Bangladesh University of Engineering and Technology

Department of Electrical and Electronic Engineering

EEE 310 PROJECT

Presented by- Group 6

HUMAN FOLLOWER SHOPPING CART

Prepared by

Nazmus Saad Lamim 1806132

MD Fazle Rafi 1806141

Sagira Zaman Eva 1806143

Nafisa Khan 1806160

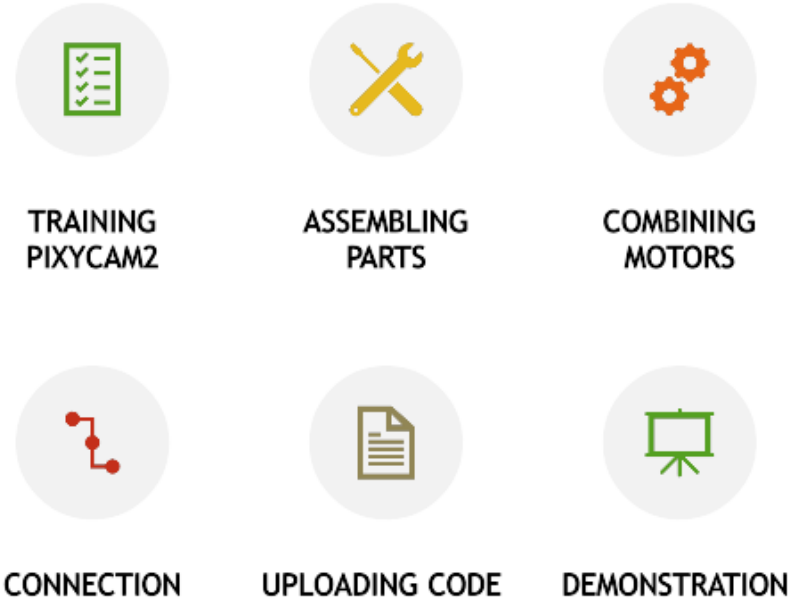
Date: August 29, 2022

Contents

Abstract

This project is about designing a human following shopping cart to ensure a hassle-free shopping experience. Our cart detects a specific object and follows its movement. We have included the option of turning left and right so that it can follow its owner easily. This project is a smaller scale example of our idea within a smaller range.

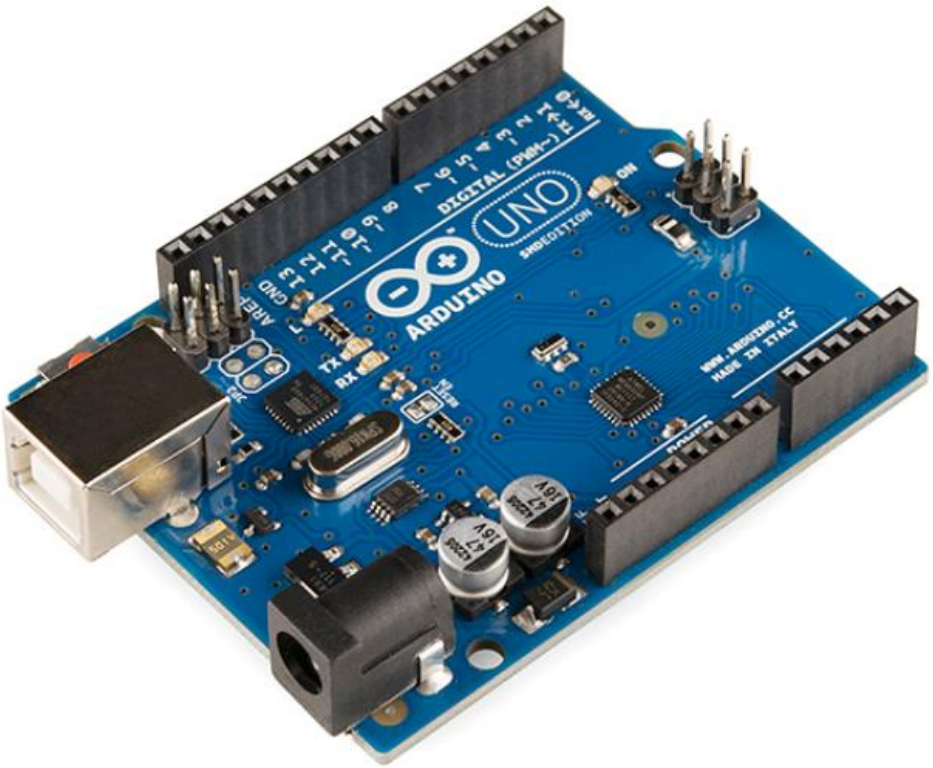
0.1 Simplified Block Diagram



0.2 Equipment

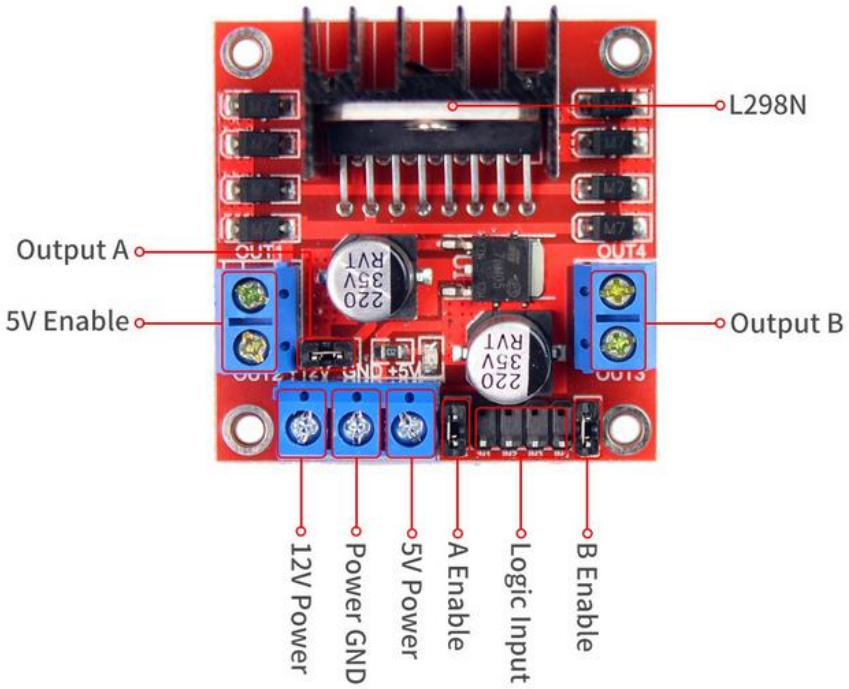
0.2.1 Arduino UNO R3

The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

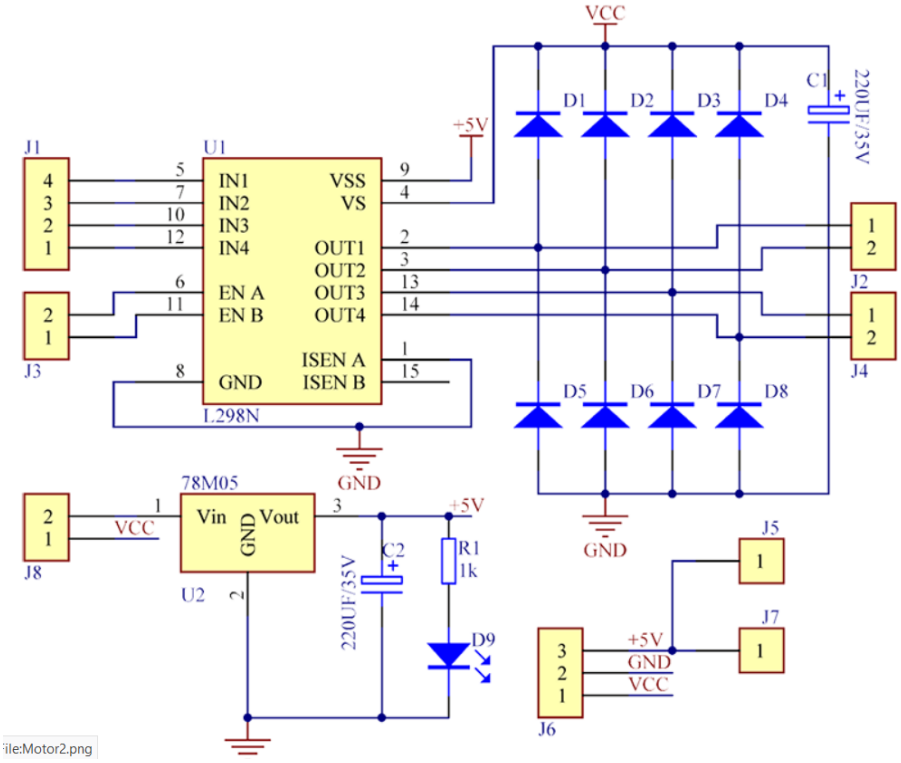


0.2.2 L298 Motor Driver

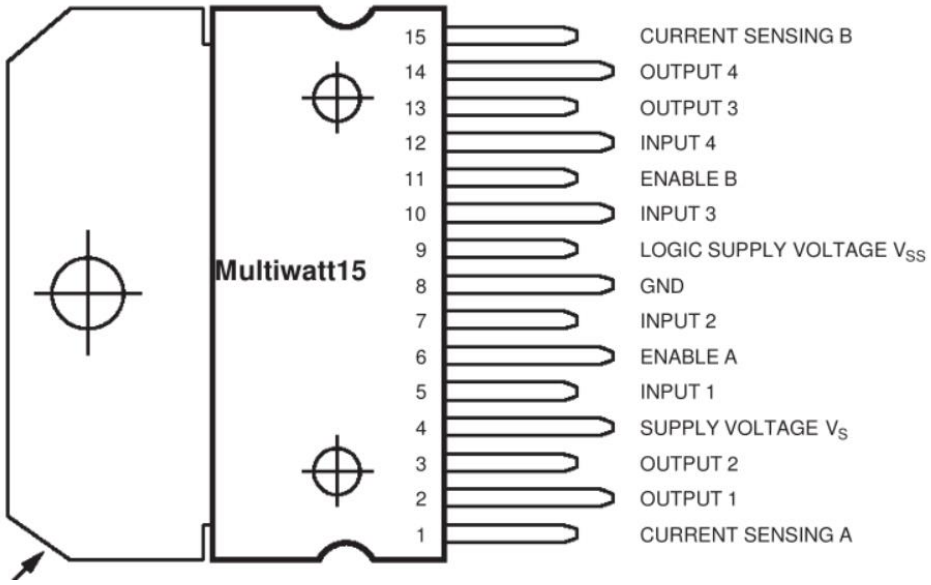
The L298N is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic level and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.



Schematic diagram:



Pin function:



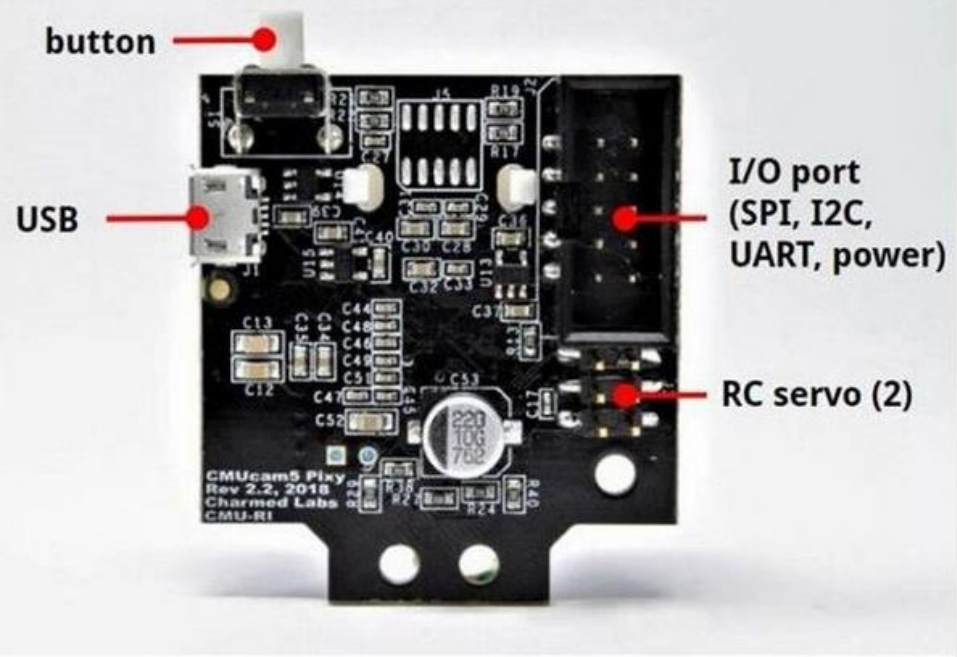
Pin	Name	Description
1; 15;	Sense A; Sense B;	The sense resistor is connected between this pin and ground to control the current of the load.
2; 3;	Out 1; Out 2;	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	Vs	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5; 7;	Input1 ; Input2;	TTL Compatible Inputs of the Bridge A.
6; 11;	Enable A; Enable B;	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	GND	GND
9	Vss	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12;	Input3; Input4;	TTL Compatible Inputs of the Bridge B.
13; 14;	Out 3; Out 4;	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.

0.2.3 Pixycam2 CMUcam5

The PixyCam2 is a fast vision sensor for DIY robotics and similar applications. It is the second version of pixy which is faster, smaller, and more capable than the original, adding line tracking/following algorithms as well as other features.

- Pixy2 detects lines, intersections, and small barcodes, intended for line-following robots
- Image sensor: Aptina MT9M114, 1296×976 resolution with integrated image flow processor
- Lens field-of-view: 60° horizontal, 40° vertical

- Improved frame rate – 60 frames-per-second
- Tracking algorithms have been added to color-based object detection
- Connects to Arduino with included cable. Also works with Raspberry Pi, BeagleBone and similar controllers
- All libraries for Arduino, Raspberry Pi, etc. are provided
- C/C++ and Python are supported
- Communicates via one of several interfaces: SPI, I2C, UART, USB or analog/digital output
- Configuration utility runs on Windows, MacOS and Linux
- Integrated light source



0.2.4 4WD Robot Chassis Kit

General specification:

Motor Operating Voltage (VDC)	3 ~ 6
Motor Shaft Length (mm)	8.5
Motor Shaft Diameter (mm)	5.4mm-(round side) and 3.5mm- (flat side)
Motor Rated Speed After Reduction (RPM)	200
Motor Rated Torque (Kg-Cm)	0.8
Motor Gearbox Shape	Straight
Motor Length (mm)	70
Motor Width (mm)	19
Motor Height (mm)	22
Motor Weight (gm)	28 (each)
Wheel Color	Black (Tyre) Yellow (Rim)
Wheel Internal Diameter (ID)(mm)	51
Wheel Load Capacity (Kg/Wheel)	2.5
Wheel Tyre Grip Material	Rubber
Wheel Body Material	Plastic
Wheel Diameter(mm)	65
Wheel Width (mm)	27
Wheel Weight(gm)	34 (each)

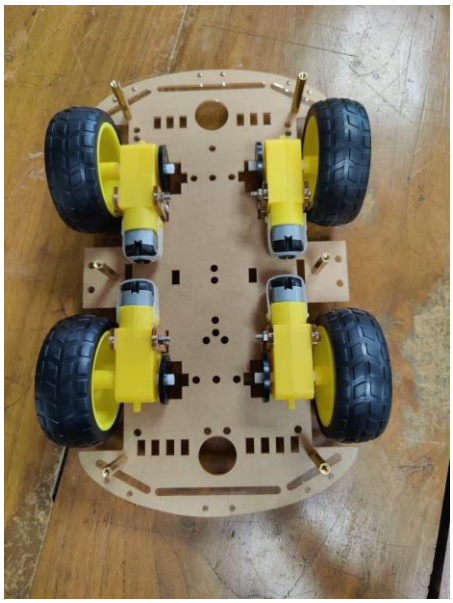
Shipment Weight	0.4 kg
Shipment Dimensions	30 × 18 × 5 cm

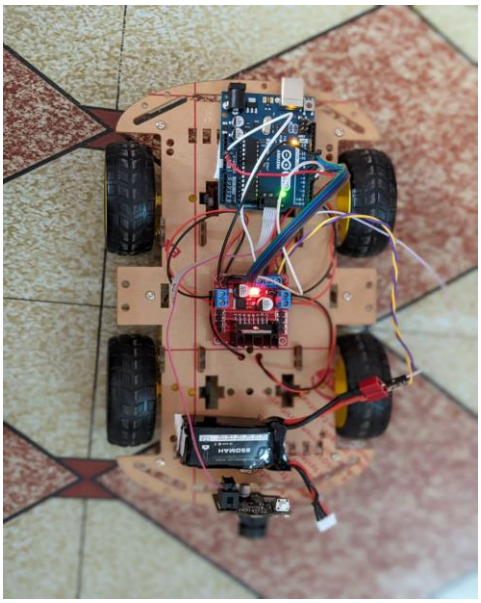
0.2.5 11.1 V LIPO Battery (850 mAh)

0.2.6 Jumper Wires

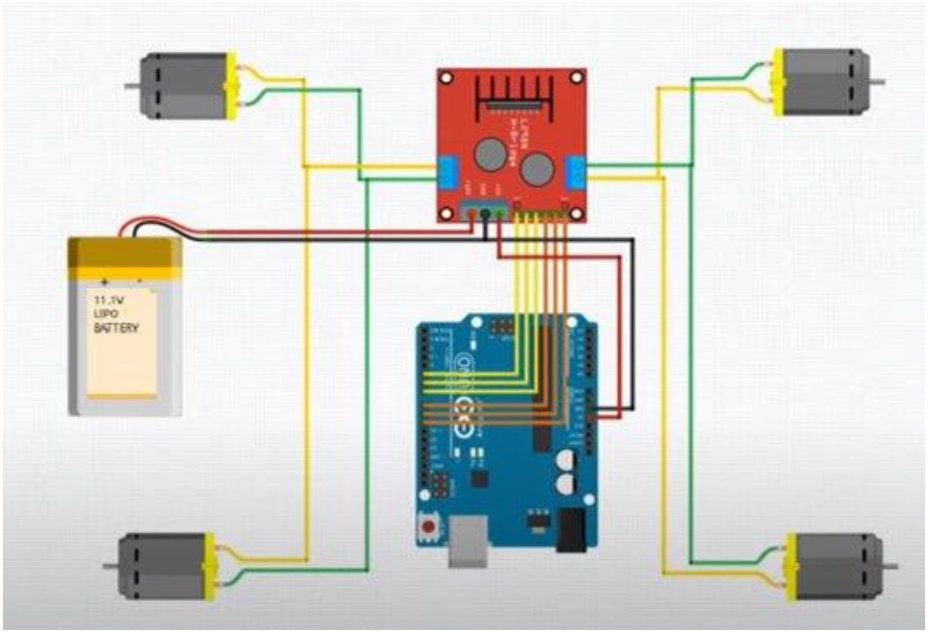
0.3 Assembly and Connections

On the 4WD Chassis kit, 4 motors along with wheels are mounted on one side and on the other side, the Arduino UNO, motor driver, LIPO battery and pixycam2 are mounted.

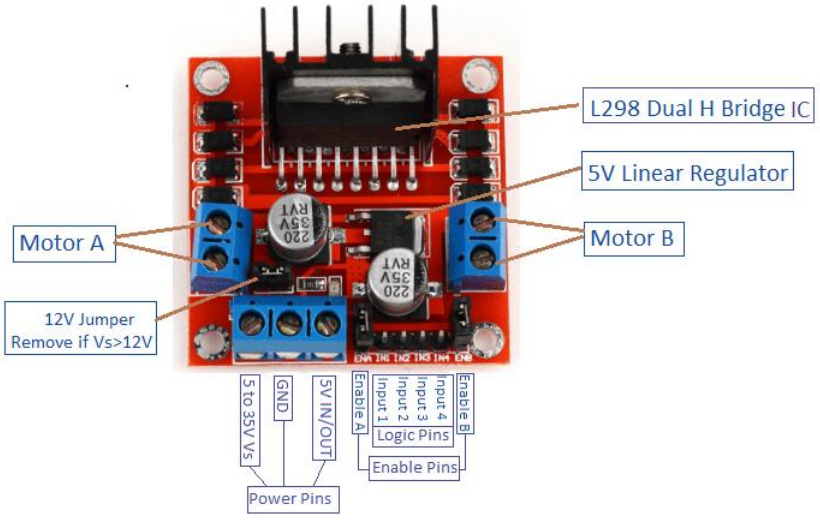




The devices on the Chassis are connected as per the following connection diagram.

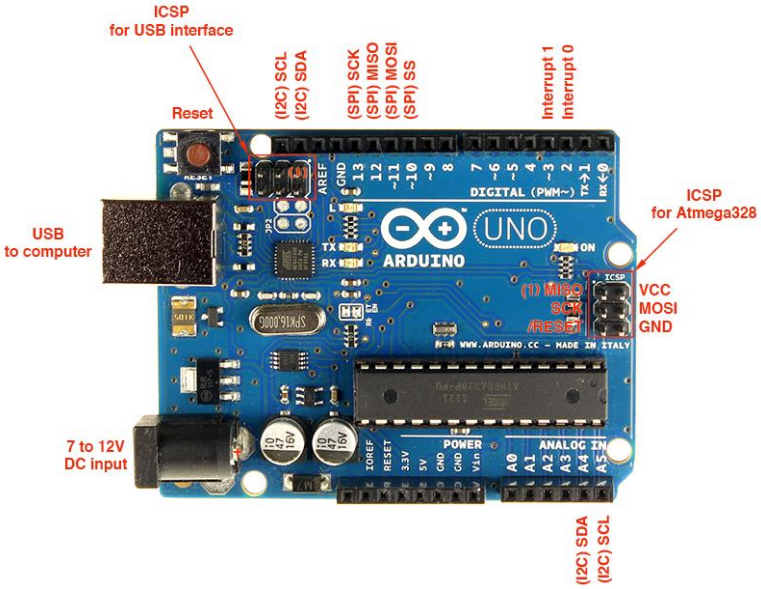


Two of the motors are connected to the Motor A port of the driver whereas the rest two are connected to the Motor B port. The 11.1 V LIPO battery connects to the 5 to 35 V Vs and the GND port. The motors are driven by the driver during operation, either rotating in forward or backward direction. The battery along with the Arduino UNO maintains power supply.

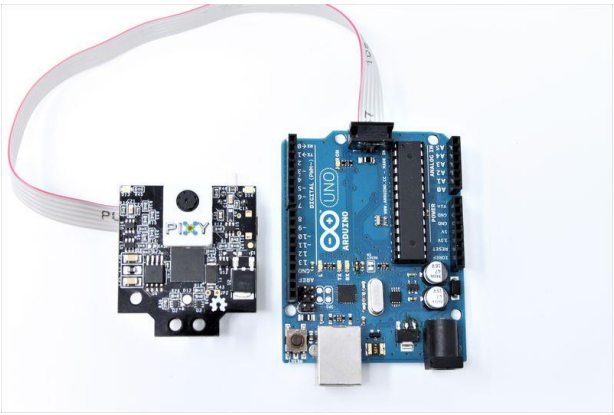


The next connection to the driver comes from the Arduino UNO. Another power pin (the 5 V IN/Out) connects to the 5 V power pin of the Arduino, and the GND

pin connects to the GND power pin of the Arduino. This connection is for the required power supply during operation. There is another set of connections from the Arduino to the Motor driver. The logic pins of the driver connect to the Digital pins of the Arduino, to pins D6, D7, D8 and D9. This brings the proper command from the Arduino to the driver that then drives the motors accordingly. The final two pins connecting the driver to the Arduino are the speed control pins, the Enable A and B pins of the driver connect to the D5 and D10 of the Arduino.



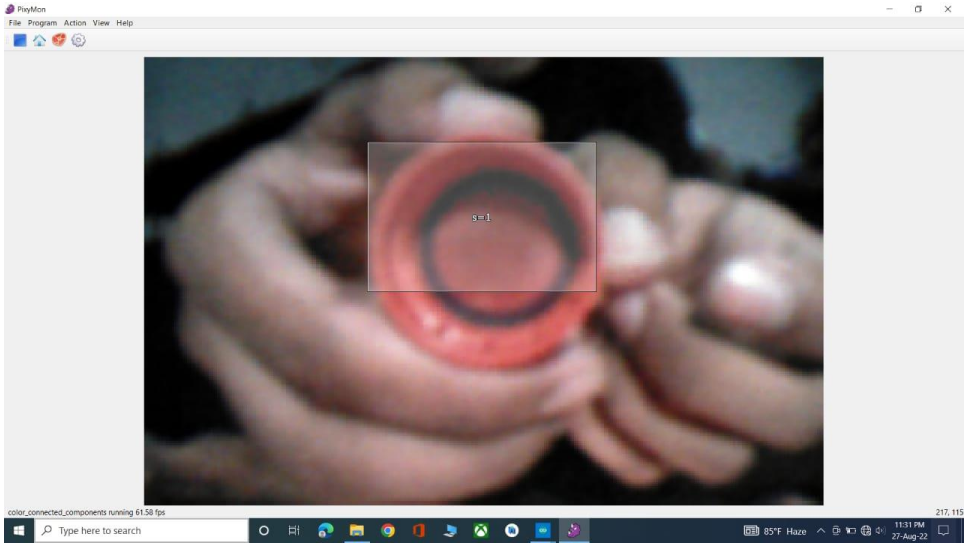
The final connection of the entire setup is between the Arduino and the Pixycam2. The connection is done via a 6-pin Ribbon cable provided with the pixycam. The detection information from the cam is passed to the Arduino by dint of this connection.



0.4 Working Procedure

0.4.1 PixyMon Output

PixyMon is the configuration utility for pixy that we ran on our Windwos computer. Through this, we were able to configure and see what the pixy sees. For our reference object, we were able to see the block pixy has detected.



PixyMon sends four data to the Arduino, the height, width and coordinates (x and y) of the object detected. The Arduino Uses these values to compute the position of the center of the object in pixy's field of view.

0.4.2 Code uploaded to Arduino UNO

```
#include <SPI.h>
#include <Pixy2.h>
Pixy2 pixy;
////////////////////////////////////////////////////////
// ENA IN1 IN2 IN3 IN4 ENB
int myPins[6] = {5, 6, 7, 8, 9, 10};
float deadZone = 0.40;
int baseSpeed = 130;
////////////////////////////////////////////////////////
int cont = 0;
```

```
int signature, x, y, width, height;
float cx, cy, area;
void setup() {
Serial.begin(9600);
Serial.print("Starting...\n");
pixy.init();
 for (int i = 0; i< 6; i++) {
pinMode(myPins[i], OUTPUT);
 }
}
void loop() {
 float turn = pixyCheck();
 if (turn > -deadZone&& turn <deadZone) {
  turn = 0;
 }
 if (turn < 0) {
moveRobot(-60, 70);
 }
 else if (turn > 0 && turn < 500) {
moveRobot(70, -60);
 }
 else if (turn == 500) {
moveRobot(0, 0);
 }
 else {
moveRobot(60, 60);
 }
delay(1);
}
float pixyCheck() {
 static int i = 0;
 int j;
 uint16_t blocks;
 char buf[32];
 // grab blocks!
 blocks = pixy.ccc.getBlocks();
```

```
 if (blocks)
 {
  signature = pixy.ccc.blocks[0].m_signature;
  height = pixy.ccc.blocks[0].m_height;
  width = pixy.ccc.blocks[0].m_width;
  x = pixy.ccc.blocks[0].m_x;
  y = pixy.ccc.blocks[0].m_y;
  cx = (x + (width / 2));
  cy = (y + (height / 2));
  cx = mapfloat(cx, 0, 316, -1, 1);
  cy = mapfloat(cy, 0, 208, 1, -1);
  area = width * height;
  //      Serial.print("sig: ");
  //      Serial.print(signature);
  //      Serial.print(" x:");
  //      Serial.print(x);
  //      Serial.print(" y:");
  //      Serial.print(y);
  //      Serial.print(" width: ");
  //      Serial.print(width);
  //      Serial.print(" height: ");
  //      Serial.print(height);
  //      Serial.print(" cx: ");
  //      Serial.print(cx);
  //      Serial.print(" cy: ");
  //      Serial.println(cy);
 }
 else {
cont += 1;
  if (cont == 100) {
cont = 0;
    cx = 500;
  }
 }
 return cx;
}
```

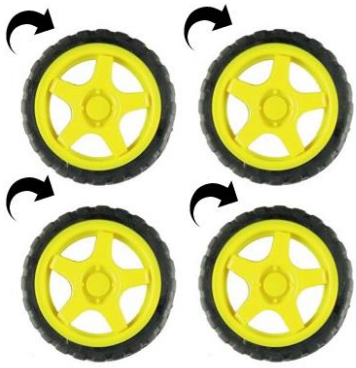
```
float mapfloat(long x, long in_min, long in_max, long out_min, long out_max)
{
  return (float)(x - in_min) * (out_max - out_min) / (float)(in_max - in_min) + out_min;
}
void moveRobot(int leftSpeed, int rightSpeed)
{
  if (leftSpeed> 0) {
digitalWrite(myPins[1], 0);
digitalWrite(myPins[2], 1);
  }
  else {
digitalWrite(myPins[1], 1);
digitalWrite(myPins[2], 0);
  }
  if (rightSpeed> 0) {
digitalWrite(myPins[3], 0);
digitalWrite(myPins[4], 1);
  }
  else {
digitalWrite(myPins[3], 1);
digitalWrite(myPins[4], 0);
  }
  if (leftSpeed == 0 &&rightSpeed == 0)
{ digitalWrite(myPins[1], 0);
digitalWrite(myPins[2], 0);
digitalWrite(myPins[3], 0);
digitalWrite(myPins[4], 0);
  }
analogWrite(myPins[0], abs(leftSpeed));
analogWrite(myPins[5], abs(rightSpeed));
}
```

0.4.3 Arduino Command to Motor Driver

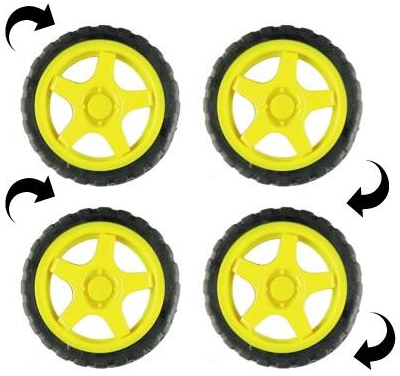
Arduino receives information about the object from the pixy and immediately sends the corresponding command to the motor driver. Arduino decides whether the driver will drive the motors or not, and if it does, how the motors will move. Arduino also checks the speed of the rotation so that the cart does not overspeed.

0.4.4 Orientation of Motor Rotation

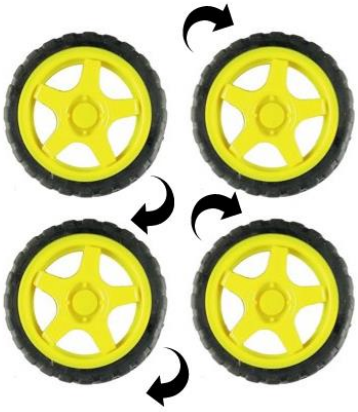
When the object is directly in front of the pixy, all the four wheels attached to the motors rotate in the forward direction, moving the cart straight forward.



If the object is inclined to the right of the pixy's viewing range, the right-side wheels rotate in the backward direction while the left-side wheels still move in the forward direction. This makes the whole cart make a right turn and follow the object properly.



Similarly, for the inclination to the left, the left-side wheels rotate in the backward direction while the right-side wheels change back into the forward direction. The cart thus turns left, and the object is properly followed.



0.5 Operational Output

In our project, we trained the pixycam2 to detect a red Coca Cola bottle cap. The pixy can detect the object well. When the cap is in a desired range in front of the pixy2, it blinks a bright red light, which means the detection is done. The pixy sends the information about the object to Arduino UNO. The Arduino controls the motor driver. The motor driver drives the motors that consequently drive the wheels to move accordingly. We have shown that the cart follows the direction where the object is going even if it makes turns to the right, left or even a 360^{o} rotation.

0.6 Challenges Encountered

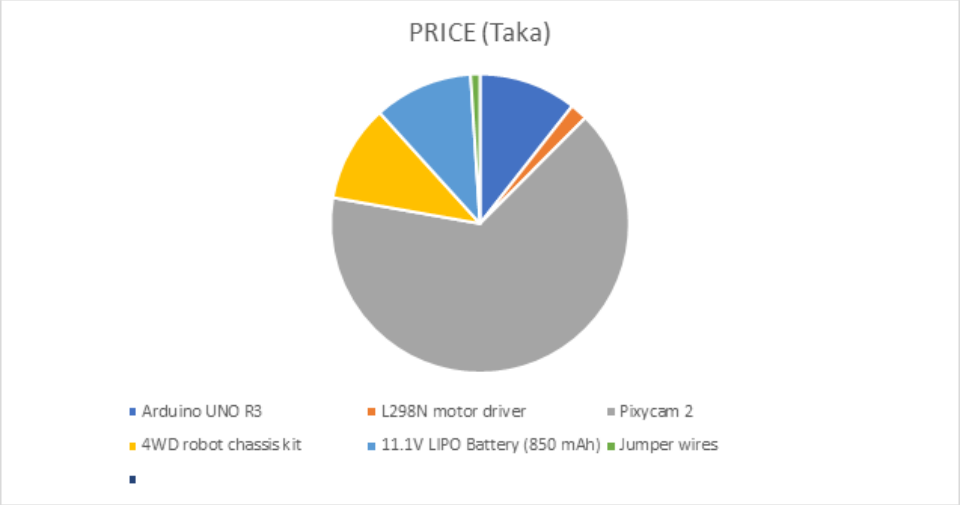
The Arduino board needs more than 7V to power up. The battery that we initially used was only capable of supplying 6V. Therefore, we switched to an 850 mAh LIPO battery. The connection of ENA and ENB pins of the motor driver and the jumper wires was a little bit tricky, however, we could overcome it and were able

to ensure proper speed control. Also, the Pixycam2 needs proper lighting condition to work perfectly, otherwise the detection can be hampered.

0.7 Cost Analysis

Our total expenditure is recorded as:

EQUIPMENT	PRICE (Taka)
Arduino UNO R3	990
L298N motor driver	175
Pixycam 2	6100
4WD robot chassis kit	980
11.1V LIPO Battery (850 mAh)	1000
Jumper wires	95
Total	9340



A substantial portion of our expenditure is for the sensor camera pixy-cam 2. We of course have cheaper alternatives for sensing objects, but they are not compatible with our project. For example:

1. There was a little bit cheaper option for us, a smart machine vision named ‘JeVois’, but unfortunately which is still not available in Bangladesh.
2. Using regular and phone camera: Though some smart cameras in phones can detect some objects, we need OpenCV (an open-source computer vision and machine learning library) expert. It is because we need hardcore

programming to use this option. Consequently, we must discard phone cameras which leaves us with the easiest option that is using pixy camera.

0.8 Applications and Future Prospects

Now we can go two ways with our shopping cart:

i. Upgrading:

With some advanced technologies like GPS and incorporating machine learning our project can be much more efficient. We can use the gesture control device to add new dimension to our device. This is a promising device and can be used in so much extensive application. We are stating a few of them.

ii. Extensive application:

The uses of object following robots are many. They are in this case:

- A prime use can be in childcare. In our country, in working class families or for single mothers who must do all the household deeds alone, this can be a lifesaver.
- We can also build wheelchairs for disabled people. The application will be like childcare.
- Again, we can use this as modification of the working robots in industry or use them separately.

Though we need extensive research both on the work environment and our robot for implementing all that.

0.9 List of Contributors

1806132 – Nazmus Saad Lamim

1806141 – MD. Fazle Rafi

1806143 – Sagira Zaman Eva

1806160 – Nafisa Khan

Reference

GitHub - murtazahassan/Object-Following-Robot