Deccan Education Society's

Kirti M. Doongursee College of Arts, Science and Commerce

[NAAC Accredited: "A Grade"]

M.Sc. [Computer Science]

Practical Journal

PAPER: PSCSP401

Roll Number [_____]

Department of Computer Science and Information Technology

Department of Computer Science and Information Technology

Deccan Education Society's

Kirti M. Doongursee College of Arts, Science and Commerce

[NAAC Accredited: "A Grade"]

C E R T I F I C A T E

This is to certify that Mr./Mrs. _____

of M.Sc. (Computer Science) with Roll No._____ has completed **9**

Practicals of Paper **PSCSP401** under my supervision in this College during the year 2022-2023.

Lecturer-In-Charge **H.O.D.**

Dept of CS & IT

Date: Date:

Examined by: **Remarks:**

Date: _____

Index

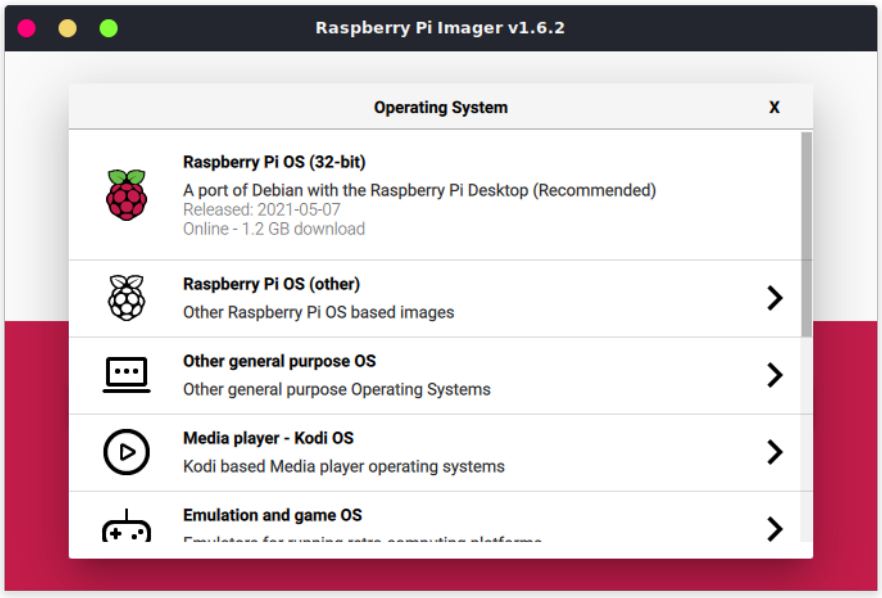
Practical No. 1

Aim: Making Raspberry pi headless, and reaching it from the network using WiFi and SSH.

Procedure:

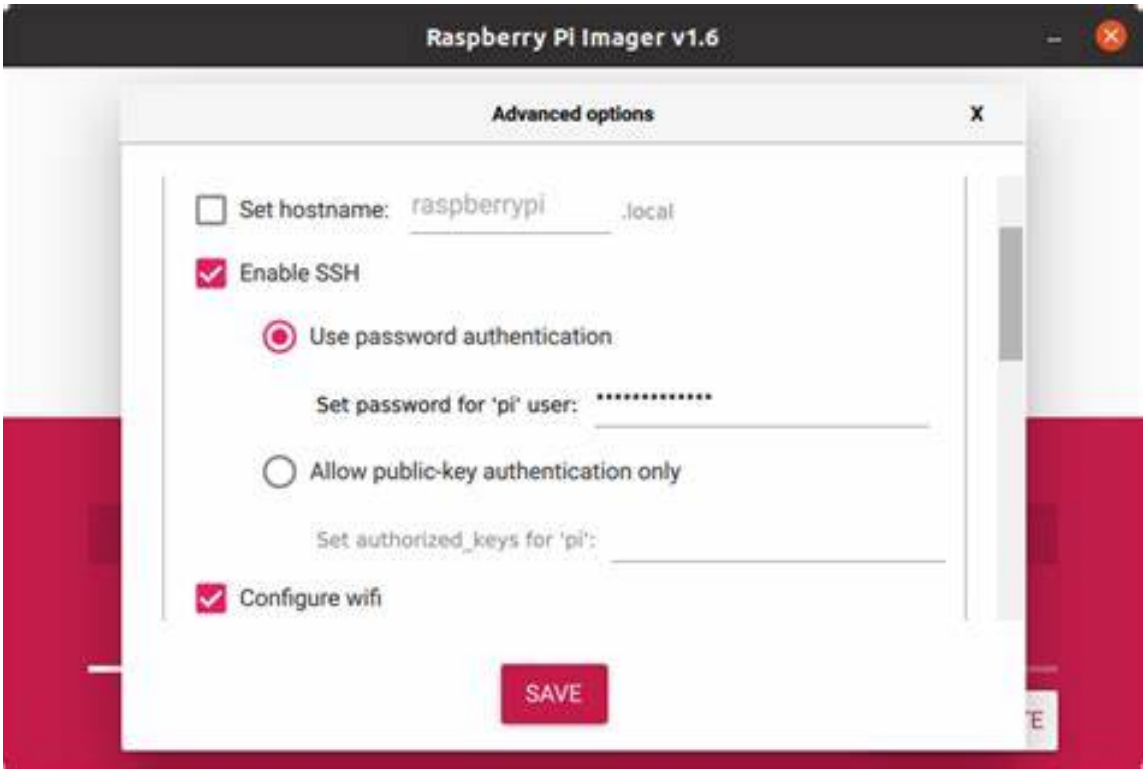
Step 1: Go to https://www.raspberrypi.com/software/ and click on the software tab.
Download raspberry pi imager and connect SD card to laptop.

Step 2: Select operating system (raspberry pi OS 32-bit)

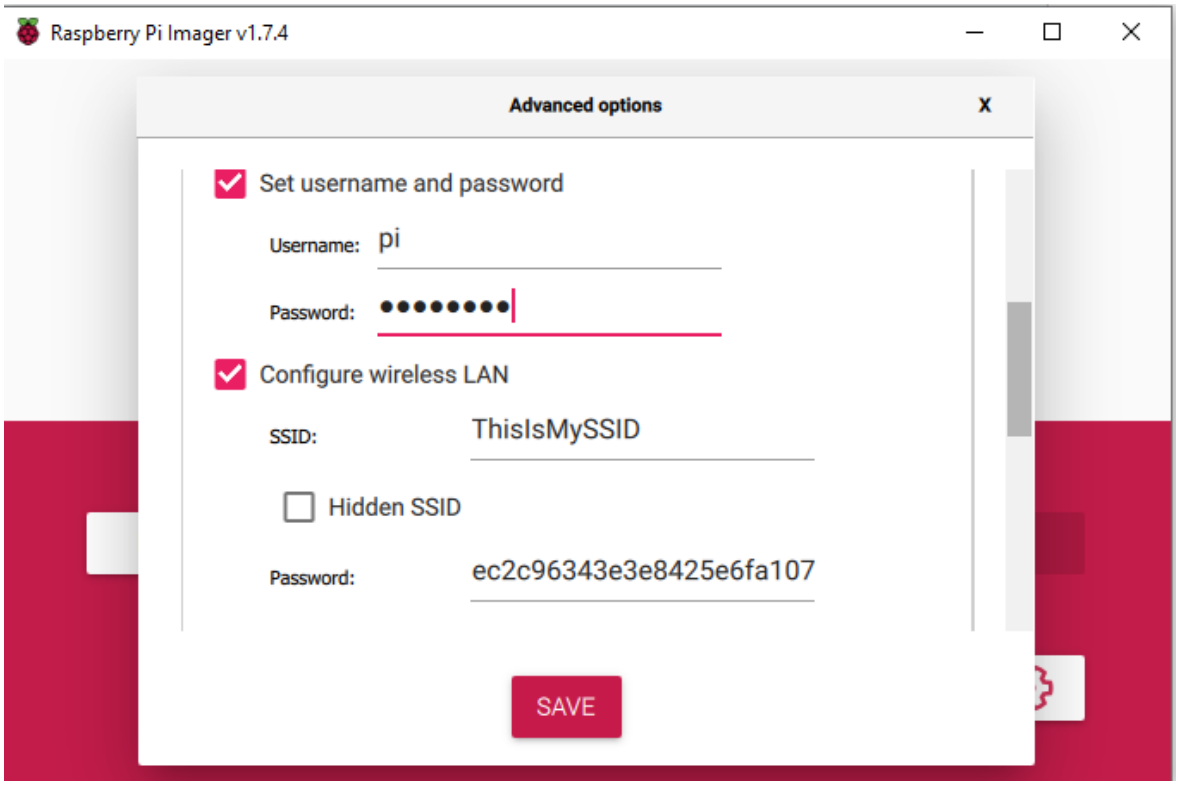


Step 3: Select storage

Step 4: Click on Setting Icon and set hostname. Click to enable SSH.

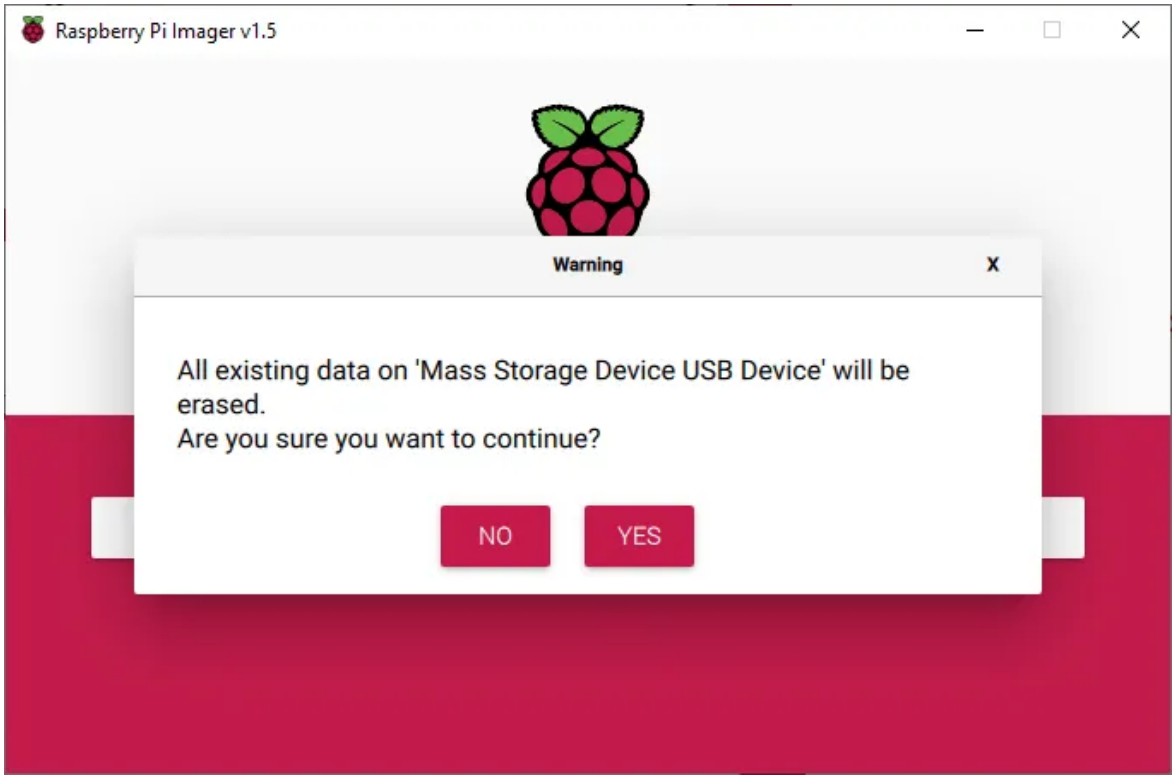


Step 5: Click to configure wireless LAN. And select a country.



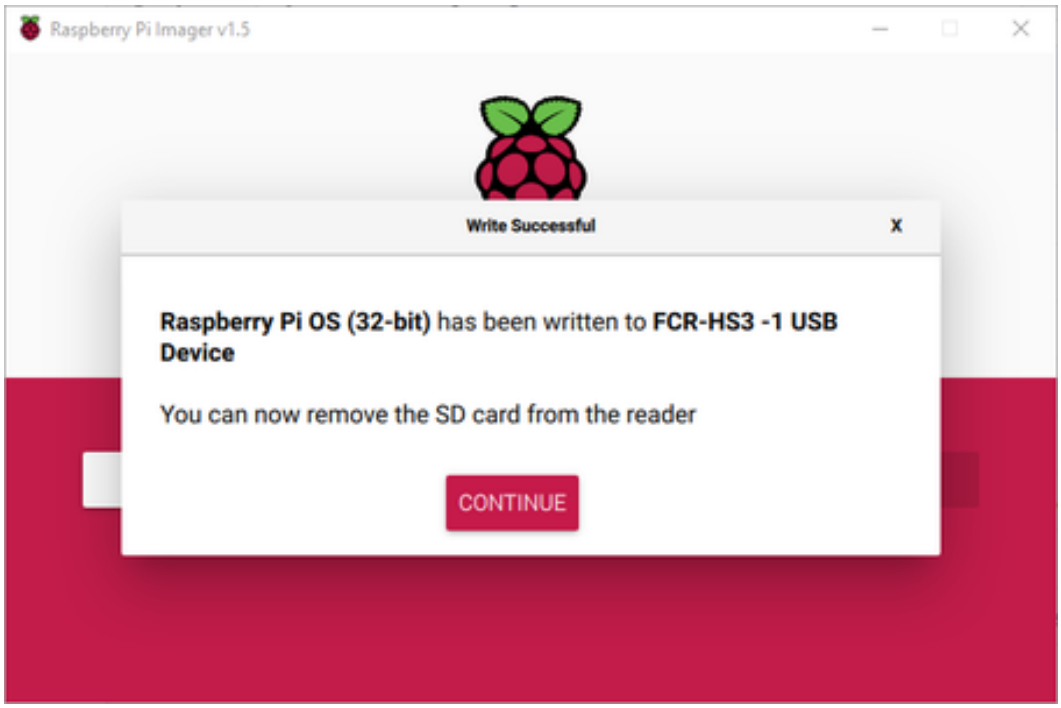
Step 6: Click on write.

Step 7: Click on yes.



Step 8: Verifying raspberry pi.

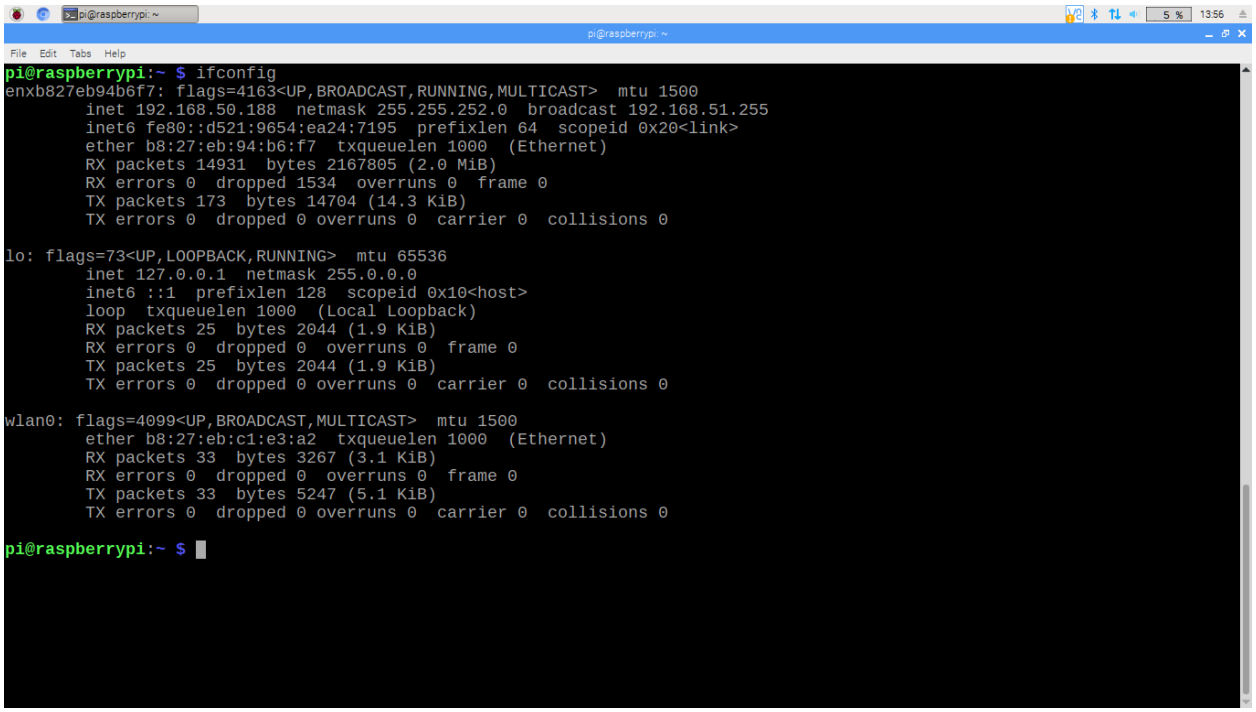
Step 9: Click on continue.



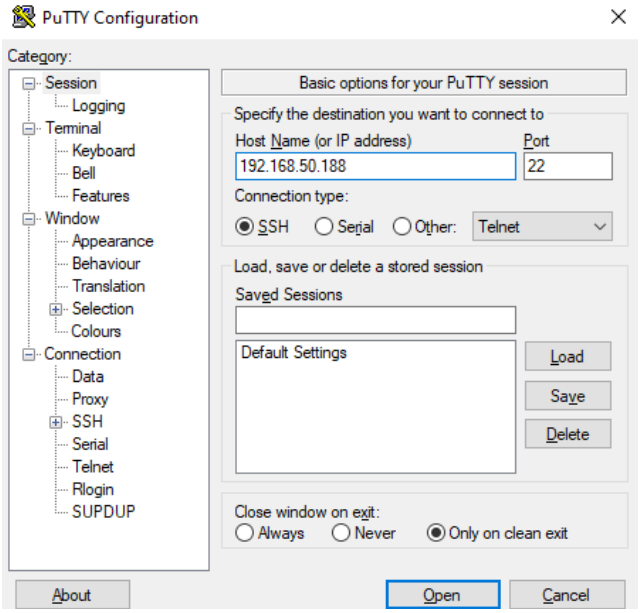
Step 10: Check the connection in your mobile hotspot of raspberry pi.

Step 11: Open terminal and execute following command:

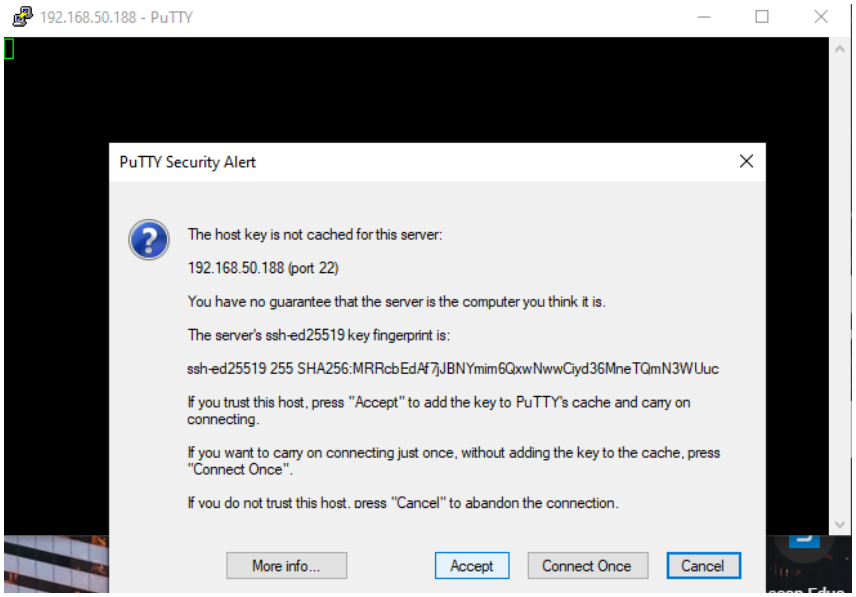
1. ifconfig



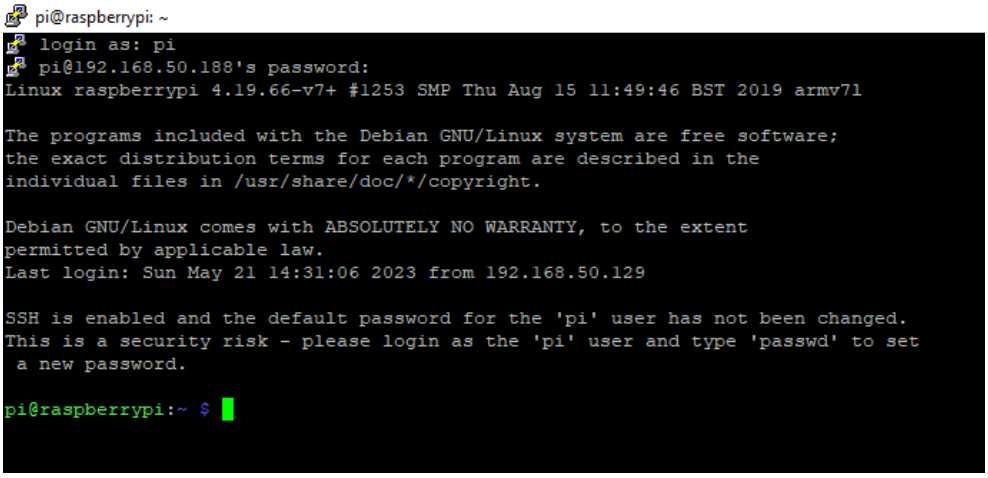
Step 12: Open Putty and add Hostname as raspberry pi ip address and click on open.

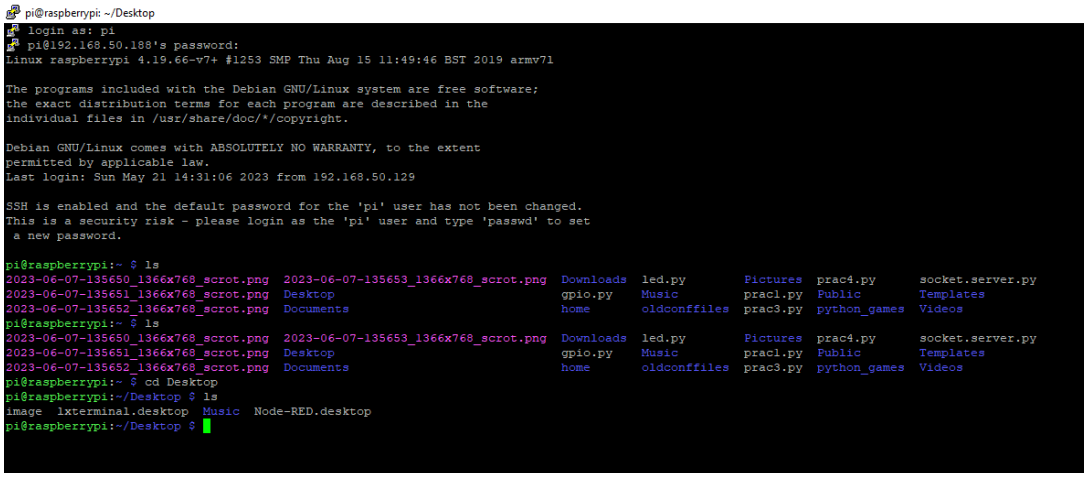


Step 13: Click on Accept



Step 14: Enter username and password:



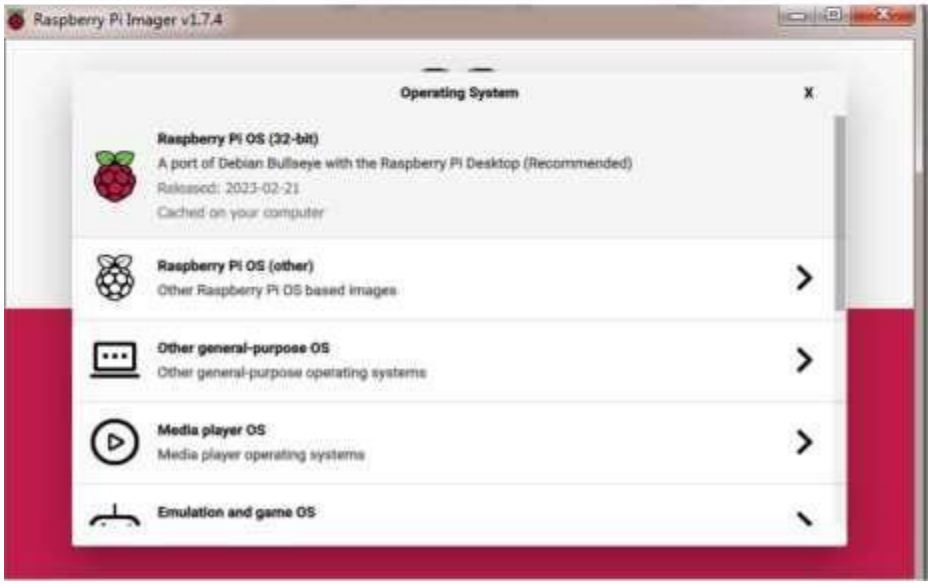


Practical No. 2

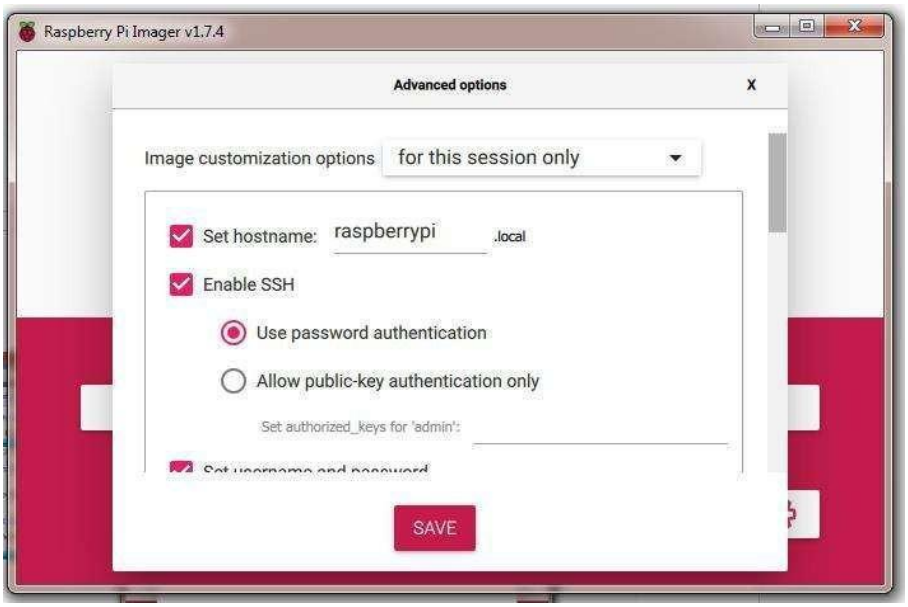
Aim: Using sftp upload files from PC.

Procedure:

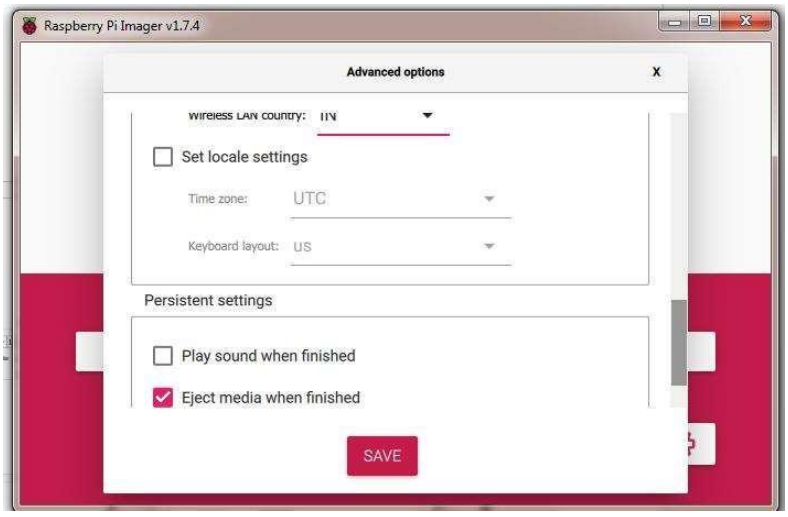
Step 1: Install the Raspberry Pi Imager

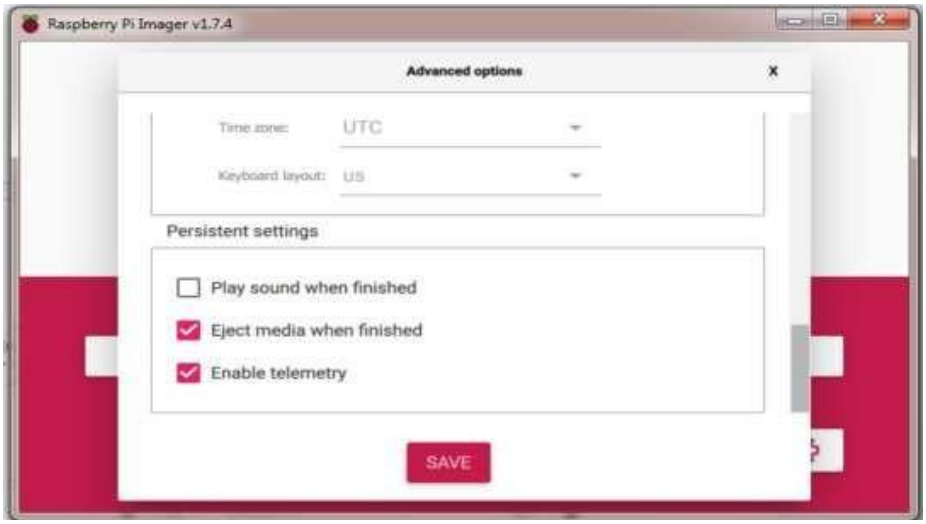


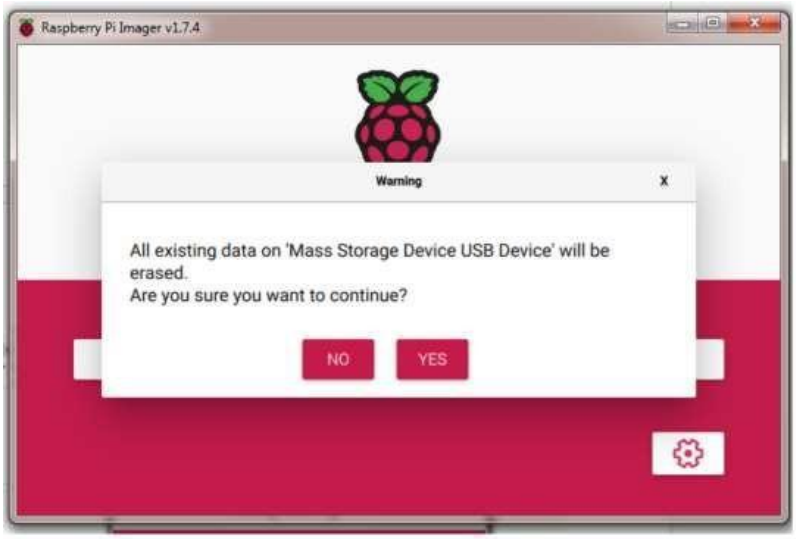
Step 2: Set Hostname , enable SSH, Set Username and Password.



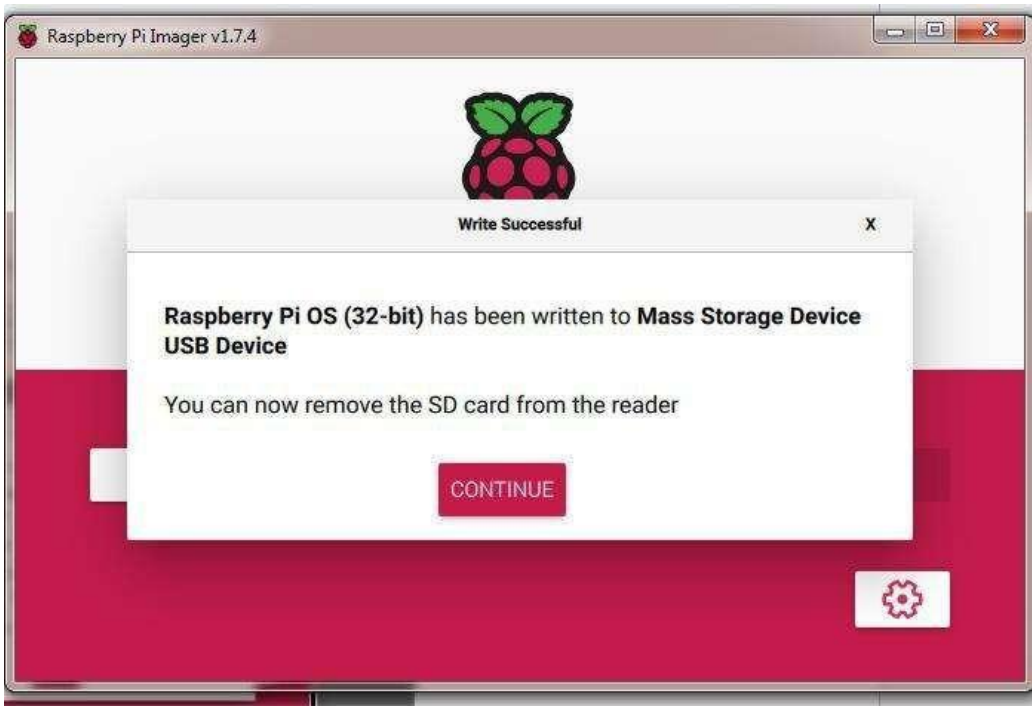
Step 3: Set SSID and Password of WiFi which is used




Raspberry Pi Imager v1.7.4
Advanced options
Time zone: UTC
Keyboard layout: US
Persistent settings
Play sound when finished
Eject media when finished
Enable telemetry
SAVE


Raspberry Pi Imager v1.7.4
Warning
All existing data on 'Mass Storage Device USB Device' will be erased.
Are you sure you want to continue?
NO
YES

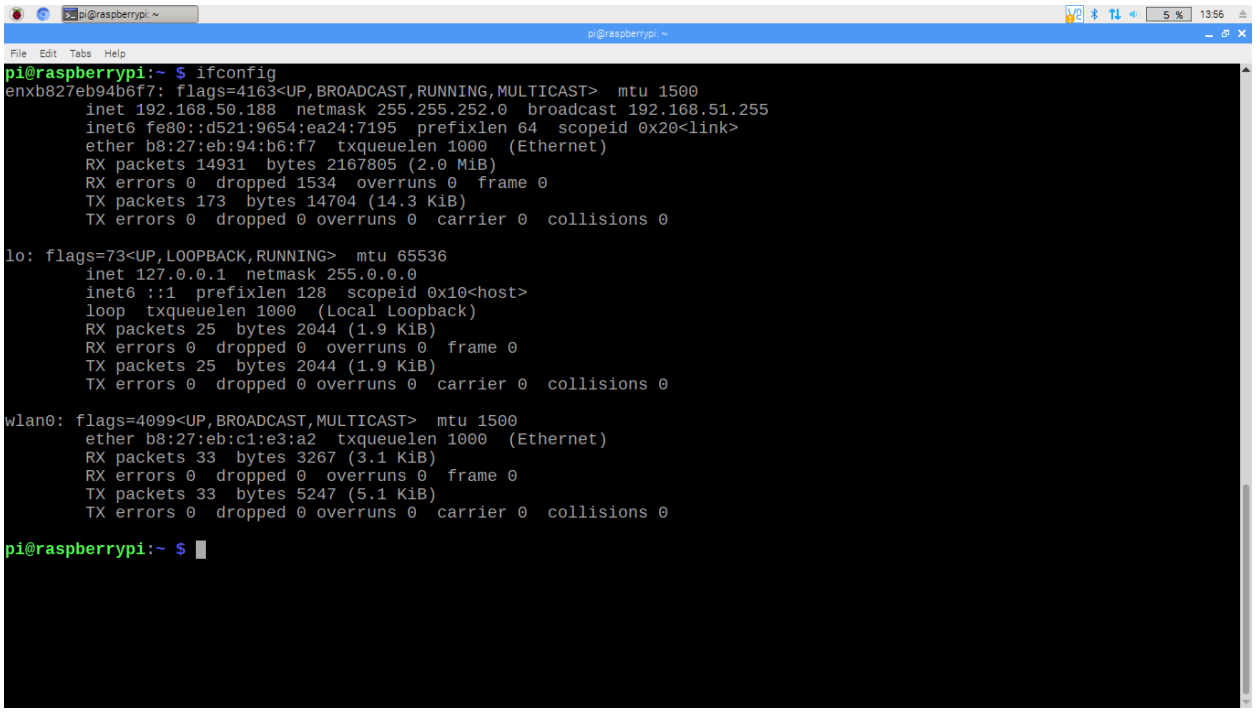

Raspberry Pi Imager v1.7.4
Raspberry Pi
Operating System
Storage
WRITE
Verifying... 6%
CANCEL VERIFY



Step 4: Connect Raspberry Pi WiFi and Laptop WiFi to Mobile device.

Step 5: Open terminal and execute following command:

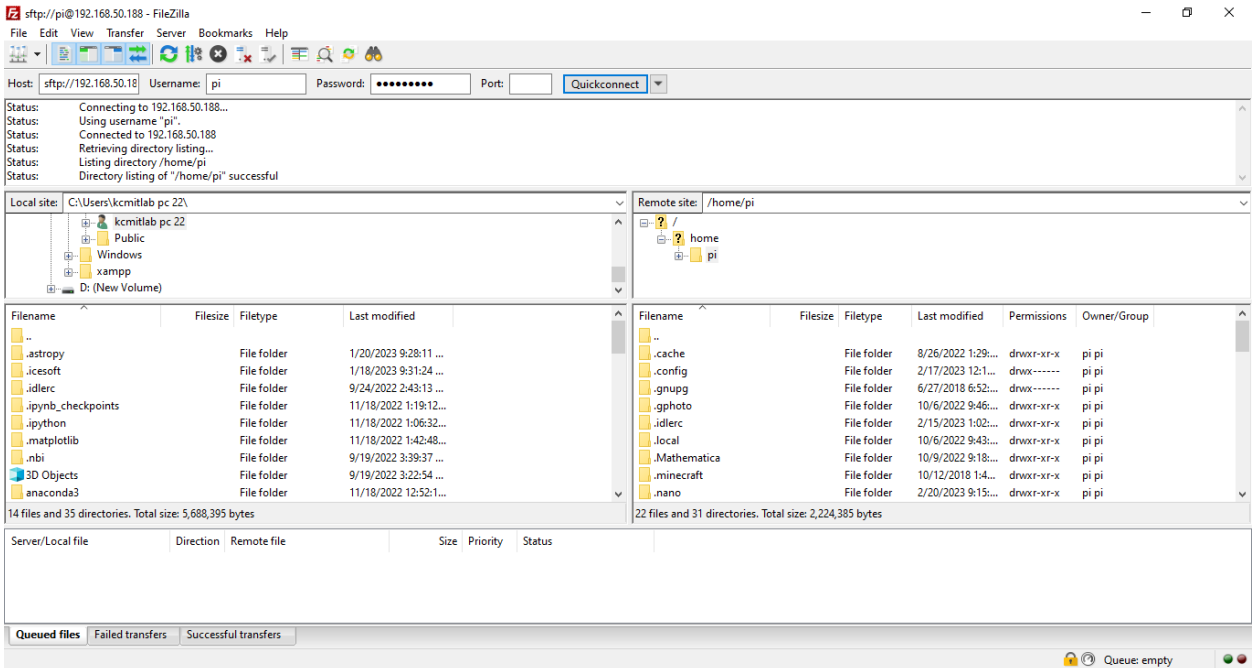
1. if config



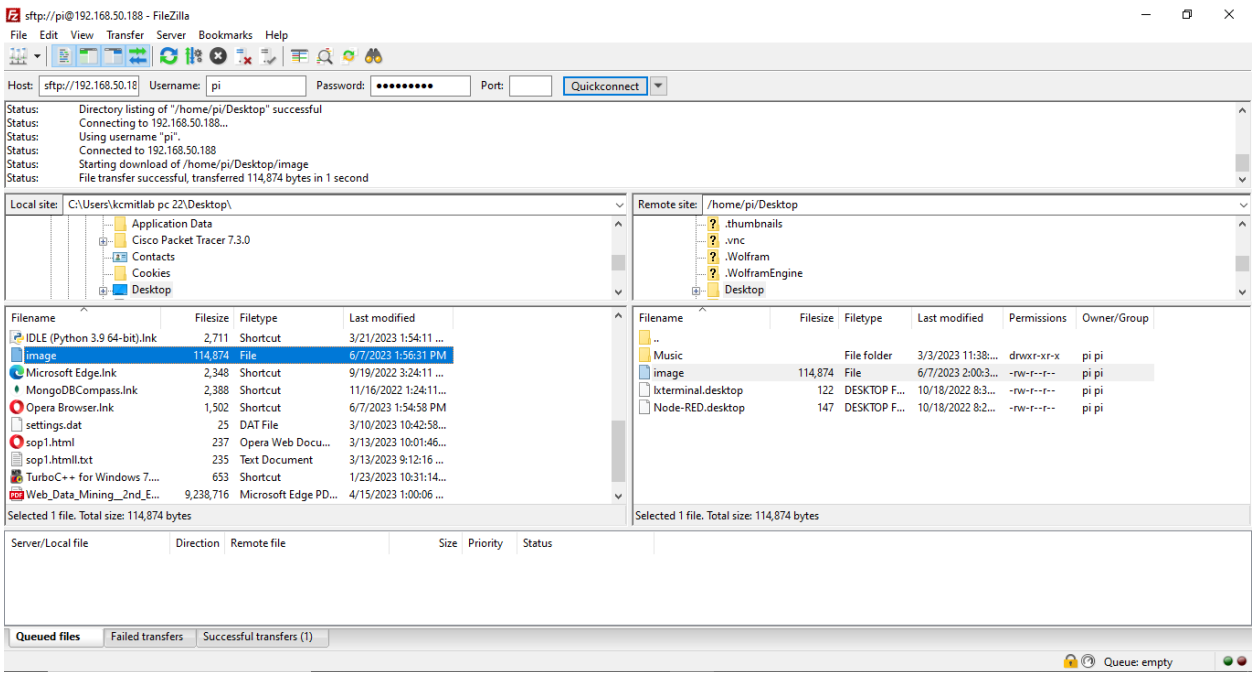
Step 6: Copy ip address and download Filezilla (Client)

Step 7: Open filezilla:

- In host paste ip address
- Username: pi
- Password: raspberry and click on Quickconnect



Step 8: Drag and Drop any file from Local site to Remote site



Practical No. 3

Aim: Write a Python code to test motor

Code:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(false)

motor_1A = 4
motor_1B = 17
motor_2A = 23
motor_2B = 24

GPIO.setup(motor_1A, GPIO.OUT)
GPIO.setup(motor_1B, GPIO.OUT)
GPIO.setup(motor_2A, GPIO.OUT)
GPIO.setup(motor_2B, GPIO.OUT)

seq = [[1, 0, 0, 1], [1, 0, 0, 0], [1, 1, 0, 0],
        [0, 1, 0,0], [0, 1, 1, 0], [0, 0, 1, 0],
        [0, 0, 1, 1], [0, 0, 0, 1]]
delay = 0.005

for i in range(10):
    for h_step in range(8):
        for pin in range(4):
            GPIO.output(motor_1A, seq[h_step][0])
            GPIO.output(motor_1B, seq[h_step][1])
            GPIO.output(motor_2A, seq[h_step][2])
```

```
            GPIO.output(motor_2B, seq[h_step][3])
            time.sleep(delay)

for h_step in reversed(range(8)):
    for pin in range(4):
        GPIO.output(motor_1A, seq[h_step][0])
        GPIO.output(motor_1B, seq[h_step][1])
        GPIO.output(motor_2A, seq[h_step][2])
        GPIO.output(motor_2B, seq[h_step][3])
        time.sleep(delay)

GPIO.cleanup()
```

Practical No. 4

Aim: Write a Python code to run motor in Predetermined path

Code:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(false)

motor1 = 4
motor2 = 17
motor3 = 23
motor4 = 24

GPIO.setup(motor1, GPIO.OUT)
GPIO.setup(motor2, GPIO.OUT)
GPIO.setup(motor3, GPIO.OUT)
GPIO.setup(motor4, GPIO.OUT)

seq = [[1, 0, 0, 1], [1, 0, 0, 0], [1, 1, 0, 0],
       [0, 1, 0, 0], [0, 1, 1, 0], [0, 0, 1, 0],
       [0, 0, 1, 1], [0, 0, 0, 1]]

path = [[1, 0, 0, 0], [1, 1, 0, 0], [0, 1, 0, 0],
       [0, 1, 1, 0], [0, 0, 1, 0], [0, 0, 1, 1],
       [0, 0, 0, 1], [1, 0, 0, 1]]

delay = 0.05

for step in path:
```

```
    GPIO.output(motor1, step[0])
    GPIO.output(motor2, step[1])
    GPIO.output(motor3, step[2])
    GPIO.output(motor4, step[3])
    time.sleep(delay)

GPIO.cleanup()
```

Practical No. 5

Aim: Develop Python code for testing the sensors.

Code:

```
int pirsensor = 0;

void setup() {
    pinMode(2, INPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT)
}

void loop() {
    pirsensor = digitalRead(2);

    if (pirsensor == HIGH) {
        digitalWrite(13, HIGH);
        tone(12, 500, 500);
    }

    digitalWrite(13, LOW)

}
```

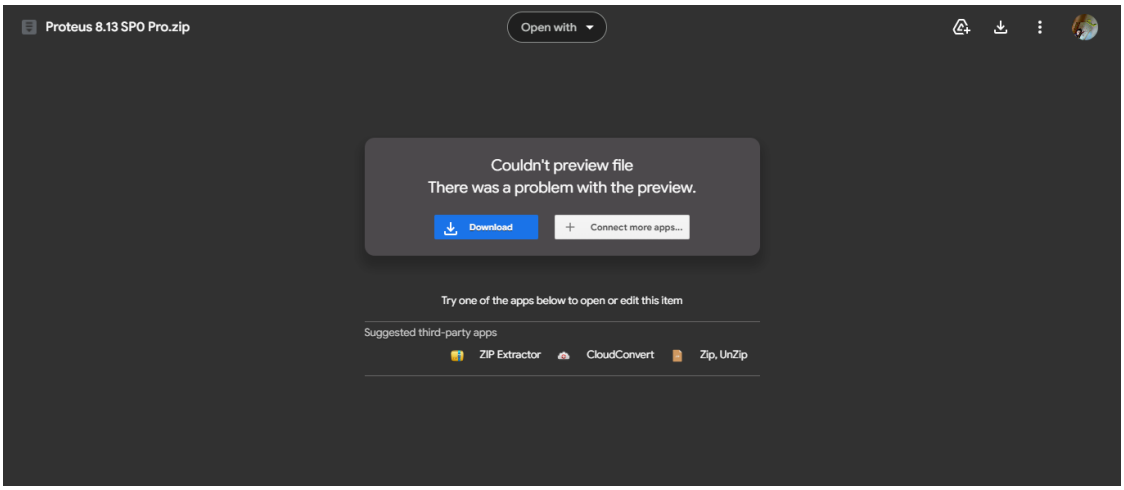
Practical No. 6

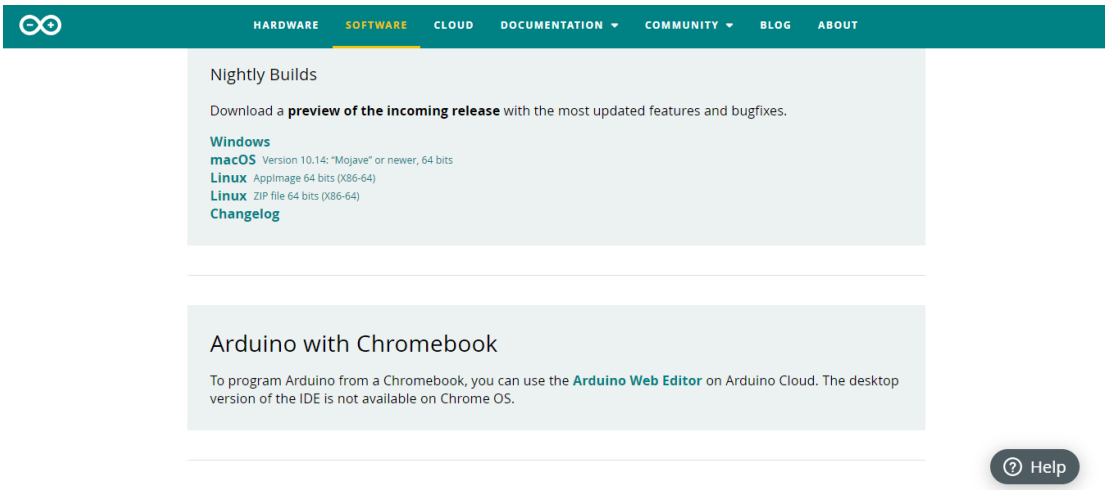
Aim: Add the sensors to the Robot object and develop the line-following behavior code.

Components Used:

1. Arduino uno
2. IR sensors (x2)
3. L298 Motor Controller
4. DC motors (x2)
5. Logic Toggle (x2)
6. Ground
7. Power, DC

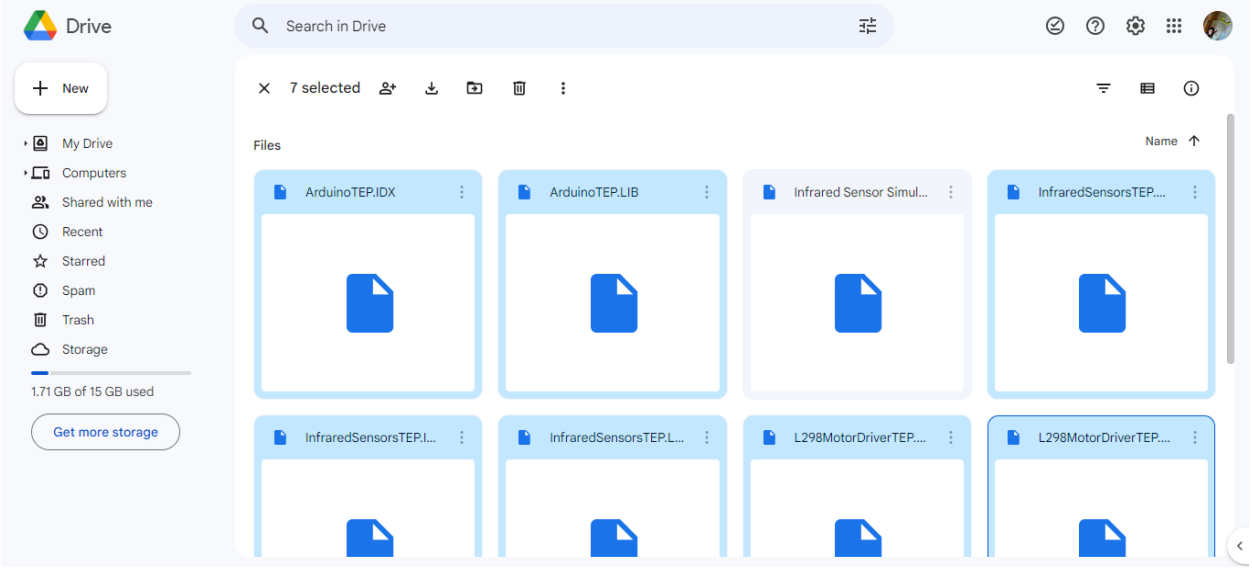
Step 1: Download and install Proteus and Arduino IDE.



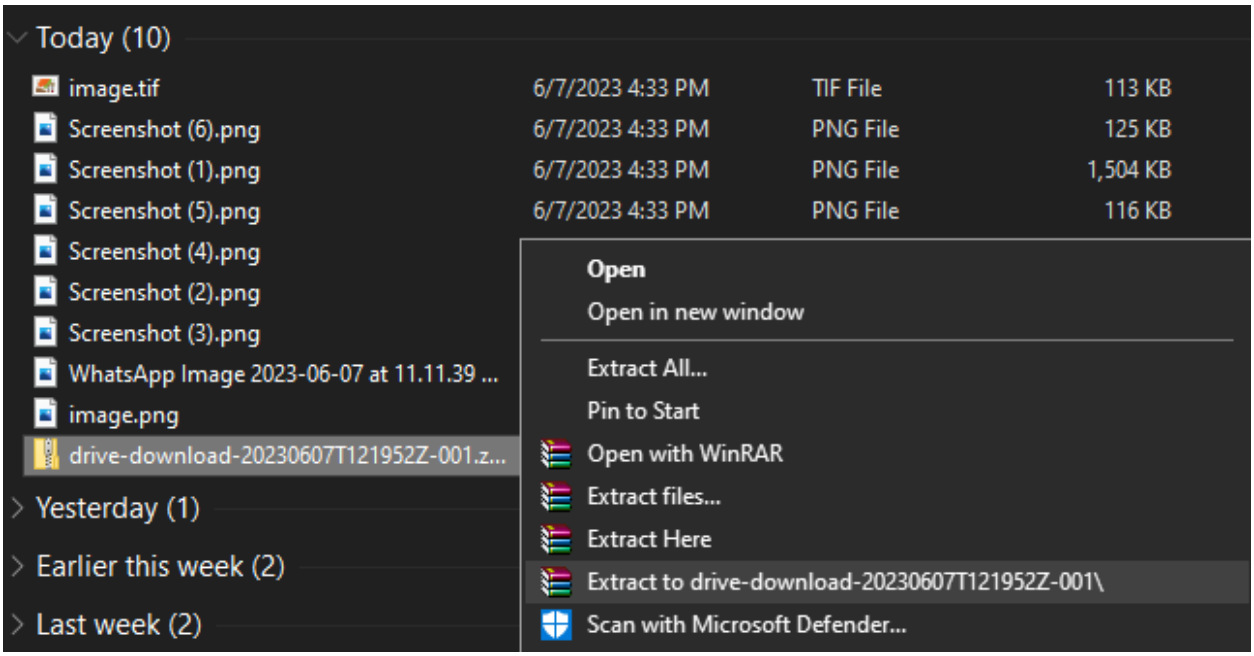


Step 2: Download Libraries: https://drive.google.com/drive/folders/18G4d2w5Sot_XCv7rKVMpC90EIM1FT8_f

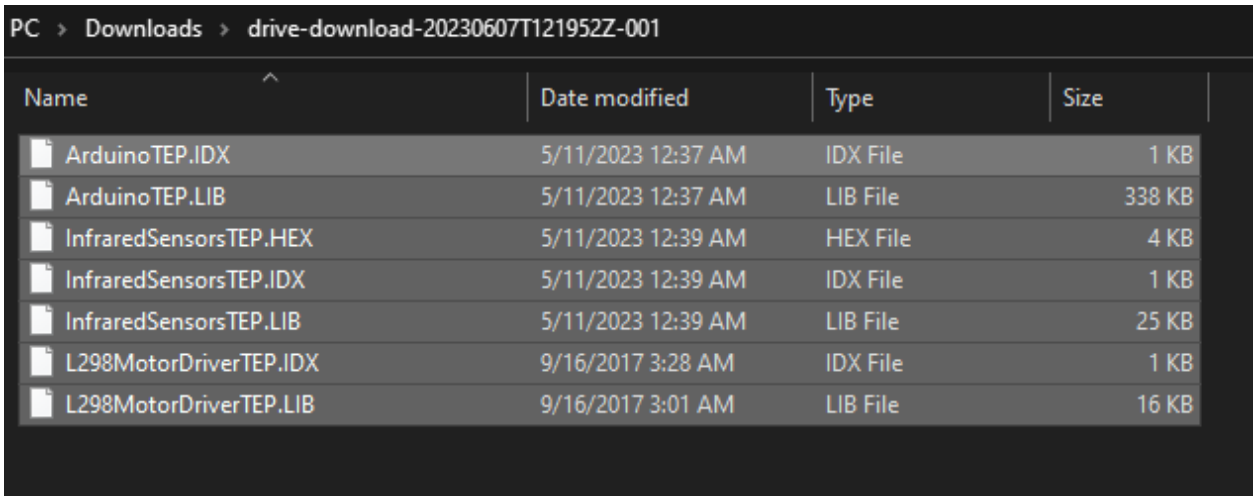
Download selected libraries:

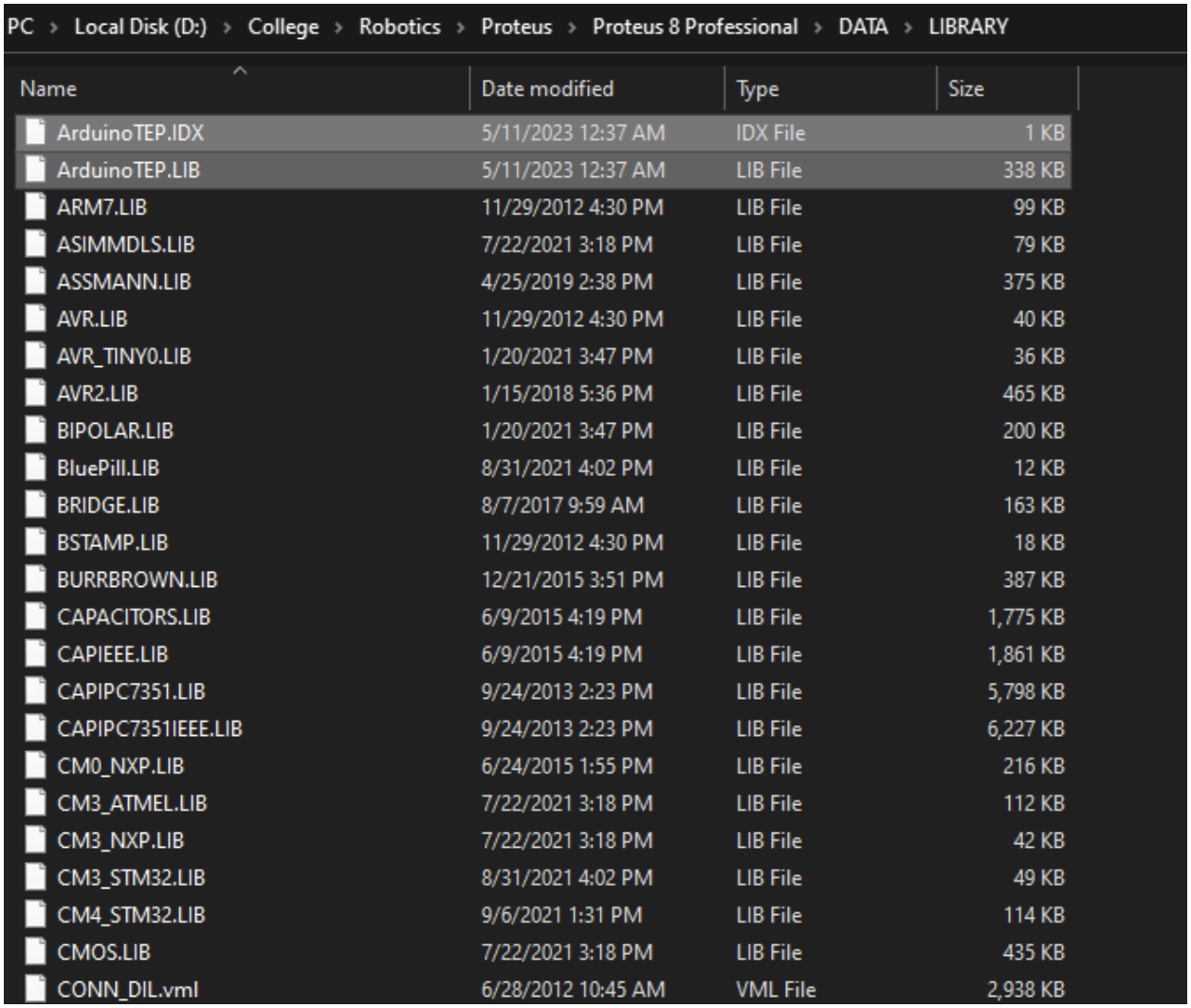


Step 3: Extract zip file.

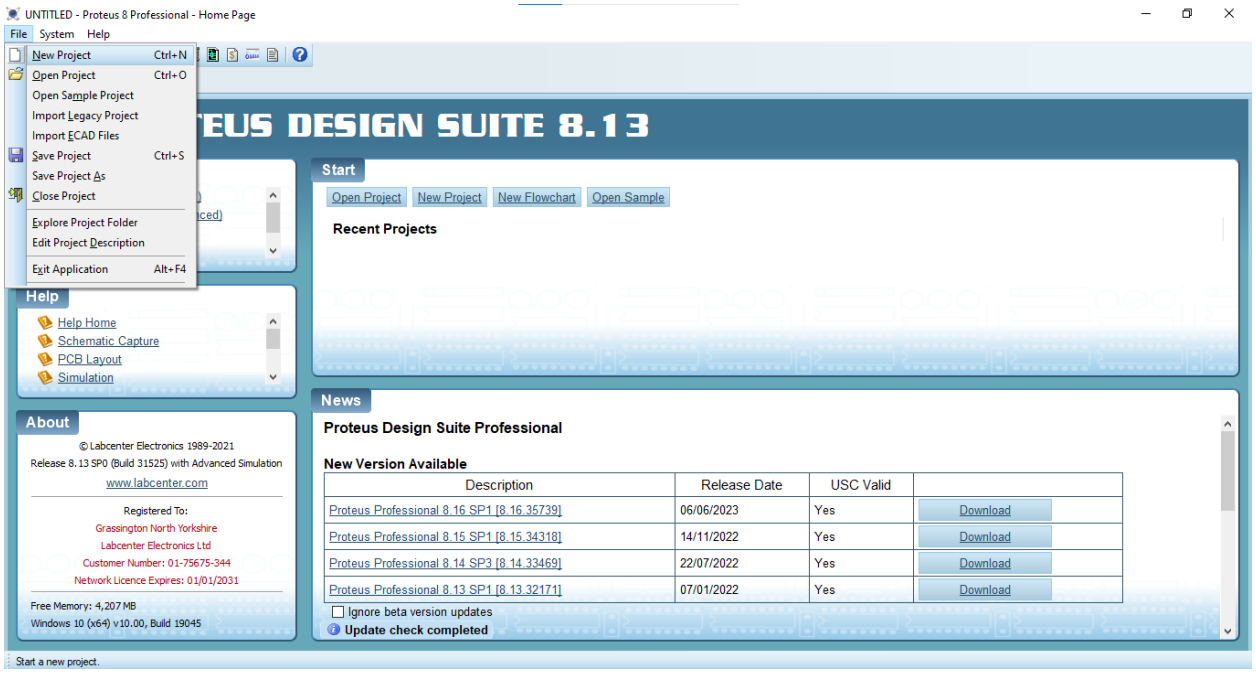


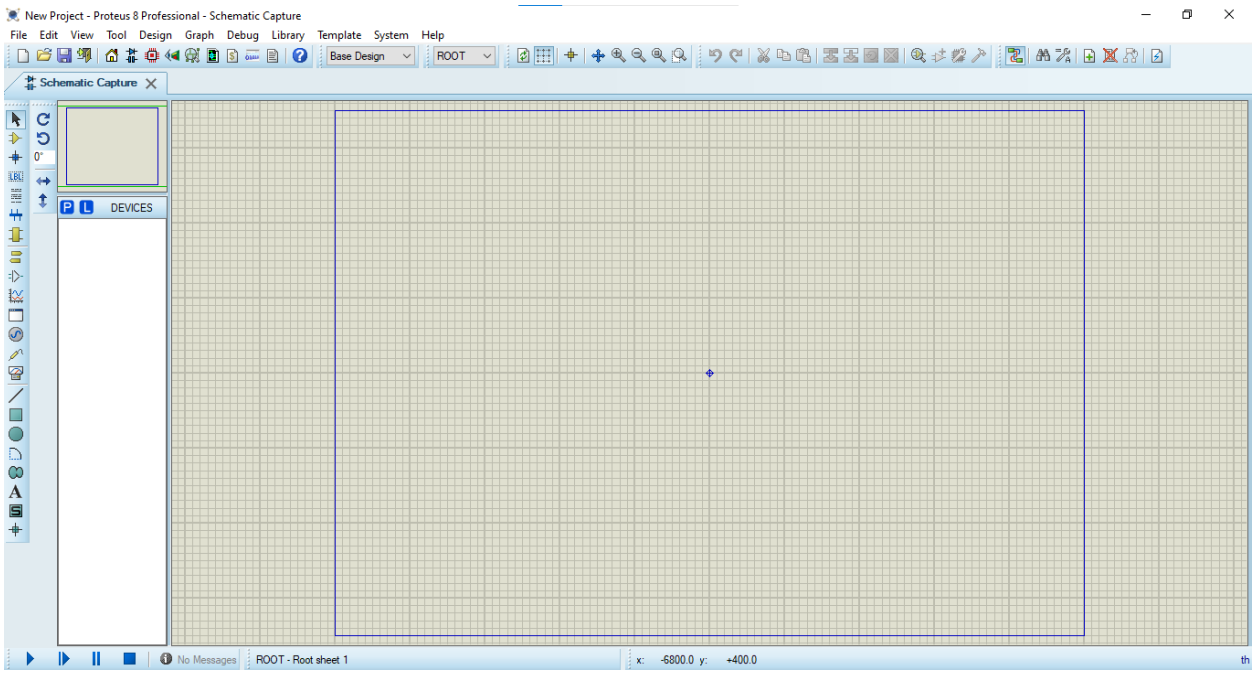
Step 4: Copy file and Paste in Proteus/DATA/LIBRARY folder



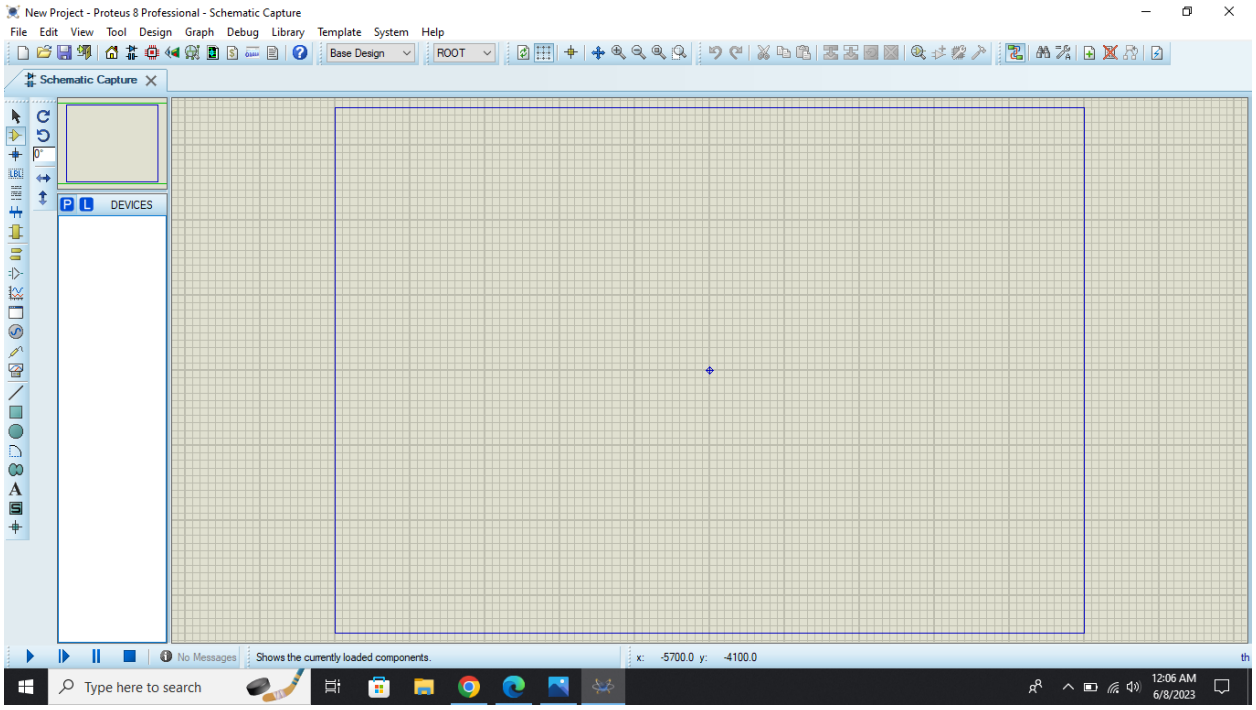


Step 5: Open Proteus > and Click on File > Create New Project.

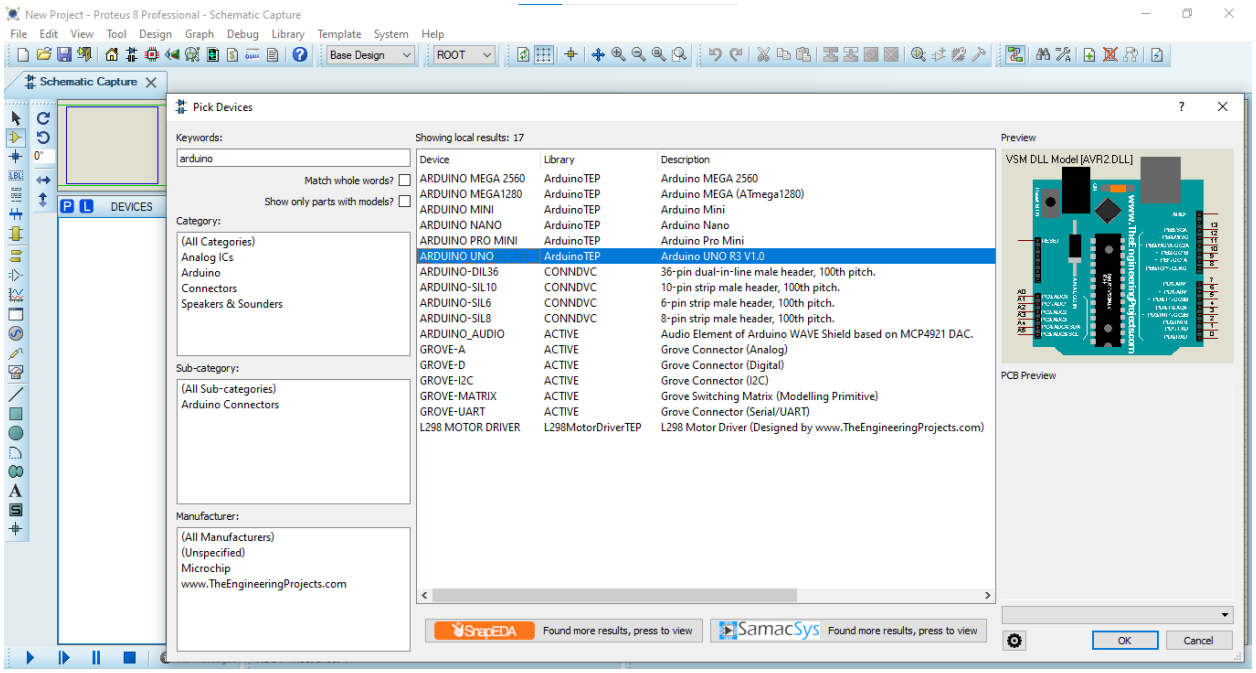




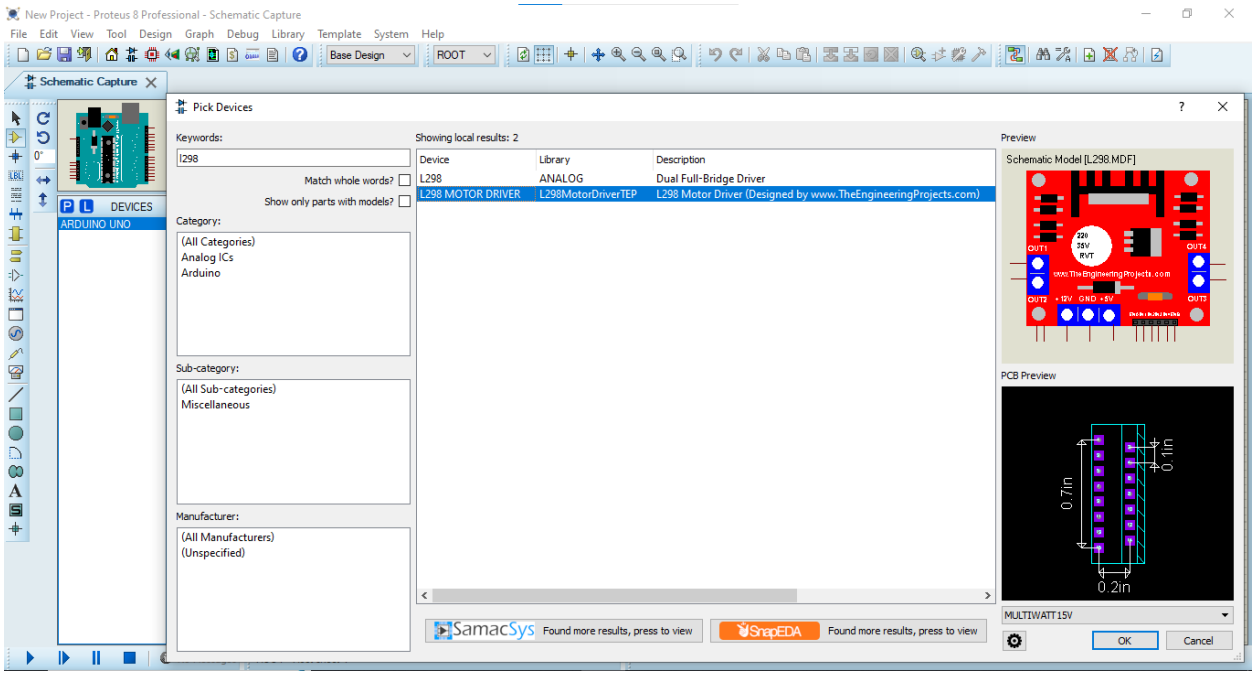
Step 6: Add components



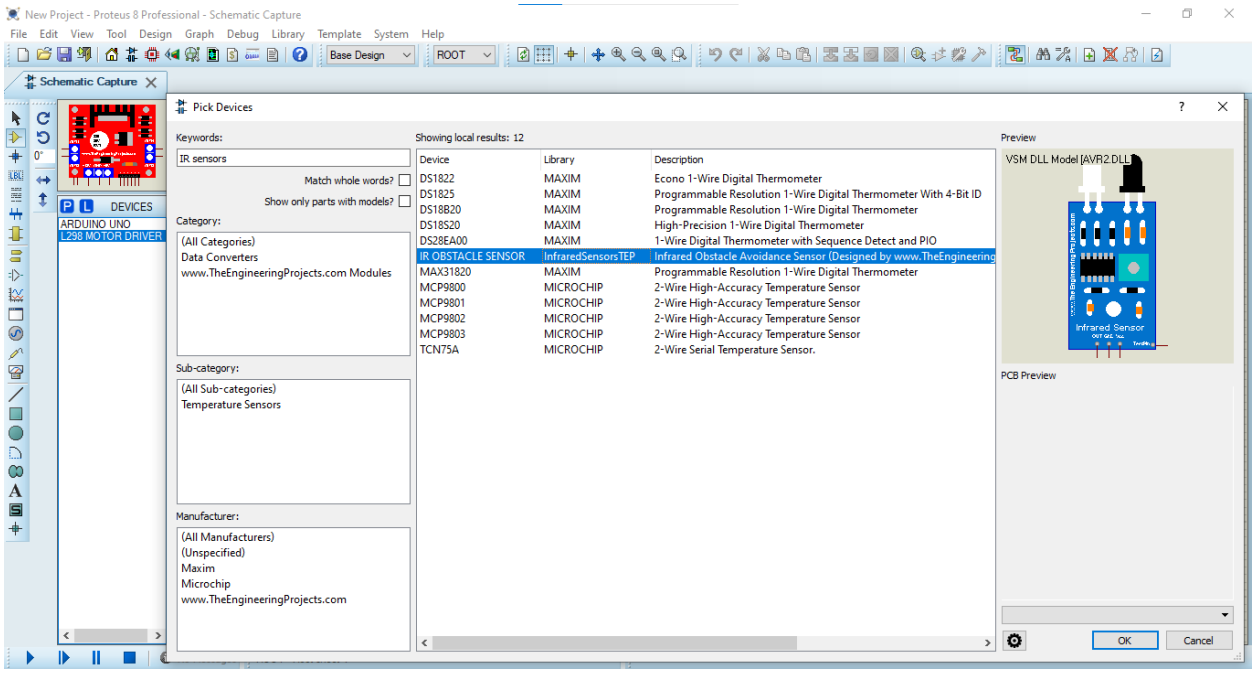
1. Add Arduino UNO



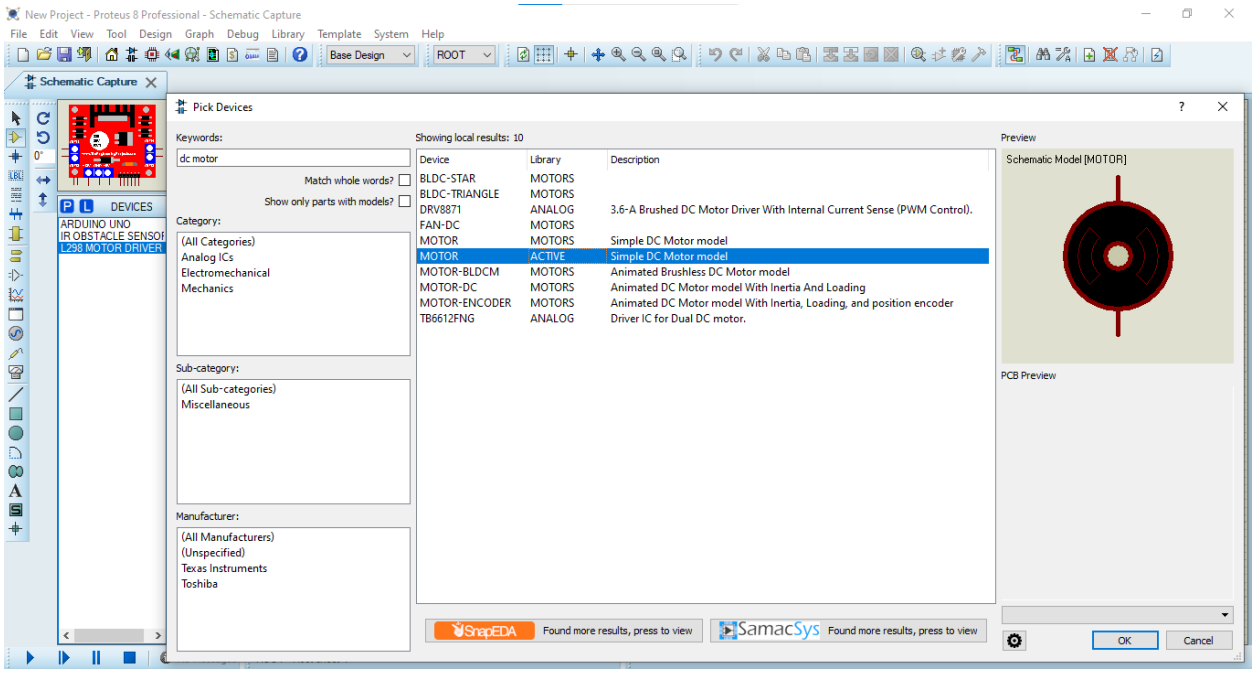
2. Add L298 Motor Driver



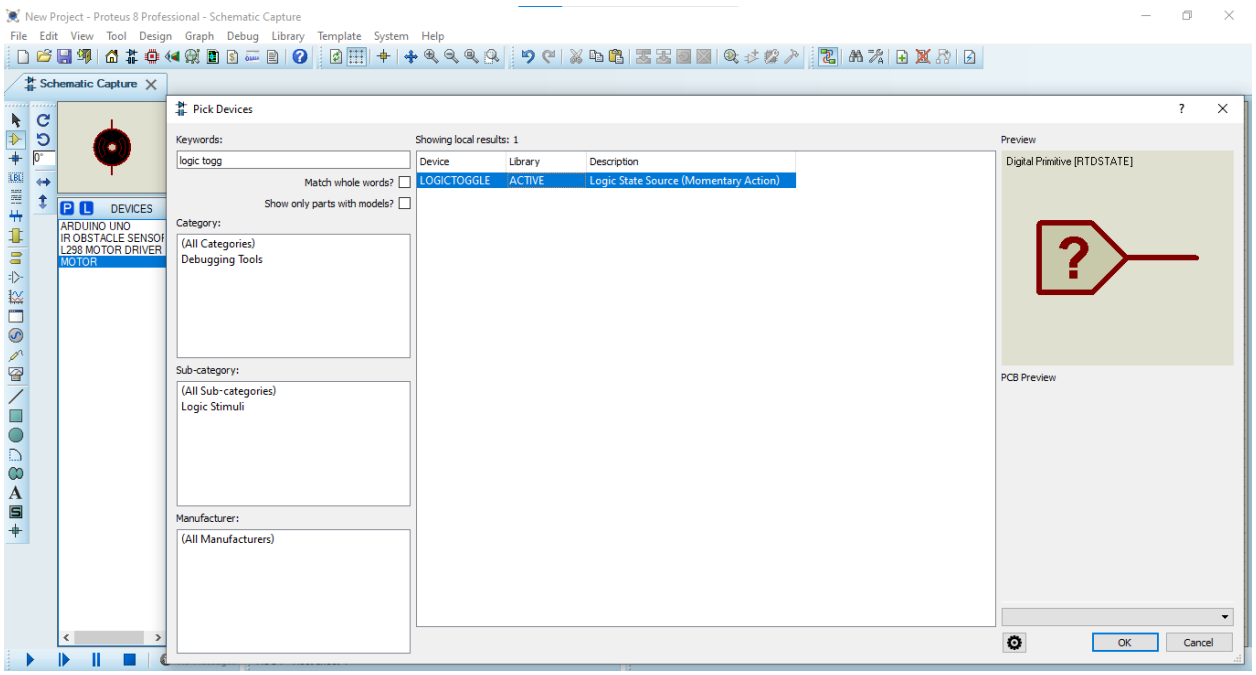
3. Add IR Obstacle Sensor



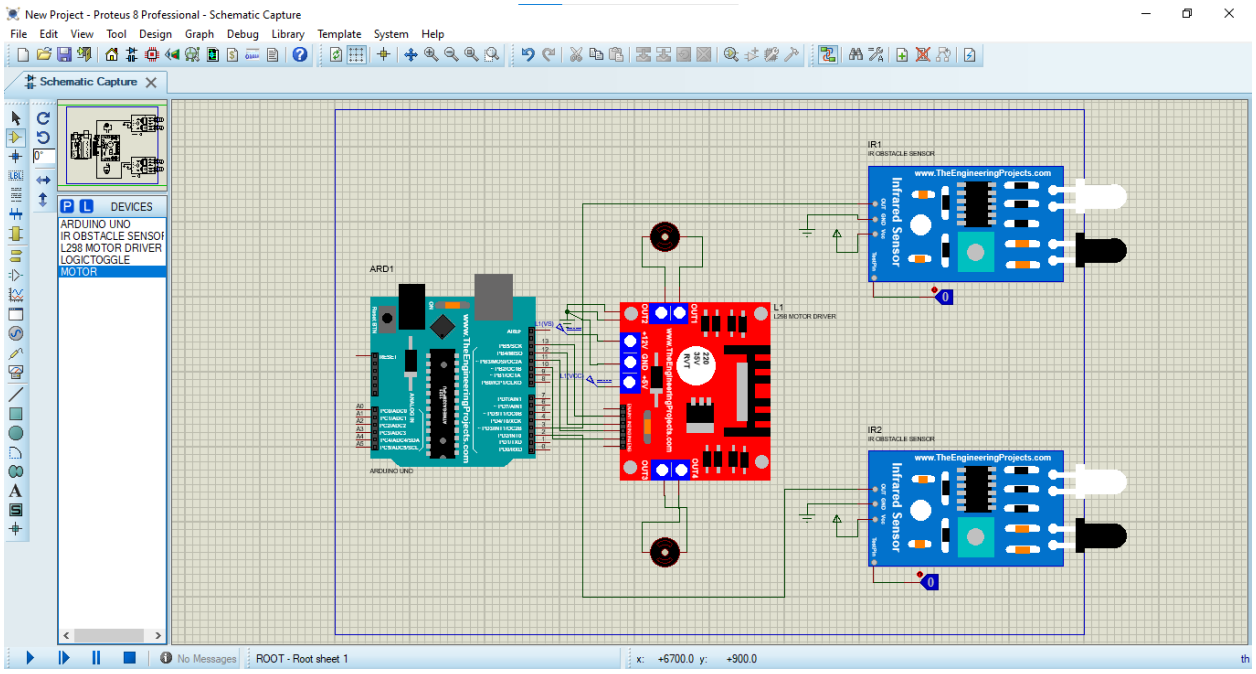
4. Add DC Motor



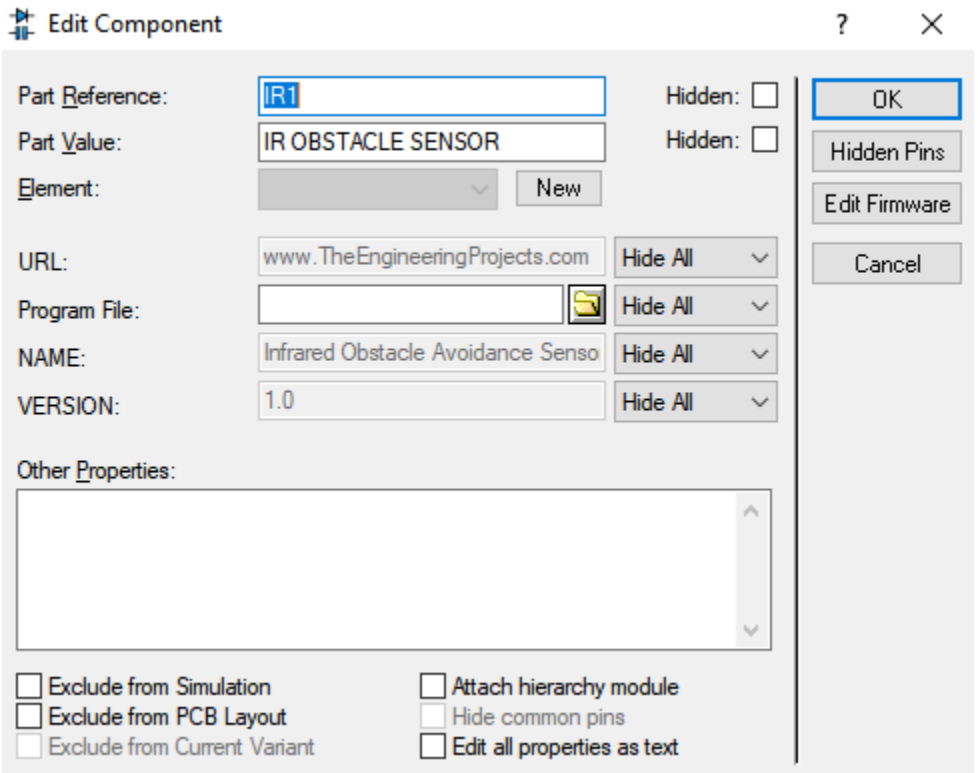
5. Add Logic Toggle

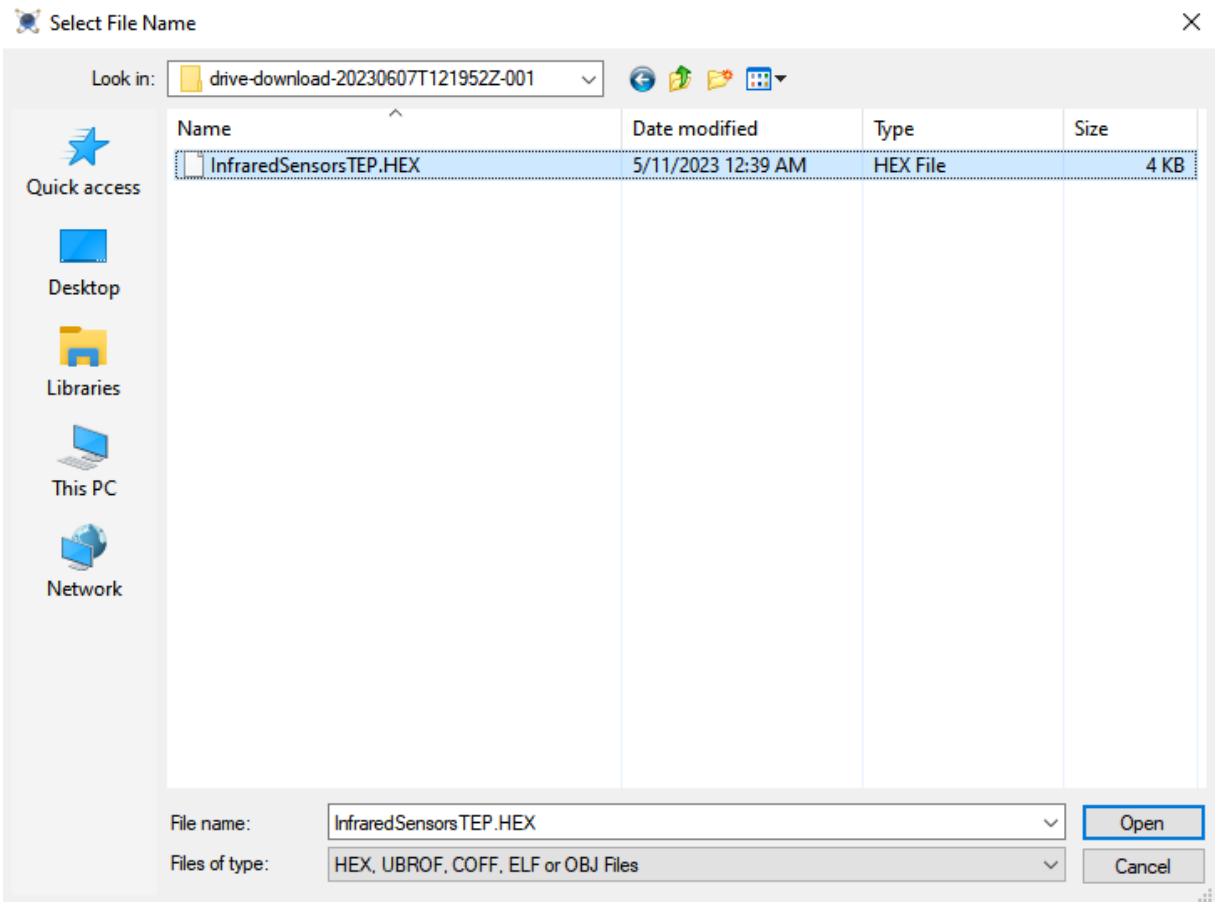


Step 7: Make Connection

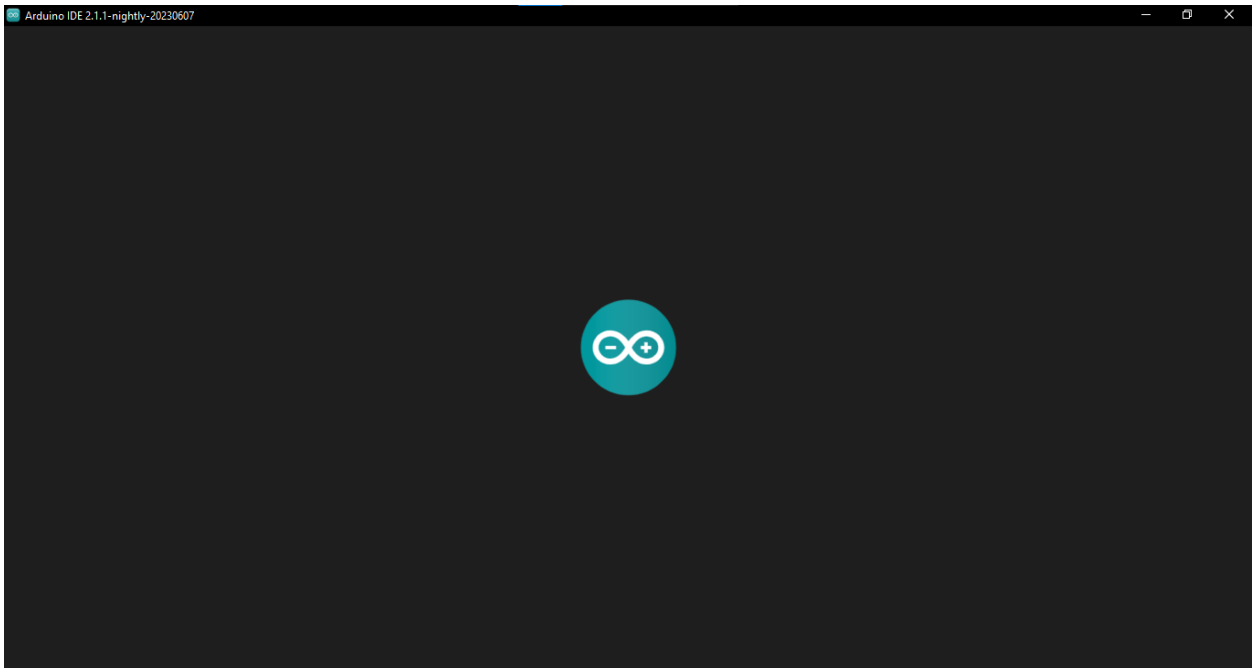


Step 8: Double click on IR Sensor and Add Library (for both)

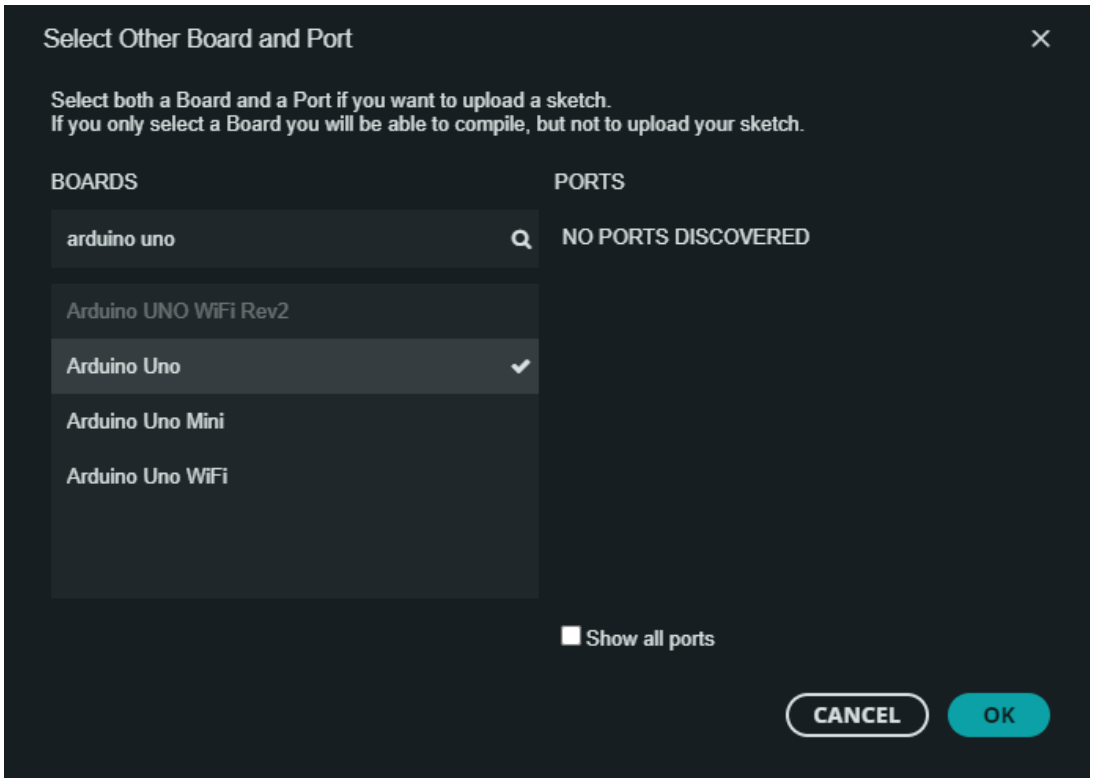




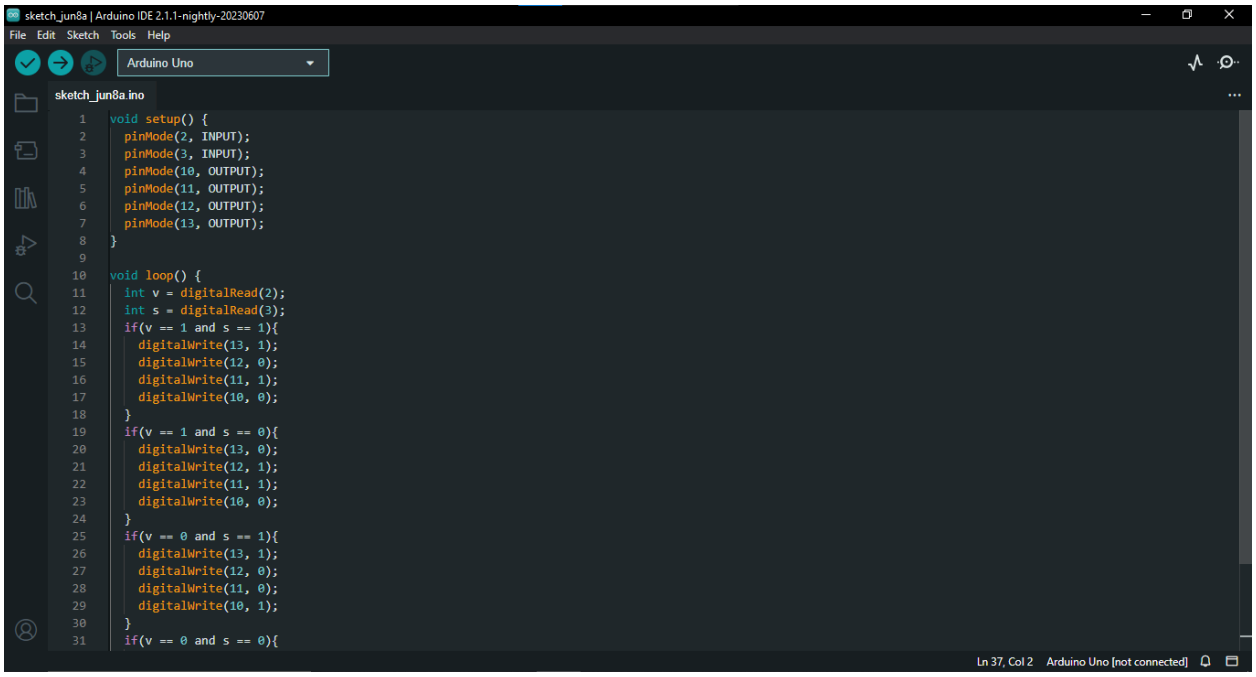
Step 9: Write a program for Arduino UNO using Arduino IDE.



Step 10: Select Arduino Uno board and Click on Ok



Step 11: Write code



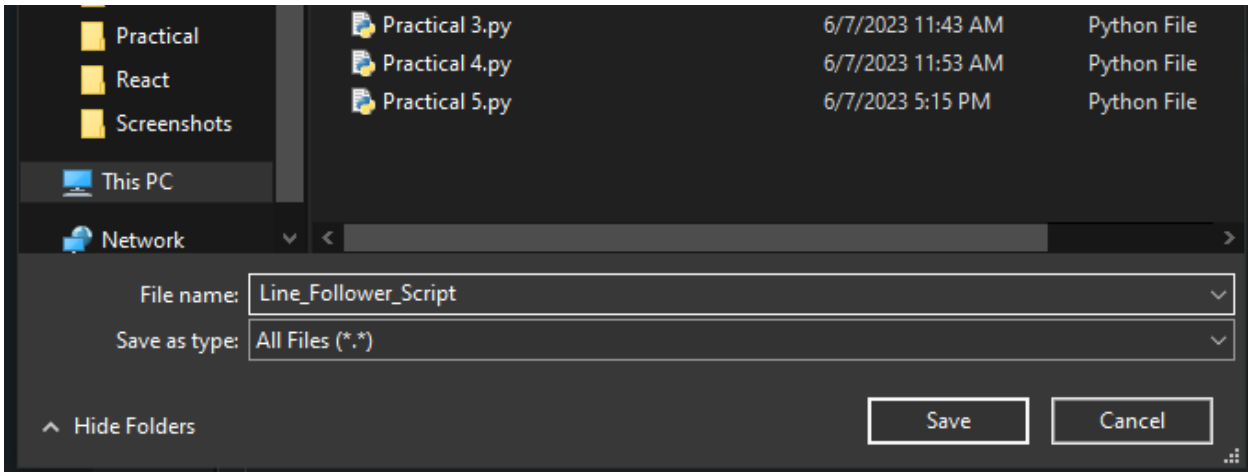
Code:

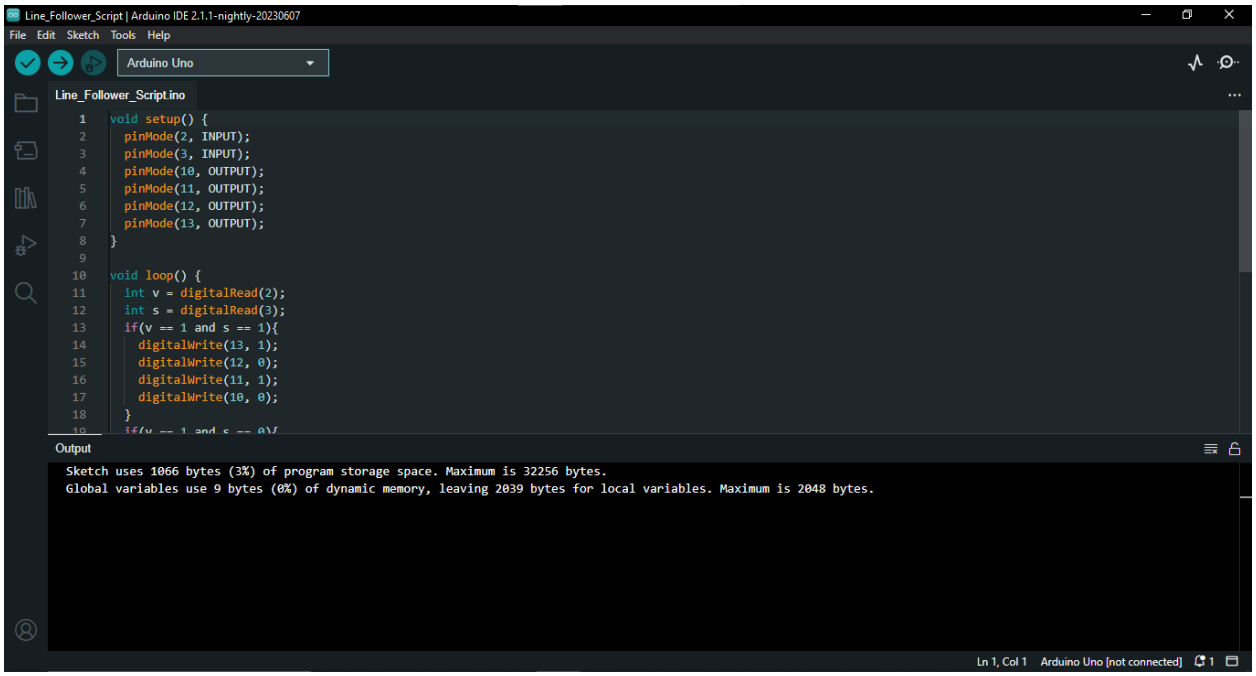
```
void setup() {
  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  int v = digitalRead(2);
  int s = digitalRead(3);
  if(v == 1 and s == 1){
    digitalWrite(13, 1);
    digitalWrite(12, 0);
    digitalWrite(11, 1);
    digitalWrite(10, 0);
  }
  if(v == 1 and s == 0){
    digitalWrite(13, 0);
    digitalWrite(12, 1);
    digitalWrite(11, 1);
    digitalWrite(10, 0);
  }
  if(v == 0 and s == 1){
    digitalWrite(13, 1);
    digitalWrite(12, 0);
    digitalWrite(11, 0);
    digitalWrite(10, 1);
  }
```

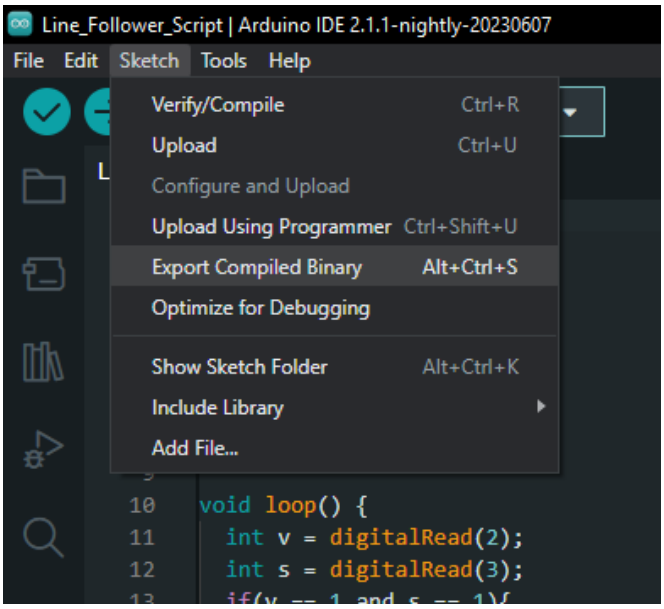
```
  if(v == 0 and s == 0){
    digitalWrite(13, 0);
    digitalWrite(12, 1);
    digitalWrite(11, 0);
    digitalWrite(10, 1);
  }
}
```

Step 12: Save the code and Click on verify

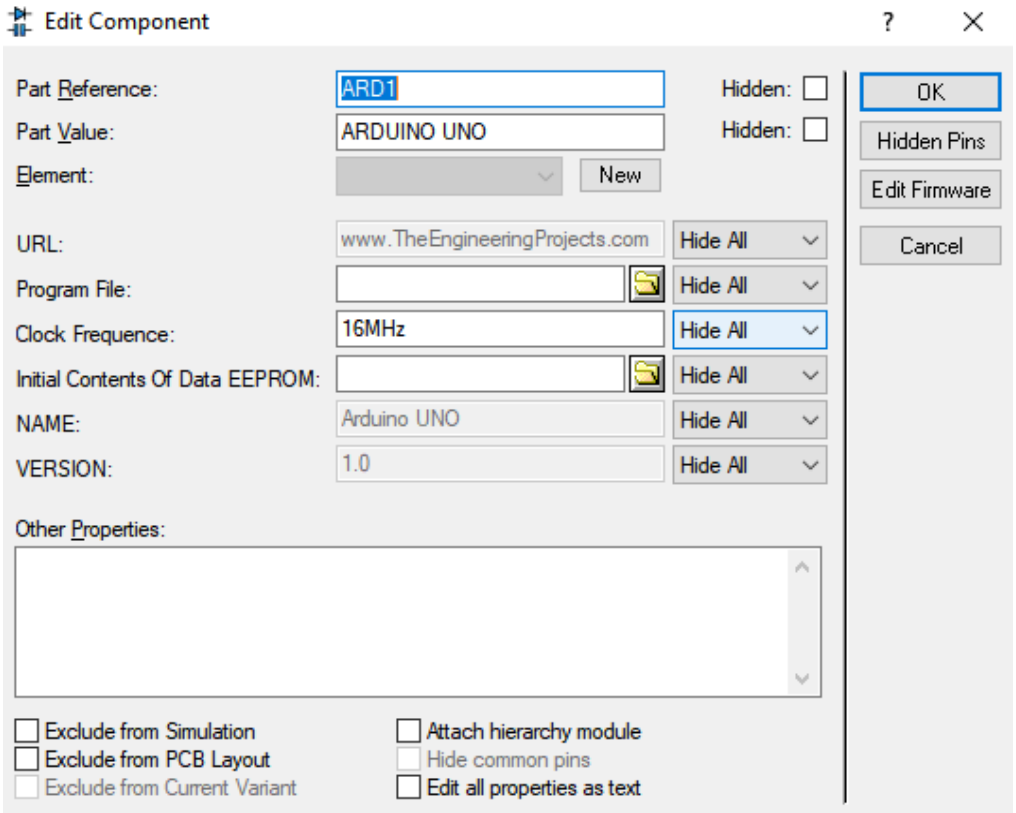


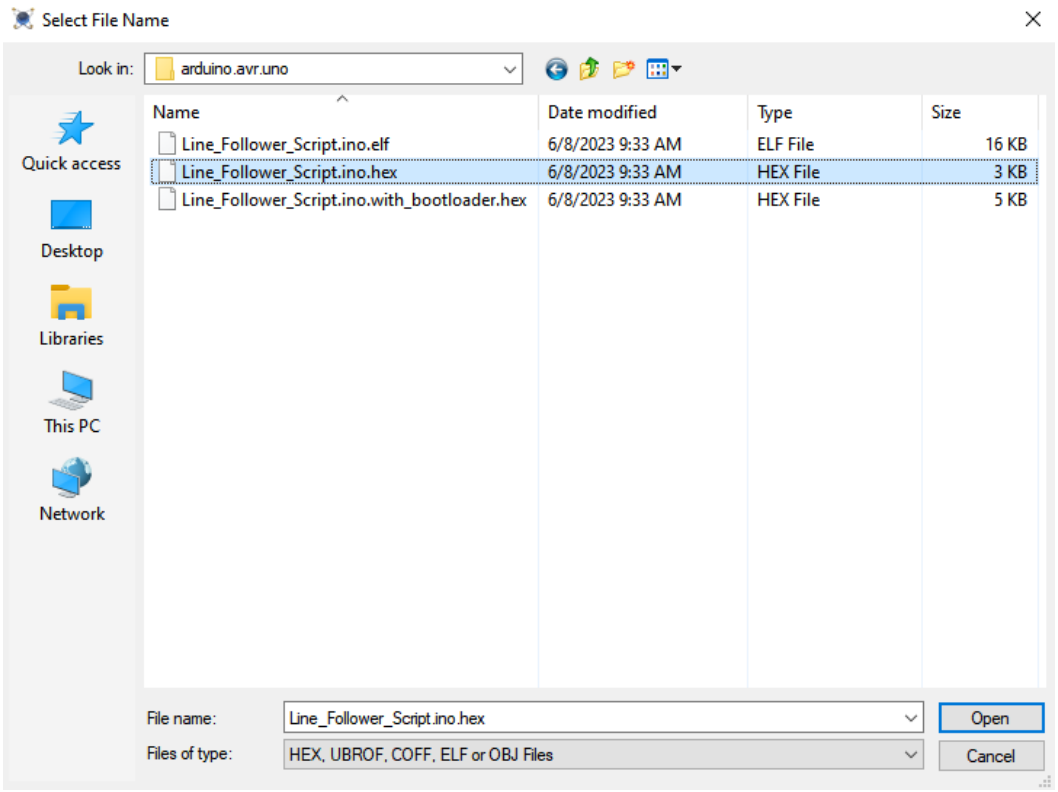


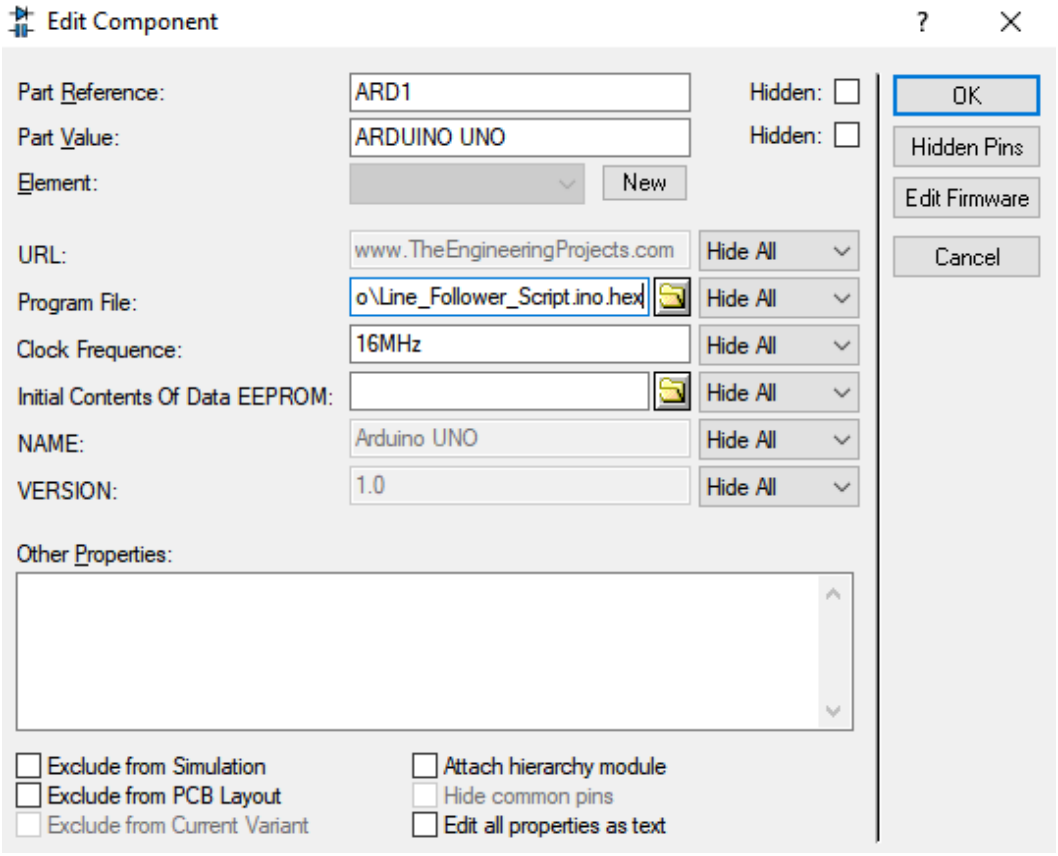
Step 13: Go to Sketch and select Export Compile Binary



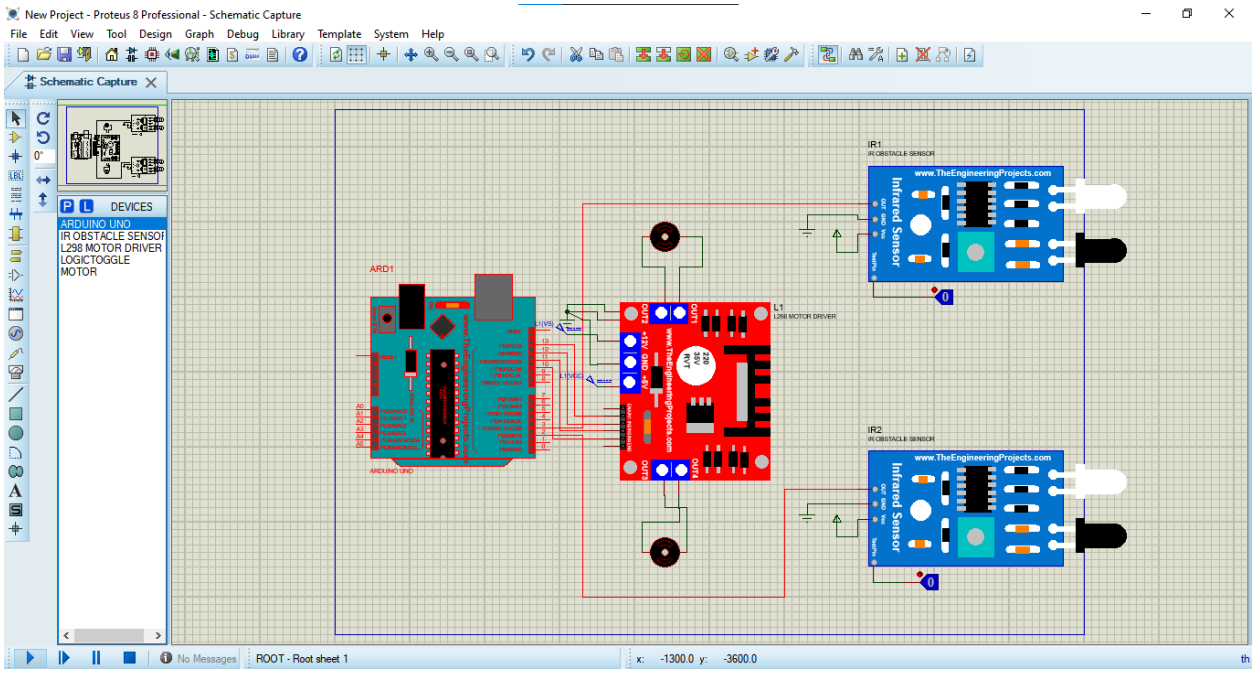
Step 14: Go to Proteus and Double click on Arduino UNO and browse Program file and select Exported Compile Binary File



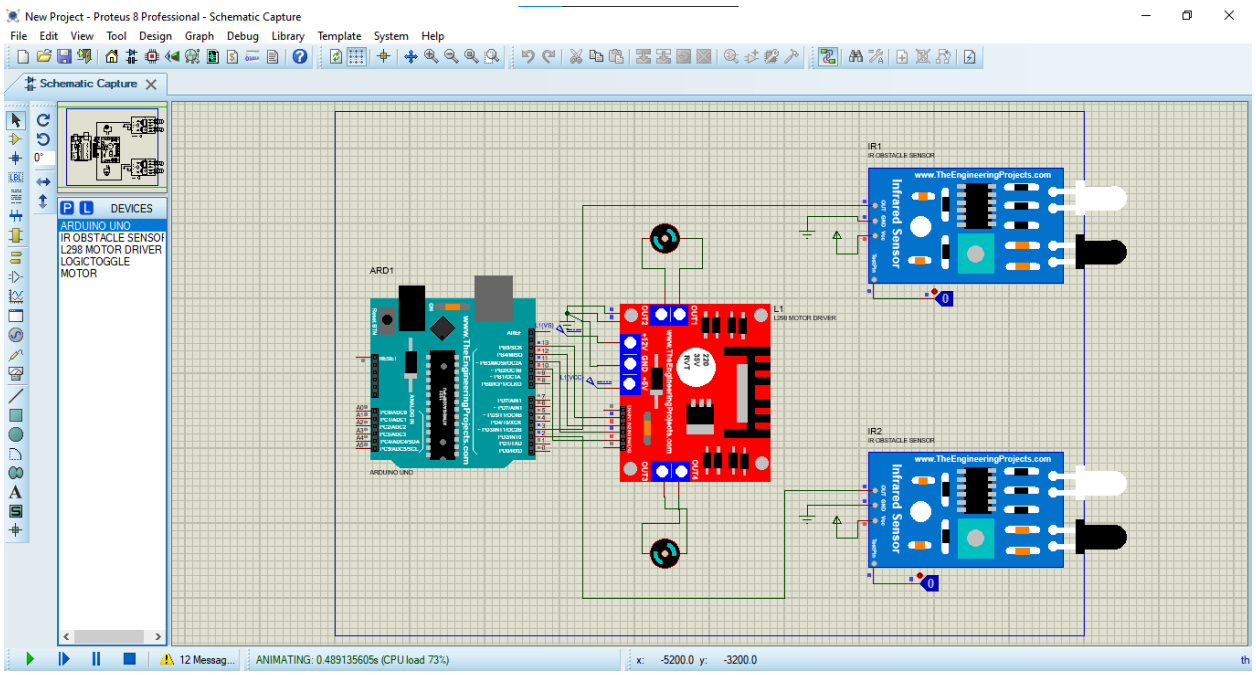

Select File Name
Look in:
arduino.avr.uno
Quick access
Desktop
Libraries
This PC
Network
Name
Date modified
Type
Size
Line_Follower_Script.ino.elf
6/8/2023 9:33 AM
ELF File
16 KB
Line_Follower_Script.ino.hex
6/8/2023 9:33 AM
HEX File
3 KB
Line_Follower_Script.ino.with_bootloader.hex
6/8/2023 9:33 AM
HEX File
5 KB
File name:
Line_Follower_Script.ino.hex
Open
Files of type:
HEX, UBROF, COFF, ELF or OBJ Files
Cancel


Edit Component
Part Reference:
ARD1
Hidden:
Part Value:
ARDUINO UNO
Hidden:
Element:
New
URL:
www.TheEngineeringProjects.com
Hide All
Program File:
o\Line_Follower_Script.ino.hex
Hide All
Clock Frequence:
16MHz
Hide All
Initial Contents Of Data EEPROM:
Hide All
NAME:
Arduino UNO
Hide All
VERSION:
1.0
Hide All
Other Properties:
Exclude from Simulation
Exclude from PCB Layout
Exclude from Current Variant
Attach hierarchy module
Hide common pins
Edit all properties as text
OK
Hidden Pins
Edit Firmware
Cancel

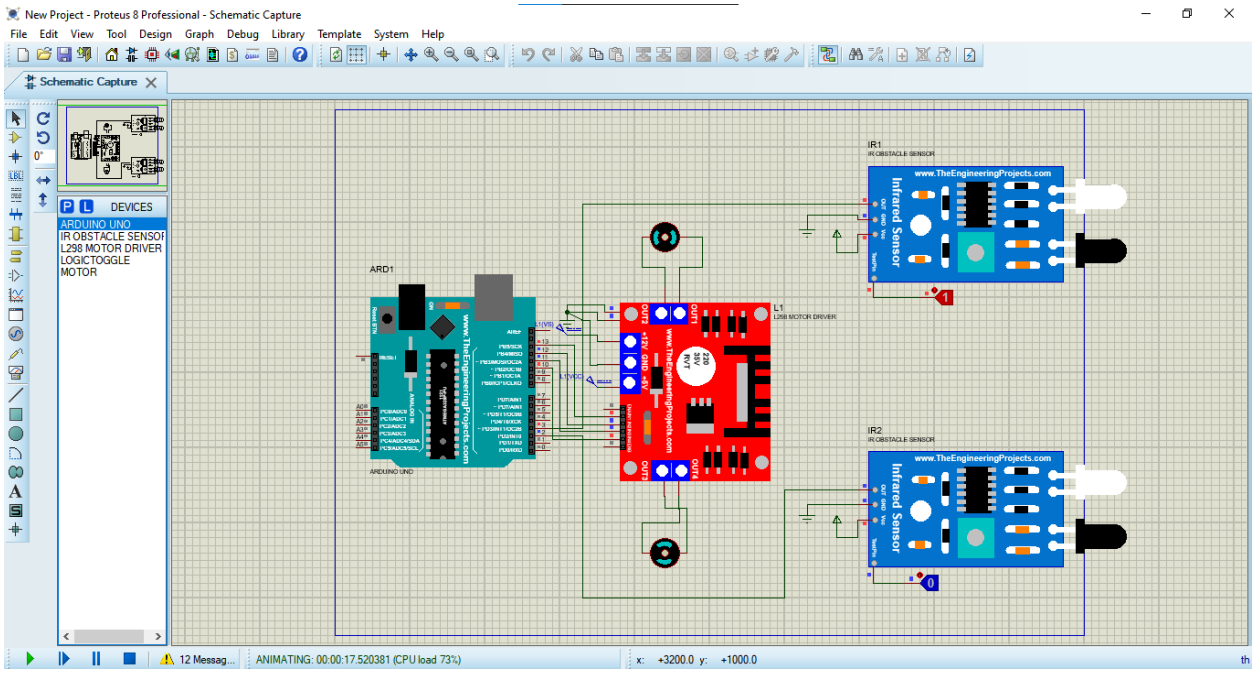
Step 15: Click on Run Simulate Button



Motor is start rotating



And it toggle to 1 Motor run in Clockwise direction



Practical No. 7

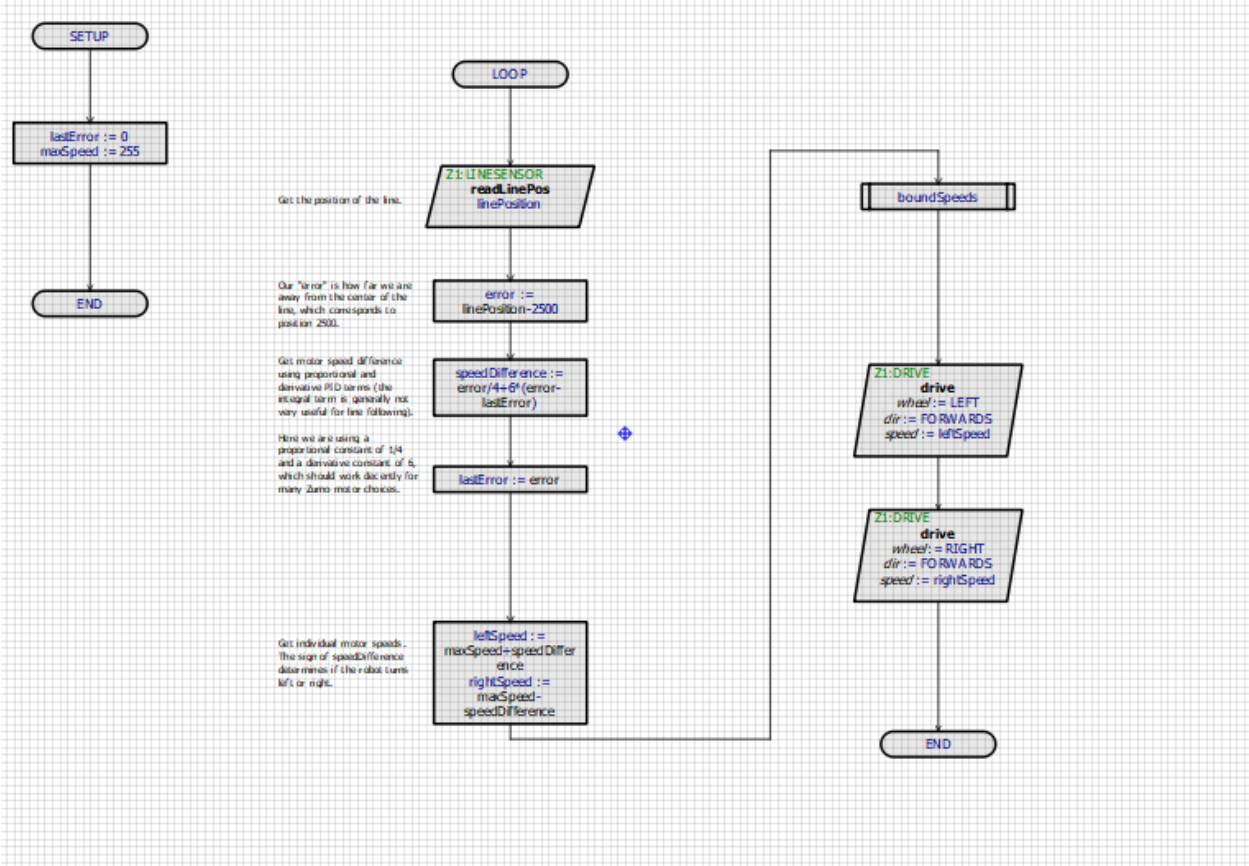
Aim: Add the sensors to the robot object and develop the line follower behavior code

Components Used:

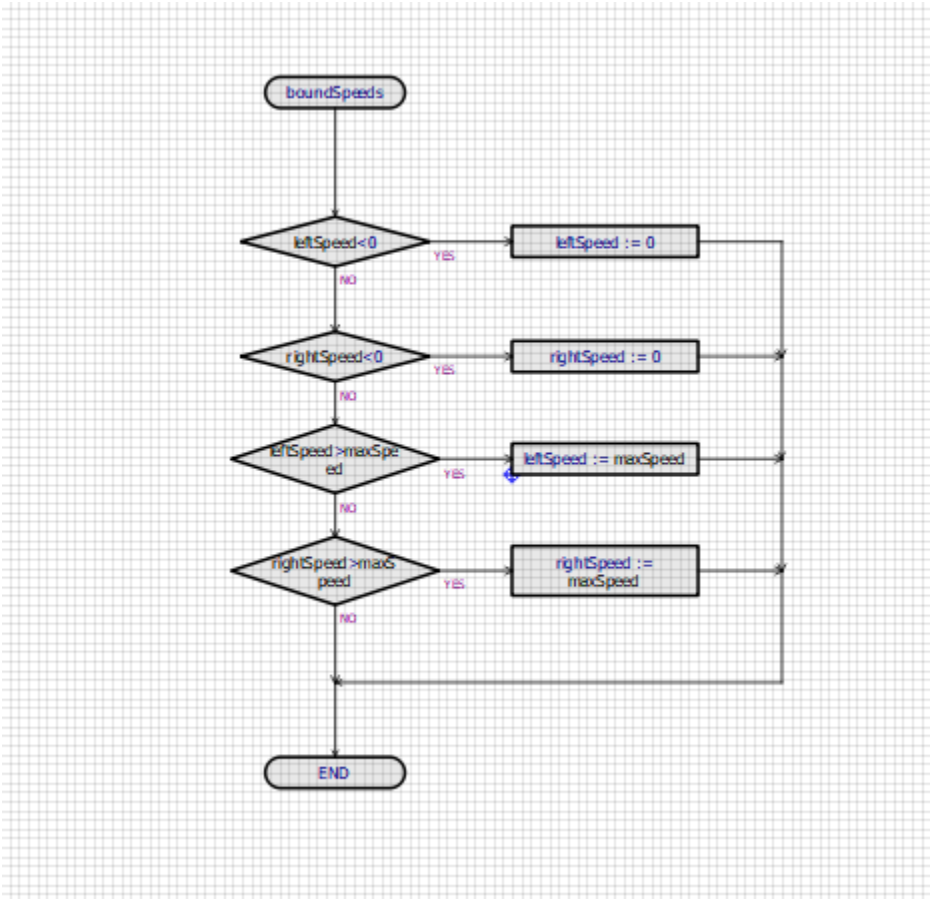
1. Arduino
2. Button
3. Zumo robot

Flow Chart:

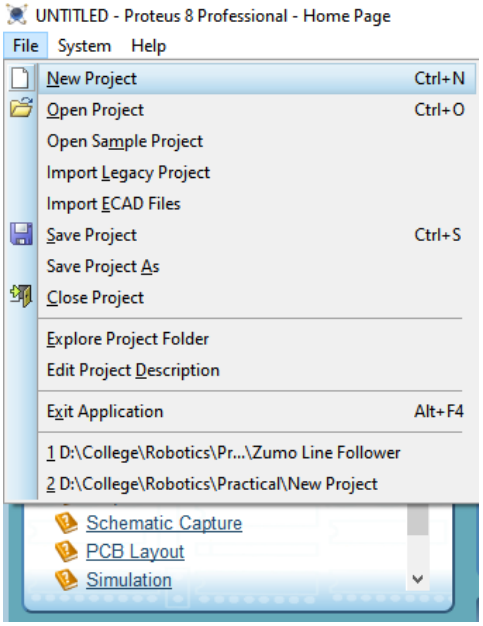
Main

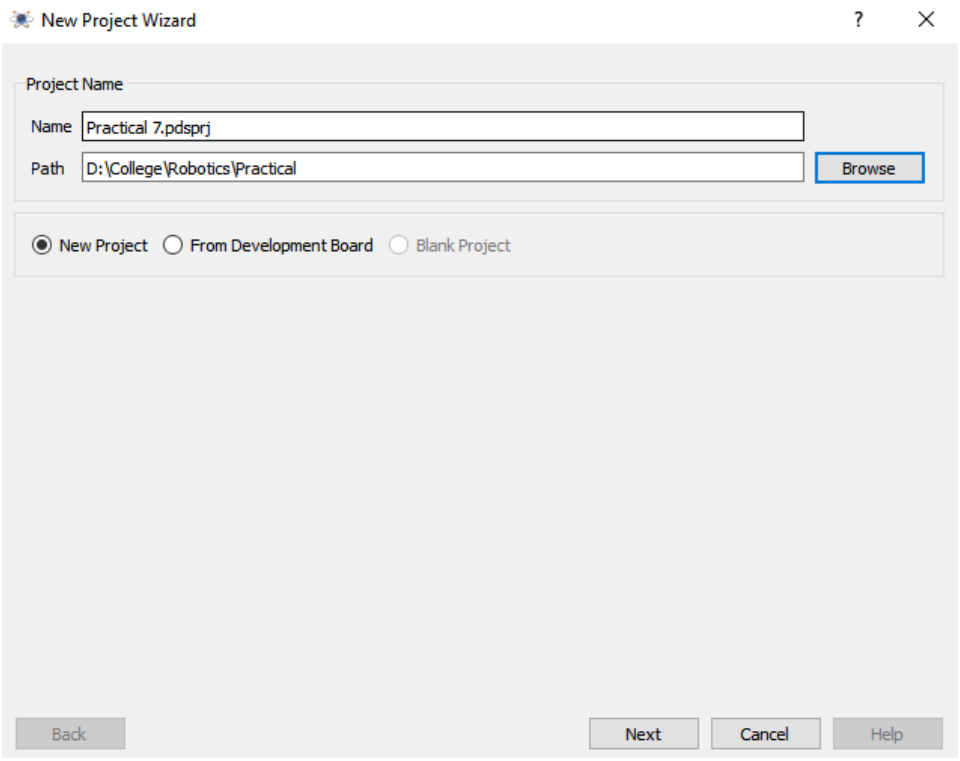


Subroutine

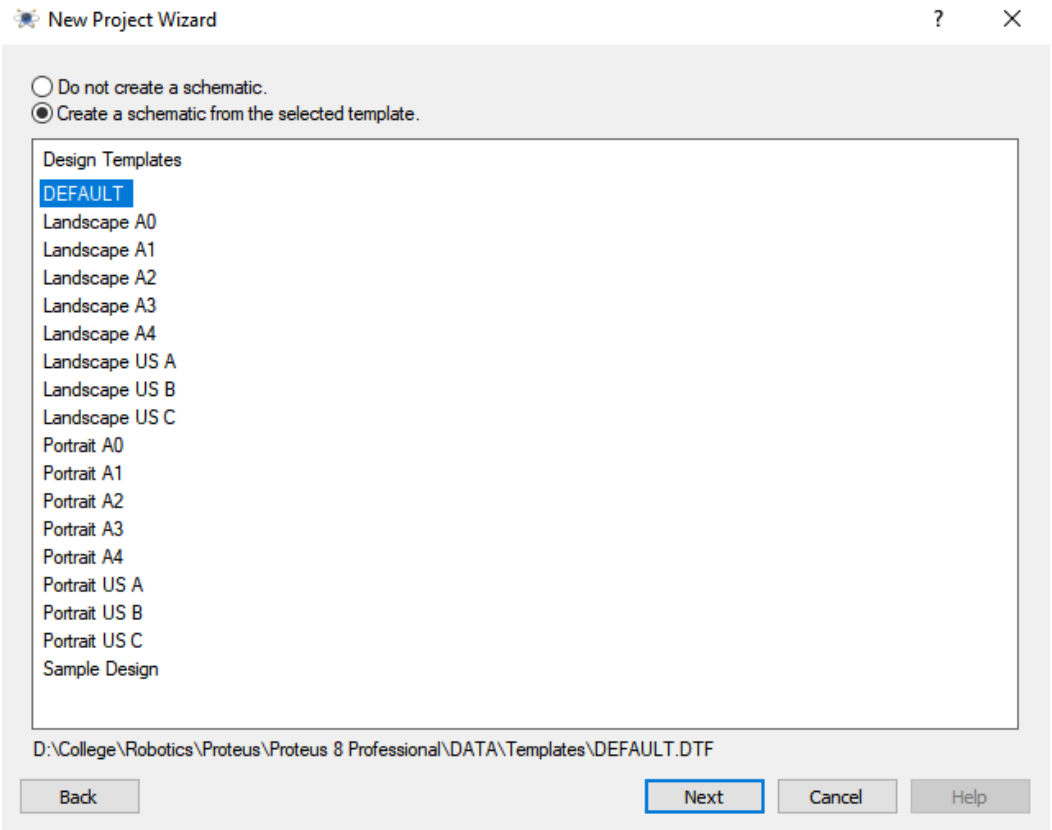


Step 1: Open Proteus > Click on File > Click on New Project

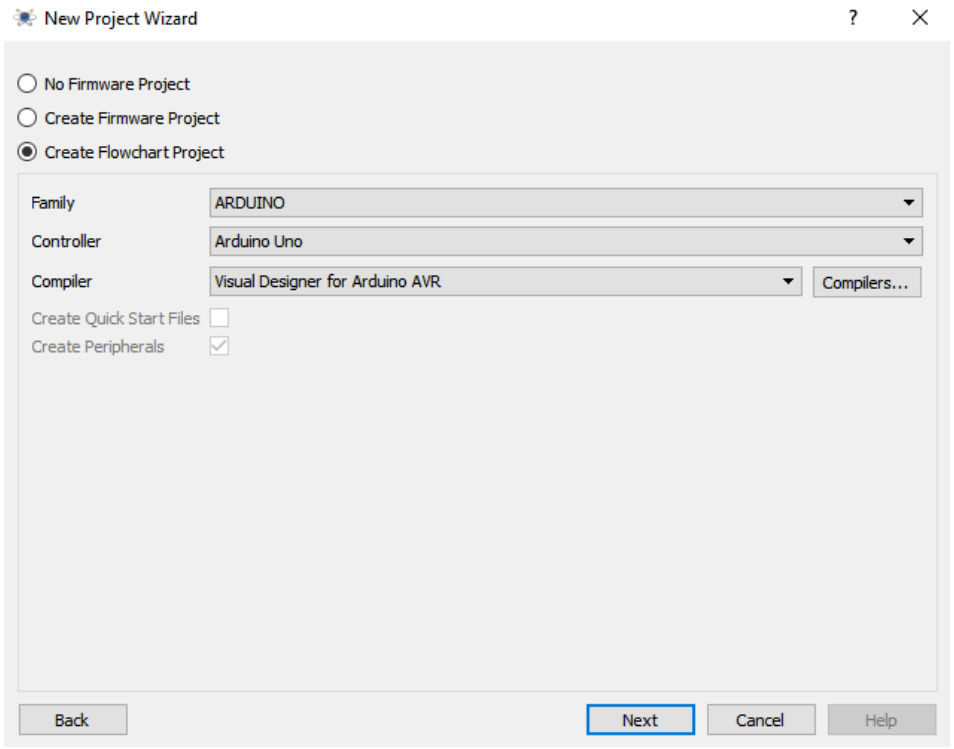




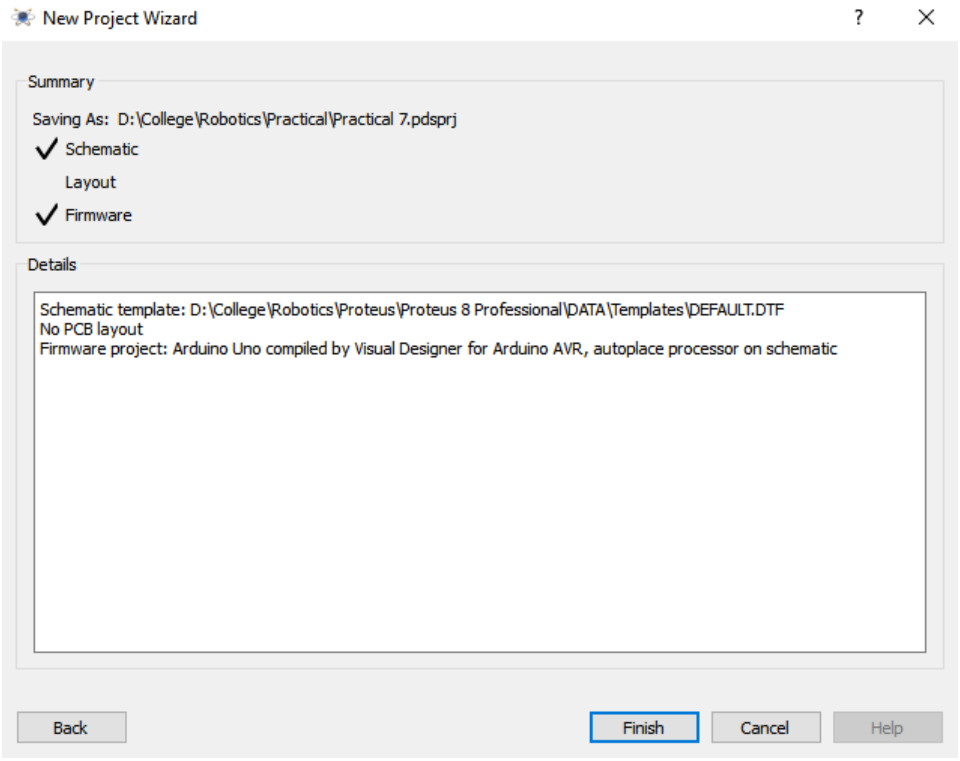
Set it to Default and Click on Next



Now Select Create Flowchart Project option

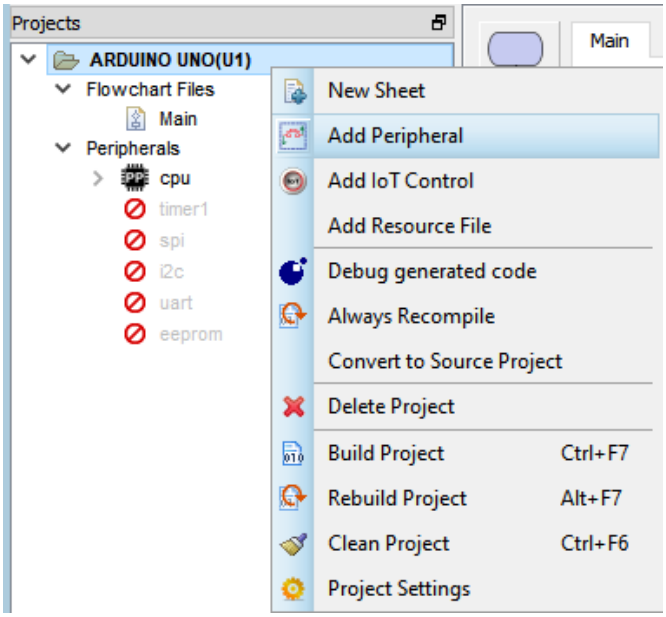


Then Click on Finish

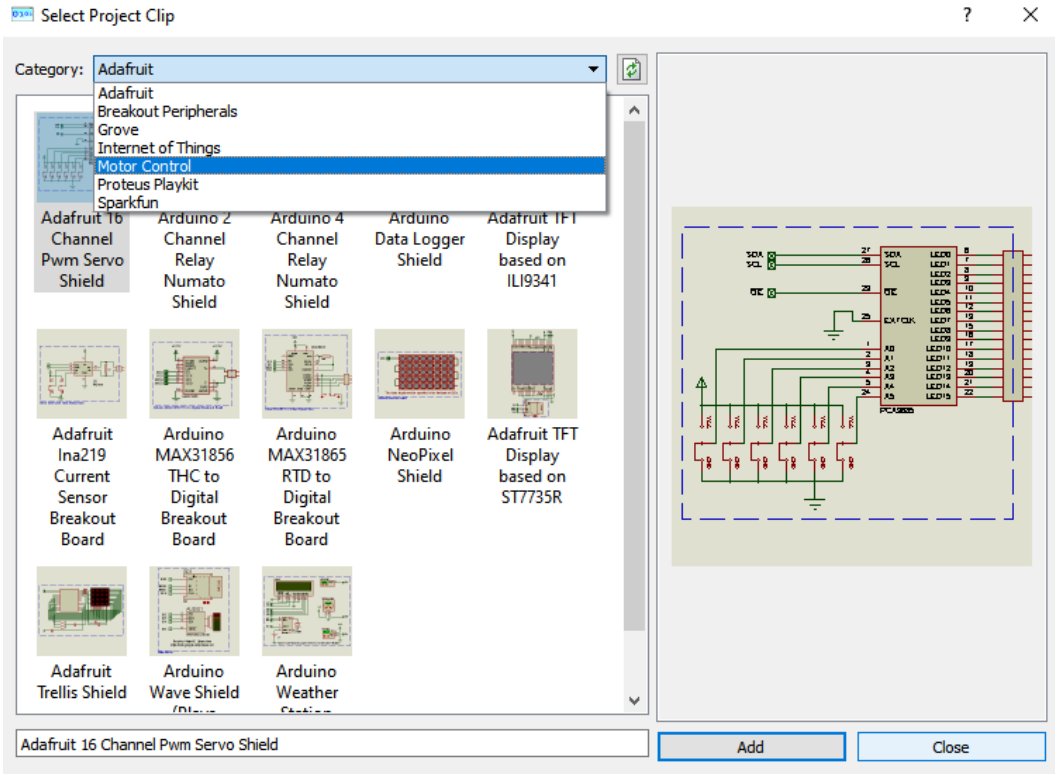


Step 2: Add Arduino Zumo Robot

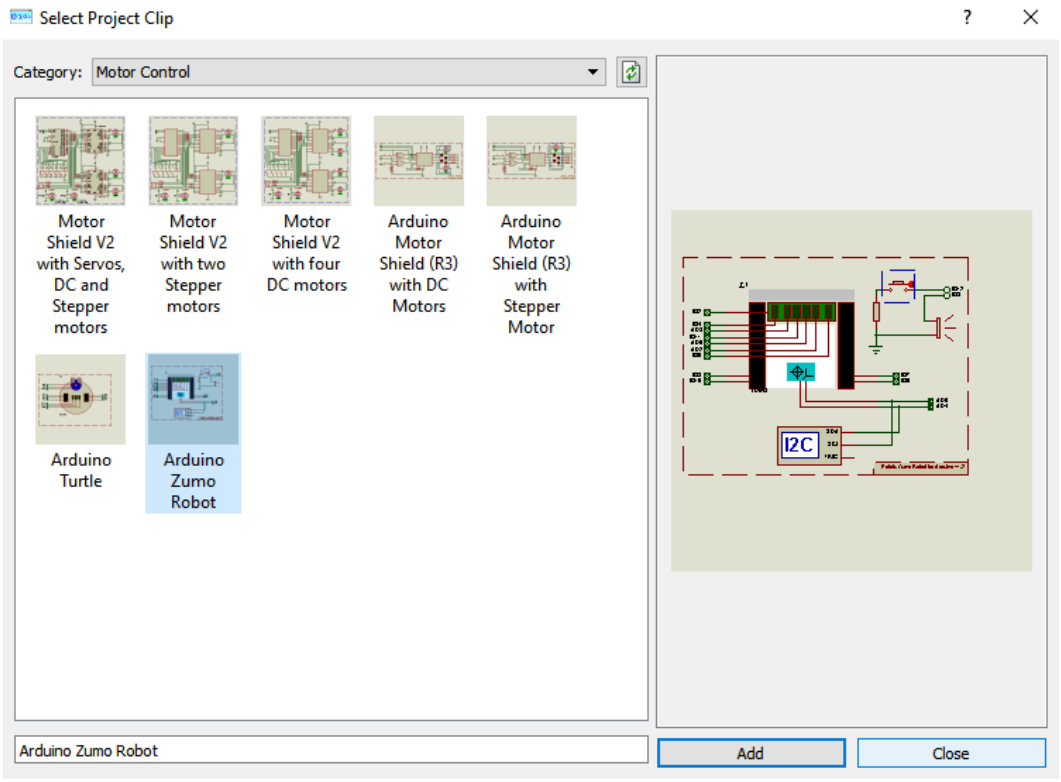
Right click on Project Name -> Click on Add Peripherals

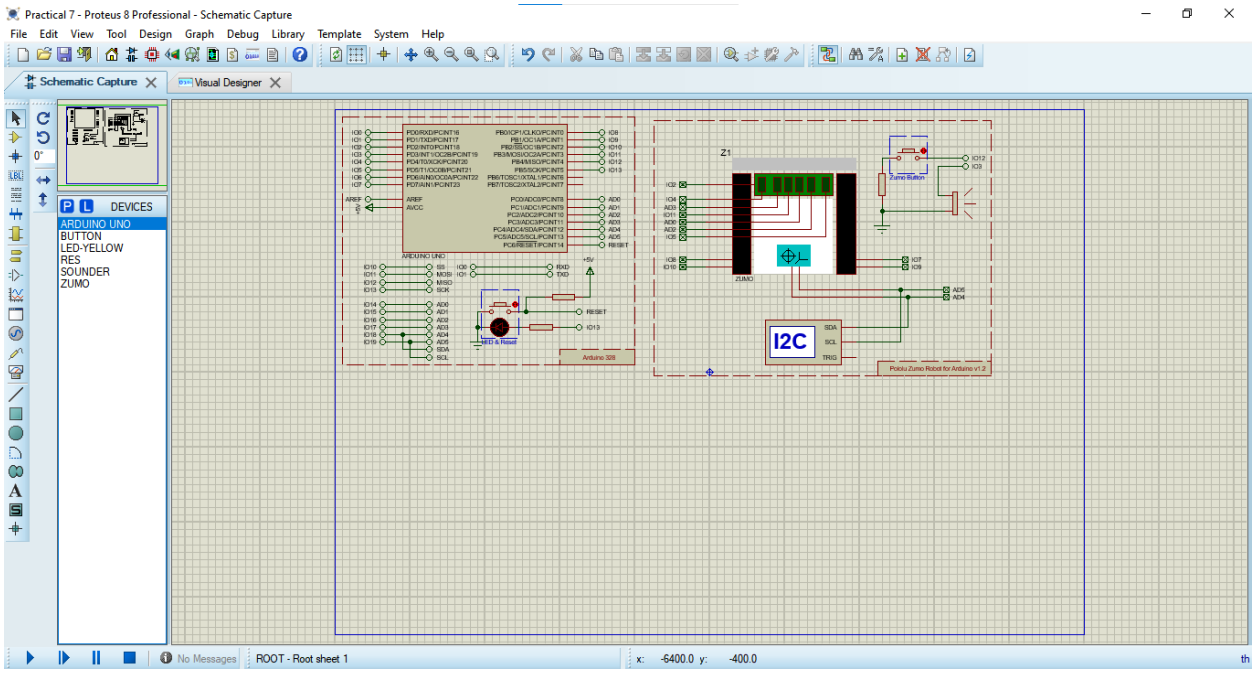


From Dropdown -> Select Motor Control

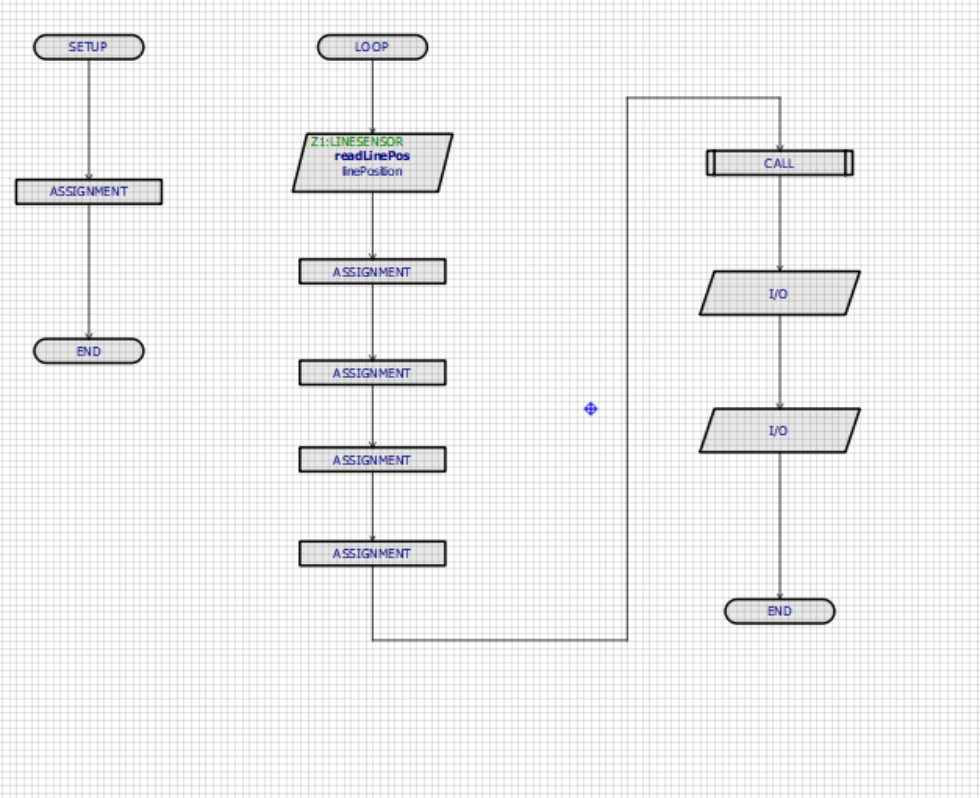


And then Select Arduino Zumo Robot and Click on Add

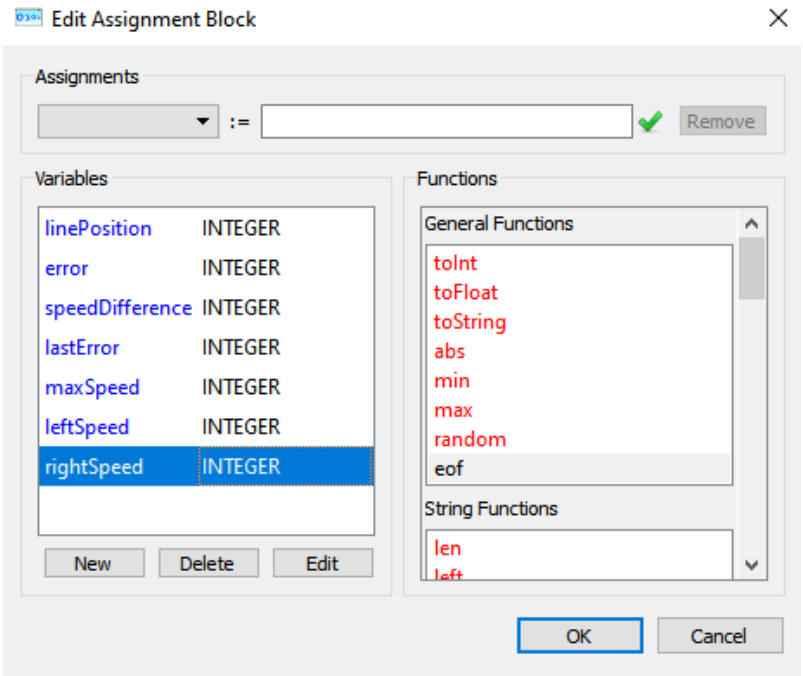




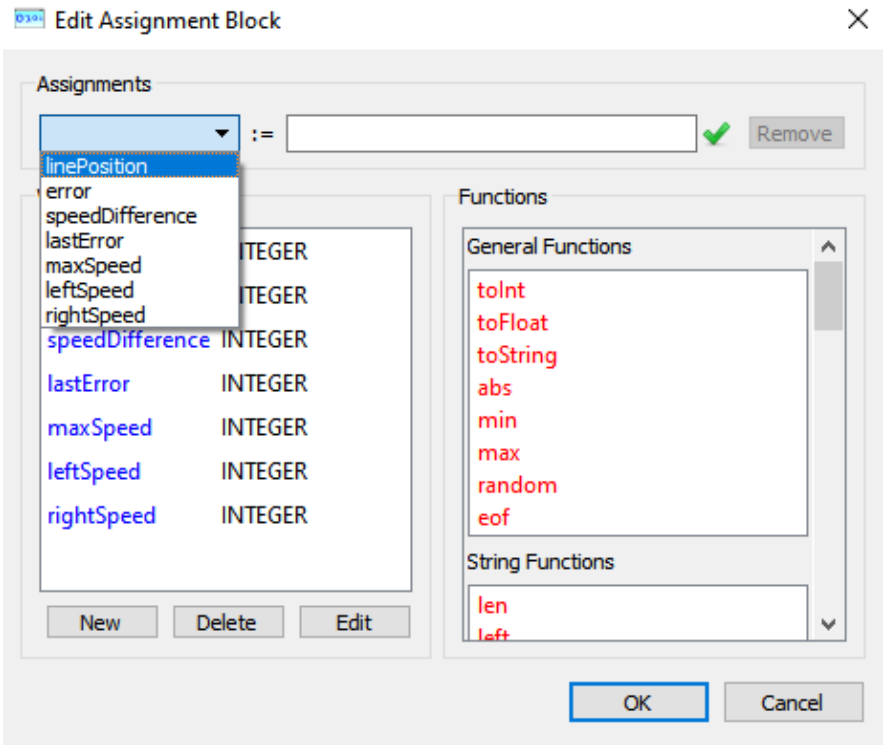
Step 3: Click on Visual Design and create Flow Chart

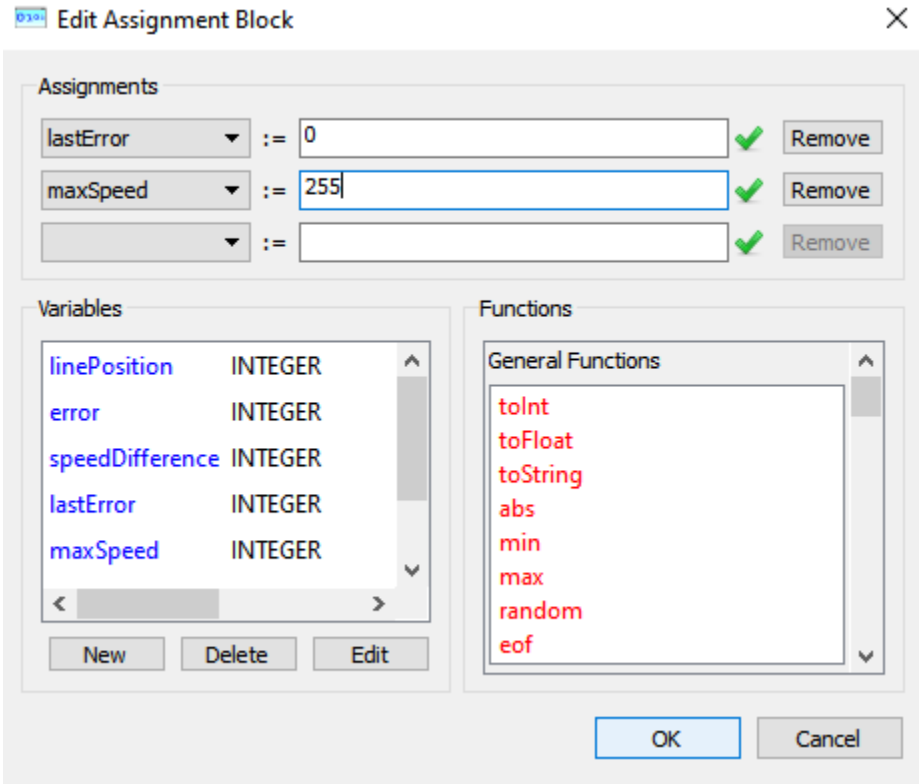


Step 4: Add variables

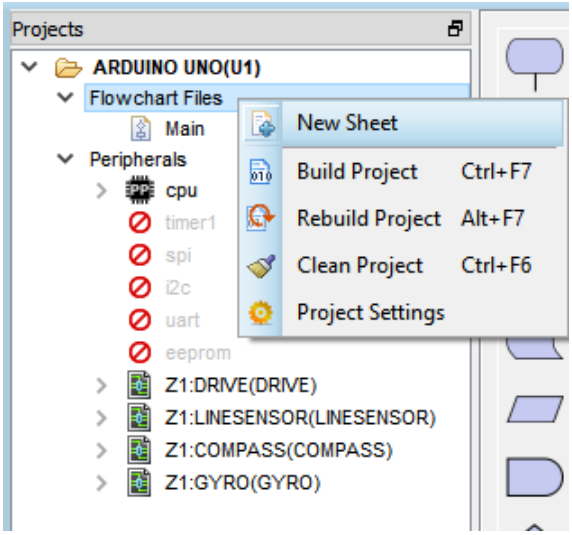


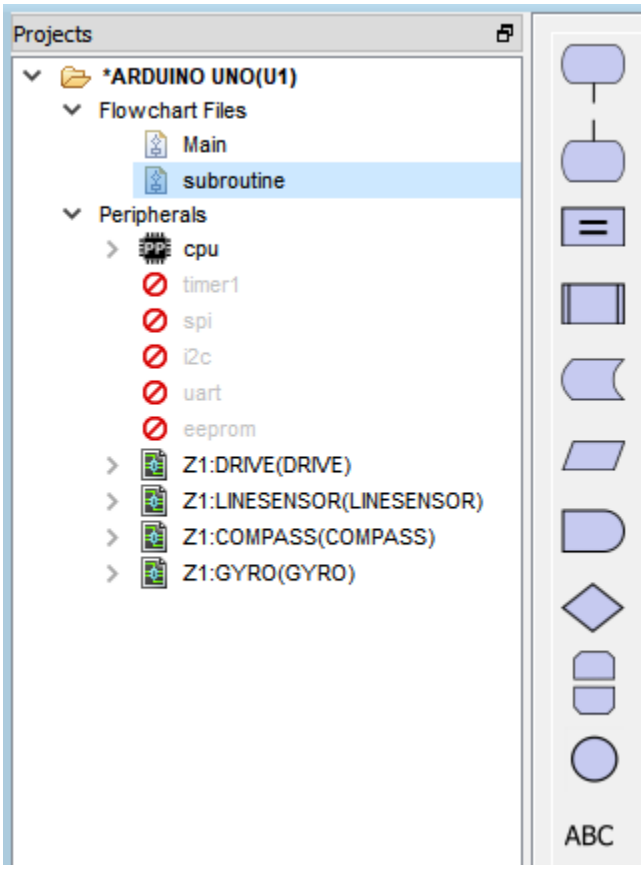
Step 5: Now set Assignments



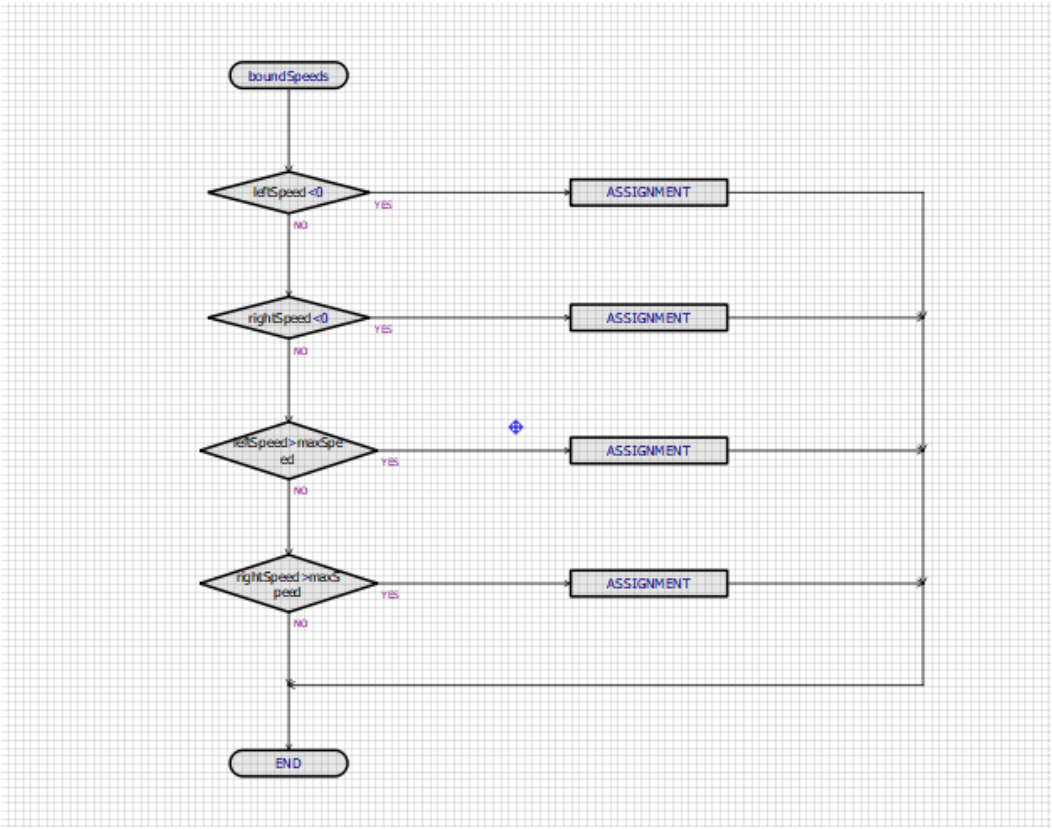


Step 6: Right click on Flowchart Files -> Click on New Sheet

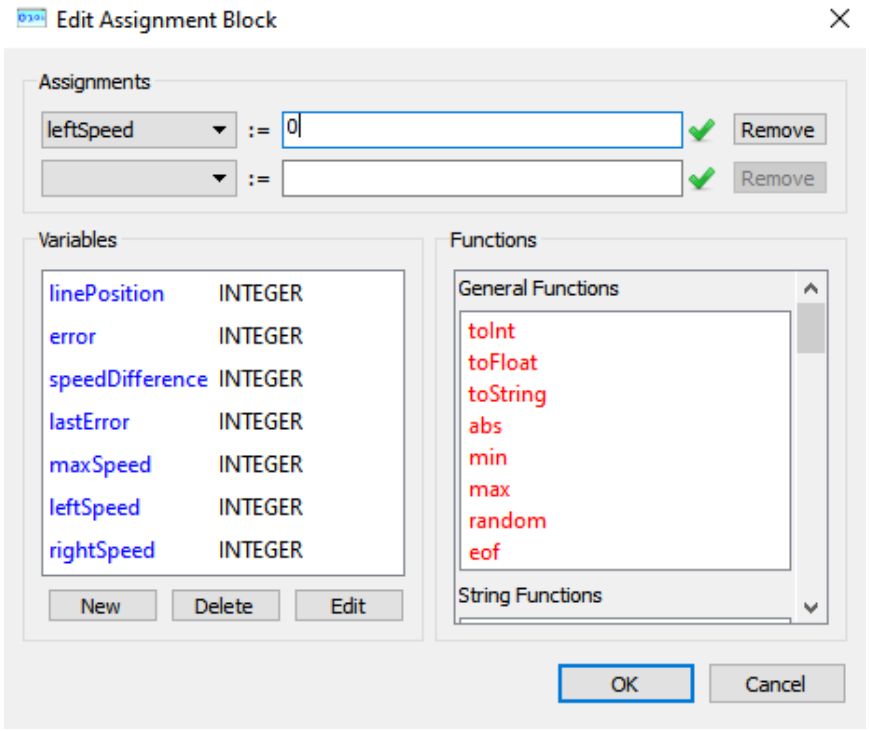




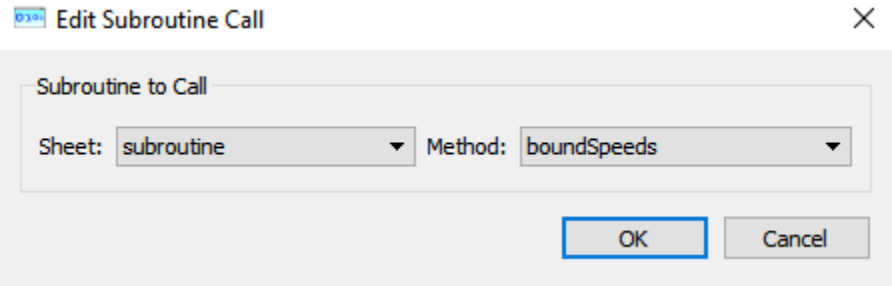
And create flow chart



Then Add Assignments

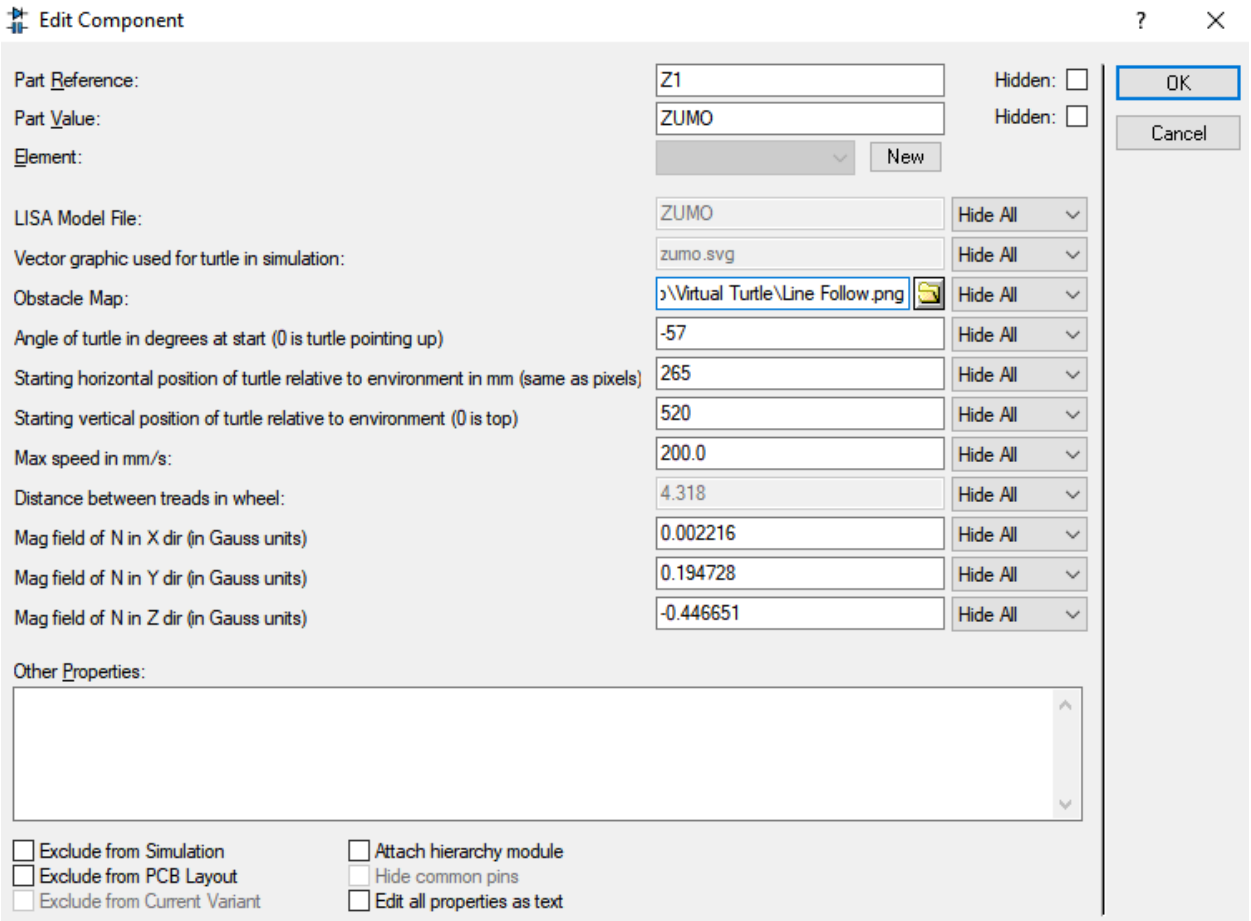


In first flow chart -> Add subroutine call

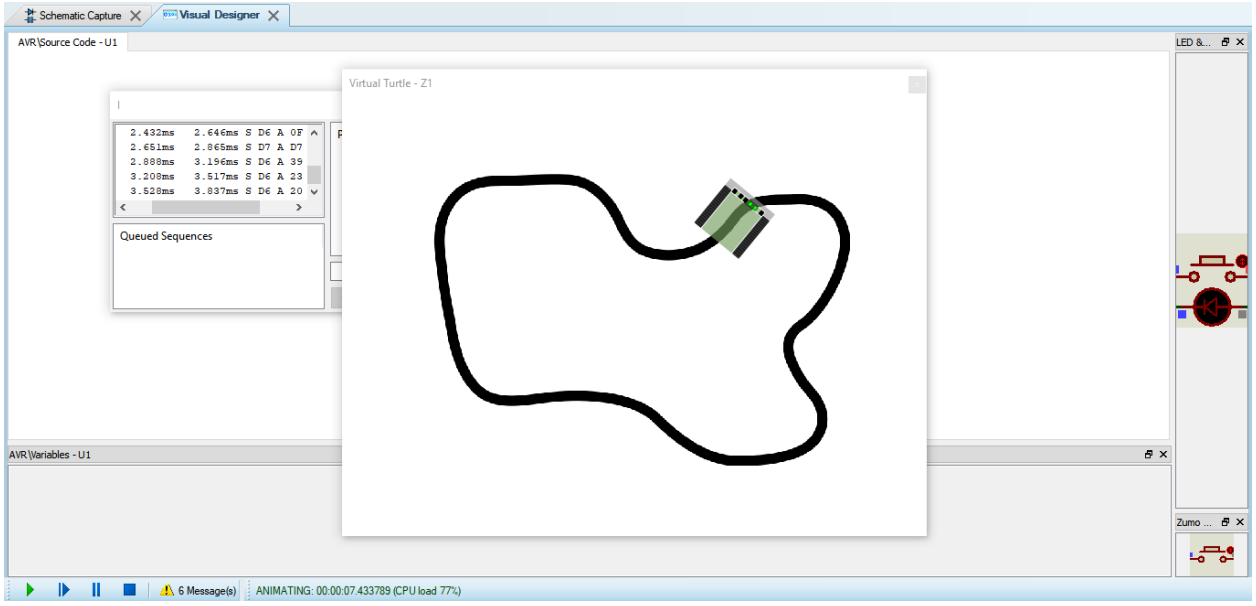


Step 7: Double click on Zumo Z1

- Add Obstacle Map
- Set Angle of turtle in degree to -57
- Set Starting horizontal position to 265
- Set Starting vertical position to 520
- and then click on Ok



Step 8: Click on Run Simulation Button



Practical No. 8

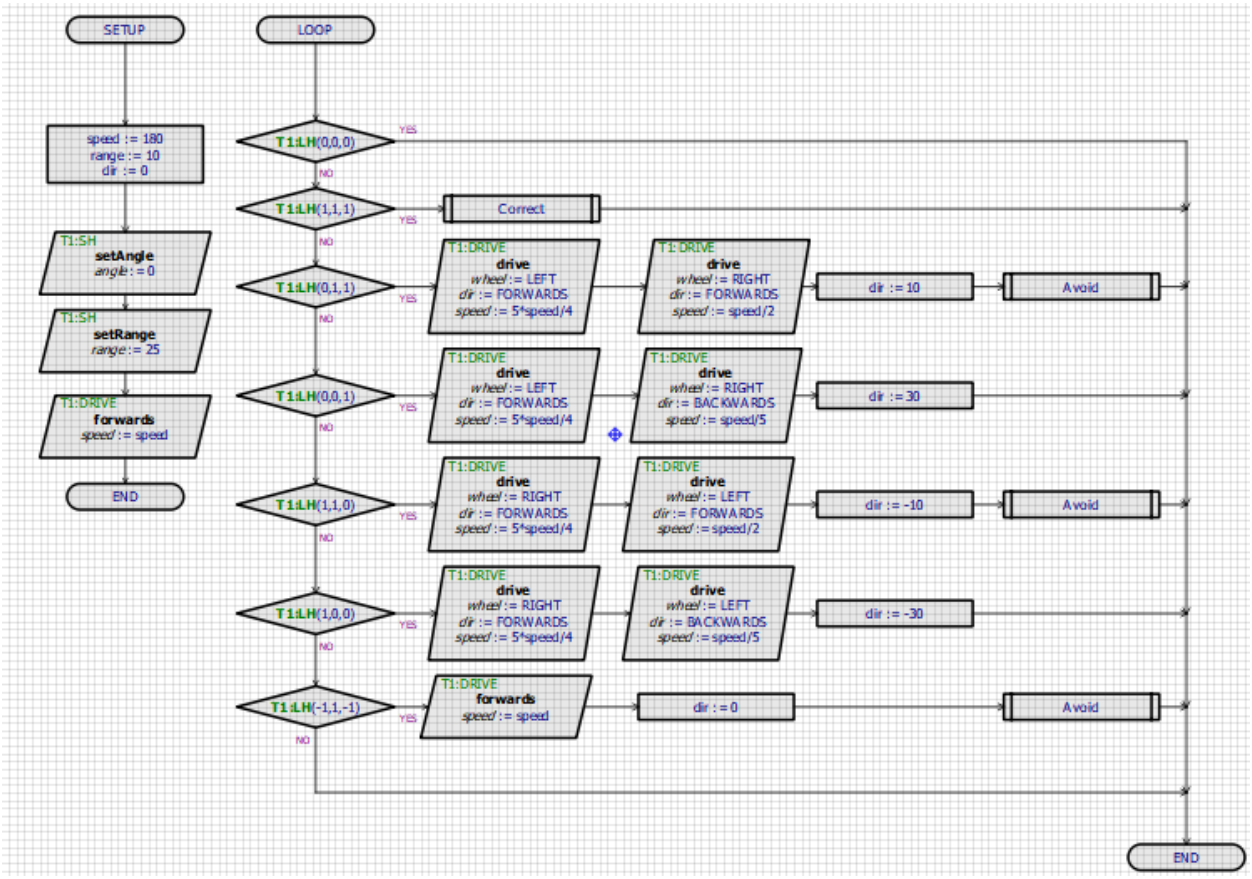
Aim: Virtual turtle Follow and Avoid

Components Used:

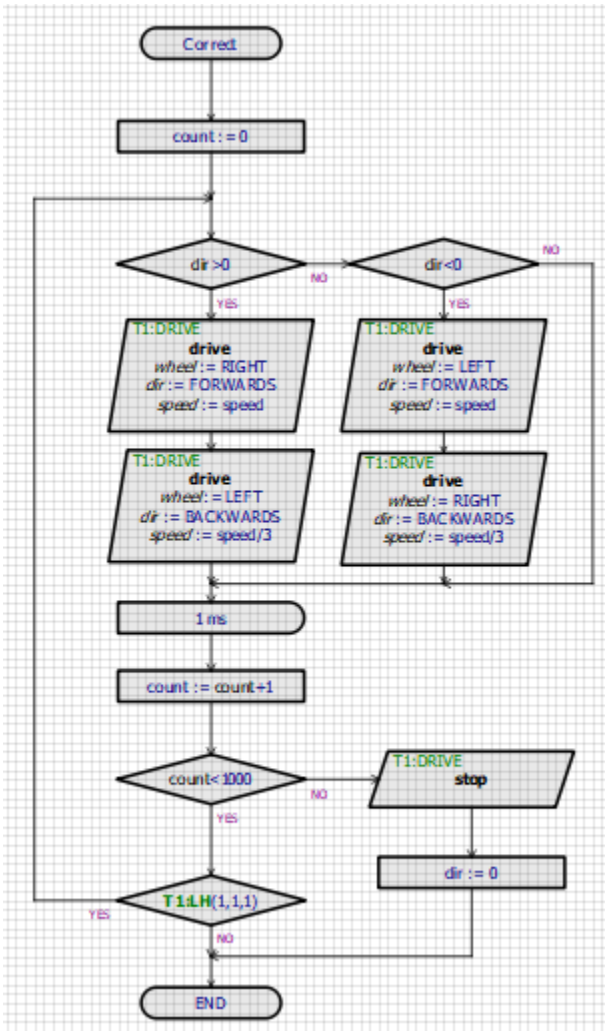
1. Arduino UNO
2. Arduino Turtle

Flow Chart:

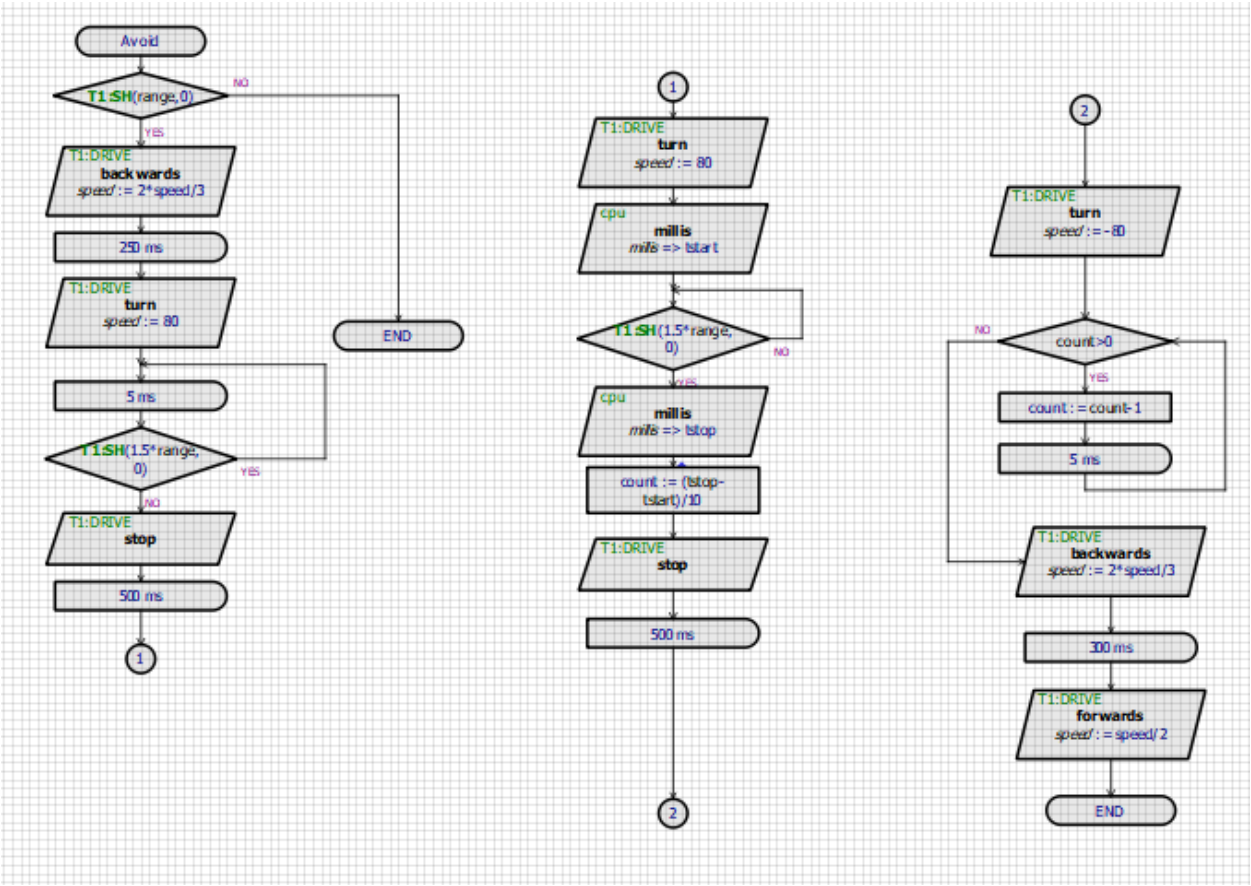
Main



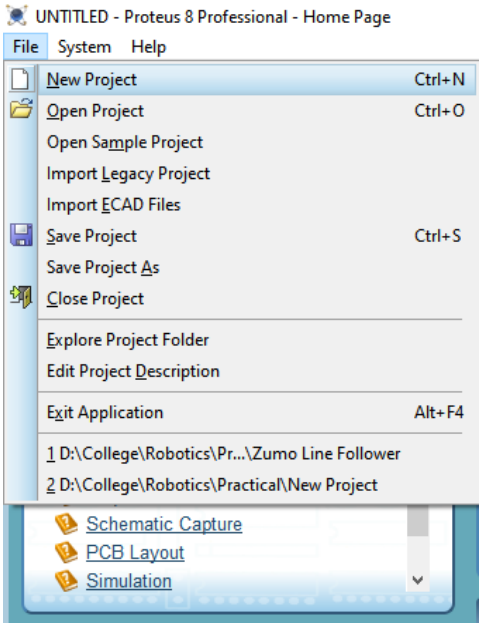
Correct

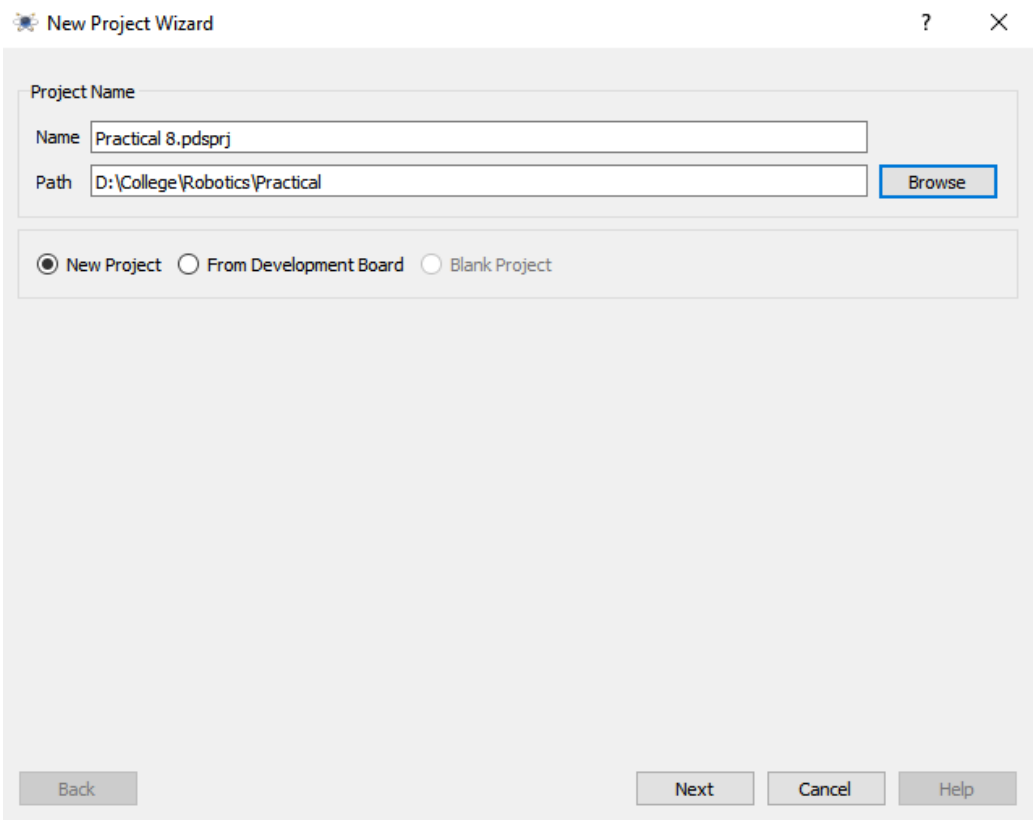

Correct
count : = 0
dir >0
NO
YES
dir <0
NO
YES
T1:DRIVE
drive
wheel : = RIGHT
dir : = FORWARDS
speed : = speed
T1:DRIVE
drive
wheel : = LEFT
dir : = FORWARDS
speed : = speed
T1:DRIVE
drive
wheel : = LEFT
dir : = BACKWARDS
speed : = speed/3
T1:DRIVE
drive
wheel : = RIGHT
dir : = BACKWARDS
speed : = speed/3
1 ms
count : = count+1
count< 1000
NO
YES
T1:DRIVE
stop
dir : = 0
T1:LH(1,1,1)
YES
NO
END

Avoid

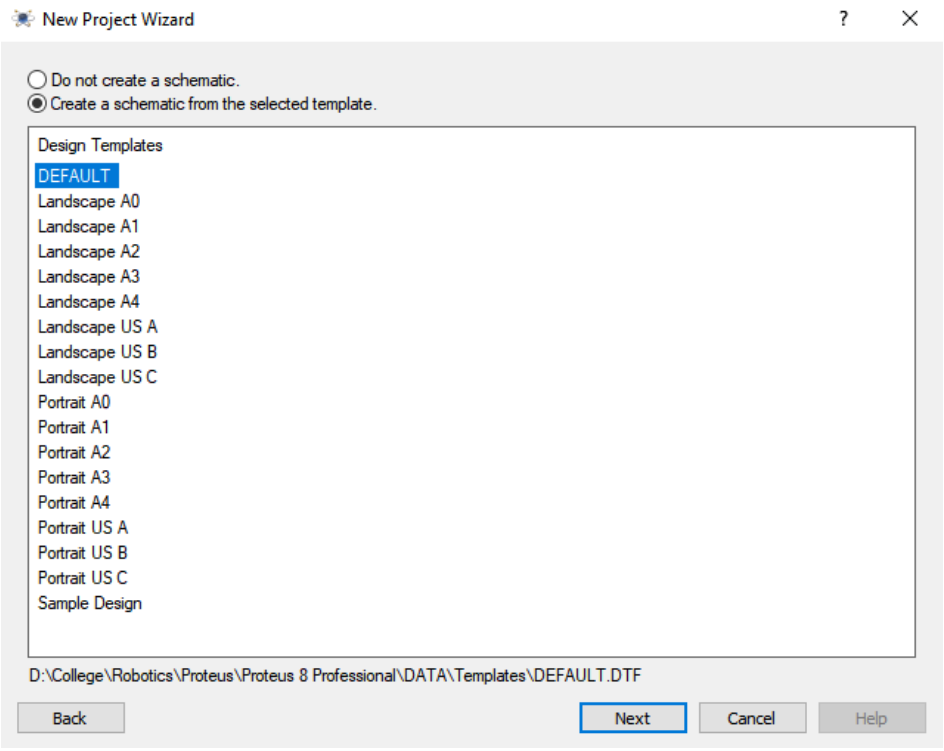


Step 1: Open Proteus > Click on File > Click on New Project

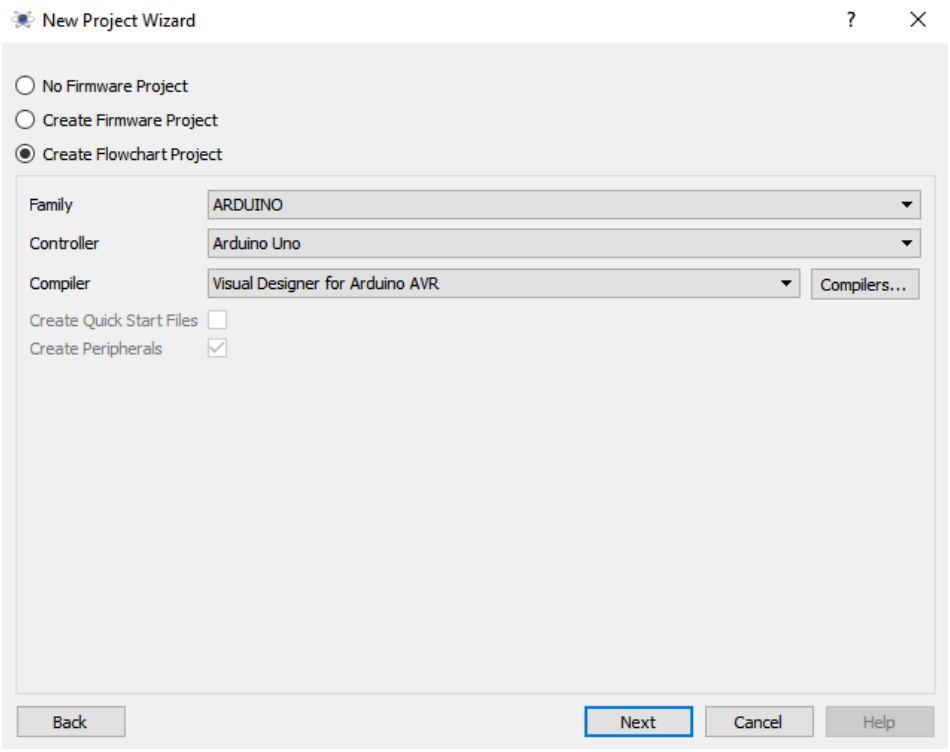




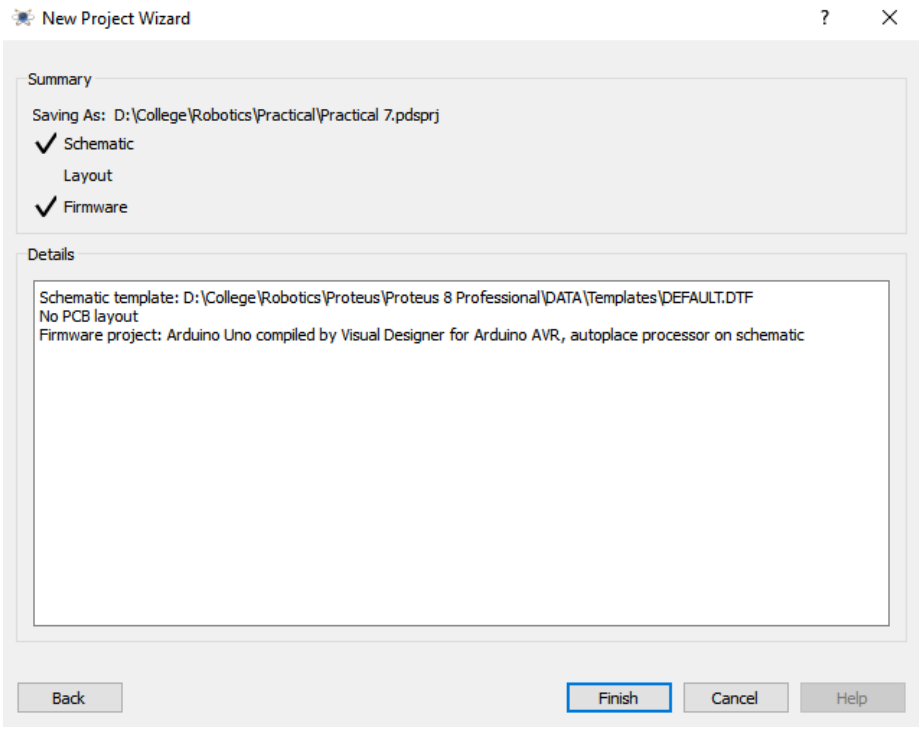
Set it to Default and Click on Next



Now Select Create Flowchart Project option

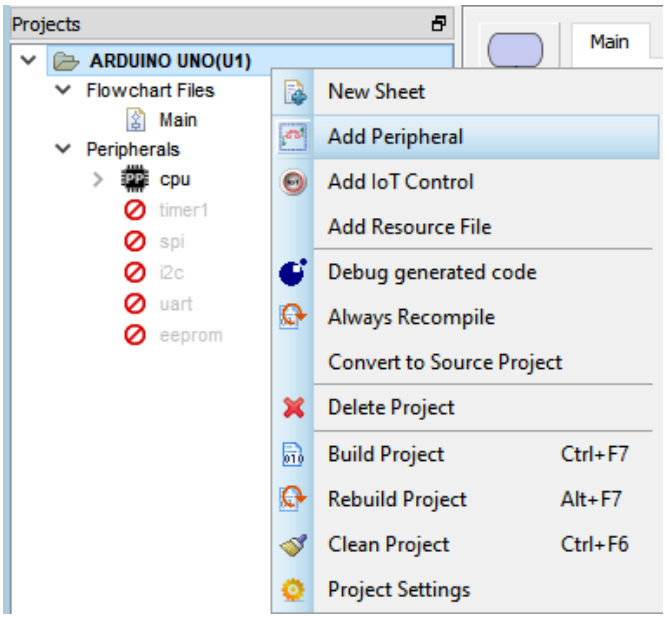


Then Click on Finish

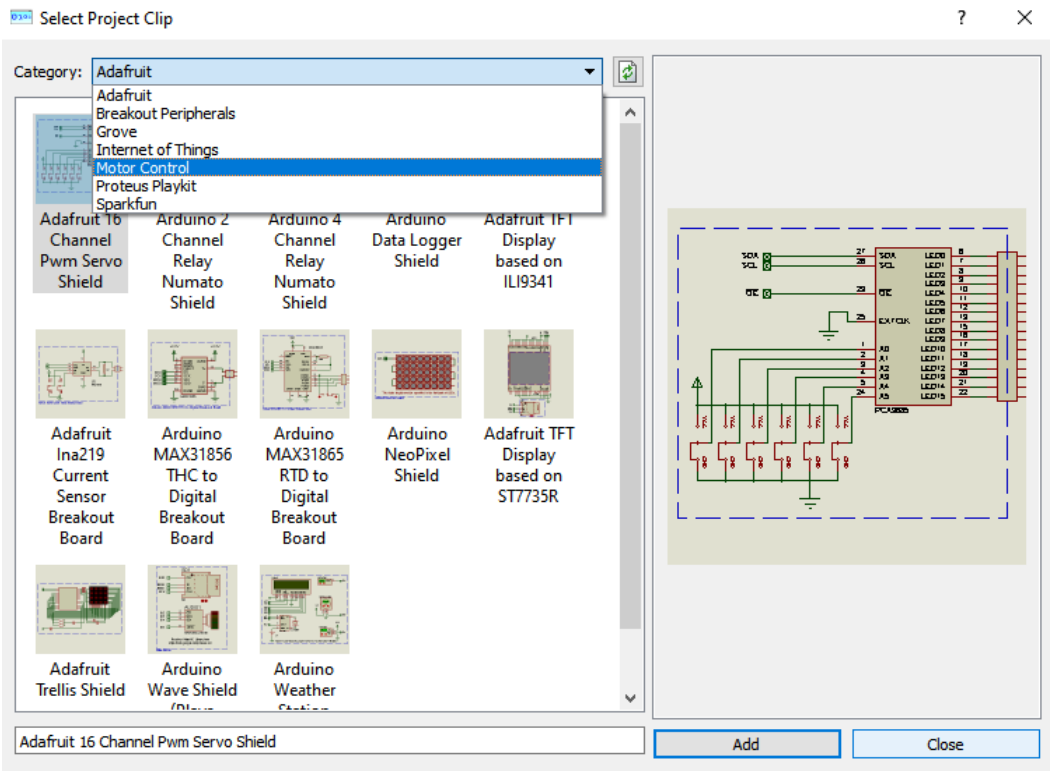


Step 2: Add Arduino Turtle

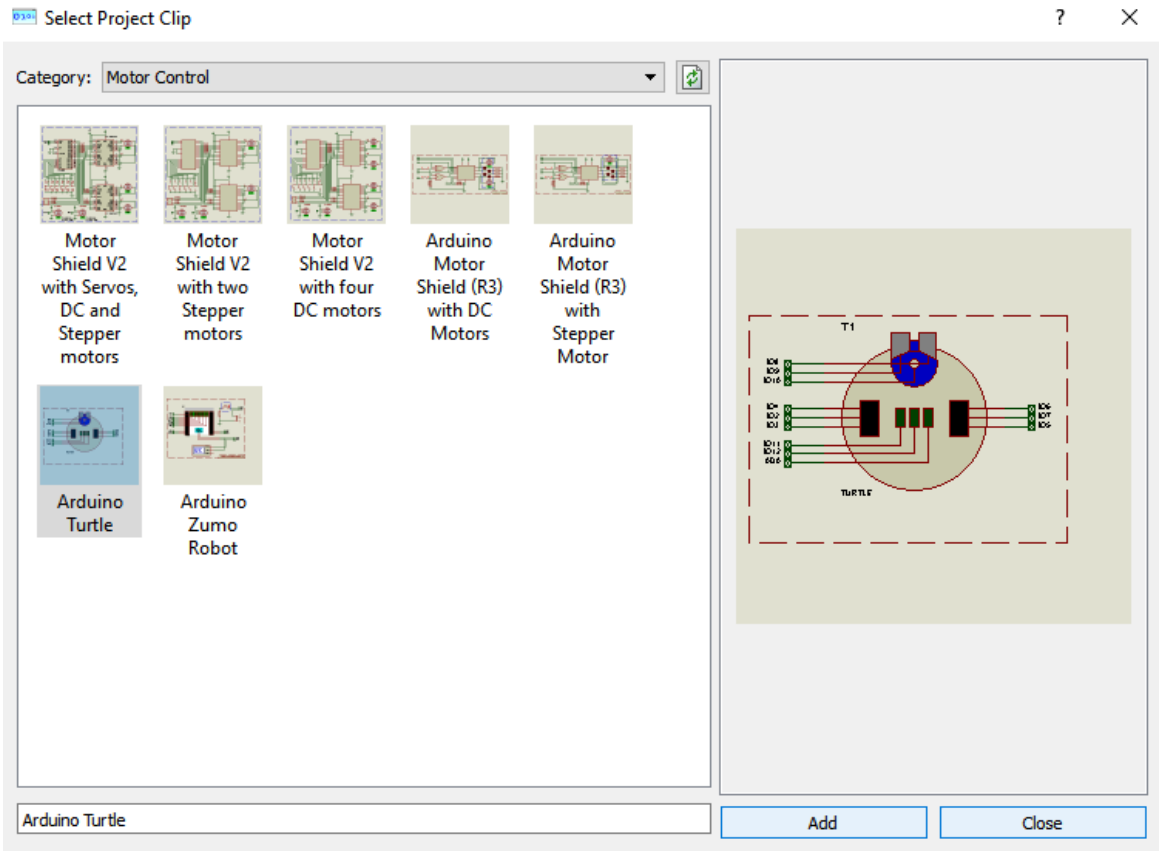
Right click on Project Name -> Click on Add Peripherals

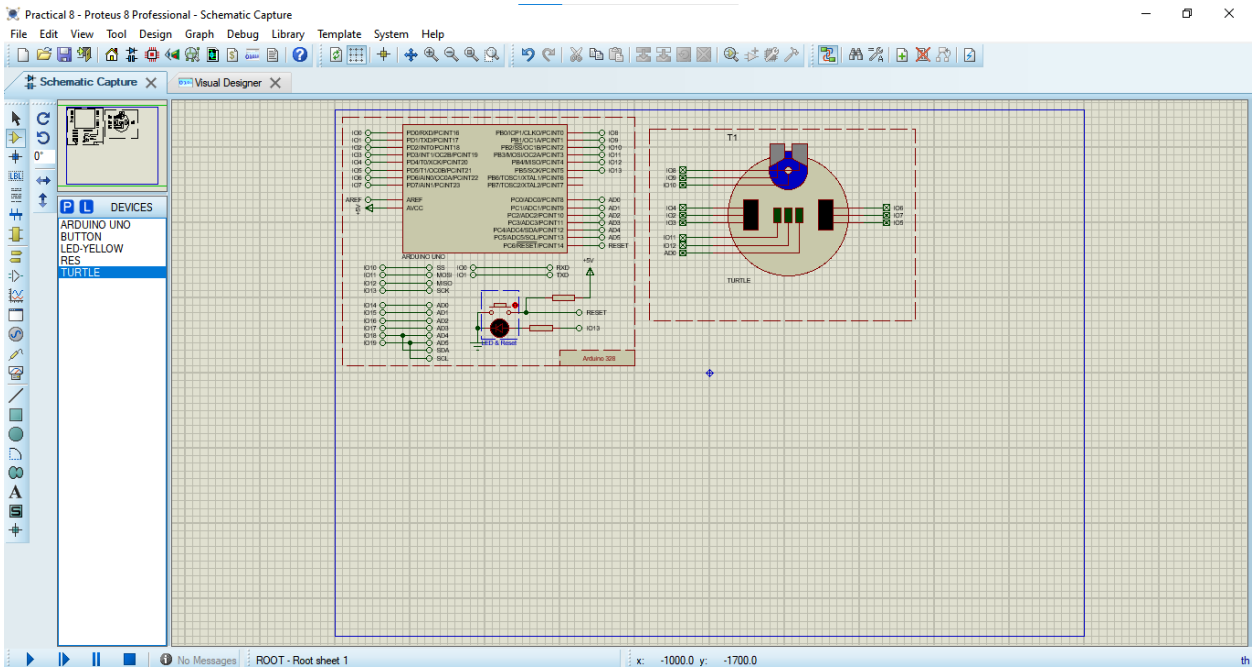


From Dropdown -> Select Motor Control

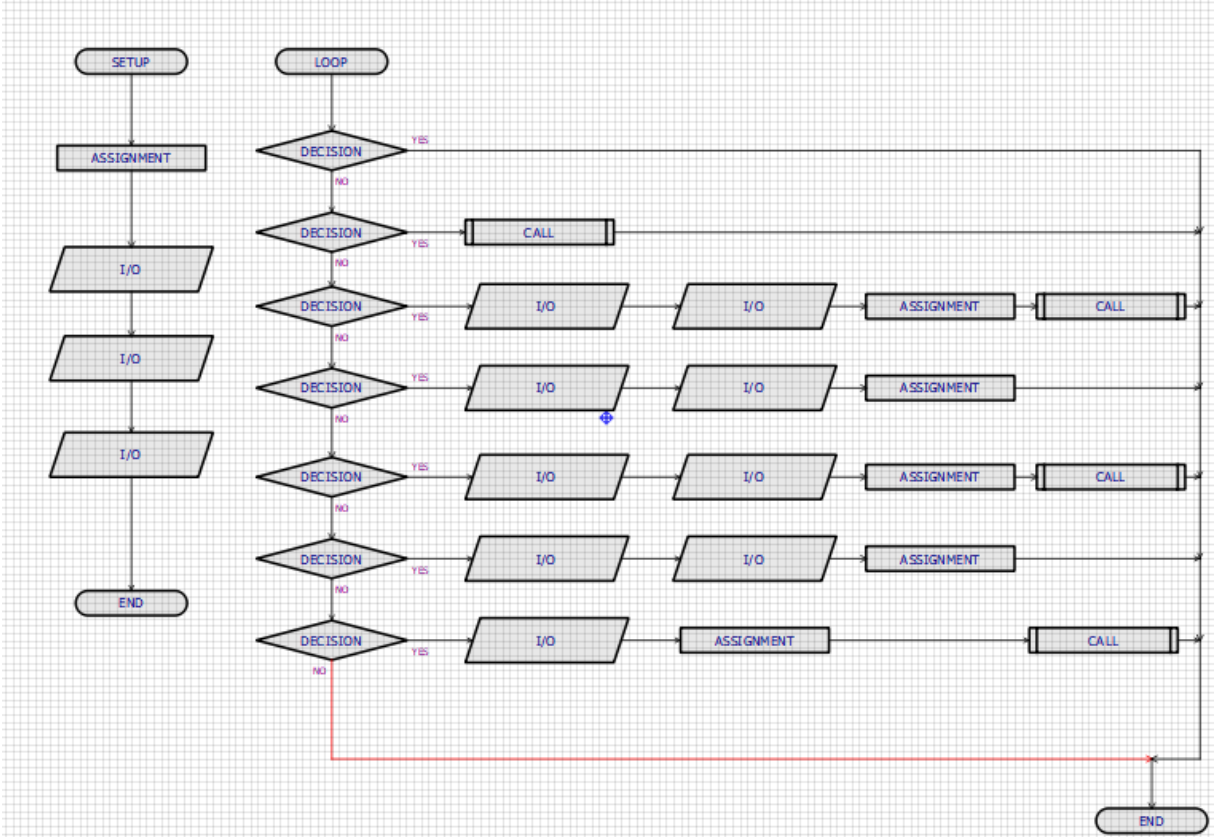


And then Select Arduino Turtle and Click on Add

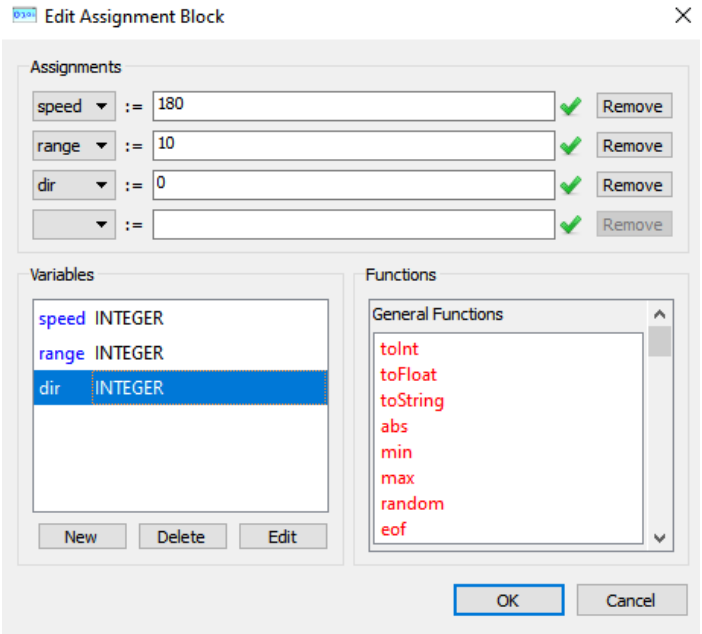




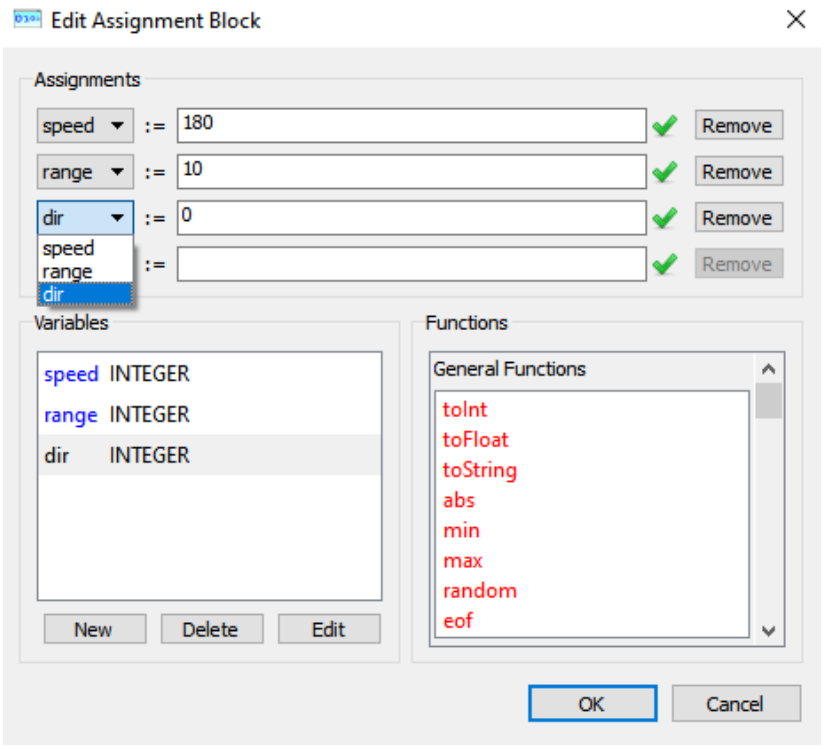
Step 3: Click on Visual Design and create Flow Chart

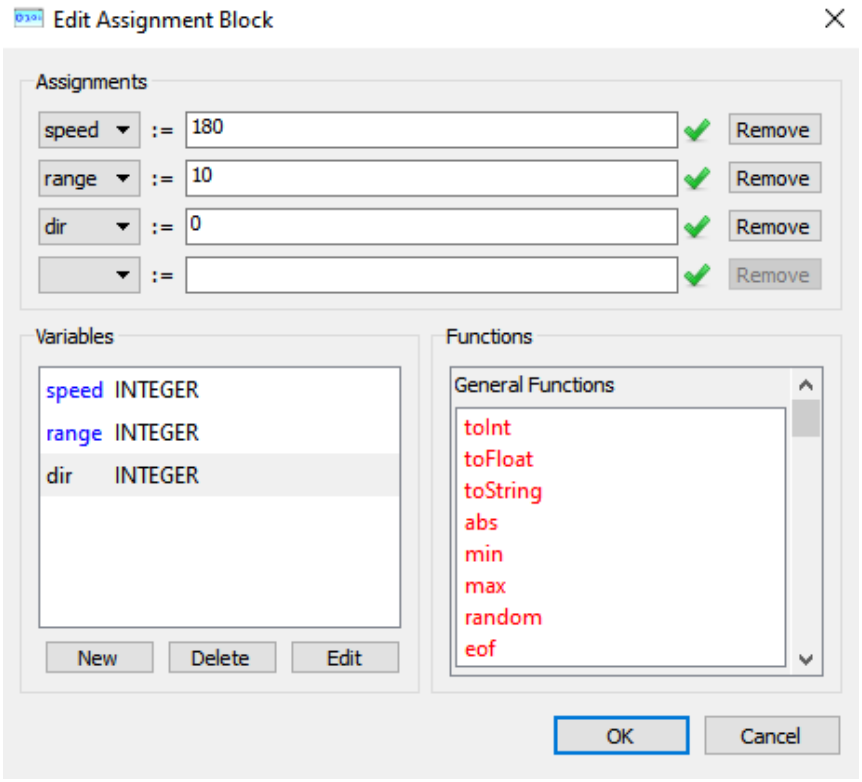


Step 4: Add variables

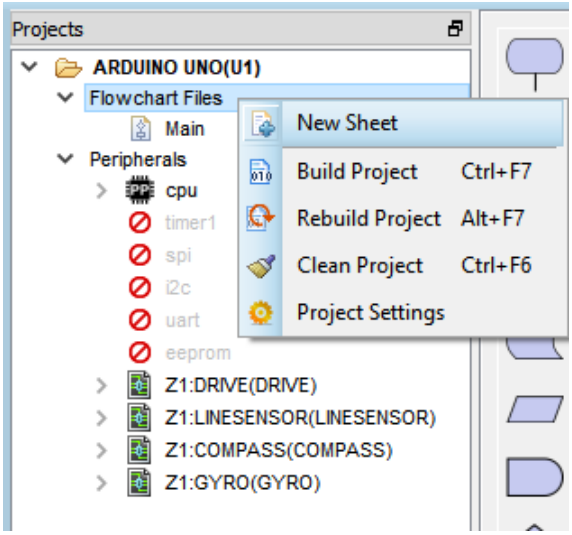


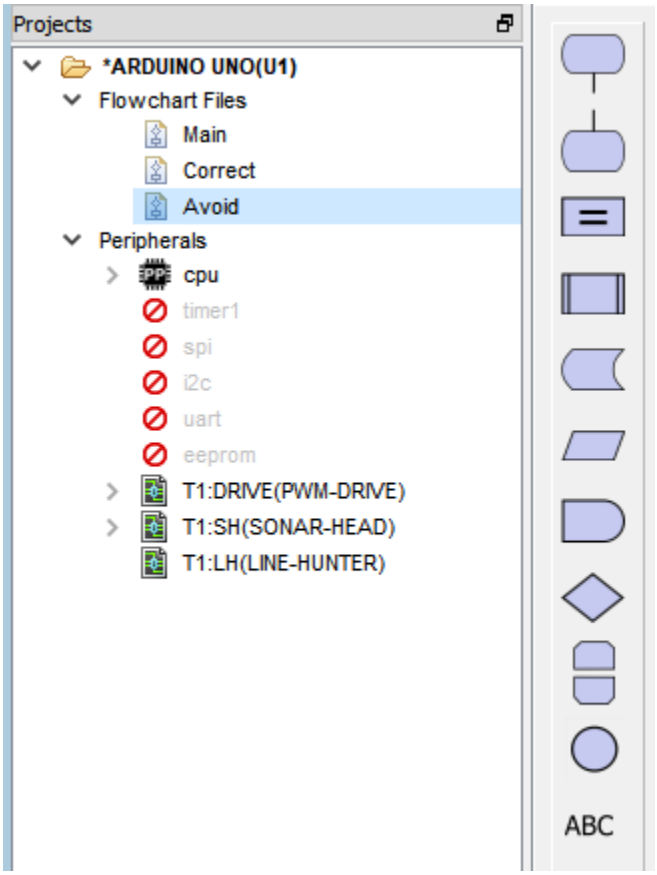
Step 5: Now set Assignments



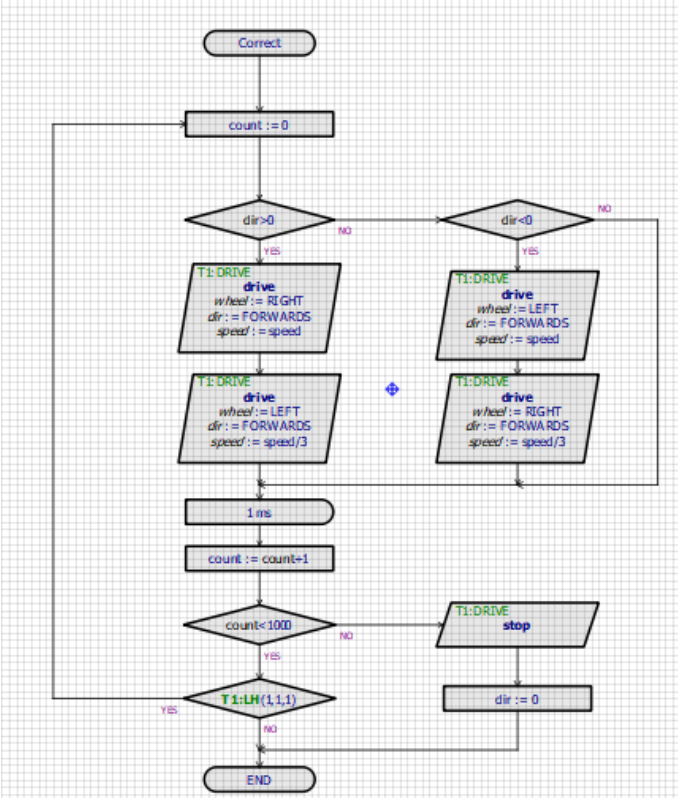


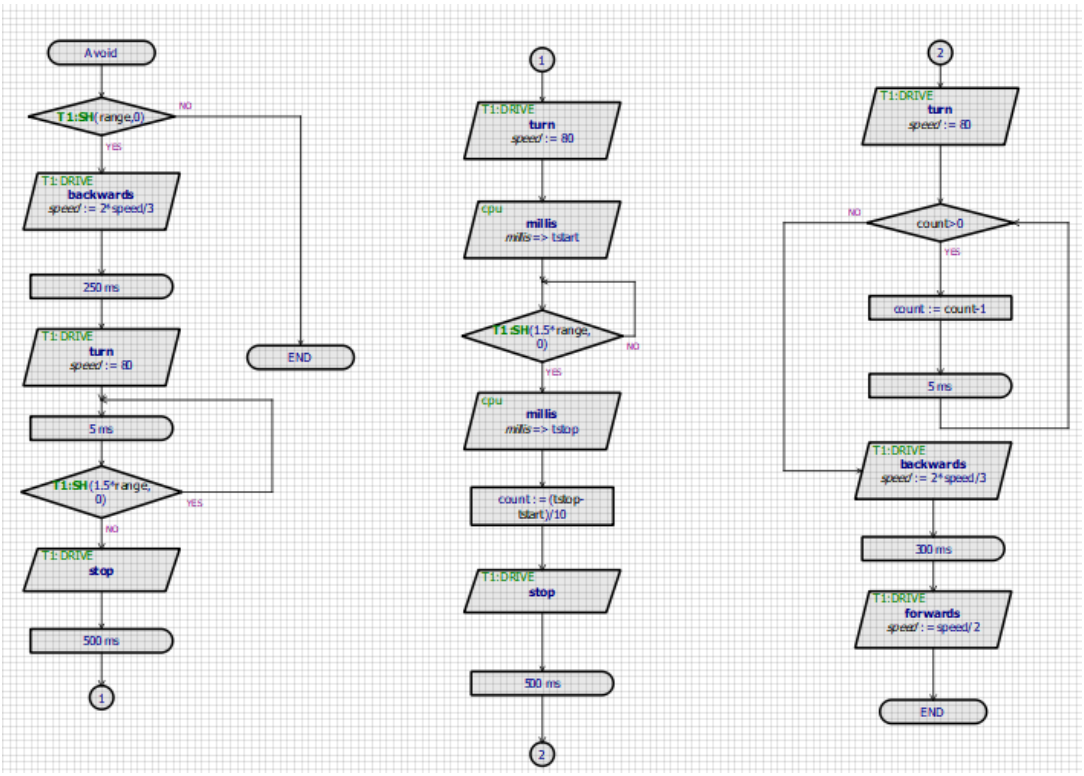
Step 6: Right click on Flowchart Files -> Click on New Sheet



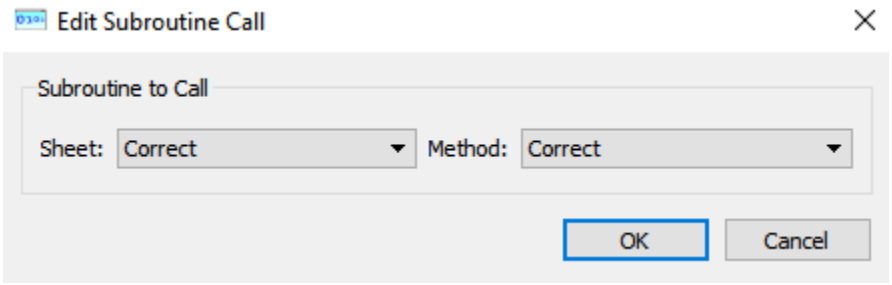


Create flow chart for Correct and Avoid



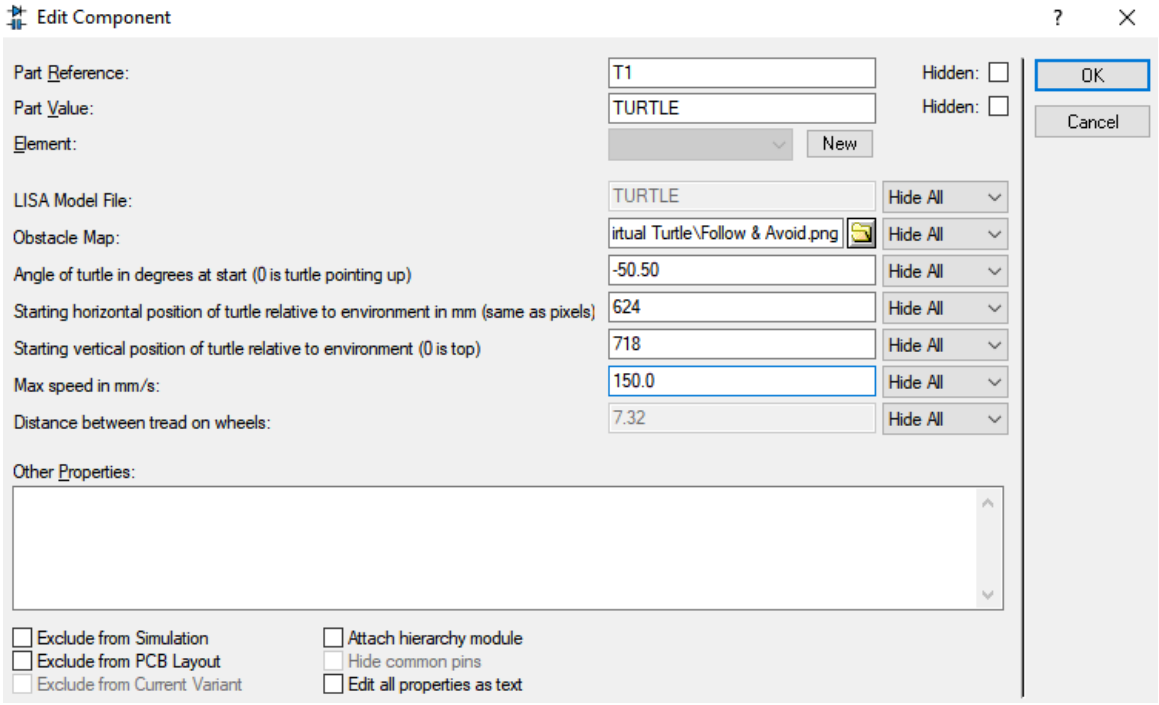


Step 7: In Main chart add Correct and Avoid chart

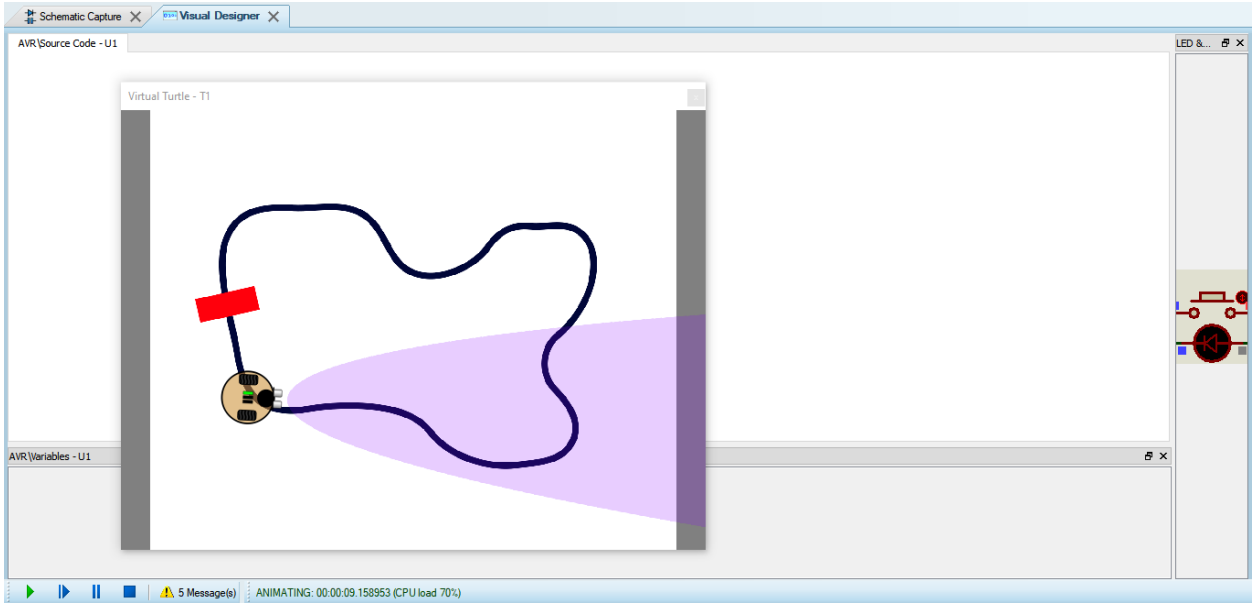


Step 8: Double click on Turtle

- Add Obstacle Map
- Set Angle of turtle in degree to -50.50
- Set Starting horizontal position to 624
- Set Starting vertical position to 718
- and then click on Ok



Step 8: Click on Run Simulation Button



Practical No. 9

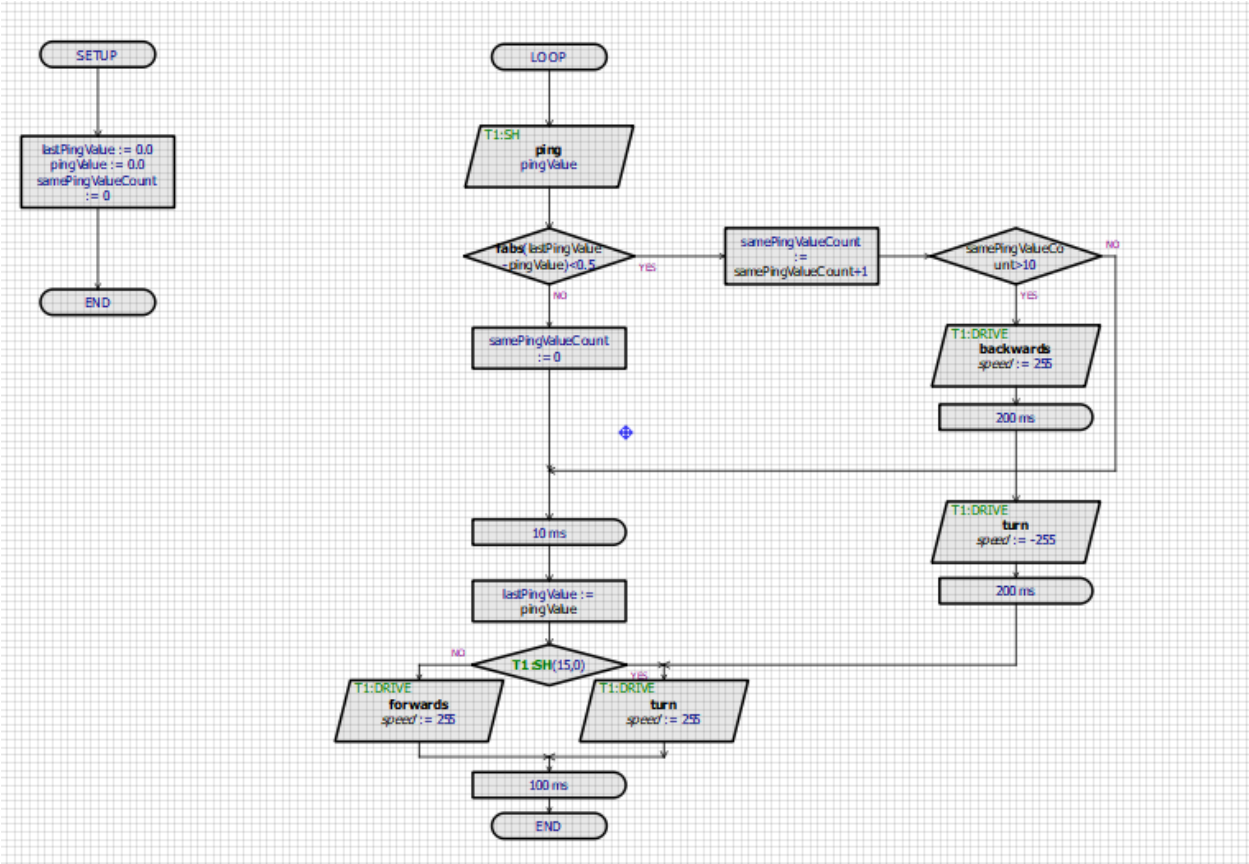
Aim: Avoid Obstacles

Component Used:

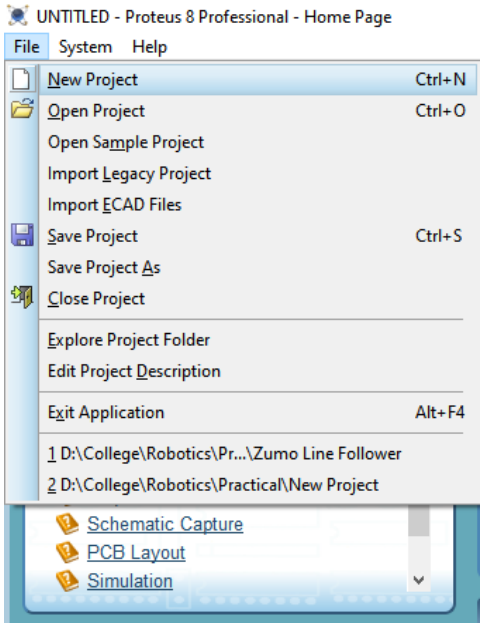
1. Arduino UNO
2. Arduino Turtle

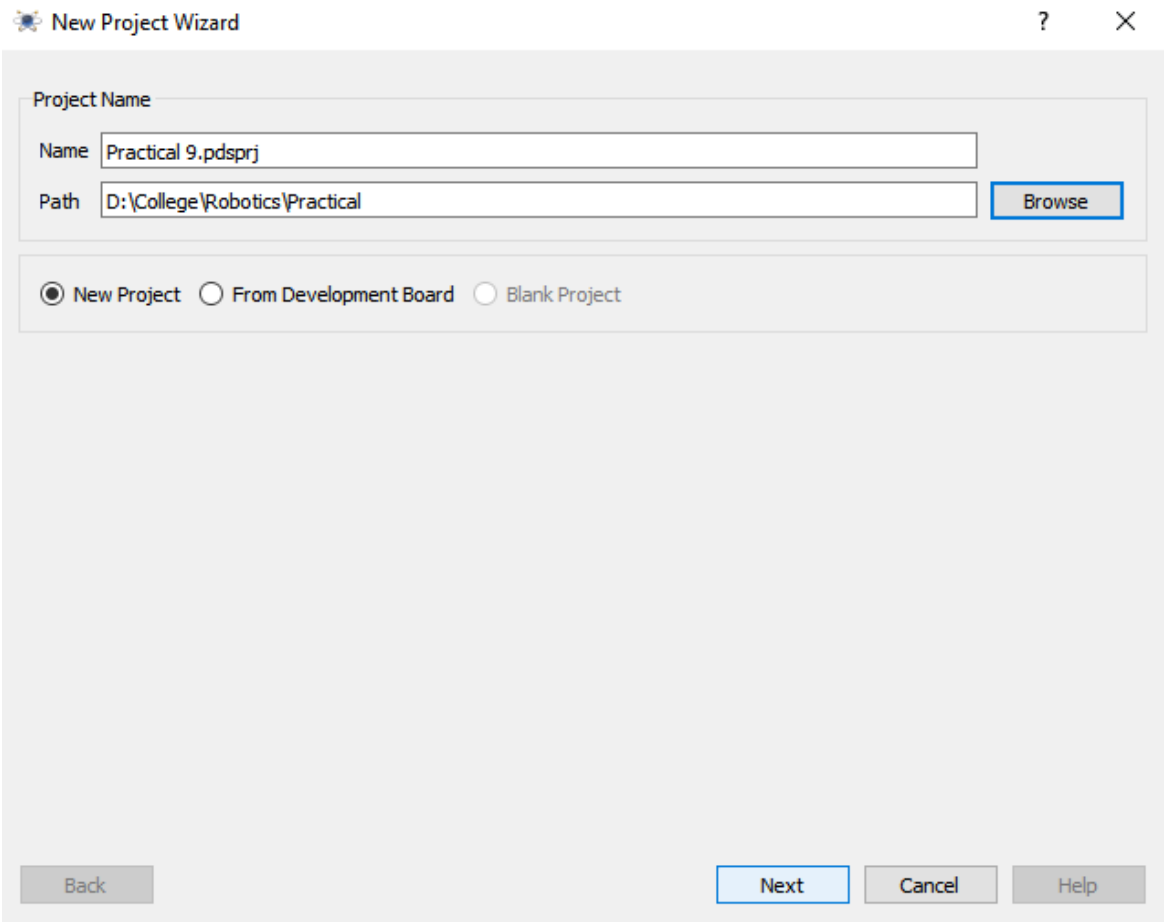
Flow Chart:

Main

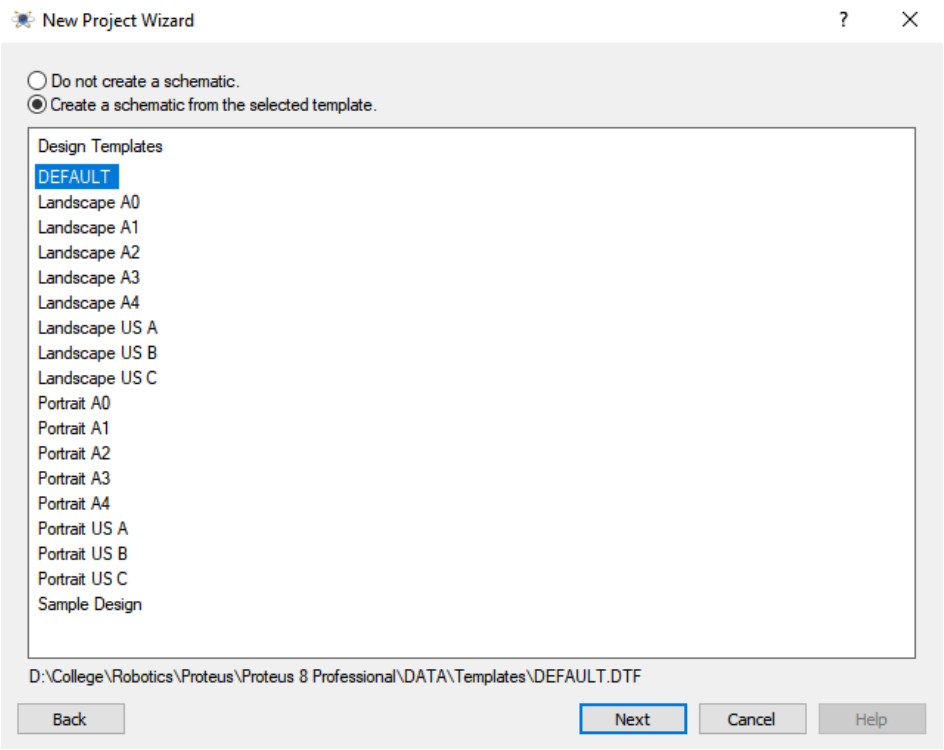


Step 1: Open Proteus > Click on File > Click on New Project

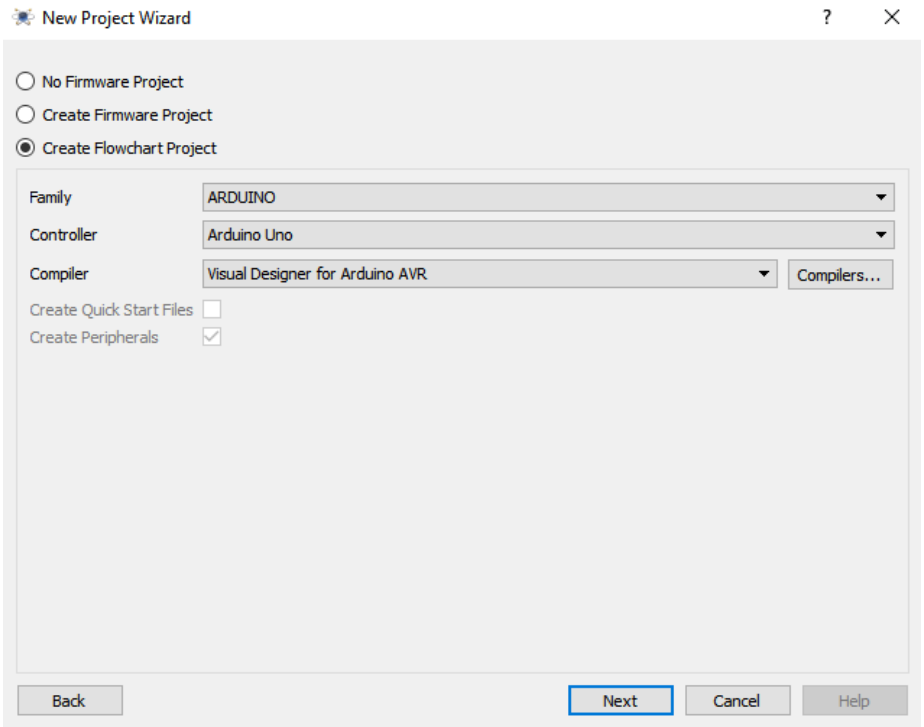




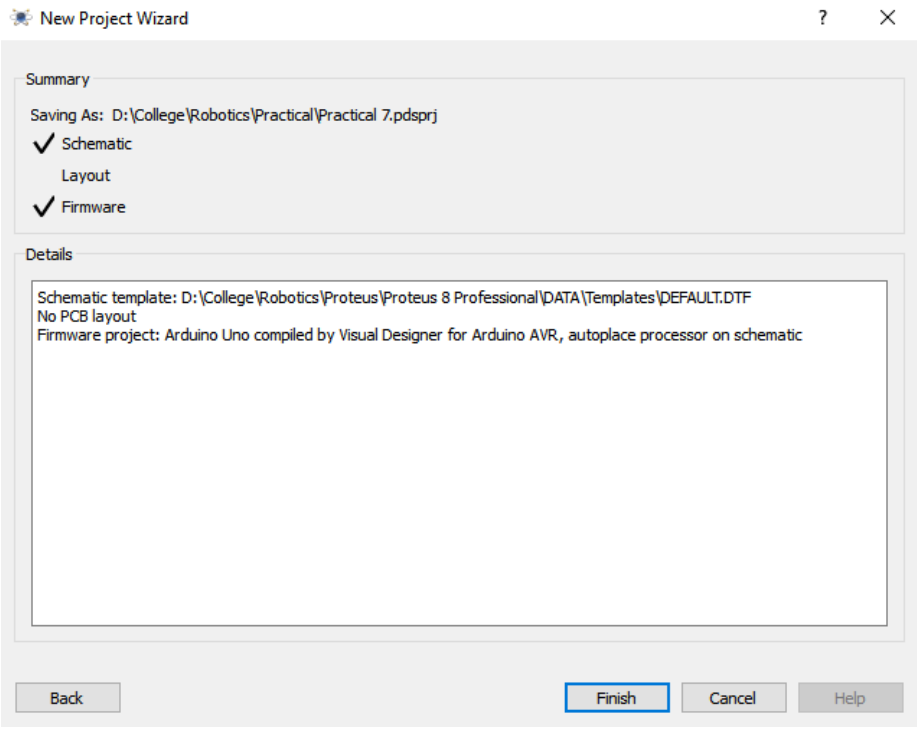
Set it to Default and Click on Next



Now Select Create Flowchart Project option

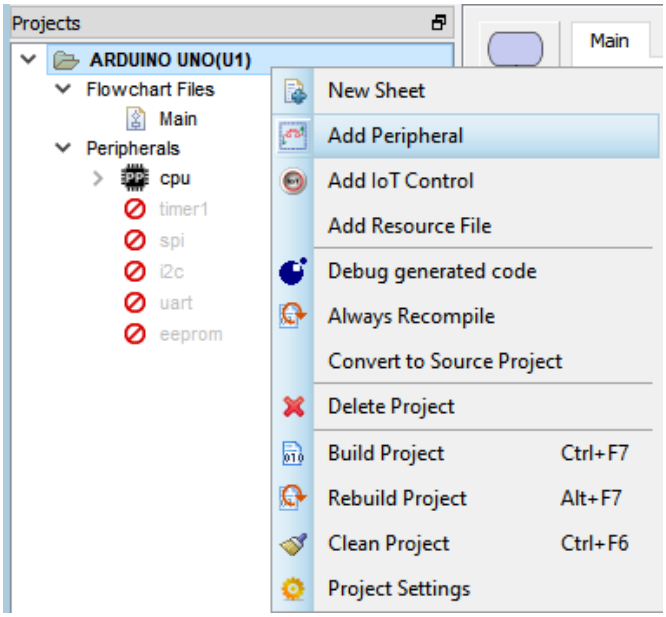


Then Click on Finish

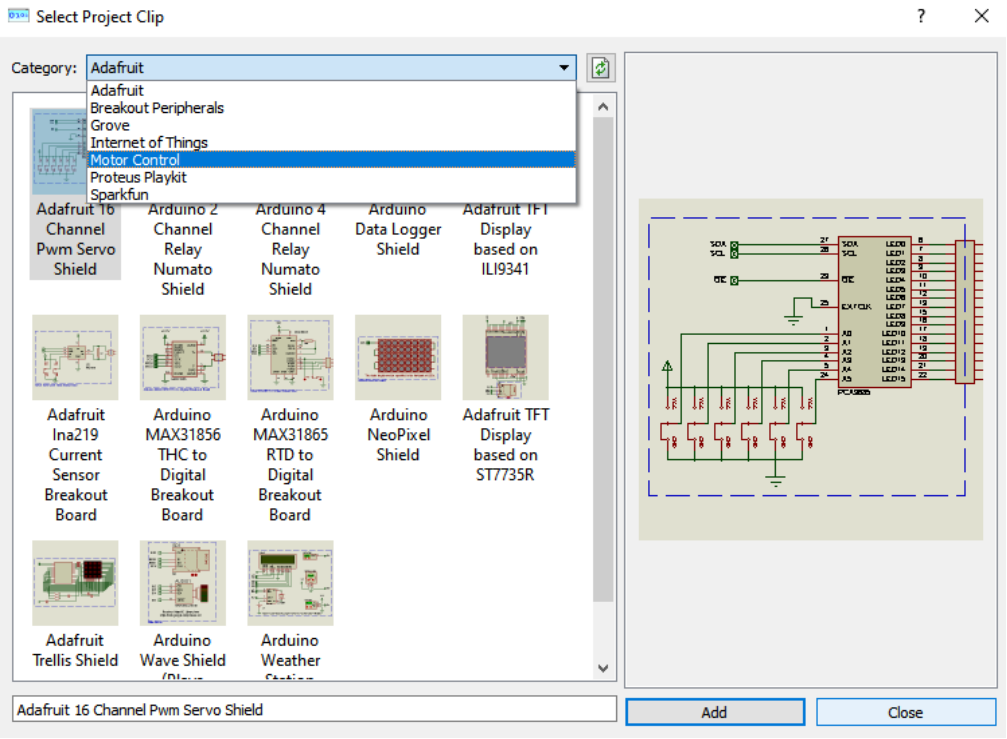


Step 2: Add Arduino Turtle

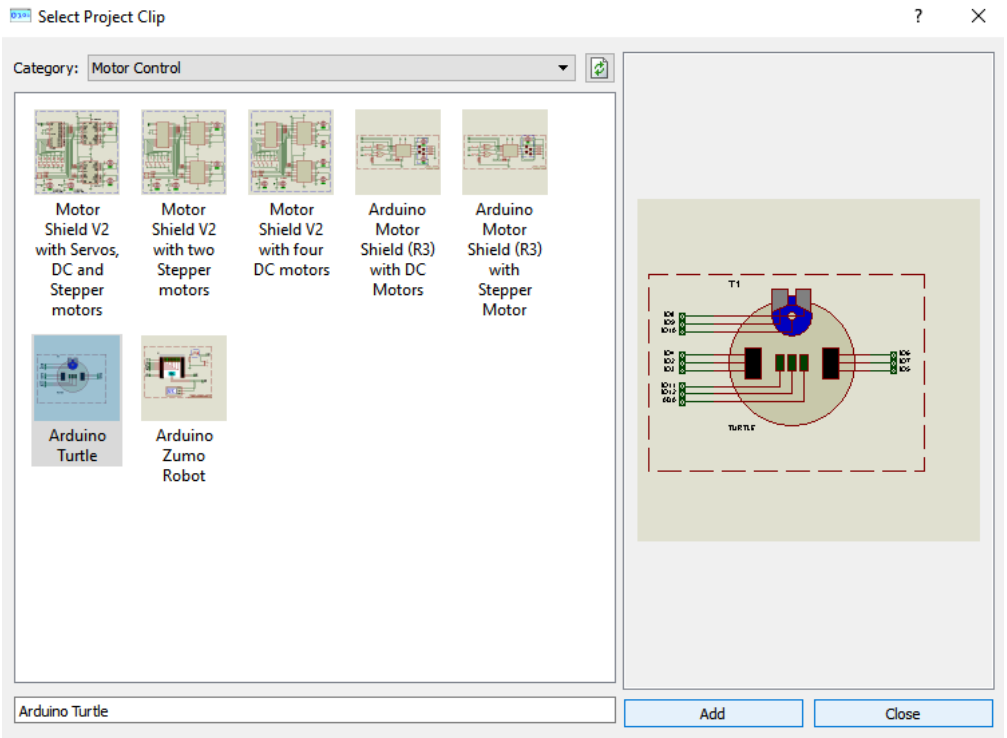
Right click on Project Name -> Click on Add Peripherals

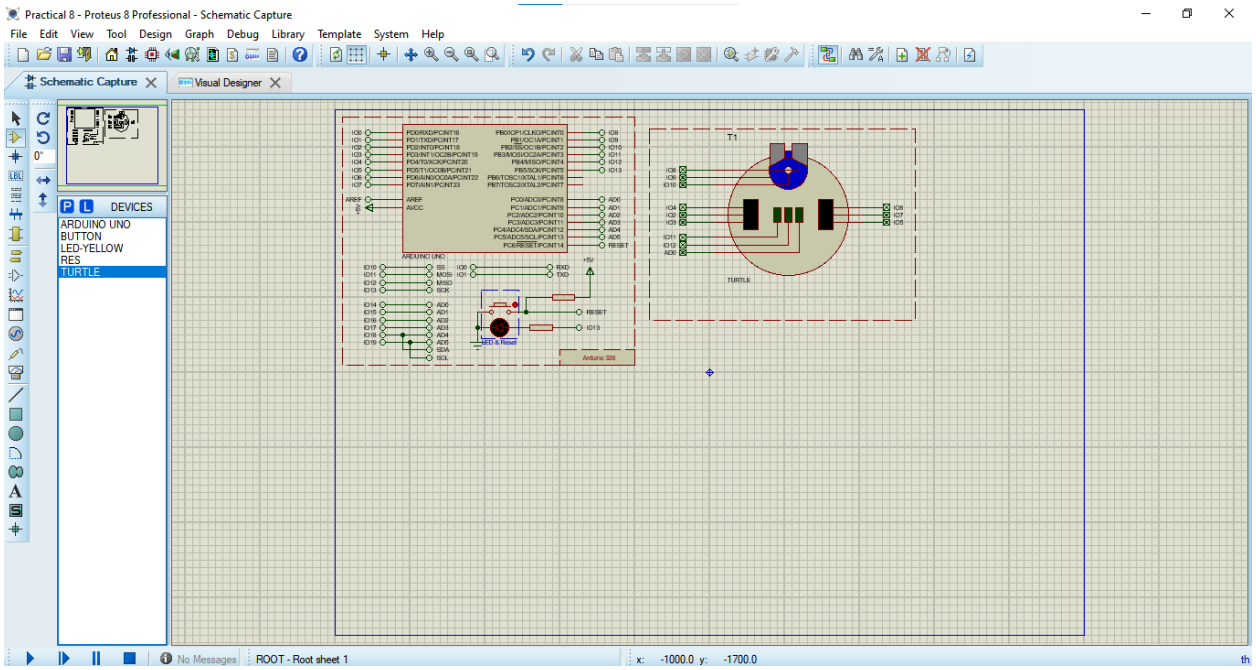


From Dropdown -> Select Motor Control

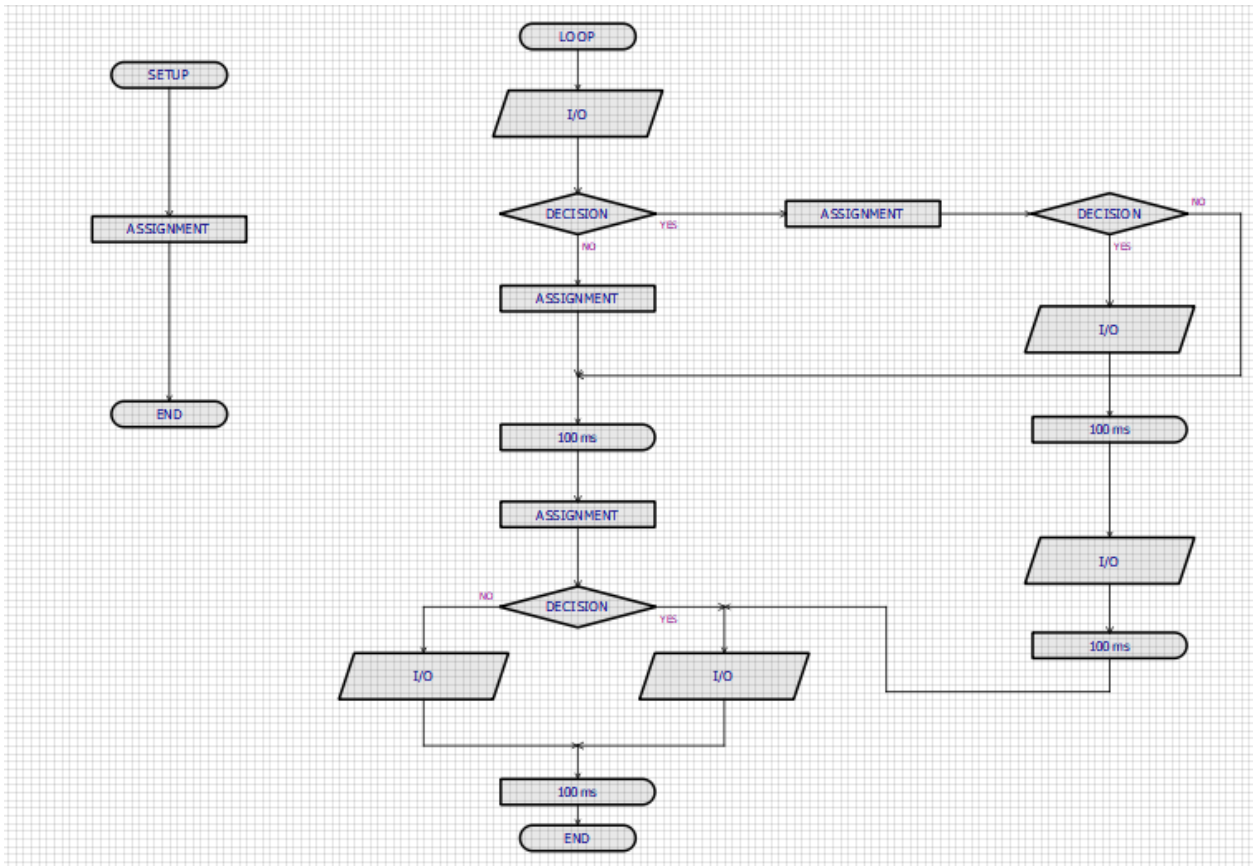


And then Select Arduino Turtle and Click on Add

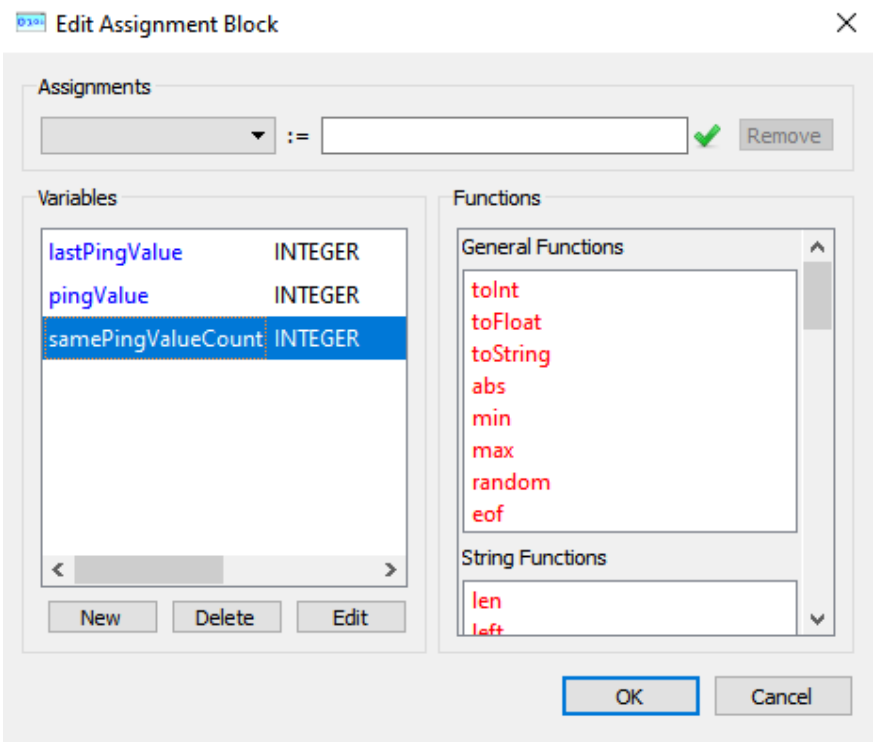




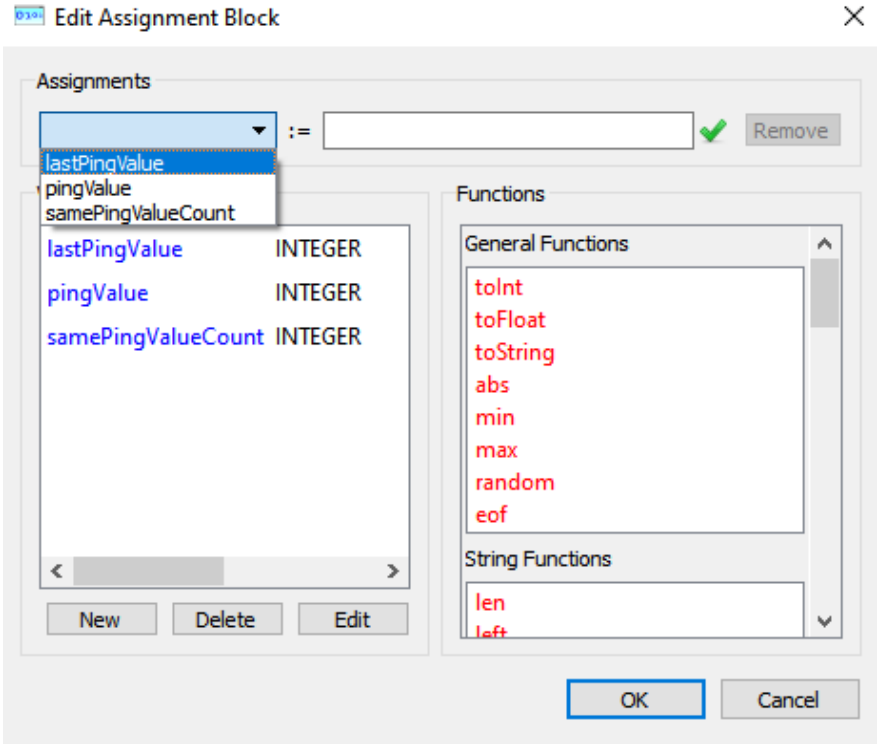
Step 3: Click on Visual Design and create Flow Chart

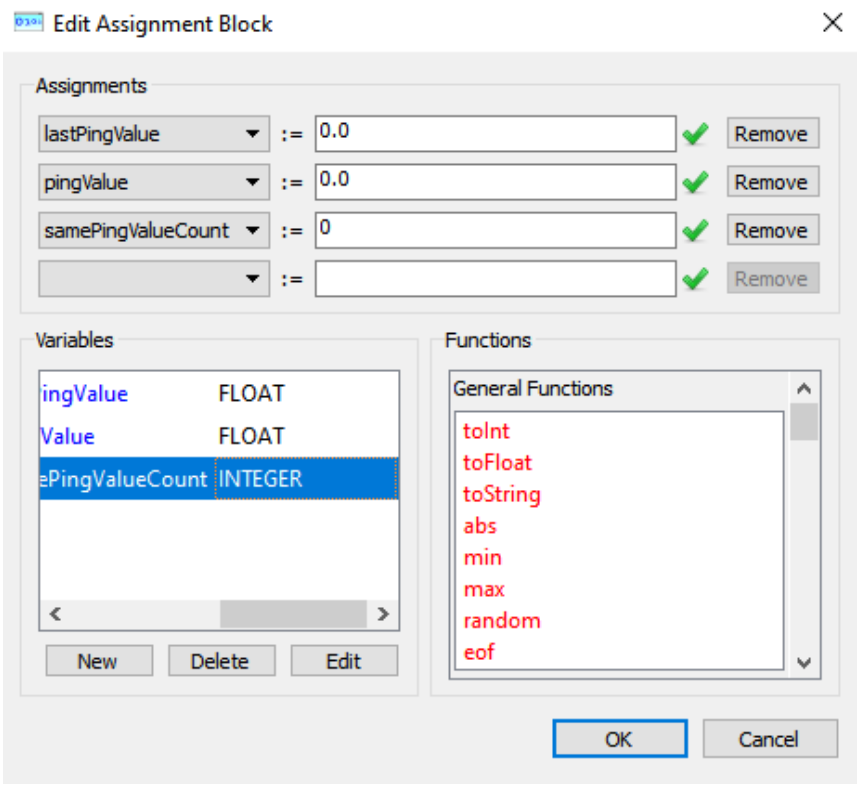


Step 4: Add variables

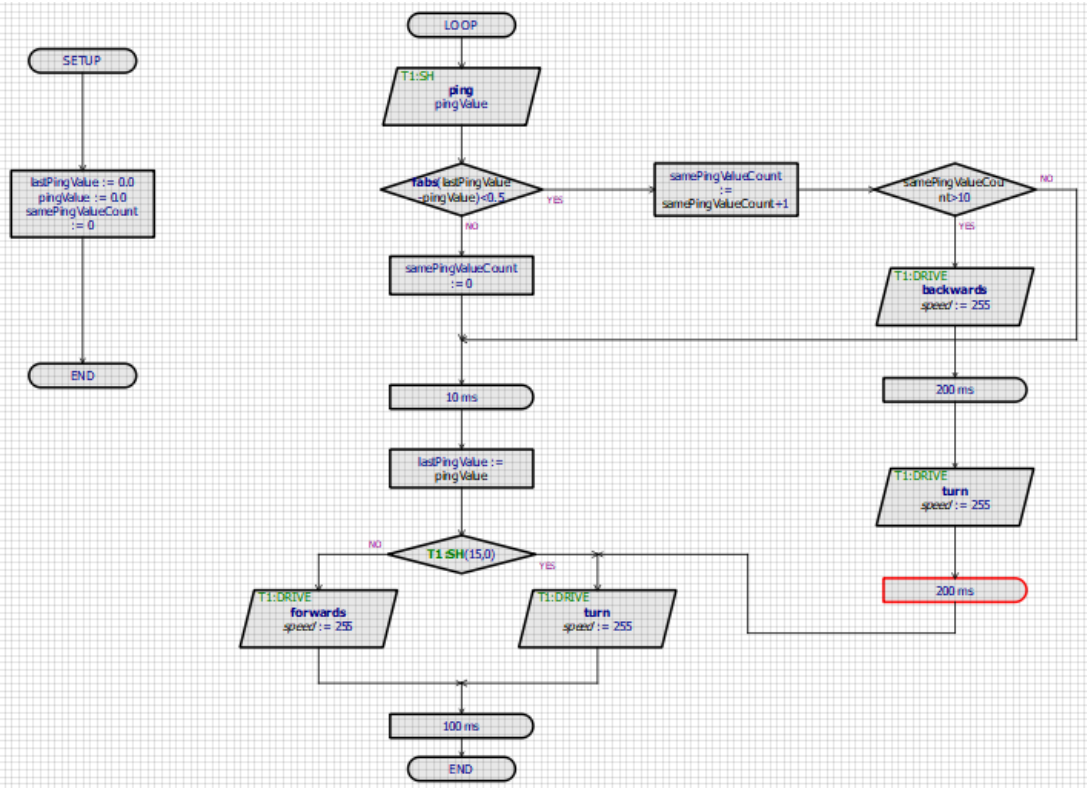


Step 5: Now set Assignments



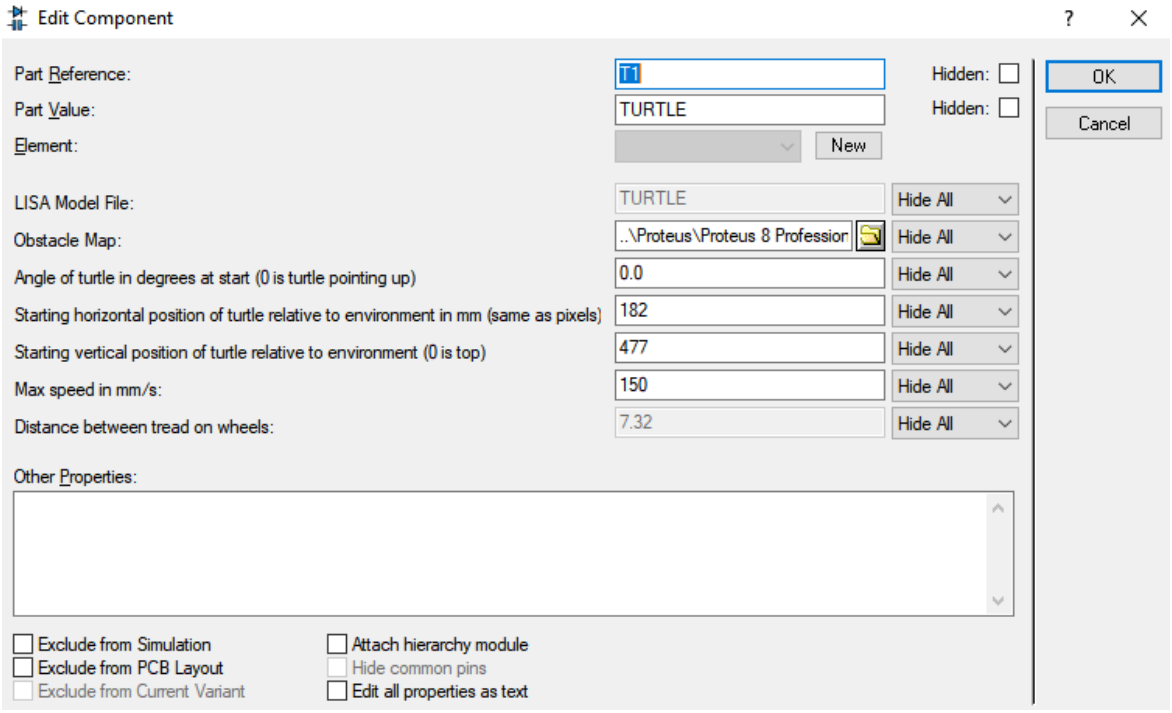

Edit Assignment Block
Assignments
lastPingValue := 0.0 Remove
pingValue := 0.0 Remove
samePingValueCount := 0 Remove
:= Remove
Variables
ingValue FLOAT
Value FLOAT
ePingValueCount INTEGER
New
Delete
Edit
Functions
General Functions
toInt
toFloat
toString
abs
min
max
random
eof
OK
Cancel

Flowchart


SETUP
lastPingValue := 0.0
pingValue := 0.0
samePingValueCount := 0
END
LOOP
T1:SH
ping
pingValue
fabs(lastPingValue -pingValue)<0.5
YES
NO
samePingValueCount := samePingValueCount+1
samePingValueCount>10
NO
YES
samePingValueCount := 0
T1:DRIVE
backwards
speed := 255
10 ms
200 ms
lastPingValue := pingValue
T1:DRIVE
turn
speed := 255
T1:SH(15,0)
NO
YES
200 ms
T1:DRIVE
forwards
speed := 255
T1:DRIVE
turn
speed := 255
100 ms
END

Step 8: Double click on Turtle

- Add Obstacle Map
- Set Angle of turtle in degree at start is 0.0
- Set Starting horizontal position of turtle relative to environment is 182
- Set Starting vertical position of turtle relative to environment is 477
- Set Max Speed to 150
- and then click on Ok



Step 9: Click on Run Simulation Button