# Personalized Dual Drive Robots (December 2015)

EE 485 - Robotic Systems Final Technical Memorandum

Joshua Wilkins, Samara Reddy, Lu Min Aung

***Abstract*—Robots play a fundamental role in society and are essential in many applications. Research and development of robotic systems will persevere well into the technologically advancing future. This revolutionary change in a developing society manifests the need for radical changes in the educational realm. Creating a rudimentary knowledge base for any structure is the basis on which all knowledge is obtained. This project focuses on the establishment of this fundamental knowledge base through the formation of a differential drive robot. Focusing on fulfilling the necessary requirements including line following, obstacle avoidance, and speed control, multiple robots were constructed. In addition, a personality, behavior, or specific character was needed of the robots. Each robot was assigned a particular personality including bluetooth control and light following in addition to the aforementioned requirements. The original idea involved these individual robots having the ability to join together through the use of electro-permanent magnets, however this concept never came to fruition.**

***Index Terms*— Attiny, Microcontroller, H-Bridge, Encoder, Photocell, Arduino**

## I. The Proposal

This project began with the conceptualization of an idea through the brainstorming process. From its fruition, a project proposal was created and submitted for design review (Outlined below).

This robot will be comprised of several smaller robots that can be connected through electromagnets. These robots will be approximately 3-4cm in length and each will be able to perform a unique designated task, shown below in robot characteristics. Each of the smaller robots will be modular so that new and exciting creations can easily be joined to the larger body. The personalities of each robot will vary, but the required ones include line following, encoder usage, and obstacle avoidance.

Each robot will be able to communicate with each other through a modulated IR signal when connected through the electromagnets. The robot will start off as one joint robot and would then of course have the ability to separate and perform their own unique personality. A central controller may be used, such as an app on a phone with bluetooth connection to one of the robots, to designate the desired operation of the joint robot. The alternative way of implementing this however, is to have a precedence scheme so that when one or more smaller robots are connected, it follows only one behavior instead of the phone app. Note that the behaviors are ordered by precedence level in the flowchart.

Robot Characteristics:

1) Line Following - 1 QTR Sensor
2) Obstacle Avoiding - Sharp IR Sensor
3) Encoder Usage - Hall Effect Sensors
4) H-Bridge Control - SN754410
5) Temperature Sensing - BMP180
6) Light Following - Photo-cell
7) Bluetooth Connection - HC-05 module

With the generic outline and idea behind program flow shown below in figure 1.

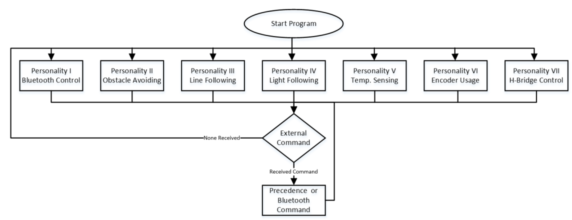


Fig. 1: Generic approach to Program Flow

## II. Theory

During the development of this design, several components were needed, stated above in the robot characteristics. Each of these components are described in detail below.

### A. *QTR Sensor*

The Pololu QTR-1RC reflectance sensor carries a single infrared (IR) LED and phototransistor pair. It has the ability to read the reflectance by withdrawing the externally supplied voltage and timing how long it takes the output voltage to decay due to the integrated phototransistor. Shorter decay time is an indication of greater reflection. This measurement approach has several advantages, especially when multiple units are used including the fact that no ADC is required, it has improved sensitivity over voltage-divider analog output, and parallel reading of multiple sensors is possible with most microcontrollers. The circuitry and component are shown below in figure 2.

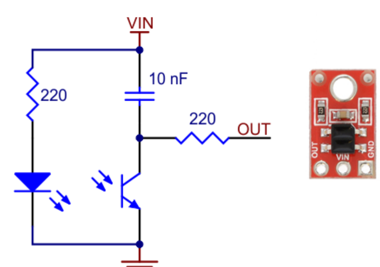


Fig. 2: QTR Sensor Component and Circuitry

### B. IR Sensor

The Sharp distance sensors are ideal for accurate distance measurements. Interfacing to most microcontrollers is straightforward: the single analog output can be connected to an ADC for taking distance measurements, or the output can be connected to a comparator for threshold detection. The detection range of this version is approximately 10 cm to 80 cm. A sharp ir distance sensor is shown below in figure 3.

Fig. 3: A Sharp GP2D12 IR Sensor

The general output of an ir sensor is shown below in figure 4. As can be seen, it is a non-linear relationship between distance and voltage. However, this was not a concern in the project since only relative distances were needed.

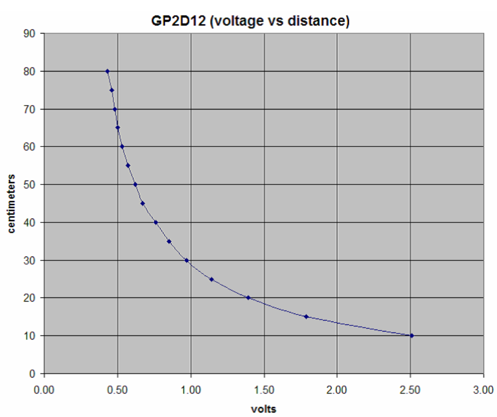


Fig. 4: Measured and calibrated IR distances

### C. Hall Effect Sensor

Hall Effect sensors are transducers that convert magnetic field energy into electrical energy. Common applications include proximity switches, positioning, speed detection, and current sensing. Generally a Hall effect sensor is combined with circuitry that allows the device to act in a digital mode. Implemented correctly, an encoder can be created with the use of one or more hall effect sensors as shown in figure 5 below.

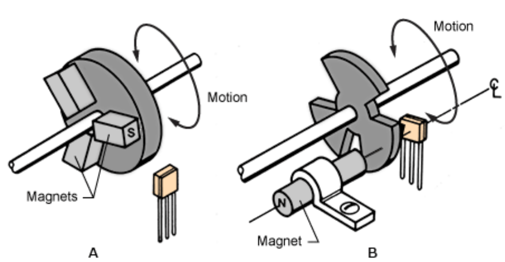


Fig. 5: Encoder made from a Hall Effect Sensor

### D. Encoders

Closed loop control of a robot involves the integration of the robots actuator, sensors and control logic. Closed loop control of a motors velocity and position on a robot can achieved using feedback via encoders. The feedback of the motors current speed can help reduce the error of obtaining a specified motor speed under varying load conditions.

The most commonly used encoders are magnetic or optical and they both operate by using their respective sensors to count changes in regions or ticks on an encoder wheel that spins with the motor shaft. Magnetic encoders were implemented, however optical ones are shown below since they are easier to understand due to the fact they are visible. Standard optical encoders use sector disks with black and white strips, as seen in Figure 6.

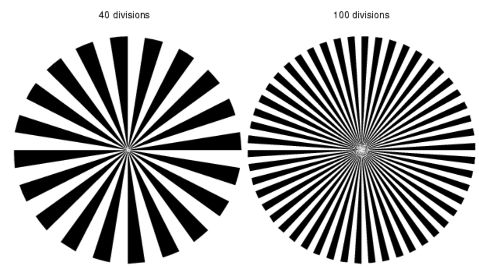


Fig. 6: Relative Sector Wheel

The optical encoder detects the transition between the regions of black and white by using an LED and a photo-diode. The LED constantly generates light and the photo-diode will detect reflected light during white segments and not detect during black segments. The output of the optical encoder is a square pulse representing the light and dark regions of the sector disk.

The position and current speed of the motor shaft can be determined by counting the number of ticks that passed in a given period of time. This type position and velocity calculation is relative based. The current position is relative to the number of ticks it has encountered. The resolution of the calculations for relative encoding is dependent on the number of ticks on the sector wheel.

Using the above mentioned data, the speed of a motor can more accurately be implemented through the use of a feedback system. This is outlined in appendix VI-A.

### E. H-Bridge

An H-bridge is an electronic circuit that enables a voltage to be applied across a load in either direction. These are most often used to control the direction of a DC motor, but have applications in many other kinds of power electronics. The term H bridge is derived from the typical graphical representation of such a circuit. An H bridge is built with four switches as shown in figure 7 below. By configuring which switches are closed, a forward or reverse bias is applied to the load or motor.

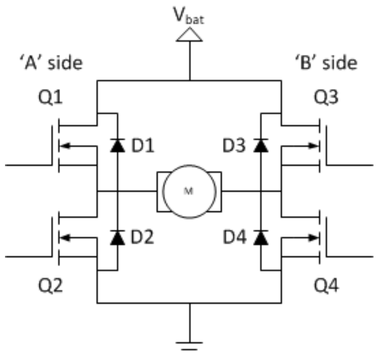


Fig. 7: H-Bridge Configuration

An H-bridge arrangement is generally used to reverse the polarity of a motor, but can also be used to actively brake the motor. This is where the motor comes to a more abrupt stop, as the motor's terminals are shorted as opposed to letting the motor spin freely to a stop as in passive braking. Furthermore, the speed of a motor can be controlled with a pwm signal to the enable pin of an h-bridge.

### F. BMP180

The Bosch BMP180 is an ideal sensor for measuring temperature and pressure accurately. Because pressure changes with altitude, this sensor can also be used as an altimeter! The sensor is soldered onto a PCB with a 3.3V regulator, I2C level shifter and pull-up resistors on the I2C pins. Its component image is shown below in figure 8.

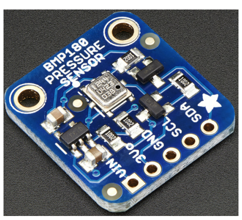

Fig. 8: BMP180 Component Image

### G. PhotoCell

A photocell, photo detector, or a photo-resistor is a light sensor. A photocell changes resistance depending on the amount of light it is exposed to. By implementing the circuit shown below in figure 9, a voltage divider circuit, the resistance across the photocell can be found and converted into a light measurement.

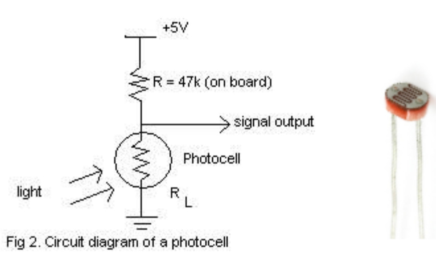


Fig. 9: PhotoCell Component Image and Circuit

### H. BlueTooth Module

The HC-05 component is a bluetooth to uart converter. This means that it takes incoming bluetooth data and converts it into serial data for communication. It is also capable of the reverse; sending data from serial through bluetooth to another device. This device is shown in figure 10.

Fig. 10: Bluetooth HC-05 Component Image

## III. Results

Apart from the extensive use of various components, some information was needed on magnets. This led to the discovery of electro-permanent magnets (EPM), capable of being magnetic one moment and non-magnetic the next based on a current pulse and its direction. The generic setup of such a device is shown below in figure 3.

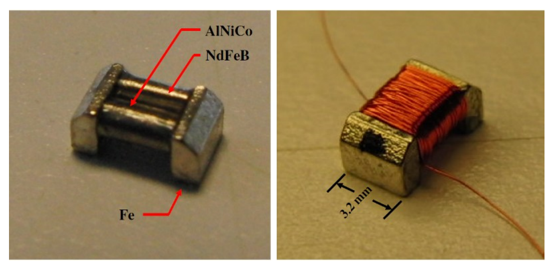


Fig. 11: Generic Setup of EPM

By using two different types of magnets, the polarity of one of the magnets can be reversed to allow the magnetic field to extend or flow circularly within itself. This is shown in figure 12.

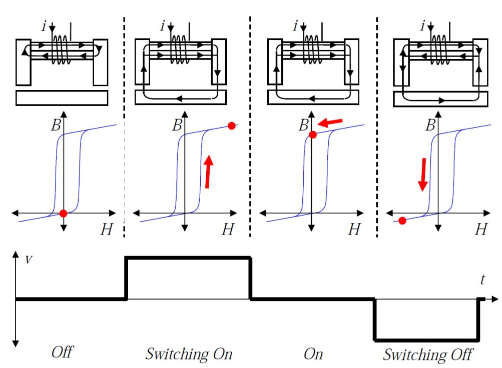


Fig. 12: Functionality of an EPM

Unfortunately, despite our best efforts, we could not succeed in getting an EPM to work. Furthermore, as stated in the design proposal, small motors such as pager motors, shown in figure 13 were going to be used in the design to limit the size of the robots.

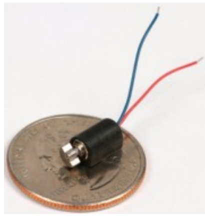
Fig. 13: Pager Motor

The first attempt at the creation of one of these small robots is shown in figure 14 below.

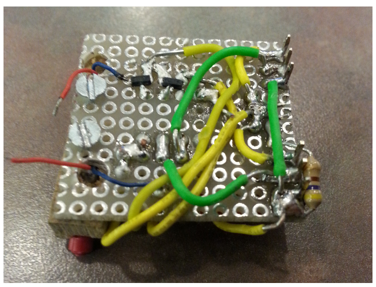
Fig. 14: First Attempt at Small Robot

There are two main reasons this did not work; battery power was not regulated and continuously dropped below the required voltage levels of the circuitry and second, the motors were incapable of moving that much weight.

In addition, the original design was made from attiny85 microcontrollers as shown in figure 15 below.

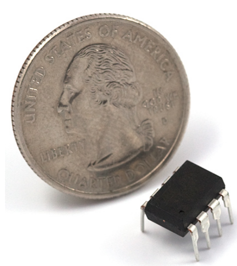
Fig. 15: Attiny85 Component Image

The whole design was then revamped to include the use of bigger motors and microcontroller these are shown below in figures 16 and 17 respectively.


Fig. 16: Incorporated Larger Motors

Fig. 17: Arduino Uno Component Image

The first robot, shown in figure 18 below, was made to line follow and light follow. The determining factor for which behavior it followed was the current temperature in the room. The source code is presented in appendix VI-B.

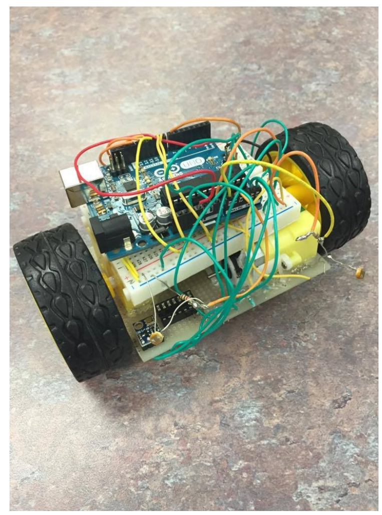
Fig. 18: Robot1: Line, Light, and Temperature

The second robot, shown in figure 19 below, was made to obstacle avoid. The source code is presented in appendix VI-C.

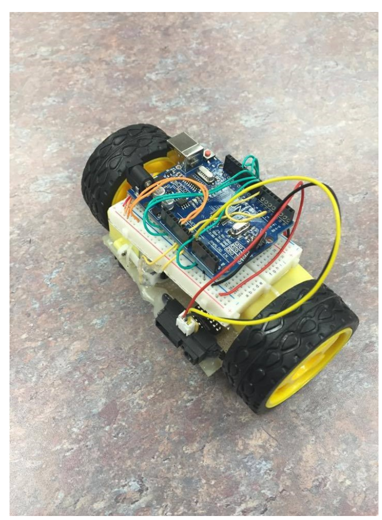
Fig. 19: Robot2: Obstacle Avoidance

The third robot, shown in figure 20 below, was made to follow commands from a bluetooth application on an android device, also shown in figure 20. The source code is presented in appendix VI-C.

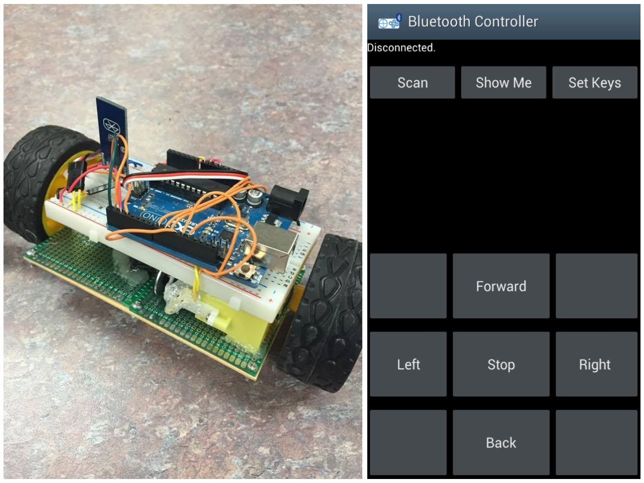


Fig. 20: Robot3: Bluetooth Control

## IV. Discussion

Initially, the intent of this project was to assemble several micro sized robots with dimension of approximately 1 square inch. However, due to unforeseen obstacles, it was realized midway through the development process that of which proved unfavorable for the mechanics to work. For one, the wheels lacked the size and power to set the robot in motion. Soldering transpired to be burdensome, and the desired voltage across the motors was not acquired as expected using a 3V battery. Perhaps the implementation of a voltage regulator could have remedied this problem.

Furthermore, prior to making the larger scaled robots, the idea was to implement these robots with the capability of interconnection, utilizing electro-permanent-magnets. Multiple attempts were made but they were to no avail. Further experimentation led to the testing of electromagnets instead of electro-permanent magnets, but unfortunately they did not work as anticipated. The existing issues, aside for the interconnectivity, were fixed by building the robots on a bigger scale while using a 9V battery instead of the 3V. Future developments may include in-depth exploration of the electromagnets to connect the robots into one.

Despite pitfalls, the knowledge and experience gained from this project are irreplaceable. As Albert Einstein once stated, ”The only source of knowledge is experience. This project epitomizes this statement. Working in theory, compared to field work itself, are two completely different things. One could easily be convinced of Murphys law out in the physical world. Of course there are also neglected or forgotten factors present for the sake of convenience during theoretical calculations.

## V. Conclusion

Despite the numerous accounts of failure, the final result still accomplished nearly all of the designated tasks originally set. As stated, this project was not about new discoveries or accomplishments, but the formation of a fundamental knowledge base to provide some semblance of a structure to which the basis of future knowledge is built. As stated by Henry Ford, ”Failure is the opportunity to begin more intelligently.” This is not necessarily to state the overall failure of the project, but quite the opposite. Not only were the main requirements of the project fulfilled, but some of the personality constraints were as well, even after such great obstacles.

# VI. Arduino Source Code

## A. Feedback Control

```
const int motor1PinA=12,motor1PinB=13,motorEN=11;
volatile double cPos = 0;

int spead, speedDiff;
double diff = 0, lpos = 0, revos, aspeed, eqspeed;
unsigned long time1 = 0, time2 = 0;
bool dir;
unsigned long t1 = 0, t = 0, t2 = 0;

void setup() {
  digitalWrite(2, HIGH);
  pinMode(motorEN, OUTPUT);
  pinMode(motor1PinA, OUTPUT);
  pinMode(motor1PinB, OUTPUT);
  digitalWrite(motor1PinA, LOW);
  digitalWrite(motor1PinB, HIGH);
  attachInterrupt(0,encod, RISING);
  analogWrite(motorEN, 255);
  t1 = millis();
}

void loop() {
  t = t2 - t1;
  if (t > 500) {
    t1 = millis();
    double x = cPos;
    Serial.println(x);
    double revs = x*384;
    double rps = revs/(t * 1000);
    double rpm = rps*60;
    Serial.println(rpm);
    cPos = 0;
  }
  t2 = millis();
}

void encod() {
  noInterrupts();
  cPos++;
  interrupts();
}
```

## B. Robot 1 Implementation

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>

Adafruit_BMP085_Unified bmp =
    Adafruit_BMP085_Unified(10085);
float temp;

#include <QTRSensors.h>
#define NUM_SENSORS   1
#define TIMEOUT       2500
#define EMITTER_PIN   QTR_NO_EMITTER_PIN

QTRSensorsRC qtrrc((unsigned char[]) {7},
    NUM_SENSORS, TIMEOUT, EMITTER_PIN);
unsigned int sensorValues[NUM_SENSORS];
const int m1PB=8,m1PA=9, m2PA=13,m2PB=12;
const int eP1 = 10, eP2 = 11;
int spead;

void setup(void) {
  Serial.begin(9600);
  if(!bmp.begin()) while(1);

  pinMode(m1PA, OUTPUT); pinMode(m2PA, OUTPUT);
  pinMode(m1PB, OUTPUT); pinMode(m2PB, OUTPUT);
  pinMode(eP1, OUTPUT); pinMode(eP2, OUTPUT);
  digitalWrite(eP1, HIGH); digitalWrite(eP2, HIGH);
  digitalWrite(m1PA, LOW); digitalWrite(m1PB, HIGH);
  digitalWrite(m2PA, LOW); digitalWrite(m2PB, HIGH);
}

void loop(void) {
  sensors_event_t event;
  bmp.getEvent(&event);

  if (event.pressure) {
    bmp.getTemperature(&temp);
    temp = temp*1.8 + 32;
  }

  if (temp > 70) lineFollow();
  else lightFollow();
}

void lineFollow() {
  qtrrc.read(sensorValues);
  sensorValues[0] /= 10;
  Serial.println(sensorValues[0]);
  if (sensorValues[0] < 235) {
    analogWrite (enablePin1, 50);
    analogWrite (enablePin2, 175);
  } else {
    analogWrite (enablePin1, 150);
    analogWrite (enablePin2, 25);
  }
}

void lightFollow(){
  int lVal = analogRead(A0);
  int rVal = analogRead(A1);
  int diff = abs(rVal - lVal);

  if (diff <= 100) {       // Go straight if ~ same
    analogWrite(enablePin1,175);
    analogWrite(enablePin2,175);
  }else if (rVal > lVal) {  // Go right if brighter
    analogWrite(enablePin1,200);
    analogWrite(enablePin2,50);
  } else { // Go left if brighter
    analogWrite(enablePin1,50);
    analogWrite(enablePin2,200);
  }
}
```

### C. Robot 2 Implementation

```
#include <SoftwareSerial.h>
SoftwareSerial BT(1,2);

const int motor1PinB = 8, motor1PinA = 9;
const int motor2PinA = 13, motor2PinB = 12;
const int enablePin1 = 10, enablePin2 = 11;
int speadA, speadB;

void setup() {
  BT.begin(9600);

  pinMode(motor1PinA, OUTPUT); pinMode(motor2PinA,
      OUTPUT);
  pinMode(motor1PinB, OUTPUT); pinMode(motor2PinB,
      OUTPUT);
  pinMode(enablePin1, OUTPUT); pinMode(enablePin2,
      OUTPUT);

  digitalWrite(enablePin1, HIGH); digitalWrite(
      enablePin2, HIGH);
  digitalWrite(motor1PinA, HIGH); digitalWrite(
      motor1PinB, LOW);
  digitalWrite(motor2PinA, HIGH); digitalWrite(
      motor2PinB, LOW);
}

void loop() {
  if (BT.available()){
    char c = BT.read();
    if (c == 'F') forward();
    else if (c == 'B') reverse();
    else if (c == 'R') right();
    else if (c == 'L') left();
    else if (c == 'S') stopper();
    else {}
  }
  analogWrite(enablePin1,speadA);
  analogWrite(enablePin2,speadB);
}

void forward() {
  digitalWrite(motor1PinA, HIGH);
  digitalWrite(motor1PinB, LOW);
  digitalWrite(motor2PinA, HIGH);
  digitalWrite(motor2PinB, LOW);
  speadA = 175; speadB = 145;
}

void reverse() {
  digitalWrite(motor1PinA, LOW);
  digitalWrite(motor1PinB, HIGH);
  digitalWrite(motor2PinA, LOW);
  digitalWrite(motor2PinB, HIGH);
  speadA = 215; speadB = 175;
}

void right() {
  forward();
  speadB = 75;
}

void left() {
  forward();
  speadA = 75;
}

void stopper() {
  digitalWrite(motor1PinA, LOW);
  digitalWrite(motor1PinB, LOW);
  digitalWrite(motor2PinA, LOW);
  digitalWrite(motor2PinB, LOW);
  speadA = 0; speadB = 0;
}
```

### D. Robot 3 Implementation

```
const int motor1PinB = 8, motor1PinA = 9;
const int motor2PinA = 13, motor2PinB = 12;
const int enablePin1 = 10, enablePin2 = 11;

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

  pinMode(motor1PinA, OUTPUT); pinMode(motor2PinA,
      OUTPUT);
  pinMode(motor1PinB, OUTPUT); pinMode(motor2PinB,
      OUTPUT);
  pinMode(enablePin1, OUTPUT); pinMode(enablePin2,
      OUTPUT);

  digitalWrite(enablePin1, HIGH); digitalWrite(
      enablePin2, HIGH);
  digitalWrite(motor1PinA, HIGH); digitalWrite(
      motor1PinB, LOW);
  digitalWrite(motor2PinA, HIGH); digitalWrite(
      motor2PinB, LOW);
}

void loop() {
  int sensorValue = analogRead(A0);

  Serial.println(sensorValue);

  if (sensorValue >= 200) {
    digitalWrite(motor1PinA, LOW);
    digitalWrite(motor1PinB, HIGH);
  } else {
    digitalWrite(motor1PinA, HIGH);
    digitalWrite(motor1PinB, LOW);
  }

  analogWrite(enablePin1,175);
  analogWrite(enablePin2,150);
}
```

## REFERENCES


[1] ”WHAT IS ARDUINO?” Arduino. Web. 20 Sept. 2015.

[2] Tiny robot family. (n.d.). Retrieved December 13, 2015, from https://hackaday.io/project/581-tiny-robot-family

[3] Electropermanent Magnets: Programmable Magnets with Zero Static Power Consumption Enable Smallest Modular Robots Yet — Hizook. (n.d.). Retrieved December 13, 2015, from http://www.hizook.com/blog/2010/12/07/electropermanent-magnets-programmable-magnets-zero-static-power-consumption-enable-s