

ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT REVISION

---

# A MICRO-PENETROMETER FOR SNOW AND SOIL STRUCTURAL ANALYSIS

---

## Team #12

CHENGHAO MO (cmo8@illinois.edu)

XING SHEN (xings2@illinois.edu)

ZHEYAN WU (cmeng10@illinois.edu)

CHENXIAN MENG (zheyaw2@illinois.edu)

April 18, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem and Solution Overview . . . . .	1
1.2	Visual Aid . . . . .	2
1.3	Highlevel-List . . . . .	2
<b>2</b>	<b>Design</b>	<b>3</b>
2.1	Block Diagram . . . . .	3
2.2	Mechanical Description . . . . .	4
2.3	Penetrate Subsystem . . . . .	5
2.3.1	Force Sensor . . . . .	5
2.3.2	Voltage Amplifier . . . . .	6
2.4	Moving Subsystem . . . . .	7
2.4.1	Motor Unit . . . . .	7
2.5	Compute Unit . . . . .	10
2.5.1	Microcontroller (ATK-DMF407) . . . . .	10
2.5.2	Bluetooth Module (ATK-MW579) . . . . .	13
2.6	Power System . . . . .	15
2.7	Software Subsystem . . . . .	16
2.8	Calculation . . . . .	17
2.8.1	Power Consumption . . . . .	17
2.8.2	ADC Conversion Period . . . . .	19
2.9	Tolerance Analysis . . . . .	20
<b>3</b>	<b>Cost and Schedule</b>	<b>23</b>
3.1	Cost Analysis . . . . .	23
3.1.1	Labor . . . . .	23
3.1.2	Parts . . . . .	23
3.1.3	Grand Total . . . . .	25
3.2	Schedule . . . . .	25
<b>4</b>	<b>Safety</b>	<b>30</b>
<b>5</b>	<b>Ethical Issue</b>	<b>31</b>
	<b>References</b>	<b>32</b>
	<b>Appendix A ADC + Bluetooth</b>	<b>33</b>
	<b>Appendix B Software + Motor</b>	<b>36</b>

# 1 Introduction

## 1.1 Problem and Solution Overview

In various fields such as agricultural production, archaeology, and disaster warning, the analysis of soil (sand, snow) is particularly important. The use of mechanical penetration can detect the bonding forces of soil (sand, snow) at different depths, and serve for subsequent data analysis. The existing analytical instruments have many problems, such as too large volume, limited application scenarios, insufficient operation convenience, and insufficient accuracy[1], [2]. We hope to have a portable, simple to operate, intuitive results, and adaptable to different scenarios for people to use.

We envisioned designing a machine that could drive a rod with a force sensor into soil (sand, snow) and record the force it received in real time. After integrating the data records, the penetration force characteristics of the sample at different depths could be visually reflected by color. Furthermore, this machine could also achieve multiple sampling and analysis of small-scale ground samples without moving the device by changing the horizontal position of the rod.

The specific implementation method is as follows: The controller controls the movement of the mechanical structure. During the movement, the data collected by the force sensor can be transmitted to the computer in real time by Bluetooth, and stored in the computer. The data analysis and result presentation can be carried out through the supporting software on the computer. After a single sampling, the rod will be reset to the initial height. If necessary, the horizontal position of the rod will be changed and the sample will be taken again. In addition, in the case of Bluetooth connection, users can also control the machine by using Bluetooth devices.

Our solution is to design an mechanical product which can test the force when drugging into the soil with pressure sensor and move xyz axis automatically. Also, we need to design the control parts to control the motor movement and read the transient data of sensor. The data can be displayed on computer with Bluetooth block.

## 1.2 Visual Aid

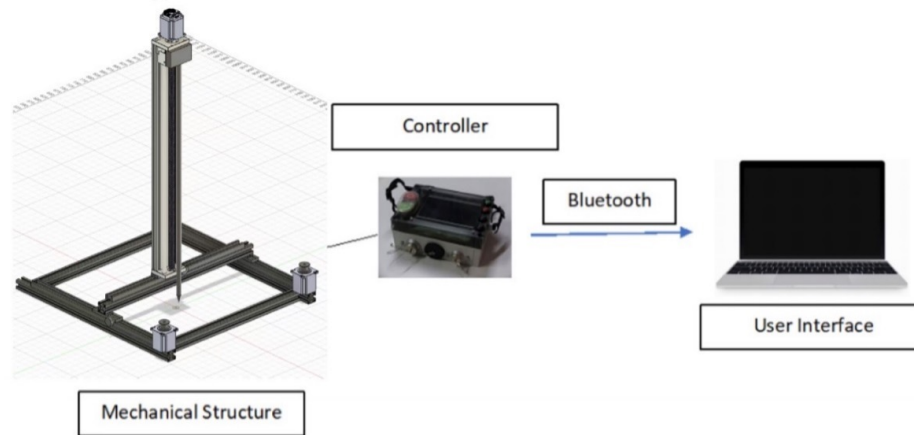


Figure 1: Visual Aid

## 1.3 Highlevel-List

1. The total mass of the machine shall not exceed 10KG.
2. The error of data acquisition shall not exceed 0.1mm.
3. The rod with a powerful sensor shall have a horizontal movement range of at least 40cm\*40cm.
4. The detector is capable of penetrating soil with a compactness of 2 MPa.

## 2 Design

Our Penetrometer Mainly consists of three systems, Mechanical System, Data Processing System and Control System. The Mechanical System includes the mechanical rod and force sensor, responsible for penetrating soil or snow and measuring the force encountered. The Data processing system ensures the data transfer, form force sensor to micro controller and form micro controller to the computer. The Control System includes the user interface on the computer and a STM32 MicroController. Below is our block diagram of the Penetrometer

### 2.1 Block Diagram

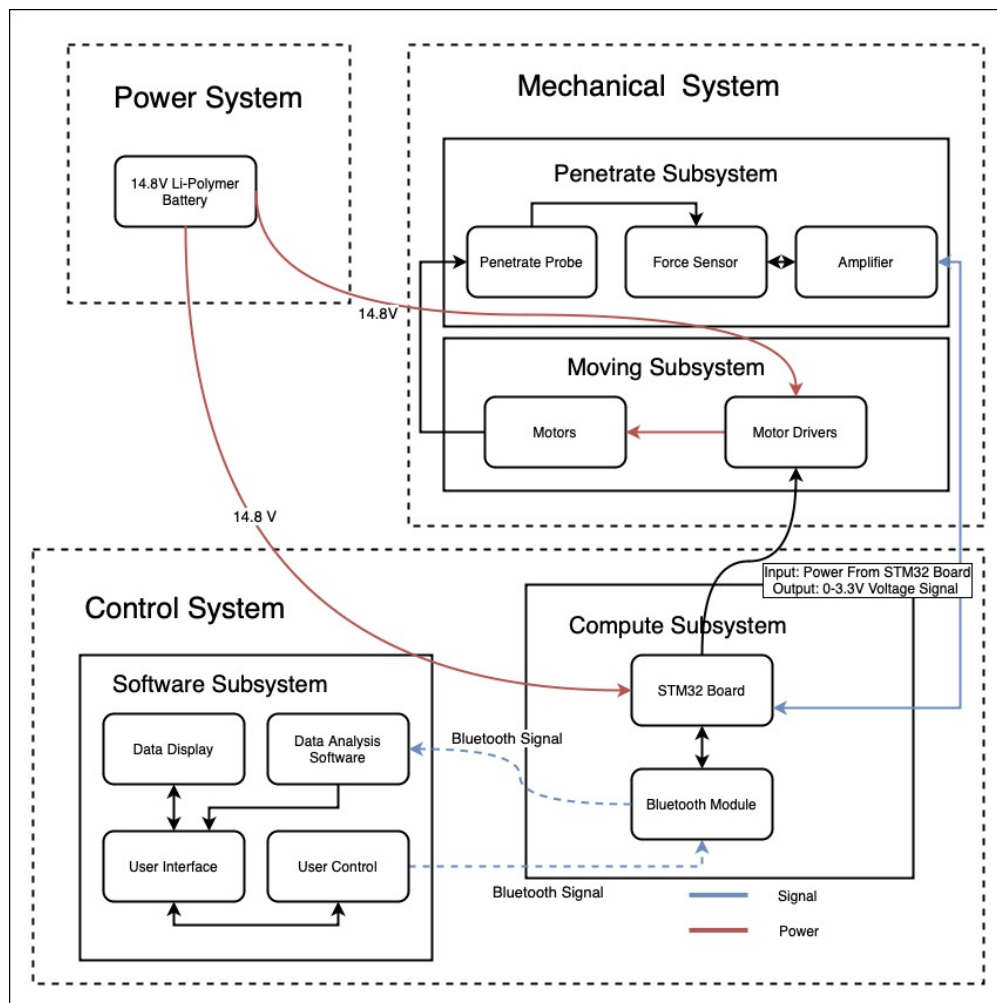


Figure 2: Block Diagram

## 2.2 Mechanical Description

The mechanical structure we designed allows the probe to move within a horizontal range of approximately 50mm \* 50mm, and can overcome a pressure of approximately 2MPa. In addition, the screws on the base can keep the instrument level on uneven ground. The mechanical sensor is placed at the tip of the detection rod.

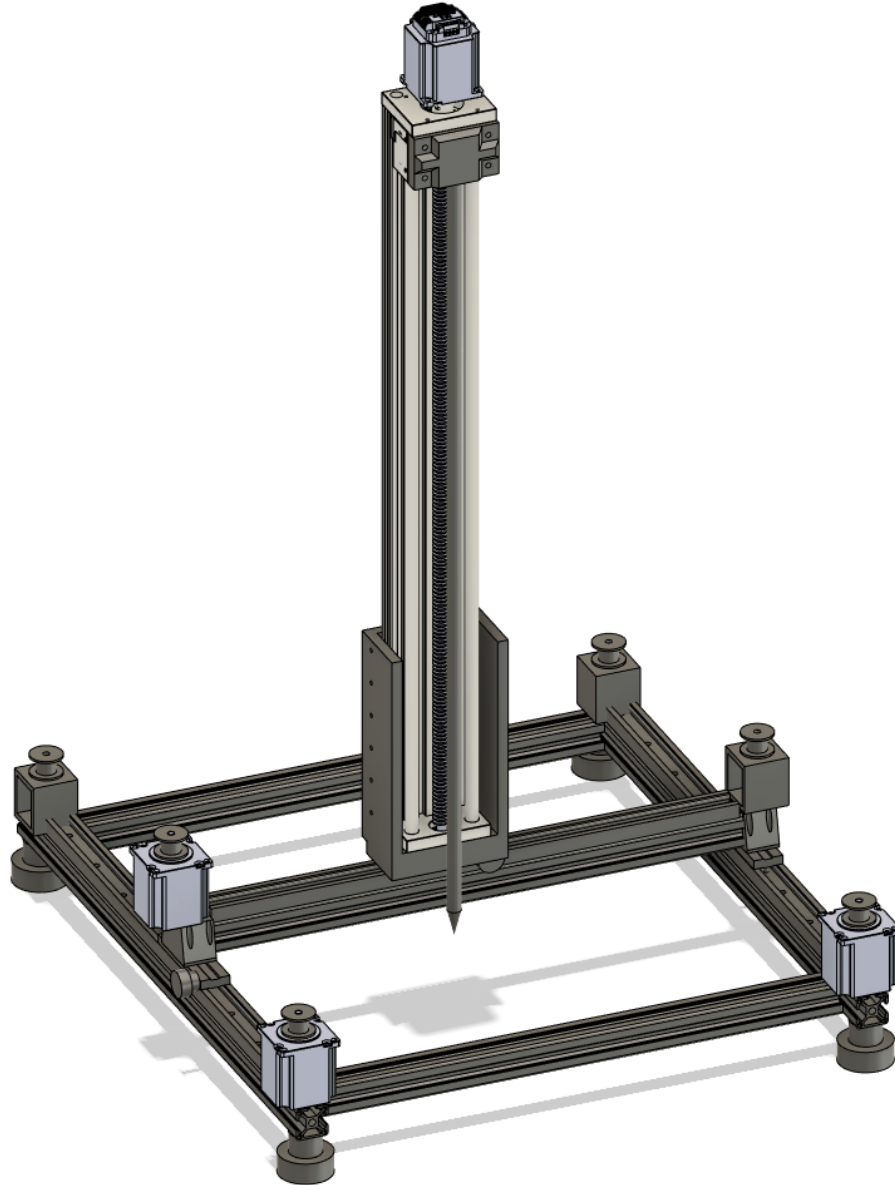


Figure 3: Mechanical Description

## 2.3 Penetrate Subsystem

The penetrate subsystem is composed of a penetrate probe and a force sensor, the force sensor should sending the magnitude of the force and send it to the micro controller.

### 2.3.1 Force Sensor

Input: : Physical force

Output: voltage change of signal

The force sensor has a built-in Force Sensitive Resistor (FSR) whose resistance varies according to the applied pressure. The force sensor here, with a 0-300N range and a sensitivity resolution of  $1.0 \pm 10\%$  mV/V, is a strain gauge-based transducer designed to convert applied force into a measurable electrical signal. When interfaced with an STM32F407IGT6 microcontroller, the sensor's excitation wires are connected to a stable 5V supply and ground, while the signal wires are linked to the microcontroller's ADC inputs to facilitate precise data acquisition. This setup enables the conversion of the mechanical force into digital values, which the STM32 controller can process and interpret. Calibration is imperative to correlate the ADC readings to actual force measurements accurately, considering the sensor's specified sensitivity and ensuring that the 10% tolerance is accounted for in the measurement system. This integration not only provides a reliable method to measure forces within its capacity but also allows for real-time monitoring and analysis when coupled with the microcontroller's computational capabilities, making it a suitable choice for a wide range of applications demanding force quantification.

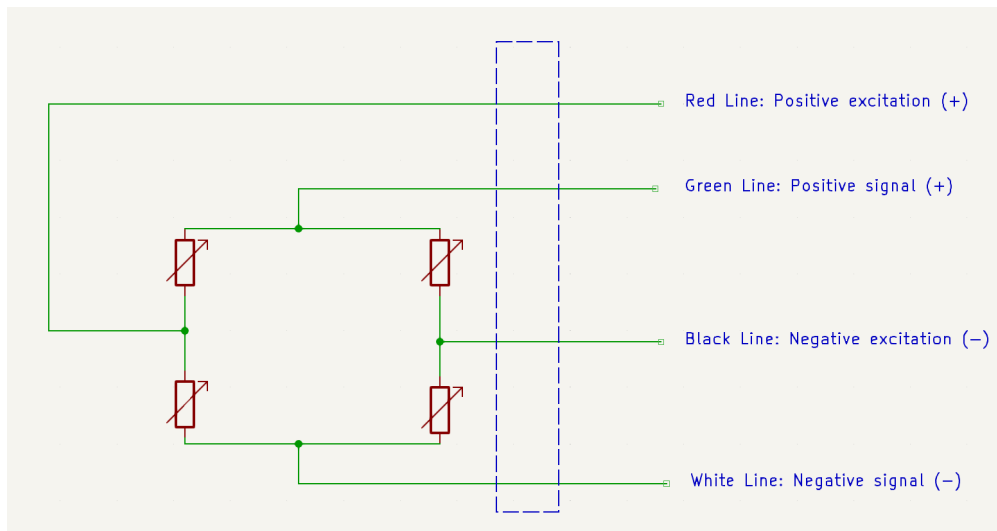


Figure 4: Force Sensor Schematic Diagram

Requirement	Verification Procedure
<p>The force sensor must display pressure levels on a screen, which vary with the applied force. The sensor must accurately measure pressure corresponding to the force applied by different displacements of an elastic spring with an accuracy of 10%.</p>	<ol style="list-style-type: none"> <li>1. Place block between Force Sensor and elastic spring to measure displacement vs. force. Measure the voltage at each level of force and then plot Force vs. Voltage with data obtained and testify the trend of the plotting.</li> <li>2. Put standard weights of different weights on the force sensor and check the force read from the sensor <math>F</math> with standard value (<math>S</math>). Use the formula for Error Rate, And the Error Rate should be less than 10%. <math display="block">\text{Error Rate} = \left  \frac{F - S}{S} \right  \times 100\%</math> </li> </ol>

Table 1: Force Sensor Requirements and Verification

### 2.3.2 Voltage Amplifier

Input: Positive Signal(Force Sensor), Negative Signal(Force Sensor)

Output: Amplified Voltage(+), Amplified Voltage(-), Power Supply for Sensor

The voltage amplifier in this part plays a critical role in enhancing the force sensor's output signal for proper digital interpretation by the STM32F407 microcontroller. Given that the force sensor's output is approximately 1mV, a considerably low signal, the voltage amplifier is selected for its capability to amplify this signal by a factor of 1000x. This degree of amplification is essential to boost the minute voltage to a measurable range for the microcontroller's ADC. Furthermore, the voltage amplifier is designed to operate on a 5V supply, conveniently sourced from one of the STM32's onboard voltage output ports. This integration ensures that the amplified signal is both within the optimal operating voltage for the microcontroller and that it does not require an additional external power source, thereby streamlining the design and reducing the complexity of the overall system.



Requirement	Verification Procedure
Amplification Factor of 500	Apply a constant input voltage of 0.015V to the voltage amplifier. Connect the amplifier's output to an oscilloscope. Adjust the on-board adjustable resistance until the average output voltage reads as 7.5V (0.015V × 500). Observe the stability and consistency of the output signal.
Tolerance within ±5%	After the amplification is set to 7.5V, measure the upper and lower voltage bounds on the oscilloscope. Ensure these values are within 5% of the 7.5V target (ranging from 7.125V to 7.875V).
Validation Across Different Inputs	Repeat the verification procedure for additional input voltages of 0.010V, 0.012V, and 0.016V. Expected amplified outputs will be 5.0V, 6.0V, and 8.0V respectively. Adjust the resistance as necessary for each and record the output voltages, checking each against the ±5% tolerance requirement.

Table 2: Verification Procedures for Amplifier Requirements

## 2.4 Moving Subsystem

### 2.4.1 Motor Unit

High-performance stepper motor driver (ATK-PD5050S)

The ATK-PD5050S module [3] is a versatile and rugged stepper motor driver that enables motors to dig down at a steady speed. The module operates from a 12 to 50V DC supply voltage range and has an output current of up to 5.0A, effectively driving a two-phase hybrid stepper motor at a constant speed. It boasts cutting-edge features including high-resolution microstepping for detailed motion control, load-based power optimisation for energy savings, and a low-resonance chopper algorithm that minimises vibration for improved motion accuracy. Integrating the ATK- PD5050S with an STM32F407IGT6 microcontroller typically involves connecting the drive's power input to the microcontroller's controlled voltage output, aligning the motor connections with the drive's terminals, and establishing communication for pulse, direction, and enable functions via optically isolated control signals. This integration will facilitate sophisticated motor control through the microcontroller's firmware, which generates PWM signals to control motor direction and to engage or disengage the motor as required by the application.

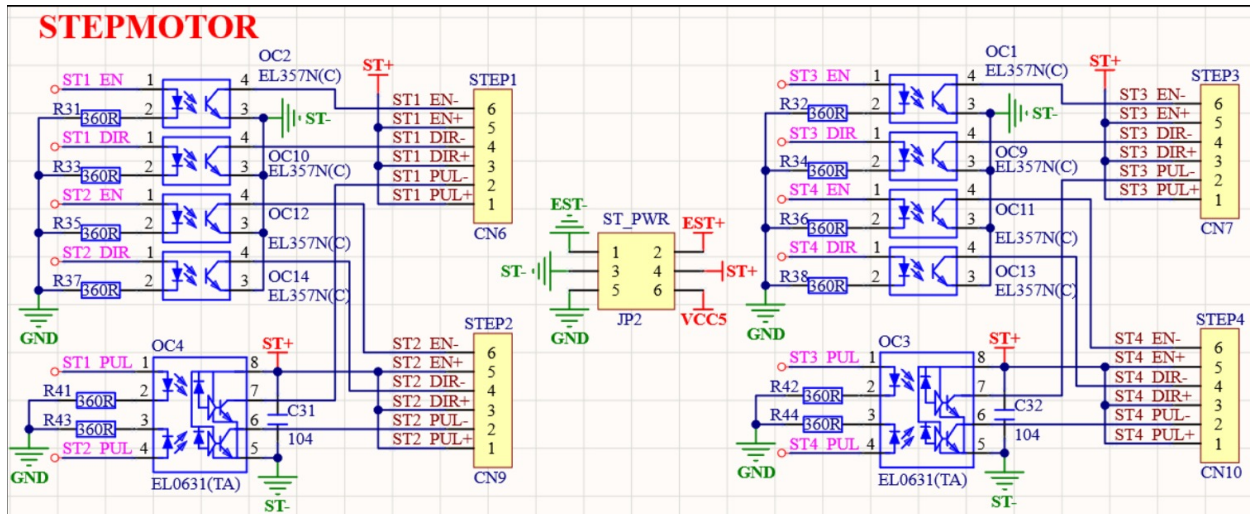


Figure 5: Motor Unit Schematic Diagram

Pin	Name	Description
ENA-	Enable negative	When the negative enable signal is effective, the internal logic signal is valid. When the signal is invalid, the internal logic is suspended, and the output is low impedance, high level, and the internal clock is stopped.
ENA+ (5V)	Enable positive (5V)	When the positive enable signal is effective, the internal logic signal is valid. When the signal is invalid, the internal clock is suspended, and the output is high impedance, high level.
DIR-	Direction negative	When the negative direction signal is effective, the internal logic determines the direction, and when the signal is invalid, the direction is determined externally.
DIR+ (5V)	Direction positive (5V)	When the positive direction signal is effective, the internal logic determines the direction.
PUL-	Pulse negative	When the negative pulse signal is effective, the internal logic is triggered, and when the signal is invalid, the external high level maintains the current state.
PUL+ (5V)	Pulse positive (5V)	When the positive pulse signal is effective, the internal logic is triggered, and the speed of the pulse is 200kHz.

Table 3: Motor Driver Pin Descriptions

ATK-PD5050S Interface	F407 Corresponding Interface
ENA-	PF15
ENA+	ST+
DIR-	PF14
DIR+	ST+
PUL-	PI5
PUL+	ST+

Table 4: Motor Driver Interface Pin Correspondences

Requirement	Verification Procedure
The motor unit will be able to control the rod at a velocity of 15 mm/s with error rate below 5%.	<ol style="list-style-type: none"> <li>1. Set parameters of speed and time for a motor</li> <li>2. Calculate the distance traveled by the motor under ideal conditions</li> <li>3. Measure the actual distance traveled by the motor</li> <li>4. Repeat step 1 4 for several times</li> <li>5. Repeat step 1 5 for every motors and get the result</li> </ol>

Table 5: Motor Driver Requirements and Verification

Since the motor control all the movements of the device, so it is important for us to test the accuracy of the motor. Based on the below equation, we can calculate the accuracy of motor.

$$Accuracy = \frac{\sum Speed_{Motor} * Diameter_{DrivingWheel} * time}{MovingDistance} / number_{test} \quad (1)$$

We hope that this accuracy should be large than 95%. On x and y axis, the large accuracy means the probe can be control to the right place. Since we want to detect several place in a 50cm \* 50cm square, we should keep every position accurate. Also, for the z axis, we also hope we can generate a diagram with force and distance, which means the motor should be accurate enough to represent its position.

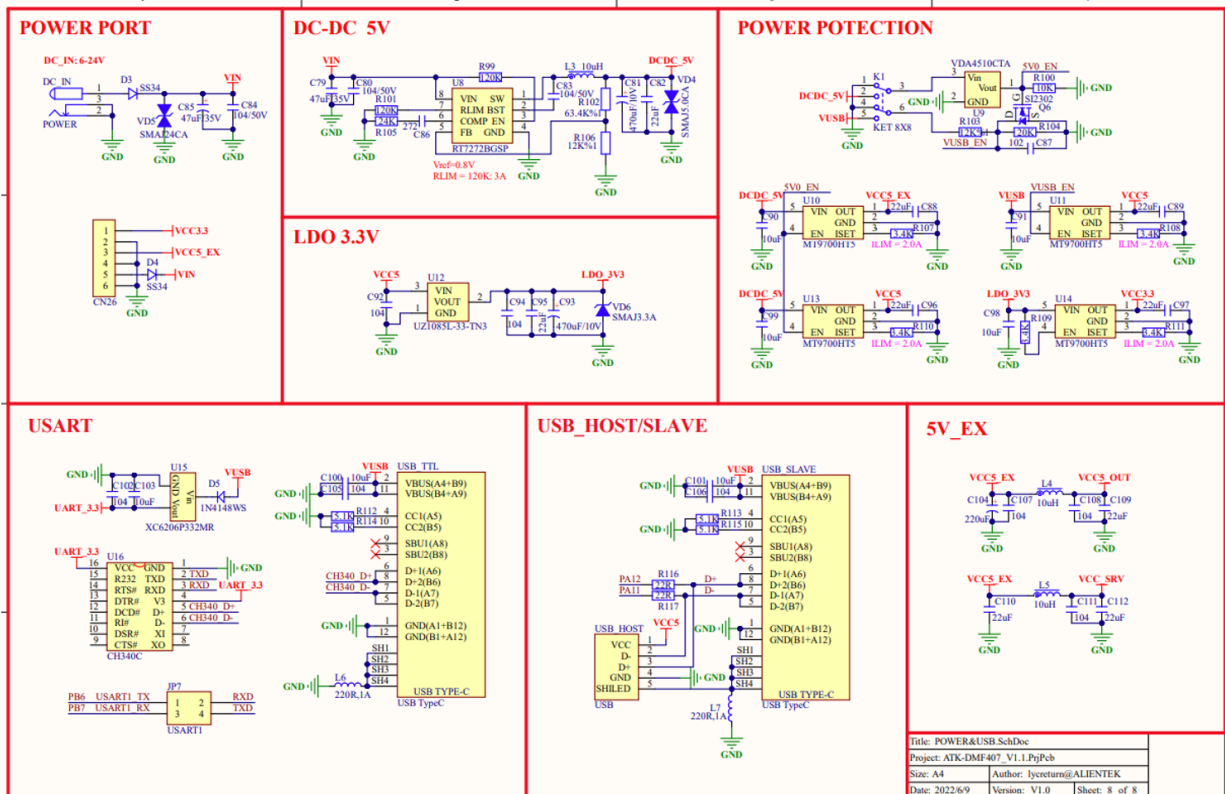
Besides, we should also test the accuracy for different motors. The accuracy for each motor may be different. Which means when we setting parameters and controlling the motors, we should assign different parameters to different motors. So this step should be repeat several times for every motors.

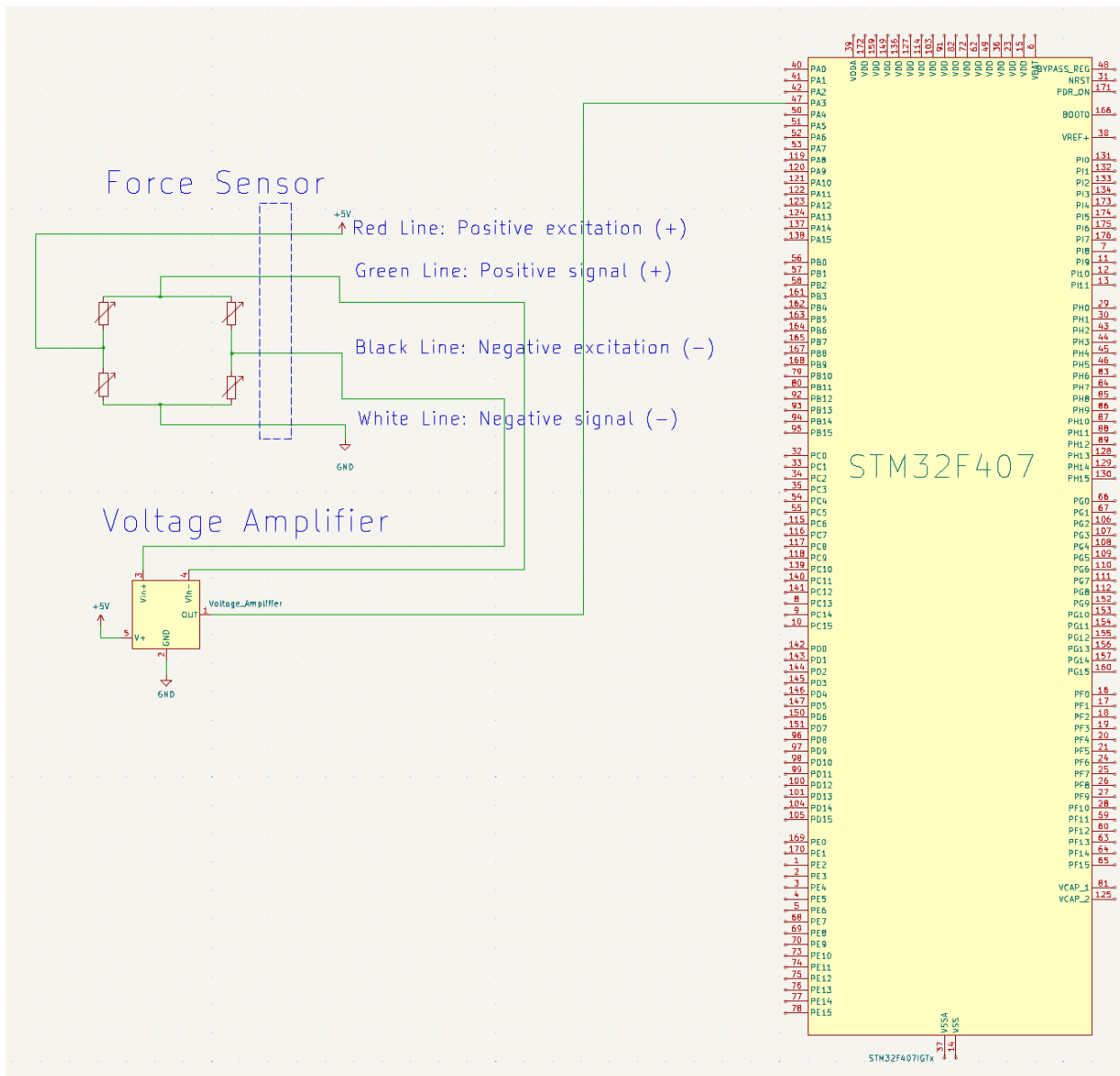
## 2.5 Compute Unit

### 2.5.1 Microcontroller (ATK-DMF407)

Input: 5V(USB) or DC6V 24V(DC005), voltage signal from sensor, data from motor driver  
 Output: command to motor driver, data of force sensor to Bluetooth Module

The microcontroller (ATK-DMF407)[4] serves as the central hub for interfacing with the Bluetooth module (ATK-MW579), the force sensor (DYMH-106), and the motor unit (ATK-PD5050S). It connects to the Bluetooth module utilizing TX and RX pins for bi-directional data exchange, allowing for wireless communication with other devices using UART. The force sensor is linked to an analog GPIO pin of the microcontroller, enabling the measurement of variable voltages that correspond to physical forces applied to the sensor. This analog signal is then digitized by the microcontroller for processing. For the motor unit, the microcontroller uses digital GPIO pins to send control signals (enable, direction, and pulse) to the stepper motor driver, dictating the motor's operational state, direction, and speed. This intricate network of connections between the microcontroller and the various components facilitates a seamless integration, enabling the microcontroller to collect data, process inputs, and control outputs, thus forming the backbone of a sophisticated sensor and control system.





<b>Component Interface</b>	<b>ATK-MW579 Pin</b>	<b>ATK-PD5050S Interface</b>	<b>F407 Corresponding Interface</b>
Power Supply (VCC)	VCC	-	5V
Ground (GND)	GND	-	GND
UART Transmit (TXD)	TXD	-	PB11
UART Receive (RXD)	RXD	-	PB10
Status Indicator	STA	-	PF6
Wake Up (WKUP)	WKUP	-	PC0
Enable Negative	-	ENA-	PF15
Enable Positive	-	ENA+	ST+ (Supply Voltage Positive)
Direction Negative	-	DIR-	PF14
Direction Positive	-	DIR+	ST+ (Supply Voltage Positive)
Pulse Negative	-	PUL-	PI5
Pulse Positive	-	PUL+	ST+ (Supply Voltage Positive)

Table 6: Microcontroller Pin Interface Mapping

Requirement	Verification Procedure
ADC Readings Within $\pm 5\%$ Tolerance of Actual Input	<ol style="list-style-type: none"> <li>1. Apply the output voltage from the voltage amplifier to the ADC input of the STM32. Simultaneously measure this voltage using a voltmeter to obtain an accurate reference value. Record this value.</li> <li>2. Record the ADC reading from the STM32 for the applied voltage.</li> <li>3. Calculate the percentage error between the STM32 ADC reading and the actual voltage measured by the voltmeter to ensure it is within the <math>\pm 5\%</math> tolerance.</li> <li>4. Repeat the procedure with multiple sets of voltage values to perform error testing and comparison. This will ensure consistency and reliability of the ADC across a range of input voltages.</li> </ol>

Table 7: ADC Verification Procedures

**2.5.2 Bluetooth Module (ATK-MW579)**

Input: 3.3V 5V power supply and pin connection with microcontroller

Output: data transmitted to host device (Bluetooth protocol)

The Bluetooth Module modulates 1Mbps enhanced data rate with complete 2.4GHz radio[5].

<b>Pin</b>	<b>Name</b>	<b>Description</b>
1, 3, 9, 19, 20	GND	Ground
2	ANT	Antenna
4	SLEEP	Sleep Mode
5	WKUP	Wake Up, Active High
6	RELOAD	Reload Counter
7	RXD	UART Receive (RX) Pin
8	TXD	UART Transmit (TX) Pin
10	3V3	Power Supply (3.3V-5V)
11-15, 18	NC	Not Connected
16	LINK	Connection Status Indicator (Active Low: Connection, Active High: No connection, Flashing: Transmitting)
17	RST	Reset (Active Low)

Table 8: Pin Descriptions of the Bluetooth Module

Connection with STM32 dev board.

<b>ATK-MW579 Pin</b>	<b>F407 Corresponding Interface</b>
VCC	5V
GND	GND
TXD	PB11
RXD	PB10
STA	PF6
WKUP	PC0

Table 9: ATK-MW579 to F407 Interface Mapping



Requirement Description	Verification Procedure
Module must be properly initialized with UART settings for communication.	Verify the UART settings on the module match the system's requirements by sending AT commands and ensuring appropriate responses are received.
Module must be discoverable by the host device for connection establishment.	Perform a scan on the host device and verify that the ATK-MW579 is listed among the discoverable devices.
Module must be able to establish a Bluetooth connection with the host device.	Initiate a connection from the host to the module and verify successful connection establishment by checking the module's LINK pin or LED status.
Module must transmit data with integrity to the host device over an established Bluetooth connection.	<ol style="list-style-type: none"> <li>1. Send a predefined data string from the module to the host and confirm receipt on the host side. Use a logic analyzer or software to capture and verify the transmitted data.</li> <li>2. Compare the received data on the host device against the original string sent from the module to ensure data integrity.</li> </ol>

Table 10: Bluetooth Module Requirements and Verification

## 2.6 Power System

Input: Li-Polymer Battery (14.8 V, 48 Wh)

Output:

1. DC6V 24V for Microcontroller (ATK-DMF407)
2. DC12 50V for High-performance stepper motor driver (ATK-PD5050S)

Our project's power system is centered around a highly efficient Li-Polymer Battery, which boasts a 14.8V, 48 Wh specification, designed to cater to the diverse energy needs of our setup. This battery adeptly provides a variable DC output range, offering 14.8V for the microcontroller (ATK- DMF407) , and it is within range of DC 12V to 50V for a high-performance stepper motor driver (ATK-PD5050S), ensuring both precision and power. Furthermore, critical components like the Bluetooth module and the force sensor derive their power indirectly through the microcontroller, showcasing our system's integrated approach to power distribution and management for optimal functionality and efficiency.

<b>Requirement</b>	<b>Verification Procedure</b>
Supply $\pm 14.8V \pm 5\%$ power for both Micro-controller and motor driver.	Use a multimeter to check if the voltage output are specified values $\pm 14.8V \pm 5\%$ . And use an oscilloscope to check if the voltage signal is steady.

Table 11: Power System Requirements and Verification

## 2.7 Software Subsystem

Our Software subsystem mainly consists of 4 parts. Data Display, Data Analysis, User Interface and User Control. Users can use User Control to set parameters, such as the drilling depth, drilling positions and so on. The Data Analysis part will analysis the bluetooth signal sent by Bluetooth Module, then send it to the User Interface. The User Interface will display the position, the depth and the force values, and send a figure to Data Display, where a 3d figure will show on the computer screen. This is the flow chart for our software subsystem.

<b>Requirement</b>	<b>Verification Procedure</b>
The software should let the user control the whole device, including control the motors and the probe	In the user interface, test all the function of each button, and check the movement of the device.
The software should also receive data from STM32 board, and generate a figure about the force and distance.	Compare the force in the figure and the force data in the STM32 board, check whether the software generate the right figure.

Table 12: Power System Requirements and Verification

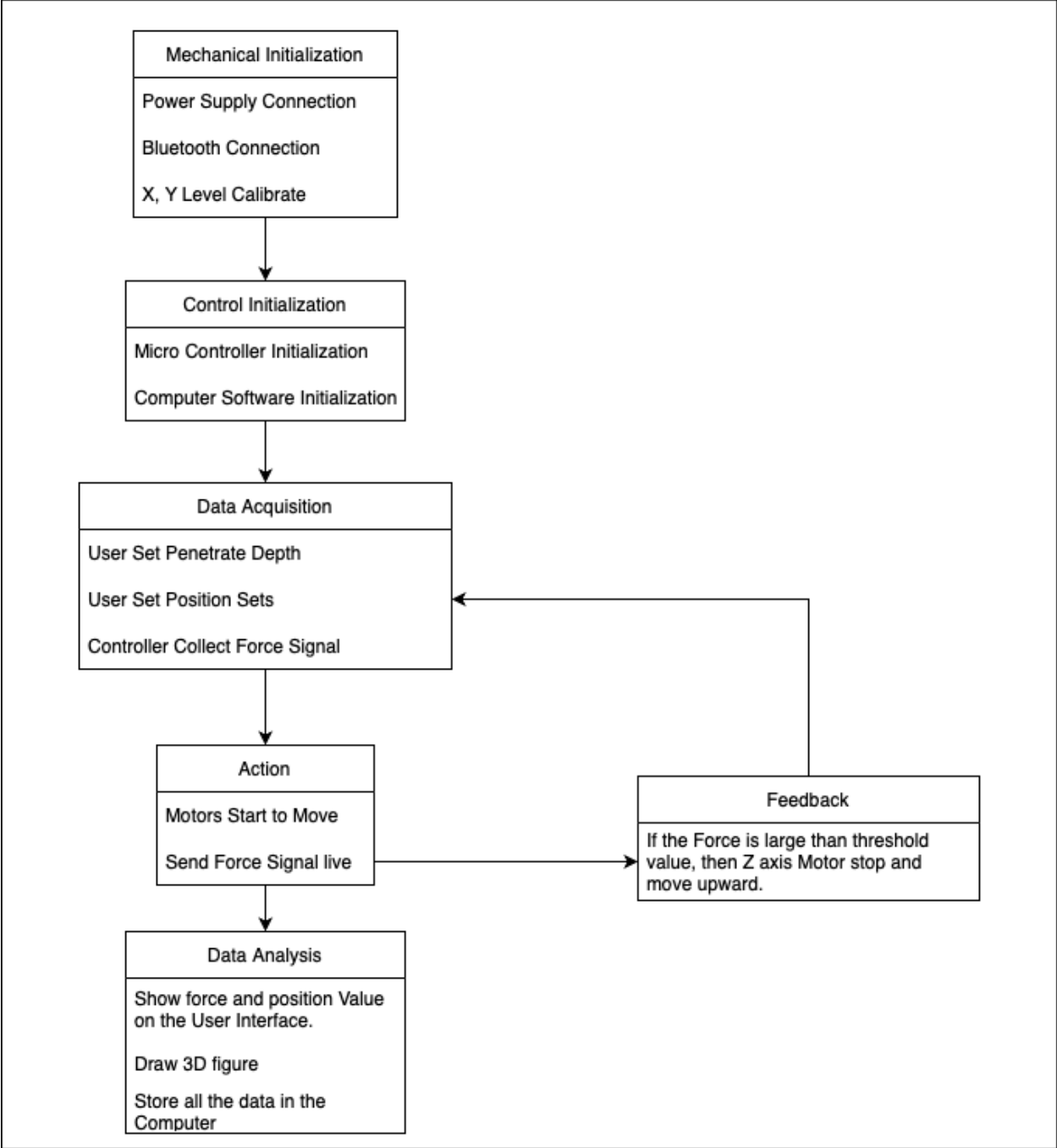


Figure 8: Software Subsystem Flow Chart

## 2.8 Calculation

### 2.8.1 Power Consumption

**Battery Specifications:**

1. Type: Lithium Polymer (Li-Poly)

2. Voltage: 14.8 V
3. Energy Capacity: 48 Watt-hours (Wh)
4. Battery Capacity:  $\frac{48}{14.8} = 3.24Ah$

Load	Voltage	Current (Active Mode)
Microcontroller	+6V~24V	+140mA
Bluetooth Module	+3.3V	+8.7mA
Motor Driver	+12V-50V	+0.75A
Total		0.8987A

Table 13: Electrical Specifications

### Drilling Operation Specifications:

1. Target Depth: 50 cm
2. Drilling Rate: 20 mm/s
3. Time to Drill One Hole: Approximately 25 seconds (0.00694 hours)

**Power Consumption and Drilling Capacity:** Each drilling operation consumes roughly 0.0924 Wh of the battery's energy. Based on the total energy capacity of 48 Wh, the system can perform approximately 500 drilling operations on a full charge. This estimate assumes constant current consumption as per the system's active mode specifications and includes the energy expenditure for the drilling process. For the experiment of going out to investigate the soil structure, the 500 boreholes of this equipment can meet our needs.

### Important Considerations:

1. The number of holes drilled is contingent upon the current draw being consistent with the system's active mode consumption.
2. The efficiency of the drilling process and the hardness of the soil are variables that can impact the actual number of holes that can be drilled. Soil that is more compact or contains rocks may require more energy to drill.
3. Battery performance can degrade over time, and environmental factors like temperature can influence the effective capacity of the battery.

An analysis of theoretical calculations of power consumption for drilling operations shows that with an available lithium polymer battery with an energy capacity of 48 Wh and a calculated battery capacity of 3.24 Ah at 14.8 V, the system is capable of performing approximately 500 drilling operations before needing to be recharged, providing for the fulfilment of field expeditions. This calculation assumes a constant current consumption consistent with the specified activity pattern of the system, considering a total current consumption of all components of 0.8987 A. However, the efficiency of these operations

is dependent on the homogeneity of the soil composition and the physical state of the battery, which may vary due to factors such as soil density and ambient temperature. Therefore, while the theoretical ability to perform 500 boreholes under optimal conditions provides a robust baseline for operational planning, actual performance may be biased by real-world variables and the inherent degradation of cell efficiency over time.

## 2.8.2 ADC Conversion Period

For enhanced precision in the penetrometer's soil analysis, the Analog-to-Digital Converter (ADC) sampling time has been configured to 480 clock cycles. And our goal is to achieve a resolution, targeting a sampling density of 200 samples/mm. This decision is inspired by the high sampling rate of 242 samples/mm used by a comparable project by WSL for their Snow MicroPen (SMP4) [6].

Given the STM32F407's ADC clock frequency ( $f_{ADC}$ ) of 21 MHz, the time per ADC clock cycle is:

$$T_{\text{cycle}} = \frac{1}{f_{ADC}} = \frac{1}{21 \times 10^6 \text{ Hz}} \approx 47.619 \times 10^{-9} \text{ s}$$

Therefore, the ADC conversion time ( $T_{\text{conv}}$ ) is calculated as:

$$T_{\text{conv}} = (\text{SMP} + 12) \cdot T_{\text{cycle}} = (480 + 12) \cdot 47.619 \times 10^{-9} \text{ s} \approx 23.43 \times 10^{-6} \text{ s}$$

To maintain the penetrometer's speed at 20 mm/s and to ensure a high sampling rate, the time to sample each millimeter of penetration is:

$$\text{Time per mm} = \frac{1 \text{ s}}{20 \text{ mm/s}} = 50 \text{ ms/mm}$$

For a desired 200 samples per millimeter, the maximum allowable time per sample is:

$$\text{Time per sample} = \frac{\text{Time per mm}}{\text{Number of samples per mm}} = \frac{50 \text{ ms}}{200} = 250 \mu\text{s}$$

Comparing this to our conversion time:

$$\text{Number of ADC samples within } 250 \mu\text{s} = \frac{250 \mu\text{s}}{T_{\text{conv}}} \approx \frac{250 \mu\text{s}}{23.43 \mu\text{s}} \approx 10.67$$

The calculation indicates that with a conversion time of approximately 23.43  $\mu\text{s}$ , our penetrometer can achieve 10.67 samples within the time it takes to penetrate 1 mm of soil. It meets our target sampling density, and the increased ADC cycle count should enhance the accuracy of our measurements.

By comparing our designed sampling rate with that of an established instrument, we validate the adequacy of our system's resolution for detailed soil analysis.

## 2.9 Tolerance Analysis

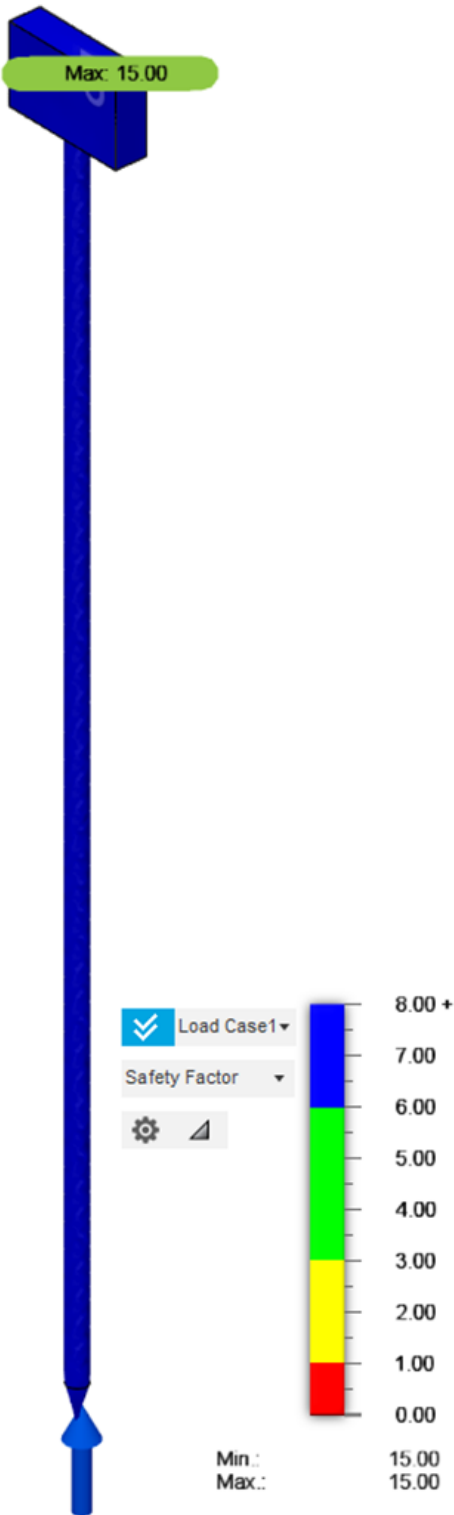
Due to the longer length of the detection rod we used and the greater force we experienced during using, we used simulation to calculate the safety factor of the rod when working in soil. The safety factor formula is:

$$SF = \frac{UltimateLoad(Strength)}{AllowableLoad(Stress)} \quad (2)$$

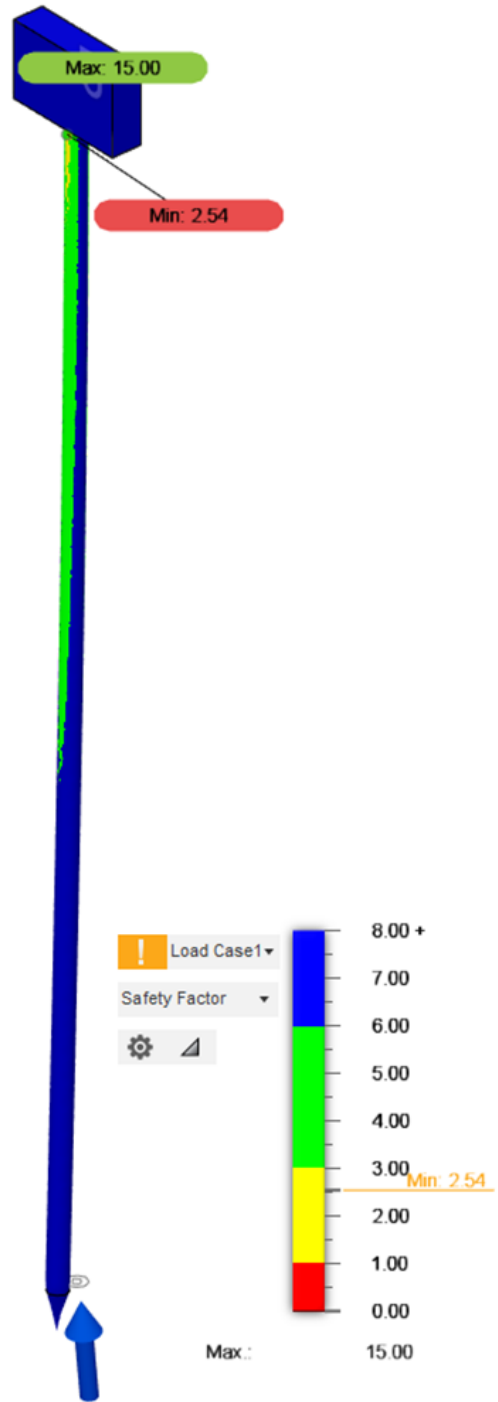
Research has shown that the penetration pressure of soil is between 0.8MPa and 1.9MPa. We calculate based on the formula  $F = P * \frac{\pi}{4}d^2$ , and get the force acting on the detection rod is approximately 260N. So we set the conditions for the rod tip to be subjected to a pressure of 260N, as well as a forward and backward tilt of 5 ° and a lateral tilt of 5 °, and obtained four different results to simulate the force situation of the rod when encountering uneven ground. From the results obtained, it can be seen that the safety factors of the detection rods are all above 2.5, which is within an acceptable range.

In the figures, the legend on the side tells us that colors represent different safety factor values, with the blue part having the highest safety factor, indicating that this part is least likely to breakage.

What's more, to ensure the drill goes down vertically, we are going to put gradienter on aluminum profile to test whether it's horizontal. And we use screws on the bottom of four corners to adjust the aluminum base to make it horizontal. Since the z axis is perpendicular to the aluminum profile, the drill is vertical obviously if the aluminum profile is horizontal.

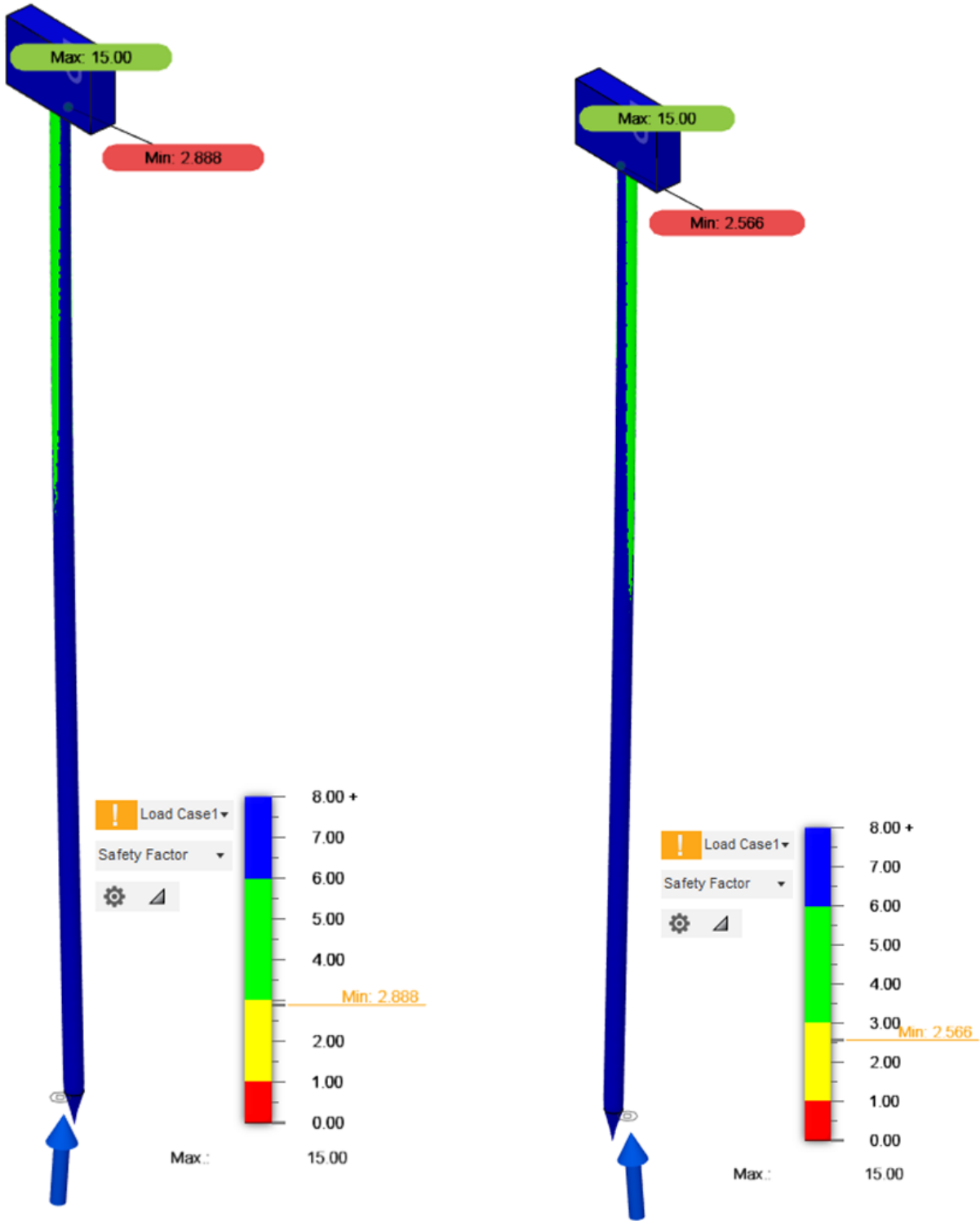


(a) Safety factor under 260N, tilt angle of 0°



(b) Safety factor under 260N, tilt angle of 5° backward

Figure 9: Safety factors for 0° and 5° backward tilt angles.



(a) Safety factor under 260N, tilt angle of 5° forward

(b) Safety factor under 260N, tilt angle of 5° lateral

Figure 10: Safety factors for 5° forward and lateral tilt angles.



### 3 Cost and Schedule

#### 3.1 Cost Analysis

##### 3.1.1 Labor

Name <sup>1</sup>	Hourly Rate <sup>1</sup>	Hours <sup>1</sup>	Total <sup>1</sup>	Total x2.5 <sup>1</sup>
Chenghao Mo <sup>1</sup>	¥30 <sup>1</sup>	200 <sup>1</sup>	¥6000 <sup>1</sup>	¥15000 <sup>1</sup>
Xing Shen <sup>1</sup>	¥30 <sup>1</sup>	200 <sup>1</sup>	¥6000 <sup>1</sup>	¥15000 <sup>1</sup>
Zheyuan Wu <sup>1</sup>	¥30 <sup>1</sup>	200 <sup>1</sup>	¥6000 <sup>1</sup>	¥15000 <sup>1</sup>
Chenxian Meng <sup>1</sup>	¥30 <sup>1</sup>	200 <sup>1</sup>	¥6000 <sup>1</sup>	¥15000 <sup>1</sup>
<b>Total<sup>1</sup></b>			¥60000 <sup>1</sup>	

Table 14: Labor Costs<sup>1</sup>

##### 3.1.2 Parts

Table 15: Component Costs

Description	Quantity	Manufacturer	Vendor	Cost/unit	Total cost
Li-Polymer Battery (14.8 V, 48 Wh)	1	Wei Zhen	Tao Bao	¥34.20	¥34.20
GGP Dual optical axis ball screw module guide rail with motor	1	Mei Ke Transmission	Tao Bao	¥310.00	¥310.00
57HB56L4/1.2Nm2 motor		Planetary Deceleration	Tao Bao	¥50.00	¥100.00
3030N1-Aluminum profile	4	YYDS	Tao Bao	¥24.00/m	¥56.00
Continued on next page					

**Table 15 – continued from previous page**

<b>Description</b>	<b>Quantity</b>	<b>Manufacturer</b>	<b>Vendor</b>	<b>Cost/unit</b>	<b>Total cost</b>
3030ED-Aluminum profile	1	YYDS	Tao Bao	¥21.00/m	¥12.00
Manufacture service for Aluminum profile	4	YYDS	Tao Bao	¥1.00	¥4.00
3030 Corner code	10	YYDS	Tao Bao	¥1.20	¥12.00
Delivery fee for Aluminum profile	1	YYDS	Tao Bao	¥12.00	¥12.00
DYM-106 Micro pressure sensor	1	Ocean Sensor	Tao Bao	¥234.00	¥234.00
Bluetooth module	1	Zheng Dian Atom	Tao Bao	¥56.36	¥56.36
Stepper motor driver	3	Zheng Dian Atom	Tao Bao	¥114.86	¥344.58
Motor control development board	1	Zheng Dian Atom	Tao Bao	¥532.38	¥532.38
Plum blossom handle screw	4	Wu Xi Quality	Tao Bao	¥0.74	¥2.80
Large head screw	1	Guwan Ji	Tao Bao	¥7.82	¥7.82
Guide rail slide	3	Zhejiang Zhenhao	Tao Bao	¥30.00	¥90.00
Guide rail slide with lock	3	Zhejiang Zhenhao	Tao Bao	¥48.00	¥144.00
Belt pulley	4	AET Hardware	Tao Bao	¥9.00	¥36.00

Continued on next page

**Table 15 – continued from previous page**

<b>Description</b>	<b>Quantity</b>	<b>Manufacturer</b>	<b>Vendor</b>	<b>Cost/unit</b>	<b>Total cost</b>
Belt	2	Libose	Tao Bao	¥9.00	¥18.00
<b>Total</b>					<b>¥2006.14</b>

### 3.1.3 Grand Total

<b>Section</b>	<b>Total</b>
Labor	¥60000
Parts	¥2006.14
<b>Grand Total</b>	<b>¥62006.14</b>

Table 16: Grand Total Costs (Labor + Parts)

## 3.2 Schedule

Table 17: Project Schedule and Task Allocation

<b>Week</b>	<b>Task</b>	<b>Responsibility</b>
	Mechanical design and assemble	Zheyang Wu, Chenxian Meng
	Controlling and software system	Xing Shen, Chenghao Mo
1/22/2024	Project Selection Form (Mon)	All
	PCB Design Exercise (Fri)	All
	Team Contract (Fri)	All
2/26/2024	Request for Approval (Wed)	All
	Decide the function of the product and allocate the work	All
3/4/2024	Proposal Early due (Thur)	All
	Discussed with professor and decided basic parameters	Xing Shen
Continued on next page		

**Table 17 – continued from previous page**

<b>Week</b>	<b>Task</b>	<b>Responsibility</b>
	Search for information about motors and sensors	Chenghao Mo
	Search for parts for aluminium profile and other mechanical parts	Zheyang Wu
	Study the CAD models for the whole structure	Chenxian Meng
3/11/2024	Proposal (Mon)	All
	Decide the control system with a micro control unit and PCB design	Xing Shen
	Select a specific force sensor and learn how to transmit the data to a computer	Chenghao Mo
	Design the structure and buy some mechanical parts	Zheyang Wu
	Buy ball screw module guide rail with motor and battery	Chenxian Meng
3/18/2024	Learn how to connect Bluetooth unit and force sensor to the micro control unit	Xing Shen
	Get the sensor and try to read the data in practice	Chenghao Mo
	Build up the first version of main structure	Zheyang Wu
	Build up the first version of main structure	Chenxian Meng
3/25/2024	Project Proposal Regrade (Tue)	All
	Design Document (Wed)	All
	Try to use the Bluetooth control in practice	Xing Shen
Continued on next page		

**Table 17 – continued from previous page**

<b>Week</b>	<b>Task</b>	<b>Responsibility</b>
	Use the sensor the control system to read the data Make adjustment of basic structure and design the probe Test the real force in soil with sensor and estimate the range.	Chenghao Mo Zheyang Wu Chenxian Meng
4/1/2024	Teamwork Evaluation (Thur) Connect the control system with driver and motor and make adjustment Use Bluetooth unit to connect sensor and read the data Design the suitable probe with correct testing range 3D printing the probe and test it in the soil	All Xing Shen Chenghao Mo Zheyang Wu Chenxian Meng
4/8/2024	Individual Progress Report (Wed) Fix the problem the control the motor as desired Fix the problem about sensor transmitting and realize the function Design the rest xy axis about its automatic moving function Buy the belts, belts pulley and relevant connections	All Xing Shen Chenghao Mo Zheyang Wu Chenxian Meng
4/15/2024	Design Document Revision (Fri) Study the rest two axis of motor control Perfect the function of interaction function	All Xing Shen Chenghao Mo
Continued on next page		

**Table 17 – continued from previous page**

<b>Week</b>	<b>Task</b>	<b>Responsibility</b>
	Assemble the rest part of xy axis Assemble the rest part of system with control system	Zheyang Wu Chenxian Meng
4/22/2024	Connect all control system together and find problem Continue test the part of testing data and reading data Test the work process and find the problem Make adjustment of the structure	Xing Shen Chenghao Mo Zheyang Wu Chenxian Meng
4/29/2024	Fix the problems of control system Fix the problems of data reading process and presentation Complete the whole mechanical structure Make sure the physical work of the product	Xing Shen Chenghao Mo Zheyang Wu Chenxian Meng
5/6/2024	Main Mock Demo Final Report draft (Fri) Find the potential problems and fix them	All All All
5/13/2024	Final Demo Block Final Presentation Final Individual Design Report draft	All All All
5/20/2024	Final Individual Design Report (Tue) Functionality Demonstration Video (Tue)	All All
Continued on next page		

**Table 17 – continued from previous page**

<b>Week</b>	<b>Task</b>	<b>Responsibility</b>
	Final Report (Tue)	All
	Teamwork Evaluation II (Tue)	All

## 4 Safety

There are several potential safety hazard with our project. First is in the moving subsystem. If we overcharge the Li battery, then it may be brought to an extreme high temperature or even explosion. In this case, we will use a timer to control the charge time of the Li battery, when the battery interface shows that it have been fully charged, then we should immediately stop the charging. Besides, after we shut done the machine, we should disconnect the battery with the motor and micro controller, in order to make these electronic device safe.

Besides, consider the working environment, our electronic device cannot working when it is raining, snowing, which may cause electronic device irreversible damage. For the temperature, our device is allowed to work on a large range of temperature, from  $-20^{\circ}C$  to  $40^{\circ}C$ . Both high or low temperature may cause damage to the electronic devices.

Since there are several Al material in our device, its corners may cause injury to users, so we should make sure every sharp corner is rounded. Meanwhile, the probe is also in a sharp corn shape, so when we turn off the machine, we should make sure a protection cover is add to the probe.

Furthermore, all members of the team have completed the lab safety training before engaging with any lab work. The trainings completed include both general laboratory safety training and electrical safety training.



## 5 Ethical Issue

Our project's commitment to ethical engineering practices is rooted in the principles outlined by the IEEE Code of Ethics[7]. We integrate these principles within the scope of our micro-penetrometer development:

1. **Public Safety:** We prioritize the safety and well-being of the public by ensuring the reliable and safe operation of our penetrometer, conducting extensive tests to mitigate potential hazards.
2. **Honesty:** We adhere to honesty in reporting findings and claims based on data obtained by our penetrometer, ensuring accuracy and verifiability.
3. **Technological Understanding:** We aim to enhance technological understanding and its appropriate application in soil and snow analysis, with implications for agriculture and disaster management.
4. **Technical Competency:** We maintain technical competence, embarking on tasks within our expertise, and disclose any limitations in knowledge or abilities.
5. **Criticism and Credit:** We welcome criticism constructively, acknowledge errors, and appropriately credit the contributions and efforts of others involved in the project.
6. **Avoiding Harm:** We are committed to avoiding harm, by designing a safe-to-handle device and securely enclosing all electronic components to prevent misuse or accidents.
7. **Professional Support:** We support our peers in professional development and uphold the ethical standards set forth by IEEE, fostering a culture of ethical engineering.

Each principle is actively practiced and reflected in our project methodology, ensuring an alignment of our technological advancements with ethical engineering standards.

## References

- [1] D. E. Rolston, M. N. A. Bedaiwy, and D. T. Louie, "Micropenetrometer for in situ measurement of soil surface strength," *Soil Science Society of America Journal*, vol. 55, no. 2, pp. 481–485, 1991. DOI: 10.2136/sssaj1991.03615995005500020031x.
- [2] M. Schneebeli and J. B. Johnson, "A constant-speed penetrometer for high-resolution snow stratigraphy," *Annals of Glaciology*, vol. 26, pp. 107–111, 1998.
- [3] OpenEDV, *Documentation for the ATK-2MD4850 Module*, <http://www.openedv.com/docs/modules/other/ATK-2MD4850.html>, [Online; accessed 28-September-2023], 2023.
- [4] OpenEDV, *ATK-DMF407 Development Board Documentation*, <http://www.openedv.com/docs/boards/stm32dj/ATK-DMF407.html>, [Online; accessed 28-September-2023], 2023.
- [5] OpenEDV, *ATK-BLE02 IoT Module Documentation*, <http://www.openedv.com/docs/modules/iot/ATK-BLE02.html?highlight=%E8%93%9D%E7%89%99>, [Online; accessed 28-September-2023], 2023.
- [6] W. I. for Snow and A. R. SLF. "'SMP4 Manual'." (2024), [Online]. Available: [https://www.wsl.ch/fileadmin/user\\_upload/SLF/Services\\_Produkte/Forschungsgeraete/SMP4/SMP4\\_Manual\\_0.84.pdf](https://www.wsl.ch/fileadmin/user_upload/SLF/Services_Produkte/Forschungsgeraete/SMP4/SMP4_Manual_0.84.pdf) (visited on 04/13/2024).
- [7] Institute of Electrical and Electronics Engineers, *IEEE Governance Documents*, Accessed: 2023-09-28, 2023. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>.

## Appendix A ADC + Bluetooth

```
#include "./SYSTEM/sys/sys.h"           // Include system header
    for SYS functions
#include "./SYSTEM/delay/delay.h"       // Include header for
    delay functions
#include "./SYSTEM/usart/usart.h"       // Include header for
    USART functions
#include "./BSP/LED/led.h"              // Include header for LED
    control
#include "./BSP/KEY/key.h"              // Include header for key
    interface
#include "./BSP/LCD/lcd.h"              // Include header for LCD
    interface
#include "./BSP/ATK_MW579/atk_mw579.h" // Include header for
    ATK_MW579 Bluetooth module
#include "./BSP/ADC/adc.h"              // Include header for ADC
    functions

// Bluetooth and display configuration defines
#define DEMO_BLE_NAME          "ATK-MW579"           // Define the
    Bluetooth device name
#define DEMO_BLE_HELLO        "HELLO ATK-MW579"     // Define the
    power-on welcome message
#define DEMO_BLE_ADPTIM       5                     // Define the
    advertisement interval speed

/**
 * @brief Display experiment information on the screen.
 */
void show_mesg(void)
{
    // Display information on the LCD
    lcd_show_string(10, 10, 220, 32, 32, "STM32", RED);
    lcd_show_string(10, 47, 220, 24, 24, "ATK-MW579", RED);

    // Print information through the serial port (USART)
    printf("\n*****\r\n");
    printf("STM32\r\n");
    printf("ATK-MW579\r\n");
    printf("*****\r\n\n");
}

/**
 * @brief Bluetooth functionality and setup.
 */
void bluetooth(void)
{
```

```

uint8_t ret;          // Variable to store return values of Bluetooth
                    // functions
uint8_t key;         // Variable to store key press values
uint8_t *recv_dat;  // Pointer to store received data from Bluetooth

uint16_t adcx;      // Variable to store ADC value
float temp;         // Temporary float variable for calculations
float voltage;      // Variable to store voltage calculation result

// Initialize ATK-MW579 Bluetooth module with a baud rate of 115200
ret = atk_mw579_init(ATK_MW579_UART_BAUDRATE_115200);
if (ret != 0) // If initialization failed, enter an infinite loop
and toggle LED
{
    printf("ATK-MW579 init failed!\r\n");
    while (1)
    {
        LED0_TOGGLE();
        delay_ms(200);
    }
}

// Configure ATK-MW579 module settings
atk_mw579_enter_config_mode();
ret = atk_mw579_set_name(DEMO_BLE_NAME); // Set Bluetooth name
ret += atk_mw579_set_hello(DEMO_BLE_HELLO); // Set welcome message
// Add other configuration settings here...
if (ret != 0) // If configuration failed, enter an infinite loop
and toggle LED
{
    printf("ATK-MW579 config failed!\r\n");
    while (1)
    {
        LED0_TOGGLE();
        delay_ms(200);
    }
}

// Restart UART reception
atk_mw579_uart_rx_restart();

while (1) // Main loop
{
    // ADC processing and display logic
    // Get ADC result average
    adcx = adc_get_result_average(ADC_ADCX_CHY, 10); // ADC channel
    and number of averages

```

```

    lcd_show_xnum(134, 110, adcx, 5, 16, 0, BLUE); // Display ADC
        value

    // Voltage calculations
    temp = (float)adcx * (3.3f / 4096); // Convert ADC value to
        voltage
    voltage = temp; // Copy to voltage variable for UART
        transmission
    adcx = (uint16_t)temp; // Convert to integer for display
    lcd_show_xnum(134, 130, adcx, 1, 16, 0, BLUE); // Display
        integer part of voltage

    // Extract and display decimal part of voltage
    temp -= adcx;
    temp *= 1000;
    lcd_show_xnum(150, 130, (uint16_t)temp, 3, 16, 0x80, BLUE); //
        Display decimal part

    // Toggle LED and scan for key presses
    LED0_TOGGLE();
    delay_ms(100);
    key = key_scan(0);

    // Key press handling
    // Send data through UART on key press
    // Wake up Bluetooth module on key press
    // ... (Add key press handling logic here)

    // Receive data from the master device through UART
    recv_dat = atk_mw579_uart_rx_get_frame();
    if (recv_dat != NULL)
    {
        printf("%s", recv_dat); // Print received data
        atk_mw579_uart_rx_restart(); // Restart reception
    }

    delay_ms(10); // Short delay for loop timing
}
}

/**
 * @brief Main function where the program starts execution.
 */
int main(void)
{
    // Initialize hardware and peripherals
    HAL_Init(); // Initialize HAL library

```

```

sys_stm32_clock_init(336, 8, 2, 7); // Configure the system clock
delay_init(168); // Initialize delay functions
usart_init(115200); // Initialize USART with baud
    rate of 115200
led_init(); // Initialize LED interface
key_init(); // Initialize key interface
lcd_init(); // Initialize LCD screen
adc_init(); // Initialize ADC

// Display messages and start main application
show_mesg(); // Display experiment information
lcd_show_string(30, 67, 200, 16, 16, "STM32", RED); // Display "
    STM32"
lcd_show_string(30, 87, 200, 16, 16, "ADC TEST", RED); // Display "
    ADC TEST"
lcd_show_string(30, 107, 200, 16, 16, "ADC1_CH3_VAL:", BLUE); //
    Display ADC value label
lcd_show_string(30, 127, 200, 16, 16, "ADC1_CH3_VOL:0.000V", BLUE);
    // Display ADC voltage label

bluetooth(); // Enter the main Bluetooth
    function
}

```

## Appendix B Software + Motor

```

import asyncio
from bleak import BleakScanner, BleakClient

async def run_bleak():
    stm32_address = "38:3B:26:36:26:21" # Directly using the address
        as confirmed

    async with BleakClient(stm32_address) as client:
        if await client.is_connected():
            print(f"Connected to {stm32_address}")

            # Subscribe to notifications on a characteristic
            notify_uuid = "9ecadc24-0ee5-a9e0-93f3-a3b50300406e"
            def notification_handler(sender, data):
                print(f"Notification from {sender}: {data.decode('utf
                    -8')}") # Decode from UTF-8 when receiving

            await client.start_notify(notify_uuid, notification_handler
                )
            print("Subscribed to notifications. Listening for messages

```

```

        from STM32...")

# Loop to send messages repeatedly
while True:
    user_input = input("Enter the message to send (type '
        exit' to quit): ")
    if user_input.lower() == 'exit':
        break

    data_to_send = user_input.encode('utf-8') # Convert to
        UTF-8 encoded bytes
    write_uuid = "9ecadc24-0ee5-a9e0-93f3-a3b50200406e"
    await client.write_gatt_char(write_uuid, data_to_send,
        response=False)
    print("Data sent to STM32")

# Unsubscribe to clean up
await client.stop_notify(notify_uuid)
print("Unsubscribed from notifications and exiting.")

else:
    print("Failed to connect to the device.")

asyncio.run(run_bleak())

#include "./SYSTEM/sys/sys.h"
#include "./SYSTEM/delay/delay.h"
#include "./SYSTEM/usart/usart.h"
#include "./BSP/LED/led.h"
#include "./BSP/KEY/key.h"
#include "./BSP/LCD/lcd.h"
//#include "demo.h"
#include "./BSP/ATK_MW579/atk_mw579.h"
#include "./BSP/ADC/adc.h"
#include "./BSP/TIMER/stepper_tim.h"
#include "./BSP/STEPPER_MOTOR/stepper_motor.h"
#include <string.h>

#include "./BSP/RTC/rtc.h"
#include "./USMART/usmart.h"

#define DEMO_BLE_NAME            "ATK-MW579"                /*
    */
#define DEMO_BLE_HELLO          "HELLO ATK-MW579"         /*
    */
#define DEMO_BLE_ADPTIM         5                          /*
    */

```

```

/**
 * @brief
 * @param
 * @retval
 */
void show_mesg(void)
{
    /*      L      C      D      */
    lcd_show_string(10, 10, 220, 32, 32, "STM32", RED);
    lcd_show_string(10, 47, 220, 24, 24, "ATK-MW579", RED);
    //lcd_show_string(10, 76, 220, 16, 16, "ATOM@ALIEN TEK", RED);

    /*                      */
    printf("\n");
    printf("*****\r\n");
    printf("STM32\r\n");
    printf("ATK-MW579\r\n");
    //printf("ATOM@ALIEN TEK\r\n");
    printf("*****\r\n");
    printf("\r\n");
}

void bluetooth(void)
{
    uint8_t ret;
    uint8_t key;
    uint8_t *recv_dat;

    uint16_t adcx;

    uint8_t start_hour, start_min, start_sec, start_ampm;
    uint8_t hour, min, sec, ampm;
    uint8_t year, month, date, week;
    uint8_t tbuf[40];

    float temp;
    float voltage;

    uint8_t id = 1;
    uint8_t flag = 0, t = 0;
    uint8_t dir = 0;
    uint16_t set_speed = 100;

    uint8_t start_t;

    char buf[32];

```



```

/* ATK- M W 5 7 9 */
ret = atk_mw579_init(ATK_MW579_UART_BAUDRATE_115200);
if (ret != 0)
{
    printf("ATK-MW579 init failed!\r\n");
    while (1)
    {
        LED0_TOGGLE();
        delay_ms(200);
    }
}

/*      ATK      -MW579 */
atk_mw579_enter_config_mode();
ret = atk_mw579_set_name(DEMO_BLE_NAME);
ret += atk_mw579_set_hello(DEMO_BLE_HELLO);
ret += atk_mw579_set_tpl(ATK_MW579_TPL_P0DBM);
ret += atk_mw579_set_uart(ATK_MW579_UART_BAUDRATE_115200,
    ATK_MW579_UART_DATA_8, ATK_MW579_UART_PARI_NONE,
    ATK_MW579_UART_STOP_1);
ret += atk_mw579_set_adptim(DEMO_BLE_ADPTIM);
ret += atk_mw579_set_linkpassen(ATK_MW579_LINKPASSEN_OFF);
ret += atk_mw579_set_leden(ATK_MW579_LEDEN_ON);
ret += atk_mw579_set_slavesleepen(ATK_MW579_SLAVESLEEPEN_ON);
ret += atk_mw579_set_maxput(ATK_MW579_MAXPUT_OFF);
ret += atk_mw579_set_mode(ATK_MW579_MODE_S);
if (ret != 0)
{
    printf("ATK-MW579 config failed!\r\n");
    while (1)
    {
        LED0_TOGGLE();
        delay_ms(200);
    }
}

/*                               */
printf("Connection Success\r\n");
atk_mw579_uart_rx_restart();

while (1)
{

    rtc_get_time(&hour, &min, &sec, &ampm);
    rtc_get_date(&year, &month, &date, &week);
    sprintf((char *)tbuf, "Time:%02d:%02d:%02d", hour, min, sec);
}

```

```

lcd_show_string(30, 150, 210, 16, 16, (char*)tbuf, RED);

adcx = adc_get_result_average(ADC_ADCX_CHY, 10);
/*
    ADC10
*/
lcd_show_xnum(134, 110, adcx, 5, 16, 0, BLUE);
/*
    ADC
*/

temp = (float)adcx * (3.3 / 4096);
/*
    3
    .1111
*/
voltage = (float)adcx * (3.3 / 4096);
adcx = temp;
/*
    adcadcx16
*/
lcd_show_xnum(134, 130, adcx, 1, 16, 0, BLUE);
/*
    3
    3
    */
temp -= adcx;
/*
    3
    .1111 - 3 = 0.1111 */
temp *= 1000;
/*
    10000
    .1111
    111
    .1
*/
lcd_show_xnum(150, 130, temp, 3, 16, 0X80, BLUE);
/*
    111
*/

LED0_TOGGLE();
delay_ms(100);
key = key_scan(0);

switch (key)
{
    case KEY0_PRES:
    {
        /*
            */
        atk_mw579_uart_printf("adc:%f\r\n", voltage);
        break;
    }
    case KEY1_PRES:
    {

```

```

        /*          ATK          -MW579 */
        atk_mw579_wakeup_by_uart();
        break;
    }
    default:
    {
        break;
    }
}

/*          */
//  stepper_pwm_speed(set_speed, ATIM_TIMX_PWM_CH1);
recv_dat = atk_mw579_uart_rx_get_frame();
if (recv_dat != NULL)
{
    printf("%s", recv_dat);

    const char *start = "power";
    if (strncmp((const char*)recv_dat, start, strlen(start)) ==
        0) {
        flag = !flag;
        if(flag)
        {
            //          stepper_stop(id);
            stepper_pwm_speed(set_speed+100, ATIM_TIMX_PWM_CH1)
                ;
            stepper_star(id, dir);

            start_hour = hour;
            start_min = min;
            start_sec = sec;
            printf("start_hour:%d; start_min: %d; start_sec: %d
                \r\n", start_hour, start_min, start_sec);
            flag = !flag;
        }
        else
        {
            printf("sdhfksdjfksjd\r\n");
            stepper_stop(id);
        }
    }

    const char *change = "change";
    if (strncmp((const char*)recv_dat, change, strlen(change))
        == 0)
    {

```

```

        dir = !dir;
        stepper_star(id, dir);
        stepper_pwm_speed(set_speed+100,ATIM_TIMX_PWM_CH1);
        printf("hour:%d; min: %d; sec: %d\r\n",hour, min, sec);
        delay_ms(1000*((hour-start_hour)*60*60 + (min-start_min
            )*60 + (sec - start_sec)));
        stepper_stop(id);
        dir = !dir;
    }

    atk_mw579_uart_rx_restart();
}

if ((t % 20) == 0)
{
    LED0_TOGGLE(); /* 200ms , LED0 */
}
delay_ms(10);
}
}

int main(void)
{
    HAL_Init(); /* HAL */
    sys_stm32_clock_init(336, 8, 2, 7); /* ,168Mhz */
    delay_init(168); /* */
    usart_init(115200); /* 115200 */
    usmart_dev.init(84); /* USMART */
    led_init(); /* LED */
    key_init(); /* */
    lcd_init(); /* LCD */
    show_mesg(); /* */
    stepper_init(0xFFFF, 168 - 1);
    rtc_init(); /* RTC */
    rtc_set_wakeup(RTC_WAKEUPCLOCK_CK_SPRE_16BITS, 0); /* WAKE
        U P , 1 */
    adc_init(); /* ADC */
    lcd_show_string(30, 67, 200, 16, 16, "STM32", RED);
    lcd_show_string(30, 87, 200, 16, 16, "ADC TEST", RED);
    //lcd_show_string(30, 136, 200, 16, 16, "ATOM@ALIENTEK", RED);
    lcd_show_string(30, 107, 200, 16, 16, "ADC1_CH3_VAL:", BLUE);
    lcd_show_string(30, 127, 200, 16, 16, "ADC1_CH3_VOL:0.000V", BLUE);
    /* */
    bluetooth();
}

```