1. Introduction

In today's busy lifestyle, taking care of plants can be challenging. It's easy to forget to water the plants when you're away from home or have a hectic schedule. To address this issue, I developed a smart plant management system using Raspberry Pi and Telegram. This system allows users to remotely monitor the status of their plants and water them when necessary. By leveraging Telegram's group chat functionality, friends and family can collaboratively manage the plants, making it more efficient and convenient.

This report provides a detailed explanation of the project's main idea, the software and hardware components used, and the code implementation. It also discusses the challenges encountered during the project and the solutions implemented to overcome them. The smart plant management system serves as an example of how IoT technology and Telegram bots can be effectively utilized in real-life applications.

2. Main Idea

The main idea of this project is to manage a plant using a Telegram chatbot. The system includes three primary commands: status, water, and last watered.

- The status command checks the soil moisture level of the plant and provides a real-time update to the user.
- The water command allows users to remotely water the plant.
- The last watered command records the last time the plant was watered.

Additionally, every morning at 9 AM, the system measures the soil moisture level. If the moisture level exceeds the threshold, it records the time in last watered. If the moisture level is below the threshold, it executes the water command to water the plant.

By using these commands, users can easily monitor and manage their plants remotely, ensuring they receive the necessary care even when the users are away from home. The integration of Telegram's group chat functionality also allows friends and family to collaboratively manage the plant, making the process more efficient and convenient.

3. Software

3-1. Libraries and Software Used

This project utilizes several libraries and software tools to manage the plant using a Raspberry Pi and a Telegram chatbot.

The primary programming language used for scripting and automation is **Python**. To control the GPIO pins on the Raspberry Pi, the **RPi.GPIO library** is employed. This library is essential for activating the relay module, which controls the water pump.

For interacting with the Grove sensors and modules, the **Grove.py library** is used. Specifically, it reads data from the soil moisture sensor connected to the Grove Hat on the Raspberry Pi. To install this library, the command pip install grove.py is used.

The **python-telegram-bot library** is crucial for interfacing with the Telegram Bot API. It handles the Telegram bot commands such as status, water, and last watered. This library can be installed using pip install python-telegram-bot.

To schedule periodic tasks, the **schedule library** is used. It is particularly useful for scheduling the daily check of soil moisture every morning at 9 AM. This library can be installed with pip install schedule.

The **logging library** provides a flexible framework for emitting log messages from Python programs. It is used to log various events and errors for debugging and monitoring purposes.

The datetime library supplies classes for manipulating dates and times. It is used to record the time when the plant is watered.

Finally, the **threading library** is used to run scheduled tasks in a separate thread, allowing the main program to continue running smoothly.

These libraries and software components work together to create a system that can monitor and manage the plant remotely through a Telegram chatbot. The combination of these tools ensures that the plant receives the necessary care even when the user is away from home, making the process efficient and convenient.

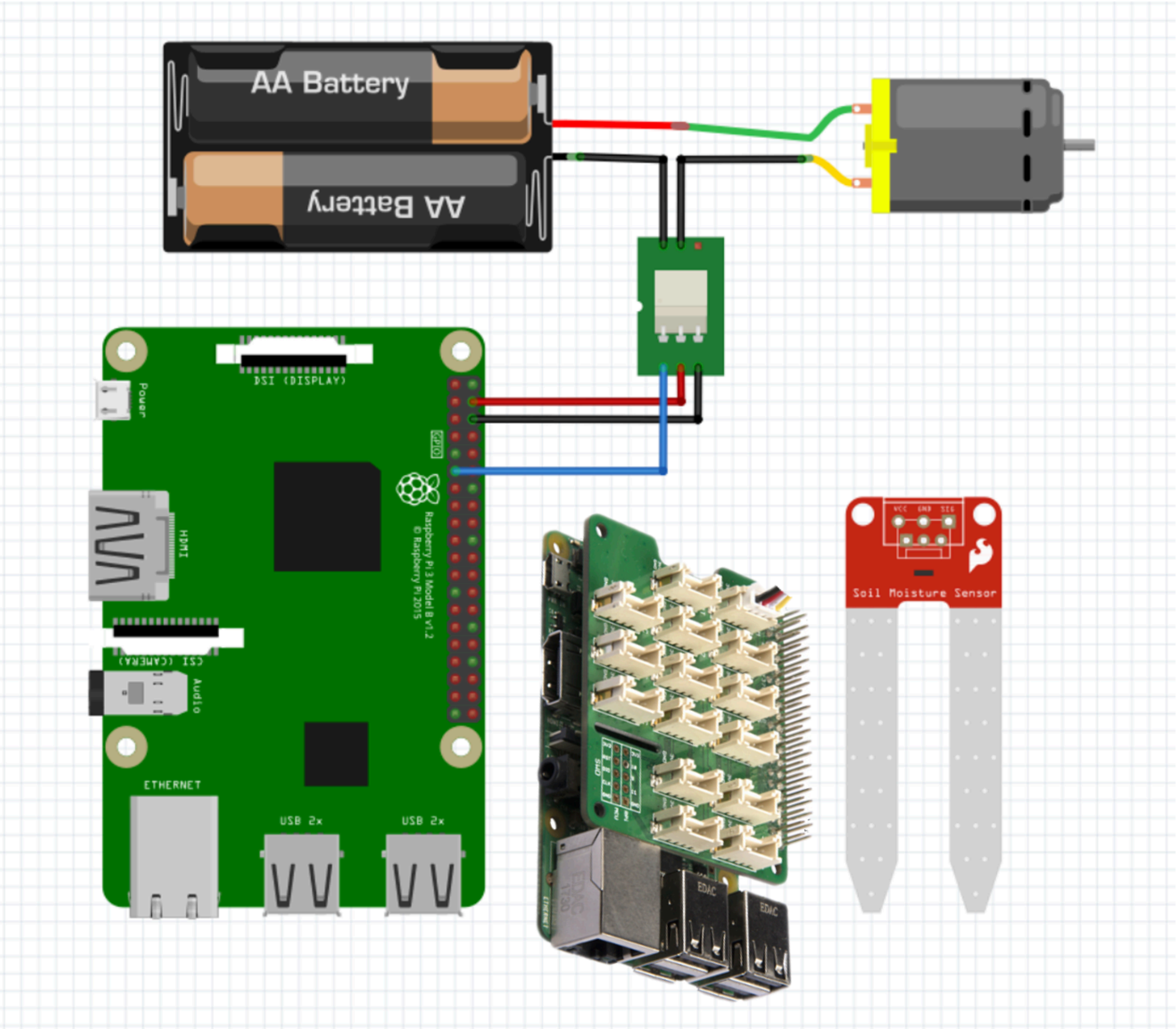
3-2. Challenges Faced

One of the significant challenges encountered during this project was the installation of the Grove.py library. The installation process was difficult and repeatedly failed, which was frustrating and time-consuming. To resolve this issue, I decided to reinstall the Raspberry Pi image provided by Grove. This image comes with the Grove modules pre-installed, which eliminated the installation issues I was facing. Using the pre-configured image from Grove saved a considerable amount of time and ensured that all necessary libraries were correctly set up. This solution allowed me to proceed with the development of the project without further delays.

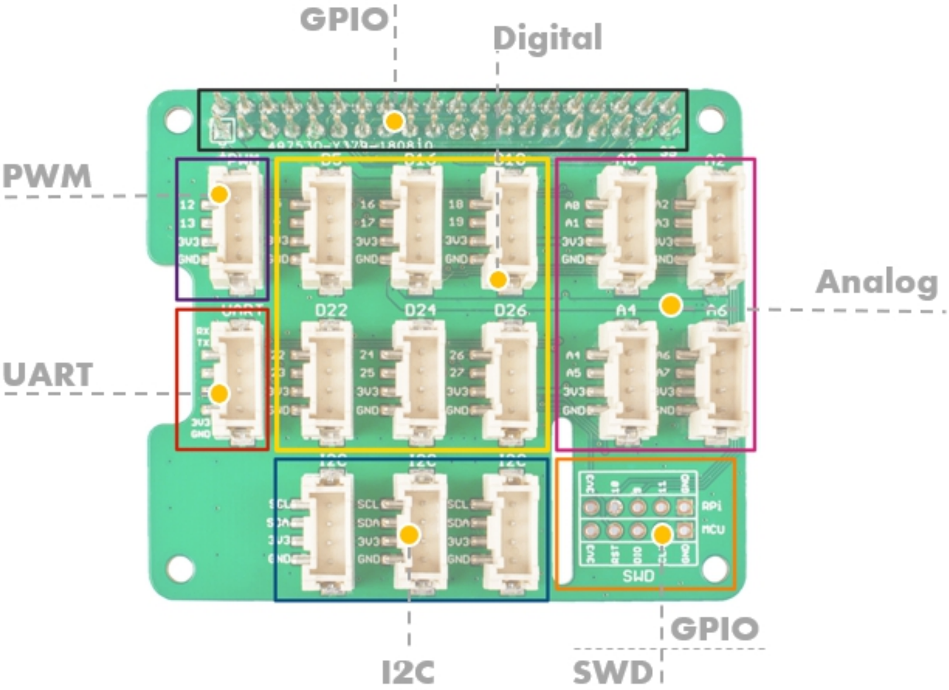
4. Hardware

4-1. Circuit Diagram

The circuit diagram for this project includes a Grove Hat, a soil moisture sensor, a 9V battery, and a water pump.



4-2. Details of the Devices



The **Grove Hat** is used because the Raspberry Pi cannot directly read analog sensors. By using the Grove Hat, the Raspberry Pi gains the ability to interface with analog sensors. This significantly simplifies the circuit, as it allows easy connection of various Grove modules through its standardized ports. In this project, the Grove Hat is connected to the Raspberry Pi and provides the necessary analog port for the soil moisture sensor from Grove.

The Grove Hat for Raspberry Pi expands the Raspberry Pi's connectivity options, allowing it to interface with various Grove sensors and modules. It provides multiple analog and digital ports, simplifying the connection process and reducing the need for additional wiring. The Grove Hat enables the Raspberry Pi to read analog sensor data, which it cannot do natively.

The **soil moisture sensor from Grove** is connected to one of the analog ports on the Grove Hat. This sensor measures the moisture level in the soil and provides an analog output that corresponds to the moisture level. The use of the Grove soil moisture sensor, together with the Grove Hat, simplifies the circuit design and makes the connections straightforward.