Final Design
UAVPayload^{TAQ}
Sampling Tube Subsystem

Prepared by	J.N____________________________ Jeremy Naylor, Sampling Tube Design Lead	Date	28/10/2022
Checked by	M.B____________________________ Marissa Bowen, Project Manager & Enclosure Design Lead	Date	28/10/2022
Approved by	M.B____________________________ Marissa Bowen, Project Manager & Enclosure Design Lead	Date	28/10/2022
Authorised for use by	________________________________ Dr. Felipe Gonzalez, Project Coordinator	Date	________________

Queensland University of Technology
Gardens Point Campus
Brisbane, Australia, 4001.

Revision Record

Document Issue/Revision Status	Description of Change	Date	Approved
0.1	**Draft Submission**	21/10/22	J.N
1.0	**Final Submission**	28/10/22	J.N
1.1	**Final Alterations**	28/10/22	J.N

Table of Contents

Paragraph **Page No.**

List of Figures

Figure **Page No.**

List of Tables

Table **Page No.**

Definitions

AQ	Air Quality Sensor
ED	Enclosure Design
GPIO	General Purpose Input Output
HLO	High-Level Objective
PPM	Pulse Position Modulation
PWM	Pulse Width Modulation
QUT	Queensland University of Technology
ST	Sampling Tube
WVI	Web Visualization Interface

1 Introduction

This document is a final design of UAVPAYG19 for the subsystem of Sampling Tube (ST). The final design report is an essential part of the System Engineering Design Approach as it provides an overview of the subsystem and details the design of the final solution. The final design report is the last report to be made before the acceptance testing of the integrated solution.

This document outlines the requirements provided in RD/1 and how these requirements were achieved for the sampling tube subsystem to help prove that the client requirements were achieved. A Subsystem Architecture will be provided with each final design documentation, describing the interfaces that the subsystems will contain to show integration. These documents will help show the client and group members how this subsystem has fulfilled its requirements, containing all the details and features of the ST subsystem.

1.1 Scope

The following documentation will introduce the final design of the Sampling Tube subsystem. It will contain a table of detailed requirements that the subsystem and how they should be met. All interfacing connections will be documented in the system architecture and outlined to show the integration of other required subsystems. As this is the final design, no additional changes will be made to this document upon completion.

1.2 Background

The Queensland University of Technology's Airborne System Lab (ASL) has commissioned the group UAVPAYG19 to design and develop a payload capable in detecting specific objects, recording air quality data to be displayed on a web interface and to pierce a ground sample. This payload is to be attached to a S500 UAV which completed an automated flight path. The payload is mounted on the bottom of the UAV using a proved bracket. This payload must contain all components to complete its required tasks. These components are:

- Raspberry Pi 3b+
- Raspberry Pi Camera
- Pimoroni enviro sensor
- DF15RSMG 360 Degree Motor

The project is required to identify three targets, a valve (In open or closed position), a fire extinguisher and an ArUCO marker. The Pimorino sensor is to be used to record air temperature, pressure humidity, light and potentially hazardous gas level data. This data along with a live feed of the Raspberry Pi Camera is to be visualized on a Web Interface. Lastly, a soil sample must be obtained using a sampling mechanism.

2 Reference Documents

2.1 QUT Avionics Documents

Table 1 – QUT Avionics Documents

RD/1	UAVPayloadTAQ System Requirements	System-Requirments (UAVPayloadTAQ 2022)
RD/2	Advanced Sensor Payload for UAV Target Detection and Air Quality Monitoring in GPS Denied Environments. UAVPayloadTAQ - 2022	Customer Needs - UAVPayload-TAQ - EGH455-2022v1
RD/30	UAVPAYG19-FD-ED-01	Enclosure Design Final Design Report
RD/31	UAVPAYG19-AQS-FD-01	Air Quality Sensor Final Design Report
RD/32	UAVPAYG19-TAIP-FD-01	Target Acquisition and Image Processing Final Report
RD/34	UAVPAYG19-WVI-02	Web Visualization Interface Final Design Report
RD/22	UAVPAYG-19-ST-TR-01	Sampling Tube Test Report

2.2 Non-QUT Documents

Table 2 – Non-QUT Documents

Doc-1	DF15RSMG 360 Degree Motor	DFROBOT, “DF15RSMG 360 Degree Motor (20kg)”, n.d. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SER0035_Web.pdf. [Accessed 14/08/2022]
Doc-2	NASA's Perseverance Rover Sample Tubes	National Aeronautics and Space Administration, “A Martian Roundtrip: NASA's Perseverance Rover Sample Tubes”, 2020. [Online]. Available: https://mars.nasa.gov/news/8822/a-martian-roundtrip-nasas-perseverance-rover-sample-tubes/. [Accessed 22/08/2022]
Doc-3	STYROFOAM™ HD 300F-GV-X Extruded Polystyrene Foam – Technical Specification	The Dow Chemical Company, “STYROFOAM™ HD 300F-GV-X Extruded Polystyrene Foam - Sales Specification”, 2014, p. 2. [Online]. Available: https://extrudedpolystyrene.com.au/wp-content/uploads/2014/02/SS_STYROFOAM%E2%84%A2-HD-300F-GV-X-Extruded-Polystyrene-Foam.pdf. [Accessed 20/08/2022]

3 Subsystem Introduction

The Sampling Tube subsystem provides the capability for High-Level Objection HLO-M-6 to be met for the client. This subsystem is designed to be 3D-printed and make use of a singular motor that is provided as specified in RD/1. The ST subsystem is positioned on the underside of the payload enclosure, connected by a 3-pin cable. The design of the ST is inspired by NASA's Perseverance sampling tubes seen in Doc-2. To meet the client's requirements, the sampling tube protrude into the simulated soil with a diameter of 10mm and depth of 10mm, outlined in the subsystem requirements in section 3.1.

To allow this subsystem to work, it has been integrated with various other subsystems, outlined in section 4. In general, the sampling tube subsystem is attached directly to the body of the enclosure subsystem and has electrical connections to the Raspberry Pi via GPIO pins passing through the Air Quality (AQ) subsystem and 5V electrical adapter. The servo motor can be controlled by a remote user-interface in the Web Visualization Interface (WVI) subsystem, or manually by directly running scripts on the Pi.

An industry standard design process has been followed throughout the development of this subsystem to help make efficient use of available materials to reduce the environmental impact of the design. This has included multiple small-scale iterations and 3D-print tests before scaling up and attaching to the enclosure subsystem and integrating with the other subsystems.

3.1 Subsystem Requirements

Table 3 – Subsystem System Requirements

Requirement	Description	Verification
REQ-M-01	The UAVPayloadTAQ shall remain under the maximum weight of 320 g and comply with an IP41 rating. The air quality sensors must be exposed to the environment to allow for accurate reading.	Demonstration/ Measured
REQ-M-08	The payload shall include a sampling tube design to collect a simulated soil sample. The payload system must protrude or push into the simulated soil. A mark must be left on the simulated soil (10mm deep, 10mm diameter hole), to ensure the sampling tube has made contact with the soil.	Demonstrated
REQ-M-09	The payload shall activate the sampling tube mechanism to collect a simulated soil sample only after the UAV has landed on a designated Aruco marker. Once the soil is sampled the sampling tube must retract to its original position.	Demonstrated
REQ-M-14	Developed solution shall conform to the systems engineering approach.	Demonstrated

4 Subsystem Architecture

For the ST subsystem to operate, several interfaces have been made available with other subsystems and components. The following section will cover each of these interfaces and the process flow behind them. Figure 1 represents how the Sampling Tube subsystem has been integrated within the overall system.

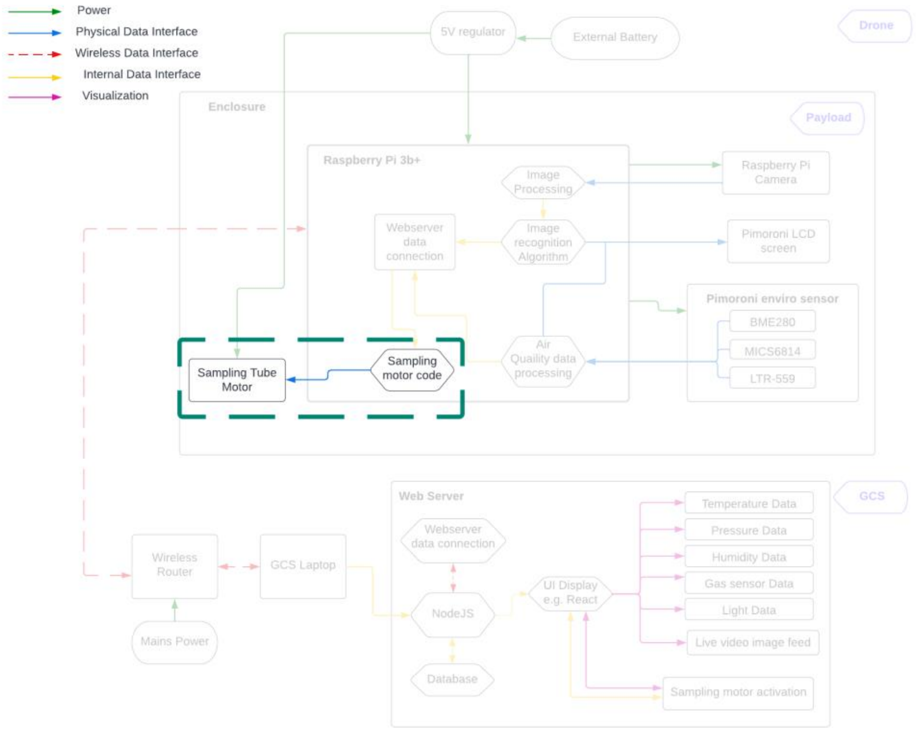


Figure 1 – Systems Architecture Diagram

4.1 Interfaces

Table 4 – List of Applicable Interfaces for Sampling Tube

Interface	Component/System	Interface Description
I1	Enclosure	The Sample Tube is connected to the enclosure subsystem through a physical mount protruding from the bottom of the enclosure. This mount attaches to the servo motor through the provided screw holes. The sampling tube is then fitted onto the same mounting holes.
I2	Electrical Adapter	A 3-pin lead is attached to the body of the servo motor, which is directly connected to the 5V regulator power adapter. This adapter provides a 2-pin lead for pulse position modulation (PPM) and grounding.
I3	Enviro+ Sensor	The 2-pin lead from the electrical adapter is connected to the Enviro+ Sensor subsystem through the ground and GPIO #4 pins that are passed through to the Raspberry Pi.
I4	Raspberry Pi 3B+	The Raspberry Pi 3B+ receives the passed-through signal from the Enviro+ Sensor to be processed and controlled electronically.
I5	Web Interface	The Web Visualization Interface subsystem provides controls to the user over Wi-Fi to allow the software on the Raspberry Pi to control the PPM signal on GPIO pin #4.

4.2 Interface Diagram

A detailed interface connection diagram is shown in Figure 2. Using the information in Table 4 – List of Applicable Interfaces for Sampling Tube, all the connections are provided. As the PPM cable passes through the power adapter and the Enviro+ Sensor to access GPIO pin #4, these elements and subsystems have also been included in the interface diagram.

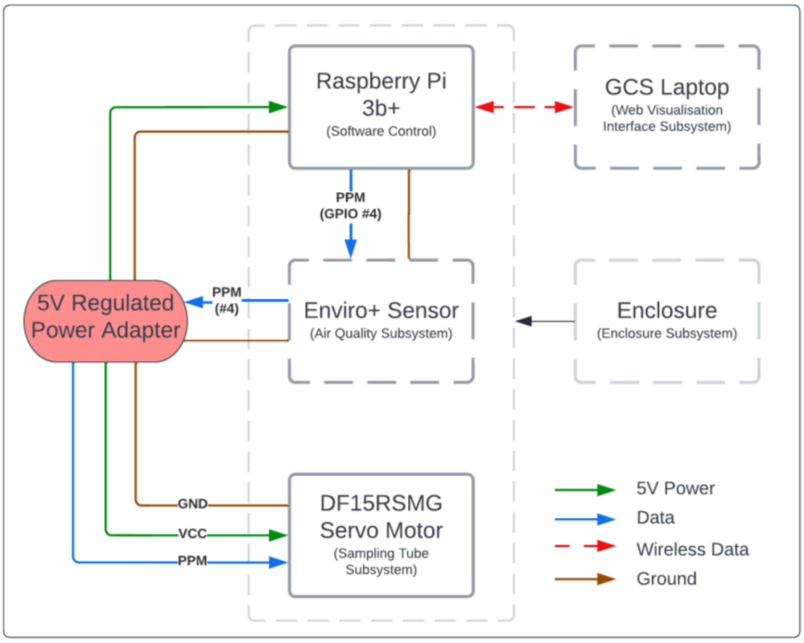


Figure 2 – Sampling Tube Interface Diagram

5 Design

As supplied by the project supervisor, the hardware used in the construction of the payload is: a Raspberry Pi 3b+, Raspberry Pi Camera, Pimoroni Enviro+ Sensor and a 360-degree servo motor. The following section will cover all related design elements that were needed to develop the Sampling Tube subsystem.

5.1 Hardware Specifications

The following is the list of hardware specifications that are being interfaced with for the Sampling Tube subsystem. Relative information that is needed for the interfacing to occur will be provided in this section.

5.1.1 DF15RSMG 360 Degree Motor

The supplied servo motor is the DF15RSMG 360 Degree Motor, with the technical specifications provided in Doc-1. This motor is a part of the Sampling Tube Subsystem and has a stall torque of 19.3kg/cm @ 7.4V. The servo motor provides rotation power to the sampling tube to allow it to extend and retract, and also providing enough power to cut into the simulated soil to achieve REQ-M-08.

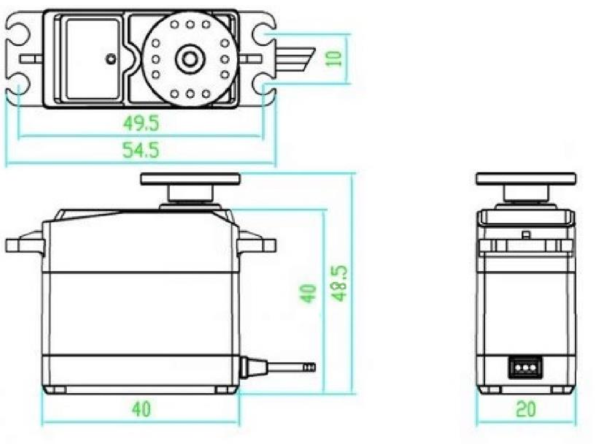


Figure 3 - 360 Degree Motor Scientific Diagram

Table 5 - Key Specifications of 360 Degree Motor

Length	**40mm**
Width	20mm
Height	40mm
Bolt Size	M4
Torque	15.1kg*cm
VCC	5-7.4V
PPM Voltage	3-5V
PPM Resolution	2us
PPM Width	400us-2550us

5.1.2 Raspberry Pi 3B+

The Raspberry Pi 3B+, being used as the payload control computer, has a 40-pin connector that the Enviro+ Sensor is completely attached to. Therefore, the pass-through pin on the Enviro+ Sensor is required so that the Raspberry Pi can control the servo motor. Figure 4 shows all the pins that are provided by the Raspberry Pi that the Enviro+ Sensor is connected to. The script that controls the PPM signal uses GPIO pin 4, identified as pin 7 in Figure 4. The Raspberry Pi is housed within the ED Subsystem, defined in RD/6.

Figure 4 - Raspberry Pi 3B+ GPIO pins

5.1.3 Pimoroni Enviro+ Sensor

The Pimoroni Enviro+ Sensor device contains several pass-through pins on the front of the device. This device is used as part of the Air Quality Subsystem, shown in RD/7. The relevant pins '#4' and 'GND' are connected to the 2-pin cable that comes from the provided 5V regulator power adapter. Pin '#4' is connected to the PPM signal cable and the 'GND' pin is connected to the ground cable. This pin can be seen on the top-left pinout in Figure 5.

Figure 5 - Pimoroni Enviro+ Sensor

5.2 Sampling Tube design

The following subsection covers the physical final design of the ST subsystem. These parts were designed in Autodesk's Fusion 360 application. The general idea was to use a power-screw design where threads are used to move the drilling part up and down so that a soil sample can be taken. The main restriction for the design was that it could not be more than 6.4cm in height, starting from the mounting position at the bottom off the Enclosure Subsystem. The following drawings in Figure 6 and Figure 7 are the complete assembly of the parts that will be outlined in this subsection. Figure 6 shows the isometric view from the outside of the assembly, while Figure 7 shows a cross-section of the complete assembly where you can see the internal driver, external shaft, and internal shaft. Generally, to meet REQ-M-01, the design is between 2-3mm thick, but may be slightly thicker in some areas. Full technical drawings can be found in Appendix 7.4.

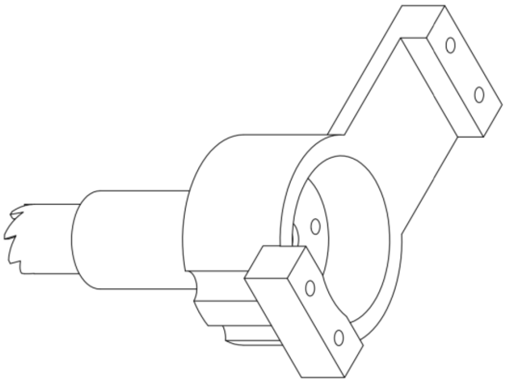


Figure 6 – Isometric view of assembly

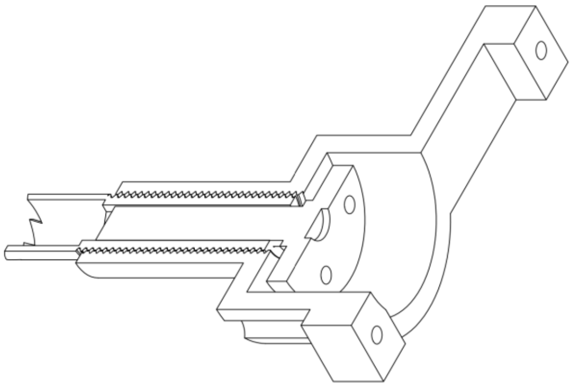


Figure 7 – Cross-sectional isometric view of assembly

5.2.1 ST Enclosure

The external shaft is designed to be a fixed body that attach to the mounts on the servo motor, which is attached to the ST Subsystem mounts on the ED Subsystem. The ST enclosure is used as the fixed body that allows the drive shaft and cutting tool to be rotated up and down in a linear direction. To ensure that the ST enclosure doesn't interfere with the servo motor or internal parts, it was designed to fit around the servo motor output interface, giving a wider body in the design. To ensure that the bolts can make it through, some surface area of the external shaft was carved out, as it can be seen at the bottom right of Figure 8. Full technical drawings can be found in Appendix 7.3. During the testing of the ST subsystem outlined in RD/22, some changes were made to the preliminary design in RD/9, resulting in the indented cup area to be made 3mm wider and 2mm deeper, extending the length of the ST enclosure by 2mm. These changes were necessary to ensure that the ST subsystem could achieve the outlined requirements.

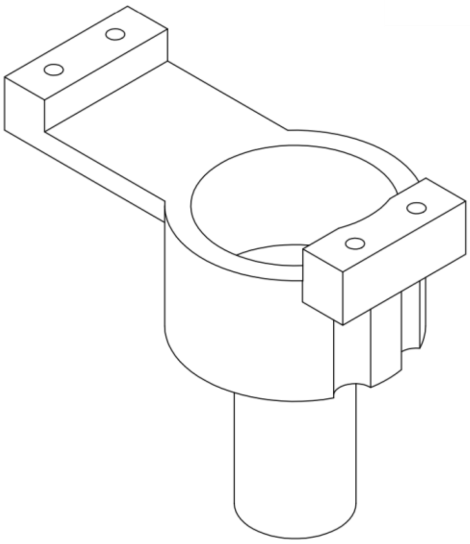


Figure 8 – Isometric view of external shaft final design

5.2.2 Cutting Tool

The cutting tool is designed to move up and down the ST Enclosure, while also containing a hole saw at the end. As the cutting tool is rotating, the cutting action of the hole saw helps cut into the Styrofoam, allowing for a soil sample to be taken achieving REQ-M-08. Figure 9 shows the isometric view of this part. As it can be seen, the thread covers the entire length of the part that fits within the ST Enclosure, leaving the external cutting part with no threads for strength. This is to ensure that the part does not move all the way up the shaft to prevent damage internally to the ST Enclosure. Also, a hexagonal slot is shown on the left, which is where the drive shaft can freely mover linearly while also providing rotational power to the part. Full technical drawings can be found in Appendix 7.2. As extra clearance was determined due to a change in the Enclosure Subsystem, this part was slightly modified to add an extra 9mm of length externally so that it can reach the simulated soil to meet REQ-M-08. As per the ST test reports in RD/22, the cutting tool could cut into the expanded polystyrene.

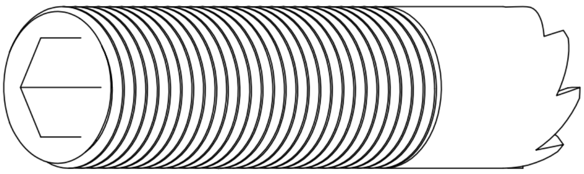


Figure 9 – Isometric view of the internal shaft with hole saw cutting tip

5.2.3 Drive Shaft

To drive the cutting tool and move it up and down within the ST enclosure, the drive shaft connects the cutting tool to a nylon motor output mount connected to the output of the servo motor. During assembly, this part connects to the nylon motor output mount with four wide-thread screws that are 2mm wide. To account for the screws, space is provided within the ST enclosure so that the drive shaft can move with minimal friction. Figure 10 contains an isometric view of the part showing the plate that connects to the nylon motor output mount and the hexagonal shaft that connects into the cutting tool. Full technical drawings can be found in Appendix 7.1. During testing a slight modification to the design was necessary as the screw that mounts the nylon motor output mount creates a 1mm gap. This can be found in the ST test report RD/22. To solve this, the base of the drive shaft was thickened by 1mm, and 1mm high circle was added around the base of the hexagonal shaft for added strength.

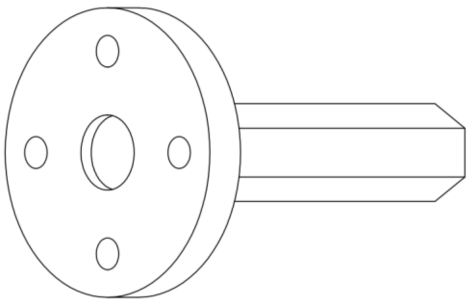


Figure 10 – Isometric view of internal driver

5.2.4 Thread design

A thread design was developed with the smallest pitch possible for a resin 3D printer that would be strong enough to hold some sort of power force. A pitch of 1mm was chosen for the thread using an ISO Metric standard thread design of M10 for a 10mm diameter. To ensure that enough force would be present to penetrate the expanded polystyrene, a cutting tool head was developed to conduct a real-world test. A test was conducted with the assistance of Dr. Veronica Gray to determine that extruded polystyrene using the tool head had linear compaction force requirement of 5N to crush the material. This test is outlined in RD/22. Using this information and measurements from the designed threads and tool head we can estimate the torque required to operate the device.

$$\tan(\theta) = \frac{height}{circumference}$$

$$\theta = \tan^{-1}\frac{1}{2 \times \pi \times (\frac{9.2685}{2})}$$

$$\theta = 1.9669°$$

With 5N of linear force measured, we can turn that into rotation force by flattening a single rotation knowing the angle and diameter of the thread.

$$\tan(\theta) = \frac{vertical}{horizontal}$$

$$F = \frac{5}{\tan(1.9669)}$$

$$F = 145.59N$$

As this force is applied at a radius of 4.63425mm, we can determine the torque from compression.

$$T_c = F \times d$$

$$T_c = 145.59 \times 4.63425$$

$$T_c = 0.6747Nm$$

To determine the cutting force a few parameters are needed. Assuming worst case scenario using high compressive strength extruded polystyrene from Dow Chemical. The shear force of the Styrofoam was chosen to be 450kPa from the technical specification sheet Doc-3. To first calculate the torque required, the cross-sectional area where the force is being applied.

$$J = \frac{\pi}{2}({r_{ext}}^4 - {r_{int}}^4)$$

$$J = \frac{\pi}{2}\left(\frac{0.01^4}{2} - \frac{0.008^4}{2}\right)$$

$$J = \frac{369\pi}{2 \times 10^{12}}$$

The shear torque can then be found by using the shear stress formula and the radius of the cutting tool.

$$\gamma = \frac{T_s \times r}{J}$$

$$T_s = \frac{450 \times 10^3 \times \frac{369\pi}{2 \times 10^{12}}}{0.005}$$

$$T_s = 0.0522Nm$$

The total torque required is shown below.

$$T = T_s + T_C$$
$$T = 0.6747 + 0.0522$$
$$T = 0.7269$$

As the total torque output of the motor is 1.481Nm, the system was expected to function as expected with plenty of margin for error. In testing of the ST subsystem in RD/22, it was confirmed that the torque available could cut into the expanded polystyrene.

5.3 Software Design

To operate the motor, a python script was developed to control the PPM signal out of GPIO pin #4 of the Raspberry Pi. Although in the documentation for the servo motor it says to use PPM, pulse-width modulation (PWM) can be used. A python library known as "GPIO" was imported to make the implementation as simple as possible. Using a frequency of 50Hz to communicate with the servo motor, the duty cycle can be evaluated with the following equation when the upper limit of 2500us and lower limit of 500us are defined using a minimum of 400us pulse-width.

$$\% = \frac{\left(400 + \frac{2500 - 500}{2} + time_{ns}\right)}{\frac{1}{50} \times (1 \times 10^6)} \times 100$$

The following values were computed as control states for the motor using a 500us drift zone around the nominal value and are listed in Table 1. The further you move away from the 'Off' position of 7% duty-cycle, the faster the servo motor will turn. The direction was determined in the ST test report RD/22 while completing the software test.

Table 6 – Duty-cycle percentages to control servo motor

Control	State
Reverse/Up	(fast) 2 → 6.5 (slow)
Off	7
Forward/Down	(slow) 7.5 → 12 (fast)

The software implementation is a modification of test motor code presented in the preliminary report RD/9. The new code works by being implemented as a class that is initialised. The class has a method that will take an integer value to change the motors state. A value greater than zero will extend the cutting tool, smaller than zero to retract the cutting tool, or zero to ensure it is not moving. This command is received from the WVI Subsystem, in which is converted to a PWM signal to send to the servo motor. The code implementation can be found in Appendix 7.5.

Additional code was developed so that the cutting tool can be moved into position once it was mounted. It allows the servo motor to be pulsed by running a python script from command line. This was necessary to fine tune the motor and make deployment faster. The code can be found in Appendix 7.6.

6 Conclusion

The Sampling Tube Subsystem final design has been created to meet all high level and system requirements supplied by the project supervisor. The ST Subsystem has interactions with the Air Quality System, Enclosure Design, and Web Visualisation Interface Subsystems, while also having contact with components such as the Raspberry Pi and Power Adapter. By using the servo motor, the requirements provided by RD/1 and RD/2 are sufficiently met. The follow on from this report will be to complete the acceptance tests to ensure that the ST Subsystem correctly integrates with all the other subsystems and that all requirements are met. Table 7 shows the requirements related to ST final design and how they have been met.

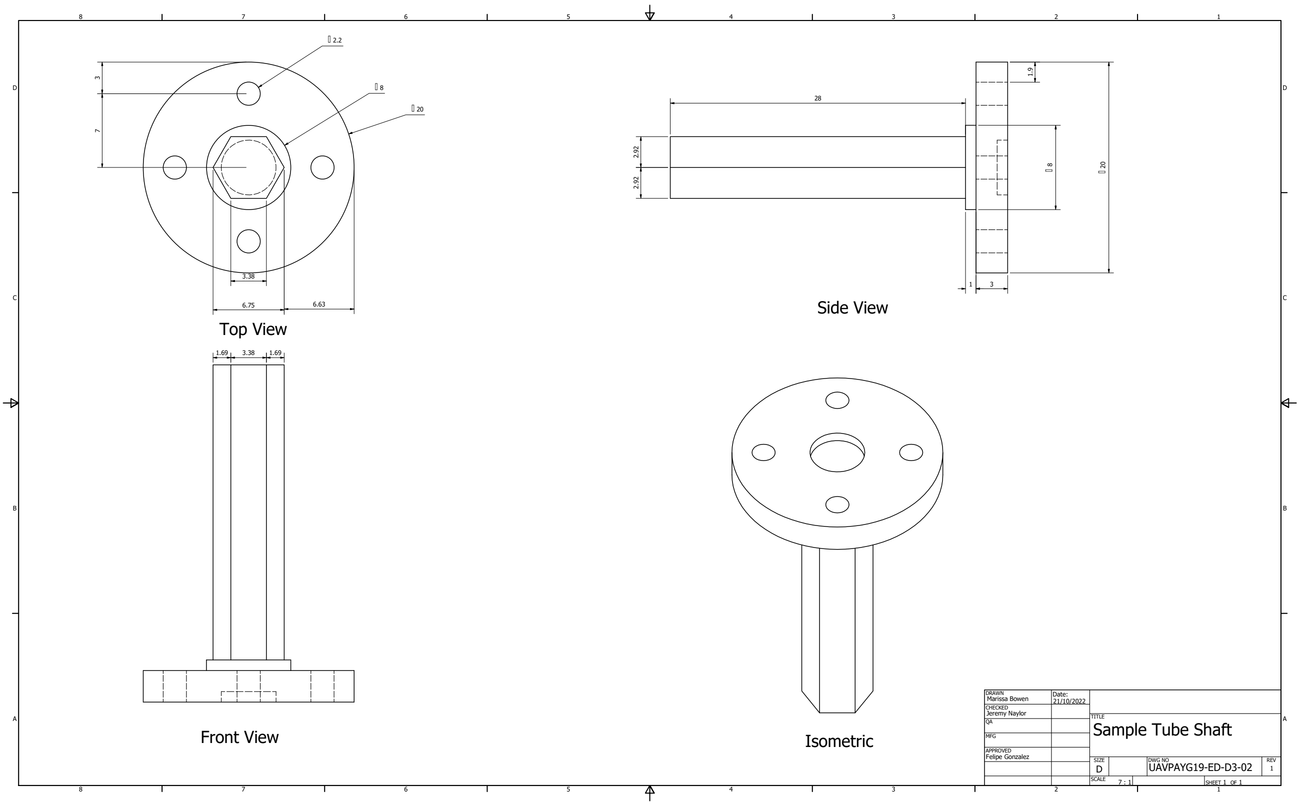
Table 7 – Requirements met by final design

Requirement	**Description**	**Requirements Met**
REQ-M-01	The UAVPayloadTAQ shall remain under the maximum weight of 320 g and comply with an IP41 rating. The air quality sensors must be exposed to the environment to allow for accurate reading.	**Met**:- The final sampling tube design has been estimated to weigh 9g after 3D printing excluding the servo motor. This will help ensure weight is distributed for the other subsystems.
REQ-M-08	The payload shall include a sampling tube design to collect a simulated soil sample. The payload system must protrude or push into the simulated soil. A mark must be left on the simulated soil (10mm deep, 10mm diameter hole), to ensure the sampling tube has made contact with the soil.	**Met**:- The final design of the sampling tube mechanism allows for the payload to protrude into simulated soil with a diameter of 10mm and a depth of 10mm. The system has been designed to have extra length if needed when deployed on a drone.
REQ-M-09	The payload shall activate the sampling tube mechanism to collect a simulated soil sample only after the UAV has landed on a designated Aruco marker. Once the soil is sampled the sampling tube must retract to its original position.	**Met**:- The final design of the sampling tube mechanism allows for the sampling tube to extend and retract when an activation command is sent from the WVI Subsystem.
REQ-M-10	Preliminary designs shall be completed by week 7.	**Met**:- This document has been completed and submitted in week 6.
REQ-M-14	Developed solution shall conform to the systems engineering approach.	**Met**:- All choices that have been made for the sampling tube design have been derived from the system requirements and design has followed the “V” model approach. This includes the preliminary design RD/9, ST test reports RD/22, final design, and other documents provided.

7 Appendix

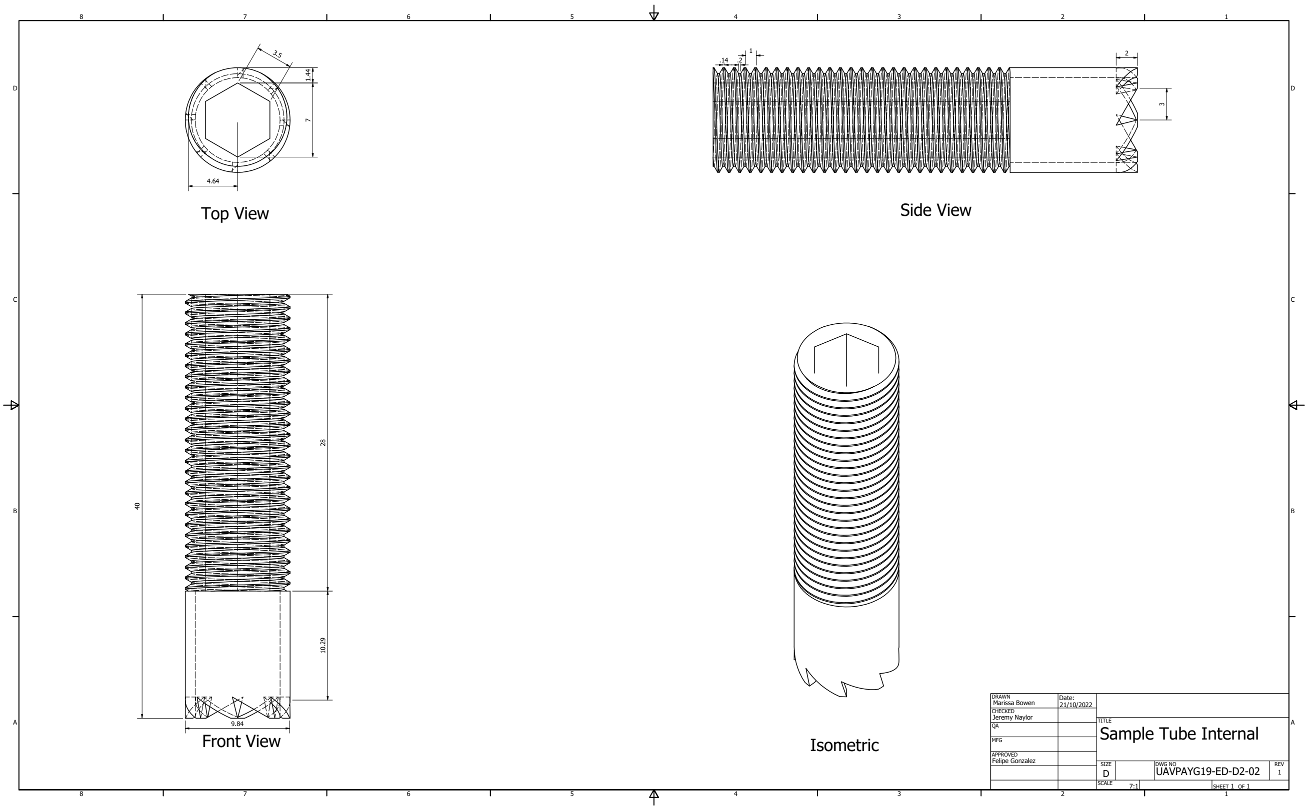
7.1 Drive Shaft

The following is a technical drawing of the shaft that drives the cutting tool up and down the shaft. This part connects to the nylon adapter at the output of the servo motor.



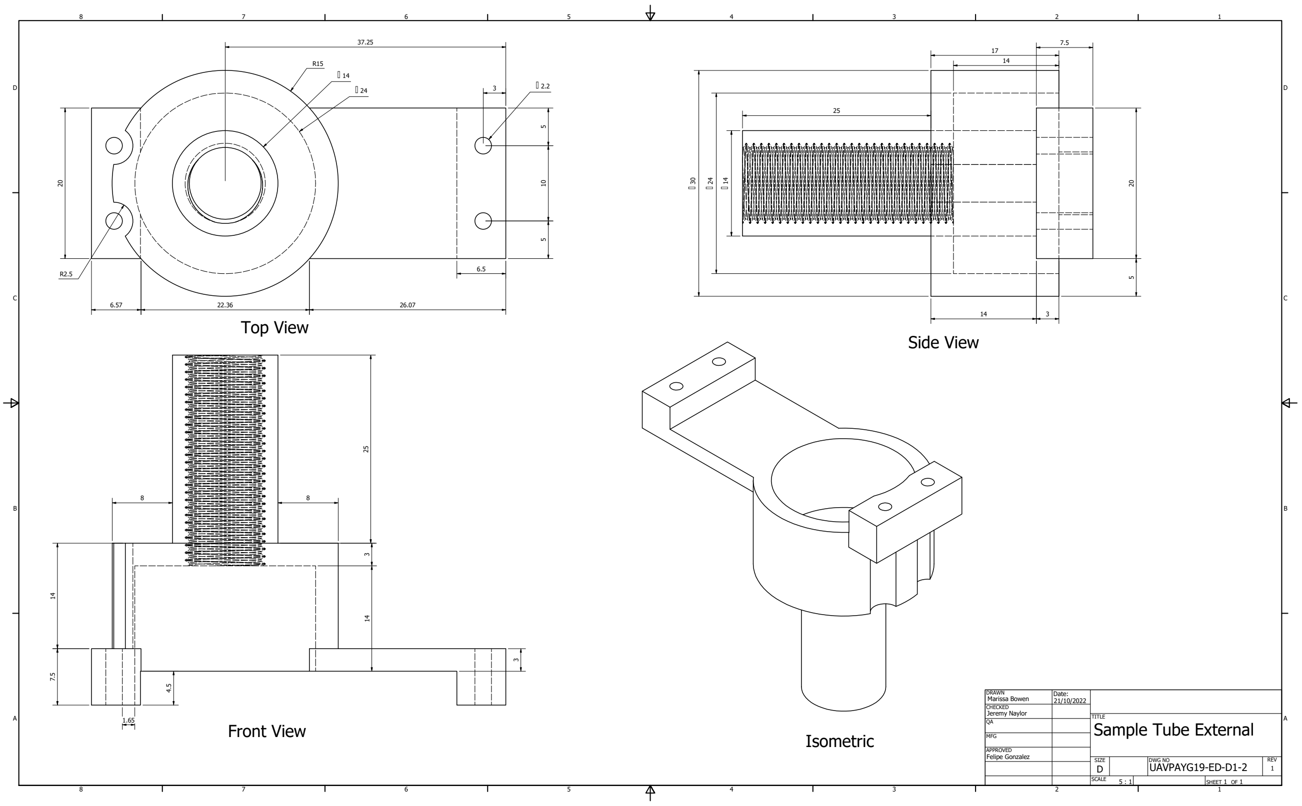
7.2 Cutting Tool

The following is a technical drawing of the cutting tool that moves up and down when powered by the motor. This part contains the cutting tool that will be used to take a sample and a hexagonal interior to drive it.



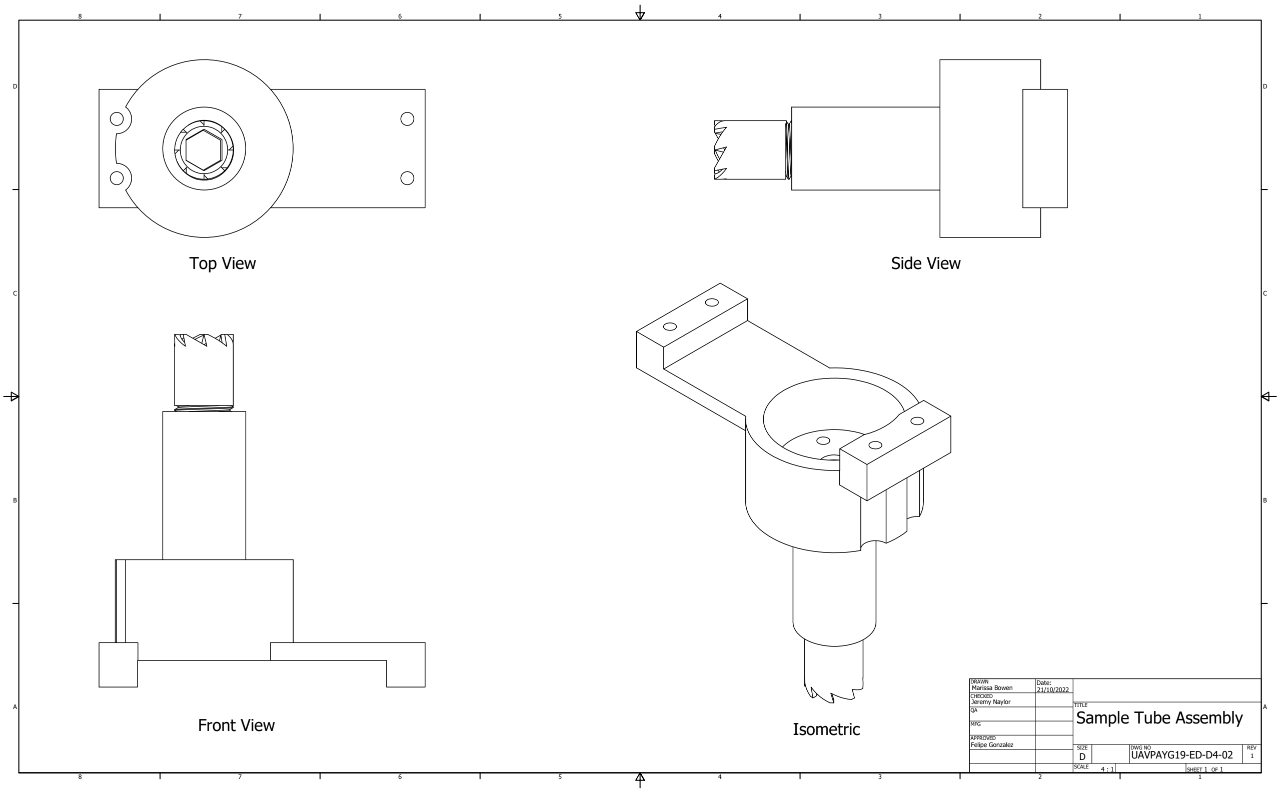
7.3 ST Enclosure

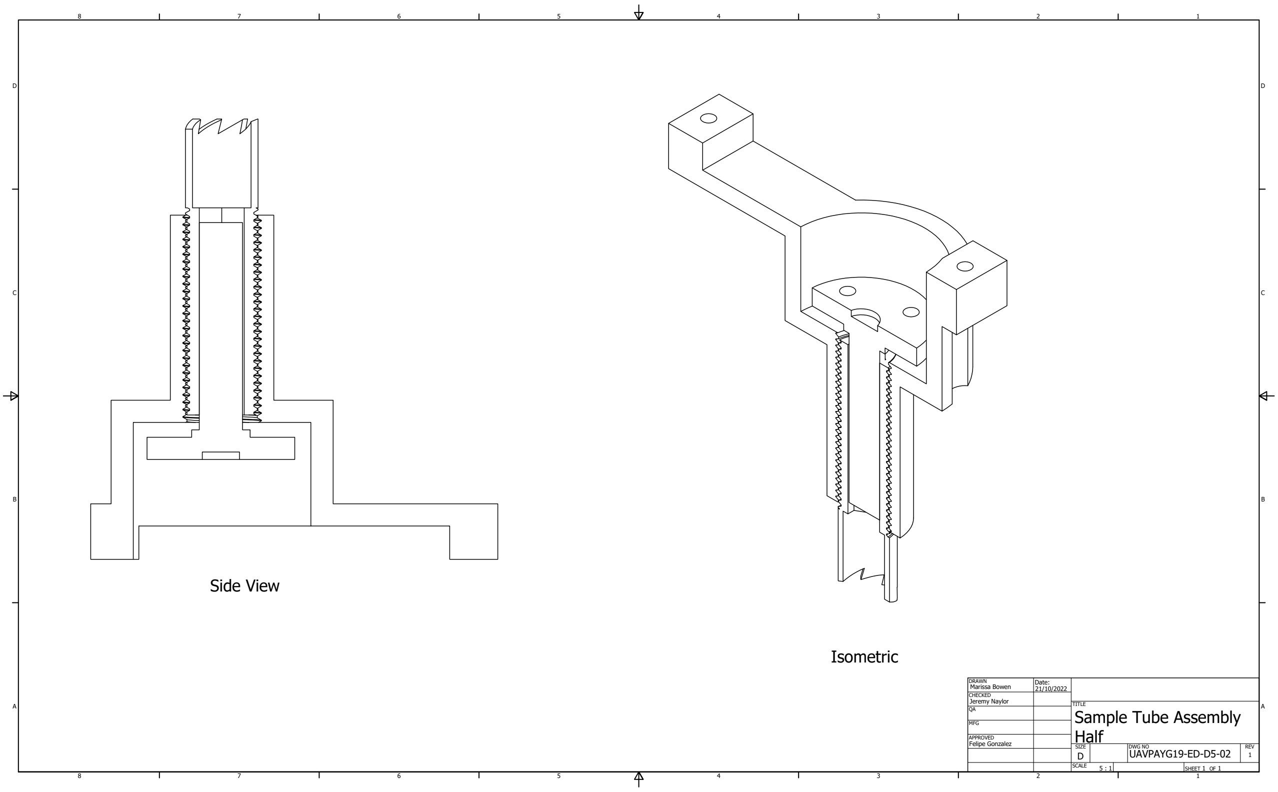
The following is a technical drawing of the ST Enclosure that is fixed to the mounting points of the servo motor. This part encloses all the other parts and keeps them attached to the servo motor.



7.4 Complete Assembly

The following two technical drawings are a completed assembly of the parts. An intersection drawing is also provided to see the internals of the completed assembly.





7.5 Servo Motor Code

The following code was used to drive the servo motor within the python environment. It was deployed as a class along-side other web-integrated code.

```python
import RPi.GPIO as GPIO
import time


# speed% = (<time_ns>-1.5)*100
# speed setting: <time_ns>/((1/50)*1000*1000)*100
# min value: 2 = Reverse/Up (max speed) -> 6.5
# max value: 12 = Forward/Down (max speed) -> 7.5
NOMINAL=7 # the 'zero' PWM %age
RANGE=5   # maximum variation above/below NOMINAL
MODIFIER=3# the value to change by, must be smaller than RANGE
#TIME=5    # time in seconds to keep motor on
CYCLES=18 # number of cycles to perform

class servo:
    def __init__(self):
        self.p = None


    def start(self):
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(4, GPIO.OUT, initial=False)
        self.p = GPIO.PWM(4,50)
        self.p.start(NOMINAL)

    # 0 = off, 1 = down, -1 = up
    def motor_change(self, mode):
        p = self.p
        if mode > 0:
            # Extend tube down
            #p.ChangeDutyCycle(NOMINAL+MODIFIER)
            #time.sleep(TIME)
            for x in range(CYCLES):
                p.ChangeDutyCycle(NOMINAL+5)
                time.sleep(1)
                p.ChangeDutyCycle(NOMINAL-1)
                time.sleep(0.25)
                p.ChangeDutyCycle(NOMINAL)
                time.sleep(0.25)
        elif mode < 0:
            # Bring tube up
            #p.ChangeDutyCycle(NOMINAL-MODIFIER)
            #time.sleep(TIME)
            for x in range(CYCLES):
                p.ChangeDutyCycle(NOMINAL-5)
                time.sleep(1)
                p.ChangeDutyCycle(NOMINAL+1)
                time.sleep(0.25)
                p.ChangeDutyCycle(NOMINAL)
                time.sleep(0.25)
        p.ChangeDutyCycle(NOMINAL)

    def cleanUp(self):
        GPIO.cleanup()
```

7.6 Servo Pulse Motor Code

The following code was used to pulse the motor so that it could be fine-tuned into the correct position when all components were mounted together.

```python
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.OUT, initial=False)
p = GPIO.PWM(4,50)

# speed% = (<time_ns>-1.5)*100
# speed setting: <time_ns>/((1/50)*1000*1000)*100
# min value: 2 = Reverse/Up (max speed) -> 6.5
# max value: 12 = Forward/Down (max speed) -> 7.5
NOMINAL=7 # the 'zero' PWM %age
RANGE=5   # maximum variation above/below NOMINAL

p.start(7)

print("To exit: Ctrl+C")
while(True):
    userinput = int(input("Enter Number Of Rotations. Max +/-2. Eg: '1' or '-2'\n"))
    if not isinstance(userinput, int) or userinput > 2 or userinput < -2:
        print("Invalid Input")
        exit()

    if userinput > 0:
        for x in range(abs(userinput)):
            p.ChangeDutyCycle(NOMINAL+5)
            time.sleep(1)
            p.ChangeDutyCycle(NOMINAL-1)
            time.sleep(0.25)
            p.ChangeDutyCycle(NOMINAL)
            time.sleep(0.25)
    else:
        for x in range(abs(userinput)):
            p.ChangeDutyCycle(NOMINAL-5)
            time.sleep(1)
            p.ChangeDutyCycle(NOMINAL+1)
            time.sleep(0.25)
            p.ChangeDutyCycle(NOMINAL)
            time.sleep(0.25)

GPIO.cleanup()
```