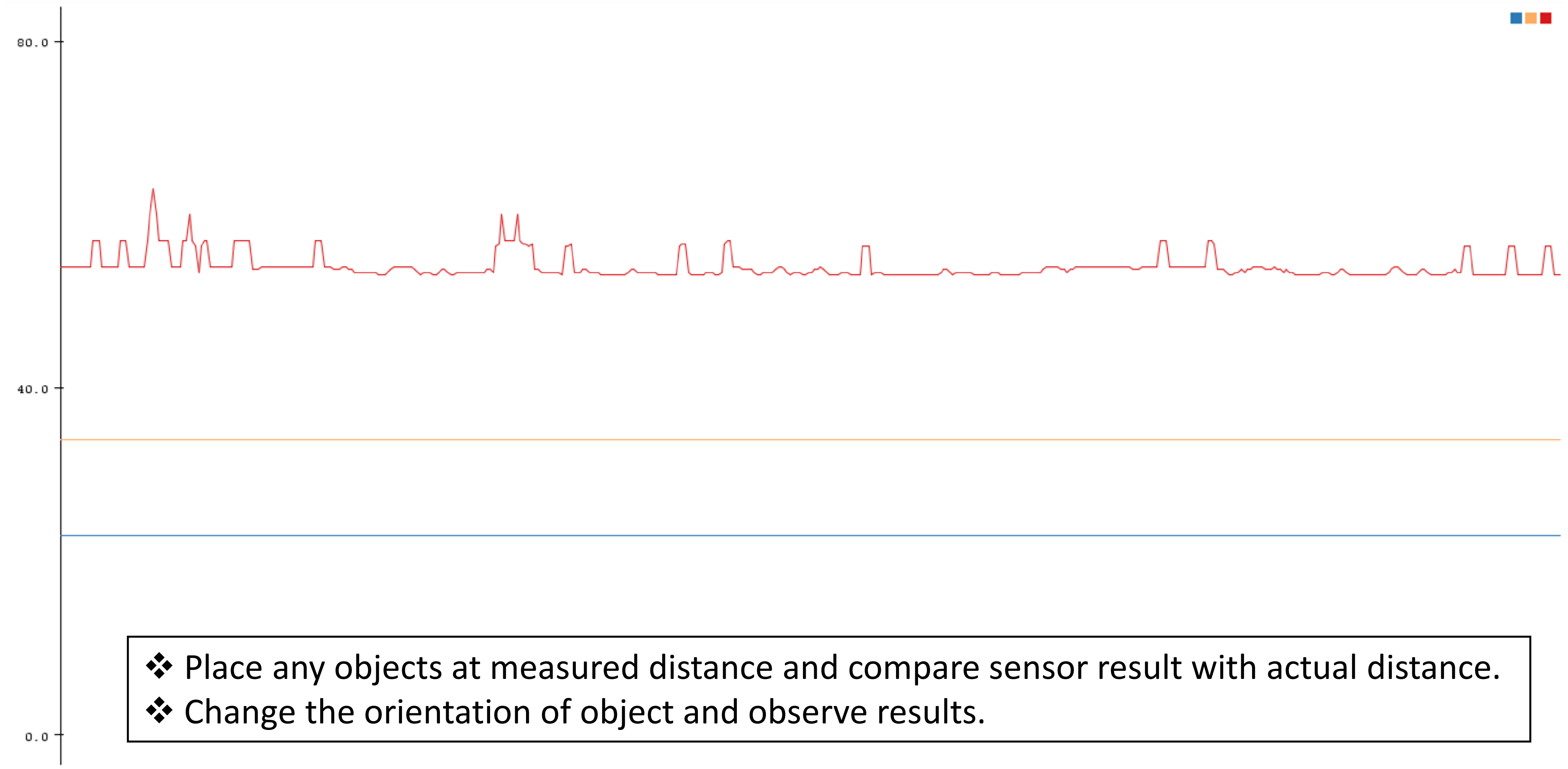


Activity 4: Obstacle Detection

- **Step 1**: Open **Activity 4** from Activities folder.
- **Step 2**: **Select** correct Arduino **board** and **port** as you did in previous activities.
- **Step 3:** Verify if all three ultra-sonic sensors are installed on PCB
- **Step 4:** Download the latest "NewPing" library from the following url https://bitbucket.org/teckel12/arduino-new-ping/downloads
- **Step 5:** Install the library by **Sketch->Include Library->Add .Zip Library**
- **Step 6:** Compile Program
- **Step 7:** Upload program in Arduino
- **Step 8:** Open Serial plotter
  Tools->Serial Plotter



Activity 4: Obstacle Detection

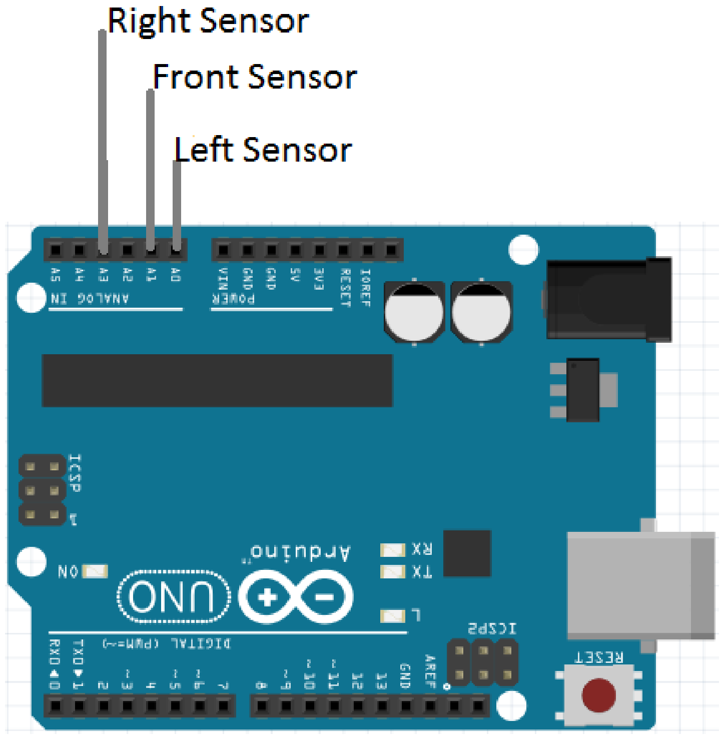


❖ Place any objects at measured distance and compare sensor result with actual distance.
❖ Change the orientation of object and observe results.

Activity 4: Obstacle Detection

```
#include <NewPing.h>

const int LEFT = A0;
const int FRONT = A1;
const int RIGHT = A3;

const int MAX_DISTANCE = 400;
unsigned long Front, Left, Right;

NewPing sonarLeft(LEFT, LEFT, MAX_DISTANCE);
NewPing sonarFront(FRONT, FRONT, MAX_DISTANCE);
NewPing sonarRight(RIGHT, RIGHT, MAX_DISTANCE);

void setup() {
   Serial.begin(115200);
   pinMode(LEFT, OUTPUT);
   pinMode(FRONT, OUTPUT);
   pinMode(RIGHT, OUTPUT);
}
```

```
void loop() {
   ReadUSSensors(&Front, &Left, &Right);
   Serial.print(Front);
   Serial.print(" " );
   Serial.print(Right);
   Serial.print(" " );
   Serial.println(Left);
}

void ReadUSSensors(long *Front, long *Left, long *Right)
{
  *Front = sonarFront.ping_cm();
  *Right = sonarRight.ping_cm() ;
  *Left = sonarLeft.ping_cm();
}
```



Mobile Robot: Actuators

Interaction with environment

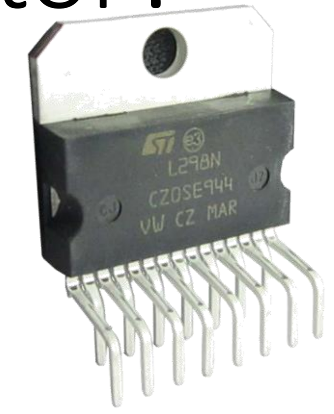
Activity 5 : Moving your robot

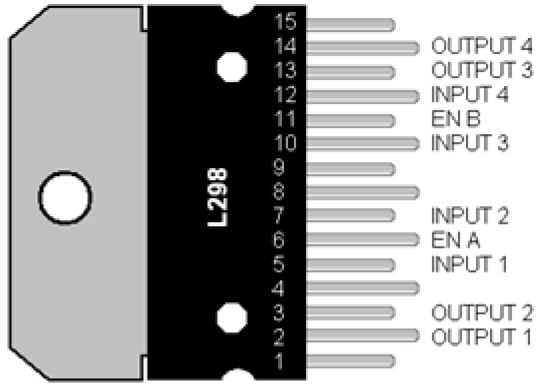
- Getting Started:
  - Move the robot on a straight line for exactly 1m
  - Rotate the robot on the same point for exactly 90° Clockwise
  - Move the robot on a square path of side 1m

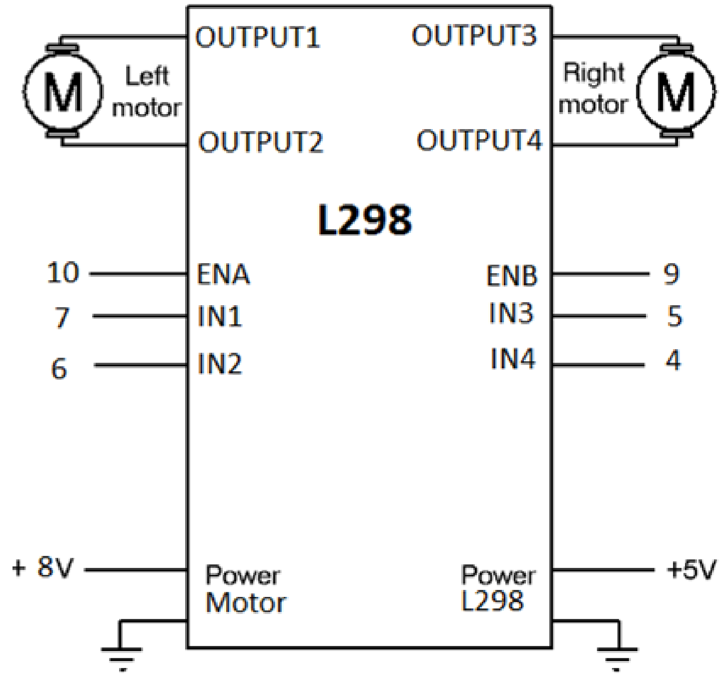
- To do:
  - Move the robot on the same square path 3 times and note down the final position during each trial. What do you observe?

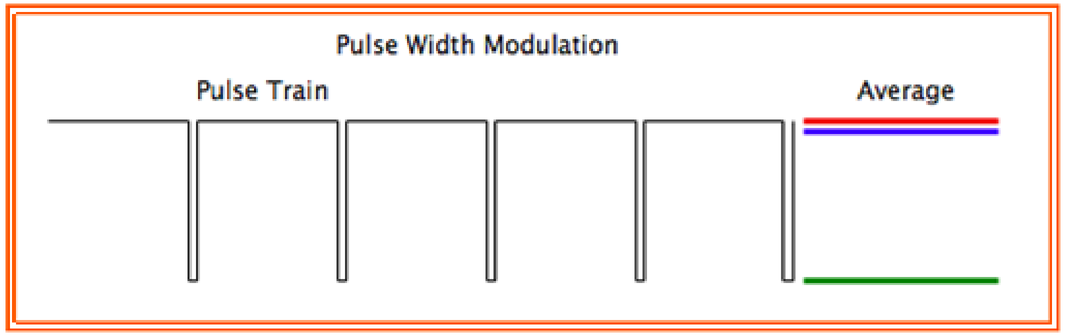
How to control a DC brush motor?

- Each motor requires three digital output pins for control
- Motor direction is controlled using changing the state of two pins (**IN1/IN2**)
- Motor Speed is controlled by changing PWM (Pulse Width Modulation) on pin (**ENA**)



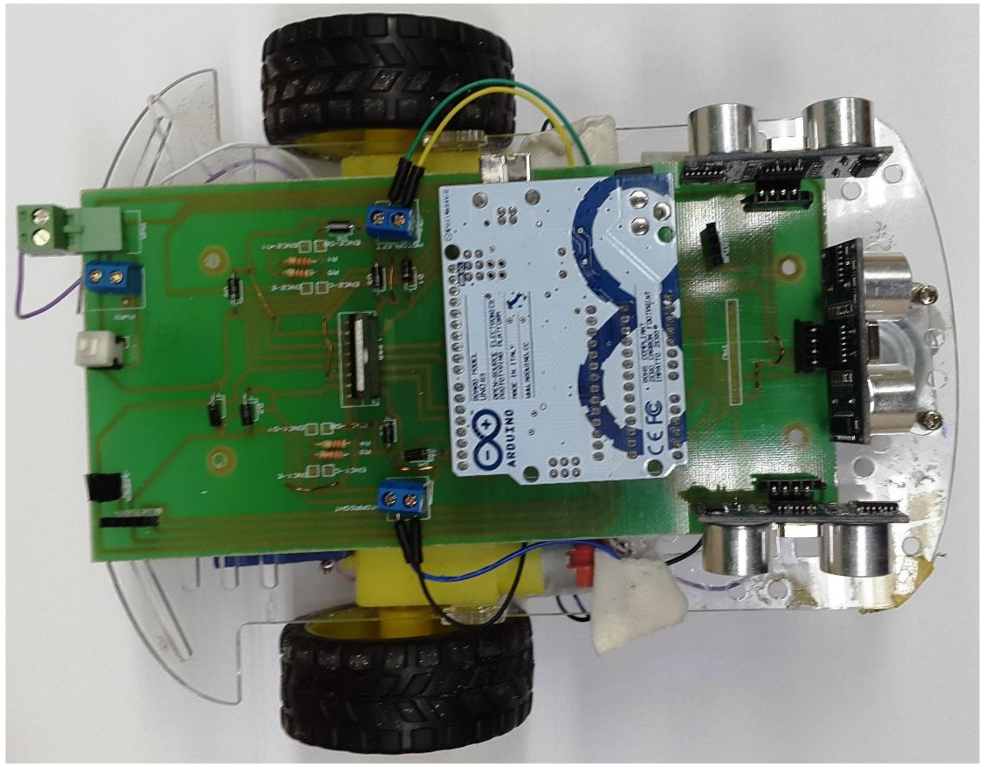






Activity 5: Moving your robot

- **Step 1**: Open **Activity 5** from Activities folder.
- **Step 2**: Select correct Arduino **Board** and COM **port** as you have done in previous activities
- **Step 3:** Verify Motor pin connections
- **Step 4:** Compile Program
- **Step 5:** Upload program in Arduino



❖Please verify that motors wires are properly connected.

Activity 5: Moving your robot

```
#define IN1 7
#define IN2 6
#define IN3 5
#define IN4 4
#define ENA 10
#define ENB 9
const int SPEED = 128;
void setup() {
  pinMode (IN2, OUTPUT);
  pinMode (IN1, OUTPUT);
  pinMode (IN4, OUTPUT);
  pinMode (IN3, OUTPUT);
  pinMode (ENA, OUTPUT);
  pinMode (ENB, OUTPUT);
}
```

```
void loop() {
  Motors ( SPEED, SPEED);
  delay(1000);
  Motors (SPEED, - SPEED);
  delay(500);
  Motors (SPEED, SPEED);
  delay(1000);
  Motors (SPEED, - SPEED);
  delay(500);
  Motors (SPEED, SPEED);
  delay(1000);
  Motors (SPEED, - SPEED);
  delay(500);
  Motors (SPEED, SPEED);
  delay(1000);
  Motors (SPEED, - SPEED);
  delay(5000);
}
```

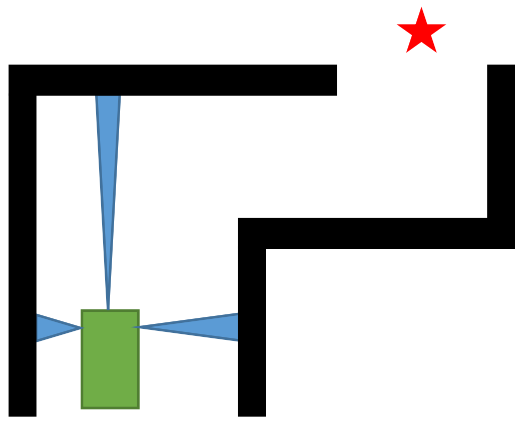
```
void Motors ( int LeftPWM, int RightPWM) {
  LeftPWM = constrain (LeftPWM, -255, 255);
  RightPWM = constrain (RightPWM, -255, 255);
  analogWrite (ENA, abs(LeftPWM));
  analogWrite (ENB, abs(RightPWM));
  if ( LeftPWM >=0){
     digitalWrite (IN1, HIGH);
     digitalWrite (IN2, LOW);
  }
  else if ( LeftPWM < 0){
     digitalWrite (IN1, LOW);
     digitalWrite (IN2, HIGH);
  }
  if ( RightPWM >= 0){
     digitalWrite (IN3, HIGH);
     digitalWrite (IN4, LOW);
  }
  else if ( RightPWM < 0){
     digitalWrite (IN3, LOW);
     digitalWrite (IN4, HIGH);
  }}
```

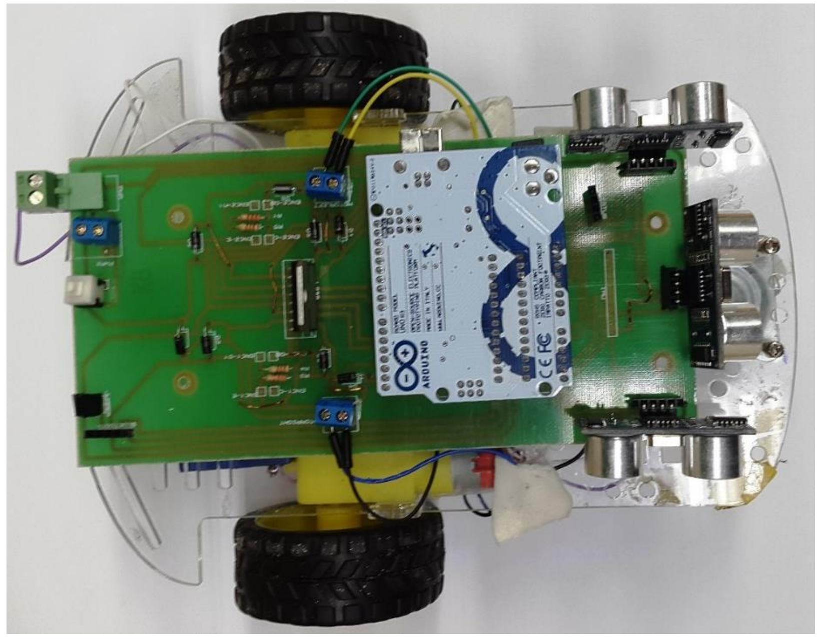
❖Please verify if robot moves in required direction. If not then motors connections accordingly.

Challenge: Wall-Following robot

- Program your wheeled mobile robot to navigate along the wall in the environment and avoid colliding with obstacles.
- Use ultrasonic sensors to determine obstacles and distance from the wall in order to decide the turning direction of the robot
- Use the provided code skeleton to program your robot

  Complete navigation conditions enclosed in **/* condition */** for your robot to follow a wall

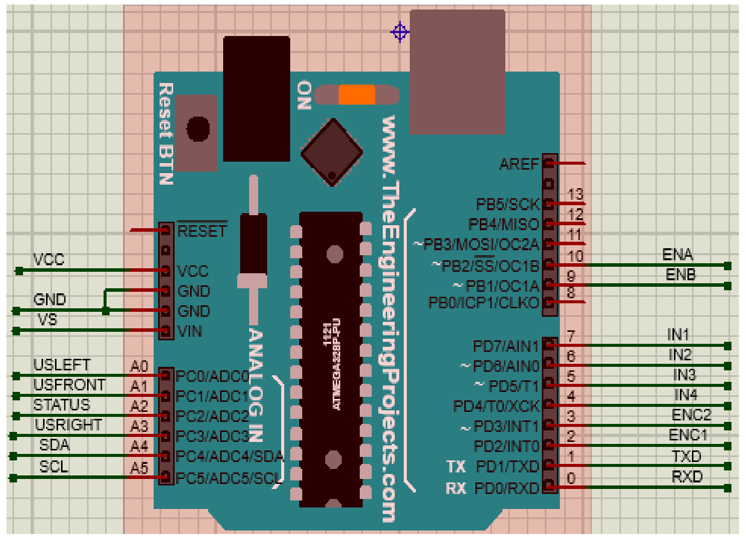




Challenge: Wall-Following robot

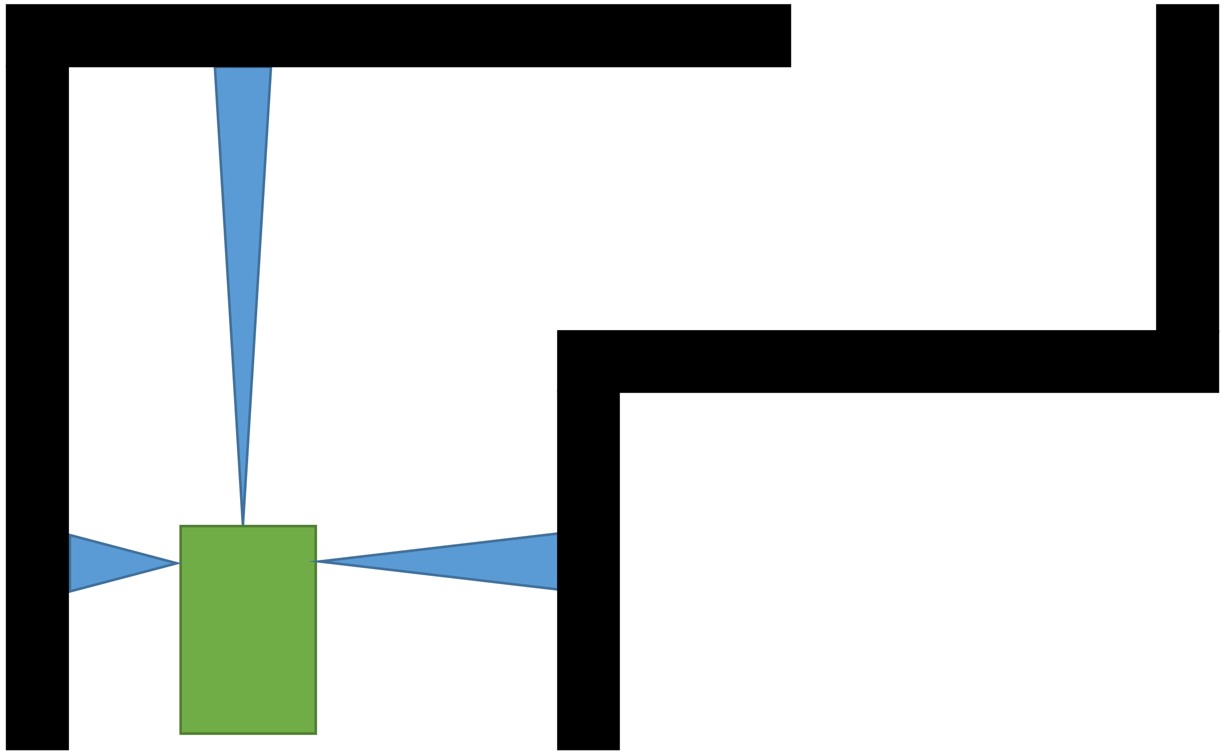
```
#include <NewPing.h>
const int LEFT = A0; // Trigger Pin
const int FRONT = A1; // Echo Pin
const int RIGHT = A3; // Echo Pin
const int IN1 = 7;
const int IN2 = 6;
const int IN3 = 5;
const int IN4 = 4;
const int ENA = 10;
const int ENB = 9;

const int MAX_DISTANCE = 400;
long Front, Left, Right;
int leftMotorSpeed = 128, rightMotorSpeed = 128;

NewPing sonarLeft(LEFT, LEFT, MAX_DISTANCE);
NewPing sonarFront(FRONT, FRONT, MAX_DISTANCE);
NewPing sonarRight(RIGHT, RIGHT, MAX_DISTANCE);
```

```
void setup()
{
   Serial.begin(115200);

   pinMode(LEFT, OUTPUT);
   pinMode(FRONT, OUTPUT);
   pinMode(RIGHT, OUTPUT);

   pinMode (ENA, OUTPUT);
   pinMode (ENB, OUTPUT);

   pinMode (IN1, OUTPUT);
   pinMode (IN2, OUTPUT);
   pinMode (IN3, OUTPUT);
   pinMode (IN4, OUTPUT);
}
```



Challenge: Wall-Following robot

```
void loop() {
 ReadUSSensors(&Left, &Front, &Right);

  if ( /*Condition to stop Robot*/ ) {
   Motors(0, 0);
 }
 else if ( /*Condition to move robot Backward */ ) {
   Motors (-leftMotorSpeed, -rightMotorSpeed);
 }
 else if ( /*Condition to move Forward*/ ) {
   Motors (leftMotorSpeed, rightMotorSpeed);
 }
 else {
   if ( /*Condition to turn robot left*/ ) {
     Motors (leftMotorSpeed, -rightMotorSpeed);
   }
   else {
     Motors (-leftMotorSpeed, rightMotorSpeed);
   }
 }
 PrintInfo ();
}
```



Challenge: Wall-Following robot

```
void ReadUSSensors(long *Front, long *Left, long *Right){
  *Front = sonarFront.ping_cm();
  *Right = sonarRight.ping_cm() ;
  *Left = sonarLeft.ping_cm();
}

void Motors ( int LeftPWM, int RightPWM) {
  LeftPWM = constrain (LeftPWM, -255, 255);
  RightPWM = constrain (RightPWM, -255, 255);
  analogWrite (ENA, abs(LeftPWM));
  analogWrite (ENB, abs(RightPWM));
  if ( LeftPWM >=0){
     digitalWrite (IN1, HIGH);
     digitalWrite (IN2, LOW);
  }
  else if ( LeftPWM < 0){
     digitalWrite (IN1, LOW);
     digitalWrite (IN2, HIGH);
  }
```

```
 if ( RightPWM >= 0){
     digitalWrite (IN3, HIGH);
     digitalWrite (IN4, LOW);
  }
  else if ( RightPWM < 0){
     digitalWrite (IN3, LOW);
     digitalWrite (IN4, HIGH);
  }}

void PrintInfo () {
  Serial.print( " LeftMotorSpeed: ");
  Serial.print(leftMotorSpeed );
  Serial.print( " RightMotorSpeed: ");
  Serial.print(rightMotorSpeed );
  Serial.print("  Left: ");
  Serial.print(Left);
  Serial.print( " Front: ");
  Serial.print(Front );
  Serial.print( "  Right: ");
  Serial.println(Right);
}
```

Schematic & Layout

ArduinoUNOShield Schematic.PDF

ArduinoUNOShield Layout.pdf