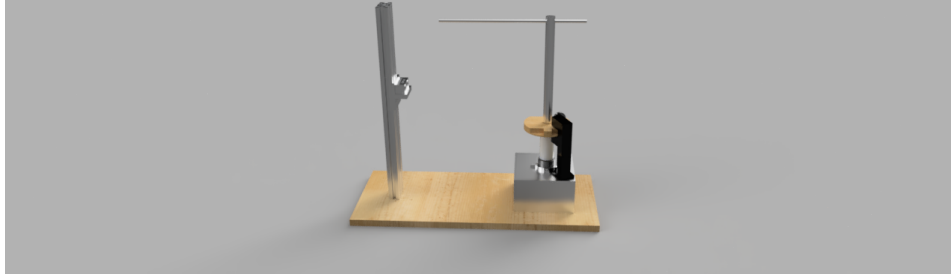


**User Manual for CMONEY**  
*Centripetal Machine Obtaining Neat Evidence for You*

Blueberry Pi

PHYS 351



## 1 Introduction

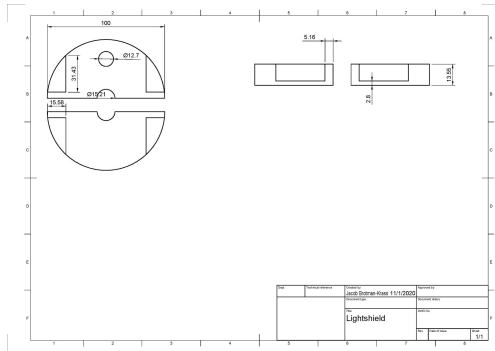
The 102 Physics Lab requires students to calculate centripetal force using a complicated and rather antiquated machine. Students must spin a wheel to go as steady as possible, while simultaneously recording the number of rotations using a stopwatch. This is inefficient and riddled with error.

Our goal with the C-MONEY (Centripetal Machine Obtaining Neat Evidence for You) is to make this process computerized/automated/motorized using the Raspberry Pi 4 for efficient and accurate data collection.

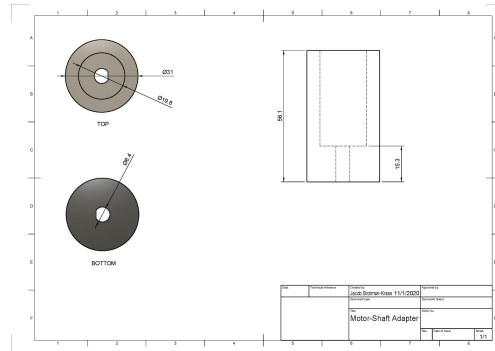
In order to calculate centripetal force, there are three basic variables one must determine: mass, radius, and frequency:  $a_C = 4\pi^2 r f^2$ , where  $a_C$  is the centripetal acceleration,  $r$  is the radius, and  $f$  is the frequency of rotation. Subsequently, the centripetal force is calculated with the formula  $F_C = ma_C$ .

While the mass can be measured very easily on an electronic scale by the user, the other two variables are less obvious. With CMONEY, the machine does the work for you. A ultrasonic range-finder sensor is used to measure the radius with a resolution of about 0.3 cm (very precise), while a laser-photodiode system is used to measure rotation frequency. CMONEY's comprehensive code only requires the user to input mass and choose a spinning speed, and in a short time, the user is presented with useful and clear data.

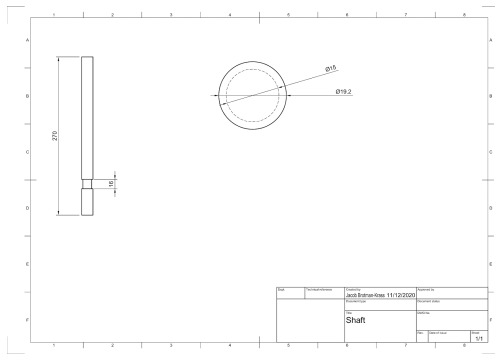
This product makes data collection for centripetal force far more user friendly. It eliminates the need for constant realigning of the mass, unlike the original process. Moreover, the spinning is automated, so motion is smooth and periodic, while the frequency detection system is extremely precise - far more so than 'eyeballing.' With easy to acquire data provided by CMONEY, an introductory physics student will understand the effects of mass, radius, and velocity in no time. In the remainder of this section, the parts, circuit schematics, pictures, and tutorial videos are provided for details on how the device functions.



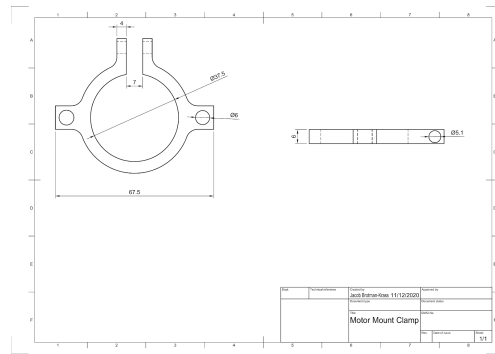
(a) Light-shield



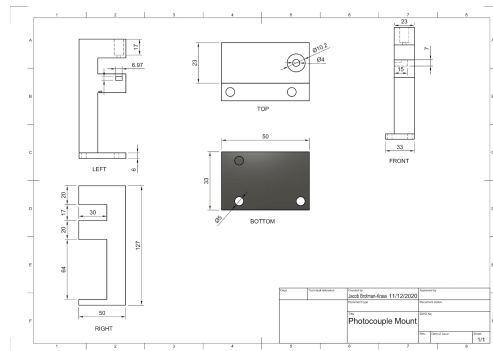
(b) Motor-Shaft Adapter



(c) Shaft



(d) Motor-Mounting Clamp



(e) Photocouple Moun

Figure 1: CAD Schematics

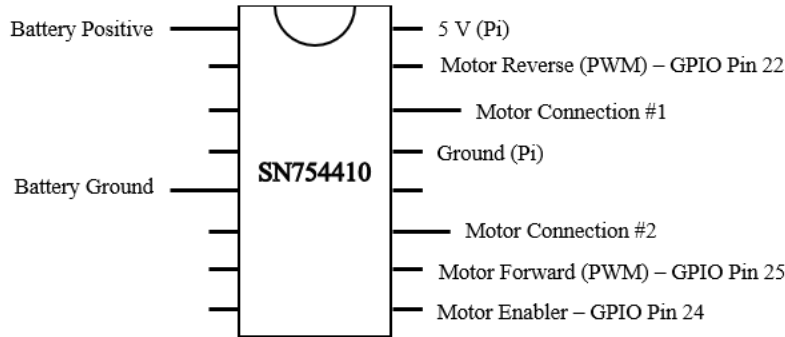


Figure 2: Configuration of the SN754410 Motor Driver Chip as used in the CMONEY device.

## 2 Circuitry

The core mechanisms of this device relies on three components. First, a DC motor and driver circuit to ensure full mobility and speed of the electronic circuit. Second, a coupled laser and photodiode system to account for the total rotational frequency. Lastly, the range finder circuit uses an ultrasonic sensor to determine overall distance between the mass and the centripetal central axis of the spinning motor system.

### 2.1 DC Motor and Driver

This machine uses the SN754410 Motor Driver Chip to spin a brushless DC motor. Connections are based on standard pulse width modulation commands and an external voltage of 24 V (although the motor will run plenty fine with voltages as low as 9 V) and paired ground attached to the left hand side of the chip. This power should come from a DC power source of some form. In its current iteration, a red positive and black ground cord are attached for power delivery. Future iterations of the design will include a 24 V wall power adapter and transformer, that turns AC wall-voltage into 24 DC volts, so the motor can be powered by directly plugging it in. For accessibility, the DC power source can be replaced with a properly configured 9 V battery.

The overall motor is connected to the Pi using 3 GPIO pins and 2 Pi pins that act as a 5V source and a ground. GPIO pin 25 allows for the motor to spin in the forwards direction, GPIO pin 22 allows for the motor to spin in the backwards direction, and GPIO 24 is an enabler that is always on so long as the code is running such that it can allow the other GPIO pins pulse width modulation to function properly. Pulse width modulation commands control the total rotation speeds for the device, and internally three speeds are set - “SLOW” at 33.3% of max speed, “MEDIUM” at 66.7%, and “FAST” at 100%.

### 2.2 Laser Diode/Photodiode Coupled System

This is the basic circuit that allows for the laser pointer to function without burning out any connected devices. Whenever the laser is not in use, please put the switch in the off position.

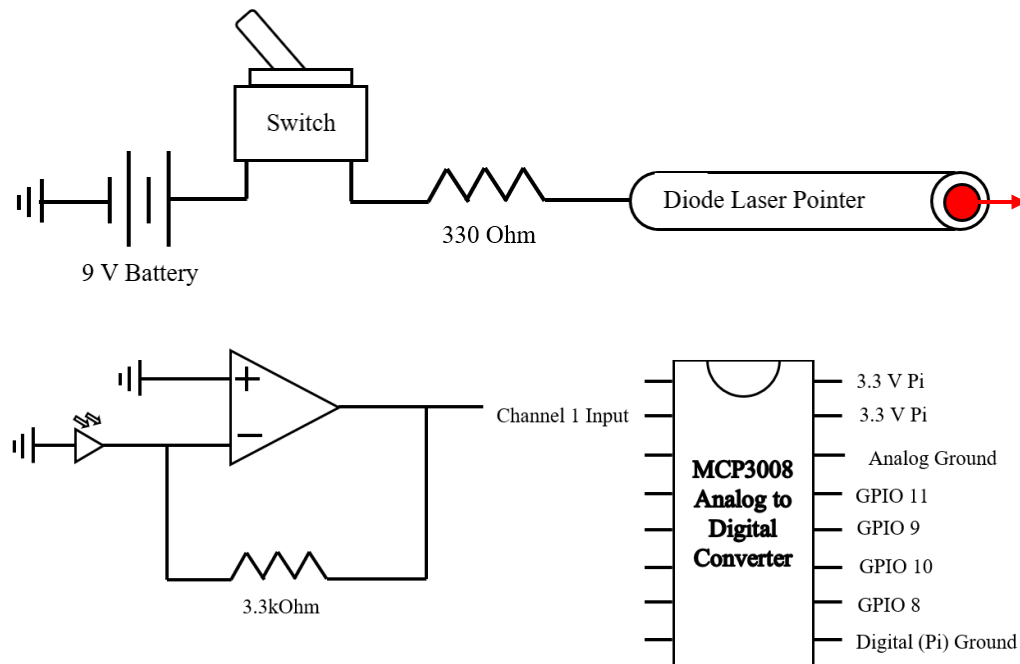


Figure 3: Configuration of the photodiode detector and analog to digital converter as used in the CMONEY device.

This photodiode detector circuit is attached beneath the laser pointer underneath a spinning wheel with a hole cut out such that it can properly measure frequency. The code accounts for an ambient light resulting in a constant voltage of around 200 mV. This is accounted for in the device's code which converts voltage output of the photodiode system into digital peaks and troughs that can be read by the Raspberry Pi. For signal amplification, the feedback resistor and a +/- 15 V directed to the op amp (OP27) can amplify the current created by the photodiode. The rotation-based inconsistencies in current (not the overall amplitude measured) are what allow the device to measure frequency through some clever code that finds the peaks in a given period. These signals are quickly processed through the MCP3008 analog to digital converter which utilizes a spi connection to communicate with the Raspberry Pi computer.

### 2.3 Ultrasonic Range Finder Circuit

The ultrasonic sensor circuit functions with relatively simple connections. The echo pin is directly connected to GPIO pin 15 from the Raspberry Pi, while the voltage pin is connected to 5 V and the ground pin is connected to the Pi's ground. The trigger's 5V pulse is stepped down through a voltage divider to more manageable levels ( 3.1 V) for the Raspberry Pi's GPIO Pin 5.

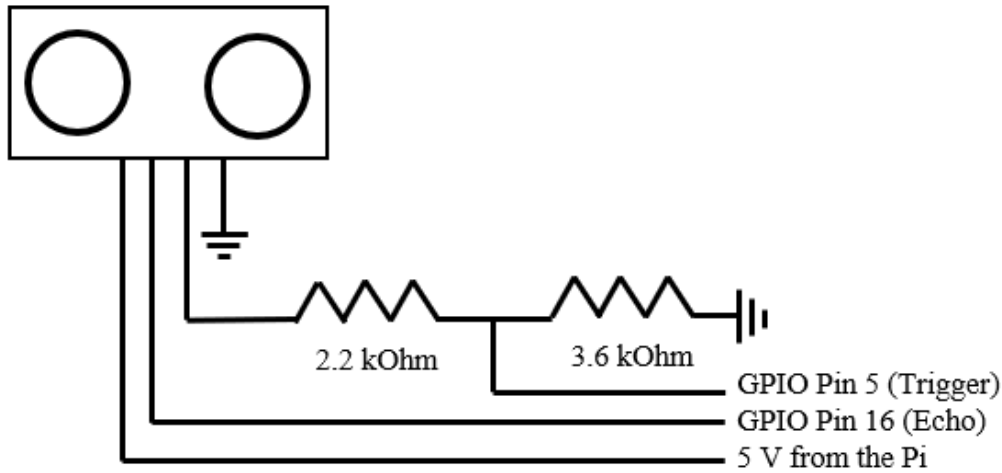


Figure 4: Configuration of the ultrasonic sensor and subsequent voltage divider as used in the CMONEY device

### 3 Design Specifications

To construct a successful product, our design team has pooled together our best ideas. The specifications of our centripetal force apparatus require a machine that can routinely find the centripetal force of a variable applied mass at a variable distance. Our design does this by measuring the frequency of rotation to determine an angular velocity  $\omega = d/dt$ , and thus velocity  $v = 2\pi\omega$ , from which a centripetal force can be calculated,  $F_c = mv^2/r$ . Thus, the centripetal force may be measured based only on an input mass and radius.

The frequency of rotations caused by a brushless DC motor requires a photodetector circuit which includes a photodiode and a battery-powered, switch-controlled laser. A disk is attached to the central rod with a hole in it for the light to pass through, making voltage spikes from the photodiode much more distinct by preventing the noise created by ambient light. Our python program does the internal math allowing for experimenters to only insert a mass into the terminal when ready. The machine uses an ultrasonic probe to determine the distance ( $r$ ) of the mass from the center spinning access.

The rod arm adapter weighs 50 g

The accuracy of the Analog Digital Converter(ADC) is based on the input resistance and ranges from 75-200 kbps in regards to sample rate.

The ADC operates over the range 2.7V - 5.5V for Vdd

The maximum extension of the rod holding the weight is 15 cm.

Our team uses a brushless C23 DC motor in our product. It directly converts direct current (DC current) into mechanical energy. The advantage of using a brushless DC motor is two-fold: minimal overheating and longevity. Both advantages are enabled by permanently lubricated bearings within the C23.

The DC motor requires a voltage supply of 9V., though we use it at an operating voltage of 12-24 V. The peak torque of the C23 motor is 2.189 N\*m.

The emf is directly proportional to the speed of the motor; this varied emf is done

precisely via rapid “on” and “off” switching from a Raspberry Pi GPIO pin. The ratio of time “on” over time “off” is known as the duty cycle. The emf our motors speed is proportional to is the magnitude of voltage outputted by the Pi multiplied by the duty cycle(0%-100%).

We use an HC SR04 distance sensor to measure the displacement of the mass along the rod. The HC SR04 works by sending eight 40 kHz ultrasound waves and measuring the amount of time for the echo to return to the sensor. This process requires four wires between the sensor and the Raspberry Pi: 5V Supply, Trigger, Echo, and Ground. The Raspberry Pi triggers the sensor to send its ultrasound waves through the Trigger connection with a  $10\mu\text{s}$  high level (on) signal. The sensor sends a high signal to the Raspberry Pi through the Echo connection lasting the duration of the time between sending and receiving a single signal. The time of the Echo pulse can be called using code on the Raspberry Pi and the test distance,  $d$ , is given by  $d = (t * v_{\text{sound}})/2$ , since the ultrasound wave must travel the distance between the sensor and the object measured (the mass in our case) twice. The accuracy of this distance measurement is 0.3 cm within the range of 2 cm and 500 cm.

An essential component of our apparatus is the photodetector circuit. The photodetector circuit provides the means by which the user will be able to gather essential information about the rotating mass. Such information, most notably the frequency of rotation, is critical to operating this machine.

Another component that is integral to this apparatus is the ultrasonic range finder. Obtaining the radius from this device is essential to computing the centripetal force as it is dependent upon knowing the radius.

## 4 Tutorial

Ready to get started? Great!

1. Make sure you have: various masses, a DC power supply, a monitor with HDMI cable and working keyboard mouse.
2. Measure the mass on a provided scale. Make sure to write down the mass in kilograms since CMONEY will need that information.
3. Place the mass securely on the arm. Rotate the rod to ensure that the arm points straight down the middle of the board pointed at the beam extrusion holding the ultrasonic sensor, so a proper device length can be accurately measured. In this step, masses may be added or removed, and the overall radius of rotation may also be manipulated, yet if the securing nut is loosened to adjust the rod ensure it is as tight as possible before engaging the motor to spin in Step 8.
4. Plug in the white power cord into the Pi’s power outlet, and attach the HDMI cord into the Raspberry Pi computer through the slit in the body of the device. Plug in the keyboard and mouse to the exposed USB cables on the side of the Raspberry Pi. After it loads, you should see the Pi desktop on your monitor.
5. Plug in the red (positive) and black (ground) cords into a DC power supply with anywhere from 9-20V.\* If you have followed the optional electrical steps and configured the power supply with a 24 Volt wall transformer, then you may plug the cord directly into the wall power. If you are using a 9 V battery directly attach it to the positive and negative terminals in the circuit schematic.

6. Right click on the file called `centripetal_force_03.py` and click open with Python 3 (IDLE).
7. Flip the switch on the aluminum extrusion containing the spinning motor to turn on the laser pointer, so the photodiode can track the rotational speed. If the light is dim or off try to pop out the foam back panel and replace the 9 V battery.
8. Hit the F5 key to run the program.
9. Follow program prompts. Note that all words should be typed with capital letters. The steps should read as follows:
  - (a) "Please enter the mass in kilograms. Be sure to account for the mass holder:"  
- Type in a number corresponding to your measured mass in kilograms plus the mass holder (about 50 grams).
  - (b) "Photodiode output is now set up." - This is the program checking its spi communication with the photodiode.
  - (c) "Getting radius" - This is the program checking the distance between the ultrasonic sensor and the placed mass. At this point it is critical that the mass and cross beam is placed pointing towards the aluminum extrusion where the ultrasonic sensor is secured. If it is not pointed this direction, shut off the code with the command "Ctrl C" in the terminal and restart the program.
  - (d) "Radius is [number] meters" - Ensure that this seems right, (it should be between 0.01 and 0.35 m). If something is wrong check the FAQ
  - (e) "Motor is all set to go!", "Run Motor", "Input FAST, MEDIUM, SLOW to get a spinning speed:" - Type into the terminal one speed using capital letters spelled as presented. Take a step back, the motor is now prepared to run.
  - (f) "Total time was [] seconds And [] rotations. Frequency was [] rotations per second Angular velocity was [] radians per second. Angular acceleration was [] radians per second squared. Centripetal force is: [] Newtons!"
  - (g) Indicate whether or not you would like to continue with another trial: "Would you like to try again with a different mass or different speed? Enter YES or NO (in all caps):" If you wish to engage the same process again with a different mass and radius, ensure the bar is manually pointed back towards the aluminum extrusion holding the ultrasonic sensor and repeat the above steps.
10. When done, a csv file can be found on the Pi desktop with all the appropriate data - mass as row 1, radii as row 2, frequencies as row 3, and centripetal forces as row 4. Insert a USB stick into the ports on the Raspberry Pi (under the white cover) to transfer the file to your own computer.

## 4.1 Warnings

- Do not stare directly at the laser pointer, it can cause eye pain or damage.
- After choosing a spinning speed for the motor, ensure everything and everyone is out of the way of the rotation radius to avoid being hit by the spinning aluminum rod and added masses.



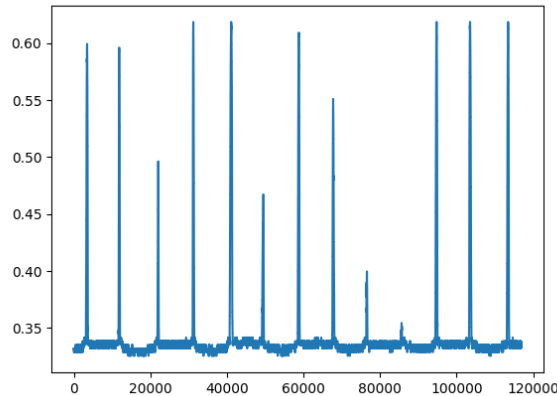
- Ensure that the securing bolt for the rod's cross beam is as tight as possible to no projectiles are launched during the rotation.
- Turn off the laser and unplug the motor when not in use. This will keep the components from burning out.
- Stand by the motor's auxiliary power while it spins, so, if any errors occur, the motor can be shut down.
- Do not disconnect any of the wires. If any wires become disconnected get assistance or carefully follow the circuit diagram to repair any damage done to the machine.
- Do not place conductive material on any of the board's circuitry.

## 4.2 FAQ's

- I followed all the instructions, why is the motor not spinning?
  - Double check that the motor is supplied with an external voltage from the DC power supply, anywhere from 9 to 24 volts (24 volts recommended by the provided power supply).
- The radius doesn't make sense!
  - Make sure you start with your rod pointed directly down the center of the board, so the mass hangs directly in front of the sensor. Otherwise, it can't find the mass and the sensor will give you a weird number for the radius.
- The motor is spinning, so why is there an output of zero rotations and wildly off numbers?
  - It is likely that you have forgotten to switch on the laser pointer - if this is the issue, it is easy to resolve. If the laser does not turn on after flipping the switch, the battery may need to be replaced. In which case, remove the foam panel and replace the 9 V battery.
  - If the battery is not at fault, uncomment (remove the `#`) to the following code lines so see the photodiode output:

```
plt.plot(volt)
plt.show()
```

If this graph does not show sharp peaks, like this graph,



you may need to order a new photodetector or resecutre the current one within its casing.

- I have pressed “Ctrl C” but the motor does not stop!
  - First of all, unplug the power to the motor to ensure everyone nearby is safe. If you are not rushed and the motor is spinning but controlled, try to go to the command prompt and “Interrupt Execution” manually with the mouse.
- Why is the motor spinning so slowly?
  - It is likely that you are not supplying enough current or power to the motor. In which case double check the power lines into the motor and ensure they are getting somewhere between 12-24 Volts either from the wall or an external DC power supply.

## 5 Appendix

### 5.1 Team Members

Name	Role	Contact
Des Dozier	Coder/Media	dcdozier@email.wm.edu
Gabrielle “G” Jawer	Coder/Leader	grjawer@email.wm.edu
Jonathan Palumbo	Electrical	jpgalumbo@email.wm.edu
Colm McDermott	Electrical/CAD	ctmcdermott@email.wm.edu
Josh Dunn	CAD	jpdunn01@email.wm.edu
Jacob Brotman-Krass	CAD/Materials	jmbrotmankrass@email.wm.edu

## 5.2 Components

<i>Name</i>	<i>Datasheet</i>	<i>How to Purchase</i>	<i>Cost</i>
Aluminum Rod	<a href="https://www.mcmaster.com/8974K22/">https://www.mcmaster.com/8974K22/</a>	McMaster Carr, SKU: 8974K22 (select 1' length)	\$2.06
Aluminum Shaft	See CAD drawing	Made in-house, see Contact Info	\$25.00
Aluminum Housing			
Mass-Hanging Assembly	See CAD drawing	Made in-house, see Contact Info	\$10.00
Photocouple Mount	See CAD drawing	Made in-house, see Contact Info	\$10.00
Lightshield	See CAD drawing	Made in-house, see Contact Info	\$15.00
Foam	N/A	Michaels Department Store	\$10.00
Wooden Base	<a href="https://www.rockler.com/1-2-baltic-birch-plywood-12-x-30">https://www.rockler.com/1-2-baltic-birch-plywood-12-x-30</a>	Rockler, Item #63404	\$7.99
1/8" x 10-24 Set Screw	<a href="https://www.mcmaster.com/91375A242/">https://www.mcmaster.com/91375A242/</a>	Pack of 50--McMaster Carr, SKU: 91375A242 Check local hardware store for purchase of individual units	\$10.25 per pack of 50 screws
M5 Nut	<a href="https://www.mcmaster.com/90592A012/">https://www.mcmaster.com/90592A012/</a>	Pack of 100--McMaster Carr, SKU: 90592A012 Check local hardware store for purchase of individual units	\$1.96 per pack of 100 nuts
M5 Bolts: 15 mm, 25 mm	15 mm: <a href="https://www.mcmaster.com/91290A231/">https://www.mcmaster.com/91290A231/</a> 25 mm: <a href="https://www.mcmaster.com/91290A252/">https://www.mcmaster.com/91290A252/</a>	15 mm: Pack of 50--McMaster Carr, SKU: 91290A231 25 mm: Pack of 50--McMaster Carr, SKU: 91290A252 Check local hardware store for purchase of individual units	15 mm: \$10.34 per pack of 50 screws 25 mm: \$8.23 per pack of 50 screws

<i>Name</i>	<i>Datasheet</i>	<i>How to Purchase</i>	<i>Cost</i>
Brushless DC Motor	<a href="https://www.moog.com/content/dam/moog/literature/MCG/moc23series.pdf">https://www.moog.com/content/dam/moog/literature/MCG/moc23series.pdf</a>	anaheimautomation.com	\$90
Ultrasonic Range Finder (HC-SR04)	<a href="https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf">https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf</a>	sparkfun.com	\$3.95
Raspberry Pi 4	<a href="https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf">https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf</a>	<a href="https://www.raspberrypi.org/">https://www.raspberrypi.org/</a>	\$35
Analog to Digital Converter (MCP3008)	<a href="https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf">https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf</a>	<a href="https://www.adafruit.com">https://www.adafruit.com</a>	\$3.75
DC Motor Driver (SN754410)	<a href="https://www.ti.com/lit/ds/symlink/sn754410.pdf">https://www.ti.com/lit/ds/symlink/sn754410.pdf</a>	sparkfun.com	\$2.35
Laser Pointer	<a href="https://www.laserglow.com/img/specifications/GAN.pdf">https://www.laserglow.com/img/specifications/GAN.pdf</a>	Can purchase on amazon, office stores, etc	\$20
Switch	[?]	Hardware stores	\$1.00
Op Amp (OP27)	<a href="https://www.analog.com/media/en/technical-documentation/data-sheets/OP27.pdf">https://www.analog.com/media/en/technical-documentation/data-sheets/OP27.pdf</a>	digikey.com	\$3.75
<b>Total Cost:</b>			<b>\$270.11</b>

## 6 Code

All source code, including test codes, are available on Github:  
 (<https://github.com/grjawer/CMONEY>)

### 6.1 Import Libraries

The following precedes all codes:

```
#import useful libraries
import spidev
import matplotlib.pyplot as plt
from scipy import signal
import time
import RPi.GPIO as GPIO
```

```

import numpy as np
import scipy
from scipy.signal import find_peaks
import pandas as pd

```

## 6.2 Motor Test

```

#Set up motor
GPIO.setmode(GPIO.BCM)
GPIO.setup(25,GPIO.OUT)
GPIO.setup(22,GPIO.OUT)
GPIO.setup(24,GPIO.OUT)
print("Motor is all set to go!")
forward = GPIO.PWM(25,100)
reverse = GPIO.PWM(22,100)

#Drive motor
forward.start(0)           #start with forward spin
reverse.start(0)

print("Run Motor")
GPIO.output(24,GPIO.HIGH)  #Tells the motor its go time
speed = input("Input FAST, MEDIUM, or SLOW to get a spinning speed:")
if speed == 'SLOW':
    forward.ChangeDutyCycle(33.3) #Forward spin speed (0%-100% Duty Cycle)
    reverse.ChangeDutyCycle(0)    #Reverse spin speed (0%-100% Duty Cycle)
    #Note:
    if Forward is not zero,
    Reverse should be zero
    and vice versa
    time.sleep(5)                #5 seconds to get smooth speed
elif speed == 'MEDIUM':
    forward.ChangeDutyCycle(66.7)
    reverse.ChangeDutyCycle(0)
    time.sleep(5)
elif speed == 'FAST':
    forward.ChangeDutyCycle(100)
    reverse.ChangeDutyCycle(0)
    time.sleep(5)
else:
    print("Not a valid command, try again.")

```

```

print("Now Stop")
GPIO.output(24,GPIO.LOW)      #Tells the motor its time to stop
forward.stop()
reverse.stop()

GPIO.cleanup()

```

### 6.3 Photodetector Test

```

#Set up communication with photodiode circuit
spi = spidev.SpiDev() #open spi bus
spi.open(0,0) #open(bus, device)
spi.max_speed_hz=1000000
spi.mode = 0b00 #spi modes; 00,01,10,11
print("Photodiode output is now set up.")

def read_adc(channel):
    if not 0 <= channel <= 7:
        raise IndexError('Invalid. enter 0, 1, ..., 7' )
    """datasheep page 19 about setting sgl/diff bit to high, hence we add 8 = 0b10
    left shift 4 bits to make space for the second byte of data[1]"""
    request = [0x1, (8+channel) << 4, 0x0] # [start bit, configuration, listen spa
    data = spi.xfer2(request) #data is recorded 3 bytes: data[0] - throw away, dat
    data10bit = ((data[1] & 3) << 8) + data[2] #shfit bits to get the 10 bit data
    return data10bit

#Gather data for frequency
volt = []
print("Getting data now... press Ctrl + c when done")
t0 = time.time()      #inital time
try:
    while 1:
        v_volt = read_adc(1) * 3.3 / 1024 #get voltage from photodiode circuit
        volt.append(v_volt) #add voltage to list, light is hitting
except KeyboardInterrupt:
    pass
tf = time.time()      #final time

#Calculate centripetal force

```

```

plt.plot(volt)
plt.show()

peaks, _ = find_peaks(volt, height=0.35, distance = 5000)
rotations = len(peaks)
t_total = tf - t0
print('Total time was', t_total, 'seconds')
print(rotations, 'rotations')
frequency = rotations / t_total          #frequency!
print('Frequency was ', frequency, 'rotations per second')
omega = 2 * np.pi * frequency          #angular velocity!

```

#### 6.4 Rangefinder Test

```

#Set up communication with distance sensor and get initial radius
GPIO.setmode(GPIO.BCM)
trig = 5
echo = 16
GPIO.setup(trig,GPIO.OUT)
GPIO.setup(echo,GPIO.IN)
T_room = 26 #room temperature is 26 C.
print (f'{trig} is the trigger, and {echo} is the echo')
try:
    distlist = []

    for i in range(0,10):
        GPIO.output(trig,GPIO.LOW)
        print (f'a 10us pulse is sent to {trig}, ultrasound emitted.')
        time.sleep (0.2) # waiting for 1 second to start
        GPIO.output(trig,GPIO.HIGH)
        time.sleep(0.00001) #supply a 10 us pulse to the trigger input pin to star
        GPIO.output(trig,GPIO.LOW)
        while GPIO.input(echo) == 0:
            t0 = time.time() #record time now as t0
        while GPIO.input(echo) == 1:
            t1 = time.time() #record the time that the echo pin recieves the reflec
        t = t1-t0 #calculate the time interval
        v = 20.05* math.sqrt(273.16+T_room)
        d = t*v/2 #calculate the distance in m
        dm = round(d,2)
        dcm = round(100*d, 2)

```

```

        din = round(39.37*d,2)
        print (f'The distance is {round(d,2)} m, {round(100*d, 2)}cm, {round(39.37

        distlist.append(dcm)

    davg = sum(distlist)/len(distlist)
    print(davg)
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()

```

## 7 centripetal\_force\_03.py

```

def read_adc(channel):
    if not 0 <= channel <= 7:
        raise IndexError('Invalid. enter 0, 1, ..., 7' )
    """datasheep page 19 about setting sgl/diff bit to high, hence we add 8 = 0b10
    left shift 4 bits to make space for the second byte of data[1]"""
    request = [0x1, (8+channel) << 4, 0x0] # [start bit, configuration, listen spa
    data = spi.xfer2(request) #data is recorded 3 bytes: data[0] - throw away, dat
    data10bit = ((data[1] & 3) << 8) + data[2] #shfit bits to get the 10 bit data
    return data10bit

def Find_cforce(): #Gets centripetal force from user input of given mass and set

    mass = float(input("Please enter the mass in kilograms, making sure to account

    #Set up communication with distance sensor and get initial radius
    GPIO.setmode(GPIO.BCM)
    trig = 5
    echo = 16
    GPIO.setup(trig,GPIO.OUT)
    GPIO.setup(echo,GPIO.IN)
    T_room = 26 #room temperature is 26 C.
    print('Getting radius...')
    #print (f'{trig} is the trigger, and {echo} is the echo')

    try:
        distlist = []

```



```

for i in range(0,30):
    GPIO.output(trig,GPIO.LOW)
    time.sleep (0.2) # waiting for 1 second to start
    GPIO.output(trig,GPIO.HIGH)
    time.sleep(0.00001) #supply a 10 us pulse to the trigger input pin to
    GPIO.output(trig,GPIO.LOW)
    while GPIO.input(echo) == 0:
        t0 = time.time() #record time now as t0
    while GPIO.input(echo) == 1:
        t1 = time.time() #record the time that the echo pin recieves the re
    t = t1-t0 #calculate the time interval
    v = 20.05* math.sqrt(273.16+T_room)
    d = t*v/2 #calculate the distance in m
    dm = d + 0.01 #to account for thickness of mass holder
    #print (f'The distance is {round(d,2)} m')

    distlist.append(dm)

    davg = sum(distlist)/len(distlist)
    #print(davg)
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()

distance = 0.375
radius = distance - davg #radius is distance from shaft to mass = (total di
print('Radius is ', radius, 'meters')

#Set up motor
GPIO.setmode(GPIO.BCM)
GPIO.setup(25,GPIO.OUT)
GPIO.setup(22,GPIO.OUT)
GPIO.setup(24,GPIO.OUT)
print("Motor is all set to go!")
forward = GPIO.PWM(25,100)
reverse = GPIO.PWM(22,100)

#Drive motor
forward.start(0) #start with no spin

```

```

reverse.start(0)

print("Run Motor")
GPIO.output(24,GPIO.HIGH)      #Tells the motor its go time
speed = input("Input FAST, MEDIUM, or SLOW to get a spinning speed:")
if speed == 'SLOW':
    forward.ChangeDutyCycle(33.3) #Forward spin speed (0%-100% Duty Cycle)
    reverse.ChangeDutyCycle(0)    #Reverse spin speed (0%-100% Duty Cycle)
    #Note: if Forward is not zero, Reverse should be zero
    time.sleep(.5)               #5 seconds to get smooth, constant speed
elif speed == 'MEDIUM':
    forward.ChangeDutyCycle(66.7)
    reverse.ChangeDutyCycle(0)

    time.sleep(.5)
elif speed == 'FAST':
    forward.ChangeDutyCycle(100)
    reverse.ChangeDutyCycle(0)

    time.sleep(.5)
else:
    print("Not a valid command, try again.")

#Gather data for frequency
volt = []
print("Getting data now... please wait.")
t0 = time.time()                #initial time
for i in range(10000):          #spin for about 10 seconds
    v_volt = read_adc(1) * 3.3 / 1024 #get voltage from photodiode circuit i
    volt.append(v_volt)           #add voltage to list
    time.sleep(.001)
tf = time.time()                #final time

print("Now Stop")
GPIO.output(24,GPIO.LOW)       #Tells the motor its time to stop
forward.stop()
reverse.stop()

GPIO.cleanup()

#Calculate centripetal force

```

```

    peaks, _ = find_peaks(volt, height=0.25, distance = 500)    #light is hitting s
    rotations = len(peaks)    #number of peaks = number of rotations
    t_total = tf - t0    #total time for rotations
    print('Total time was', t_total, 'seconds')
    print('And', rotations, 'rotations.')
    frequency = rotations / t_total    #frequency!
    print('Frequency was ', frequency, 'rotations per second')
    omega = 2 * np.pi * frequency    #angular velocity!
    print('Angular velocity was ', omega, 'radians per second')
    alpha = radius * omega    #angular acceleration
    print('Angular acceleration was ', alpha, 'radians per second squared')
    force = mass * radius * omega**2    #CENTRIPETAL FORCE!
    print('Centripetal force is:', force, 'Newtons!')

    #plt.plot(volt)
    #plt.show()

    return mass, radius, frequency, force

go = 'NO'
while go != 'YES':
    print("Let's find centripetal acceleration! Make sure everything is plugged in")
    go = input("Are you ready? Enter YES or NO (in all caps):")

#Set up communication with photodiode circuit
spi = spidev.SpiDev() #open spi bus
spi.open(0,0) #open(bus, device)
spi.max_speed_hz=1000000
spi.mode = 0b00 #spi modes; 00,01,10,11
print("Photodiode output is now set up.")

Masses = []
Radii = []
Frequencies = []
Forces = []

Mass, Radius, Frequency, Force = Find_cforce()

Masses.append(Mass)
Radii.append(Radius)

```

```

Frequencies.append(Frequency)
Forces.append(Force)

again = 'YES'
while again == 'YES':
    again = input("Would you like to try again with a different mass or different
    if again == 'NO':
        pass
    Mass1, Radius1, Frequency1, Force1 = Find_cforce()
    Masses.append(Mass1)
    Radii.append(Radius1)
    Frequencies.append(Frequency1)
    Forces.append(Force1)

# write to csv:
filename = '/home/pi/Desktop/' + str(time.time()) + '.csv'
with open(filename, 'w', newline='') as csvfile:
    writer = csv.writer(csvfile, delimiter=' ', quotechar='|', quoting=csv.QUOTE_M
    writer.writerow(Masses)
    writer.writerow(Radii)
    writer.writerow(Frequencies)
    writer.writerow(Forces)

print('Thank you for using CMONEY! Your data is available as a csv file on the Pi

```