Development of a Portable Electroluminescence Measurement System for Photovoltaic Modules –2024S1

A thesis submitted in partial fulfilment of the degree of
Bachelor of Engineering (Honours)

by
Neil Arthur Bradbury
U5841150

Supervisor: Dr Marco Ernst
Examiner: Mr Anh Bui

College of Engineering and Computer Science
The Australian National University

May 2024

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of the author's knowledge, it contains no material previously published or written by another person, except where due reference is made in the text.

Neil Bradbury
7 June 2024

Acknowledgements

I like to thank the following for their contributions to my project over the past year:

Dr Marco Ernst for providing me the opportunity to work on this project and the supporting feedback given during the project. This project was an invaluable experience which gave me great exposure to both software and hardware engineering.

Mr Chris Jones who patiently and greatly assisted with the electronics PCB design work. Chris' expertise, guidance and feedback during this project significantly expanded my understanding of practical electronics and how to build electronics circuits for real world applications.

Mr Ahn Bui for providing the support as the second supervisor for this project.

The ENGN8170 project team, *Team Solar*, for their previous work. Thanks to their efforts in the documentation, I was able to follow and extend their work in 2023 and 2024.

Abstract

Electroluminescence (EL) imaging is an effective method for analysing PV modules to detect faults. A portable EL measurement system (PELMS) device is a EL imaging device that can be used on PV modules in situ (i.e. in the field) rather than in a lab. A cheaper low cost (PELMS) device would be of benefit to Solar Panel owners and users. The ENGN8170 project team, *Team Solar* developed a prototype PELMS device that used an innovative modulation technique to eliminate external lighting noise the enhances EL image quality. This project extended the work on this prototype by increasing the image resolution, creating a PCB of the circuitry, optimising the power on the PCB, increasing the current rating, and improving the circuitry enclosure. The outcomes of this project were software that could generate 12-bit resolution Red, Green and Blue (RGB) EL images, a PCB circuit board of the current regulator designed to handle 20 A that is ready to be tested and an improved electronics enclosure for the Current Regulator device.

Content

List of figures

List of tables

Nomenclature

Abbreviations

ANU	Australian National University
bpp	bits-per-pixel
DC	Direct Current
EL	Electroluminescence
GUI	Graphical User Interface
GPIO	General Purpose Input/Output
IR	Infrared
ISP	Image Signal Processor
PCB	Printed Circuit Board
PELMS	Portable Electroluminescence Measurement System
PID	Potential Induced Degradation
PNG	Portable Network Graphics
PV	Photovoltaic
RF	Radio Frequency
RGB	Red, Green, Blue

Chapter 1 Introduction

The goal of this project was to continue the work of a portable electroluminescence measurement system (PELMS) for photovoltaic (PV) modules that had been previously developed in Semester 1 2023 by the ENG8170 student team, *Team Solar*.

Electroluminescence (EL) imaging is a test that can be used to identify defects in photovoltaic (PV) modules or solar panels. EL imaging has industrial applications to the solar panel industry in areas such as quality control in PV module production and fault detection in PV modules employed in the field [1, p. 10].

Two main methodologies for EL imaging are [2, p. 1]:

- conducting EL imaging in an indoor lab with optimal conditions such as minimal light, optimal temperature, and specialised equipment. This method is the costliest but produces the most accurate imaging results.
- conducting EL imaging in-situ using a portable EL measurement device. This method produces less accurate imaging results than the lab but is cheaper. The imaging is performed faster, and the PV module doesn't need to be relocated. This method was the focus of the previous work by *Team Solar* and for this project.

In 2023, *Team Solar* developed a working PELMS prototype using a portable computing system, high quality camera and portable power source with an attached current regulator. They implemented a unique image processing algorithm used to eliminate external lighting noise that enhanced the quality of the EL images.

The scope of the design and development work was to extend the *Team Solar* 2023 prototype by implementing deliverables that were based on 5 recommendations from *Team Solar* documentation ([2] and [3]). The recommendations are given in Table 1 and the 5 deliverables developed for this project to meet the recommendations are given in Table 2.

	2023 PELMS Recommendation
1	Increase image capture from 8-bit low-resolution to 12-bit resolution to improve image detail.
2	Develop a more portable and reliable enclosure, this will be an enclosure that houses components securely and prevents dust and water ingress.
3	Implement power management to optimise power consumption.
4	Create a Printed Circuit Board (PCB) for circuitry currently seated on a prototyping board.
5	Increase the current rating above 10 A to allow the device to scale to work with larger PV modules.

Table 1: 2023 PELMS Recommendations

	2024 PELMS Deliverables
1	Develop the Python code to handle 12-bit resolution images and redevelop/update the Graphical User Interface (GUI).
2	Design and construction of an improved enclosure.
3	Modification of the electrical circuitry to optimise power.
4	Construction of the PCB for the electrical circuitry to replace the prototyping board.
5	Recalculate the component values of the electric circuit to handle loads above 10 A.

Table 2: 2024 PELMS Deliverables

This design and development project was completed using a combination of engineering hardware and software skills from the following engineering-related fields:

- Software Development – used to develop Python code to manipulate images and update a GUI for deliverable 1.
- PCB Design – used to design and develop a PCB with optimised power and increased current capacity, for deliverables 3, 4 and 5.
- Electronics Enclosure Design – used to design an enclosure that can house the electrical circuitry for deliverable 2.

The work in this project contributes to the ongoing development of a PELMS device that provides for the availability of a low cost EL imaging device for solar panels users. An example of the utility of a PELMS device could be for a rural user who may be disconnected from the main power grid and is reliant on an independent solar power microgrid [2, p. 1]. A low cost PELMS device could be used to identify solar panel defects in place rather than having to dismantle and transport the panels for testing in a lab space. Enhancement of the PELMS device can enable solar panel users to perform their own inspections and assist with solar panel maintenance and management.

Chapter 2 Background

The PELMS device is a device that uses EL imaging to identify defects in PV modules. The workings of the device are based on:

- the principle that PV modules can emit near infrared (IR) light that is detectable by digital cameras to produce EL images, and
- a modulation technique implemented in the 2023 PELMS device that was designed to remove the external lighting noise that enhance the EL image quality.

Team Solar developed a PELMS device that produced EL images using the modulation technique. The system that they developed provided the foundation for this project.

2.1 Photovoltaic Modules and Electroluminescence Imaging

Photovoltaic (PV) modules are configurations of PV cells, which are cells constructed from semiconductor materials that generate electrical current when exposed to sunlight. In general PV modules operate by absorbing light to produce direct current (DC) electrical power. The inverse phenomenon is also possible due to the principal of reciprocity. In this circumstance a DC electrical power source injects current into a PV module that causes the PV module to produce light [2, p. 1]. The light produced is emitted in the near IR range and cannot be seen by the human eye [4, p. 1254].

The phenomenon of producing a near IR light by injecting electricity into a PV module is electroluminescence (EL) and the technique of capturing an image of the near IR light using an appropriate detector is EL imaging [4, p. 1254]. The detector in most cases is a digital camera that can capture the near IR light emitted by the PV module. An example of an EL image of a PV module is shown in Figure 1. A test current is applied to the PV module which emits the near IR light captured in the EL image. The portions of the image where the PV module has darker PV cells are cells which have been affected by potential induced degradation (PID) and are in a failure state (i.e. they are not emitting any light).

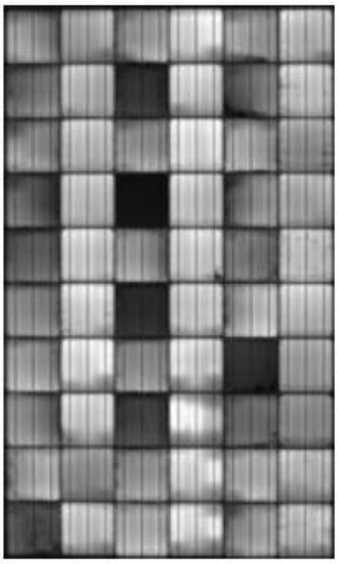

Figure 1: EL image of PV module (from [1, pp. 45, Fig 2-27b)].

An industrial application of EL imaging is to identify defects in PV modules. An example of PV module defects identified using EL imaging is shown in Figure 2 where the EL image shows a PV module with cracked cells. When using a PELMS device EL imaging can be used for defect inspection at various points in the lifetime of the PV module, such as quality control in production, mechanical damage during transportation and quality degradation in field [1, p. 10]. This information is useful for suppliers and users of PV modules so they are informed when damage is likely to occur e.g. during transportation or installation and can take preventative and/or precautionary steps e.g. more protection in transportation or more specialised tools for installation etc.

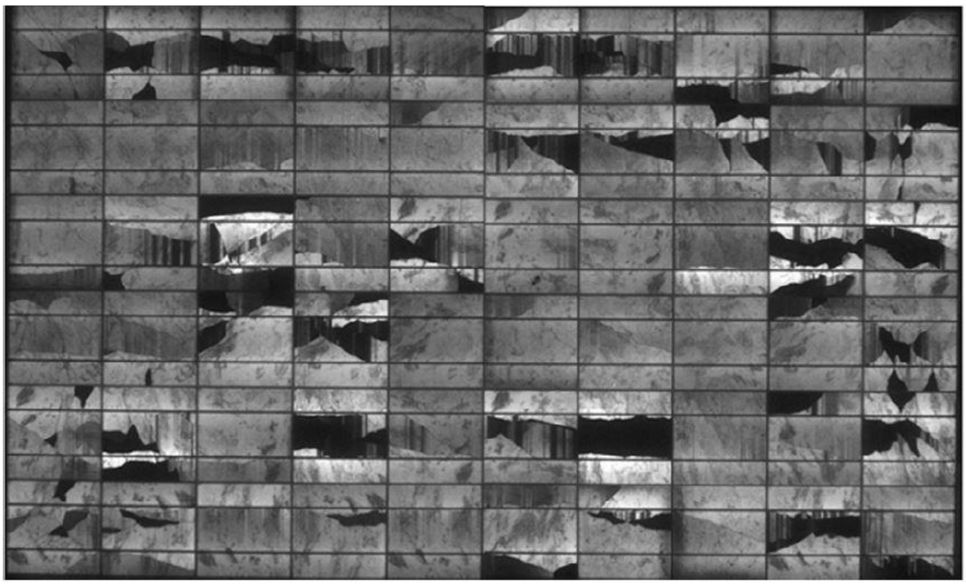

Figure 2: EL image of PV module with cracked cells (from [1, pp. 68, Fig 3-12])

2.2 PELMS Modulation Technique

The 2023 PELMS device uses a unique image processing algorithm to eliminate the external lighting noise in an EL image [2, p. 1].

When an EL image of a PV module is captured using a digital camera the image is composed of a visible light component and the near IR light component. The visible light component can be considered external lighting noise, and the near IR light component is the measurement information of the PV module, so the goal to improve the quality of the EL image is to reduce or eliminate the external lighting noise. The modulation technique used to do this in the PELMS system is shown in Figure 3. The PELMS device uses a digital camera that takes a sequence of *dark* and *bright* images of the PV module where:

- *dark* images are images of the PV module with no current injected into the module and contain only the visible light component of the PV module image, and
- *bright* images are images of the PV module with a suitable current injected into the module that causes the module to emit near IR light and so the image contains both a visible light component and a near IR light component of the PV module image.

The orange diamond in Figure 3 mark the times that the camera is triggered to capture an image and the blue line shows the current level injected into the PV module. The *dark* images are captured when no current (i.e. 0 A) is injected into the module and the camera is triggered (i.e. when an orange diamond is visible) and *bright* images are captured when current (i.e. 10 A) is injected into the module and the camera is triggered.

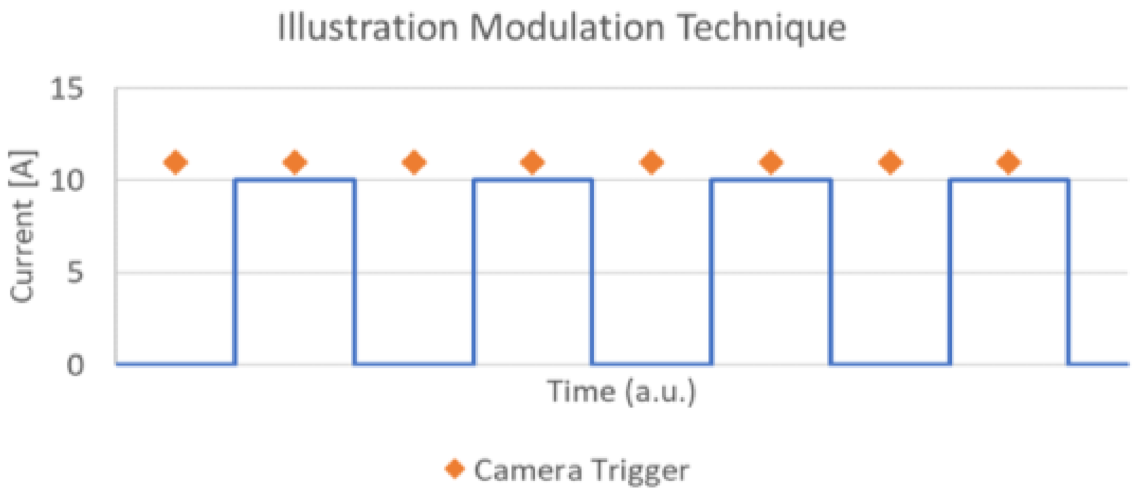

Figure 3: Modulation Technique Timing (from [2, pp. 1, Fig 1.]).

The PELMS device captures a sequence of *dark* and *bright* images. The difference between consecutive *dark* and *bright* images is taken which will produce a sequence of *difference* images. The *difference* image should contain only the near IR component of the image i.e. by subtracting the *dark* image from the *bright* image the visible light component is removed leaving behind only the near IR component of the image). This sequence of images can then be summed together and averaged to produce the higher quality EL image.

2.3 PELMS 2023 Development

In Semester 1 of 2023, the ENGN8170 project team, *Team Solar*, built a PELMS device that successfully implemented the modulation technique previously described. The testing configuration for the system is shown in Figure 4. The developed system imaged one solar panel and produced EL images within 5 minutes and the resultant image of a PV module under test is shown in Figure 5.

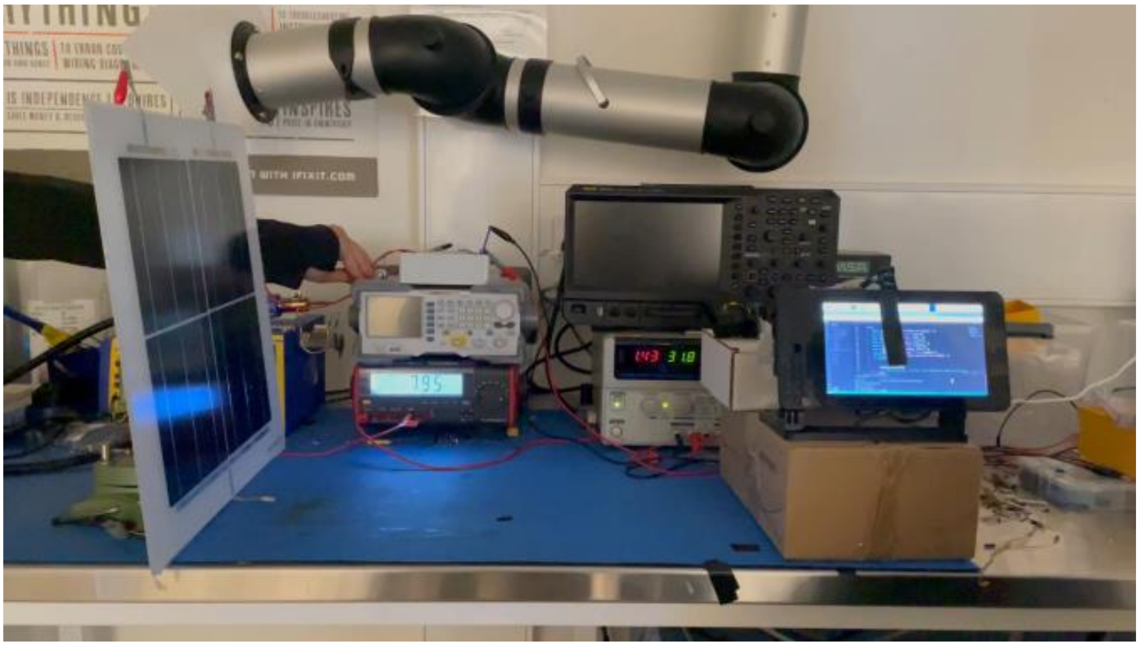

Figure 4: Testing setup of *Team Solar* 2023 PELMS system (from [2, pp. 8, Fig E.1]).

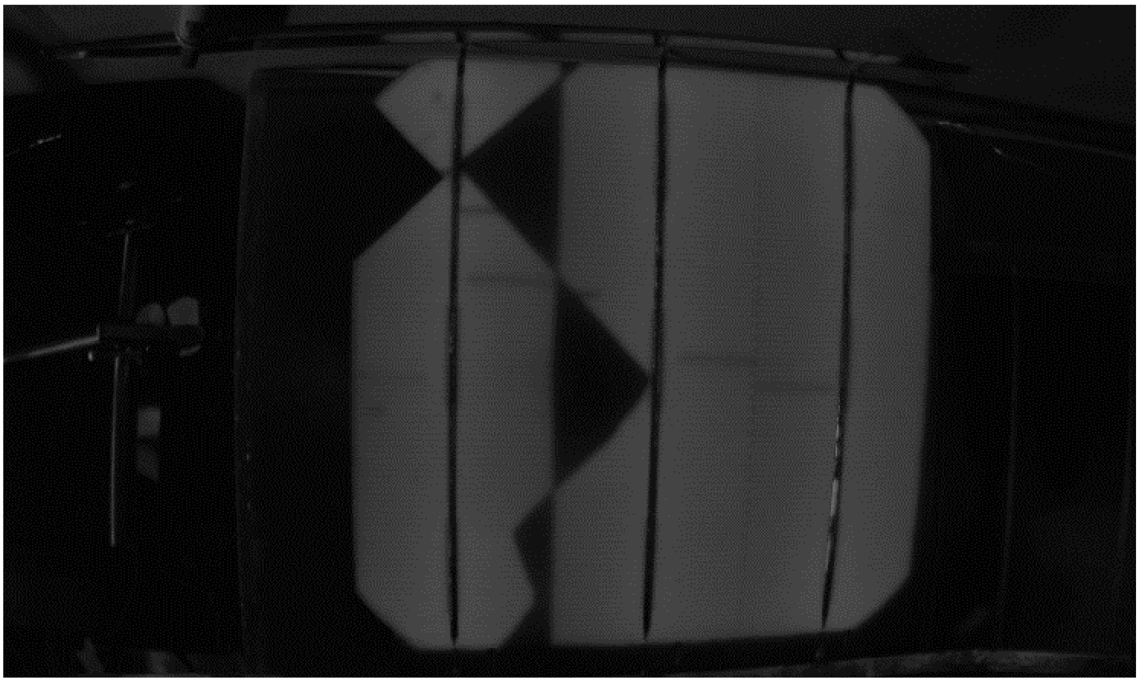

Figure 5: EL image produced by *Team Solar* 2023 PELMS system (from [2, pp. 5, Fig 8]).

The developed PELMS system was built comprising of cheaper more readily available components such as Raspberry Pi and Raspberry Pi HQ cameras and *Team Solar* documented their system in their project report as well as recommendations for future work [2].

The system block diagram of the *Team Solar* 2023 PELMS device is shown in Figure 6. The system is divided two subsystems:

- the Electrical Circuit subsystem, and
- the Camera Algorithm subsystem.

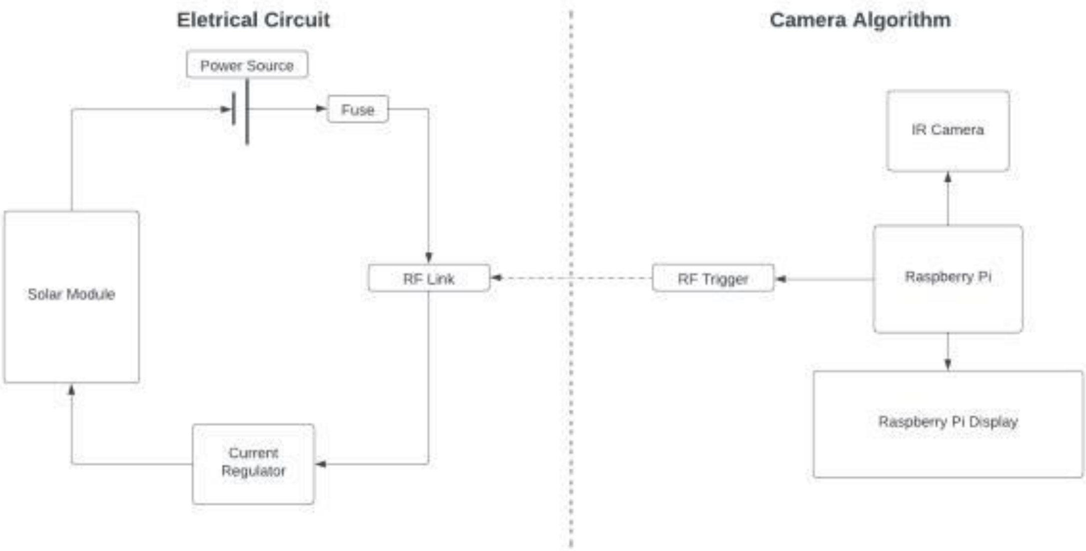

Figure 6: *Team Solar* 2023 PELMS System block diagram (from [2, pp. 2, Fig 2]).

The Camera Algorithm subsystem is used to take images of the PV module. The subsystem is controlled by the Raspberry Pi which is connected to:

- the IR camera which takes the images of the PV module,
- the Raspberry Pi Display which interfaces with the user, and
- the Radio Frequency (RF) Trigger which transmits the ON or OFF signal to the Electric Circuit subsystem via the RF Link to signal the Current Regulator to inject current into the PV module from the Power Source.

When a sequence of dark and bright images of the PV module have been taken the Raspberry Pi software will process the images to generate the higher quality EL image. The details of the Camera Subsystem components are given in Table 3.

Component	Description
Raspberry Pi	Controller for the Camera Algorithm subsystem. In the measurement mode it instructs the RF trigger to turn the current ON and the IR Camera to capture a *bright* image and then it instructs the RF trigger to turn the current OFF and the IR Camera to capture a *dark* image.
Raspberry Pi Display	The user interface to the Camera Algorithm subsystem. The user can select functions to start the measurement testing using the GUI (the GUI display is shown in Figure 7).
IR Camera	The IR Camera is a Raspberry Pi HQ 12.3 MP Camera. The camera will take images of the Solar Panel when actuated by the Raspberry Pi.
RF Trigger	The RF Trigger is a circuit that transmits an RF signal to the RF Link to signal the Current Regulator to inject or stop current flow into the Solar Panel. The RF Trigger is implemented using a Linx LR-433-EVAL Transmitter board.

Table 3: Camera Algorithm subsystem components.

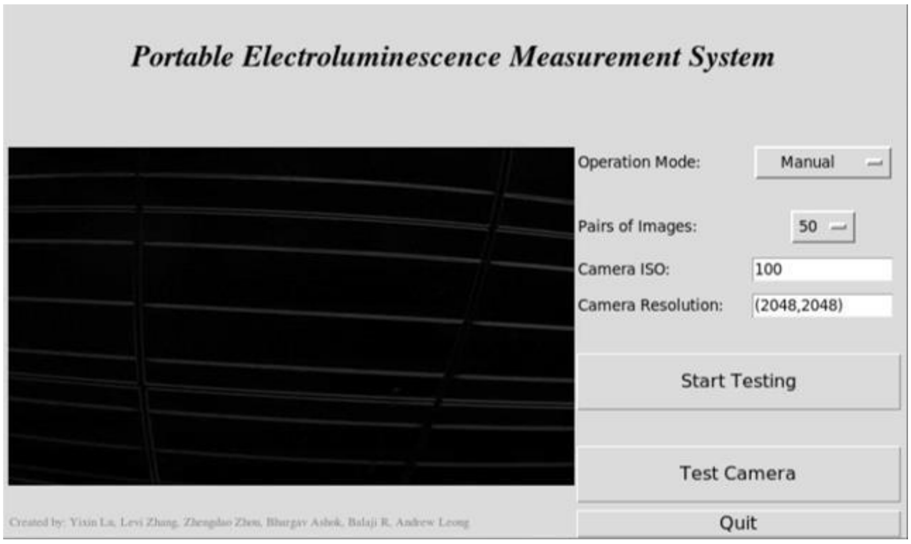

Figure 7: *Team Solar* 2023 PELMS GUI (from [2, pp. 5, Fig 7]).

The Electrical Circuit subsystem's primary function is to inject a regulated DC current into the Solar Panel (i.e. PV module under test) from the Power Source via the Current Regulator when the RF Link receives an ON signal from the RF trigger and to inject a current of 0 A when the RF Link receives an OFF signal. The Current Regulator and RF link were built to the same physical unit as shown in Figure 8. The details of the Camera Subsystem components are given in Table 4.

Component	Description
Solar Module	This was the PV module under test.
Power Source	The power source during the testing was a power supply system that would generate supply the current to inject into the PV Module.
Fuse	The fuse was in series with the Power Source and RF Link / Current Regulator to protect the RF Link / Current Regulator circuitry.
RF Link	The RF Link was a circuit that received an RF signal from the RF trigger that would signal the current regulator to inject current into the PV module. The RF Trigger is implemented using a Linx LR-433-EVAL Receiver board (seen on the left-side of the Current Regulator circuit shown in Figure 8).
Current Regulator	The current regulator was a circuit that would use the power source to inject a fixed current into the PV module.

Table 4: Electrical Circuit subsystem components.

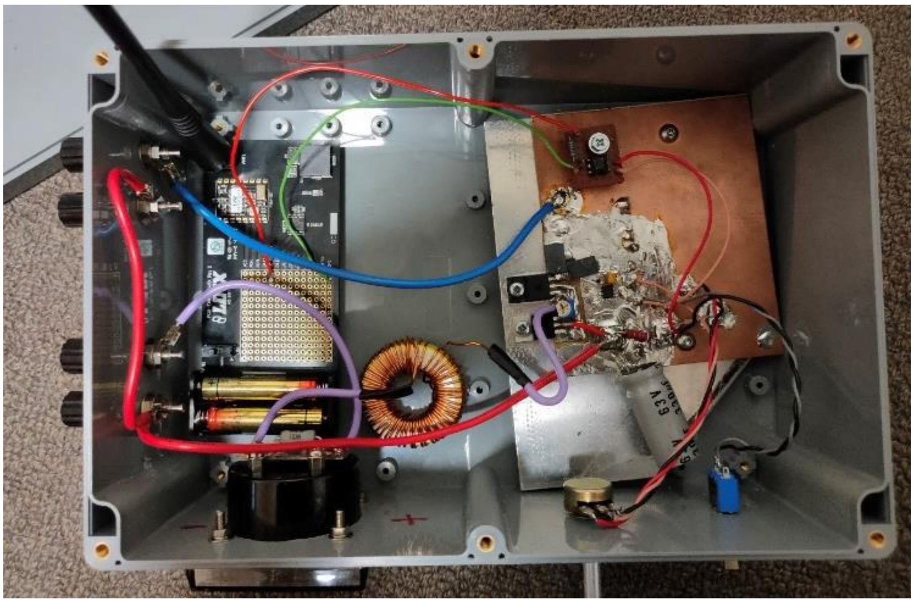

Figure 8: Team Solar 2023 PELMS Current Regulator Circuit (from [2, pp. 8, Fig F1])

The 2023 PELMS device described in this section was the initial prototype and had limitations that were discussed in the technical report and are documented in Table 1. This was the foundation work for the deliverables that were described in Table 2. The work completed to build the deliverables of this design project to as closely as possible use the same system architecture, terminology, and principles.

Chapter 3 Methodology

The approach I used for this project was to divide and allocate the work for the 5 deliverables to 3 engineering-related specialist fields that have tools and techniques to accomplish each deliverable. The specialist fields identified were:

- **Software Development** – the processes and practices of developing software. For this project I would develop Python software for the Raspberry Pi GUI that was used in the Camera Algorithm subsystem,
- **PCB Design** – the processes and practices for designing a PCB for an electronic circuit. For this project I would design a PCB for the Current Regulator circuit prototype with modifications to the power configuration and current rating, and
- **Electronic Enclosure Design** – the least common of the three fields but applicable for this project, this field relates to practices and processes for enclosing electronics projects so that the electronics is protected and still functions correctly. For this project I would design an enclosure for the Current Regulator circuit that protected the internal circuitry, enabled user access to Current Regulator controls and enable the Current Regulator to connect reliably to peripherals such as the Solar Panel (i.e. PV module) and the Power Source (i.e. E-bike battery).

The deliverables were mapped to their relevant subsystem and to the specialist field best suited to achieve that deliverable and then tabulated in Table 5. The work performed in each specialist field for each deliverable is discussed in the follow sections.

Deliverable	Subsystem	Specialist Field
Develop Python code to handle 12-bit resolution images and redevelop/update the GUI.	Camera Algorithm	Software Development
Design and construct an improved enclosure,	Electrical Circuit	Electronic Enclosure Design
Modify electrical circuitry to optimise power,	Electrical Circuit	PCB Design
Construction of a PCB board for the electrical circuit to replace the prototyping board, and	Electrical Circuit	PCB Design
Recalculate the component values of the electric circuit to handle loads above 10A.	Electrical Circuit	PCB Design

Table 5: Deliverables mapped to Subsystem and Field.

3.1 Software Development

Software development work was used to achieve deliverable 1 described in Table 2 which was:

- Develop Python code to handle 12-bit resolution images and redevelop/update the Graphical User Interface (GUI).

The work completed to achieve this deliverable, while broadly categorised as software development, also required a background understanding of computer vision principles. The software developed for this deliverable was programmed using Python and made extensive use of a computer vision called OpenCV. The background subjects that I had previously studied relating to software development and computer vision were:

- ENGN2219 Computing for Engineering
- ENGN4528 Computer Vision
- ENGN2218 Electronic Systems and Design

ENGN2219 was a general introduction to programming course primarily in MATLAB which is programming language that has many similarities to the Python programming language. ENGN4528 was a computer vision subject that covered relevant theory of image processing in particular how handle and manipulate red, green, and blue (RGB) images – which is the format of the images used in the PELMS systems. ENG2218 was an electronics subject which covered digital electronics this background and was useful when working with the general-purpose input/output (GPIO) pins to interface the Raspberry Pi with the RF Trigger. A lot of Python development support is found online in various websites on the internet. Websites such as *Stack Overflow* and other Python library pages will have sample code which can modified to suit a software developer's purpose.

The specific libraries that I used in the Python software development work were the following:

- *tkinter* – a Python library used for GUI development [5],
- *RPi.GPIO* – a Python library used controlling GPIO pins on a Raspberry Pi [6], the GPIO pins would be used to connect to the RF Link to raise the signal HIGH or LOW which in turn would turn the Current Regulator ON or OFF,
- *Picamera2* - a libcamera-based Python interface to Raspberry Pi cameras [7], this is the library that will have commands to interface and operate the Raspberry Pi Camera,

- *OpenCV* – is a Python implementation of the OpenCV library [8], which is an open-source computer vision library which contains image manipulation methods functions which are used to manipulate the image data from the camera.
- *NumPy* – is Python toolbox with mathematical and numerical computing tools [9], which is used for mathematically manipulating image data.

Using the previous Python code from the *Team Solar* 2023 PELMS GUI as a baseline I redeveloped the software with similar features. The primary difference between my code and the 2023 Python code was the image processing. The 2023 Python code produced 8-bit gray scale 480 x 480 portable network graphics (PNG) image file and my 2024 Python code developed used the original raw 12-bit Bayer image data to produce 16-bit RGB PNG image files with the pixel resolution proportional to the camera pixel resolution which was either full resolution 4056 x 3040 or quarter resolution 2028 x 1520. A discussion on how 12-bit Bayer image data is retrieved from the Raspberry Pi camera system and the algorithm I used is given in the following sections.

3.1.1 Raspberry Pi Camera System and Bayer image data

The raw 12-bit Bayer image data is the raw image data from the camera. The IR Camera used in the Camera Algorithm subsystem is a Raspberry Pi HQ (High Quality) Camera as shown in Figure 9 a) with the following features [10]:

- Sony IMX477R stacked, back-illuminated sensor,
- 12.3 megapixels = 4056 x 3040 P = 12330240 P ≈ 12.3 MP,
- Output: RAW12/10/8

This means the camera has 4056 x 3040 pixel light sensors and on top of the pixel light sensors is a colour filter array which will filter the light that hits the pixel light sensor. Many digital cameras (including the Raspberry Pi HQ) use a Bayer colour filter array which is shown in Figure 9 b) which is 2x2 pattern of 2 green filter and 1 red and 1 blue filter, so that each pixel captures the light intensity of green light or red light or blue light and stores this intensity as a digital 12-bit value. Therefore, one Bayer data image frame is composed of 4056 x 3040 12-bit values which are the green, red and blue light intensities organised in the Bayer colour filter array pattern.

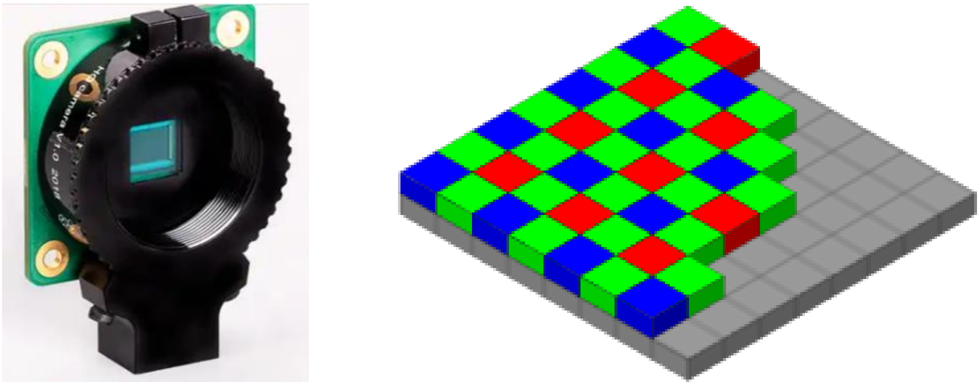

Figure 9: a) Raspberry Pi HQ Camera (from [10]) b) Bayer Colour filter array (from: [11])

The processing of the Bayer image data in the Raspberry Pi camera system is shown in Figure 10, where the sequence of events are as follows [12, p. 17]:

1. The camera module delivers 12-bit Bayer image frames through a flat ribbon cable to the CSI-2 Receiver and then transfers the incoming image into Memory.
2. The raw 12-bit Bayer image data is output to the Raw Stream, or passed to the Raspberry Pi Image Signal Processor (ISP),
3. If the raw 12-bit Bayer image frames are passed to the ISP, the ISP will output two image frames for every input image frame, the two output frame types are Main and Low resolution and are output to the Main Stream and Low resolution Stream respectively.

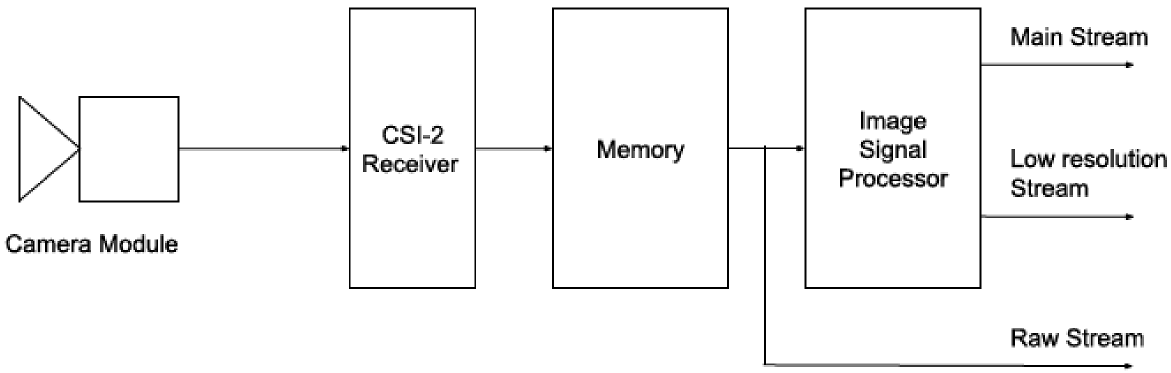

Figure 10: Raspberry Pi camera system (from [12, pp. 17, Fig 2]).

The 3 possible outputs from the camera system are:

- Main Stream – the primary quality image stream that can be in either RGB or YUV format and has typically 8-bit resolution and configurable frame size.
- Low resolution Stream – a viable quality image stream and is in YUV format and has typically 8-bit resolution and is a smaller frame size than main image, and
- Raw Stream – the original raw 12-bit Bayer image data output with fixed frame size relative to the raw frame size of the camera sensor.

The data stream I used for the PELMS system image processing in the software was the Raw Stream because this used the full camera resolution and full 12-bit intensity information. The software I developed used this 12-bit information but stored it in 16-bit unsigned integers for processing algorithm (later discussed) and RGB image data. The reason for using 16-bit unsigned integers was to be able to retain the 12-bit information and the software would only work in byte (or 8-bit) oriented data structures.

The Picamera2 Python software library contains the functions to interface with the Raspberry Pi Camera system including functions to capture images, capture video stream and configure camera parameters. The original Picamera library was used in the PELMS 2023 Python code but didn’t have the libraries to capture the 12-bit Bayer image data.

Picamera2 is the software library that contains Python software that communicates with the Raspberry Pi Camera. Raspberry Picamera2 has 3 configuration modes which are:

- *Still* – capture that uses 4056 x 3040 raw frame for still image capture from a main, lores or raw image stream,
- *Video* – capture that uses 4056 x 3040 raw frame for video image capture from a main, lores or raw image stream,
- *Preview* – capture that uses 2028 x 1520 raw frame for still or video image capture from a main, lores or raw image stream.

I used the *Still* image or *Preview* image configuration because either configuration could use the raw 12-bit data. In most cases I used the *Preview* image because it was the smaller size which meant there was less data to transfer and I experienced some software failures when trying to work with the higher rate *Still* image configurations.

3.1.2 12-bit image Processing Algorithm

The raw 12-bit Bayer image data needs to be converted to a 16-bit RGB image data format. There isn't a specific function to do this but there are generic steps that can be used to perform this function. A single 12-bit Bayer data image frame of at the camera resolution of the Raspberry Pi HQ camera will produce a 4056 x 3040 frame of image data where each pixel contains a 12-bit intensity value of the light that the colour filter passed through. The 16-bit RGB image data that we are interested in will be a 3 x 4056 x 3040 frame of image data comprised of:

- one 4056 x 3040 frame of 16-bit values of red image intensity values,
- one 4056 x 3040 frame of 16-bit values of green image intensity values, and
- one 4056 x 3040 frame of 16-bit values of blue image intensity values. (N.B. the 12-bits of information from the Bayer image is packed (or converted) to a 16-bit RGB value)

A sample of the Python code used to convert a raw 12-bit Bayer image format to 16-bit RGB format is given below:

```
# Capture the VL image:
# Get a raw Bayer frame with 12-bit values packed into 16-bits.
# Convert Bayer 12-bit array to RGB 16-bit and write to file by
# doing the following steps:
#   - Convert to 12-bit value to 16-bit by shifting 4 bits
#   - Convert from Bayer format to RGB format
img_VL_Bayer_12bit = picam2.capture_array("raw").view(np.uint16)
img_VL_Bayer_16bit = img_VL_Bayer_12bit * 16
img_VL_RGB_16bit =
cv2.cvtColor(img_VL_Bayer_16bit,cv2.COLOR_BAYER_RG2RGB)
```

A description of the lines in this sample code implementing the Bayer image conversion is as follows:

- the command: `picam2.capture_array("raw").view(np.uint16)`captures the 12-bit Bayer data and packs it into unsigned 16-bit integers.
- the command: `img_VL_Bayer_12bit * 16` is used to shift the 12-bit 4-bits so that the data scales from having values of 0 to 4096 (= 2^{12}), to having values in the range 0 to 65536 (=2^{16}).

- the command:
  `cv2.cvtColor(img_VL_Bayer_16bit,cv2.COLOR_BAYER_RG2RGB)` is an OpenCV command that will create an the red, green and blue arrays. The intermediate values for a given colour in where the light intensity was not directly capture and are interpolated from the other values.

Using this algorithm, we use the full 12-bit intensity information of each pixel in the Bayer image frame data to produce 3 red, green, and blue intensity values with 16-bit resolution. If we used the processed image data from the ISP – the ISP will have truncated the 12-bit data to 8-bits, and we would be working with only 8-bit resolution of the light intensities.

3.2 PCB Design

The PCB Design work was used to achieve deliverables 3, 4 and 5 described in Table 2 which were:

- Modification of the electrical circuitry to optimise power.
- Construction of the PCB for the electrical circuitry to replace the prototyping board.
- Recalculate the component values of the electric circuit to handle loads above 10 A.

The PCB design goal was to develop a PCB that satisfied all 3 deliverables and would involve building a PCB of the original 2023 Current Regulator circuit that has the RF link on boarded (i.e. so that board was configured to supply power to all circuit elements from the Power Source) and has the circuit rated to 20 A rather 10 A. The background subjects that I had studied relating to PCB design were:

- ENGN2218 Electronic Systems and Design
- ENGN4625 Power Systems

ENGN2218 covered analog and digital electronic systems and how electronic circuits worked which provided the fundamentals of how to design a PCB. ENGN4625 was a Power Systems course which provided me with fundamental concepts in power system design which was useful in understanding concepts such as:

- how inductors and capacitors are used to reduce the effects of transient and ripple current,

- the temperature and power impact of current on components and
- practical experience using LT Spice software which could be used for modelling the behaviour of circuits.

PCB design is a hardware area, and so mistakes can be more costly than when mistakes are made in software development. If mistakes are made in hardware such as choosing incorrect component value the failure can result in damaged components that needs to be replaced, whereas in software the code simply will fail and can be corrected as needed. Also, there can be delays in finding parts suitable that may need to be ordered.

I was fortunate to have the assistance of an electronics engineer, Mr Chris Jones, at the various stages of my PCB design work. Chris assisted greatly with proof checking my PCB designs i.e. that component value calculations were suitable and providing advice on industry practices such as tolerance for component values.

The primary tool for the PCB design work used was a free software suite for electronic design automation called KiCad. The basic KiCad process to develop a PCB is:

- edit a circuit schematic as shown in Figure 11 a),
- generate a PCB footprint based on the schematic as shown in Figure 11 b), and
- generate Gerber files from the PCB design which can be passed on to a manufacturer.

An example of an uploaded KiCad Gerber files to a manufacturer website is shown in Figure 12. There were 3 PCB manufacturers that were considered for PCB manufacturing they were:

- OSH Park (US-based),
- PCBWay (China), and
- JLCPCB (China).

In this project we settled on JLCPCB as the preferred manufacturer who supplied 5 PCBs for $55.00 AUD with a build and delivery time of approximately 1 week.

My approach to develop the PCB was to design 3 revisions of the PCB and to develop features with each revision. The reason for doing this was that I was relatively inexperienced with PCB design and didn’t want to implement too much complexity too quickly.

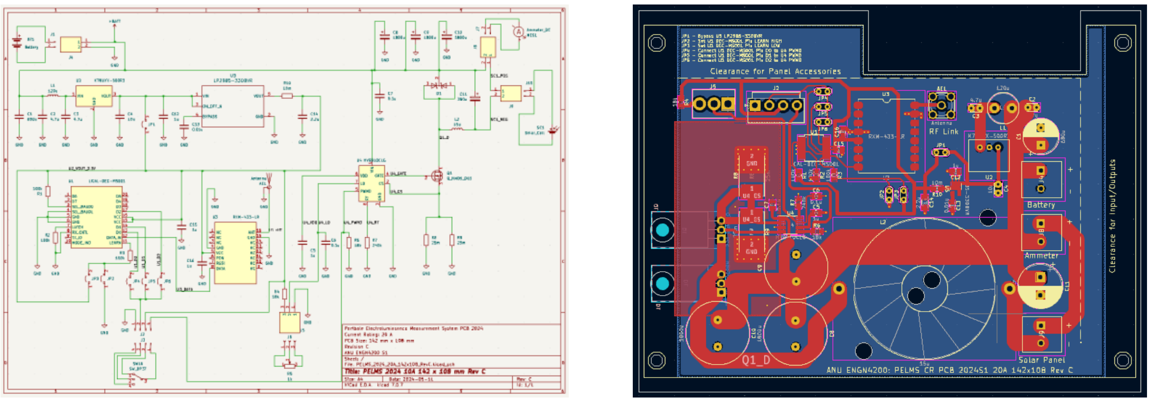

Figure 11: KiCad PCB design process. a) KiCad schematic b) KiCad PCB footprint.

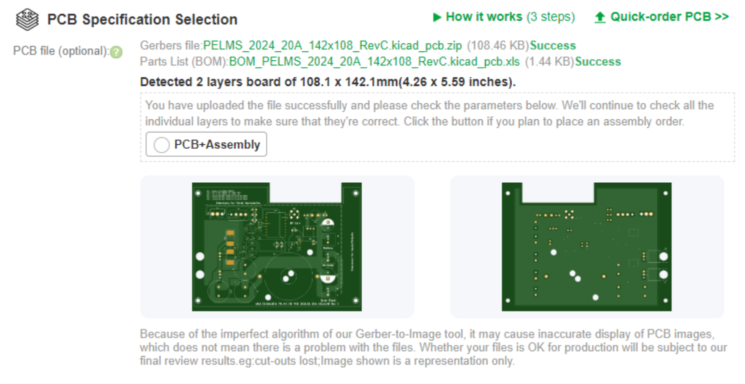

Figure 12: Upload PCB Gerber files to a manufacture website (i.e. PCBWay).

The first revision developed was revision A and this was developed as KiCad project: *PELMS 2024 10A 142x110 Rev A*. The goal of the revision A PCB was to implement the Current Regulator circuit from the 2023 design (shown in Figure 13). The original circuit design was documented in the handover [3] and technical report [2], however there were some inconsistencies with component values so I determined what would be the correct values for a 10 A rated circuit. The RF link was still off-board, and the circuit was developed on a smaller footprint of 142 mm x 110 mm. In this circuit design heat considerations were first implemented, the primary being the modification of track widths, which I could determine reasonable value from various vendor websites so lighter current paths were thinner than heavy current paths, and the placement of transistor Q1 and diode D1 underneath the board for attachment to an Aluminium plate that could draw heat away from the circuit which was a suggestion from Mr Chris Jones. The main track that would be carrying 20 A (i.e. if the board was used to capacity) had a track width of 10 mm, and this value was based on the PCBWay Track calculator that determined a track width of approximately 10 mm would result in a temperature rise of 20 °C – this was a conservative figure because as we would not be running the current at 20 A on a 100 % duty cycle but rather a 50% duty cycle so this track width would be suitable. The PCB Revision A schematic and footprint are shown in Figures 20 and 21 of Appendix A.

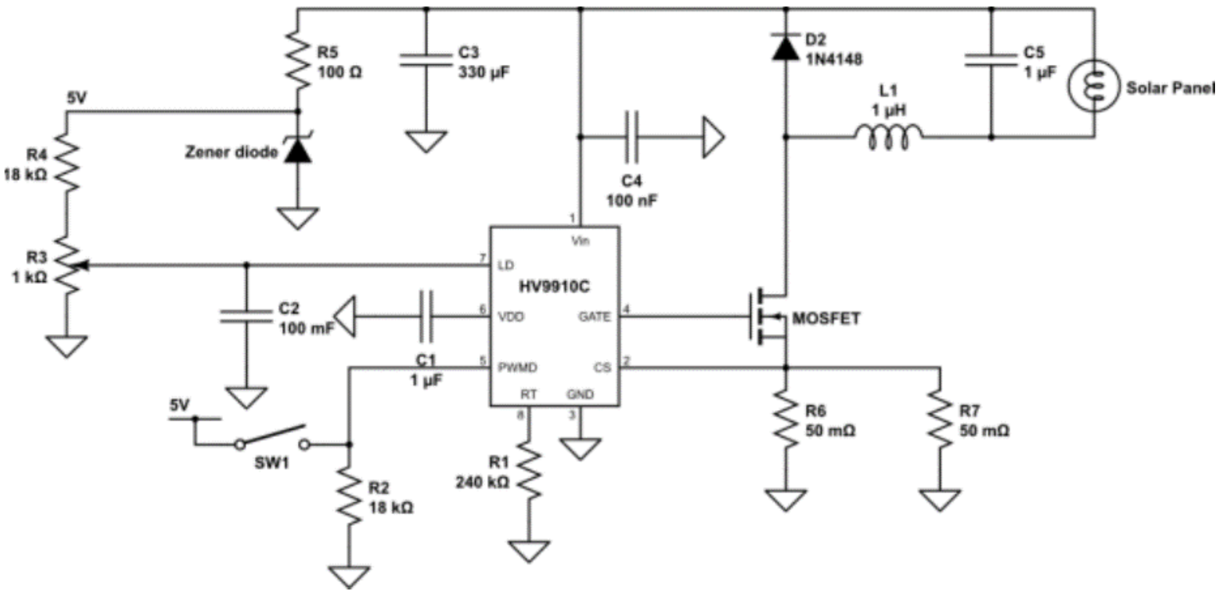

Figure 13: The 2023 Current Regulator circuit schematic.

The second revision developed was revision B and developed as KiCad project: *PELMS 2024 10A 142x110 Rev B*. Revision A had implemented a PCB of the original circuit design and revision B would be on-boarding the RF Link, the design would still be rated to 10 A so in this revision extra components be added. The design was built on the same footprint as Revision A of 142 mm x 110 mm, but the following components were added to the design:

- RF receiver chip (RXM-433-LR) – the circuit that received the 433 MHz signal as part of the RF Link,
- decoder chip (LICAL-DEC-MS001) – the chip to decode the output from the RF receiver to select a data line also part of the RF Link,
- DC/DC converter chip (K78U03-500R3) – this was to convert the power source voltage from 52V to 3.3 V to power the RF receiver and decoder chips, and
- an antenna mount this was the mount for the antenna to provide the RF signal to the RF receiver.

There was a significant increase in the number and complexity of signalling tracks which needed to be routed through the PCB. In the completed version of this revision all components in the Current Regulator were now powered off the same power source (i.e., the 52 V e-bike battery). The PCB Revision B schematic and footprint are shown in Figures 22 and 23 of Appendix A.

The third and final revision was revision C and developed as KiCad project: *PELMS 2024 10A 142x110 Rev C*. This was the revision where the PCB design would upgrade the current rating from 10 A to 20 A. In this revision some of the component values needed to be recalculated or upgraded so that all components could handle a 20 A current, additionally the track widths needed to be checked that they could handle 20 A. The main updates were the values of the current sensing resistors and the upgrade of the inductor.

One significant problem with revision C was finding a suitable inductor for the 20 A circuit. To determine the suitable inductor, I used LT Spice software with modified simulations developed by Mr Chris Jones and the *Team Solar* to observe the ripple current for different values of inductors. In the previous revision the inductor had been a 10 A rated 380 uH inductor which at 10 A in the LT Spice simulation resulted in current ripple of 0.06 %, so the output current through the Solar Panel would be in the range of 9.994 – 10.006 A. The LT Spice simulations are shown in Figure 14. The only 20 A rated inductor that would fit the PCB was a 15 uH inductor and using an LT Spice the simulated ripple current was 1.26% so that output current through the Solar Panel would be in the range of 19.748 – 20.252 A. The LT Spice simulations are shown in Figure 15. Despite the current ripple performance being worse I thought this still acceptable, so I used the 15 uH inductor in Revision C. The PCB Revision B schematic and footprint are shown in Figures 24 and 25 of Appendix A.

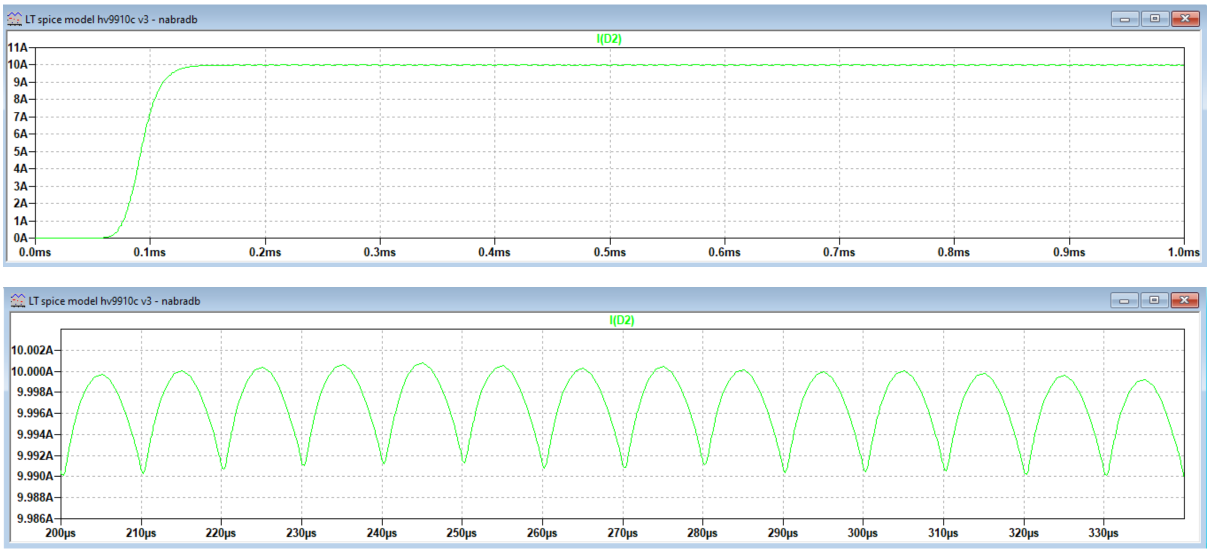

Figure 14: LT Spice simulation for 380 uH inductor used in 10 A circuit.

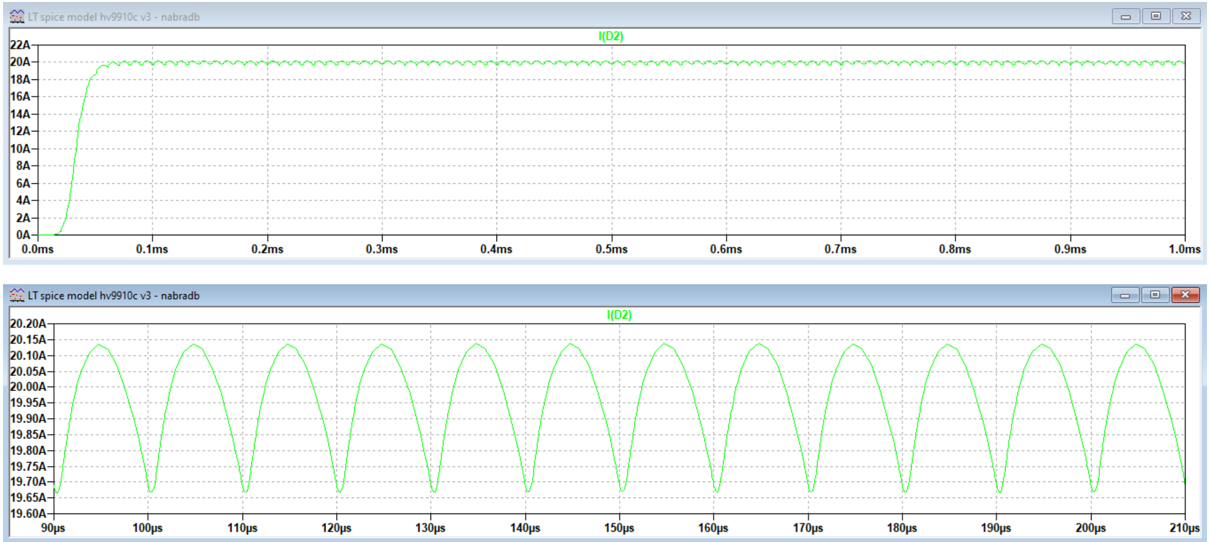

Figure 15: LT Spice simulation for 15 uH inductor used in 20 A circuit.

3.3 Electronic Enclosure Design

The Electronic Enclosure Design work in this project was used to achieve deliverable 2 described in Table 2 which was:

- Design and construction of an improved enclosure.

The enclosure was for the Circuit Regulator of the Electrical Circuit subsystem. The PCB of the Circuit Regulator (from the 2024 design) needed to be fit into a 171 mm x 121 mm enclosure. The enclosure itself was already fabricated but needed modifications to enable panel mounting for controls and secure ingress and egress of cables to the PCB.

Like PCB design, Electronic Enclosure design is a hardware discipline and problematic because mistakes can cause damage and be costly e.g. an incorrectly placed hole can't be undone. Related studies I had completed in this field were taken at Canberra Institute of Technology (CIT):

- CIT ANUQ Electronics Workshop Practices

This subject had been a hands-on workshop tool-based subject, during which I practised soldering, drilling holes for cable glands and cutting holes to fit panel mount items such as displays. As this was the area that I had little background knowledge. I relied heavily on internet and vendor research.

The selected enclosure was a 171 mm x 121 mm x 55 mm and is shown in Figure 16. This prefabricated enclosure is rated to IP65, an ingress protection code, where:

- the first digit is solid particle protection and a value of 6 is *Dust-tight – No ingress of dust; complete protection against contact,* and
- the second digit is liquid ingress protection and a value of 5 is *Water jets; Water projected by a nozzle (6.3 mm (0.25 in)) against enclosure from any direction shall have no harmful effects*.

Figure 16: Sealed Poly carbonate 171 x 121 x 55 mm enclosure with a clear lid (from [13]).

The approach for the enclosure design was to:

- identify ingress/egress points and panel mount controls,
- generate some initial solutions ingress/egress points and panel mount controls,
- build a mock-up to identify and test the location of fittings.

The ingress/egress and peripherals identified, and the initial solutions are given in the description column shown in Table 6.

Component	Description
Power Source (i.e. Battery)	This required a 20 A connection to the e-bike battery with fuse in place. The Power Source (a 52 V E-bike battery) used 30 PP Anderson connectors. A special housing was required to house the Anderson connectors and a suitable housing was identified on eBay.
Solar Panel (i.e. PV Module)	This required a 20 A connection to the PV Module. The solution here was to uses 35 A rated alligator clips connected via 12 AWG 25 A rated wire egressing via cable gland.
Current Level Controller	Used a 1 kΩ potentiometer with a knob.
Mode Switch	Used a rotary switch – that could mechanically switch various positions.

Table 6: Ingress/Egress points and panel mount controls for electronic enclosure.

The mock-up (see Figure 17) was built using a cardboard PCB to get an approximation for the PCB and holes were placed with respect to the location of the various ingress/egress points. From these approximations I was able to determine and record the measurements for the hole locations which are shown in Appendix B Figure 26 and 27.

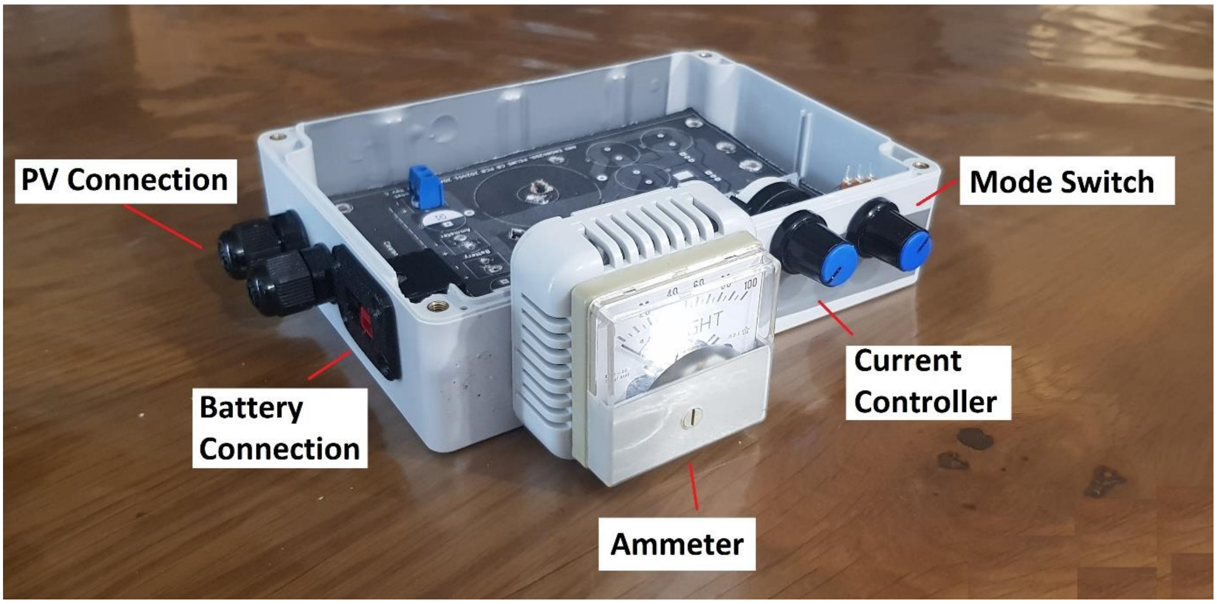

Figure 17: PCB Cardboard mock-up used to estimate enclosure measurements.

Chapter 4 Results and Analysis

The project outcomes are documented in the three engineering-related fields and are discussed in terms of results and analysis. In the results section the outputs from the project work are discussed in terms of products generated and the capabilities and limitations of the products. In the analysis section the products are evaluated against the 2023 PELMS system and the limitations of the products are discussed.

4.1 Software Development

4.1.1 Software Development Results

The goal of the software development work was used to achieve deliverable 1 described in Table 2 which was:

- Develop Python code to handle 12-bit resolution images and redevelop/update the Graphical User Interface (GUI).

The Python code was for the software that would run on the Raspberry Pi in the Camera Algorithm subsystem. The code would provide a GUI and process the image data. In this project I developed two python GUIs based on the PELMS 2023 GUI. The two programs were:

- *pelms-2024-s1_dev_v2024-05-06a.py*
- *pelms-2024-s1-thin_dev_v2024-06-03.py*

The reason for developing two programs was because of some of the difficulties I was having working with the Picamera2 libraries. The GUI *pelms-2024-s1_dev_v2024-05-06a.py* was the first GUI developed and had capabilities that extended the PELMS 2023 GUI which included:

- *Camera Viewer* – this function was a preview mode that would be used during camera setup, effective this was video stream,
- *File Transfer* – this function enabled files to be transferred from the PELMS device to other area in the operating filesystem on the Raspberry Pi, and
- *EL Measure* – this was the EL measurement function which would generate the EL images of the PV module under test.

The main menu GUI for *pelms-2024-s1_dev_v2024-05-06a.py* is given in Figure 18. The code *pelms-2024-s1_dev_v2024-05-06a.py* had stability issues and intermittent failures during interaction with the Raspberry Pi camera when using the Picamera2 library functions. The program would execute correctly in some instances and then fail under similar conditions.

To try to resolve whether the instability issues were with the actual code itself or more systematic issues (e.g. an operating system update, library incompatibility etc.) I implemented the program *qtdemo.py* script from the Picamera2 manual [12, pp. 56,56], this program despite using the PyQt library used many of the functionality in terms of accessing the camera which were also used in my applications. This program had the same problems as my original code, *pelms-2024-s1_dev_v2024-05-06a.py*. Given this result, I felt that there were some other, more systematic issues that were causing the instability, and the problem was not simply poor programming.

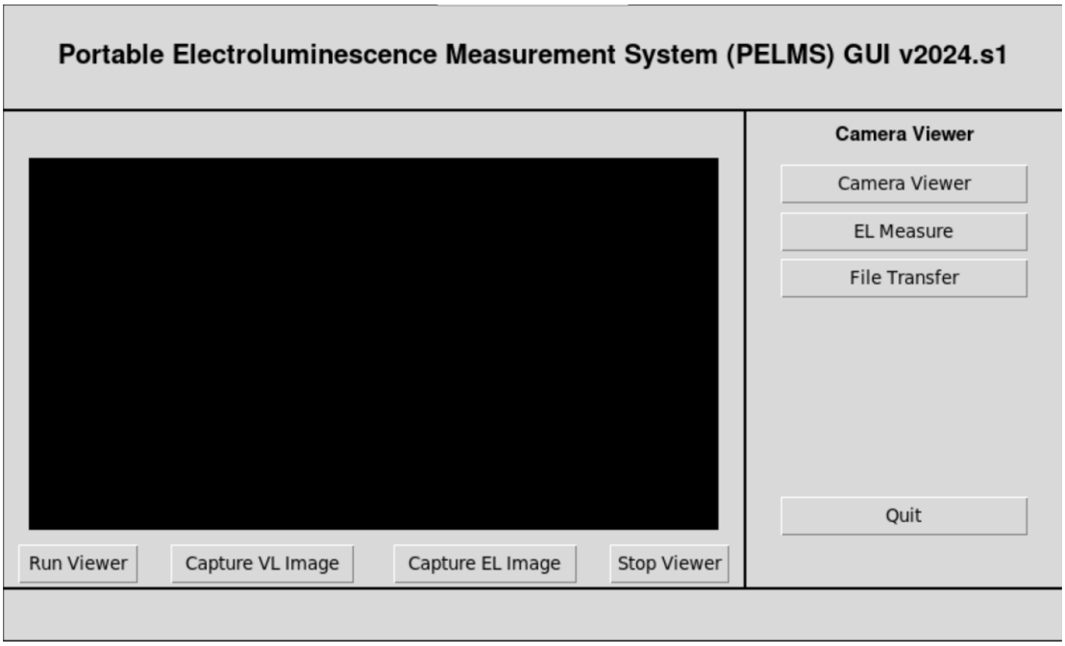

Figure 18: PELMS GUI v2024.s1 *pelms-2024-s1_dev_v2024-05-06a.py*.

The Picamera2 functions where I had the least errors and failure was when working with the "raw" stream data. This was fortunate in that the raw 12-bit data was what was to be used for image processing. A possible reason for the raw data not being an issue is that the raw stream bypasses the Image Signal Processor, so if stability issues were related to communications with the Image Signal Processor of the Raspberry Pi camera they could be avoided or reduced by working with the "raw" stream data.

The reduced functionality program written to work with this was *pelms-2024-s1-thin_dev_v2024-06-03.py*. The was a "thin" PELMS GUI that had simpler and less functionality than the previous code and the GUI is shown in Figure 19. This had a similar interface to the PELMS 2023 and simpler functionality. The user can select:

- the Camera Resolution to be either 2028x1520 or 4056x3040,
- the number of image pairs,
- whether to keep the intermediate images, and
- the Test Mode which is one of the following:
  - PASSIVE means no image is produced – and can be used as a "dry run" to check the image and adjust the camera placement,
  - ACTIVE means produce an EL image in 16-bit RGB and 16-bit GRAY scale and writes the files to the Desktop, and
  - ACTIVE-T is the same as ACTIVE but produces an EL image that implements a threshold – this is primarily available because it can produce a threshold image using fewer image pairs and in lower light conditions and is useful to check the current is flowing through the panel.

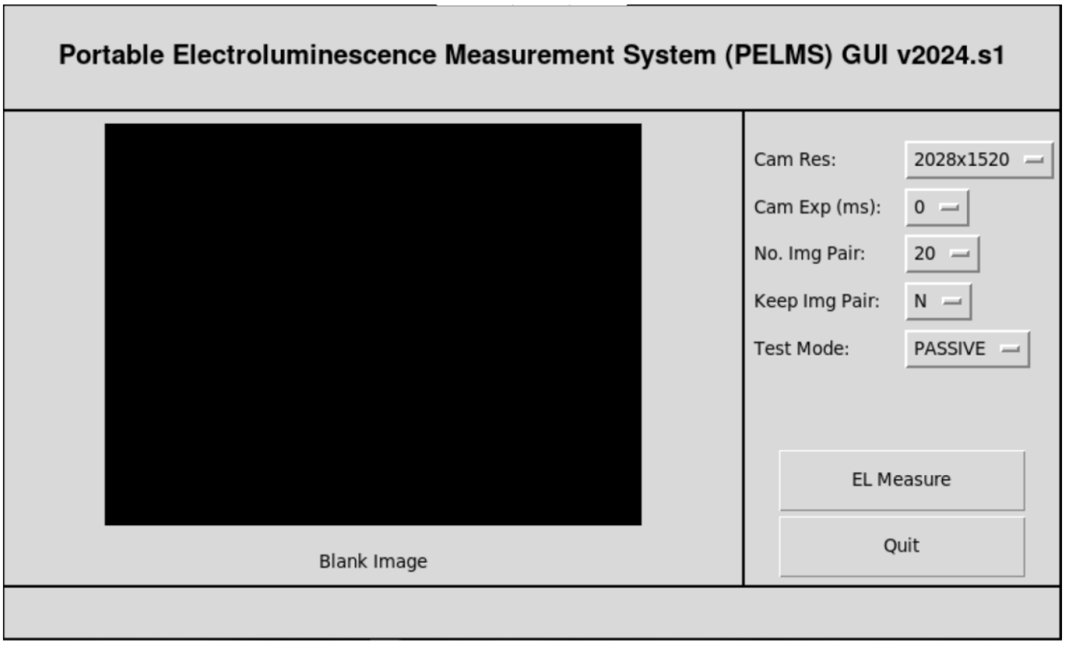

Figure 19: PELMS GUI v2024.s1 *pelms-2024-s1-thin_dev_v2024-06-03.py*.

4.1.2 Software Development Analysis

The imaging functionality of the *Team Solar's* 2023 PELMS code was compared with the 2024 PELMS code *pelms-2024-s1-thin_dev_v2024-06-03.py*. In both cases the software was run on the Raspberry Pi hardware and used the PELMS 2023 Current Regulator (as at this time I have not completed assembly of the 2024 Current Regulator).

The comparison test was to generate images of a solar panel adjusting 3 parameters, were:

- the number of PV cells used for EL imaging – in the tests I used 1, 2 and 3 PV cells,
- the currents injected into the PV Cell – in the tests I used 7A, 8A and 9A, and
- the number of image pairs used to generate EL images – in the tests I used 10, 20 and 50 pairs.

The image results for the PELMS 2023 imaging software are given in Appendix C (Figures 35, 37 and 39) and the image results for the PELMS 2024 imaging software are given in Appendix D (Figures 42, 44 and 46). When the image was compared the primary advantages of the 2024 images over the 2023 images were:

- higher resolution – the 2023 EL images were grayscale 480 x 480 png files, while the 2024 EL images were 2028x1520 16-bit grayscale and RGB images, and
- initial image quality – the 2023 images often needed software adjustment to view the detail of the near IR light while in the 2024 images the near IR i.e. EL can be seen without software adjustment.

There were limitations of the software, which were:

- limited resolution – the image resolution of 2028x1520 worked fine in most cases but the resolution of 4056x3040 would often fail – this may be a data transfer issue.
- exposure time had no effect - the Picamera2 exposure time value could be set (i.e. in software I set the value and then checked it) but didn't seem to have any effect and I would have to research further as to why this happened.

Overall, the software does improve the image quality, but the software needs to be more stable, and the exposure time further researched.

4.2 PCB Design

4.2.1 PCB Design Results

The goal of the PCB design and development work was used to achieve deliverables 3, 4 and 5 described in Table 2 which were:

- Modification of the electrical circuitry to optimise the power,
- Construction of the PCB for the electrical circuit to replace the prototyping board, and
- Recalculation of the component values of the electric circuit to handle loads above 10A.

In this project I designed 3 revisions of Electrical Circuit system Circuit Regulator PCB in KiCad with the following features:

- the RF component was onboarded on to the PCB Board – so the RF Link was no longer powered independently,
- the circuit was implemented in a PCB as opposed to a prototyping board which meant that the tracks for the circuit were in-built, the components were seated correctly, and the PCB can be easily replicated, and
- the component values for the circuit were rated to 20 A.

The PCB design was passed on to a PCB manufacturer JLC PCB and the physical PCB was built. The final PCB board is shown in Figure 20. The components for the PCB have been ordered and the PCB is currently being assembled.

4.2.2 PCB Design Analysis

At this stage the PCB is still in an assembly state, so I haven't been able to perform analysis as to whether the PCB is functional or not. The test to check the final functionality of the board will be to setup the Current Regulator and attach the power and perform the following tests:

- Set the Current Regulator in a Current ON mode and check the output current with a multi-meter can range for 0 to 20 A. Not also in this test for how the heat is handled.
- Set the Current Regulator to an RF Trigger mode and the output is triggered ON/OFF with the software output.

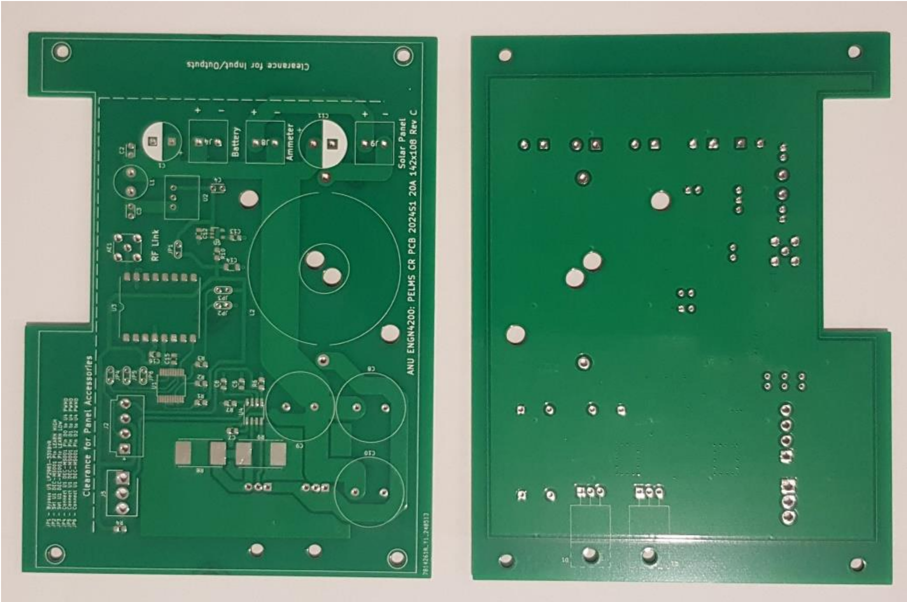

Figure 20: Current Regulator PCB Board

4.3 Electronics Enclosure Design

4.3.1 Electronics Enclosure Design Results

The goal of the software development work was used to achieve deliverable 2 described in Table 2 which was:

- Design and construction of an improved enclosure.

In this project I designed the electronics enclosure based on the porting the Current Regulator circuit from the original prototype in a Sealed ABS Enclosure of dimensions 240 x 160 x 90 mm to the Sealed Poly carbonate 171 x 12 x 55 mm enclosure shown in Figure 16.

The enclosure was designed based on the ingress/egress points and panel mount controls for the enclosure given in Table 6. The profile of the enclosure was documented in Appendix B, and I constructed the enclosure by drilling the appropriate holes and mounting the components such as cable glands, potentiometer, and control switches.

The electronic enclosure was fitted with peripherals and the PCB and is shown in Figure 21.

Figure 21: Enclosure fitted with peripheral and PCB.

4.3.2 Electronics Enclosure Design Analysis

The 2023 and 2024 Circuit Regulator enclosures are compared and shown in Figures 22, 23 and 24. The improvements of the 2024 Current Regulator enclosure over the 2023 Current regulator enclosure that can be viewed are:

- physical footprint reduction (see Figures 22 to 24) – the original 2023 prototype was fitted into a 240 x 160 mm base footprint and the new 2024 enclosure has a base footprint of 171 x 121 mm,
- secure fitting of the PCB and component (see Figure 22) – the original 2023 prototype didn’t secure the base to the enclosure and new 2024 enclosure has the PCB attached to the enclosure base by 4 x M3 screws.
- secure egress/ingress attachments – the original 2023 prototype simply uses banana terminals to connect to the Power Source and Solar Panel and the new 2024 enclosure uses and Anderson connection to the Power Source and alligator clips to the Solar Panel.

There was a limitation to the enclosure design which was the height of enclosure. Originally the enclosure design had been for an enclosure of 171 x 121 x 55 mm. The capacitors on the PCB have a heigh of 35 mm which makes them to tall to fit in the 55 mm (this is because the PCB is seated off the approximately 22 mm off the base. The resolution for this was to use a higher lid which means the 2024 enclosure will have dimension of 171 x 121 x 80 mm.

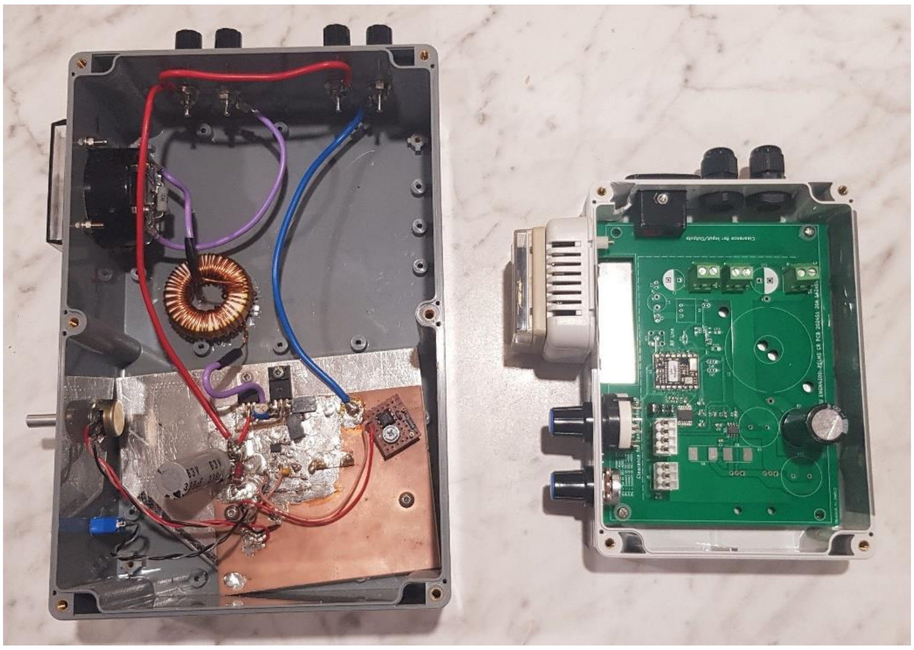

Figure 22: 2023 (left) and 2024 (right) Current Regulator Enclosure top view comparison.

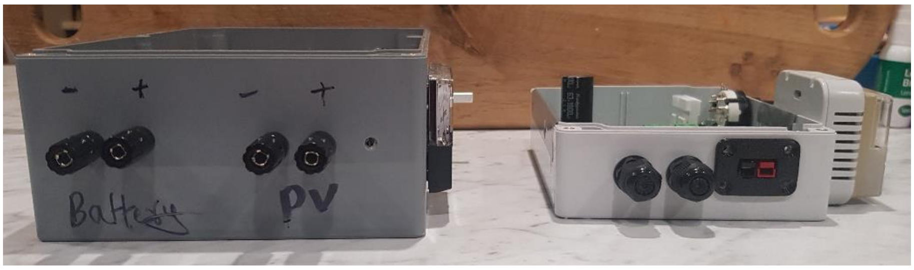

Figure 23: 2023 (left) and 2024 (right) Current Regulator Enclosure front view comparison.

Figure 24: 2023 (left) and 2024 (right) Current Regulator Enclosure side view comparison.

Chapter 5 Conclusions and outlook

This project developed progress against the 5 deliverables mentioned in Table 2. Only deliverables 1 and 2 could be considered successful with limitations. Deliverables 3, 4 and 5 were dependent on the PCB board that was built to work which at this stage has not been tested.

The software development work for deliverable 1 resulted in the development of the PELMS GUI software that generated 12-bit resolution EL images using the raw 12-bit camera data. The images provided higher resolution images and greater initial image quality than the 2023 data, but the software was really limited to images of 2028 x 1520 and needs to be modified to improve the software stability.

The PCB development work for deliverables 3, 4 and 5 did result in the development of PCB board that:

- implemented a PCB of the 2023 current regulator prototype circuit,
- modified the RF Link components to use the same power source instead of independent batteries which simplified the power configuration, and
- used components values that rated the circuit to 20 A.

At this stage the PCB is still being assembled and has not yet been tested and so I will need to wait to see whether these deliverables were successful.

The Electronics Enclosure design development work for deliverable 2 did result in development of an improved enclosure that:

- reduced the base footprint from 240 x 160 mm to 171 x 121 mm,
- fitted the PCB and internal components in a more secure manner, and
- provided more secure ingress/egress cables to the Power Source and Solar Panel.

The primary limitation for the enclosure design was the height of the capacitors in the PCB meant that the enclosure needed a higher lid which change the enclosure height from 55 mm to 80 mm.

The things that work well during the project were:

- the KiCad software process for PCB development – this was extremely streamline in that changes to the PCB could be performed quite quickly and the process to develop the PCB was straightforward,
- the Picamera2 library when it worked was also relatively straightforward to use, and
- the enclosure design was also a reasonably straightforward process in that many off the shelf components can be used to build an enclosure – the only real bespoke enclosure piece required was mounting panel for the Anderson power connectors which was ordered off eBay.

The things I found difficult during the project were:

- component adjustments could have a ripple effect i.e. a change in value in one component may result in other components to be changed (e.g. with higher rating) and this could also affect the size of components which affected the enclosure design,
- sourcing components could at times be challenging in that most components needed to be ordered a head of time and if I made a mistake in the order then there was a delay. Most of the components needed to be sourced online – some components could be source from Jaycar but typically these more expensive.

I think that this project still has lots of potential future work. I think that there may still be some improvements that could be made in the hardware space in terms of improving the enclosure (i.e. moving the controls) and modifying some of the part placement but I think if the 20A Current Regulator works then most of the improvements will be for the user interface. An area in this project where I think significant work could be done is in the actual computer vison / image processing and software. I felt that the GUI I developed while functional was still simplistic – I think that a more sophisticated programmer could develop a much more powerful GUI that would put more image processing functionality on the Raspberry Pi.

References

[1] U. Jahn and M. Herz, "Report IEA-PVPS T13-10:2018 Review on Infrared and Electroluminescence Imaging for PV Field Applications," International Energy Agency Photovoltaic Power Systems Program, 2018.

[2] A. Leong, B. Radhakrishnan, B. Ashok, L. Zhang, Y. Lu and Z. Zhou, "Development of a Portable Electroluminescence Measurement System for Photovoltaic Modules," Australian National University, Canberra, 2023.

[3] A. Leong, B. Radhakrishnan, B. Ashok, L. Zhang, Y. Lu and Z. Zhou, *Development of a Portable Electroluminescence Measure System for Photovoltaic Modules - Handover Document,* Canberra: Australian National University, 2023.

[4] M. Owen-Bellini, D. B. Sulas-Kern, G. Perrin, H. North, S. Spataru and P. Hacke, "Method for In Situ Electroluminescence Imaging of Photovoltaic Modules Under Varying Environmental Conditions," *IEEE Journal of Photovoltaics,* vol. 10, no. 5, pp. 1254 - 1261, 2020.

[5] Python Software Foundation, “tkinter — Python interface to Tcl/Tk,” Python Software Foundation, 21 May 2024. [Online]. Available: https://docs.python.org/3/library/tkinter.html. [Accessed 22 May 2024].

[6] Python Software Foundation, "RPi.GPIO 0.7.1," Python Software Foundation, 7 February 2022. [Online]. Available: https://pypi.org/project/RPi.GPIO/. [Accessed 22 May 2024].

[7] Python Software Foundation, "picamera2 0.3.18," Python Software Foundation, 18 April 2024. [Online]. Available: https://pypi.org/project/picamera2/. [Accessed 22 May 2024].

[8] Python Software Foundation, "opencv-python 4.9.0.80," Python Software Foundation, 1 January 2024. [Online]. Available: https://pypi.org/project/opencv-python/. [Accessed 22 May 2024].

[9] NumPy team, "NumPy," NumPy team, 16 September 2023. [Online]. Available: https://numpy.org/. [Accessed 23 May 2024].

[10] Raspberry Pi, "Raspberry Pi High Quality Camera," Raspberry Pi, 22 May 2024. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/. [Accessed 22 May 2024].

[11] Wikipedia contributers, "Color filter array," Wikipedia, The Free Encyclopedia, 2024 May 22. [Online]. Available: https://en.wikipedia.org/wiki/Color_filter_array. [Accessed 2024 May 28].

[12] Raspberry Pi Ltd, "The Picamera2 Library," 29 April 2024. [Online]. Available: https://datasheets.raspberrypi.com/camera/picamera2-manual.pdf. [Accessed 22 May 2024].

[13] Jaycar Electronics, "Jaycar Electronics - Sealed Polycarbonate Enclosure 171 x 121 x 55 - Clear Lid," Jaycar Electronics, 15 May 2024. [Online]. Available: https://www.jaycar.com.au/sealed-polycarbonate-enclosure-171-x-121-x-55-clear-lid/p/HB6248. [Accessed 15 May 2024].

[14] "Picamera," [Online]. Available: https://picamera.readthedocs.io/en/release-1.13/. [Accessed 7 November 2023].

[15] Raspberry Pi & Raspberry Pi Foundation, "Picamera2," [Online]. Available: https://pypi.org/project/picamera2/. [Accessed 7 November 2023].

[16] Various, "Stack Overflow," Stack Overflow, 2024. [Online]. Available: https://stackoverflow.com/. [Accessed 2024 May 23].

Appendix A: PCB Design KiCad Schematics and Footprints

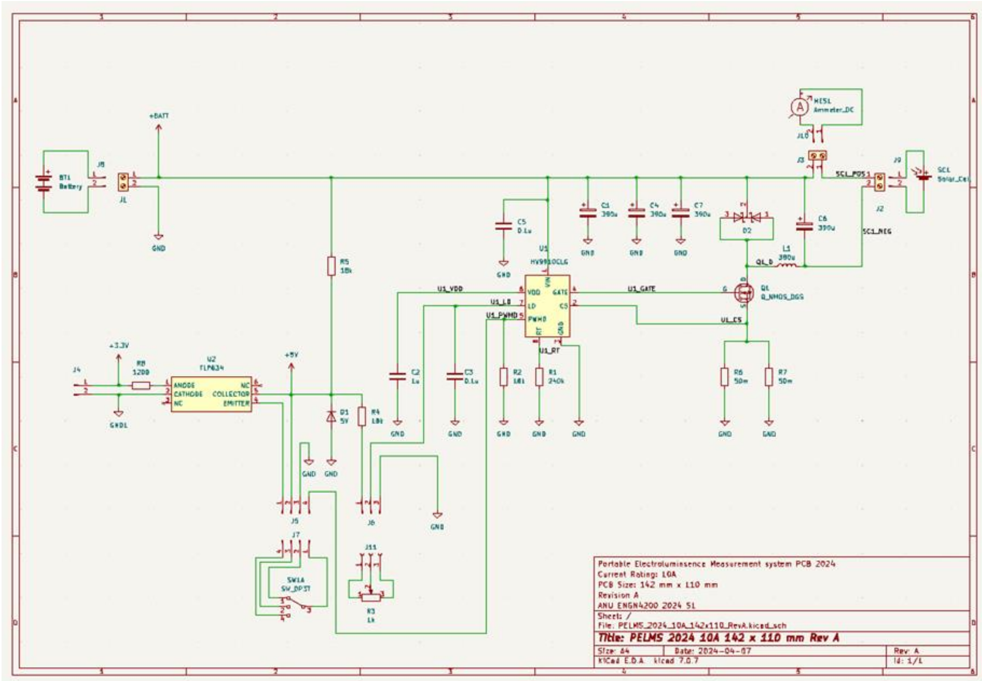

Figure 25: PELMS 2024 10A 142x110 Rev A – Schematic

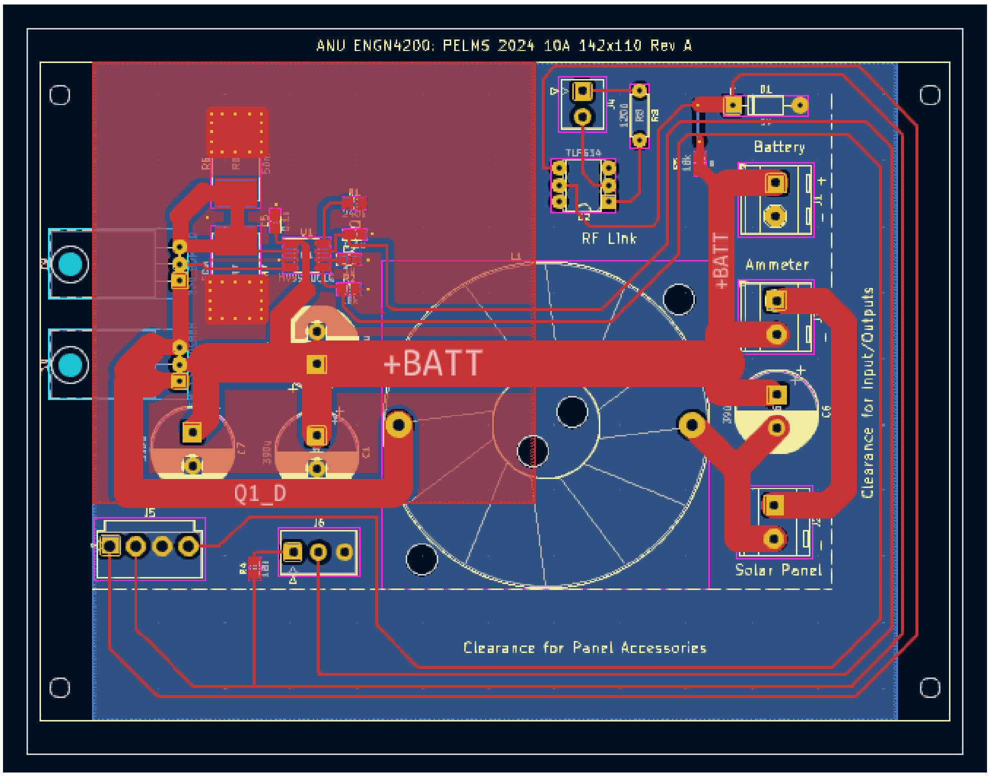

Figure 26: PELMS 2024 10A 142x110 Rev A – PCB Footprint

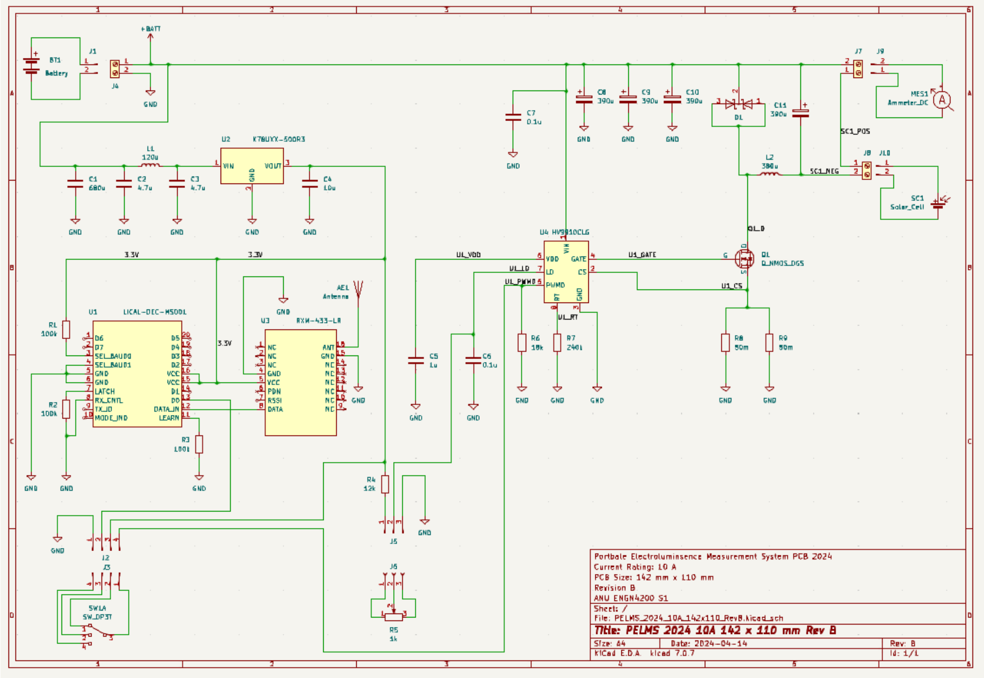

Figure 27: PELMS 2024 10A 142x110 Rev B – Schematic

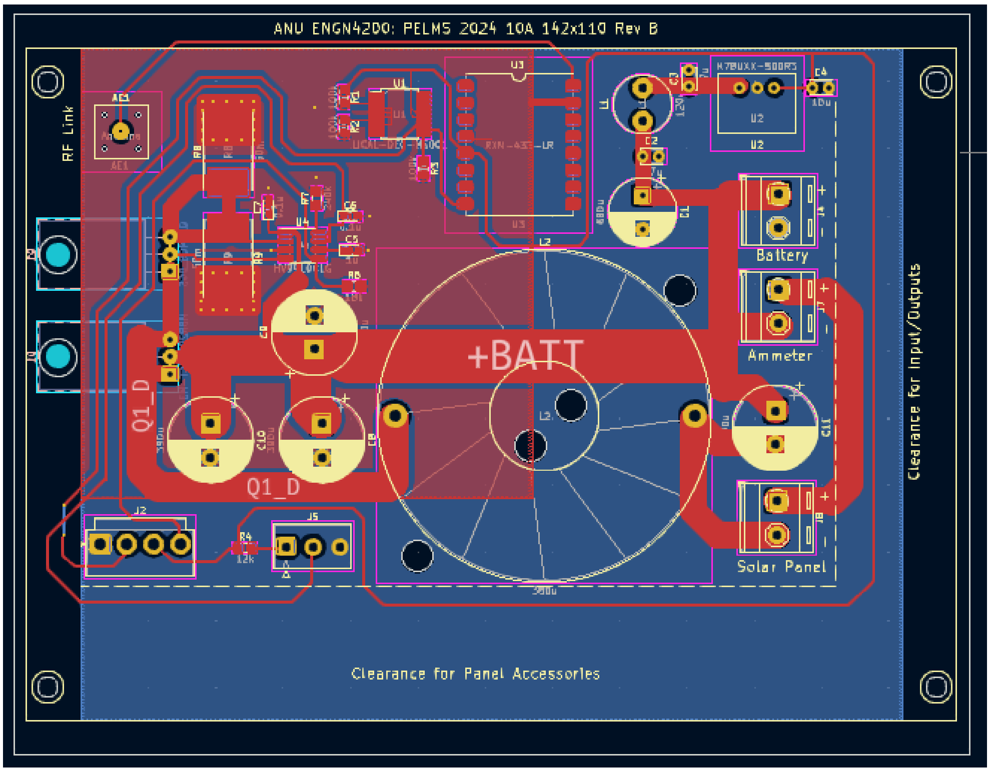

Figure 28: PELMS 2024 10A 142x110 Rev B – PCB Footprint

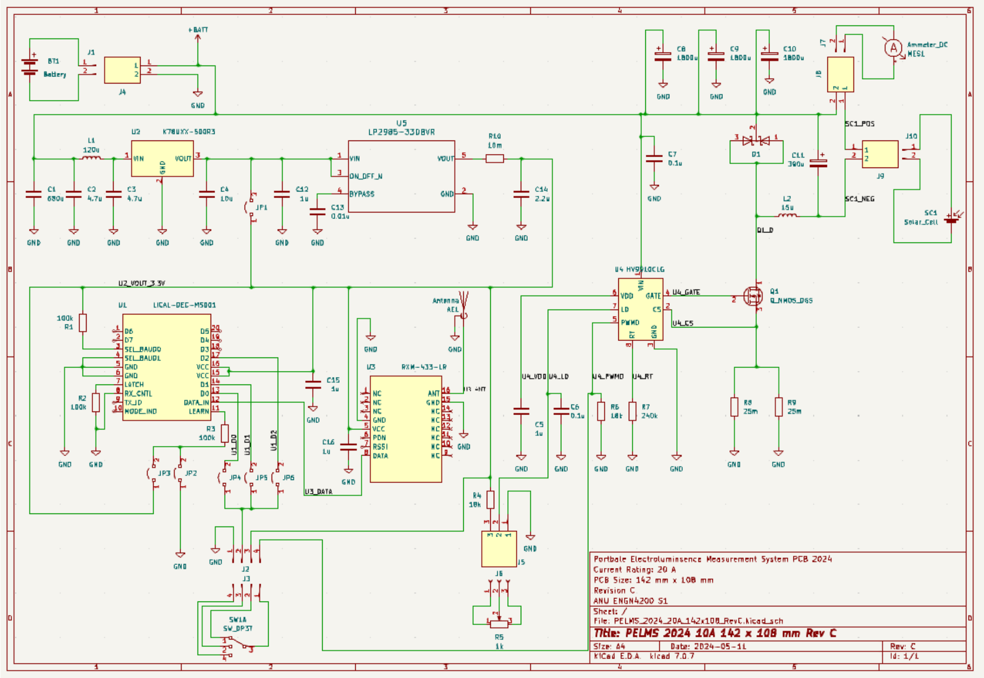

Figure 29: PELMS 2024 10A 142x110 Rev C – Schematic

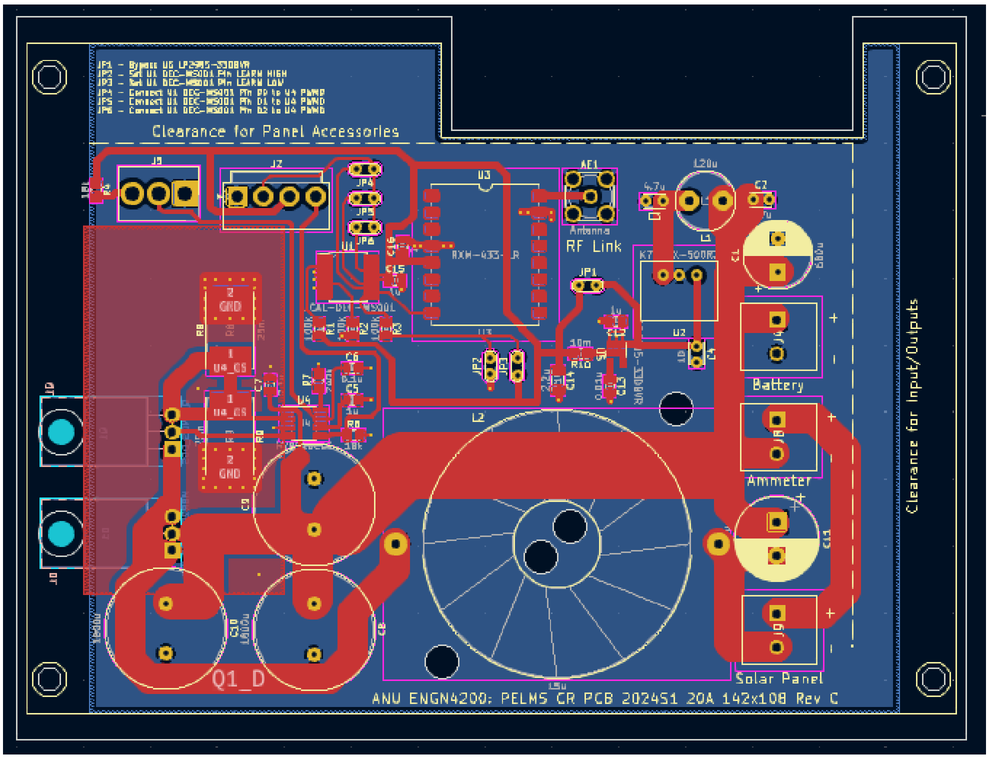

Figure 30: PELMS 2024 10A 142x110 Rev C – PCB Footprint

Appendix B: Enclosure Design Profiles

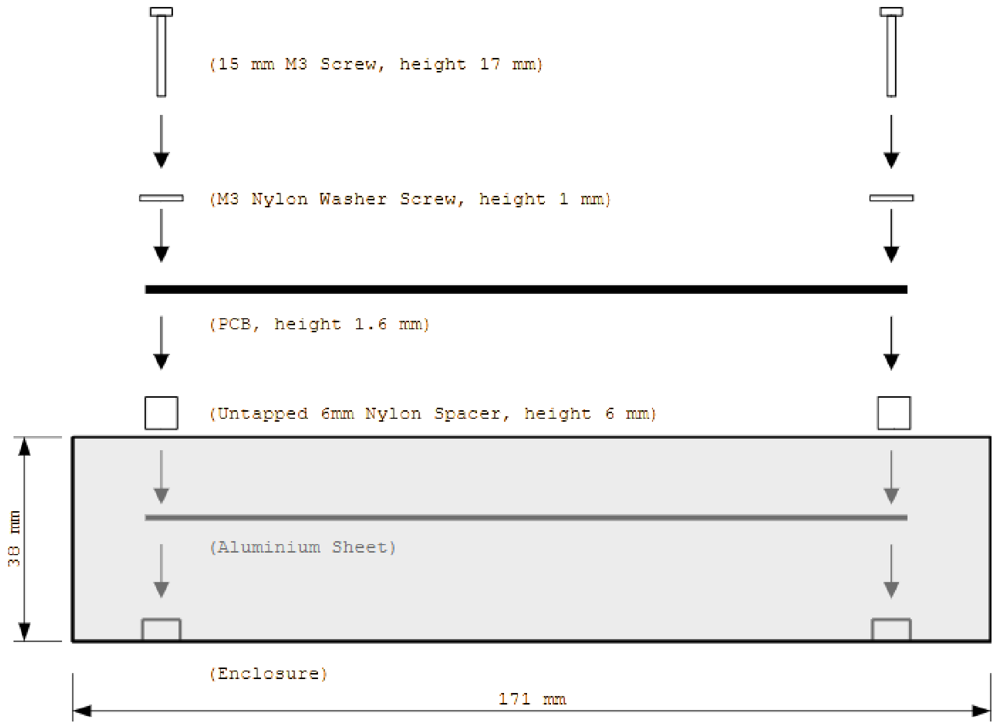

Figure 31: Current Regulator Enclosure – side profile.

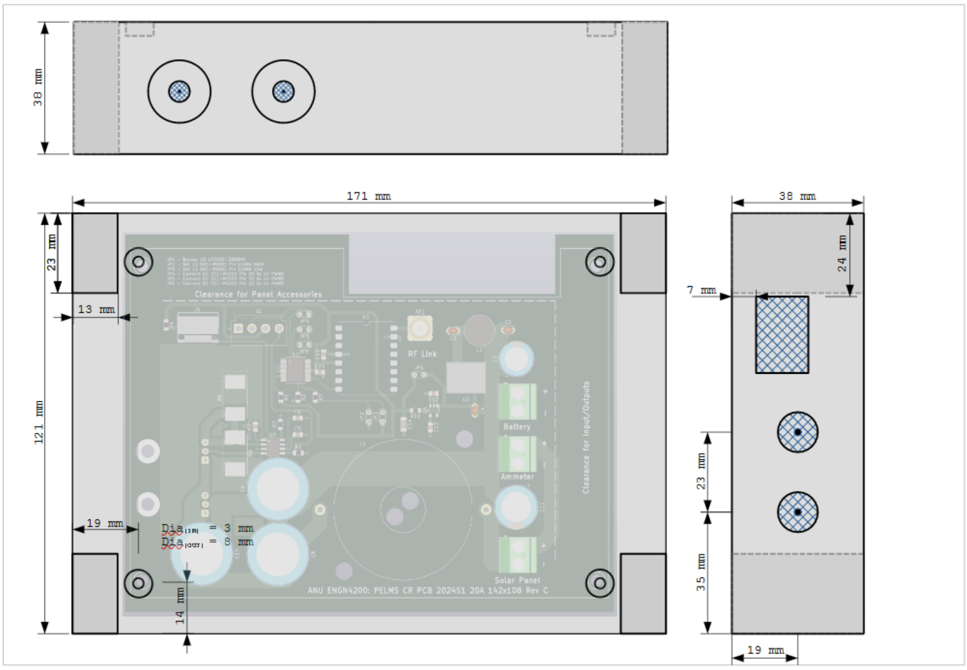

Figure 32: Current Regulator Enclosure – top, front, side profile.

Appendix C: PELMS 2023 Imaging Software Test

The PELMS 2023 Imaging Software Test was a test that generate some images of a PV module with 3 PV cells. The images were generated in three areas, which were:

- generating EL images for 1, 2 and 3 PV cells,
- generating EL images for currents of 7A, 8A and 9A, and
- generating EL images based on 10, 20 and 50 pairs.

The initial setup for the test is shown in Figure 28. The resulting images are given in Figures 30, 32 and 34 (Figures 29, 31, and 33 are simply refence images of the Solar Panel under test).

Figure 33: PELMS 2023 Imaging Software Base Testing Setup.

The EL images for 1, 2 and 3 PV Cells are shown in Figure 30. The left column contains the original image produced and the right column displays the images after image software adjustment (e.g. adjust contrast, brightness etc.). All the EL images were generated using a current of 8 A with 50 image pairs. The top row shows a 1 x PV Cell image, the middle row shows a 2 x PV cell image, and the bottom row shows a 3 x PV cell image.

The EL images for 7A, 8A and 9A injection currents are shown in Figure 32. The left column contains the original image produced and the right column displays the images after image software adjustment (e.g. adjust contrast, brightness etc.). All the EL images were generated using a 1 x PV cell with 50 image pairs. The top row shows a 7A EL image, the middle row shows an 8A EL image, and the bottom row shows a 9A cell image.

The EL images for 10, 20 and50 pairs are shown in Figure 34. The left column contains the original image produced and the right column displays the images after image software adjustment (e.g. adjust contrast, brightness etc.). All the EL images were generated using a 1 x PV cell with 8A injection current. The top row shows a 10 pair EL image, the middle row shows a 20 pair EL image, and the bottom row shows a 50 pair EL image.

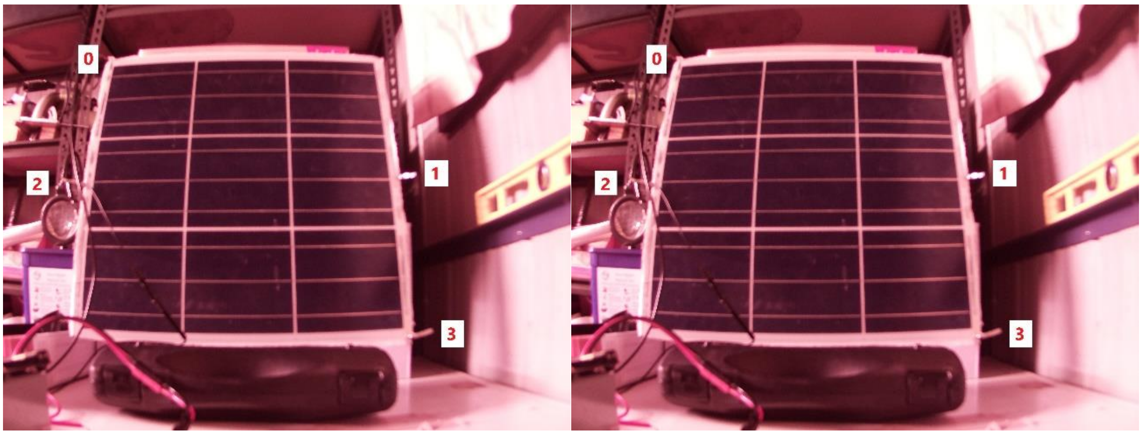

Figure 34: Reference Solar Panel Image for PELMS 2023 Software Testing.

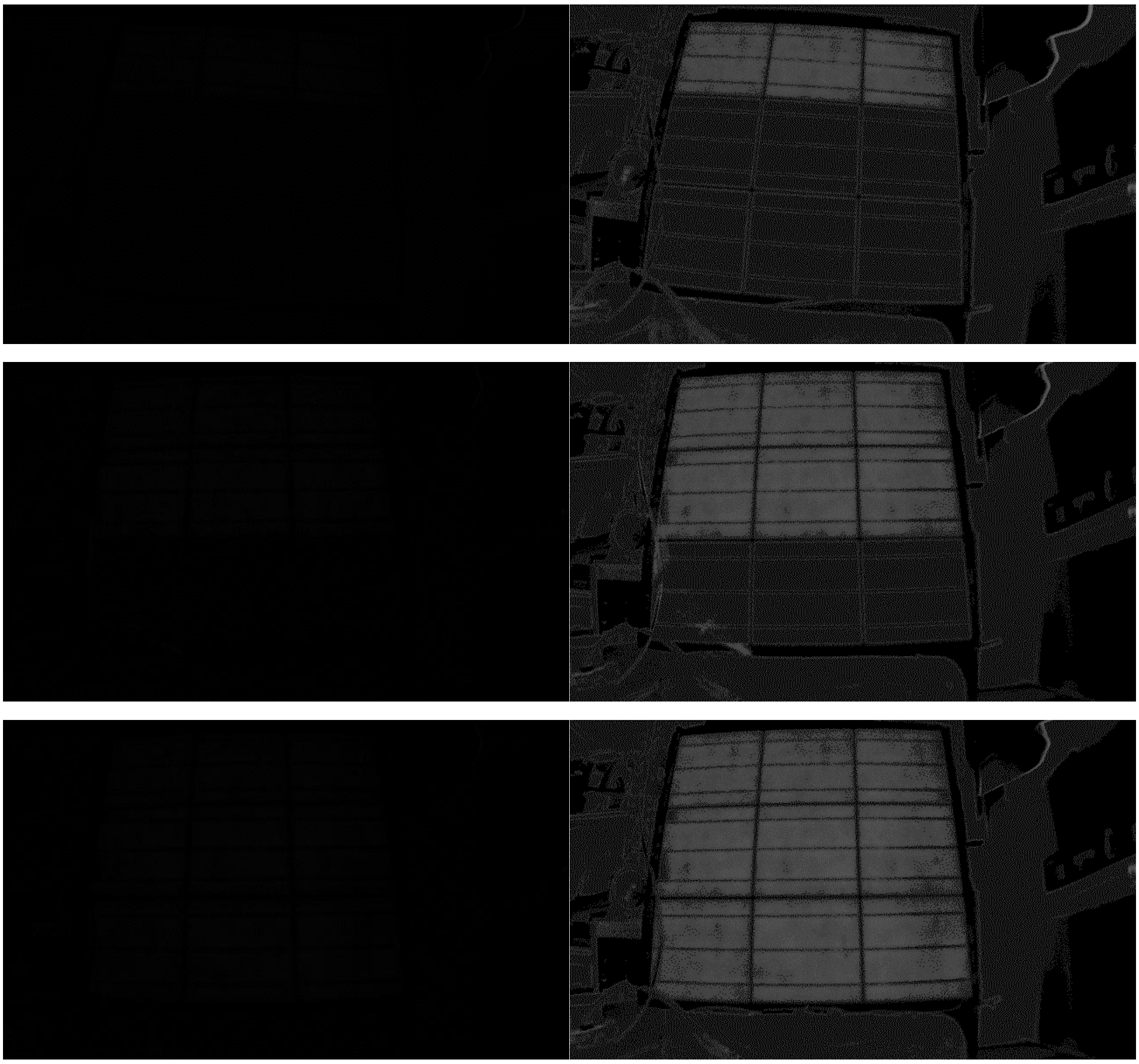

Figure 35: PELMS 2023 Software EL images for 1, 2 and 3 PV Cells.

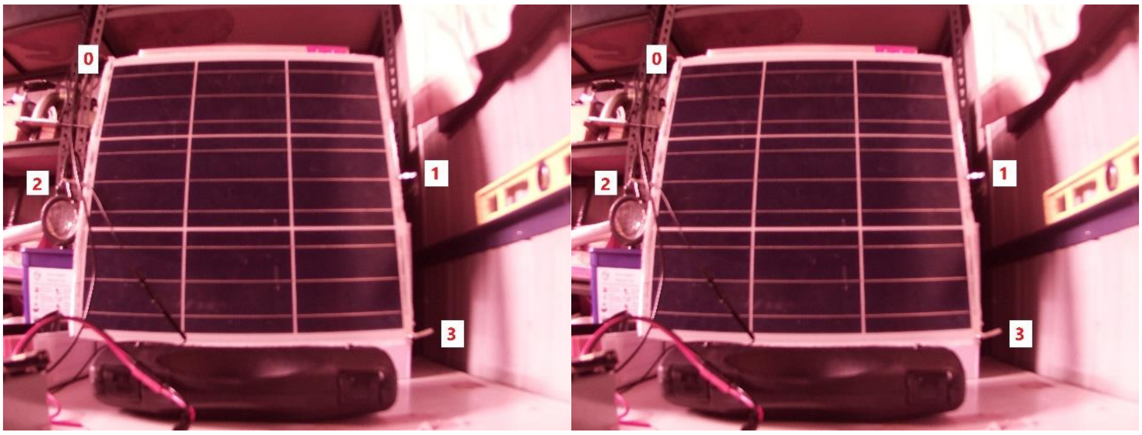

Figure 36: Reference Solar Panel Image for PELMS 2023 Software Testing.

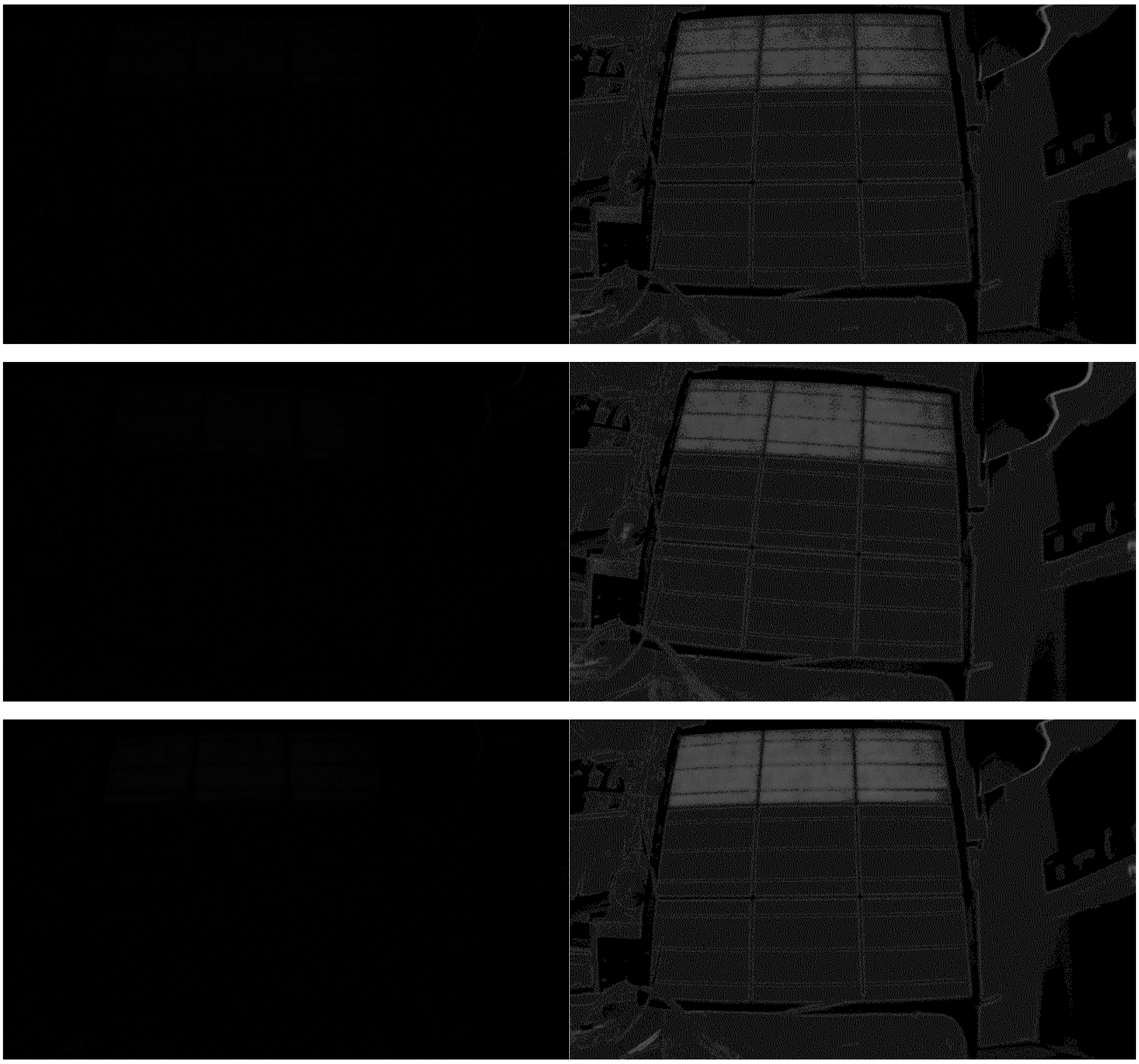

Figure 37: PELMS 2023 Software Image taken input current of 7A, 8A and 9A.

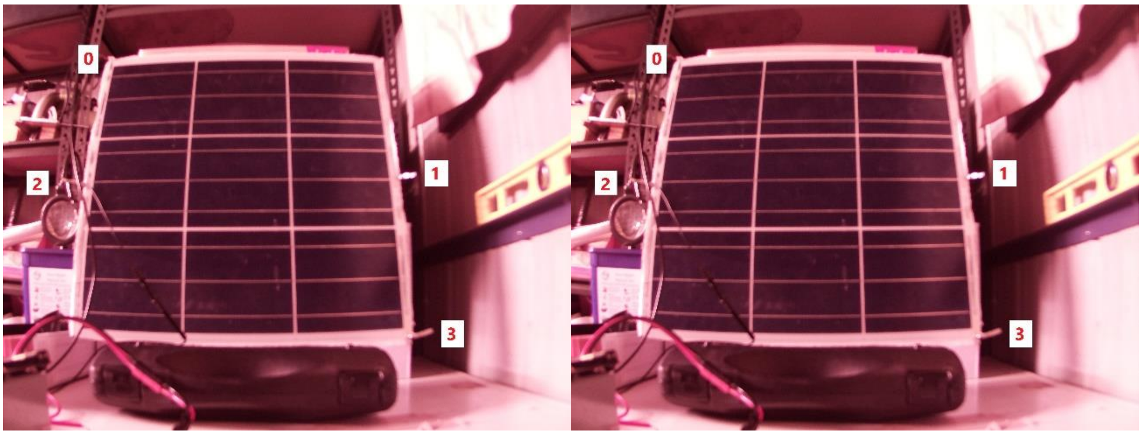

Figure 38: Reference Solar Panel Image for PELMS 2023 Software Testing.

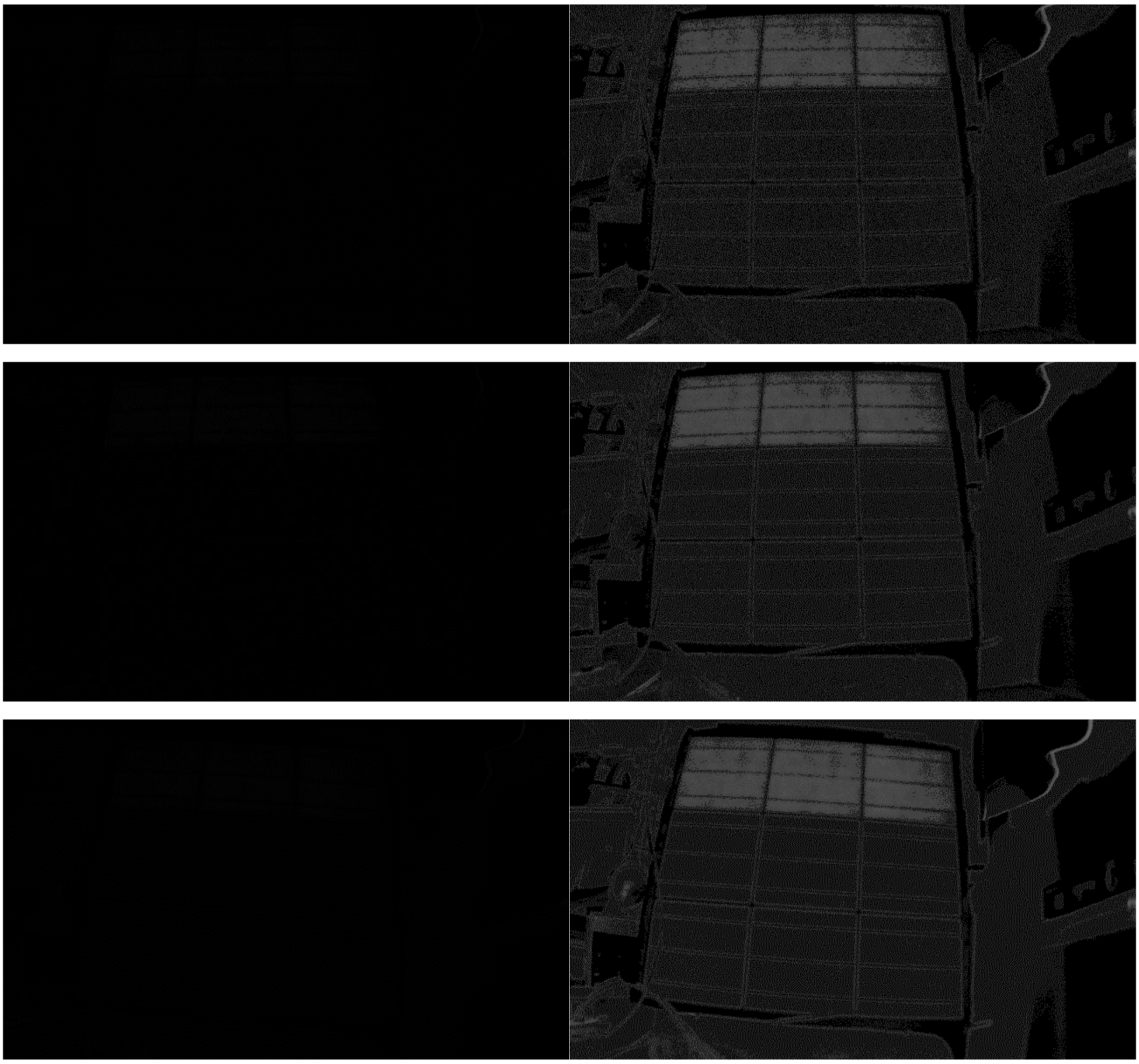

Figure 39: PELMS 2023 Software Image taken from 10, 20 and 50 pairs.

Appendix D: PELMS 2024 Imaging Software Test

The PELMS 2024 Imaging Software Test was a test that generate some images of a PV module with 3 PV cells. The images were generated in three areas, which were:

- generating EL images for 1, 2 and 3 PV cells,
- generating EL images for currents of 7A, 8A and 9A, and
- generating EL images based on 10, 20 and 50 pairs.

The initial setup for the test is shown in Figure 35. The resulting images are given in Figures 37, 39 and 41 (Figures 36, 38, and 40 are simply refence images of the Solar Panel under test).

Figure 40: PELMS 2024 Imaging Software Base Testing Setup.

The EL images for 1, 2 and 3 PV Cells are shown in Figure 37. The left column contains the original image produced and the right column displays the images after image software adjustment (e.g. adjust contrast, brightness etc.). All the EL images were generated using a current of 8 A with 50 image pairs. The top row shows a 1 x PV Cell image, the middle row shows a 2 x PV cell image, and the bottom row shows a 3 x PV cell image.

The EL images for 7A, 8A and 9A injection currents are shown in Figure 39. The left column contains the original image produced and the right column displays the images after image software adjustment (e.g. adjust contrast, brightness etc.). All the EL images were generated using a 1 x PV cell with 50 image pairs. The top row shows a 7A EL image, the middle row shows an 8A EL image, and the bottom row shows a 9A cell image.

The EL images for 10, 20 and50 pairs are shown in Figure 41. The left column contains the original image produced and the right column displays the images after image software adjustment (e.g. adjust contrast, brightness etc.). All the EL images were generated using a 1 x PV cell with 8A injection current. The top row shows a 10 pair EL image, the middle row shows a 20 pair EL image, and the bottom row shows a 50 pair EL image.

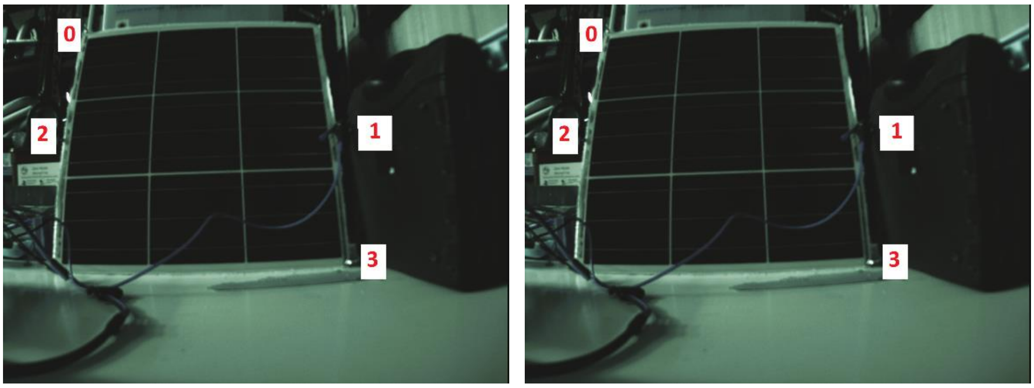

Figure 41: : Reference Solar Panel Image for PELMS 2024 Software Testing.

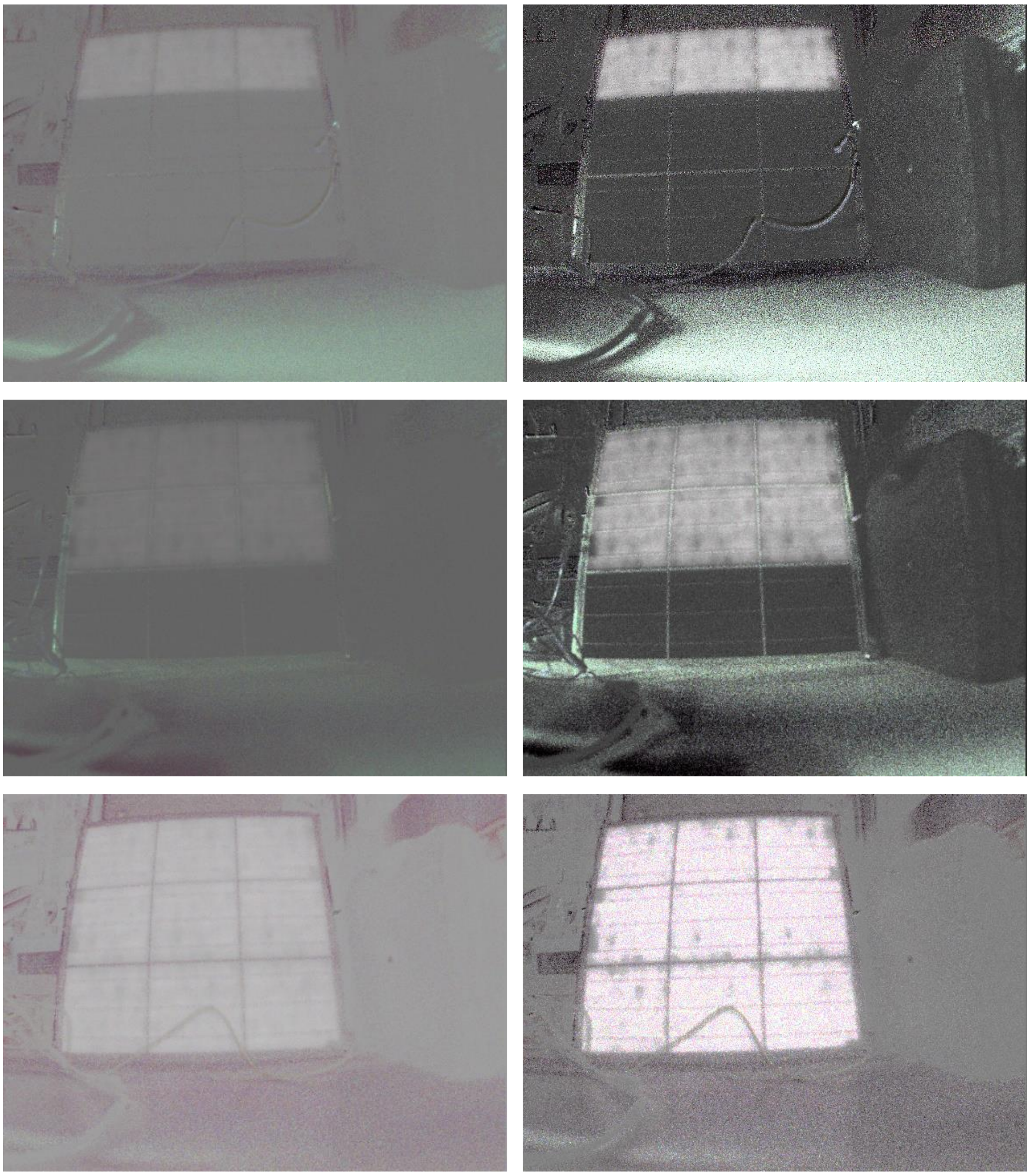

Figure 42: PELMS 2024 Software EL images for 1, 2 and 3 PV Cells.

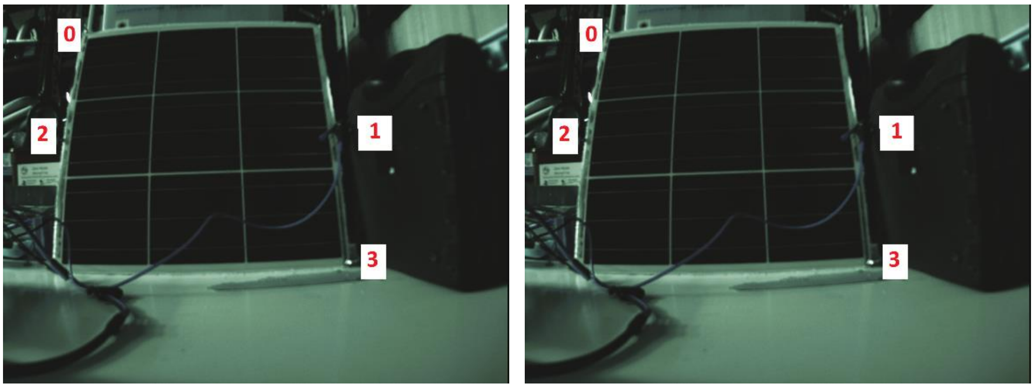

Figure 43: Reference Solar Panel Image for PELMS 2024 Software Testing.

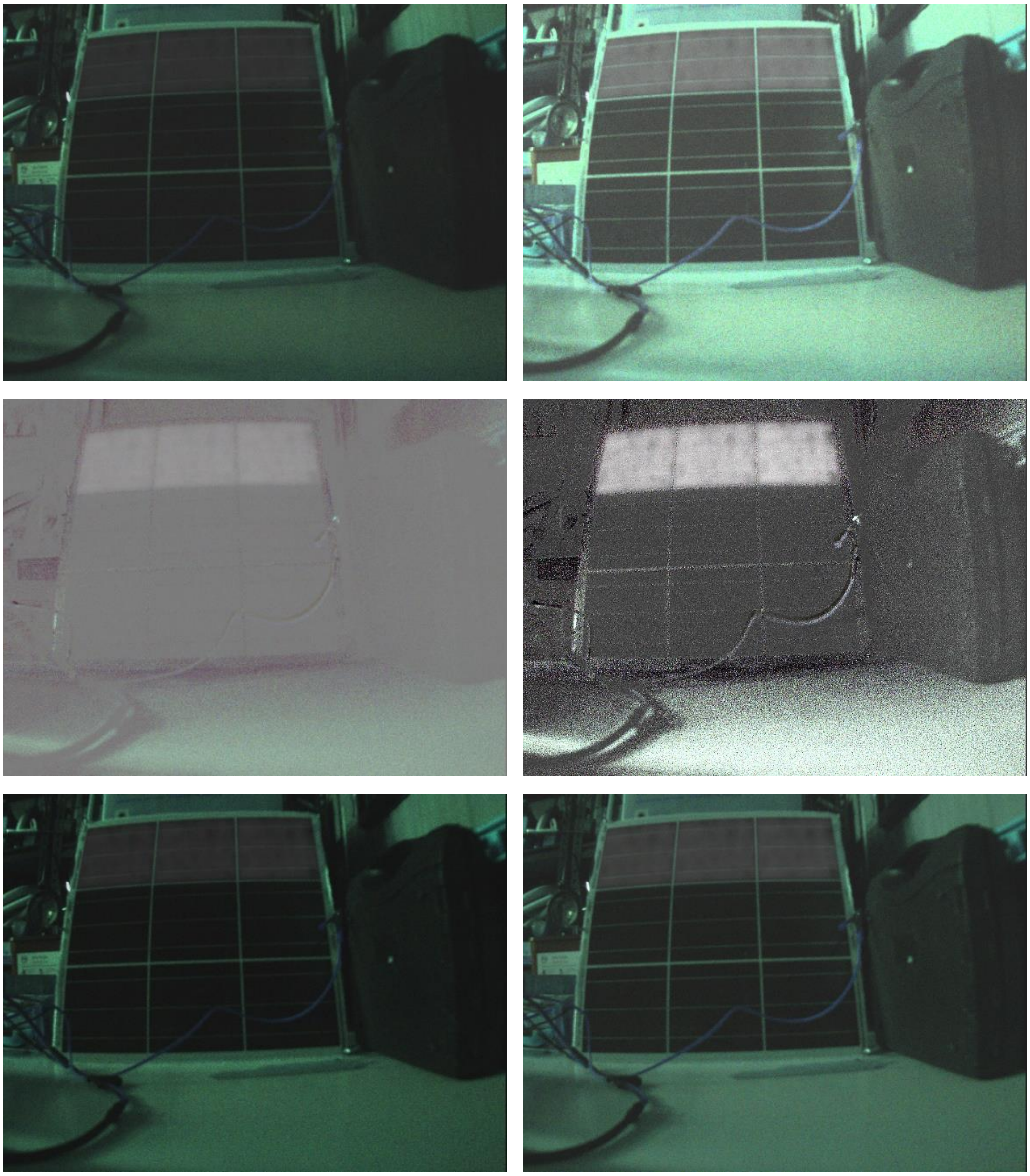

Figure 44: PELMS 2024 Software Image taken input current of 7A, 8A and 9A.

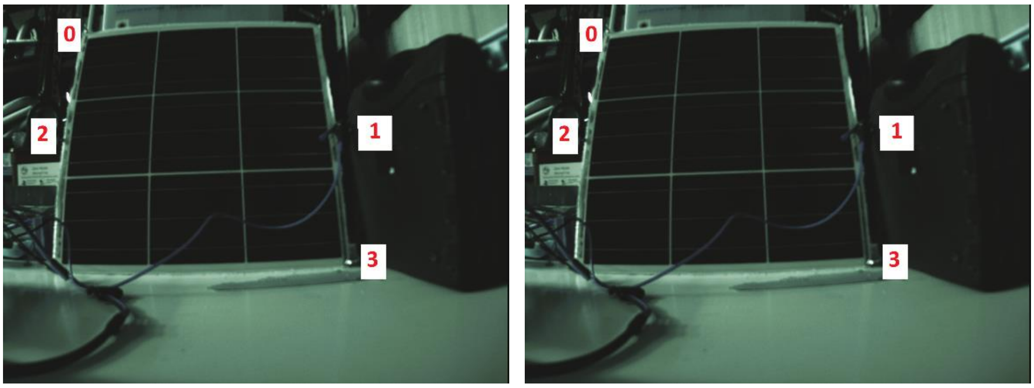

Figure 45: Reference Solar Panel Image for PELMS 2024 Software Testing.

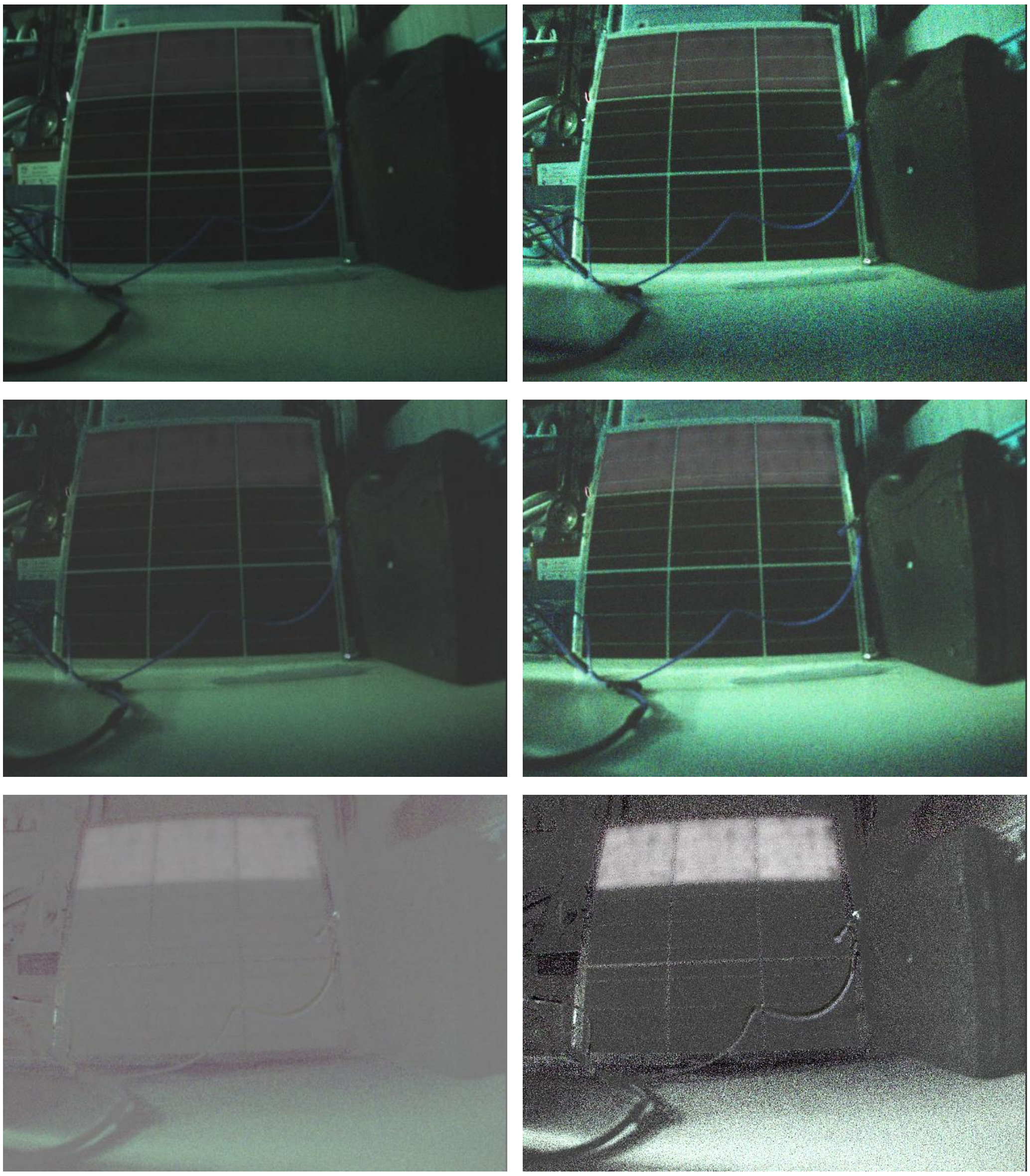

Figure 46: PELMS 2024 Software Image taken from 10, 20 and 50 pairs.