## RASPBERRY PI CODE TO GET THE LSA VALUES FROM THE BOT:

```
import time

import Adafruit_ADS1x15

adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1

print('Reading ADS1x15 values, press Ctrl-C to quit...')

print('| {0:>6} | {1:>6} | {2:>6} | {3:>6} |'.format(*range(4)))
print('-' * 37)

while True:
    values = [0]*4

   values[0] = adc.read_adc(0, gain=GAIN)
    print((values[0]*60)/29791,values[0])
```

## RASPBERRY PI CODE TO MAKE THE ROBOT MOVE MANUALLY:

```
Import RPi.GPIO as gpio
from time import sleep
import signal
import Adafruit_ADS1x15
TIMEOUT = 1
gpio.setmode(gpio.BOARD)
```

```
adc = Adafruit_ADS1x15.ADS1115()
GAIN=1

rf_pwm = 18
rf_dir = 16

rb_pwm = 26
rb_dir = 24

lf_pwm = 11
lf_dir = 13

lb_pwm = 19
lb_dir = 21

gpio.setup(rf_pwm, gpio.OUT)
gpio.setup(rf_dir, gpio.OUT)

gpio.setup(rb_pwm, gpio.OUT)
gpio.setup(rb_dir, gpio.OUT)

gpio.setup(lf_pwm, gpio.OUT)
gpio.setup(lf_dir, gpio.OUT)

gpio.setup(lb_pwm, gpio.OUT)
gpio.setup(lb_dir, gpio.OUT)

rf = gpio.PWM(rf_pwm, 100)
rb = gpio.PWM(rb_pwm, 100)
lf = gpio.PWM(lf_pwm, 100)
lb = gpio.PWM(lb_pwm, 100)
```

```
rf.start(0)
rb.start(0)
lf.start(0)
lb.start(0)
#while True:
ss = adc.read_adc(0,gain=GAIN)
#  print ss

setpoint= (ss*60)/29791
last = setpoint
i=0
def line(last):
   #values = [0]*4
   value = adc.read_adc(0, gain=GAIN)

  current = (value*60)/29791

  error = 0.77*(current-setpoint) +0.0*(last-current)+(current-
setpoint+i)*0.00
   error=int(error)

   iii(error)
   last = current
   return last

print("reached")

def iii(change):

#         signal.signal(signal.SIGALRM, iii)
 #         signal.alarm(TIMEOUT)
#         s = input_i()
```

```
        # disable the alarm after success
#        signal.alarm(0)
    s="w"
       rf.start(60+change/2)
    rb.start(60+change/2)
    lf.start(60-change/2)
    lb.start(60-change/2)
    if s=="w":
      print("forward", change)
      gpio.output(rf_dir, gpio.HIGH)
      gpio.output(rb_dir, gpio.HIGH)
      gpio.output(lf_dir, gpio.LOW)
      gpio.output(lb_dir, gpio.LOW)
'''
    if s=="s":
      print("backward")
                gpio.output(rf_dir, gpio.LOW)
                gpio.output(rb_dir, gpio.HIGH)
                gpio.output(lf_dir, gpio.LOW)
                gpio.output(lb_dir, gpio.HIGH)

    if s=="d":
      print("right")
                gpio.output(rf_dir, gpio.LOW)
                gpio.output(rb_dir, gpio.LOW)
                gpio.output(lf_dir, gpio.HIGH)
                gpio.output(lb_dir, gpio.HIGH)

    if s=="a":
      print("left")
```

```
                    gpio.output(rf_dir, gpio.HIGH)
                    gpio.output(rb_dir, gpio.HIGH)
                    gpio.output(lf_dir, gpio.LOW)
                    gpio.output(lb_dir, gpio.LOW)

    if s=="f":
                   print("stop")
#                    break
'''
rf.start(0)
rb.start(0)
lf.start(0)
lb.start(0)
a=input()
while True:
  last = line(last)
rf.start(0)
rb.start(0)
lf.start(0)
lb.start(0)

gpio.cleanup()
```

PROBLEMS FACED WITH RASPBERRY PI:

Due to sudden jerks at the start of the bot, the bot went way from line and rasberry pi was not able to get it back on line at high speed. So we were only able to make the bot follow line at 35% of its max speed.

## ARDUINO CODE FOR LINE FOLLOWING:

```
int lsa=A0;
int p=0;
int error=0;
int maxspeed=200;//int junc=13;
//int junction=0;
int dir_lf=10;
int dir_rf=6;
int dir_lb=8;
int dir_rb=4;
int pwm_lf=11;
int pwm_rf=5;
int pwm_lb=9;
int pwm_rb=3;
int positionx=0;
int setpoint=35;
int rf=maxspeed;
int rb=maxspeed;
int lb=maxspeed;
int lf=maxspeed;
```

```
void setup() {
  // put your setup code here, to run once:
  pinMode(lsa,INPUT);
  Serial.begin(9600);
}

void loop() {
  char a =Serial.read();
  if (a=='f'){
  //movemotor(40,40,40,40,0,0);
  // put your main code here, to run repeatedly:
  while (true){
  /*positionx=analogRead(lsa);
  positionx=((float)positionx/921)*70;
  //Serial.println(positionx);
  pidcalc();
  pidcalcturn();*/
  movemotor(lf,lb,rf,rb,0,1,0,1);
    Serial.println("f");
  }
  }
```

```
  else{
    int asds=1321;
  }
}

void movemotor(int lf,int lb,int rf,int rb,int dlf,int drf,int dlb,
int drb){
   digitalWrite(dir_lf,dlf);
  analogWrite(pwm_lf,lf); // Left Forward

  digitalWrite(dir_rf,drf);
  analogWrite(pwm_rf,rf); // Right Forward

  digitalWrite(dir_lb,dlb);
  analogWrite(pwm_lb,lb); // Left Backward

  digitalWrite(dir_rb,drb);
  analogWrite(pwm_rb,rb); // Right Backward



}
```

```
int last=0;
int d=0;
void pidcalc(){


  p= positionx-setpoint;
  d=last-p;
  error = -1*p+0*d;
  last=p;
  Serial.println(error);
  //i= i+p;
  //d=p-lp;
  //error=kp*p+ki*i+kd*d;
}

void pidcalcturn(){
  if (error>90){
    error=40;
  }
  if (error<0){
    rf=maxspeed-error/2;
    rb=maxspeed-error/2;
```

```
    lf=maxspeed+error/2;
    lb=maxspeed+error/2;
  }
  else{
    rf=maxspeed-error/2;
    rb=maxspeed-error/2;
    lf=maxspeed+error/2;
    lb=maxspeed+error/2;
  }
}
```

PROBLEMS FACED IN ARDUINO:

Due to mechanical errors(which will be rectified soon) the bot is getting sudden jerks and because the alignment of one of the wheels is wrong it is disbalancing the vectors and thus does not support in line following. Thus we are only able to follow line at 50% of the max speed.

SOLUTIONS:

As soon as the errors are rectified, with more testing we can make the bot move faster.