Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

HydroSystem

Project Engineering

Year 4

Conor Finnerty

G00383112

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Atlantic Technological University

2022/2023

Declaration

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at Atlantic Technological University.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

_____

Acknowledgements

I'd like to thank Niall O'Keeffe my project supervisor for all the skill he has thought me throughout the past 4 year and this year. His experience and knowledge are priceless and I'm very grateful of his support throughout my project.

Further I'd like to thank Paul Lennon, Michelle Lynch, and Brian O'Shea for all their help and feedback throughout my project.

I'd also like to thank all my fellow classmates for being very supportive with reviewing and giving feedback on my poster, report, project video and my in-person project presentation. Working with my classmates I've developed my teamwork and communication skills.

Table of Contents

1 Summary

One of my personal passions is gardening and landscaping. I worked for my brother's gardening and landscaping company last year, the most frustrating part of the work was growing hydroponic plants in ponds for clients. Hydroponic plants grow in water without soil, the problem with growing them in ponds is that a lot of bacteria will gather quickly in the pond and spoil the nutrients intended for the plant. To grow healthy and strong hydroponic plants, there's a lot of maintenance in keeping the pond bacteria free.

Therefore, I've decided to act by designing HydroSystem. HydroSystem is a pond filtration unit, it is designed to filter the bacteria infected pond water, and feed the plants when the water is below the cleanliness threshold.

This is done by combining the following hardware and software components:

- ST B-L475E-IOT01A micro controller
- STM32CubeIDE
- Turbidity sensor (to detect the NTU (Nephelometric Turbidity Unit) value present)
- DC motor attached to gear and rack (open feeding hatch to provide nutrients to the plants)
- Ubidots STEM (an online dashboard to analyze and control HydroSystem)
- Filtration pump (to clean the water)

Embedded C is written in the STM32CubeIDE to the ST B-L475E-IOT01A micro controller to firstly connect to a mobile hotspot which supplies Wi-Fi to the board, which allows it to connect to Ubidots STEM online dashboard. The turbidity sensor will actively detect the NTU levels present in the water. If the NTU threshold level is exceeded the filtration pump will turn on until the NTU level present is below the threshold. Once the filtration pump has stopped the DC motor will rotate the gear along the rack and open the feeding hatch above the plants. Ubidots will display all the live data from HydroSystem, when the NTU threshold level is broken an email will be sent to inform the user of what threshold has been breached and allow the user to turn on or off the pump as they wish.

2 Poster

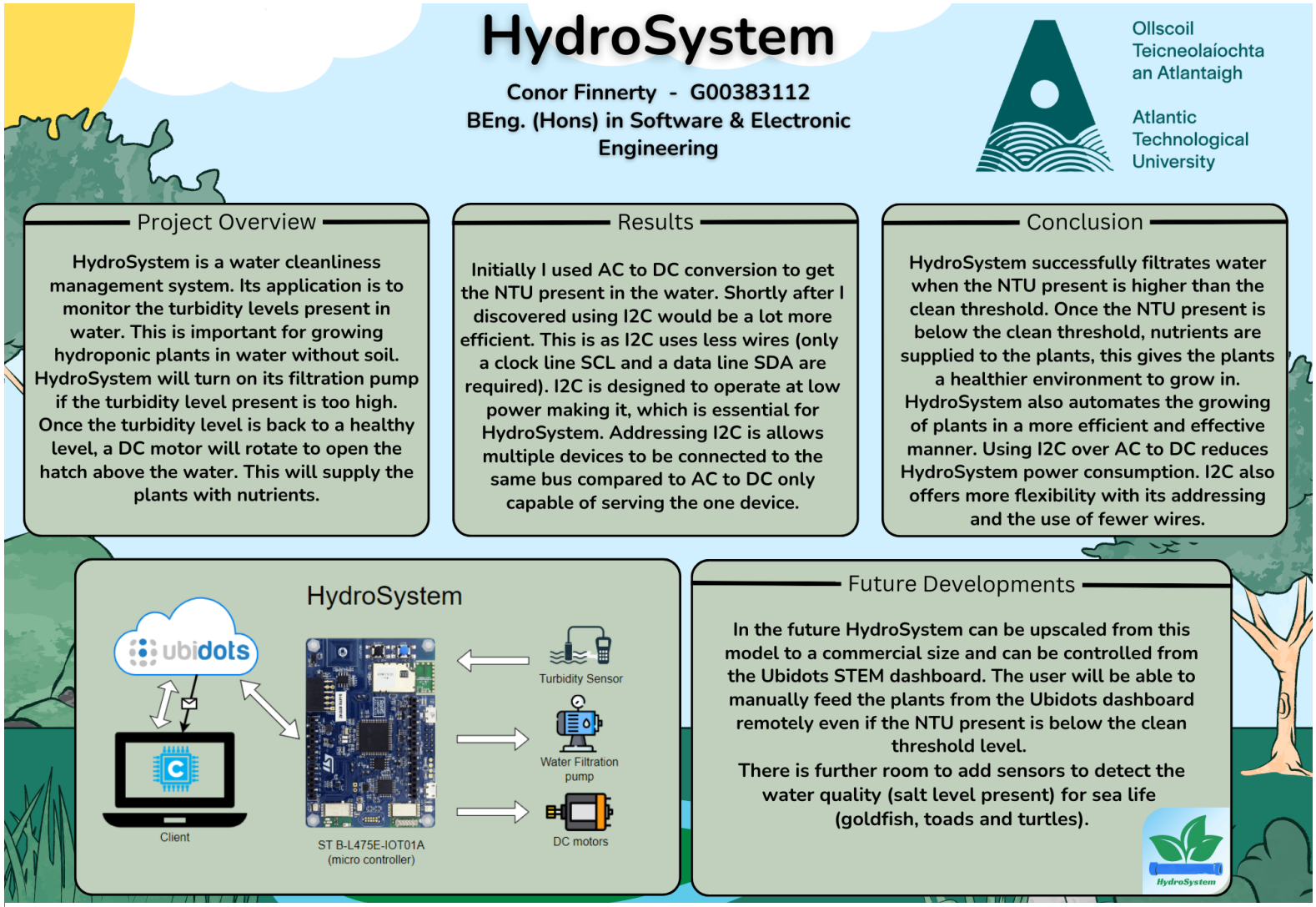


Figure 1 – HydroSystems' Poster

3 Introduction

HydroSystem is a prototype product to help grow hydroponic plants in a more efficient manner, whilst eliminating the long and boring task of keeping the water clean. HydroSystem optimizes healthy plant growth in water, by ensuring to only feed the hydroponic plants when the NTU level in the water is below the threshold which considers the water clean.

A personal hobby of mine is gardening and landscaping, as I love converting gardens from nothing to something very scenic and beautiful. Last summer I worked with my brother for his gardening and landscaping company. My favourite addition to a garden is pond filled with hydroponic plants. The problem with ponds is that they gather quite a lot of bacteria quickly, meaning they're high maintenance. The gathering of bacteria also hinders the growth and healthiness of the hydroponic plants. This requires having to visit clients at least once a month to clean their ponds with a filtration pump, filtration pumps are messy and time consuming to set up, expensive to buy and even more expensive to leave run over a long period of time.

I want to eliminate these annoying trips just to clean ponds. Therefore, I decided to act by designing HydroSystem. This will not only eliminate the trips to the clients with ponds but also ensure the hydroponic plants grow to their full blossom and stay healthy.

HydroSystem consists of a ST B-L475E-IOT01A micro-controller which is the brains of the product.

A turbidity sensor is used to monitor the NTU levels present in the water (cleanliness of the water).

A DC motor along with a Dual H-Bridge is used to open and close the feeding hatch from above.

Filtration pump, consisting of a water pump, tubing and a filter sponge, which cleans the water when the NTU levels present are too high.

Lastly Ubidots STEM is the online dashboard which allows the user to analyse and monitor the NTU levels present in the water at any time and grants the user remote manual control of the feeding hatch.

The combination of these components and technologies creates HydroSystem. It is a cost-effective solution to what is in place at the moment. HydroSystem would eliminate the labour cost, uses less power and effort to clean ponds. HydroSystem is also more environmentally friendly, it reduces the burning of fossil fuels as once it is set up it doesn't require a worker to drive out to the client. At times last summer I had to drive around 20km to the clients' house and a further 20km back to the yard.

4 Background

4.1 Turbidity

Turbidity refers to the cloudiness present in a liquid. Nephelometric Turbidity Unit (NTU)is the measurement used to detect turbidity. Nephelometry is a technique used to the turbidity presence in a liquid.

For water to be deemed safe to drink, it should be no higher than 5NTU, but the NTU presence should ideally be no more than 1NTU according to the World Health Organization. On the other hand, when growing hydroponic plants, it is recommended to not to have NTU levels greater than 10 present.

5 Project Architecture

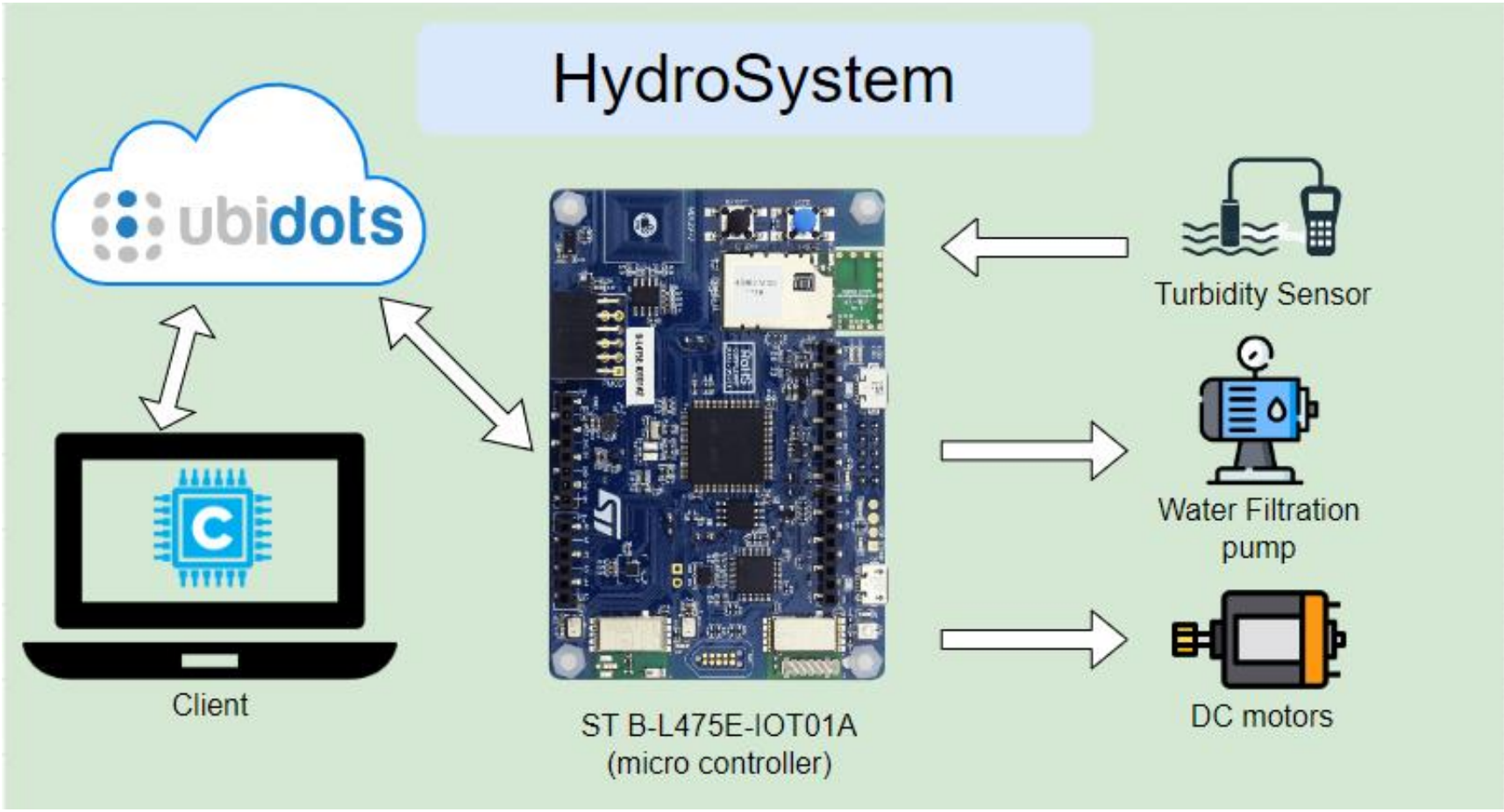


Figure 2 – HydroSystems' Architecture diagram

6 Project Plan

Personally, I found project planning very daunting, but I came to the realisation that effective project planning is vital for a successful project delivery. Therefore, I used OneNote and Jira to assist with planning out deadlines and logging updates for HydroSystem.

6.1 OneNote

OneNote is a digital notebook connected to OneDrive making it easily accessible to log notes, research, and project updates.

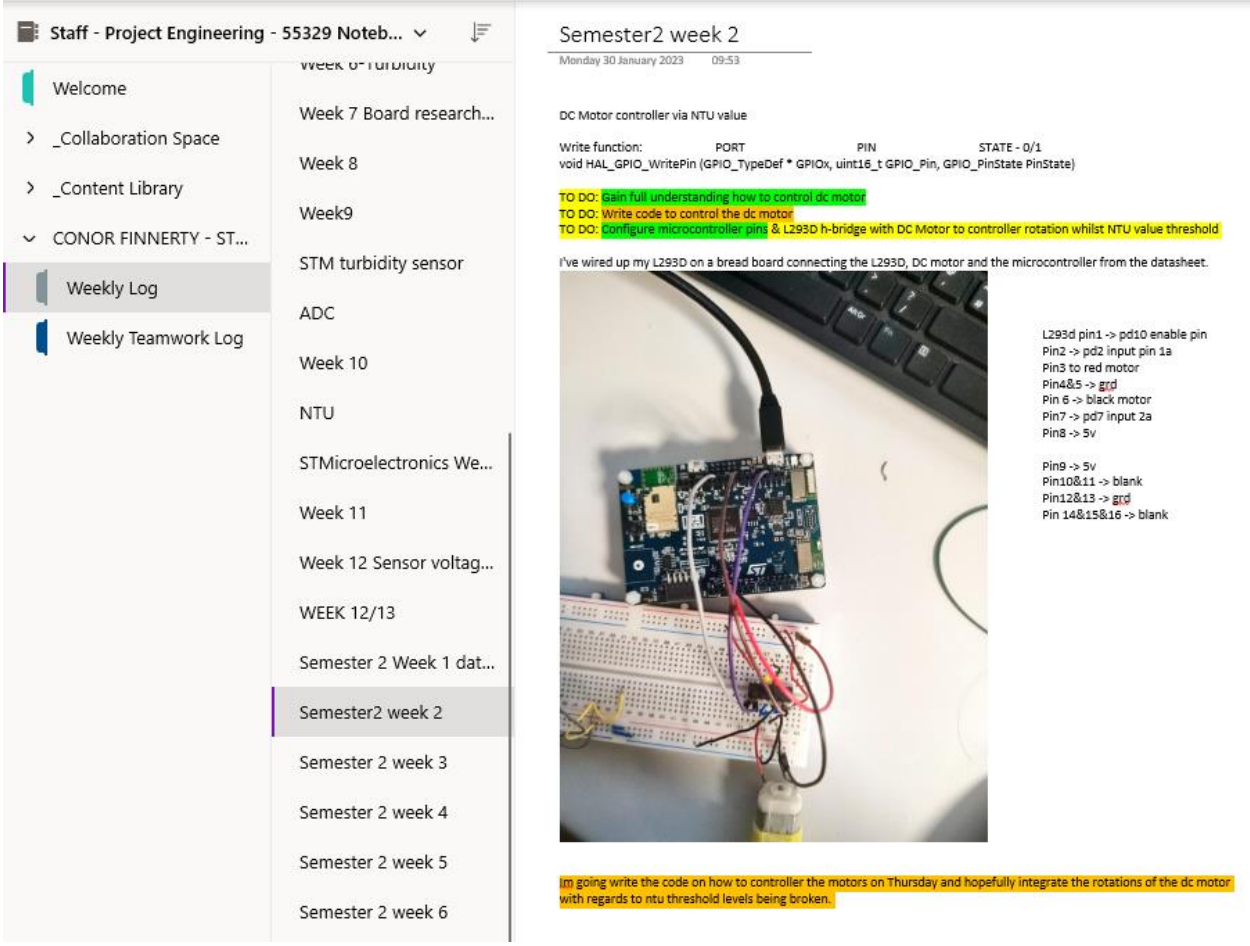


Figure 3 - OneNote

6.2 Jira

Jira is a project management tool which allows the user to map out a timeline, create sprints, reports and much more.

6.2.1 HydroSystems timeline

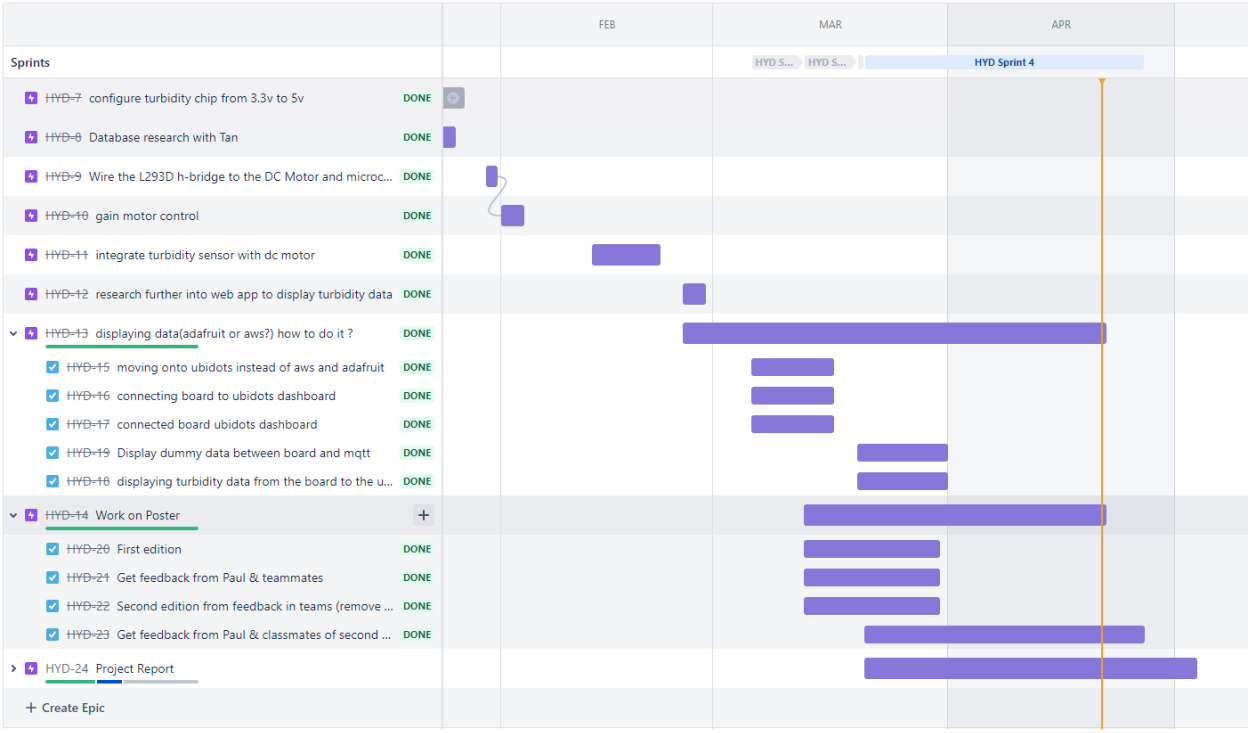


Figure 4 – Jira timeline

6.2.2 Sprint - Burnup report

Figure 5 – Jira burnup report

6.2.3 Sprint – burnup report explanation

A sprint is a to do list/work management tool, it allows the user(s) to enter an issue(s) (parts of a project the user/team is working on). Each issue will be assigned a time frame the user(s) are planning on completing the issue. The user(s) is also able to mark issues as “To do”, “In progress” and “Done”. The above image is a burnup report displaying what how many issues are in the sprint (in orange), how many issues have been completed (in green), a progression line (diagonal grey line) and a planned end line (vertical lighter grey line).

The graph on the previous page is a burnup report of one HydroSystem’s sprints for its poster, it included 4 issues. Each time the green line stepped up an issue was marked as completed. When the green line was above the progression line it states the issue was completed early, where when it was below the progression line the issue was overdue or completed late. The poster was enroute to being delivered earlier than anticipated, but it ultimately the poster was delivered later than anticipated. This didn’t mean the poster was uploaded late, just means it took longer to complete than originally planned. This occurred due to waiting on the final issue to be completed which was incorporating feedback from teammates into the poster before submitting.

7 Components and Technologies

HydroSystem consists of 1 micro controller, 2 software components, 3 hardware components and is written using Embedded C.

7.1 ST B-L475E-IOT01A

The ST B-L475E-IOT01A is a STMicroelectronics micro controller. It contains a lot of key features such as its 802.11 802.11 b/g/n compliant Wi-Fi module which allows HydroSystem to connect to the Ubidots STEM online dashboard.

The ST B-L475E-IOT01A also consists of an Ultra-low-power STM32L4 Series MCUs based on Arm Cortex-M4 core, 32 I/O (input/Output) pins and I2C compatibility.

The Ultra-low-power STM32L4 Series MCUs based on Arm Cortex-M4 core is vital for due to the need for HydroSystem to be running for long periods of time. This is very cost effective compared to using other microcontrollers such as a NXP LPC4370 which would draw a lot more power. [1]

7.2 I2C

I2C is a communication protocol, there are a few different configurations I2C can be set up. HydroSystem uses I2C master-slave interrupt configuration, where a bus is shared between the microcontroller and other digital devices.

The interrupt feature is essential for I2C communication as it allows the devices to signal events or request attention without the need for continuous polling. When an interrupt occurs, the microcontroller temporarily pauses its current operation, executes the interrupt service routine (ISR) to handle the event, and then resumes its previous task. This efficient approach improves overall system performance and ensures timely data transfer between devices.

7.3 STM32CubeIDE

The programming language used to control HydroSystem is Embedded C. STMicroelectronics have their own IDE: STM32CubeIDE, this IDE caters for the development, debugging and running of HydroSystems. All the peripherals, pin connectivity's and cloud communication (Ubidots) are set up in the IDE.

7.4 Embedded C

Embedded C was chosen as HydroSystem's programming language as it's compatible with the ST B-L475E-IOT01A micro controller. Embedded C is great for Real Time Operating Systems it has the capability of processing data in real time, allowing the code to make very quickly decisions from live data it receives from IoT sensors and cloud dashboard components.

7.5 Turbidity sensor

The Turbidity sensor detects the NTU present in water. The ST B-L475E-IOT01A micro controller is configured with I2C Master Slave interrupt to calculate the AC signal to NTU.

7.6 L293D Dual H-Bridge Motor Driver

The L293D is an integrated circuit (IC) that provides clockwise and anti-clockwise motor control for the DC motor. The L293D doesn't only provide directional control but also speed control. There are 2 independent H-Bridges inside the IC with the capability to drive 2 dc motors each. Each H-Bridge consists of 4 transistors arranged in a configuration which provides bidirectional motor control. The built in flyback diodes protect the circuit from voltage spikes when the motor is switched off.

7.7 DC motor

The DC (Direct Current) motor is controlled by the dual H-Bridge, which provides bidirectional control and speed control. The 2 main components in the motor are a stator and an armature. The stator is static, and the armature is dynamic.

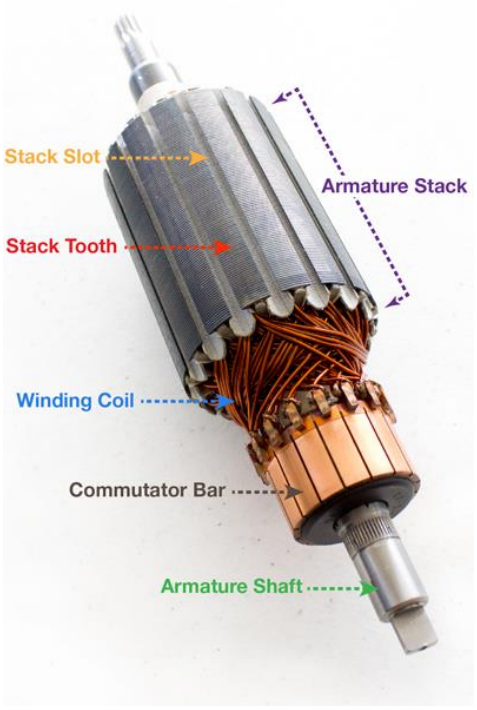


Figure 6 – Inside of DC motor [2]

The stator houses a magnetic field that causes the armature to rotate when an electric current flows through it. As the current passes through the motor's internal coil, a magnetic field is generated, which interacts with the rotor's magnets, propelling the motor into motion.

7.8 Ubidots STEM

Ubidots STEM is a cloud-based Internet of Things (IoT) online dashboard. This dashboard displays HydroSystem's data (turbidity level present) on gauge component.

The MQTT broker is set up in the code along with the structure of the MQTT broker credentials (hostname, port, security settings, username, password, and client ID). An MQTT client is created and responsible for establishing a connection to the MQTT broker, enabling communication with the broker to publish messages and subscribe to topics. The MQTT broker connects to the Wi-Fi hotspot. Once the Wi-Fi network is initialized, the IP and MAC address' are retrieved. The IP address uniquely identifies the device on the network and enables communication with other devices over the network. The MAC address is used by the ethernet protocol to identify the device on the local network.

The online dashboard provides a variety of components; HydroSystem uses the gauge component and Ubidots STEM Events for email notifications.

The gauge component provides a live visual of the data being collected from the Turbidity sensor.

7.8.1 MQTT broker

An MQTT broker is a publish/subscribe system that enables communication between devices in IoT (Internet of Things) applications. An MQTT broker acts as a post office that receives and routes messages between devices connected to the network. Devices can subscribe to topics on online dashboards and receive messages are published to those topics.

Devices that wish to communicate with each other using MQTT must first connect to the broker. The broker receives messages from devices and routes them to the destination based on their subscribed topics. This way, devices don't need to know specific IP address' or location of the other devices they are communicating with, as the broker handles the routing of messages between devices.

7.9 Ubidots STEM Events

Ubidots STEM events are a feature of the Ubidots STEM that allow you to trigger actions based on specific events or conditions.

Events can be thought of as triggers that are associated with data published or subscribed to the linked Ubidots STEM dashboard. An event is when 1 or a series of changes occur causing the trigger to activate and the set action is performed. These changes include things like:

- The data variable breaking/reaching the set threshold once or a set number of times
- Reinitializing the data variable
- Inactivity of a variable over time
- Entering or exiting the set geofence perimeter[3]

Once an event is triggered, the set action is performed. These actions can include things like:

- Sending an email or SMS notification
- Posting a message to a Slack channel
- A pre-recorded phone call to the set phone number
- Publishing a message to a MQTT topic
- Updating a data variable in the Ubidots STEM dashboard
- Running a custom script or webhook

By defining events and actions, you can create a flexible and responsive system that can adapt to changing conditions and automatically respond to critical events or situations.

7.10 Filtration Pump

The filtration pump is a combination of a water pump, tubing and a filter sponge. The pump has a motor which rotates the impeller. The faster the impeller (rotor in the pump) is rotated the

quicker the water will flow. The filter sponges are attached to 2 ends of the tubing, which filters out the turbidity.

Unfortunately, with extremely long delivery times and wrong pump being sent, this feature wasn't implemented due to the date of the delivery deadline.

8 Code

8.1 Get turbidity in NTU

This function calculates the turbidity in NTU from the input sample value. The function takes the unsigned 32-bit integer sample value and a pointer to the float variable ntu as the input parameters. The 2 floats variables are declared; voltage and coeff_c, which store the input voltage and the turbidity calculation coefficient.

To convert this voltage value to millivolts, the code multiplies it by a reference voltage of 5000 mV and divides it by the ADC resolution. This conversion is important to normalize the input voltage value and make it easier to compare with pre-defined voltage levels, which are also normalized in millivolts.

The input voltage is checked against different voltage levels and update the coeff_c value accordingly using an if-else ladder. These voltage levels and corresponding coefficients are defined as constants in the code.

The turbidity value in NTU is then calculated using the formula *ntu = (TURBIDITY_ADC_VOLTAGE_MAX - voltage) / coeff_c. The resulting turbidity value is stored in the ntu variable, which is passed as a pointer to the function.

The final 2 if statements are used to set the ntu value to either the minimum or maximum values if the input voltage is out of range. The function returns 0 if executed successfully.

```c
int8_t turbidity_get_ntu ( uint32_t sample, float * ntu )
{
    float voltage;
    float coeff_c;

    voltage = (float)sample;
    voltage *= TURBIDITY_VREF_5000_mV;
    voltage /= TURBIDITY_ADC_RESOLUTION;
    printf("Input voltage: %fmV\r\n", voltage);

    coeff_c = TURBIDITY_NTU_CALC_FACT_0;

    if ( voltage > TURBIDITY_VOLTAGE_LEVEL_1 )
    {
        coeff_c = TURBIDITY_NTU_CALC_FACT_1;
    }

    if ( voltage > TURBIDITY_VOLTAGE_LEVEL_2 )
    {
        coeff_c = TURBIDITY_NTU_CALC_FACT_2;
    }

    if ( voltage > TURBIDITY_VOLTAGE_LEVEL_3 )
    {
        coeff_c = TURBIDITY_NTU_CALC_FACT_3;
    }

    if ( voltage > TURBIDITY_VOLTAGE_LEVEL_4 )
    {
        coeff_c = TURBIDITY_NTU_CALC_FACT_4;
    }

    *ntu = TURBIDITY_ADC_VOLTAGE_MAX;
    *ntu -= voltage;
    *ntu /= coeff_c;

    if ( voltage > TURBIDITY_ADC_VOLTAGE_MAX )
    {
        *ntu = TURBIDITY_NTU_MIN;
    }

    if ( voltage < TURBIDITY_ADC_VOLTAGE_MIN )
    {
        *ntu = TURBIDITY_NTU_MAX;
    }

    return 0;
}
```

Figure 7 – Turbidity calculation code [4]

8.2 Header file set up for Ubidots STEM

For HydroSystem to connect to Ubidots STEM, the Wi-Fi and Cloud broker credentials header files need to be set up.

The Wi-Fi credentials set up is simple, all that's needed is the SSID (Service Set Identifier) "hydrosystem" and password "12345678" of the mobile hotspot. These credentials inform HydroSystem which network it is connecting to and gives it the necessary information to do so, which are found inside the header guard.

```
#ifndef INC_WIFI_CREDENTIALS_H_
#define INC_WIFI_CREDENTIALS_H_

#define SSID     "hydrosystem"
#define PASSWORD "12345678"


#endif /* INC_WIFI_CREDENTIALS_H_ */
```

Figure 8 – Wi-Fi Credentials header file

The cloud broker credentials are slightly more complex; it requires "CloudBroker_HostName", "CloudBroker_Port", "CloudBroker_Username", "CloudBroker_Password" and "CloudBroker_ClientID".

"CloudBroker_HostName" specifies the domain name of the MQTT broker, which is "industrial.api.ubidots.com".

"CloudBroker_Port" specifies the port number used for the MQTT broker communication. HydroSystem isn't transmitting any sensitive data, therefore the port is set to "1883". "1883" is the default port for unencrypted communication and faster than port "8883", which is used for secure communication making it slower and bigger transmission packets.

"CloudBroker_Username" specifies the API key required to authenticate with the MQTT broker, "BBFF-IPRUyfG3Nh7zGxoYOO7qRyF9HUgWrd" is the API key to connect to HydroSystem's online dashboard.

“CloudBroker_Password” specifies the password required to authenticate with the MQTT broker. For Ubidots an empty string is all that’s required.

“CloudBroker_ClientID” specifies the unique identifier for the MQTT client that is connecting to the broker. Ubidots uses the Device ID to identify the client.

```
#ifndef INC_CLOUDBROKERCREDENTIALS_H_
#define INC_CLOUDBROKERCREDENTIALS_H_

/*
 * Add host name DNS here
 * For Ubidots use "industrial.api.ubidots.com"
 */
#define CloudBroker_HostName "industrial.api.ubidots.com"

/*
 * Port = "1883" for unencrypted communication
 */
#define CloudBroker_Port "1883"

/*
 * For Ubidots use your API key
 */
#define CloudBroker_Username "BBFF-IPRUyfG3Nh7zGxoYOO7qRyF9HUgWrd"

/*
 * For Ubidots use an empty password ""
 */
#define CloudBroker_Password ""

/*
 * For Ubidots use your Device ID
 */
#define CloudBroker_ClientID "64216795aaee4c001076ea93"


#endif /* INC_CLOUDBROKERCREDENTIALS_H_ */
```

Figure 9 – CloudBrokerCredentials header file

8.3 Connecting to Ubidots STEM

With the header files all set up, HydroSystem now has the sufficient information to connect to Ubidots STEM. HydroSystem connects to the MQTT broker and communicates over the mobile hotspot connection using the TCP/IP protocol.

HydroSystem initializes the configuration of the MQTT broker credentials and the Wi-Fi network interface, followed by retrieving the IP address and MAC address of the microcontroller. HydroSystem attempts to connect to the MQTT broker using the TCP/IP socket. It creates a network socket, sets the socket options, and opens a connection to the broker. On a successful connection, a message is printed to the console and initializes the MQTT client using the MQTTClientInit() function.

The MQTT client structure contains function pointers to read and write data to the network socket, which are defined as network_rd() and network_wr() respectively. These functions read and write data to the MQTT broker.

8.3.1 TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) is the communication protocol which enables communication between devices over the internet and other networks.

TCP the Transmission Control Protocol ensures that the data transmitted over the network. The protocol breaks the data into packets and adds a header to each packet with details relating to the data, source and destination IP addresses, packet length, and a sequence number.

IP the Internet Protocol facilitates the delivery of data packets across a network. Devices on the network are assigned a unique IP address. This unique IP address is used to identify devices on the network. Devices communicate with each other by sending data packets to the IP address of the destination device. IP ensures that data packets are routed to the correct device and that the communication is arriving to the correct device.

```
//Initialise MQTT broker structure
//Fill in this section with MQTT broker credentials from header file
MQTT_Config.HostName = CloudBroker_HostName;
MQTT_Config.HostPort = CloudBroker_Port;
MQTT_Config.ConnSecurity = "0";   //plain TCP connection with no security
MQTT_Config.MQUserName = CloudBroker_Username;
MQTT_Config.MQUserPwd = CloudBroker_Password;
MQTT_Config.MQClientId = CloudBroker_ClientID;

//Initialise WiFi network
if (net_init(&hnet, NET_IF, (wifi_net_if_init)) != NET_OK) {
    printf("\n\rError");
}
else {
    printf("\n\rOK");
}
HAL_Delay(500);

printf("\n\rRetrieving the IP address.");

if (net_get_ip_address(hnet, &ipAddr) != NET_OK) {
    printf("\n\rError 2");
}
else
{
  switch(ipAddr.ipv) {
    case NET_IP_V4:
      printf("\n\rIP address: %d.%d.%d.%d\n\r", ipAddr.ip[12], ipAddr.ip[13], ipAddr.ip[14], ipAddr.ip[15]);
      break;
    case NET_IP_V6:
    default:
        printf("\n\rError 3");
  }
}

if (net_get_mac_address(hnet, &macAddr) == NET_OK) {
    printf("\n\rMac Address: %02x:%02x:%02x:%02x:%02x:%02x\r\n",
            macAddr.mac[0], macAddr.mac[1], macAddr.mac[2], macAddr.mac[3], macAddr.mac[4], macAddr.mac[5]);
}
```

Figure 10 – Initializing Ubidots STEM connection code

```
printf("Connecting to MQTT Broker\r\n\n");
//Create network socket
ret = net_sock_create(hnet, &socket, NET_PROTO_TCP);
if (ret != NET_OK)
{
  printf("\n\rCould not create the socket.\r\n");
}
else
{
    ret |= net_sock_setopt(socket, "sock_noblocking", NULL, 0);
}

ret = net_sock_open(socket, MQTT_Config.HostName, 1883, 0);
if (ret != NET_OK)
  {
    printf("\n\rCould not open the socket.");
    HAL_Delay(1000);
  }
else {
    printf("\n\rConnected to server");
    HAL_Delay(1000);
}

network.my_socket = socket
network.mqttread = (network_rd);
network.mqttwrite = (network_wr);

MQTTClientInit(&client, &network, MQTT_CMD_TIMEOUT, mqtt_send_buffer, MQTT_SEND_BUFFER_SIZE,
   mqtt_read_buffer, MQTT_READ_BUFFER_SIZE);

/* MQTT connect */
options.clientID.cstring = MQTT_Config.MQClientId;
options.username.cstring = MQTT_Config.MQUserName;
options.password.cstring = MQTT_Config.MQUserPwd;

HAL_Delay(1000);

ret = MQTTConnect(&client, &options);
if (ret != 0)
{
  printf("\n\rMQTTConnect() failed: %ld\n", ret);
}
else
{
  printf("\n\rConnected to MQTT Broker");
  HAL_Delay(1000);
}
HAL_Delay(1000);
```

Figure 11 – Connecting to MQTT broker code

8.4 Publishing Turbidity (NTU) to Ubidots STEM

MQTTYield(&client, 500); is the function used to process/send incoming and outgoing MQTT messages from/to Ubidots STEM. &client is the pointer to the MQTT client instance, with a timeout value of 500ms. The timeout value specifies how long the function should wait for incoming messages to arrive and outgoing messages to be sent before returning a 0 and allowing the code to execute.

The turbidity is obtained from the Turbidity sensor, it receives two bytes of data from the I2C bus and then converts it to a millivoltage reading. The millivoltage reading is then passed through the function called turbidity_get_ntu() which calculates the turbidity value in NTU from the millivoltage reading. The turbidity value is then published to the MQTT broker in the format of a JSON message. The NTU value is actively displayed on a gauge widget on HydroSystem's Ubidots STEM dashboard.

8.5 DC motor control

HAL_GPIO_WritePin (GPIOx [the port the pin is on], GPIO_Pin [the name of the pin], GPIO_PinState [logic level 1/0]) controls how the DC motor rotates or if its stationary. The enable pin, input pin 1 and input pin 2 are configured to a logic level of either 1 or 0. The enable pin must be set to a logic level of 1 for the motor to rotate. Once the enable pin is set to 1, the logic levels of input pins 1 and 2 can be configured as shown below to rotate the DC motor clockwise, anti-clockwise or stop rotating.

Input 1 logic level	Input 2 logic level	Motor state
0	0	Motor Off
1	0	Clockwise/forwards
0	1	Anti-clockwise/backwards
1	1	Motor off due to short circuit

The DC motor rotates clockwise or anticlockwise with respect to the NTU level present in the water. An if statement regarding the NTU level detected is used to dictate if the DC motor rotates clockwise, anti-clockwise or is stationary.

```
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */

    /*
     * Need to yield to test for a received publish message from broker
     */
    ret = MQTTYield(&client, 500);


    HAL_I2C_Master_Receive(&hi2c3, 0x9B, buffer, 2, 1000);
    sample = (((uint16_t)(buffer[0]))<<8) + buffer[1];
    voltage_mv = 5000 * sample/4096;
    printf("Sample: 0x%04x %lu decimal, voltage %lumV\r\n", sample, sample, voltage_mv);
    turbidity_get_ntu(sample, &turbidity);
    printf("NTU Reading: %f\r\n\n", turbidity);


    sprintf(speedMsg, "{\"Turbidity\": {\"value\": %f}}", turbidity);
    printf("%s\n\r", speedMsg);
    memset(&mqmsg, 0, sizeof(MQTTMessage));
    mqmsg.qos = QOS0;
    mqmsg.payload = (char *) speedMsg;
    mqmsg.payloadlen = strlen(speedMsg);
    MQTTPublish(&client, "/v1.6/devices/hydrosystem", &mqmsg);
    HAL_Delay(10000);

    if(turbidity > 10){
        printf("Feeding plants\r\n");
        printf("Motor opening feed hatch\r\n\n");
        HAL_GPIO_WritePin(IN1A_GPIO_Port, IN1A_Pin, 1);
        HAL_GPIO_WritePin(IN2A_GPIO_Port, IN2A_Pin, 0);
        HAL_GPIO_WritePin(ENABLE_GPIO_Port, ENABLE_Pin, 1);//rotates motor left
        HAL_Delay(5000);
    }else{
        printf("Motor closing feed hatch\r\n\n");
        HAL_GPIO_WritePin(IN1A_GPIO_Port, IN1A_Pin, 0);
        HAL_GPIO_WritePin(IN2A_GPIO_Port, IN2A_Pin, 1);
        HAL_GPIO_WritePin(ENABLE_GPIO_Port, ENABLE_Pin, 1);//rotates motor right
    }
  }
  /* USER CODE END 3 */
}
```

Figure 12 – Publishing turbidity to Ubidots STEM [5]

8.6 Ubidots STEM Events – Email notifications

The email notification event is created by monitoring the Turbidity variable, which is published to HydroSystems' Ubidots STEM Device. To create the event the triggers, actions and settings needs to be set up correctly.

8.6.1 Trigger – Value based

The trigger consists of 4 components:

- a variable (Turbidity)
- a condition (Greater than)
- a Trigger value (8)
- a Trigger after (0 minutes)

Ubidots STEM awaits the turbidity variable to be assigned a value greater than the trigger value of 8. With the value of turbidity being greater than the trigger value, the action (email notification) will be performed with no delay as the trigger after value is assigned to 0 minutes.

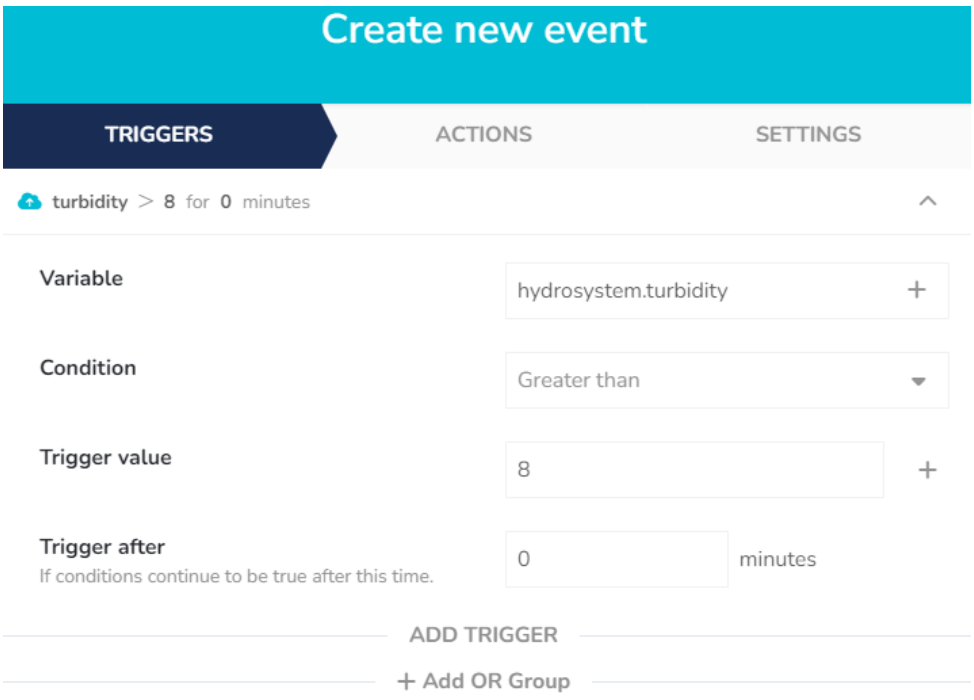


Figure 13 – Events triggers settings

8.6.2 Actions – Send email

The action consists of 4 sections:

- To (Email recipient)
- Email subject
- Message (The body of the email)
- Repeat action (resend the email)

The user's email is entered into the "To" text box, informing Ubidots STEM what email address to send the message to.

The subject is the title of the email (NTU Threshold break).

The Message is the body of the email informing the user that the NTU threshold has been broken. The message includes components which pull in the variable name, trigger value and the trigger timestamp automatically.

Repeat action allows the triggered event to be preform the action multiple times. HydroSystem doesn't require repeat action due to it being a prototype.

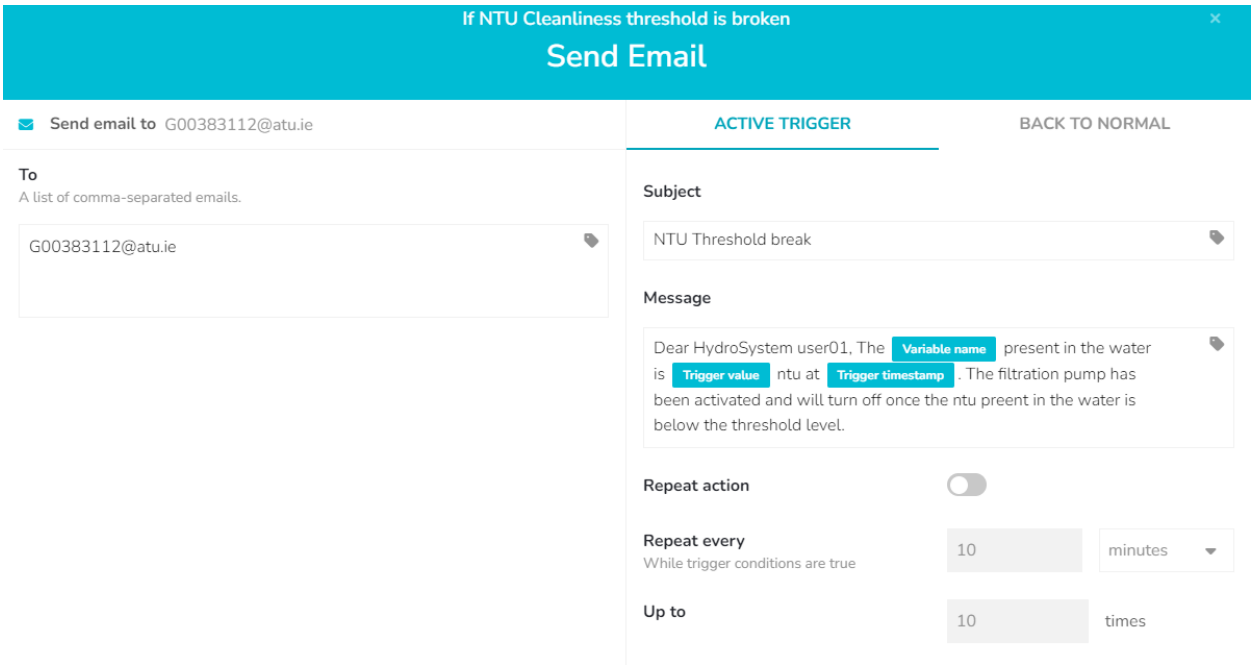


Figure 14 – Events Actions email set up

8.6.3 Settings – Send email

The settings consist of 3 sections:

- Event name
- Description
- Active windows

Event name allows the user to distinguish which event has occurred.

Description briefly describes the purpose of the event and what happens when its triggered.

Active windows are the periods where the Event can be triggered. This facilitates the user to specify the day(s) of the week, the start and end times during which the Event can be triggered.

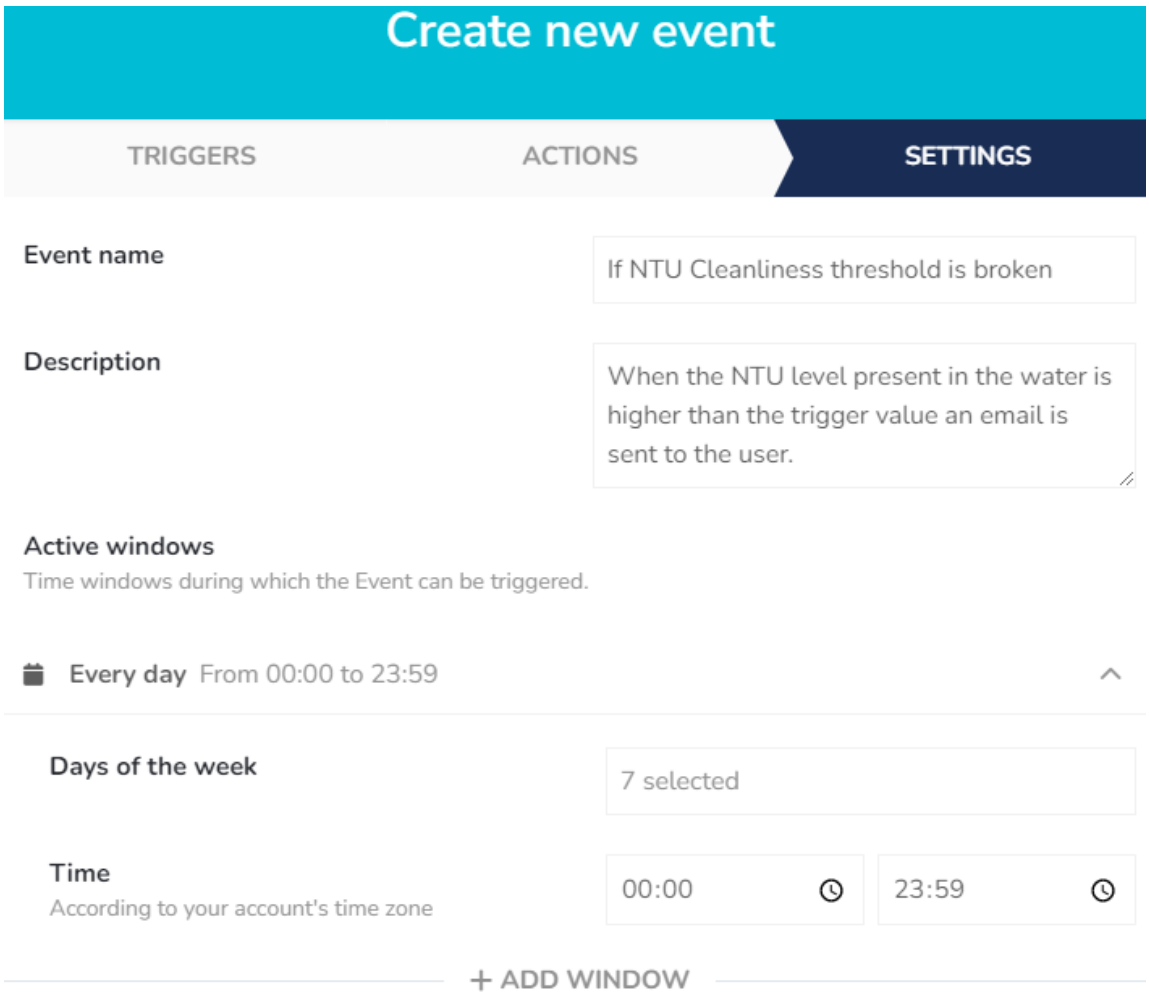


Figure 15 – Events Settings

9 Problem solving

Throughout working on HydroSystem I came across multiple problems, which I will discuss below.

9.1 Turbidity sensor

9.1.1 Components delivery time

At first I was converting an AC signal to DC to calculate the NTU. I needed a 10nF capacitor, but the college store had none left. In December there were supplier issues with extremely long delivery dates and components arriving late, which I had already experienced.

This supplier issues led me to look at an alternative way to detect the NTU level using the turbidity sensor. The solution I came to was setting up a I2C master slave interrupt to calculate the NTU.

9.1.2 Turbidity sensor initially set up on 3.3V

I had the turbidity sensor wired up to the ST B-L475E-IOT01A micro controller and all the code written correctly to display the NTU level present in the water. The sensor was returning a NTU value of 4000NTU which was the max reading the sensor will read.

From analyzing the datasheet and the input voltage I concluded there had to have been a hardware issue. I discovered the sensor can run on 3.3V and 5V and I needed it to be configured to 5V.

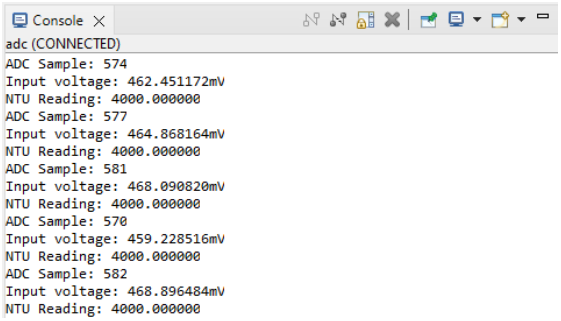
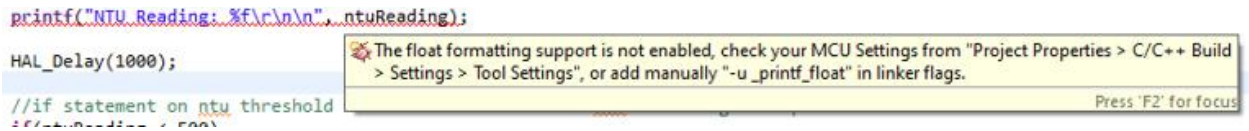


Figure 16 – HydroSystems previous AC-DC console

To achieve 5V configuration I had to desolder a surface mount resistor and solder it back on 2 pads right of its original position.

9.1.3 Printf floating formatting support

STM32CubeIDE supports float formatting when using printf(), but in early April HydroSystem stopped compiling due to an error.



To fix the formatting error the “Use float with printf from newlib-nano (-u_printf_float)” checkbox needed to be re-ticked.

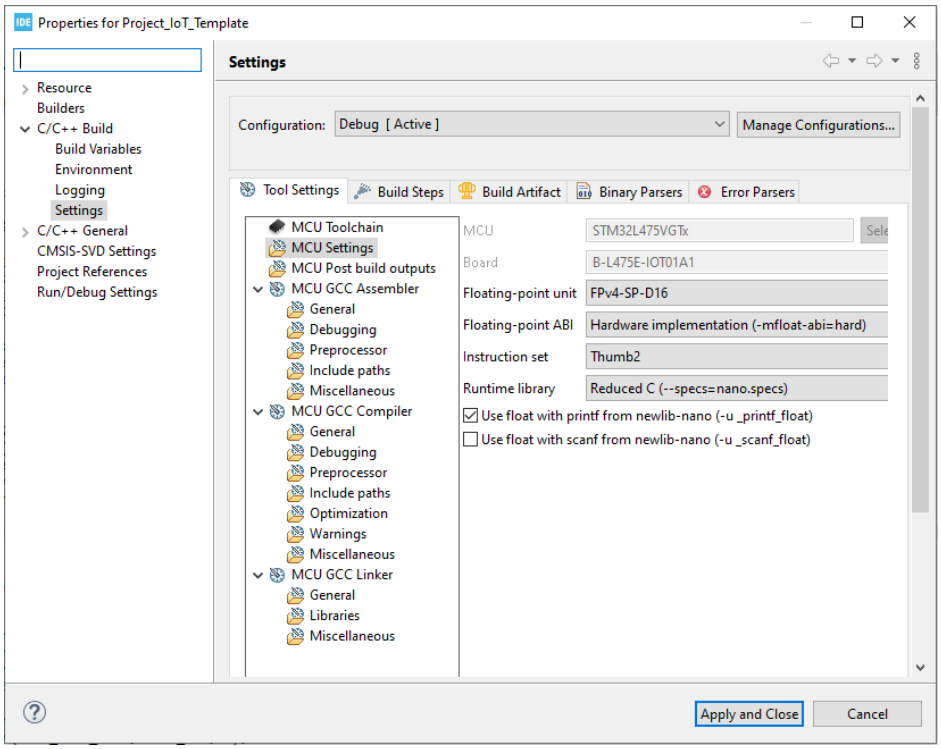


Figure 17 – Project properties

9.1.4 Project not compiling

During the middle of April HydroSystem stopped compiling again. From checking the application code, project structure and settings, I noticed the root folder and majority of the header files were removed from the Paths and Symbols Project Properties folder.

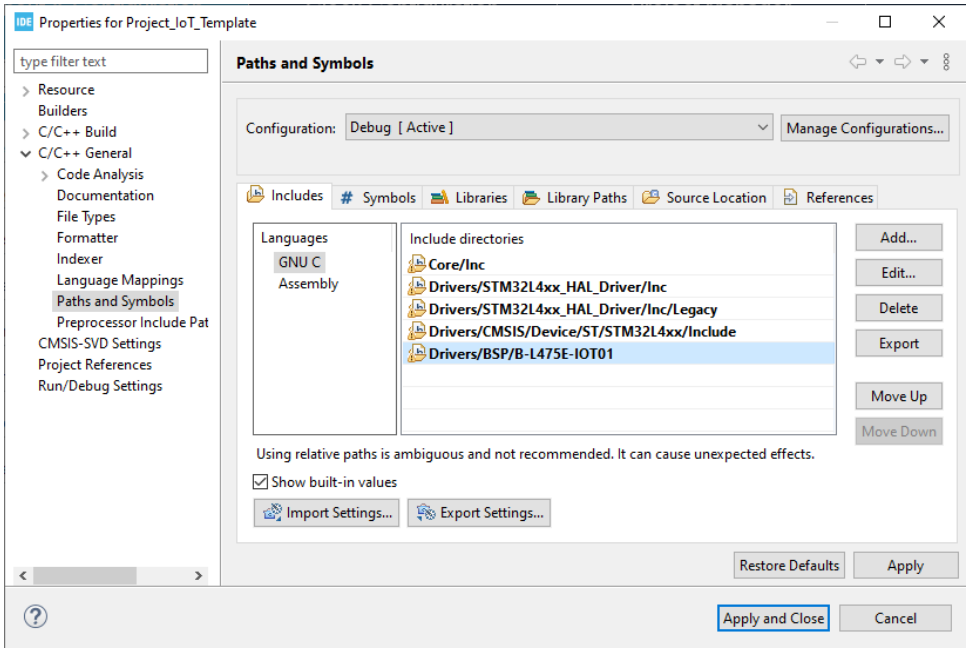


Figure 18 - Project properties

The root folder and all the missing header files had to be manually re-added 1 by 1 into the Includes GNU C section of the Paths and Symbols section of the project properties.

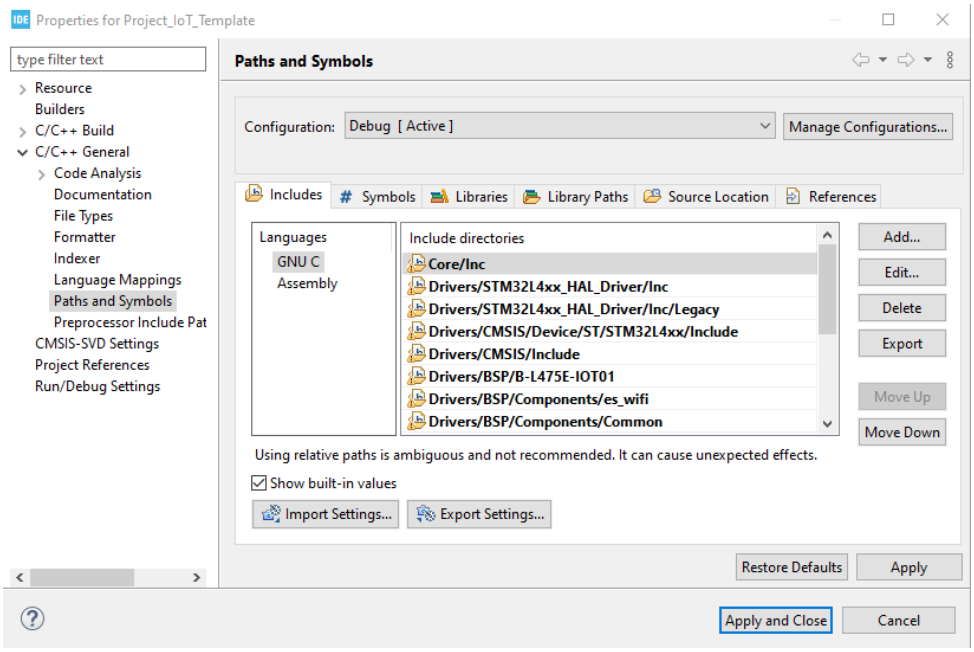


Figure 19 - Project properties

Annoyingly I discovered the root folder error would resurface every time the .ioc (pinout & configuration) file was altered. I informed my supervisor Niall about this issue, which he was

previously aware of and had already emailed STMicroelectronics about this bug. To this date the bug still hasn't been fixed and meaning the above needs to be done manually after each .ioc file configuration.

Similar to the header file error, the symbols of the project properties also disappeared.

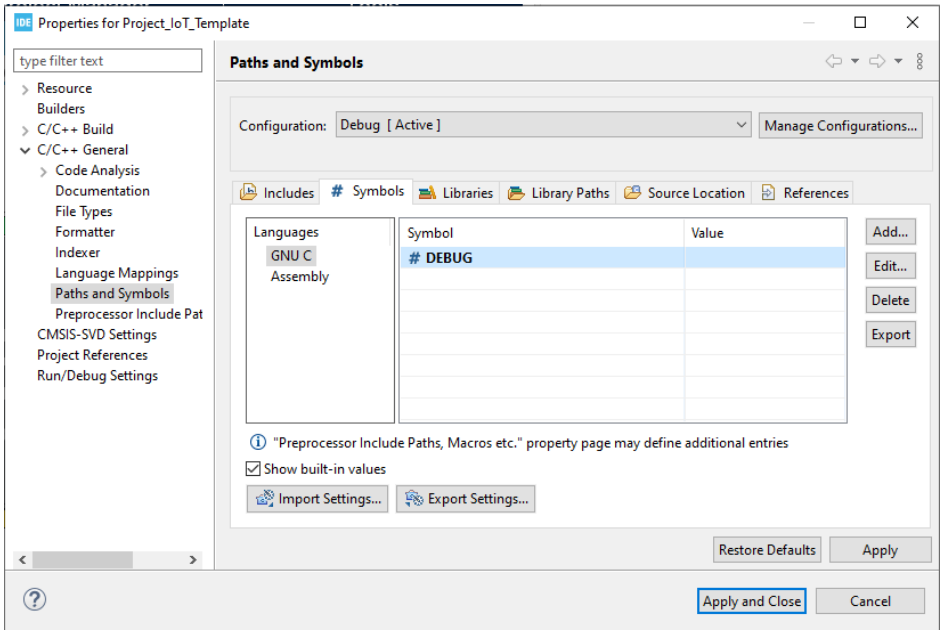


Figure 20 - Project properties

This led to having to manually re added them all back in into Symbols GNU C section of the Paths and Symbols section of the project properties.

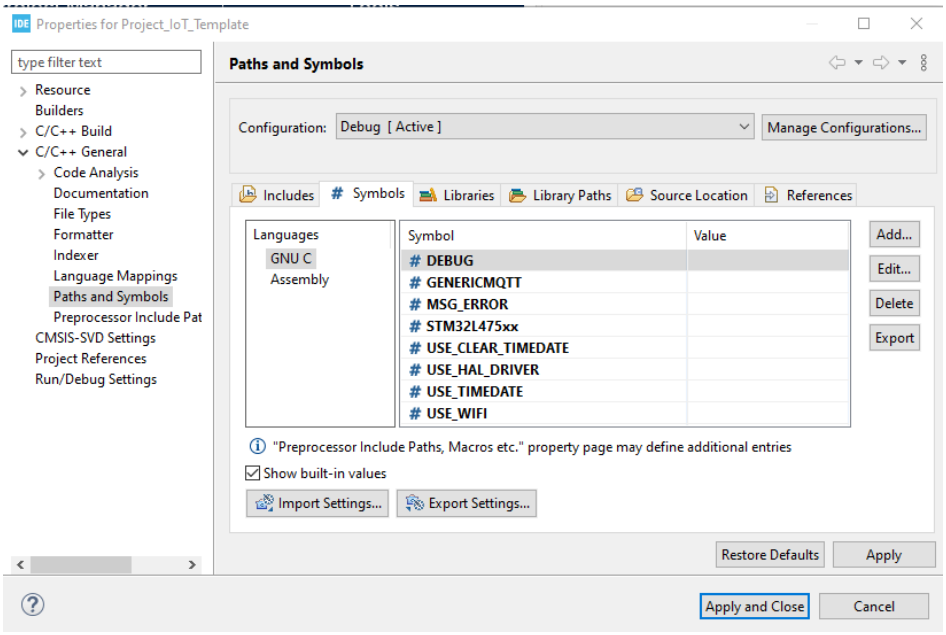


Figure 21 - Project properties

These 2 sections being refilled with the correct folders and symbols fixed the compiler error. Unlike the root folder issue, I am unaware as of how and why these files and symbols were removed from the project properties.

10 Ethics

HydroSystem is a nature-based project and aims to grow hydroponic plants in a healthier environment. HydroSystem reduces the environmental impact compared to traditional farming methods.

HydroSystem uses nutrient rich water instead of soil to provide the necessary nutrient to the roots of the plant. Compared to the traditional agricultural methods HydroSystem is more efficient when it comes to water usage and eliminates the use of synthetic fertilizers and pesticides which contribute to greenhouse gas emissions.

The use of chemicals in traditional farming is a big contributor towards to global warming. Pollution to water sources and soil further contribute to global warming. Chemical's runoff into surrounding rivers, lakes, and the ocean, which pollute the worlds water sources and soil.

If farmers would implement HydroSystem into the growing of crops such as strawberry plants which are a type of hydroponic[6], the impact of global warming would reduce further. This is due to the reduced use of tractors and other fossil fuel-based machinery.

11 Future Developments

This version of HydroSystem is prototype and very small compared to what is needed for commercial use. This leaves plenty of room for upscaling and improvement.

Sourcing the correct pump would be the first development to the prototype to demonstrate the full project working.

An upgrade I'd make, would be to either use multiple turbidity sensors or one that would cover a greater volume. This would improve the accuracy of the NTU level reported in small ponds, but also provide greater confidence of the NTU value leading to HydroSystem being able to function bug free in lakes.

It would also be more feasible to have the nutrients supplied with the filtered water. When the NTU level in the water is below the threshold an additional pump could be added to blow in the nutrients alongside the filtered water directly into the roots of the plants.

12 Conclusion

HydroSystem successfully detects the NTU present in water. Once the turbidity cleanliness threshold is broken, HydroSystem is aware it should turn on the filtration pump. Unfortunately, the pump that arrived was the incorrect pump and with it arriving so late, the filtration pump aspect can't be demoed for now. The email notification event informs the user that the turbidity cleanliness threshold has been broken and simulates the filtration pump activating as if it were part of HydroSystem.

HydroSystem will clean ponds and grow hydroponic plants in a healthier and more efficient manner than the techniques that are used in the past and present, with the addition of a filtration pump in the near future.

13 References

[1] "B-L475E-IOT01A - STM32L4 Discovery kit IoT node, low-power wireless, BLE, NFC, SubGHz, Wi-Fi - STMicroelectronics." https://www.st.com/en/evaluation-tools/b-l475e-iot01a.html (accessed May 03, 2023).

[2] "Basics of Armatures - Motor Specialty Inc." https://motorspecialty.com/news/basics-of-armatures/ (accessed May 03, 2023).

[3] D. Namiot and M. Sneps-Sneppe, "Geofence and Network Proximity," *Internet of Things, Smart Spaces, and Next Generation Networking*, vol. 8121, pp. 117–127, 2013, doi: 10.1007/978-3-642-40316-3_11.

[4] MIKROE, "LibStock - Turbidity click," Dec. 29, 2022. https://libstock.mikroe.com/projects/view/4836/turbidity-click (accessed May 03, 2023).

[5] "Publish data to a Device (v1.6)." https://docs.ubidots.com/v1.6/reference/publish-data-to-a-device (accessed May 03, 2023).

[6] "How To Grow DIY Hydroponic Strawberries | Horticulture.co.uk." https://horticulture.co.uk/strawberries/hydroponic/ (accessed May 03, 2023).