Project Title:

“Control and Navigation of a Real Mobile Robot” with Camera Vision and Image Processing

BY: FATEMEH CHANGIZIAN, SEPIDEH NASROLLAHI

Project Description:

The project involves robot navigation to come up with control policies that can control it to efficiently move around the environment in a goal-directed manner while avoiding collisions. In my work, the camera observes robot moves at the top of the environment by tracking a sequence of views and actively interact with the robot for scene understanding. The software should estimate the location of the START POINT and according to the final location (the robot's destination-END POINT) it should direct the robot to the destination. The START POINT and END POINT given as input to the software.

Project Requirements:

- Hardware part:

1. Arduino UNO as The Main Processor
2. L298N H-bridge Dual Motor Controller Module (Motor Driver)
3. 4 Wheeled Smart Car Robot Chassis
4. LM35 ±0.5°C Temperature Sensor (Used for Motor Driver)
5. NRF24L01 Wireless Transceiver Module

- Software part:

1. LabVIEW
2. NI's Vision Assistant

Project Features:

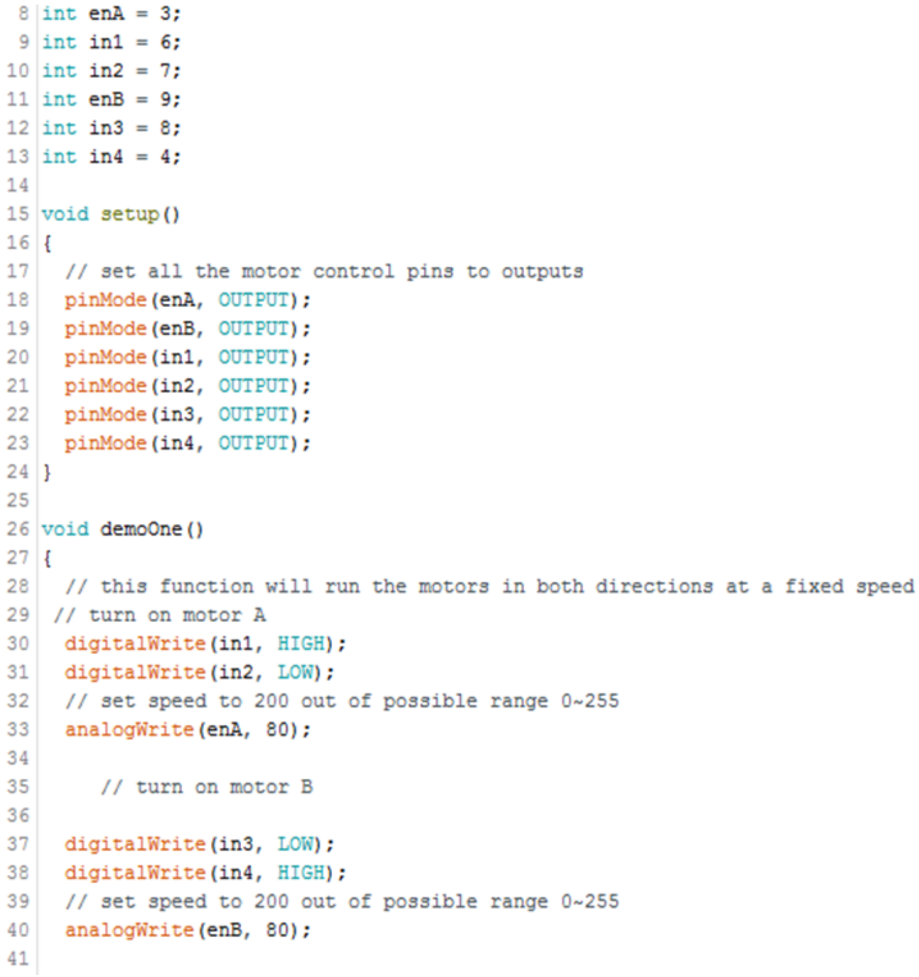
1. The ability to enter two different coordinates as destination point into the software (the robot must reach this given points at the end). The coordinates must be within the

range of the image dimensions that is taken from the camera; these dimensions must be automatically detected by the software and display to the user.

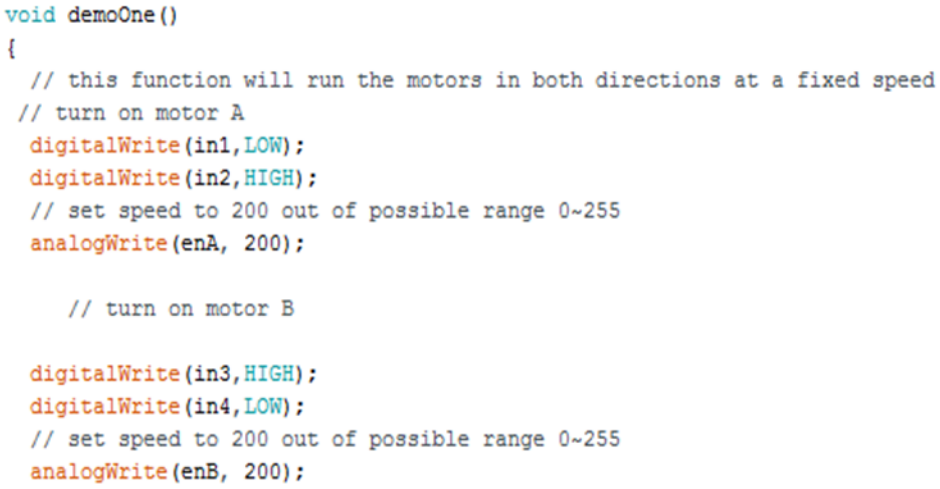
2. Ability to notify the user if the robot arrives at the target point. The announcement is displayed as a massage box. If the user confirms, the next steps will continue.
3. Ability to show the robot's motion path in the software - Ability to activate or deactivate for this option.
4. Ability to recognize the number of obstacles and identify them in the path of the robot - the possibility of activation and deactivation for this case.
5. A warning to the user if the robot is not detected or the robot is out of range of the image dimensions.
6. After the user has installed the software, it waits 1 minute to detect the robot and if the robot detects, the process will continue and if not, the error message will display - Activate and deactivate for this item is possible.
7. Ability to display the online coordinates of the robot.
8. Ability to calculate the length of the motion path and the average speed of the robot and display it to the user.
9. The ability to display online battery charge and motor driver temperature of the robot.
10. Ability to record the coordinates of the motion route of the robot as an excel file at the end (saving confirmation by the user) - Recording coordinates will be made at certain times given by the user, for example every 100 milliseconds. The stored information (excel file) should include time, line coordinates X, Y, angles, battery charge and temperature of motor driver.
11. Place the file location by default.

Programming part:

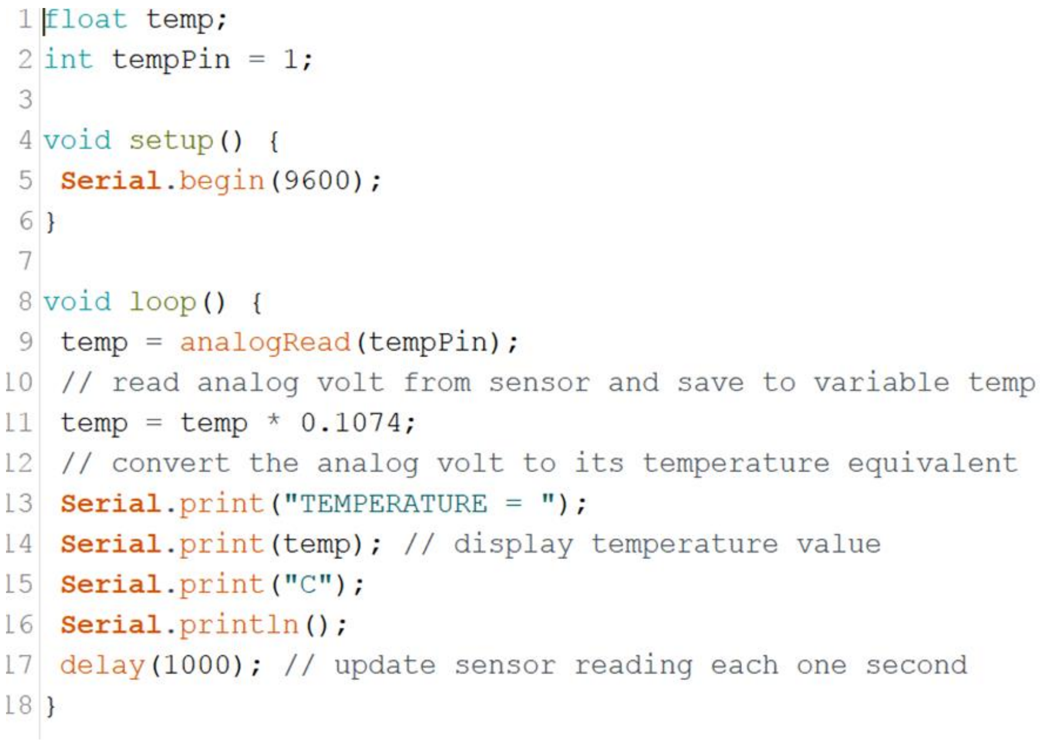
Here is the code for hardware part:



```
int enA = 3;
int in1 = 6;
int in2 = 7;
int enB = 9;
int in3 = 8;
int in4 = 4;

void setup()
{
  // set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void demoOne()
{
  // this function will run the motors in both directions at a fixed speed
 // turn on motor A
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enA, 80);

     // turn on motor B

  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(enB, 80);

```



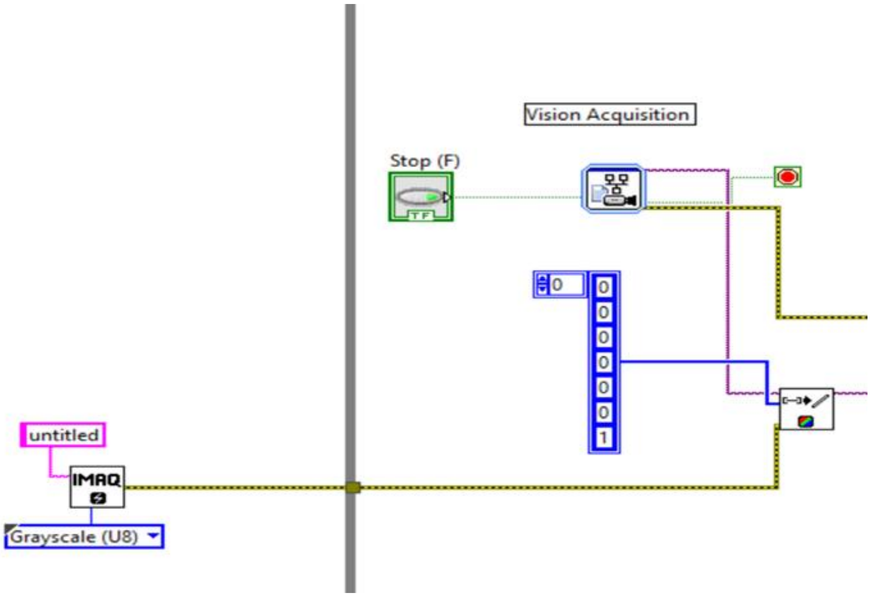
```
void demoOne()
{
  // this function will run the motors in both directions at a fixed speed
 // turn on motor A
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(enA, 200);

     // turn on motor B

  digitalWrite(in3,HIGH);
  digitalWrite(in4,LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enB, 200);
```

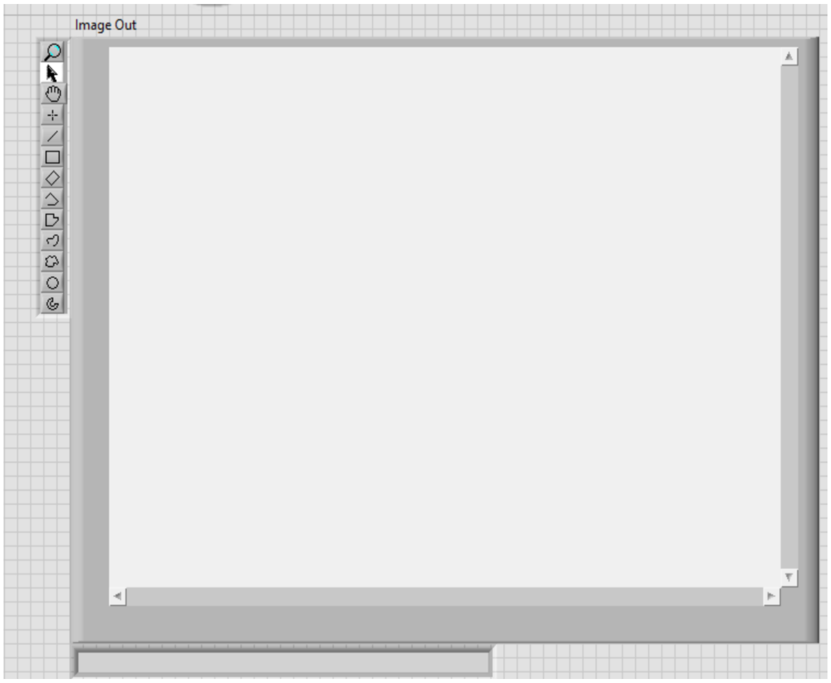


```
float temp;
int tempPin = 1;

void setup() {
 Serial.begin(9600);
}

void loop() {
 temp = analogRead(tempPin);
 // read analog volt from sensor and save to variable temp
 temp = temp * 0.1074;
 // convert the analog volt to its temperature equivalent
 Serial.print("TEMPERATURE = ");
 Serial.print(temp); // display temperature value
 Serial.print("C");
 Serial.println();
 delay(1000); // update sensor reading each one second
}
```

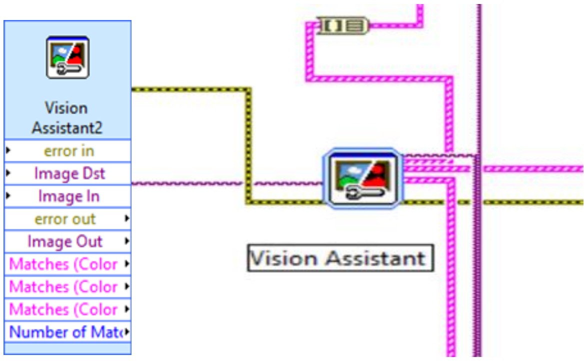
Here is the code for software part (LabVIEW):

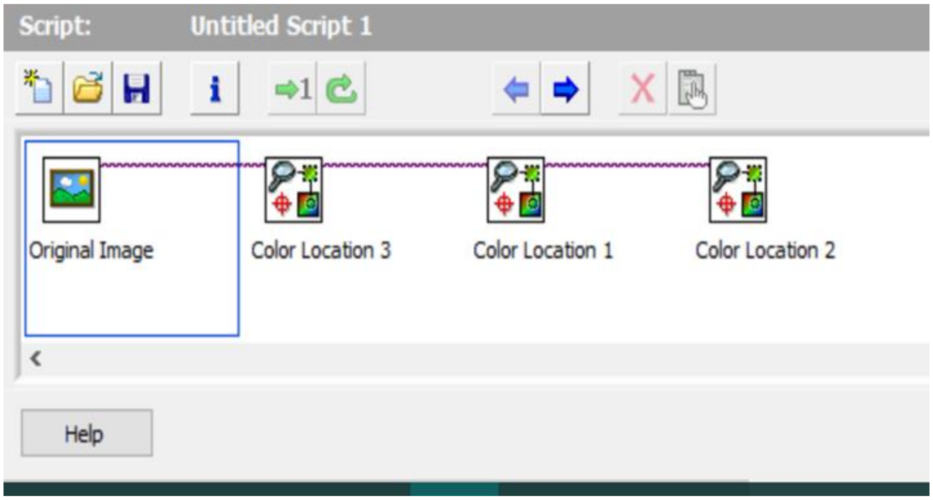
Loading the camera:



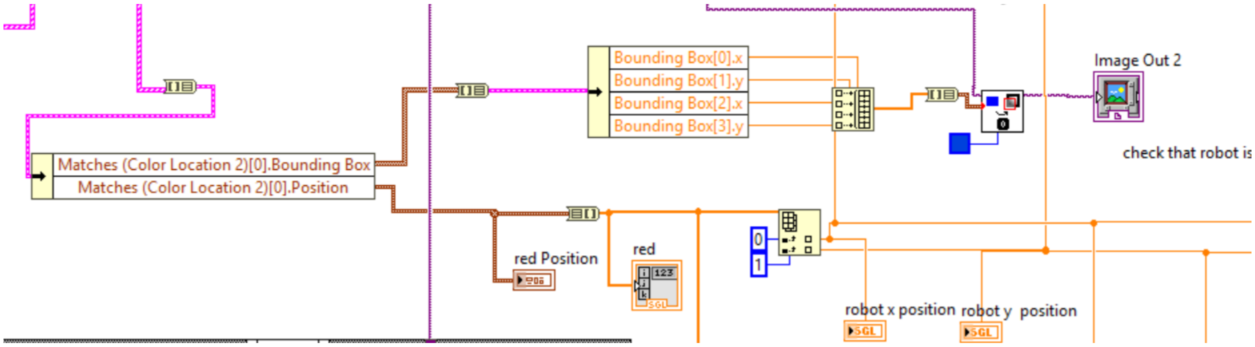


Vision assistant and detecting color location:

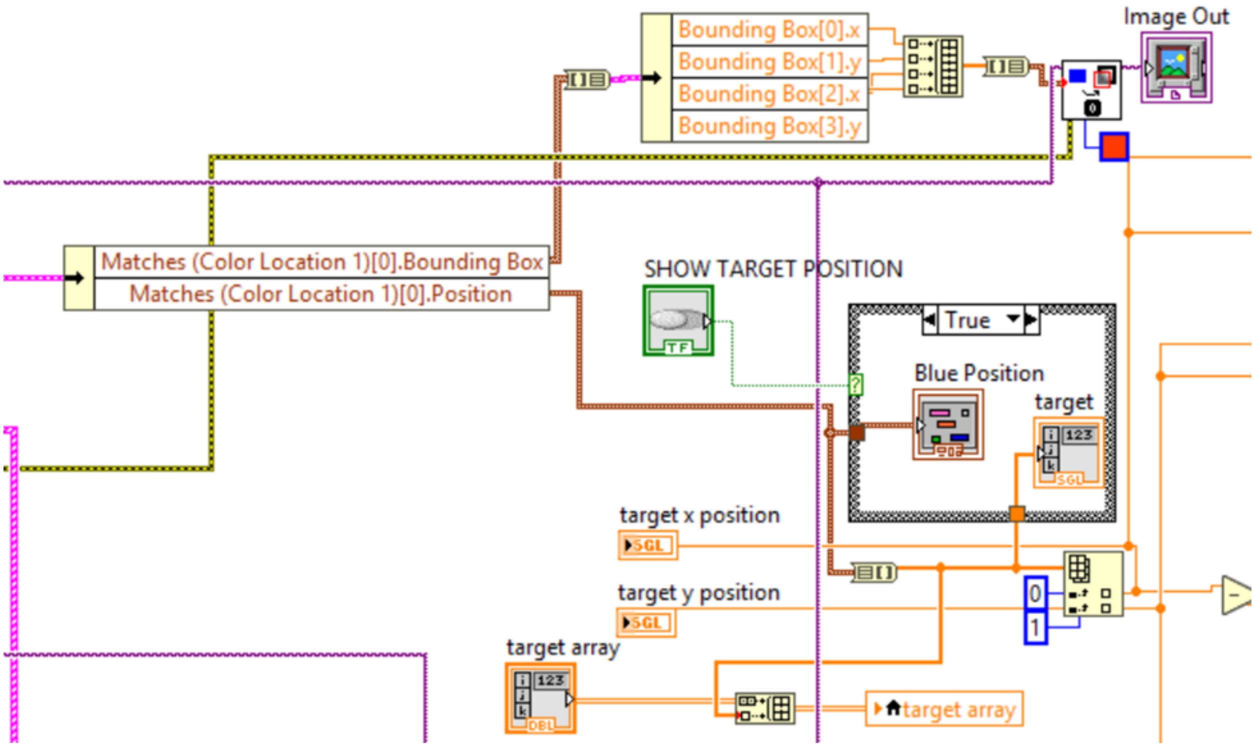




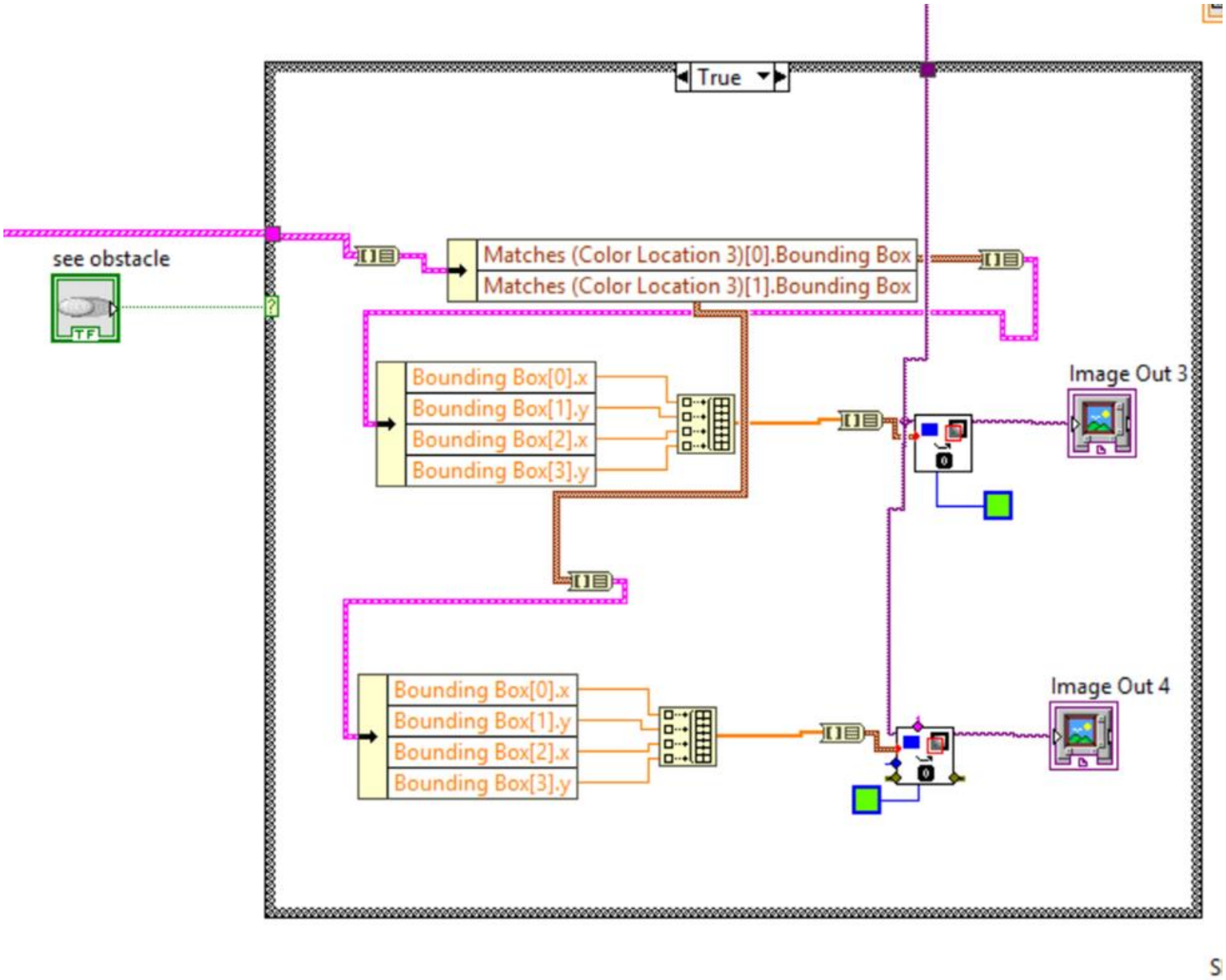
Display the online coordinates of the robot:



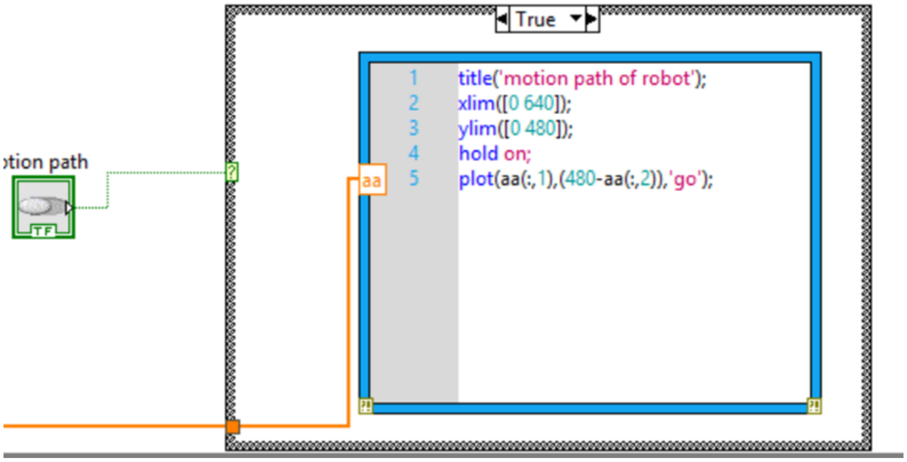
Show target position:



Detect obstacles with an optional switch:

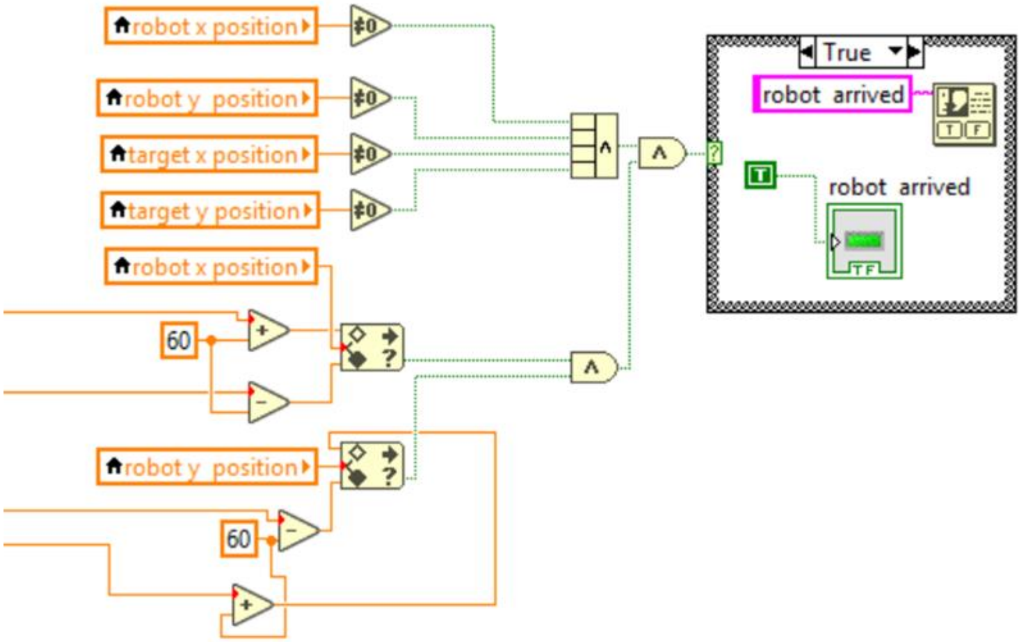


Show the robot's motion path by mathscript tool box:

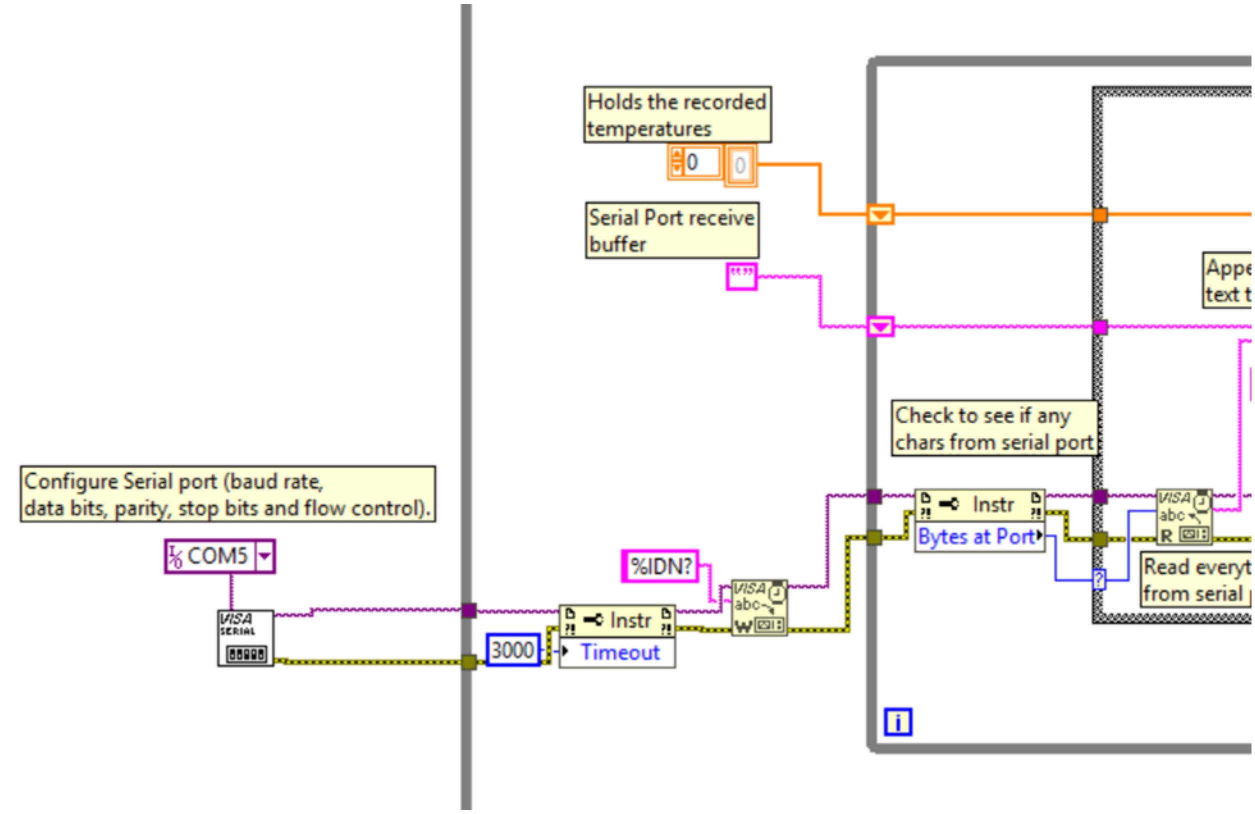


```
title('motion path of robot');
xlim([0 640]);
ylim([0 480]);
hold on;
plot(aa(:,1),(480-aa(:,2)),'go');
```

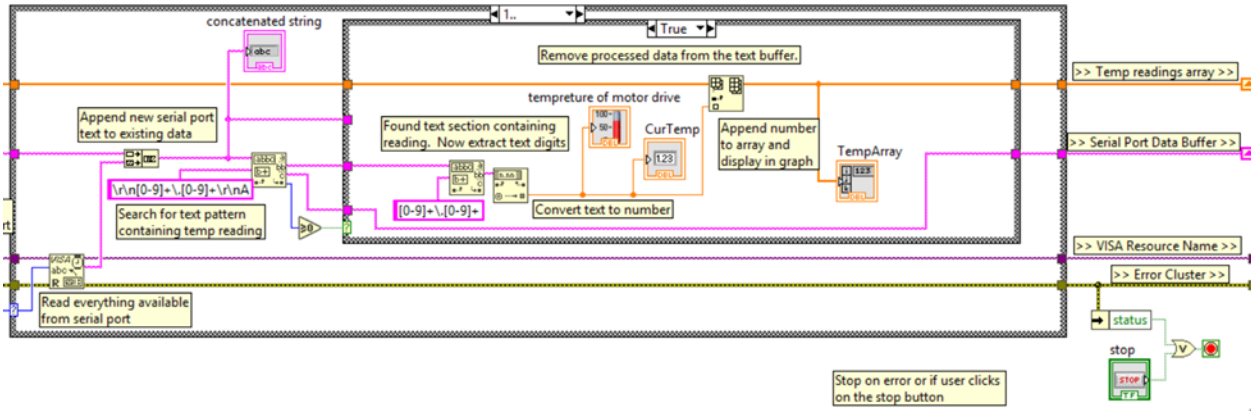
The “arrival” message will be announced to the user:



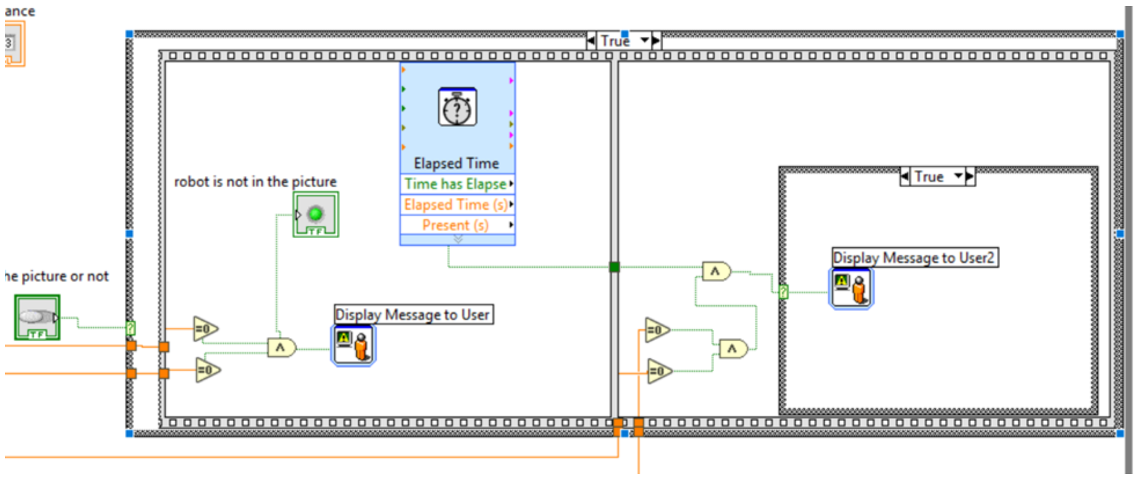
Read serial port information which is temperature of the robot’s motor driver:


Holds the recorded temperatures
0
0
Serial Port receive buffer
""
Appe
text t
Check to see if any chars from serial port
Configure Serial port (baud rate, data bits, parity, stop bits and flow control).
COM5
VISA SERIAL
%IDN?
3000
Instr
Timeout
VISA abc W
Instr
Bytes at Port
VISA abc R
Read everyt
from serial
i

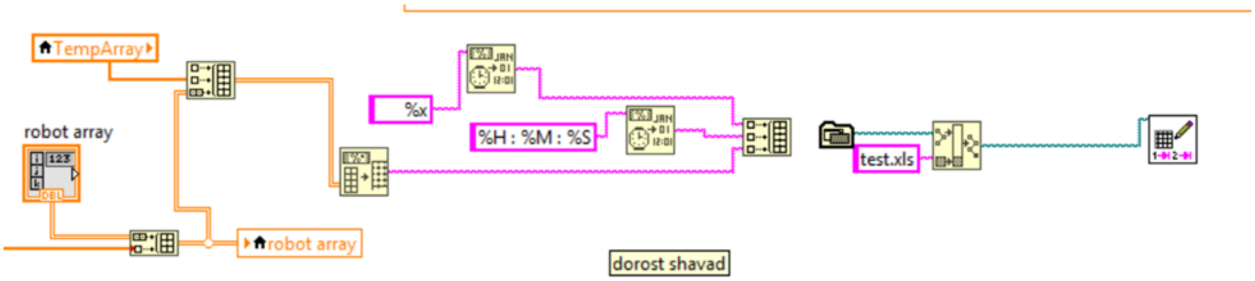
Display current temperature and an array of all measured temperatures:


concatenated string
1..
True
Remove processed data from the text buffer.
tempreture of motor drive
Append new serial port text to existing data
Found text section containing reading. Now extract text digits
CurTemp
Append number to array and display in graph
TempArray
\r\n[0-9]+\.[0-9]+\r\nA
Search for text pattern containing temp reading
[0-9]+\.[0-9]+
Convert text to number
>> Temp readings array >>
>> Serial Port Data Buffer >>
>> VISA Resource Name >>
>> Error Cluster >>
Read everything available from serial port
status
stop
Stop on error or if user clicks on the stop button

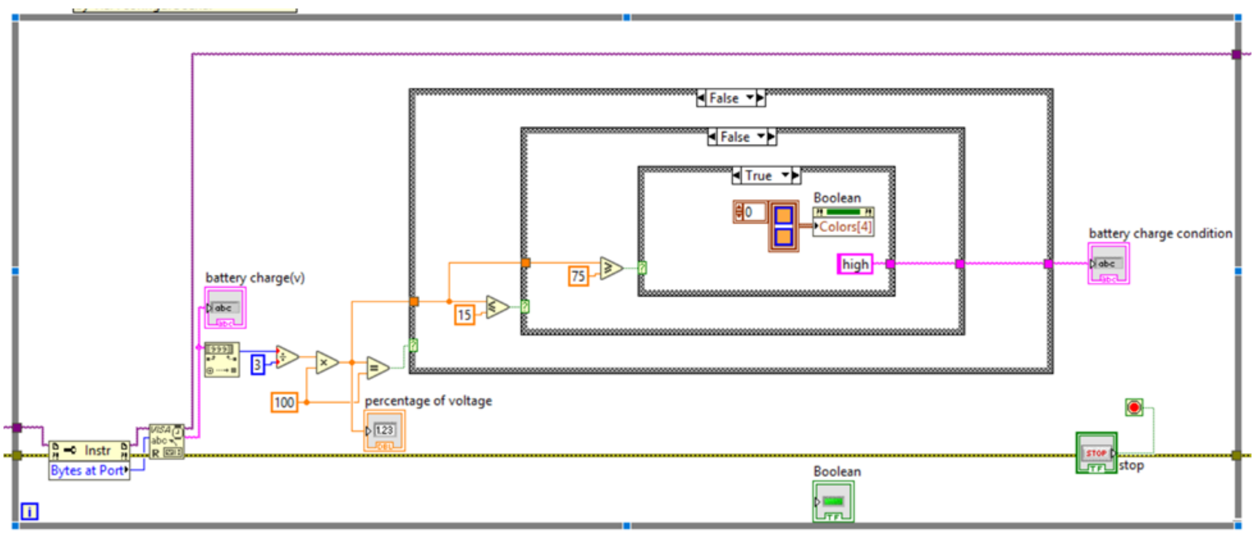
Check if the robot is not detected or the robot is out of range of the image dimensions:



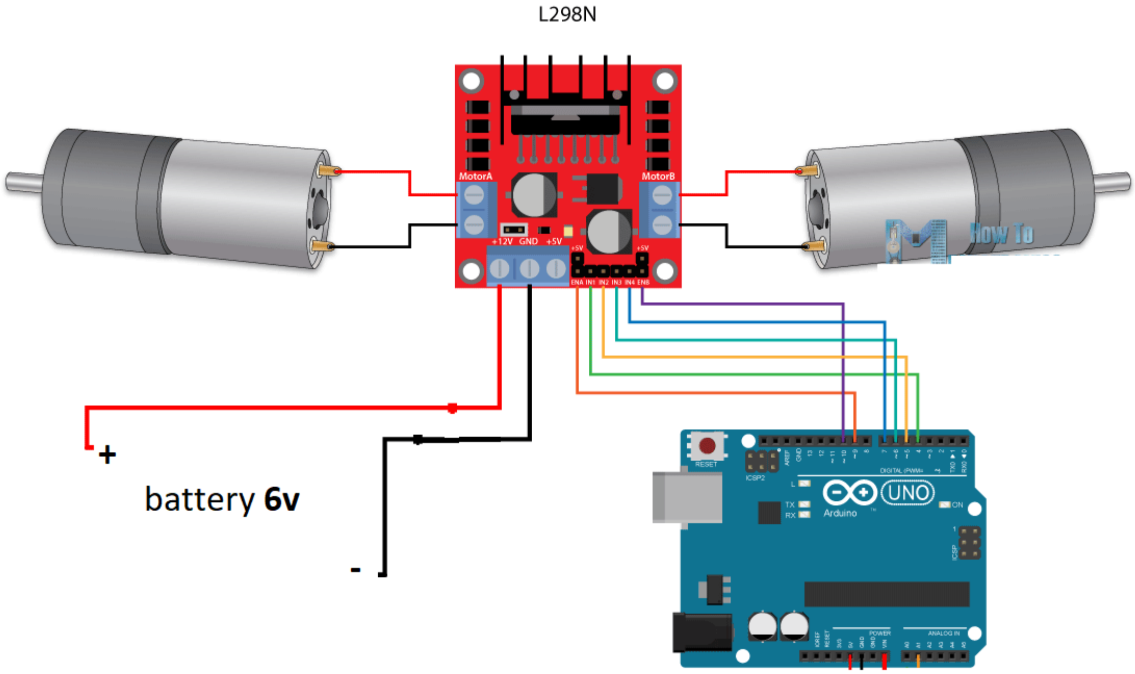
Save robot info in an excel file:

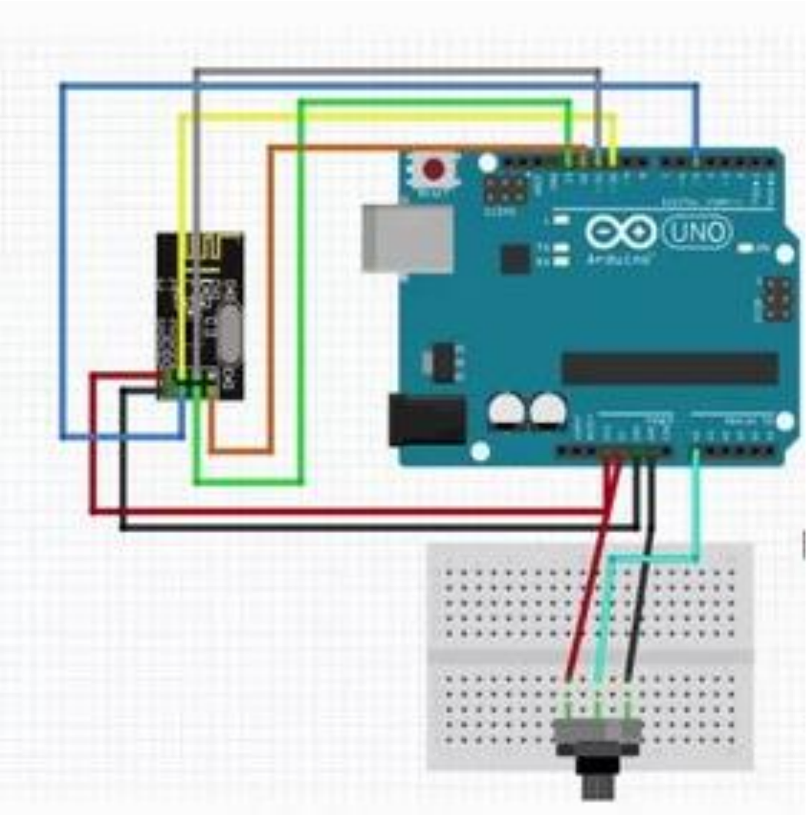


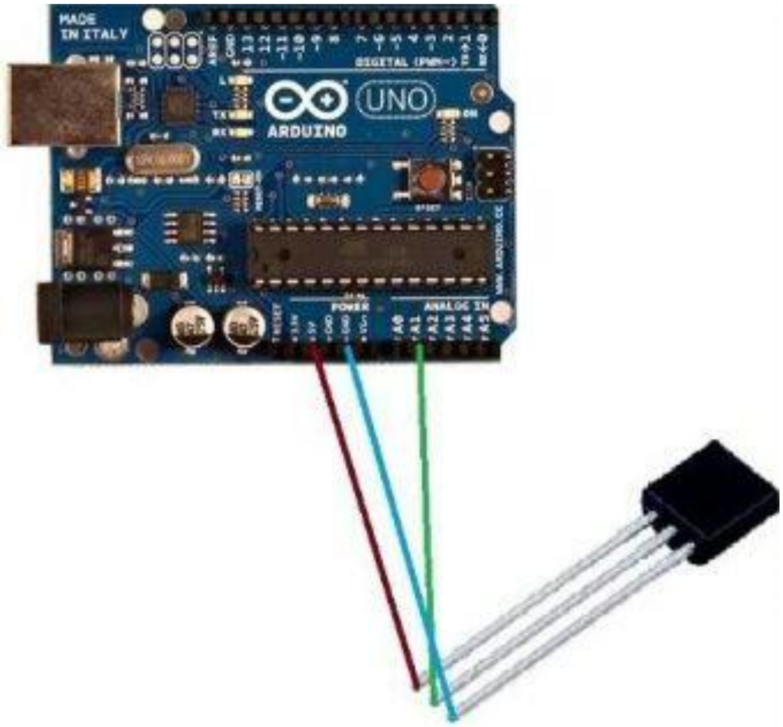
Battery charge condition:



Hardware connections:




UNO
Arduino


MADE IN ITALY
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

Real Picture of Hardware Part: