BRAVE-BOUNCING ROBOT WITH AUDIO VISUAL EMULATOR

PROJECT REPORT

OCTOBER 30, 2015

EEE 201 ANALOG INTEGRATED CIRCUITS

SLOT-C2

Faculty:- Prof. MONICA SUBASHINI

1.1 ABOUT THE PROJECT

This Fall Semester 2015-16, we decided to adopt a new approach for doing our Embedded Project in AIC. We wanted to experiment with the concept of online learning. So, we enrolled ourselves in a course offered by UC Berkeley on ‘edx’ platform. The Course name is ‘Electronic Interfaces: Bridging the Physical Worlds’.

The course involved building a wholesome project towards the end of the course that demonstrated all the basic electrical laws in action and the use of different components.

Salient Features of the project.

1. The project was about designing a Bouncing Robot that can respond to the visual and audio stimuli.

2. It employs MSP430 Micro controller that uses BJTs as switch to control DC Motor and manage analog inputs from 2 LDR's and a Microphone.

3. The Robot also equipped with a Piezo Buzzer to tell whether the Robot is awake or not.

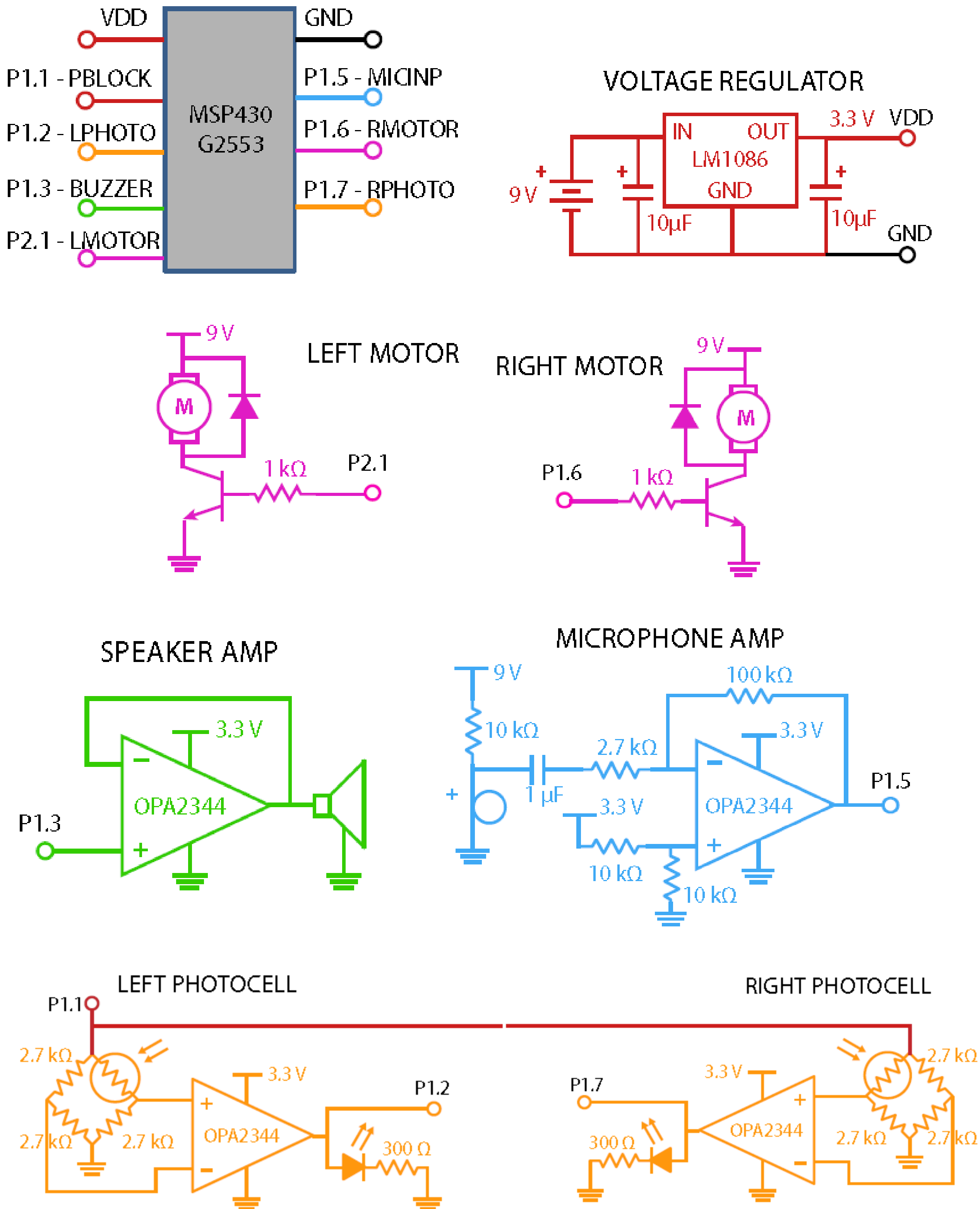
4. We have used basic components like Wheatstone bridge and Op Amps in this project.

5. We have demonstrated the use of Operational Amplifier through a Microphone Amplifier Circuit through a ‘First Order Filter Circuit’ and a Buzzer Amplifier.

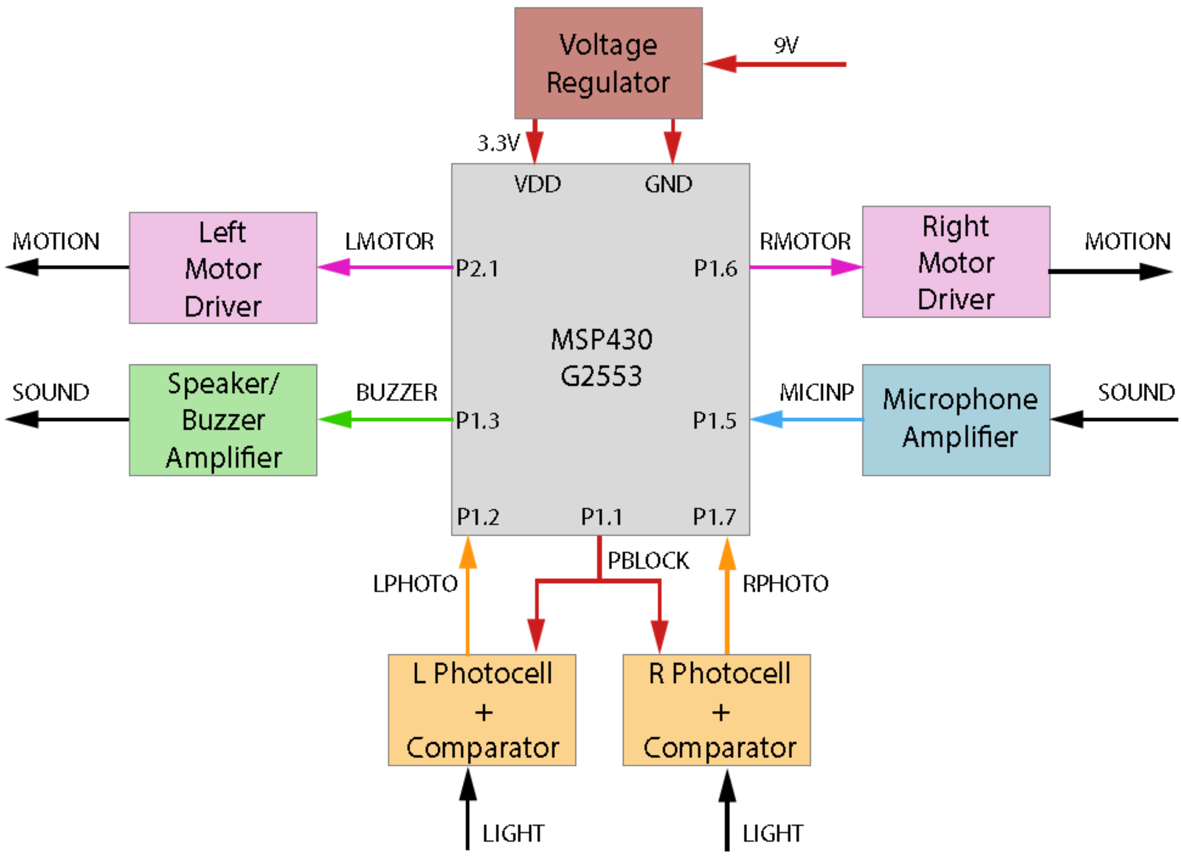
6. We have demonstrated how to use an Analog Sensor like LDR to give Digital output using an Op Amp as Comparator implemented in the Photocell Circuit.

1.2 CIRCUIT DIAGRAM

Our Robot Circuit can divided into multiple sub circuits that are being used to carry out particular operations.



1.3 ROBOT BLOCK DIAGRAM



1.4 COMPONENTS LIST

Electronic Parts

These are the main electronic parts used to build the various robot circuits.

Qty	Part	Manufacturer	Man. Part #
1	Solderless Breadboard	MULTICOMP	MCBB400
3	Red LED 2.1V 10mA 5mm	MULTICOMP	MCL053PT
2	OPA2344 Dual Amp (DIP) –	TEXAS INSTRUMENTS	OPA2344PAG4

	This makes a total of four op amps with rail-to-rail output.		
1	LM1086 3.3V Regulator	TEXAS INSTRUMENTS	LM1086CT-3.3/NOPB
2	STANDARD DIODE, 1A, 50V, DO-41	MULTICOMP	1N4001
2	PN2222 (NPN transistor)	FAIRCHILD SEMICONDUCTOR	PN2222
2	DC Motor, 9100 rpm max speed, 6V DC supply voltage, 20g-cm max torque, size 130	ADAFRUIT INDUSTRIES	711
2	Photocell	EXCELITAS TECH	VT90N1
2	RESISTOR, METAL FILM, 300 OHM, 250mW, 1%	MULTICOMP	MF25 300R
2	RESISTOR, CARBON FILM, 1KOHM, 250mW, 5%	MULTICOMP	MCF 0.25W 1K
3	RESISTOR, CARBON FILM, 10KOHM, 250mW, 5%	MULTICOMP	MCF 0.25W 10K
7	RESISTOR, CARBON FILM, 2.7KOHM, 250mW, 5%	MULTICOMP	MCF 0.25W 2K7
1	RESISTOR, CARBON FILM, 100KOHM, 250mW, 5%	MULTICOMP	MCF 0.25W 100K
1	Piezo Buzzer	MULTICOMP	MCKPT-G1340-3917
1	Electret microphone, 50 to 16000 Hz response, 10V max operating voltage	PRO SIGNAL	ABM-707-RC
2	CAPACITOR ALUM ELEC 10UF, 16V	MULTICOMP	MCGPR16V106M5X11-RH
2	CAPACITOR CERAMIC, 1UF, 25V, Y5V, +80	VISHAY BC COMPONENTS	K105Z20Y5VE5TL2
1	MSP430 LaunchPad	TEXAS INSTRUMENTS	MSP-EXP430G2

Power

The robot runs on 9V DC. There are some options to power your robot.

Battery Power – Requires a 9V battery and a 9V snap connector.
Battery power will allow your robot to move around freely, but you may deplete the charge on a number of batteries before finishing the course!

Qty	Part	Manufacturer	Man. Part #
1	9V Battery		
1	9V snap connector	KEYSTONE	234

1.5 MSP430 /ENERGIA CODE

```
int PBLOCK = P1_1;  // set PBLOCK as P1.1 alias
int BUZZER = P1_3;  // set BUZZER as P1.3 alias
int LMOTOR = P2_1;  // set LMOTOR as P2.1 alias
int RMOTOR = P1_6;  // set RMOTOR as P1.6 alias
int LPHOTO = P1_2;  // set LPHOTO as P1.2 alias
int RPHOTO = P1_7;  // set RPHOTO as P1.7 alias
int MICINP = A5;    // set MICINP as A5   alias

int MPOW   = 255  // set motors to use 100  0% PWM (possible values 0-255)
int MICTHRESH = 600; // set microphone trigger threshold (possible values 0-1023)

void setup()
{
 // set outputs
  pinMode(LMOTOR, OUTPUT);
  pinMode(RMOTOR, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(PBLOCK, OUTPUT);

 // set inputs
  pinMode(LPHOTO, INPUT);
  pinMode(RPHOTO, INPUT);
}

void loop()
{
  int i;
```

```
int val;
int MICTHRESH = 600;
int maxval;

analogWrite(LMOTOR,0);      // turn off both motors
analogWrite(RMOTOR,0);

// beep a bunch of times!
for (i=1; i<5; i++)
{
  beep(BUZZER, 1000, 100*i);
  delay(100*i);
}

// listen to the microphone for ~100 ms
maxval = 0;
for (i=1; i<100; i++)
{
  val = analogRead(MICINP);
  if (val > maxval)
        {
    maxval = val;
  }
        delay(1000);
}

// If the largest voltage detected is above 1.77 V (3.3*550/1024),
// commence the "beep dance" response
if (maxval > MICTHRESH)
{
```

```
  // Make the "siren" noise by alternating 1200 Hz and 800 Hz tones
  for (i=0; i<5; i++)
  {
     beep(BUZZER, 1200, 100);
     beep(BUZZER, 800,  100);
  }
  // Shake motors back and forth rapidly
  for (i=0; i<3; i++)
  {
     analogWrite(RMOTOR, MPOW);
     delay(200);
     analogWrite(RMOTOR, 0);
     analogWrite(LMOTOR, MPOW);
     delay(200);
     analogWrite(LMOTOR, 0);
  }
  // Make a series of tones with increasing frequency from 300-100 Hz
  // then come back down
  for (i=30; i<100; i+=1)
        {
    beep(BUZZER, 10*i, 10);
        }
  for (i=100; i>30; i-=1)
        {
    beep(BUZZER, 10*i, 10);
        }
}
// read the status of photocells and adjust motor output
digitalWrite(PBLOCK, HIGH);     // supply 3.3V to the power rail
delayMicroseconds(20);          // delay briefly to allow comparator outputs to settle
```

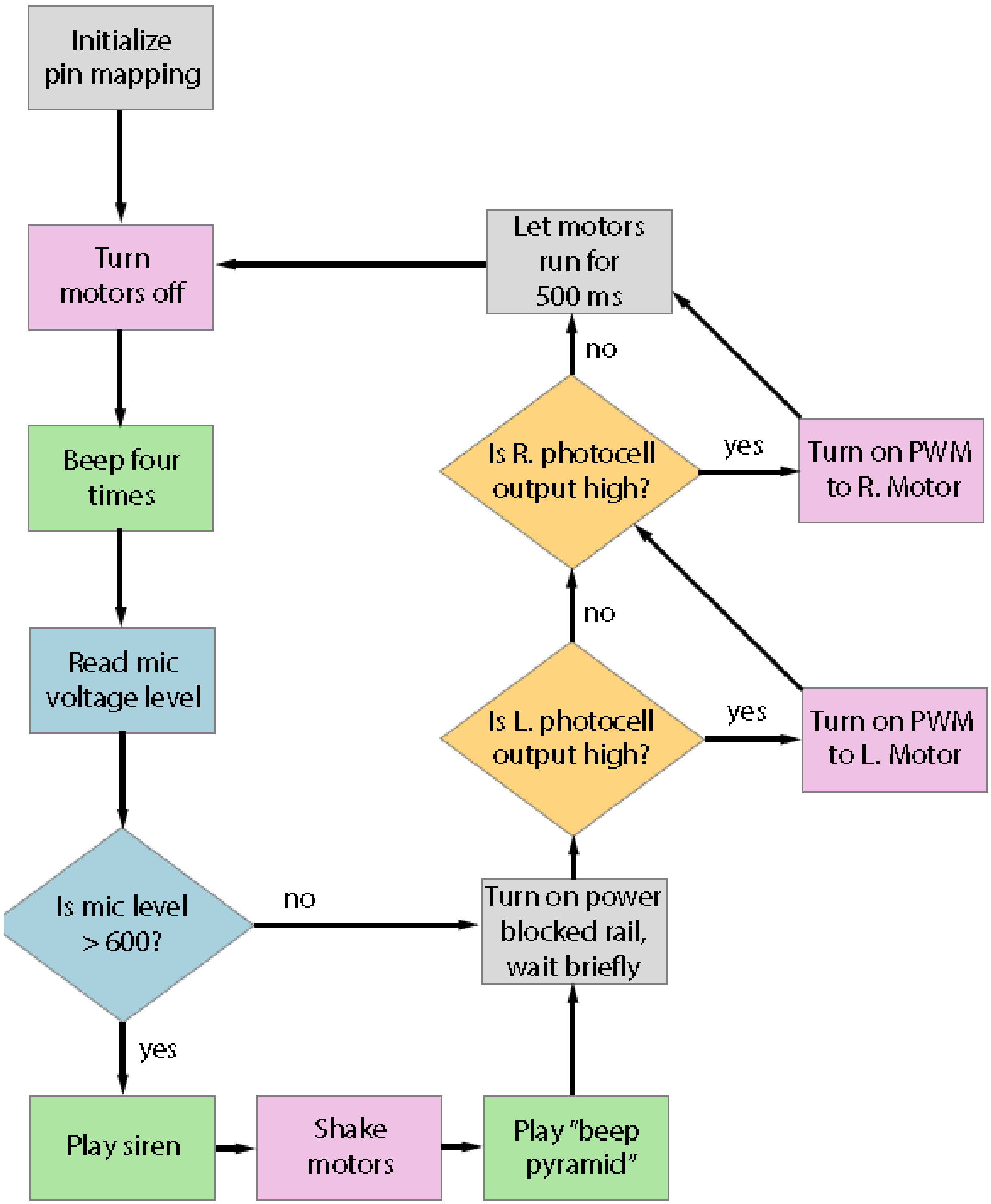
```
 if (digitalRead(LPHOTO))        // check each photocell/circuit output and determine
 {
   analogWrite(LMOTOR, MPOW);    // whether to run on the left motor...
 }
 if (digitalRead(RPHOTO))        // ...or right motor
 {
   analogWrite(RMOTOR, MPOW);
 }
 digitalWrite(PBLOCK, LOW);      // turn the power-blocked rail off

 sleep(500);                     // wait 500 ms

}

void beep(int pin, int freq, long ms)      //generate a square wave at a given frequency for ms miliseconds
{
   int k;
   long semiper = (long) (1000000/(freq*2));
   long loops = (long)((ms*1000)/(semiper*2));
          for (k=0;k<loops;k++)
          {
      digitalWrite(pin, HIGH);  //set buzzer pin high
      delayMicroseconds(semiper);  //for half of the period
      digitalWrite(pin, LOW);   //set buzzer pin low
      delayMicroseconds(semiper);  //for the other half of the period
          }
}
```

1.5.1 CODE FLOWCHART

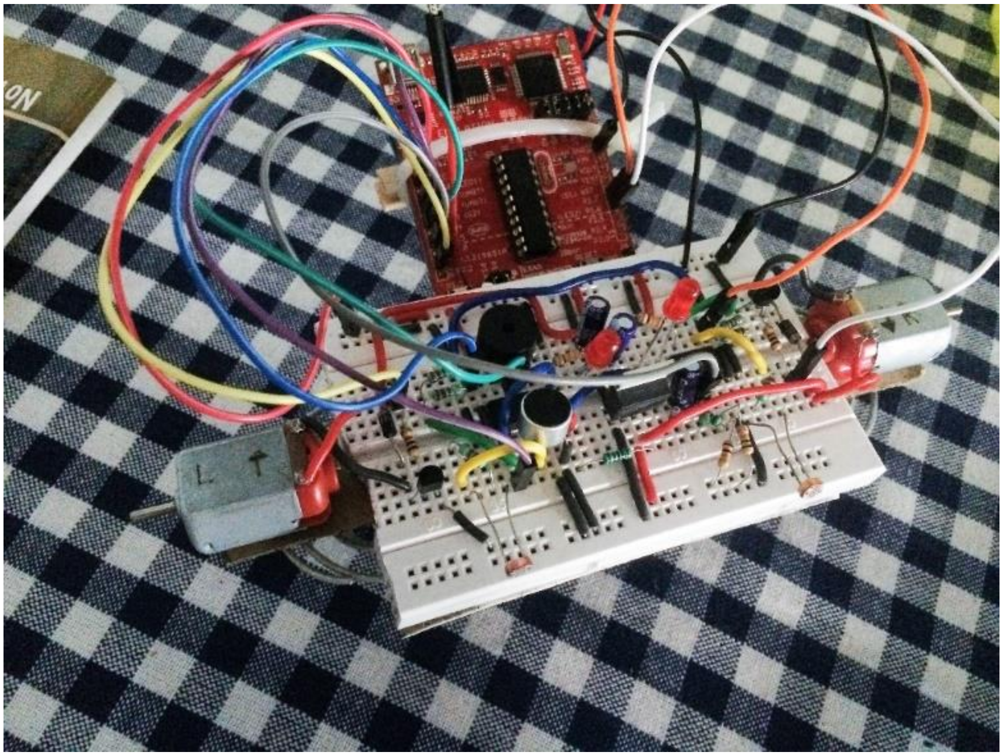


1.6 MAKING THE BOT

Assembling the robot was an enriching experience. We have tried to list down what we learnt in the points below.

1. Gathering all the required components was the first step, The Op-Amps used in this project are very exclusive and weren't available in the local stores so we had to order in online form Texas Instruments.
2. Since this is a bouncing robot we needed springs to make it bounce, and the springs we required were of specific dimensions that was again new hurdle for us, we tried many options but finally we got the springs from torch lights
3. Before putting the whole robot together, we had to test each and every sub circuit separately, involved in this project like microphone amplifier, motor driver, buzzer amplifiers, photocell etc.
4. Next, we had to build a chassis or a support system on which the whole circuit would be mounted. We wanted to make it sturdy as well as Eco-friendly, so we used ice cream sticks and cardboard stripes.
5. Finally, we mounted the whole circuit onto the chassis using zip ties.
6. After, we had assembled the whole robot, our next step was test run.
7. During the test run, our aim was to analyze our robot and look for any problems. We came across certain problems with working of motors, so we did some troubleshooting to make it work.

1.6 HOW IT FINALLY LOOKS?



Finally, we managed to bring all the components together and get all the components to work in a harmonious manner. The Result is a wonderful Bot that is a treat to both eyes and young learners to teach them about the basic of robotics.

Thanks for reading our report, hope you found it interesting.

1.8 TEAM DETAILS

Akhil Punia 14BEE0100

Shaik M. Umar Hassan 14BEE0094