EEE088F 2022

Documentation & Software

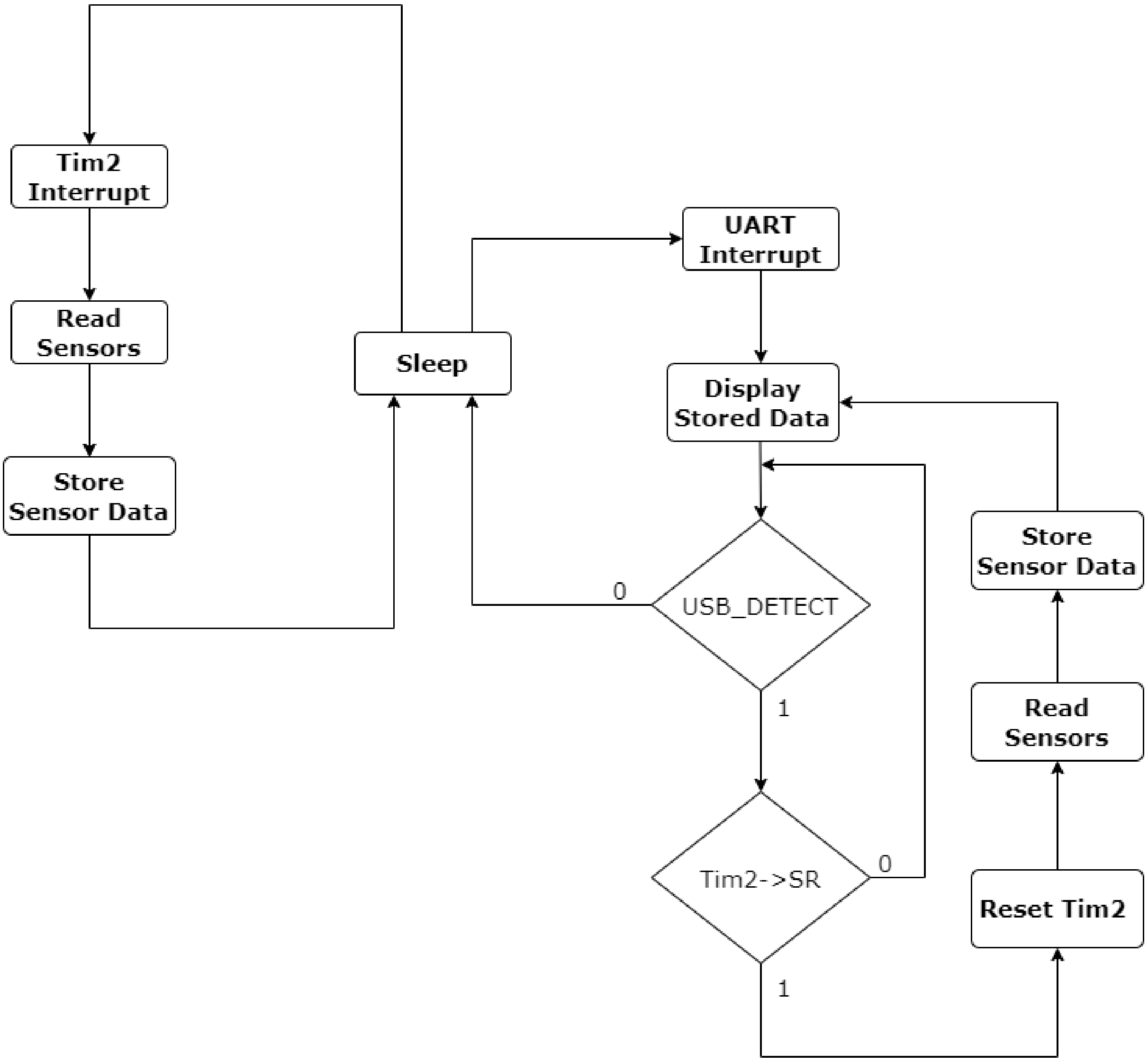
Autonomous Plant Monitoring HAT for Discovery Board with STM32FC051

JNSRYA006
TRFDEV001
CRGALE002

23/05/2022

Section A: Software

Q1 Code Structure [5]



*When the USB is plugged in, the device is in a high power polling state.

Figure 1: Code Block Diagram

Q2 Code Reference [10]

The following functions were used in the code to communicate and interface between the HAT, STMDiscovery Board and the computer.

Microcontroller Functions:

```
void HAL_TIM_PeriodElapsedCallback(TIM_handleTypeDef *htim)
```

Function that runs when Tim2 Interrupt is triggered

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
```

Function triggered by user typing "Hello HAT"

```
void debugPrintln(UART_HandleTypeDef *uarthandle, char_out[ ])
```

Function that improves output to screen

```
int main(void)
```

Main function

Sensor functions:

```
uint8_t DS18B20_Start (void)
```

Initialises the digital sensor

```
void DS18B20_Write (uint8_t data)
```

Takes a byte of data (as an argument) and writes it to the digital sensor

```
uint8_t read (void)
```

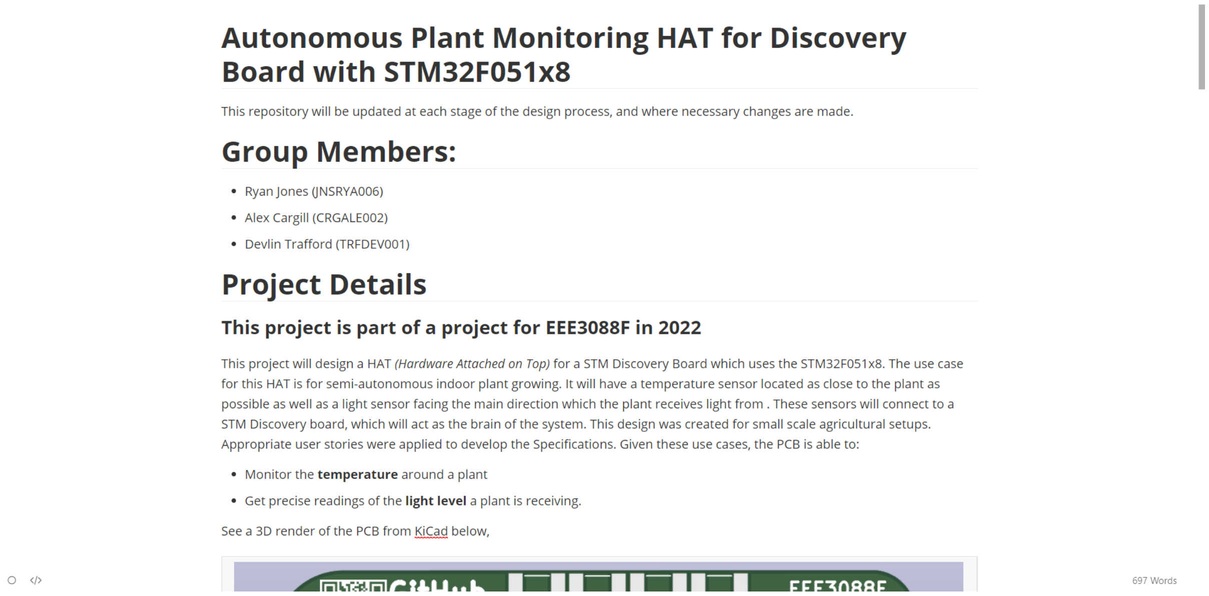
Returns a byte from the digital sensors built in memory

The Full Code Reference document is available on GitHub here

Section B: Learning documentation:

Q3 PCB Getting Started README: [10]

The README.md file can be found on the GitHub here



Autonomous Plant Monitoring HAT for Discovery Board with STM32F051x8

This repository will be updated at each stage of the design process, and where necessary changes are made.

Group Members:

- Ryan Jones (JNSRYA006)
- Alex Cargill (CRGALE002)
- Devlin Trafford (TRFDEV001)

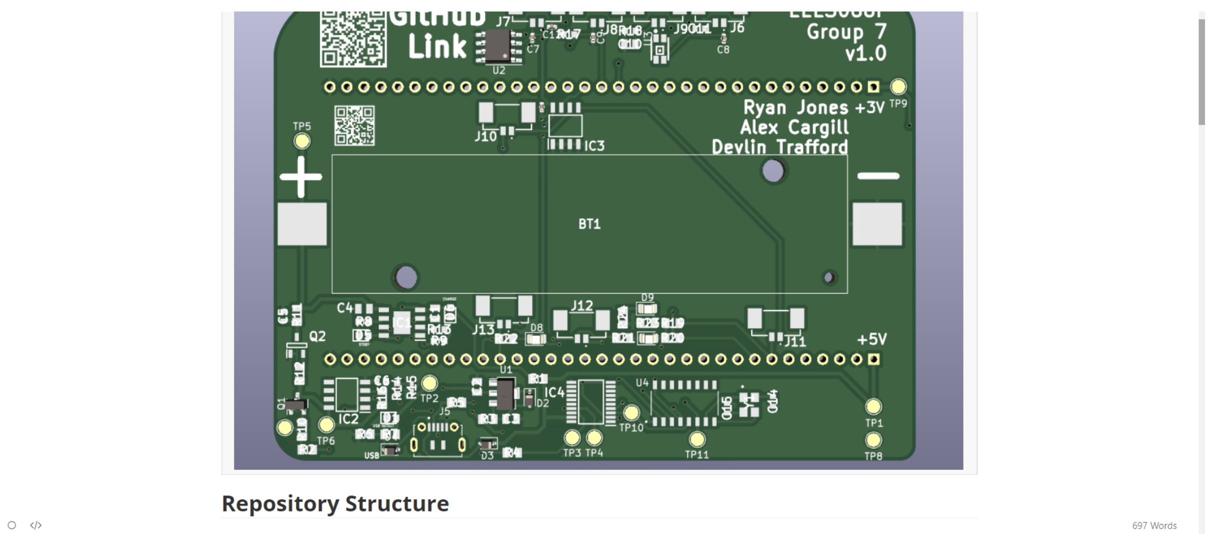
Project Details

This project is part of a project for EEE3088F in 2022

This project will design a HAT *(Hardware Attached on Top)* for a STM Discovery Board which uses the STM32F051x8. The use case for this HAT is for semi-autonomous indoor plant growing. It will have a temperature sensor located as close to the plant as possible as well as a light sensor facing the main direction which the plant receives light from . These sensors will connect to a STM Discovery board, which will act as the brain of the system. This design was created for small scale agricultural setups. Appropriate user stories were applied to develop the Specifications. Given these use cases, the PCB is able to:

- Monitor the **temperature** around a plant
- Get precise readings of the **light level** a plant is receiving.

See a 3D render of the PCB from KiCad below,



Repository Structure

Each sub-folder contains different aspects relating to different parts of the project

- ***PCB:*** The KiCad Project, Schematics, PCB & Layout and Project Libraries
- ***Firmware:*** Any software developed for the hardware or instructions on where to find relevant sofware in cases where it makes more sense to have this in its own dedicated repo
- ***Docs:*** Project documentation, Draft documents, ERC & DRC reports, project handins, schematic screenshot & documents and component datasheets
- ***Production:*** The final gerber files, BOM, Budget, or anything required by the fabrication houses
- ***Simulation:*** Any simulation files (SPICE) or design stage generated results (matlab or excel)
- ***CAD:*** Any 3D models or mechanical designs for enclosures or support

Hardware Requirements

- □ STMDiscovery Board with an STM32F051x8 µC
- □ 1 x MicroUSB to USB-A Cable
- □ 1x MiniUSB to USB-A Cable
- □ 8x Male to Female Jumpers

Software Requirements

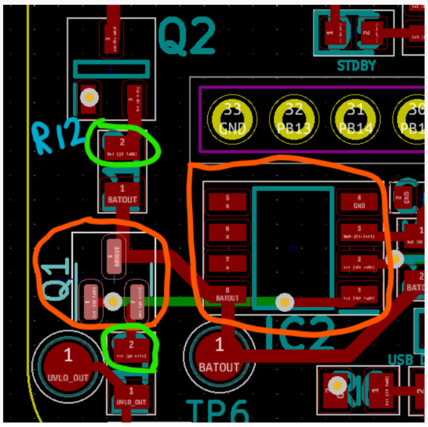
- □ Computer
  - *Windows, Linux or MACOS*
- □ STM32Cube IDE
  - [Download available online]()

- □ PuTTY

- □ PuTTY
  - [Download available online]()

How to short the UVLO circuit

- On the LHS of the board, remove resistor **R12 & R10** using pliers and twisting and pulling the resistors
- Q1 & IC1 (Circled in **Orange**) are the MOSFET & Op-Amp in the UVLO circuit that needs to be removed by soldering one end of a wire to the top pad (Pad 2) of R12 to the top pad (Pad 2) of R10 (Circled in **Green**)



How to connect the power module to the rest of the board

- On the top, RHS of the board, solder a wire between **TP9** *OR* the **+3V** pin pad and the right hand pin of **J7**
  - This will connect the power from the power module to the rest of the circuit
- If connected correctly, when a battery is inserted or a USB is connected, a **Red** diode (**D9**) will turn on to indicate the sensing and microcontroller sub-circuits are receiving power.

How to connect the PCB to the STMDiscovery Board

The following locations **on the PCB** need to be soldered to the male end of the jumper:

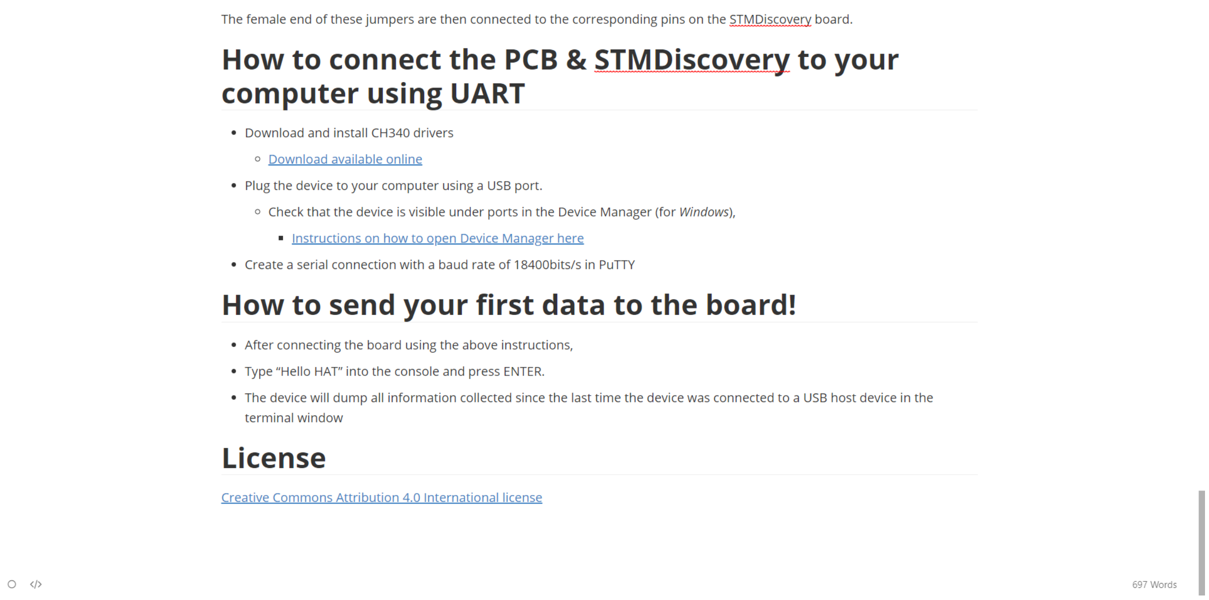
(*Refer to the KiCad PCB file [here]() for the exact location of these*)

- PA1
- PA2
- PA9
- PA10
- PB6
- PB7

The female end of these jumpers are then connected to the corresponding pins on the STMDiscovery board.

How to connect the PCB & STMDiscovery to your computer using UART

- Download and install CH340 drivers



The female end of these jumpers are then connected to the corresponding pins on the STMDiscovery board.

How to connect the PCB & STMDiscovery to your computer using UART

- Download and install CH340 drivers
  - Download available online
- Plug the device to your computer using a USB port.
  - Check that the device is visible under ports in the Device Manager (for *Windows*),
    - Instructions on how to open Device Manager here
- Create a serial connection with a baud rate of 18400bits/s in PuTTY

How to send your first data to the board!

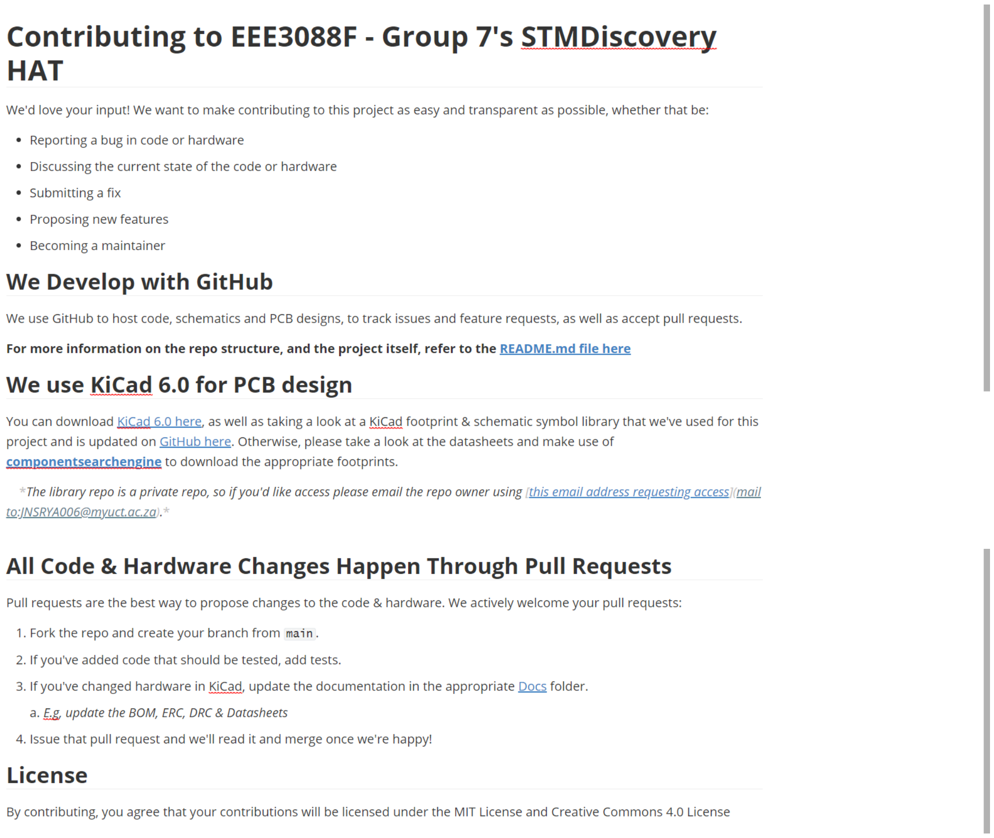
- After connecting the board using the above instructions,
- Type "Hello HAT" into the console and press ENTER.
- The device will dump all information collected since the last time the device was connected to a USB host device in the terminal window

License

Creative Commons Attribution 4.0 International license

Q4 CONTRIBUTING How-To Guide: [5]

The CONTRIBUTING.md file can be found on the GitHub here



Contributing to EEE3088F - Group 7's STMDiscovery HAT

We'd love your input! We want to make contributing to this project as easy and transparent as possible, whether that be:

- Reporting a bug in code or hardware
- Discussing the current state of the code or hardware
- Submitting a fix
- Proposing new features
- Becoming a maintainer

We Develop with GitHub

We use GitHub to host code, schematics and PCB designs, to track issues and feature requests, as well as accept pull requests.

For more information on the repo structure, and the project itself, refer to the README.md file here

We use KiCad 6.0 for PCB design

You can download KiCad 6.0 here, as well as taking a look at a KiCad footprint & schematic symbol library that we've used for this project and is updated on GitHub here. Otherwise, please take a look at the datasheets and make use of **componentsearchengine** to download the appropriate footprints.

The library repo is a private repo, so if you'd like access please email the repo owner using [this email address requesting access](mailto:JNSRYA006@myuct.ac.za).

All Code & Hardware Changes Happen Through Pull Requests

Pull requests are the best way to propose changes to the code & hardware. We actively welcome your pull requests:

1. Fork the repo and create your branch from `main`.
2. If you've added code that should be tested, add tests.
3. If you've changed hardware in KiCad, update the documentation in the appropriate Docs folder.
   a. *E.g, update the BOM, ERC, DRC & Datasheets*
4. Issue that pull request and we'll read it and merge once we're happy!

License

By contributing, you agree that your contributions will be licensed under the MIT License and Creative Commons 4.0 License