Cherry Pi

Scientific Instrumentations Lab

Professor Ran

# Rotational Inertia Generator User Manual

Story:

**Design Background:**

Rotational Inertia Generator (R.I.G.) is the latest design in gauging centripetal acceleration, centripetal force, and radial orbit distance across a variety of applied masses. While many other centripetal force generators rely on a calculated moment of inertia rod, R.I.G. is able to simulate near-point mass rotation through the application of a bead chain. An overall design brief of the system is as follows:

A 6-18 V DC motor is mounted to the head of the overall machine frame. Attached to the motor is a collar with an attached bead chain (See Figure 1). The mechanism connecting the bead chain to the collar is a chain holder. Attached to the other end of this bead chain is a hook with an appropriate mass. As the motor rotates the mass holds a radial path of motion around the center of the motor. Attached to the frame approximately 22 cm away from the motor is an infrared distance sensor that measures the closest distance from the mass to the sensor. The end result, when subtracted from this distance of 22 cm, is the radius of the orbit. A piezoelectric strip is connected to the top of the frame and with each rotation of the motor is hit by an extrusion in the motor. The extrusion is created by a 50 mm screw put in the top hole of the motor collar. As the motor moves the screw makes contact with the piezoelectric strip resulting in an input signal. This provides the Raspberry pi a signal for each rotation inputting a period.

The radius *r* and period *T* values may be used to derive the centripetal acceleration and, with the mass *m*, centripetal force. The expressions

$a_c = 4\pi^2 r/T^2$ and F = ma

describe this relationship.

Customers that desire a state of the art machine that accurately describes the radial distance of motion of a mass, analyses the centripetal acceleration of motion, and does not rely on a rod moment of inertia term should find this device extremely appealing.

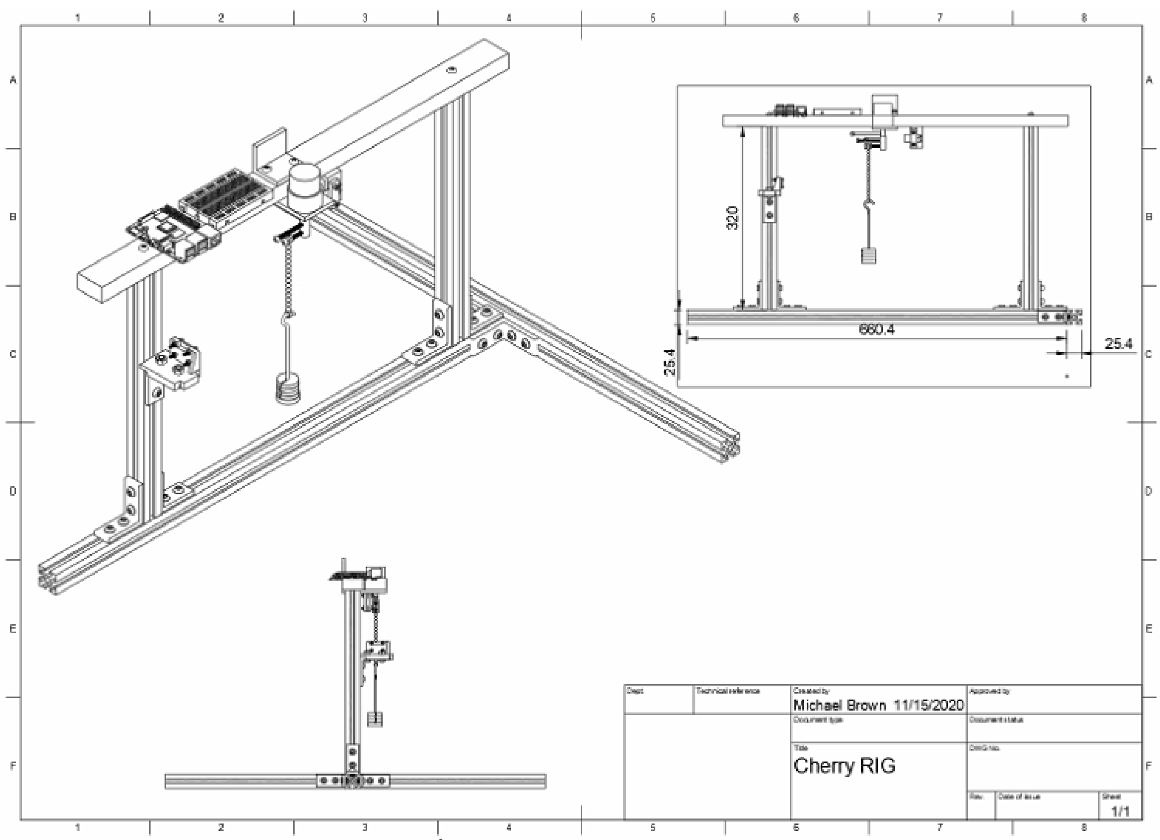



*Figure 1: Diagram of the Centripetal Force Detector Frame*

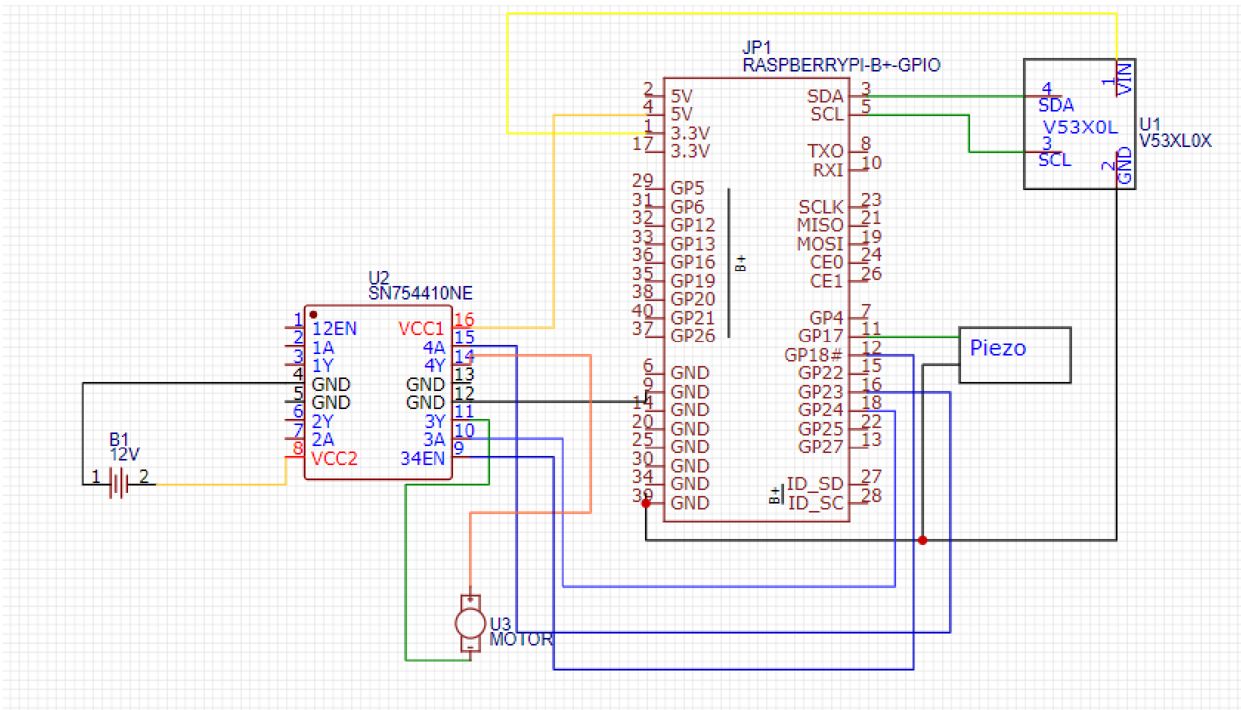



*Figure 2: Circuit Schematic for R.I.G.*

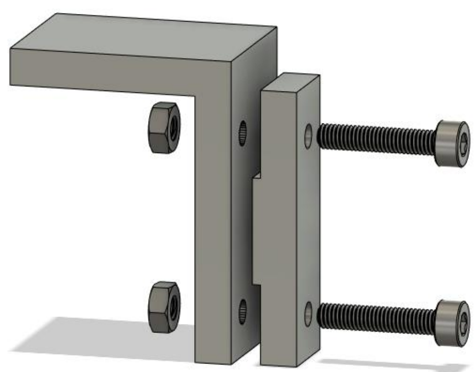


*Figure 3: Piezoelectric mounting Strip.*

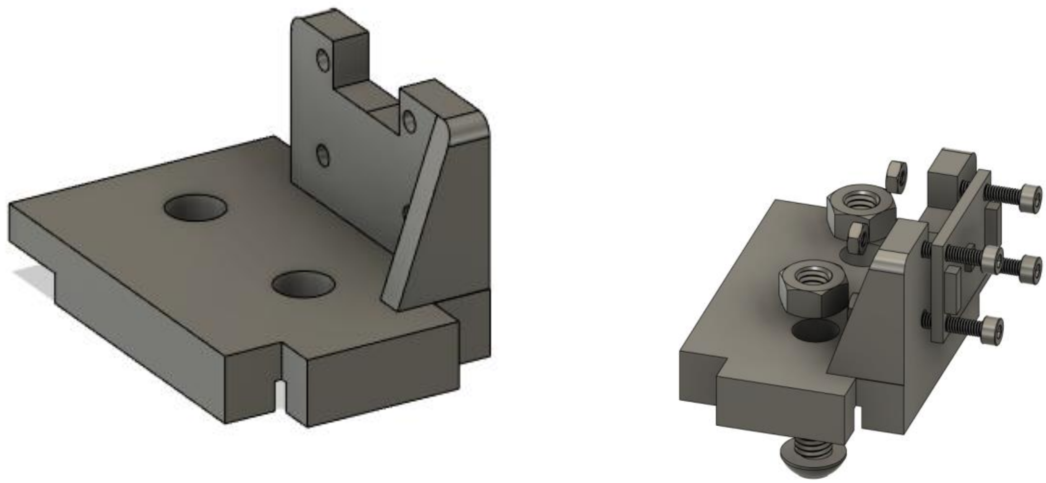


*Figure 4: Distance Sensor Mounting Piece.*

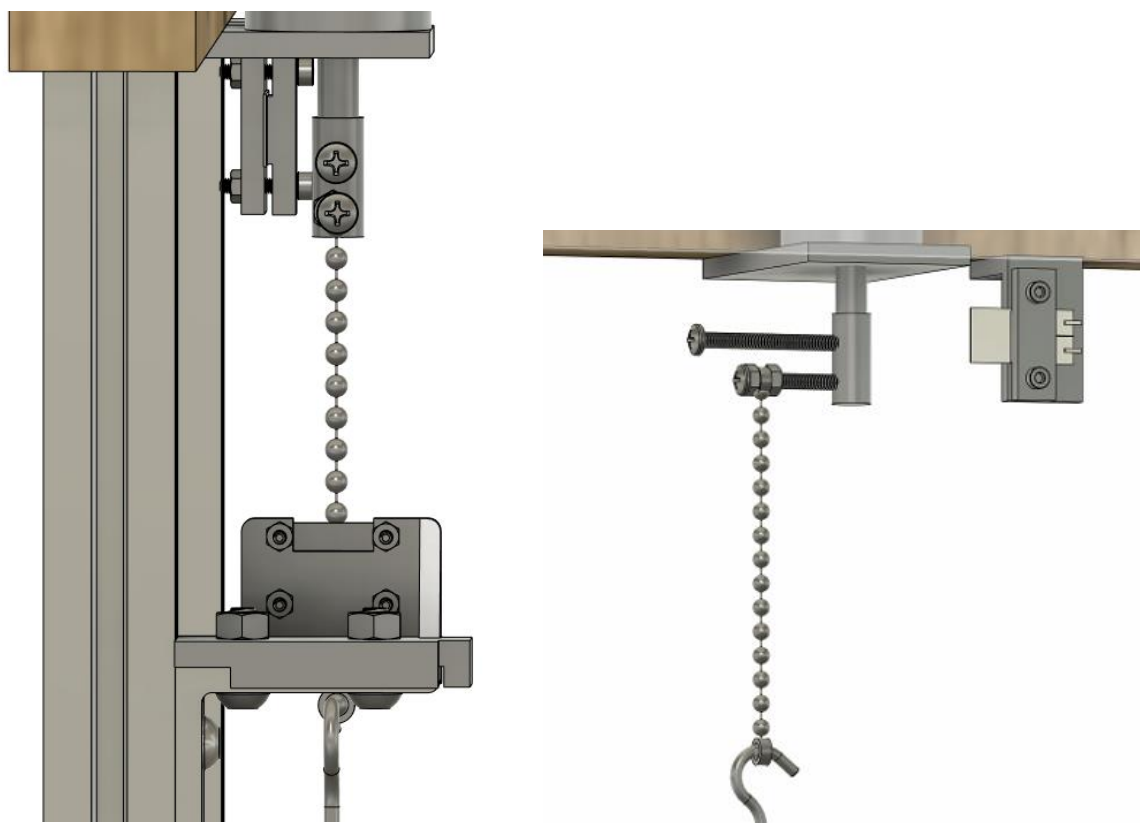


*Figure 5: Bead Chain Mount.*

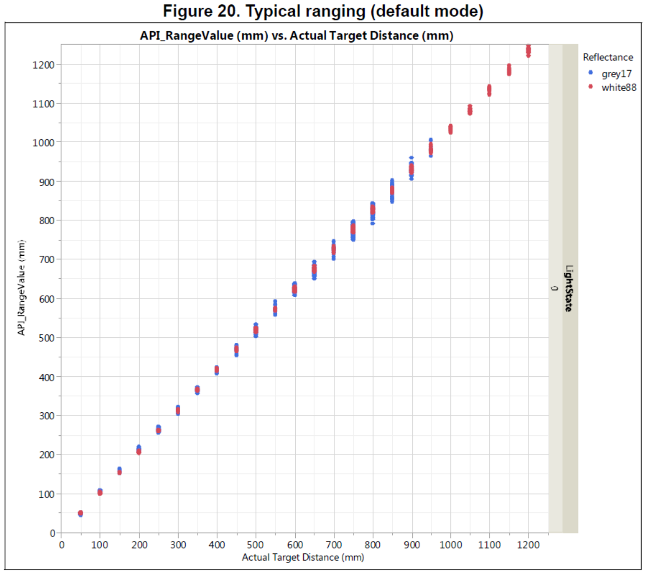
(OTHER MEDIA)

Specifications:

The DC Motor operates within a range of 1.4 - 18 Volts. Its rotation speed at 12V is 81 RPM, where RPM is proportional to the voltage it receives. It has a 36 mm face with a bottom centered rotating shaft. The motor is a brushless gear motor and as such is generally resistant to overheating.

The Adafruit VL53L0X requires 2.8 Volts to run, allowing it to easily run from the 3.3 - 5.0 Volt output sourced from our Raspberry Pi. When running, the VL53L0X has an operable range of 50 millimeters to 1200 millimeters and communicates through an I2C. When powering and communicating with the VL53L0X, the sensor takes both clock and data measurements represented on the component by the SCL and the SDA pins, respectively. The precision of the VL53L0X is subject to both distance and the reflectance of the surface it is measuring. To minimize this error, the mass contained in the system has been covered in masking tape to create a matte surface and the error margins for distance have been accounted for in the R.I.G.'s operational code. The error of the sensor is as follows:

Figure 20. Typical ranging (default mode)




*Source: Adafruit VL53L0X Datasheet*

*(https://www.adafruit.com/product/3317#technical-details)*

A key component of the circuit is the piezoelectric strip which is able to accurately gauge a period by detecting contact with a motor extrusion. This is necessary for gauging the period in the equation determining the centripetal acceleration.

The other core component of the machine is the infrared laser sensor as described above. LAsers are ideal devices for measuring distance due to their linear property of cohesion, laser light travels in a singular line. The dictator gauges the distance of the mass from the sensor and code on the raspberry pi computer subtracts the distance from the measure total range (22 cm) to attain a radius.

A Raspberry pi is a small circuit board computer able to carry out a broad range of functions. Here it is used to drive the motor and sensors as well as compile the data into acceleration and force values.

Piezoelectric material uses a crystalline lattice to transmit an electrical signal when stress is received. This signal is in turn proportional to the stress on the piezoelectric material and proportional to the force applied. As such a rotating bar hitting the piezo at a regular interval will send a signal of the period of rotation of the motor.

The infrared laser sensor works by shooting a low intensity infrared laser ~ 700 nm to 1 mm. With a known speed of light ~ $3*10^8$ m/s the sensor is able to gauge the lights time of flight to the mass and reflection back and through the expression $v = x/t$ and $2d = x = v*t = c*t$ arrives at the distance between the mass and the sensor.

The central DC motor operates off of induction as wire wrapped around the central shaft induces a magnetic torque that drives the rotation of the central beam. Connecting this to a chain allows a rotation to precipitate over the mass attached.

<u>Tutorial:</u>

**Instructions**

Before operating with the device please cover the mass, if reflective, with tape or some sort of sleeve. Attach the mass to the hook and secure the hook to the bead chain. Measure the mass with the bead chain and record its mass in kilograms.

In order to effectively set up the device insert 9 V batteries into the battery pack and plug in the HDMI of the Raspberry pi computer into a respective monitor and plug in the mouse and keyboard to the Raspberry pi USB ports. Click on the directory “RIG” then the file “input.py” on the desktop of the device.

The dialogue prompt should read the following: “Insert mass in grams,” please insert a value for the mass in grams. For 10 g mass insert “10”. After this the prompt “What direction of spin?(type plus or minus),” will appear, type “plus” into the command prompt for clockwise and “minus” into the prompt for counterclockwise. Next, the prompt will read “Desired duty cycle:“ please set the duty cycle to a value from 0 to 100%. For example, insert “100” for a 100% duty cycle, insert “50” for a 50% duty cycle, and so on. After this the prompt will read “how long before taking measurements:,” please insert a delay time in seconds: 15 for a 15 second delay (reccomended), 20 for a 20 second delay ect.. Last the monitor will display “desired number of data points:” please insert the desired number of data points. We suggest an average value of 100 data points but this should determine the overall precision of your measurements.

Expect two CSV files to be saved under the name “dataperiod.csv,” and “datadistance.csv.” After the code runs a graph of the distance over time for the mass will appear on your screen. In the console window there should be measurements of the average radius, period, centripetal acceleration, and centripetal force. If desired, insert a USB into the machine to attain any data files. Save any files you want to the USB and close the opened programs.

**Warnings:**

- Do not leave the 9 V power supply connected to the device when not in use.
- Keep approximately one foot away from the center of the device at all times.
- Keep conductive materials away from the circuit board at all times.
- Do not stare at the laser sensor as the device is in operation.
- Do not disconnect any of the wires, if wires become dislodged please get assistance and carefully follow the circuit diagram to repair the machine.

- Make sure mass is securely fixed to the bead chain to avoid unwanted slipping.

**Frequently Asked Questions:**

1. Why is the motor not stopping?

    Check the wires are going into the motor and make sure that there is no external power. At worst something has shorted, use a Digital Multimeter to gauge the voltage at each wire. Connect the black end to a ground and the red end to a place on the machine that should be measured.

2. Why is the distance value coming out so large (or not at all)?

    Make sure that the sleeve or tape is covering all reflective parts of the mass. It is possible the laser is being reflected instead of picking up the mass's distance value.

3. Why is the motor spinning slower than normal?

    Check that the 9 volt battery supply is not drained. If it is, chances are the motor is not spinning at maximum capacity and changing the batteries is a simple fix. If the motor is still spinning slowly make sure the duty cycle is not set too low.

4. If the bead chain, or another component, breaks is there anything that can be done to fix it?

    Yes, the bead chain is strapped to a bead chain hook. Simply dismounting the chain and adding a new one will repair the device. Chains may be removed or added like a standard curtain bead chain drawstring. The blueprint and components sheets in this document should have locations, components, and replacement part sites where these items may be purchased.

5. The motion of the mass is not circular -- is the device broken?

The device is not broken and probably simply needs more time to adjust to the circular path. Adjusting the delay time as seen in the instructions will most likely correct this issue. Additionally, moving the mass further out on the screw will mean that the chain is set apart from the motor spindle. This will further allow a circular motion.

Appendix:

**Roles:**

**Simon:** Electrical / Media = Design Manager — Contact: sjdavenport@email

**Jameson:** Coding / Electrical = Software Designer — Contact: gjcrouse@email.wm.edu

**Jordan:** CAD Engineer/ Materials = CAD Iterator — Contact: jekoseski@email.wm.edu,

**Brennen:** Electrical = Construction Engineer

**Meley:** CAD Engineer = CAD Manager — Contact: mbbrown01@email.wm.edu

**Douglas:** Leader/ Coding = Project Manager — Contact: djpalumbo@email.wm.edu

Components:

Total Cost: $75.23. *Raspberry Pi not included

| Component | Price | Amount | Purchase Location |
|---|---|---|---|
| Brushless Gear DC Motor | $16.21 | 1 | https://usa.banggood.com/DC-12V-300RPM-Geared-Motor-High-Torque-37GB-550-Gear-Reducer-Motor-p-1068575.html?utmsource=googleshopping&utm_medium=cpc_organic&gmcCountry=US&utm_content=minha&utm_campaign=minha-usg-pc&currency=USD&createTmp=1&cur_warehouse=CN |
| DC Motor Shaft Collar | , Price = $4.99, | 1 | https://www.sparkfun.com/products/12325 |

| | | | |
|---|---|---|---|
| Piezo film tab | $5.37 | 1 | https://www.digikey.com/en/products/detail/te-connectivity-measurement-specialties/1002794/279646 |
| 50 mm, ¼ Diameter, Screw (Check) | $2.00 | 1 | https://www.homedepot.com/p/M4-0-7-x-50-mm-Phillips-Pan-Head-Stainless-Steel-Machine-Screw-2-Pack-842948/204283773 |
| Infrared Laser Sensor | $14.95 | 1 | https://www.adafruit.com/product/3317 |
| Bead Chain | $3.64 | 1 | https://www.homedepot.com/p/Commercial-Electric-3-ft-Chrome-Beaded-Chain-with-Connector- |

| | | | |
|---|---|---|---|
| | | | 82405/306596879?g_store=4635&source=shoppingads&locale=en-US |
| Aluminum Rail | $3.29 | Per 250 mm | https://openbuildspartstore.com/v-slot-20x20-linear-rail/ |
| Tee Nuts M5 | $2.99 | Pack of 10 | https://openbuildspartstore.com/tee-nuts-m5-10-pack/ |
| M5 Screws | $0.89 | Pack of 10 | https://openbuildspartstore.com/low-profile-screws-m5-10-pack/ |
| L Bracket | 2 X ($2.45) | 2 Packs of 4 | https://www.homedepot.com/p/Everbilt-1-in-Steel-Zinc-Plated-Corner-Brace-4-Pack-13619/203170051 |
| Wood Top Row | $5.22 | 1 | https://www.homedepot.com/p/2-in-x-4-in- |

| | | | |
|---|---|---|---|
| | | | x-96-in-Prime-Whitewood-Stud-058449/312528776 |
| Wood Screw | $2.50 | Pack of 30 | https://www.homedepot.com/p/SPAX-8-x-1-in-Philips-Square-Drive-Flat-Head-Full-Thread-Zinc-Coated-Multi-Material-Screw-30-per-Box-4101010400251/202040993 |
| Screw Mount Links | $8.28 | 1 (pack of 10) | https://www.mcmaster.com/3606T331/ |

Source Code:

` Below is the source code for R.I.G. broke into headed sections. In the image below the upper four code blocks import modules while the laser code block sets up the infrared laser distance sensor. The i2c command sets up the sensor and the laser is set up with the adefruit command. Last, a definition of the distance is set which is multiplied by a scale of 8.5/11 accounting for the slope error of the distance sensor.

```
#modules used in the code
import adafruit_vl53l0x #adafruit environment to control the laser sensor

#standard RPi code blocks
import RPi.GPIO as GPIO
import board
import busio
import time
import csv

#number manipulation
import numpy as np
import scipy as sp
import statistics as stats
import matplotlib.pyplot as plt

#sets up the board
GPIO.setmode(GPIO.BCM)

#laser code block
i2c = busio.I2C(board.SCL, board.SDA) #sets up
laser = adafruit_vl53l0x.VL53L0X(i2c)
def distance(self):
    return 8.5/11*self.range
```

The piezoelectric code is set to pin 11 with a GPIO setup. This is followed by a motor script that sets the PWM port of the motor pin at pin 18 as pin 23 and 24 were used to supply power in the forwards and reverse motion respectively.

```
#Piezo code
piezo = 11
GPIO.setup(piezo,GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

#Motor Code
Motor1E = 18 #  Enable pin 1 of the controller IC
Motor1A = 23 #  Input 1 of the controller IC
Motor1B = 24 #  Input 2 of the controller IC
GPIO.setup(Motor1A,GPIO.OUT)
GPIO.setup(Motor1B,GPIO.OUT)
GPIO.setup(Motor1E,GPIO.OUT)

forward=GPIO.PWM(Motor1A,100) # configuring Enable pin for PWM
reverse=GPIO.PWM(Motor1B,100) # configuring Enable pin for PWM

forward.start(0)
reverse.start(0)
```

Next are the data analysis scripts where a distance value is calculated from subtracting total distance by the measured distance yielding a radius. This distance value is appended to the list dist_small. Acceleration, a_c, is calculated with the expressions a = 4*(pi^2)*r/(T^2) where r is divided by 1,000 to convert centimeters to meters. Centripetal force, F_c, multiplies this value by mass.

```
#Data analysis code blocks
def avg_dist(dist):
    '''
    Takes a list of distance measurements and outputs a radius.
    '''
    dist_small = []
    r_big  = stats.mean(dist)
    print('measured mean is: ', r_big)
    low_high= dist.copy()
    low_high.sort()
    for i in range(3):
        dist_small.append(low_high[i])
    r_small = stats.mean(dist_small)
    return (210 - r_small)


def a_c(radius, period, mass, F_or_a= 'F'):
    '''
    Outputs the centripetal force or acceleration, use 'F' or 'a' for desired output,
    Make sure to use average distance output for dist variable.
    '''
    a_c = 4*np.pi**2 *radius/period**2/1000
    F_c = mass*a_c/1000
    if F_or_a == 'F':
        return F_c
    elif F_or_a == 'a':
        return a_c
```

The test code section runs through the previous script in order driving the motor while measuring the distance and period. Time.time is used in association with other lists to measure an overall period and radius.

```
#test code
m = float(input('What is the mass being measured? '))
sign = input('What direction of spin?(type plus or minus) ')
duty_cycle = int(input('Desired duty cycle:  '))
wait = float(input('how long before taking measurements:  '))
num_d = int(input('desired number of distance data points:  '))
num_T = int(input('desired number of period data points:  '))
t = []
dist = []
period = []
try:
        if sign == 'plus':
            GPIO.output(Motor1E,GPIO.HIGH)
            forward.ChangeDutyCycle(duty_cycle)
            reverse.ChangeDutyCycle(0)
        elif sign == 'minus':
            GPIO.output(Motor1E,GPIO.HIGH)
            forward.ChangeDutyCycle(0)
            reverse.ChangeDutyCycle(duty_cycle)
    #tickers
        i=0
        j=0
    #timing
        time.sleep(wait)
        ti = time.time()
    #distance while loop
        while i< num_d:
                l = distance(laser)
                if l < 240:
```

```
        ti = time.time()
  #distance while loop
    while i< num_d:
            l = distance(laser)
            if l < 240:
                dist.append(l)
                t.append(time.time()-ti)
                i+=1
    told = time.time()

    while j< num_T + 1:
            if GPIO.input(11)==1:
                tnew = time.time()
                period.append(tnew-told)
                told = time.time()
                j+=1
                time.sleep(.2)

    period.pop(0)
    GPIO.cleanup()
    r = avg_dist(dist)
    T = stats.mean(period)
    print('Period is', T, ' seconds')
    print('radius is', r, ' mm')
    print("Centripetal force is:", a_c(r, T, m, 'F'), ' N')
    print('Centripetal acceleration is:', a_c(r, T, m, 'a'), ' m/s^2')
  #outputs data csv files. If taking multiple data points make sure to create new csv files so they are not overwritten
    with open('distance_data.csv', mode = 'w') as f:
```

Last two csv lists are output with the distance and period values. A keyboard interrupt is set to stop the system early.

```
        period.pop(0)
        GPIO.cleanup()
        r = avg_dist(dist)
        T = stats.mean(period)
        print('Period is', T, ' seconds')
        print('radius is', r, ' mm')
        print("Centripetal force is:", a_c(r, T, m, 'F'), ' N')
        print('Centripetal acceleration is:', a_c(r, T, m, 'a'), ' m/s^2')
    #outputs data csv files. If taking multiple data points make sure to create new csv files so they are not overwritten
        with open('distance_data.csv', mode = 'w') as f:
            writer =csv.writer(f)
            writer.writerow(t)
            writer.writerow(dist)
        with open('period_data.csv', mode = 'w') as f:
            writer =csv.writer(f)
            writer.writerow(period)
        plt.figure()
        plt.plot(t, dist, '.')
        plt.show()
except KeyboardInterrupt:
    GPIO.cleanup()
```