MONASH University
Engineering

# Assessment cover sheet

| Unit and student details | | | | | |
|---|---|---|---|---|---|
| **Unit code** | ECE 4075 | **Unit title** | Real Time Embedded Systems | | |
| If this is a group assignment, each student must include their name and ID number and sign the student statement. | | | | | |
| **Student ID** | 29439701 | **Surname** | Mak | **Given names** | Clifton |
| **Student ID** | | **Surname** | | **Given names** | |
| **Student ID** | | **Surname** | | **Given names** | |
| **Student ID** | | **Surname** | | **Given names** | |

| Assessment details | | | |
|---|---|---|---|
| **Title of assignment /lab** | Pendulum Mimicking System | **Authorised group assignment** | Yes ☐ No ☒ |
| **Lecturer/tutor** | Dr. Vishnu Monn Baksaran | **Tutorial day and time** | Friday, 3-6pm |
| **Due date** | 30th May 2021 | **Date submitted** | 30th May 2021 |
| **Has any part of this assessment been previously submitted as part of another unit/course?** | | | Yes ☐ No ☒ |

## Submission date and extensions

All work must be submitted by the due date. If an extension of work is required, please complete and submit a Special Consideration application (in-semester assessment task) form to your examiner/lecturer/tutor.

## Plagiarism and collusion

***Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations.***

**Plagiarism**: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

**Collusion**: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

## Student statement and signature

**Student Statement**

I have read the university's Student Academic Integrity Policy and Procedures.

I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://www.monash.edu/legal/legislation/current-statute-regulations-and-related-resolutions

I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.

No part of this assignment has been previously submitted as part of another unit/course.

I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:

i. provide to another member of faculty and any external marker; and/or
ii. submit it to a text matching software; and/or
iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.

I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

| | | | |
|---|---|---|---|
| **Student signature** | Clifton | **Date** | 30th May 2021 |
| **Student signature** | | **Date** | |
| **Student signature** | | **Date** | |
| **Student signature** | | **Date** | |

**Please note that it is your responsibility to retain a copy of your assignment/laboratory work**

# Real-time pendulum mimicking system using Quarter Period and Lookup Table

Clifton Mak Ren Ming
29439701
ECE4075 – Real time embedded systems
*Electrical and Computer Systems Engineering*
Monash University Malaysia
cmak0003@student.monash.edu

***Abstract*— Using the quarter period and lookup table algorithm, a real-time pendulum mimicking system was successfully developed. The quarter period was used to ensure that the motor can move back to its zeroth position without triggering the photoelectric sensor interrupt, therefore reducing delays. The lookup table was used to set the speed and position of the motor more accurately. The system had minimal delays in motion and is very accurate. The PID controller used was also successful in handling any overshoots of the motor from the desired position.**

***Keywords—Pendulum, Quarter Period, Photoelectric sensor, Lookup table, PID***

## I. INTRODUCTION

In this assignment, a real-time pendulum mimicking system is designed, where a DC motor with a pointer mimics the motion of the swinging pendulum with minimum delays and errors. The objectives of the assignment are as follows [1]:

1. Develop and practice real time embedded system design, testing and debugging techniques.
2. Understand the trade-off between hardware and software implementations in a real time embedded system by studying a software hardware co-design problem.
3. Interpret timing information from a sensor and its relationship to a physical system.
4. Design modules that respond to sensor changes within hard deadlines.
5. Design a PID motor controller with PWM output to produce desired motions in real time.

A simple pendulum is a type of ideal pendulum whereby the pendulum bob is replaced by a small particle that is known as a point mass [2]. The structure of the simple rod pendulum can be seen in Figure 1. In the simple pendulum, the only force acting on it is the gravitational force.

According to the authors from [2], the motion of the simple rod pendulum follows the linear simple harmonic motion system. The period for one swing is given by the following equation.

$$T = 2\Pi\sqrt{\frac{l}{g}} \tag{1}$$

where T is the period for one oscillation, l is the length of the rod and g is the gravitational acceleration.

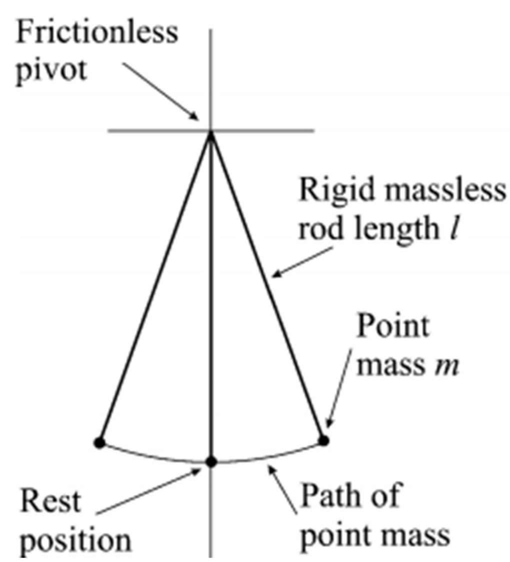



**Figure 1: Simple Pendulum Structure [2]**

The acceleration of the simple rod pendulum due to centripetal force is given by:

$$|a| = \omega^2 r \tag{2}$$

where $a$ is the normal acceleration, $\omega$ is the angular velocity and r is the length of the rod.

Lastly, the velocity of the point mass as it passes through rest position is given by:

$$v = \sqrt{2gl(1 - cosB)} \quad (3)$$

where v is the velocity of the point mass, g is the gravitational acceleration, l is the length of the rod and B is the angle of swing at the end of the arc.

In this assignment, the mimicking system will be implemented using the quarter period and lookup table. The quarter period will ensure that the motor/pointer moves downwards when the pendulum swings downwards. The lookup table on the other hand is used to give pre-obtained speed and position values for the motor.

## II. DESIGN

Specify the applied pendulum mimicking technique. Include the following in your design (using sub-sections):

*a)* Pendulum mimicking structure.

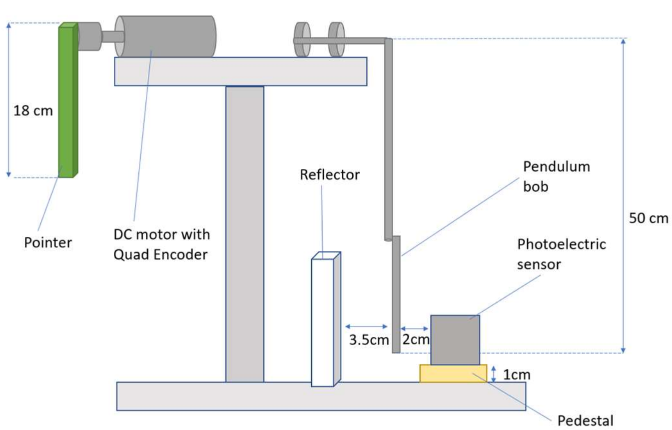



**Figure 2: Technical drawing of Pendulum Mimicking Structure**

Figure 1 shows the technical drawing of the pendulum mimicking structure. The motor will be used to control the pointer which will be used to indicate the effectives of the mimicking algorithm. The pendulum bob will swing and interrupt the photoelectric sensor beam. Furthermore, the photoelectric sensor is also placed atop a pedestal to ensure that photoelectric sensor beam can hit the pendulum bob. Additionally, take note that the quadrature encoder is inside the DC motor.

*b)* Circuit schematics.

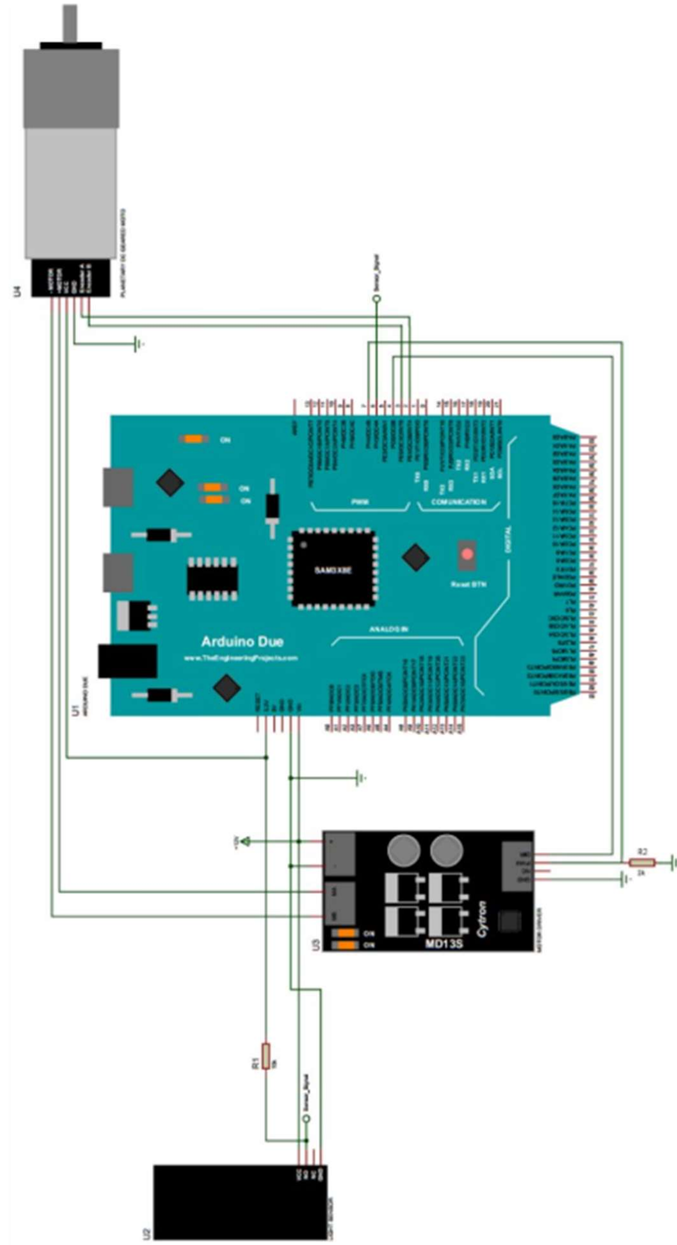


**Figure 3: Overall Circuit Schematics [1]**

*c)* Determining direction and speed of the pendulum

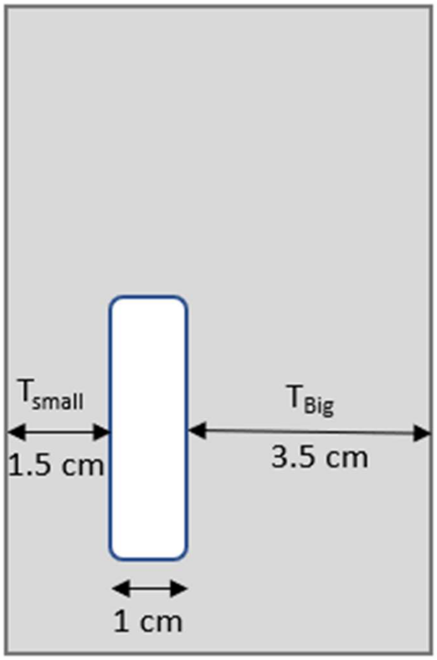



**Figure 4: Pendulum bob structure**

Figure 4 shows the structure of the pendulum bob. It can be seen on the pendulum bob that there is a small 1 cm hollow section separating the bob into two uneven sections, that are 1.5cm and 3.5cm long respectively. The determination of the direction algorithm is based on the uneven sections on the pendulum bob.

The photoelectric sensor will trigger an interrupt on a rising and falling edge signal. When the pendulum interrupts the photoelectric beam, a rising edge will occur and when the photoelectric beam first gets uninterrupted after getting triggered initially, a falling edge will occur. By using this principle, if the pendulum swings from the right, the interrupt will first be triggered by a rising edge from the entering of the smaller section of the pendulum bob. It will then be triggered by a falling edge as the photoelectric beam passes through the hollow section, after which it will be triggered again by a rising edge as the beam cuts the bigger section of the pendulum bob. And when the pendulum bob leaves the sensing area, a falling edge will trigger the interrupt. This is clearly illustrated in Figure 5.

Referring to Figure 5, the timing between Rising Edge 1 and Falling Edge 1 will be recorded as $T_{small}$ and the timing between Rising Edge 2 and Falling Edge 2 will be recorded as $T_{big}$. Due to the uneven sections on the pendulum bob, the duration for one side would be smaller ($T_{small}$) than the other ($T_{big}$). If the pendulum is swinging from the right, the same process occurs but from the opposite direction. After obtaining the durations, a comparison will be made. If $T_{small}$ comes before $T_{big}$, the pendulum is coming from the right and if $T_{big}$ comes before $T_{small}$, the pendulum is coming from the left.

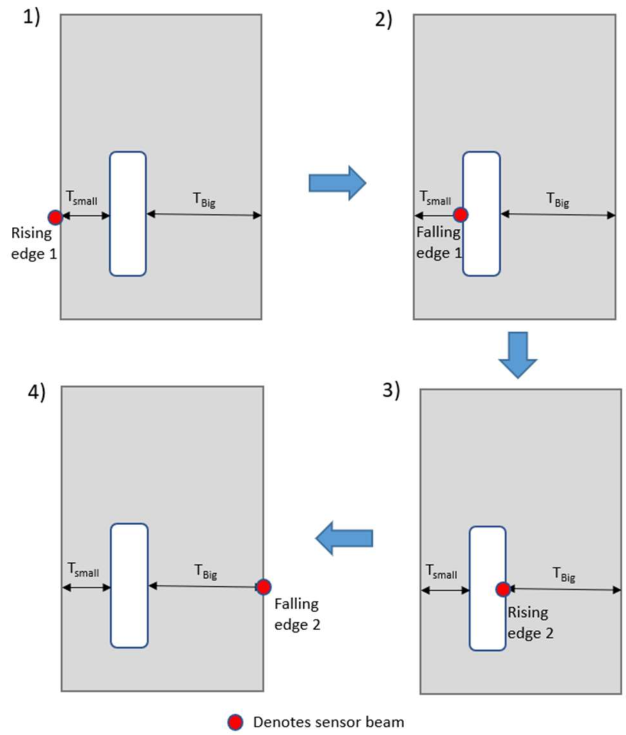



**Figure 5: Interruption of sensor beam as the pendulum swings**

The value of $T_{small}$ will then be used to find the speed of pendulum. The speed can be found based on Eq (1).

$$Speed = \frac{Distance}{Time}, \tag{1}$$

whereby Speed is the speed of the pendulum, Distance is the length of the small section (1.5cm) and Time is $T_{small}$.

As the pendulum moves faster $T_{small}$ will be shorter and as the pendulum moves slower, $T_{small}$ will be larger. The speed of the pendulum will be used as a basis to obtain the correct PWM values to set the motor speed for different scenarios.

*d)* Quarter Period

The quarter period is used as part of the mimicking algorithm. Figure 6 shows the motion of the pendulum as it swings from right to left. When the pendulum swings from the right to the sensing area of the photoelectric sensor in motion 1, the time where the pendulum first interrupts the sensor beam is recorded as enter_time. As the pendulum swings from the sensing area to the top in motion 2 and back to the sensing area again in motion 3, the time is recorded as exit_time. The duration between enter_time and exit_time is the total duration for both motions 2 and 3, which is also known as the half period.

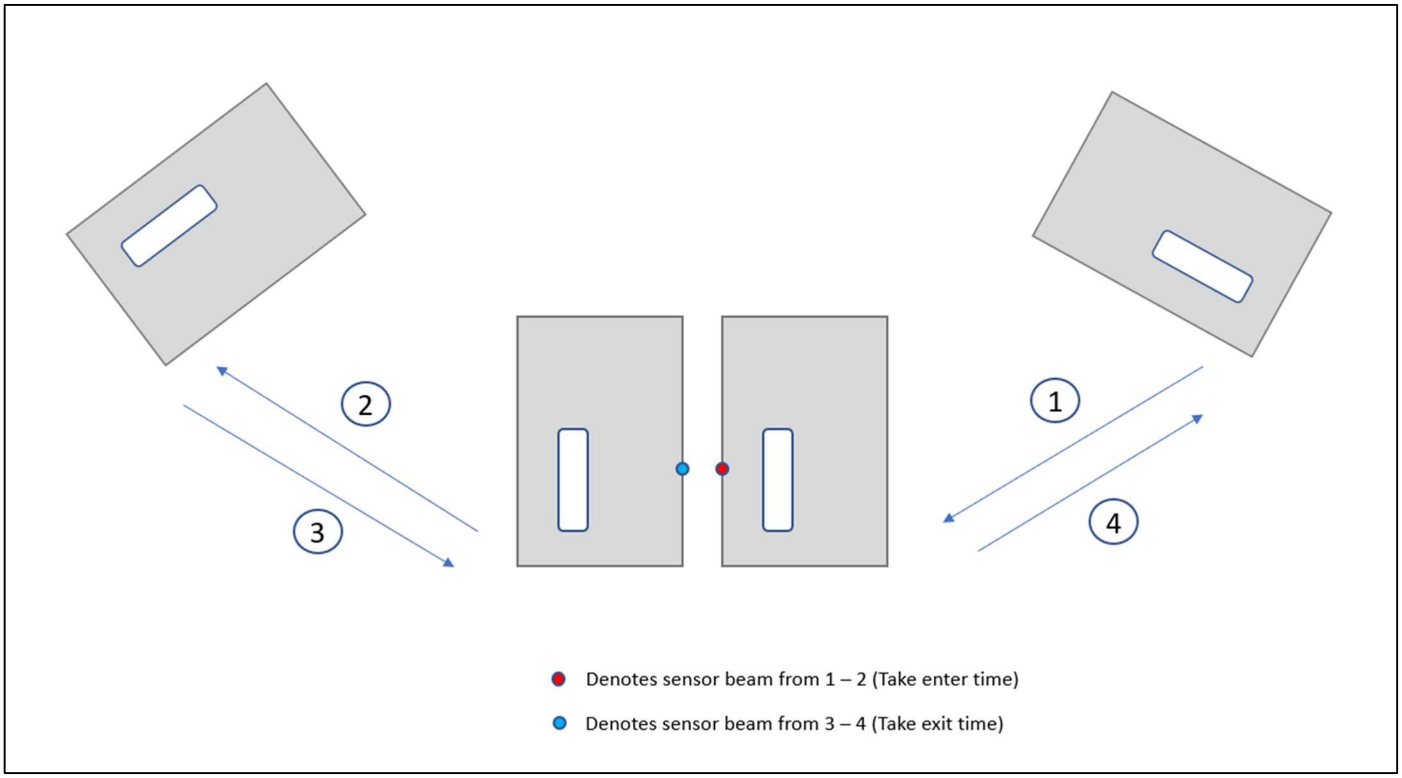



**Figure 6: Pendulum motion to illustrate Quarter Period Collection**

The quarter period can be obtained by halving the half period and it corresponds to the time it takes for the pendulum make either motion 1,2,3 or 4. It is assumed that the maximum amplitude of the pendulum would be similar on both sides for 1 iteration of motion (1,2,3,4). Therefore, the quarter period obtained will be assumed to be similar in motion 4. During motion 4, if the duration from exit_time and current time is greater than the quarter period, the motor will be adjusted to move to start going back down using motion 1. And when it interrupts the sensor beam again, the quarter period would be calculated and the whole process repeats until the pendulum stops swinging. Take note that only at the start, when the pendulum is first released, the quarter period will not be calculated in the first interruption of the sensor beam. This is because it is impossible to determine when the pendulum would be released.

*e)* Lookup Table

A pre-lookup table with PWM and setpoint position values was used to further enhance the mimicking algorithm. The setpoint position will be used by the PID controller to set the angular position of the motor. The lookup table values will be selected based on the $T_{small}$ values obtained. Table 1 summarizes the lookup table values.

**Table 1: Summary of Lookup Table**

| $T_{small}$ | PWM | Setpoint position |
|---|---|---|
| 5 | 18 | 45 |
| 7 | 18 | 42 |
| 8 | 18 | 38 |
| 9 | 17 | 26 |
| 10 | 17 | 26 |
| 11 | 17 | 26 |
| 12 | 15 | 26 |
| 13 | 15 | 26 |
| 14 | 15 | 23 |
| 15 | 11 | 23 |
| 16 | 11 | 23 |
| 17 | 11 | 23 |
| 18 | 11 | 23 |
| 19 | 11 | 21 |
| 20 | 10 | 21 |
| 21 | 10 | 14 |
| 22 | 10 | 14 |
| 23 | 10 | 14 |
| 24 | 9 | 14 |
| 25 | 9 | 11 |
| 26 | 9 | 11 |
| 27 | 9 | 11 |
| 28 | 9 | 11 |
| 29 | 8 | 11 |
| 30 | 8 | 11 |
| 32 | 8 | 8 |

| | | |
|---|---|---|
| 33 | 8 | 8 |
| 34 | 7 | 8 |
| 35 | 7 | 8 |
| 36 | 7 | 8 |
| 39 | 7 | 8 |
| 40 | 7 | 8 |
| 42 | 7 | 7 |
| 44 | 7 | 7 |
| 47 | 7 | 7 |
| 48 | 7 | 7 |
| 52 | 7 | 7 |
| 54 | 7 | 7 |

### *f)* PID controller

The PID controller is used to move the motor based on the given setpoint position. It is also used to handle any overshoots from the desired position. The setpoint position represents the desired quadrature encoder counts and the current position represents the current quadrature encoder count. The PID controller will output positive, negative values or zero depending on the setpoint position and current position. Table 2 summarizes the outputs of the PID controller and the selected Kp, Ki and Kd values. These PID output values will be used as conditions to move the motor in the right direction and speed taken from the lookup table.

**Table 2: Summary of PID outputs and Kp,Ki,Kd values**

| Condition | PID output |
|---|---|
| Current position > Setpoint position | < 0 |
| Current position = Setpoint position | 0 |
| Current position < Setpoint position | > 0 |
| Kp = 2.1<br>Ki = 0.001<br>Kd = 30 | |

### *g)* Quadrature Encoder

The quadrature encoder produces pulses from Channels A and B that are 90 degrees out of phase from each other. A timer counter was used to count the rising edges of both Channels A and B. The count values are converted to values that go from 0 to 196 for one revolution in the clockwise direction and 0 to -196 in the anti-clockwise direction. It is converted with the following equation.

$$position = (position \,\%\, \text{count\_per\_revolution}) + 1 \qquad (2)$$

where position is the current count value from the tc counter and count_per_revolution (196) is the count value for one whole revolution.

### h) Flowcharts Summarizing

The flowcharts in Figures 7, 8 and 9 summarizes the mimicking algorithm.

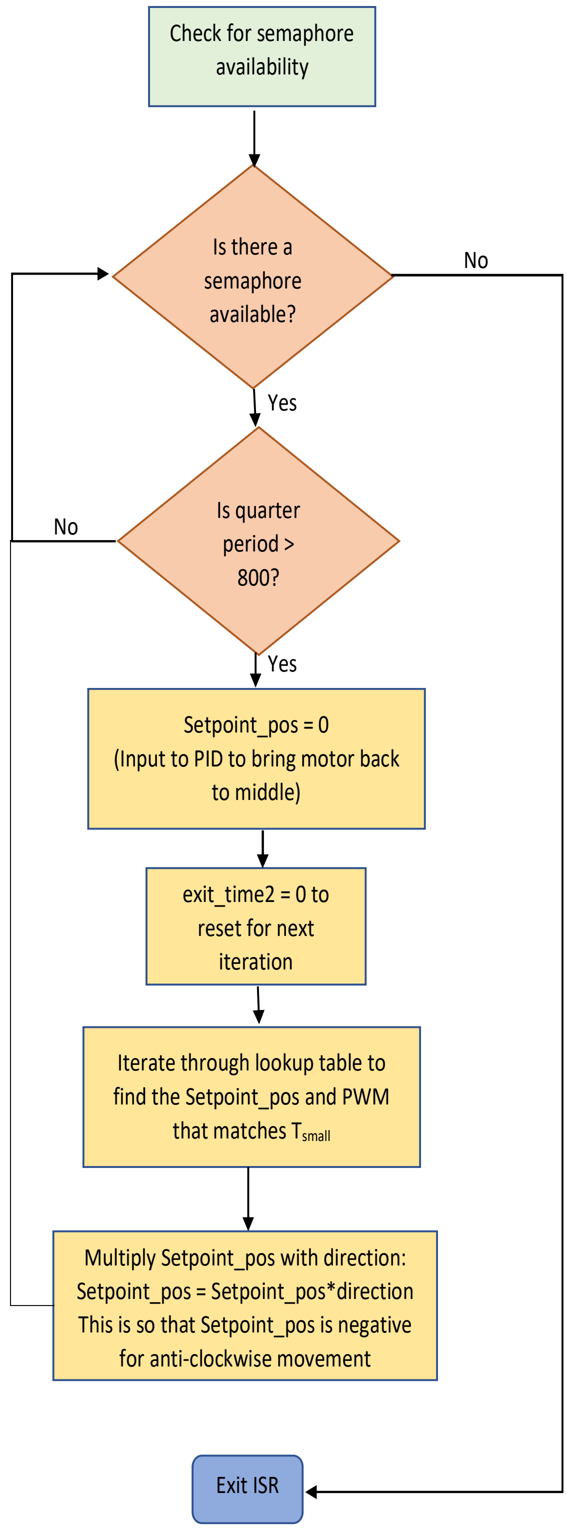



**Figure 7: Lookup Table Task**

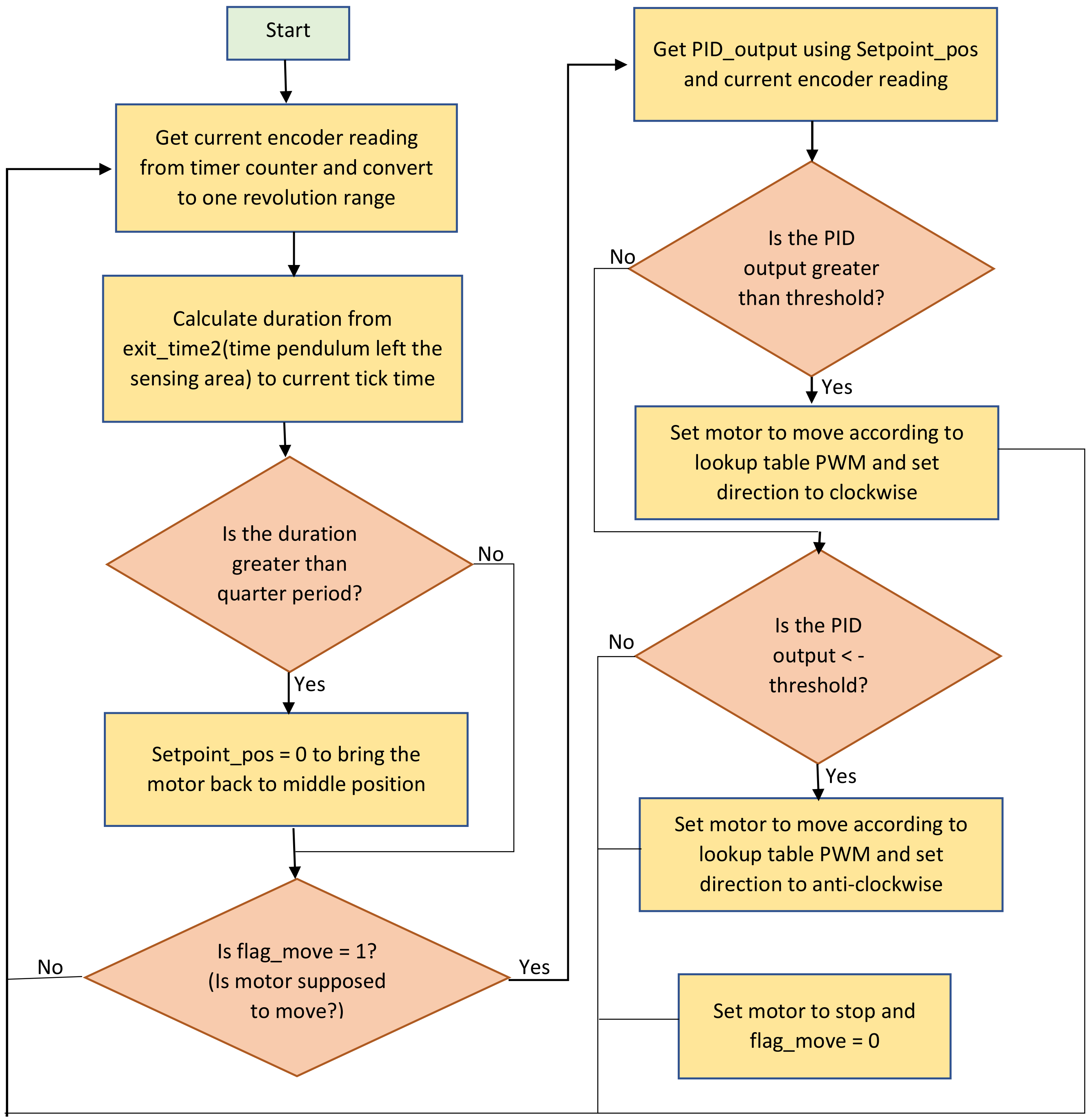



**Figure 8: Moving Motor Using PID**

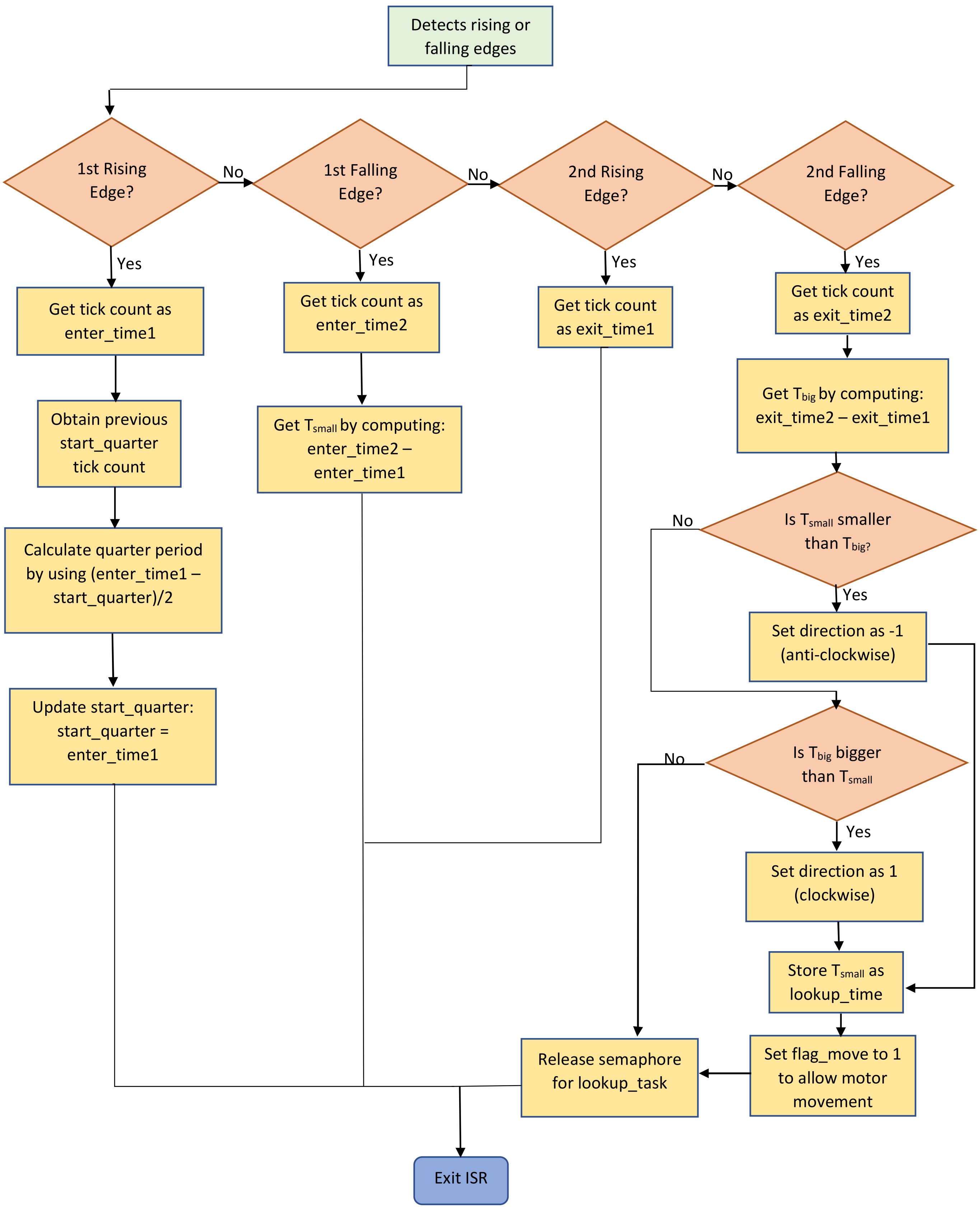



**Figure 9: Photoelectric ISR: Direction Detection and Quarter Period Computation**

## III. Results & Discussion

Test 1: Testing position accuracy at different release angles

<u>Hypothesis</u>

With the implementation of the quarter period and the lookup table, the motor and pendulum positions would still be synchronized at different motor angles.

<u>Method</u>

The pendulum was released at angles of 45°, 60° and 90°and when the motor/pointer first moves to its maximum amplitude, the PWM pin was disconnected so that the motor would stop moving, after which the angle of the pointer position would be measure relative to the middle position.

<u>Results</u>

**Table 3: Summary of Testing results at different release angles (position accuracy)**

| Angle | Test | Measured Angle (°) | Error (°) |
|---|---|---|---|
| 45° | 1 | 45.3 | +0.3 |
| | 2 | 45.4 | +0.4 |
| | 3 | 45.4 | +0.4 |
| | 4 | 45.3 | +0.3 |
| | 5 | 45.2 | +0.2 |
| | | Average | +0.32 |
| 60° | 1 | 60.3 | +0.3 |
| | 2 | 60.2 | +0.2 |
| | 3 | 60.0 | 0 |
| | 4 | 60.3 | +0.3 |
| | 5 | 60.3 | +0.3 |
| | | Average | +0.22 |
| 90° | 1 | 90.2 | +0.2 |
| | 2 | 90.2 | +0.2 |
| | 3 | 90.0 | 0 |
| | 4 | 90.3 | +0.3 |
| | 5 | 90.3 | +0.3 |
| | | Average | +0.2 |

<u>Conclusion</u>

It can be seen that the average errors are all less than 0.5 and the average percentage errors are all less than 1%, which is considered negligible. Therefore, this shows that the quarter period is collected correctly and the assumption that the quarter period on both swinging sides(left and right) would be similar holds true. Furthermore, this also proves that the lookup table position values are suitable.

Test 2: Testing speed accuracy at different release angles

<u>Hypothesis</u>

Similar to Test 1, with the implementation of the quarter period and the lookup table, the speed of the motor and pendulum would still be synchronized at different motor angles.

<u>Method</u>

The pendulum was released at angles of 45°, 60° and 90°. The accuracy/synchronization of the motor and pendulum speeds are observed visually through video recording. Note that this test can only be done after ensuring position accuracy.

<u>Results</u>

**Table 4: Summary of Testing results at different release angles (speed accuracy)**

| Angle | Test | Comments |
|---|---|---|
| 45° | 1 | Synchronized |
| | 2 | Synchronized |
| | 3 | Synchronized |
| | 4 | Synchronized |
| | 5 | Synchronized |
| 60° | 1 | Synchronized |
| | 2 | Synchronized |
| | 3 | Synchronized |
| | 4 | Synchronized |
| | 5 | Synchronized |
| 90° | 1 | Synchronized |
| | 2 | Synchronized |
| | 3 | Synchronized |
| | 4 | Synchronized |
| | 5 | Synchronized |

<u>Conclusion</u>

It can be seen that the both the motor and pendulum speeds at different release angles are synchronized. Therefore, it proves that the lookup table PWM values are suitable.

Test 3: Testing to see if pointer/motor stops at middle position when pendulum stops

<u>Hypothesis</u>

The pointer will go back to its zeroth position in the middle when the pendulum stops swinging as the photoelectric sensor interrupt is no longer being triggered.

<u>Method</u>

The pendulum will be released from any angle and will be left to swing until it stops. Then, the angular position of the pointer relative to the middle zeroth position will be measured.

<u>Results</u>

**Table 4: Summary of Stopping Angular Positions for 10 iterations**

| Test | Stopping Angular Position (°) |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 5 |
| 4 | 5 |
| 5 | 5 |
| 6 | 0 |
| 7 | 0 |
| 8 | 0 |
| 9 | 0 |
| 10 | 0 |

<u>Conclusion</u>

On an average of 10 trials, about 70% of the time, the pointer would go back to its zeroth position. About 30% of the time, the pointer would be about 5° off. This could be due to the fact that the algorithm only allows the motor to move if it complete all 4 stages of interrupts as shown in Figure 4. When the pendulum slows down, the final interrupt stage may not have been completed, and as such, the motor will not be able to move back to the zeroth position. The error however is very small and can be considered to be negligible.

Test 4: Testing suitability of Kp values

<u>Hypothesis</u>

As the Kp value decreases, the accuracy in the position of the motor relative to the pendulum decreases.

<u>Method</u>

The pendulum will be released at an angle of 90° with different Kp values. Similar to Test 1, when the motor/pointer first moves to its maximum amplitude, the PWM pin was disconnected so that the motor would stop moving, after which the angle of the pointer position would be measure relative to the middle position.

<u>Results</u>

**Table 5: Summary of Results for Different Kp values**

| Kp | Measured Angle (°) | Error (°) |
|---|---|---|
| 2.1 | 95 | -5 |
| 1.8 | 85 | -10 |
| 1.5 | 78 | -17 |
| 1.2 | 70 | -25 |
| 0.9 | 65 | -30 |

<u>Conclusion</u>

It can be seen that at different Kp values, the error in the position of the pointer increases as the Kp value decreases. This is because the Kp value directly affects the steady state error. With a lower Kp value, the steady state error increases and hence the accuracy of the pointer in mimicking the pendulum becomes worse. This proves that the selected Kp value of 2.1 is suitable as the percentage error is only approximately 0.6 %.

## IV. Conclusion

From the results obtained, it is clear the proposed algorithm works efficiently to enable mimicking of the motor/pointer with the swinging pendulum. Using the quarter period and lookup table algorithm, the position and speed accuracies produced negligible errors. Using the PID controller, the algorithm was able to help the motor to recover from any overshooting from the desired position. A limitation would be that when the system is first started, the pendulum is required to swing ¾ of an oscillation to collect the quarter period before the motor starts moving to mimic the pendulum. Future work could involve estimating the initial quarter period based on $T_{snall}$ or $T_{big}$. A lookup table could be used where pre-tested quarter periods based on different $T_{small}$ or $T_{big.}$ That way, initial delays in starting the mimicking system can be avoided. Another limitation in the system is that the pointer does not completely go back to its zeroth position all the time when the pendulum stops swinging. There will sometimes be a small deviation. This happens as the software algorithm requires a certain condition to be fulfilled before allowing the motor to move. This condition relies on the interruption of the photoelectric sensor beam as explained in Test 3. Future work could attempt to solve this issue by attempting to set an internal software timer that tells the motor to move back to its zeroth position if it is stuck in the wrong state(where the motor is not told to go back to its zeroth position yet).

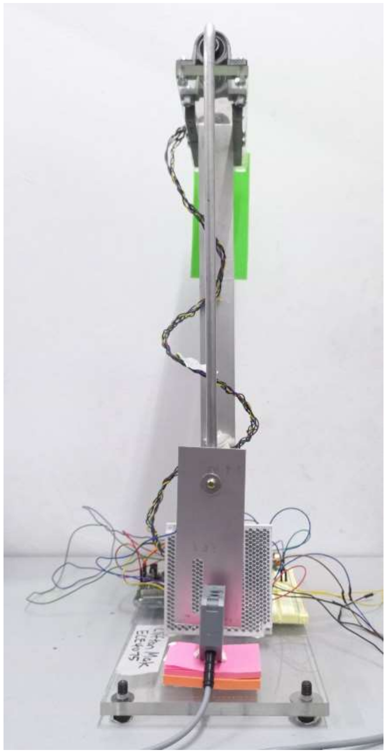
## References

[1] V.B.Monn,"Pendulum Mimicking System," Monash University, Malaysia, 2021.

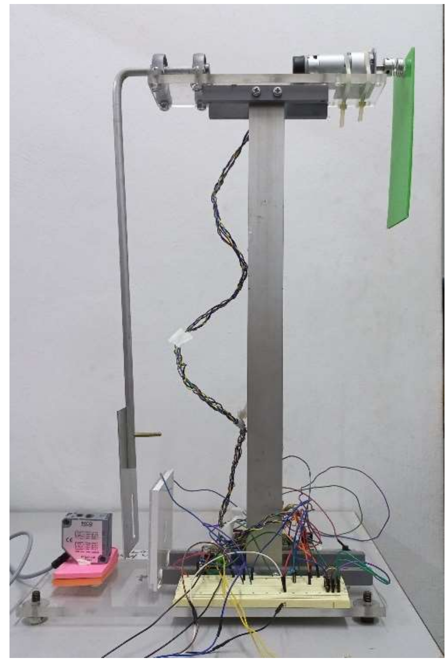
[2] L. P. Pook, Understanding Pendulums A Brief Introduction, 1st ed. 2011. ed. Dordrecht : Springer Netherlands : Imprint: Springer, 2011.

Appendix

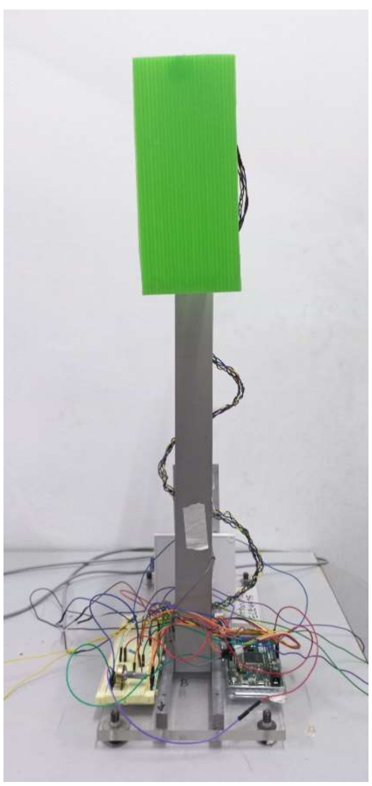
Pendulum Structure From Different Angles



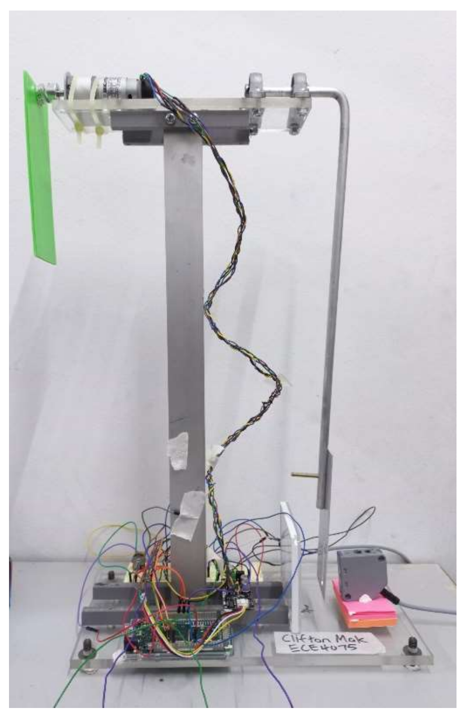
Front view



Left view



Back view



Right view