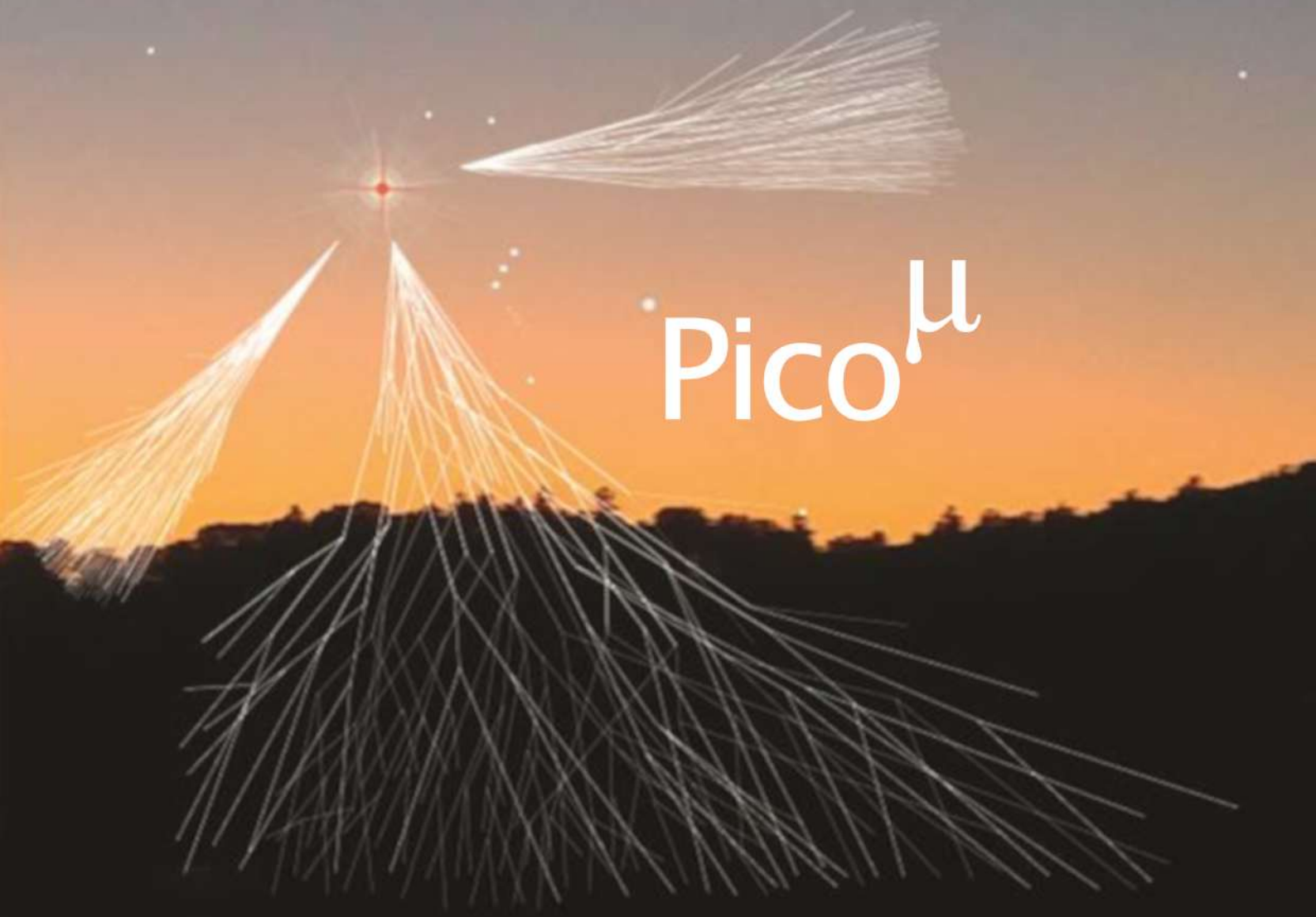
UK Radio Astronomy Association

PicoMuon Detector User Manual



https://ukraa.com/

The UK Radio Astronomy Association
A charitable incorporated organisation
Registered charity in England and Wales No. 1123866

Acknowledgements

Design Team

The UKRAA PicoMuon detector was developed by
Paul Hearn, Richard Knott and Dave Hardwick from the original work
of the Cosmic Watch detector.

Testing Team

The UKRAA PicoMuon detector was tested by
Paul Hearn, Andrew Thomas, Richard Knott, Dave
Hardwick, Simon Street and John Thain.

Production Team

The initial batch of the UKRAA PicoMuon detectors was produced by
Paul Hearn and Richard Knott.

Contributors

The following authors have contributed to the UKRAA PicoMuon detector
User Manual: Paul Hearn, John Thain, Andrew Thomas, Richard Knott,
Dave Hardwick and Dave Hilton

Table of Contents

Introduction

UKRAA

The UK Radio Astronomy Association (UKRAA) is a non-profit-making charitable incorporated organisation.

It was established by the Radio Astronomy Section of the British Astronomical Association (BAA) to facilitate the production and sale of radio astronomy products.

This Manual describes how to connect and use the UKRAA PicoMuon detector.

Any suggestions or recommendations for improvement of this Manual would be appreciated. See the **Contacts** page for further details.

The UKRAA PicoMuon detector

The UKRAA PicoMuon detector

The PicoMuon detector will detect ionising radiations – β-particles, γ-rays and muons

To eliminate the recording of β-particles and γ-rays from cosmic ray generated muons, the UKRAA PicoMuon detector has two scintillator assemblies, separated by 17.0mm, that can directly record coincidence events associated with cosmic ray muons.

The detected events are displayed on the OLED screen, output through a serial connection or recorded to an inserted microSD card.

The PicoMuon detector is an active device and requires a +5V supply via a micro-USB cable from suitable wall outlet, power-bank or suitable computer with USB type A socket.

Support

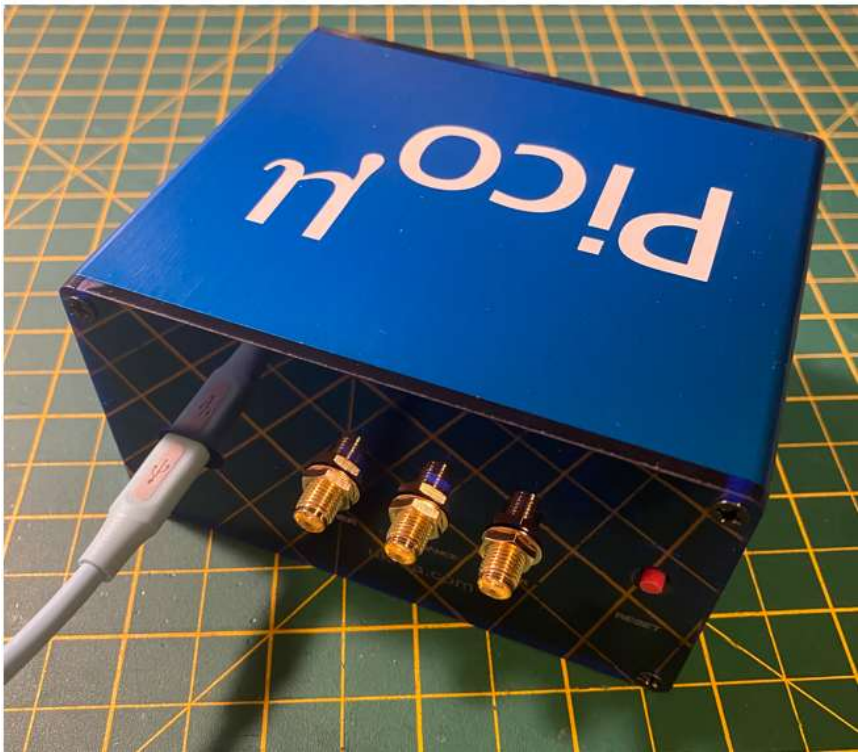
All users of the UKRAA PicoMuon detector are encouraged to make use of the support available from UKRAA, and others, for setting up and operation the detector.

For ongoing support there is a user forum where discussion, questions and ideas can be shared.

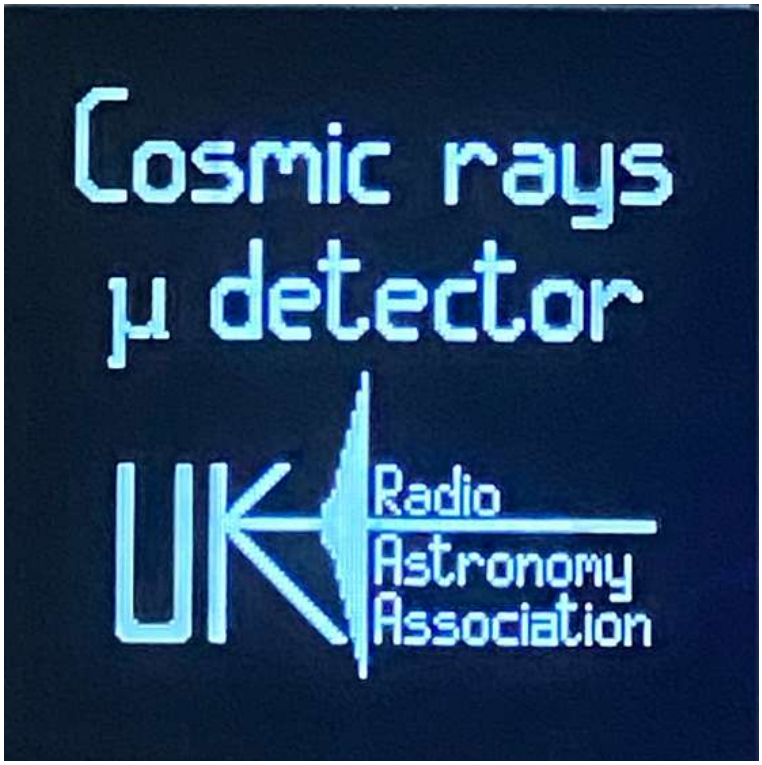
Please see the **Contacts** section for details.

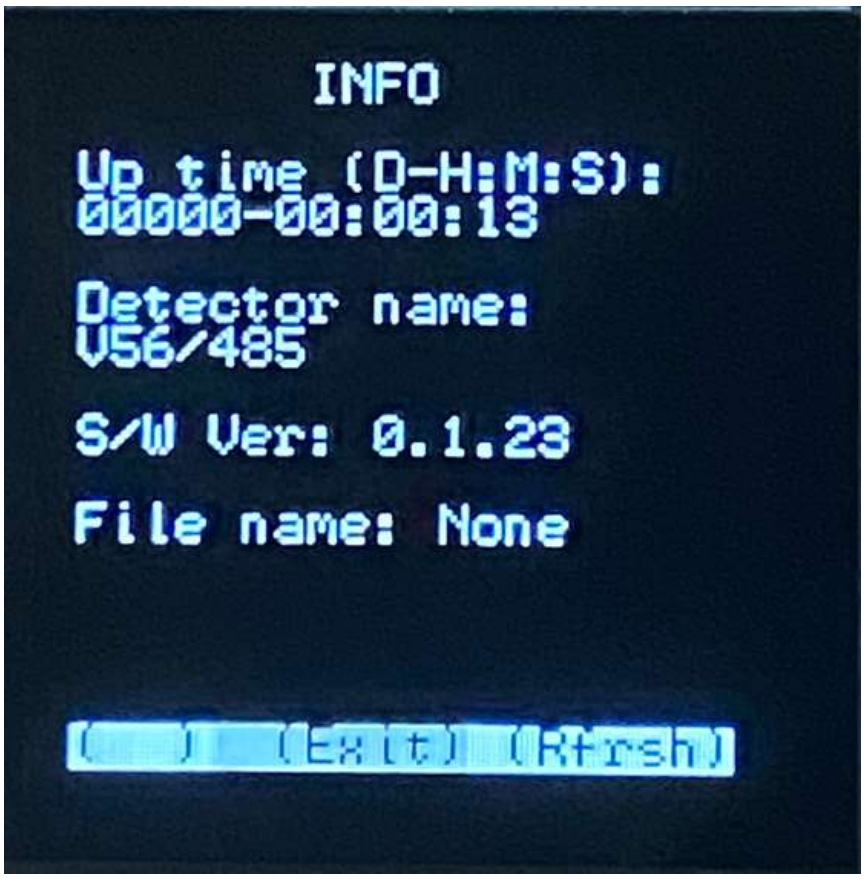
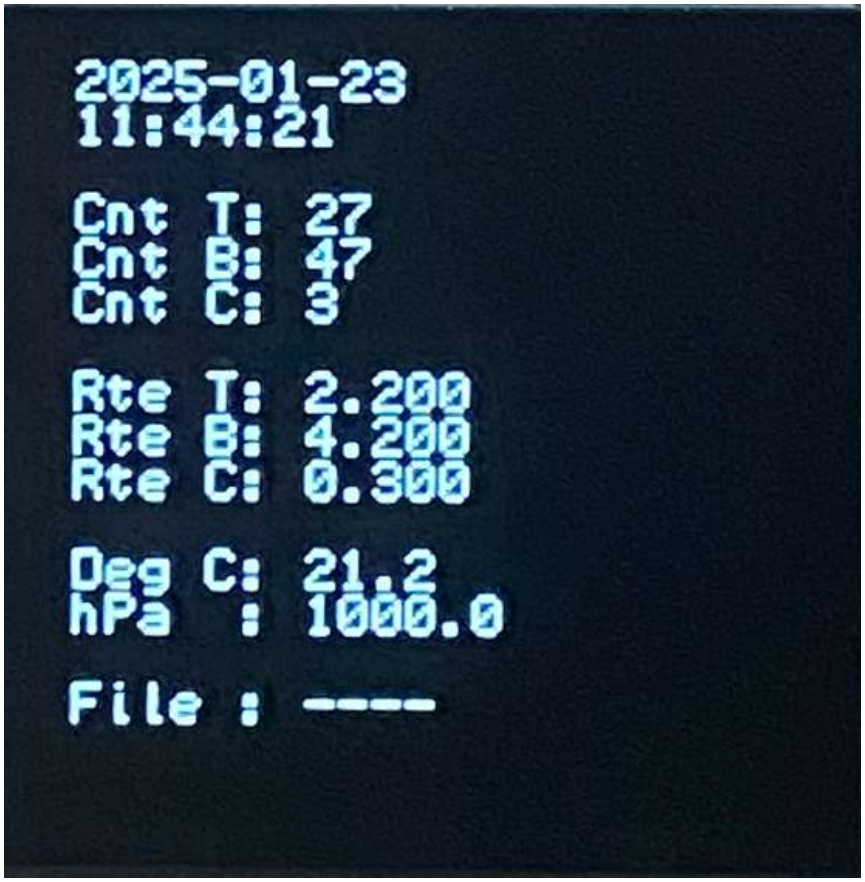
Unboxing and first power on

You will receive your PicoMuon detector in a white postal box.	
Upon opening the box you will see the included micro-USB cable, for connecting your PicoMuon detector to a suitable +5V DC supply – this can be a USB power outlet, a USB connector on a PC or a USB power bank.	

Removing the micro-USB cable, the PicoMuon detector is wrapped in corrugated cardboard for transit. Remove the cardboard and unwrap to reveal the PicoMuon detector **NOTE: Please retain the corrugated cardboard and the white postal box should you have need to return the unit to UKRAA.**	
Plug the micro-USB cable into the micro-USB socket located from the rear panel of the PicoMuon detector.	

Plug the USB type A connector into a suitable USB +5V DC adapter. **Note: picture indicates UK power plug – other countries power plugs will be different.**	
Insert the USB +5V DC adapter into a mains outlet and switch on. **Note: picture indicates UK power socket – other countries power sockets will be different.**	

The PicoMuon detector draws less than 1W, as measured with a USB power meter. So any suitable USB type A +5V DC adapter that can supply more than 1W should be suitable for powering the unit.	
The PicoMuon detector is now powered. Observing the front panel you will see all three LEDs illuminated at 100% for 5 seconds. You will then see all three LEDs illuminated at preset user configurable brightness as well as the UKRAA PicoMuon splash screen for 5 seconds.	

You will now see the **INFO** screen displayed, this is described in the **OLED display – INFO screen** – see this section for more information. The INFO screen will be displayed for 3 second.	
You will now see the **Runtime Display** screen displayed, this is described in the **OLED display – Runtime Display –** see this section for more information. **Congratulations** – your PicoMuon detector is now detecting β–particles, γ–rays and muons. The PicoMuon detector will remain on this screen, unless user input from the button switches located on the front panel.	

Front panel

DOWN button

Used to control the cursor on the OLED display screen when moving between menus

ENTER button

Used to access the PicoMuon menus and confirm action

UP button

Used to control cursor on the OLED display screen when moving between menus

OLED display

128x128 pixel monochrome display.

Displays various information relating to the function / working of the PicoMuon display.

MicroSD socket

Used to input user configuration parameter variables , via detector.ini file, on microSD card

Used to store event data on microSD card

Rear panel

RPi Pico micro-USB

Plug suitable micro-USB cable into this port to power the PicoMuon detector and enable data transfer from the PicoMuon detector to suitable serial interface

Reset switch

Press to action

Will restart the PicoMuon detector when pressed.

Also used to enable upload of new UF2 code in conjunction with RPi Pico **BOOT SEL** button.

Lower event sma connector

Lower detector event pulse can be accessed via this sma connector.

See Typical scope traces for shape of pulse

Coincidence event sma connector

Coincidence event pulse can be accessed via this sma connector.

See Typical scope traces for shape of pulse

Upper event sma connector

Upper detector event pulse can be accessed via this sma connector.

See Typical scope traces for shape of pulse

OLED display – Runtime Display screen

Date and time

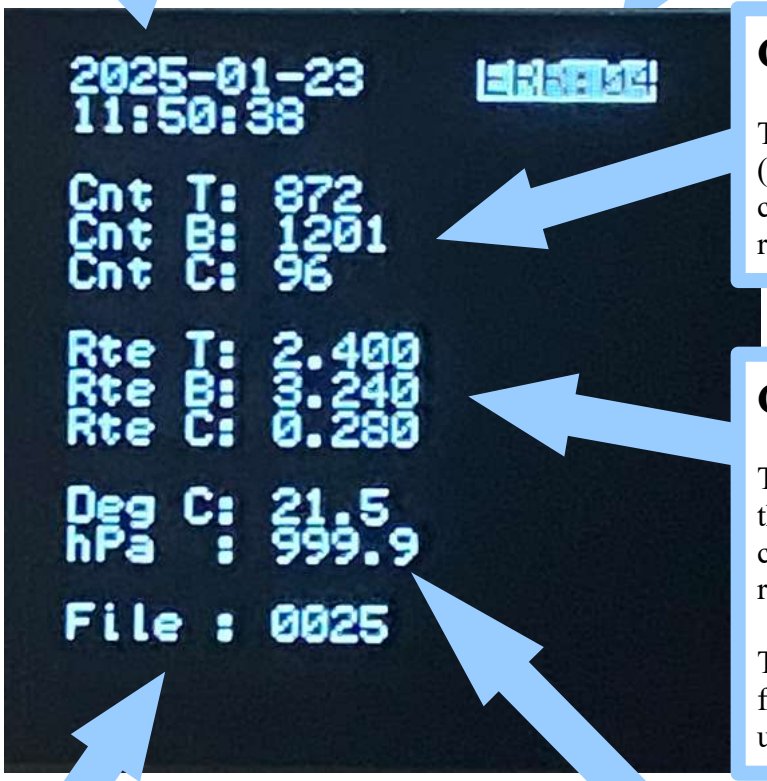
The current date (YYYY-MM-DD) and time (HH:MM:SS), taken from the DS3231 RTC at start-up, will be displayed here.

The time will increment each second.

ERROR code

Any error code will be displayed in the top right of the OLED display

See ERROR codes section for meaning



Counts

The total counts for the top (T), bottom (B) and coincidence (C) events are recorded here.

Count rates

The current count rate for the top (T), bottom (B) and coincidence (C) events are recorded here.

This is a rolling count rate for the last 60 seconds – in units of counts per second

File

Provides the current file name that is being used for saving event data to when a microSD card is inserted into the microSD card slot.

If no microSD card inserted will display

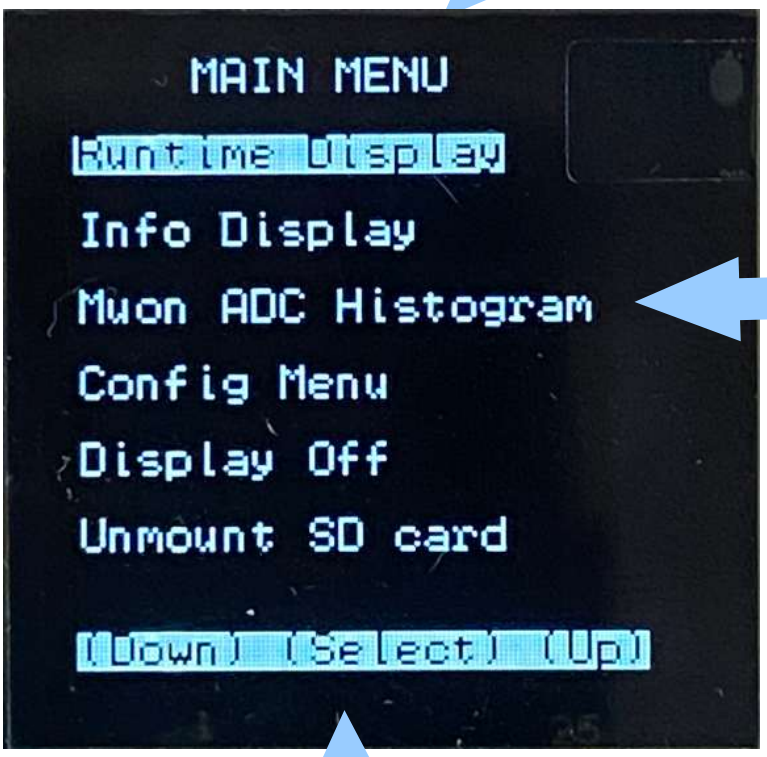
File : ----

Temperature and pressure

Displays the current temperature and pressure recorded from the BMP280 sensor.

Temperature is in deg C and pressure is in hPa

OLED display – MAIN MENU screen



Menu title

This will vary depending on which menu you are accessing

Available sub-menus

A list of available sub-menus will be displayed for the user to access.

The **Unmount SD card** will only be displayed if an microSD card is inserted into the microSD card slot

Button annotation and action

Displays the actions relating to the user pressing the down (left), enter (centre) and up (right) buttons located on the upper right of the front panel.

OLED display – INFO screen

The **Info Display** is accessed through the **MAIN MENU** by using the **DOWN** button to highlight **Info Display** and then pressing the **ENTER** button.

Menu title

This will vary depending on which menu you are accessing

Detector uptime

This displays the time, in days-hours:minutes:seconds, since the detector has been powered on.

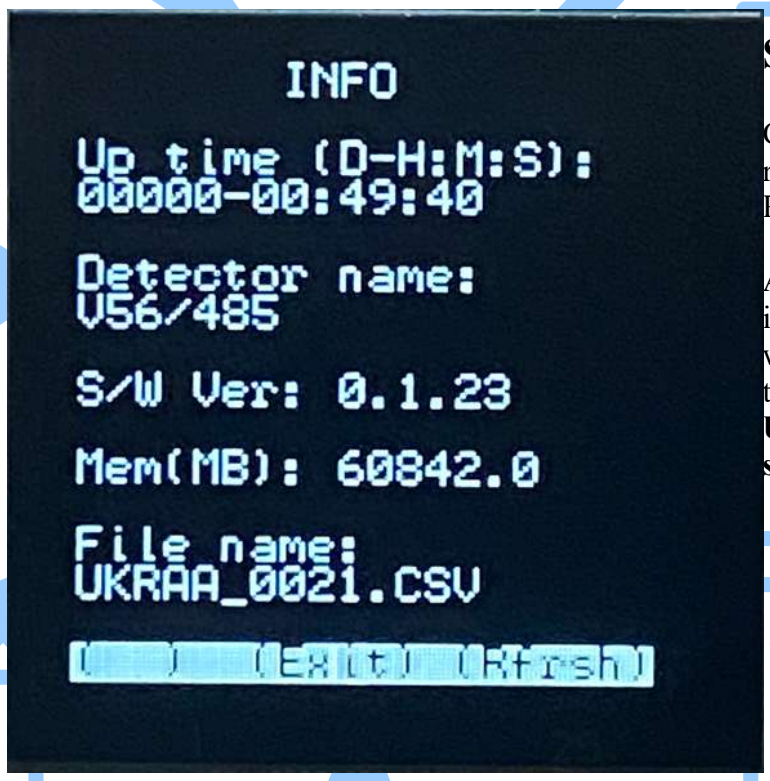
Selecting refresh will refresh the uptime displayed.

Detector name

User defined detector name.

Default is model-serial number.

Changed by user using detector.ini file.



Software Version

Current version of the running software on the PicoMuon detector.

As bugs/enhancements are identified – the core code will need to be updated by the user. Follow **Uploading updated software** for instruction.

File name

Full file name (including user defined file prefix – here **UKRAA_**) of file used to store event data on inserted microSD card.

If no microSD card inserted will display **File name: None**

Available memory on microSD

Current free capacity on the inserted microSD card.

As event data is stored to the microSD card – this will decrease.

If no microSD card is inserted then this line is not displayed..

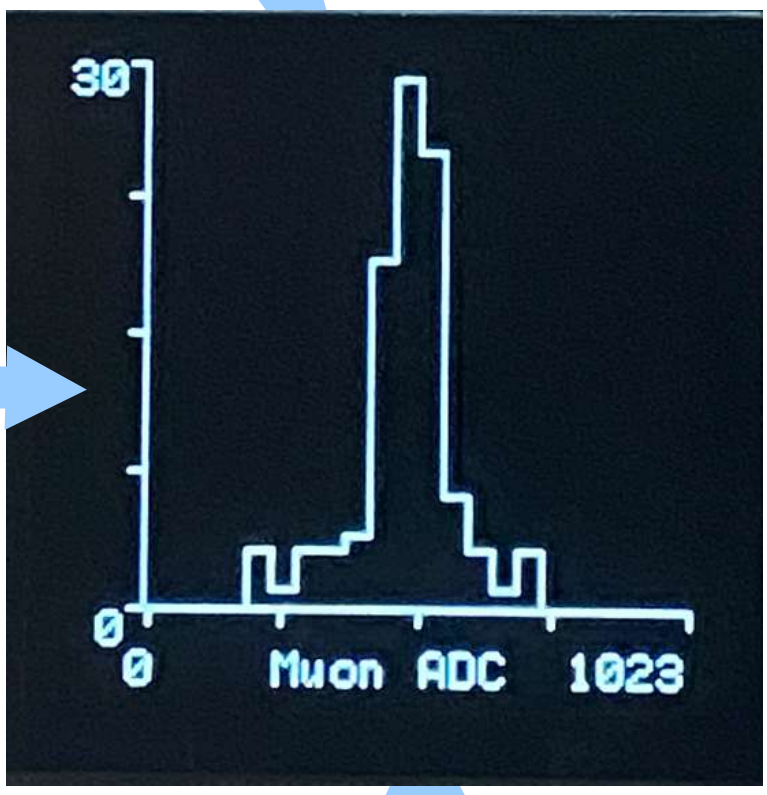
Button annotation and action

Displays the actions relating to the user pressing the down (left), enter (centre) and up (right) buttons located on the upper right of the front panel.

OLED display – Muon ADC Histogram screen

ADC histogram

The graph will update as time progresses and provides a rolling 5 minute snap shot of the recorded ADC values associated with coincidence (muon) events recorded by the PicoMuon detector.



Frequency of ADC value

Y-axis represents the frequency of ADC values recorded for the last 100 events.

Scale is 0 to 30.

Muon ADC value

X-axis represents the ADC values of recorded muon events in the last 5 minutes.

Scale is 0 to 1023 ADC units, binned at 20 ADC units – remember there is a threshold below which event data is not recorded.

Although the PicoMuon detector is not calibrated – the recorded ADC values gives an indication of the energy deposited by the ionising radiation event recorded by the detector

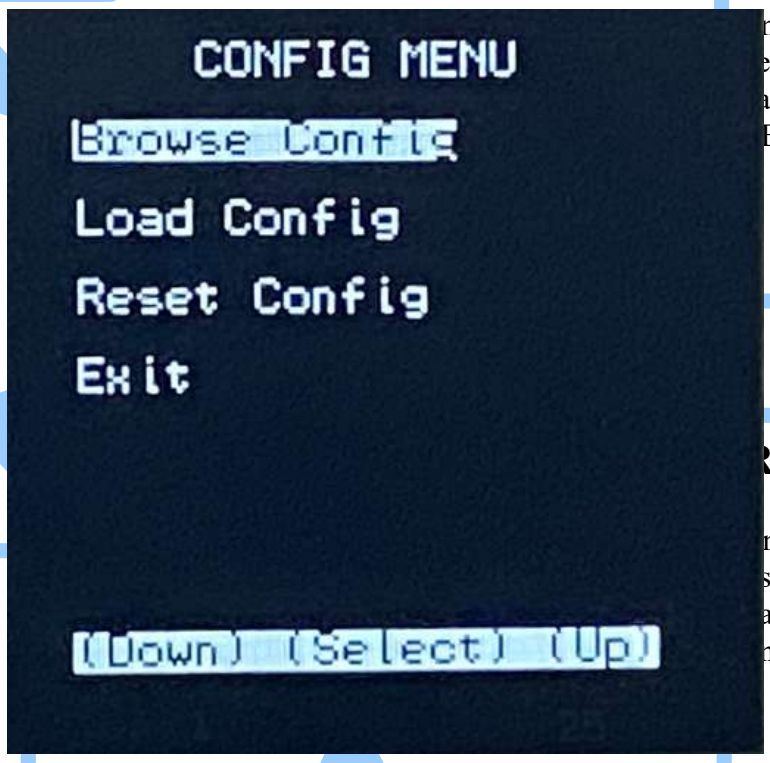
OLED display – CONFIG MENU screen

Browse Config

Enables user to scroll through the current EEPROM stored configuration parameter variables

Load Config

nables user to upload ew configuration arameter variables to the EPROM



Exit

Enables user to exit this menu.

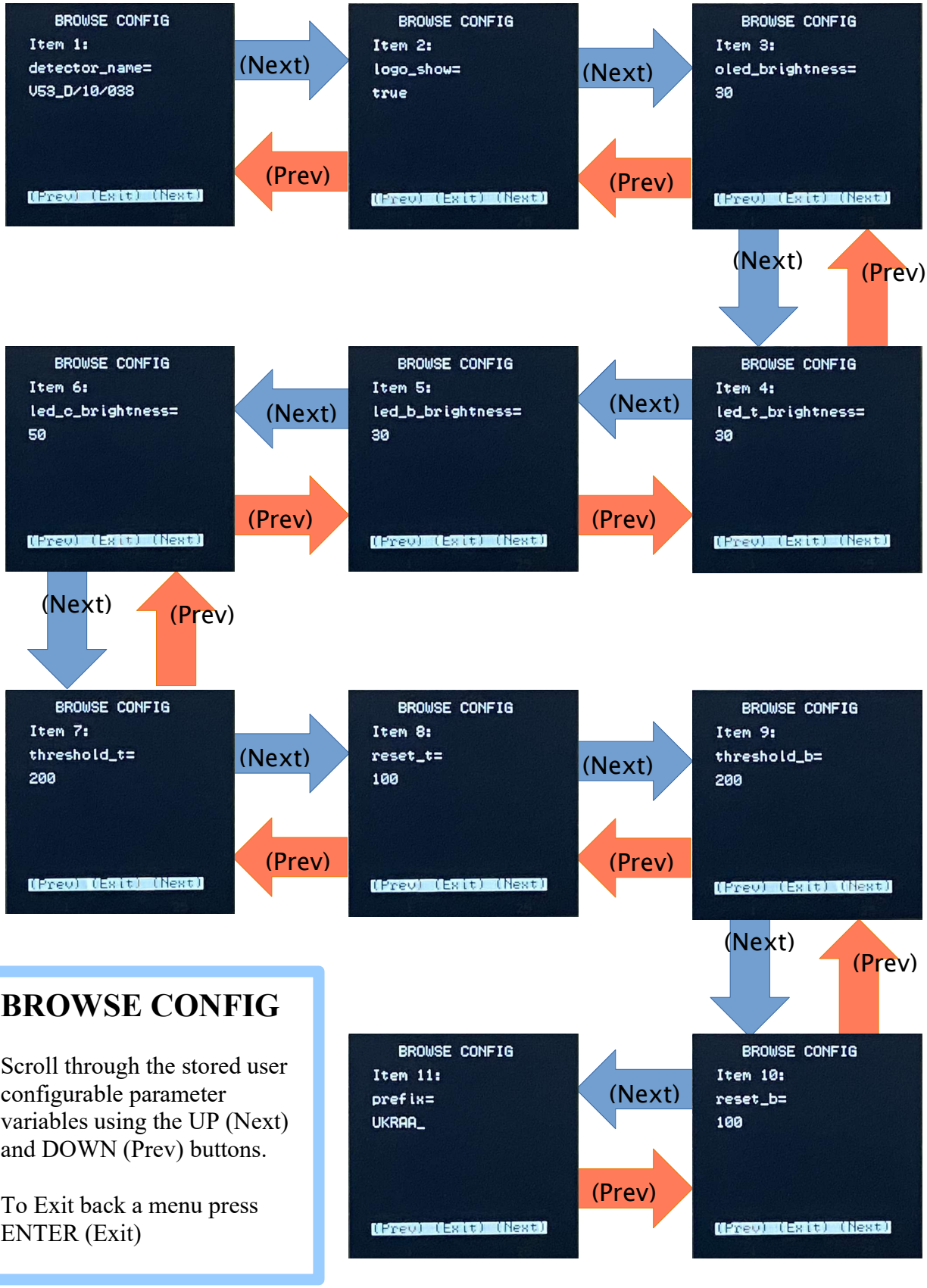
eset Config

nables user to reset the ser configurable arameter variables stored n the EEPROM

Button annotation and action

Displays the actions relating to the user pressing the down (left), enter (centre) and up (right) buttons located on the upper right of the front panel.

OLED display – BROWSE CONFIG screens



BROWSE CONFIG

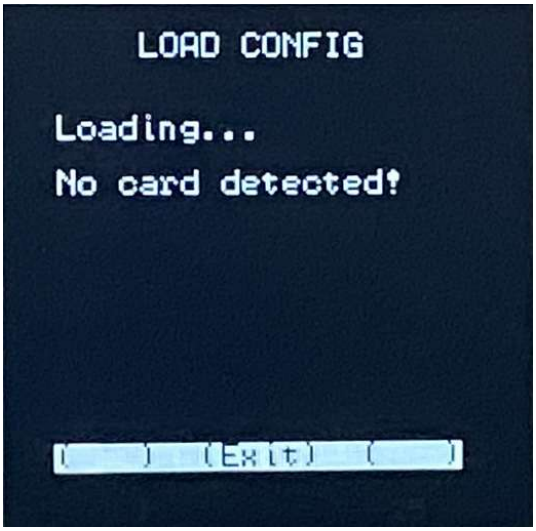
Scroll through the stored user configurable parameter variables using the UP (Next) and DOWN (Prev) buttons.

To Exit back a menu press ENTER (Exit)

OLED display – LOAD CONFIG screens

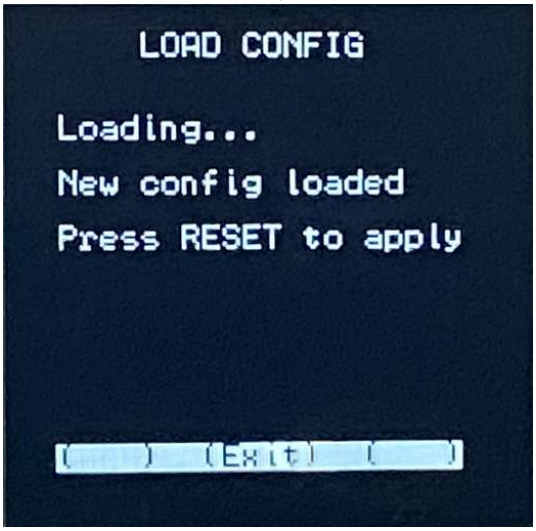
Selecting the **Load Config** option from the **CONFIG MENU** screen will bring up one of two screens

No microSD card inserted into the microSD slot



Action:
Press ENTER (Exit)
Insert microSD card into microSD card slot
Re-select Load Config

microSD card inserted into the microSD slot – with valid detector.ini file



The new configuration has been stored to the EEPROM.

The changed user parameter variables will be used once the PicoMuon detector is RESET.

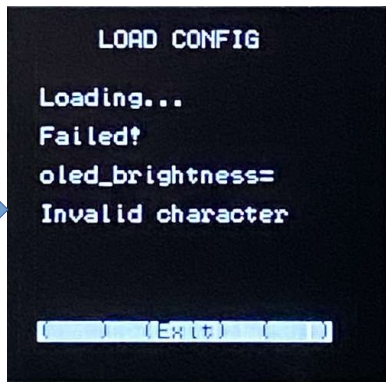
Press the **RESET** switch located on the rear panel to apply new configuration variables

OLED display – Invalid LOAD CONFIG screen

If there was any deviation from the expected user configurable parameter variables within the **detector.ini** file on the microSD card when loading new user configurable parameter variables – eg the inclusion of an alpha character when expecting a numerical character, or the variable is outside of the expected numerical range for that parameter, then a **Failed!** message will be displayed, along with the reason.

Invalid character

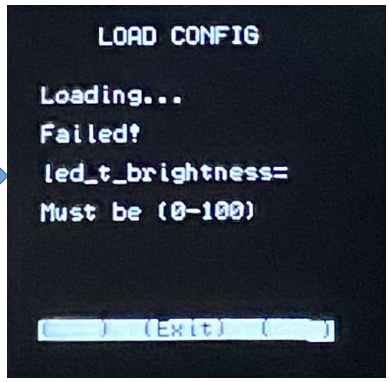
Action: correct error in **detector.ini** file and redo.



Invalid value - 1

```
[leds]
led_t_brightness = 300
```

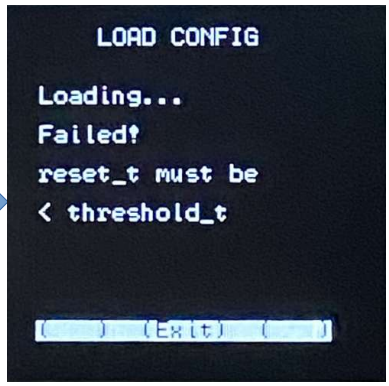
Action: correct error in **detector.ini** file and redo.



Invalid value - 2

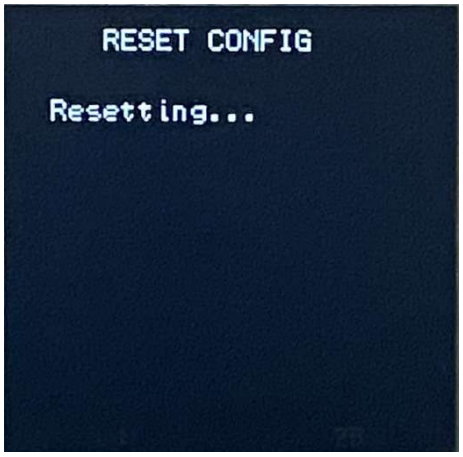
```
[event]
threshold_t = 200
reset_t = 300
```

Action: correct error in **detector.ini** file and redo.



OLED display – RESET CONFIG screen

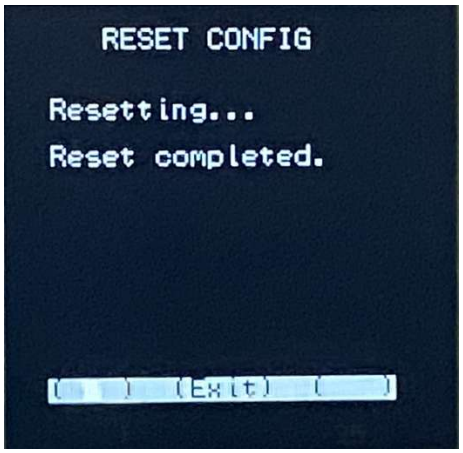
Selecting theReset **Config** option from the **CONFIG MENU** screen will bring up the following screen



This will reset the stored user configurable parameter variables,

Once complete you will see the following screen.

Press **ENTER** (Exit) to return to previous menu screen.



Base configuration values after RESET CONFIG actioned

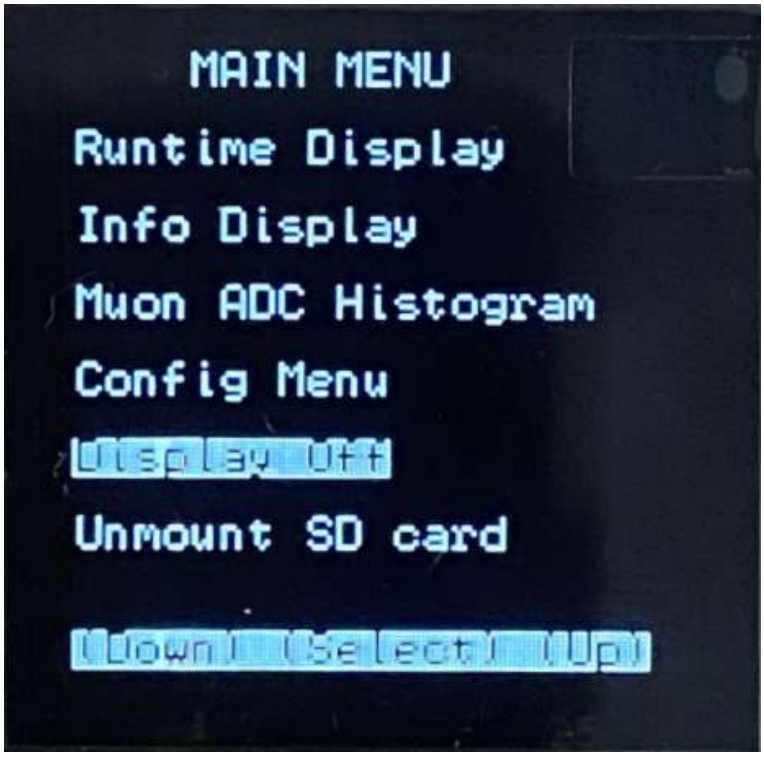
detector_name=Undifined
logo_show=true
oled_brightness=30
led_t_brightness=30
led_b_brightness=30
led_c_brightness=50
threshold_t=200
reset_t=100
threshold_b=200
reset_b=100
prefix=UKRAA_

OLED display – Display off

OLED displays have a unique feature of “burn in” – where if the display has the same unchanging text for a long period of time then some of the pixels will either fade or not function.

To avoid this happening you have the opportunity to turn your OLED display off when not needed.

This is available through the **MAIN MENU**.



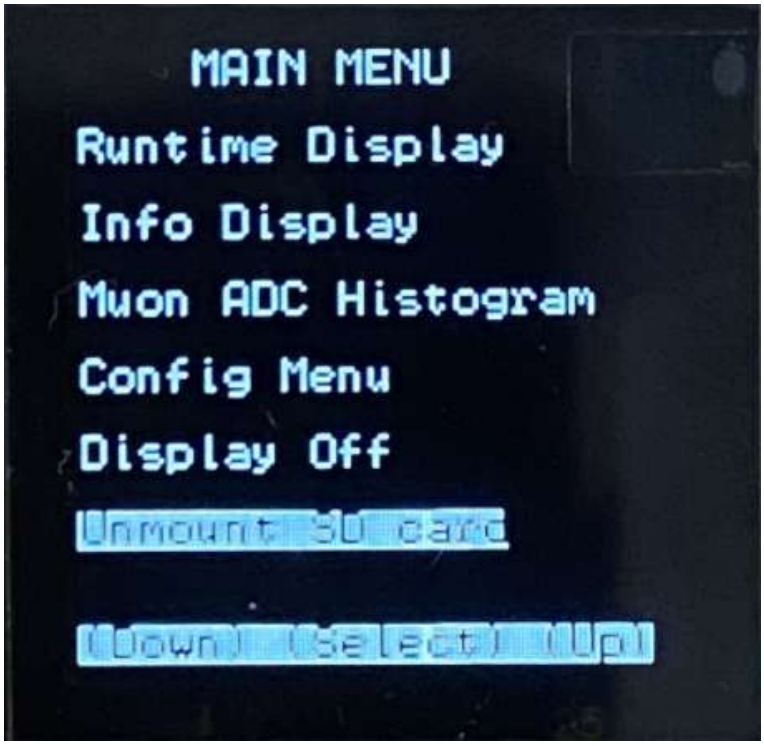
Selecting this option when the OLED display is not needed will improve the longevity of the PicoMuon OLED display.

Pressing the **ENTER** button on the front panel will bring the OLED display back on.

OLED display – Unmount SD card

To be able to safely eject an inserted microSD card, there is an option available via the **MAIN MENU** that enables this to happen.

It is always sensible to **Unmount** the inserted microSD card from the PicoMuon detector – rather than just pulling it out while still writing data to it.



Changing user configuration values

1) The PicoMuon detector has user configurable parameter variables, that can be set using a text file named **detector.ini**.

2) The structure of the **detector.ini** file is as follows:

```
; UKRAA PicoMuon detector - User configuration file
;
; Follow the instructions in the UKRAA PicoMuon detector manual for editing this file
;
; Follow the instructions in the UKRAA PicoMuon detector manual for using this file
;

[name]                          ; detector name configuration
detector_name = V55_D/11/043    ; 16 characters maximum

[start_up]                      ; Show logo/info screen on OLED at start-up
logo_show = true                ; [true] = show, false = don't show

[display]                       ; OLED configuration
oled_brightness = 30            ; Brightness of OLED display - [0 to 100]%

[leds]                          ; LEDs configuration
led_t_brightness = 30           ; Brightness of bottom LED display - [0 to 100]%
led_b_brightness = 30           ; Brightness of top LED display - [0 to 100]%
led_c_brightness = 50           ; Brightness of coincidence LED display - [0 to 100]%

[event]                         ; Event detection configuration
threshold_t = 200               ; Threshold level for top event to be recorded [0 to 4095]
reset_t = 100                   ; Reset level for next top event to be recorded - smaller than threshold
threshold_b = 200               ; Threshold level for bottom event to be recorded [0 to 4095]
reset_b = 100                   ; Reset level for next bottom event to be recorded - smaller than threshold

[microsd]                       ; microSD configuration
prefix = UKRAA_                 ; microSD card filename prefix
```

3) The **detector.ini** file is separated into 6 sections, that the user can modify to their own requirements.

4) If you wish to edit the **detector.ini** file, it is recommended to use a plain text editor such as “notepad++”, https://notepad-plus-plus.org/.

5) Only change the text after the “=” for each of the parameters, Do not change the parameter name – these are used by the PicoMuon detector code to register and store the correct variable to the onboard EEPROM.

6) If **detector.ini** is modified – save the modified file and then copy onto a suitable microSD card – do not change the file name.

7) Insert this microSD card into the microSD card slot located on the front panel of the PicoMuon detector.

8) Access the menu screen of the PicoMuon detector by pressing the ENTER button located on the front panel of the PicoMuon detector. Navigate to the **Config Menu** and then select **Load Config.**

9) Once the parameters and variables from the **detector.ini** file have been stored on the onboard EEPROM of the PicoMuon detector, removal of power will not remove/alter the stored parameter variable values.

Table of parameters/variables from detector.ini file

Parameter	Variable
detector_name	Name the user wishes to describe their PicoMuon detector. Can be maximum of 16 characters. **Preset value = detector model~serial number – e.g. V56~500**
logo_show	Show UKRAA logo at start up. "true" = show logo – this is the preset value "false" = don't show logo **Preset value = true**
events_all	Enable logging of all events. "true" = log Top, Bottom and Coincidence events. "false" = log Coincidence events only. **Preset value = true**
oled_brighness	Sets the brightness of the OLED display, set between 0 to 100. 0 would be off. **Preset value = 30**
led_t_brightness	Sets the brightness of the LED that illuminates when the PicoMuon detector records an event from the upper detector. Set between 0 to 100. 0 would be off. **Preset value = 30**
led_b_brightness	Sets the brightness of the LED that illuminates when the PicoMuon detector records an event from the lower detector. Set between 0 to 100. 0 would be off. **Preset value = 30**
led_c_brightness	Sets the brightness of the LED that illuminates when the PicoMuon detector records a coincidence event. Set between 0 to 100. 0 would be off. **Preset value = 50**
threshold_t	Sets the threshold value of the upper RPi Pico ADC for which the PicoMuon detector records an event from the upper detector.

	Set between 0 to 4095. 0 would be off. **Preset value = 200**
Parameter	**Variable**
reset_t	Sets the reset value of the upper RPi Pico ADC for which the PicoMuon detector is enabled to record a subsequent event from the upper detector. Set between 0 to 4095 and also lower then threshold_t. 0 would be off – not a good idea as the previous event could still be above threshold_t value and be recorded as another event. **Preset value = 50**
threshold_b	Sets the threshold value of the lower RPi Pico ADC for which the PicoMuon detector records an event from the lower detector. Set between 0 to 4095. 0 would be off. **Preset value = 200**
reset_b	Sets the reset value of the lower RPi Pico ADC for which the PicoMuon detector is enabled to record a subsequent event from the lower detector. Set between 0 to 4095 and also lower then threshold_t. 0 would be off – not a good idea as the previous event could still be above threshold_t value and be recorded as another event. **Preset value = 50**
prefix	Sets the prefix of the filename when storing data to microSD. Use ASCII text characters. **Preset value = UKRAA_**

Event data – microSD

1) Events detected by the PicoMuon detector can be stored directly to a suitable microSD card, inserted into the microSD card slot located via the front panel.
2) When a microSD card is inserted into the PicoMuon detector, the card will be detected and a file will be created for storing event data.
3) The file name consists of {prefix}+{number}+.csv – where:
   - {prefix} = user *prefix* parameter variable defined in **detector.ini**
   - {number] = sequential number generated by code – starting 000, and incrementing each time PicoMuon is reset, or microSD card is unmounted and then reinserted.
   - .csv = file type defined
   - So typical file name would be UKRAA_0001.csv by default, then incrementing to UKRAA_0002.csv when PicoMuon detector is reset or microSD card is unmounted and then reinserted
4) Format of data stored to microSD is as follows:
   - Header section:

```
#######################################################################
### UKRAA: The Pico Muon Detector                                   ###
### S/W version: 0.1.23                                             ###
### Detector name: V56/485                                          ###
### Thresholds: T(200 ,100 ), B(200 ,100 )                          ###
#######################################################################
```

   - This is 71 characters wide and 6 lines of plain text, comprising
     - Software version currently being run by the PicoMuon detector
     - The name of the detector
     - The threshold and reset ADC values, set for the upper (T) and lower (B) SiPM assemblies

   - There is then a single line that identifies the parameters that are recorded:

```
Date,Time,ID,Event,ADC(0-1023),ElapsedTime(mS),DeadTime(mS),Temperature(C),Pressure(hPa),DetectorName
```

   - This is 103 characters wide and a single line of plain text, comprising of data column identifiers.
   - ID indicates the detector position: T = top (upper), B = bottom (lower) and C = coincidence (muon).

- Finally there is the event data recorded to the microSD card relating to the column headers shown above.

  2025-01-24,10:21:04,B,1,251,87,1,20.2,992.4,V56/485
  2025-01-24,10:21:04,T,1,82,235,0,20.2,992.4,V56/485
  2025-01-24,10:21:05,T,2,231,1075,1,20.2,992.4,V56/485
  2025-01-24,10:21:05,B,2,92,1193,6,20.2,992.4,V56/485

5) Here is an example of a data file opened in notepad:

```
##########################################################################
### UKRAA: The Pico Muon Detector                                      ###
### S/W version: 0.1.23                                                ###
### Detector name: V56/485                                             ###
### Thresholds: T(200 ,100 ), B(200 ,100 )                             ###
##########################################################################
Date,Time,ID,Event,ADC(0-1023),ElapsedTime(mS),DeadTime(mS),Temperature(C),Pressure(hPa), DetectorName
2025-01-24,10:21:04,B,1,251,87,1,20.2,992.4,V56/485
2025-01-24,10:21:04,T,1,82,235,0,20.2,992.4,V56/485
2025-01-24,10:21:05,T,2,231,1075,1,20.2,992.4,V56/485
2025-01-24,10:21:05,B,2,92,1193,6,20.2,992.4,V56/485
2025-01-24,10:21:05,B,3,275,1346,6,20.2,992.4,V56/485
2025-01-24,10:21:05,B,4,94,1398,6,20.2,992.4,V56/485
2025-01-24,10:21:05,B,5,120,1415,6,20.2,992.4,V56/485
2025-01-24,10:21:05,T,3,138,1473,1,20.2,992.4,V56/485
2025-01-24,10:21:05,B,6,603,1632,6,20.2,992.4,V56/485
2025-01-24,10:21:05,B,7,112,2045,6,20.2,992.4,V56/485
```

6) The csv file can be opened in any data processing software for the user to manipulate as necessary. Here is the same data file open in a spreadsheet:

	A	B	C	D	E	F	G	H	I	J
1	##									
2	### UKRAA: The Pico Muon Detector ###									
3	### S/W version: 0.1.23 ###									
4	### Detector name: V56/485 ###									
5	### Thresholds: T(200	100)	B(200	100)	###					
6	##									
7	Date	Time	ID	Event	ADC(0-1023)	ElapsedTime(mS)	DeadTime(mS)	Temperature(C)	Pressure(hPa)	DetectorName
8	24/01/2025	10:21:04	B	1	251	87	1	20.2	992.4	V56/485
9	24/01/2025	10:21:04	T	1	82	235	0	20.2	992.4	V56/485
10	24/01/2025	10:21:05	T	2	231	1075	1	20.2	992.4	V56/485
11	24/01/2025	10:21:05	B	2	92	1193	6	20.2	992.4	V56/485
12	24/01/2025	10:21:05	B	3	275	1346	6	20.2	992.4	V56/485
13	24/01/2025	10:21:05	B	4	94	1398	6	20.2	992.4	V56/485
14	24/01/2025	10:21:05	B	5	120	1415	6	20.2	992.4	V56/485
15	24/01/2025	10:21:05	T	3	138	1473	1	20.2	992.4	V56/485
16	24/01/2025	10:21:05	B	6	603	1632	6	20.2	992.4	V56/485
17	24/01/2025	10:21:05	B	7	112	2045	6	20.2	992.4	V56/485

Event data – serial output

1) The PicoMuon detector provides event data via the serial interface. The format of the data is similar to that recorded to the microSD card, except there is no header and the single line that identifies the parameters that are recorded does not include date or time, as it is anticipated that the computer that the PicoMuon detector is connected to will provide suitable date and time.

ID,Event,ADC(0-1023),ElapsedTime(mS),DeadTime(mS),Temperature(C),Pressure(hPa),DetectorName

2) The user can thus utilise a simple script to record and store the event data from the PicoMuon detector and then analyse the data as they wish.

3) Below is an example of a simple Python script, to record and store the event data from the PicoMuon detector, run on a RPi4 single board computer.

```
# imports
import serial
from datetime import datetime, timezone

# definitions
def main():
    # set up usb serial variables
    ser = serial.Serial(port = '/dev/ttyACM0',
                        baudrate = 115200,
                        bytesize = serial.EIGHTBITS,
                        parity = serial.PARITY_NONE,
                        stopbits = serial.STOPBITS_ONE)

    for line in ser:
        # define output file to write data to
        outfile = open('/home/pi/UKRAA_PicoMuon/data.txt', 'a')
        # if data, write to file
        if line:
            # date and time
            timetowrite = (datetime.strftime(datetime.now(timezone.utc), '%Y-%m-%d')
                           + " "
                           + datetime.strftime(datetime.now(timezone.utc), '%H:%M:%S.%f'))
            # data received from PicoMuon detector
            texttowrite = (line.decode('utf-8', 'ignore').strip())

            print(timetowrite + ',' + texttowrite, file = outfile)

# main
if __name__ == "__main__":
    main()
```


4) The data stored from the above script has the following format:

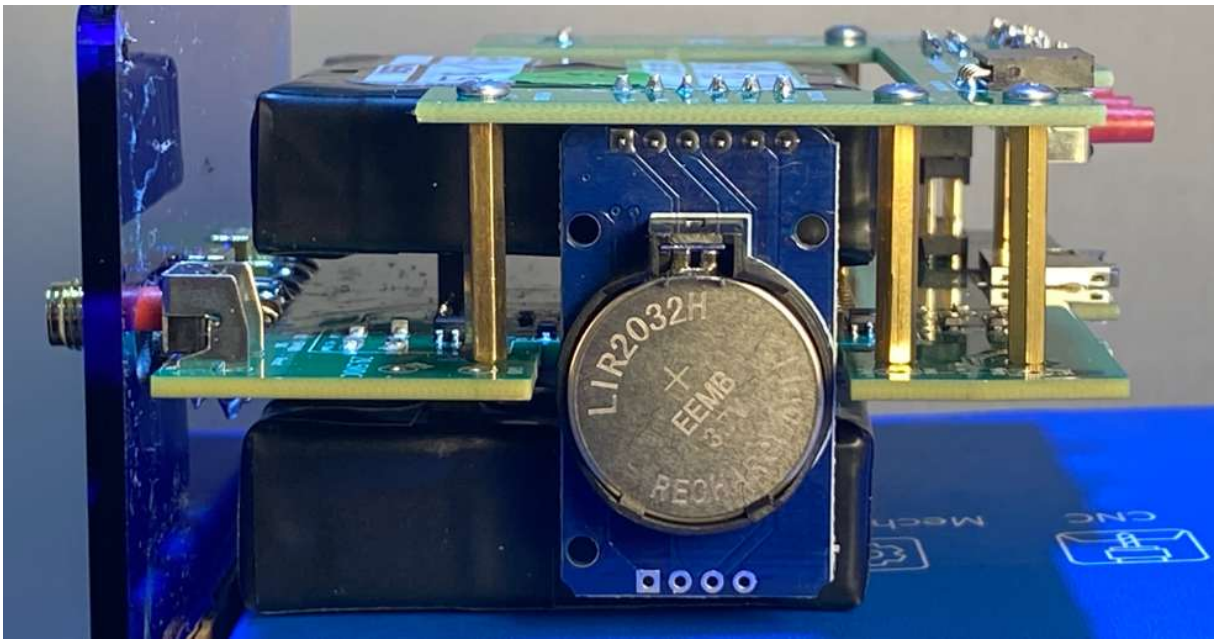
```
2025-01-24 12:16:04.418779,T,15217,54,6901909,8232,20.6,993.5,V56/485
2025-01-24 12:16:04.419031,B,21260,116,6901959,19132,20.6,993.5,V56/485
2025-01-24 12:16:04.419245,B,21261,70,6902080,19132,20.6,993.5,V56/485
2025-01-24 12:16:04.419610,B,21262,51,6902395,19132,20.6,993.5,V56/485
2025-01-24 12:16:05.427845,B,21263,239,6903081,19133,20.6,993.5,V56/485
2025-01-24 12:16:05.428655,T,15218,66,6903615,8232,20.6,993.5,V56/485
2025-01-24 12:16:06.438108,B,21264,60,6903921,19137,20.6,993.5,V56/485
2025-01-24 12:16:06.438560,T,15219,420,6903935,8235,20.6,993.5,V56/485
2025-01-24 12:16:06.438830,T,15220,60,6904168,8235,20.6,993.5,V56/485
2025-01-24 12:16:06.439119,B,21265,276,6904326,19137,20.6,993.5,V56/485
2025-01-24 12:16:06.439362,T,15221,433,6904416,8235,20.6,993.5,V56/485
2025-01-24 12:16:06.439585,B,21266,579,6904518,19137,20.6,993.5,V56/485
2025-01-24 12:16:06.439784,T,15222,444,6904518,8235,20.6,993.5,V56/485
2025-01-24 12:16:06.439990,B,21267,253,6904795,19137,20.6,993.5,V56/485
2025-01-24 12:16:07.447902,T,15223,205,6904929,8236,20.6,993.5,V56/485
2025-01-24 12:16:07.448119,T,15224,100,6905097,8236,20.6,993.5,V56/485
2025-01-24 12:16:07.448314,B,21268,350,6905329,19138,20.6,993.5,V56/485
2025-01-24 12:16:07.448723,T,15225,416,6905410,8236,20.6,993.5,V56/485
2025-01-24 12:16:08.456954,B,21269,297,6905980,19140,20.6,993.5,V56/485
2025-01-24 12:16:08.457178,B,21270,139,6906534,19140,20.6,993.5,V56/485
2025-01-24 12:16:08.457792,B,21271,50,6906921,19140,20.6,993.5,V56/485
2025-01-24 12:16:09.466057,T,15226,115,6907005,8238,20.6,993.5,V56/485
2025-01-24 12:16:09.466957,T,15227,138,6907141,8238,20.6,993.5,V56/485
```

5) A full set of Python code is available, from the UKRAA GitHub page, that will get, process and present data from the PicoMuon detector onto your smart phone at home using a RPi4 or RPi5.

https://github.com/UKradioastro/UKRAA_PicoMuon

Replace RTC button cell

1) Should you need to remove/replace the button cell from the DS3231 RTC, please contact UKRAA for advice/instruction for resetting the DS3231RTC date/time following button cell replacement.
2) Should the CR2032 button cell need to be replaced on the DS3231 RTC module, then the following actions need to be taken.
3) Unplug the PicoMuon detector from any external power source.
4) Remove all x8 case screws, x4 from the front panel and x4 from the rear panel. It is not necessary to remove the x3 ¼" sma nuts and washers – the rear panel can remain attached to the PicoMuon main PCB.
5) Pull the front panel forward to disengage the OLED display from the upper OLED PCB – careful not to bend the pins on the 1x4 header of the OLED display secured to the front panel.
6) The upper aluminium case can be removed and the PicoMuon PCB assembly removed from the lower case.
7) Set the PicoMuon PCB assembly on a flat surface with the CR2032 button cell visible, as shown in the picture below.



8) To remove the CR2032 button cell – pull back on the upper spring clip of the button cell holder. This will release the button cell from the button cell holder, and remove.
9) Insert a fresh CR2032 button cell into the button cell holder from the lower black clip, push towards the top of the button cell until it clicks in behind the upper metal spring clip. **Ensure that +ve is facing outwards.**

10) The date/time will now need to be reset on the DS3231 RTC – contact UKRAA for instruction on completing this before progressing further – see Contact details at end of manual.

11) Replace the PicoMuon PCB assembly into the lower aluminium case and then replace the upper aluminium case

12) Align the 1x4 head of the OLED display with the 1x4 socket on the upper OLED PCB and push home, again being careful to correctly engage the 1x4 header with the upper OLED PCB.

13) Replace all x8 case screws: x4 from the front panel and x4 from the rear panel

14) Plug the PicoMuon detector into suitable external power source.

15) Battery replaced, real time clock date/time updated – good to go.

Install new release of software

1) Like any provided software, there will be certain bugs that were not captured during pre-release and initial-release testing – these are free!
2) Once software bug issues have been raised through the GitHub repository for this product https://github.com/Ukradioastro/CpicoMuon, and addressed, a new release of software will be issued. This will be in the form of a .UF2 file. All users for whom UKRAA has suitable contact details, will be made aware of this new software release. The user is then able to update their software, should they wish to do so.
3) Other software issues that are raised through the GitHub repository for this product https://github.com/Ukradioastro/CpicoMuon, and are identified as enhancements may not be expedited into the next release of code – but will be reviewed and included if deemed appropriate.
4) To install new release of software, the user has to have access to the **BOOT SEL** button, located on the RPi Pico seen in the upper left of the picture below.

5) To gain access to the RPi Pico **BOOT SEL** button, remove the x2 front pane lower aluminium case screws and the x2 rear panel lower aluminium case screws. Then remove the lower aluminium case from the PicoMuon detector.

6) On a suitable PC, download the latest release of software for the PicoMuon detector.

7) Connect the PicoMuon detector to this PC using the supplied USB cable with the PicoMuon detector.

8) While holding the RPi Pico **BOOT SEL** button down also press the **reset** button located on the rear panel of the PicoMuon detector.

9) Now release the **Reset** button, while still holding the RPi Pico **BOOT SEL** button down.

10) Now release the RPi Pico **BOOT SEL** button.

11) You will now be able to access the RPi Pico as a mass storage device on your PC.

12) Open a file explorer of the RPi Pico

13) Open a file explorer where you downloaded the new .UF2 release and drag this .UF2 file to the RPi Pico file explorer. The code will be automatically installed and the file explorer should close. This implies successful installation of new software to the PicoMuon detector.

14) Note, the new release will not impact the stored user configuration data held by the AT24C32 EEPROM – unless explicitly informed in the release notes associated with the new .UF2.

15) Replace the lower aluminium case of the PicoMuon detector. Replace the x2 front panel lower case screws and the x2 rear panel lower aluminium case screws.

16) Upgrade complete – good to go.

Error codes and meaning.

The PicoMuon detector has a series of error codes that are displayed on the OLED display when there is a fault condition. Below is a table of error codes, cause and remedy.

Error code	Name	Cause	Remedy
ERR:00	ERROR_NONE	N/A	No action necessary.
ERR:01	ERROR_DS3231_READ	DS3231 read error – unable to read the external RTC.	Replace 2032 button cell and reset time.
ERR:02	ERROR_QUEUE_OVERLOW	Internal debug message. Memory structure used to buffer up the events coming from the high speed event capture routine has overflown.	Should never occur. Reboot if does.
ERR:03	ERROR_SD_CARD	SD Card interface failed to initialise.	Try different SD card. Reboot.
ERR:04	ERROR_SD_MOUNT	Failed to mount file system on SD card.	Check SD card file format.
ERR:05	ERROR_SD_FILE	Unable to create or access file on SD card.	Check SD card contents.
ERR:06	ERROR_SD_SPACE	Free space on SD card has dropped below 5.0 MB.	Free up space on SD card or replace with new one.
ERR:07	ERROR_RING_BUFF	Internal debug message indicating a ring buffer overflow.	Should never occur. Reboot if does.
ERR:08	ERROR_NO_CONFIG	Not used.	Should never occur.

Error code	Name	Cause	Remedy
ERR:09	ERROR_BAD_CONFIG	Invalid parameter/value detected in user configuration.	Check configuration content and fix.
ERR:10	ERROR_BAD_PRESSURE	Internal debug message. Invalid temperature and or pressure value detected.	Should never occur. Reboot if does.
ERR:21	ERROR_DEBUG_EVENTS_STOPPED	Internal debug message. Event capture has stopped.	Should never occur. Reboot if does.

Typical Oscilloscope Traces

1) Using the three sma connectors from the rear panel of the PicoMuon detector you have access to the raw upper and lower event pulses, range in value from 0V to 3V3 (**threshold_t** and **threshold_b** set at 0.161V), that are read by the RPi Pico ADC, and to the coincidence pulse (0V to 3V3) generated by the RPI Pico when it identifies a coincidence event – this is a modulated pulse with direct connection to the coincidence LED.

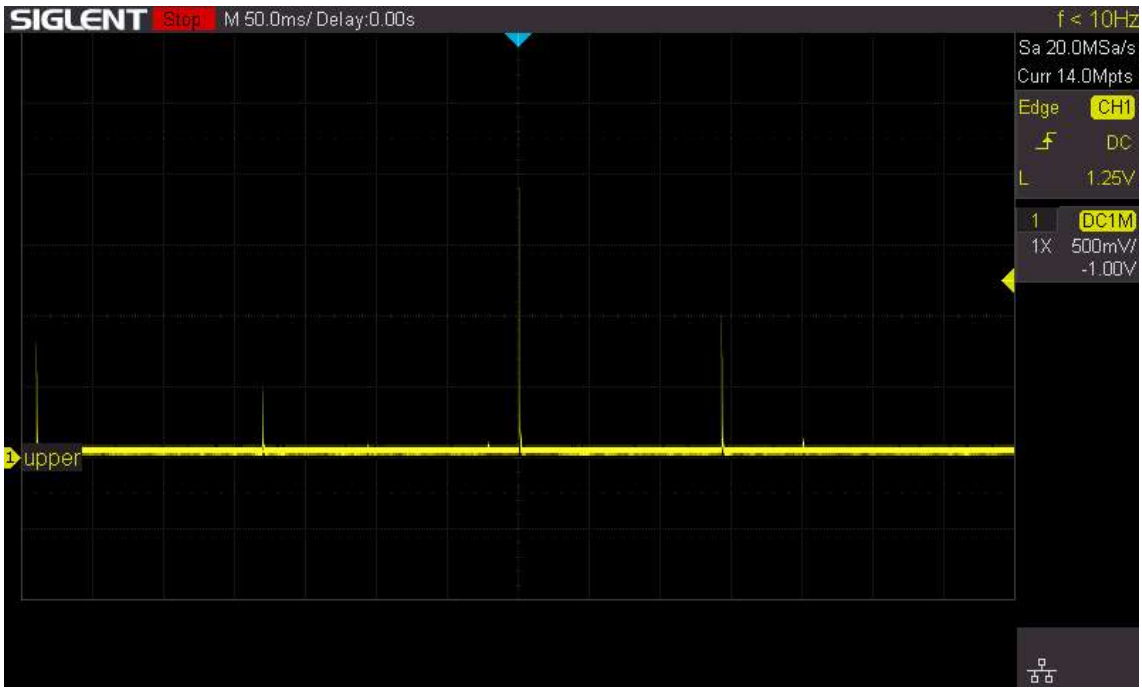
2) The figure below shows a typical pulse trace for the upper SiPM signal following an event. Details of scope settings are included in the figure.

3) It can be seen in the figure above that the event pulse is approximately 8 x 200μs length – 1.6s.

4) The figure below shows a typical pulse trace for the lower SiPM signal following an event. Again details of the scope settings are included in the figure.

5) Again, it can be seen in the figure above that the event pulse is approximately 8 x 200μs length – 1.6s.

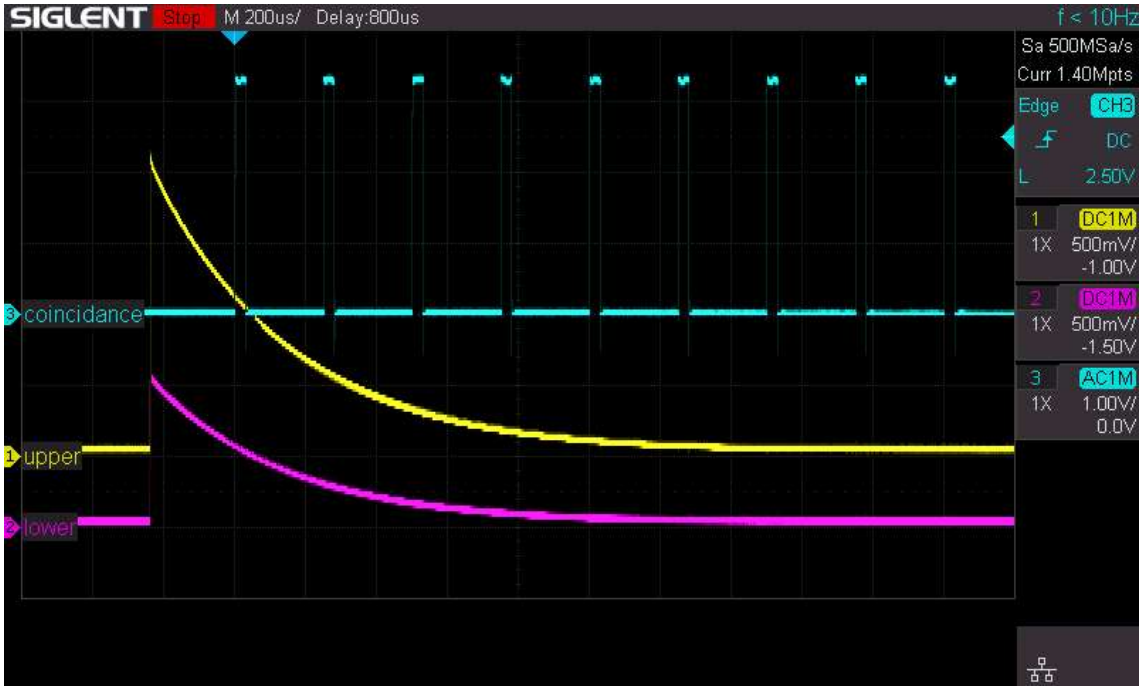
6) The following scope trace shows how the use of **threshold_t** and **threshold_b** is used to discriminate the recording of “noise” events recorded by the SiPMs.



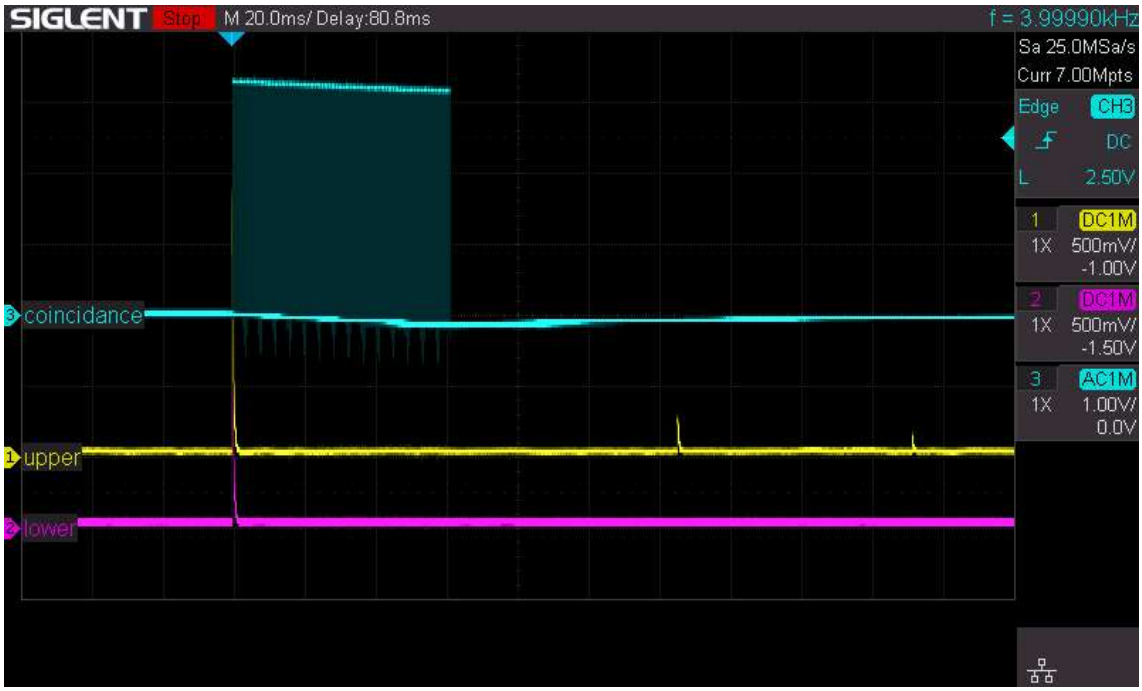
7) The trigger threshold of the scope traces above was set at 1.25V and it can be seen that the first two pulses on the left of the figure did not trigger the oscilloscope, these would not be recorded as events as they are below the threshold. It was the central pulse that triggered the oscilloscope.

8) It can also be seen that the amplitude of the two pulses on the right of the figure would equally have not triggered the oscilloscope, again, these would not be recorded as event as then are below the trigger threshold.

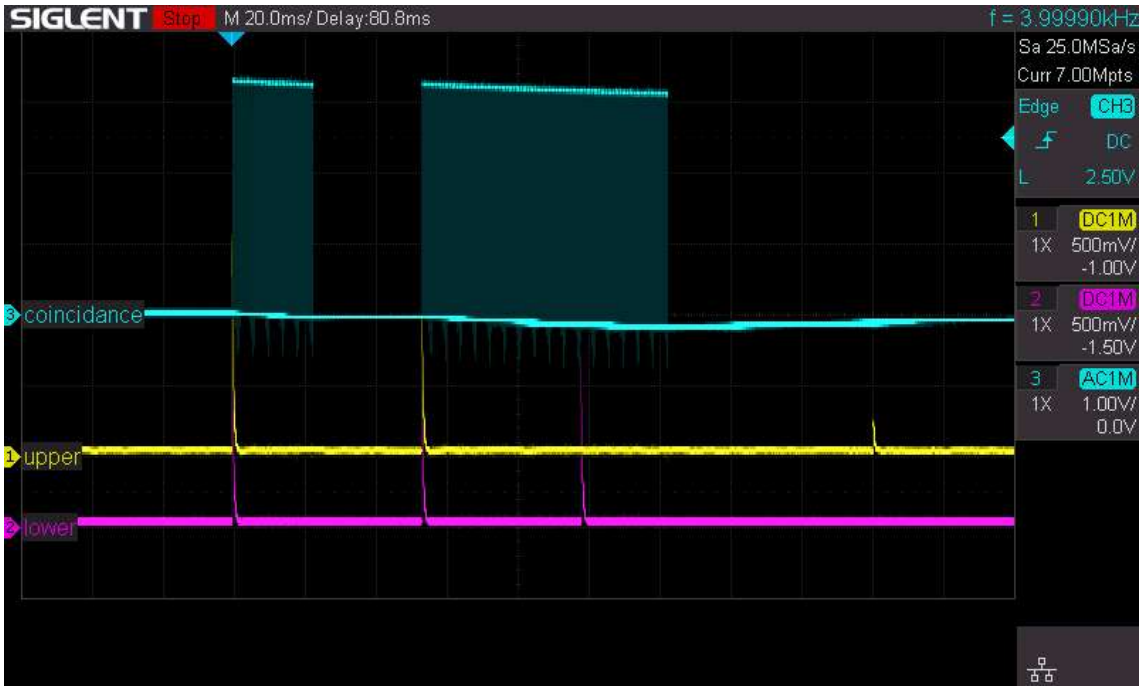
9) The scope trace below illustrates a coincidence event – note that there is about 200µs between the upper/lower pulse event registration and the output of the coincidence sma going high. This is a pulsed high/low output (PWM) as it also drives the front coincidence LED.



10) The scope trace below shows that the coincidence sma output is pulsed high/low for approximately 60ms (from 0V to 3V3) following a coincidence event.



11) The scope trace below records two coincidence events within a period of about 50ms.



12) There is a third event recorded by the lower SiPM, at about 100ms from the first event, but no event on the upper SiPM recorded at the same time – hence not a coincidence event.

13) There is also a third event shown of the upper SiPM trace, at about 180ms from the first event, but this is below the threshold value and would not be recorded by the PicoMuon detector as an event as it is below the threshold level.

Contacts

The UK Radio Astronomy Association

83 Cornwallis Avenue
TONBRIDGE
Kent
TN10 4LL

A charitable incorporated organisation

Registered Charity in England and Wales No. 1123866.

E-mail: picomuon@ukraa.com

Website: ukraa.com

BAA Radio Astronomy Group

Website: BAA Radio Astronomy

Muon detector forum

Website: muondetector@groups.io

Appendix 1 – PicoMuon detector Specifications

Basic information	
Size:	104mm x 85mm x 60mm
Mass:	350g
Power requirements	
Type:	Via micro USB to RPi Pico board
Voltage:	5V DC
Current peak:	150mA
Current nominal:	70mA
Scintillator	
Material:	BC408
Emission λmax:	425nm
Size:	50mm x 50mm x 10mm (nominal)
Rise time:	0.9ns
Decay time:	2.1ns
Detector	
Type:	Silicon PhotoMultiplier (SiPM)
Size:	6mm x 6mm
Peak λ response:	425nm
Processor	
Model:	Raspberry Pi Pico
Core:	RP2040 microcontroller
Speed:	Up to 133MHz
SRAM:	264KB
Flash memory:	2MB

Display	
Type:	OLED
Size:	1.5”
Pixels:	128×128
Drive:	sh1107
Interface:	I2C
Environment monitoring	
Type:	BMP280 pressure sensor
Interface:	I2C
Pressure range:	950hPa to 1050hPa @ 25^{0}C
Temperature range:	-40^{0}C to +85^{0}C
Real Time clock	
Type:	DS3231 I2C-integrated RTC
Interface:	I2C
Accuracy:	About 30 minutes per year
EEPROM	
Type:	AT24C32 2-wire Serial EEPROM
Interface:	I2C
Capacity:	4kB @128 pages of 32 bytes
SD card slot	
Type:	microSD card size
Interface:	SPI
Capacity:	Tested to 64GB

Appendix 2 – Regulatory Compliance

RoHS

In Great Britain and Northern Ireland, the Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment Regulations 2012 (as amended) are the underpinning legislation. The Restriction of Hazardous Substances Directive or RoHS regulations restrict of the use of the following ten hazardous substances in electronic and electrical equipment products falling within the Directive:

- Lead
- Mercury
- Cadmium
- Hexavalent chromium
- Polybrominated biphenyls
- Polybrominated diphenyl ethers
- Bis(2-ethylhexyl) phthalate (DEHP)
- Butyl benzyl phthalate (BBP)
- Dibutyl phthalate (DBP)
- Diisobutyl phthalate (DIBP)

UKRAA confirms that the suppliers of the components and materials used in the UKRAA PicoMuon detector have stated that such components and materials are RoHS compliant and that reasonable steps have been taken to confirm these statements.

WEEE

Electrical and electronic equipment (EEE) is regulated to reduce the amount of waste electrical and electronic equipment (WEEE) incinerated or sent to landfill sites. Reduction is achieved through various measures which encourage the recovery, reuse and recycling of products and components. The Waste Electrical and Electronic Equipment Regulations 2013 (as amended) is the underpinning UK legislation. (gov.uk website)

If a customer purchases a new product from UKRAA which falls within the WEEE Directive to replace an existing one (of similar function to the one that has been sold) and intends to dispose of the existing one, then the customer can request that we take back the existing product and recycle it. Any customer wishing to take advantage of this facility should contact us.

You may also use alternative local WEEE recycling provision. Please do not place WEE in the general waste.

Revision History

Revision	Date	Author	Status
Draft A	2025-01-18	R Knott	Internal draft for peer review
Draft B	2025-01-30	R Knott	Draft following first peer review
Draft B1	2025-01-31	R Knott	Specification done
Draft B2	2025-03-09	R Knott	Error codes done + Display Off – Appendix 2

Outstanding work

A charitable incorporated organisation

Registered charity in England and Wales No. 1123866.