

CS

**LANC**  
**Video Camera Control**

by

Jack D. Anderson

Supervisor: Dr. R. Lock

Department of Computer Science  
Loughborough University

April 2020

# Abstract

Due to cumbersome existing solutions, this project aimed to develop a remote video camera control system that imposes minimal hardware overhead to the user. Extensive market research and stakeholder interaction led to the design and realisation of an integrated system using custom hardware and software. This included experimental testing of a single-wire serial communication protocol at the physical layer for automatic control bus termination. The system created was found to be commercially viable within the resource and personnel limitations of a single-author project. Full testing in the intended environment was not possible due to coronavirus travel restrictions.

Keywords: Video camera, Remote control, Embedded, Local application control bus, Controller Area Network, Padded Jittering Operative Network.

## Acknowledgements

Firstly, I would like to thank my supervisor, Dr. Russell Lock, for invaluable expertise and encouragement – not least for agreeing to take this project on!

Thanks also to Loughborough Students' Union Media for allowing access to their broadcast equipment and for entrusting me with a large part of it throughout the coronavirus pandemic. I would like to highlight the support received from Jack Connor-Richards and Joshua Gray acting as stakeholders for this project, their contribution of ideas for features and requirements was particularly useful.

Finally, I would like to thank Giovanni Blu Mitolo and Fred Larsen for valuable assistance and encouragement during testing of PJON.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>Contents</b>	<b>iii</b>
<b>List of Acronyms</b>	<b>vi</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Project Background . . . . .	1
1.2 Project Aims . . . . .	2
1.3 Project Objectives . . . . .	3
1.4 Industrial Stakeholders . . . . .	3
1.5 Risks to Project . . . . .	4
1.6 Limitations . . . . .	6
1.7 Document Structure . . . . .	6
<b>2 Literature Review</b>	<b>7</b>
2.1 Project Management . . . . .	7
2.1.1 Waterfall Model . . . . .	7
2.1.2 Incremental Development . . . . .	7
2.1.3 Agile Methods . . . . .	8
2.2 Hardware Subsystem . . . . .	8
2.2.1 Embedded Systems . . . . .	8
2.2.2 Control Communication Interface . . . . .	9
2.2.3 Hardware Dependent Protocols . . . . .	10
 LANC Video Camera Control	 iii

2.2.4	Electromagnetic Compatibility . . . . .	10
2.2.5	Computer-Aided Design . . . . .	11
2.3	Software Subsystem . . . . .	12
2.3.1	Languages . . . . .	12
2.3.2	Software Integration . . . . .	12
2.4	Testing . . . . .	13
2.4.1	Hardware . . . . .	13
2.4.2	Software . . . . .	14
2.5	Summary . . . . .	15
<b>3</b>	<b>Gap Analysis</b>	<b>16</b>
3.1	Competitor Comparison . . . . .	16
3.1.1	Competitor Summary . . . . .	16
3.1.2	Competitor Detail . . . . .	17
3.1.3	Summary . . . . .	23
3.2	Market Research . . . . .	23
3.2.1	Industry Survey . . . . .	23
3.2.2	Stakeholder Interviews . . . . .	32
3.2.3	Summary . . . . .	34
<b>4</b>	<b>Requirements</b>	<b>35</b>
<b>5</b>	<b>Design</b>	<b>37</b>
5.1	System Design . . . . .	37
5.2	Hardware Design . . . . .	39
5.2.1	Master Control Unit . . . . .	41
5.2.2	Camera Control Unit . . . . .	43
5.3	Software Design . . . . .	44
5.3.1	AVR Microcontroller . . . . .	45
5.3.2	ARM Microcontroller . . . . .	45
5.3.3	Single-Board Computer . . . . .	46
<b>6</b>	<b>Implementation</b>	<b>48</b>
6.1	Hardware Implementation . . . . .	48
6.1.1	Electronics . . . . .	48
6.1.2	Enclosures . . . . .	54
6.2	Software Implementation . . . . .	58
6.2.1	AVR Microcontroller . . . . .	58

6.2.2	ARM Microcontroller . . . . .	59
6.2.3	Single Board Computer . . . . .	62
6.2.4	Open-Source Summary . . . . .	67
<b>7</b>	<b>Testing</b>	<b>68</b>
7.1	Hardware Testing . . . . .	68
7.2	Software Testing . . . . .	69
7.3	System Testing . . . . .	70
<b>8</b>	<b>Evaluation</b>	<b>71</b>
8.1	Technical Requirements . . . . .	71
8.2	Commercial Viability . . . . .	73
<b>9</b>	<b>Conclusion</b>	<b>74</b>
	<b>References</b>	<b>75</b>
	<b>Appendices</b>	<b>80</b>
<b>A</b>	<b>Industry Survey</b>	<b>81</b>
A.1	Consent Page . . . . .	81
A.2	Participant Details . . . . .	83
A.3	Camera Control Systems . . . . .	85
<b>B</b>	<b>PJON</b>	<b>87</b>
B.1	Overview . . . . .	87
B.2	Testing . . . . .	87
B.3	Conclusion . . . . .	90
<b>C</b>	<b>Printed Circuit Boards</b>	<b>91</b>
C.1	Master Control Unit . . . . .	91
C.1.1	Schematic . . . . .	91
C.1.2	Board Layout . . . . .	97
C.1.3	Bill of Materials . . . . .	106
C.2	Camera Control Unit . . . . .	108
C.2.1	Schematic . . . . .	108
C.2.2	Board Layout . . . . .	113
C.2.3	Bill of Materials . . . . .	117
<b>D</b>	<b>Mechanical Drawings</b>	<b>120</b>

## List of Acronyms

- API** Application Programming Interface.
- ARM** Advanced Reduced Instruction Set Computing Machine.
- AVR** Alf and Vegard's Reduced Instruction Set Computing Processor.
- BBC** British Broadcasting Corporation.
- BSD** Berkeley Standard Distribution.
- CAD** Computer-Aided Design.
- CAM** Computer-Aided Manufacturing.
- CAN** Controller Area Network.
- CCU** Camera Control Unit.
- DC** Direct Current.
- EEC** European Economic Community.
- EMC** Electromagnetic Compatibility.
- FIFO** First-In, First-Out.
- GNU** GNU's Not Unix! (recursive acronym).
- GPIO** General Purpose Input/Output.
- GPL** GNU General Public License.
- GUI** Graphical User Interface.

**HTTP** Hypertext Transfer Protocol.

**I<sup>2</sup>C** Inter-Integrated Circuit.

**IBM** International Business Machines Corporation.

**IC** Integrated Circuit.

**IDE** Integrated Development Environment.

**IP** Internet Protocol.

**ISM** Industrial, Scientific, and Medical.

**JSON** JavaScript Object Notation.

**LANC** Local Application Control Bus System.

**LDO** Low-Dropout Regulator.

**LED** Light-Emitting Diode.

**LGPL** GNU Lesser General Public License.

**LoRa** Long Range.

**LSU** Loughborough Students' Union.

**LSU Media** Loughborough Students' Union Media.

**LSUTV** Loughborough Students' Union Television.

**MCU** Master Control Unit.

**MIT** Massachusetts Institute of Technology.

**MoSCoW** Must Should Could Won't.

**MVP** Minimum Viable Product.

**N/A** Not Applicable.

**OLED** Organic Light-Emitting Diode.

**OS** Operating System.



**P-value** Probability Value.

**PCB** Printed Circuit Board.

**PDF** Portable Document Format.

**PJDL** Padded Jittering Data Link.

**PJON** Padded Jittering Operative Network.

**PLA** Polylactic Acid.

**PSU** Power Supply Unit.

**REST** Representational State Transfer.

**RF** Radio Frequency.

**RJ45** Registered Jack 45.

**RS-232** Recommended Standard 232.

**RS-422** Recommended Standard 422.

**RTC** Real Time Clock.

**SBC** Single-Board Computer.

**SMA** Sub-Miniature Version A.

**SMART** Specific, Measurable, Achievable, Realistic, Time-Bound.

**SPI** Serial Peripheral Interface.

**SPSS** Statistical Product and Service Solutions.

**SRAM** Static Random-Access Memory.

**TCP** Transmission Control Protocol.

**UART** Universal Asynchronous Receiver-Transmitter.

**UDP** User Datagram Protocol.

**USB** Universal Serial Bus.

LANC Video Camera Control

# List of Figures

1.1	An example of camera settings on a Sony camera . . . . .	2
3.1	Blackmagicdesign camera control software . . . . .	18
3.2	Blackmagicdesign camera control panel - right angle view . . . . .	18
3.3	Blackmagicdesign camera control panel - top view . . . . .	19
3.4	Sony RMB170 . . . . .	20
3.5	Sony MSU1000 . . . . .	20
3.6	Datavideo RMC300C . . . . .	21
3.7	Datavideo MCU200J . . . . .	22
3.8	BroadcastRF . . . . .	22
3.9	Question 1 Plot — Time in Industry . . . . .	24
3.10	Question 2 Plot — Type of Company . . . . .	25
3.11	Question 3 Plot — Type of Productions . . . . .	26
3.12	Question 4 Plot — Viewership Size . . . . .	26
3.13	Question 5 Plot — Primary Role . . . . .	27
3.14	Likert Results Plot . . . . .	28
3.15	Ranking Results Plot . . . . .	29
5.1	System Design Initial Sketch . . . . .	38
5.2	System Design Overview Diagram . . . . .	39
5.3	Master Control Unit Overview Diagram . . . . .	42
5.4	Master Control Unit ARM Diagram . . . . .	42
5.5	AVR CAN Bus Auto-Termination Diagram . . . . .	43
5.6	Camera Control Unit Overview Diagram . . . . .	44
5.7	Camera Control Unit ARM Diagram . . . . .	44
6.1	PCB Layout Design . . . . .	51
6.2	PCB Assembly Component Placement . . . . .	52

6.3	Assembled CCU PCBs . . . . .	53
6.4	Assembled MCU PCB . . . . .	53
6.5	CCU Enclosure Base CAD View . . . . .	55
6.6	CCU Enclosure Tally CAD View . . . . .	55
6.7	CCU Enclosure Assembly CAD Render 1 . . . . .	56
6.8	CCU Enclosure Assembly CAD Render 2 . . . . .	56
6.9	CCU Enclosure Base Slice View . . . . .	57
6.10	Completed CCU . . . . .	58
6.11	Web GUI Main View . . . . .	64
6.12	Web GUI About View . . . . .	65
6.13	Web GUI Help View . . . . .	66
7.1	Hardware Testing of a CCU . . . . .	69
B.1	PJDL Mode One 2,000m Oscilloscope Trace . . . . .	88
B.2	PJDL Mode Four 2,000m Oscilloscope Trace . . . . .	89
C.1	MCU Schematic – Sheet One . . . . .	92
C.2	MCU Schematic – Sheet Two . . . . .	93
C.3	MCU Schematic – Sheet Three . . . . .	94
C.4	MCU Schematic – Sheet Four . . . . .	95
C.5	MCU Schematic – Sheet Five . . . . .	96
C.6	MCU All Layers . . . . .	98
C.7	MCU Top Copper . . . . .	99
C.8	MCU Bottom Copper . . . . .	100
C.9	MCU Top Solder Mask . . . . .	101
C.10	MCU Bottom Solder Mask . . . . .	102
C.11	MCU Top Silkscreen . . . . .	103
C.12	MCU Bottom Silkscreen . . . . .	104
C.13	MCU Solder Paste . . . . .	105
C.14	CCU Schematic – Sheet One . . . . .	109
C.15	CCU Schematic – Sheet Two . . . . .	110
C.16	CCU Schematic – Sheet Three . . . . .	111
C.17	CCU Schematic – Sheet Four . . . . .	112
C.18	CCU All Layers . . . . .	113
C.19	CCU Top Copper . . . . .	114
C.20	CCU Bottom Copper . . . . .	114
C.21	CCU Top Solder Mask . . . . .	115

C.22 CCU Bottom Solder Mask . . . . .	115
C.23 CCU Top Silkscreen . . . . .	116
C.24 CCU Bottom Silkscreen . . . . .	116
C.25 CCU Solder Paste . . . . .	117
D.1 CCU Enclosure Assembly Drawing . . . . .	121
D.2 CCU Enclosure Base Drawing . . . . .	122
D.3 CCU Enclosure Top Drawing . . . . .	123
D.4 CCU Enclosure Sundries Drawing . . . . .	124

# List of Tables

1.1	Risk Likelihood . . . . .	4
1.2	Risk Consequence . . . . .	4
1.3	Project Risks and Mitigation Plan . . . . .	4
3.1	Competitor Product Summary . . . . .	17
3.2	Hypothesis One . . . . .	30
3.3	Hypothesis Two . . . . .	31
3.4	Hypothesis Three . . . . .	32
3.5	Thematic Analysis of Stakeholder Interviews . . . . .	32
5.1	Cabling Conductor Signal Assignment . . . . .	41
6.1	Open-Source Components . . . . .	67
8.1	Requirements Evaluation Summary . . . . .	71
B.1	PJDL Experimental Maximum Range Summary . . . . .	89
C.1	Master Control Unit Bill of Materials . . . . .	106
C.2	Camera Control Unit Bill of Materials . . . . .	117

# 1 Introduction

## 1.1 Project Background

The television industry in the United Kingdom generated an estimated 6.37 billion pounds through subscription revenue in 2018 (Ofcom 2019). In the second quarter of 2019 alone the BBC News television channel had a reach of over 18 million (BARB 2019). Video cameras are used in the production of live television where having remote control over the camera system aids in matching camera shots (Bermingham et al. 1994, p. 44).

Camera control systems are used by a range of industry stakeholders including: television studios, production companies, national broadcasting agencies, and student media organisations. An example of an existing camera control system is Blackmagicdesign's ATEM Camera Control however this system often requires an additional wired connection to each camera as well as requiring cameras that are specifically compatible with the system (Blackmagicdesign 2019).

Control systems allow remote adjustment of camera settings such as:

**Focus** is adjusted so that the desired subject is sharp in the image. Autofocusing systems may struggle with moving subjects, such as sports players (R. Lewis 1991, p. 20).

**Gain** is the electronic amplification of the video signal and should be used sparingly to avoid image quality degradation (Cheshire 1982, p. 130). Low light conditions often require greater gain to achieve a desirable output.

**Iris** aperture controls how much light enters the camera, allowing exposure control in tricky lighting conditions (R. Lewis 1991, p. 13). This effects how bright or dark the image appears.

**Shutter Speed** is the length of time the camera sensor is exposed to light for each frame of video. Whilst also influencing the exposure of the image, shutter speed should usually be set to a multiple of the frame rate to prevent the shutter being visible in the image.

**White Balance** adjusts the camera's reference to white, allowing faithful reproduction of colours under different light sources (R. Lewis 1991, p. 13).

**Zoom** gives a choice of viewing angles allowing a scene to be varied without changing lenses (Cheshire 1982, p. 92).

As can be seen from figure 1.1, camera settings can be complicated and cluttered on the physical camera interface as a result of space limitations. A remote camera control system is not necessarily restricted in the same way.



Figure 1.1: An example of camera settings on a Sony camera

## 1.2 Project Aims

This project aims to allow control over a range of video camera models with minimal additional hardware burden to the user, in the hope of reducing the cost and technical overhead involved.

The overall system aims to allow any video camera with a compatible control protocol to be remotely controlled from a simple software interface. This will aid in calibrating camera settings from a distance.

Due to the cost and scale of existing camera control systems, it was not possible to produce a fully-fledged competitor product with the resources available for this project. Therefore, a minimum viable product was researched and implemented to assess the feasibility of developing a commercial product from this project.

### 1.3 Project Objectives

For the project as a whole to succeed, the following SMART objectives had to be met with reference to the time-frames of the final-year project module (Doran 1981, pp. 35-36).

1. Complete a literature review to identify the most appropriate technologies to be used in the design and implementation of the system.
2. Perform market research of relevant industry stakeholders, existing solutions, and technologies to aid the requirements design.
3. Develop hardware subsystems as necessary to allow the system to operate in line with the project aims.
4. Develop software subsystems to allow operation of the system and integration with relevant protocols as required to achieve the project aims.
5. Perform complete testing of the system on all levels to ensure satisfactory operation to the system design.
6. Document project progression in this report to record the project process, decisions, and outcomes.

### 1.4 Industrial Stakeholders

As mentioned in the project background (1.1), there are multiple general industrial stakeholders to a camera control system. Specifically for this project, the main stakeholder was Loughborough Students' Union Media (LSU Media) as they kindly allowed use of their broadcast equipment for development and testing purposes.

Within LSU Media there were two main contacts: Jack Connor-Richards and Joshua Gray. Jack Connor-Richards is an alumnus of Loughborough University, continuing to be heavily involved in LSU Media and now working for the BBC as a senior broadcast engineer. He was chosen for his wide experience in production and the technicalities of live television broadcasts, as well as a strong background in the maintenance and support of the underlying systems. Joshua Gray is the current Loughborough Students' Union Television (LSUTV, a subsection of LSU Media) station manager and is responsible for oversight of all activities the station undertakes. Joshua was selected for his general knowledge and use of broadcast equipment as well as being a target user of this project.



## 1.5 Risks to Project

Projects have an underlying need to manage uncertainty, which can impose risk, in order to increase the likelihood of success. The risks to a project can be viewed as threats to the success of the project. These can be assessed in terms of the size of the negative impact of a particular risk and its likelihood (Chapman and Ward 2003, pp. 3-5).

In order to assess the risks of this project, the following equation and definitions were used (C. W. Dawson 2009, pp. 82-83):

$$\text{risk impact} = \text{likelihood} \times \text{consequence}$$

Table 1.1: Risk Likelihood

Likelihood	Score
Low	1
Medium	2
High	3

Table 1.2: Risk Consequence

Consequence	Score
Very Low	1
Low	2
Medium	3
High	4
Very High	5

Table 1.3 details the risks to this project that were identified. It additionally shows the risk impact score, risk mitigation plan, and new risk impact score assuming the mitigation is successful.

Table 1.3: Project Risks and Mitigation Plan

Identified Risk	Impact Score	Mitigation Plan	New Score
Use of new technologies slow progress	Likelihood: 2, Consequence: 4, Score: 8	Allow adequate time and reduce use of new technologies	Likelihood: 1, Consequence: 4, Score: 4
Time limitations due to other commitments	Likelihood: 3, Consequence: 4, Score: 12	Ensure good time management by maintaining a Gantt chart	Likelihood: 2, Consequence: 4, Score: 8
Data Loss	Likelihood: 2, Consequence: 5, Score: 10	Back up data regularly	Likelihood: 2, Consequence: 2, Score: 4

<b>Continuation of Table 1.3</b>			
<b>Identified Risk</b>	<b>Impact Score</b>	<b>Mitigation Plan</b>	<b>New Score</b>
Overrun due to poor time estimation	Likelihood: 3, Consequence: 5, Score: 15	Allow for a large contingency period in case of overrun	Likelihood: 2, Consequence: 2, Score: 4
Requirements Inflation	Likelihood: 2, Consequence: 4, Score: 8	Perform adequate research and allow enough time for excellent requirements definition	Likelihood: 1, Consequence: 4, Score: 4
Multiple hardware iterations cause delays	Likelihood: 3, Consequence: 4, Score: 12	Allow extra time for hardware design to allow design checking, contingency period as backup	Likelihood: 2, Consequence: 3, Score: 6
Under utilisation of resources	Likelihood: 2, Consequence: 3, Score: 6	Ensure weekly task planning with goals	Likelihood: 1, Consequence: 3, Score: 3
Inadequate testing due to overrun in earlier project stages	Likelihood: 3, Consequence: 4, Score: 12	Allow adequate time for earlier stages, contingency period as backup	Likelihood: 2, Consequence: 3, Score: 6
Hardware Failure	Likelihood: 2, Consequence: 5, Score: 10	Ensure good hardware design through research and adequate design checks	Likelihood: 1, Consequence: 5, Score: 5
Open-source dependencies becoming unmaintained or unavailable	Likelihood: 3, Consequence: 4, Score: 12	Reduce dependencies where possible	Likelihood: 2, Consequence: 4, Score: 8
Closed-source protocol changes requiring rework	Likelihood: 2, Consequence: 5, Score: 10	If possible, avoid closed-source protocols	Likelihood: 1, Consequence: 5, Score: 5
Damage to borrowed equipment caused by poor design	Likelihood: 2, Consequence: 5, Score: 10	Ensure adequate design checking and aim for fail-safe designs	Likelihood: 1, Consequence: 5, Score: 5
Supplier Delays	Likelihood: 2, Consequence: 5, Score: 10	Plan for longer than agreed delivery times, contingency period as backup	Likelihood: 2, Consequence: 2, Score: 4
Illness	Likelihood: 3, Consequence: 4, Score: 12	Avoid unnecessary risks, use contingency period if unable to work	Likelihood: 2, Consequence: 3, Score: 6
<b>End of Table</b>			

## 1.6 Limitations

The main limitations of this project were imposed as a result of available hardware and time constraints.

Due to the need to develop and test with real broadcast hardware in order to ensure compatibility, the system could only work with the hardware available in LSU Media. LSU Media's video cameras are all controllable via Sony's LANC protocol. The remainder of their broadcast system is based around Blackmagicdesign's ATEM product range. Therefore, this project is limited to compatibility with the LANC camera control protocol and the ATEM control protocol. A future commercial product could be expanded to other protocols with additional work.

Since this project was carried out as an undergraduate final year project, limited time and personnel resources were available and the author was the only active workforce with limited oversight from the supervisor.

## 1.7 Document Structure

This report is made up of the following chapters in addition to this introduction. Chapter 2 is a literature review, analysing sources of relevance to this project. Chapter 3 continues the research phase by exploring existing solutions and performing market research as a gap analysis. Next, the project requirements form chapter 4. With the requirements defined, the report then covers the design (chapter 5) and implementation (chapter 6) of the project system. This is followed by chapter 7 covering all aspects of testing. Chapter 8 evaluates the outcome of the project including its commercial viability. Finally, chapter 9 concludes with lessons learnt.

**It is recommended to view this document in a fully-featured PDF viewer so that the contents may be easily navigated using bookmarks.** Section references, citations, and acronyms also link to the appropriate part of the document.

## 2 Literature Review

### 2.1 Project Management

In order for a project to be carried-out effectively, a number of inter-linked activities must be managed (ISO/IEC 2010, p. v). Several different approaches in how to define the activities and the management process are compared below.

#### 2.1.1 Waterfall Model

The waterfall model in principle requires all processes to be defined before software development begins (Sommerville 2016, p. 47). Futrell, D. F. Shafer and L. I. Shafer (2002, pp. 147-152) states that this model is best suited to projects with easily defined requirements that do not need to change throughout the project. This may not be the case with this project as the requirements may develop with stakeholder and user interaction. On the other hand, B. Hughes and Cotterell (2009, pp. 82-83) suggests that the waterfall model is well suited to projects with limited resources by restricting the process to a largely “one-shot or once-through model”, which could be beneficial to this project due to its strict time-frame with reduced scope for rework.

#### 2.1.2 Incremental Development

The incremental method breaks the process down into individual components which are completed in sequence, each giving benefit to the user (B. Hughes and Cotterell 2009, p. 88). This approach gives more flexibility in requirements changes when compared to the waterfall model (Sommerville 2016, p. 50) which could benefit this project if used alongside rapid stakeholder feedback. However this method makes it harder to track the project progress due to the potential for rework to an unknown extent (Futrell, D. F. Shafer and L. I. Shafer 2002, pp. 147-152) which could be problematic with the author’s relative inexperience in project management.

### 2.1.3 Agile Methods

The agile mindset focuses on the reduction of overheads in the development process (Somerville 2016, p. 66). In the context of this project's limited resources, any reduction in overheads could be considered beneficial however B. Hughes and Cotterell (2009, pp. 92-93) highlights the team-oriented nature of agile methods which would, on reflection, seem poorly suited to a project with one active developer.

## 2.2 Hardware Subsystem

### 2.2.1 Embedded Systems

Embedded systems make use of a microprocessor to control a range of functions (Heath 2002, p. 2). Some considerations regarding embedded system design are explored below.

#### Microcontroller Families

Predko (1999, pp. 3-5) states that different processor architectures influence performance differences and general capabilities of their respective microcontroller family.

One such family is the Motorola MC68HC05 which is designed for low cost and reduced power consumption (Heath 2002, p. 23). Given the resources of this project, low cost could be beneficial however J. Hughes (2016, p. 14) highlights the versatility and ease of programming of the Atmel AVR microcontroller family when compared to the MC68HC05. Additionally, in the small quantities needed for this project, the cost differences would be negligible.

In contrast, the Intel 8051 microcontroller is one of the most popular architectures ever produced however application design is considerably different to most microcontrollers and the standard of development tools available is relatively poor (Predko 1999, pp. 153-155, 189-190, 197). This would not seem to lend itself to straightforward system design for this project.

In comparison to the archaic yet common 8-bit processor families already discussed, 32-bit ARM Cortex-M based microcontrollers allow much higher performance whilst maintaining a very low cost (Martin 2013, pp. 1-3). As this project aims to demonstrate a concept, a microcontroller with an excess of resources could improve the chance of success, especially if requirements change after the hardware has been designed. Later commercial products could optimise microcontroller choice once the requirements are more thoroughly known. In addition, the author has considerable experience in using the ARM Cortex-M family as well as owning the required hardware programming devices.

## Hardware Construction

Development kits could be used for hardware components presenting the most risk (White 2012, p. 36). In the context of this project, multiple development kits could be joined together to give the desired hardware functionality through known working designs. This could significantly reduce the amount of custom design required and save considerable time in development.

On the other hand, a high-density Printed Circuit Board (PCB) design could allow the overall size and weight to be reduced (Coombs 2008, p. 22.3). Despite the increased work required, Tooley (2015, pp. 370-371) states that PCBs result in neat, professional designs that are well suited to prototyping due to the ease of duplication and modification. The ability to easily modify the design could be useful in this project if hardware iterations are required. Furthermore, small size could allow easier mounting and placement of control circuitry near a camera.

### 2.2.2 Control Communication Interface

#### Communication Media

Despite wire links providing an ideal data transmission conveyance for most control applications, Morris (1983, pp. 228-229) highlights the attractive qualities of Radio Frequency (RF) transmission. In particular, the less cumbersome connections between nodes since no wires are needed as well as the ease of connecting multiple devices to one control centre.

Furthermore, The National Institute of Justice (2002, pp. 3-6) states that the major advantages of RF systems include the ability to work over large distances and through obstacles. Alternatively, when considering cable-based systems it states that they work in scenarios where radio systems cannot be used or will not give adequate reception however, cables can restrict mobility and increase system setup times.

Radio systems would appear to align well to this project's aim to impose minimal hardware burden (see 1.2).

#### Radio Frequency Bands

Generally speaking, it is unlawful to make use of wireless telegraphy except under licence (*Wireless Telegraphy Act* 2006, pt. 2 c. 1). Exceptions to this include bands allocated for Industrial, Scientific, and Medical (ISM) applications designed to generate and use radio frequency energy locally (International Telecommunication Union 2012, sec. 1.15). The use of an ISM band for low-power devices results in the end user not having to seek an individual licence from a regulatory authority (Texas Instruments 2005, p. 2).

By reducing regulatory requirements on the end user, the ISM band could enhance the accessibility of this project to a wider customer base with restricted access to funding for licences.

### 2.2.3 Hardware Dependent Protocols

Some of the protocols used in this project will depend upon the hardware to be interfaced with. As this project aims to integrate different systems, the available protocols will already be defined by what the equipment manufacturer has provided. A few protocols are now reviewed.

One potential protocol to use for wired connections is RS-232, this protocol can be used as a simple general purpose serial interface and is present on many devices (Dallas Semiconductor 1998). However, Dallas Semiconductor highlight the unnecessary limitations imposed by the standard's 20kb/s data rate as well as potential issues with long cable lengths.

Texas Instruments (2013, pp. 2-3) detail the RS-422 standard, in particular highlighting how it overcomes RS-232's data rate and cable length issues however noting the extra complexities required in the hardware to support this.

Mid-range and many older Sony video cameras make use of LANC protocol. Vrancic and Smith (2006) state that this protocol conforms to RS-232 timing standards and can allow synchronisation and control of many useful settings on supporting cameras. Due to Sony not releasing a specification of this standard, reverse-engineering by Boehmel (2018) is the de-facto standard relied upon to make use of the protocol.

Blackmagicdesign's ATEM family of broadcast equipment use a custom protocol for control of many aspects, including cameras, with substantial reverse-engineering work on the protocol and open-source libraries available from Skaarhoj (2018).

### 2.2.4 Electromagnetic Compatibility

Electromagnetic Compatibility (EMC) ensures that electrical and electronic systems perform their tasks satisfactorily without causing undue interference to other systems (Chatterton and Houlden 1991, pp. 1-2, 16-17). They go on to state all goods placed on the market within the EEC since 1992 must conform to EMC directives.

Within Europe, it is necessary to issue a 'Declaration of Conformity' listing applied standards (Montrose and Nakauchi 2004, p. 14-16). They conclude that full-compliance EMC testing can be extremely expensive for products produced in small quantities and that alternative approaches to the necessary due diligence such as following standards and performing pre-compliance testing may be more suitable.

Williams (2001, pp. 36-37) states that self certification is the method expected for most manufacturers to demonstrate compliance with the relevant European standards as it does not require any testing other than what is needed to assure the manufacturer that they have indeed met the standards. This testing could be achieved in-house, given sufficient expertise.

As this project has limited access to funding or test facilities, following industry best-practices and regulatory standards may be the simplest way to ensure compliance with the appropriate regulations.

### **2.2.5 Computer-Aided Design**

Computer-Aided Design (CAD) has the potential to shorten the product development cycle, improve quality, and reduce costs (Chang 2013, p. 2). Therefore, it would seem appropriate to explore the use of CAD in the context of this project.

#### **Mechanical Design**

The use of CAD software, in particular AutoCAD, for mechanical design increases the precision, modifiability, and efficiency of the drawings produced (Byrnes 2010, pp. 12-13). Ruiz and Jack (2010, p. 2) states that SolidWorks is one of the most popular 3D mechanical CAD packages available, in particular highlighting its ease-of-use and powerful tool-set.

Efficiency and modifiability could be especially useful in this project if designs require changes since there is limited time available for rework. Furthermore, the author has considerable experience using SolidWorks and this familiarity could mitigate some of the risks explored in 1.5.

#### **Printed Circuit Board Design**

In relation to PCB design, CAD tools can be used to turn electrical circuit schematics into a physical package by following wiring rules and ensuring compliance with manufacturing limitations as defined by the designer (Coombs 2008, p. 14.8). EAGLE CAD provides an easy to learn and use PCB schematic and board layout editor, whilst also allowing implementation of complex designs (Aono 2011, p. 1). Additionally, the author has access to and experience using EAGLE CAD for PCB design.



## 2.3 Software Subsystem

### 2.3.1 Languages

#### Embedded Software

Martin (2013, pp. 17-18) describes the long history and support of C/C++ in use with ARM microprocessors, highlighting the extensive toolchains available including open-source software that could significantly speed up development of common components.

Opposed to this, Venkataramanan (2013) describes how MicroPython offers ease-of-use over C/C++ whilst maintaining the ability to process in real-time, which is important in a control system.

Finally, Pascoa (2017) details that despite assembly language being expected to deliver higher performance than compiled code such as C/C++ and MicroPython, recent improvements in compiler optimisation results in assembly language being comparatively slower unless written by a talented expert in assembly language — something the author is not.

#### Utility Software

M. Dawson (2010, pp. 3-5) states that C/C++ offers speed advantages over Python due to its lower level approach. However, Dawson goes on to praise Python for its ability to ‘glue’ code from other languages together when speed is necessary as well as the ease-of-use when object-orientation is desired for varied amounts of a project.

Whereas, Java can be particularly useful in projects that can be easily broken down through object-orientation however does not lend itself well to one-off projects or experimentation so may be less suited to this project despite clear applicability of object-oriented techniques (Waldo 2010, pp. 2-3).

An alternative approach could be to use a web interface instead of a typical desktop application so that software only needs to be installed once to be readily available wherever needed. However, a major constraint of web applications is the dependence on a network connection, something that may not always be under the users’ control in the operating environment for this project (Avesta Group 2019).

### 2.3.2 Software Integration

Integration between software components of this project will to some extent be limited by hardware design and compatibility. A few approaches are now explored however other design choices will ultimately influence actual implementation.

If inter-computer networks can be utilised, integration based on the simple and low latency User Datagram Protocol (UDP) could be implemented however the Transmission Control Protocol (TCP) could be beneficial as a result of reliable message delivery (Comer 2015, pp. 449-451, 459-461). For this project, this would give flexibility as custom protocols could be defined on top of either UDP or TCP although this may be unnecessarily complex and time-consuming.

A browser based application could make use of WebSockets to allow two-way communication between a client and remote host, with the advantage of well defined protocol (Fette and Melnikov 2011, pp. 4-5). A well defined protocol could be less risky to this project in comparison to a custom design although it could limit adaptability.

## 2.4 Testing

### 2.4.1 Hardware

#### **Electronic Assemblies**

In addition to basic bare-board testing by the PCB manufacturer, testing of the complete electronic assembly is required to ensure requirements are met. This can be achieved through both visual and automated inspection, the latter requiring complex equipment (Coombs 2008, p. 52.1). Feldmann and Sturm (1994) state that to ensure a quality level close to zero defects, visual, circuit, functional, and thermal testing must be completed.

Subsequently, only visual inspection and functional testing would seem suitable for this limited-scope project.

#### **EMC Testing**

As detailed in 2.2.4, EMC compliance is necessary for electronic systems. Montrose and Nakauchi (2004, pp. 15-16, 57-66) describe the potential for relatively relaxed pre-compliance testing within Europe. This can be achieved with limited equipment, such as oscilloscopes and spectrum-analysers, that do not have to adhere to the rigorous standards of full-compliance test equipment.

Conversely, Williams (2001, pp. 36-37) details that testing is not strictly required when using the self-certification method in Europe. Any testing would purely be to assure the manufacturer that their design performs correctly, providing it has been designed to the relevant standards.

Due to limited resources, this project would be unlikely to gain access to full-compliance testing therefore pre-compliance style testing in order to validate the design may be possible

with the author's test equipment.

### **2.4.2 Software**

In order for software testing to have high probability of success, formal test techniques must be used (W. E. Lewis 2000, p. 19). This subsection reviews a few different methods for testing software.

#### **Unit Testing**

Unit testing could allow individual components to be tested (Royer 1993, pp. 132-133), perhaps even automatically. However, automatic testing requires considerable work and requires tools to be learnt which could pose an additional strain on the author (W. E. Lewis 2000, p. 227-228).

#### **Integration Testing**

Integration tests make use of linking and testing program modules in order so that their correct functioning in the complete system can be ensured (ISO/IEC/IEEE 2017, p. 231). In addition, the British Computer Society Working Group on Testing (1986, pp. 72-76) state that ideally integration testing should be carried out by a team independent of the programming team. This is clearly impractical for this project.

#### **System Testing**

System testing is the first opportunity to test the whole software against the specification and can be seen to do this in a destructive manner by attempting unexpected uses (British Computer Society Working Group on Testing 1986, pp. 82-83). Again, this type of testing is usually carried out by an independent team however strict measuring against the requirements specification during black-box testing could prevent bias in this project (Software Testing Fundamentals 2019).

#### **User-Acceptance Testing**

Acceptance testing can be used to demonstrate the system's ability to meet the original objectives and requirements (W. E. Lewis 2000, p. 201). Royer (1993, pp. 148-149) details how formal demonstrations can be used to perform this task, noting that retesting is necessary for any errors found. As discussed in 2.1.2, rapid stakeholder feedback can be utilised to ensure the project meets the requirements. This could be used as a form of repeated user acceptance testing for this project.

## 2.5 Summary

The literature review has discussed many approaches to the management, implementation, and testing of the project. Whilst some advantages and disadvantages have been proposed, the uniqueness of this project suggests that there may be no single correct way to solve a problem.

Some key themes developed include the size and efficiency benefits available through the use of custom printed circuit boards, the ease of use and simple hardware requirements offered by Radio Frequency communications, and the likely need to self-certify for Electromagnetic Compatibility compliance due to limited project resources.

## 3 Gap Analysis

### 3.1 Competitor Comparison

This section identifies and compares several pre-existing competitor products that are a relevant influence on this project. First an overview comparison of product features is made, followed by in-depth exploration of each individual competitor. These products were chosen for comparison due their features and overlap with the potential market this project could enter.

The identified competitor products are:

- Blackmagicdesign's ATEM camera control (Blackmagicdesign 2019).
- Sony's professional camera control panel range (Sony 2019).
- Datavideo's camera control system (Datavideo Technologies Co. 2019).
- BroadcastRF's wireless camera control packages (BroadcastRF 2019).

#### 3.1.1 Competitor Summary

Table 3.1 summarises the key features of each of the identified competitor products.

The main features that are comparable across the four products are:

**Cost of Basic System** — cost of the cheapest entry-level product from the manufacturer's range that achieves camera control

**Cost of Flagship System** — cost of the top-of-the-range offering from the manufacturer

**Control Interface** — method with which the user interacts with the system

**OS Support** — in the case of software interaction, Operating System (OS) support of the application

**Compatibility** — range of camera systems from different manufacturers that can be controlled

**Tally Indication** — ability of the system to indicate to the camera operator that the camera is live

**Connection** — main means of communication between system components

**Control Protocol** — protocol used to transport commands between system components

**Max. Cameras** — maximum number of cameras that can be controlled by the flagship product

Table 3.1: Competitor Product Summary

Product	Blackmagicdesign	Sony	Data Video	BroadcastRF
<b>Cost of Basic System</b>	£0 (Software Only)	£1,600	£1,200	£200 (Hire Only)
<b>Cost of Flagship System</b>	£2,500	£8,900	£10,000	£300 (Hire Only)
<b>Control Interface</b>	Hardware Panel / Software	Hardware Panel	Hardware Panel	Hardware Panel
<b>OS Support</b>	Windows / Mac	N/A	N/A	N/A
<b>Compatibility</b>	Limited to BMD	Limited to Sony	Good	Excellent
<b>Tally Indication</b>	Yes	No	No	Unclear
<b>Connection</b>	Wired	Wired	Wired	Wireless
<b>Control Protocol</b>	Custom IP	Custom Wired / Custom IP	Mixed Custom IP / Custom Wired	Custom Wireless
<b>Max. Cameras</b>	4	12	4	Unclear

### 3.1.2 Competitor Detail

#### Blackmagicdesign

Blackmagicdesign are the only competitor identified that have a software control solution. This software interface allows control of all the main parameters of the cameras whilst maintaining a relatively simple user interface (see fig. 3.1).

Despite the software itself being zero-cost, it is limited to compatibility with their own (relatively expensive) range of broadcast cameras and associated equipment. They are also

the only competitor to include tally indication within their control system, as a result of the extensive integration with the rest of their suite.



Figure 3.1: Blackmagicdesign camera control software (Blackmagicdesign 2019)

The flagship product from Blackmagicdesign is their hardware camera control panel (see figs. 3.2 and 3.3). An interesting feature of this product is the ability to 'call' a camera to alert the camera operator that someone is trying to communicate with them. The system also allows camera settings to be stored to a 'scene' so that they can be easily recalled later. The hardware panel itself is particularly well designed, giving a very professional appearance.



Figure 3.2: Blackmagicdesign control panel - right angle view (Blackmagicdesign 2019)



Figure 3.3: Blackmagicdesign camera control panel - top view (Blackmagicdesign 2019)

### Sony

Sony's entry level product, the RMB170, allows a basic set of features for controlling one camera but is not well suited to an environment with multiple cameras as many units would be required. As can be seen from fig. 3.4, the main advantage of this product is the small size and portability. A unique feature of the RMB170 is the ability to control the camera's recording media and playback transport however this has limited use within a live production environment. Similarly to Blackmagicdesign, Sony's products are only compatible with their own cameras.





Figure 3.4: Sony RMB170 (Sony 2019)

Sony's flagship product, the MSU1000, incorporates a touchscreen interface as well as conventional hardware controls (see fig. 3.5). This gives the ability to quickly page between multiple settings, potentially allowing a wider range of control parameters. As this presents a hybrid between hardware and software control interfaces, it may allow new features to be more easily delivered through firmware update as the need to redesign the controls is partially mitigated, preventing the need for a whole new product. Of course, this benefit may not be realised in practice depending on Sony's update strategy.



Figure 3.5: Sony MSU1000 (Sony 2019)

## Datavideo

Datavideo's entry level product, the RMC300C, allows control of 96 cameras — far more than any other product. This is achieved by switching between banks of 12 cameras. Figure 3.6 shows the unique use of a tablet and accompanying application to present the full control interface of the device, a novel way to expand the available control without increasing hardware complexity. The shortcoming of this product is its inability to interface with cameras other than Datavideo's own, which are not typical television broadcast cameras.



Figure 3.6: Datavideo RMC300C (Datavideo Technologies Co. 2019)

Datavideo's flagship products, such as the MCU200J, offer compatibility with other manufacturer's cameras however this comes with the substantial limitation that particular versions of the product are limited to work with one camera manufacturer's equipment only. This prevents the potential benefit to use one control system with a diverse range of cameras.

Figure 3.7 shows a screenshot of the product webpage, highlighting the relatively limited control interface. However, the form factor would allow mounting in nineteen inch equipment racks which are commonplace within the television industry.



Figure 3.7: Datavideo MCU200J (Datavideo Technologies Co. 2019)

**BroadcastRF**

BroadcastRF present two major differences from all of the previously explored systems. Firstly, they have the only product utilising wireless communications as the link to the camera. This massively reduces the need for additional cables to the cameras and allows the possibility of controlling a camera that is also sending its video wirelessly, for a completely wireless solution. Secondly, their products are only available on a hire basis which exposes a less obvious business model that could be used if this project was developed into a commercial product.

Figure 3.8 shows the lack of information available about BroadcastRF’s products however they claim excellent compatibility with a wide range of camera systems.

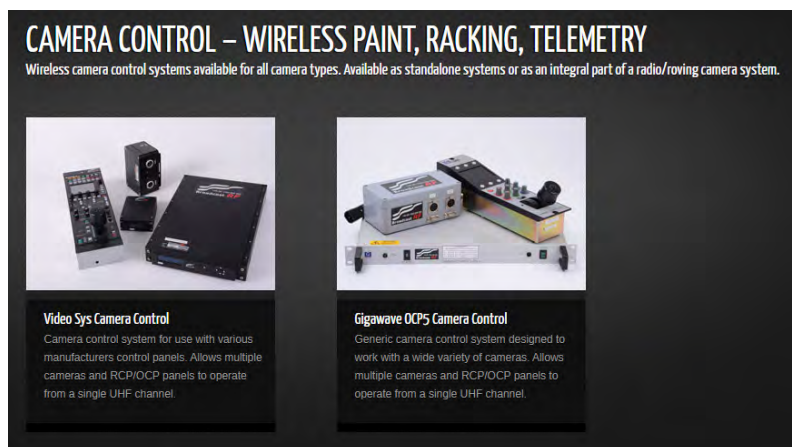


Figure 3.8: BroadcastRF (BroadcastRF 2019)

### 3.1.3 Summary

From the evaluation of competitor products, it can be seen that there is a gap in the market for any product that can achieve control of multiple manufacturers' cameras in one device. Additionally, any product making use of wireless communications could present a threat to most of the products currently available due to their cumbersome equipment requirements. There is also a general lack of software control systems for most camera types, and a lack of low-cost hardware control interfaces for all broadcast cameras.

## 3.2 Market Research

### 3.2.1 Industry Survey

In order to obtain a better understanding of the general feelings towards camera control systems and their features, a survey of industry professionals was carried out. Prior to commencement of the study, the Loughborough University ethical clearance checklist for data collection was completed.

The survey consisted of an information and consent section, a section to collect information about the participant, and a section to assess the participants' attitudes towards certain statements. See appendix A for the complete survey design.

In order to reach a representative sample of participants that would have relevant industry experience, participants were recruited from online forums organised for television broadcasting, video engineering, and associated fields. Convenience sampling was used due to this project's lack of resources.

The total number of participants reached was 58. The remainder of this subsection analyses the results of the survey, question by question, followed by statistical testing for specific hypotheses using IBM SPSS.

#### **Question One — Time in Industry**

The first question intended to gauge the participants' experience in terms of time within the industry. Figure 3.9 shows a plot of the results obtained which suggest that a good range of participants were sampled as both relatively new and long established industry professionals were reached.

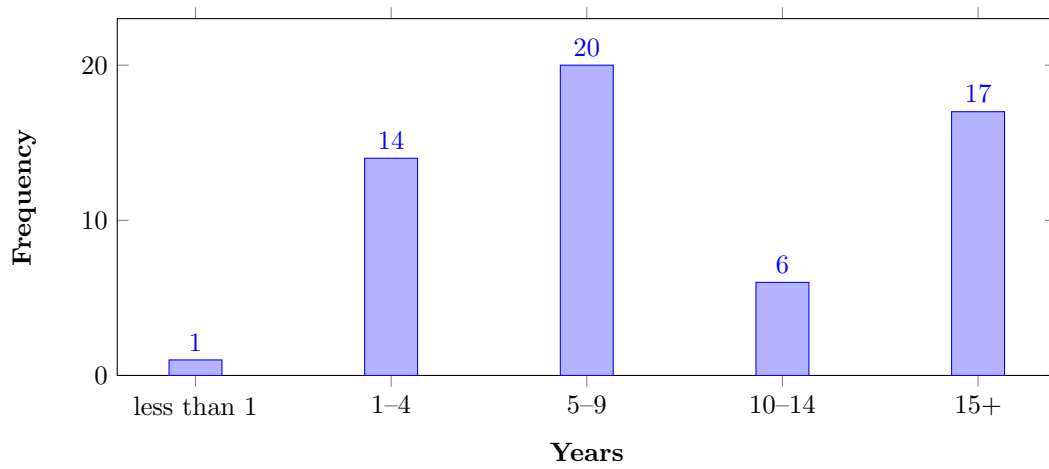


Figure 3.9: Question 1 Plot — Time in Industry

### Question Two — Type of Company

As there are many business models and related types of companies within the television industry, question two was designed to establish the type of company the participants worked for. Figure 3.10 shows the wide distribution of company types represented by the participants, with production companies being the most represented.

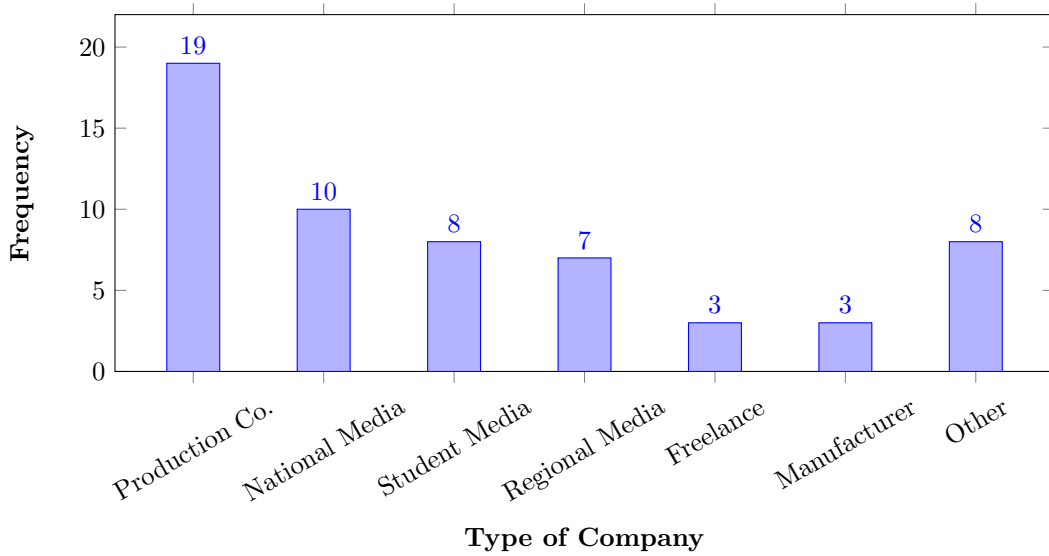


Figure 3.10: Question 2 Plot — Type of Company

### Question Three — Type of Production

Question three assessed the type of productions undertaken by the participants. Since it is commonplace for organisations to produce multiple types of content, multiple selections were accepted. The results are shown in figure 3.11.

Live productions were more heavily selected than recorded productions, however both saw good representation. The most commonly selected types of production were ‘Live Outside Broadcast’ and ‘Live Studio’ which were the two environments initially envisioned for the use of this project. The other responses show that there may be applicability of this project to a wider audience than initially expected.

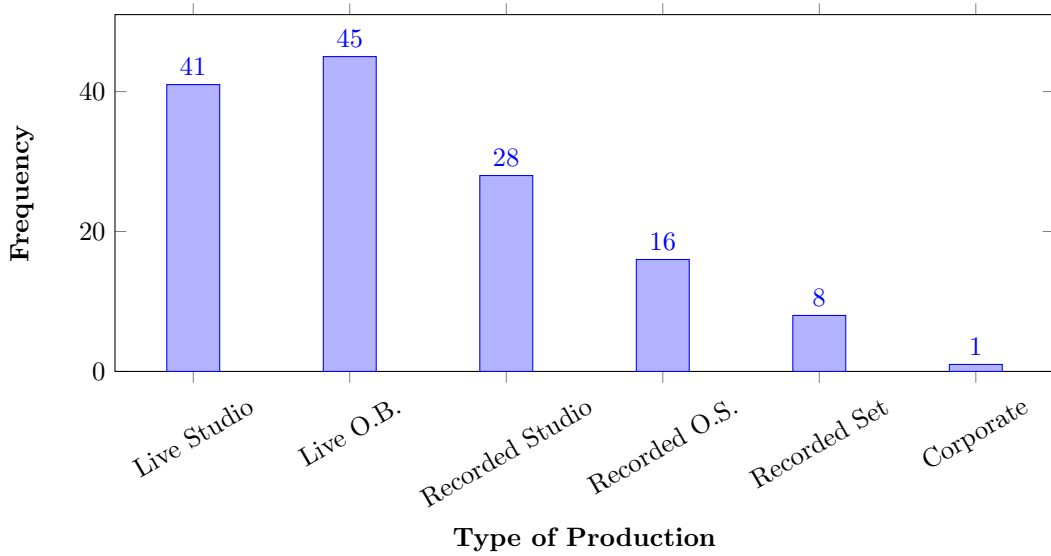


Figure 3.11: Question 3 Plot — Type of Productions

**Question Four — Viewership Size**

The viewership size of productions worked on by the participants formed an even distribution as shown in figure 3.12. This may suggest industry interest from both large and small scale producers.

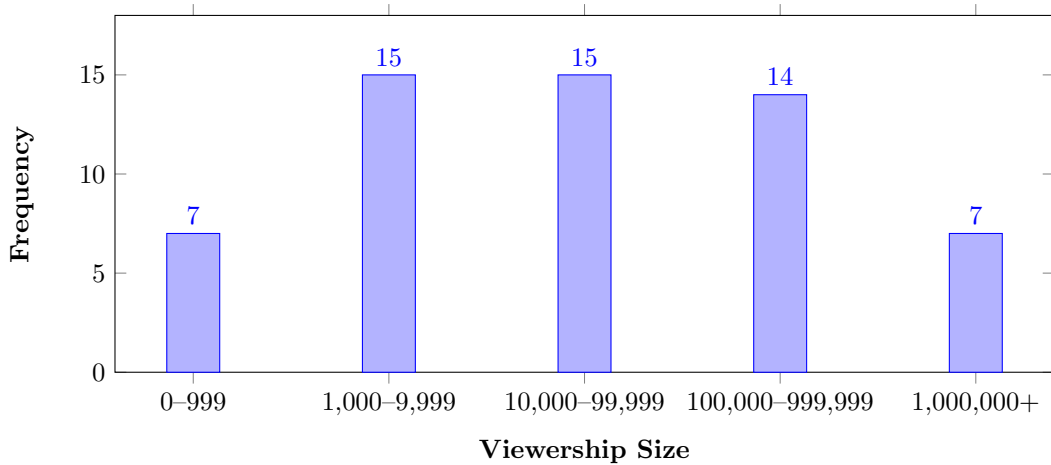


Figure 3.12: Question 4 Plot — Viewership Size

### Question Five — Primary Role

The fifth question aimed to gain an understanding of the type of role participants have in productions. Figure 3.13 shows that the majority of responses were for more senior roles within a production team. If engineer and technical support responses were grouped due to their similarity, they would represent almost half of the responses obtained. This would suggest that most of the participants have a good understanding of the production environment and underlying technical operation.

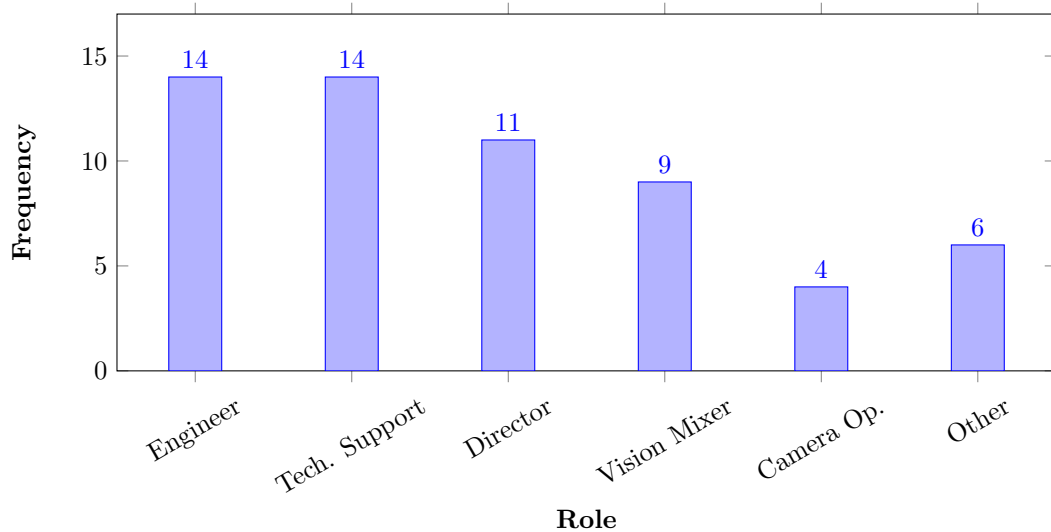


Figure 3.13: Question 5 Plot — Primary Role

### Question Six — Camera Control System Experience

The final question about the participants' experience determined whether the participants had ever used a camera control system before. 93.1% of respondents indicated that they had used a camera control system before. This is a desirable outcome since participants with experience of similar systems should be more well informed about potential advantages and disadvantages to such systems. However, it is possible that due to exposure to existing solutions, participants may be less open to alternative approaches and new technologies.

### Question Seven — Likert Statements

In order to gauge the participants' feelings towards certain statements, a five-point Likert scale was used. Figure 3.14 shows the results obtained. Whilst most statements received an overwhelmingly positive response, there were two major exceptions.



The first was the response to cabling and infrastructure of camera control systems being a burden on productions. The results for this statement were mostly split, with a slightly higher tendency to disagree. This would suggest that whilst cabling of control systems can pose a burden for some productions, there are many where this is not the case.

Secondly, the response to wireless connections being preferred was mostly negative or neutral, with a small but not negligible number of positive responses. This could suggest that most participants did not favour wireless connections although there may be circumstances in which they are useful.

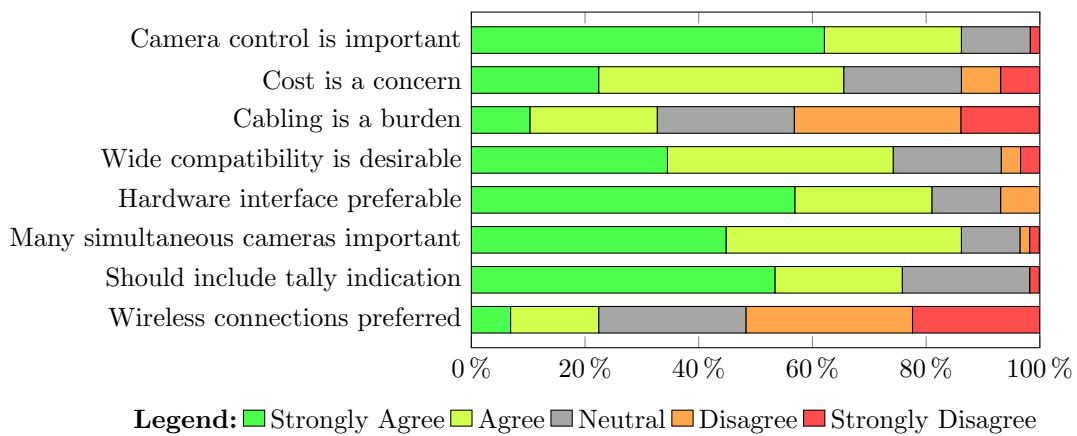


Figure 3.14: Likert Results Plot

**Question Eight — Selection Criteria**

To determine the order of importance of criteria used when selecting a camera control system, a ranking question was used. Figure 3.15 shows the results where ‘Most Important’ indicates that the criterion was ranked first or second most important and ‘Least Important’ indicates it was ranked last or second-last most important. The results have been sorted to give an impression of the overall ranking across all participants.

Reliability and ease of use are clearly the most important criteria when selecting a camera control system, this was as expected. However, the size of the equipment and wireless functionality were least important to participants which brings into question this project’s tendency towards wireless connections.

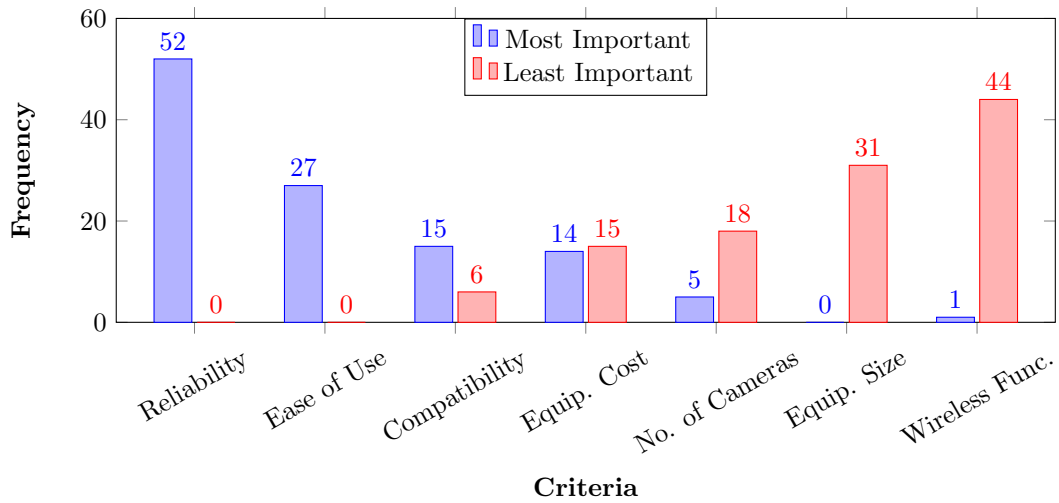


Figure 3.15: Ranking Results Plot

### Question Nine — Open Response

Lastly, an optional open response question was provided to allow participants to make further comments. A total of 9 responses were obtained.

A range of points were made, centring on the following:

- Budget will influence responses in real-world scenario with cost most preventative to student media
- State-of-the-art camera systems allow control data to be sent over power and video cabling
- Vendor support is very important
- Hardware panels are useful for live events such as sport whereas software panels are good for remote support
- Wireless connectivity is not desirable in professional settings due to reliability

The comments surrounding cost and budget were exactly as would be expected. Clearly, student media being a largely underfunded endeavour would struggle to invest in an expensive system. State-of-the-art systems integrate their control into the cameras directly, allowing for remote camera control by default. However, such new systems are a very expensive investment not suitable for all organisations. Additionally, they generally require a

single manufacturer's products to be used throughout the entire broadcast system which is not always practical.

The comments surrounding vendor support highlighted that any product known to be backed by helpful and responsive customer service would be much more likely to be chosen over a product that is not. This reinforces the importance of having excellent documentation so that the user can fix their own issues. In addition, this would make it easier for a customer support department to give useful responses if a commercial product were developed from this project.

Comments relating to hardware and software control panels detailed that hardware panels are useful when fast or continuous adjustments are needed, such as during live sport. Software control, on the other hand, gives the advantage that remote support can be given and if implemented correctly the software could be used over the internet, allowing control over vast distances.

Responses commenting on wireless connectivity focused on fears that connections could be lost, reducing the reliability of the system.

### Hypothesis One

From the above overview of the results, the following null hypothesis was created: "There is no association between time in industry and preference for wireless connections". This hypothesis was chosen to explore the idea that participants who had been within the industry for a long time may be less open to new technologies such as wireless connections.

The statistical test used is Fisher's Exact Test due to the sample size being insufficient to use the Pearson Chi-Square method as a large percentage of the expected values are below 5. A P-value of less than 0.05 would indicate a significant result thereby requiring rejection of the null hypothesis (McDonald 2015). Table 3.2 shows the cross-tabulation used within IBM SPSS to compute the P-value.

		Wireless connections between the control station and cameras would be preferred					Total
		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	
Time in Industry	less than a year	0	0	0	0	1	1
	1-4 years	4	4	2	3	1	14
	5-9 years	3	5	8	3	1	20
	10-14 years	0	2	2	1	1	6
	15+ years	6	6	3	2	0	17
Total		13	17	15	9	4	58

Table 3.2: Hypothesis One – IBM SPSS Cross-tabulation

Fisher's exact test yields  $P=0.460$ , therefore the result is not significant and the null hypo-

thesis is accepted.

### Hypothesis Two

The second null hypothesis is: “There is no association between company type and concern towards cabling”. This hypothesis was chosen to explore the idea that different types of companies might have different feelings towards infrastructure. For example, large national companies may have access to their own equipment or premises that can be permanently setup whereas smaller organisations may have to rely on shared or hired resources.

Again, the statistical test used is Fisher’s Exact Test due to the sample size being insufficient to use the Pearson Chi-Square method. As with hypothesis one, a P-value of less than 0.05 would indicate a significant result. Table 3.3 shows the cross-tabulation used within IBM SPSS to compute the P-value.

		Remote camera control systems' cabling and infrastructure are a burden on productions					
		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Total
Company Type	Production Company	5	5	3	4	2	19
	National Media	0	4	3	3	0	10
	Student Media	0	4	0	3	1	8
	Regional Media	1	3	2	1	0	7
	Freelance	1	1	0	0	1	3
	Manufacturer	0	0	2	1	0	3
	Other	1	0	4	1	2	8
Total		8	17	14	13	6	58

Table 3.3: Hypothesis Two – IBM SPSS Cross-tabulation

Fisher’s exact test yields  $P=0.216$ , therefore the result is again not significant and the null hypothesis is accepted.

### Hypothesis Three

Finally, the following null hypothesis was chosen: “There is no association between production role and preference for a hardware interface”. This hypothesis was chosen to explore the idea that participants in different roles may have differing preference for hardware and software interfaces. For example, someone responsible for technical support may prefer software as this allows for remote support.

Again, the statistical test used is Fisher’s Exact Test due to the sample size being insufficient to use the Pearson Chi-Square method. As with the previous hypotheses, a P-value of less than 0.05 would indicate a significant result. Table 3.4 shows the cross-tabulation used within IBM SPSS to compute the P-value.

		A hardware control panel is preferable over a software solution				
		Disagree	Neutral	Agree	Strongly Agree	Total
Production Role	1	2	2	1	9	14
	2	0	2	6	6	14
	3	0	2	3	6	11
	4	1	0	3	5	9
	5	0	1	0	3	4
	6	1	0	1	4	6
Total		4	7	14	33	58

Table 3.4: Hypothesis Three – IBM SPSS Cross-tabulation

Fisher's exact test yields  $P=0.547$ , therefore the result is again not significant and the null hypothesis is accepted.

### 3.2.2 Stakeholder Interviews

In order to further explore the themes developed from the industry survey (see 3.2.1), stakeholder interviews were carried out on an individual basis with the key stakeholders identified in section 1.4.

The interviews took the form of verbal questioning with open-ended responses and follow-up questions. In order to efficiently analyse the qualitative data obtained, thematic analysis was used to identify interesting patterns and responses (Maguire and Delahunt 2017).

Table 3.5 breaks down the themes identified within the interviews, showing key patterns.

Table 3.5: Thematic Analysis of Stakeholder Interviews

Theme	Pattern
Control Functionality	Should control: iris, shutter speed, white balance, focus, and zoom. Control of recording stop/start not important.
Physical Form	Mounting should be shoe-mount compatible. Wired connections should make use of common cables and connectors. Master control unit with camera unit slaves if direct control not practical.
Power	Power over data cable preferred, external battery required for wireless modes. External PSU would be tolerable but not ideal. Internal batteries unacceptable. Batteries should be rechargeable.

<b>Continuation of Table 3.5</b>	
<b>Theme</b>	<b>Pattern</b>
Wireless vs. Wired	Wired preferred for perceived reliability. Wireless cameras would require wireless control too. If mixture of wired and wireless functionality, units capable of both with mode selection would be preferred. Redundancy good – could fail-over to wireless with status indication.
Protocol	Chained-bus topology preferred. Could be Internet Protocol – concerns raised around network configuration. CAN bus identified as suitable. Physical termination should be automatic. Master unit could also present USB interface for simple setup.
Compatibility	Must support LANC protocol control of cameras.
Capacity	Must support at least four cameras, more is better.
User Interface	Hardware panel easier to teach. Software panel probably necessary for configuration. Software preferred.
Usability	Must be clearly labelled. Simple setup desirable, plug-and-play excellent. Use of control interface should be as simple as possible. Must have clear user and technical documentation.
Safety	Must be durable to reduce risk of harm to users. Resistance to liquid damage desirable.
System Expansion	Must present external control protocol for future expansion. Could be network protocol to master control unit. Possibility to add pan and tilt control in future is highly desirable. Ability to support non-LANC cameras in future desirable, new camera units acceptable if required.
Bonus Features	Could include ‘call’ function to alert camera operators. Could include tally indication, needs to be easily seen. Call and tally must be easily distinguished between.
<b>End of Table</b>	

### **3.2.3 Summary**

In summary, the survey of industry professionals identified attitudes that were not as initially anticipated. In particular, the survey results initially questioned the use of wireless technology altogether. However, interviews with stakeholders identified that wireless technology is indeed necessary. Overall, a combined approach of both wired and wireless connections would be ideal. Stakeholder interviews have provided critical insights into the practical application of this project and will heavily influence the system design and requirements.

## 4 Requirements

This chapter lists the high-level requirements for this project. The requirements are based upon the work within the literature review and gap analysis chapters (2 and 3 respectively). Natural language specification is used to aid stakeholder understanding, with standardised formatting and keywords to reduce ambiguity (Sommerville 2016, pp. 120-122).

In this context, **must** indicates a requirement that has to be met in order for the project to be considered successful, **should** indicates a requirement that is highly desirable but not critical for the system to succeed, and **could** indicates additional requirements that would extend the project's usefulness. **Won't** denotes features that stakeholders have agreed will not be implemented but may be added in the future. Text in parentheses details the relationship between the requirement and the real-world, including why it is important and indicating *stakeholders* where appropriate. This format is commonly referred to as MoSCoW analysis (IIBA 2009, p. 102).

1. The system **must** control at least four cameras. (Typical *LSU Media* productions include approximately four cameras, more advanced productions will use more.)
2. The system **must** control at least the following camera functions: iris, white balance, focus, and zoom. (These are typical parameters that *camera operators* require assistance with from *technical personnel* during productions.)
3. The system **must** support LANC equipped video cameras. (All of the video cameras currently owned by *LSU Media* allow control via LANC protocol.)
4. The system **must** use cables and connectors that are commonplace. (Reduces overall investment required by *purchasing organisations* and increases the likelihood of *technical personnel* being able to replace broken cables quickly.)
5. The system **must not** use internal batteries for power. (Ensures constant power supply or easy battery replacement by *camera operators*.)



6. The system **must** use wired connections. (Satisfies reliability concerns of *technical personnel*.)
7. The system **must** have a software user-interface. (Allows increased portability of the system and gives *technical personnel* the ability to provide remote assistance.)
8. The system **must not** cause undue hazard to safety. (Safety of *all users* and *the public* is paramount.)
9. The system **must** provide an interface for external third-party control systems. (Ensures longevity of the system by allowing *purchasing organisations* and *technical personnel* to build custom integrations that communicate with the system.)
10. The system **must** be well documented. (Comprehensive documentation for both *users* and *technical personnel* will ensure the system is used to its full potential.)
11. The system **should** be shoe-mount compatible. (All of the video cameras currently owned by *LSU Media* have shoe-mounts for easy mounting of accessories.)
12. The system **should** fail to an inactive state. (If the system fails, for example due to a broken cable, *camera operators* would be able to manually adjust settings without interference from the system.)
13. The system **should** allow daisy-chaining of control cables. (Reduces cable requirements and allows for easier setup by *non-technical personnel*.)
14. If control cables require physical termination, the system **should** do this automatically. (Prevents system failure due to incorrect setup by *non-technical personnel*.)
15. The system **should** include tally indication. (Assists *camera operators* by notifying them when their camera is in use.)
16. The system **could** allow for future addition of pan and tilt control. (Ensures longevity of the system by allowing *purchasing organisations* to expand functions later.)
17. The system **could** include a camera call function. (Assists *gallery personnel* by allowing them to attract the attention of *camera operators*.)
18. The system **won't** be resistant to liquid-damage. (Increases durability of the system and increases safety of *all users* and *the public*.)
19. The system **won't** use wireless connections as a backup. (Redundancy satisfies reliability concerns of *technical personnel* as well as allowing control of cameras that are being operated completely wirelessly.)

## 5 Design

This chapter details the design phase of the project. The high-level system design is explored first followed by the hardware and software design. The main influences upon the design came from the stakeholder interviews through the requirements (3.2.2 and 4 respectively).

The system design is the level at which most of the requirements are tackled, the later hardware and software design stages can be viewed as supporting the needs of the overall system, with the exception of a few specific requirements.

### 5.1 System Design

Figure 5.1 shows an initial hand-drawn sketch exploring ideas of how to fulfil the requirements of the system.

The use of a CAN bus was attractive from the start of the design process due to the mature, well-understood nature of the technology. By selecting a CAN bus to carry the messaging between physical components of the system, requirements 6 and 13 are met. The additional requirement (19) of wireless capability is met in hardware through the use of LoRa packet radio, a low power and licence-free solution that can still provide good range of many kilometres indoors (Murata Manufacturing Co. Ltd. 2020). Requirement 19, as a MoSCoW won't, is intended as something that could be included in a future release. For this to be possible, the underlying hardware must provide a means of wireless communication which is why it is met in the hardware stage.

A system topology was needed to decide how data is controlled within the whole system. Since the system must be able to interface to external third-party systems (requirement 9), there is a possibility for unknown data and intentions to be present at a later date. For this reason, a master unit is incorporated as the boundary between the known and unknown portions of the system. The master is accessible by third-parties over internet protocol so that configuration and control can be easily established wherever needed. Any commands can be processed before they are sent over CAN or LoRa which should allow for better reliability. Additionally providing a simpler interface for third-parties since they would not

need to understand how the CAN bus or LoRa radio side of the system is implemented.

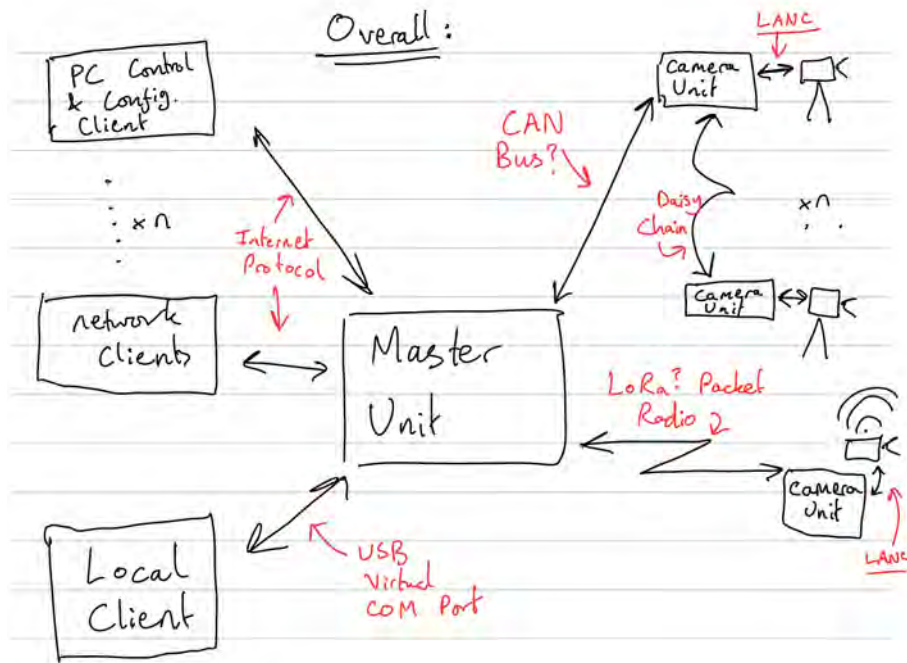


Figure 5.1: System Design Initial Sketch

Figure 5.2 shows the finalised overview diagram of the system. The system limits are defined, showing no practical limit to the number of control clients. A single local control client is shown, realised through a USB serial connection to the master unit, intended to allow use of the system without complex network setup if advanced features are not required.

The number of cameras that can be controlled is limited by the length of the CAN bus (for a given bus speed), the input impedance of hardware CAN transceivers, and by the number of addresses implemented in a chosen radio module. These limits are not likely to be exceeded in even the most advanced configurations since both should typically exceed 100 nodes. Whilst this may seem excessive when compared to the requirement (1) to control at least four cameras, it allows for future system expansion and would be impossible to change without hardware redesign.

An advantage of this system is the ability to add multiple types of Camera Control Unit (CCU) in the future. This could allow for cameras that require an interface other than LANC. One disadvantage of how the system is arranged is that a failure of the Master Control Unit (MCU) would result in failure of the entire system. However, the likelihood of failure can be mitigated during the hardware design stage.

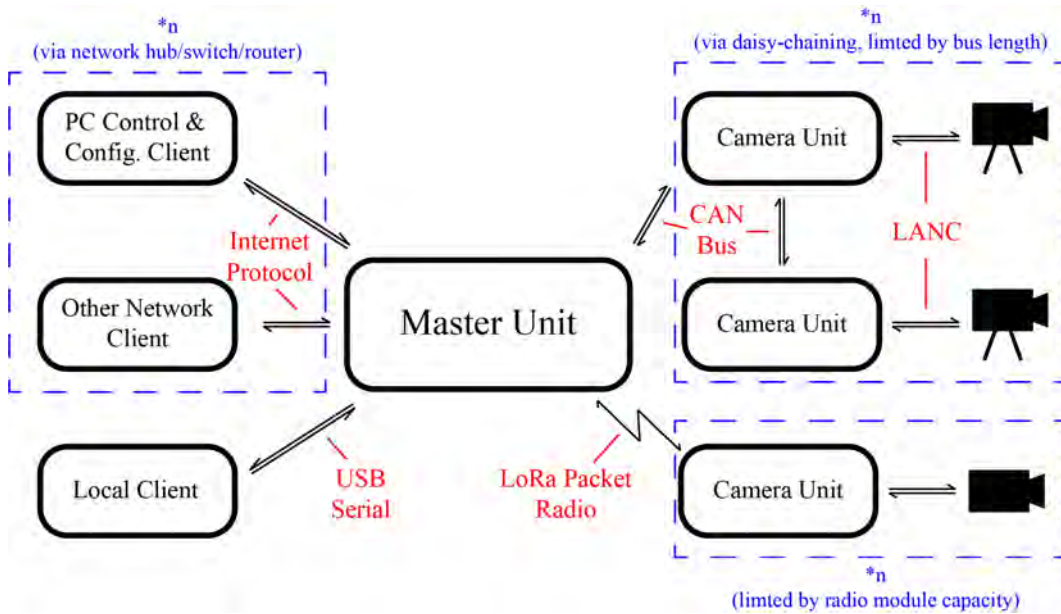


Figure 5.2: System Design Overview Diagram

## 5.2 Hardware Design

As a result of the overall system design (5.1), two unique hardware units are identified. This section details the design process for the hardware elements of the Master Control Unit followed by the Camera Control Unit.

Before the either hardware element could be directly tackled, a solution to several universal hardware design requirements was needed, these are explored first.

### CAN Bus Termination

A CAN bus has a chain topology with differential signalling lines, requiring termination resistors at each end of the chain. A typical solution to this is manual termination by skilled system users. However, requirement 14 specifies automatic termination if possible. One simple method for automatic termination could have been to designate separate ‘in’ and ‘out’ CAN ports on each device then make use of unused conductors to provide neighbour sensing using voltage levels. Such a solution, whilst simple, has potential for failure if devices are incorrectly connected together. To provide an intelligent solution, a simple low-speed signalling protocol was desired, preferably single-wire based so as to not require termination itself as well as to reduce need for many conductors in cabling between devices.

A potential protocol to use could have been the Dallas 1-Wire protocol since it meets the single conductor and no termination requirements. Unfortunately, 1-Wire is a single master, multiple slave protocol which would not work well for the intended purpose since two neighbouring nodes would have to configure themselves to be a master and a slave without any communication between them (Maxim Integrated Products Inc. 2008).

An alternative identified was Padded Jittering Operative Network (PJON). PJON is an open-source protocol that can be implemented with different ‘strategies’, one of which makes use of a single wire physical layer and all of which make use of addressing (including broadcast) which lends itself better to this application. Another advantage of PJON’s single wire strategy (Padded Jittering Data Link, PJDL) is that it can be implemented in software on very simple microcontrollers, this supports hardware design where automatic termination is handled by a separate microcontroller since little extra resources are required.

During experimental testing of suitability for this project, PJDL was found to work on wires up to 2km long. Since automatic termination only needs to work over the length between each CAN node, as opposed to the length of the whole CAN bus which is limited to 1km at the slowest speed, PJDL’s 2km range is more than adequate. The complete details of testing carried out to determine the suitability of PJON for this purpose are included in appendix B. As a result of satisfactory testing and related support from the protocol’s developers, PJON was selected for use in automatic termination of the CAN bus.

### **Inter-Device Cabling**

The cabling between devices must carry multiple separate signals (CAN bus high, CAN bus low, PJDL, Ground) as well as preferably providing power to some nodes. This rules out cables commonly used for audio or video since they typically do not have enough separate conductors to carry all the signals. The most common cable that meets the requirements is structured network cabling such as category 5e or category 6, containing four twisted pairs for a total of eight conductors. Network cabling is usually connected with RJ45 type connectors which are also suitable for the needs of this project.

A potential risk with network cabling is the existence of ‘crossover’ cables where one pair of conductors is swapped with another to service legacy Ethernet devices. With bad design, a crossover cable could, for example, reverse the power supply voltage or connect the power supply to the CAN bus. However, due to the well-defined nature of which pairs are swapped, it was relatively simple to mitigate this risk. Table 5.1 shows the conductor assignment chosen so that network cabling could be used for this project.

In a crossover cable the orange and green pairs are transposed which, for the chosen signal assignment, would result in no change since the solid marked and dashed conductors of those pairs each carry the same respective signal (power supply and power ground). This

has the added advantage of providing supplementary conductor area which allows for a greater electrical current to pass safely when compared to using only a single pair for power.

Table 5.1: Cabling Conductor Signal Assignment

RJ45 Pin	Conductor Marking	Signal Carried
1	Orange Dash	Power Ground
2	Orange Solid	Power Supply
3	Green Dash	Power Ground
4	Blue Solid	CAN High
5	Blue Dash	CAN Low
6	Green Solid	Power Supply
7	Brown Dash	PJDL Ground
8	Brown Solid	PJDL Data

### 5.2.1 Master Control Unit

In order to meet the design requirements of being able to interface with internet protocol as well as CAN and LoRa, a design based around a single board computer was created. This allows use of a Linux based operating system to facilitate simple third-party facing software design, with interfacing to an ARM microcontroller on a custom Printed Circuit Board to support the CAN and LoRa requirements. Figure 5.3 shows how these components are connected, with the addition of several other features.

A Real Time Clock was included to allow accurate timekeeping from the Single-Board Computer even if an internet connection is not present. An OLED display is also present to provide basic information to users, such as system status and connection details. The design makes use of commercial-off-the-shelf power supplies to handle mains alternating-current to low voltage direct-current conversion. Power supply protection components are included in the custom PCB part of the design to increase system safety.

An ARM microcontroller is specified to handle the communication over CAN and LoRa due to the ARM family's relatively fast speed, 32-bit architecture, and large program memory. An AVR microcontroller is additionally specified to allow separated handling of automatic CAN bus termination using PJON. Due to PJON's minimalist hardware requirements, it can run on some of the smallest AVR-based microcontrollers available. The choice to separate PJON onto a distinct microcontroller should simplify software design whilst posing little hardware overhead due to the simplicity of AVR microcontrollers.

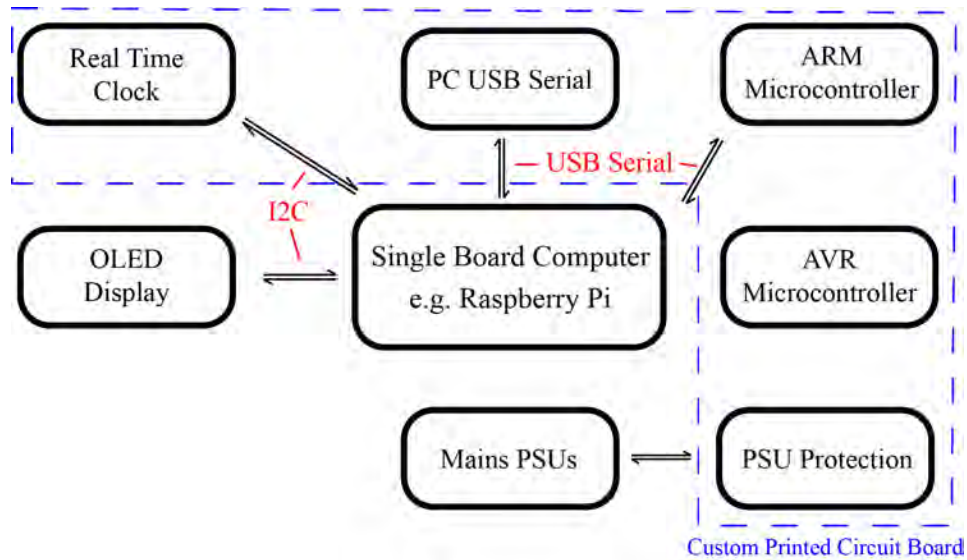


Figure 5.3: Master Control Unit Overview Diagram

Figure 5.4 shows the arrangement of the ARM subsystem. The CAN controller and transceiver provide interfacing between the ARM microcontroller and the CAN bus using Serial Peripheral Interface (SPI), a hardware interface common to most ARM microcontrollers. The LoRa radio module is also interfaced using SPI to allow use of common modules and software libraries. A temperature sensor and power supply status monitoring are included for added safety and information for the user. Hardware settings make use of the microcontroller’s General Purpose Input/Output (GPIO) to configure any settings that may not be possible or desirable to configure using a software interface.

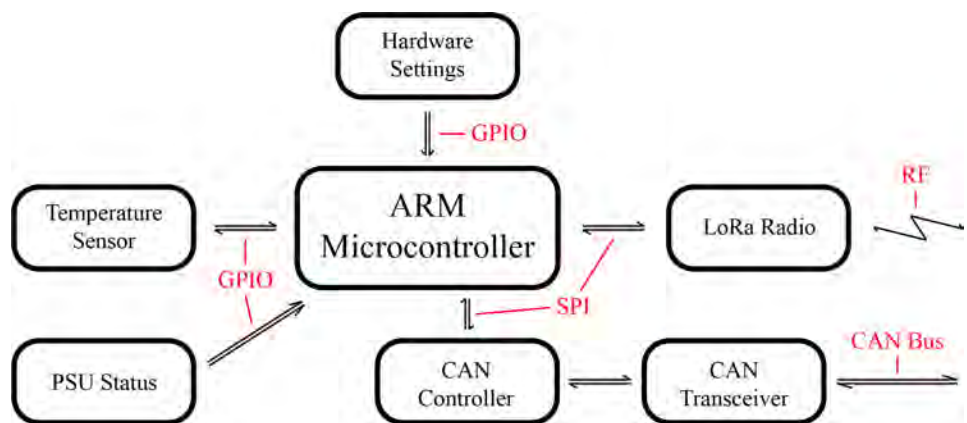


Figure 5.4: Master Control Unit ARM Diagram

Figure 5.5 highlights the simplicity of the AVR subsystem. GPIO is used for all functionality, resulting in no requirement for support of complex hardware interfaces on the microcontroller. The PJON bus allows for detection of neighbour nodes so that appropriate action can be taken. A relay is used to allow the CAN termination resistor to be switched in or out of the circuit as required. Status output to LED allows the termination state to be seen locally to the device to aid troubleshooting. Additional status output to the ARM microcontroller allows termination status to be read remotely from software if necessary.

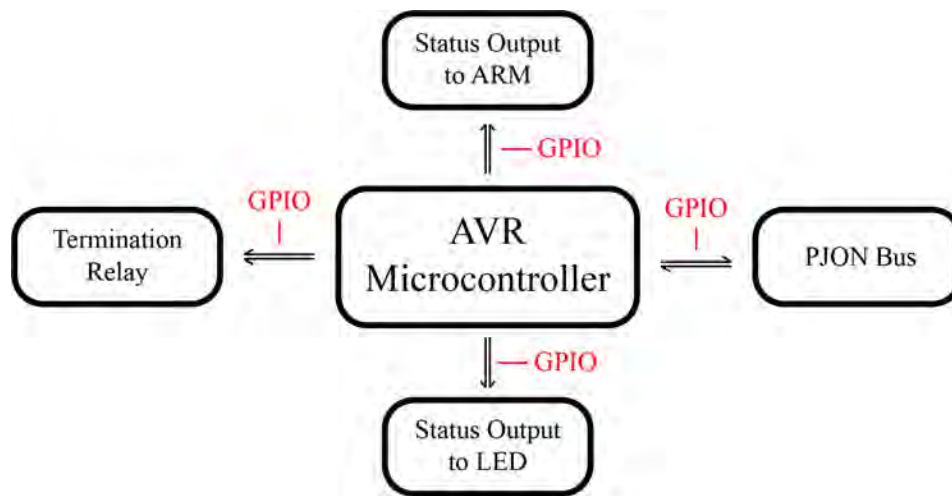


Figure 5.5: AVR CAN Bus Auto-Termination Diagram

### 5.2.2 Camera Control Unit

The Camera Control Unit design is comparably simpler than the Master Control Unit as can be seen in figure 5.6. A custom PCB is used for the whole design in order to reduce size and increase reliability. Similarly to the Master Control Unit, an ARM microcontroller is used to provide the bulk of processing capability with an additional AVR microcontroller to handle automatic CAN bus termination. Power supply protection is again included to ensure maximum safety.

Note: The AVR subsystem design is identical to that of the Master Control Unit so is not repeated in this subsection, see figure 5.5.



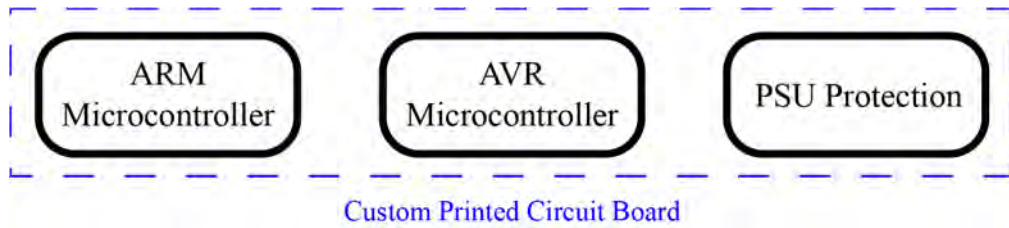


Figure 5.6: Camera Control Unit Overview Diagram

As can be seen in figure 5.7, the CCU ARM subsystem design is based on the MCU design with a few alterations. Importantly, a LANC circuit is added to allow control of cameras. Tally lights are included to provide information to camera operators, meeting requirement 16. An RS-232 interface is added so that the UART hardware transceiver common to most ARM microcontrollers can be interfaced to future additional hardware, for example a pan and tilt motor controller.

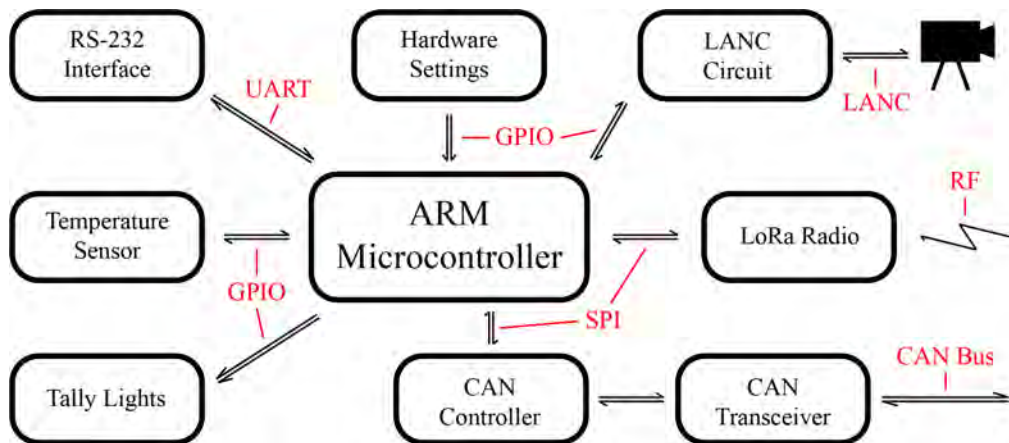


Figure 5.7: Camera Control Unit ARM Diagram

### 5.3 Software Design

The software design element of the project was necessarily influenced by the hardware design. As a result of the chosen hardware design, there are three major groups of software required: AVR microcontroller, ARM microcontroller, and Single-Board Computer. This section details the design of each of these software groups.

Note that some features included in the hardware design are not part of the software design since they are not mandatory requirements of the project as a whole which aims only

to achieve Minimum Viable Product (MVP) functionality as discussed in 1.2.

### 5.3.1 AVR Microcontroller

The required functionality of the AVR microcontroller software is relatively straightforward. It must use PJDL to determine if there are neighbours on both sides of the bus and if so terminate the CAN bus by actuating a relay. In addition, it must signal the termination status on GPIO pins.

### 5.3.2 ARM Microcontroller

The purpose of the ARM microcontroller software is to carry out the computation required on the PCBs, especially any interfacing to the physical layer. The software design for the ARM microcontroller in the MCU and CCUs can be viewed as supporting the desired functionality of the overall system within the constraints and limitations imposed by the hardware design.

The MCU and CCU ARM software have a wide overlap since they both need to use the CAN bus to communicate as well as interface with other hardware components that are identical on both circuit boards. Both pieces of software aim to be as simple as possible for their intended purpose to avoid stretching the limited resources of their microcontroller hosts.

#### Master Control Unit

The design of the MCU ARM software is to facilitate the following:

- Configure IC chips and read hardware settings on start-up.
- Communicate over USB serial protocol with the Single-Board Computer.
- Provide MCU hardware status to the SBC.
- Formulate commands received from the SBC into messages sent on the CAN bus to CCUs.
- Receive CAN messages from CCUs and pass them to the SBC.
- Display basic status information via PCB hardware (e.g. LEDs).

### Camera Control Unit

The CCU design differs slightly from the MCU, requiring the following facilities:

- Configure IC chips and read hardware settings on start-up.
- Communicate over the CAN bus with the MCU.
- Provide CCU hardware status to the MCU.
- Convert messages from the MCU into LANC commands to an attached camera.
- Display tally status on PCB LEDs.
- Signal fatal error states to the user so that they can report issues.

The final point, signalling fatal errors, is important since it allows a camera operator to be aware of CCU failure. This should prevent potential issues such as camera operators relying on incorrect tally status in the event of CAN controller IC failure.

### 5.3.3 Single-Board Computer

The software for the SBC is the boundary between the internal workings of the system and the outside world in so much as all external software must interface through the software on the SBC. This can be broken down into two distinct parts which could actually be run from entirely different computers but for this system they make most sense to be both provided by the SBC.

The first part, referred to as the web API, is a server program connecting to the MCU's USB serial port and providing a web interface to external components. The second part, the web GUI, is an entirely client-side processing website being served from a static web server on the SBC.

#### Web API

The purpose of the web API server is to maintain the state of the system and provide software access to control it. This requires a program with the following features:

- Communicate with the MCU via USB serial.
- Maintain the state of the system.
- Expose a web API for external software to interface through.
- Retrieve tally status from a networked camera switcher.

## Web GUI

Whilst the web GUI software actually runs on the user's computer, it is accessible from a web server on the SBC so for the purposes of this discussion is classed as SBC software. The purpose of the web GUI is to provide the main user interface to the system, allowing oversight and control of all features. This requires the following design features:

- Use client-side technologies to communicate with the web API.
- Provide a clear overview of the system status.
- Allow user control of connected cameras.
- Include user guidance documentation.
- Serve all components locally without need for an internet connection.

By serving all components required to run the client-side aspect from local sources, the web GUI will be useable without internet connection. This is critical since it is common for the control network used in broadcasts to be isolated from the internet for security and stability reasons.

The reason for providing the main user interface to the system as a website is to prevent the need for software installation on user computers. This way, the software can easily be maintained to a current version without need for administrator assistance to upgrade physical machines. This has an additional benefit of allowing flexible use of computer equipment during broadcasts as there is no concern over which computers have the required software as only a standard web browser is required.

## 6 Implementation

This chapter details the implementation of the project based around the designs produced in chapter 5. Starting with the hardware implementation, including PCB and enclosure production, then moving on to discuss the software implementation.

### 6.1 Hardware Implementation

#### 6.1.1 Electronics

Before anything else could be completed, the electronic implementation was needed in order to allow for manufacturing of PCBs in a timely manner. Completed PCBs also allow dimensions to be known so that enclosures can be created.

##### **Power Supplies**

A major decision within the electronic implementation was how to handle power supplies. As identified in 5.1, the Master Control Unit could be a single point of failure within the system and so dual, redundant power supplies have been used. This way, in a mission-critical application, two diverse sources of power can be used such that if either one fails, the MCU would not. This resulted in two off the shelf mains power supplies being used to feed a single device. To counteract electrical issues that could arise during various power supply failure modes, advanced eFuse hot-swap controller Integrated Circuits (ICs) were used. An eFuse can protect against multiple fault types including reverse-current, reverse-polarity, over and under-voltage, as well as over-current conditions, the only protection an ordinary fuse provides (Texas Instruments 2018).

Next, power supply voltage levels needed to be determined. As previously shown in table 5.1, the design calls for power to be carried over the control cabling which is structured network cabling. Since network cabling is typically constructed using a thin gauge conductor, the current carrying capacity is limited. In order to allow the most devices possible to be powered in this way, the voltage needs to be as high as possible so that current is reduced

for the same power according to the voltage-current product law,  $P = IV$ . The highest DC voltage that is widely considered safe for deployment by non-electricians is around 50V. For this reason, 48VDC was selected as the power supply voltage to be carried over the bus wiring, the same voltage level used by Power-over-Ethernet systems.

Since the CCU power supply provision over control cabling was limited and not appropriate when used in a wireless-only mode, a second power supply input for CCUs was needed. In order to maintain compatibility with a wide range of common power supplies, an eFuse redundant system was again used. This way, CCUs can be powered either over the control cabling or with external power supplies, with the option to supply both as well.

For both the MCU and CCU circuits, a switch-mode voltage regulator design was used to convert the 48V transmission voltage to a useful 5V supply. A switch-mode design was selected for its efficiency and, as an additional benefit, less heat is produced which simplifies design considerably. For 3.3V supply, an Low-Dropout Regulator (LDO) was used to convert the 5V supply. LDOs are a linear design so convert excess voltage into waste heat however the amount of waste energy is low in the conversion from 5V to 3.3V therefore this is not a concern.

### Connectors

In order to ensure robustness and ease of use, connector choice was very important. The design specified that network cables would be used for the control bus lines however a typical RJ45 connector is not particularly robust. For this reason, the considerably more expensive and superior Ethercon connector was used. Ethercon retains compatibility with standard network cables whilst also allowing the use of an additional metal shell around the male RJ45 connector, greatly reducing chance of breakages.

For the CCU external power input connector, a 5.5mm by 2.1mm DC barrel style connector was chosen since it is by far the most common connector found on low-voltage power supplies. This way, any readily available power supply should be suitable thanks to the combination of wide input voltage range and common connector.

The wireless antenna connection uses an SMA connector since this is one of the most common antenna connectors with  $50\Omega$  impedance that is an appropriate size for the application. Suitable antennas for the operating frequency are readily available with SMA connectors.

The LANC connector could not easily be a common connector since the camera end of the cable, in most cases, needs to be a 2.5mm plug which is uncommon. Since the cable would have to be custom made regardless of the CCU connector, a 5-pin MiniDIN connector was used. MiniDIN is far more robust than a 2.5mm plug and has the advantage of not being likely to be incorrectly connected since it is not very common.

For the RS-232 connector, a 3-pin MiniDIN was used. This maintains the size and style of other connectors used as well as preventing incorrect connection with the LANC port as a result of MiniDIN connector keying.

### Integrated Circuits

Many Integrated Circuits (ICs) were used to save space and make electronic design easier. Particular ICs worth noting include the eFuses already discussed, integrated CAN controller and transceiver, UART to RS-232 interface, GPIO expander, and microcontrollers. An advantage of many of the ICs selected is the availability of open-source software libraries that simplify their use from software.

The CAN controller and transceiver used (MCP25625) are integrated into a single package to reduce space occupied and provide easy circuit design. Multiple open-source libraries exist for this IC which would allow reduced workload during software implementation.

The RS-232 interface IC (MAX3232) allows use of 3.3V UART interface common to most microcontrollers to provide a standards compliant 12V RS-232 interface all without extra power supply voltages. This also allows the hardware UART of the microcontroller to be used as an RS-232 interface with no additional software requirements.

A GPIO expander IC (MCP2008) was used to connect hardware address coding switches to the ARM microcontroller without requiring 8 GPIO pins. By interfacing over I<sup>2</sup>C, the address coding switches can be read from software without a large pin requirement. An additional advantage of the MCP2008 is built-in pull-up resistors which reduce the total component count.

For the AVR microcontroller, the ATtiny84A was selected since it measures only 3mm by 3mm and provides just enough functionality for its purpose without becoming unnecessarily difficult to use as it has 8 kilobytes of program memory. The ARM microcontroller used is the ATSAMD21G18A, with an ARM Cortex-M0+ based instruction set, offering moderate performance of 48MHz clock speed with low power consumption and a wide-range of open-source libraries available due to its use in many Arduino derivatives.

### Single Board Computer

For the MCU Single-Board Computer, a Raspberry Pi model 3B+ was selected. Despite a newer model, the 4B, having been released, the 3B+ is preferable since it has enough performance for the application whilst not demanding as much power as the 4B and also not requiring active cooling. Whilst there are many other manufacturers of SBCs, the Raspberry Pi foundation has far better product support and software stability, making it preferable since this project has a large scope without a custom compiled Linux kernel.

### Printed Circuit Boards

Once the main blocks of the electronic design had been identified, electronic schematics were produced using Eagle CAD to detail the complete circuitry and components required. From these schematics, PCB layout designs were created so that the needed Computer-Aided Manufacturing (CAM) files could be sent to the fabrication factory. Complete schematics, board layouts, and bill of materials for each PCB are included in appendix C. Figure 6.1 shows an example of the view seen when using Eagle CAD to design PCB layouts.

Due to the small quantity of circuit boards required, the only cost-effective PCB fabrication companies are located in Shenzhen, China. As a result of initially the Chinese new year, followed by holiday extensions and quarantine procedures due to the coronavirus pandemic, PCB fabrication was delayed by several months. Once combined with domestic complications due to coronavirus, this resulted in complete use of the original contingency period allowed.

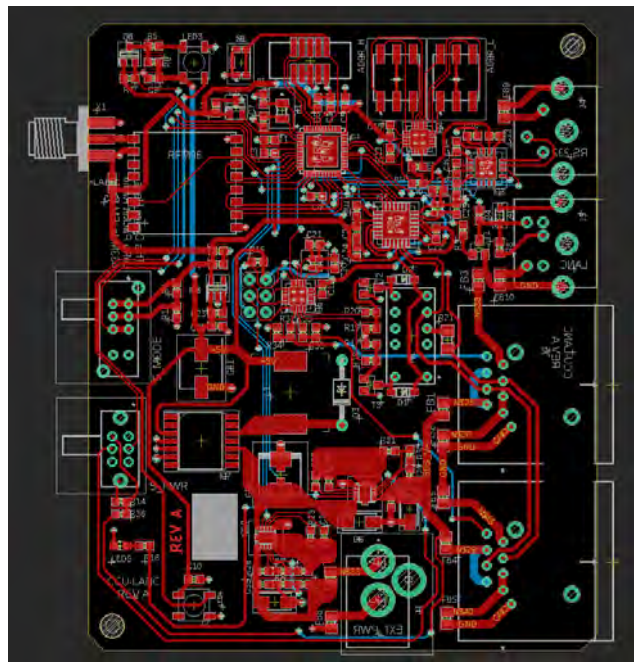


Figure 6.1: PCB Layout Design

Once the PCBs and electronic components were eventually received, assembly began immediately. This involved first depositing solder paste onto the circuit boards, using a stencil produced by the fabrication company using the CAM solder paste layer. Next, the surface-mount electronic components were placed into their respective positions by hand,



the smallest of which measuring just 1.6mm by 0.8mm – less than a grain of rice. Figure 6.2 shows a circuit board with solder paste applied after some of the components had been placed. In mass-production, placement of surface-mount components is carried out by a pick-and-place robot however this is beyond the resources of this project.

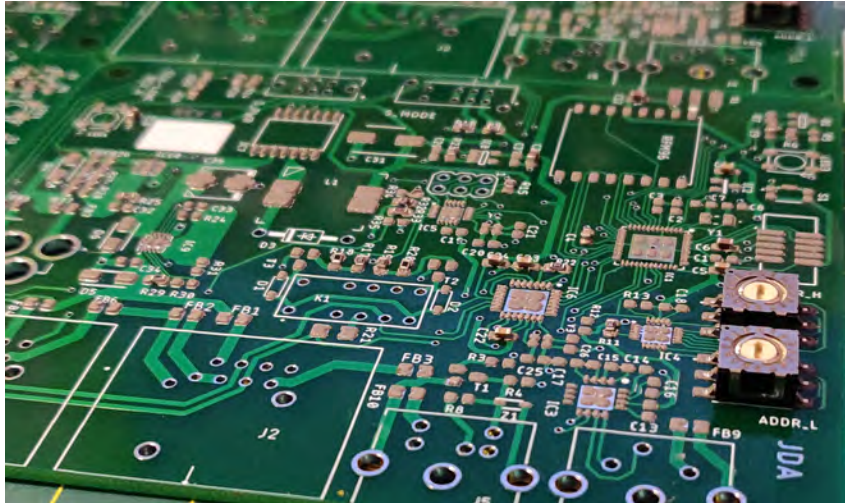


Figure 6.2: PCB Assembly Component Placement

Once all of the surface-mount components had been placed onto each of the PCBs, the circuit boards were heated in a home-made reflow oven to a specific temperature profile. This allowed the solder paste to melt and the components to become electrically connected to the board without damage from overheating. Once the boards had cooled down, they were visually inspected for quality of solder joints. Due to the low-precision nature of the equipment and placement techniques used, some solder bridges were found which were then manually fixed using a hot-air rework station. After another thorough inspection, the surface mount components were finished. Finally, the through-hole components, such as connectors and switches, were hand soldered into place completing the PCB assembly process. Figures 6.3 and 6.4 show the completed CCU and MCU PCBs respectively.

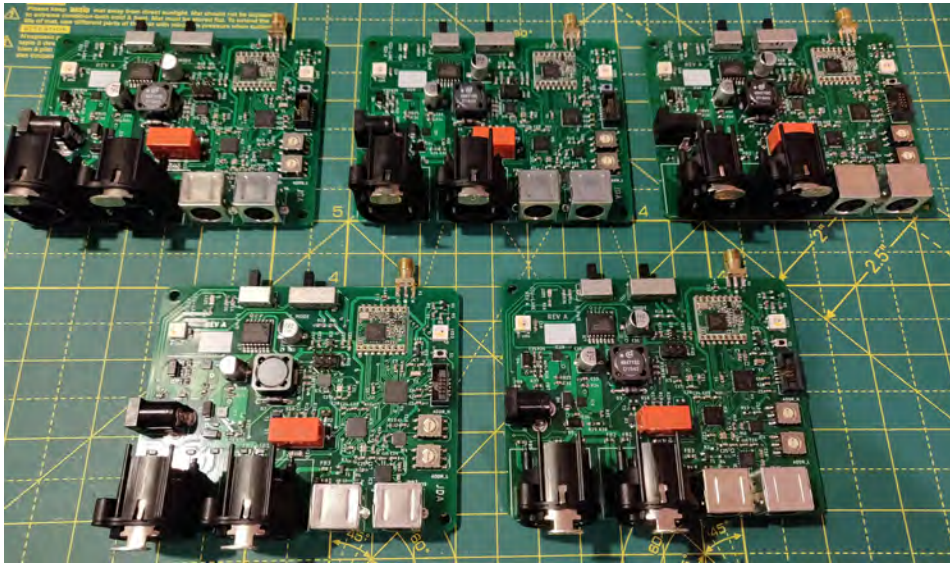


Figure 6.3: Assembled CCU PCBs

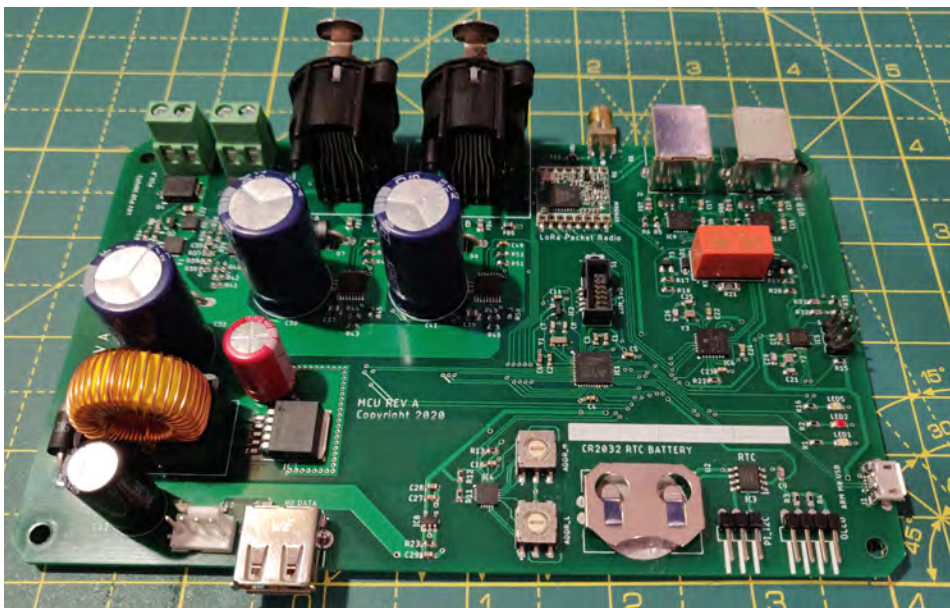


Figure 6.4: Assembled MCU PCB

### 6.1.2 Enclosures

This subsection explores the enclosures created to protect the electronics and provide suitable mounting methods. Due to limited resources, manufacturing methods were limited to low-cost processes that the author was able to carry out. A final product being brought to market could make use of advanced technologies such as injection-moulded plastic which would give a more refined result. The two primary influences on enclosure design were ensuring that suitable mechanical protection was given to electronics as well as compatibility with existing broadcast hardware.

#### Master Control Unit

A standard 19-inch rack-mounted box, similar to those used in computer server rooms, was chosen for the MCU enclosure. It is common for broadcast hardware to be built into mobile flight cases for easy transport, these are typically furnished with 19-inch rails for hardware mounting. By selecting a metal rack-mount box, compatibility with industry standard practices could be met whilst also giving a metal shell that could be earthed for proper electrical safety. It is anticipated that this implementation would easily pass a portable appliance test, for example.

To prevent risk of damage to the MCU PCB, connectors exposed on the rear of the enclosure were ‘through’-style such that they could be easily replaced if broken, without need to repair the circuit board.

#### Camera Control Units

The enclosures for the CCUs were designed using Solidworks CAD so that they could be manufactured using the author’s 3D printer. This allowed for rapid prototyping where several design iterations could be trialled over the course of a few days. Several different ideas were tested, primarily to ensure that parts fitted together correctly despite the relatively poor tolerances of a non-commercial 3D printer. Additionally, trials were conducted to determine how best to label switches and ports. The design that worked best was to emboss the text into the part surface.

Figure 6.5 shows the base part, designed to hold the PCB. The hole marked ‘A’ allows a heat-set brass insert to be used to provide a threaded hole for securing the PCB with screws. The hole marked ‘B’ allows a similar, larger, insert to provide a thread for mounting adapters. This threaded hole provides a 1/4-inch thread commonly found on camera tripods, microphone stands, and shoe-mount adapters which is the reason for its inclusion.

Above hole ‘B’, a square cut-out allows a piece of 1.5mm thick steel to be inserted during assembly. This was added to prevent mounting screws being over-inserted into the enclosure

and risking damage to the PCB above. The steel piece is held in place by a 3D-printed hexagon plug that is affixed with cyanoacrylate glue into the area marked 'C'. The use of a hexagonal shape was a compromise between a large surface area for glue adhesion and a shape that would reliably print. A slotted design could have been used instead however this would have been difficult to print due to overhangs as well as requiring strength in a plane that is relatively weak in standard 3D prints.

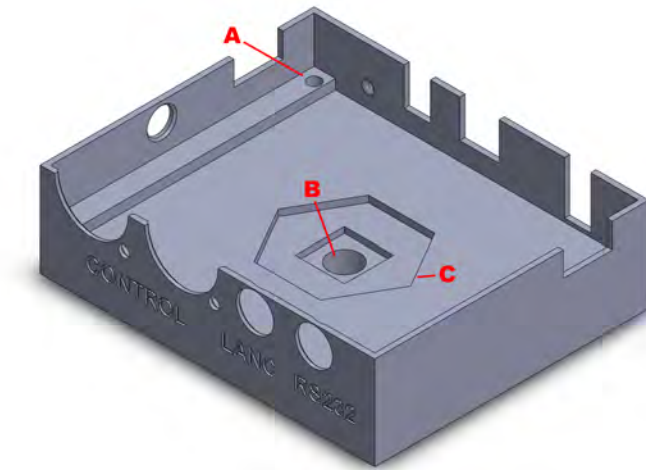


Figure 6.5: CCU Enclosure Base CAD View

In order to pass light from the tally LEDs through the enclosure, square holes in the enclosure base and top were created. Into these holes, a part printed using translucent plastic was inserted, the design of which is shown in figure 6.6. To ensure most of the light was directed through the translucent material, a reflector was glued to the sloping arms at the back. This was easily achieved with the shiny side of common kitchen foil.



Figure 6.6: CCU Enclosure Tally CAD View

Figures 6.7 and 6.8 show photo-realistic renders of the CCU enclosure, complete with base, lid, and two tally parts as well as Ethercon connector 3D designs provided by the connector manufacturer. Complete mechanical drawings of the 3D printed parts can be found in appendix D.



Figure 6.7: CCU Enclosure Assembly CAD Render 1



Figure 6.8: CCU Enclosure Assembly CAD Render 2

In order for the 3D designs to be printed, they had to be sliced into layers and commands that could be understood by the 3D printer. Figure 6.9 shows an example view of this sliced representation. In total, to create all prototypes and final parts, around 1kg of 3D printer filament was used. The plastic used was PLA which has a reduced environmental impact compared to conventional plastics since it is plant-based. At end-of-life it can be recycled, commercially composted, or incinerated however the use of metal inserts and cyanoacrylate glue may compromise the ease of recycling in this case.

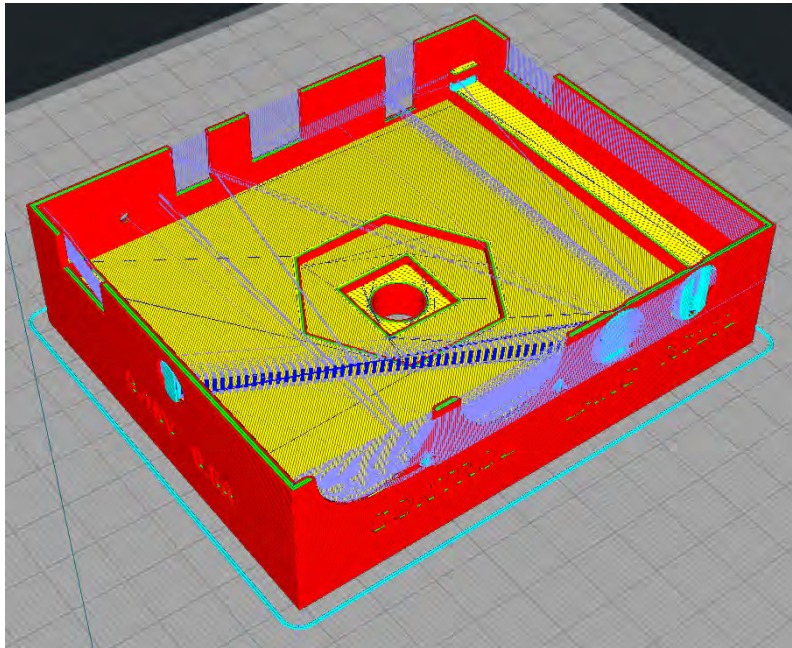


Figure 6.9: CCU Enclosure Base Slice View

Figure 6.10 shows a completed CCU enclosure which, whilst not perfect, achieves a high standard finish. The translucency of the tally parts and the effectiveness of the reflector design was a particularly good outcome, slightly better than anticipated.



Figure 6.10: Completed CCU

## 6.2 Software Implementation

This section details the software implementation, again it is important to note that due to this project's limited resources, the implementation aims to produce a Minimum Viable Product. Due to requirements for future improvement possibilities, the hardware implementation went further than the MVP level since reproduction of hardware would be costly. This is not the case for software so some hardware features are not implemented at the software level in this project.

### 6.2.1 AVR Microcontroller

Implementation of the AVR software was fairly straightforward, requiring only use of the previously discussed PJON library (Mitolo 2020a), licensed under the Apache license version 2, as well as core files (Konde 2020), licensed under GPL version 3, to allow use of the Arduino IDE to program the ATtiny84A microcontroller in C++.

The program begins by initialising the microcontroller pins and PJON buses. The program then repeatedly loops, first attempting to determine if there is a neighbour on a particular PJON bus by sending a packet and checking for acknowledgement. In order for this to be successful, the program then spends a relatively long time in PJON receive mode. This ensures that packets from neighbours are acknowledged. After multiple loops of sending

a packet and receiving on both PJON buses, the termination state is set based on whether a neighbour has ever been seen on both buses. This repeated attempt to send and receive prior to deciding if a neighbour is present helps to reduce false negatives that could be, for example, caused by packet collision.

After the decision whether to terminate the CAN bus is made, the entire process loops so that neighbours must again be found. This allows detection of neighbours being removed from the bus which in turn allows dynamic connection and disconnection of nodes from the overall system without the need to restart any of the hardware devices to achieve correct CAN bus termination.

In total, the compiled program uses 4,478 bytes or 54% of the ATtiny84A's flash program memory and 87 bytes or 16% of the available SRAM for global variables. This could be improved through the use of compiler link time optimisation however the gains would be minimal and the program in its current state does not come close to consuming all of the available resources.

### 6.2.2 ARM Microcontroller

Both ARM programs were implemented in C++ using Atmel Studio 7 (itself relying on some core files from the Arduino IDE) to compile and program the microcontroller memories. The main shared functionality between the MCU and CCU programs was communication over the CAN bus through the MCP25625 controller IC. For communication to be successful, a protocol on top of CAN needed to be defined so that devices could correctly process received messages. The extended CAN frame format was chosen which allows a 29 bit identifier field and an 8 byte data field.

Due to the way CAN medium access arbitration works, identifiers with a lower numerical value are effectively prioritised. To ensure proper utilisation of the CAN bus, the structure of identifiers had to be carefully considered. It is important to note that a CAN node can use multiple identifiers, in this way it is not strictly address-based identification since CAN is effectively a data streaming protocol. The information that this system needs to pass in the identifier field is: command number, device type, device address.

The command number decides what the data portion of the message means, for example a command to display tally status would inform the recipient that the data field contains four bytes representing red, green, blue, and white colour values. The device type indicates the type of device the information pertains to, for example an MCU or CCU. The device address represents the value physically encoded using hardware switches on the PCBs to uniquely identify a particular device of a particular type.

By carrying all three of these values in the identifier field, it is possible to use CAN's remote transfer request frame to send an identifier in the form of a question, with a specific



device responding with the same identifier as a data frame. For example, a remote transfer request for a CCU's status would be formed of the command number for device status, the device type for a CCU, and the targeted CCU's address. The response would be the same with a data field containing status information. Additionally, keeping these values separated allows command numbers to be specific to the device type targeted, providing plenty of scope for system expansion in the future.

The order chosen for the identifier information was the most significant 8 bits holding the command number, this way the command number determines message priority. Next, 5 bits of padding to fill the identifier field to correct length followed by 8 bits of device type then 8 bits of device address. Keeping each segment to one byte long simplifies programming as single-byte variables can be used before bit-shifting into correct places in the identifier. The padding bits were placed in the middle since, due to the way extended CAN identifiers work, part way through the padding other bits within the CAN frame header are positioned. This should make it easier to read the CAN identifier on an oscilloscope, if it was ever needed for debugging purposes.

To interface with the MCP25625 CAN IC, a library for the MCP2515 (Pereslegin 2020), licensed under the MIT licence, was used. This was possible since the MCP2515 is equivalent to the controller portion of the MCP25625. The library uses an object-oriented approach and provides access to the IC's incoming message filters which allow unwanted messages to be filtered out without use of the microcontroller's processing time.

### Master Control Unit

Implementation of the MCU program used two open-source libraries. These were the previously discussed MCP2515 library and the Adafruit MCP23008 library (Adafruit Industries 2019), licensed under the BSD licence. The MCP23008 library provides an object interface to the GPIO expander which is used to read the hardware address coding switches. These switches provide an 8-bit device address.

The program begins by initialising the microcontroller pins and connected ICs. Since the MCU processes CAN messages between all devices, the incoming filter is configured to accept messages with any identifier. The program then repeatedly loops, processing any received CAN or USB serial messages. No messages are sent automatically on an interval basis. Automatic status messages to the SBC could reduce requests for information however this introduces the risk of saturating the serial link with unimportant messages.

In order to make use of temperature sensors onboard CCUs, the MCU needed to be able to receive floating-point data types over the CAN bus. Since CAN and the library used only transport raw bytes, a solution was needed to convert between an array of bytes and a float type. The following code shows the function developed:

```

float bytesToFloat(unsigned char byte0, unsigned char byte1,
                  unsigned char byte2, unsigned char byte3){
    union {
        unsigned int input;
        float output;
    } u;

    u.input = (byte3 << 24) | (byte2 << 16) | (byte1 << 8) | byte0;

    return u.output;
}

```

The solution developed was particularly interesting since it will produce the correct result regardless of whether the sender or receiver are little or big endian architectures. This is achieved by defining the first data byte to be the least significant, increasing until the fourth data byte being the most significant. The function uses this definition to bit-shift the individual bytes into the correct bits of a four-byte integer. By using a C++ union, the same four bytes of memory can be written as an integer and then read as a float.

The compiled program uses 28,140 bytes, or 11%, of the microcontroller's program memory and 4,160 bytes, or 13%, of the SRAM. This is as expected since the limiting factor when selecting a microcontroller model was the hardware interfaces offered. There remains plenty of memory for future expansion of program functionality, a desirable situation to be in.

### Camera Control Unit

The implementation of the CCU program used the same libraries as the MCU for reading the MCP23008 and communicating over CAN. Unlike the the MCU, the CCU uses the CAN IC's message filters to only accept messages matching the device type and address of the CCU. This means that the microcontroller does not need to waste processing time deciding whether messages received are relevant. Additionally, the Adafruit Neopixel library (Adafruit Industries 2020), licensed under the GNU LGPL, was used to handle control of the tally LEDs. This allowed simple addressing of each LED separately without the need to implement the strictly timed data protocol from scratch.

In order to communicate with cameras using LANC protocol, a heavily modified version of code by Rosén (2017), licensed under the GNU GPL, was used. The main reason for modification was to simplify the code as this project does not make use of the relatively meaningless responses from the cameras. Since different cameras implement LANC

differently, the CCU program forms LANC messages from bytes sent by the MCU. This way, different cameras can be used with the CCU by simply defining new configurations of LANC commands on the SBC.

Similarly to the MCU, the CCU program operates by initially configuring the microcontroller and connected ICs. In order to notify the user of errors during startup, the CCU can pulse the tally LEDs red. Upon successful start-up, the program continues by repeatedly looping, processing any accepted CAN messages by displaying tally status, sending a command to an attached camera, or responding with a CAN message back to the MCU. A reverse version of the float unpacking function is used to format the CCU temperature into bytes for sending over CAN.

The compiled CCU program uses more program memory than the MCU due to the need for extra hardware interfacing routines. The program memory required is 37,768 bytes, or 14%, and the SRAM used is 4,148 bytes, or 13%. Again, this leaves a healthy amount free for future improvement.

### 6.2.3 Single Board Computer

The SBC programs fall into two parts, as discussed in the design (see 5.3.3). The web API is implemented in Python due to its high-level approach and speed of development. The web GUI runs as client-side JavaScript, depending on the functionality of the web API. These two parts are now explored in more depth, starting with the API.

#### Web API

The web API is implemented as a Python program, using the Flask framework (The Pallets Projects 2020). Flask, which is licensed under the BSD licence, allows easy creation of REST APIs. The use of Flask, and the REST principle in general, was selected for simplicity of programming. This greatly assisted in developing the program within the limited time available for this project. Additionally, PySerial (Liechti 2019), also licensed under BSD, was used to provide USB serial communication with the MCU. PyATEM (Sxpert 2018), licensed under LGPL, provides tally information by communicating with a network-connected camera switcher.

For the program to maintain the state of the system and communicate with the MCU, multi-threading was used. A total of five background threads plus the main Flask application were created. One thread handles incoming serial messages from the MCU, updating states as required. Another thread handles outgoing serial messages to the MCU using a FIFO queue. Other processes place serial messages onto the queue for this thread to send. This avoids the complication of preventing multiple threads from attempting to send a message

at the same time. A third thread handles use of the PyATEM library, processing new tally states as they are available. The final two threads are used to poll the MCU and CCUs for updated status information. They make use of delays between each poll to prevent saturation of the serial port.

The program as a whole is multi-paradigm, using object class models to represent the MCU and the CCUs. This allows for simple scaling of the number of CCUs connected without any changes to the way they are represented as computational data. A class variable is used to hold a list of all CCU instances, allowing easy access by other functions and threads. Whilst in some applications this could risk a memory leak, this system has a naturally bound upper limit to the number of CCUs that can be attached. This is due to the maximum number of CAN nodes supported on a bus and should be around 100 devices.

The API paths exposed are as follows:

**GET** / Displays a title page identifying the API.

**GET** /**ccu** Takes a CCU device address as an argument and returns the matching CCU object as JSON. If there is no matching object, a HTTP 404 status is returned.

**GET** /**ccu/addresses** Returns a JSON array of CCU object device addresses.

**PUT** /**ccu/type** Sets the camera type of a CCU, device address and camera type passed as arguments. Returns 204 if successful, 404 if no matching CCU or camera type was found.

**PUT** /**ccu/tally** Sets the tally number of a CCU, device address and tally number passed as arguments. Returns 204 if successful, 404 if no matching CCU was found.

**PUT** /**ccu/command** Sends a command to a CCU, for example 'zoom in'. Device address, command, and (optionally) a number of repeats passed as arguments. Returns 204 if successful, 404 if no matching CCU or command was not found.

**PUT** /**ccu/discover** Instructs the MCU to manually search for CCUs.

**GET** /**cameraTypes** Returns a JSON array of camera type names.

**GET** /**mcu** Returns the MCU object as JSON.

**PUT** /**mcu/power** Accepts buses with boolean states as parameters and instructs the MCU to turn on or off power to the bus as requested.

## Web GUI

The web GUI is implemented as a static website, served from a web server on the SBC. Client-side JavaScript is used to make API calls and update the GUI on the user's web browser. Three open-source libraries are used for GUI styling, all licensed under the MIT licence. They are: jQuery (JS Foundation 2020), Bootstrap 4 (Otto and Thornton 2020), and Popper.js (Zivolo 2020).

On initial page load, the JavaScript program will query the API for a list of CCU addresses. If this list is empty, the program will automatically issue the command to scan for CCUs. This makes using the system simpler since CCUs can be found completely automatically if they are connected prior to system start. In case CCUs are added to the system later, the MCU can be instructed to scan manually using the GUI button 'Find CCUs'. The main GUI view is shown in figure 6.11.

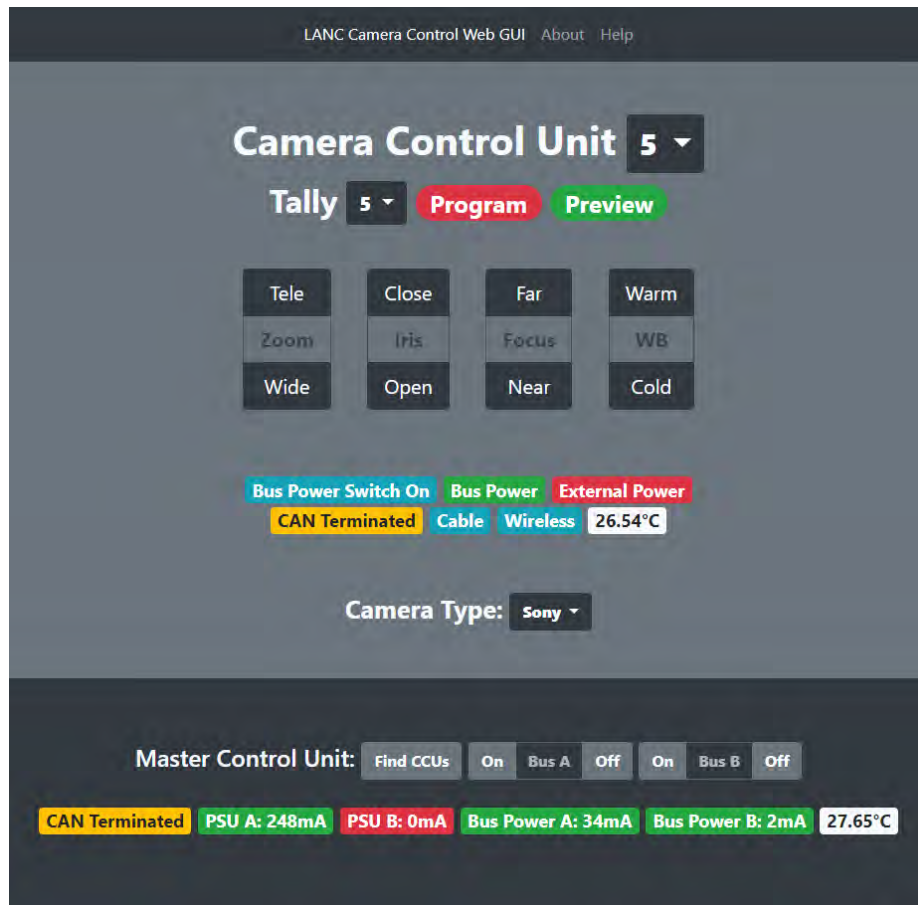


Figure 6.11: Web GUI Main View

To prevent a cluttered interface, only one CCU can be selected at a time. This should not be an issue since camera adjustments are not usually made in a live environment. Tally information is displayed near the top to warn the user if the camera is being used. All available status information for both the currently selected CCU and the MCU are displayed to the user. Whilst not intended for use on mobile devices, the design is responsive so could be used on phones and tablets if situations demanded it. However, it is expected that the system will usually run on a wired connection due to technical users' concerns for reliability of wireless connections.

Figure 6.12 shows the 'About' pop-up box which displays basic information about the web GUI and its licensing to the user. This functionality was included to allow attribution of open-source components and provide additional information to the user.

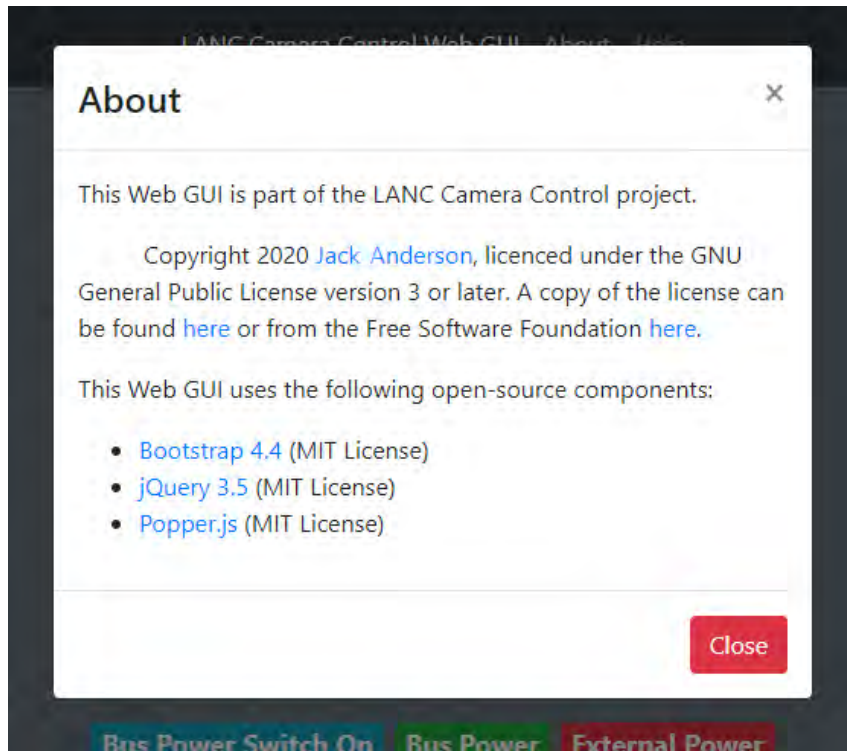


Figure 6.12: Web GUI About View

Figure 6.13 shows the 'Help' pop-up box which is intended to provide the user with all the basic information required to use the web GUI. This guidance should assist the user and hopefully ensure the system is found to be useful. It is also a requirement of the system to have good user documentation, which this forms a part of.

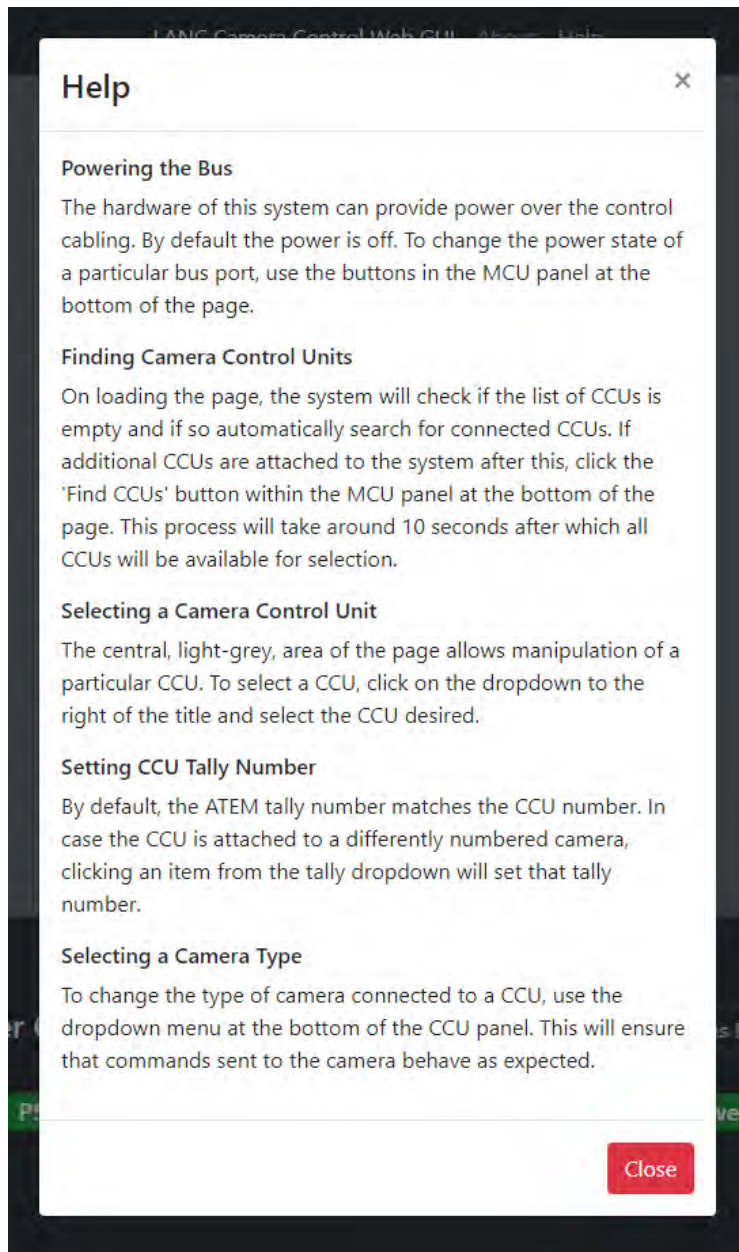


Figure 6.13: Web GUI Help View

### 6.2.4 Open-Source Summary

Table 6.1 summarises all the open-source components used in the software elements of this project. The only other components used were those distributed with the programming languages. In the case of the microcontroller programs, this table does not include components distributed with the Arduino IDE or Atmel Studio 7.

Table 6.1: Open-Source Components

Name	Author	Licence	Used In
PJON	Mitolo (2020a)	Apache 2.0	AVR
ATTinyCore	Konde (2020)	GPLv3	AVR
MCP23008	Adafruit Industries (2019)	BSD	ARM (both)
Neopixel	Adafruit Industries (2020)	LGPLv3+	ARM (CCU)
MCP2515	Pereslegin (2020)	MIT	ARM (both)
Serial to LANC	Rosén (2017)	GPLv3+	ARM (CCU)
Flask	The Pallets Projects (2020)	BSD-3	Web API
PySerial	Liechti (2019)	BSD-3	Web API
PyATEM	Sxpert (2018)	LGPLv2.1+	Web API
jQuery	JS Foundation (2020)	MIT	Web GUI
Bootstrap 4	Otto and Thornton (2020)	MIT	Web GUI
Popper.js	Zivolo (2020)	MIT	Web GUI



## 7 Testing

This chapter explores the testing that was carried out to ensure that the system developed behaved as was intended. First the hardware testing is detailed, followed by the software and system testing. Unfortunately, testing was severely impacted by the coronavirus pandemic, as is discussed in more detail where relevant.

### 7.1 Hardware Testing

Aside from the visual inspection and rework performed during assembly of the PCBs, several other hardware tests were performed. Firstly, power was applied to the circuit boards and voltages checked to be within expected ranges. All circuit boards passed this test initially, as only external power was tested in the case of the CCU PCBs. When bus power was later applied to the CCUs, two did not power on. This was found to be caused by poor solder joints to the bus power eFuse, something that was missed during visual inspection. Both units had their eFuse reseated after which they worked as expected.

Further testing was only possible once basic prototypes of microcontroller code could be compiled. In the case of the AVR microcontrollers, one would not program and had to be soldered again. Further basic testing found all other ICs to work. Once the AVR program had been completed, it was programmed to all PCBs. This revealed a fault with the MCU CAN termination relay, which had to be replaced with a new part. It is likely that the original relay was delivered faulty from the supplier as there was no sign of external damage. Once the new part was fitted, the CAN termination behaved as expected which removes the possibility of a design flaw causing the original fault. Initial hardware testing is shown in figure 7.1.



Figure 7.1: Hardware Testing of a CCU

It would have been preferable to assess the completed electronics for EMC compliance. However, due to the coronavirus lock-down, no suitable test site was reachable. Despite the author possessing some of the equipment required, the electromagnetic radiation noise floor is far too high in an ordinary home to capture meaningful results. For this reason, no EMC testing was carried out. It is likely that a Faraday cage, in the form of a metal enclosure or copper tape, would be required for the CCUs. This cannot be determined without experimental testing.

No destructive load testing of the CCU enclosures was carried out. However, they are not expected to carry any meaningful mechanical loads so such testing would likely be redundant.

## 7.2 Software Testing

Due to this project's interaction with hardware and physical components such as video cameras, it was not possible to achieve automated unit testing. Any automated unit testing that could have been carried out would have been minimal, adding unnecessary development overhead. However, manual testing was performed as each unit was created. This allowed mistakes to be spotted quicker than if testing had only occurred once a program

was completed.

Integration testing happened naturally in this project as each new software component relied upon the last to achieve the desired functionality. However, it was cumbersome to continually test microcontroller programs due to the author only owning one hardware device programmer for each microcontroller type. This resulted in any functionality requiring two devices to be programmed to cause a complete rearrangement of connections.

Across all software developed for this project, 87% of tests were passed first time. The most significant test that failed initially was configuration of the CAN IC. After debugging the SPI bus with an oscilloscope, the cause was found to be the LoRa radio module. Both the CAN IC and the LoRa radio interface to the microcontroller on the same SPI bus. Since the LoRa functionality was not implemented in software, the LoRa chip select pin had been left in a default state. This had caused both devices to attempt to respond to the microcontroller at the same time. Luckily this was a simple fix, requiring only a software pull-up resistor to be enabled on the LoRa chip select pin.

### 7.3 System Testing

Full system testing was the most impacted by the coronavirus pandemic. Fortunately, LSU Media loaned the author two video cameras and a camera switcher just prior to the start of the lock-down. This allowed for basic system testing to be carried out, including confirmation of the ability to actually control the cameras. Prior to the coronavirus outbreak, it had been planned to perform full user testing followed by acceptance testing in a broadcast environment. In the end, this was not possible.

Informal user testing occurred through internet-based communication. Unfortunately, since this project requires hands-on interaction with hardware, little useful information was gained. However, some improvements to the web GUI were made based on recommendations from potential users. It is hoped that once the coronavirus restrictions are relaxed, complete testing will be possible.

An area that needs particular scrutiny during future testing is the web API. Without feedback from potential users of this functionality, it is hard to determine if all useful features have been implemented correctly. Additionally, the total length of the CAN bus tested during complete system testing has been restricted by the length of network cabling owned by the author. Testing throughout an entire building, such as LSU, would be highly desirable.

## 8 Evaluation

This chapter evaluates the outcome of the project against the technical requirements. It then briefly explores the commercial viability of this project.

### 8.1 Technical Requirements

This section compares the project outcome against the requirements defined in chapter 4. It continues by discussing potential improvements that could be made to the system on a technical level. Table 8.1 shows that most requirements intended to be implemented were satisfied. Entries marked with an asterisk (\*) have been assessed to a limited extent due to coronavirus restrictions. Further interaction with stakeholders in a broadcast environment would allow more detailed outcomes to be determined.

Table 8.1: Requirements Evaluation Summary

<b>Req.</b>	<b>Type</b>	<b>Outcome</b>
1	Must	Satisfied
2	Must	Satisfied
3	Must	Satisfied
4	Must	Satisfied
5	Must	Satisfied
6	Must	Satisfied
7	Must	Satisfied
8	Must	Satisfied
9	Must	Satisfied*
10	Must	Satisfied*
11	Should	Satisfied
12	Should	Satisfied
13	Should	Satisfied

<b>Continuation of Table 8.1</b>		
<b>Req.</b>	<b>Type</b>	<b>Outcome</b>
14	Should	Satisfied
15	Should	Satisfied
16	Could	Satisfied
17	Could	Future Work
18	Won't	Future Work
19	Won't	Future Work
<b>End of Table</b>		

The only requirement not originally specified as work for a future release that was not completed is requirement 17. This was not satisfied since a non-blocking camera call function was difficult to implement in the time available. Additionally, stakeholder interaction had highlighted that a camera call function must be visually distinctive from tally functionality. This is hard to assess without user input under broadcast conditions.

Requirement 18 is based around resistance to liquid damage. It was never intended to be accomplished within this project due to the lack of access to or funding for advanced manufacturing techniques. However, it would be feasible to achieve this with future work by simply redesigning the CCU enclosures. Requirement 19 calls for automatic communication redundancy through wireless connections. This could have consumed an entire project by itself so was intended for future work. However, in order for this to be possible to add later, the hardware developed as part of this project had to support wireless functionality. LoRa radio modules were included on all PCBs specifically to allow for this.

A particular success of the system developed is the number of cameras that can be controlled. The theoretical limit of the system is around 100 CAN nodes, this is plenty of capacity for future expansion. On the other hand, a slight oversight is the implementation of requirement 16. This calls for the ability to add pan and tilt control at a later date. This was achieved by the addition of an RS-232 port to the CCUs. Despite fully meeting the requirement, the port presented provides no electrical power to connected devices. This was chosen since a pan and tilt motor would require significantly more power than the CCU could deliver. However, if this port was used for other accessories, a small power provision would be extremely useful.

Finally, the implementation chosen for the web API results in a polling approach to retrieving data. This approach was chosen to speed up development. Whilst being fully functional, it needs more strenuous testing in a larger implementation of the system to determine its suitability. Had project resources not been a factor, it is likely that a web socket based system would have been preferred. This would allow automatic pushing of

relevant data to API users without polling requests. Future versions of the system could easily change the way the API is implemented. Such changes would be best made early in the life of a product to reduce the disruption caused to third-party integrated systems.

## 8.2 Commercial Viability

In comparison to the existing products identified in 3.1, this project offers several distinct advantages. Firstly, providing power over the control cables significantly reduces the overhead of using the system. Additionally, the maximum number of cameras that can be connected outperforms all competitors aimed at the broadcast market. Provision of an API that allows for external systems to integrate with the system provides greater possibilities for automation. If future work was carried out to implement the redundant wireless functionality possible through the hardware implementation, a unique product would result.

The costs of producing hardware components for this system, whilst not insignificant, are well below the sales prices of competitor products. Once economies of scale are considered, there is no doubt that a healthy margin would exist. The biggest obstacle to developing this project into a commercial offering is the need for EMC testing. However, with the right investment this could easily be overcome.

## 9 Conclusion

In conclusion, this project set out to develop a LANC camera control system and determine its commercial viability. The project aim of creating a system with minimal hardware burden that allows remote control of camera settings was achieved. Despite only aspiring to build a Minimum Viable Product, a fully-featured hardware platform was created. The overall system shows real promise as a commercial product, neatly filling a perceived gap in the current market.

The project objectives (see 1.3) were all met with the exception of objective 5. This objective was not met due to the coronavirus pandemic preventing full testing to occur. Overall, this was a highly satisfactory outcome and complete testing will hopefully happen once restrictions are lifted.

On a personal level, the author has completed a large project requiring integration of numerous components. In particular, the combination of hardware and software development was particularly enjoyable. A great deal of personal development has occurred as a result of having to manage a project with such different aspects. The ability to move between multiple programming paradigms freely is a great skill that has been acquired from this project. Lastly, the experimental testing of a communications protocol at the physical layer was especially rewarding (see appendix B).

## References

- Adafruit Industries (2019). *Adafruit MCP23008 Library*. URL: <https://github.com/adafruit/Adafruit-MCP23008-library>.
- (2020). *Adafruit Neopixel Library*. URL: [https://github.com/adafruit/Adafruit%7B%5C\\_%7DNeoPixel](https://github.com/adafruit/Adafruit%7B%5C_%7DNeoPixel).
- Aono, Kenji (2011). *Application Note: PCB Design with EAGLE*. Tech. rep. Michigan State University. URL: [https://www.egr.msu.edu/classes/ece480/capstone/spring11/group05/documents/app%7B%5C\\_%7Dnote%7B%5C\\_%7Ds11%7B%5C\\_%7Ddt5%7B%5C\\_%7Dkenji.pdf](https://www.egr.msu.edu/classes/ece480/capstone/spring11/group05/documents/app%7B%5C_%7Dnote%7B%5C_%7Ds11%7B%5C_%7Ddt5%7B%5C_%7Dkenji.pdf).
- Avesta Group (2019). *Desktop Applications vs Web Apps*. URL: <https://www.avestagroup.net/DetailsEN.aspx?PostID=1006%7B%5C%7DCataType=5%7B%5C%7DCataID=1006> (visited on 24/11/2019).
- BARB (2019). *Quarterly reach of BBC News in the United Kingdom (UK) from 1st quarter 2012 to 2nd quarter 2019*. Tech. rep. URL: <https://www.statista.com/statistics/290866/bbc-news-viewers-reached-quarterly-uk/>.
- Birmingham, Alan et al. (1994). *The Video Studio*. 3rd ed. ISBN: 0240513924.
- Blackmagicdesign (2019). *ATEM Camera Control Panel - Tech Specs*. URL: <https://www.blackmagicdesign.com/uk/products/atemcameracontrolpanel/techspecs/W-ABP-04> (visited on 16/10/2019).
- Boehmel, Manfred (2018). *How SONY's LANC(tm) protocol works*. URL: <http://www.boehmel.de/lanc.htm> (visited on 28/10/2019).
- British Computer Society Working Group on Testing (1986). *Testing in Software Development*. Ed. by M. Ould and C. Unwin. Cambridge University Press. ISBN: 0521337860.
- BroadcastRF (2019). *CAMERA CONTROL – WIRELESS PAINT, RACKING, TELEMETRY*. URL: <http://broadcastrf.com/camera-control/> (visited on 17/11/2019).
- Byrnes, David (2010). *AutoCAD 2011 For Dummies*. 1st ed. ISBN: 9780470649688.
- Chang, Kuang-Hua (2013). *Product Design Modeling Using CAD/CAE*. 1st ed. ISBN: 9780123985170.
- Chapman, Chris and Stephen Ward (2003). *Project Risk Management*. 2nd ed. ISBN: 0470853557.



- Chatterton, Paul A. and Michael A. Houlden (1991). *EMC: Electromagnetic Theory to Practical Design*. 1st ed. ISBN: 047192878X.
- Cheshire, David (1982). *The Video Manual*. 1st ed. ISBN: 0861340302.
- Comer, Douglas E. (2015). *Computer Networks and Internets*. 6th ed. ISBN: 9781292061825.
- Coombs, Clyde F., ed. (2008). *Printed Circuits Handbook*. 6th ed. ISBN: 9780071467346.
- Dallas Semiconductor (1998). *Application Note 83: Fundamentals of RS-232 Serial Communications*. Tech. rep. URL: <http://ecee.colorado.edu/%7B~%7Dmccclure1/dan83.pdf>.
- Datavideo Technologies Co. (2019). *Camera Control*. URL: <https://www.datavideo.com/eu/category/Camera%20Control> (visited on 17/11/2019).
- Dawson, Christian W. (2009). *Projects in Computing and Information Systems*. 2nd ed. ISBN: 9780273721314.
- Dawson, Michael (2010). *Python Programming for the Absolute Beginner*. 3rd ed. ISBN: 9781435455009.
- Doran, George T. (1981). “There’s a S.M.A.R.T. way to write management’s goals and objectives”. In: *Management Review* 70.11.
- Feldmann, K. and J. Sturm (June 1994). “Closed loop quality control in printed circuit assembly”. In: *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part A* 17.2, pp. 270–276. ISSN: 10709886. DOI: 10.1109/95.296409. URL: <http://ieeexplore.ieee.org/document/296409/>.
- Fette, I. and A. Melnikov (Dec. 2011). *The WebSocket Protocol*. Tech. rep. DOI: 10.17487/rfc6455. URL: <https://www.rfc-editor.org/info/rfc6455>.
- Futrell, Robert T., Donald F. Shafer and Linda I. Shafer (2002). *Quality Software Project Management*. 1st ed. ISBN: 0130912972.
- Heath, Steve (2002). *Embedded Systems Design*. 2nd ed. ISBN: 9780080477565.
- Hughes, Bob and Mike Cotterell (2009). *Software Project Management*. 5th ed. ISBN: 0077122798.
- Hughes, J.M. (2016). *Arduino: A Technical Reference: A Handbook for Technicians, Engineers, and Makers*. 1st ed. ISBN: 9781491934500.
- IIBA (2009). *A Guide to the Business Analysis Body of Knowledge*. Ed. by Kevin Brennan. 2.0. ISBN: 0981129218.
- International Telecommunication Union (2012). *ITU Radio Regulations: Article 1*. URL: <http://life.itu.int/radioclub/rr/art1.pdf>.
- ISO/IEC (2010). *ISO/IEC TR 24774 Systems and Software Engineering — Life Cycle Management — Guidelines for Process Description*. Tech. rep. URL: <https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>.
- ISO/IEC/IEEE (2017). *ISO/IEC/IEEE 24765: Systems and software engineering — Vocabulary*. Tech. rep. IEEE. DOI: 10.1109/IEEESTD.2017.8016712. URL: <https://ieeexplore.ieee.org/servlet/opac?punumber=8016710>.

- JS Foundation (2020). *jQuery*. URL: <https://github.com/jquery/jquery>.
- Konde, Spence (2020). *ATTinyCore*. URL: <https://github.com/SpenceKonde/ATTinyCore>.
- Lewis, Roland (1991). *The Video Maker's Handbook*. 2nd ed. ISBN: 0330323555.
- Lewis, William E. (2000). *Software Testing and Continuous Quality Improvement*. 1st ed. ISBN: 0849398339.
- Liechti, Chris (2019). *PySerial*. URL: <https://github.com/pyserial/pyserial>.
- Maguire, Moira and Brid Delahunt (2017). "Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars". In: *All Ireland Journal of Higher Education* 9.3. URL: <https://ojs.aishe.org/index.php/aishe-j/article/view/335>.
- Martin, Trevor (2013). *The Designer's Guide to the Cortex-M Processor Family*. 1st ed. ISBN: 9780080982960.
- Maxim Integrated Products Inc. (2008). *Guide to 1-Wire Communication*. URL: <https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1796.html> (visited on 08/04/2020).
- McDonald, John H. (2015). *Fisher's exact test of independence*. URL: <http://www.biostathandbook.com/fishers.html> (visited on 04/01/2020).
- Mitolo, Giovanni Blu (2020a). *Padded Jittering Operative Network*. URL: <https://github.com/gioblu/PJON>.
- (2020b). *PJDL v4.1*. Tech. rep. URL: <https://github.com/gioblu/PJON/blob/master/src/strategies/SoftwareBitBang/specification/PJDL-specification-v4.1.md>.
- Montrose, Mark I. and Edward M. Nakauchi (2004). *Testing for EMC Compliance*. 1st ed. IEEE Press. ISBN: 047143308X.
- Morris, D.J. (1983). *Communication for Command and Control Systems*. 1st ed. ISBN: 008027597.
- Murata Manufacturing Co. Ltd. (2020). *LoRa Module FAQ*. URL: <https://www.murata.com/en-eu/support/faqs/products/lpwa/lorahardware/0008> (visited on 07/04/2020).
- Ofcom (2019). *Television industry revenue in the United Kingdom (UK) from 2004 to 2018, by source (in billion GBP)*. Tech. rep. URL: <https://www.statista.com/statistics/236899/tv-industry-revenue-in-the-uk-by-source/>.
- Otto, Mark and Jacob Thornton (2020). *Bootstrap*. URL: <https://github.com/twbs/bootstrap>.
- Pascoa, Jose A. (2017). *Need for Speed - C++ versus Assembly Language*. URL: <https://www.codeproject.com/Articles/1182676/Need-for-Speed-Cplusplus-versus-Assembly-Language> (visited on 23/11/2019).
- Pereslegin, Dmitry (2020). *Arduino MCP2515 CAN interface library*. URL: <https://github.com/autowp/arduino-mcp2515>.
- Predko, Myke (1999). *Handbook of Microcontrollers*. 1st ed. ISBN: 0079137172.

- Rosén, L (2017). *Serial to LANC (Control-L)*. URL: <https://create.arduino.cc/projecthub/L-Rosen/serial-to-lanc-control-l-70f735>.
- Royer, Thomas C. (1993). *Software Testing Management: Life on the Critical Path*. 1st ed. ISBN: 0135329876.
- Ruiz, Alex and Gabi Jack (2010). *SolidWorks 2010: No Experience Required*. 1st ed. ISBN: 9780470638170.
- Skaarhoj (2018). *BlackMagic ATEM Switcher Protocol*. URL: <https://www.skaarhoj.com/fileadmin/BMDPROTOCOL.html> (visited on 24/11/2019).
- Software Testing Fundamentals (2019). *System Testing*. URL: <http://softwaretestingfundamentals.com/system-testing/> (visited on 22/11/2019).
- Sommerville, Ian (2016). *Software Engineering*. 10th ed. ISBN: 1292096136.
- Sony (2019). *Camera Control Panels*. URL: [https://pro.sony/en%7B%5C\\_%7DGB/products/camera-control-panels](https://pro.sony/en%7B%5C_%7DGB/products/camera-control-panels) (visited on 17/11/2019).
- Sxpert (2018). *PyATEM*. URL: <https://github.com/sxpert/PyATEM>.
- Texas Instruments (2005). *ISM-Band and Short Range Device Regulatory Compliance Overview*. Tech. rep. URL: <https://www.ti.com/lit/an/swra048/swra048.pdf>.
- (2013). *AN-1031: TIA/EIA-422-B Overview*. Tech. rep. URL: <https://www.ti.com/lit/an/snla044b/snla044b.pdf>.
- (2018). *Basics of eFuses*. Tech. rep. URL: <https://www.ti.com/lit/an/slva862a/slva862a.pdf>.
- The National Institute of Justice (2002). *Guide for the Selection of Communication Equipment for Emergency First Responders: NIJ Guide 104-00: Volume I*. Tech. rep. U.S. Department of Justice. URL: <https://www.ncjrs.gov/pdffiles1/nij/191160.pdf>.
- The Pallets Projects (2020). *Flask*. URL: <https://github.com/pallets/flask/releases>.
- Tooley, Mike (2015). *Electronic Circuits: Fundamentals and Applications*. 4th ed. ISBN: 9781317572336.
- Venkataramanan, Madhumita (Dec. 2013). *Micro Python: more powerful than Arduino, simpler than the Raspberry Pi*. URL: <https://www.wired.co.uk/article/micro-python> (visited on 23/11/2019).
- Vrancic, Damir and Steven L. Smith (Feb. 2006). “Permanent synchronization of camcorders via LANC protocol”. In: *SPIE 6055, Stereoscopic Displays and Virtual Reality Systems XIII*. Ed. by Andrew J. Woods et al., p. 60550I. DOI: 10.1117/12.677721. URL: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.677721>.
- Waldo, Jim (2010). *Java: The Good Parts*. 1st ed. ISBN: 9780596803735.
- White, Elecia (2012). *Making Embedded Systems*. 1st ed. ISBN: 9781449302146.
- Williams, Tim (2001). *EMC for Product Designers*. 3rd ed. ISBN: 0750649305.

## REFERENCES

Jack D. Anderson

- Wireless Telegraphy Act* (2006). URL: <http://www.legislation.gov.uk/ukpga/2006/36/part/2>.
- Zivolo, Federico (2020). *Popper.js*. URL: <https://github.com/popperjs/popper-core>.

# Appendices

# A Industry Survey

This appendix shows the market research industry survey carried out in 3.2.1 as seen by the 58 participants.

## A.1 Consent Page



**Camera Control System Market Research**

\* Required

**About this study**

Investigator Details:  
Jack Anderson, Epinal Way, LE11 3TU, UK.

We would like to invite you to take part in our study. Before you decide we would like you to understand why the research is being done and what it would involve for you. Talk to others about the study before making a decision if you wish.

What is the purpose of the study?  
This study aims to assess the industry concerns regarding camera control systems and gain insight into desired features and properties of a future camera control system.

Who is doing this research and why?  
This research is being carried out by a final year computer science undergraduate student as part of a final year project. This study is part of a Student research project supported by Loughborough University.

Are there any exclusion criteria?  
Any participant under the age of 18.

What will I be asked to do?  
You will be asked to complete an anonymous questionnaire relating to your position and experience within the industry and your feelings towards camera control systems.

Once I take part, can I change my mind?  
After you have read this information, if you are happy to participate we will ask you to indicate your consent below. However if at any time, before, during or after the survey you wish to withdraw from the study please just contact the main investigator. You can withdraw at any time, for any reason and you will not be asked to explain your reasons for withdrawing. However, once the results of the study have been submitted (expected to be by 01/01/2020), it may not be possible to withdraw your individual data from the research.

How long will it take?  
Around 10 minutes.

Are there any disadvantages or risks in participating?  
There are no risks expected other than those usually present when accessing the internet.

Data Protection Privacy Notice  
Loughborough University will be using information/data from you in order to undertake this study and will act as the data controller for this study. This means that the University is responsible for looking after your information and using it properly.

What personal information will be collected from me and how will it be used?  
No personal information (such as your name and contact details) shall be collected. Data collected during the study will be anonymised so that it is not personally identifying.

What is the legal basis for processing my personal information?  
Personal data will be processed on the public task basis. Individuals' rights to erasure and data portability do not apply if you are processing on the basis of public task. However, individuals do have a right to object.

How will the anonymised data/results collected from me be used?  
The results of the study will be used as part of an individual final year project.

How long will the anonymised data/results be retained?  
Anonymised results shall be retained indefinitely in the form of the final report for the project.

I have some more questions; who should I contact?  
Please contact the investigator listed at the top of the first page of this document.

What if I am not happy with how the research was conducted?  
If you are not happy with how the research was conducted, please contact the Secretary of the Ethics Approvals (Human Participants) Sub-Committee, Research Office, Hazlerigg Building, Loughborough University, Epinal Way, Loughborough, LE11 3TU. Tel: 01509 222423. Email: [researchpolicy@lboro.ac.uk](mailto:researchpolicy@lboro.ac.uk)

The University also has policies relating to Research Misconduct and Whistle Blowing which are available online at <http://www.lboro.ac.uk/committees/ethics-approvals-human-participants/additionalinformation/codesofpractice/>.

If you require any further information regarding the General Data Protection Regulations, please see: <https://www.lboro.ac.uk/privacy/research-privacy/>.

1. Have you read the description above and consent to voluntarily take part in this research? \*

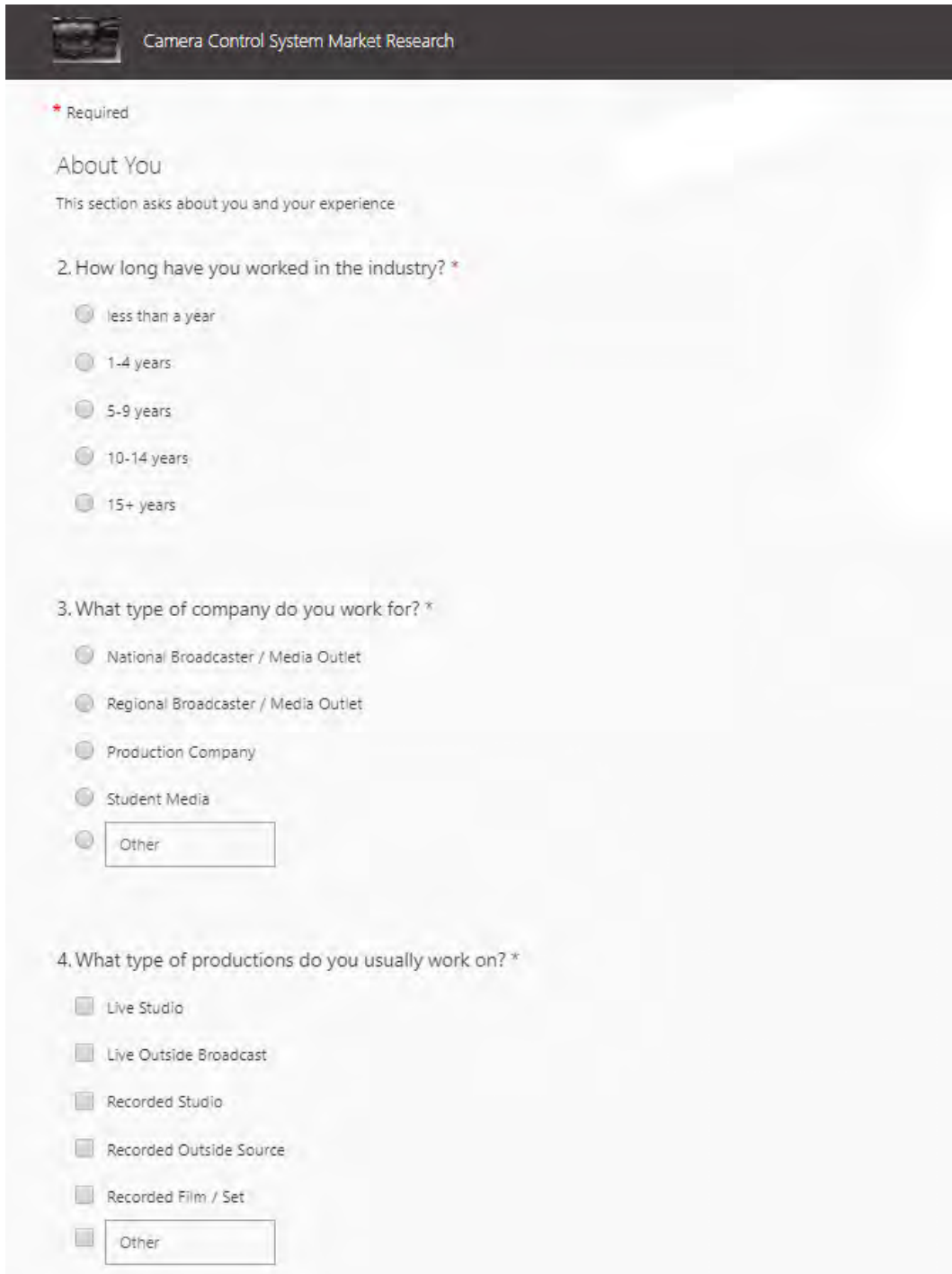
*If no, please close this survey and do not continue to answer any questions*

Yes

No

**Next**

## A.2 Participant Details



**Camera Control System Market Research**

\* Required

### About You

This section asks about you and your experience

2. How long have you worked in the industry? \*

- less than a year
- 1-4 years
- 5-9 years
- 10-14 years
- 15+ years

3. What type of company do you work for? \*

- National Broadcaster / Media Outlet
- Regional Broadcaster / Media Outlet
- Production Company
- Student Media
- Other

4. What type of productions do you usually work on? \*

- Live Studio
- Live Outside Broadcast
- Recorded Studio
- Recorded Outside Source
- Recorded Film / Set
- Other



5. What is the typical viewership size for productions you work on?

- 0-999
- 1,000-9,999
- 10,000-99,999
- 100,000-999,999
- 1,000,000+


6. Which role do you primarily undertake in productions? \*  
*If your role varies, select the one you undertake most frequently*

- Sound Operator
- Technical Support
- Runner
- Camera Operator
- Vision Mixer
- On-screen Talent
- Camera Control
- Graphics
- Director
- Floor Manager
- 

7. Have you ever used a camera control system before? \*

- Yes
- No

### A.3 Camera Control Systems


Camera Control System Market Research

Camera Control Systems

This section focuses on camera control systems

8. Please indicate how you feel towards the following statements

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Having a remote camera control system is important	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Remote camera control system cost is a concern	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Remote camera control systems' cabling and infrastructure are a burden on productions	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Compatibility of the system with a large range of cameras is desirable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A hardware control panel is preferable over a software solution	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Being able to control many cameras at once is important	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Camera control systems should also include tally indication	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Wireless connections between the control station and cameras would be preferred	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. Please rank the following in order of importance when selecting a camera control system  
*most important first (towards the top)*

Max. Number of Controlled Cameras
Reliability
Ease of Use
Wireless Functionality
Camera Compatibility
Equipment Size
Cost

10. Any further comments?

Enter your answer

## B PJON

This appendix details the testing that was carried out to determine the suitability of PJON for automatic CAN bus termination.

### B.1 Overview

Padded Jittering Operative Network (PJON) is an open-source network protocol that can use many different types of physical layer technology. Applied to this project for automatic CAN bus termination, the Padded Jittering Data Link (PJDL) strategy can make use of a single wire to provide low-speed asynchronous serial communications. An important feature of PJDL is support for multi-master communication, a clear advantage when compared to the Dallas 1-Wire protocol which dictates master-slave topology (Mitolo 2020b).

At the time of initial investigation into the use of PJDL for this project, the maximum known working range was just 100m. Since a CAN bus has potential to reach up to 1,000m at the slowest speed, more testing was required to determine if PJDL would be suitable to detect neighbour nodes since it is reasonable to expect links between nodes to exceed 100m in large CAN implementations.

PJON was invented by Giovanni Blu Mitolo and initially released in 2010, since then it has been improved by over fifty contributors. During the author's investigation and later testing of PJDL, Giovanni and Fred Larsen provided useful advice as well as hot-fixes for suspected issues within the underlying software implementation.

### B.2 Testing

In order to verify and further test the maximum range of a PJDL bus, suitable wiring was needed. With the permission of Loughborough Students' Union, spare installed network lines that run around their building were used for the test. A major advantage of using installed wiring is that it will inevitably pass countless amounts of other cabling, giving a realistic electromagnetic environment potentially full of interfering signals.

PJDL defines four modes equating to four bitrates, each of which requires separate testing to determine maximum range. To simplify testing and reduce the amount of time needed with access to the network cable infrastructure, only the fastest and slowest modes were tested extensively as the other modes' maximum range lies between those two.

To conduct the test, two Arduino Nano clones were used which are based around an 8-bit AVR Atmel ATmega328P microcontroller. Each end of the PJDL bus was tied to ground using an  $8.2k\Omega$  resistor with an additional  $60\Omega$  series resistor between the bus and the microcontroller pin, as recommended for signal integrity and safety. The implementation of the PJDL specification for these microcontrollers is the SoftwareBitBang library. Within the library are automated network test examples which were used for the tests to determine success rate of messages.

Mode one, the slowest of the four modes, was the first to be tested. Eventually a maximum range of 2,000m was found although this was not a limitation of PJDL but rather a limitation of the maximum cable length available to test with. Figure B.1 shows that the waveforms observed in this test, whilst distorted from their ideal 'square' shape were still more than adequate to facilitate communication. It was found that messages could be reliably transmitted with no observable failure rate increase when compared to 100m of cabling.

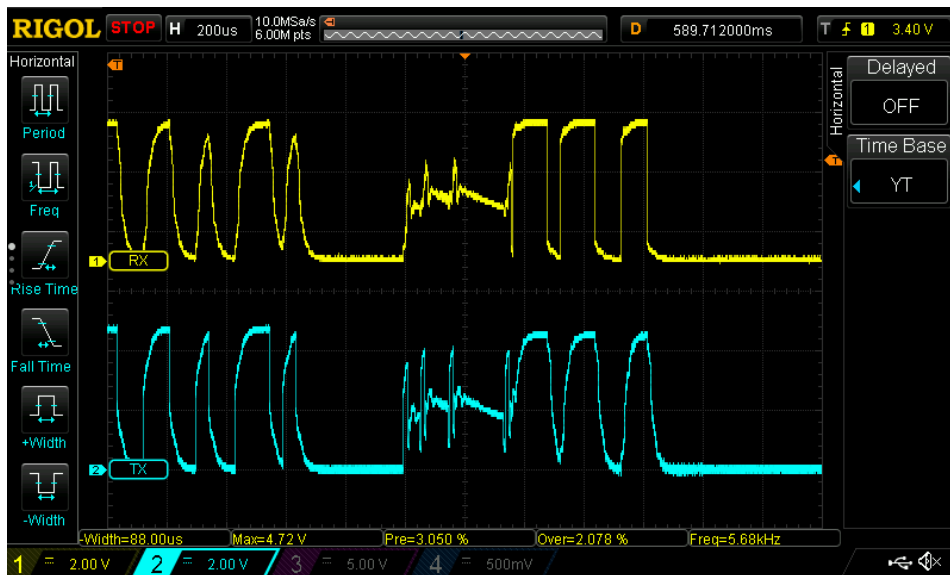


Figure B.1: PJDL Mode One 2,000m Oscilloscope Trace

On the other hand, mode four completely broke down at this range and was unable to pass messages. Figure B.2 shows just how distorted the signals became, the excessive small peaks are the result of failed message acknowledgement. The maximum range that mode four was found to work at reliably was 800m.



Figure B.2: PJDL Mode Four 2,000m Oscilloscope Trace

Table B.1 summarises the maximum range found for each mode of PJDL. Whilst this testing does not guarantee PJDL will always work at these ranges, it is fairly safe to assume that mode one would work reliably at 1,000m range in any reasonably likely conditions.

Table B.1: PJDL Experimental Maximum Range Summary

PJDL Mode	Maximum Range
1	2,000m
2	1,600m
3	1,200m
4	800m

### **B.3 Conclusion**

From the testing carried out by the author, it is reasonably safe to assume that PJDL is capable of carrying out the functionality required for this project. The timely support and encouragement received from the development community further underlines the chance for success in using this protocol. One outcome of this investigation was a new version of the PJDL specification (version 4.1) being released to document the newly found maximum ranges, a result of scientific investigation that the author found highly rewarding.

## C Printed Circuit Boards

This appendix contains the complete details of the custom Printed Circuit Boards for the Master Control Unit and the Camera Control Units. The schematics detail the interconnection of electronic components and circuit designs. Board layouts show the physical arrangement of components for Printed Circuit Board manufacture. Finally, the bill of materials list each component required by the designs including cost.

### C.1 Master Control Unit

#### C.1.1 Schematic

This subsection shows the electronic schematic sheets created by the author in Eagle PCB CAD to achieve the electronic functionality required within the Master Control Unit. Sheet one, figure C.1, details the ATSAM21G18A ARM M0+ microcontroller and supporting circuits. The remaining sheets, figures C.2 to C.5, show the circuits providing the rest of the electronic functionality of the design. It can be seen that a large amount of the circuitry deals with power supply redundancy and protection.



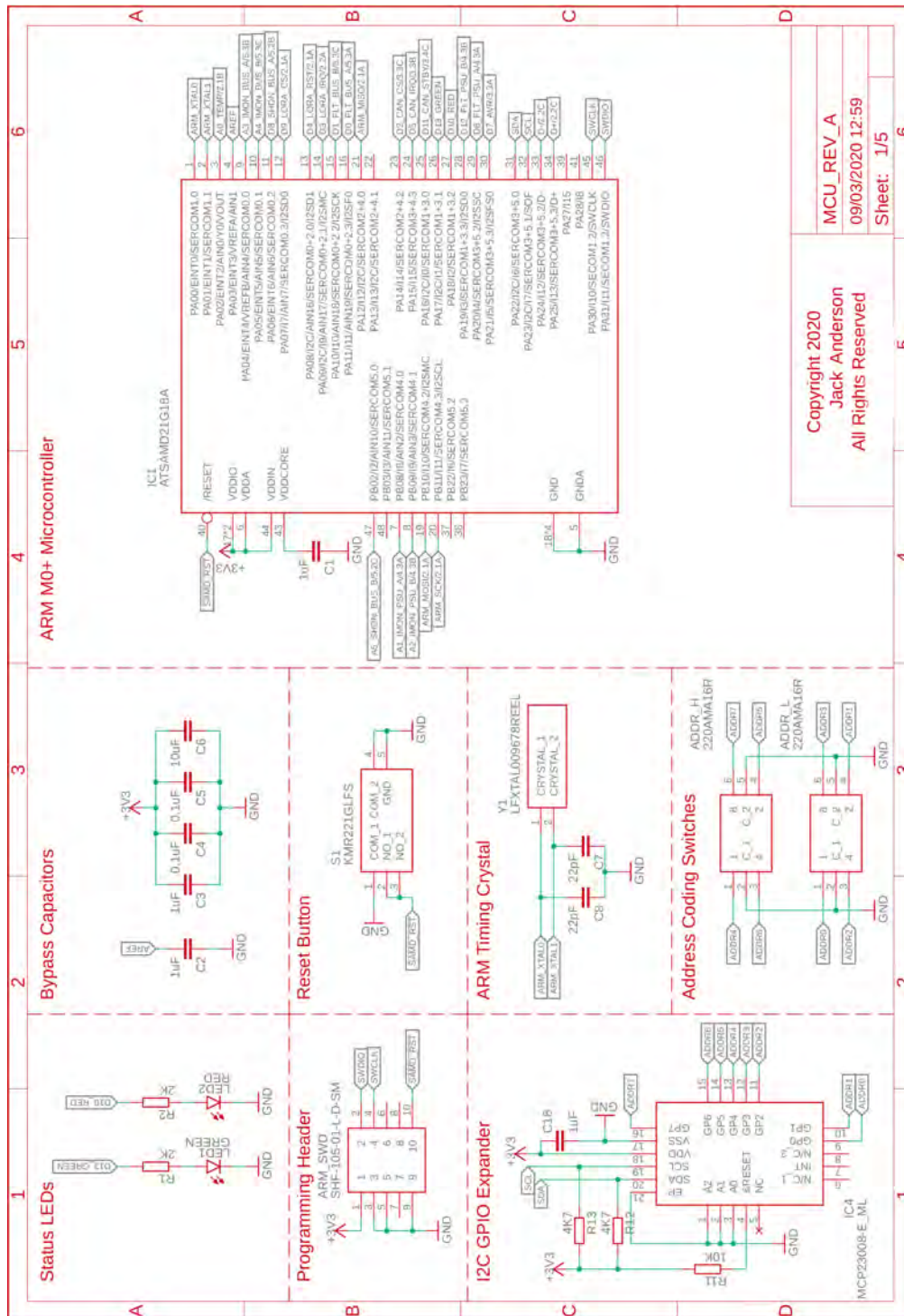


Figure C.1: Master Control Unit Schematic – Sheet One

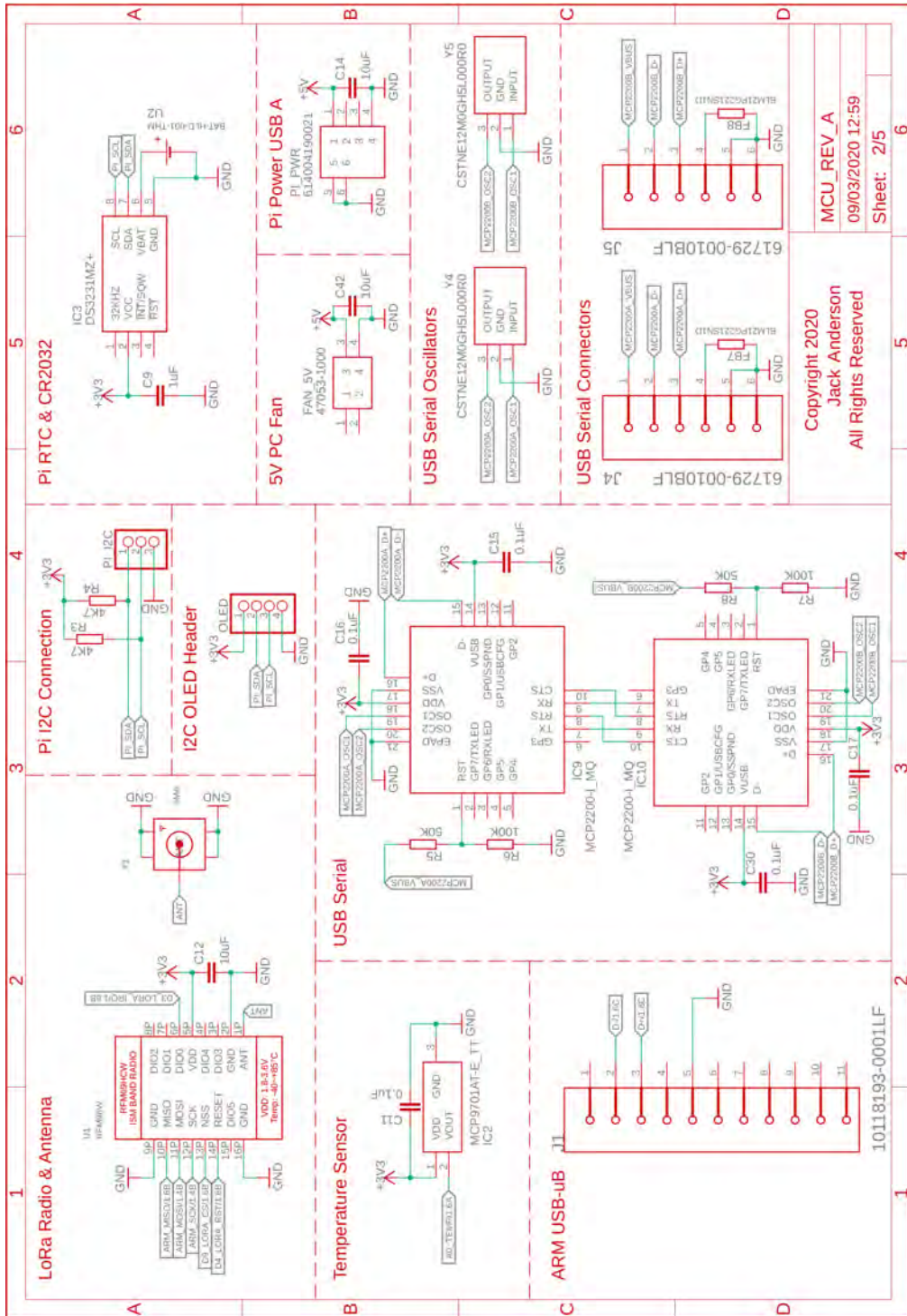
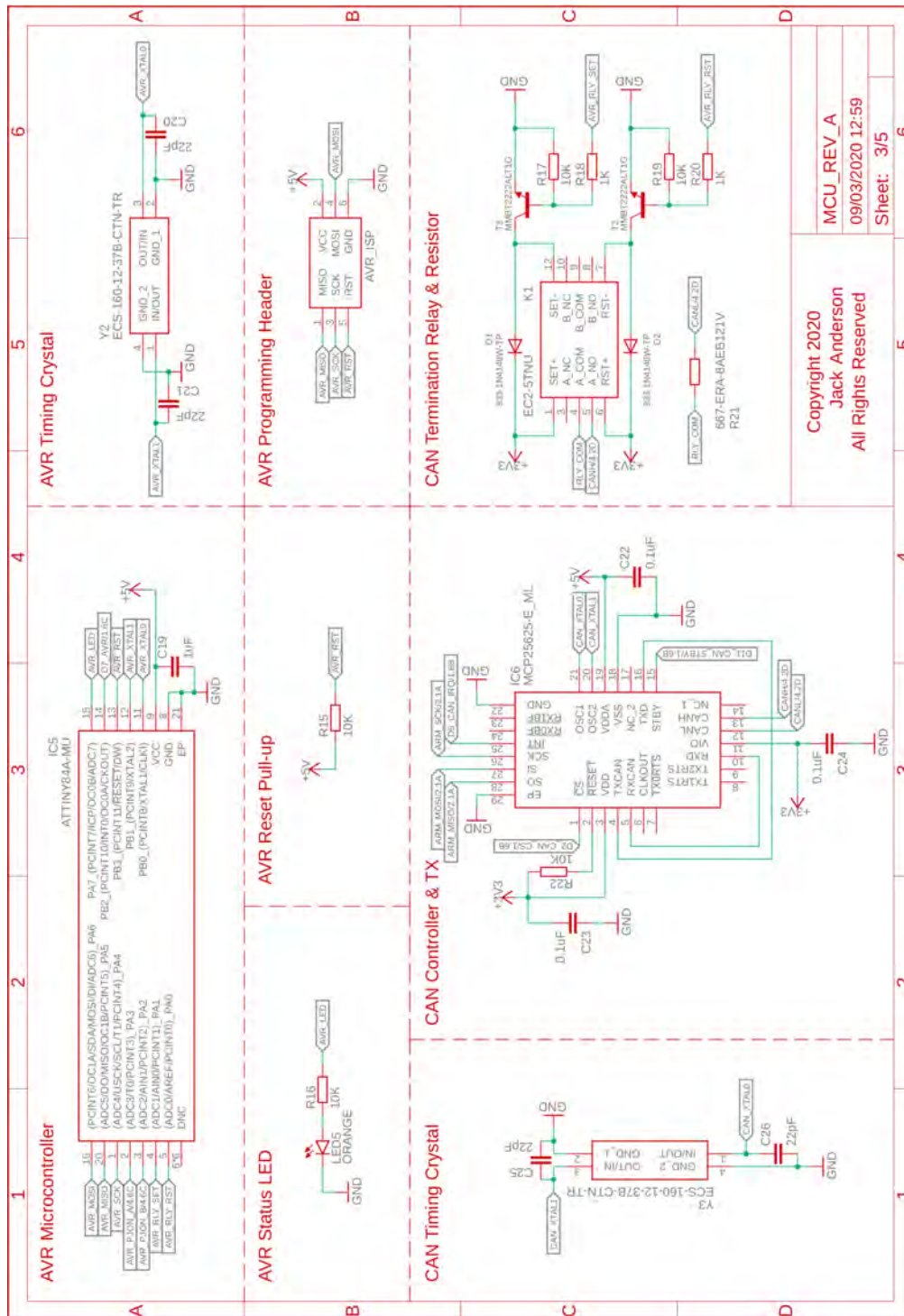


Figure C.2: Master Control Unit Schematic – Sheet Two



Copyright 2020  
Jack Anderson  
All Rights Reserved

MCU\_REV\_A  
09/03/2020 12:59  
Sheet: 3/5

Figure C.3: Master Control Unit Schematic – Sheet Three

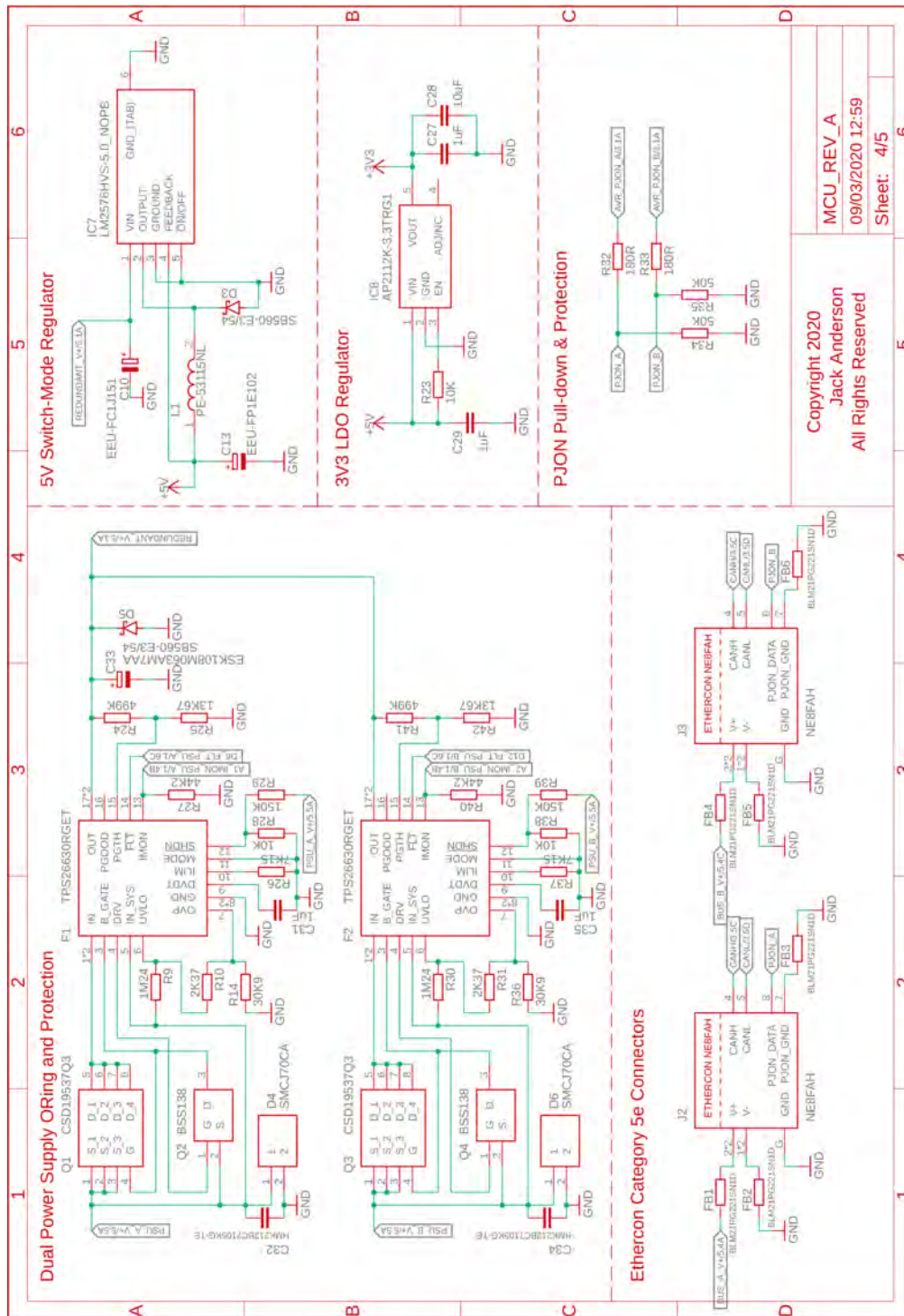


Figure C.4: Master Control Unit Schematic – Sheet Four

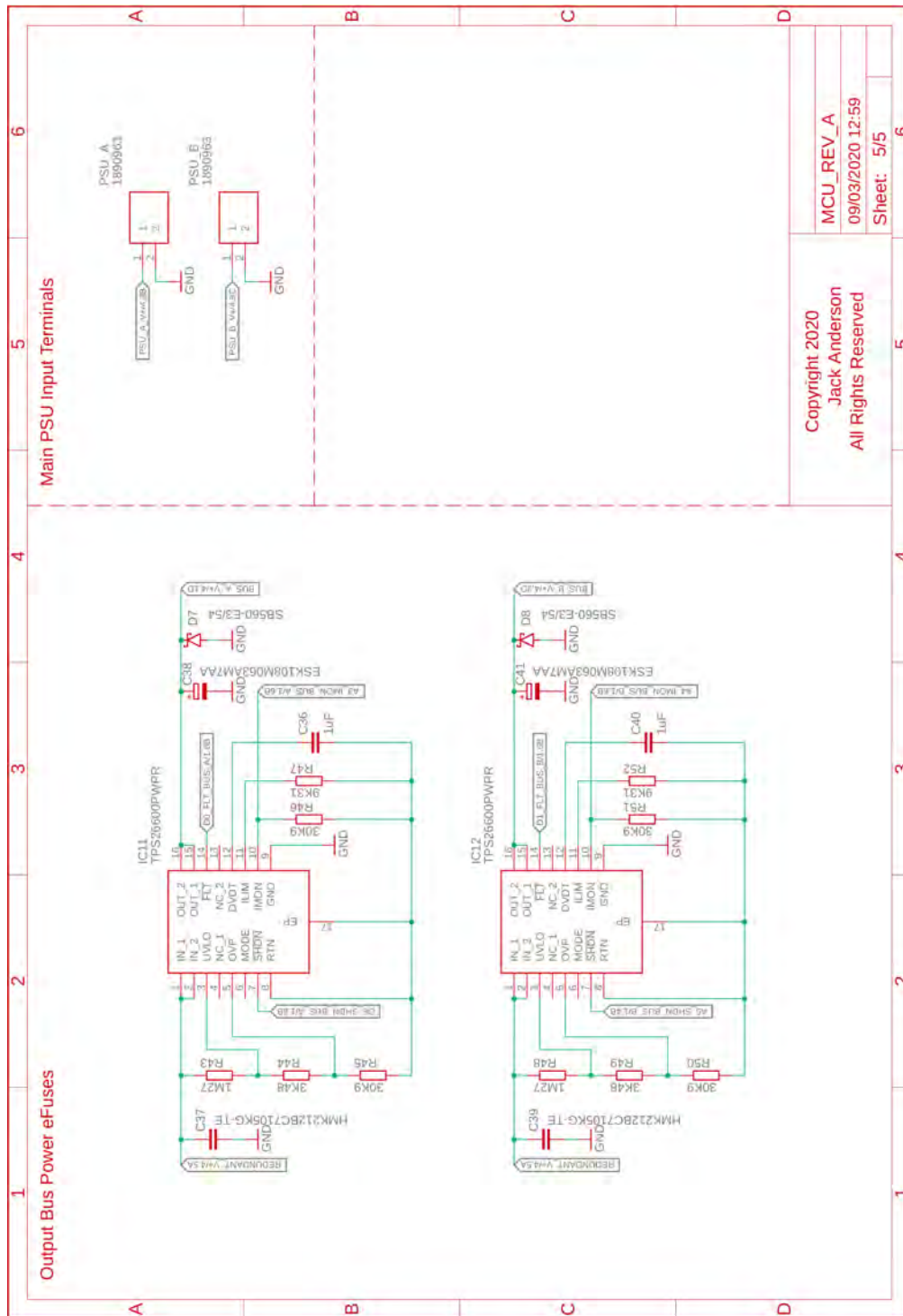


Figure C.5: Master Control Unit Schematic – Sheet Five

### C.1.2 Board Layout

This subsection shows the physical board layout design layers as required for the manufacture of the Master Control Unit circuit board. Figure C.6 shows all the layers, useful to understand how signals are connected across the entire board. Figures C.7 and C.8 show the top and bottom copper layers, with black areas indicating where copper remains after processing by the manufacturer. Figures C.9 and C.10 show the top and bottom solder mask layers, with black areas indicating where the green solder mask is removed leaving exposed copper. Figures C.11 and C.12 show the silkscreen layers, with black areas indicating text or other information printed onto the circuit board. Finally, figure C.13 shows the solder paste layer, with black areas indicating holes in the stencil used for placing solder onto the board during the assembly process carried out by the author.

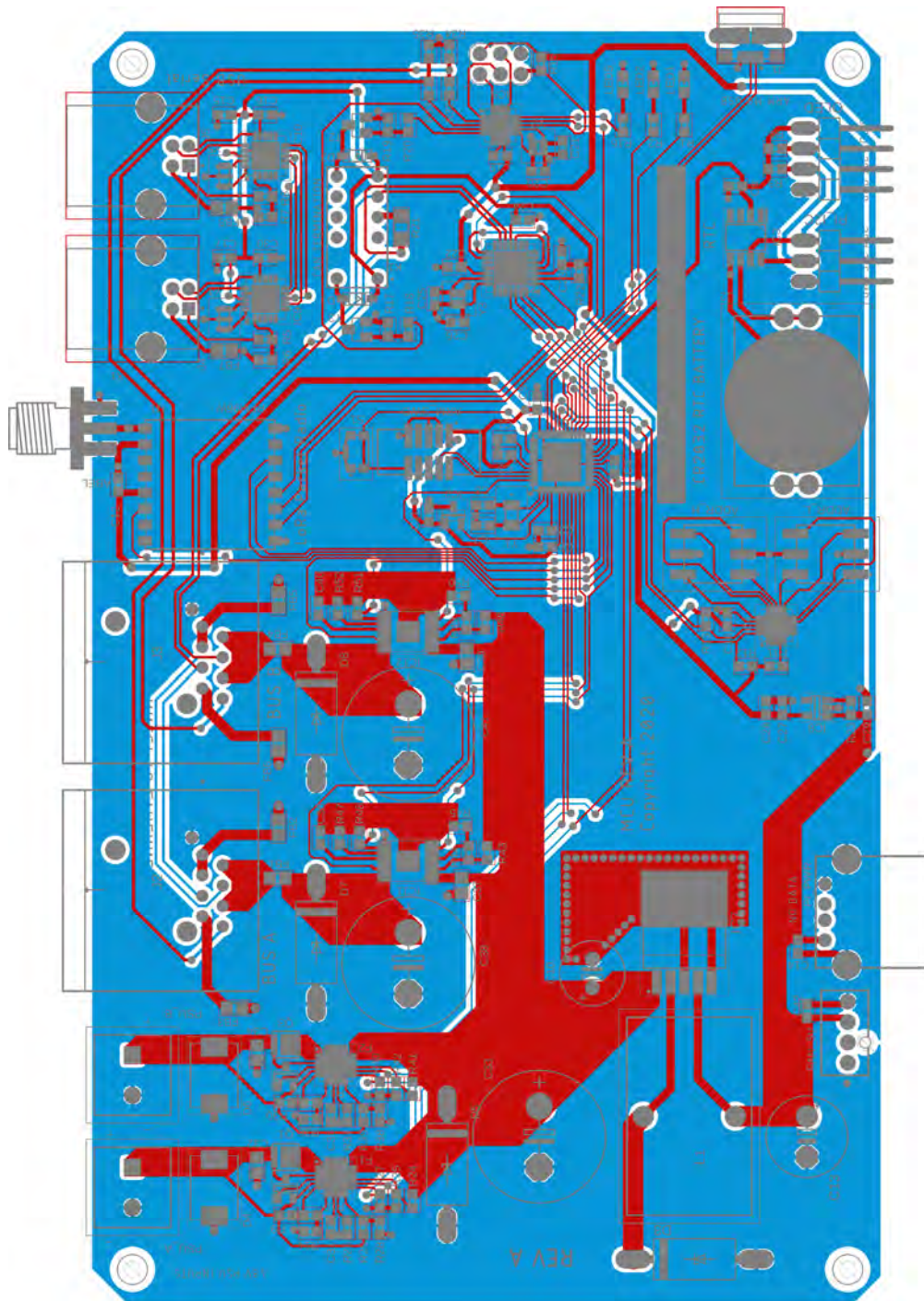


Figure C.6: Master Control Unit All Layers

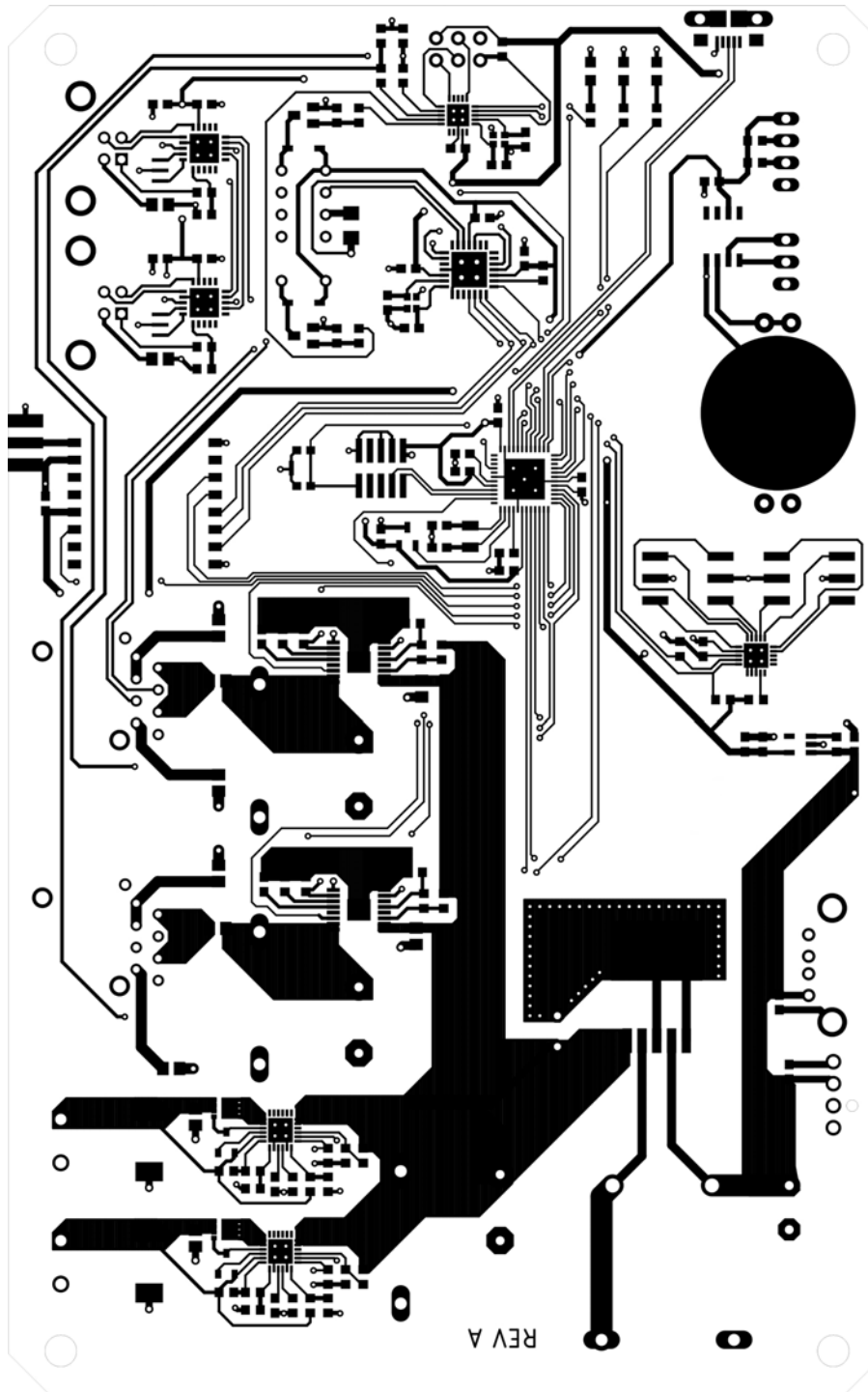


Figure C.7: Master Control Unit Top Copper



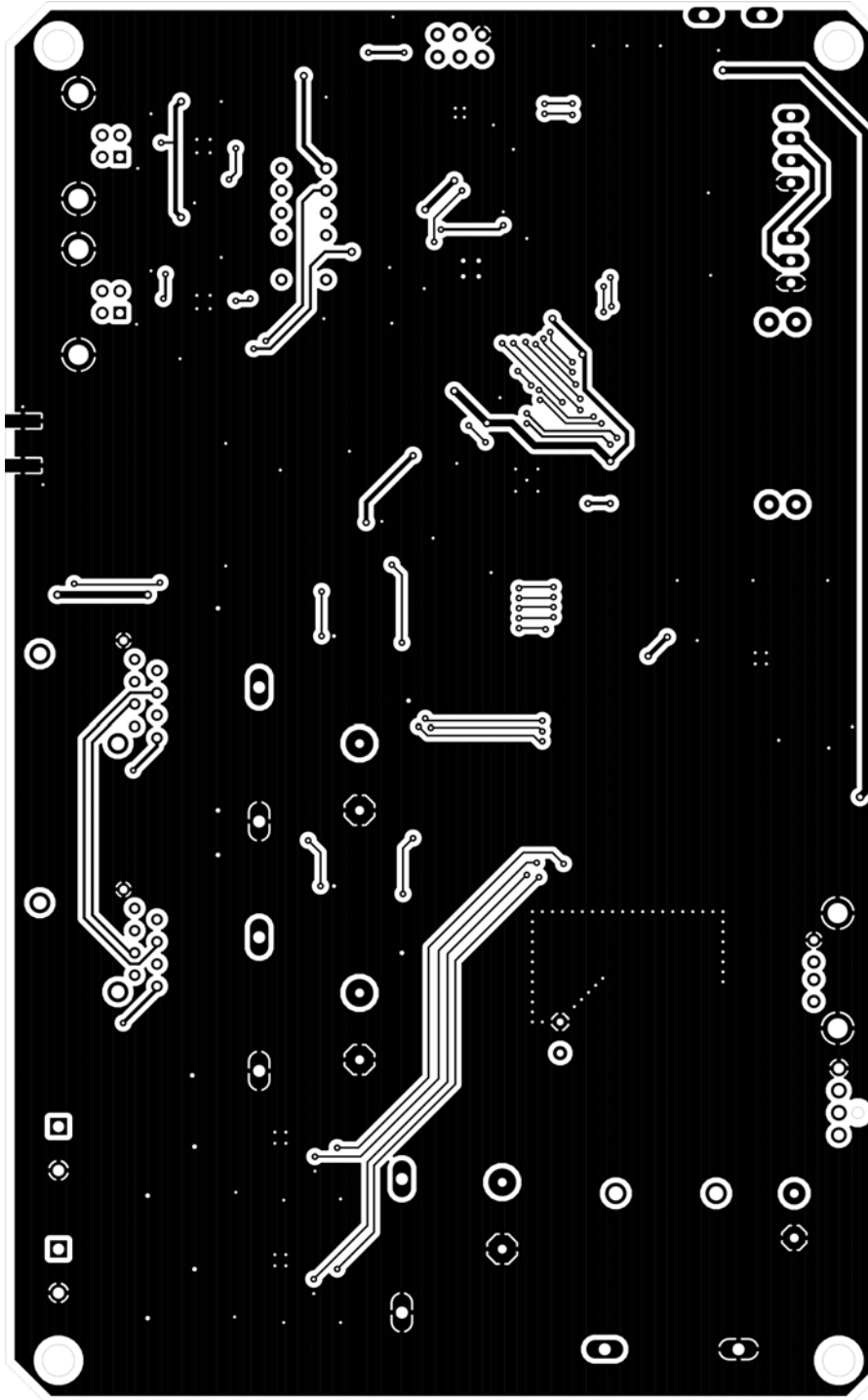


Figure C.8: Master Control Unit Bottom Copper

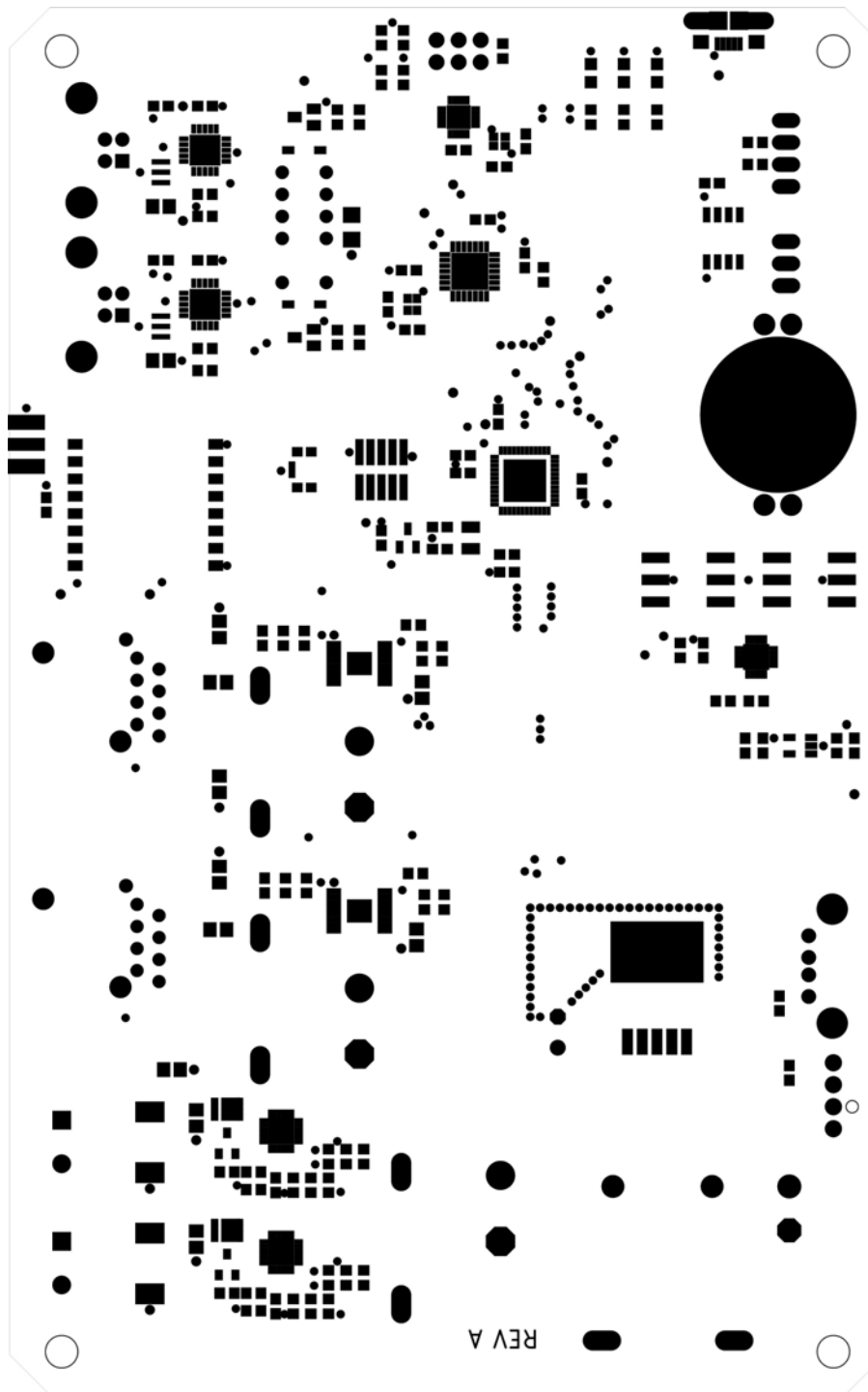


Figure C.9: Master Control Unit Top Solder Mask

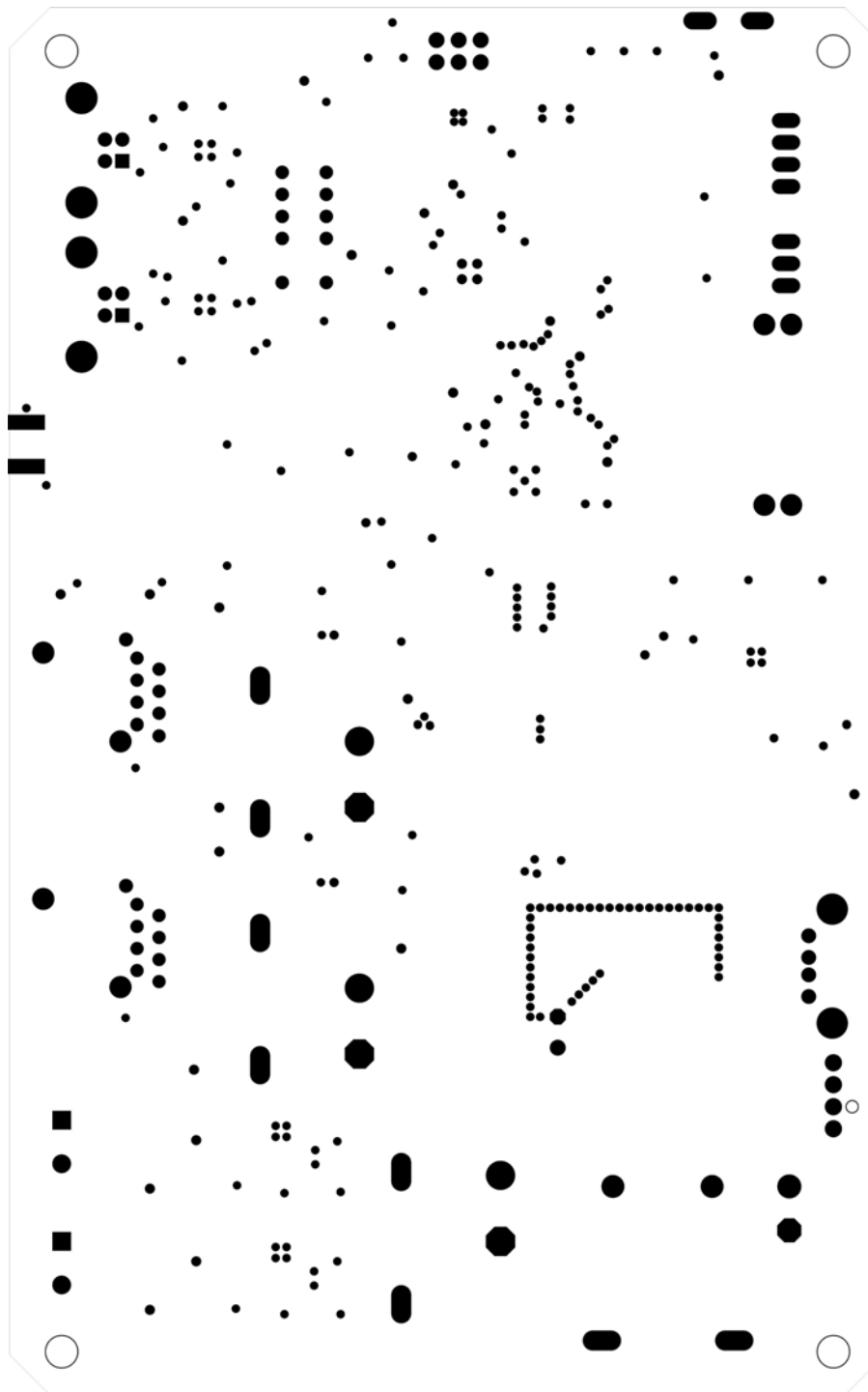


Figure C.10: Master Control Unit Bottom Solder Mask

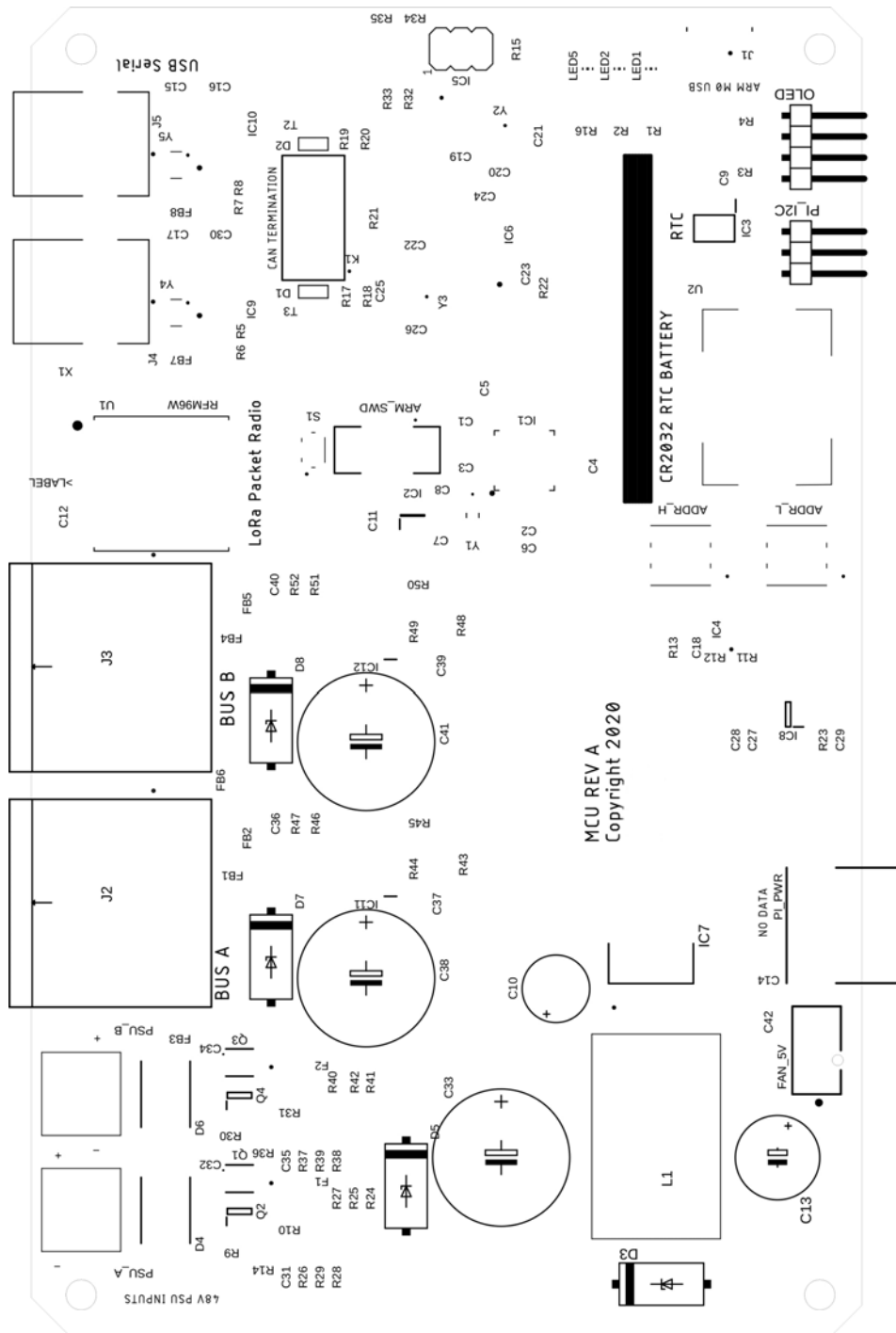


Figure C.11: Master Control Unit Top Silkscreen

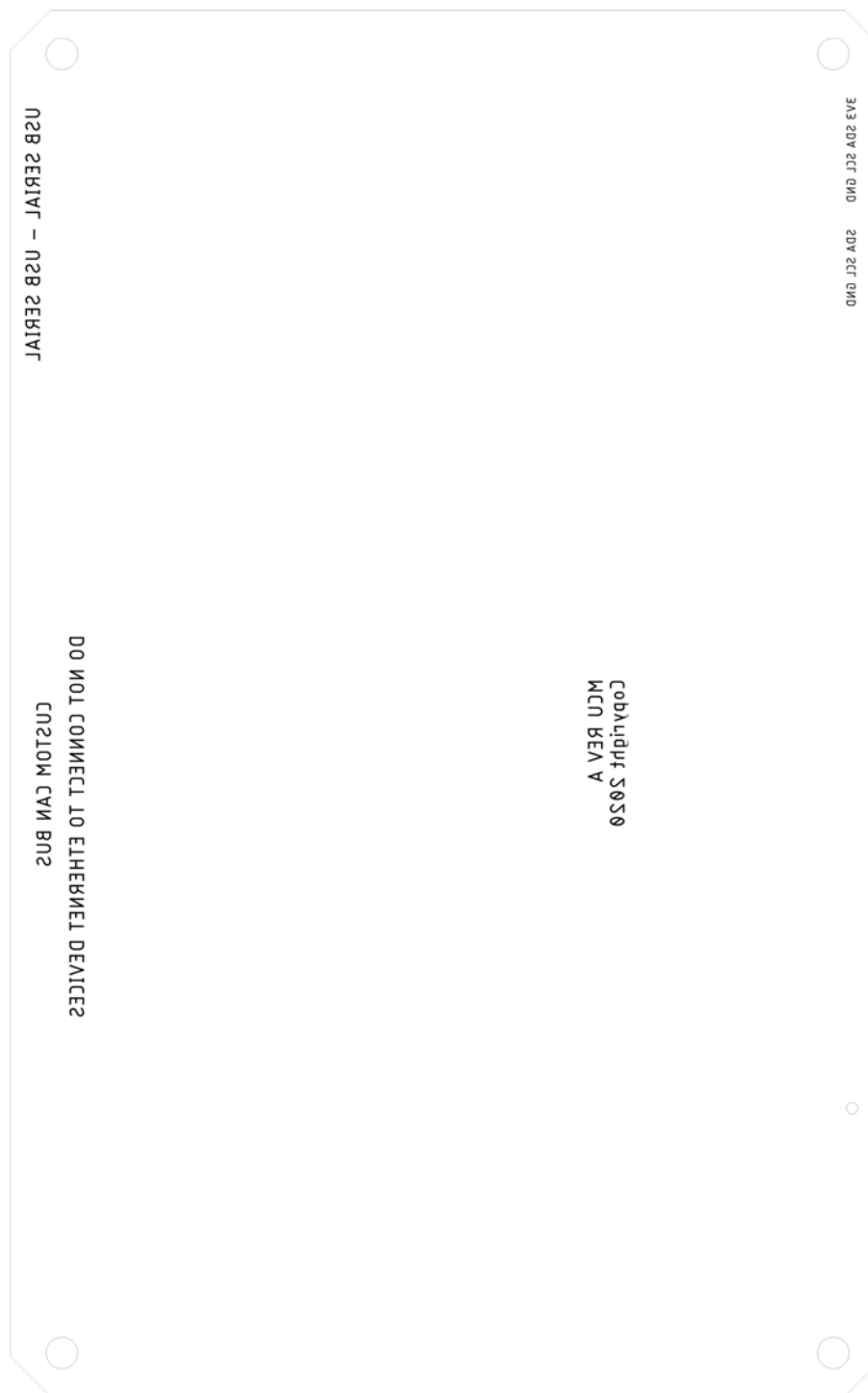


Figure C.12: Master Control Unit Bottom Silkscreen

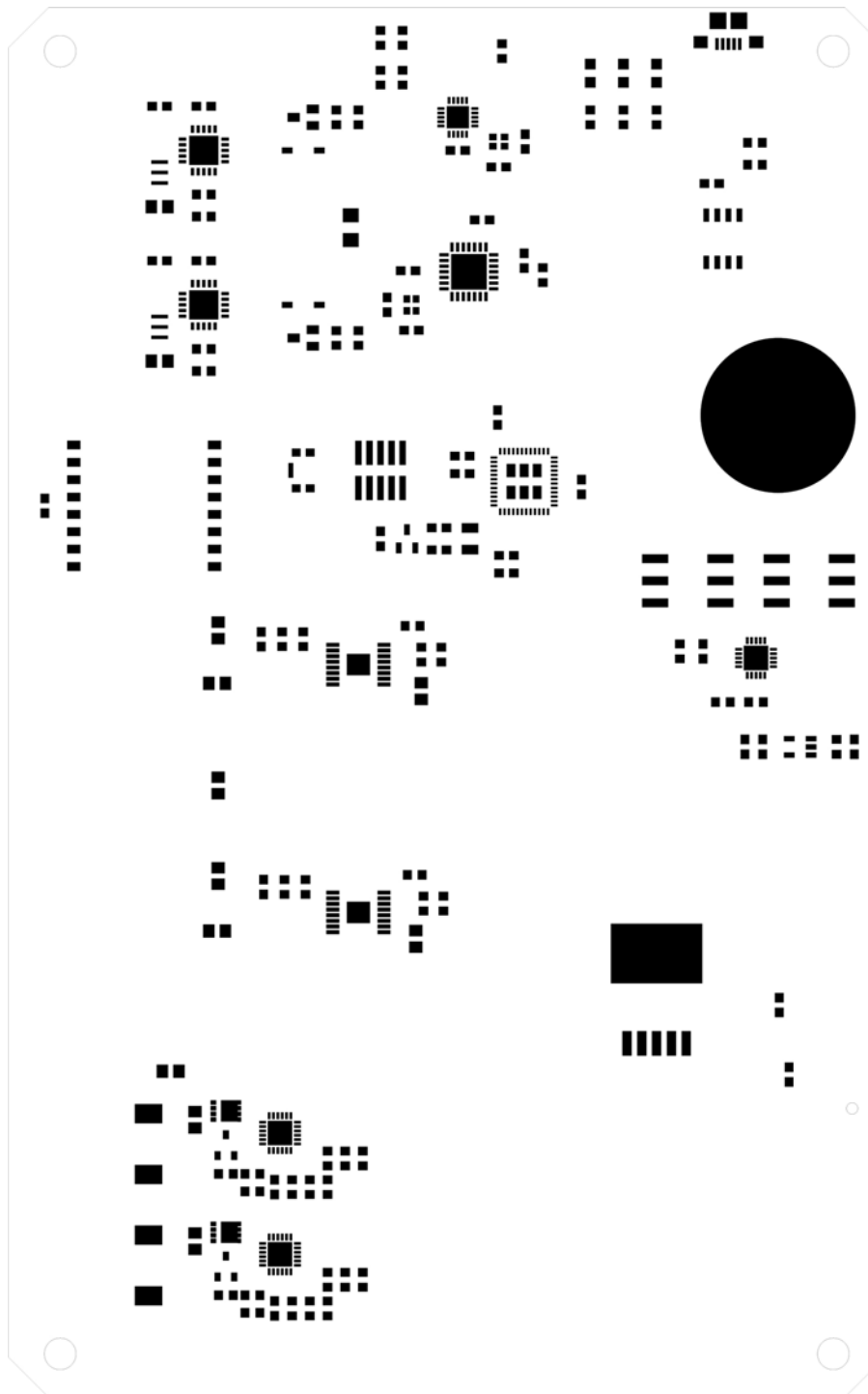


Figure C.13: Master Control Unit Solder Paste

### C.1.3 Bill of Materials

This subsection shows the bill of materials required to assemble one Master Control Unit circuit board. Table C.1 lists each manufacturer part number with a short description, quantity required per PCB, and extended cost for that line. In quantities required to assemble a single PCB, the total cost of electronic components for the Master Control Unit circuit board is £85.55. Note: this does not include the cost of the PCB, enclosure, or other electronics required separate to the PCB. This should be seen as a worst-case scenario device cost since pricing drops considerably at larger quantities.

Table C.1: Master Control Unit Bill of Materials

Part Number	Description	Qty.	Ext. Cost
885012206071	Capacitor $0.1\mu F$ 25V	10	£0.34
ERJ-3EKF1003V	Resistor $100k\Omega$	2	£0.15
ERJ-3EKF1002V	Resistor $10k\Omega$	9	£0.69
GRM188R61E106KA73D	Capacitor $10\mu F$ 25V	5	£1.40
ERJ-3EKF1372V	Resistor $13.7k\Omega$	2	£0.15
ERJ-3EKF1503V	Resistor $150k\Omega$	2	£0.15
ERJ-U03J181V	Resistor $180\Omega$	2	£0.15
1890963	Screw Terminal	2	£1.79
ERJ-3EKF1001V	Resistor $1k\Omega$	2	£0.15
RC0603FR-071M24L	Resistor $1.24M\Omega$	2	£0.15
CRCW06031M27FKEA	Resistor $1.27M\Omega$	2	£0.15
CC0603KRX5R8BB105	Capacitor $1\mu F$ 25V	12	£0.50
220AMA16R	Coded Switch	2	£3.52
885012006053	Capacitor $22pF$ 50V	6	£0.36
ERJ-3EKF2001V	Resistor $2k\Omega$	2	£0.15
ERJ-3EKF2371V	Resistor $2.37k\Omega$	2	£0.15
ERJ-3EKF3092V	Resistor $30.9k\Omega$	6	£0.46
ERJ-3EKF3481V	Resistor $3.48k\Omega$	2	£0.15
ERJ-3EKF4422V	Resistor $44.2k\Omega$	2	£0.15
47053-1000	Fan Header	1	£0.35
ERJ-3EKF4993V	Resistor $499k\Omega$	2	£0.15
ERJ-3EKF4701V	Resistor $4.7k\Omega$	4	£0.31
RT0603BRD0750KL	Resistor $50k\Omega$	4	£1.18
614004190021	USB A Connector	1	£1.01
ERA-8AEB121V	Resistor $120\Omega$	1	£0.51

Continuation of Table C.1			
Part Number	Description	Qty.	Ext. Cost
ERJ-3EKF7151V	Resistor 7.15k $\Omega$	2	£0.15
1N4148W-TP	Diode	2	£0.26
ERJ-3EKF9311V	Resistor 9.31k $\Omega$	2	£0.15
10118193-0001LF	USB <i>u</i> B Connector	1	£0.35
61729-0010BLF	USB B Connector	2	£1.52
AP2112K-3.3TRG1	LDO Regulator 3.3V	1	£0.36
ATSAMD21G18A-MU	ARM Microcontroller	1	£2.32
ATTINY84A-MU	AVR Microcontroller	1	£0.60
BAT-HLD-001-THM	Battery Holder	1	£0.22
BLM21PG221SN1D	Ferrite Bead	8	£0.62
BSS138	N-Channel MOSFET	2	£0.40
CSD19537Q3	Power MOSFET	2	£1.94
CSTNE12M0GH5L000R0	Resonator 12MHz	2	£0.57
DS3231MZ+	Real Time Clock	1	£5.90
EC2-5TNU	Latching Relay	1	£2.13
ECS-160-12-37B-CTN	Crystal 16MHz	2	£0.70
860080775016	Electro. Cap. 150 <u>F</u> 63V	1	£0.45
EEU-FP1E102	Electro. Cap. 1000 <u>F</u> 25V	1	£0.82
ESK108M063AM7AA	Electro. Cap. 1000 <u>F</u> 63V	3	£2.42
150080GS75000	LED Green	1	£0.14
HMK212BC7105KG-TE	Capacitor 1 <u>F</u> 100V	4	£1.46
KMR221GLFS	Tactile Switch	1	£0.42
LM2576HVSX-5.0	Switching Regulator 5V	1	£4.71
MCP2200-I/MQ	USB-UART Interface	2	£3.26
MCP23008-E/ML	GPIO Expander	1	£0.85
MCP25625-E/ML	CAN Controller	1	£1.95
MCP9701AT-E/TT	Temperature Sensor	1	£0.24
MMBT2222ALT1G	Transistor NPN	2	£0.20
NE8FAH	Modular Connector	2	£7.30
APT2012NW	LED Orange	1	£0.16
PE-53115NL	Inductor 150 <u>H</u>	1	£2.36
AP2012HD	LED Red	1	£0.13
SB560-E3/54	Schottky Diode Rectifier	4	£1.64
SHF-105-01-L-D-SM	Programming Header	1	£2.30
SMCJ70CA	TVS Diode 1.5kW	2	£0.87



<b>Continuation of Table C.1</b>			
<b>Part Number</b>	<b>Description</b>	<b>Qty.</b>	<b>Ext. Cost</b>
TPS26600PWPR	eFuse	2	£7.14
TPS26630RGET	eFuse	2	£9.34
CON-SMA-EDGE-S	SMA RF Connector	1	£1.35
LFXTAL009678	Crystal 32.7680kHz	1	£0.38
RFM96W	LoRa Radio Module 433MHz	1	£3.25
<b>End of Table</b>			

## C.2 Camera Control Unit

### C.2.1 Schematic

This subsection shows the electronic schematic sheets covering the Camera Control Unit. As with the Master Control Unit, sheet one, figure C.14, details the ATSAM21G18A ARM M0+ microcontroller and supporting circuits. The remaining sheets, figures C.15 to C.17, show the circuits providing the rest of the electronic functionality of the design. In comparison to the master control unit, there is less power supply circuitry as there is no new power output to other devices requiring protection. It is interesting to note that LANC requires only a limited amount of supporting components as can be seen on sheet two, figure C.15.

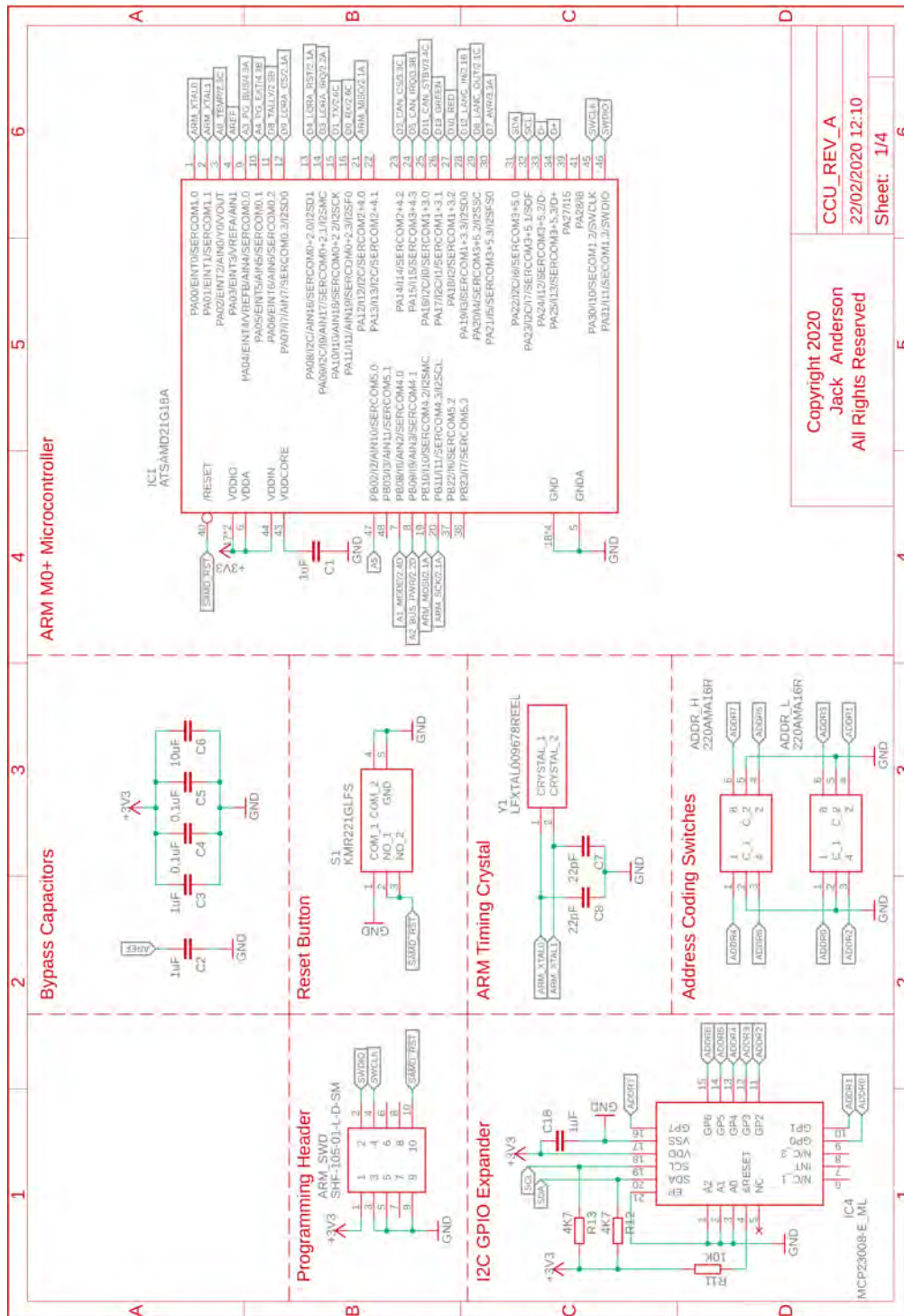


Figure C.14: Camera Control Unit Schematic – Sheet One

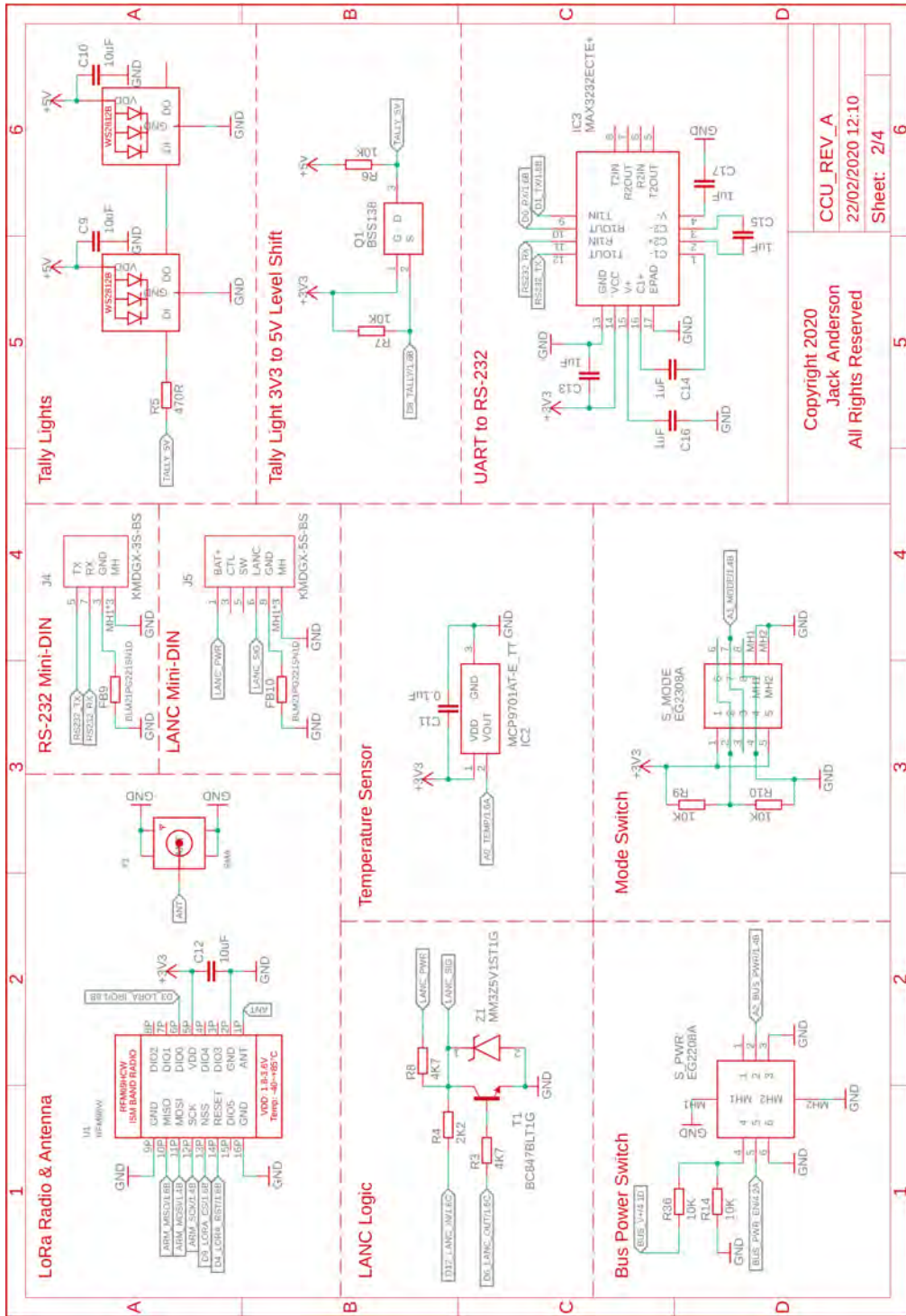


Figure C.15: Camera Control Unit Schematic – Sheet Two

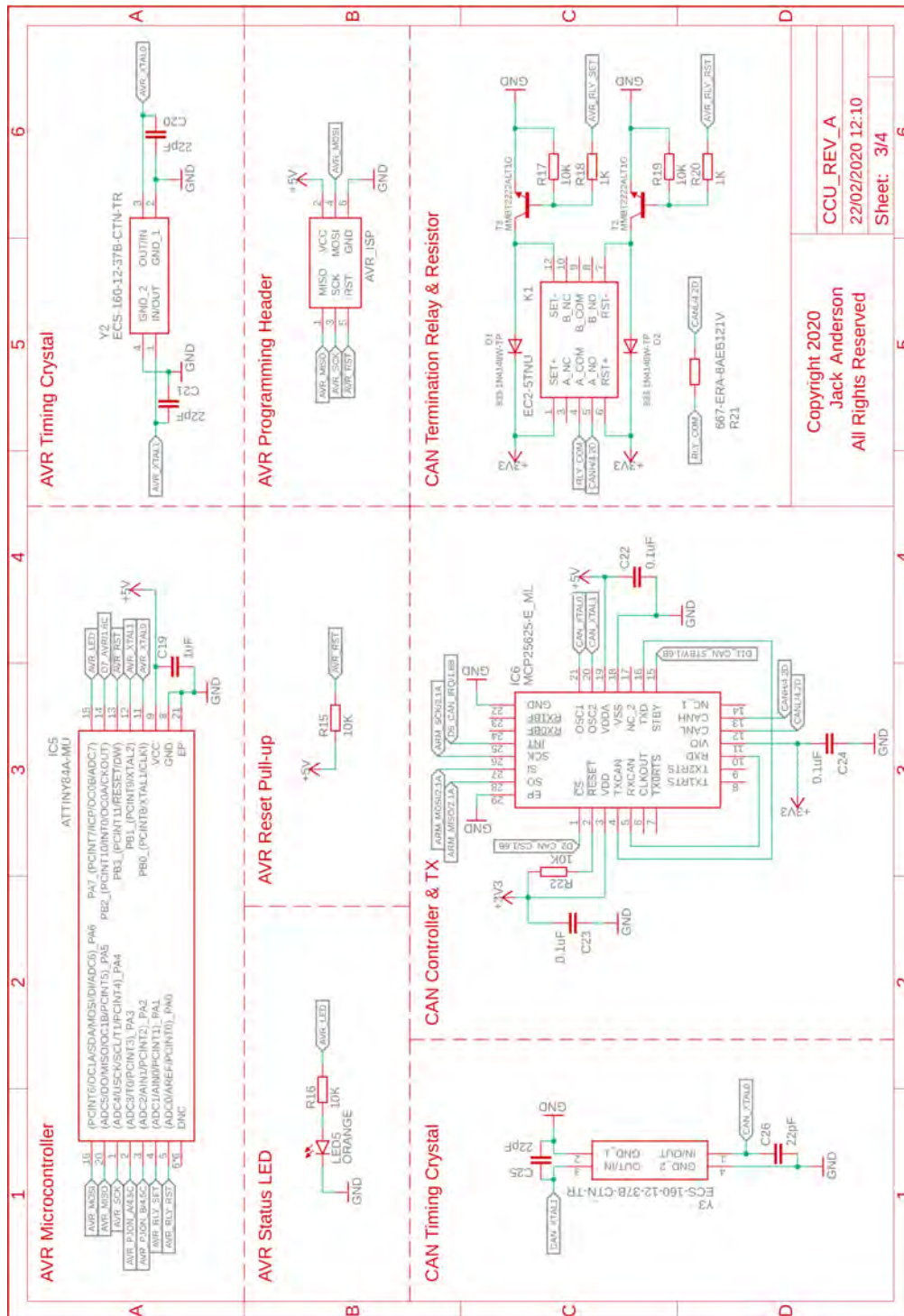


Figure C.16: Camera Control Unit Schematic – Sheet Three

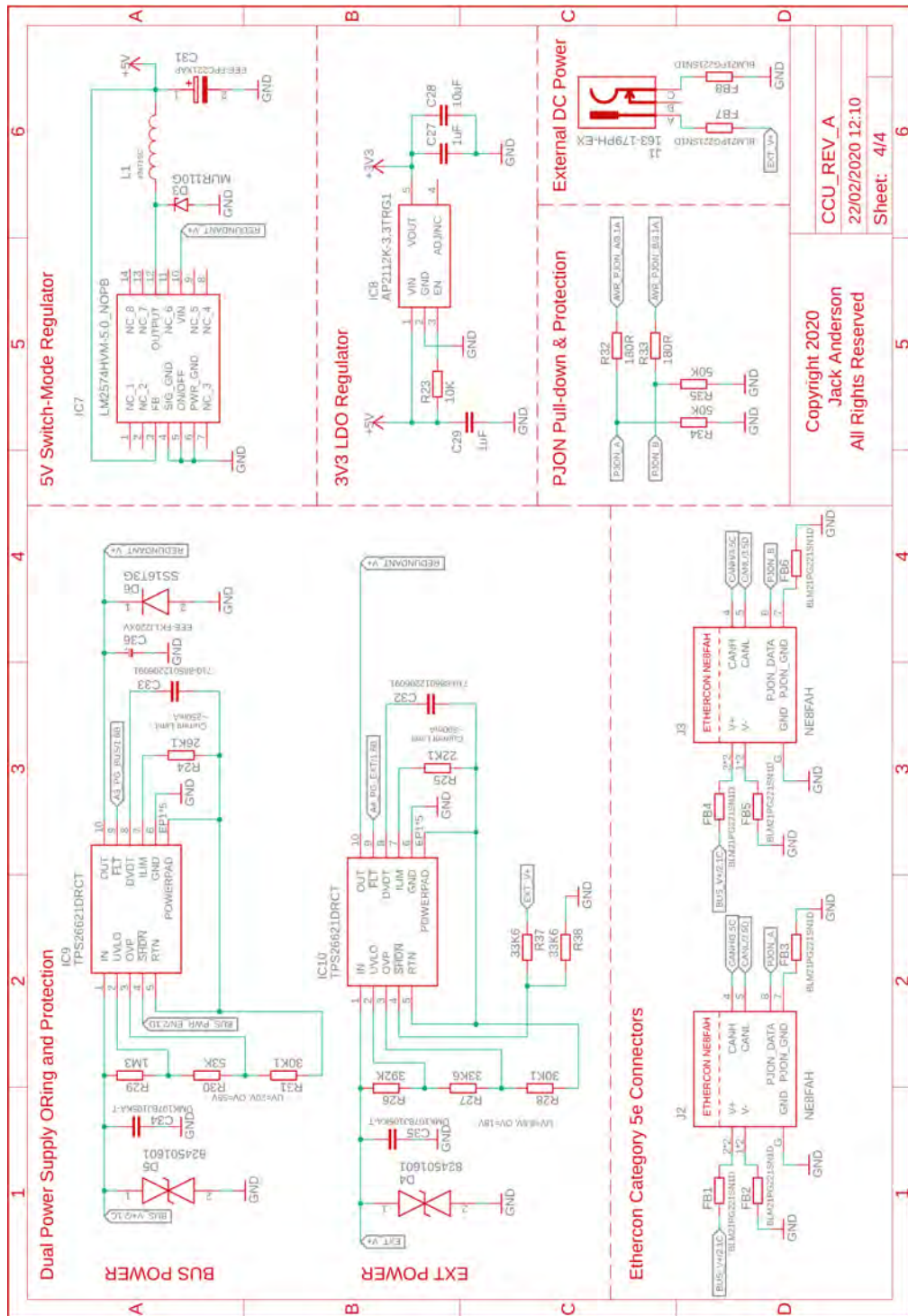


Figure C.17: Camera Control Unit Schematic – Sheet Four

## C.2.2 Board Layout

This subsection shows the physical board layout design layers as required for the manufacture of the Camera Control Unit circuit board, similar to the Master Control Unit. Figure C.18 shows all the layers, useful to understand how signals are connected across the entire board. Figures C.19 and C.20 show the top and bottom copper layers, with black areas indicating where copper remains after processing by the manufacturer. Figures C.21 and C.22 show the top and bottom solder mask layers, with black areas indicating where the green solder mask is removed leaving exposed copper. Figures C.23 and C.24 show the silkscreen layers, with black areas indicating text or other information printed onto the circuit board. Finally, figure C.25 shows the solder paste layer, with black areas indicating holes in the stencil used for placing solder onto the board during the assembly process carried out by the author.

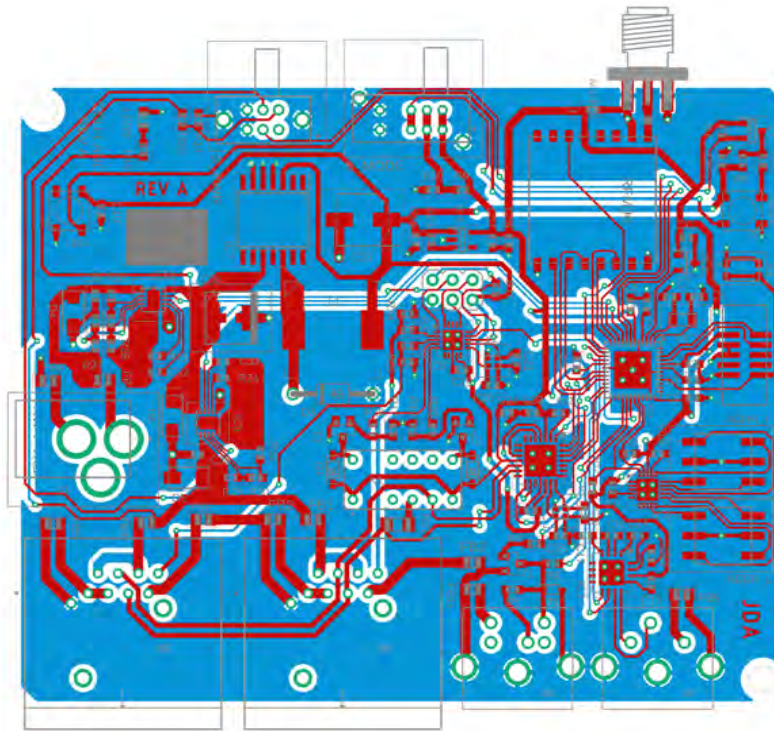


Figure C.18: Camera Control Unit All Layers

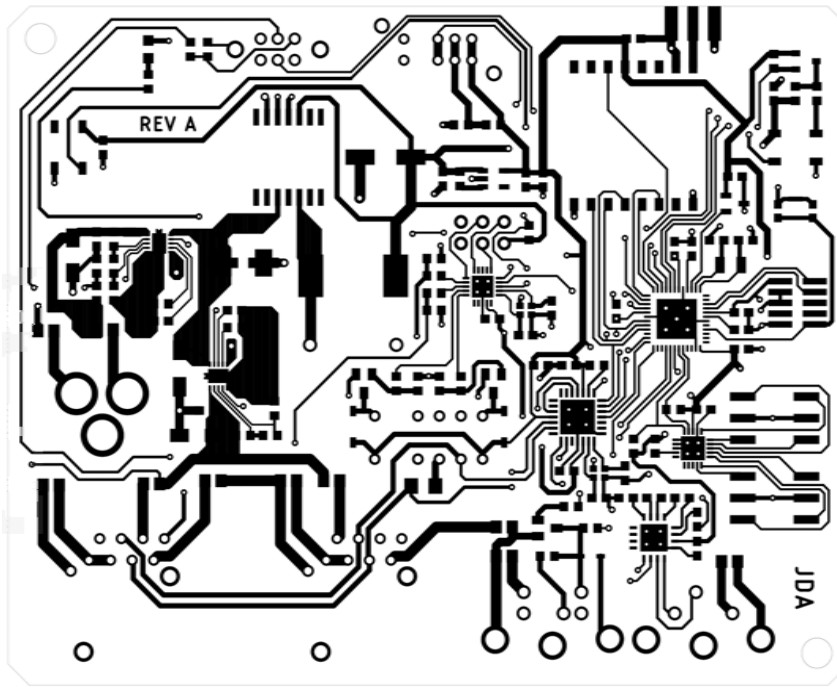


Figure C.19: Camera Control Unit Top Copper

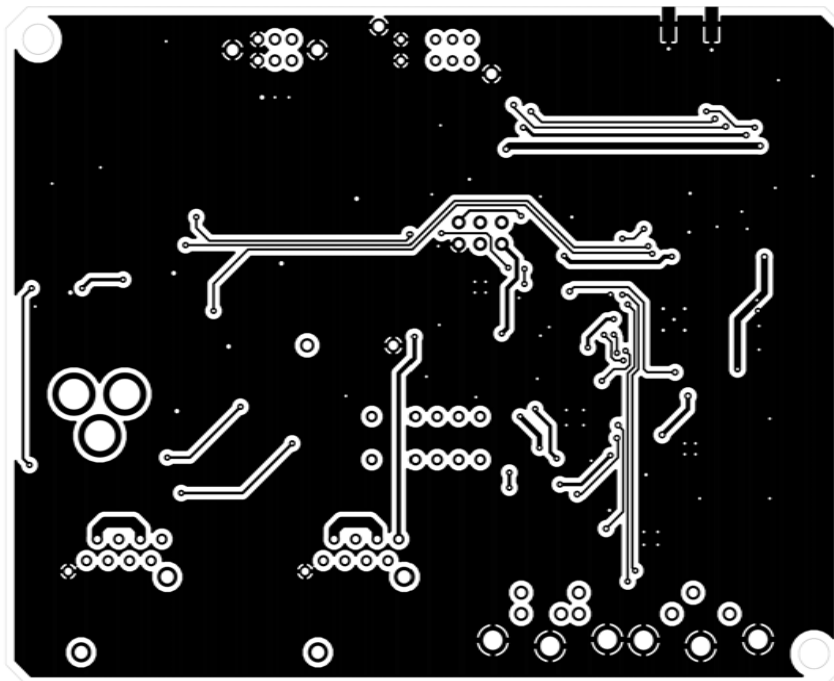


Figure C.20: Camera Control Unit Bottom Copper

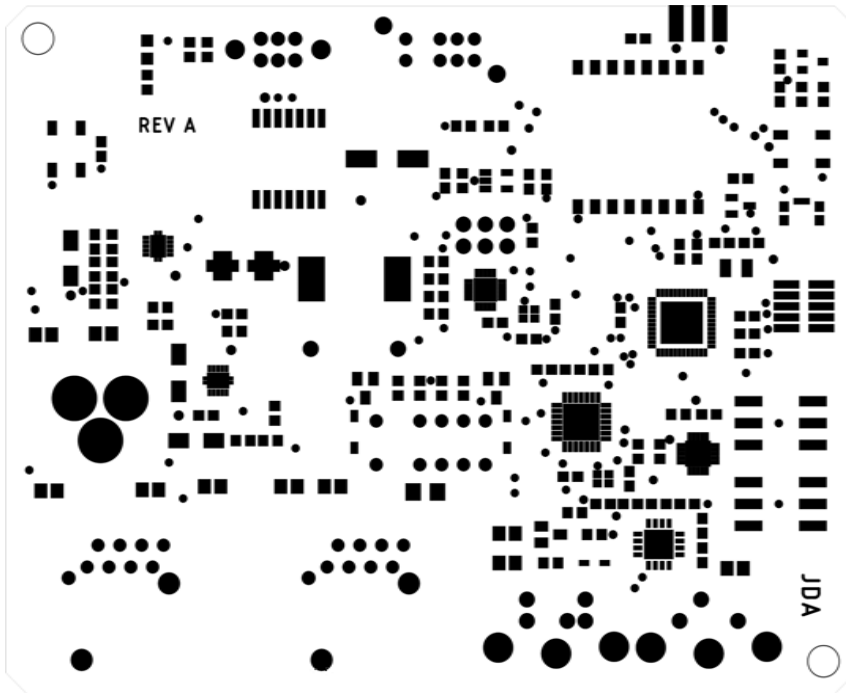


Figure C.21: Camera Control Unit Top Solder Mask

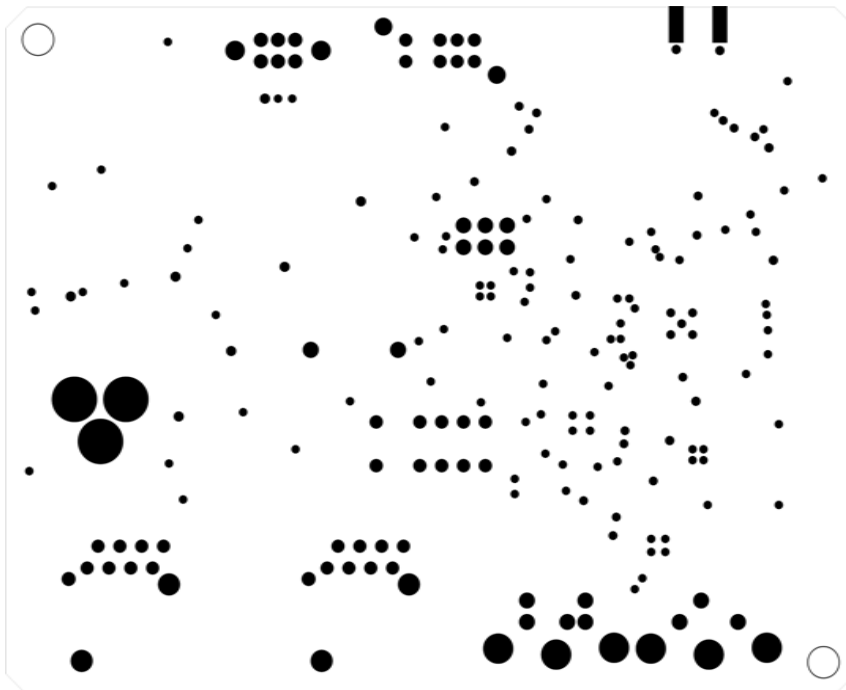


Figure C.22: Camera Control Unit Bottom Solder Mask



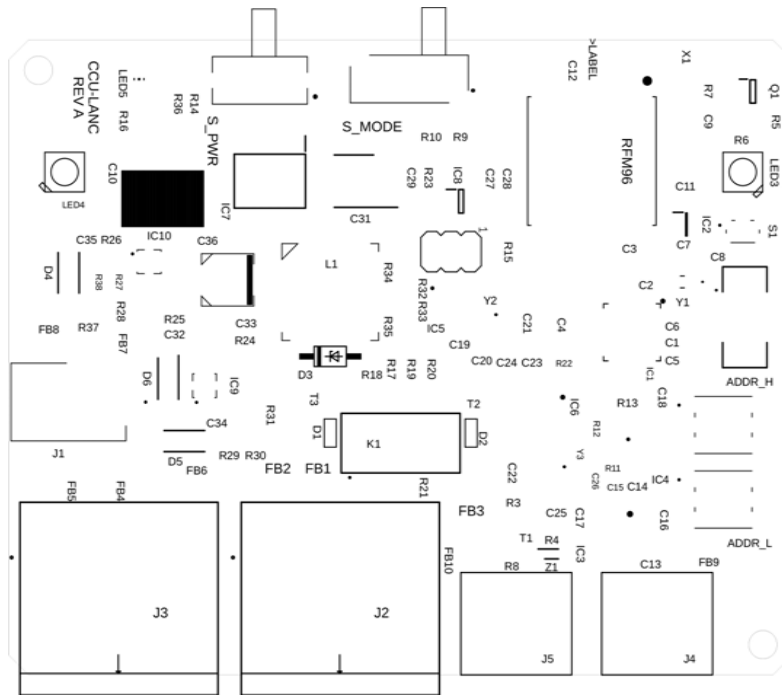


Figure C.23: Camera Control Unit Top Silkscreen

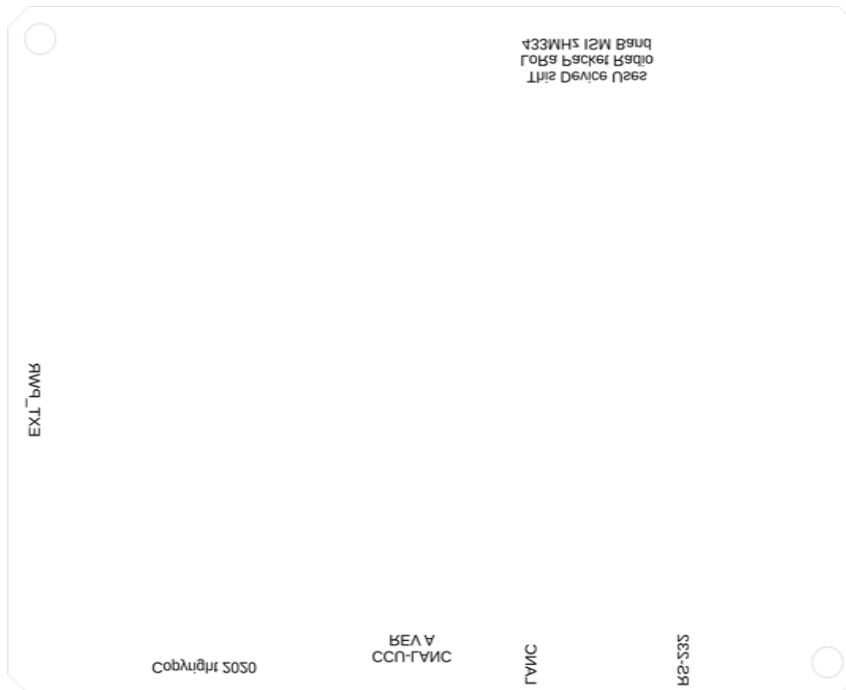


Figure C.24: Camera Control Unit Bottom Silkscreen

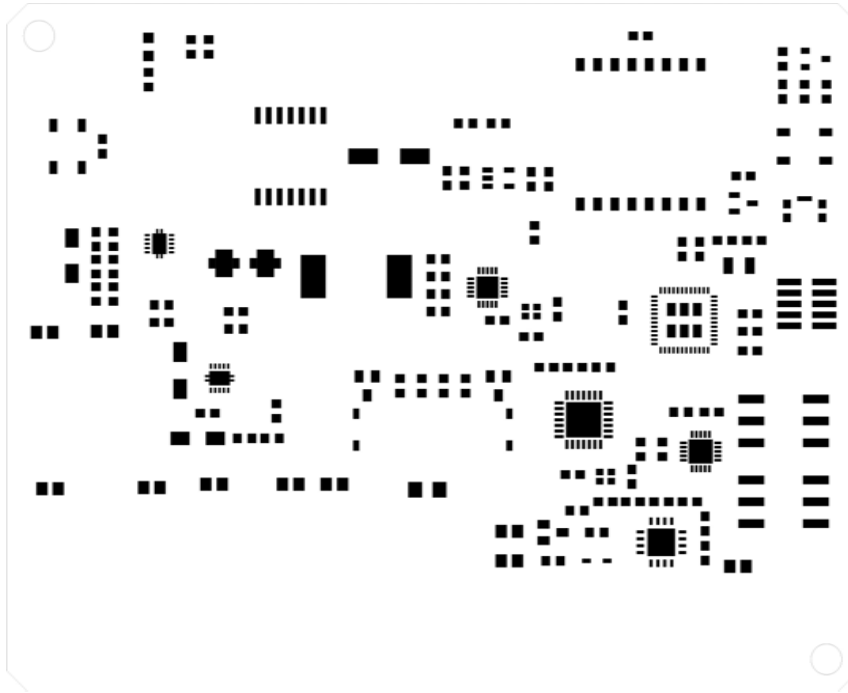


Figure C.25: Camera Control Unit Solder Paste

### C.2.3 Bill of Materials

This subsection shows the bill of materials required to assemble one Camera Control Unit circuit board. Table C.2 lists each manufacturer part number with a short description, quantity required per PCB, and extended cost for that line. In quantities required to assemble a single PCB, the total cost of electronic components for the Camera Control Unit circuit board is £55.67. Note: this does not include the cost of the PCB or enclosure. However unlike the Master Control Unit, the Camera Control Unit does not contain any other electronics separate from the PCB. Again, this should be seen as a worst-case scenario device cost since pricing drops considerably at larger quantities.

Table C.2: Camera Control Unit Bill of Materials

Part Number	Description	Qty.	Ext. Cost
885012206071	Capacitor 0.1 $\mu$ F 25V	6	£0.20
ERJ-3EKF1003V	Resistor 100k $\Omega$	13	£0.44
GRM188R61E106KA73D	Capacitor 10 $\mu$ F 25V	5	£1.40
163-179PH-EX	DC Power Connector	1	£0.96
ERJ-U03J181V	Resistor 180 $\Omega$	2	£0.15

Continuation of Table C.2			
Part Number	Description	Qty.	Ext. Cost
ERJ-3EKF1001V	Resistor $1k\Omega$	2	£0.15
ERJ-3EKF1304V	Resistor $1.3M\Omega$	1	£0.08
CC0603KRX5R8BB105	Capacitor $1\mu F$ 25V	12	£0.50
220AMA16R	Coded Switch	2	£3.52
ERJ-3EKF2212V	Resistor $22.1k\Omega$	1	£0.08
885012006053	Capacitor $22pF$ 50V	6	£0.36
ERJ-3EKF2612V	Resistor $26.1k\Omega$	1	£0.08
ERJ-3EKF2201V	Resistor $2.2k\Omega$	1	£0.08
ERJ-3EKF3012V	Resistor $30.1k\Omega$	2	£0.15
ERJ-3EKF3402V	Resistor $34k\Omega$	3	£0.23
ERJ-3EKF3923V	Resistor $392k\Omega$	1	£0.08
ERJ-PB3B4700V	Resistor $470\Omega$	1	£0.19
48471SC	Inductor $470\mu H$	1	£0.78
ERJ-3EKF4701V	Resistor $4.7k\Omega$	4	£0.31
RT0603BRD0750KL	Resistor $50k\Omega$	2	£0.59
RN73H1JTTD5302D100	Resistor $53k\Omega$	1	£0.19
ERA-8AEB121V	Resistor $120\Omega$	1	£0.51
885012206091	Capacitor $0.022\mu F$ 50V	2	£0.10
824501601	TVS Diode 400W	2	£0.43
1N4148W-TP	Diode	2	£0.26
AP2112K-3.3TRG1	LDO Regulator 3.3V	1	£0.36
ATSAMD21G18A-MU	ARM Microcontroller	1	£2.32
ATTINY84A-MU	AVR Microcontroller	1	£0.60
BC847BLT1G	Transistor NPN	1	£0.09
BLM21PG221SN1D	Ferrite Bead	10	£0.41
BSS138	N-Channel MOSFET	1	£0.20
EC2-5TNU	Latching Relay	1	£2.13
ECS-160-12-37B-CTN	Crystal 16MHz	2	£0.70
EEE-FK1J220XV	Electro. Cap. $22\mu F$ 63V	1	£0.72
EEE-FPC221XAP	Electro. Cap. $220\mu F$ 16V	1	£0.54
EG2208A	Slide Switch DPDT	1	£0.62
EG2308A	Slide Switch DP3T	1	£0.59
KMDGX-3S-BS	Mini-Din Connector 3P	1	£0.75
KMDGX-5S-BS	Mini-Din Connector 5P	1	£0.82
KMR221GLFS	Tactile Switch	1	£0.42

<b>Continuation of Table C.2</b>			
<b>Part Number</b>	<b>Description</b>	<b>Qty.</b>	<b>Ext. Cost</b>
LFXTAL009678	Crystal 32.7680kHz	1	£0.38
LM2574HVM-5.0	Switching Regulator 5V	1	£3.57
MAX3232ECTE+	RS-232 Interface	1	£3.14
MCP23008-E/ML	GPIO Expander	1	£0.85
MCP25625-E/ML	CAN Controller	1	£1.95
MCP9701AT-E/TT	Temperature Sensor	1	£0.24
MM3Z5V1ST1G	Zener Diode 5.1V	1	£0.09
MMBT2222ALT1G	Transistor NPN	2	£0.20
NE8FAH	Modular Connector	2	£7.30
APT2012NW	LED Orange	1	£0.16
SHF-105-01-L-D-SM	Programming Header	1	£2.30
CON-SMA-EDGE-S	SMA RF Connector	1	£1.35
SS16T3G	Schottky Diode Rectifier	1	£0.12
TPS26621DRCT	eFuse	2	£4.58
UMK107BJ105KA-T	Capacitor 1 $\mu$ F 50V	2	£0.34
KMDLAX-5P	Mini-Din Plug 5P	1	£2.11
1N4934G	Diode Rectifier	1	£0.18
WS2812B	Addressable RGB LED	2	£0.47
RFM96W	LoRa Radio Module 433MHz	1	£3.25
<b>End of Table</b>			

## D Mechanical Drawings

This appendix contains mechanical drawings of the Camera Control Unit CAD parts and assembly. Figure D.1 shows the assembled enclosure comprised of base, top, plug, and two tally parts. The Ethercon connectors are also shown since the manufacturer provides CAD files for this purpose. Figure D.2 shows the base enclosure part which is the piece the PCB attaches to as well as the mounting screw insert in the bottom. Figure D.3 shows the top part of the enclosure which is attached to the bottom part using a combination of screws that mount to the Ethercon connectors and simple clips that snap into the bottom part. Figure D.4 shows the plug and tally parts, sundry items that complete the 3D printed components of the assembly.

Note: only basic dimensions are shown since the parts are intended for CAM using the full CAD models so it is not appropriate to include complete dimensions.

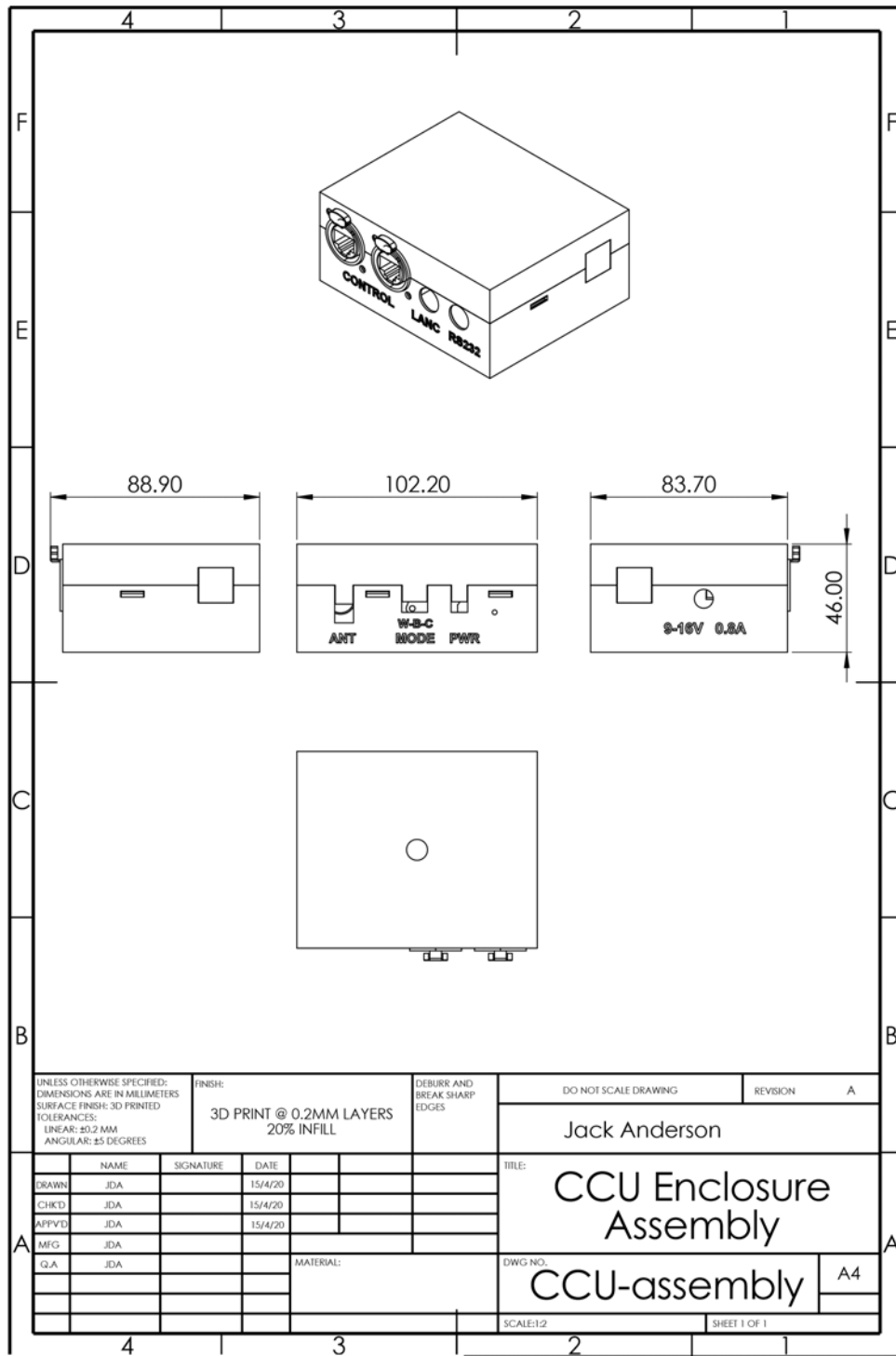


Figure D.1: CCU Enclosure Assembly Drawing

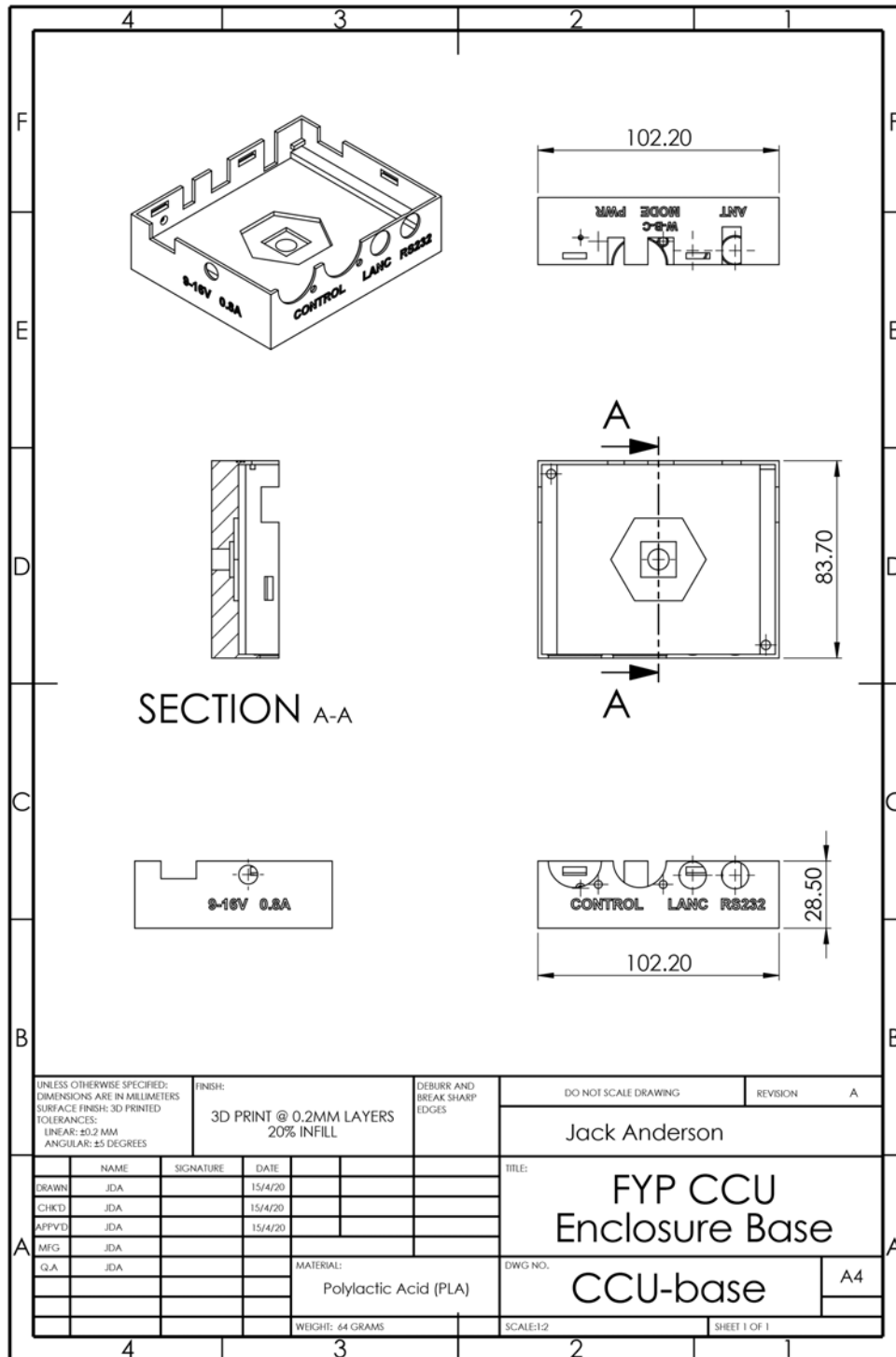


Figure D.2: CCU Enclosure Base Drawing

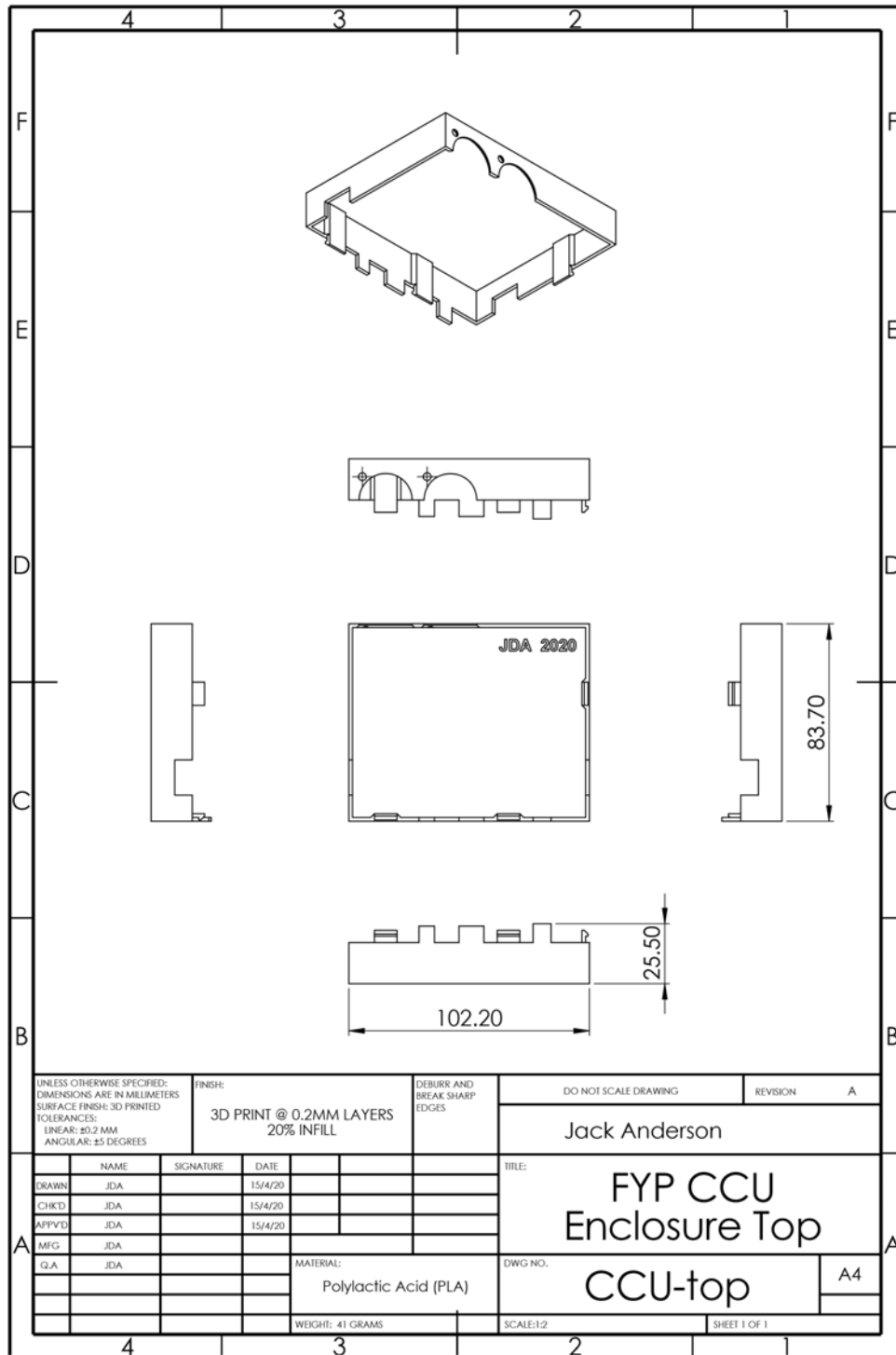


Figure D.3: CCU Enclosure Top Drawing



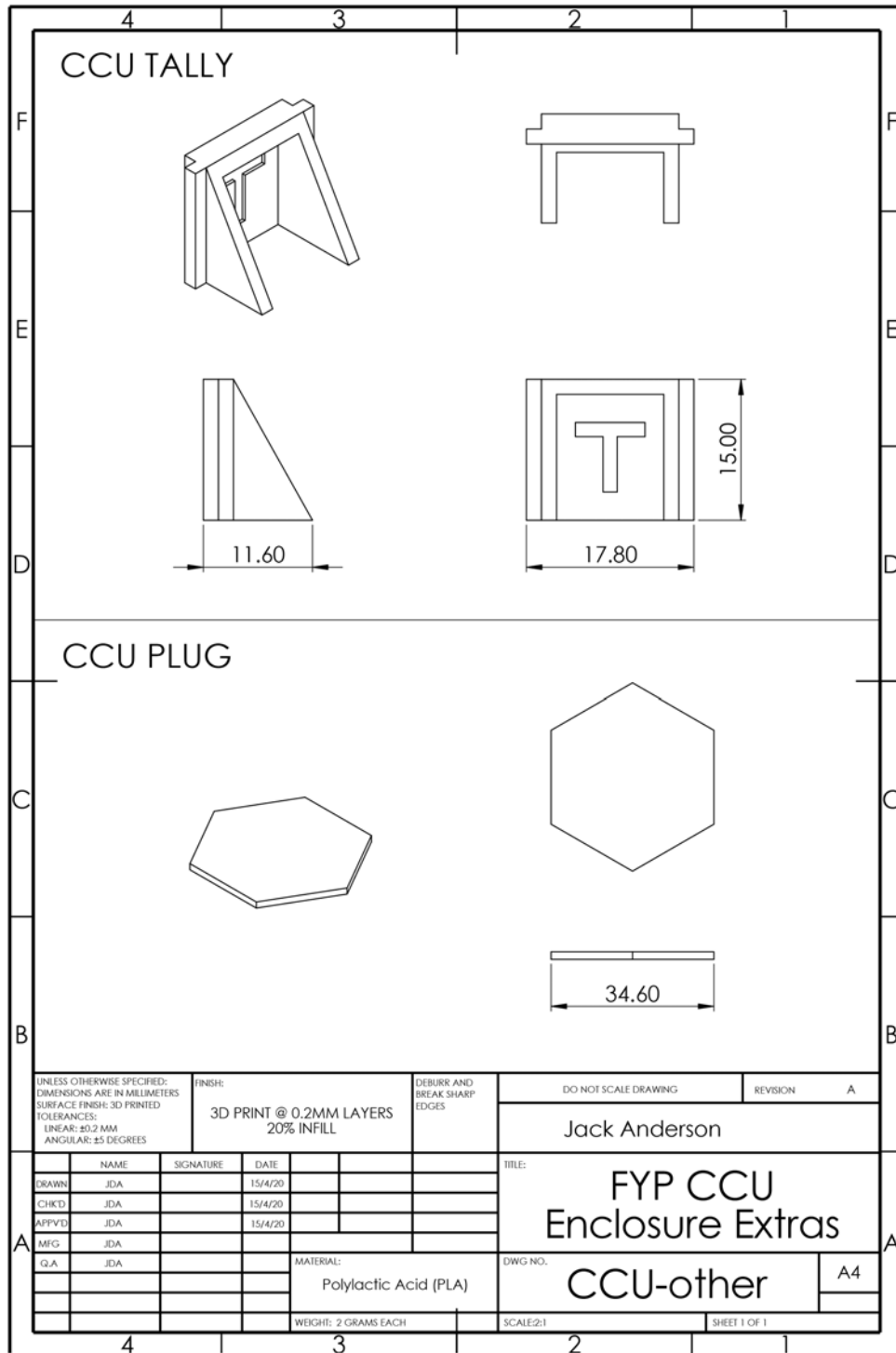


Figure D.4: CCU Enclosure Sundries Drawing