**Logbook**

**Planning:**

26/10/23:

I have decided on what I will create for this project, it will be a smart plant pot that aims to make it easier for the user to take care of the plant inside it.

12/11/23:

Having done Assignment 1 and used MQTT in question 2 I am going to apply what I have learned to send messages to the user’s phone for this project.

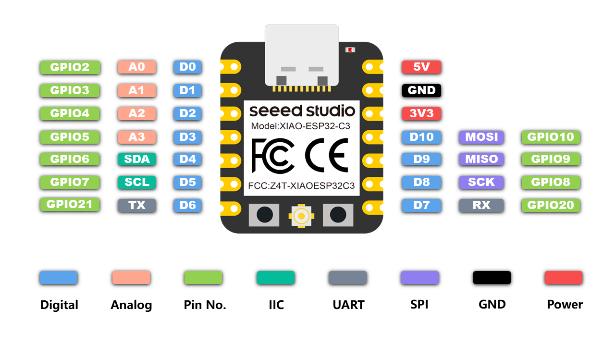
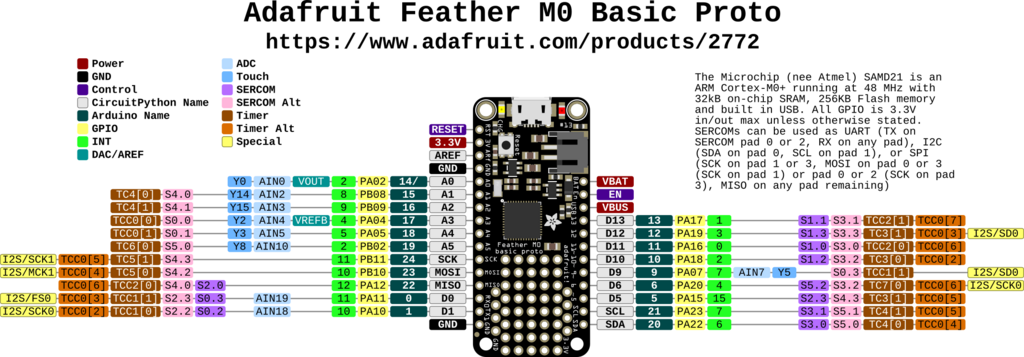
**Starting Work on the device:**

10/11/23:

I have started work on the sensors for the plant pot. I am using a DHT22 and BH1750. The DHT22 has humidity and temperature sensors in it and the BH1750 is a light sensor. The board I am using is the Xiao C3 ESP32. Currently the code doesn’t seem to work, and I am unsure why.

15/11/23:

I have changed the board I am using to a Adafruit ESP32 Feather as the old board has very little documentation and few pins which limits me heavily.

 Xiao C3 Pinout ESP32 Feather Pinout

In addition to this I am using a Thermistor and resistive humidity sensor rather than a DHT22. This means I can just take analogue readings from a wire instead of trying to find a library that works with both the board and sensor I have.

17/11/23:

A diagram of a circuit board

Description automatically generatedI have changed how the board is wired. I am using a bread board to do my wiring and have set up the circuit like so:

(Ignore that the

board says Uno)

Such that the sensors are in series with resistors and the data wires take a reading from between the two. This allows me to find the resistance of the sensors at any time by taking a reading then doing:

Resistor In Series Resistance / ((Max Reading Possible / Reading) – 1)

Which provides me with the resistance of the component.

All this code has been placed into a separate file for now but will be combined later.

18/11/23:

I have added flags from the lectures to the code I created for Assignment 1 question 2 to create a new part for this project which will send messages when the appropriate flags have been triggered. I have flags for water, temperature and light which send corresponding messages to the user’s phone via MQTT.

20/11/23:

I have added code that converts the resistance of the thermistor into and actual temperature reading via a method called the Steinhart equation. It takes a known resistance and temperature at that resistance as well as a beta coefficient which is calculated by a pair of known resistances and temperatures, and it converts resistance to temperature for thermistors:

(1 / (ln(Reading / Known Resistance) / Beta Coefficient) + (1 / (Known Temperature + 273.15))) = Temperature

The temperature is given is kelvin so to convert to Celsius I just subtract 273.15.

24/11/23:

The humidity sensor was giving incorrect readings, after taking it to the Makerspace it turned out that the Humidity sensor had a much higher resistance than originally thought as the thermistor was a 1.5k ohm one and it was assumed that the humidity sensor was the same but it was actually around 10m ohms so I swapped out the resistor in series with it to one closer to it’s resistance and the readings became more useful.

29/11/23:

I have added code for and wired the light meter. The BH1750 takes 4 wires 1 3.3v, 1 ground, 1 to SCL and 1 to SDA. I then am using a library for the sensor which requires me activate the SCL/SDA wires then set up the sensor into One-time High-resolution mode. After which I take a reading which ranges from around 0 – 600 with 500 being the average reading of a bright room and less than 10 being a dark area.

1/12/23:

I have combined the code for the MQTT and sensors, but the sensors have stopped working as a result. Have been debugging for most of the day.

7/12/23:

The Wi-Fi was interfering with the readings, so I have moved the code around so that readings are taken before anything happens with the Wi-Fi. The Wi-Fi was interfering because the Wi-Fi takes control of some of the data pins on the board and when I turned the Wi-Fi off after I was done with it, it seems to also have shutdown those pins.

8/12/23:

I have added deep sleep to the device so that it will save power by only taking readings every hour or so or whenever the plant is watered it will also wake up via an interrupt.

To save data between the deep sleeps I found a library that allows be to write to flash memory to store variables, so I use this to store the set-up information about the Wi-Fi and plant that the user gives, and store variables needed to find spikes in humidity.

9/12/23:

I have added a function to take a few readings and average then to avoid anomalies from causing errors.

I have then tested the device and it works.

**Reflections:**

My approach to this project was to split the device to sperate parts to work on individually then combine one by one to avoid errors from different parts interfering with each other. For example when I combined the sensors and Wi-Fi code together they caused issues btu I was able to resolve them fairly easily since the issue only happened after combining the code.

The part I found the easiest in this project was the programming as not much code was required, and Arduino has a simple arrangement with the setup and loop methods.

The most difficult part of this project was finding specifications and working out how the different components and parts worked as a few different parts/libraries interfered with each other and caused issues when logically then wouldn’t but due to how they have been implemented by the people who made them they did.

If I was to do this project again, I would start with a different Arduino board from the beginning as the original board I was using was far more limited than the one I changed to and if I had been using it from the start, I could have potentially had time to add more features to the device.

The main thing I feel I learned is to make my work modular such that I can isolate the issues that occur more easily and avoid having to search all the code to find what is causing issues.

I would like specific feedback on how I have implemented the wake up from deep sleep caused by the user watering the plant as I feel like I could have done this better and the deep sleep implementation in general along with the flag system I feel isn’t very well implemented.

I felt that the work was very interesting, and it was fun to work around the issues that happened although when I had to read datasheets and check through the different libraries to find issues it was annoying, especially when the issues are caused by things they don’t mention. Although I enjoyed the project overall.