UNIVERSITY *of York*

Department of Electronic Engineering

Design, Construction and Test Assessment Report

Jan – May 2022

Year of Study: Second Year

Group Number: A2

Group Members: Connall Shurey
Wijdaan Masud Rizavi
Benjamin McCann
Nojus Jukna
Srikanth Jakka

Report by: Connall Shurey

Last Modified: April 26th, 2022

Table of Contents

1 Summary of Design, Construction and Testing of the Product

1.1 Product Concept

1.1.1 Main Aim

The aim of the Design, Construction and Test Assessment, is to develop a microprocessor-based digital multimeter capable of measuring:

- DC and AC voltages of up to +/- 10V with a frequency of up to 1MHz
- Currents up to +/- 100mA with a frequency of up to 1MHz
- Resistance from 1Ω to 1MΩ

The assessment is designed to give experience facing complex technical tasks where you do not currently have training in all the required technical areas. Completing the task requires researching how an STM32F4 microprocessor operates, how to program it, and how to implement analogue-to-digital conversions to take readings. It is also required to investigate how modern digital multimeters work and how we can use the STM32F4 microprocessor to create a multimeter to the above specifications. This report contains information on the process undergone to achieve this task.

1.1.2 Market Analysis

Market research shows that the digital multimeter market is flooded with multimeters for industrial use that can handle large measurements of up to 1000V and cost upwards of £25. The cost of these standard multimeters increases exponential as more features are implemented. Competing in such a market requires a high-end product that is difficult to provide in a time span of 5 months with limited experience designing measuring devices.

To make our product stand out, we are aiming to target the education sector. By providing a cheap product between £10 - £15, that matches the required standards for a classroom environment, we provide an attractive product for teachers and electronic hobbyists. The education sector is looking for multimeters that are cheap and do not require large measurement capabilities or extra features.

This is perfect for our product, voltages above 10V rarely appear in the classroom as it poses a safety risk to students and most classrooms are kept at close to room temperature. Considering these conditions, our multimeter does not require the expensive components necessary to tackle the large voltages or handle the extreme temperatures a multimeter may be subjected to in an industrial environment, allowing us to design a multimeter with an extremely low production cost.

1.1.3 Conceptual Product Design

Our digital multimeter will be designed around a STM32F4 microprocessor. Initial research indicates that the microprocessor has several in-built ADCs (analogue-to-digital converters) to convert analogue voltages to a digital reading. However, the maximum rated voltage of the ADC is below the 10V magnitude in our product specification. To combat this, the multimeter needs a voltage scaler and protection circuitry to prevent accidental damage to the ADC. With the voltage measuring hardware, considering the formula Voltage = Current * Resistance, the multimeter can then measure:

- Current by measuring voltage across a known resistor
- Resistance by measuring the resultant voltage from outputting a known current through a resistor

This allows the multimeter to obtain the 3 basic readings through voltage readings alone.

The conceptual hardware design showing the inputs and outputs for the microprocessor can be seen in **Figure 1**. It contains the functionality to take our 3 basic measurements and the voltage scaler, selector, and protection circuitry required for the microprocessor.

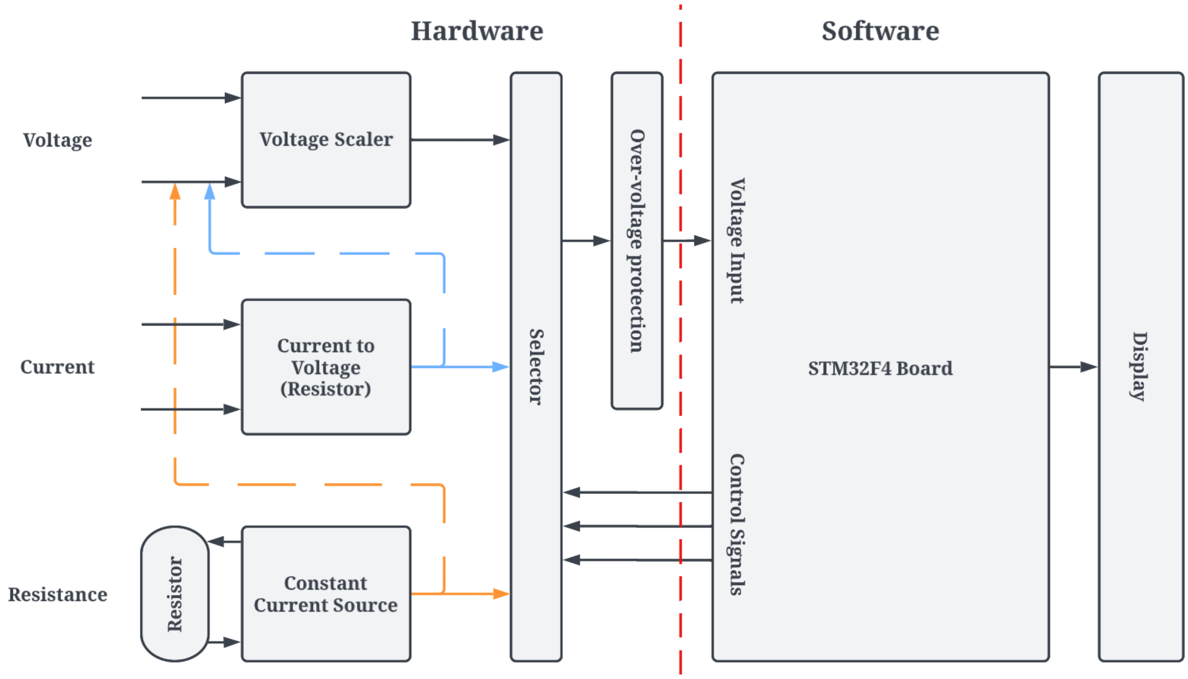

Figure 1 - Conceptual Digital Multimeter Hardware Design

The conceptual software design seen in **Figure 2**, uses the conceptual hardware design as a guide of our required microprocessor board inputs. The STM32F4 Board has a single programmable button for users which is used in the software design to cycle through measuring modes.

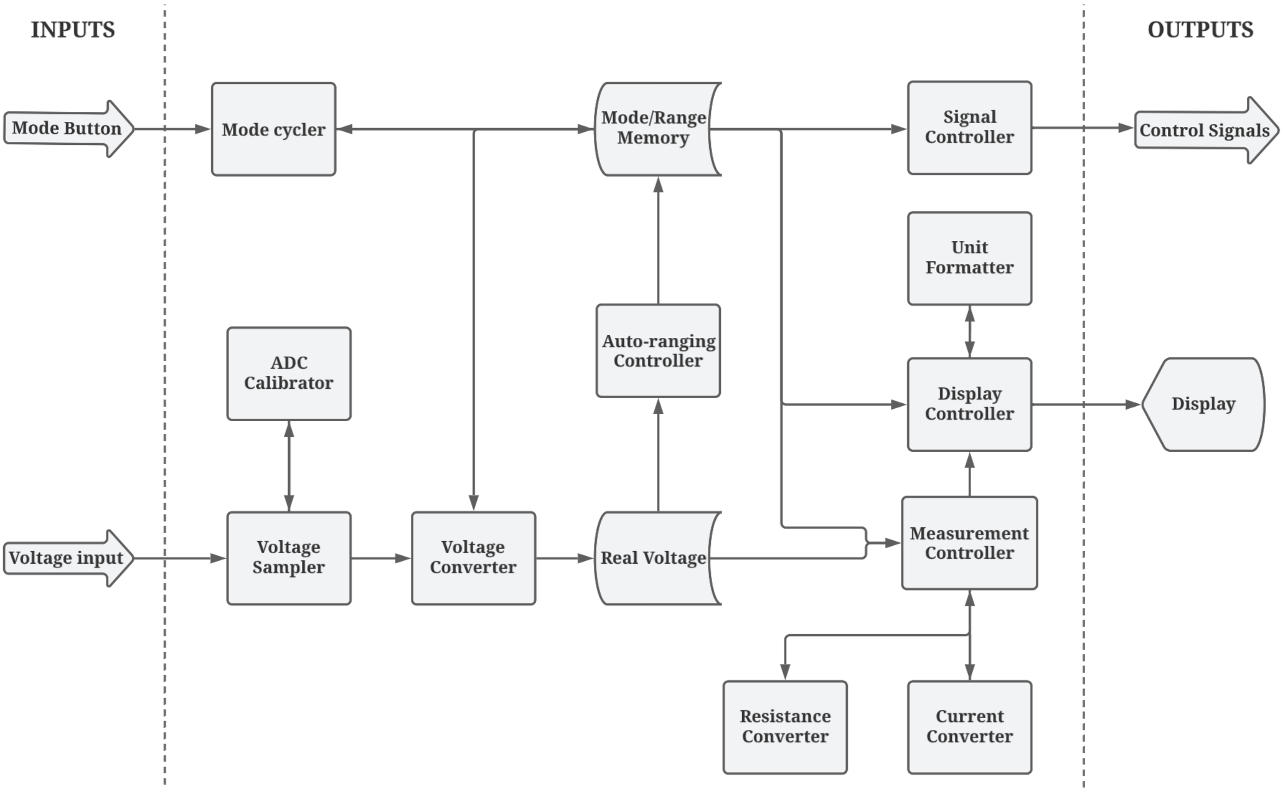

Figure 2 - Conceptual Digital Multimeter Software Design

An auto-ranging feature was included in our conceptual design, though not in the original specification. Without this feature users would be required to manually switch to the desired resolution of the measurement (V, mV, uV, etc). The STM32F4 Board only has one user input which is already used for cycling through modes and since auto-ranging can be achieved through software, we can reduce the number of components required in the device, further reducing the product cost. Auto-ranging is also an attractive feature to the education sector, as students may not have the knowledge required to correctly range a multimeter.

1.1.4 Work Breakdown

Figure 3 shows the Work Breakdown Structure (WBS) used to make large projects more manageable:

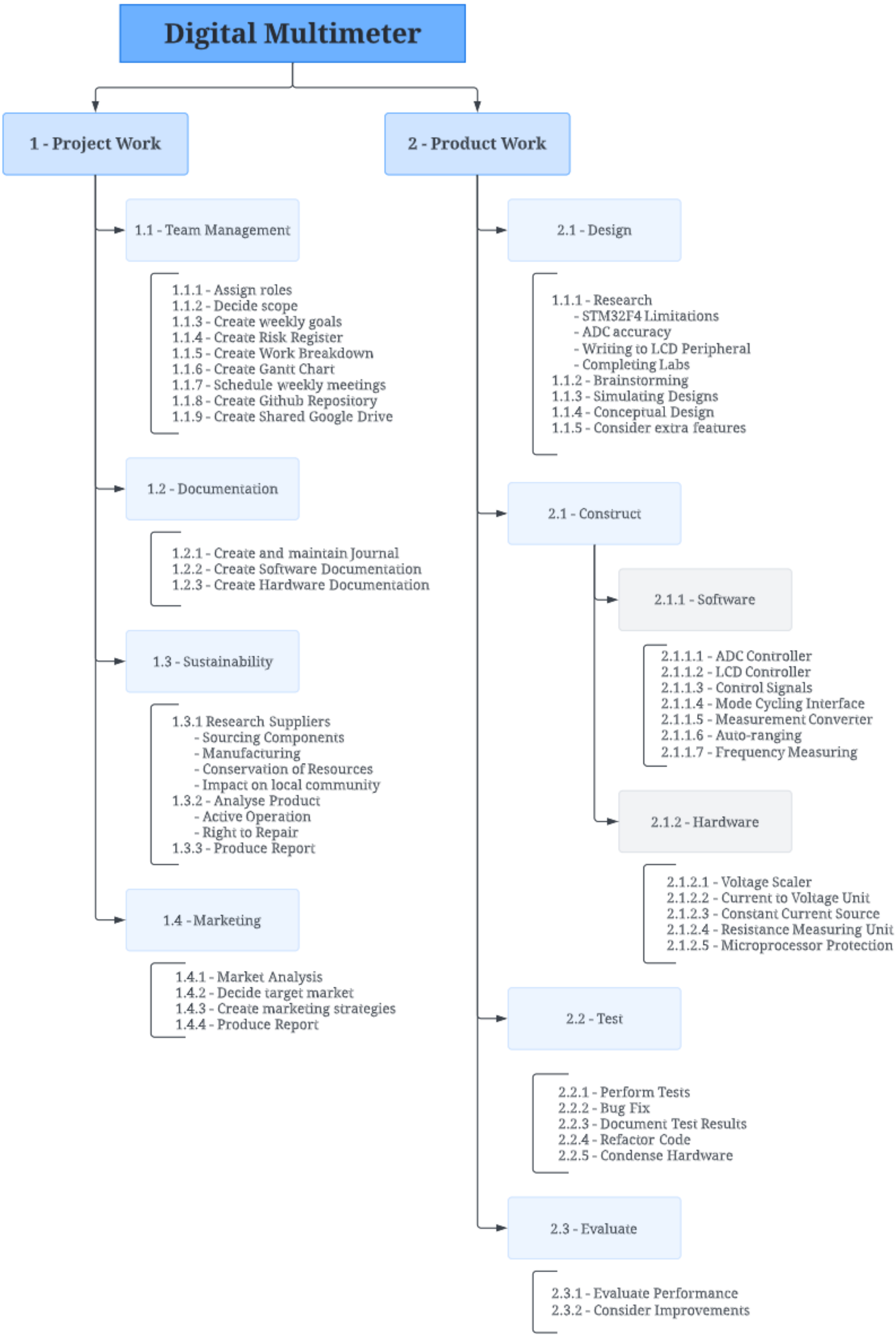

Figure 3 - Project Work Breakdown Structure

1.2 Description of Entire System

1.2.1 Hardware

The final hardware design can be in seen **Figure 4**. Using an analogue multiplexer, the device can switch between measuring the voltage across multiple sources. The resultant voltage is then scaled down from $\pm$ 10V and offset to 0 - 3V, the range the ADC (analogue-to-digital converter) operates in. Software then converts the signal from 0 - 3V back to $\pm$ 10V or $\pm$ 100mV depending on which ADC input we are measuring and prints the output to the LCD.

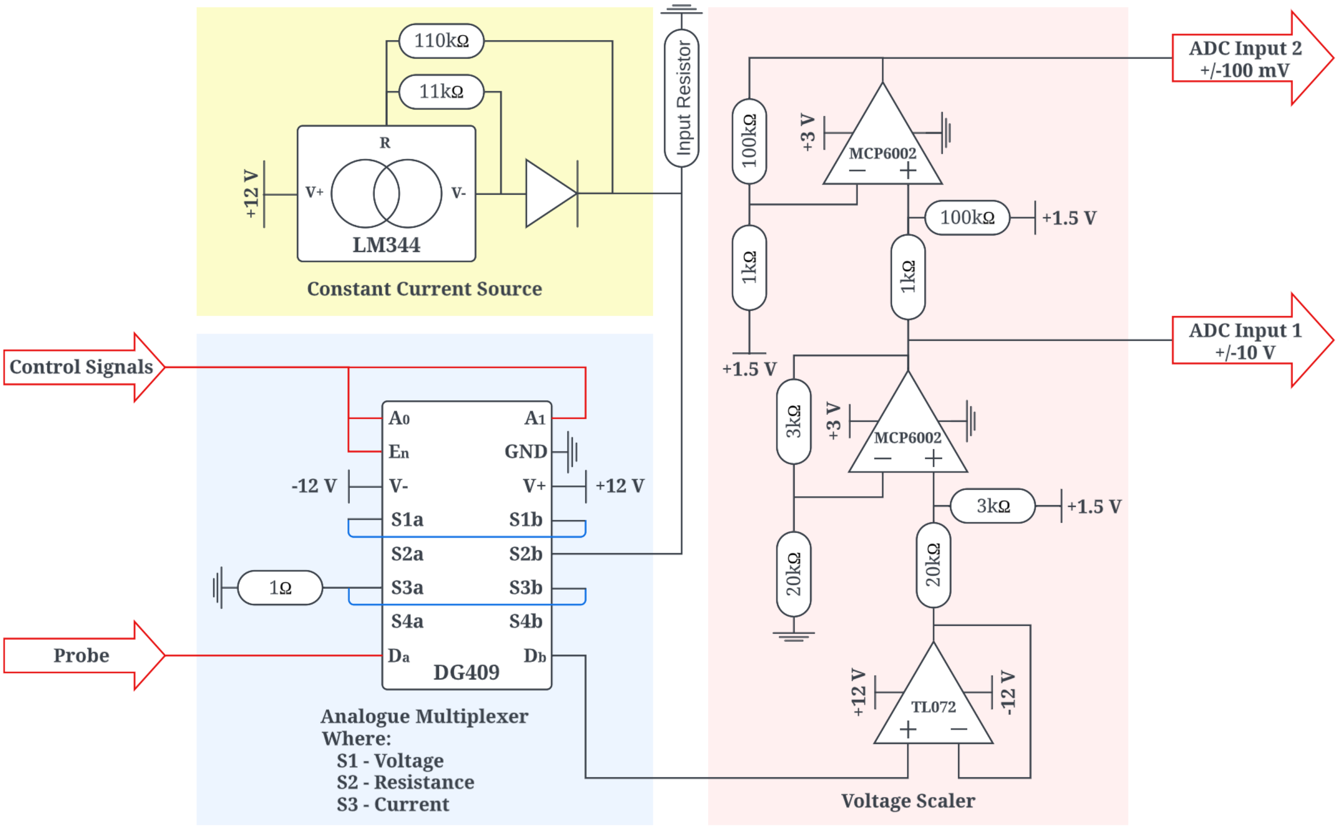

Figure 4 - Hardware Final Design

Measuring modes:

Voltage To measure voltage, the analogue multiplexer is set to S1 through control signals. The measuring circuit then outputs the scaled voltage between the probe input and ground.

Resistance To measure resistance, the analogue multiplexer is set to S2 through control signals. A constant current source outputs a known current through the Input Resistor. The measuring circuit then outputs the scaled voltage across the input resistor.

Current To measure current, the analogue multiplexer is set to S3 through control signals. Current is input through the input probe and sent through a 1Ω resistor. The measuring circuit then outputs the scaled voltage across the 1Ω resistor. Hardware specifically uses a 1Ω resistor as it has a small uncertainty, and we can save microprocessor power as the equation to find the current is:

$$Current = \frac{Voltage}{1\Omega}$$

As we are dividing by 1, software can skip the measurement conversion operation and save power.

1.2.2 Software

The Final Software Design shown in **Figure 5**, was designed to make use of the STM32s powerful ARM Cortex-M4 microprocessor, while also considering power consumption and design simplicity. Measurements are based on the microprocessors system tick interrupt that triggers every 500ms. Each time the SysTick Interrupt triggers, we take a measurement, convert it according to the mode, and print the result. It is possible to take measurements many times faster, doing so increases the multimeters accuracy by a negligible portion while largely increasing power consumption. Users change the measuring mode using the Mode Button which will update peripherals to match the selected mode. If no probe input is detected, the software will put the microprocessor into standby mode, one of the STM32s low-power states.

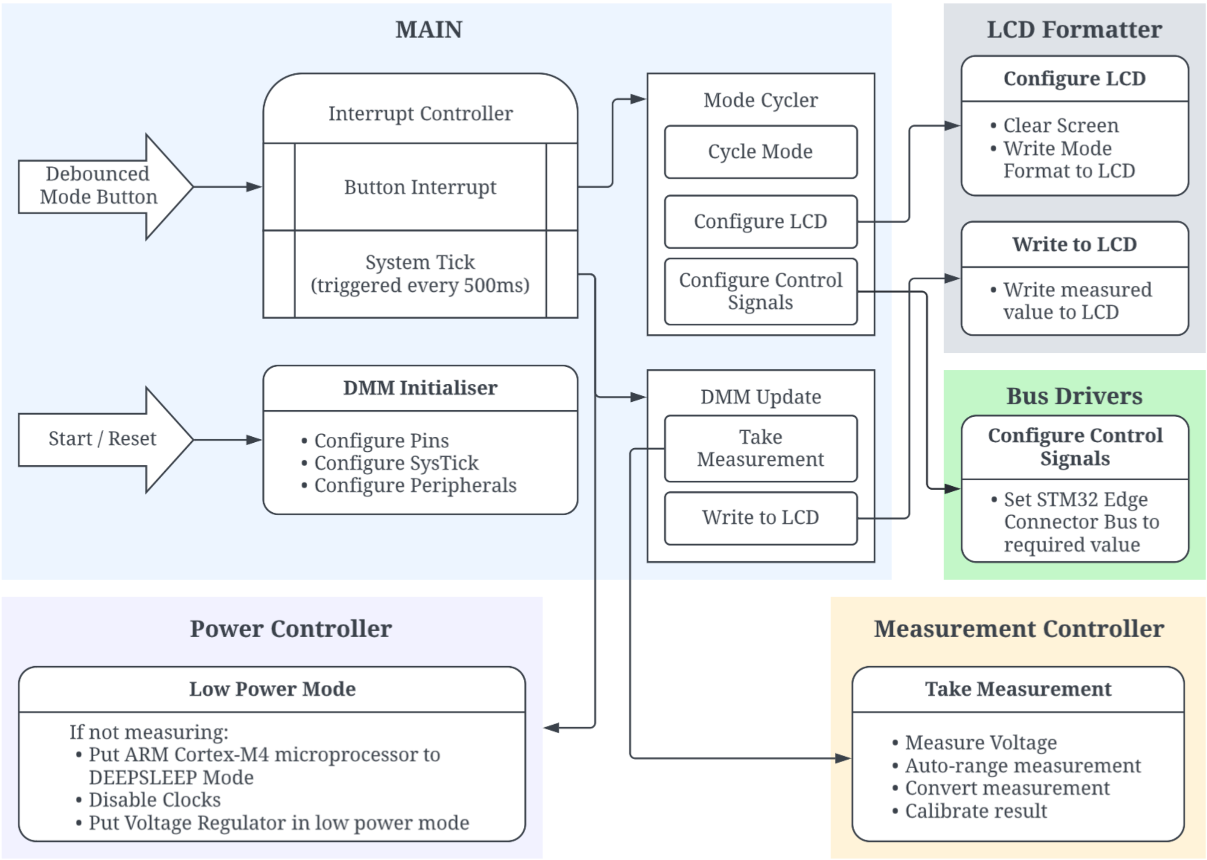

Figure 5 - Software Final Design

The software file structure for the device shown in **Figure 6**, is based on easy modification. Each module contains logic for a specific function of the program, which aids debugging and testing. This is necessary to facilitate multiple developers updating the program simultaneously. The modules are not cross dependent, so changes in a module won't affect other modules on the same level.

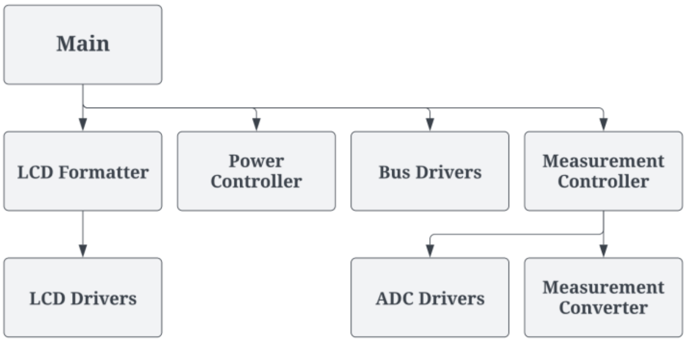

Figure 6 - Software Module Structure

1.2.3 Product Specifications

Table 1 - Product Specifications

Specifications		
Voltage DC	Accuracy[1]	$\pm$ 1mV
	Maximum Magnitude	10V
Voltage AC	Accuracy[1]	$\pm$ 10mV
	Maximum Magnitude	10V
Current	Accuracy[1]	$\pm$ 1mA
	Maximum Magnitude	100mA
Resistance	Accuracy[1]	$\pm$ 1k
	Maximum Magnitude	1MΩ
Frequency	Accuracy[1]	$\pm$ 0.1%
	Maximum Magnitude	10MHz

[1]Note: Accuracies are best accuracy of function

1.2.4 Use of the Analogue-to-Digital Converter

The STM32s built-in ADC (Analogue-to-Digital Converter) is used to convert analogue voltage readings into a digital value. The ADC compares the voltage input to a reference voltage, and outputs to a 12-bit result register, the ratio between the analogue input and the reference voltage. After the conversion the end-of-conversion flag is raised, and software can begin processing the ADC results.

To improve the accuracy of the ADC, the multimeter uses multiple methods:

Averaging When a voltage measurement is triggered, the ADC makes up to 100 conversions and returns the average result. Averaging the result effectively removes noise from the measurement. The maximum conversions can be configured for further accuracy.

Calibrating By measuring an internal reference voltage V_{REFINT} that is output from a stable voltage source that does not vary with temperature, software can calculate a calibration value. By multiplying the result of an ADC conversion with this calibration value, we can remove offset error from the ADC readings. Changes in temperature will affect the ADC voltage reference values, so new calibration values must be calculated often or else the stored calibration value can become increasingly inaccurate over time.

To convert the ADC result into a voltage, we must use the below formulae:

$$V_{Channel\_X} = ADC_{Result} * \frac{V_{REF}}{2^n}$$ *Equation 1*

$$V_{REF} = V_{REF+} - V_{REF-}$$ *Equation 2*

Where n is the number of bits in the result register. V_{REF} is not known by software, however, we know the value of an internal reference voltage, V_{REFINT} from the STM32F4s datasheet [1] which can be seen in **Table 2**:

Table 2 - Embedded internal reference voltage [1]

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{REFINT}	Internal reference voltage	–40 °C < T_A < +105 °C	1.18	1.21	1.24	V

The program measures V_{REFINT} and uses the result to calculate the calibration value, the ADCs reference voltage. We can substitute the formula to find the reference value into **Equation 1** to create an equation for our program to take measurements with:

$$V_{Channel\_X} = ADC_{Result(Channel\_x)} * \frac{V_{REFINT}}{ADC_{Result(Internal\ Reference)}}$$ *Equation 3*

Using the method in **Equation 3** to calculate voltage requires 2 conversions per result, however, provides high accuracy.

1.2.5 Constructing a Multimeter Prototype

To create a prototype for testing, a STM32F4 attached to a University of York development board was connected to a breadboard. The hardware was then built on the breadboard and test measurements were made. Lab voltage sources were used as a power source for the hardware and to reduce noise, wires on the breadboard were clipped to minimum length. Using this set up, the team was able to adjust, improve and test the design sufficiently.

1.3 Analysis of Effectiveness and Future Plans

1.3.1 Achievements against specifications

Through iterative testing, improving, and reconstructing, the multimeter reached the desired specification for each mode. As all measurements are found through voltage readings, improving the voltage measuring circuit improves all modes. The multimeter is also capable of measuring frequency up to 10Mhz by varying core clock speed. A feature not originally within our specification.

1.3.2 Description of test strategy and results

The devices software structure (see **Figure 6)** is highly modular, and modules are not dependent on modules on the same level or on a higher level of the structure. This allows us to test the functions of all the lowest level modules. Once tested and proven to be fully functional, we test the next level up, until we reach and test our main file. Through this file structure, if there are changes to a module only that individual module needs to be retested instead of the entire program. Saving valuable time.

To test hardware, a high precision multimeter was used to test key points in the circuit to ensure:

- The circuit outputs the correct result
- The output does not go above or below our output threshold (0 – 3V)
- Components do not heat up or receive damage from dangerous voltage loads

With the hardware tested and proven to be safe to connect to our microprocessor, we built our multimeter and input a range of different measurements through the labs signal generator and measured the signal with both our multimeter and a high precision multimeter. The high precision multimeters results were used to then calibrate our multimeter, and through a cycle of testing and improving, we reached the desired accuracy in the multimeters readings. A portion of a DC Voltage Test can be seen in **Table 3**.

Table 3 *– DC Voltage Test*

DC Voltage - TEST			
Signal generated (V)	High Precision Result (V)	Our Result (V)	Calibrated Result (V)
7	6.971	6.16	6.971
6	5.975	5.256	5.975
5	4.977	4.353	4.977
4	3.984	3.45	3.984
3	2.988	2.548	2.988
2	1.9897	1.645	1.9897
1	0.9944	0.742	0.9944
0	-0.0005	-0.163	-0.0005

NOTE: full ±10V range not shown due to document size restrictions

1.3.3 Quality of user interfacing

The final user interface is simple and user friendly. Due to auto-ranging which handles the complicated resolution scaling of the device, there are only 2 user inputs, the Reset Button and Mode Cycle Button. The current mode is clearly shown on the devices display along with the measured unit (e.g., V, mA, kHz) making the multimeter simple and easy to use.

1.3.4 Further work needed as a group to complete this product

The product is at a stage where we have completed all tasks within the original project scope, but there are several tasks that need completing before the device can be released to the market.

Create Device The current design needs to be reconfigured and built on a chip, to remove the University of York development board, and allow a casing to be placed over the hardware. This will shrink the multimeter to a suitable size for a handheld device and protect users from exposed circuitry.

Change Supply The multimeters power source must be changed to an integrated battery pack.

Auto-calibration Calibrating the device is a lengthy process which will increase manufacturing costs and production time. The multimeter currently becomes increasingly inaccurate over time during active operation. The ADC currently auto-calibrates independently, but the hardware does not. It may be possible to automatically calibrate the hardware through an already existing DAC (Digital-to-Analogue Converter). A future developer may be able to implement this feature by:

- Calibrating the DAC using the ADC
- Using the calibrated DAC output to calibrate the hardware

This would greatly decrease multimeter production time, factory labour requirements, and increase multimeter accuracy over time during active operation.

1.4 Analysis of Sustainability

As a designer and manufacturer of a product in the modern market, it is necessary to ensure the team, and all suppliers, are currently and in the foreseeable future practising responsible sustainability. Initially, our suppliers are STMicroelectronics and the University of York.

1.4.1 Sourcing Components

STMicroelectronics, which provides the products microprocessor states on their website [2] and in their 2021 Sustainability Report [3] that all products are conflict mineral free and that the company has reduced per fluorinated compounds (PFCs) by 26% since 2016 from production. The company has committed to be carbon neutral by 2027, confirming that the company is a responsible source of microprocessors currently and in the foreseeable future.

The University of York that provides boards to interface the microprocessor and hardware has stated in their Slavery and Human Trafficking Statement [4], that laboratory suppliers used by the university have undertaken assessments to ensure all supply chains are in accordance with the Ethical Trading Initiative Base Code. The university has committed to achieving carbon neutrality by 2030, according to their 2021-2030 sustainability plan [5].

Several small external components are used in the device, however due to being standardised components, can be sourced from a range of responsible suppliers. Initially external components are sourced from the University of York which has already been assessed by the team.

1.4.2 Manufacturing

During manufacturing, our impact on the environment is largely affected by the consumption of resources and effect on the people involved in the production of the product. Suppliers have minimised resource consumption and waste creation and provided plans for further decreasing resource consumption by 2030 which can be seen below:

The University of York currently [4]:

- Maintains 100% divestment from fossil fuels
- Produces 26MWh from 80 Solar Panels

And states in their Sustainability plan [5] will by 2030:

- Reduce indirect (Scope 3) carbon emissions by 30% against 2017-19 average baseline data
- Increase waste recycling rates to 80%
  Increase water consumption efficiency to 25%

STMicroelectronics has according to their 2021 Sustainability Report [3], currently achieved:

- -19% greenhouse gasses emissions in absolute vs 2019
- 88% of waste reused, recovered, or recycled
- 41% of water recycled and reused
- 40% renewable energy

Striving to improve the lives of the people involved in the creation of our product is important to the team, and our suppliers have shown to have similar morals with the University of York committing to an Anti-Slavery statement [4] mitigating against mistreatment of people in the companies supply chain. And STMicroelectronics state on their website [2], are a member of the Responsible Business Alliance and maintain procedures to improve the local communities, including but not limited to:

- All company sites being covered by RBA audits to ensure no human rights violations
- Internal e-learning modules to help communication and awareness of conflicts of interest
- Zero tolerance of bribery and corruption – clear policy with definitions to help employees understand what constitutes bribery and corruption
- 5 ethic committees which regularly review code of conduct and quarterly review ethical breaches and investigations
- Supporting education in countries the company operates in

1.4.3 Active Operation

Due to designing the product with sustainability during operation in mind, our product can achieve high efficiency through several energy saving features. Including a low-power mode that activates when the multimeter is not taking measurements and managing clock signals to peripherals.

The European right to repair law was also considered during the design process, the hardware components were made easily accessible and are replaceable through many responsible sources for a low price which promotes consumers repairing the product instead of disposing it.

1.4.4 End-of-life

Due to responsible sourcing of materials, the multimeter does not contain any hazardous substances or materials and can therefore facilitate recycling of the device when disposed of.

2 Individual Contribution

2.1 Role and List of Contributions

Role: Main Manager and Software Leader

Contributions as a **Main Manager**:

- Assigned roles
- Led meetings
- Made team decisions
- Assigned weekly goals
- Tracked progress
- Created Documentation
- Created Diagrams for visual representation

Contributions as **Software Leader**:

- Researched and create software design
- Assigned tasks and tracked progress
- Maintained Version control
- Created Measurement Controller, which includes:
    - ADC Driver
    - Conversion/Calibration Module
- Created Button Debouncer
- Created Bus Driver
- Created Power Controller
- Tested all Software

2.2 Technical Contribution Details

2.2.1 As Main Manager:

- Assigned roles and created the team structure seen in **Figure 8**
- Led weekly Thursday meetings and made team decisions
- Created weekly goal. example in **Figure 7**:

☑ ~~Combine software (Task for Nojus):~~

- Include Mode switching in main file as it is a high-level component

Figure 7 *- Example Weekly Goal for Nojus*

- Tracked Progress on a project planner, which can be seen in **Figure 9**
- Developed the visual diagrams in this report for use by the team

2.2.2 As Software Manager:

- Creation of Software conceptual design seen in Figure.
- Assigned Tasks and tracked progress on project planner in Figure
- Version control through managing a GitHub repository, and maintaining a backup

Created and Tested Bus Driver: A module that allows writing to a 5-bit bus on the York Development Boards edge connector. The module contained 2 important functions.

- BusJ5_init: Set Bus J5s pins to General Purpose outputs with Push-Pull resistors.
- BusJ5_set: Takes a 5-bit value and sets the bus pins to output the value. Pseudo-code below:

```
procedure BusJ5_set(num)
    clear current bus value in GPIOB Set/Reset Register
    if num <= 2^5 then
        for each bit in num do
            GPIOB Set/Reset Register |= bit << related pin position
```

Created and Tested Debouncer: A debouncer included in the main file, debouncers the user input button to prevent mode skipping. Allows the mode to be cycled only once every 2 SysTicks.

```
procedure debouncer()              // Triggers on button input
    // On Button press
    if debounceValue == 0 do
        debounceValue = 2
        Cycle Mode                 // << Mode Cycling created by Nojus

procedure SysTick_Handler()        // Triggers every 500mS
    if debounceValue > 0 do
        --debounceValue
```

Created and Tested Power Controller: A module that contains an important function to set the microprocessor to low-power mode. Function pseudo-code below:

```
procedure PWR_enterSTOPMode()
    // Configure STOP Mode
    set to enter standby mode in Power Control Register (PWR->CR)
    set voltage regulator to low-power during standby in PWR->CR
    disable peripheral clocks and System Tick
    set Cortex-M4 chip to DEEPSLEEP
    wait for interrupt
    // On wakeup
    enable clocks and system tick
```

Created and Tested Calibration and Conversion Module: Functions containing the mathematical conversions required to convert our ADC output into our multimeter input values. The module also calibrates the value. To convert the value the following formulae are used:

Conversion Output:	Conversion Formula:
±10V	(V_ADC – 1.5) * (20/3)
± 100mV	(V_ADC – 1.5) / 15
± 100mA	V_ADC / 1
0 – 1MΩ	V_ADC / (9.3 * 10^{-6})

The module also calibrations the converted result according to the formula:

$$V_{Calibrated} = \left((1 + slope\ error) * V_{ADC}\right) + offset\ error \qquad \textit{Equation x}$$

Created and Tested ADC Drivers: Functionality required to configure and use the built-in ADC to take measurements of the analogue input. Contains the following functions:

- Initialiser that sets the input pins to analogue inputs and enables and configures the ADC to run in discontinuous mode.
- ADC get conversion function that starts a conversion and returns the averaged result. Details of how the ADC is being used is explained in **Section 1.2.4** Use of the Analogue-to-Digital Converter Pseudo-code below:

```
// Return average of num conversions of input input
procedure ADC_getConversion(num, input)
    int totalResult, calibrationReference

    // Find calibration reference
    enable internal voltage reference
    set ADC input to channel 17
    repeat num times:
        run conversion
        totalResult += conversion result
    calibrationReference = totalResult / num
    disable internal voltage reference

    // Take measurement on channel x
    set ADC input to channel input
    totalResult = 0
    repeat num times:
        run conversion
        totalResult += conversion result
    Return (totalResult / num) * calibrationReference

Procedure runConversion()         // triggered by run conversion
    trigger conversion
    wait for end-of-conversion flag
    return Result Register contents
```

Created and Tested Measurement Controller: Uses the Calibration and Conversion Module and the ADC Drivers to take measurements according to the measurement mode. Pseudo-code below:

```
procedure getVoltage()
    V_ADC = ±10V ADC input conversion
    if |V_ADC| < 100mV do
        return ±100mV ADC input conversion
    return V_ADC

Procedure getCurrent()
    return ±100mV ADC input conversion

Procedure getResistance()
    return convertToResistance(±10V ADC input conversion)

Procedure getACVoltage(frequency)
    set ADC sample size to [(-0.0001 * frequency) + 100.01]
    Double vSquaredTotal
    Repeat AC_SAMPLES_NUM times:     //Note: AC_SAMPLES_NUM = 500
        vSquaredTotal += getVoltage() ^ 2
    Return square root of (vSquaredTotal / AC_SAMPLES_NUM)
```

3 Reflective Summary of Group Work – including Peer Review

3.1 Reflection of Group Structure

3.1.1 Group Structure

The structure of team 2 (Shown in **Figure 8**) throughout the assignment allowed a balance of roles where in theory, no group members would be left working harder than others. By splitting the group into a software and hardware team, and then splitting the teams into managers and developers, each developer had a manager they could go to, and each manager did not need to entirely learn all the technical complexities of the other teams' tasks. As a software manager, I could focus on researching and developing the software to the best of my ability with my team, while trusting that the hardware team was doing the same, without needing to check their work.

The managers would meet weekly to discuss team progress and bring up topics that may affect the other team such as the hardware control signals and calibration. Each developer was assigned a sub-role to give the individual an assignment to oversee and allow developers a chance to show their project management skills to the team.

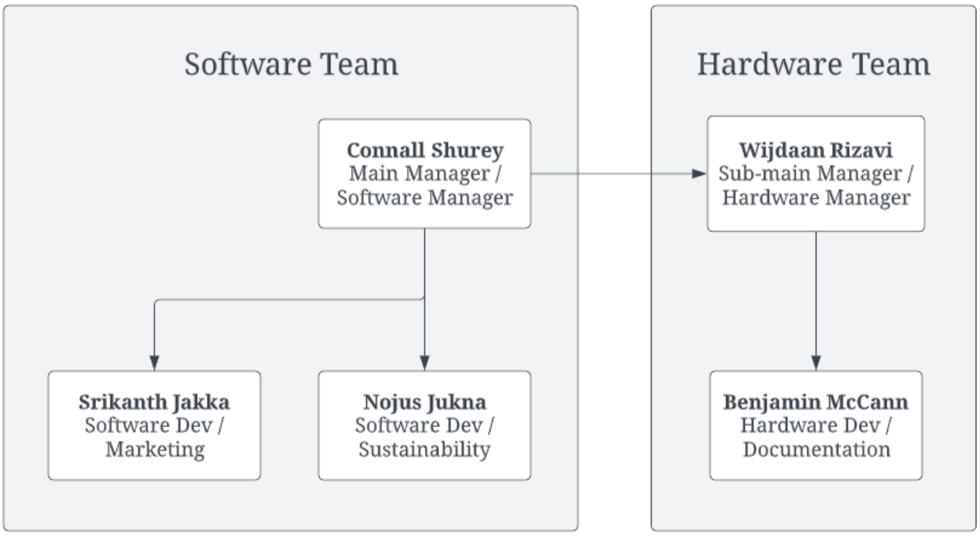

Figure 8 *- Group Structure*

In practice, the structure was exceedingly more effective than expected. Most group members quickly grew into their roles and the project naturally progressed throughout the term.

3.1.2 Time Management and project planning

Initially, the group spent a large portion of the project researching and planning the product design. Using this research, the team was able to develop conceptual designs and create a plan on a Gantt Chart to follow. For more flexibility, I translated the graph into an excel sheet for easier tracking of the groups progress, shown in **Figure 9**. This made it easier to see which week we were in, and whether we were progressing according to our plans.

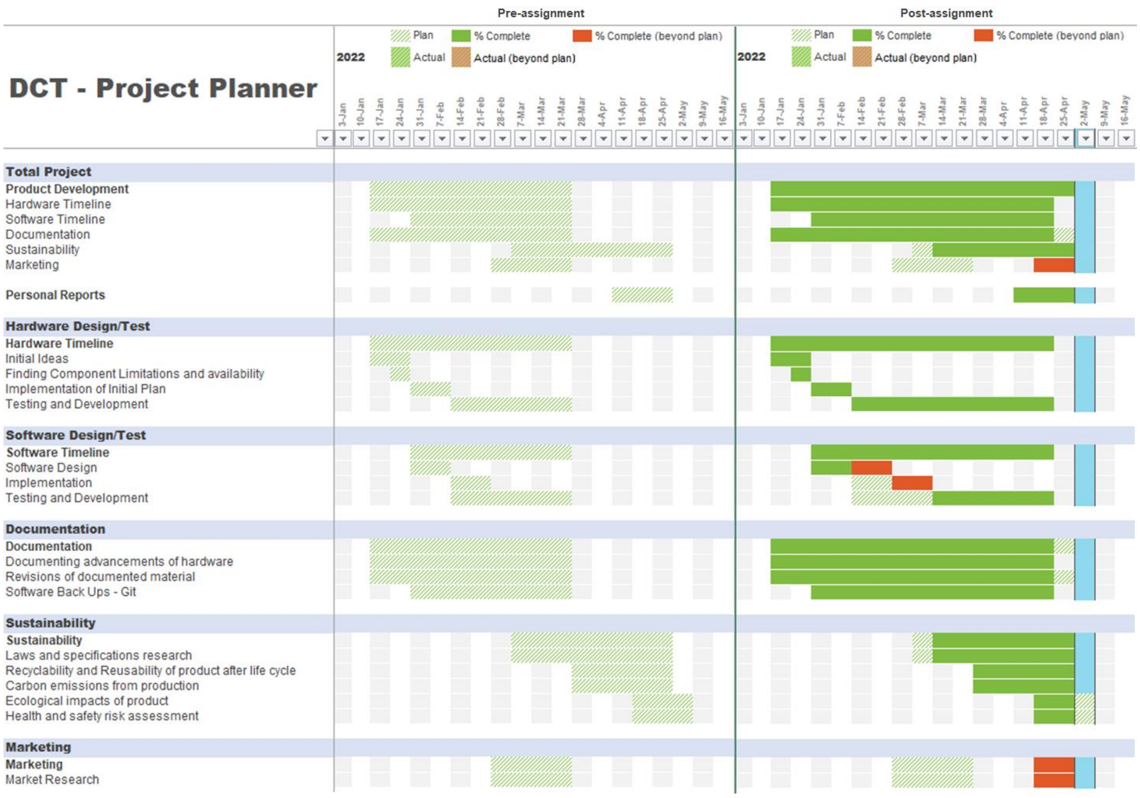

Figure 9 – Project Plan (Before and After Assignment)

The project plan was created with the aim of finishing the main product early so the team had flexibility in the timeline to tackle delays and issues. It was agreed that if the group had time left over at the end of the project, we would add additional features to the product that go beyond the current scope.

In hindsight, there were improvements that could be made to the plan, such as increasing the time assigned to researching and designing. Learning how to use tools that the team has never used before proved to be very time consuming, and with more time to learn how to use these tools proficiently, the group could have relieved pressure towards the end of the project.

It was a good decision to allow flexibility in the timeline as the group tackled several issues throughout the project which can be seen in the post-assignment project plan in **Figure 9**. I fell ill with Coronavirus on the 16th of February, which is shown in the software timeline where the team fell behind schedule. Our timeline allowed room for delays, and this had minimal impact on the overall project. However, the group was faced with a crisis during the project when the marketing report was not started on time and was delayed well beyond its allocated period. The Risk Register did provide strategies to prevent this happening, however despite applying the strategy which was effective for most of the group, the team was impacted from "Lack of Engagement" from an individual. The team showed resilience in shifting focus and tackling the issue together to finish the task promptly. Doing this meant sacrificing any extra features that would make our product stand out. However, completing all tasks within the initial scope takes priority.

3.2 Peer Review

For this assignment, each member of the group conducts a peer review in their individual report. Below I give my peers a fair review, that is based on my personal perception of effort, effectiveness, and overall contribution to the assignment. Each student below is given a mark out of 100 shared total points. The marks were calculated by giving each member a score out of 10 for time invested, the difficulty of their tasks, and the total contribution to the project. Then I calculated their share of the 100 points from the sum of those scores. This gives a fair, mostly unbiased mark.

Name: Connall Shurey **Mark: 26**

Justification: As I took on 2 major management roles, I spent a large quantity of time creating documentation, weekly goals, tracking progress and decision making. I was required to host each meeting, assign tasks, and act as a leader group members could go to in the case of issues or confusion. Whilst also conducting complex software research to make educated decisions and create software plans for the team to follow. I believe this justifies my calculated mark.

Name: Wijdaan Masud Rizavi **Mark: 25**

Justification: Wijdaan took on the role of Hardware Manager and spent many hours in the lab and at home tackling the complex hardware design and creating hardware plans. As another Lead Manager I was able to query Wijdaan at any time for advice on complex problems and overall, Wijdaan contributed largely to the hardware design. Wijdaan can be trusted with any technical problem and his opinion is highly valued by the team.

Name: Benjamin McCann **Mark: 24**

Justification: Benjamin struggled to tackle more complex problems during the project. However, never failed to try and out of the group, Benjamin was in the labs the most, testing the hardware design or working on documentation. Benjamin can always be relied on to complete work promptly and along with Nojus, helped me to cover work not completed by other members.

Name: Nojus Jukna **Mark: 24**

Justification: Nojus can be trusted with tackling highly technical problems that required many hours of research. The complicated, low-level portion of the software that wasn't created by me, was Nojus' hard work. Nojus was late to start fulfilling his role as Sustainability Leader but stepped up to not only fulfill it, but also help cover work not completed by other members.

Name: Srikanth Jakka **Mark: 1**

Justification: At the beginning of the project, Srikanth and Nojus were assigned a short programming task to get a judgement of their technical ability and work ethic. Nojus completed the task over a weekend while Srikanth had yet to complete the task, 12 weeks later. Srikanth did not start the marketing report, despite constant reminders from the team, and does not join us in the labs at any time other than the timetabled slots in which he does not complete any work. When Srikanth finished the mentioned short programming task, it was untested and did not work correctly. I had to test it myself and debug/modify it. While it feels harsh to give Srikanth such a low mark, when compared against other members, especially considering other members of the team had to complete Srikanth's work, this was the result. The only contributions Srikanth has submitted to the team is the creation of an (empty) GitHub repository, and an untested module for formatting strings.

4 Conclusion

This multimeter group project, from start to finish, has been anything but straight forward. Accomplishing even the simplest tasks provided difficulty due to a lack of knowledge and experience. Time spent learning electronics is often spent analysing circuits and rarely spent designing them ourselves. When I first opened the 1751-page reference manual [6] for the microprocessor, I never imagined it would turn into my most valuable source of information for embed programming. The project has been a valuable source of knowledge and experience that I would recommend any Engineering Student undertake. We have finished the project with a product that accomplished goals the team never expected to accomplish, and I feel privileged to have had such incredible team members by my side throughout the journey.

References

[1] STMicroelectronics, “STM32F407xx Datasheet,” August 2020. [Online]. Available: https://www.st.com/resource/en/datasheet/dm00037051.pdf.

[2] “ST Approach to Sustainability - STMicroelectronics,” STMicroelectronics, 2022. [Online]. Available: https://www.st.com/content/st_com/en/about/st_approach_to_sustainability/st_approach_to_sustainability.html. [Accessed 28 April 2022].

[3] “STMicroelectronics Sustainability Report 2021,” STMicroelectronics, 2021. [Online]. Available: https://sustainabilityreports.st.com/sr21/. [Accessed 28 April 2022].

[4] “Slavery and Human Trafficking Statement,” University Of York, 2021. [Online]. Available: https://www.york.ac.uk/about/sustainability/strategy/slavery-human-trafficking/. [Accessed 28 April 2022].

[5] “University of York Sustainability Plan 2021-2030,” 2021. [Online]. Available: https://www.york.ac.uk/media/abouttheuniversity/sustainability/University_Sustainability%20Plan_2021-2030.pdf. [Accessed 28 April 2022].

[6] STMicroelectronics, “STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs - Reference manual,” February 2021. [Online]. Available: https://www.st.com/resource/en/reference_manual/dm00031020-stm32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf.