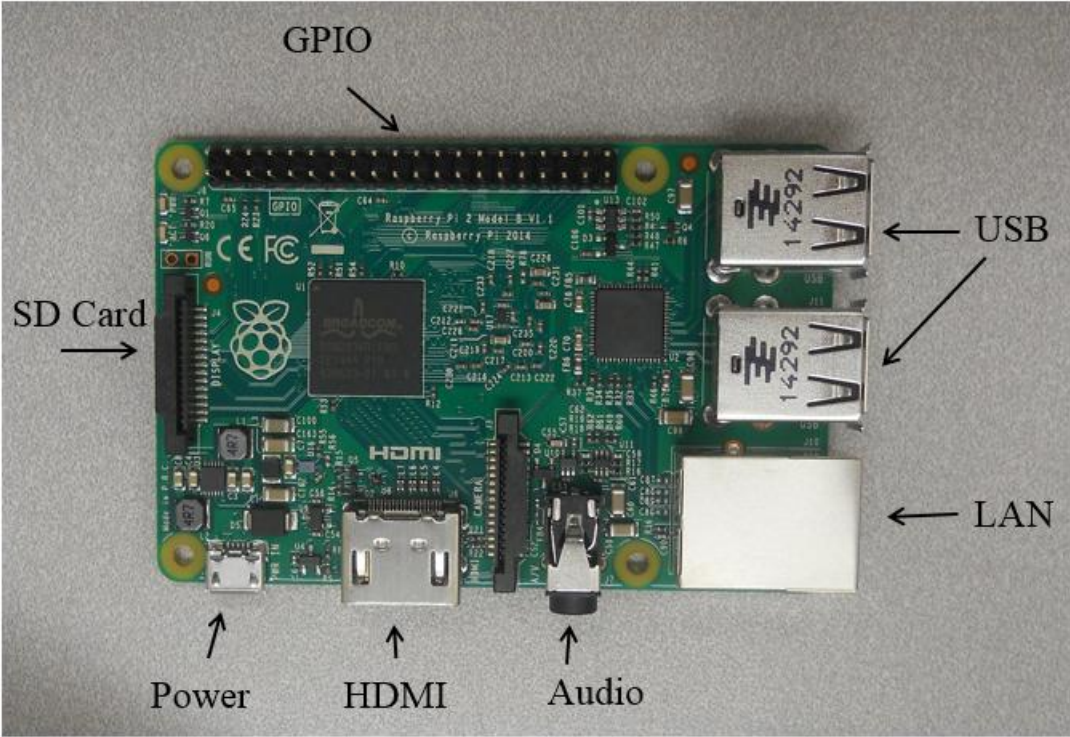
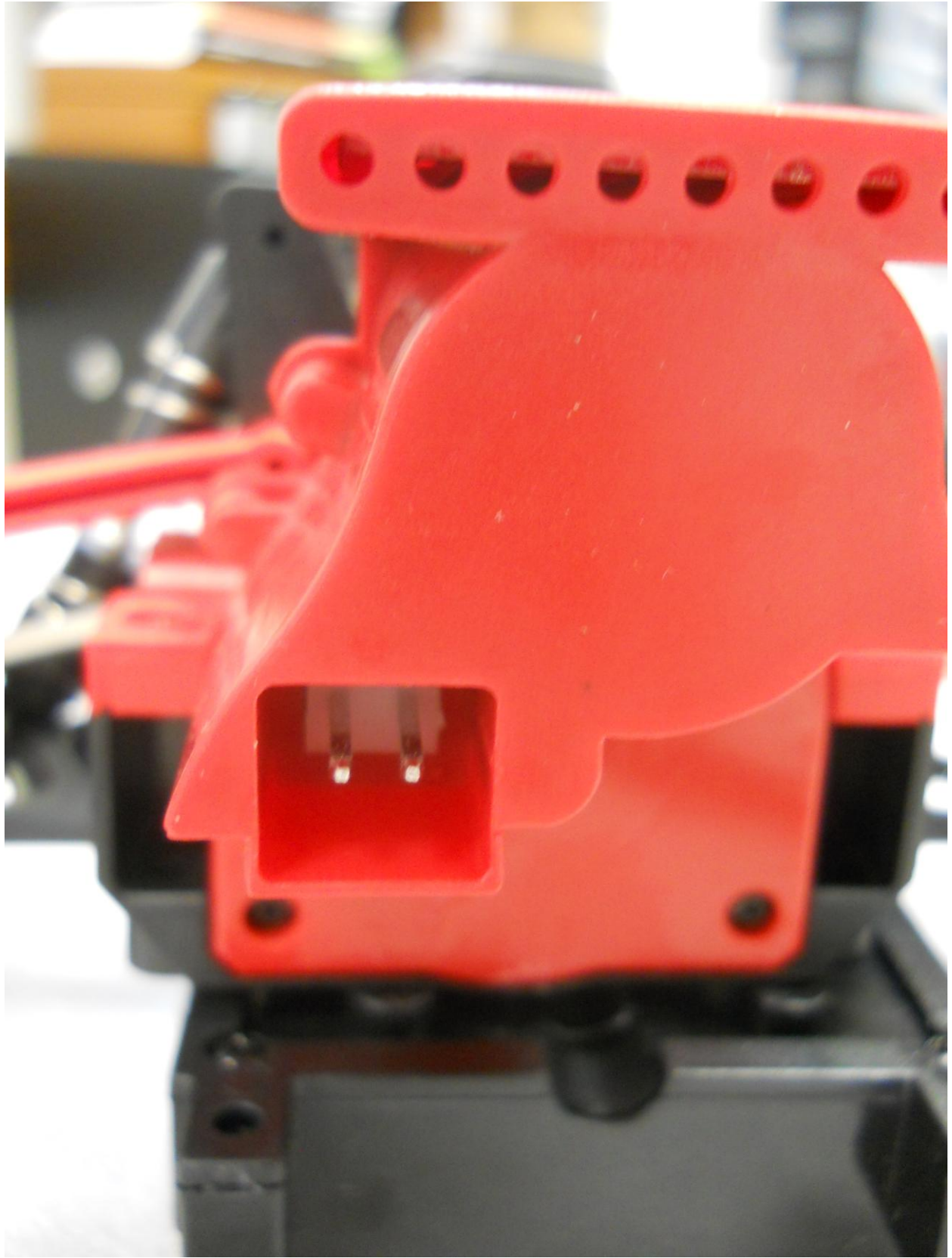


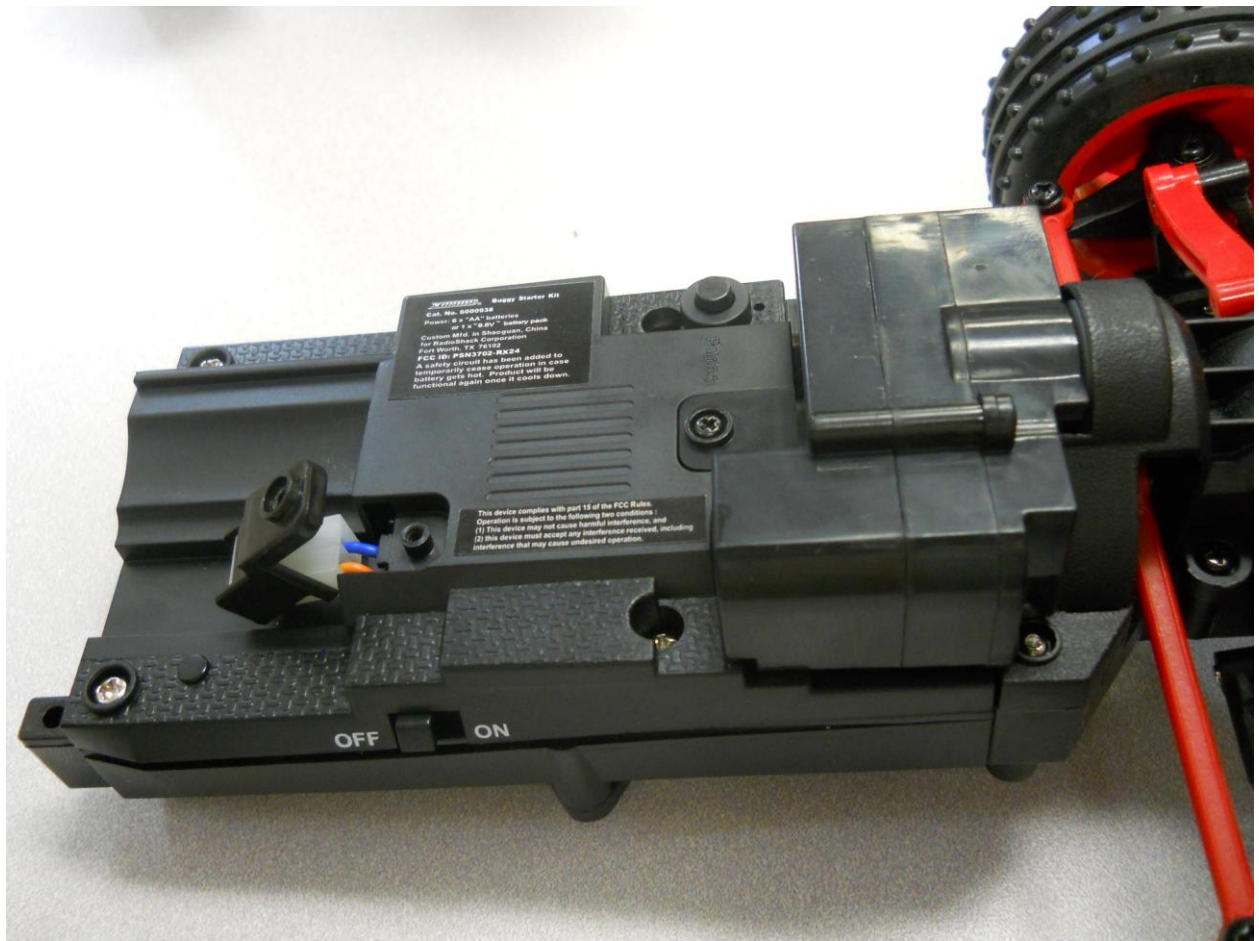
Chapter 1: Adding Raspberry Pi to an RC Vehicle





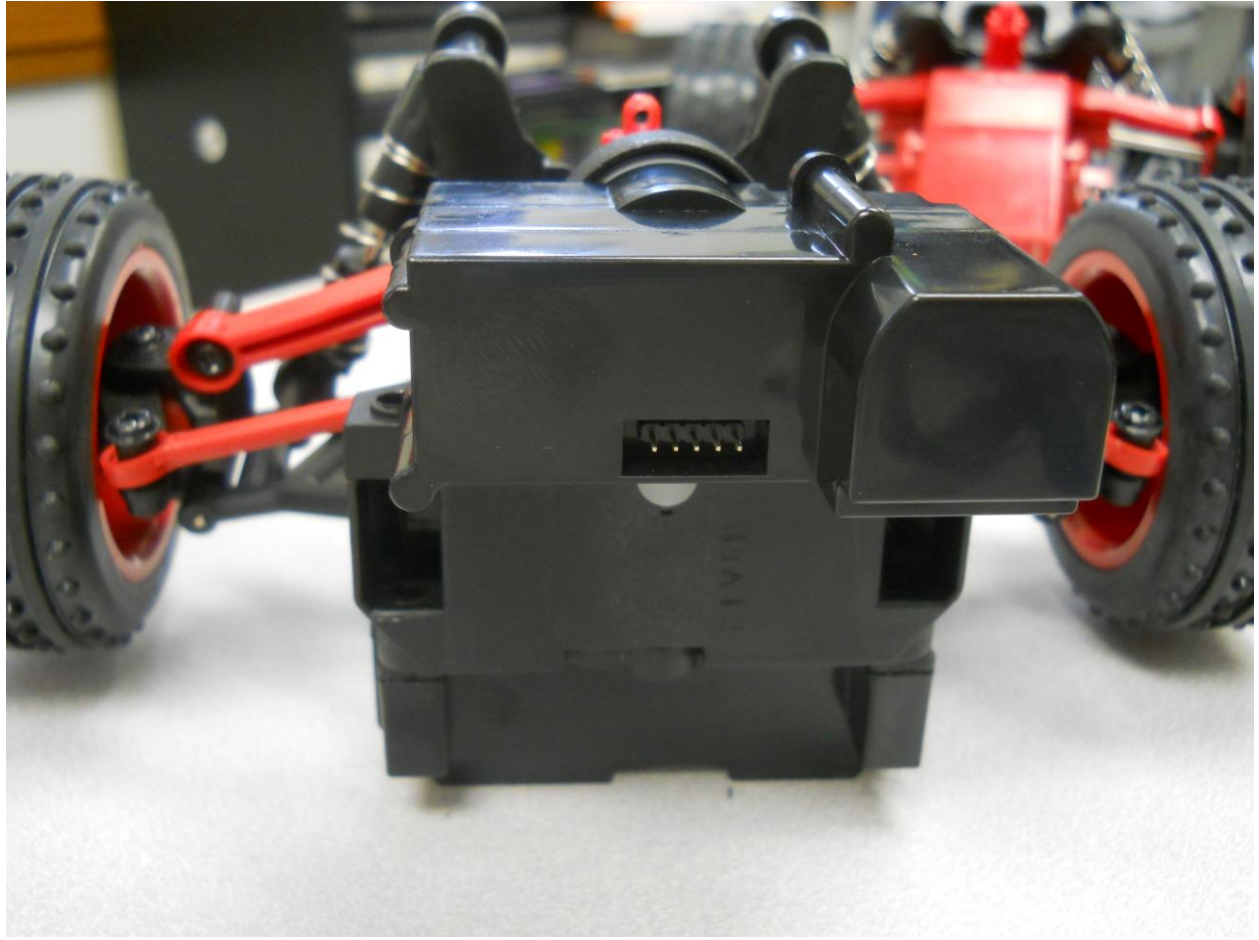


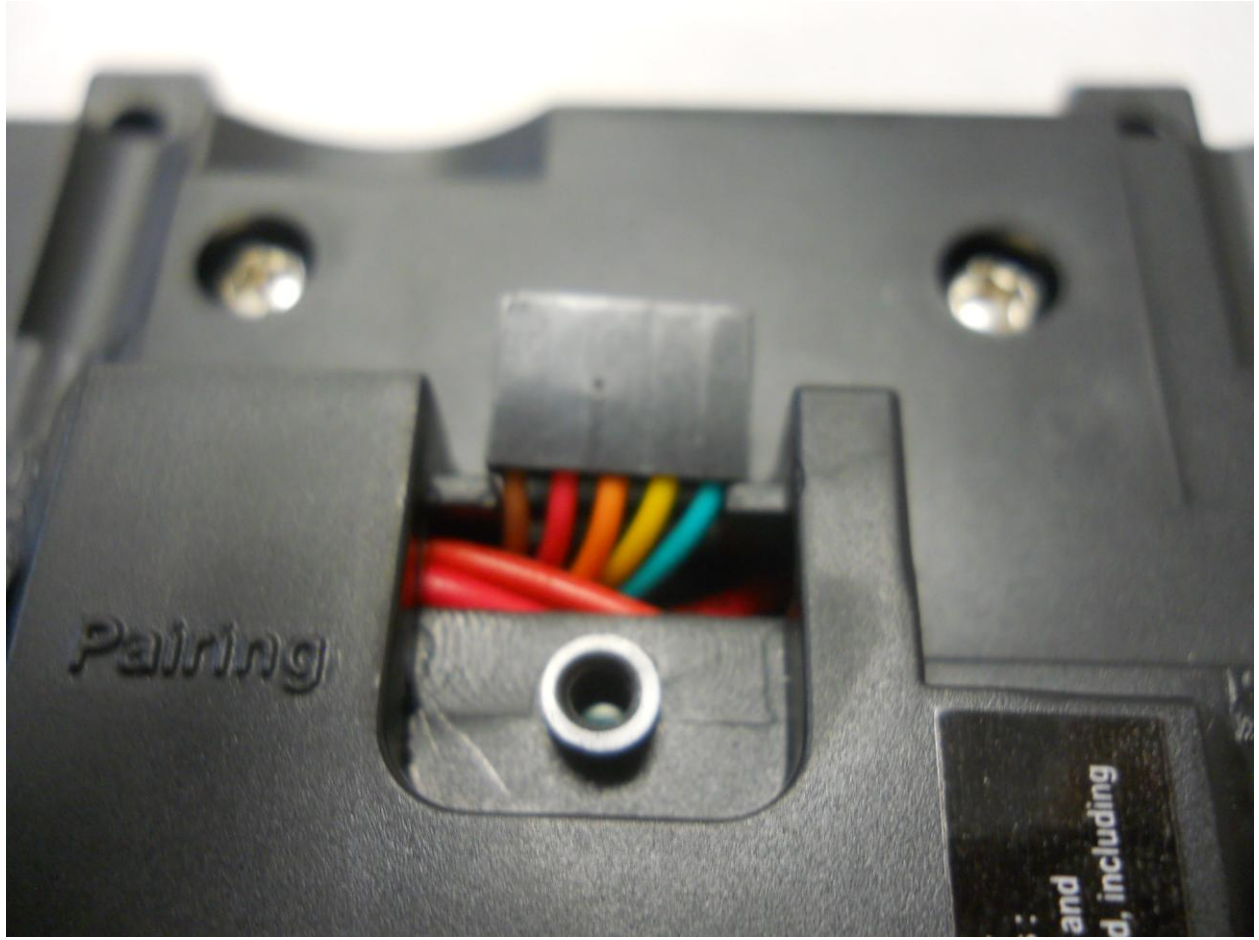




XXXXXXXXXX Super Starter Kit
Cat. No. 000000
Power: 2 x "AA" batteries
Size: 1.5 x 2.5 x 1.5" battery pack
Custom Mfg. in Shanghai, China
for Mattel/Hasbro Corporation
Fort Worth, TX 76102
FCC ID: R00000-XXXX
A safety circuit has been added to
temporarily make operation in case
battery gets hot. Product will be
functional again once it cools down.

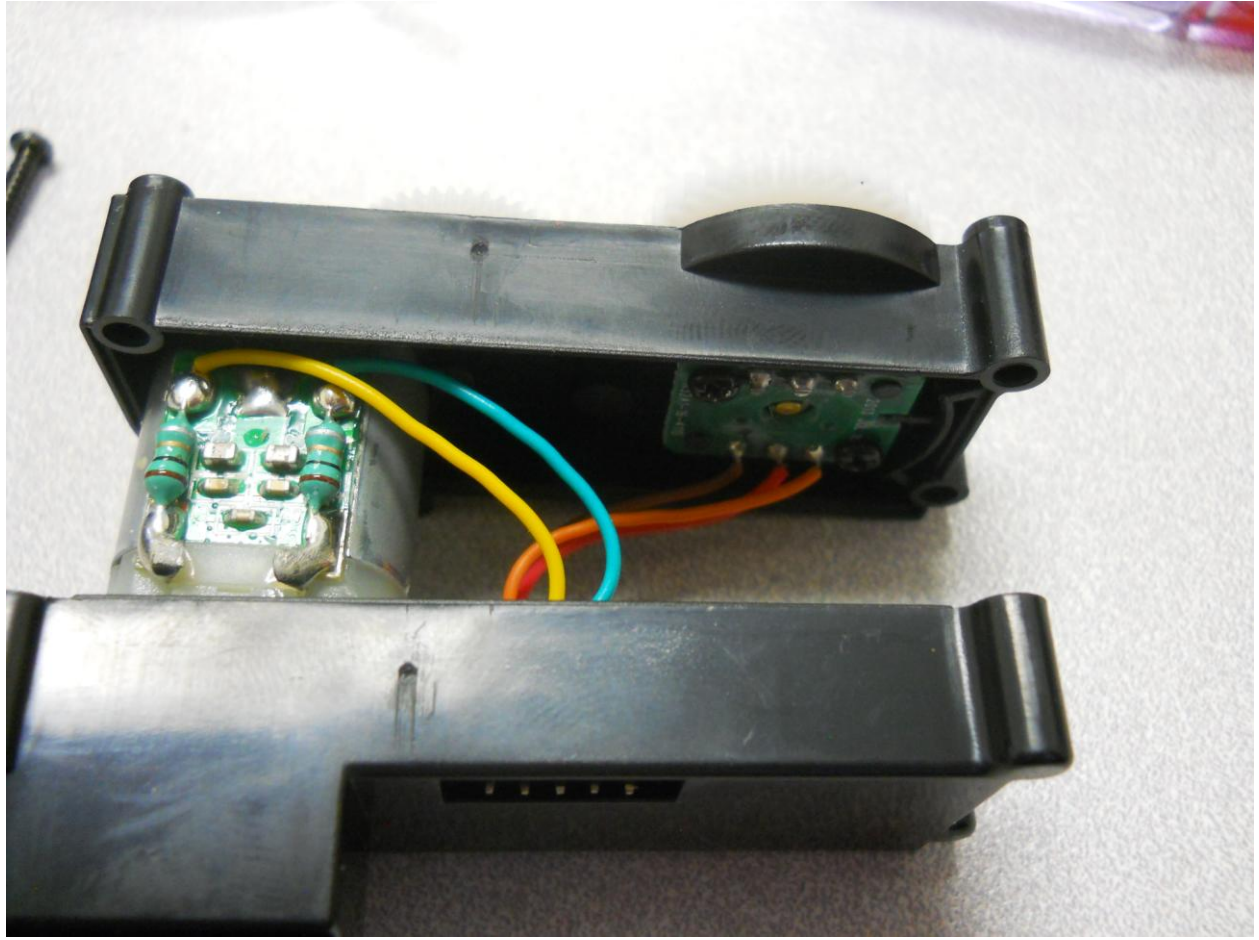
This device complies with part 15 of the FCC Rules.
Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and
(2) this device must accept any interference received, including
interference that may cause undesired operation.

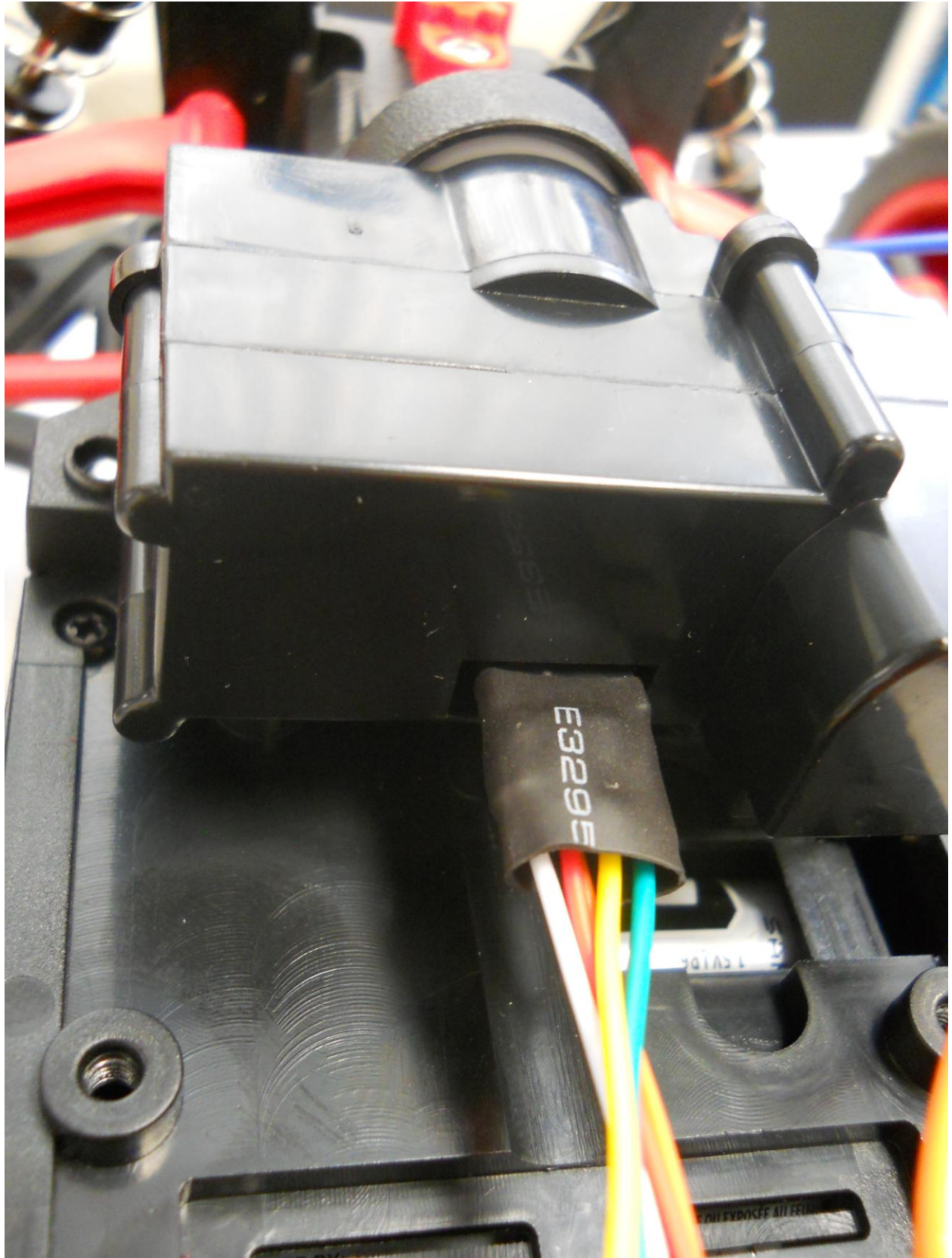


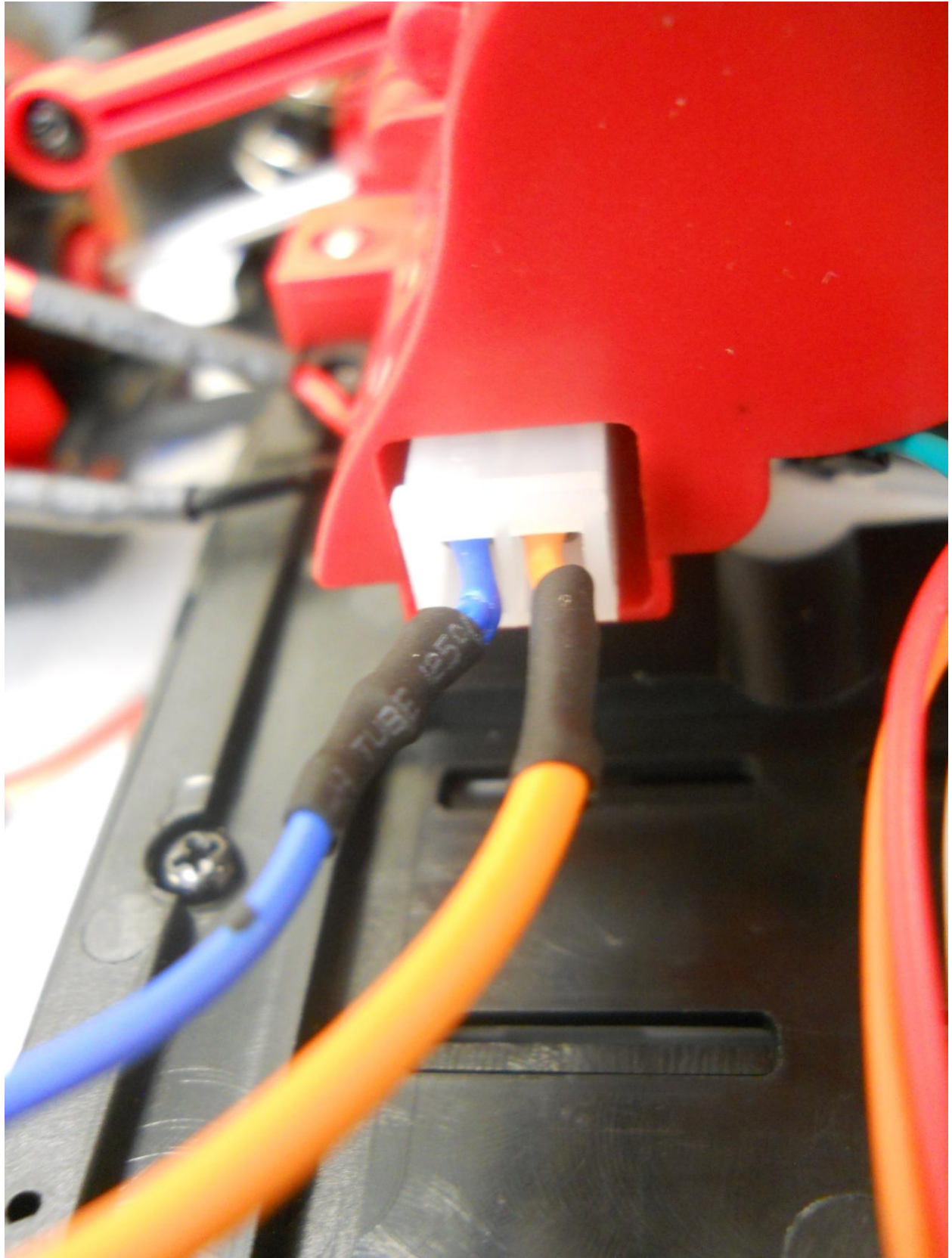


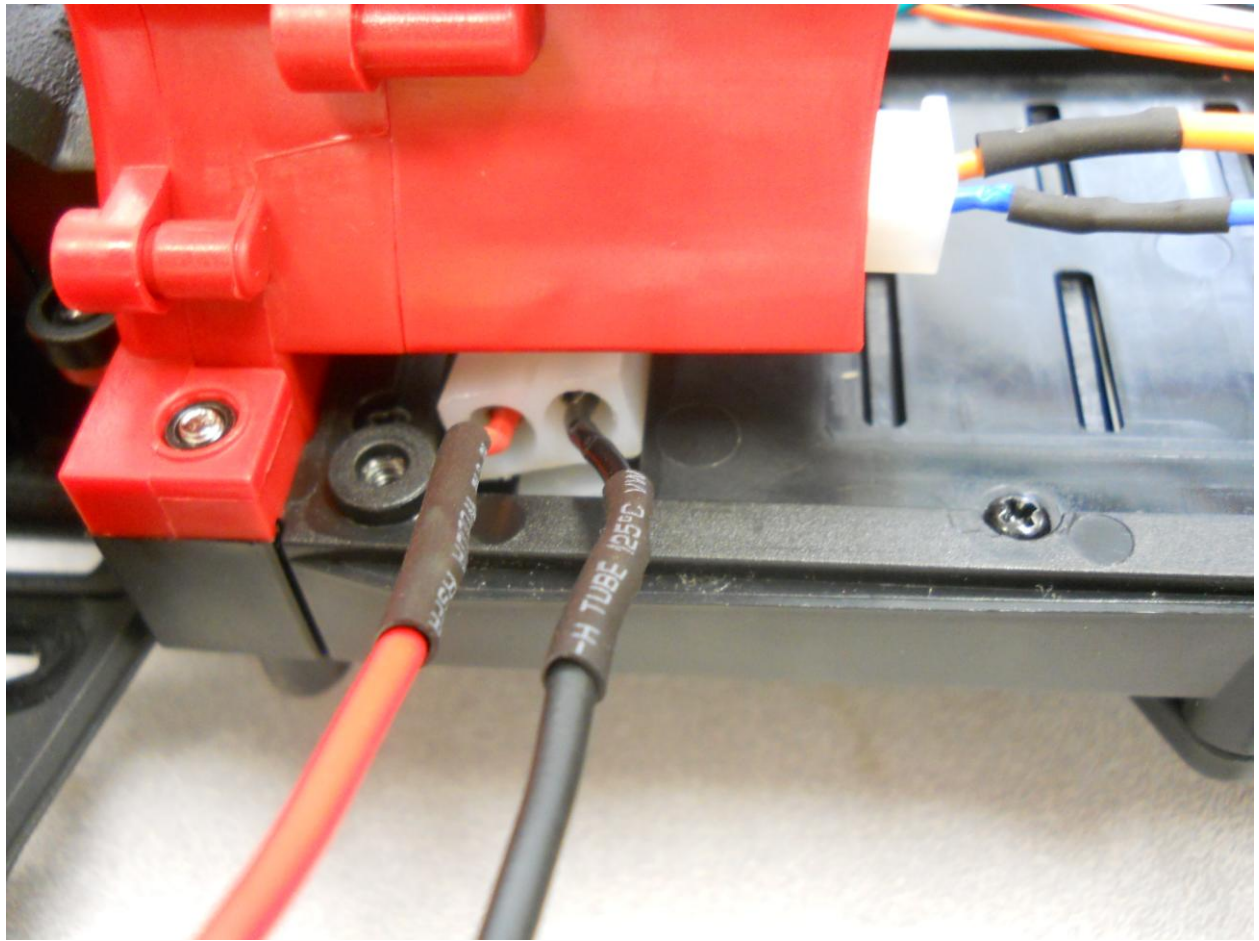
Pairing

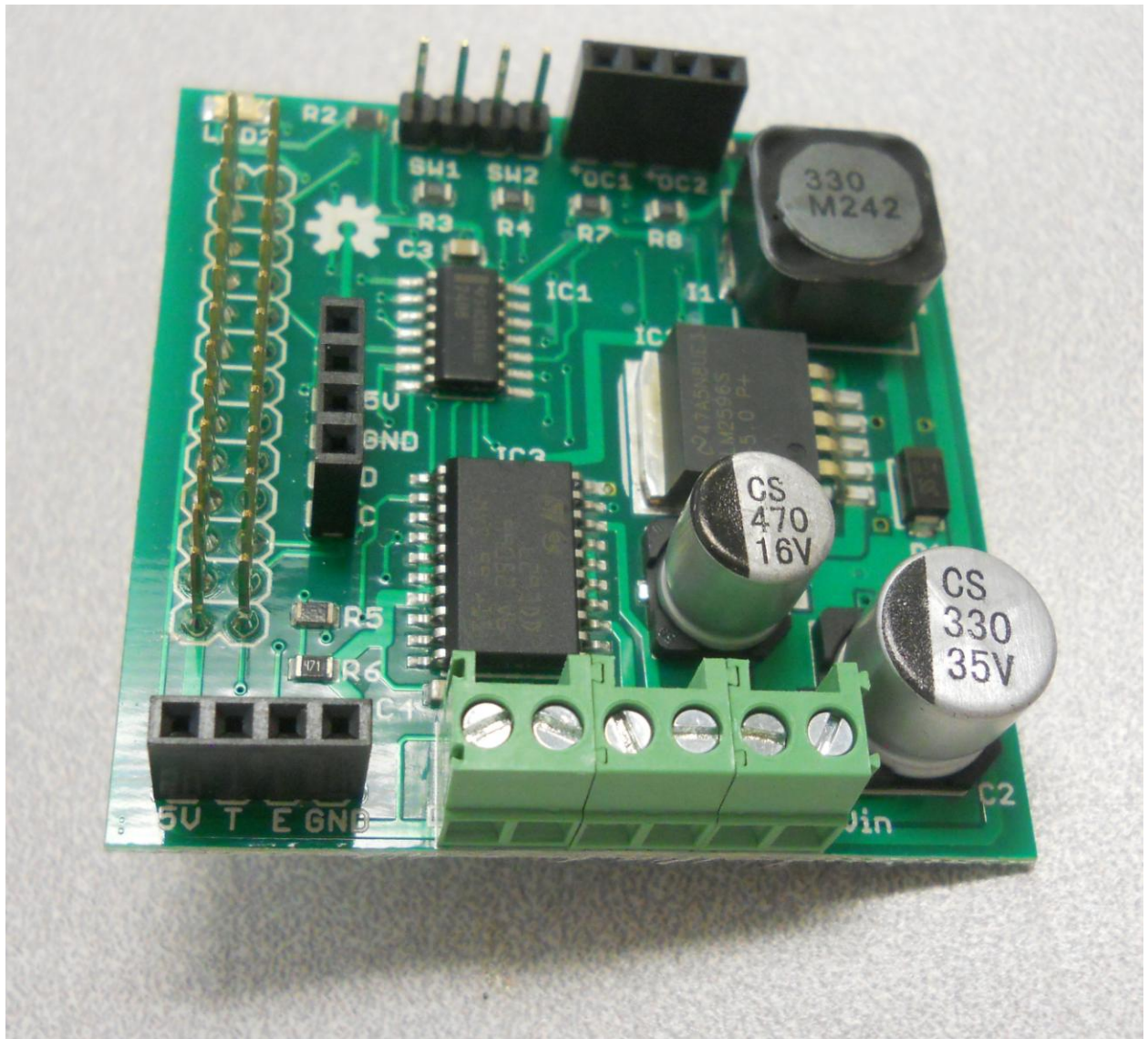
and, including

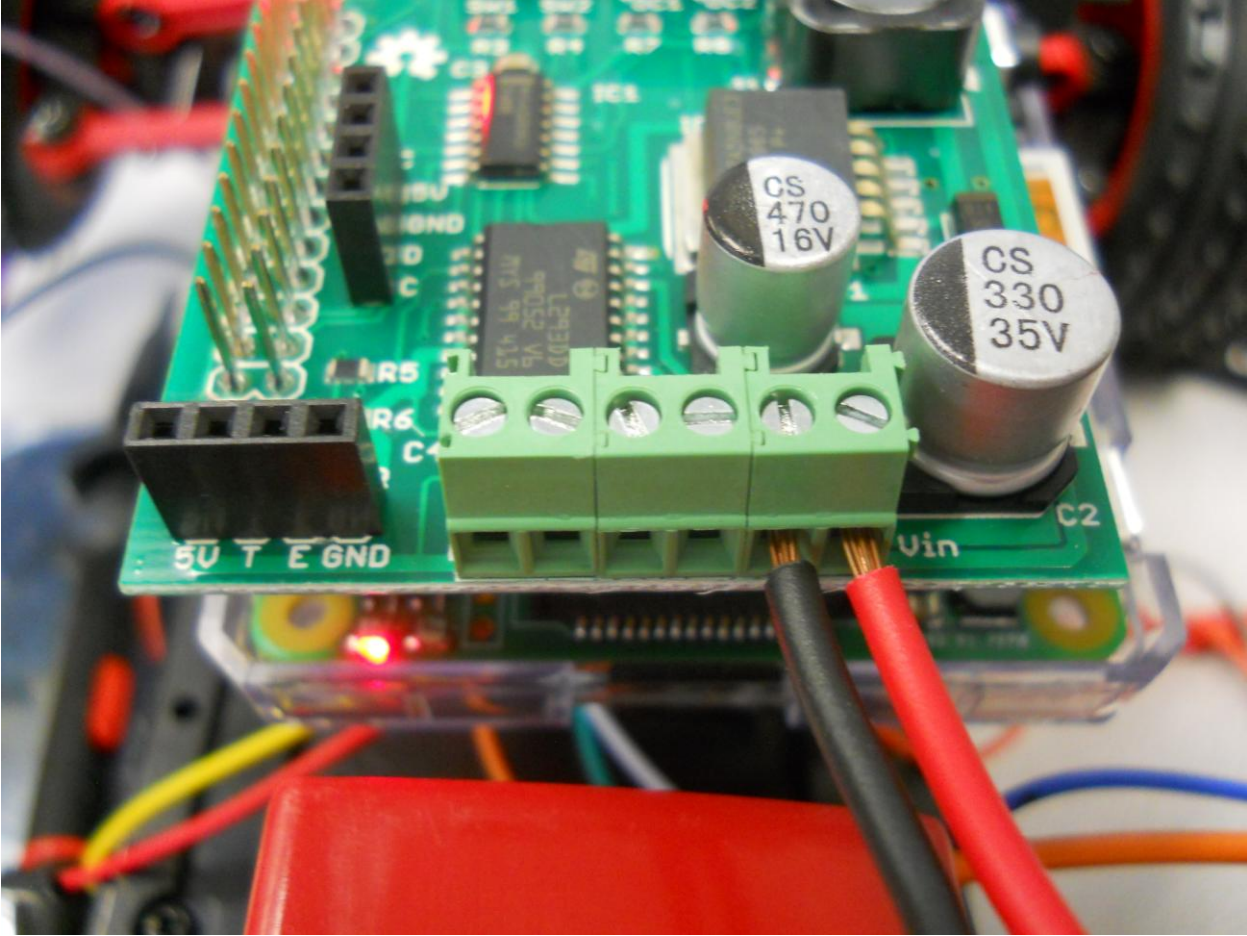


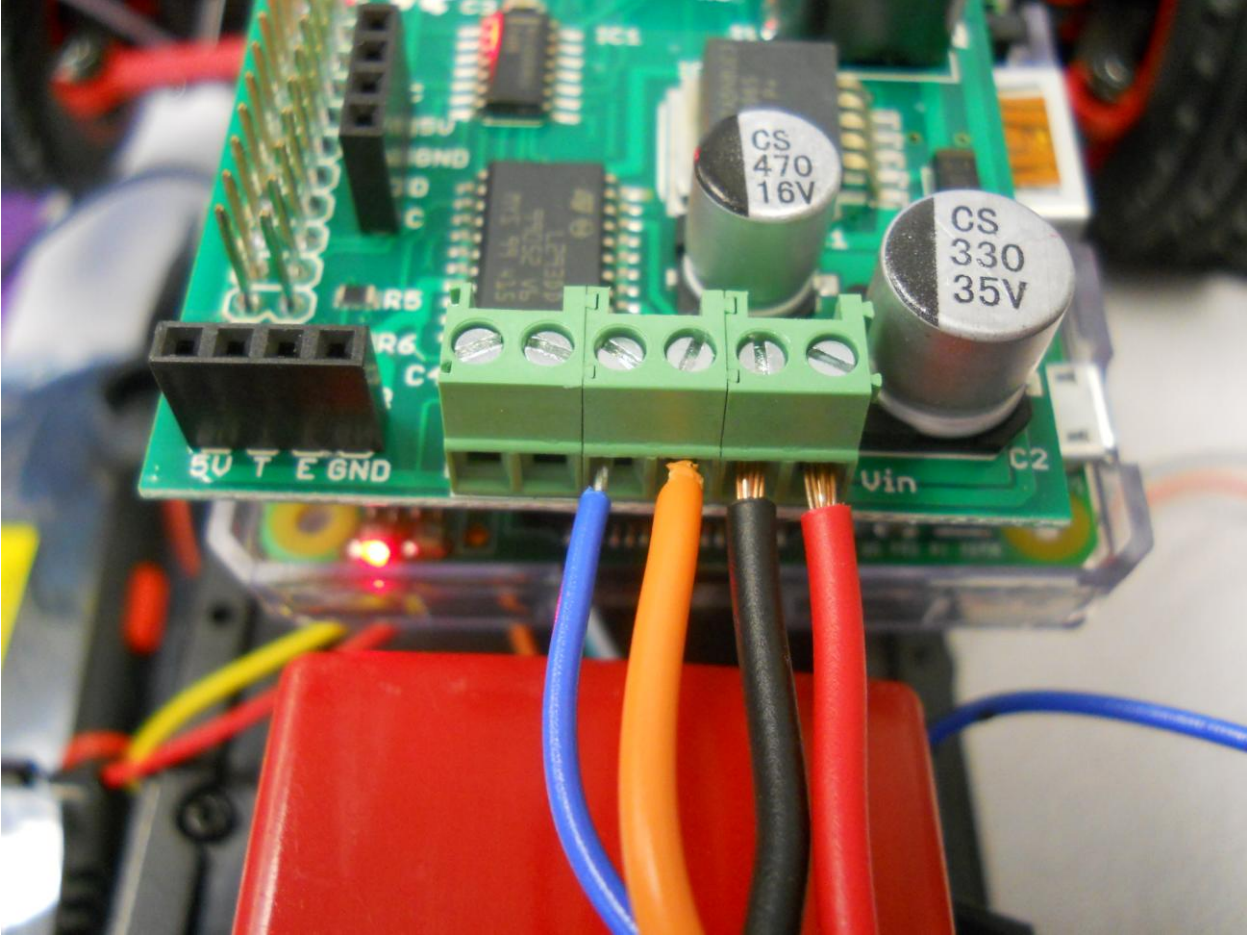


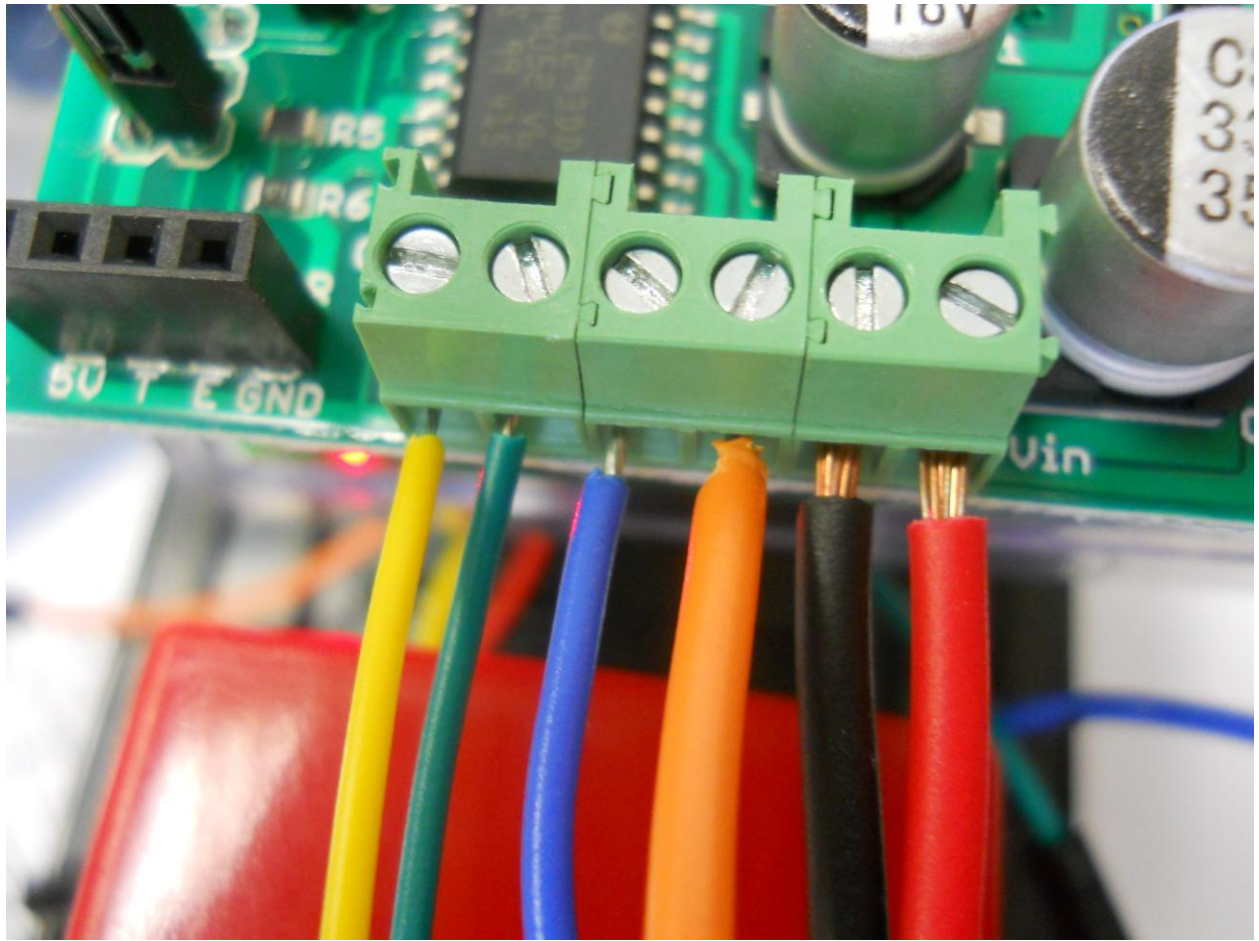




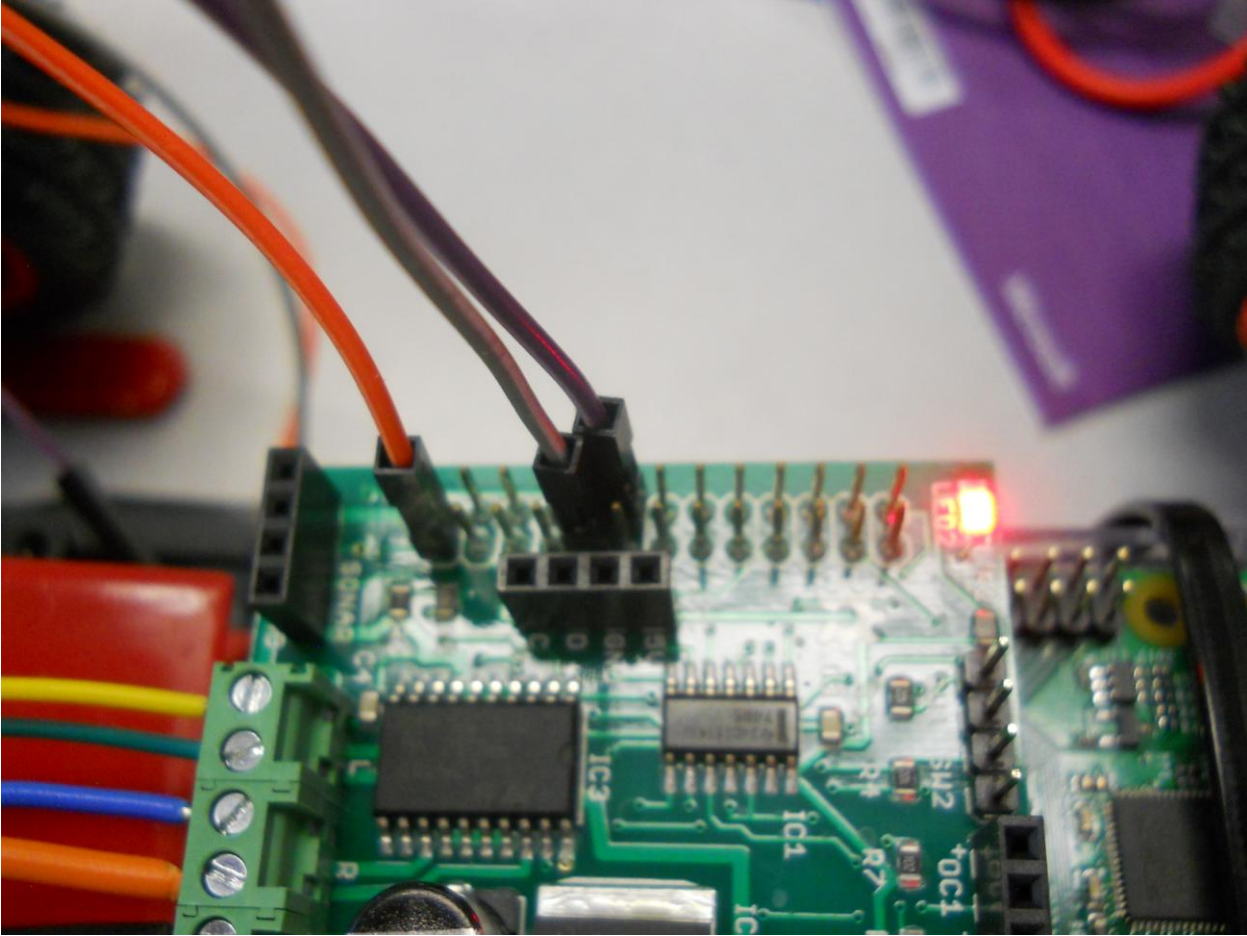








Pin 1 3.3V	<input type="checkbox"/> <input type="radio"/>	Pin 2 5V
Pin 3 GPIO2	<input type="radio"/> <input type="radio"/>	Pin 4 5V
Pin 5 GPIO3	<input type="radio"/> <input type="radio"/>	Pin 6 GND
Pin 7 GPIO4	<input type="radio"/> <input type="radio"/>	Pin 8 GPIO14
Pin 9 GND	<input type="radio"/> <input type="radio"/>	Pin 10 GPIO15
Pin 11 GPIO17	<input type="radio"/> <input type="radio"/>	Pin 12 GPIO18
Pin 13 GPIO27	<input type="radio"/> <input type="radio"/>	Pin 14 GND
Pin 15 GPIO22	<input type="radio"/> <input type="radio"/>	Pin 16 GPIO23
Pin 17 3.3V	<input type="radio"/> <input type="radio"/>	Pin 18 GPIO24
Pin 19 GPIO10	<input type="radio"/> <input type="radio"/>	Pin 20 GND
Pin 21 GPIO9	<input type="radio"/> <input type="radio"/>	Pin 22 GPIO25
Pin 23 GPIO11	<input type="radio"/> <input type="radio"/>	Pin 24 GPIO8
Pin 25 GND	<input type="radio"/> <input type="radio"/>	Pin 26 GPIO7
Pin 27 ID_SD	<input type="radio"/> <input type="radio"/>	Pin 28 ID_SC
Pin 29 GPIO5	<input type="radio"/> <input type="radio"/>	Pin 30 GND
Pin 31 GPIO6	<input type="radio"/> <input type="radio"/>	Pin 32 GPIO12
Pin 33 GPIO13	<input type="radio"/> <input type="radio"/>	Pin 34 GND
Pin 35 GPIO19	<input type="radio"/> <input type="radio"/>	Pin 36 GPIO16
Pin 37 GPIO26	<input type="radio"/> <input type="radio"/>	Pin 38 GPIO20
Pin 39 GND	<input type="radio"/> <input type="radio"/>	Pin 40 GPIO21



```
pi@raspberrypi: ~/xmod
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
from rrb2 import *

pwmPin = 18
dc = 10

GPIO.setmode(GPIO.BCM)
GPIO.setup(pwmPin, GPIO.OUT)
pwm = GPIO.PWM(pwmPin, 320)
rr = RRB2()

pwm.start(dc)
rr.set_led1(1)

rr.set_motors(1, 1, 1, 1)

print("Loop, press CTRL C to exit")
while 1:
    time.sleep(0.075)

pwm.stop()
GPIO.cleanup() █

-UU-:**--F1  xmod.py      All L23      (Python)-----
Auto-saving...done
```

```
pi@raspberrypi: ~/xmod
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
from rrb2 import *
import tty
import sys
import termios
def getch():
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    tty.setraw(sys.stdin.fileno())
    ch = sys.stdin.read(1)
    termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch
pwmPin = 18
dc = 10
GPIO.setmode(GPIO.BCM)
GPIO.setup(pwmPin, GPIO.OUT)
pwm = GPIO.PWM(pwmPin, 320)
rr = RRB2()
pwm.start(dc)
rr.set_led1(1)
var = 'n'
speed1 = 0
speed2 = 0
direction1 = 1
direction2 = 1
while var != 'q':
    var = getch()
    if var == 'l':
-UU-: **--F1 xmodControl.py Top L1 (Python)-----
```

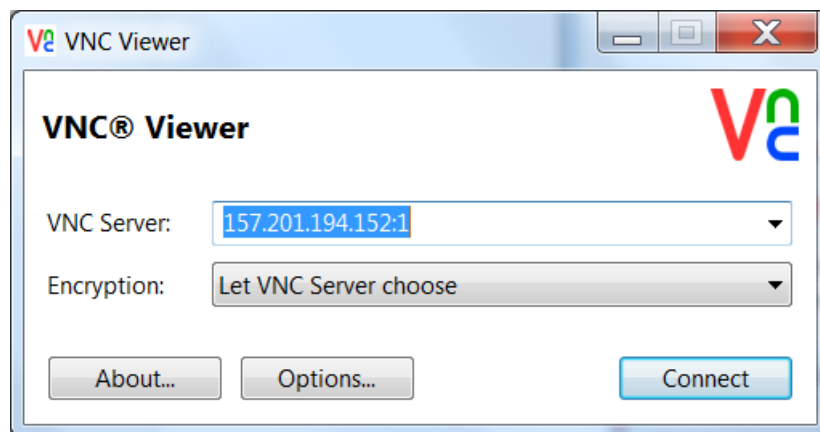


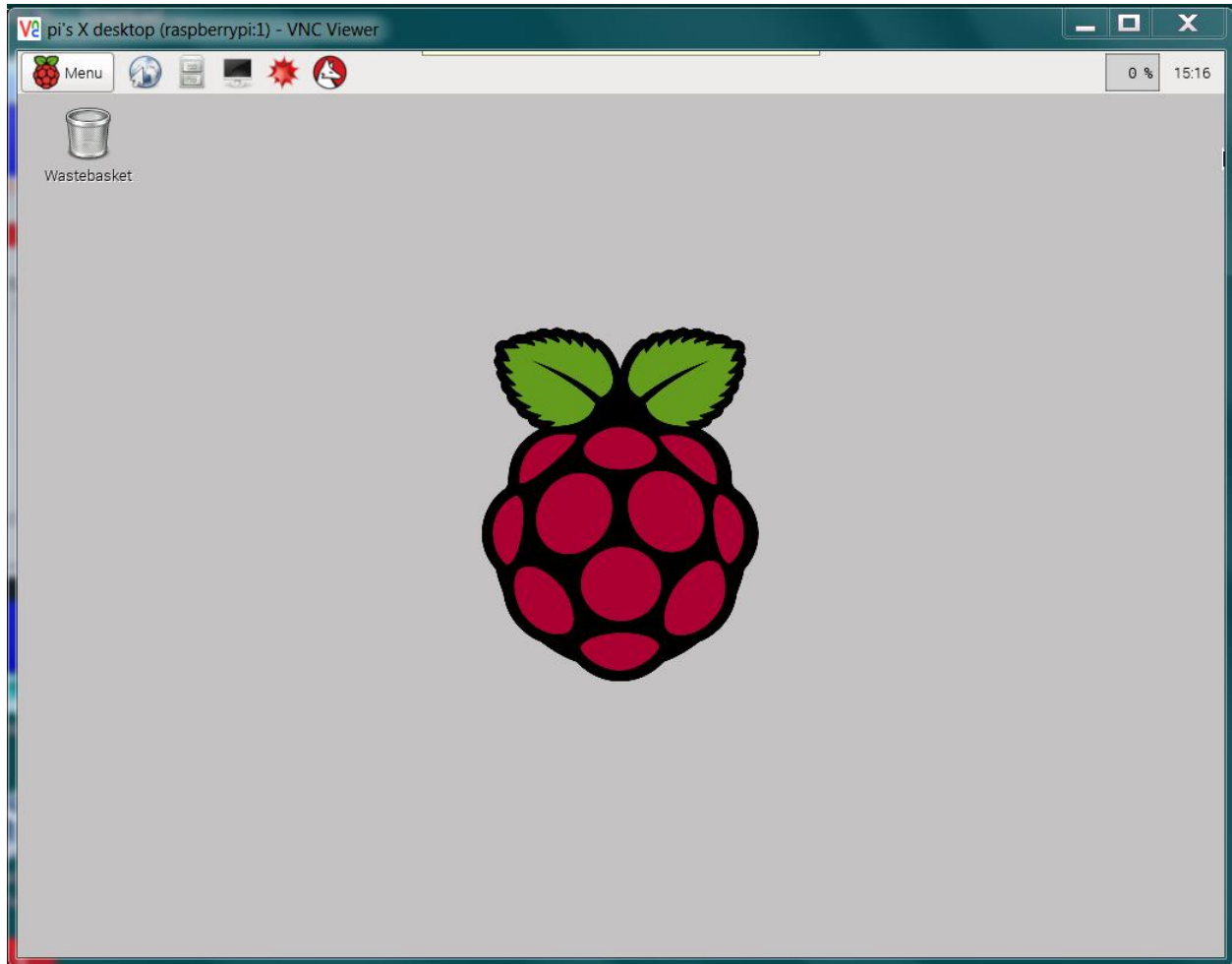
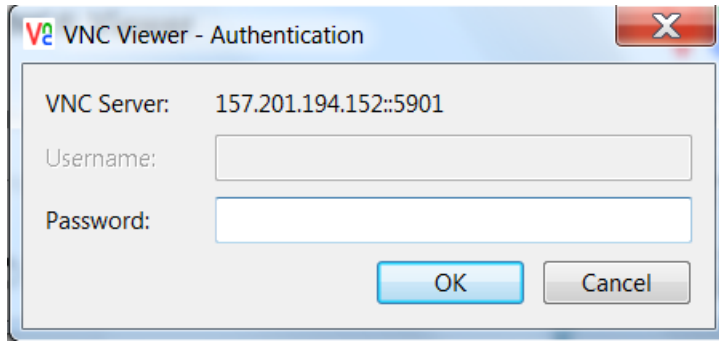
```
pi@raspberrypi: ~/xmod
File Edit Options Buffers Tools Python Help
rr.set_led1(1)
var = 'n'
speed1 = 0
speed2 = 0
direction1 = 1
direction2 = 1

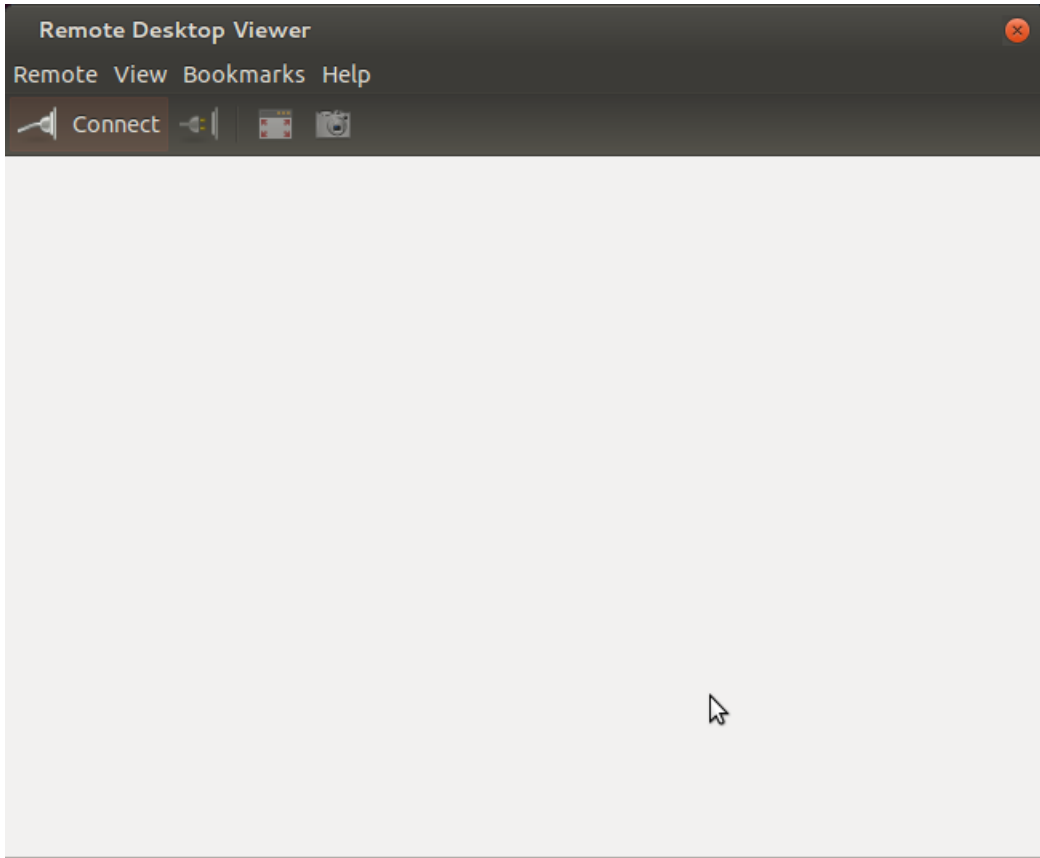
while var != 'q':
    var = getch()
    if var == 'l':
        speed1 = 0.5
        direction2 = 1
    if var == 'r':
        speed2 = 0.5
        direction2 = 0
    if var == 's':
        speed2 = 0.1
        direction = 1
    if var == 'f':
        speed1 = 1
        direction1 = 1
    if var == 'b':
        speed1 = 1
        direction1 = 0
    rr.set_motors(speed1, direction1, speed2, direction2)
    time.sleep(0.1)

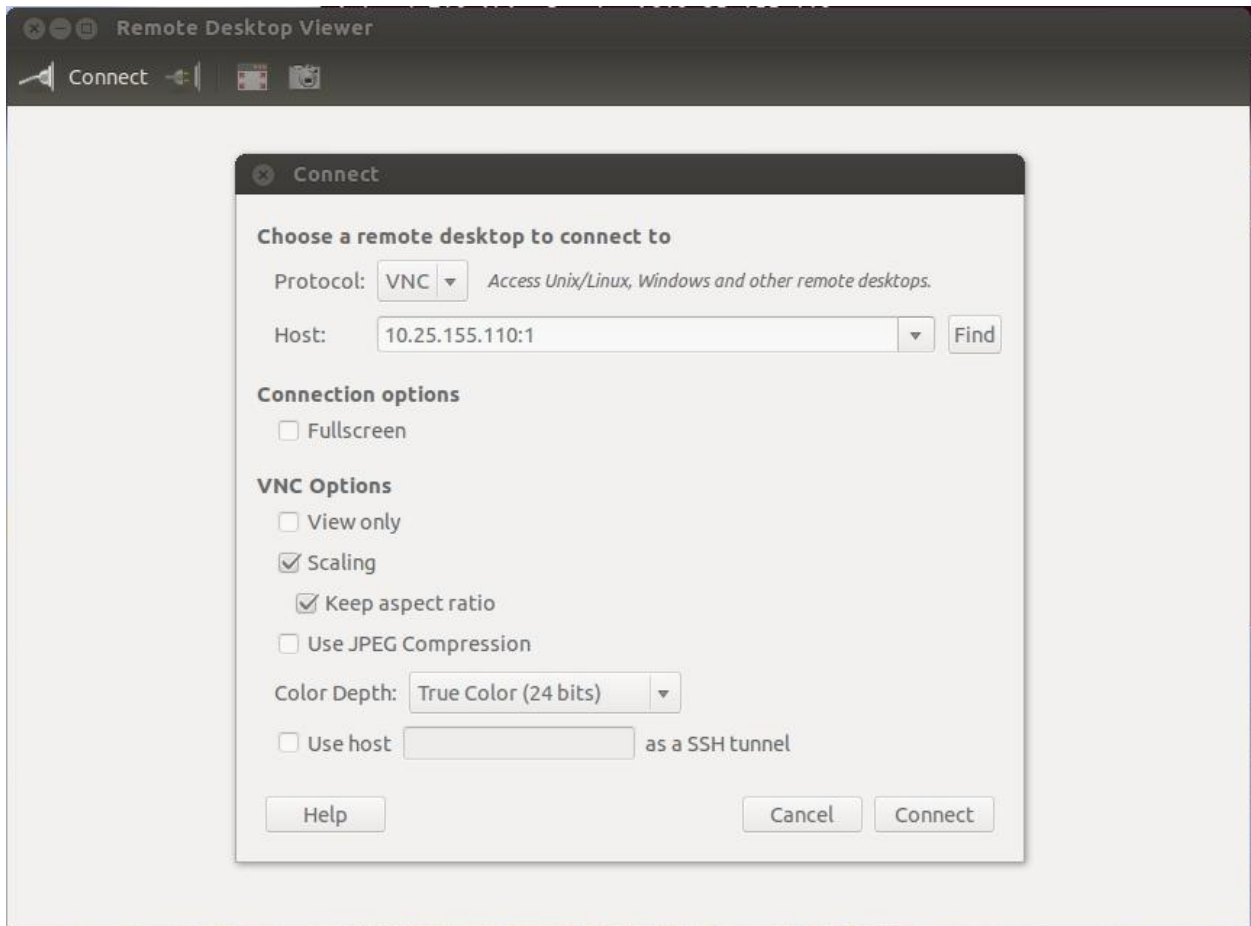
pwm.stop()
GPIO.cleanup()

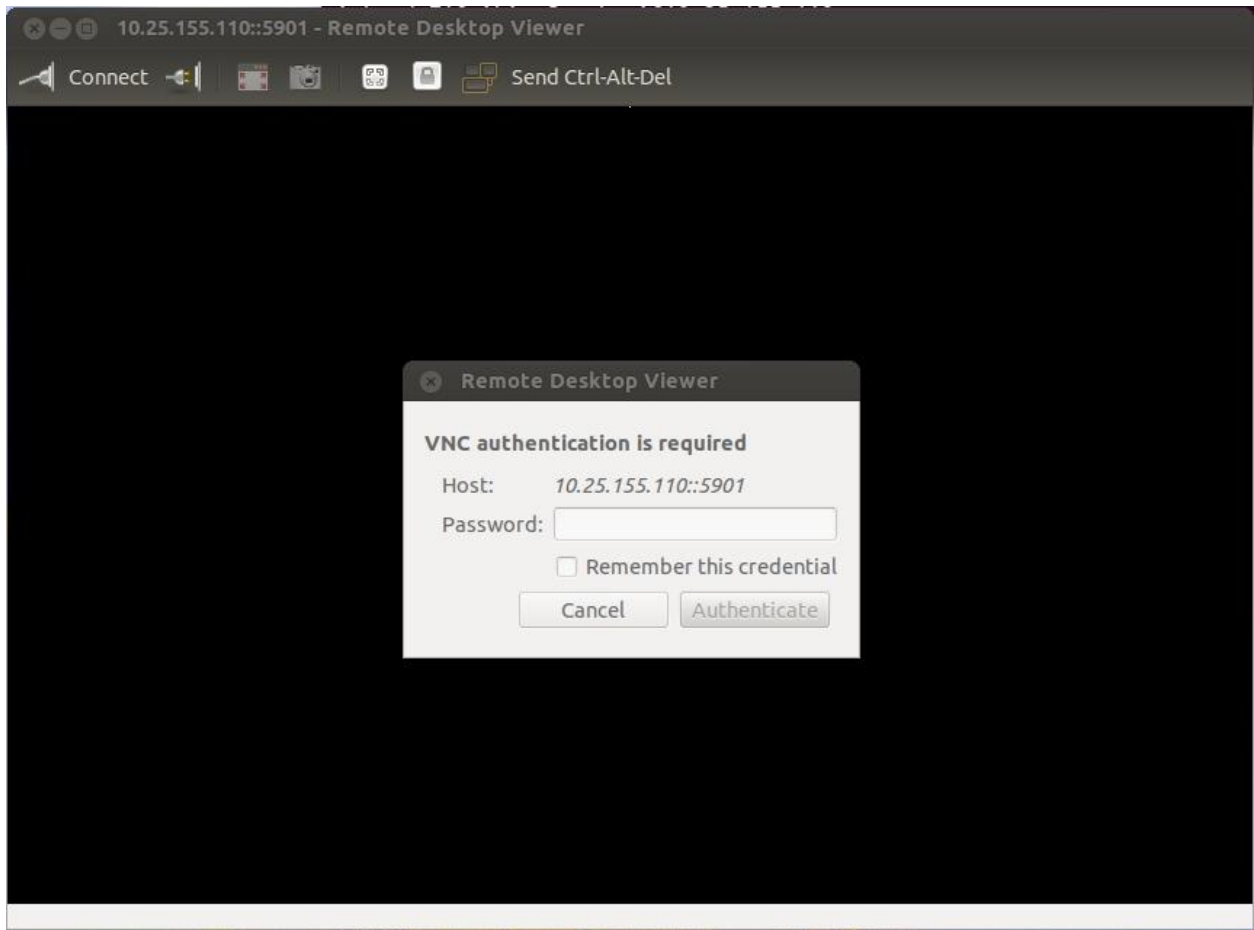
-UU-: **--F1 xmodControl.py Bot L36 (Python)
```

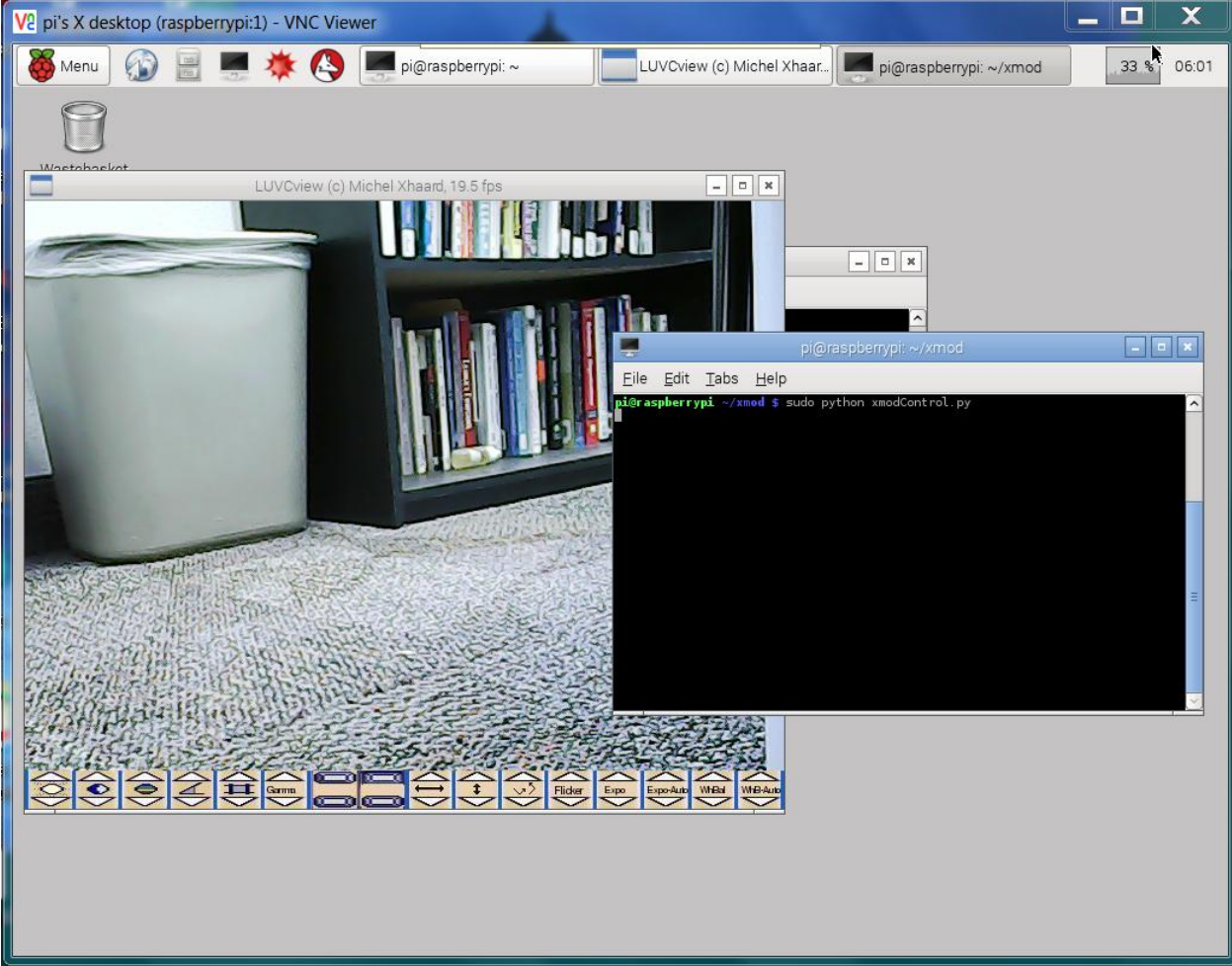








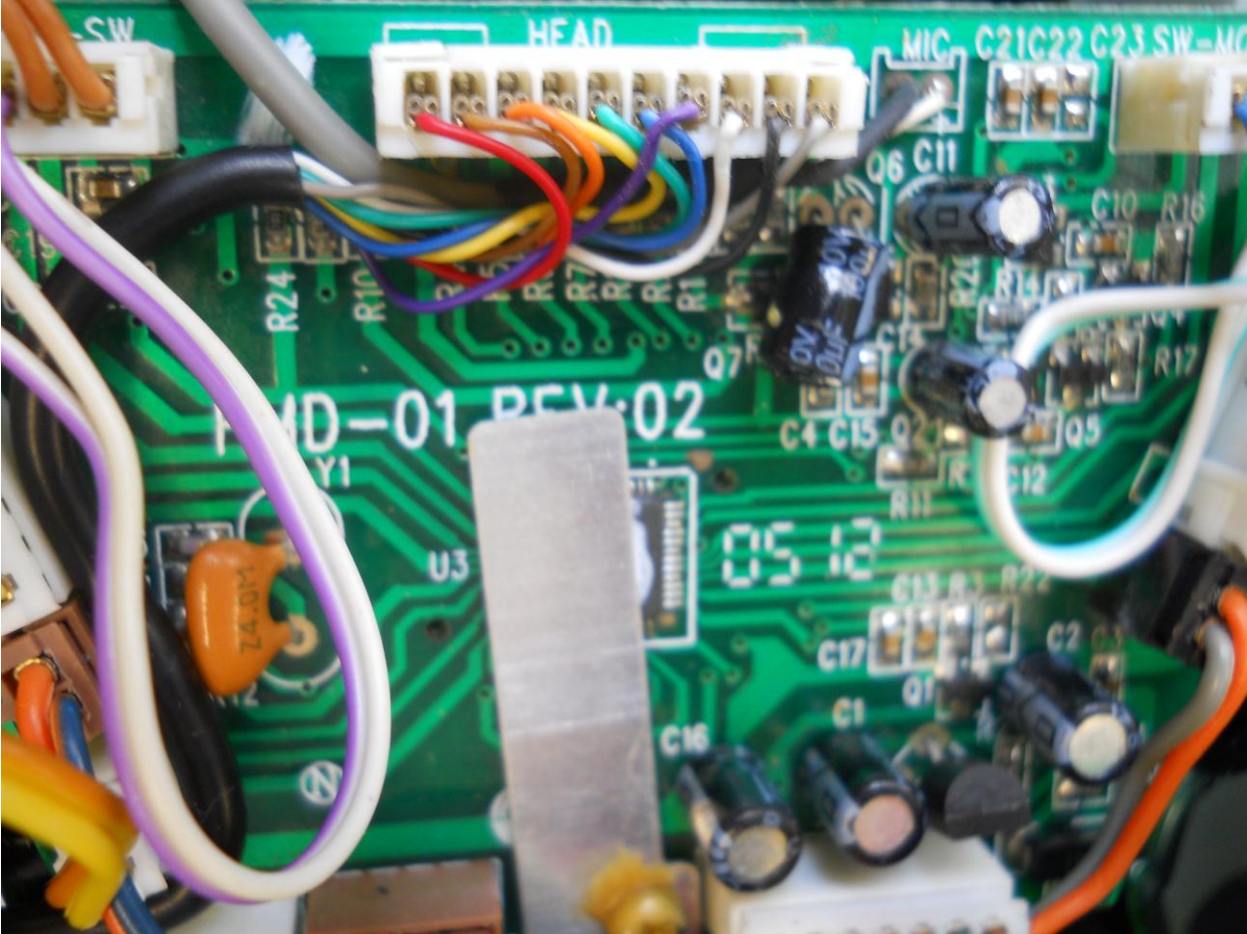


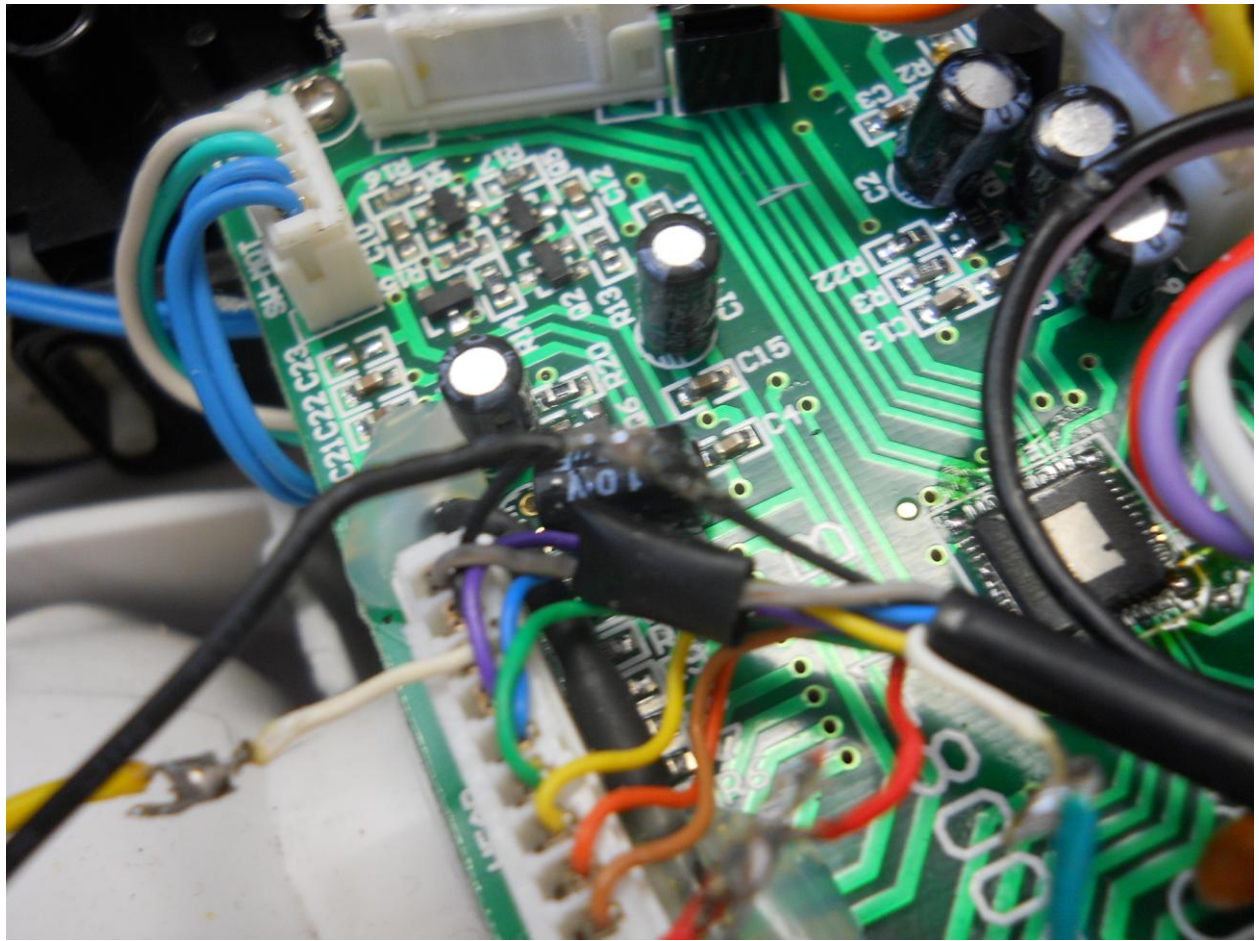


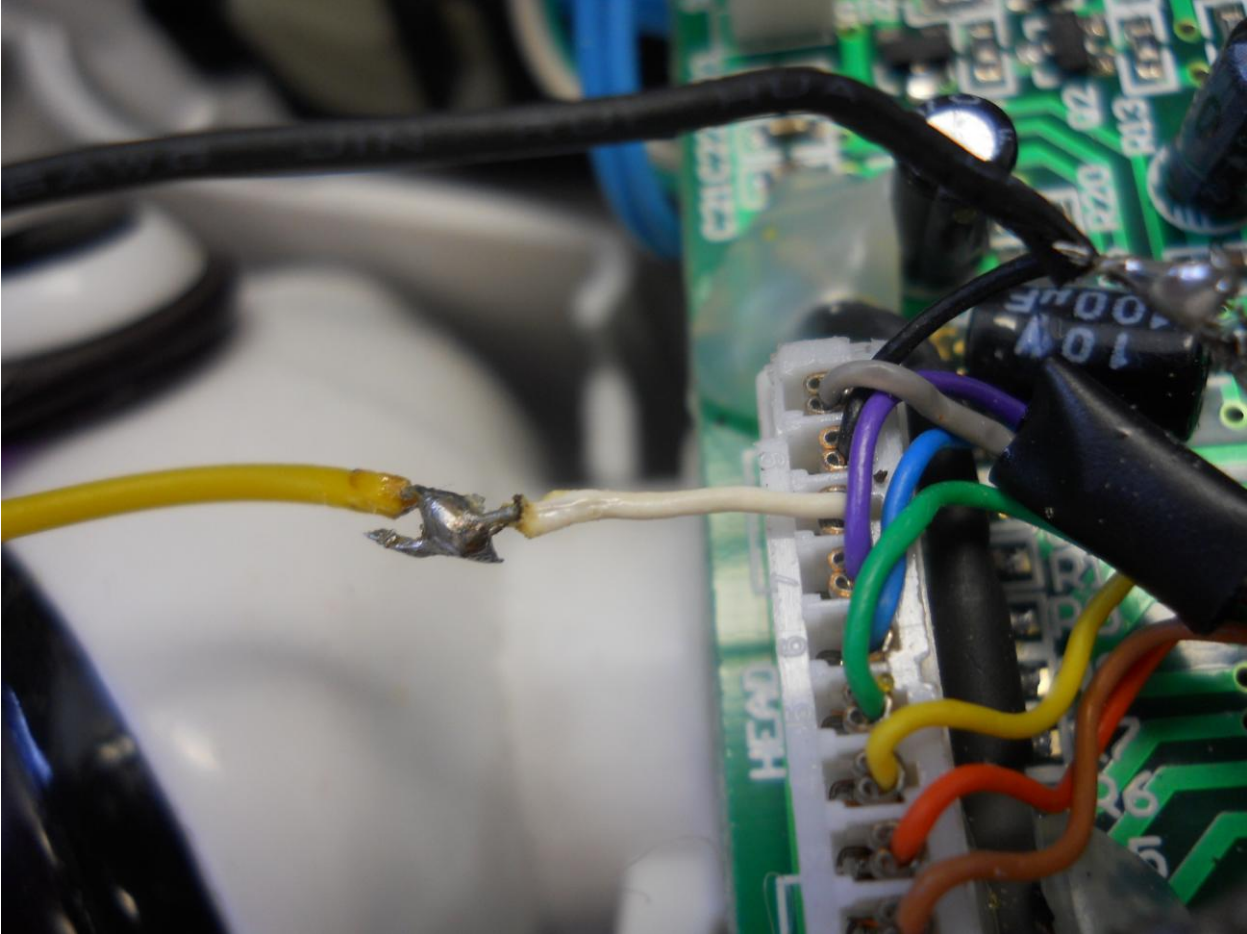
Chapter 2: Adding Raspberry Pi to a Humanoid Robot



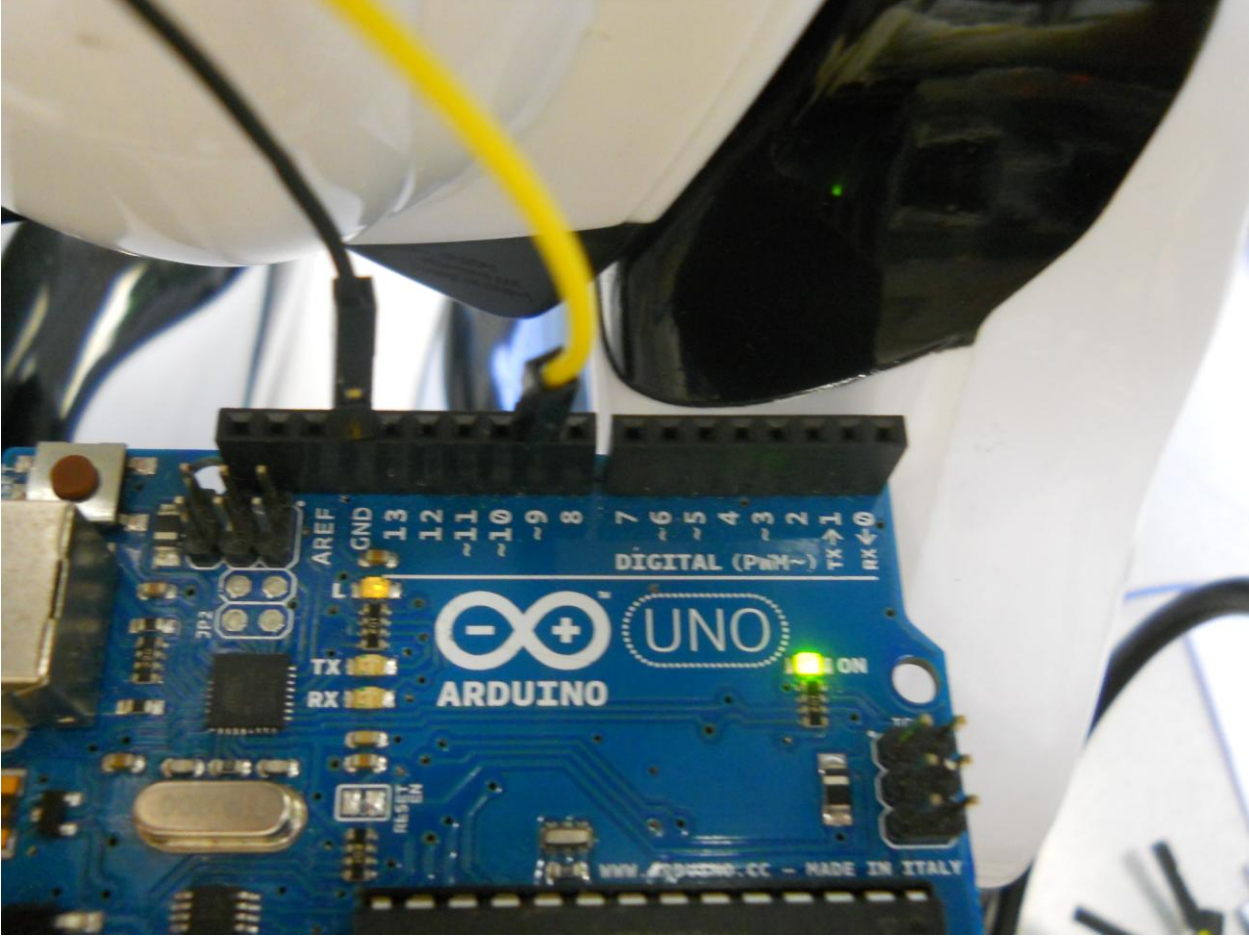












```
pi@raspberrypi: ~/wowiee
File Edit Options Buffers Tools Python Help
#!/usr/bin/python

import serial
import sys

ser = serial.Serial('/dev/ttyACM0', 9600, timeout = 1)
total = len(sys.argv)
cmdargs = str(sys.argv)

if total > 1:
    x = sys.argv[1]
    ser.write(x);
    s = ser.read(100);
#    print s

-UU-:----F1  argControl.py  All L1  (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```





```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ cat /proc/asound/cards  
0 [ALSA ]: bcm2835 - bcm2835 ALSA  
bcm2835 ALSA  
1 [Set ]: USB-Audio - C-Media USB Headphone Set  
C-Media USB Headphone Set at usb-bcm2708_usb-1.4, full spe  
ed  
pi@raspberrypi ~ $ █
```



```

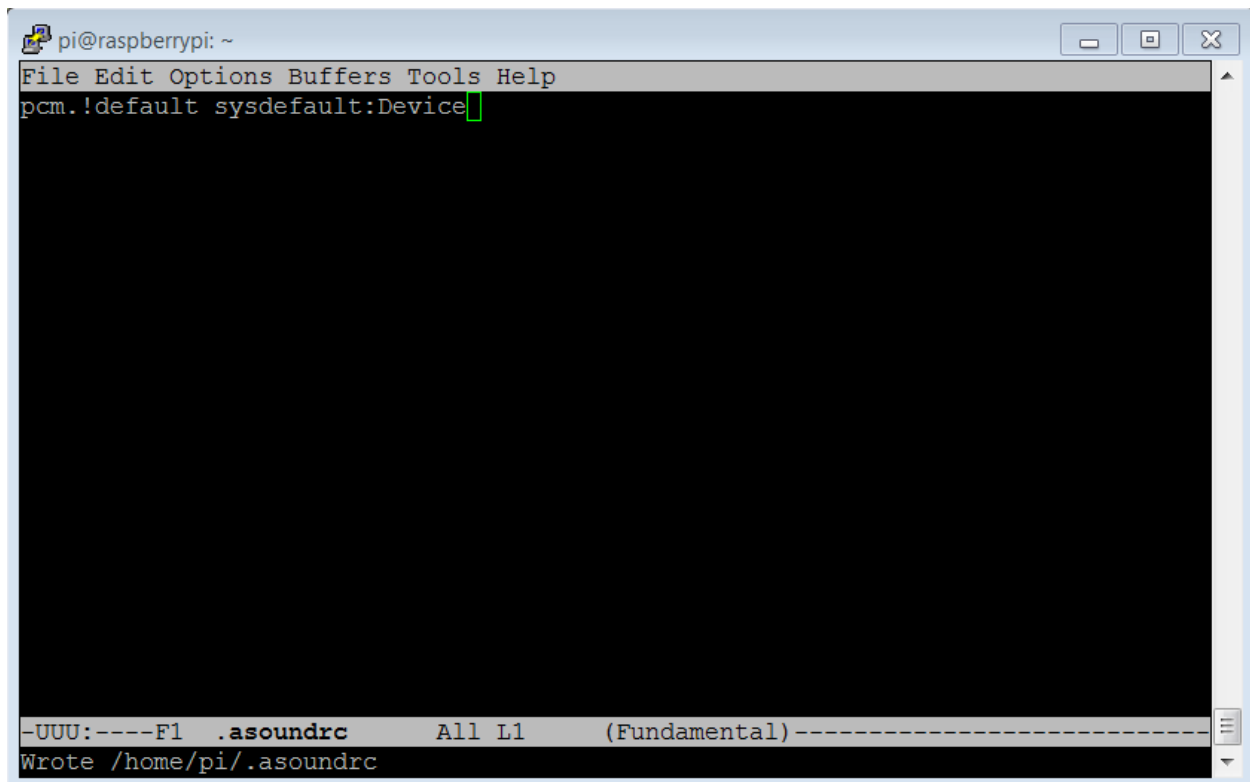
pi@raspberrypi: ~
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq AlsaMixer v1.0.25 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqk ^
x Card: C-Media USB Audio Device F1: Help x
x Chip: USB Mixer F2: System information x
x View: F3:[Playback] F4: Capture F5: All F6: Select sound card x
x Item: Speaker [dB gain: -6.63, -6.63] Esc: Exit x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
x tqqu lqqk x
x xOOx xMMx xOOx x
x mqqj mqqj mqqj x
x 66<>66 52 x
x < Speaker > Mic Auto Gain Control x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq

```

```

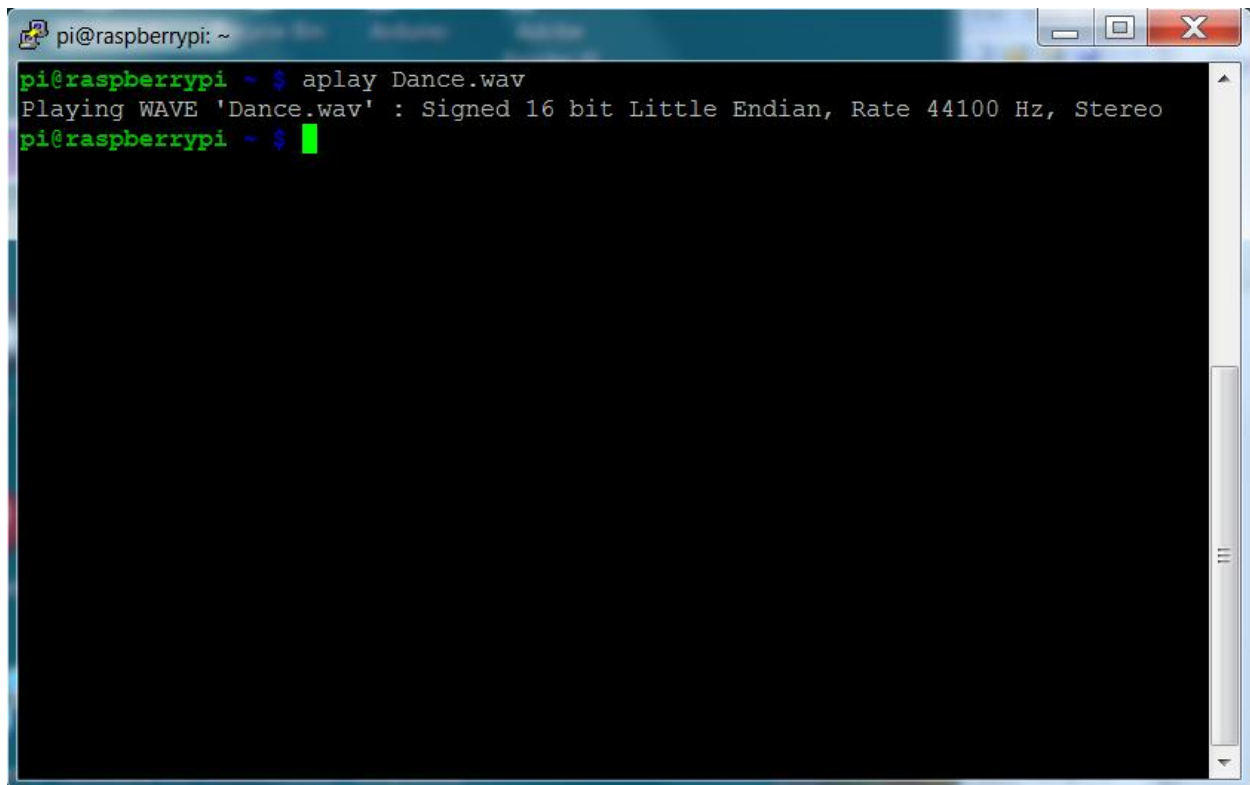
pi@raspberrypi: ~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 8/8
    Subdevice #0: subdevice #0
    Subdevice #1: subdevice #1
    Subdevice #2: subdevice #2
    Subdevice #3: subdevice #3
    Subdevice #4: subdevice #4
    Subdevice #5: subdevice #5
    Subdevice #6: subdevice #6
    Subdevice #7: subdevice #7
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
card 1: Set [C-Media USB Headphone Set], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
pi@raspberrypi ~ $ █

```



A terminal window on a Raspberry Pi. The window title is "pi@raspberrypi: ~". The menu bar contains "File Edit Options Buffers Tools Help". The text in the terminal is "pcm.!default sysdefault:Device" with a green cursor at the end. At the bottom, a status bar shows "-UUU:----F1 .asoundrc All L1 (Fundamental)-----" and "Wrote /home/pi/.asoundrc".

```
pi@raspberrypi: ~  
File Edit Options Buffers Tools Help  
pcm.!default sysdefault:Device  
-UUU:----F1 .asoundrc All L1 (Fundamental)-----  
Wrote /home/pi/.asoundrc
```



A terminal window on a Raspberry Pi. The window title is "pi@raspberrypi: ~". The text in the terminal is "pi@raspberrypi ~ \$ aplay Dance.wav" followed by "Playing WAVE 'Dance.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo" and "pi@raspberrypi ~ \$" with a green cursor at the end.

```
pi@raspberrypi ~ $ aplay Dance.wav  
Playing WAVE 'Dance.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo  
pi@raspberrypi ~ $
```



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ aplay Dance.wav  
Playing WAVE 'Dance.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo  
pi@raspberrypi ~ $ arecord -d 5 -r 48000 test.wav  
Recording WAVE 'test.wav' : Unsigned 8 bit, Rate 48000 Hz, Mono  
pi@raspberrypi ~ $ █
```

```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ ls  
Dance.wav  ocr_pi.png  python_games  test.wav  
Desktop    pocketsphinx-0.8.tar.gz  sphinxbase-0.8.tar.gz  
pi@raspberrypi ~ $ █
```

```
pi@raspberrypi: ~/sphinxbase-0.8
Linux raspberrypi 3.18.11-v7+ #781 SMP PREEMPT Tue Apr 21 18:07:59 BST 2015 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jun 19 20:18:11 2015 from grimmettr.c.byui.edu
pi@raspberrypi ~ $ ls
Dance.wav      pocketsphinx-0.8      sketchbook          start.sh
Desktop        pocketsphinx-0.8.tar.gz sphinxbase-0.8      wowie
hostapd.zip    python_games          sphinxbase-0.8.tar.gz
pi@raspberrypi ~ $ cd sphinxbase-0.8/
pi@raspberrypi ~/sphinxbase-0.8 $ ls
aclocal.m4     config.status  group              Makefile          sphinxbase.pc
AUTHORS        config.sub     include           Makefile.am       sphinxbase.pc.in
autogen.sh     configure     INSTALL           Makefile.in       sphinxbase.sln
ChangeLog      configure.in  install-sh        missing           src
config.guess   COPYING       libtool           NEWS              test
config.log     depcomp       ltmain.sh        python            win32
config.rpath   doc           m4               README            ylwrap
pi@raspberrypi ~/sphinxbase-0.8 $
```

```
pi@raspberrypi: ~
File Edit Options Buffers Tools Conf Help
include /etc/ld.so.conf.d/*.conf
/usr/local/lib
```

```
pi@raspberrypi: ~/pocketsphinx-0.8/src/programs
INFO: ngram_model_dmp.c(288): 436879 = LM.bigrams(+trailer) read
INFO: ngram_model_dmp.c(314): 418286 = LM.trigrams read
INFO: ngram_model_dmp.c(339): 37293 = LM.prob2 entries read
INFO: ngram_model_dmp.c(359): 14370 = LM.bo_wt2 entries read
INFO: ngram_model_dmp.c(379): 36094 = LM.prob3 entries read
INFO: ngram_model_dmp.c(407): 854 = LM.tseg_base entries read
INFO: ngram_model_dmp.c(463): 5001 = ascii word strings read
INFO: ngram_search_fwdtree.c(99): 788 unique initial diphones
INFO: ngram_search_fwdtree.c(147): 0 root, 0 non-root channels, 60 single-phone
words
INFO: ngram_search_fwdtree.c(186): Creating search tree
INFO: ngram_search_fwdtree.c(191): before: 0 root, 0 non-root channels, 60 singl
e-phone words
INFO: ngram_search_fwdtree.c(326): after: max nonroot chan increased to 13428
INFO: ngram_search_fwdtree.c(338): after: 457 root, 13300 non-root channels, 26
single-phone words
INFO: ngram_search_fwdflat.c(156): fwdflat: min_ef_width = 4, max_sf_win = 25
INFO: continuous.c(371): /home/pi/pocketsphinx-0.8/src/programs/.libs/lt-pockets
phinx_continuous COMPILED ON: Nov 8 2013, AT: 18:29:54

Warning: Could not find Mic element
Warning: Could not find Capture element
READY....
█
```

```
pi@raspberrypi: ~
INFO: ngram_model_arpa.c(195): Reading bigrams
INFO: ngram_model_arpa.c(533): 72 = #bigrams created
INFO: ngram_model_arpa.c(534): 14 = #prob2 entries
INFO: ngram_model_arpa.c(542): 7 = #bo_wt2 entries
INFO: ngram_model_arpa.c(292): Reading trigrams
INFO: ngram_model_arpa.c(555): 62 = #trigrams created
INFO: ngram_model_arpa.c(556): 8 = #prob3 entries
INFO: ngram_search_fwdtree.c(99): 40 unique initial diphones
INFO: ngram_search_fwdtree.c(147): 0 root, 0 non-root channels, 12 single-phone
words
INFO: ngram_search_fwdtree.c(186): Creating search tree
INFO: ngram_search_fwdtree.c(191): before: 0 root, 0 non-root channels, 12 singl
e-phone words
INFO: ngram_search_fwdtree.c(326): after: max nonroot chan increased to 194
INFO: ngram_search_fwdtree.c(338): after: 40 root, 66 non-root channels, 11 singl
e-phone words
INFO: ngram_search_fwdflat.c(156): fwdflat: min_ef_width = 4, max_sf_win = 25
INFO: continuous.c(427): /home/pi/pocketsphinx-0.8/src/programs/.libs/lt-pockets
phinx_continuous COMPILED ON: Jun 19 2015, AT: 08:53:39

Warning: Could not find Mic element
Warning: Could not find Capture element
READY....
█
```

```
pi@raspberrypi: ~/pocketsphinx-0.8/src/programs
File Edit Options Buffers Tools C Help
ps_end_utt(ps);
hyp = ps_get_hyp(ps, NULL, &uttid);
printf("%s: %s\n", uttid, hyp);
fflush(stdout);

/* Exit if the first word spoken was GOODBYE */
if (hyp) {
    sscanf(hyp, "%s", word);
    if (strcmp(word, "goodbye") == 0)
        break;
}

/* Resume A/D recording for next utterance */
if (ad_start_rec(ad) < 0)
    E_FATAL("Failed to start recording\n");
}

cont_ad_close(cont);
ad_close(ad);
}
-UU-:----F1 continuous.c 82% L331 (C/l Abbrev)-----
```

```
pi@raspberrypi: ~/pocketsphinx-0.8/src/programs
File Edit Options Buffers Tools C Help
/* Finish decoding, obtain and print result */
ps_end_utt(ps);
hyp = ps_get_hyp(ps, NULL, &uttid);
printf("%s: %s\n", uttid, hyp);
fflush(stdout);

/* Exit if the first word spoken was GOODBYE */
if (hyp) {
    sscanf(hyp, "%s", word);
    if (strcmp(hyp, "good bye") == 0)
    {
        system("espeak \"good bye\"");
        break;
    }
    else if (strcmp(hyp, "hello") == 0)
    {
        system("espeak \"hello\"");
    }
}

}
-UU-:----F1 continuous.c 80% L330 (C/l Abbrev)-----
```

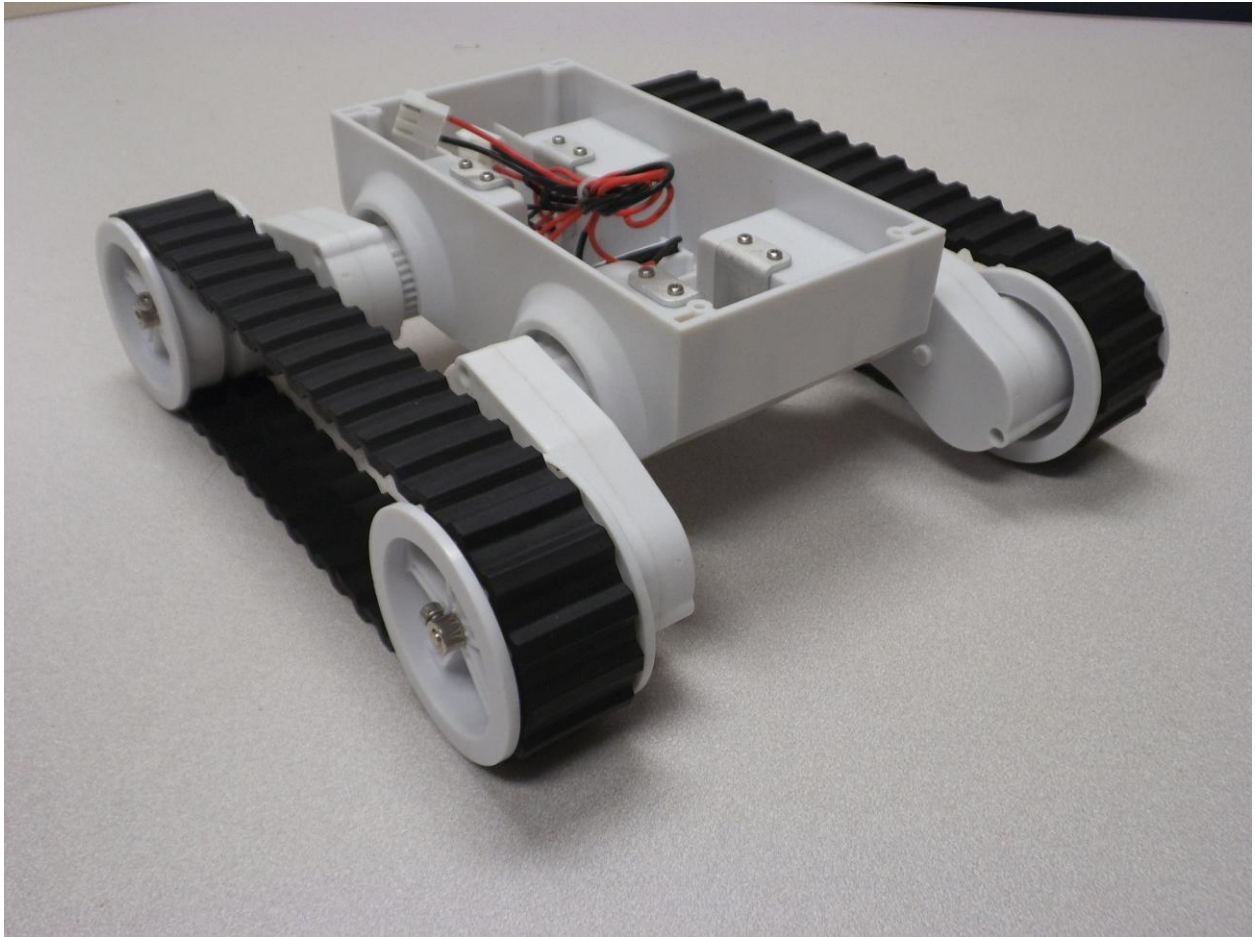


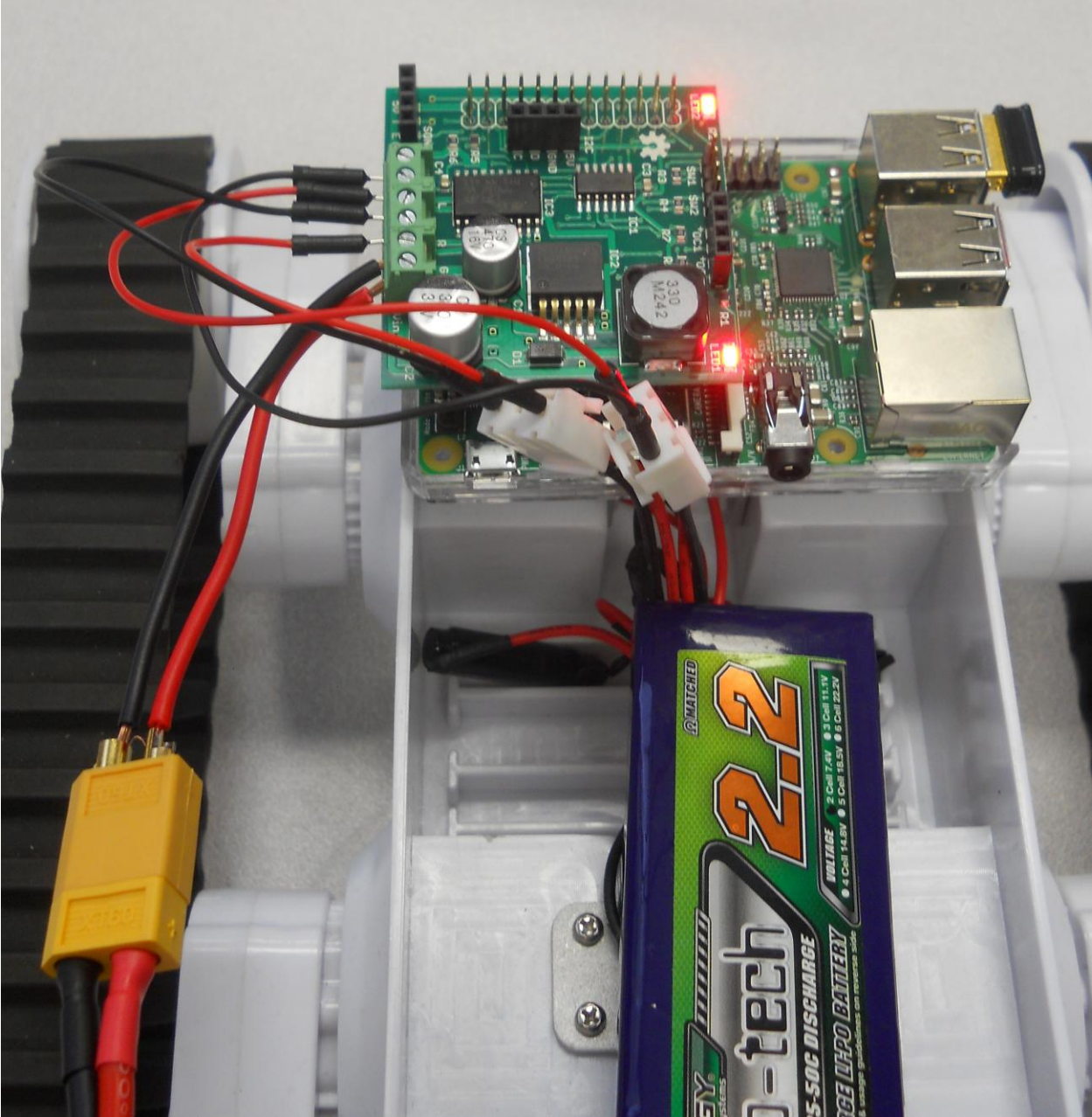
```
pi@raspberrypi: ~/pocketsphinx-0.8/src/programs
File Edit Options Buffers Tools C Help
fflush(stdout);

/* Exit if the first word spoken was GOODBYE */
if (hyp) {
    sscanf(hyp, "%s", word);
    if (strcmp(hyp, "good bye") == 0)
    {
        system("espeak \"good bye\"");
        break;
    }
    else if (strcmp(hyp, "hello") == 0)
    {
        system("espeak \"hello\"");
    }
    else if (strcmp(hyp, "roar") == 0)
    {
        system("espeak \"Roaring\"");
        system("sudo /home/pi/wowee/argControl.py A");
    }
    else if (strcmp(hyp, "hi five") == 0)
    {
        system("espeak \"hi five\"");
        system("sudo /home/pi/wowee/argControl.py u");
    }
}

/* Resume A/D recording for next utterance */
-UU-: **--F1 continuous.c 78% L344 (C/l Abbrev)
```

Chapter 3: Building a Tracked Vehicle That Can Plan Its Own Path





```
pi@raspberrypi: ~/xmod
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
from rrb2 import *
import tty
import sys
import termios
def getch():
    fd = sys.stdin_FILENO()
    old_settings = termios.tcgetattr(fd)
    tty.setraw(sys.stdin_FILENO())
    ch = sys.stdin.read(1)
    termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch
pwmPin = 18
dc = 10
GPIO.setmode(GPIO.BCM)
GPIO.setup(pwmPin, GPIO.OUT)
pwm = GPIO.PWM(pwmPin, 320)
rr = RRB2()
pwm.start(dc)
rr.set_led1(1)
var = 'n'
speed1 = 0
speed2 = 0
direction1 = 1
direction2 = 1
while var != 'q':
    var = getch()
    if var == 'l':
-UU-: **--F1 xmodControl.py Top L1 (Python)-----
```



```
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help

while var != 'q':
    var = getch()
    if var == 'l':
        speed1 = 1
        direction1 = 1
        speed2 = 1
        direction2 = 0
        stop = 1
    if var == 'r':
        speed1 = 1
        direction1 = 0
        speed2 = 1
        direction2 = 1
        stop = 1
    if var == 'f':
        speed1 = 1
        direction1 = 1
        speed2 = 1
        direction2 = 1
        stop = 0
    if var == 'b':
        speed1 = 1
        direction1 = 0
        speed2 = 1
        direction2 = 0
        stop = 0
    if var == 's':
        speed1 = 0
        direction1 = 0
        speed2 = 0
        direction2 = 0
    rr.set_motors(speed1, direction1, speed2, direction2)
    if stop == 1:
        time.sleep(1)
        rr.set_motors(0, 0, 0, 0)
GPIO.cleanup()

-UU-:----F1 track.py Bot L40 (Python)-----
```

```
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
from rrb2 import *

rr = RRB2()

def init_vehicle():
    rr.set_led1(1)

def turn_left(angle):
    rr.set_motors(1, 1, 1, 0)
    time.sleep(angle/20)
    rr.set_motors(0, 0, 0, 0)

def turn_right(angle):
    rr.set_motors(1, 0, 1, 1)
    time.sleep(angle/20)
    rr.set_motors(0, 0, 0, 0)

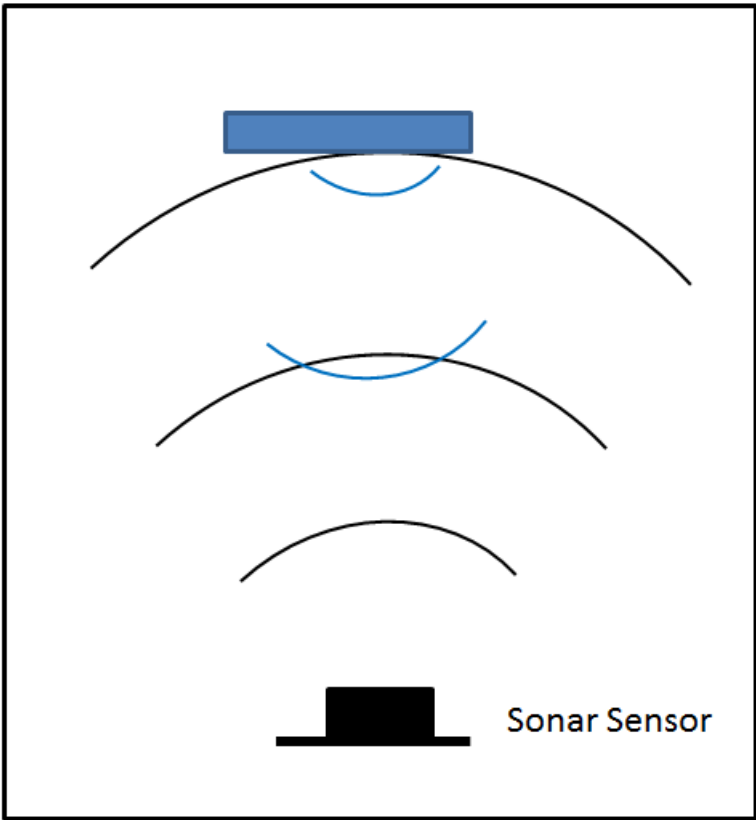
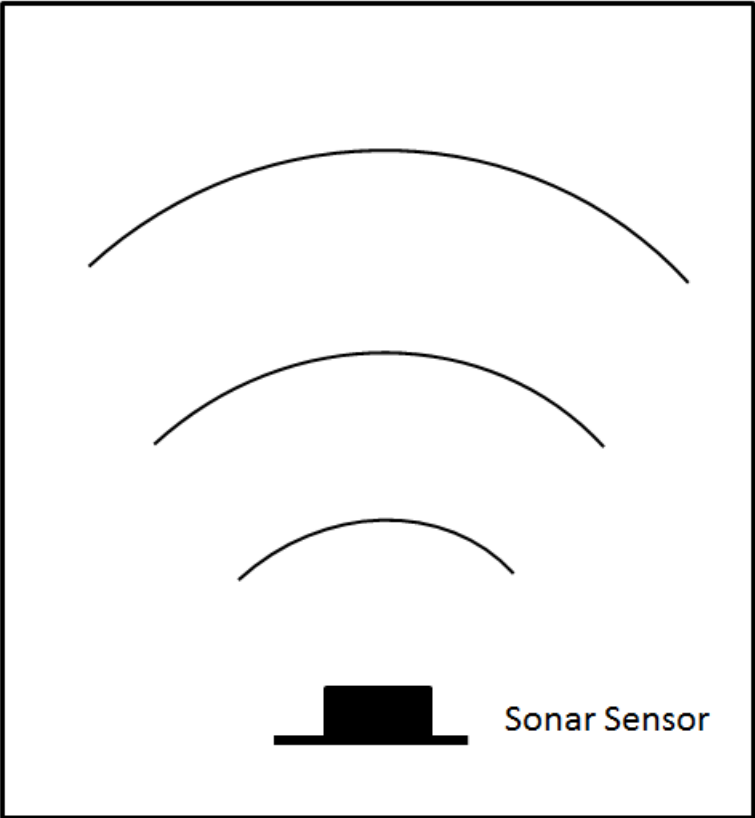
def forward(value):
    rr.set_motors(1, 1, 1, 1)
    time.sleep(value)
    rr.set_motors(0, 0, 0, 0)

def backward(value):
    rr.set_motors(1, 0, 1, 0)
    time.sleep(value)
    rr.set_motors(0, 0, 0, 0)

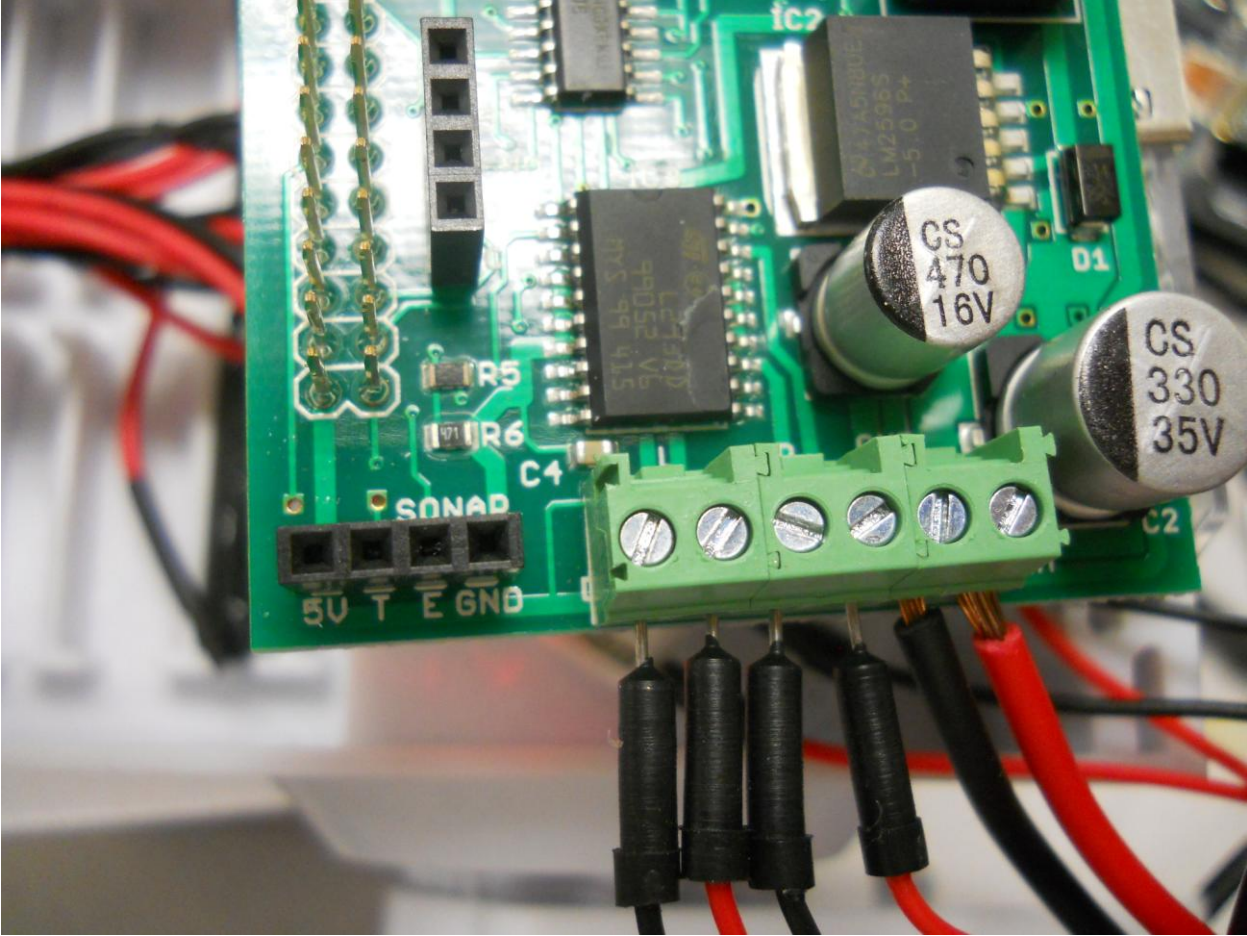
def stop():
    rr.set_motors(0, 0, 0, 0)

def cleanup():
    GPIO.cleanup()

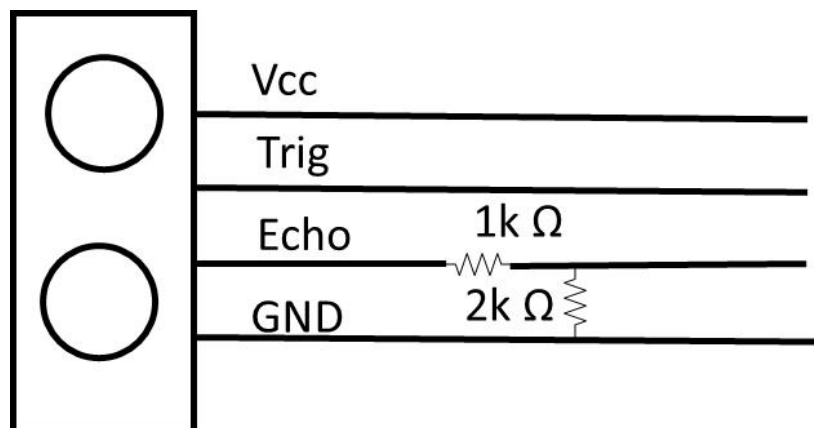
-UU-:----F1 track.py All L1 (Python)-----
```

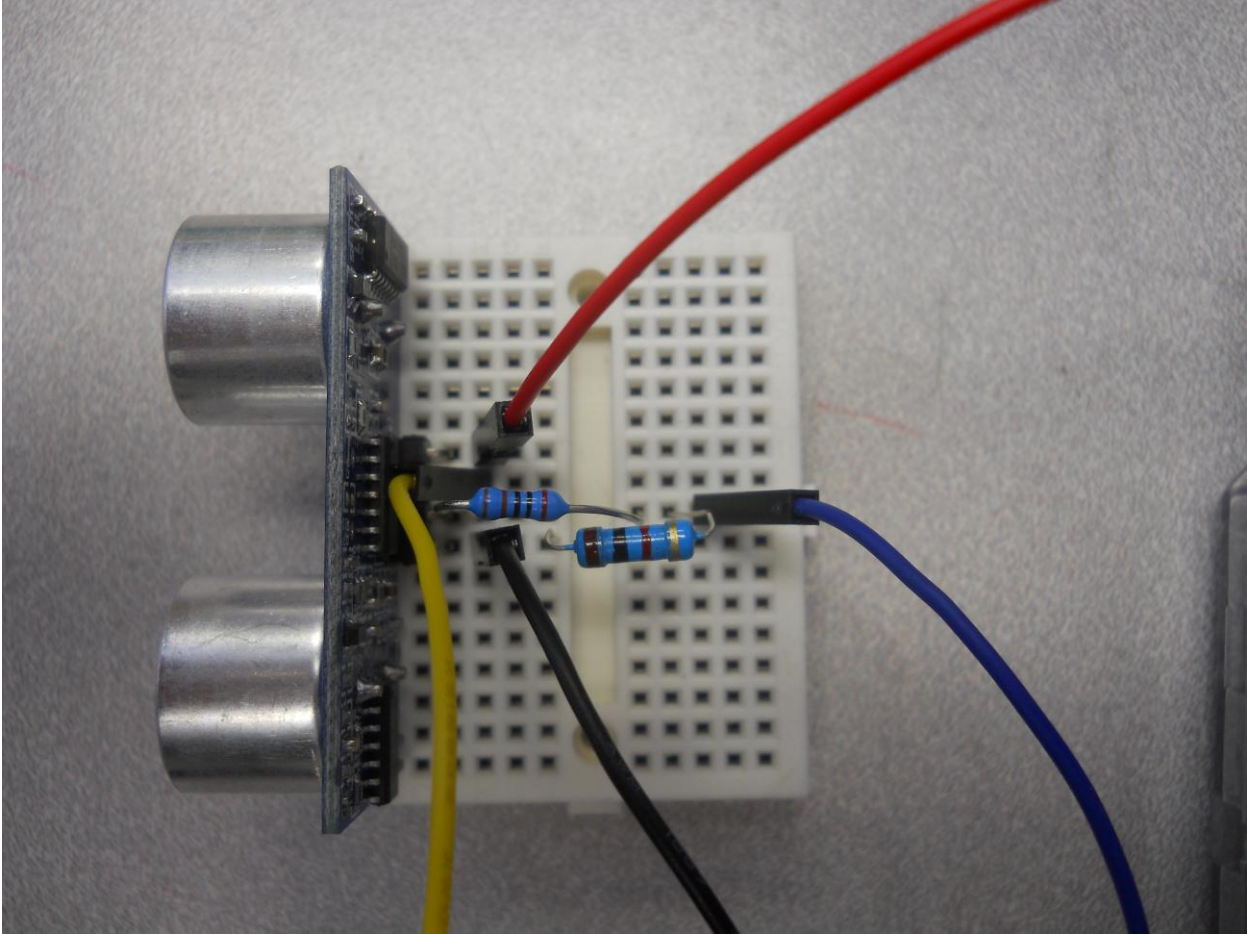


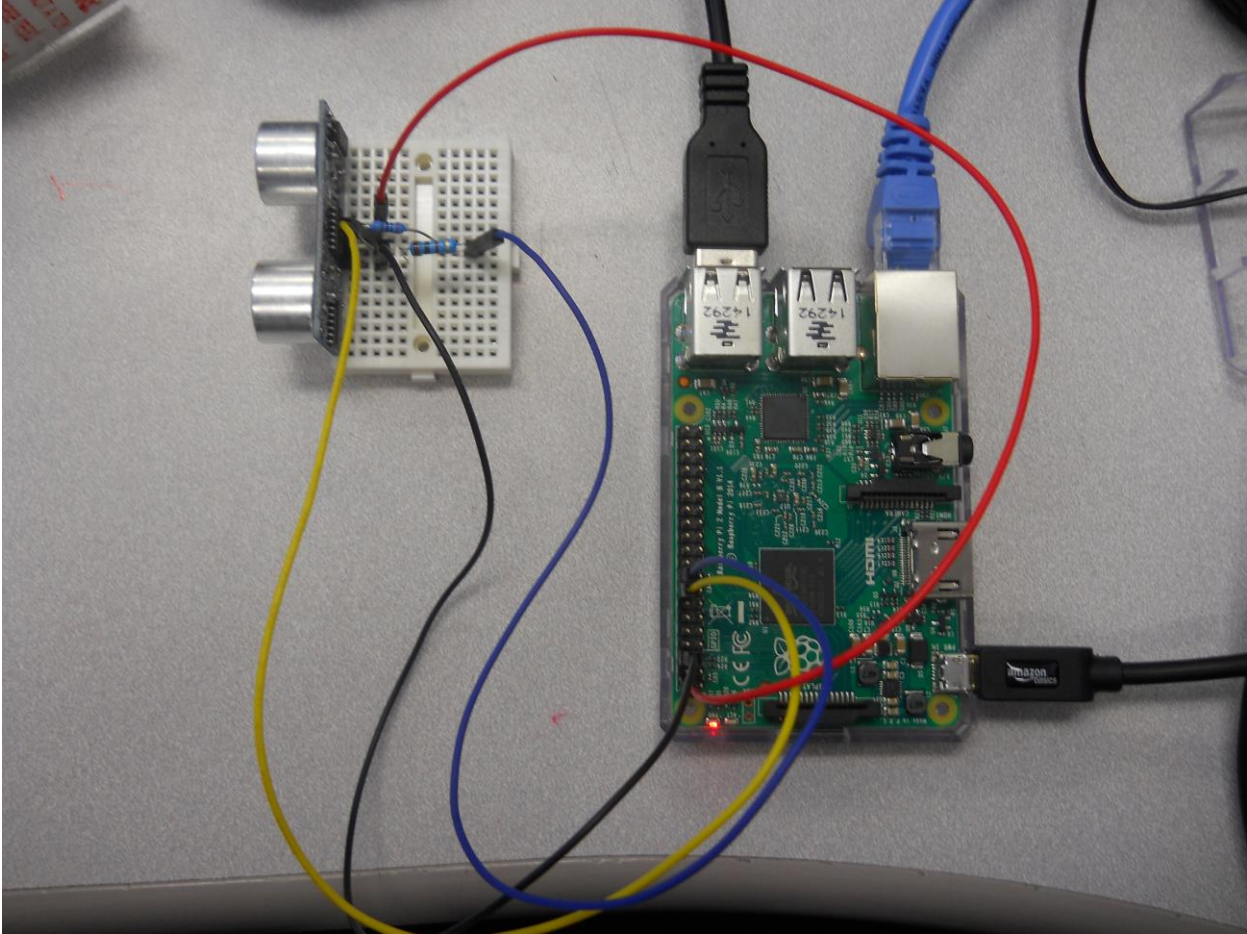




Pin 1 3.3V	<input type="checkbox"/> ○	Pin 2 5V
Pin 3 GPIO2	○ ○	Pin 4 5V
Pin 5 GPIO3	○ ○	Pin 6 GND
Pin 7 GPIO4	○ ○	Pin 8 GPIO14
Pin 9 GND	○ ○	Pin 10 GPIO15
Pin 11 GPIO17	○ ○	Pin 12 GPIO18
Pin 13 GPIO27	○ ○	Pin 14 GND
Pin 15 GPIO22	○ ○	Pin 16 GPIO23
Pin 17 3.3V	○ ○	Pin 18 GPIO24
Pin 19 GPIO10	○ ○	Pin 20 GND
Pin 21 GPIO9	○ ○	Pin 22 GPIO25
Pin 23 GPIO11	○ ○	Pin 24 GPIO8
Pin 25 GND	○ ○	Pin 26 GPIO7
Pin 27 ID_SD	○ ○	Pin 28 ID_SC
Pin 29 GPIO5	○ ○	Pin 30 GND
Pin 31 GPIO6	○ ○	Pin 32 GPIO12
Pin 33 GPIO13	○ ○	Pin 34 GND
Pin 35 GPIO19	○ ○	Pin 36 GPIO16
Pin 37 GPIO26	○ ○	Pin 38 GPIO20
Pin 39 GND	○ ○	Pin 40 GPIO21








```
pi@raspberrypi: ~
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)

trig_pin = 23
echo_pin = 24
GPIO.setup(trig_pin,GPIO.OUT)
GPIO.setup(echo_pin,GPIO.IN)

GPIO.output(trig_pin, False)
print "Waiting to settle"
time.sleep(1)
GPIO.output(trig_pin, True)
time.sleep(0.00001)
GPIO.output(trig_pin, False)

while GPIO.input(echo_pin)==0:
    start = time.time()

while GPIO.input(echo_pin)==1:
    end = time.time()

duration = end - start
distance = duration * 17150
distance = round(distance, 2)
print "Distance:",distance,"cm"
GPIO.cleanup()

-UU-:----F1  sonar_sensor.py  All L1  (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```

```
pi@raspberrypi: ~
pi@raspberrypi ~ $ sudo python sonar_sensor.py
Waiting to settle
Distance: 21.23 cm
pi@raspberrypi ~ $
```



```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)

def getDistance():
    trig_pin = 23
    echo_pin = 24
    GPIO.setup(trig_pin,GPIO.OUT)
    GPIO.setup(echo_pin,GPIO.IN)

    GPIO.output(trig_pin, False)
    time.sleep(1)
    GPIO.output(trig_pin, True)
    time.sleep(0.00001)
    GPIO.output(trig_pin, False)

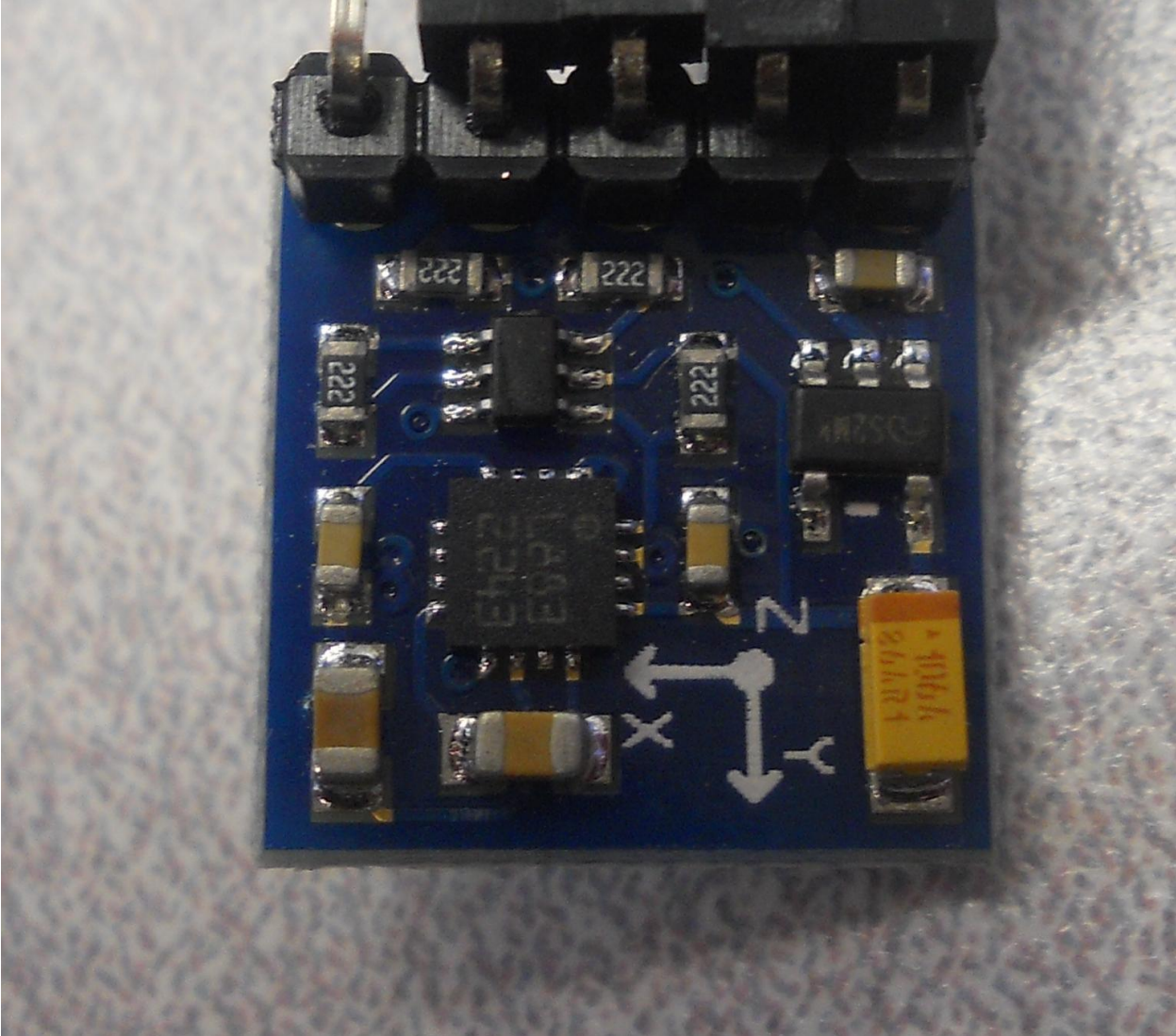
    while GPIO.input(echo_pin)==0:
        start = time.time()

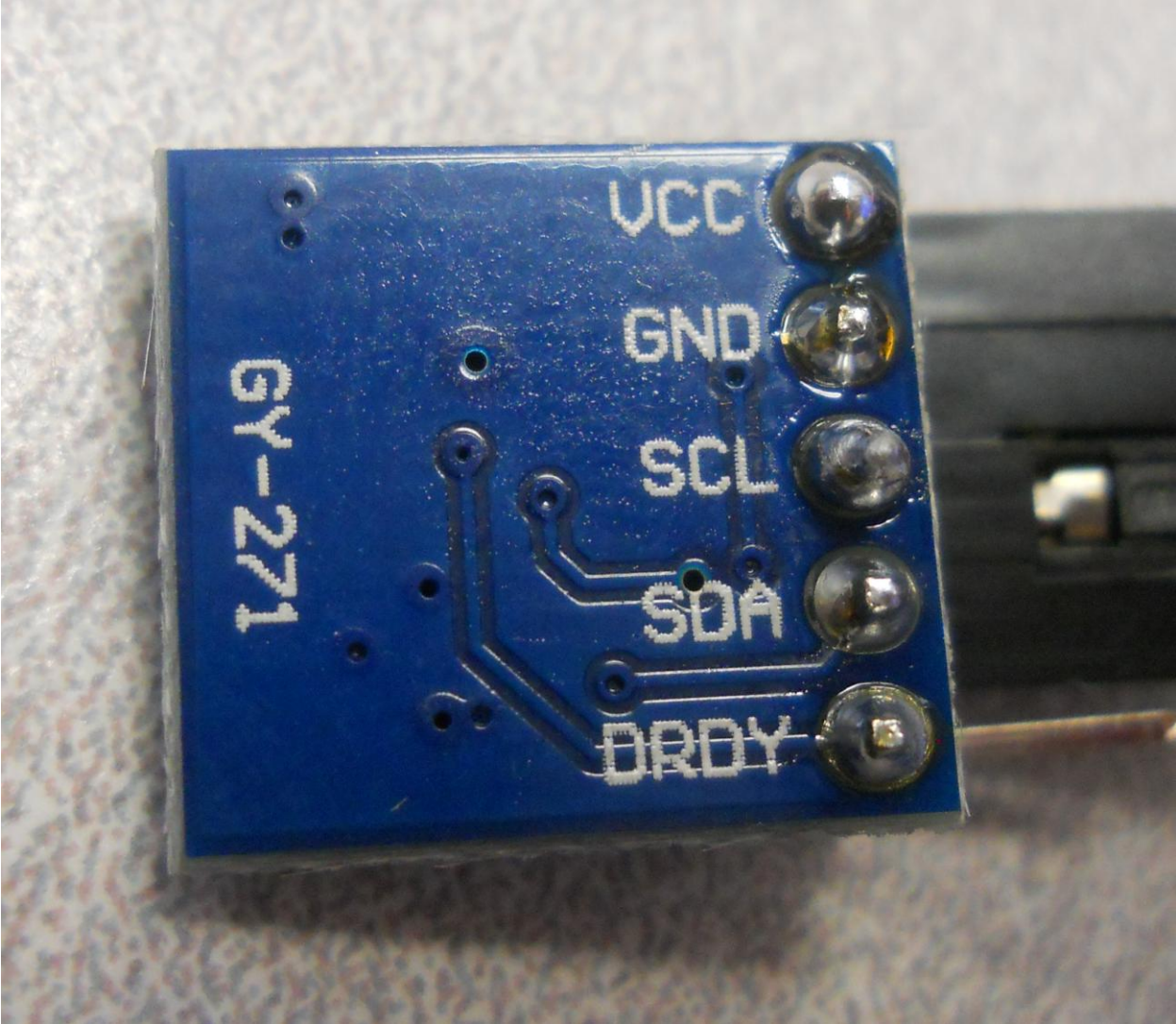
    while GPIO.input(echo_pin)==1:
        end = time.time()

    duration = end - start
    distance = duration * 17150
    distance = round(distance, 2)
    GPIO.cleanup()
    return distance

print "Distance: ", getDistance(), "cm"

-UU-:----F1 sonar_sensor.py All L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```





GY-271

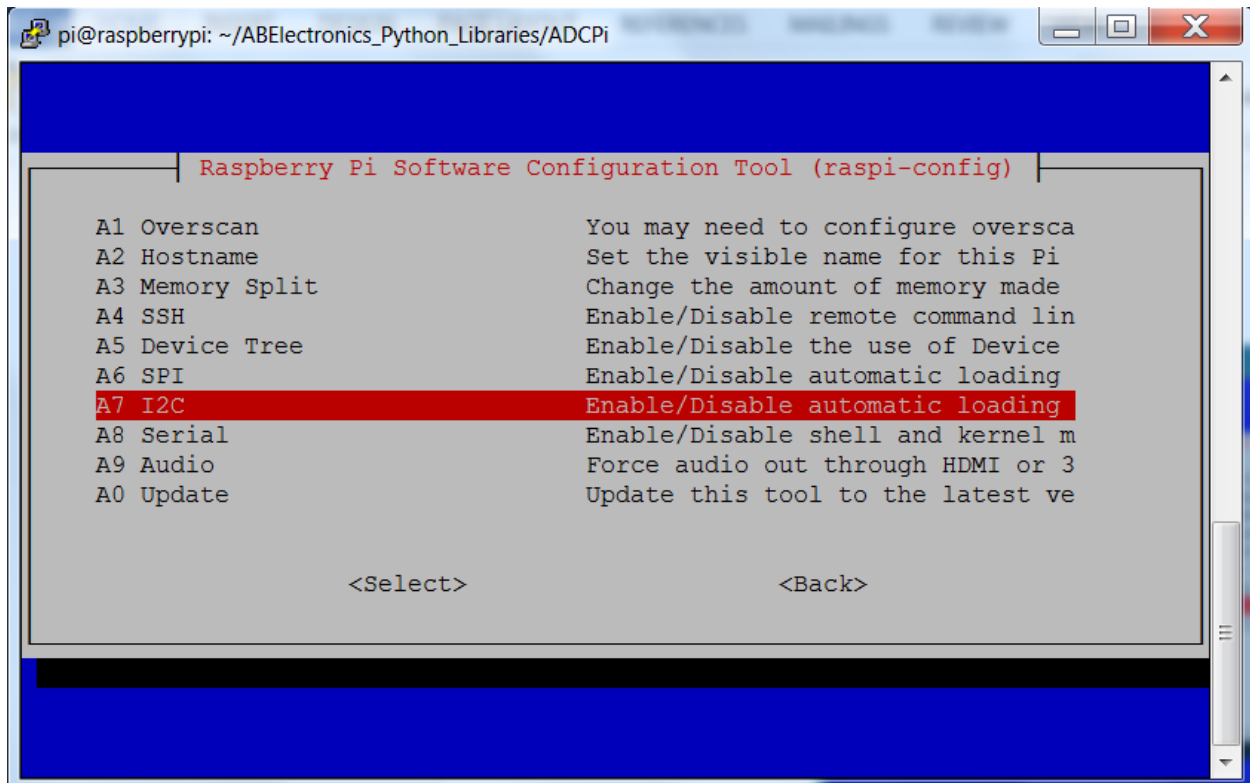
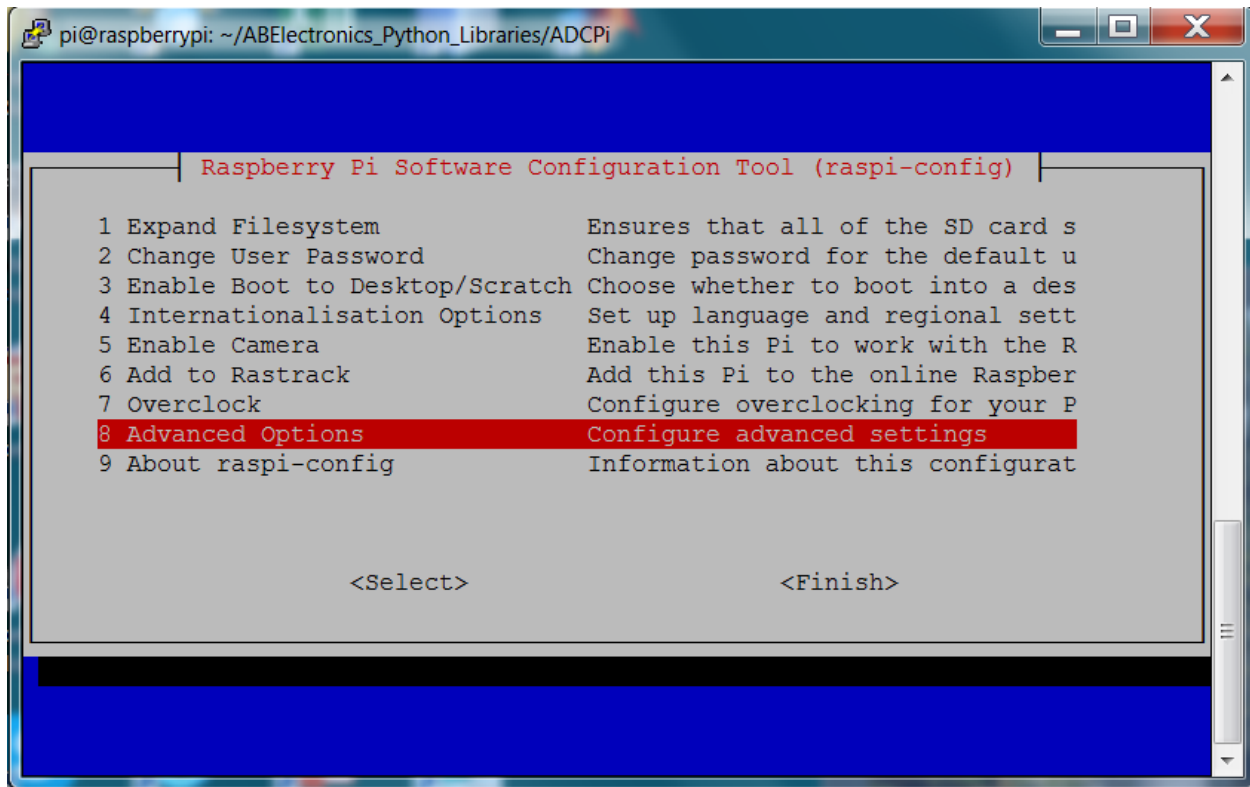
VCC

GND

SCL

SDA

DRDY



```
pi@raspberrypi: ~/ABElectronics_Python_Libraries/ADCPi
File Edit Options Buffers Tools Help
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-bcm2708
i2c-dev
-WU-:----F1 modules All I9 (Fundamental)-----
Wrote /etc/modules
```

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- 1e --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypi ~ $
```

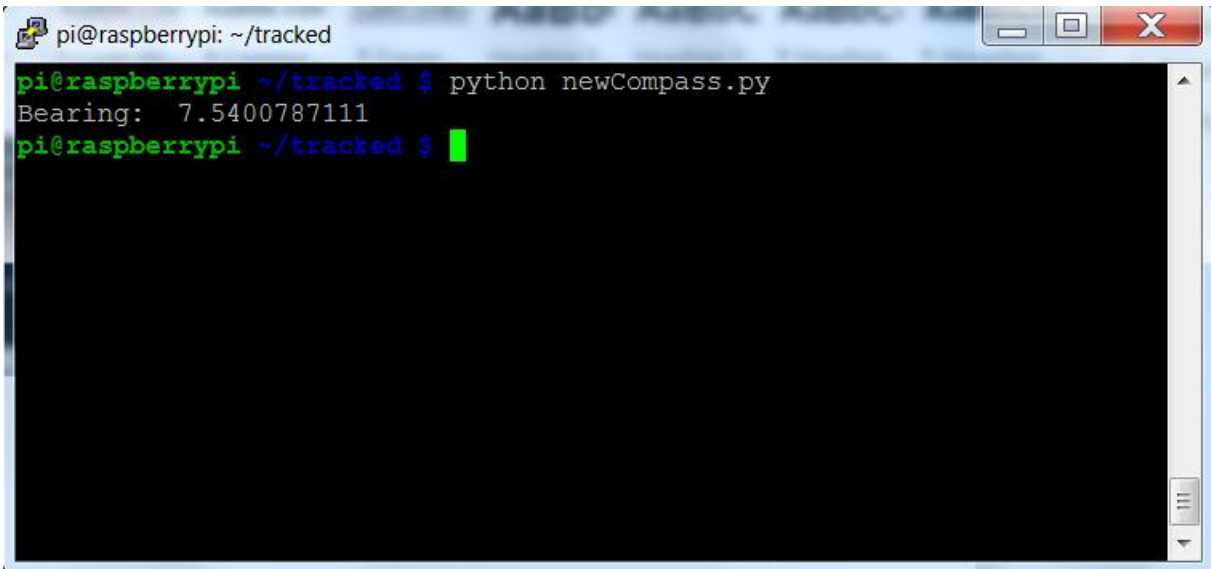


```
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import smbus
import time
import math
bus = smbus.SMBus(1)
address = 0x1e

def read_byte(adr):
    return bus.read_byte_data(address, adr)
def read_word(adr):
    high = bus.read_byte_data(address, adr)
    low = bus.read_byte_data(address, adr+1)
    val = (high << 8) + low
    return val
def read_word_2c(adr):
    val = read_word(adr)
    if (val >= 0x8000):
        return -((65535 - val) + 1)
    else:
        return val
def write_byte(adr, value):
    bus.write_byte_data(address, adr, value)

write_byte(0, 0b01110000) # Set to 8 samples @ 15Hz
write_byte(1, 0b00100000) # 1.3 gain LSB / Gauss 1090 (default)
write_byte(2, 0b00000000) # Continuous sampling
scale = 0.92
x_out = read_word_2c(3) * scale
y_out = read_word_2c(7) * scale
z_out = read_word_2c(5) * scale
bearing = math.atan2(y_out, x_out)
if (bearing < 0):
    bearing += 2 * math.pi
print "Bearing: ", math.degrees(bearing)

-UU-:**--F1 newCompass.py All L28 (Python)
```



A terminal window titled "pi@raspberrypi: ~/tracked" with standard window controls (minimize, maximize, close) in the top right corner. The terminal shows the following text:

```
pi@raspberrypi ~/tracked $ python newCompass.py
Bearing: 7.5400787111
pi@raspberrypi ~/tracked $ █
```

The prompt "pi@raspberrypi" is shown in green, the command "python newCompass.py" in blue, and the output "Bearing: 7.5400787111" in white. A green cursor is visible on the second prompt line.

```
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import smbus
import time
import math
bus = smbus.SMBus(1)
address = 0x1e

def read_byte(adr):
    return bus.read_byte_data(address, adr)
def read_word(adr):
    high = bus.read_byte_data(address, adr)
    low = bus.read_byte_data(address, adr+1)
    val = (high << 8) + low
    return val
def read_word_2c(adr):
    val = read_word(adr)
    if (val >= 0x8000):
        return -((65535 - val) + 1)
    else:
        return val
def write_byte(adr, value):
    bus.write_byte_data(address, adr, value)

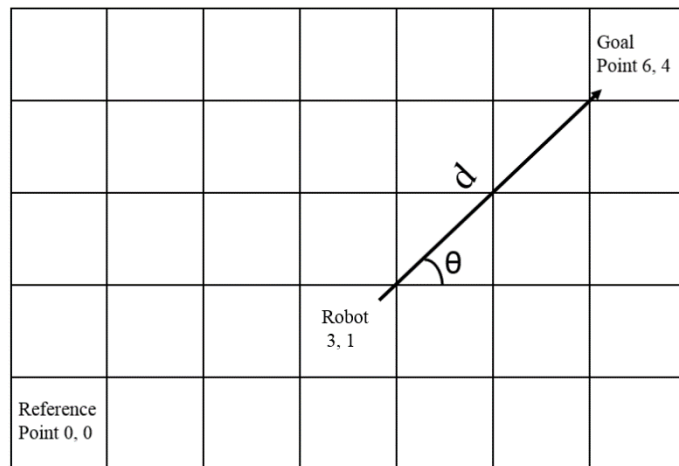
def readDirection():
    write_byte(0, 0b01110000) # Set to 8 samples @ 15Hz
    write_byte(1, 0b00100000) # 1.3 gain LSb / Gauss 1090 (default)
    write_byte(2, 0b00000000) # Continuous sampling
    scale = 0.92
    x_out = read_word_2c(3) * scale
    y_out = read_word_2c(7) * scale
    z_out = read_word_2c(5) * scale
    bearing = math.atan2(y_out, x_out)
    if (bearing < 0):
        bearing += 2 * math.pi
    print "Bearing: ", math.degrees(bearing)
    return math.degrees(bearing)

-UU-:----F1 libCompass.py All L1 (Python)-----
```

						Goal Point 6, 4
			Robot 3, 1			
Reference Point 0, 0						

$$d = \sqrt{((X_{goal} - X_{robot})^2 + (Y_{goal} - Y_{robot})^2)}$$

$$\theta = \arctan\left(\frac{Y_{goal} - Y_{robot}}{X_{goal} - X_{robot}}\right)$$



```

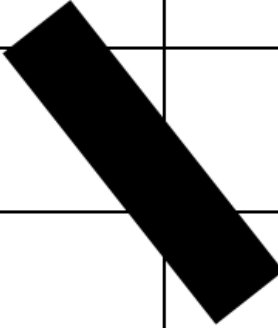
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import time
from track import *
import math

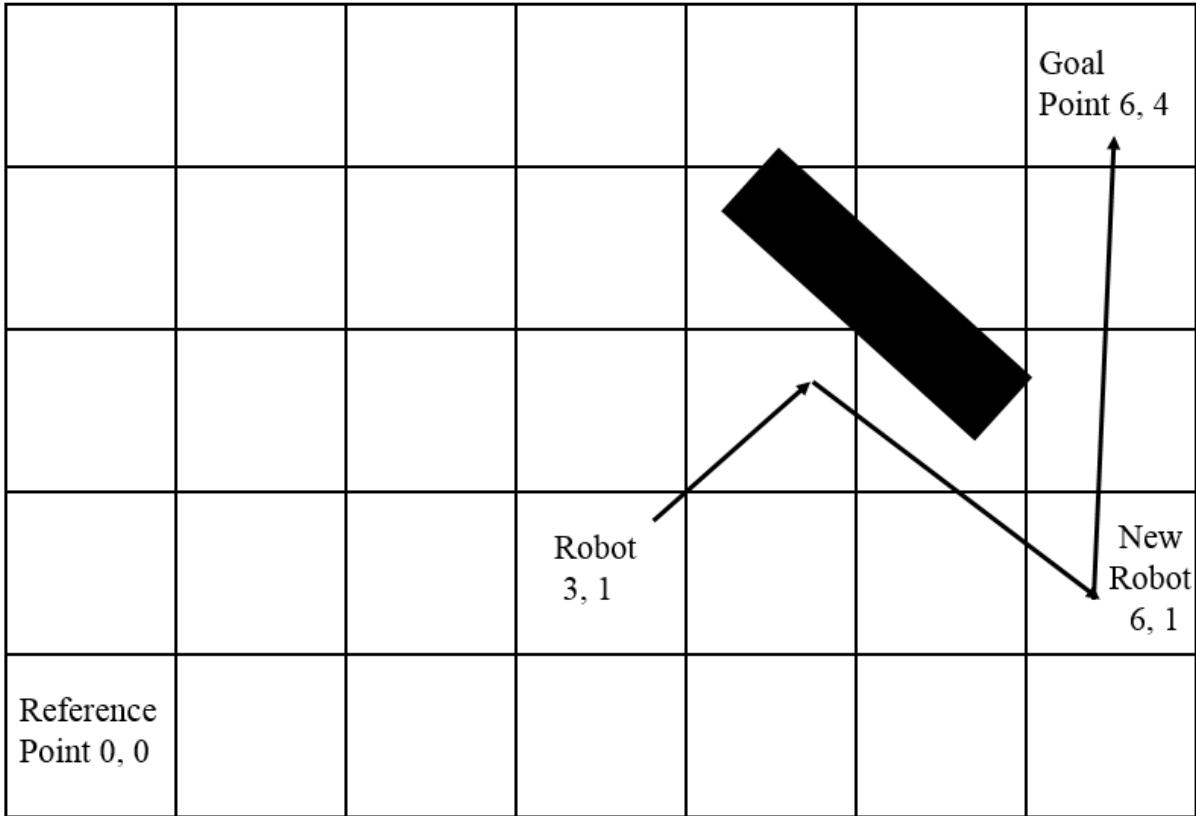
xpos_robot = int(raw_input("Robot X Position: "))
ypos_robot = int(raw_input("Robot Y Position: "))
xpos_goal = int(raw_input("Goal X Position: "))
ypos_goal = int(raw_input("Goal Y Position: "))

distance = math.sqrt((xpos_goal - xpos_robot)**2 + (ypos_goal - ypos_robot)**2)
angle = round(math.degrees(math.atan2((ypos_goal - ypos_robot), (xpos_goal - xpos_robot))))
if angle < 0:
    angle = angle + 360
print (angle)
# Turn to the right bearing
if (angle) < 180:
    turn_right(angle)
else:
    turn_left(angle)
print (distance)
forward(distance)

-UU-:----F1 robotGoal.py All L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.

```

						Goal Point 6, 4
						
			Robot 3, 1			
Reference Point 0, 0						



```
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
from rrb2 import *

rr = RRB2()

def init_vehicle():
    rr.set_led1(1)

def turn_left():
    rr.set_motors(1, 1, 1, 0)

def turn_right():
    rr.set_motors(1, 0, 1, 1)

def forward():
    rr.set_motors(1, 1, 1, 1)

def backward():
    rr.set_motors(1, 0, 1, 0)

def stop():
    rr.set_motors(0, 0, 0, 0)

def cleanup():
    GPIO.cleanup()

-UU-:----F1 track.py All L5 (Python)-----
```

```
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time
from track import *
from libCompass import *
from rrb2 import *
import math

def move_angle(angle):
    if angle < 0:
        angle = angle + 360
    bearing = readDirection()
    move_angle = bearing - angle
    if move_angle > 180:
        turn_right()
    elif move_angle < -180:
        turn_left()
    elif (move_angle) < 180 and move_angle > 0:
        turn_left()
    elif move_angle > -180 and move_angle < 0:
        turn_right()
    while(abs(angle - bearing)) > 5:
        time.sleep(.2)
        print abs(angle-bearing)
        bearing = readDirection()
    stop()
#    print "angle", bearing

def positionRobot(xpos, ypos, xpos_goal, ypos_goal):
    print xpos, ypos, xpos_goal, ypos_goal
    distance = math.sqrt((xpos_goal - ypos_robot)**2 + (ypos_goal - ypos_robot)\
**2)
    angle = round(math.degrees(math.atan2((ypos_goal - ypos_robot), (xpos_goal \
- xpos_robot))))
    print "angle",angle
    move_angle(angle)
    print distance
    return distance, angle

xpos_robot = int(raw_input("Robot X Position: "))
-UU-:-----F1 robotBarrier.py Top L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```

```
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help

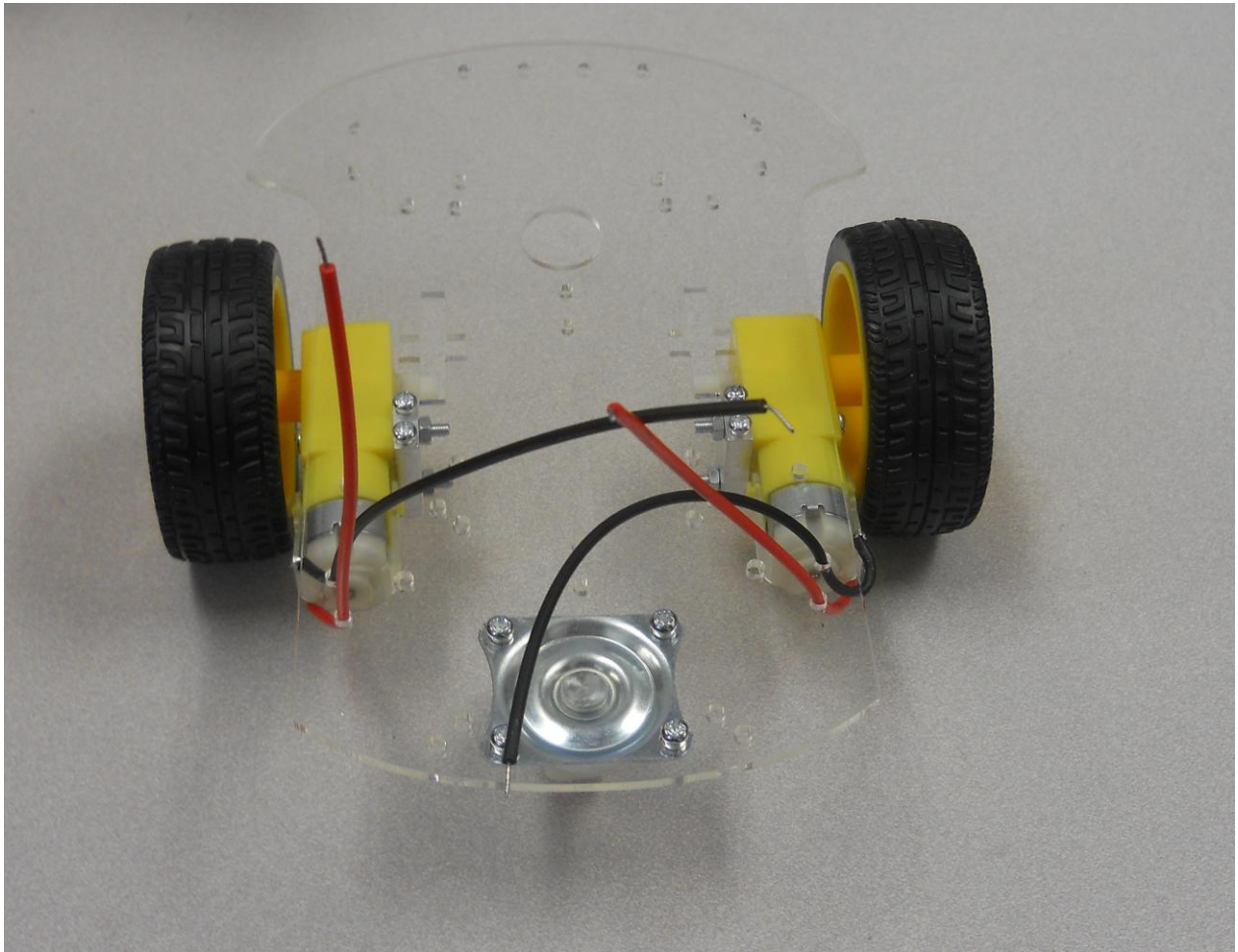
xpos_robot = int(raw_input("Robot X Position: "))
ypos_robot = int(raw_input("Robot Y Position: "))
xpos_goal = int(raw_input("Goal X Position: "))
ypos_goal = int(raw_input("Goal Y Position: "))

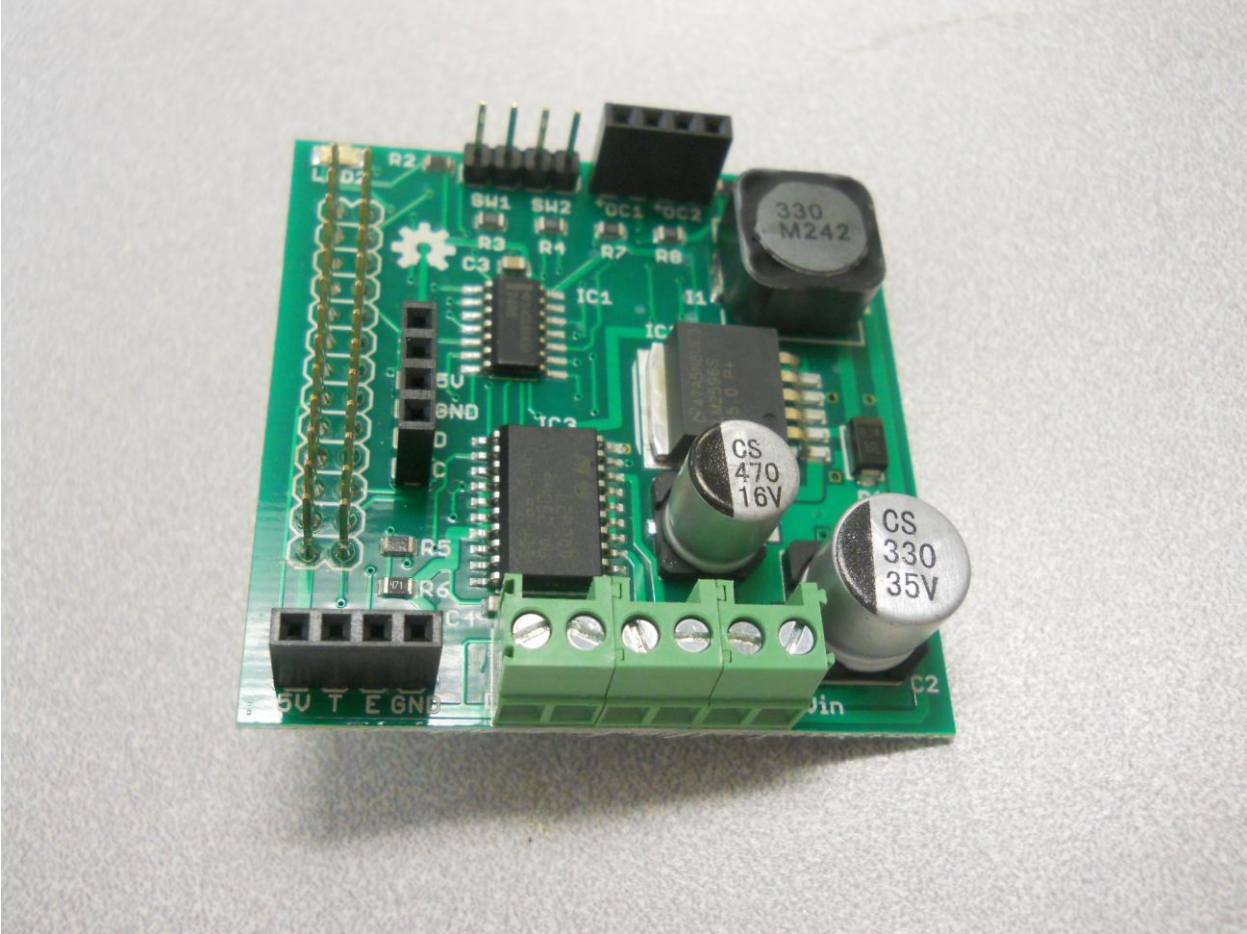
distance, angle = positionRobot(xpos_robot, ypos_robot, xpos_goal, ypos_goal)

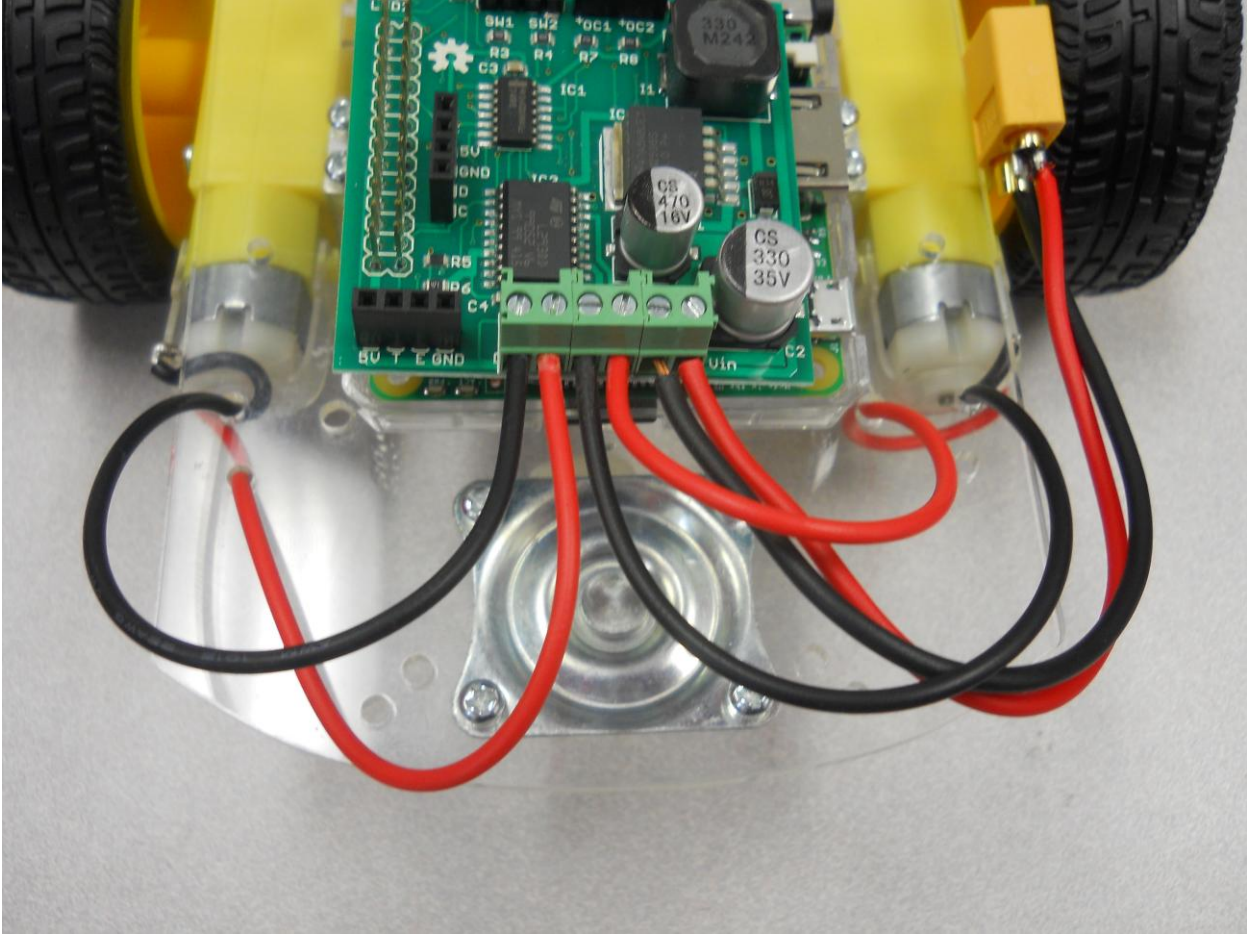
start_time = time.time()
forward()
barrier = rr.get_distance()
elapsed_time = 0

while barrier > 10 and elapsed_time < distance:
    elapsed_time = time.time() - start_time
    barrier = rr.get_distance()
    if barrier > 0 and barrier < 10:
        print "barrier", barrier
        distance traveled = elapsed_time
        new_distance = 1
        ypos_robot = ypos_robot + distance traveled * math.sin(math.radians(angle))
        ypos_goal_barrier = ypos_robot + new_distance * math.sin(math.radians(angle + 90))
        xpos_robot = xpos_robot + distance traveled * math.cos(math.radians(angle))
        xpos_goal_barrier = xpos_robot + new_distance * math.cos(math.radians(angle + 90))
        distance = positionRobot(xpos_robot, ypos_robot, xpos_goal_barrier, ypos_goal_barrier)
        start_time = time.time()
        forward()
        elapsed_time = 0
        while elapsed_time < new_distance:
            elapsed_time = time.time() - start_time
            print "Done moving around barrier"
            ypos_robot = ypos_goal_barrier
            xpos_robot = xpos_goal_barrier
            distance = positionRobot(xpos_robot, ypos_robot, xpos_goal, ypos_goal)
            start_time = time.time()
            forward()
            barrier = rr.get_distance()
            elapsed_time = 0
    stop()
print "Goal Reached"
-UU-:----F1  robotBarrier.py  40% L57  (Python)-----
```

Chapter 4: Building a Robot That Can Play Laser Tag







```
pi@raspberrypi: ~/tracked
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
from rrb2 import *

rr = RRB2()

def init_vehicle():
    rr.set_led1(1)

def turn_left():
    rr.set_motors(1, 1, 1, 0)

def turn_right():
    rr.set_motors(1, 0, 1, 1)

def forward():
    rr.set_motors(1, 1, 1, 1)

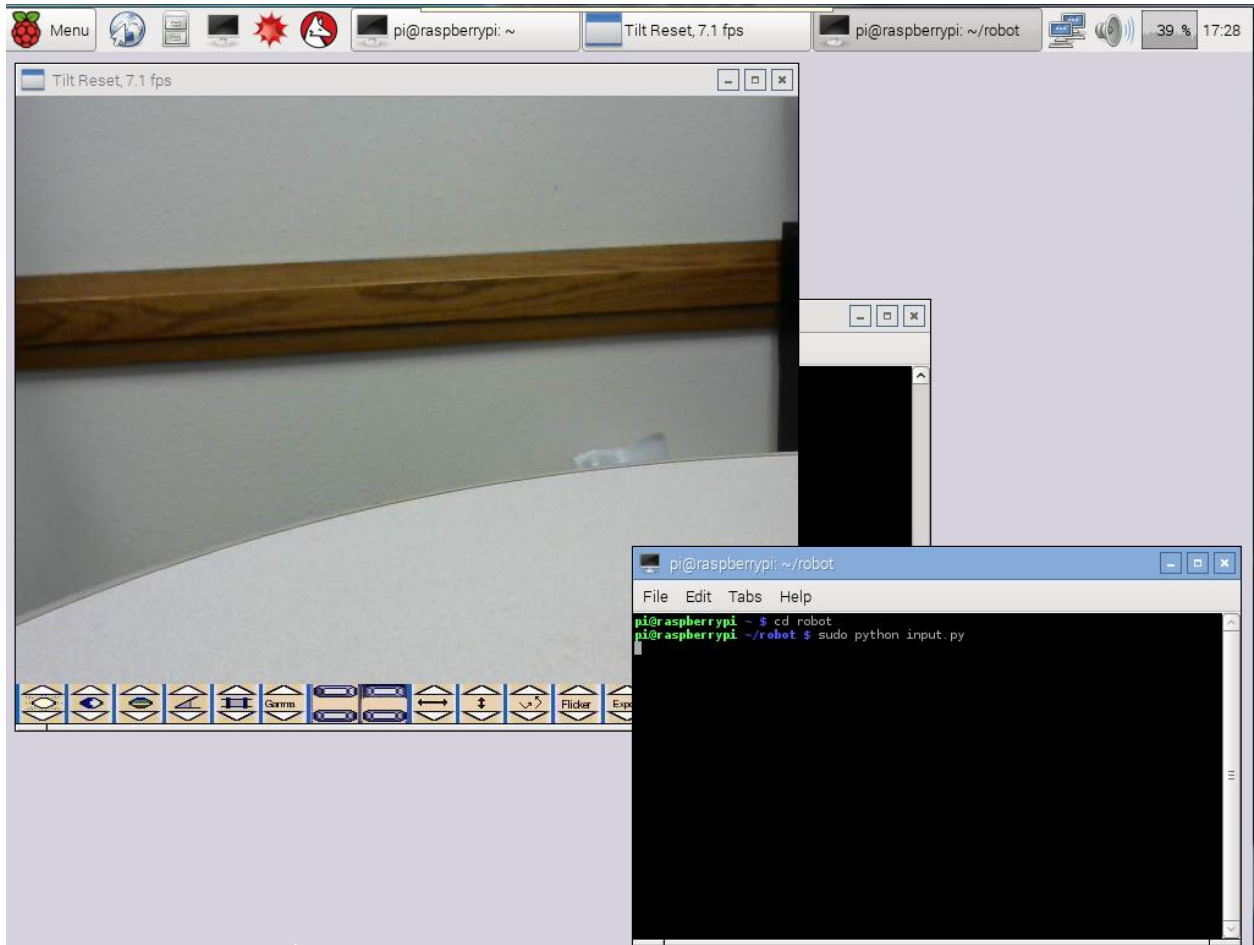
def backward():
    rr.set_motors(1, 0, 1, 0)

def stop():
    rr.set_motors(0, 0, 0, 0)

def cleanup():
    GPIO.cleanup()

-UU-:----F1 track.py All L5 (Python)-----
```

```
pi@raspberrypi: ~/robot
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
from rrb2 import *
from wheel import *
import tty
import sys
import termios
def getch():
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    tty.setraw(sys.stdin.fileno())
    ch = sys.stdin.read(1)
    termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch
var = 'n'
while var != 'q':
    var = getch()
    if var == 'l':
        turn_right()
    if var == 'r':
        turn_left()
    if var == 'f':
        forward()
    if var == 'b':
        backward()
    if var == 's':
        stop()
GPIO.cleanup()
-UU-:----F1 input.py All L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```




```
pi@raspberrypi: ~/robot
File Edit Options Buffers Tools Python Help
import pygame
import math
from PodSixNet.Connection import ConnectionListener, connection
from time import sleep
from wheel import *

class RobotGame(ConnectionListener):
    def Network_close(self, data):
        exit()
    def Network_gamepad(self, data):
        if data["type"] == 10:
            if data["info"]["button"] == 4:
                print "Fire Laser"
            if data["info"]["button"] == 5:
                print "Fire Laser"
            if data["info"]["button"] == 6:
                print "Fire Laser"
            if data["info"]["button"] == 7:
                print "Fire Laser"
        if data["type"] == 7:
            if data["info"]["value"] == 0.0:
                stop()
            else:
                if data["info"]["axis"] == 1:
                    if data["info"]["value"] > 0:
                        forward()
                    else:
                        backward()
                if data["info"]["axis"] == 2:
                    if data["info"]["value"] > 0:
                        turn_left()
                    else:
                        turn_right()

    def __init__(self):
        address=raw_input("Address of Server: ")
        try:
            if not address:
                host, port="localhost", 8000
            else:
                host,port=address.split(":")
            self.Connect((host, int(port)))
        except:
            print "Error Connecting to Server"
            print "Usage:", "host:port"

-UU-: **--F1 robot_client.py Top L1 (Python)
```

```
pi@raspberrypi: ~/robot
File Edit Options Buffers Tools Python Help
    else:
        turn_right()
def __init__(self):
    address=raw_input("Address of Server: ")
    try:
        if not address:
            host, port="localhost", 8000
        else:
            host,port=address.split(":")
            self.Connect((host, int(port)))
    except:
        print "Error Connecting to Server"
        print "Usage:", "host:port"
        print "e.g.", "localhost:31425"
        exit()
    print "Robot client started"
    self.running=False
    while not self.running:
        self.Pump()
        connection.Pump()
        sleep(0.01)

bg=RobotGame() # __init__ is called right here
while 1:
    if bg.update()==1:
        break
bg.finished()

-UU-: **--F1 robot_client.py Bot L46 (Python)-----
```

```
Python 2.7.8: flightserver - C:/Python27/flightserver
File Edit Format Run Options Windows Help
import PodSixNet.Server
from pygame import *
from time import sleep
init()
from time import sleep
class ClientChannel(PodSixNet.Channel.Channel):
    def Network(self, data):
        print data
    def Close(self):
        self._server.close(self.gameid)
class BoxesServer(PodSixNet.Server.Server):

    channelClass = ClientChannel
    def __init__(self, *args, **kwargs):
        PodSixNet.Server.Server.__init__(self, *args, **kwargs)
        self.games = []
        self.queue = None
        self.currentIndex=0
    def Connected(self, channel, addr):
        print 'new connection:', channel
        if self.queue==None:
            self.currentIndex+=1
            channel.gameid=self.currentIndex
            self.queue=Game(channel, self.currentIndex)
    def close(self, gameid):
        try:
            game = [a for a in self.games if a.gameid==gameid][0]
            game.player0.Send({"action":"close"})
        except:
            pass
    def tick(self):
        if self.queue != None:
            sleep(.05)
            for e in event.get():
                self.queue.player0.Send({"action":"gamepad", "type":e.type, "in
                self.Pump()
class Game:
    def __init__(self, player0, currentIndex):
        #initialize the players including the one who started the game
        self.player0=player0

#Setup and init joystick
j=joystick.Joystick(0)
j.init()

#Check init status
if j.get_init() == 1: print "Joystick is initialized"
```

```
Python 2.7.8: flightserver - C:/Python27/flightserver
File Edit Format Run Options Windows Help
    sleep(0.05)
    for e in event.get():
        self.queue.player0.Send({"action":"gamepad", "type":e.type, "in
    self.Pump()
class Game:
    def __init__(self, player0, currentIndex):
        #initialize the players including the one who started the game
        self.player0=player0

#Setup and init joystick
j=joystick.Joystick(0)
j.init()

#Check init status
if j.get_init() == 1: print "Joystick is initialized"

#Get and print joystick ID
print "Joystick ID: ", j.get_id()

#Get and print joystick name
print "Joystick Name: ", j.get_name()

#Get and print number of axes
print "No. of axes: ", j.get_numaxes()


#Get and print number of trackballs
print "No. of trackballs: ", j.get_numballs()

#Get and print number of buttons
print "No. of buttons: ", j.get_numbuttons()

#Get and print number of hat controls
print "No. of hat controls: ", j.get_numhats()

print "STARTING SERVER ON LOCALHOST"
# try:
address=raw_input("Host:Port (localhost:8000): ")
if not address:
    host, port="localhost", 8000
else:
    host,port=address.split(":")
boxesServe = BoxesServer(localaddr=(host, int(port)))

while True:
    boxesServe.tick()
    sleep(0.01)
```

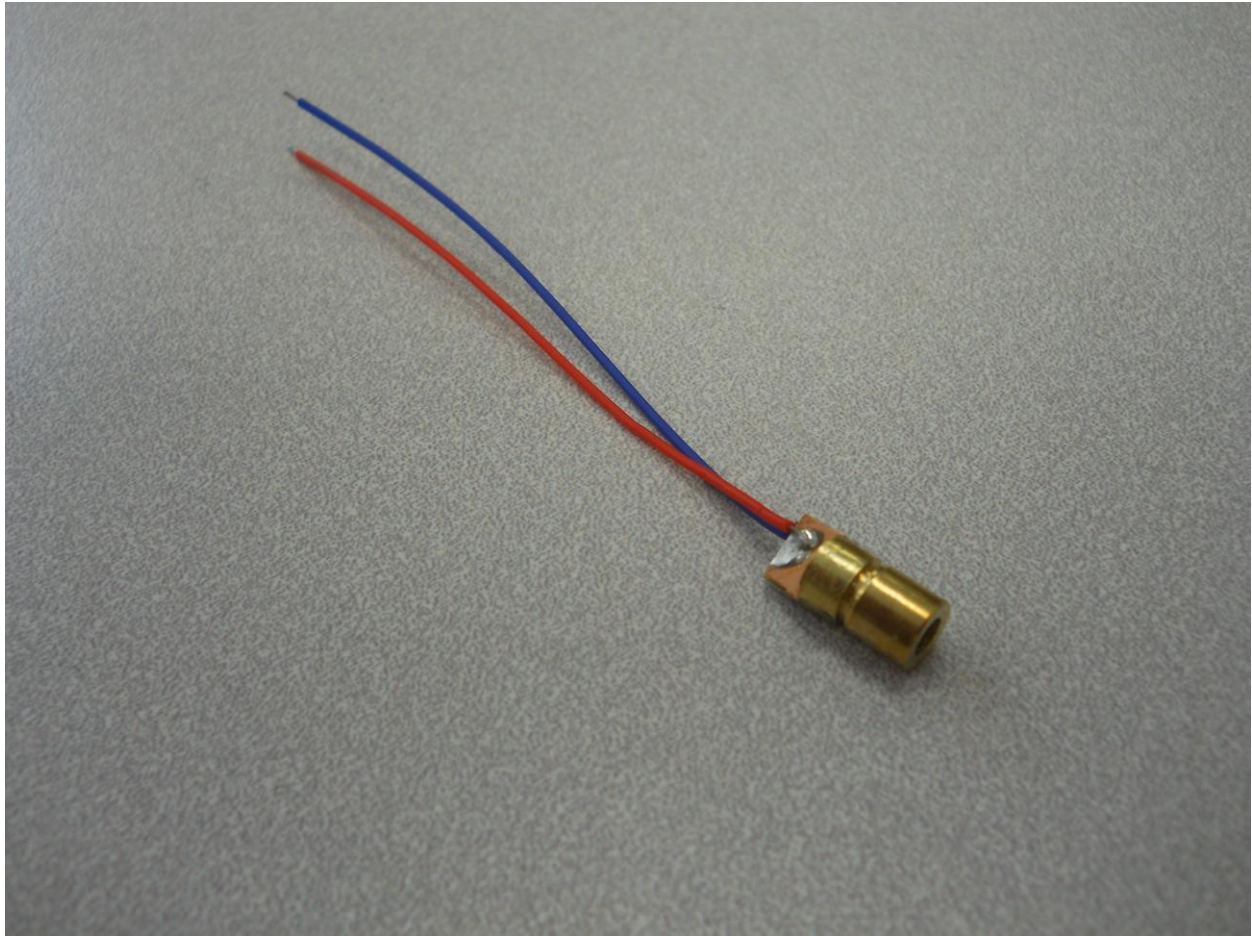


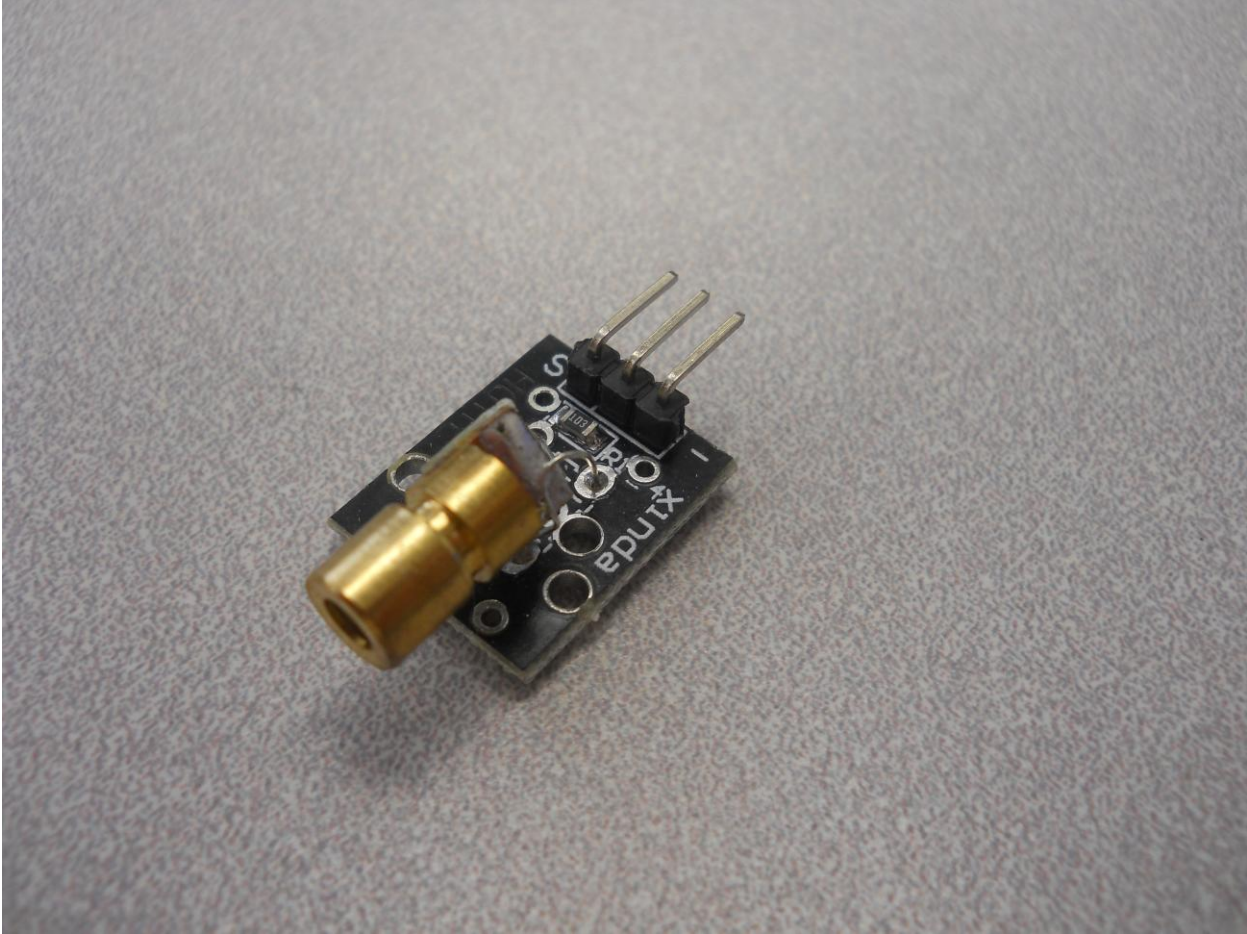
```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ python joystick.py  
Joystick is initialized  
Joystick ID: 0  
Joystick Name: 2603666 CONTROLLER  
No. of axes: 4  
No. of trackballs: 0  
No. of buttons: 12  
No. of hat controls: SDL_JoystickNumHats value:1:  
 1  
STARTING SERVER ON LOCALHOST  
Host:Port (localhost:8000): 157.201.194.150:8000  
█
```

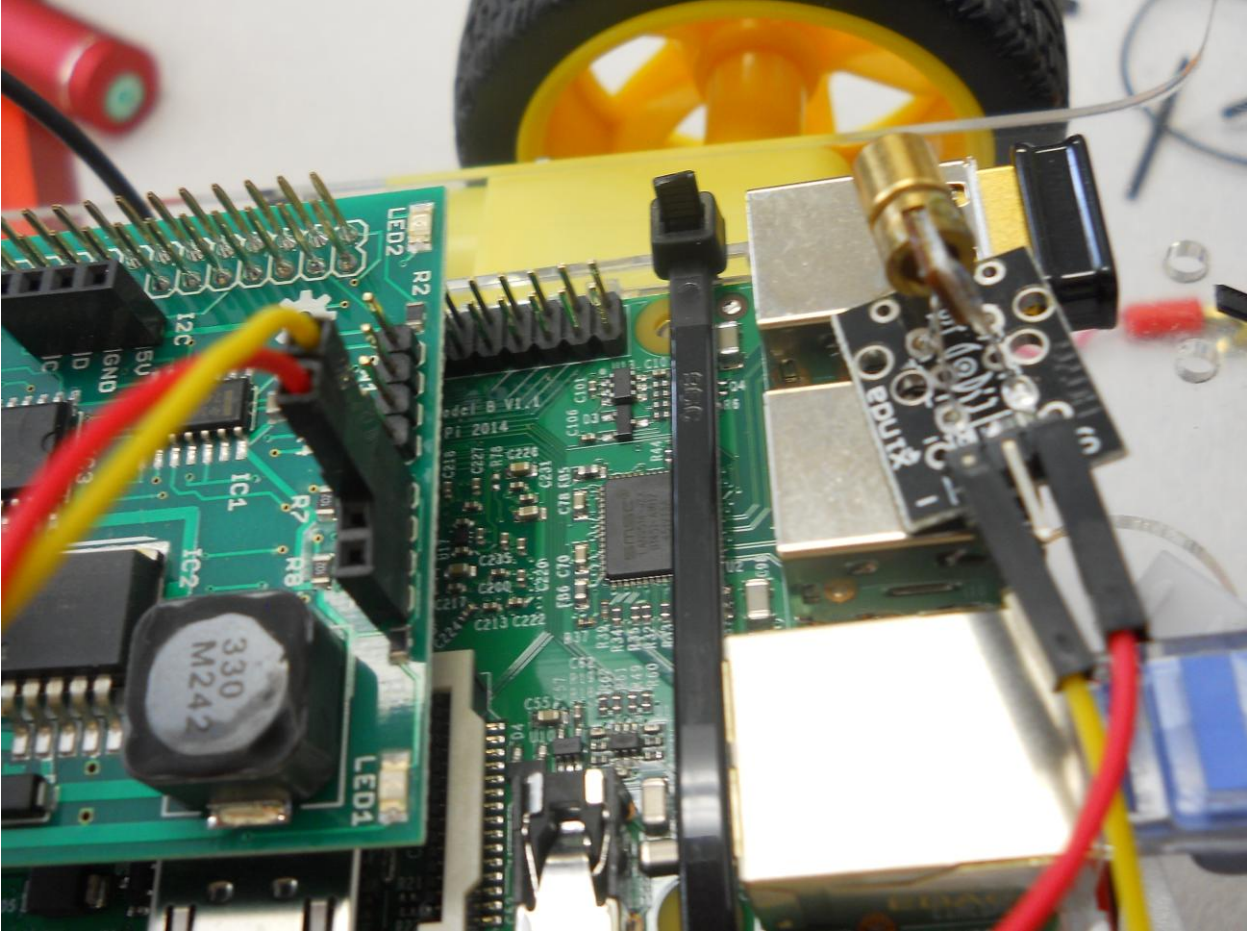
```
pi@raspberrypi: ~/robot  
pi@raspberrypi ~/robot $ sudo python robot_client.py  
Address of Server: 157.201.194.150:8000  
Robot client started  
█
```



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~$ python joystick.py  
Joystick is initialized  
Joystick ID: 0  
Joystick Name: 2603666 CONTROLLER  
No. of axes: 4  
No. of trackballs: 0  
No. of buttons: 12  
No. of hat controls: SDL_JoystickNumHats value:1:  
1  
STARTING SERVER ON LOCALHOST  
Host:Port (localhost:8000): 157.201.194.150:8000  
new connection: <_main__.ClientChannel connected 157.201.194.189:43833 at 0x217  
7b98>  
█
```







```
pi@raspberrypi: ~/robot
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time
from rrb2 import *

rr = RRB2()

def laser_on():
    rr.set_led1(1)
    rr.set_oc1(1)
    time.sleep(1)
    rr.set_led1(0)
    rr.set_oc1(0)

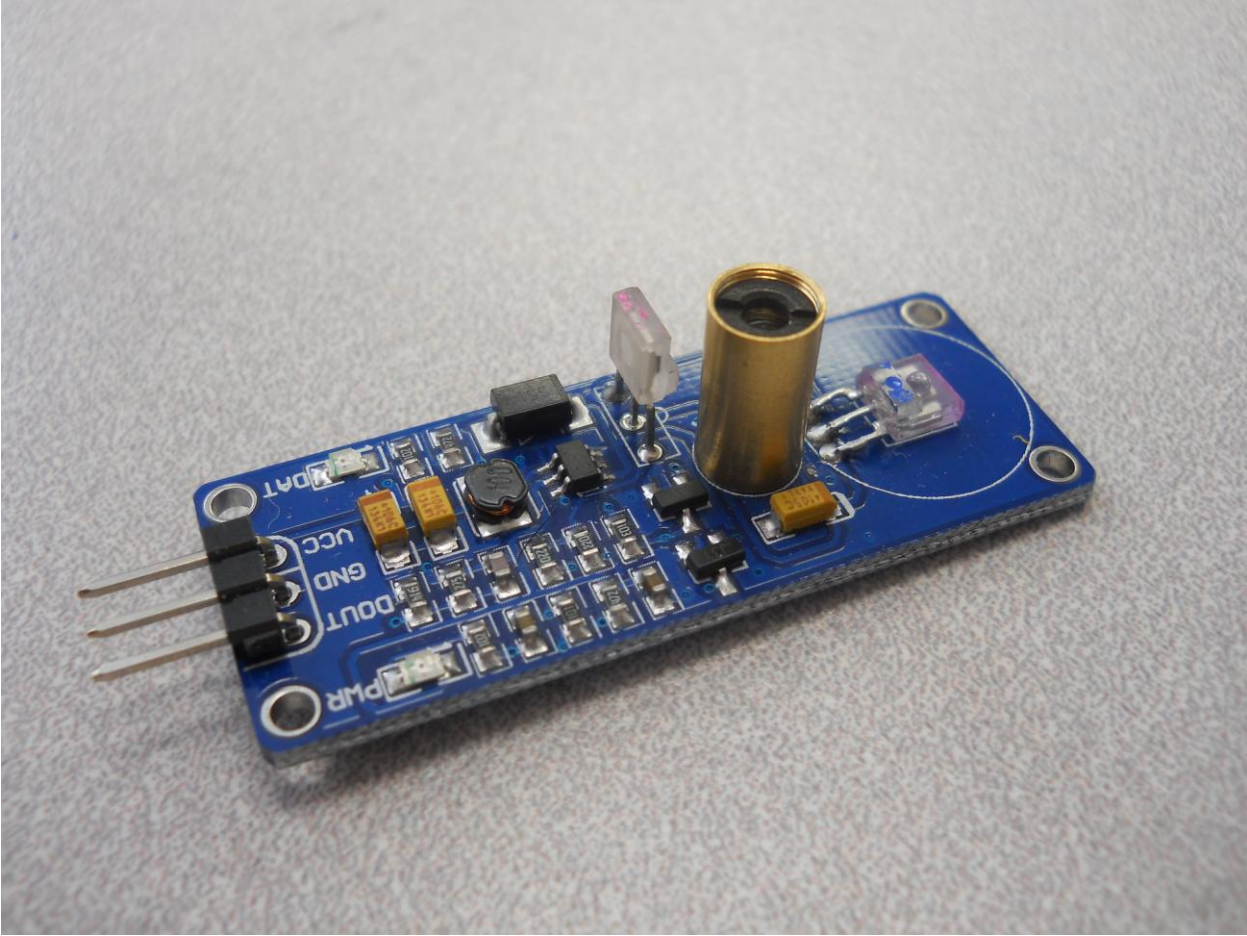
-UU-:-----F1 laser.py All L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```



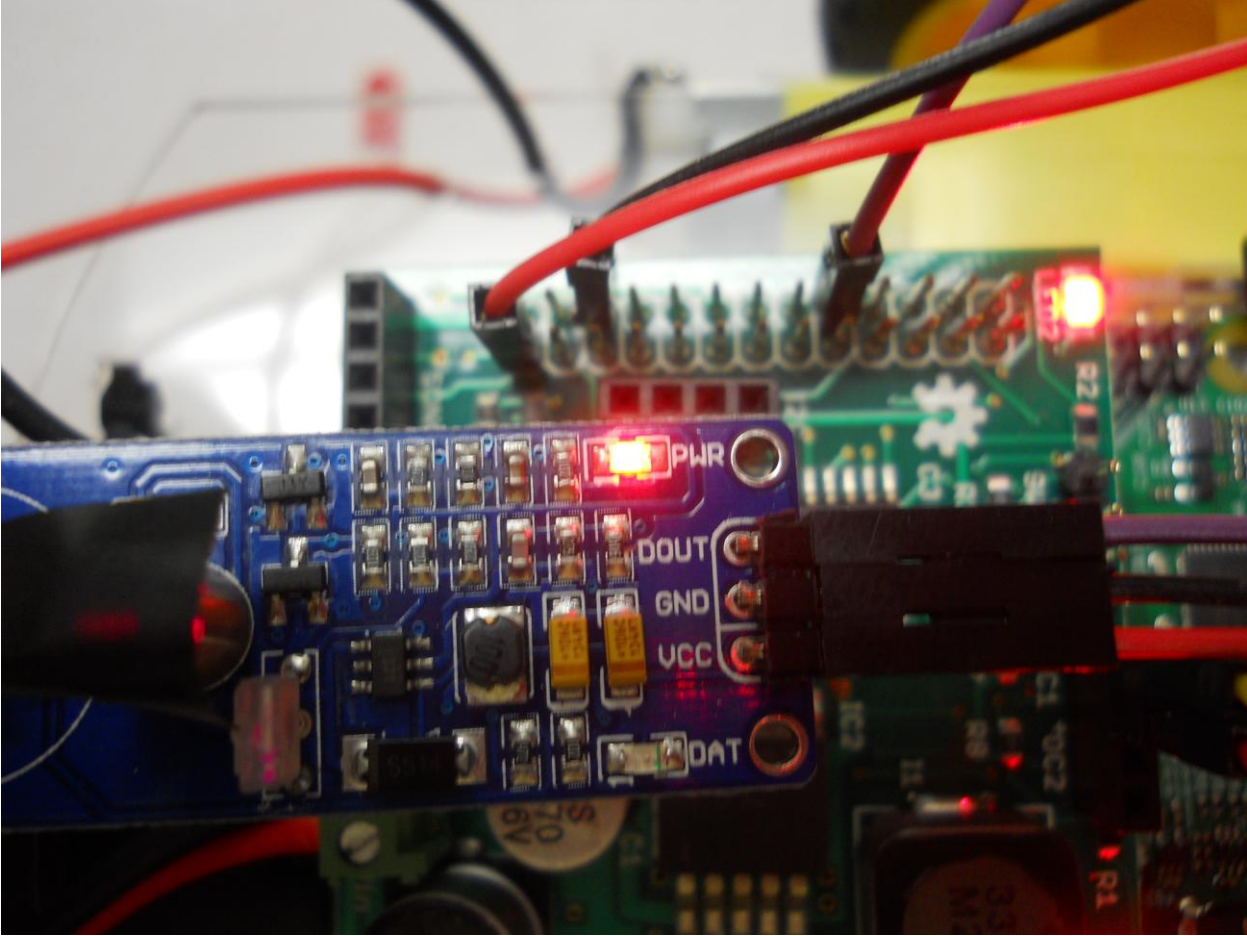
```
pi@raspberrypi: ~/robot
File Edit Options Buffers Tools Python Help
import pygame
import math
from PodSixNet.Connection import ConnectionListener, connection
from time import sleep
from wheel import *
from laser import *

class RobotGame(ConnectionListener):
    def Network_close(self, data):
        exit()
    def Network_gamepad(self, data):
        if data["type"] == 10:
            if data["info"]["button"] == 4:
                laser_on()
                print "Fire Laser"
            if data["info"]["button"] == 5:
                laser_on()
                print "Fire Laser"
            if data["info"]["button"] == 6:
                laser_on()
                print "Fire Laser"
            if data["info"]["button"] == 7:
                laser_on()
                print "Fire Laser"
        if data["type"] == 7:
            if data["info"]["value"] == 0.0:
                stop()
        else:
            pass
-UU-:----F1 robot_client.py Top L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```



Pin 1 3.3V	<input type="checkbox"/> <input type="radio"/>	Pin 2 5V
Pin 3 GPIO2	<input type="radio"/> <input type="radio"/>	Pin 4 5V
Pin 5 GPIO3	<input type="radio"/> <input type="radio"/>	Pin 6 GND
Pin 7 GPIO4	<input type="radio"/> <input type="radio"/>	Pin 8 GPIO14
Pin 9 GND	<input type="radio"/> <input type="radio"/>	Pin 10 GPIO15
Pin 11 GPIO17	<input type="radio"/> <input type="radio"/>	Pin 12 GPIO18
Pin 13 GPIO27	<input type="radio"/> <input type="radio"/>	Pin 14 GND
Pin 15 GPIO22	<input type="radio"/> <input type="radio"/>	Pin 16 GPIO23
Pin 17 3.3V	<input type="radio"/> <input type="radio"/>	Pin 18 GPIO24
Pin 19 GPIO10	<input type="radio"/> <input type="radio"/>	Pin 20 GND
Pin 21 GPIO9	<input type="radio"/> <input type="radio"/>	Pin 22 GPIO25
Pin 23 GPIO11	<input type="radio"/> <input type="radio"/>	Pin 24 GPIO8
Pin 25 GND	<input type="radio"/> <input type="radio"/>	Pin 26 GPIO7
Pin 27 ID_SD	<input type="radio"/> <input type="radio"/>	Pin 28 ID_SC
Pin 29 GPIO5	<input type="radio"/> <input type="radio"/>	Pin 30 GND
Pin 31 GPIO6	<input type="radio"/> <input type="radio"/>	Pin 32 GPIO12
Pin 33 GPIO13	<input type="radio"/> <input type="radio"/>	Pin 34 GND
Pin 35 GPIO19	<input type="radio"/> <input type="radio"/>	Pin 36 GPIO16
Pin 37 GPIO26	<input type="radio"/> <input type="radio"/>	Pin 38 GPIO20
Pin 39 GND	<input type="radio"/> <input type="radio"/>	Pin 40 GPIO21

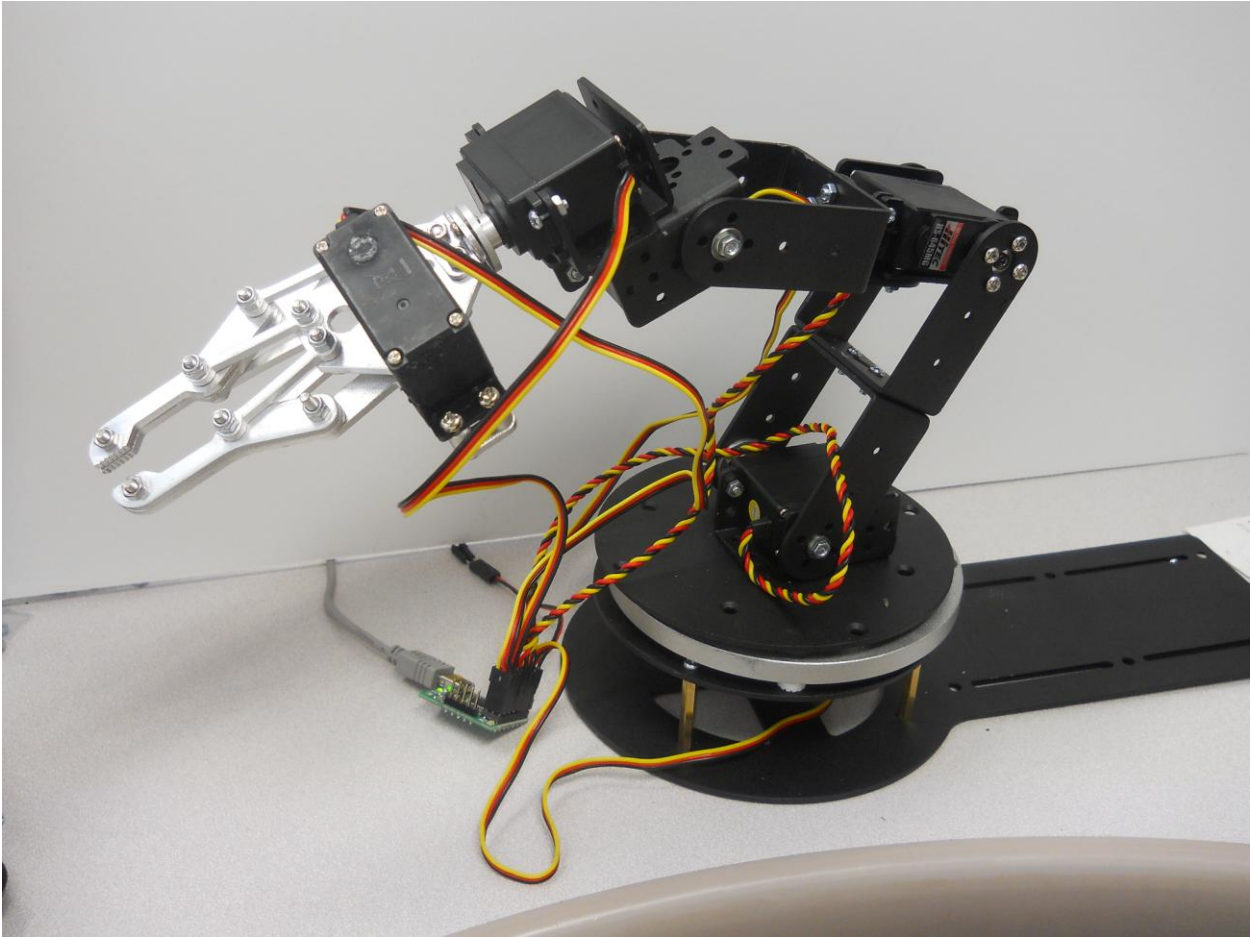


```
pi@raspberrypi: ~/robot
File Edit Options Buffers Tools Python Help
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
target_pin = 24
GPIO.setup(target_pin,GPIO.IN)
while 1:
    hit = GPIO.input(target_pin)
    if hit == 0:
        print("HIT detected");
        time.sleep(.1)

-UU-:----F1 laser target.py All L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```


Chapter 5: A Robot That Can Draw







Pololu Maestro Control Center

File Device Edit Help

Connected to: #00046711 Firmware version: 1.01 Error code: 0x0000

Status Errors Channel Settings Serial Settings Sequence Script

#	Name	Mode	Enabled		Target	Speed	Acceleration	Position
0		Servo	<input type="checkbox"/>		1500.00	0	0	0.00
1		Servo	<input type="checkbox"/>		1500.00	0	0	0.00
2		Servo	<input type="checkbox"/>		1500.00	0	0	0.00
3		Servo	<input type="checkbox"/>		1500.00	0	0	0.00
4		Servo	<input type="checkbox"/>		1500.00	0	0	0.00
5		Servo	<input type="checkbox"/>		1500.00	0	0	0.00

Save Frame 0 Apply Settings

Pololu Maestro Control Center

File Device Edit Help

Connected to: #00046711 Firmware version: 1.01 Error code: 0x0000

Status Errors Channel Settings **Serial Settings** Sequence Script

Serial mode:

- USB Dual Port
- USB Chained
- UART, fixed baud rate: 9600
- UART, detect baud rate

Enable CRC

Device Number: 12

Mini SSC offset: 0

Timeout (s): 0.00

Never sleep (ignore USB suspend)

Save Frame 0 Apply Settings

Pololu Maestro Control Center

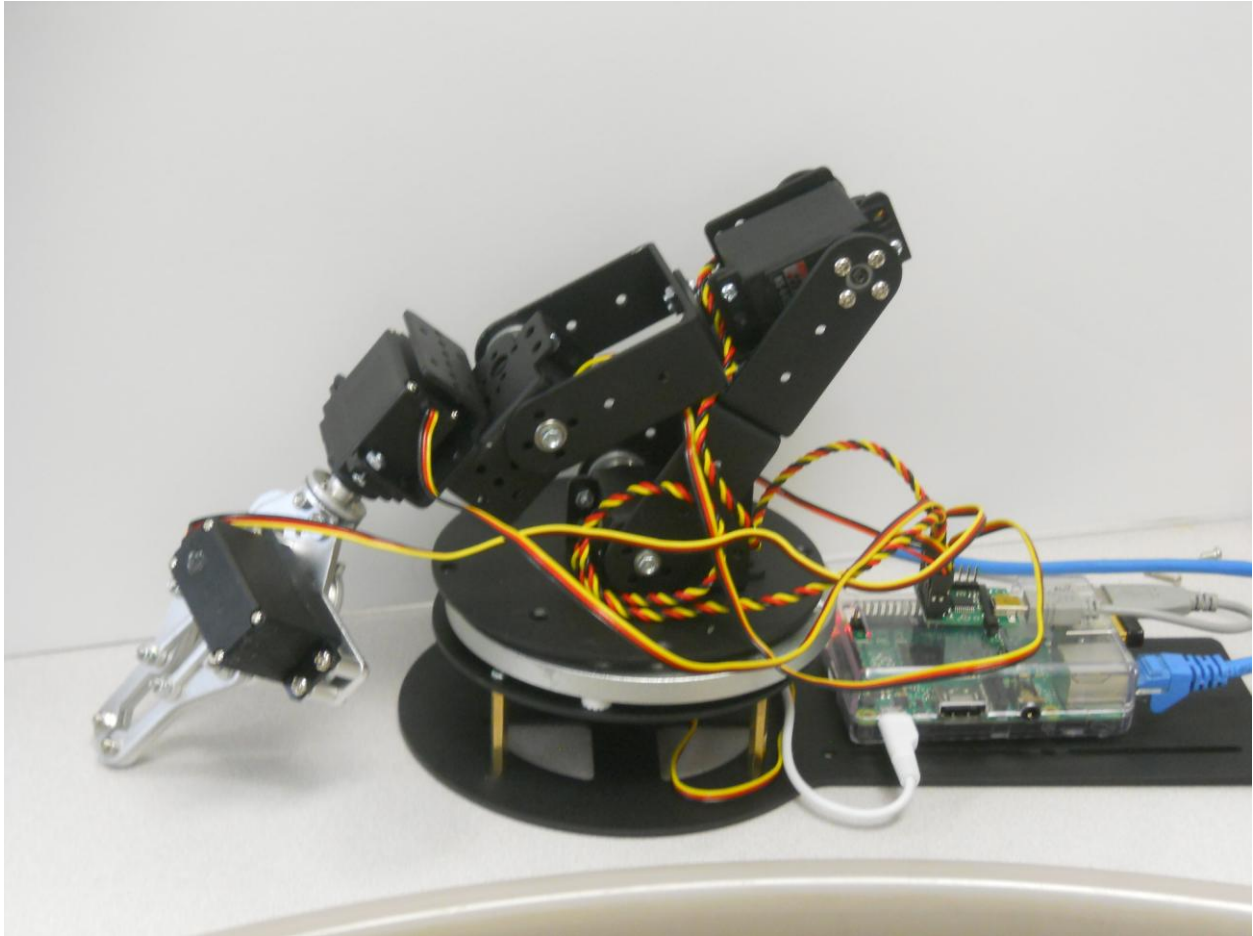
File Device Edit Help

Connected to: #00046711 Firmware version: 1.01 Error code: 0x0000

Status Errors Channel Settings **Serial Settings** Sequence Script

#	Name	Mode	Enabled		Target	Speed	Acceleration	Position
0	Servo		<input checked="" type="checkbox"/>		1500.00	0	0	1500.00
1	Servo		<input checked="" type="checkbox"/>		1500.00	0	0	1500.00
2	Servo		<input checked="" type="checkbox"/>		1500.00	0	0	1500.00
3	Servo		<input checked="" type="checkbox"/>		1500.00	0	0	1500.00
4	Servo		<input type="checkbox"/>		1500.00	0	0	0.00
5	Servo		<input type="checkbox"/>		1500.00	0	0	0.00

Save Frame 0 Apply Settings



```
pi@raspberrypi: ~/maestro_linux
pi@raspberrypi ~/maestro_linux $ ls
99-pololu.rules  FirmwareUpgrade.dll  README.txt  UsbWrapper.dll  Usc.dll
Bytecode.dll    MaestroControlCenter  Sequencer.dll  UscCmd
pi@raspberrypi ~/maestro_linux $
```



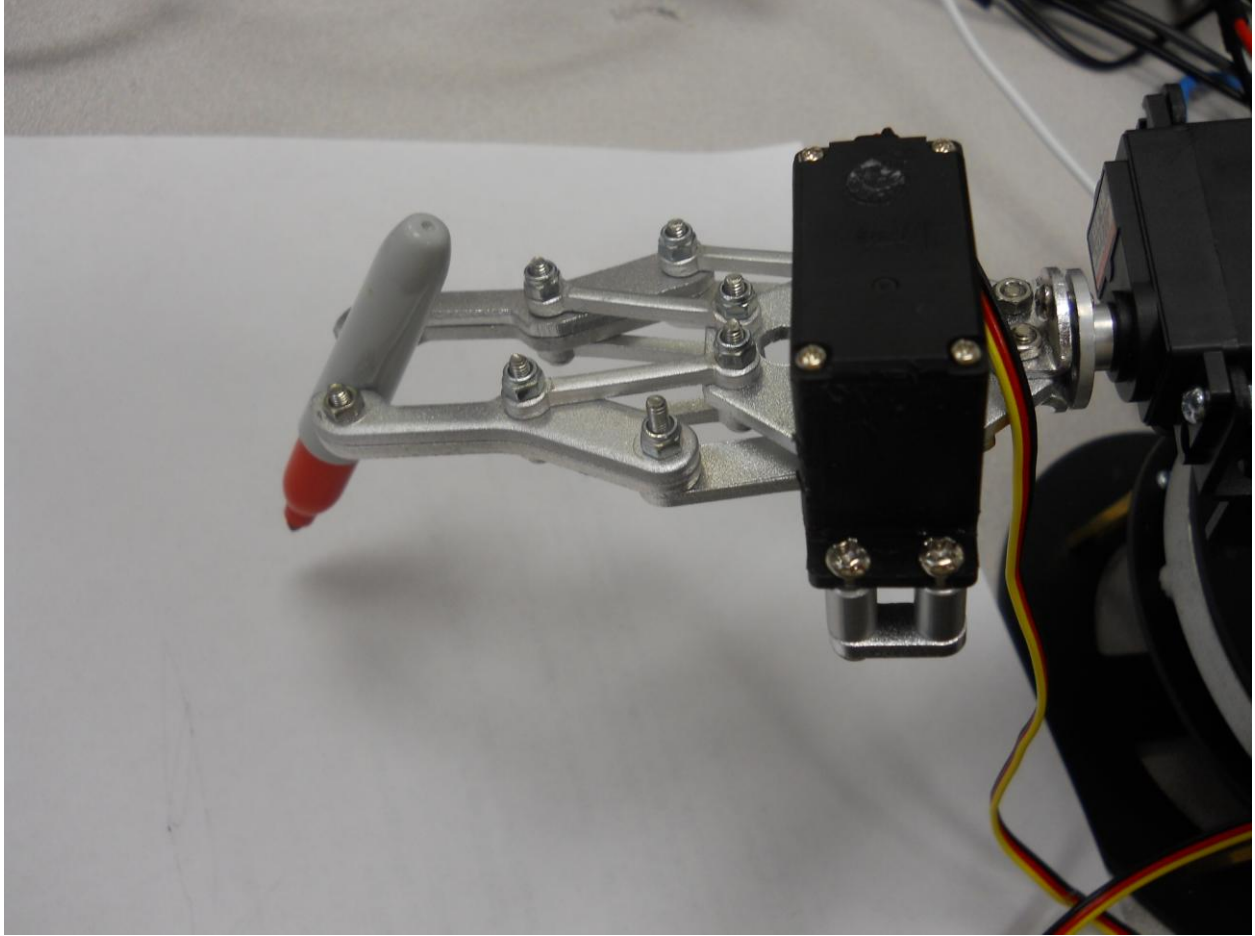
```
pi@raspberrypi: ~/maestro_linux
pi@raspberrypi ~/maestro_linux $ ./UscCmd --list
1 Maestro USB servo controller device found:
#00027392
pi@raspberrypi ~/maestro_linux $
```

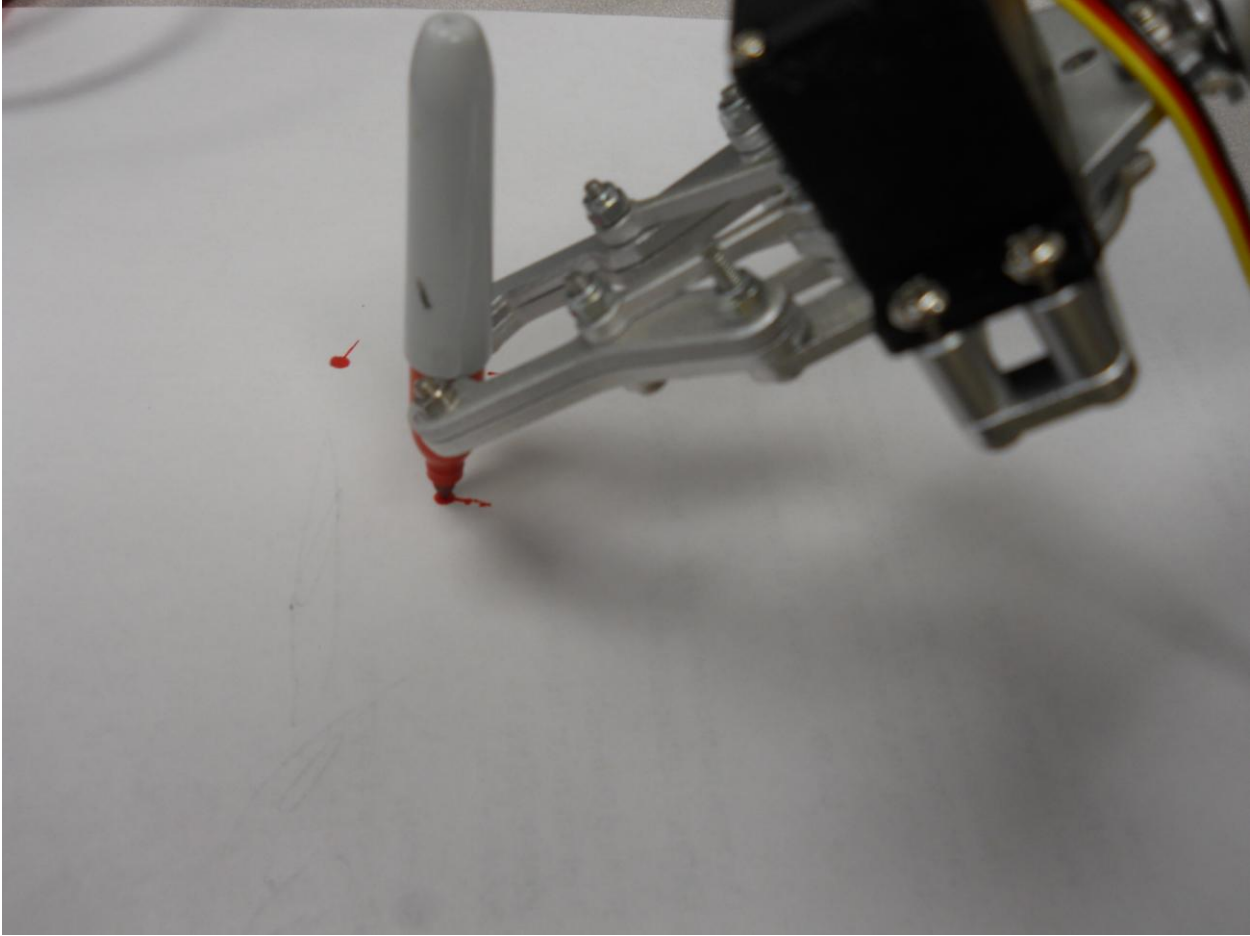
```
pi@raspberrypi: ~/maestro_linux
UscCmd, Version=1.3.0.0, Culture=neutral, PublicKeyToken=null
Select one of the following actions:
--list                list available devices
--configure FILE      load configuration file into device
--getconf FILE        read device settings and write configuration file
--restoredefaults     restore factory settings
--program FILE        compile and load bytecode program
--status              display complete device status
--bootloader          put device into bootloader (firmware upgrade) mode
--stop                stops the script running on the device
--start               starts the script running on the device
--restart              restarts the script at the beginning
--step                runs a single instruction of the script
--sub NUM              calls subroutine n (can be hex or decimal)
--sub NUM,PARAMETER  calls subroutine n with a parameter (hex or decimal)
                      placed on the stack
--servo NUM,TARGET    sets the target of servo NUM in units of
                      1/4 microsecond
--speed NUM,SPEED     sets the speed limit of servo NUM
--accel NUM,ACCEL     sets the acceleration of servo NUM to a value 0-255
Select which device to perform the action on (optional):
--device 00001430     (optional) select device #00001430
pi@raspberrypi ~/maestro_linux $
```

```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time
def setAngle(ser, channel, angle):
    minAngle = 0.0
    maxAngle = 180.0
    minTarget = 256.0
    maxTarget = 13120.0
    scaledValue = int((angle / ((maxAngle - minAngle) / (maxTarget - minTarget)\
)) + minTarget)
    commandByte = chr(0x84)
    channelByte = chr(channel)
    lowTargetByte = chr(scaledValue & 0x7F)
    highTargetByte = chr((scaledValue >> 7) & 0x7F)
    command = commandByte + channelByte + lowTargetByte + highTargetByte
    ser.write(command)
    ser.flush()
def setSpeed(ser, channel, speed):
    if speed > 127 or speed < 0:
        speed=1
    commandByte = chr(0x87)
    channelByte = chr(channel)
    highByte, lowByte = divmod(speed,32)
    highTargetByte = chr(highByte)
    lowTargetByte = chr(lowByte << 2)
    command = commandByte + channelByte + lowTargetByte + highTargetByte
    ser.write(command)
    ser.flush()
def setHome(ser):
    for i in range(0, 5):
        setAngle(ser, i ,90)

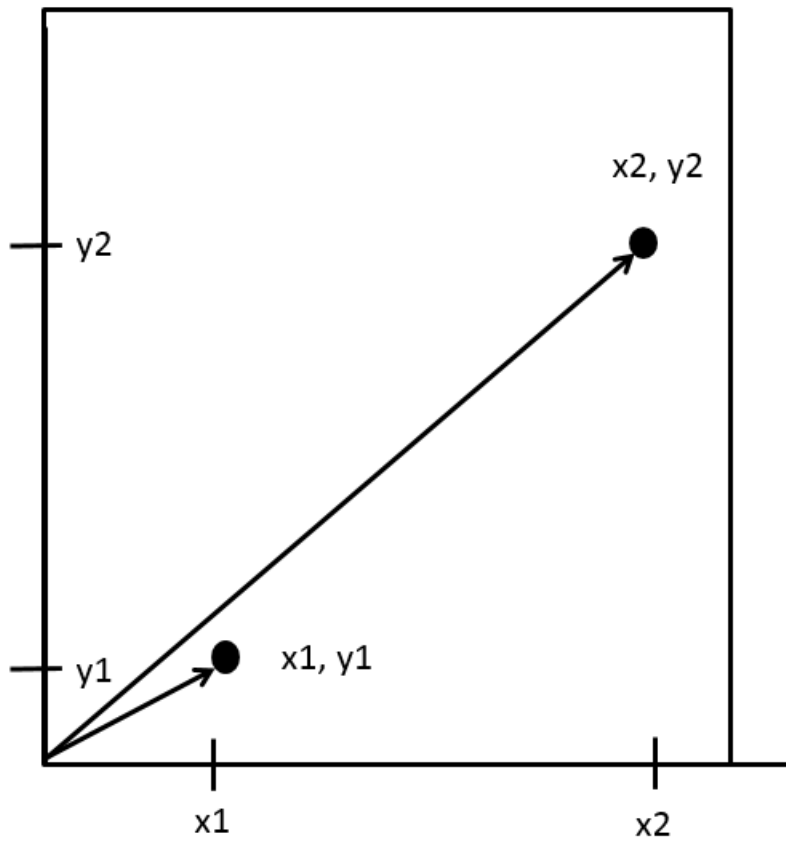
ser = serial.Serial("/dev/ttyACM0", 9600)
setHome(ser)
time.sleep(1)
while 1:
    servo = int(raw_input("Servo number: "))
    angle = int(raw_input("Angle: "))
    speed = int(raw_input("Speed: "))
    setSpeed(ser, servo, speed)
    setAngle(ser, servo, angle)
    time.sleep(.5)

-UU-:----F1 robotArm.py All L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```

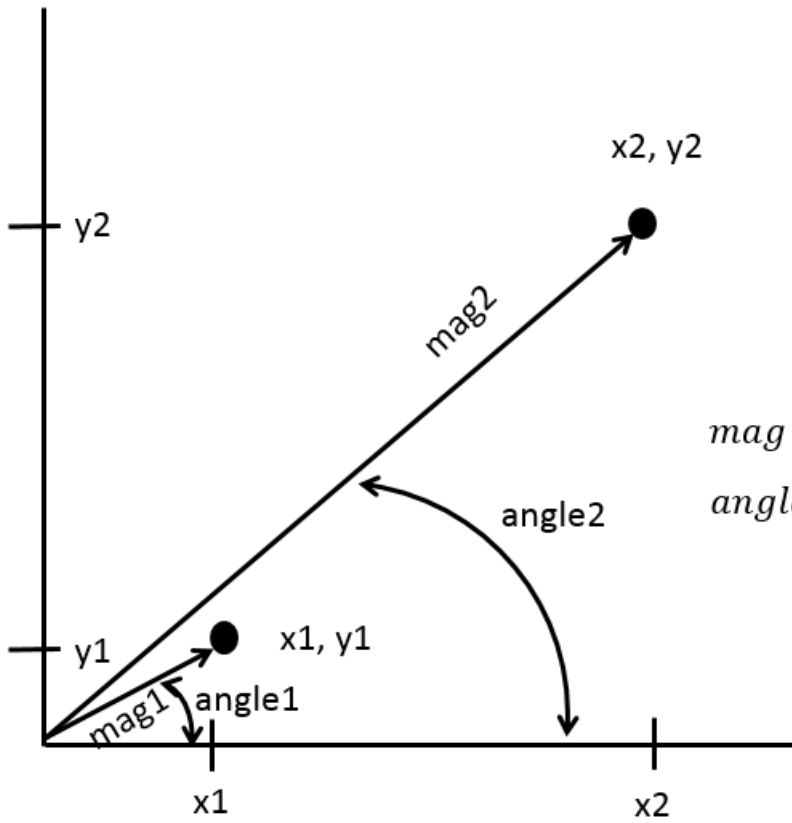






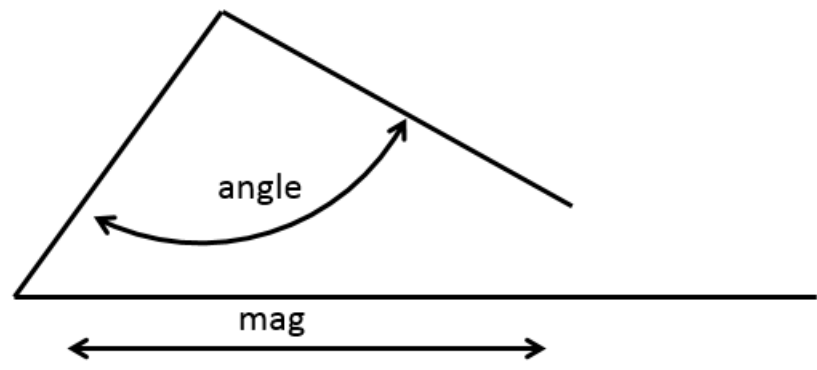


$$y = \text{mag} * \sin(\text{angle})$$



$$\text{mag} = \sqrt{(x^2 + y^2)}$$
$$\text{angle} = \arctan\left(\frac{y}{x}\right)$$

$$x = \text{mag} * \cos(\text{angle})$$



$$\text{mag} = \text{maxLength} * \sin(\text{angle})$$

```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time

def setAngle(ser, channel, angle):
    minAngle = 0.0
    maxAngle = 180.0
    minTarget = 256.0
    maxTarget = 13120.0
    scaledValue = int((angle / ((maxAngle - minAngle) / (maxTarget - minTarget)\
)) + minTarget)
    commandByte = chr(0x84)
    channelByte = chr(channel)
    lowTargetByte = chr(scaledValue & 0x7F)
    highTargetByte = chr((scaledValue >> 7) & 0x7F)
    command = commandByte + channelByte + lowTargetByte + highTargetByte
    ser.write(command)
    ser.flush()

def setSpeed(ser, channel, speed):
    if speed > 127 or speed < 0:
        speed=1
    commandByte = chr(0x87)
    channelByte = chr(channel)
    highByte, lowByte = divmod(speed,32)
    highTargetByte = chr(highByte)
    lowTargetByte = chr(lowByte << 2)
    command = commandByte + channelByte + lowTargetByte + highTargetByte
    ser.write(command)
    ser.flush()

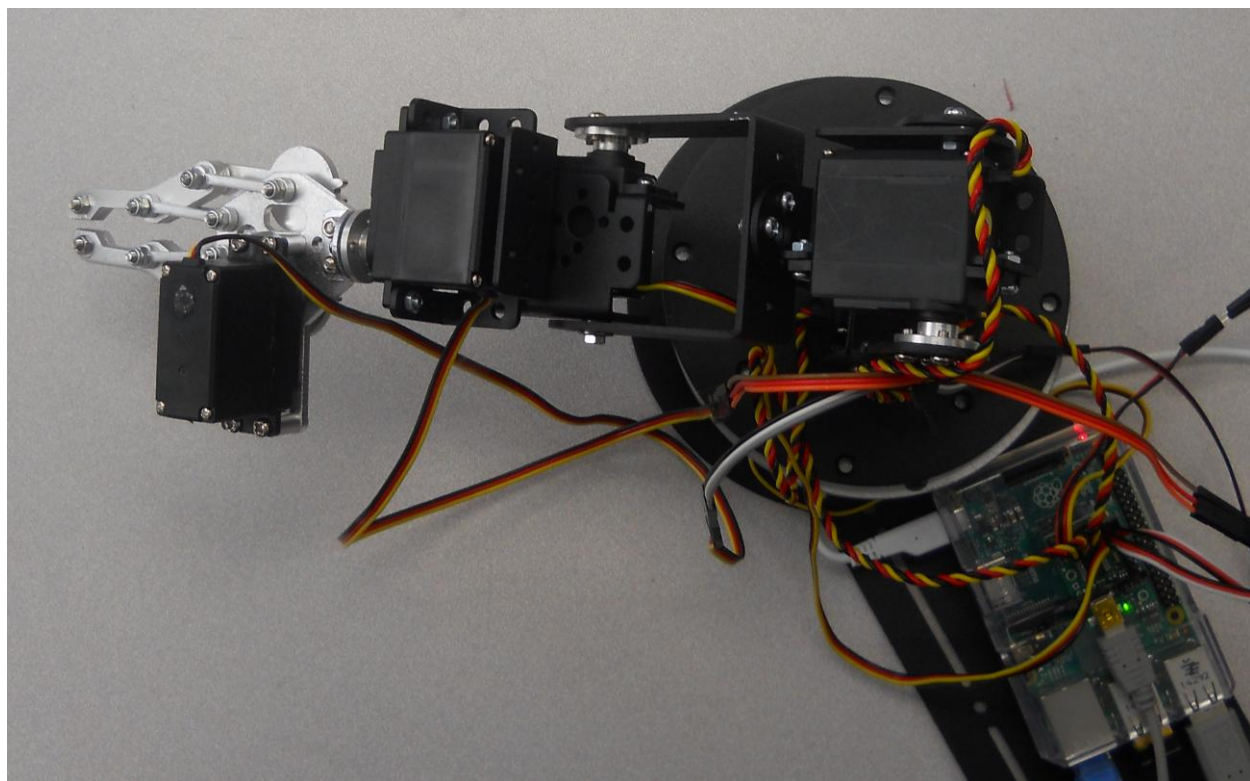
-UU-:----F1 robotArmLib.py Top L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```

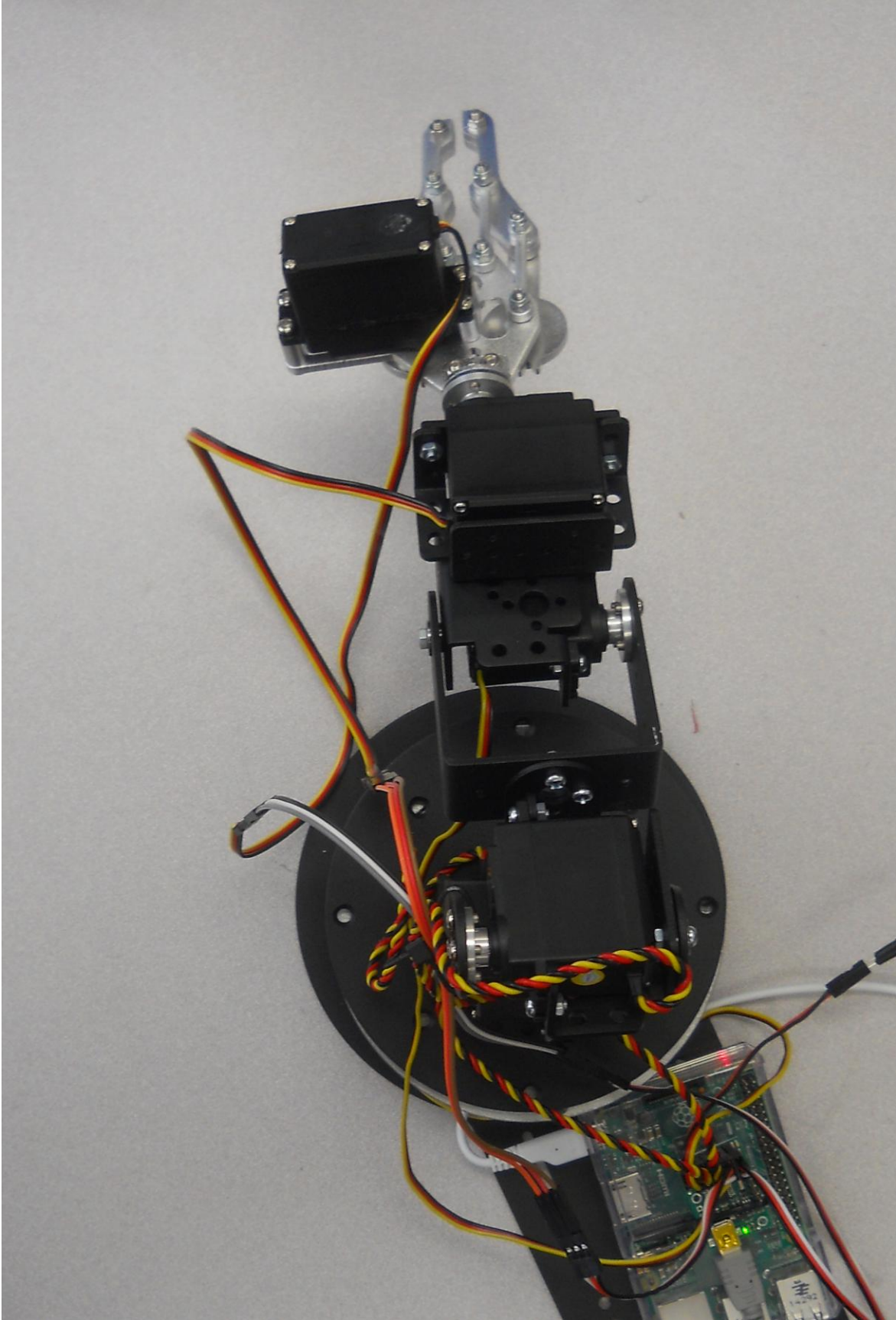
```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time
import math
from robotArmLib import *

ser = serial.Serial("/dev/ttyACM0", 9600)

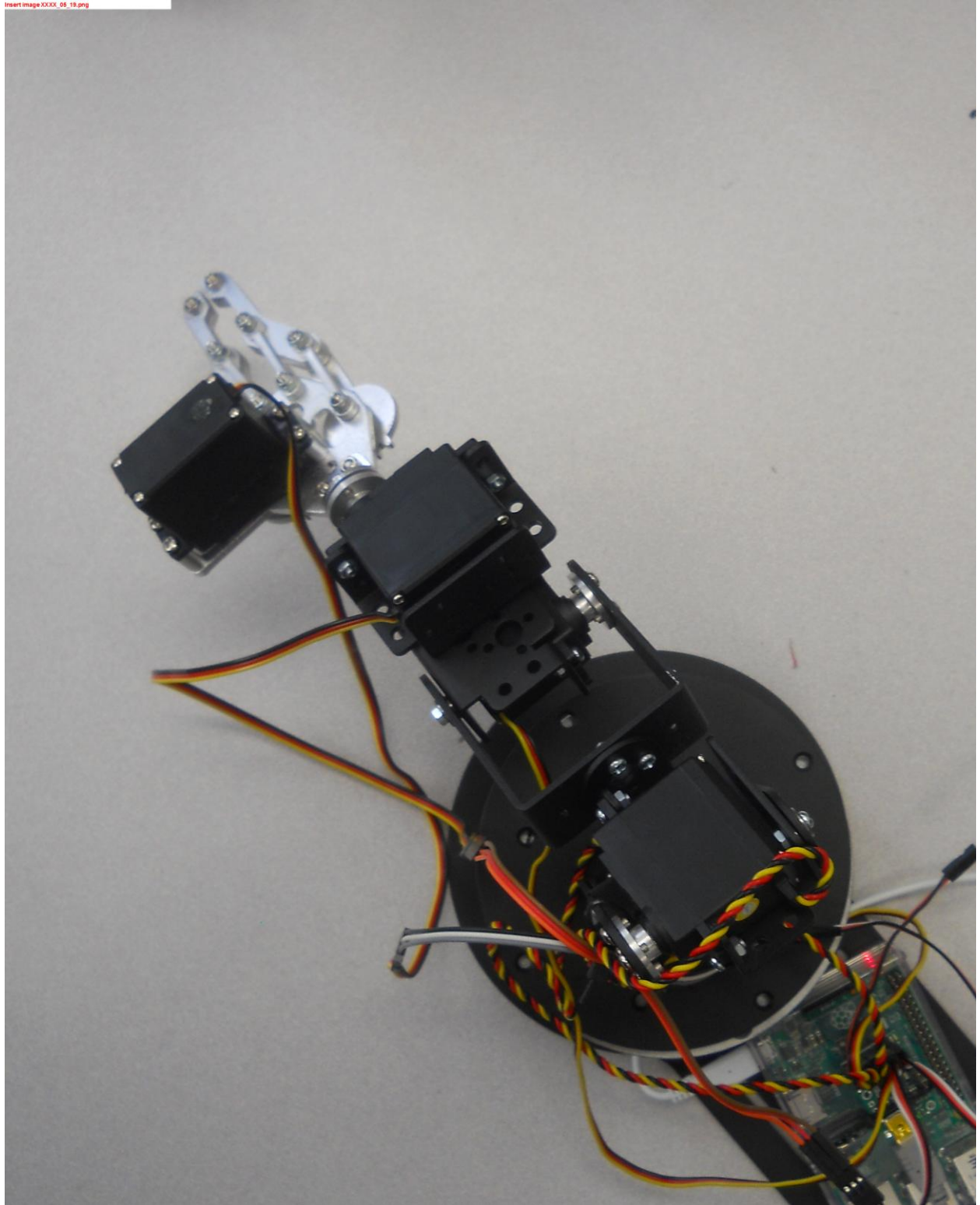
while 1:
    x = int(raw_input("x: "))
    y = int(raw_input("y: "))
    speed = 5
    servo0 = 0
    setSpeed(ser, servo0, speed)
    angle = int((math.atan2(y, x) * 360) / (2 * 3.1416))
    print angle
    setAngle(ser, servo0, angle + 40)
    time.sleep(.5)

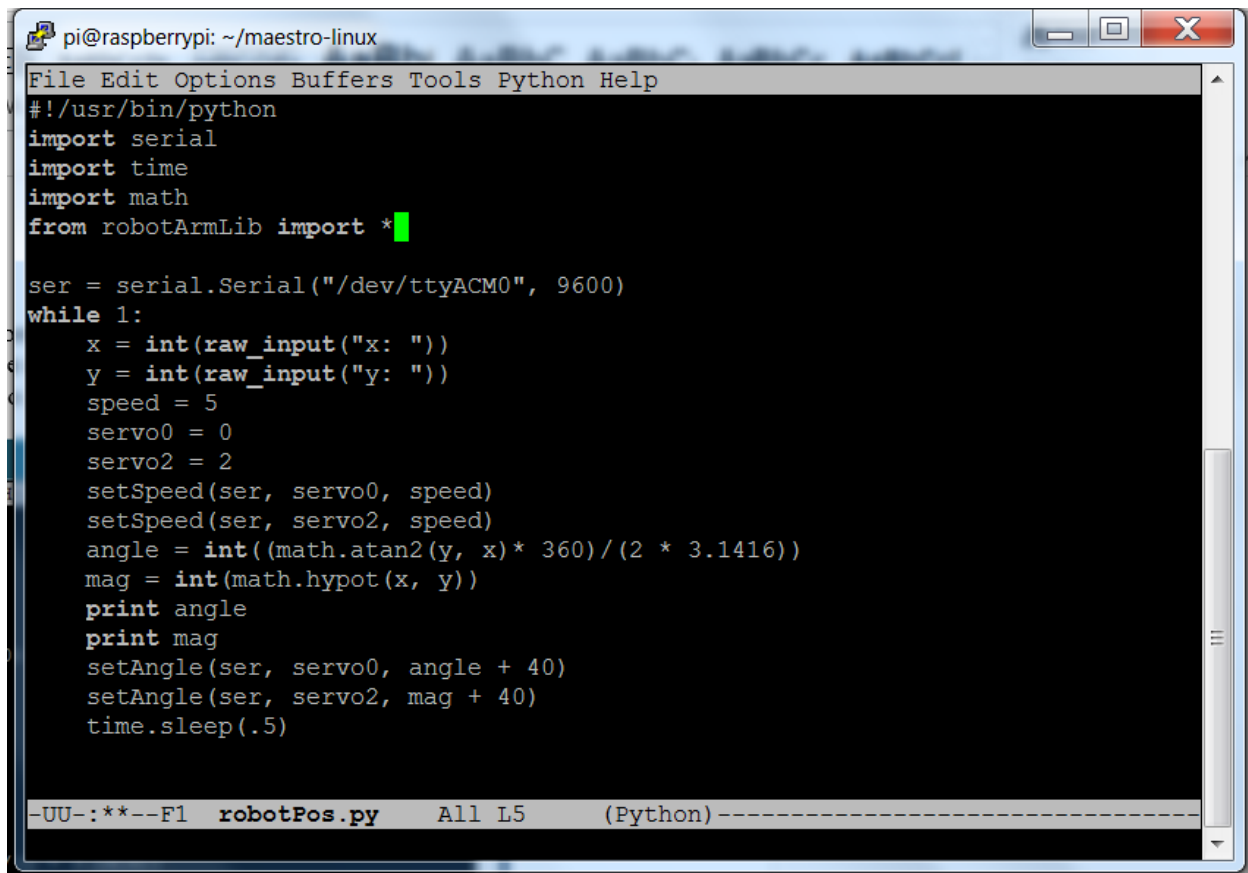
-UU-: **--F1 robotPos.py All L18 (Python)
```





insert image XXXX_05_19.png



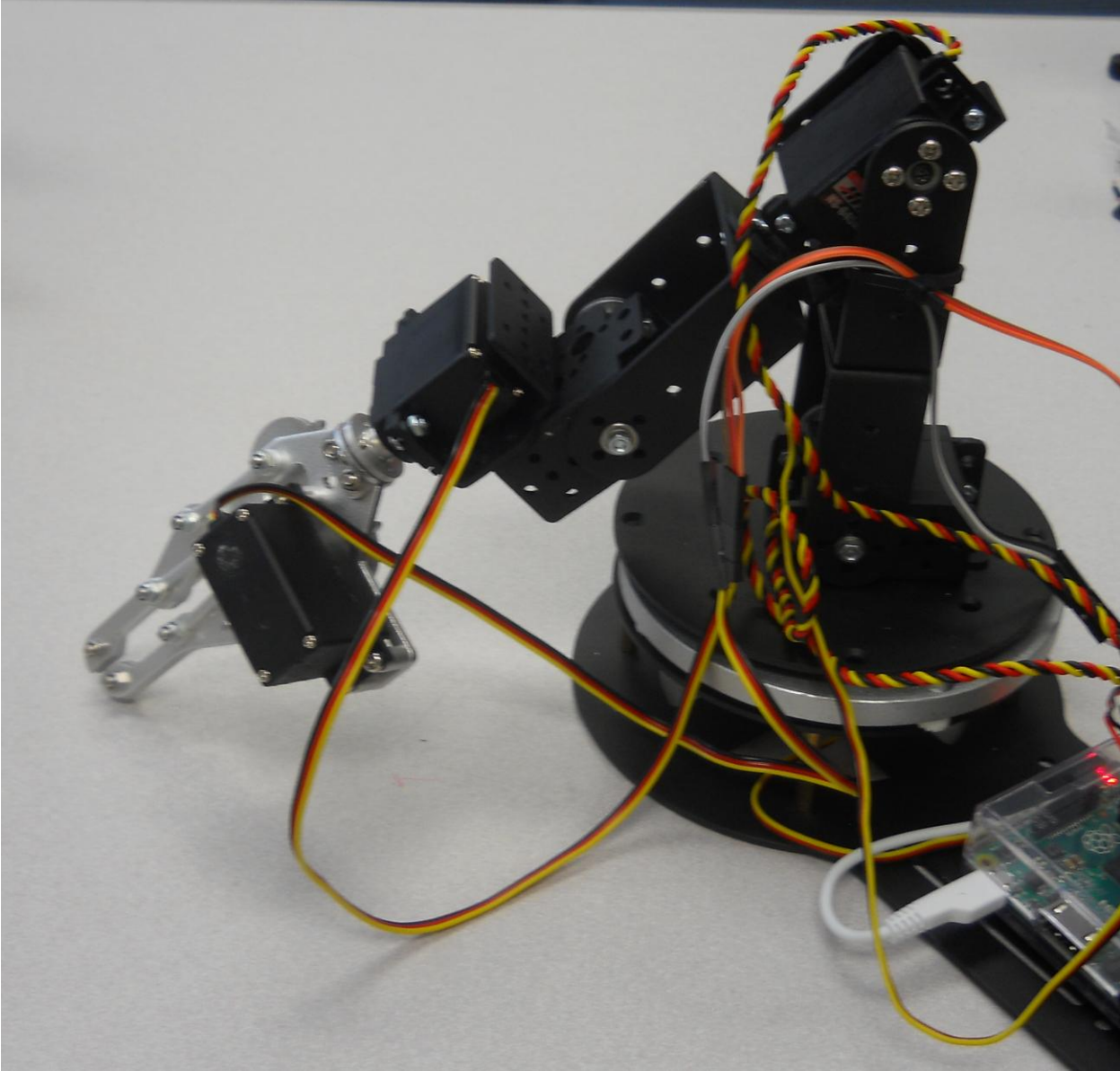


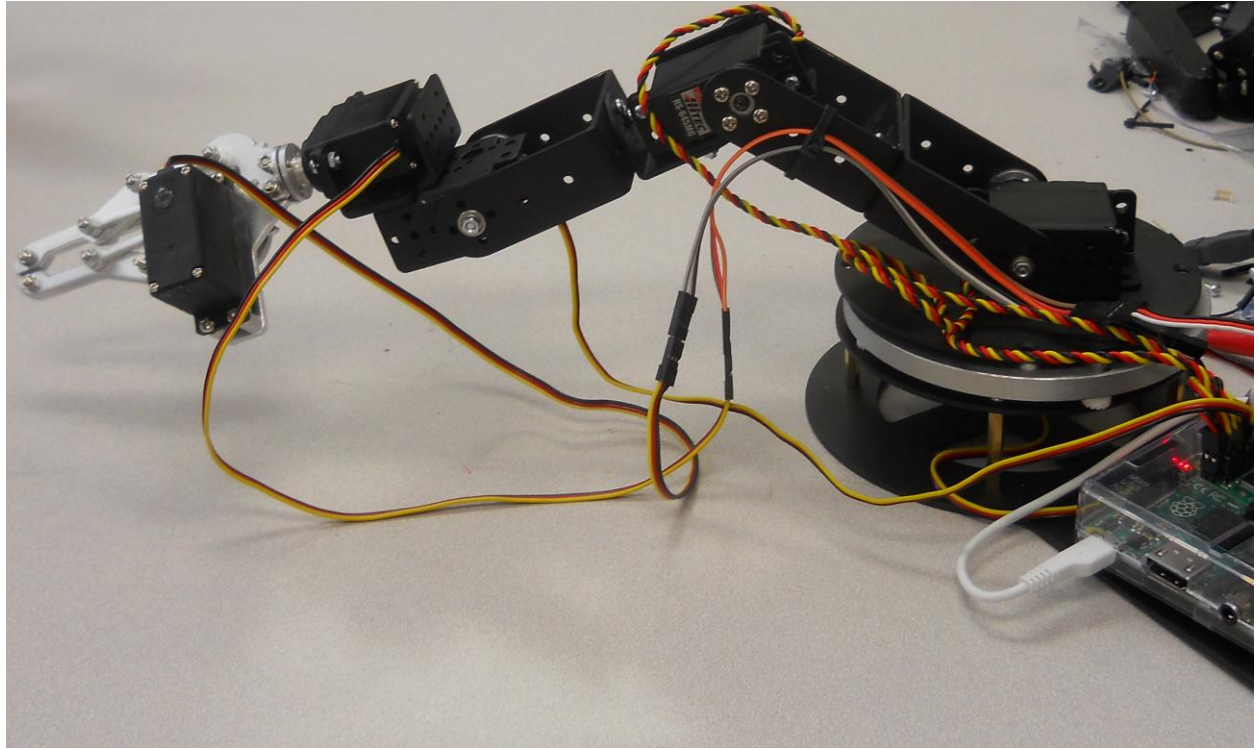
The image shows a terminal window on a Raspberry Pi. The window title is "pi@raspberrypi: ~/maestro-linux". The terminal content is a Python script for controlling a robot arm. The script imports the serial, time, and math modules, and uses the robotArmLib library. It sets up a serial connection to "/dev/ttyACM0" at 9600 baud. A while loop continuously reads user input for x and y coordinates, calculates the angle and magnitude, and sends commands to the robot arm to move the servos to the specified positions. The status bar at the bottom shows the file name "robotPos.py", line 5, and the language "Python".

```
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time
import math
from robotArmLib import *

ser = serial.Serial("/dev/ttyACM0", 9600)
while 1:
    x = int(raw_input("x: "))
    y = int(raw_input("y: "))
    speed = 5
    servo0 = 0
    servo2 = 2
    setSpeed(ser, servo0, speed)
    setSpeed(ser, servo2, speed)
    angle = int((math.atan2(y, x) * 360) / (2 * 3.1416))
    mag = int(math.hypot(x, y))
    print angle
    print mag
    setAngle(ser, servo0, angle + 40)
    setAngle(ser, servo2, mag + 40)
    time.sleep(.5)

-UU-:***-F1 robotPos.py All L5 (Python)
```

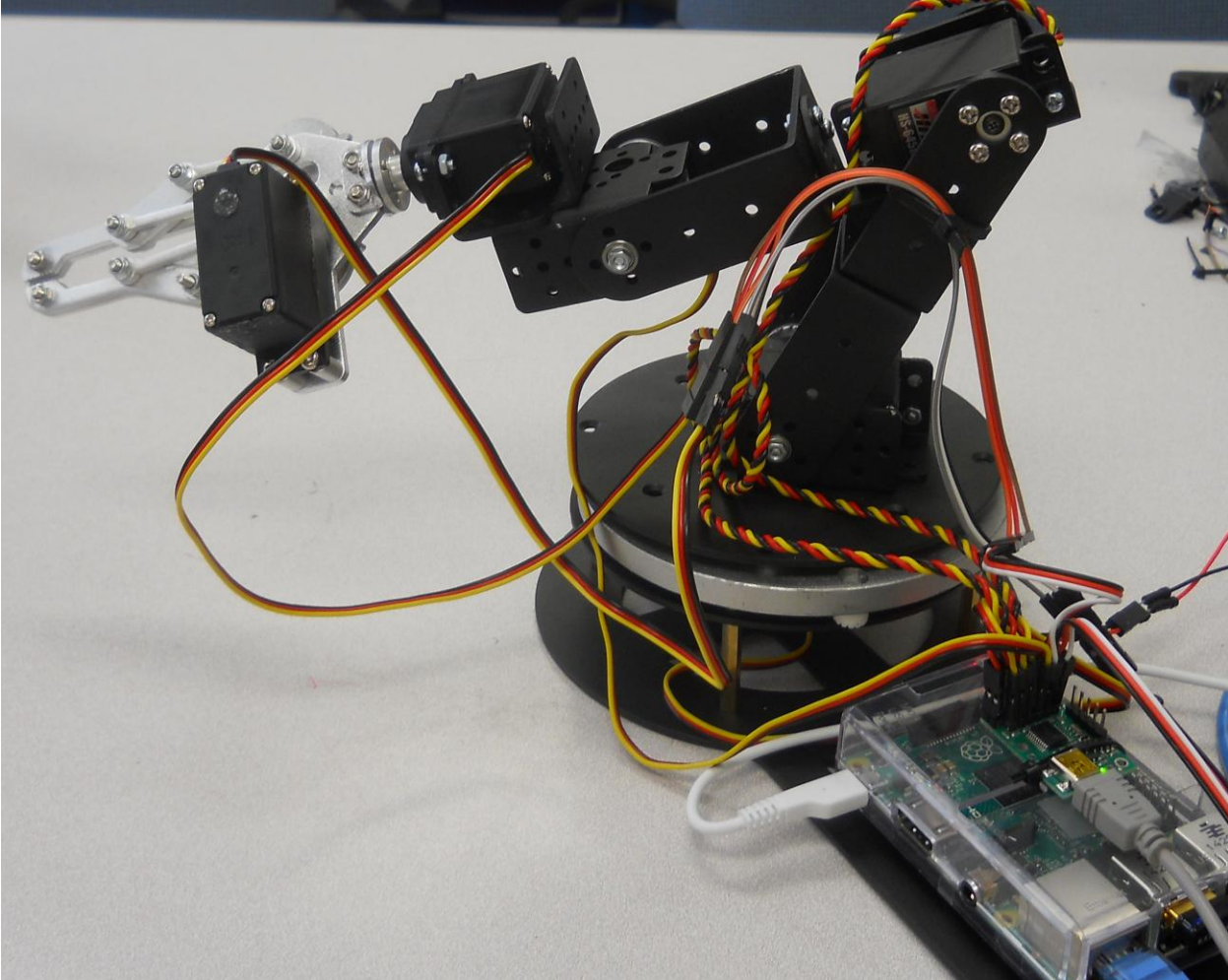




```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time
import math
from robotArmLib import *

ser = serial.Serial("/dev/ttyACM0", 9600)
while 1:
    x = int(raw_input("x: "))
    y = int(raw_input("y: "))
    speed = 5
    servo0 = 0
    servo1 = 1
    servo2 = 2
    setSpeed(ser, servo0, speed)
    setSpeed(ser, servo2, speed)
    angle = int((math.atan2(y, x)* 360)/(2 * 3.1416))
    mag = int(math.hypot(x, y))
    print angle
    mag2 = mag/2
    print mag2
    setAngle(ser, servo0, angle + 40)
    setAngle(ser, servo2, 160 - mag)
    setAngle(ser, servo1, 50 + mag2)
    time.sleep(.5)

-UU-:**--F1 robotPos.py All L8 (Python)-----
```

```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time
import math
from robotArmLib import *

def loadPen(ser):
    servo = 1
    print "Loading Pen"
    print "Enter servo 0 when loaded"
    print "Servo 5 - Claw"
    print "Servo 4 - Wrist"
    while servo != 0:
        servo = int(raw_input("Servo number: "))
        angle = int(raw_input("Angle: "))
        speed = 5
        setSpeed(ser, servo, speed)
        setAngle(ser, servo, angle)
        time.sleep(.1)

ser = serial.Serial("/dev/ttyACM0", 9600)
loadPen(ser)
while 1:
    x = int(raw_input("x: "))
    y = int(raw_input("y: "))
    speed = 5
    servo0 = 0
    servo1 = 1
    servo2 = 2
    setSpeed(ser, servo0, speed)
    setSpeed(ser, servo2, speed)
    angle = int((math.atan2(y, x) * 360) / (2 * 3.1416))
    mag = int(math.hypot(x, y))
    print angle
    mag2 = mag/2
    print mag2
    setAngle(ser, servo0, angle + 40)
    setAngle(ser, servo2, 160 - mag)
    setAngle(ser, servo1, 50 + mag2)
    time.sleep(.5)

-UU-:-----F1 robotPos.py All L1 (Python)-----
robotPos.py has auto save data; consider M-x recover-this-file
```

```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
import pygame, sys

black = 0,0,0
white = 0xFF, 0xFF, 0xFF
background_color = 0x12, 0x0E, 0x1C

width = 64
height = 32
scale = 8
canvas = pygame.Surface((width, height), pygame.SRCALPHA)
canvas.set_at((0,0), white)
canvas.set_at((width-1,height-1), black)

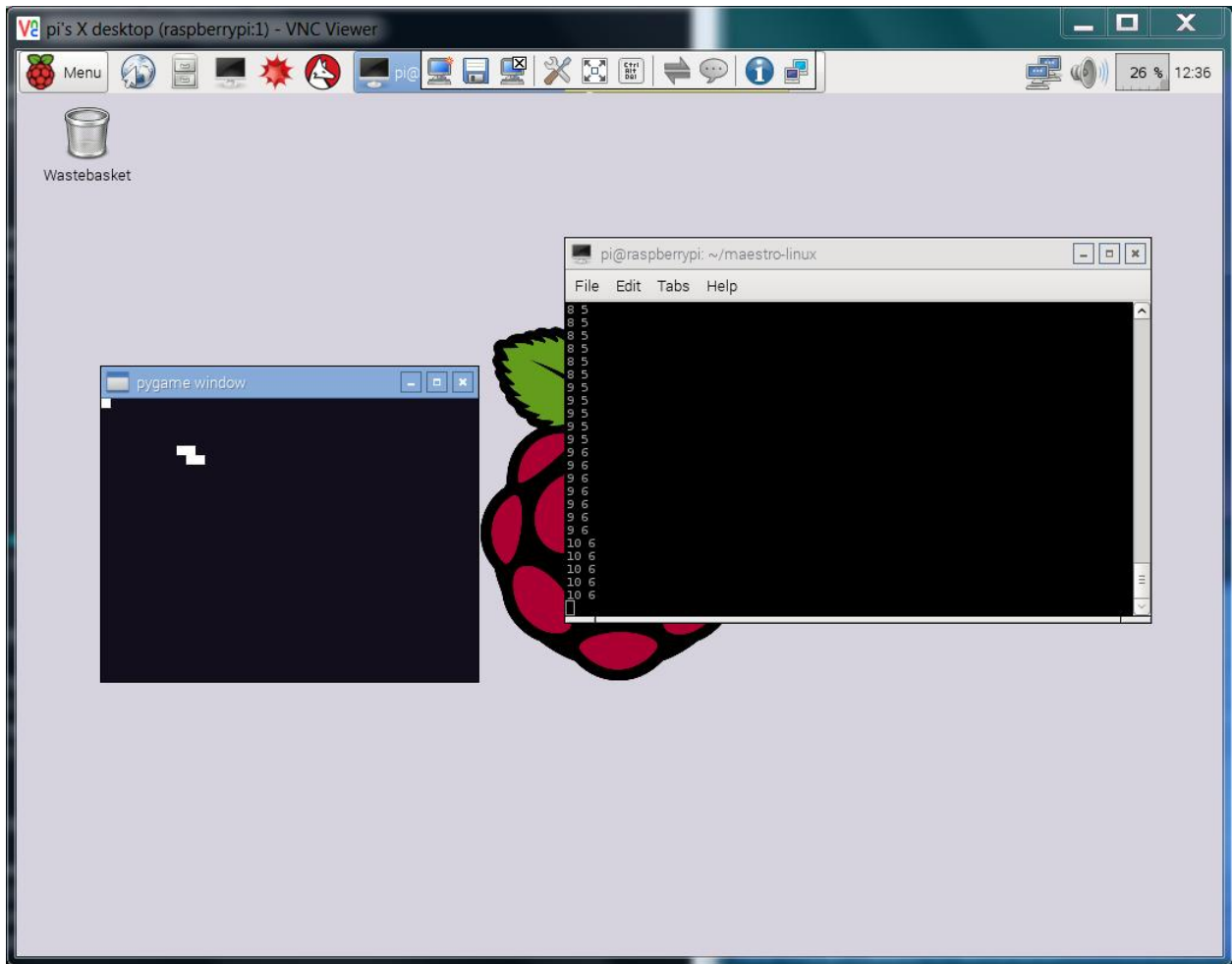
def animation_frame(screen):
    screen.fill(background_color)
    view = pygame.transform.scale(canvas, (width*scale, height*scale))
    screen.blit(view, (0, 0))

def plot((x,y)):
    x = int(x/scale)
    y = int(y/scale)
    print x, y
    if 0 <= x < width and 0 <= y < height:
        canvas.set_at((x,y), white)

def dispatch(event):
    if event.type == pygame.QUIT:
        sys.exit(0)
    if event.type == pygame.MOUSEBUTTONDOWN:
        plot(event.pos)
    if event.type == pygame.MOUSEMOTION and event.buttons != (0,0,0):
        plot(event.pos)

pygame.display.init()
screen = pygame.display.set_mode((320, 240))
while 1:
    for event in pygame.event.get():
        dispatch(event)
    animation_frame(screen)
    pygame.display.flip()

-UU-:----F1 draw.py All L1 (Python)-----
For information about GNU Emacs and the GNU system, type C-h C-a.
```



```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time
import math
from robotArmLib import *

def loadPen(ser):
    servo = 1
    print "Loading Pen"
    print "Enter servo 0 when loaded"
    print "Servo 5 - Claw"
    print "Servo 4 - Wrist"
    while servo != 0:
        servo = int(raw_input("Servo number: "))
        angle = int(raw_input("Angle: "))
        speed = 5
        setSpeed(ser, servo, speed)
        setAngle(ser, servo, angle)
        time.sleep(.1)

def setPos(ser, x, y):
    speed = 5
    servo0 = 0
    servo1 = 1
    servo2 = 2
    setSpeed(ser, servo0, speed)
    setSpeed(ser, servo2, speed)
    angle = int((math.atan2(y, x) * 360) / (2 * 3.1416))
    mag = int(math.hypot(x, y))
    print angle
    mag2 = mag/2
    print mag2
    setAngle(ser, servo0, angle + 40)
    setAngle(ser, servo2, 160 - mag)
    setAngle(ser, servo1, 50 + mag2)

-UU-:----F1 robotPosLib.py All L25 (Python)-----
Wrote /home/pi/maestro-linux/robotPosLib.py
```



```
pi@raspberrypi: ~/maestro-linux
File Edit Options Buffers Tools Python Help
import pygame, sys
from robotPosLib import *

black = 0,0,0
white = 0xFF, 0xFF, 0xFF
background_color = 0x12, 0x0E, 0x1C
ser = serial.Serial("/dev/ttyACM0", 9600)

width = 64
height = 32
scale = 8
canvas = pygame.Surface((width, height), pygame.SRCALPHA)
canvas.set_at((0,0), white)
canvas.set_at((width-1,height-1), black)

def animation_frame(screen):
    screen.fill(background_color)
    view = pygame.transform.scale(canvas, (width*scale, height*scale))
    screen.blit(view, (0, 0))

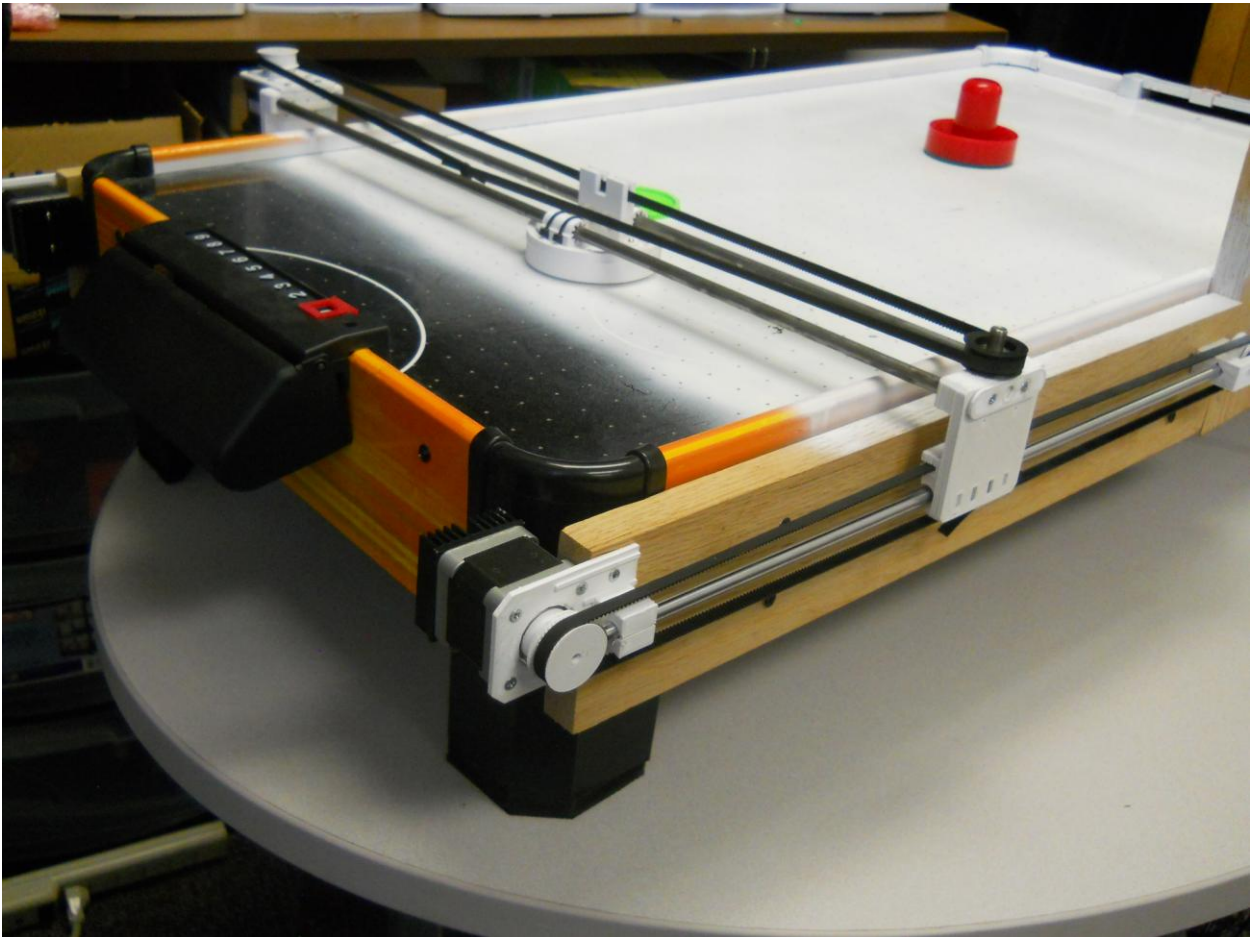
def plot((x,y)):
    x = int(x/scale)
    y = int(y/scale)
    print x, y
    setPos(ser, x + 50, y)
    if 0 <= x < width and 0 <= y < height:
        canvas.set_at((x,y), white)

def dispatch(event):
    if event.type == pygame.QUIT:
        sys.exit(0)
    if event.type == pygame.MOUSEBUTTONDOWN:
        plot(event.pos)
    if event.type == pygame.MOUSEMOTION and event.buttons != (0,0,0):
        plot(event.pos)

pygame.display.init()
screen = pygame.display.set_mode((320, 240))
while 1:
    for event in pygame.event.get():
        dispatch(event)
    animation_frame(screen)
    pygame.display.flip()

-UU-:----F1 robotDraw.py All L1 (Python)-----
```

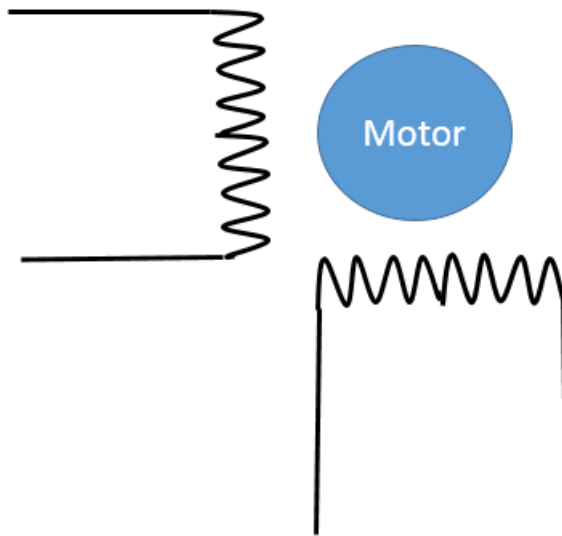
Chapter 6: A Robot That Can Play Air Hockey



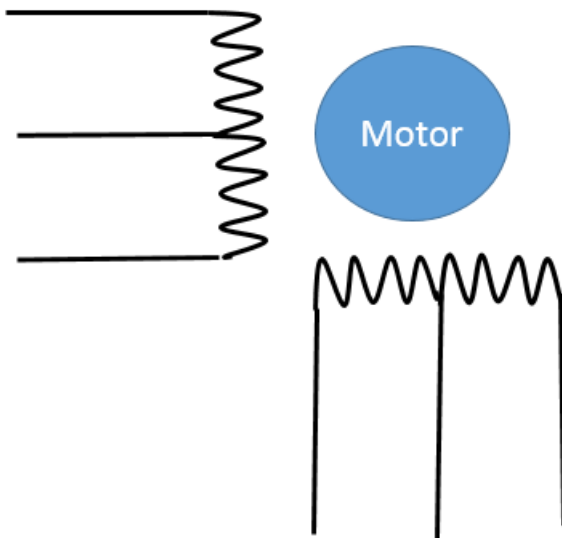


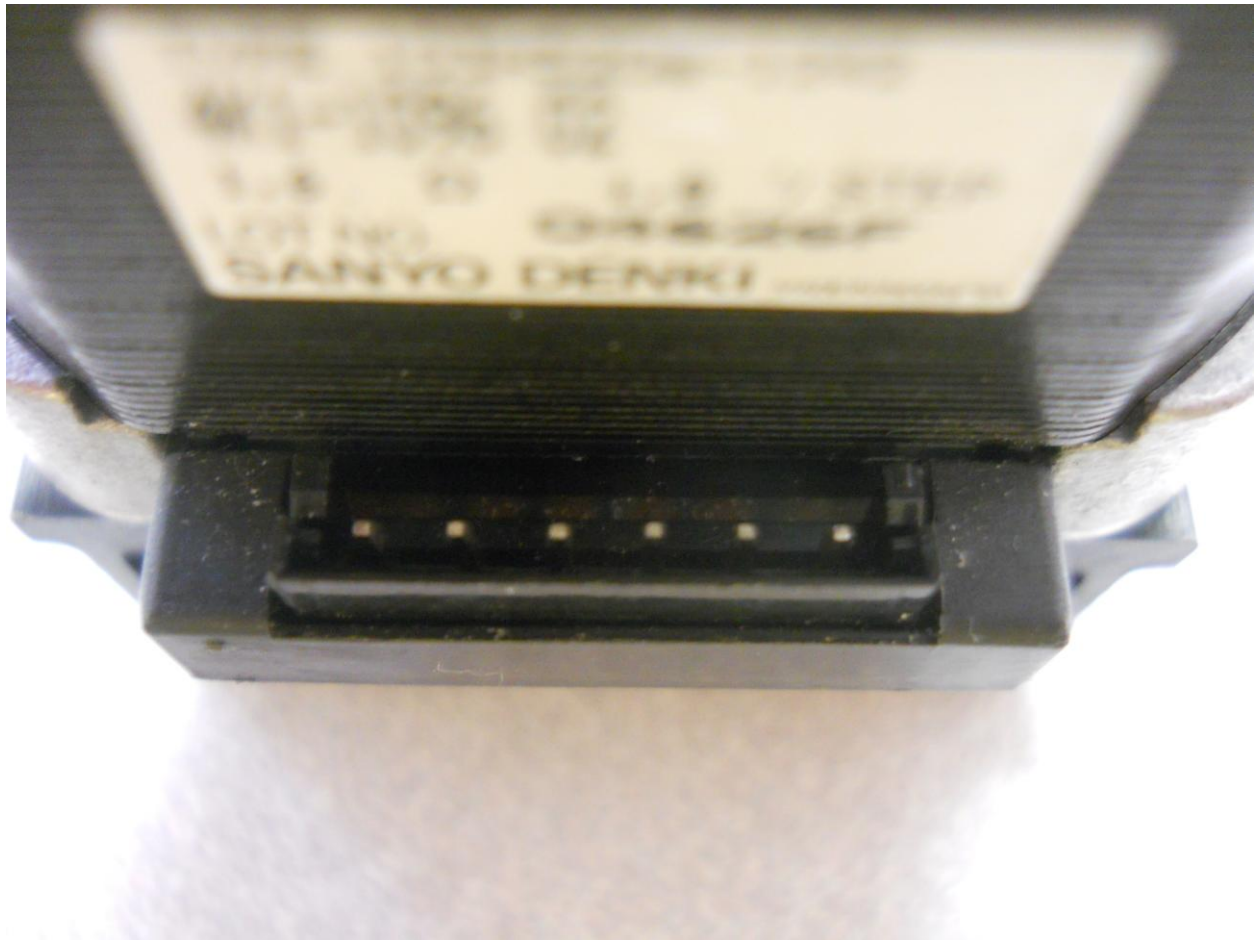
TYPE 103H5208-1242
4K1-1096 02
1.6 Ω 1.8 °/STEP
LOT NO. 04626F
SANYO DENKI MADE IN JAPAN

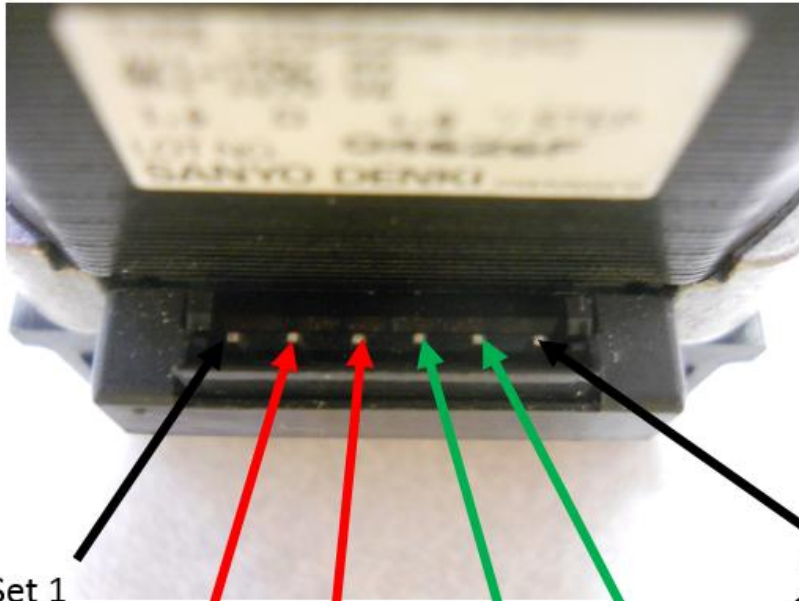
UniPolar Wiring



BiPolar Wiring







Set 1
Center

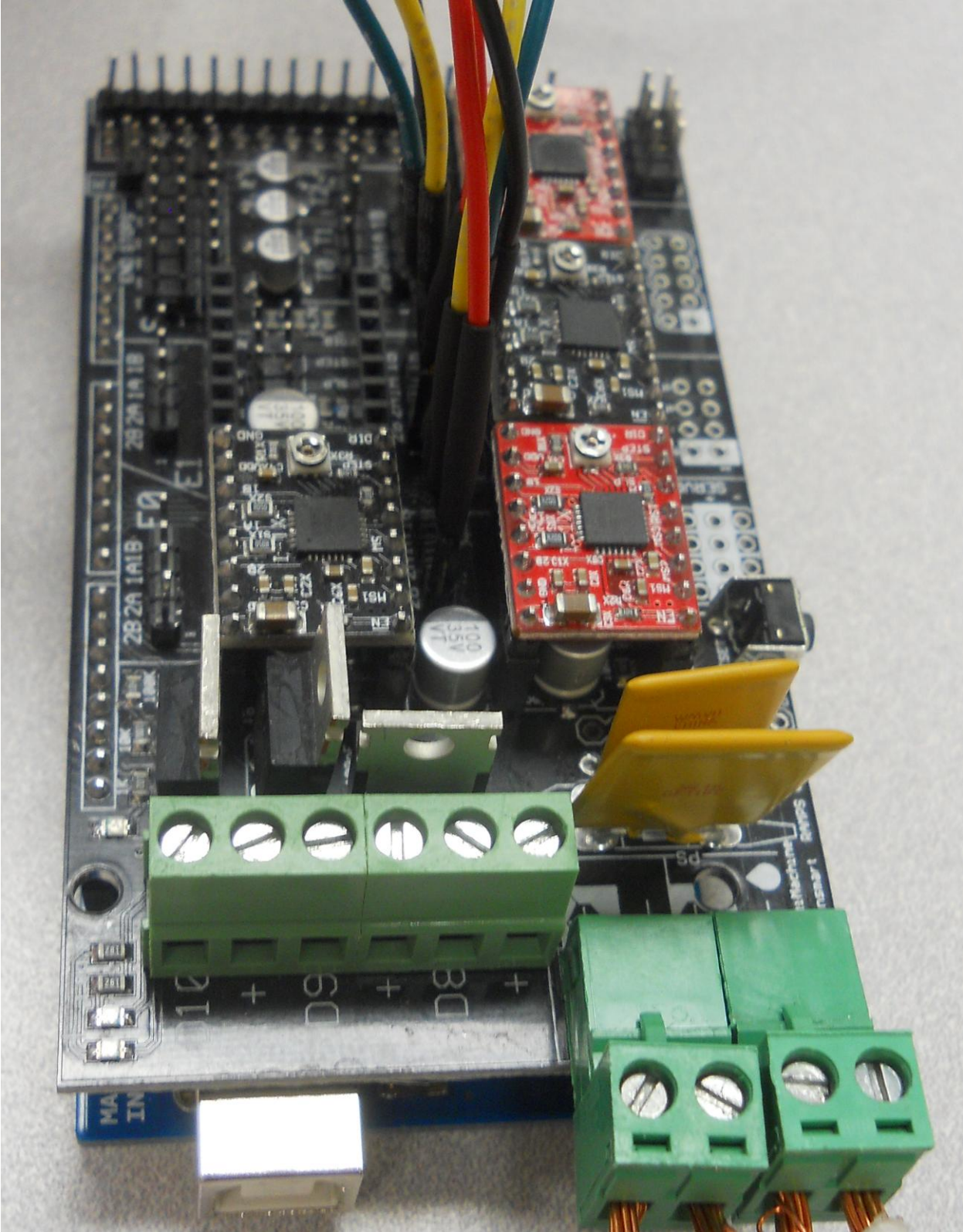
Set 1
Wire 1

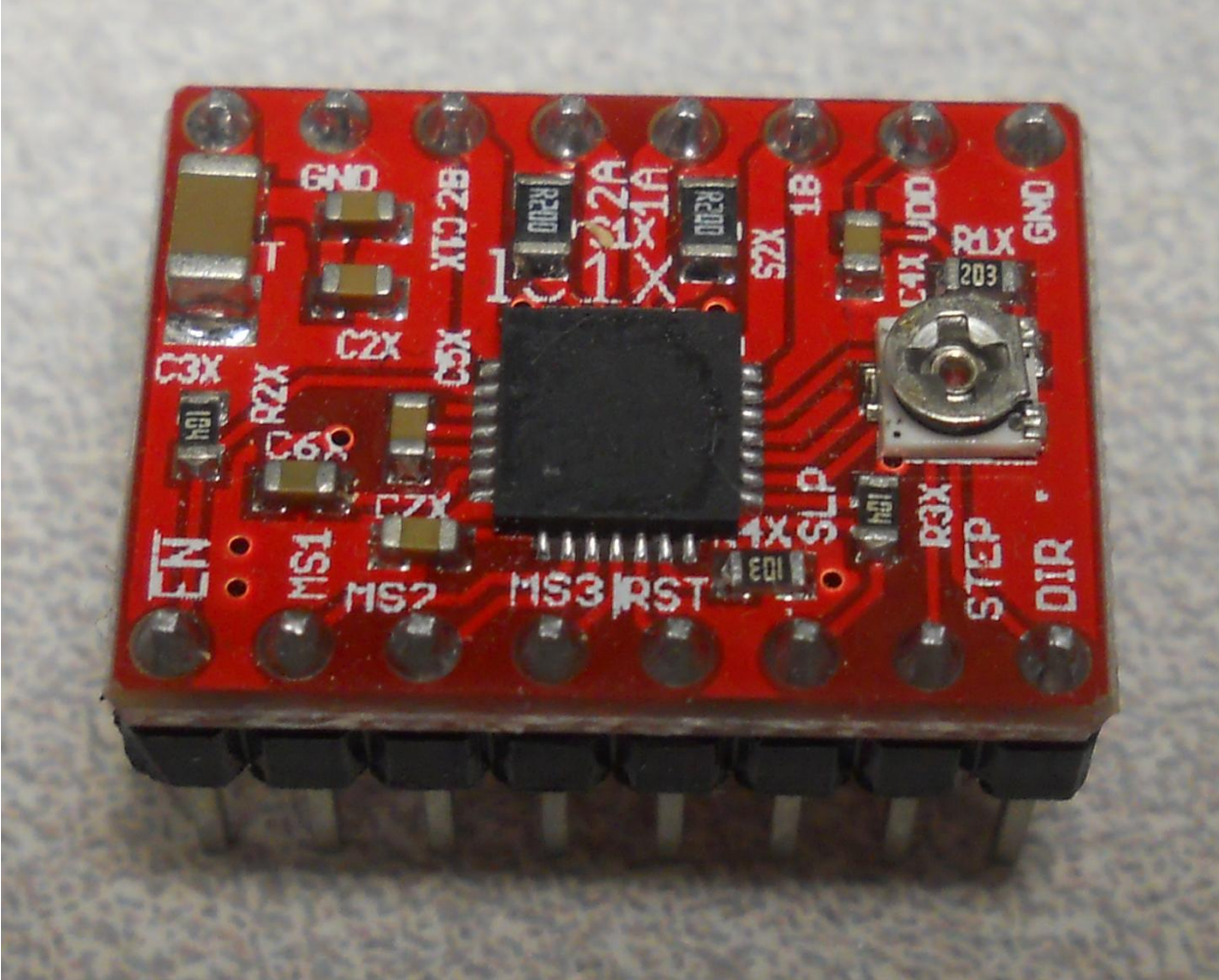
Set 1
Wire 2

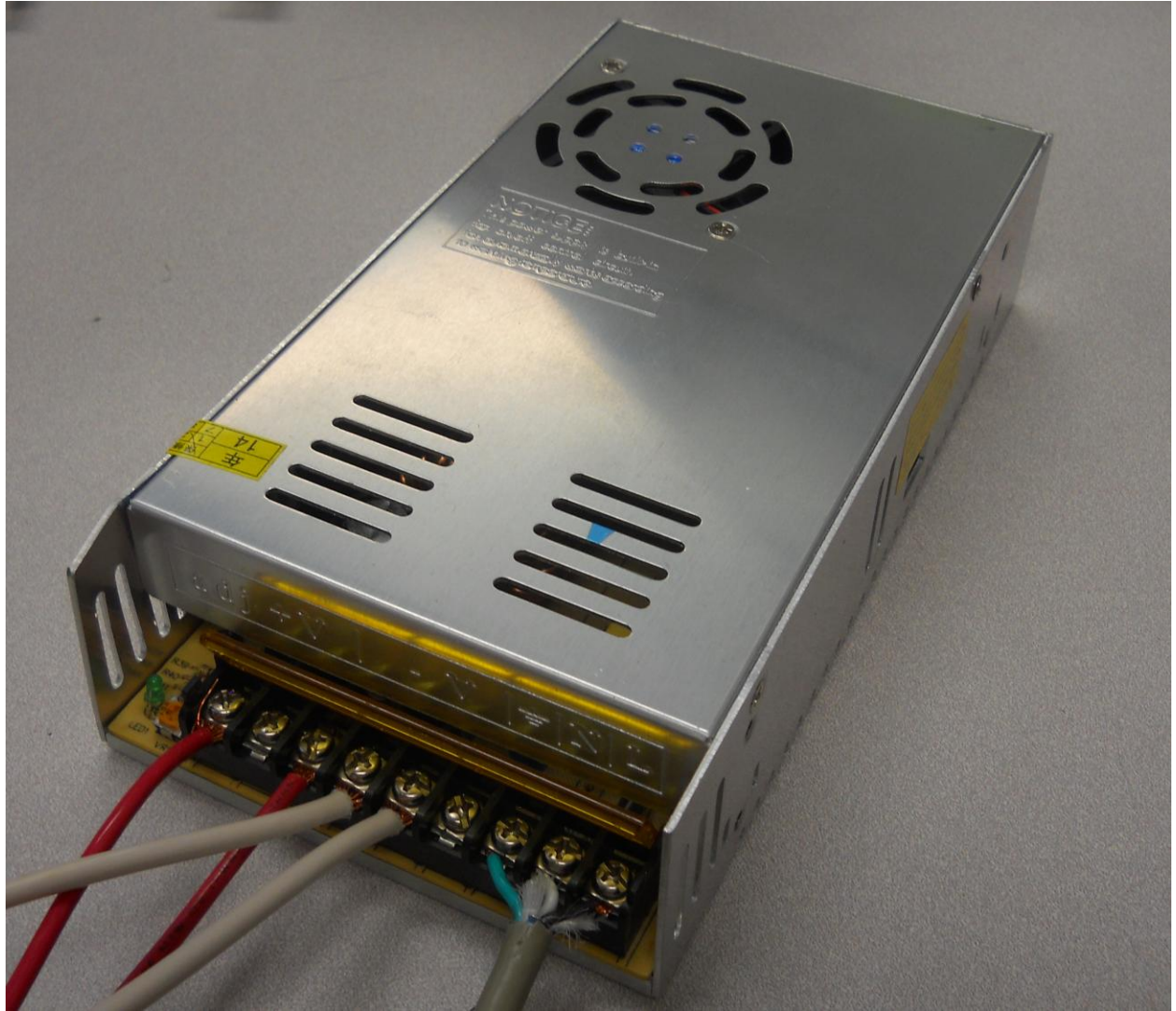
Set 2
Wire 1

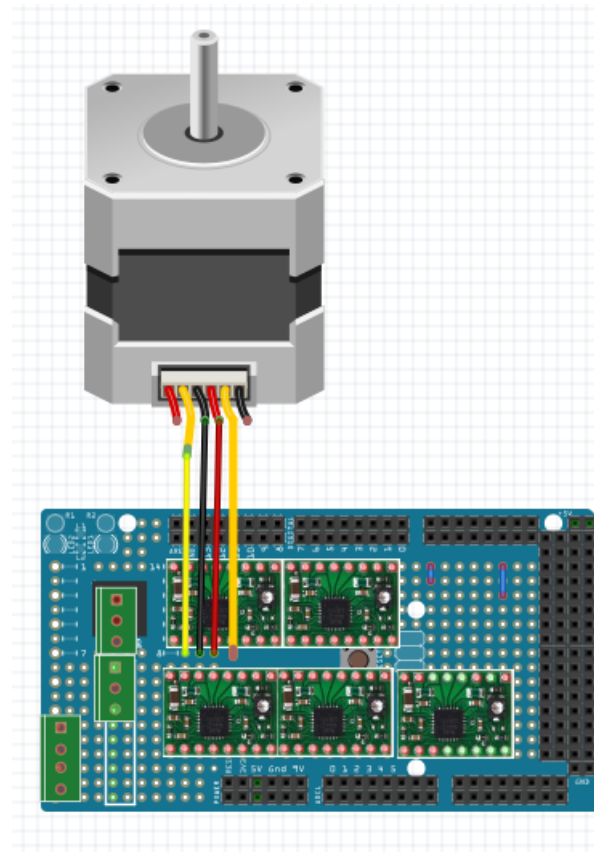
Set 2
Wire 2

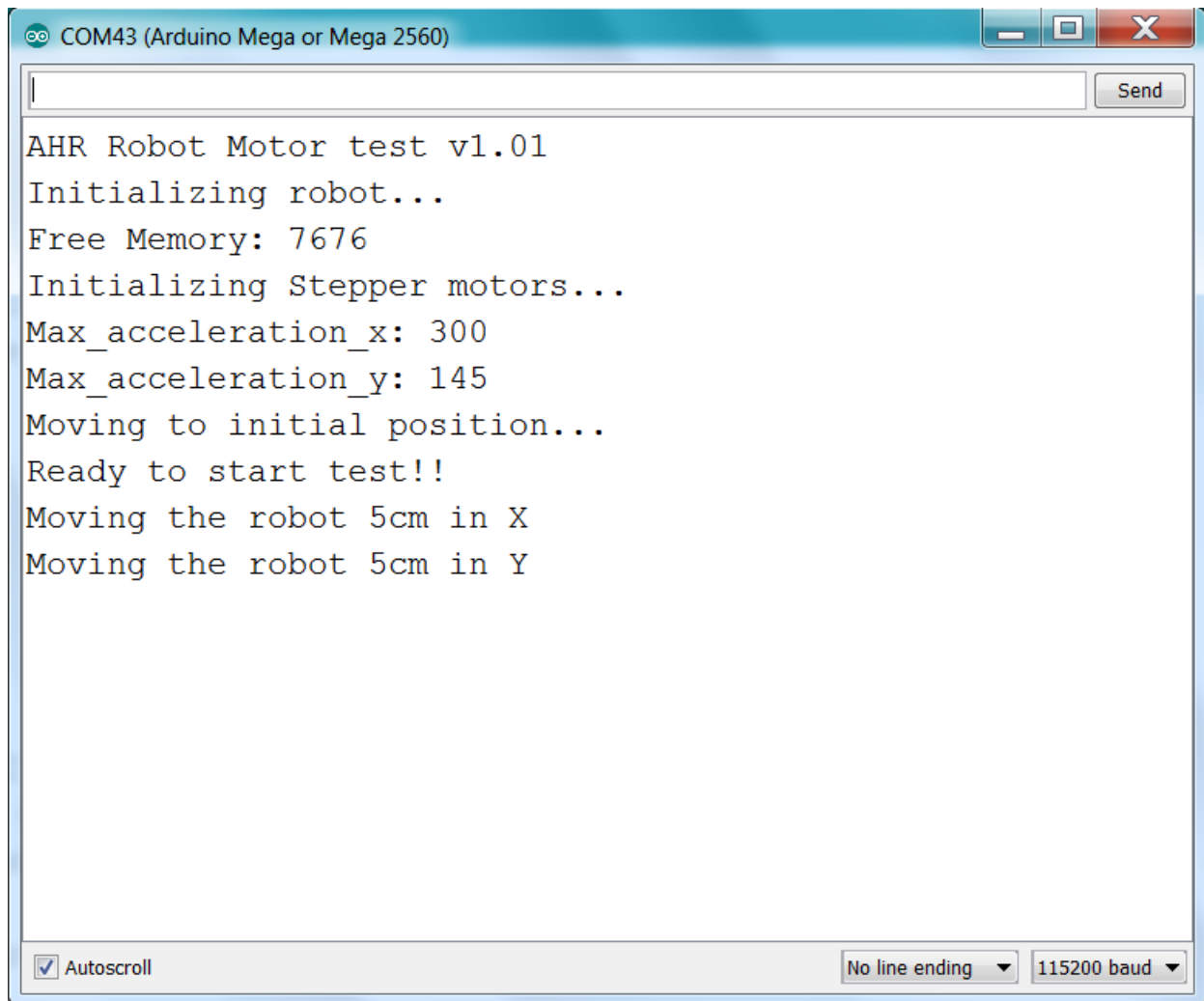
Set 2
Center











```
AHR_Motor_Test | Arduino 1.6.1
File Edit Sketch Tools Help
AHR_Motor_Test Configuration.h Definitions.h Steppers
// #define MAX_ACCEL_Y 200 //140//220
// #define MAX_SPEED_X 28000 //max 25000 for 12V // Maximun speed in steps/seg
// #define MAX_SPEED_Y 28000

// This is for the Accel ramp implementation (to smooth the intial acceleration), simplified S-profile
#define ACCEL_RAMP_MIN 2500 // The S profile is generated up to this speed
#define ACCEL_RAMP_MAX 10000

// UNCOMMENT THIS LINES TO INVERT MOTORS
#define INVERT_X_AXIS 1
#define INVERT_Y_AXIS 1 //Y-LEFT
// #define INVERT_Z_AXIS 1 //Y_RIGHT
```

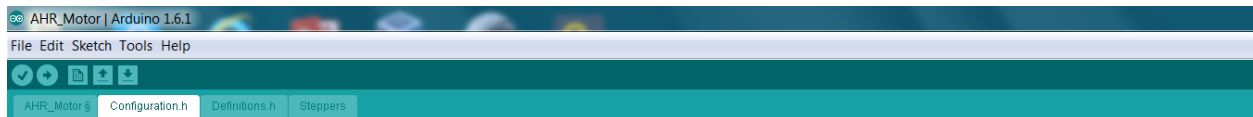
```

#define ACCEL_RAMP_MIN 2500 // The S profile is generated up to this speed
#define ACCEL_RAMP_MAX 10000
|
// UNCOMMENT THIS LINES TO INVERT MOTORS
#define INVERT_X_AXIS 1
#define INVERT_Y_AXIS 1 //Y-LEFT
//#define INVERT_Z_AXIS 1 //Y_RIGHT

// Geometric calibration.
// This depends on the pulley teeth. For 42 teeth GT2 => 19, for 40 teeth GT2 => 20, for 16 teeth T5 => 20
#define X_AXIS_STEPS_PER_UNIT 10 // With 42 teeth GT2 pulley and 1/8 microstepping on drivers
#define Y_AXIS_STEPS_PER_UNIT 10 // 200*8 = 1600 steps/rev = 1600/42teeth*2mm = 19.047, using 19 is an er

// Absolute Min and Max robot positions in mm (measured from center of robot pusher)
#define ROBOT_MIN_X 100
#define ROBOT_MIN_Y 80
#define ROBOT_MAX_X 500
#define ROBOT_MAX_Y 400

```



```

// Absolute Min and Max robot positions in mm (measured from center of robot pusher)
#define ROBOT_MIN_X 100
#define ROBOT_MIN_Y 80
#define ROBOT_MAX_X 500
#define ROBOT_MAX_Y 400

// This is the center of the table. All units in milimeters
#define ROBOT_CENTER_X 300 // Center of robot. The table is 600x1000mm, so center is 300,500
#define ROBOT_CENTER_Y 500

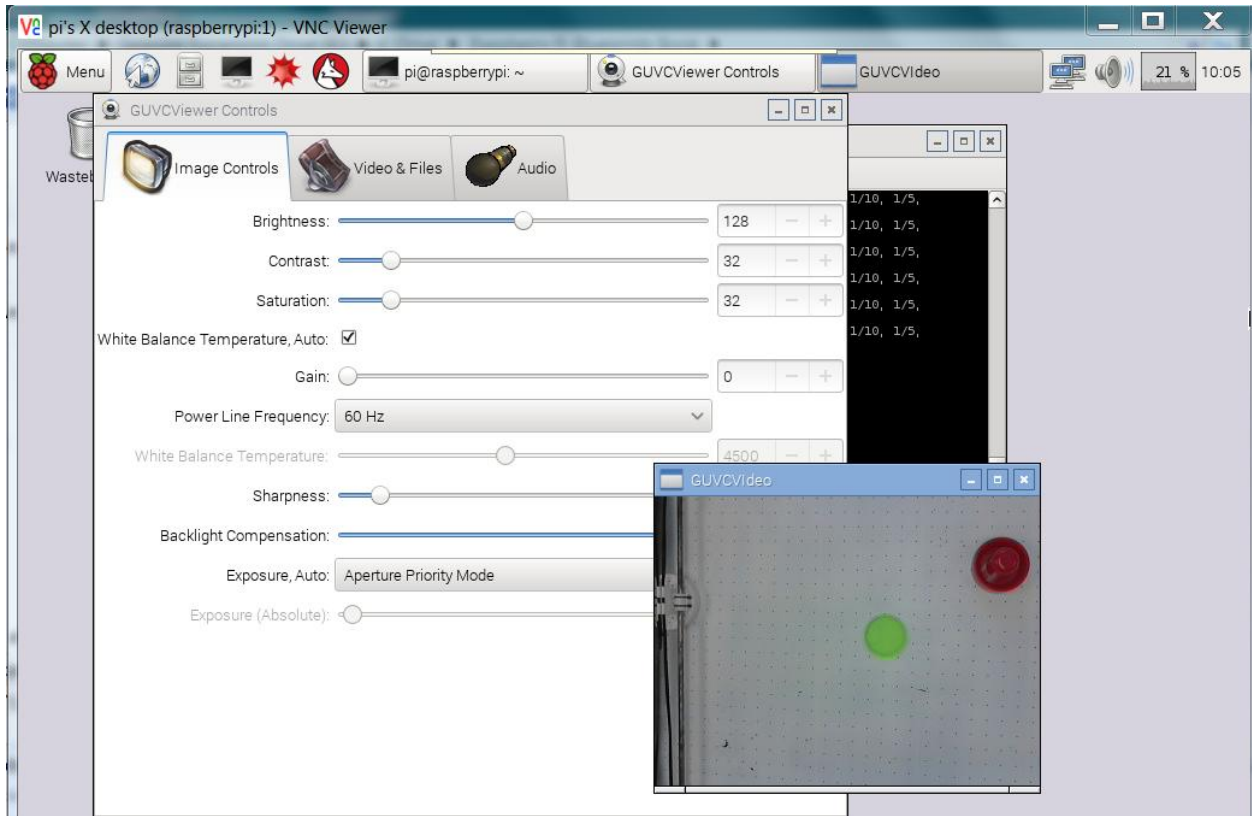
// Initial robot position in mm
// The robot must be at this position at start time
// Default: Centered in X and minimum position in Y
#define ROBOT_INITIAL_POSITION_X 300
#define ROBOT_INITIAL_POSITION_Y 45 // Measured from center of the robot pusher to the table border

// Robot defense and attack lines
#define ROBOT_DEFENSE_POSITION 95
#define ROBOT_DEFENSE_ATTACK_POSITION 220

```

```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ lsusb  
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.  
Bus 001 Device 004: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter  
Bus 001 Device 005: ID 413c:3012 Dell Computer Corp. Optical Wheel Mouse  
Bus 001 Device 008: ID 046d:0825 Logitech, Inc. Webcam C270  
Bus 001 Device 007: ID 413c:2003 Dell Computer Corp. Keyboard  
pi@raspberrypi ~ $ █
```

```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ ls /dev/v*  
/dev/vc-cma /dev/vcs /dev/vcs4 /dev/vcsa /dev/vcsa4 /dev/vcsm  
/dev/vchiq /dev/vcs1 /dev/vcs5 /dev/vcsa1 /dev/vcsa5 /dev/vhci  
/dev/vcio /dev/vcs2 /dev/vcs6 /dev/vcsa2 /dev/vcsa6 /dev/video0  
/dev/vc-mem /dev/vcs3 /dev/vcs7 /dev/vcsa3 /dev/vcsa7  
  
/dev/v4l:  
by-id by-path  
pi@raspberrypi ~ $ █
```



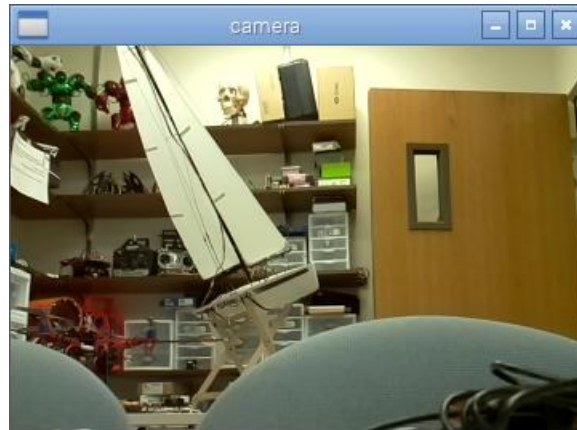
```
pi@raspberrypi: ~/examples/python
File Edit Options Buffers Tools Python Help
import cv2.cv as cv
import time

cv.NamedWindow("camera", 1)

capture = cv.CaptureFromCAM(0)
cv.SetCaptureProperty(capture, 3, 360)
cv.SetCaptureProperty(capture, 4, 240)

while True:
    img = cv.QueryFrame(capture)
    cv.ShowImage("camera", img)
    if cv.WaitKey(10) == 27:
        break
cv.DestroyAllWindows()

-UU-:----F1 myCamera.py All L6 (Python)-----
Wrote /home/pi/examples/python/myCamera.py
```

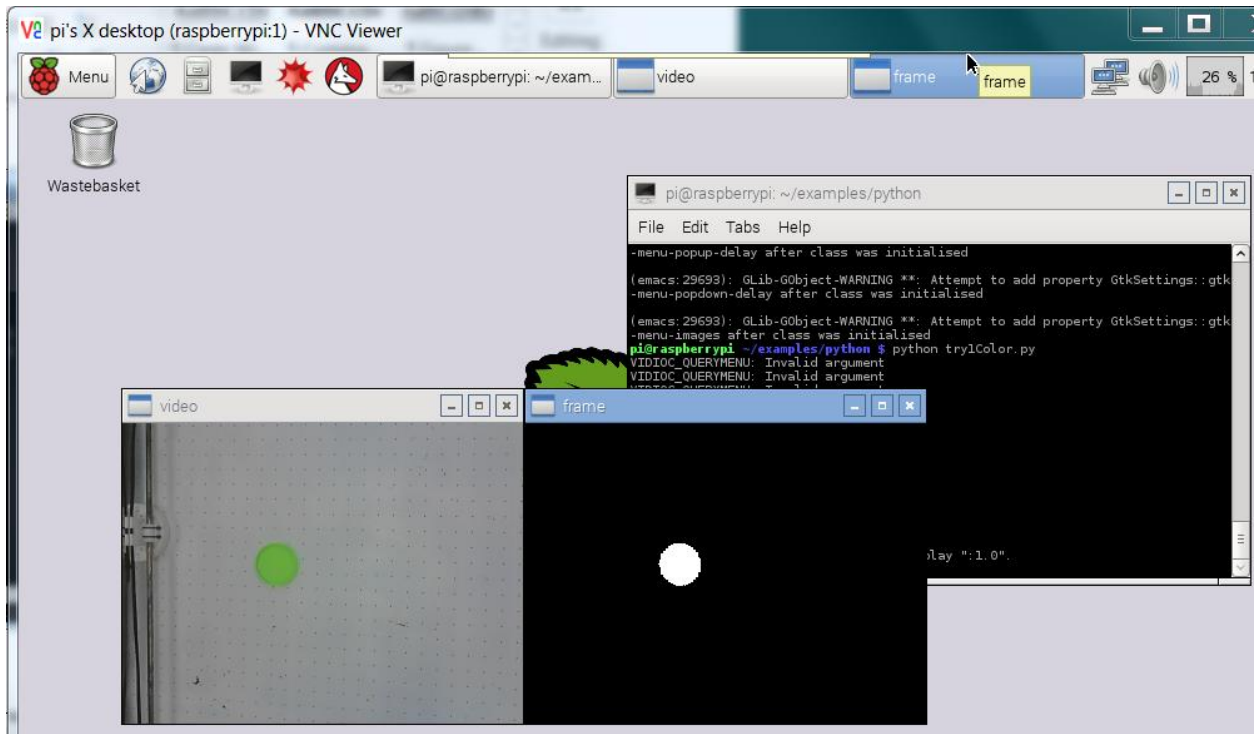



```
pi@raspberrypi: ~/examples/python
File Edit Options Buffers Tools Python Help
import numpy as np
import cv2

cap = cv2.VideoCapture(0)
cap.set(3,320)
cap.set(4,240)
low_range = np.array([10, 120, 100])
high_range = np.array([70, 255, 255])

while(cap.isOpened()):
    ret, frame = cap.read()
    hue_image = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    threshold_img = cv2.inRange(hue_image, low_range, high_range)
    cv2.imshow('video',frame)
    cv2.imshow('frame',threshold_img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()

-UU-:----F1  try1.py      All L18      (Python)-----
Wrote /home/pi/examples/python/try1.py
```



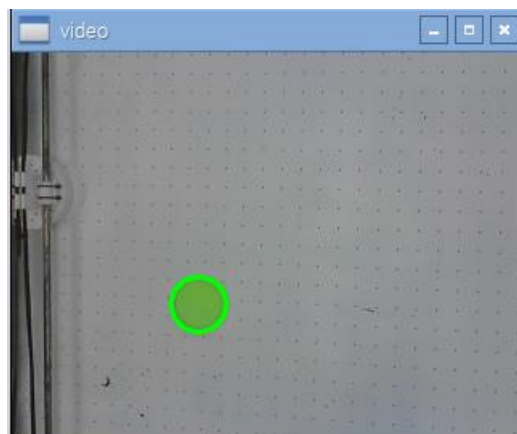
```
pi@raspberrypi: ~/examples/python
File Edit Options Buffers Tools Python Help
import numpy as np
import cv2

cap = cv2.VideoCapture(0)
cap.set(3,320)
cap.set(4,240)
low_range = np.array([10, 120, 100])
high_range = np.array([70, 255, 255])

while(cap.isOpened()):
    ret, frame = cap.read()
    hue_image = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    threshold_img = cv2.inRange(hue_image, low_range, high_range)
    contour, hierarchy = cv2.findContours(threshold_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    center = contour[0]
    moment = cv2.moments(center)
    (x,y),radius = cv2.minEnclosingCircle(center)
    center = (int(x),int(y))
    radius = int(radius)
    img = cv2.circle(frame,center,radius,(0,255,0),2)
    cv2.imshow('video',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

-UU-:***--F1  try1.py      All L22      (Python)-----
```



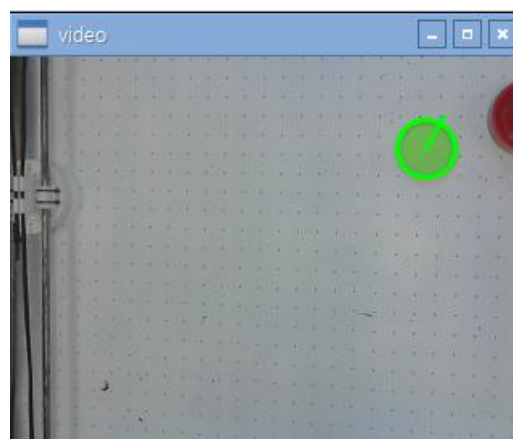
```
pi@raspberrypi: ~/examples/python
File Edit Options Buffers Tools Python Help
import cv2
import numpy as np

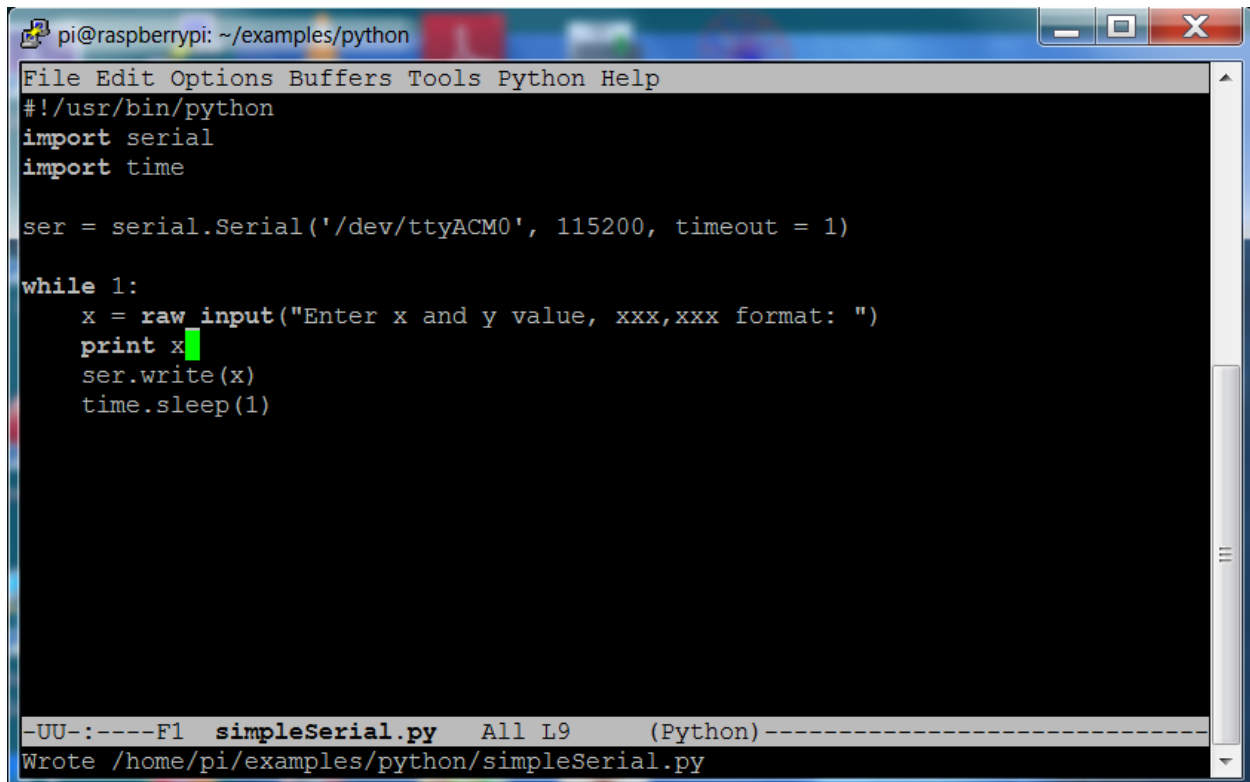
cap = cv2.VideoCapture(0)
cap.set(3, 360)
cap.set(4, 240)
low_range = np.array([10, 120, 100])
high_range = np.array([70, 255, 255])
lastX = 0
lastY = 0
deltaX = 0
deltaY = 0

while (cap.isOpened()):
    ret, frame = cap.read()
    hue_image = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    threshold_img = cv2.inRange(hue_image, low_range, high_range)
    contour, hierarchy = cv2.findContours(threshold_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    if contour:
        center = contour[0]
        moment = cv2.moments(center)
        (x,y),radius = cv2.minEnclosingCircle(center)
        center = (int(x), int(y))
        deltaX = int(x) - lastX
        deltaY = int(y) - lastY
        lastX = int(x)
        lastY = int(y)
        radius = int(radius)
        img = cv2.circle(frame, center, radius, (0, 255, 0), 2)
        img = cv2.line(frame, (lastX,lastY), (lastX + deltaX, lastY + deltaY), (0, 255, 0), 2)
        cv2.imshow('video', frame)
        if cv2.waitKey(10) == 27:
            break

cap.release
cv2.destroyAllWindows()

-UU-:***-F1 trackPuck.py All L32 (Python)-----
```





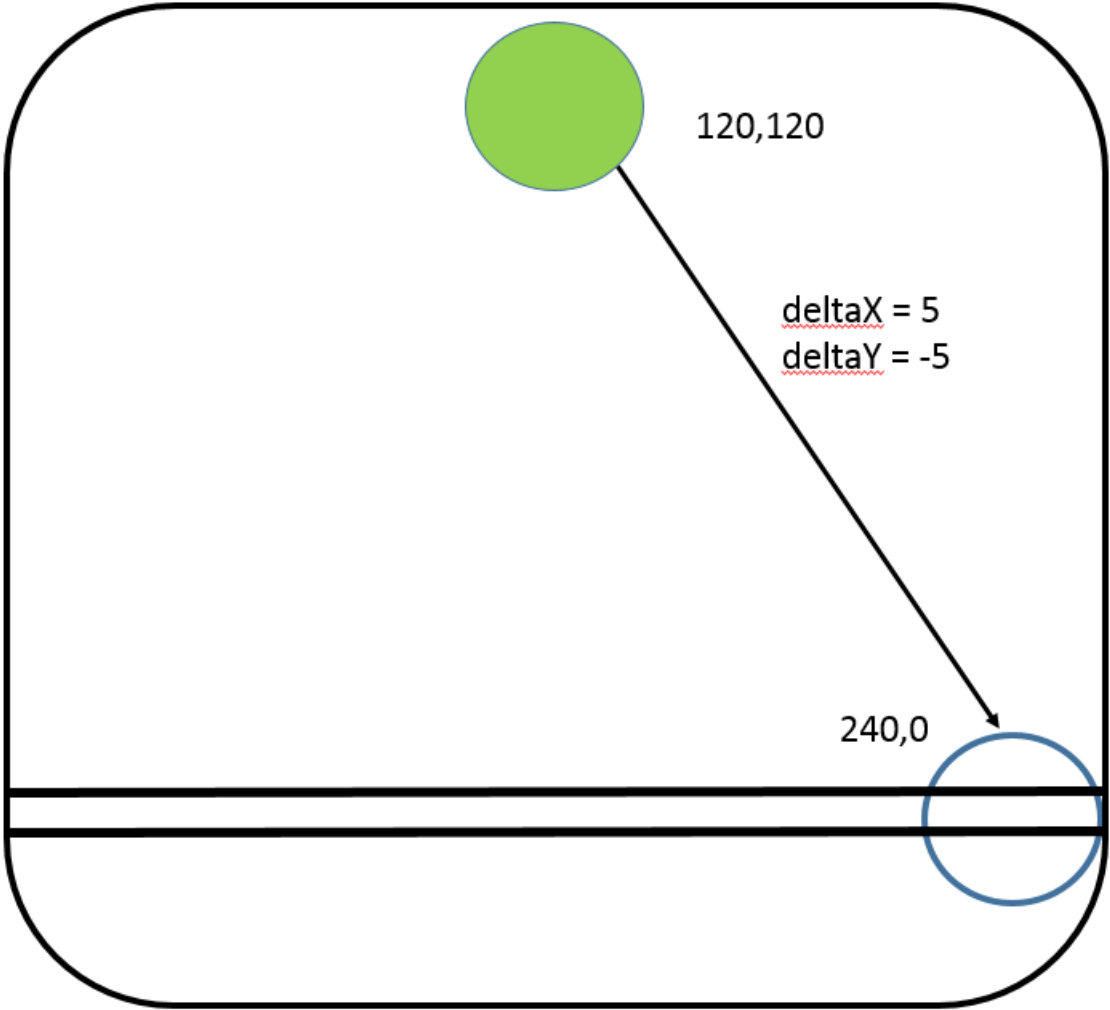
The image shows a terminal window on a Raspberry Pi. The window title is "pi@raspberrypi: ~/examples/python". The terminal content displays a Python script for serial communication. The script imports the 'serial' and 'time' modules, initializes a serial object 'ser' with parameters for the device path, baud rate, and timeout, and enters a 'while' loop that prompts for input, prints it, writes it to the serial port, and sleeps for 1 second. The status bar at the bottom indicates the file is 'simpleSerial.py', line 9, and the editor is in Python mode. A message at the bottom states "Wrote /home/pi/examples/python/simpleSerial.py".

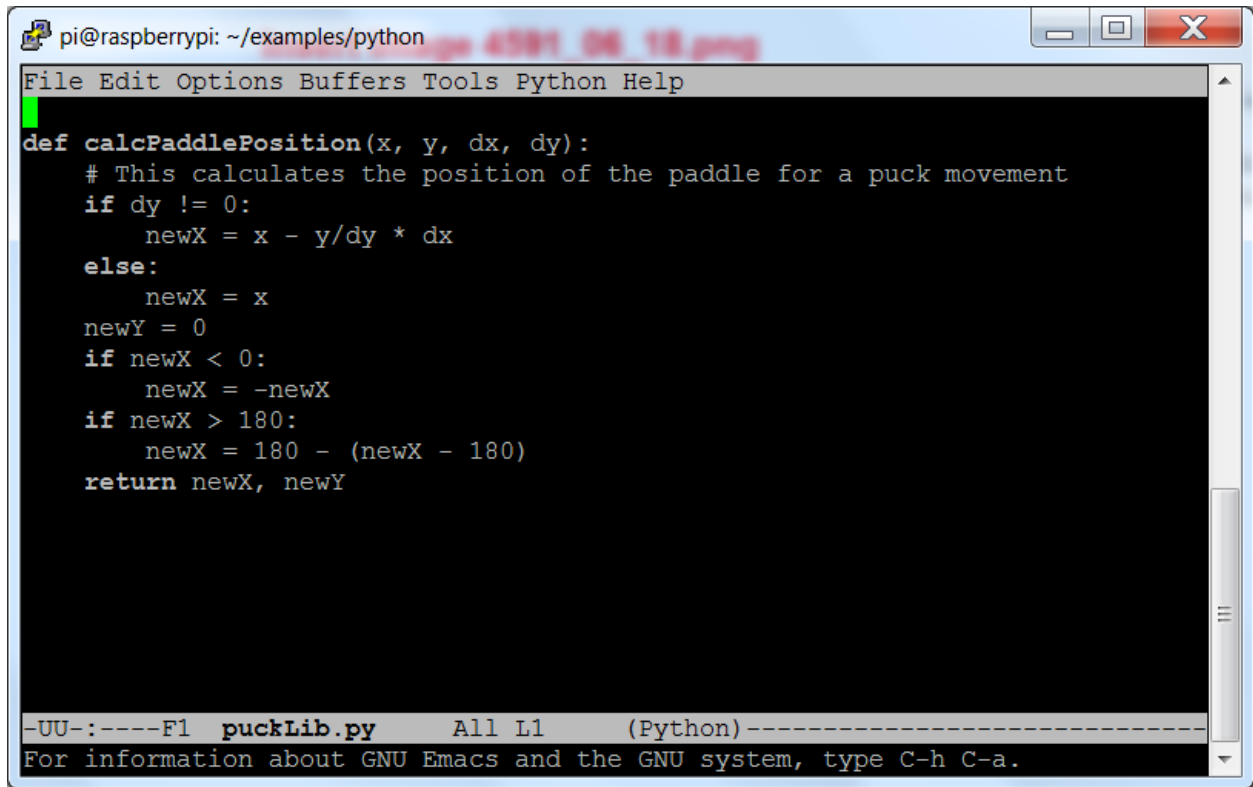
```
File Edit Options Buffers Tools Python Help
#!/usr/bin/python
import serial
import time

ser = serial.Serial('/dev/ttyACM0', 115200, timeout = 1)

while 1:
    x = raw input("Enter x and y value, xxx,xxx format: ")
    print x
    ser.write(x)
    time.sleep(1)

-UU-:----F1 simpleSerial.py All L9 (Python)-----
Wrote /home/pi/examples/python/simpleSerial.py
```



The image shows a terminal window on a Raspberry Pi. The window title is "pi@raspberrypi: ~/examples/python". The terminal content displays a Python function definition for `calcPaddlePosition` within an Emacs editor interface. The function calculates the new position of a paddle based on its current position and the movement of a puck. The code is as follows:

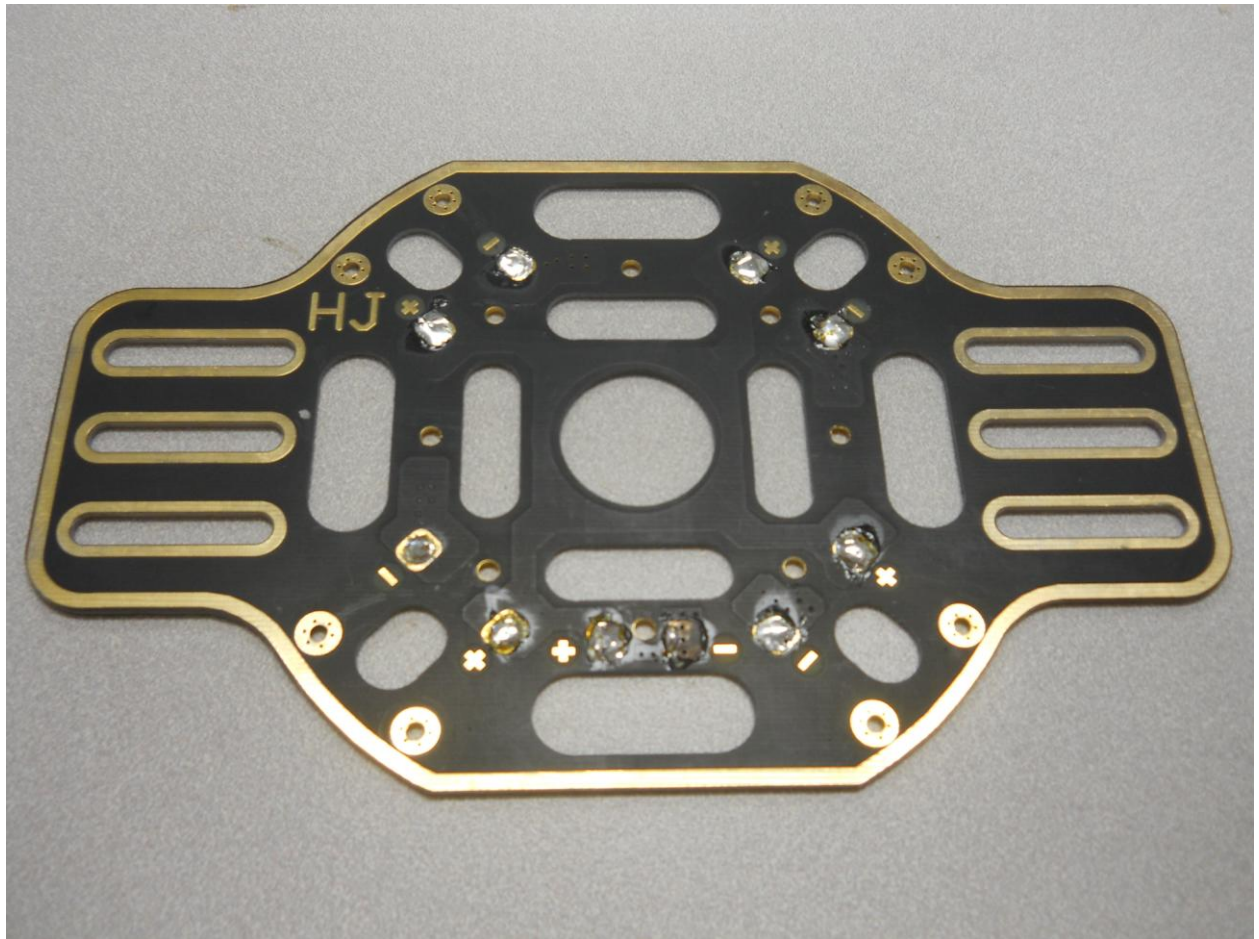
```
File Edit Options Buffers Tools Python Help
def calcPaddlePosition(x, y, dx, dy):
    # This calculates the position of the paddle for a puck movement
    if dy != 0:
        newX = x - y/dy * dx
    else:
        newX = x
    newY = 0
    if newX < 0:
        newX = -newX
    if newX > 180:
        newX = 180 - (newX - 180)
    return newX, newY
```

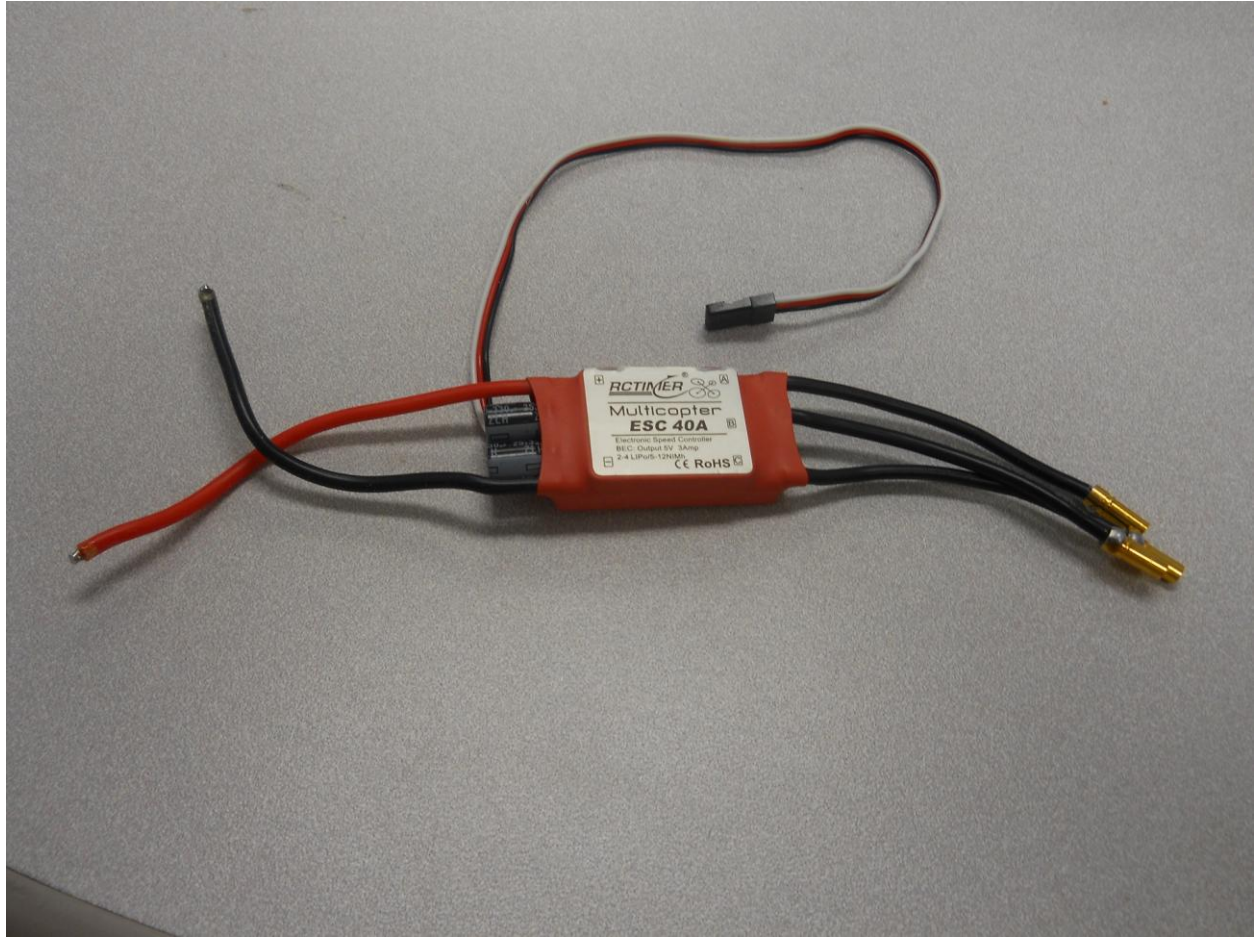
The Emacs status bar at the bottom shows: `-UU-:----F1 puckLib.py All L1 (Python)-----`. Below the status bar, it says: "For information about GNU Emacs and the GNU system, type C-h C-a."

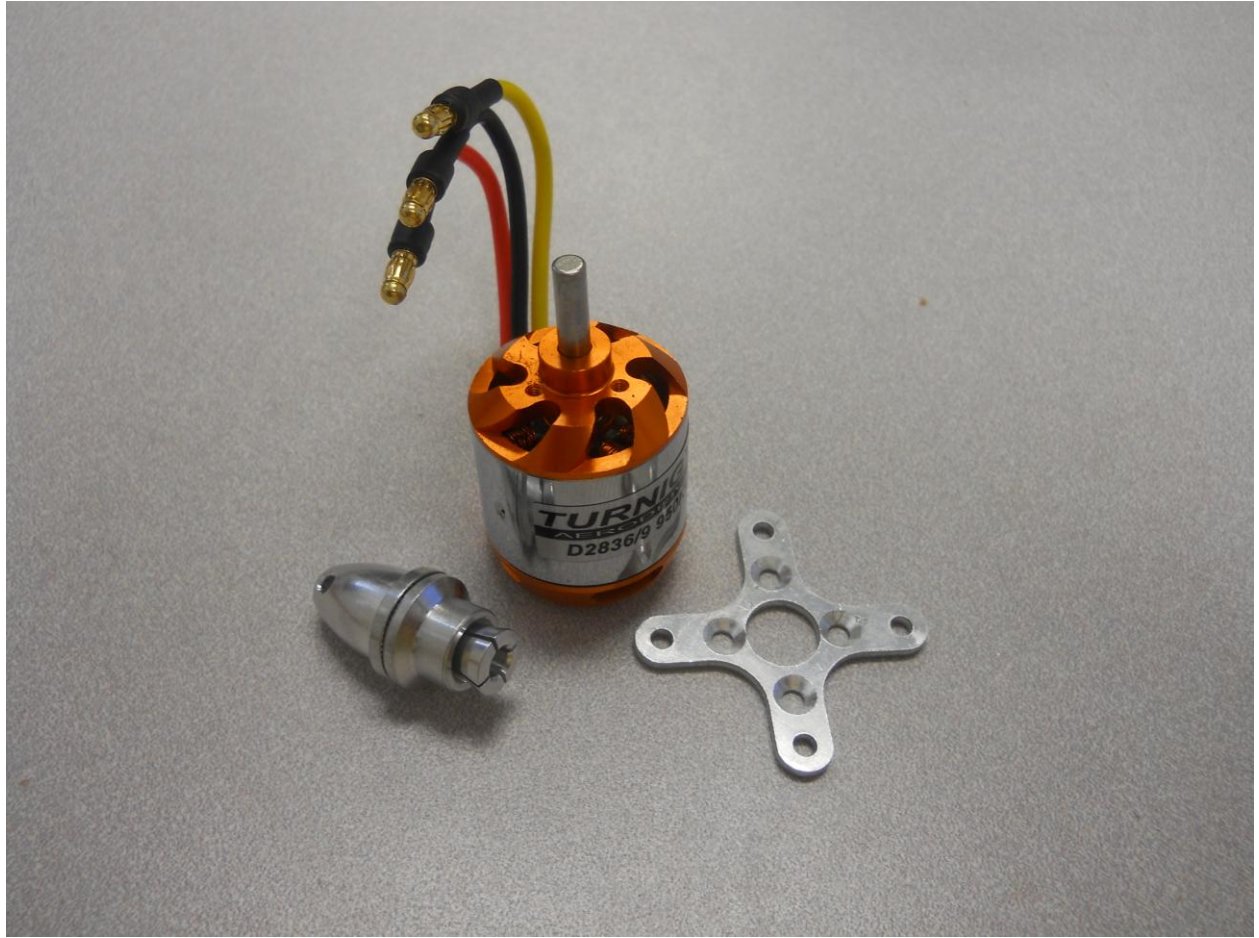
```
pi@raspberrypi: ~/examples/python
File Edit Options Buffers Tools Python Help
import cv2
import numpy as np
from puckLib import *
import time
import serial
ser = serial.Serial('/dev/ttyACM0', 115200, timeout = 1)
cap = cv2.VideoCapture(0)
cap.set(3, 180)
cap.set(4, 120)
low_range = np.array([10, 120, 100])
high_range = np.array([70, 255, 255])
lastX = 0
lastY = 0
deltaX = 0
deltaY = 0
while (cap.isOpened()):
    ret, frame = cap.read()
    hue_image = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    threshold_img = cv2.inRange(hue_image, low_range, high_range)
    contour, hierarchy = cv2.findContours(threshold_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    if contour:
        center = contour[0]
        moment = cv2.moments(center)
        (x,y),radius = cv2.minEnclosingCircle(center)
        y = 120 - y
        center = (int(x), int(y))
        deltaX = int(x) - lastX
        deltaY = int(y) - lastY
        lastX = int(x)
        lastY = int(y)
        radius = int(radius)
        img = cv2.circle(frame, (lastX, 120 - lastY), radius, (0, 255, 0), 2)
    cv2.imshow('video', frame)
    if deltaY < 1:
        newPaddleX, newPaddleY = calcPaddlePosition(lastX, lastY, deltaX, deltaY)
        xCommand = str(newPaddleX)
        yCommand = str(newPaddleY)
        command = xCommand.zfill(3) + ',' + yCommand.zfill(3)
        ser.write(command)
        time.sleep(.2)
    if cv2.waitKey(10) == 27:
        break
cap.release
cv2.destroyAllWindows()
-UU-:***-F1 trackPuck.py All L9 (Python)-----
```

Chapter 7: A Robot That Can Fly





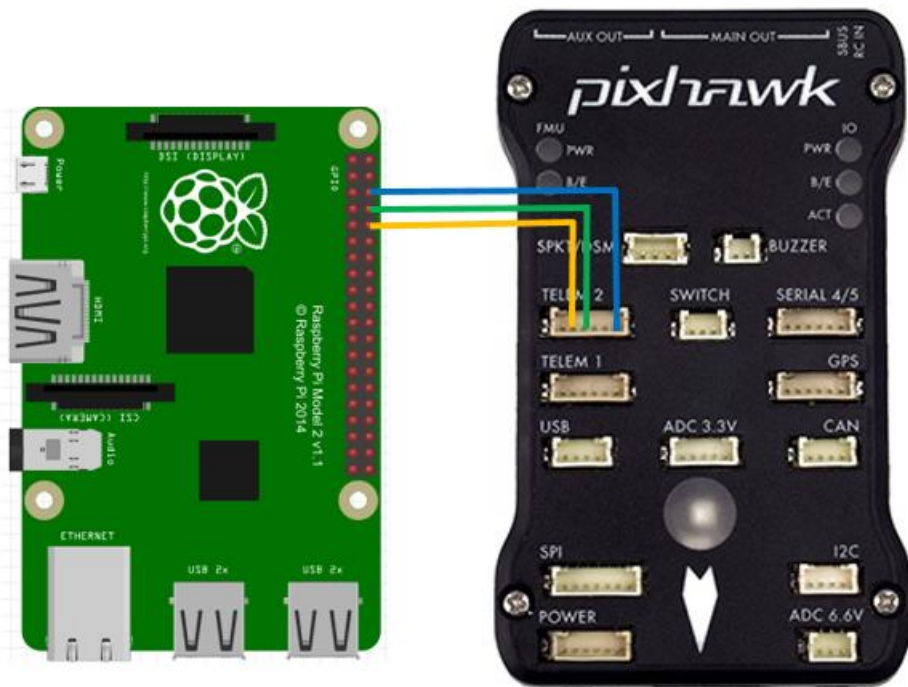




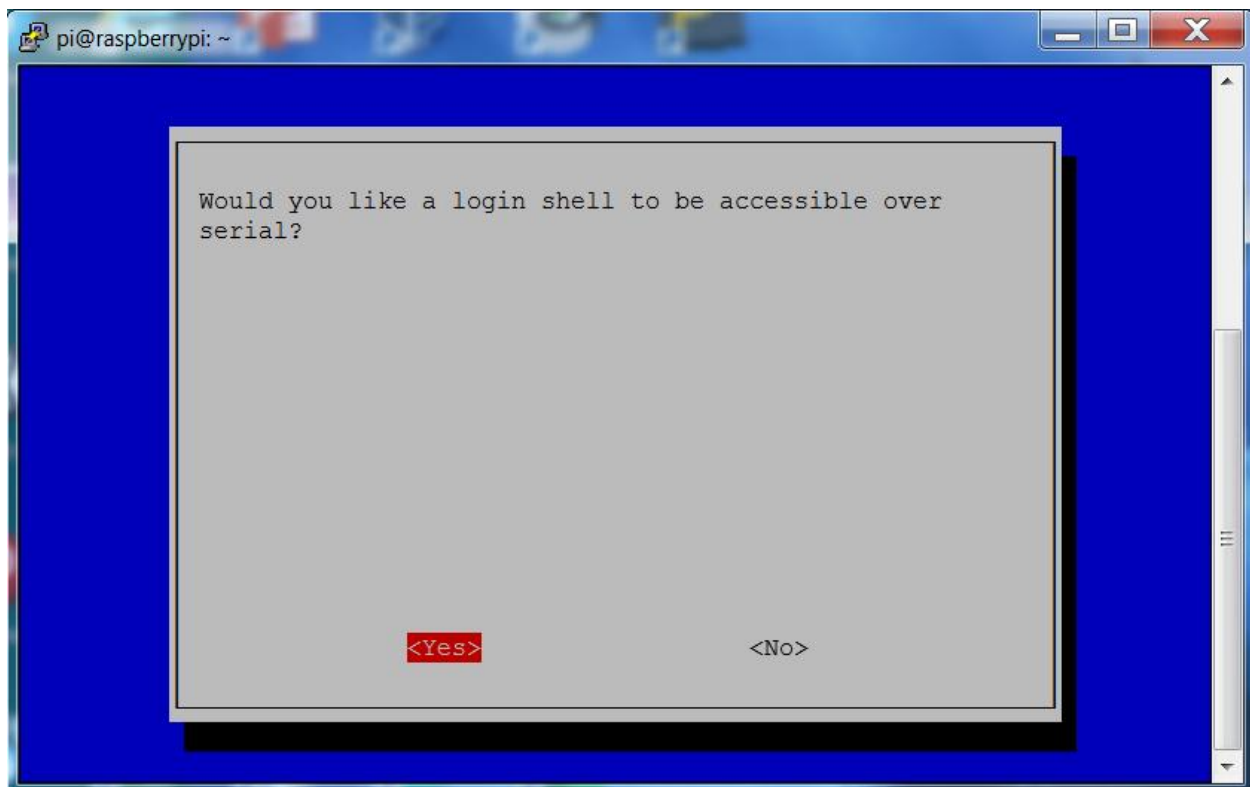
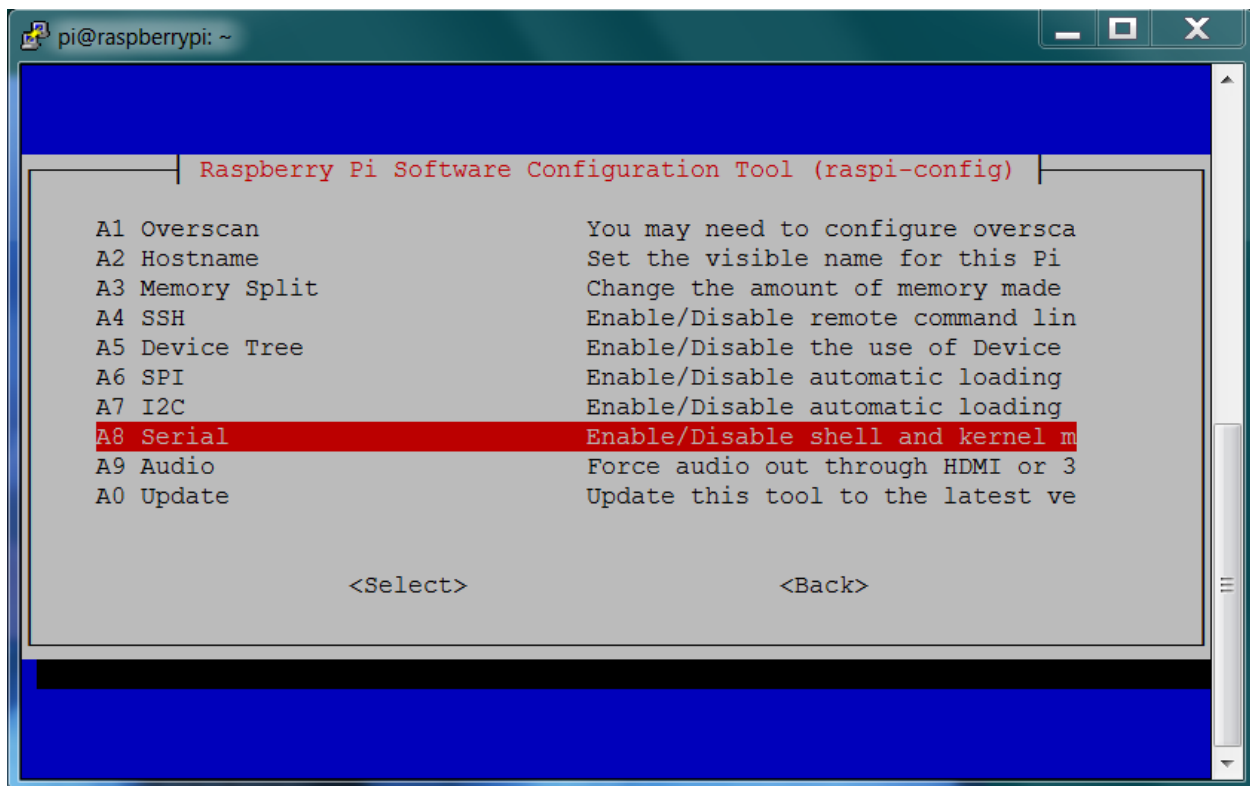








```
pi@raspberrypi: ~  
  
Raspberry Pi Software Configuration Tool (raspi-config)  
  
1 Expand Filesystem           Ensures that all of the SD card s  
2 Change User Password        Change password for the default u  
3 Enable Boot to Desktop/Scratch Choose whether to boot into a des  
4 Internationalisation Options Set up language and regional sett  
5 Enable Camera                Enable this Pi to work with the R  
6 Add to Rastrack              Add this Pi to the online Raspber  
7 Overclock                    Configure overclocking for your P  
8 Advanced Options            Configure advanced settings  
9 About raspi-config           Information about this configurat  
  
<Select>                       <Finish>
```



```
pi@raspberrypi: ~
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Sep 14 02:28:23 2015 from 116.98.25.36
pi@raspberrypi ~ $ sudo -s
root@raspberrypi:/home/pi# mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --
aircraft MyCopter
Connect /dev/ttyAMA0 source_system=255
no script MyCopter/mavinit.scr
Log Directory: MyCopter/logs/2015-09-15/flight1
Telemetry log: MyCopter/logs/2015-09-15/flight1/flight.tlog
Waiting for heartbeat from /dev/ttyAMA0
MAV> 0 0 QAonline system 1
STABILIZE> Mode STABILIZE
fence breach
APM: ArduCopter V3.2.1 (36b405fb)
APM: PX4: ce602658 NuttX: 475b8c15
APM: Frame: QUAD
APM: PX4v2 00380029 31334706 38383835
Received 417 parameters
Saved 417 parameters to MyCopter/logs/2015-09-15/flight1/mav.parm

STABILIZE> █
```

```
pi@raspberrypi: ~
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Sep 14 02:28:23 2015 from 116.98.25.36
pi@raspberrypi ~ $ sudo -s
root@raspberrypi:/home/pi# mavproxy.py --master=/dev/ttyAMA0 --baudrate 57600 --
aircraft MyCopter
Connect /dev/ttyAMA0 source_system=255
no script MyCopter/mavinit.scr
Log Directory: MyCopter/logs/2015-09-15/flight1
Telemetry log: MyCopter/logs/2015-09-15/flight1/flight.tlog
Waiting for heartbeat from /dev/ttyAMA0
MAV> 0 0 QAonline system 1
STABILIZE> Mode STABILIZE
fence breach
APM: ArduCopter V3.2.1 (36b405fb)
APM: PX4: ce602658 NuttX: 475b8c15
APM: Frame: QUAD
APM: PX4v2 00380029 31334706 38383835
Received 417 parameters
Saved 417 parameters to MyCopter/logs/2015-09-15/flight1/mav.parm

STABILIZE> param show ARMING_CHECK
STABILIZE> ARMING_CHECK      1.000000
█
```

```
pi@raspberrypi: ~/dronekit-python/examples/vehicle_state
STABILIZE> module load droneapi.module.api
STABILIZE> DroneAPI loaded
Loaded module droneapi.module.api

STABILIZE> api start vehicle_state.py
STABILIZE>
Get all vehicle attribute values:
  Location: Location:lat=0.0,lon=0.0,alt=1.38999998569,is_relative=False
  Attitude: Attitude:pitch=0.0657835155725,yaw=-3.04151630402,roll=-0.02454243041
57
Velocity: [0.0, 0.0, 0.0]
GPS: GPSInfo:fix=0,num_sat=0
Groundspeed: 0.0
Airspeed: 0.0
Mount status: [None, None, None]
Battery: Battery:voltage=0.0,current=None,level=None
Rangefinder: Rangefinder: distance=None, voltage=None
Rangefinder distance: None
Rangefinder voltage: None
Mode: STABILIZE
Armed: False
Set Vehicle.mode=GUIDED (currently: STABILIZE)
Waiting for mode change ...
Got MAVLink msg: COMMAND_ACK {command : 11, result : 0}
APM: PreArm: Need 3D Fix
GUIDED> Mode GUIDED
Set Vehicle.armed=True (currently: False)
Waiting for arming...
Got MAVLink msg: COMMAND_ACK {command : 400, result : 3}
Waiting for arming...
```

