Launch Master

Rocket Launch System

Reference Manual

207-05_001_LaunchMaster_reference

Document History

date	author	version	details
5-9-22	C.Strudwicke	0.1	Initial creation
15-11-22	C.Strudwicke	0.2	Addition of new command line items Addition of firmware update process
20-11-22	C.Strudwicke	0.3	Addition of Calibration process

Contents

1 Table of Figures

2 Table of tables

3 Overview

The Launch Master system is comprised of a Launch Controller and one or more Pad slaves.

Figure 1 Launch Master system, Launch Controller (left), Pad slave board (right)

They communicate via a wireless protocol utilizing LoRa in the 915 to 928MHz band (for Australian Models, band varies for other regions).

A Pad Slave may be configured as a 4 channel Solid motor station or a Hybrid motor station.

3.1 Features

- Launch Controller
    - High contrast sunlight readable graphical LCD screen
    - Intuitive operation, navigation via rotary encoder pushbutton
    - Key switch arming
    - Configuration via built-in GUI
    - Light and compact
    - Long range reliable communications
    - Local and remote battery voltage monitoring
    - Diagnostics screen
    - Number of Pad Slaves - 8
- Pad Slave
    - 4x pyro outputs with
        - short cct, over temperature and current limiting protection
        - Continuity measurement
    - Loadcell/strain gauge input for rocket mass
    - 2x 0,4-20mA inputs for Pressure and Temperature
    - 2 x digital inputs
    - RS-485 communications port (connection to HiPerDaq module)

3.2 System description

A Launch Controller (master) currently supports communicating with a single Pad Slave at a time, however it may be configured to communicate with up to 8 different Pad Slaves.

The wireless communication link between the master and slave utilises the LoRa protocol giving reliable and long range performance. The master polls the configured slave every 300ms, in the process sending the current commands (ie generated by the buttons and other user input hardware) and the response from the slave containing the current status of the slave such as continuity of the 4 channels, battery voltage, confirmation of current commands, analog inputs and loadcell reading.

3.2.1 System diagram

WIP

Figure 2 System diagram – Single Pad arrangement

4 Launch Controller

WIP

Figure 3 Launch Controller block diagram

4.1 Front Panel

Figure 4 Front panel layout (need a better picture)

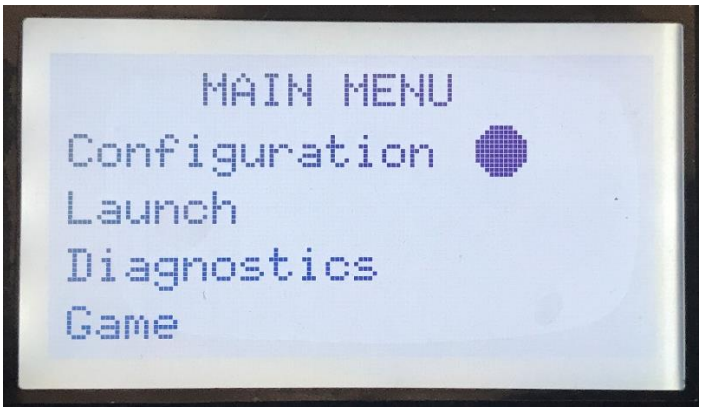
1. Monochrome 128x64 graphical LCD with adjustable backlight
2. Navigation rotary encoder button (rotation and press button)
3. Fault Lamp (red)
4. Pyro fire button
5. Power button (latching)
6. Arming key switch
7. Charge port
8. Fill button
9. Detach/Dump button
10. Status Lamp (green)

4.2 Menus & screen navigation

The navigation button is used to select the screen (from the main menu screen) by turning the button until the desired screen is selected (large circle) then pressing the button.

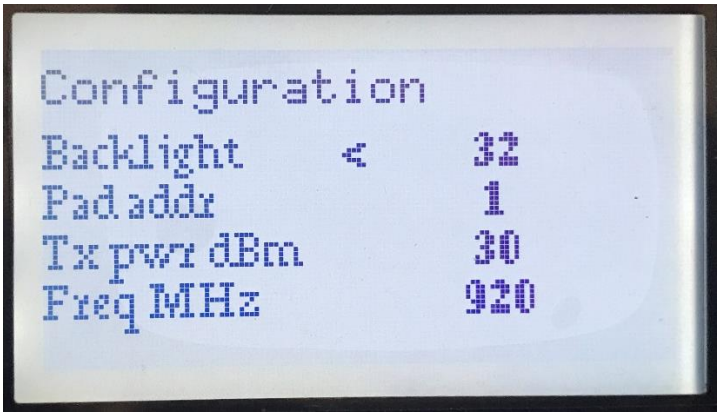
When on a sub-screen and you wish to move to another, a long press will take you back to the main screen allowing you to navigate to another.

4.2.1 Main Menu



4.3 Configuration

Typical configuration operations may be performed using the Configuration screen, accessed from the main menu.



Parameters available for edit: Backlight intensity, Pad Slave address, LC Tx Power, LC Channel (frequency)

1. Backlight intensity – adjustable from 0 to 255
2. Pad Slave address – must be set to pad of interest, between 1 and 8
3. Tx Power – adjustable from 21 to 30dBm
4. Channel Frequency – adjustable from 915 to 928 MHz in 1Mhz increments

4.4 Launch Control

This is the screen used to perform the intended function of the device.

The displayed items are self explanatory, however the symbols next to them are not.

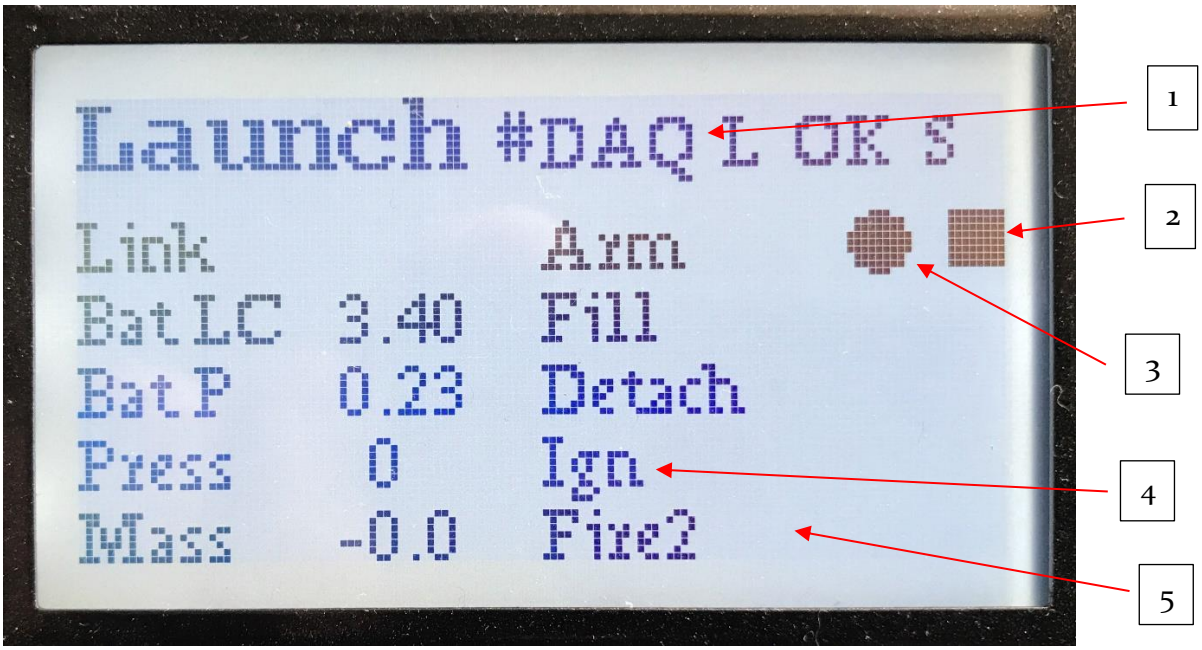


Figure 5 - Launch Screen - DAQ connected

1. If the string 'DAQ ' is shown, this indicates the HiPerDaq is present and communicating over the RS485 port
2. Square indicator – Pad Slave response – square present indicates function is implemented
3. Solid circle – Launch Controller input detected and being sent to Pad Slave
4. **Bold text** – Indicates selection – Mass & Ign1, Ign2 can be moved between using the navigation button, then pressing this selects the item
5. () indicator – Indication or continuity for that channel

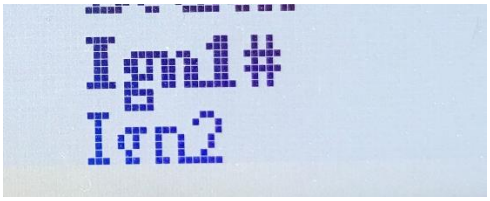


Figure 6 - Ign1 selected

4.4.1 Ignition channel selection

Rotation of the navigation button allows selection between Mass, Ign1 and Ign2 and if connected DAQ (the HiPerDaq module).

Select the Ignition channel desired then choose by short press of Navigation button. The selection will become bold and when the Fire button is pressed this is the channel that will be driven.

4.4.2 Mass Tare

Select the Mass item using the Navigation button, then tare by using a short press. The Mass should now read zero.

4.4.3 DAQ logging control

If the DAQ (HiPerDaq) is shown in the top right, it is connected and communicating. This means that it may be commanded to start logging from the Launch Controller.

Rotate the navigation button until the # indicator is shown next to DAQ, then using a short press of the navigation button, a start command is sent to the HiPerDaq. The execution status will change from **S** to **R** if successful.

Figure 7 - DAQ status - connected, non-logging mode, error

Figure 8 - DAQ connection status - logging mode, OK and state = Standby

4.5 Diagnostics

The diagnostic screen is useful for determining the quality of the wireless link.

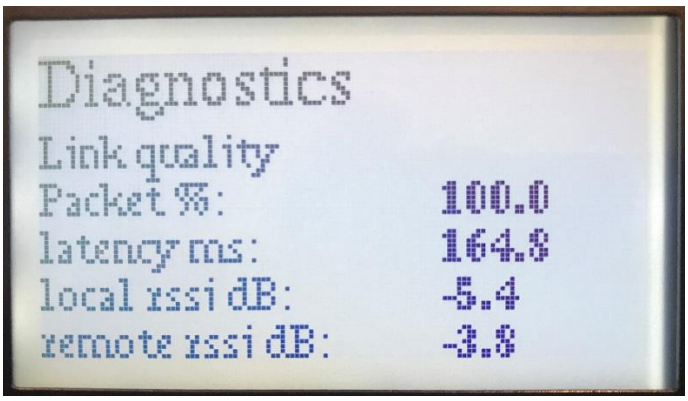


Figure 9 Diagnostics screen

The metrics available are:

1. % Successful packets: Simple percentage of packets sent from Launch controller that are responded to successfully by the current Pad slave

2. Latency: time required for Pad slave to respond (ms)
3. Local rssi: Signal level received by Launch controller from Pad slave units dBm (uncalibrated)
4. Remote rssi: Signal level received by Pad slave from Launch controller dBm (uncalibrated)

4.6 Detailed configuration

The system has numerous of configuration parameters to control its operation.

These parameters are set using the Command Line Interface (section 4).

	WIP		Units
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

5 Pad Slave

The Pad Slave is physically located near the launch pad(s) and is connected wirelessly to the Launch controller. It's power must be provided locally.

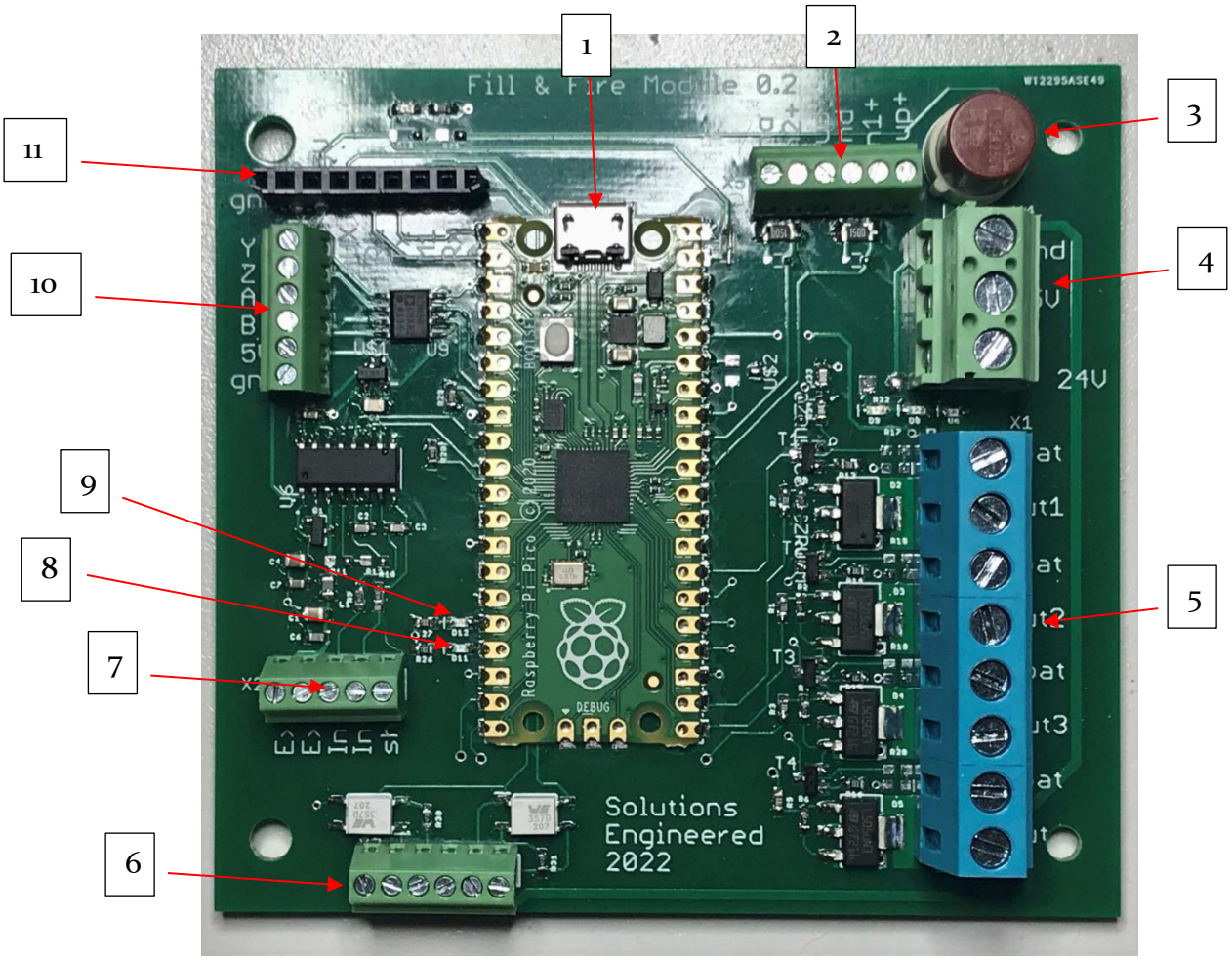


Figure 10 – Pad Slave Board

1. Raspberry Pi-PICO – USB port
2. X5 - Analog input terminals – 2 channels 0 to 20mA
3. I/O supply fuse
4. X4 - Supply terminal block
5. X1- Output terminal block – 4ch current limited
6. X?- Digital input terminal block – 2ch partial isolation
7. X2 - Strain gauge input terminal block – 4 wire
8. RED status LED
9. GREEN status LED
10. X6 - RS-485 communication port – for connection to HiPerDaq module
11. X3 - Radio connection header

5.1.1.1 Terminal pinouts

5.1.1.1.1 X1 – Output terminal block (hybrid configuration)

1	Ch1 +supply	Fill
2	Ch1 output	
3	Ch2 +supply	Detach/ Dump
4	Ch2 output	
5	Ch3 +supply	Ign1
6	Ch3 output	
7	Ch4 +supply	Ign2
8	Ch4 output	

Table 1 - Pad Slave Output Terminal block (hybrid config)

5.1.1.1.2 X1 – Output terminal block (4x Ignitor configuration)

1	Ch1 +supply	Ign1
2	Ch1 output	
3	Ch2 +supply	Ign2
4	Ch2 output	
5	Ch3 +supply	Ign3
6	Ch3 output	
7	Ch4 +supply	Ign4
8	Ch4 output	

Table 2 - Pad Slave Output Terminal block (4x ignitor configuration)

5.1.1.1.3 X2 – Load cell terminal block

1	Excitation +	Red
2	Excitation -	Black
3	Signal +	Green
4	Signal -	White
5	Shield	

Table 3 - Pad Slave Load cell terminal block

5.1.1.1.4 X3 – Radio header

1		
2		
3		
4		
5		
6		
7		

Table 4 - Pad Slave radio connection header

5.1.1.1.5 X4 – Power supply

1	GND	
2	+5V	1A
3	+24V	10A

Table 5 - Pad Slave Power supply terminal block

5.1.1.1.6 X? – Digital inputs

1	Ch1 +24V sensor supply	
2	Ch1 signal	
3	Ch1 ground	
4	Ch2 +24V sensor supply	
5	Ch2 signal	
6	Ch2 ground	

Table 6 - Pad Slave digital inputs terminal block

5.1.1.1.7 X? – Analog inputs

1	Ch1 Supply +	+24V
2	Ch1 Input	0 to 20mA
3	Ch1 Ground	0V
4	Ch2 Supply +	+24V
5	Ch2 Input	0 to 20mA
6	Ch2 Ground	0V

Table 7 - Pad Slave analog input terminals

5.1.1.1.8 X? – RS485 serial port

1	GND		
2	5V		
3	B	TX-	Connect to Z
4	A	TX+	Connect to Y
5	Z	RX-	Connect to B
6	Y	RX+	Connect to A

Table 8 - Pad Slave RS485 terminal block

5.1.2 Example connection

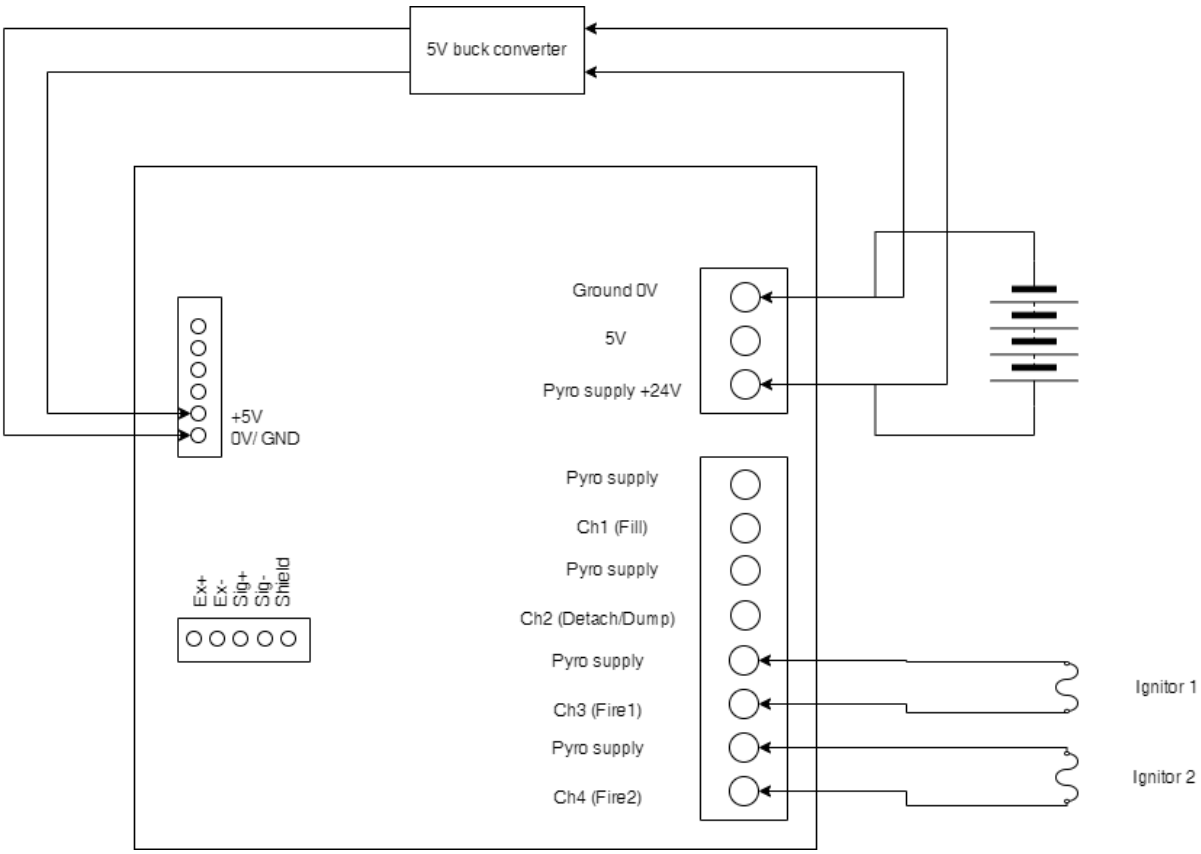


Figure 11 basic Pad Slave wiring – hybrid configuration

5.2 Configuration

The system has numerous of configuration parameters to control its operation.

These parameters are set using the Command Line Interface (section 6).

			Units	Default
1	LoRa Channel	Offset from base frequency of 850 MHz	MHz	70
2	Pad Address	Value indicating unique address from 1 to 8	none	1
3	Tx Power	Transmitter power in dBm Valid 21, 24, 27, 30	dBm	30
4	LC Scale	Gain calibration value for loadcell amplifier	Counts/kg	
5	PT scale	Pressure transducer (analog input 1) scale	Counts/psi	
6	PT zero offset*	Pressure transducer zero offset	counts	
7				
8				
9				
10				

Table 9 - Pad slave parameter table

*Not yet implemented

5.3 Commands supported

5.3.1 cmd

cmd<sp>arm<cr> sets the pad to armed

cmd<sp>disarm<cr> disarms the pad

cmd<sp>fire<sp>x<cr> fire channel x

cmd<sp>detach<cr> drives the detach channel

cmd<sp>fill<cr> toggles the fill channel (this is a latching output)

cmd<sp>status<cr> returns a comma separated list of all values

cmd<sp>debug<cr> toggles the state of debug

cmd<sp>tare<cr> tares the loadcell locally

5.3.2 keep alive

Once armed, a keep alive command must be sent at least once per second or a timeout will occur , the Pad slave will be disarmed and all outputs will be turned off.

ka<cr>

5.3.3 rspDaq

This is to support responses from a HiPerDaq module.

rspDaq<sp>mode<sp>log<cr> identifies that the HiPerDaq is in logging mode

rspDaq<sp>status<sp><mode<sp><state><sp><overall status><cr>

5.3.4 status

status<sp>json<cr> return the complete status in json format

status<cr> return complete status in csv format

5.4 Example CLI interactions

5.4.1 Parameter dump (pd)

WIP

6 Command Line Interface (CLI)

A Command Line Interface (from hereon to be called the CLI) is accessible through the USB debug port on both devices. The CLI enables configuration and debugging.

6.1 Commands

The first command to learn is the <?> as this brings up the full list of available commands:

Format: ?<CR>

6.1.1 ??

System status dump

Format: ??<CR>

6.1.2 pd

Parameter list dump triggers all parameters to be listed

Format: pd<CR>

6.1.3 ps

Save all parameters to eeprom/flash

Format: ps<CR>

6.1.4 set x y

Set parameter <x> to value <y>.

Format: set<sp><x><sp><y><CR>

There are some parameters that have multiple arguments such as date – time which has 7 arguments that are separated by a space.

6.1.5 get x

returns the value of parameter <x>

format: get<sp><param#><CR>

6.1.6 cmd x

Sends a command to the main state machine.

Format: cmd<sp><command><CR>

Currently the command list is:

- b (begin) – used to manually start the process for a given mode of operation
- e (end) – used to manually end the process underway

- debug - toggles the state of the debug parameter. When set, numerous debug statements are printed

6.1.7 ***

Force a processor reboot

Format: ***<CR>

6.1.8 Calibration

A number of calibration operations can be implemented with the cal command.

- cal<sp>c<sp>x<cr> : set calibration channel to x (0 to 3) & reset the calibration metrics
- cal<sp>r<cr> : reset the calibration metrics
- cal<sp>z<cr> : measure the channel raw reading over 5s period and use the average as the new zero offset. The channel inputs should be shorted during this process
- cal<sp>p1<sp>x<cr> : take the first of 2 calibration point readings. x is the value of real units at this point, being measured by an external device
- cal<sp>check<cr> : used to generate the new calibration values without writing them to the configuration
- cal<sp>complete<cr> : complete the calibration process for this channel. The offsets and slope are calculated and applied to the channel settings. These are not automatically saved to NVM, the ps command should be used to do this.

7 Firmware update

7.1 Launch Controller

This is an ESP32 based system and requires the following flash download software from Espressif to perform the update:

https://www.espressif.com/sites/default/files/tools/flash_download_tool_3.9.3.zip

The USB connector on the ESP32 devkit is a micro B type. Connect the ESP32 to your PC at this time.

After expanding this download, execute the file ***flash_download_tool_3.93.exe***

This dialog will appear:

Launch Master

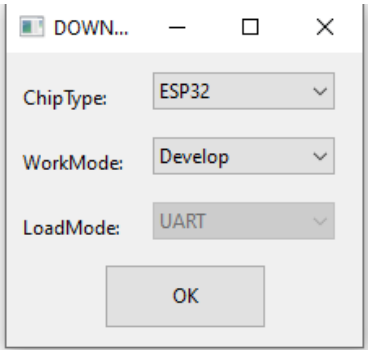


Figure 12 Espressif Systems Flash download tool - initial dialog

Select ChipType: ESP32, ensure work mode is Develop and LoadMode will be UART.

Click OK.

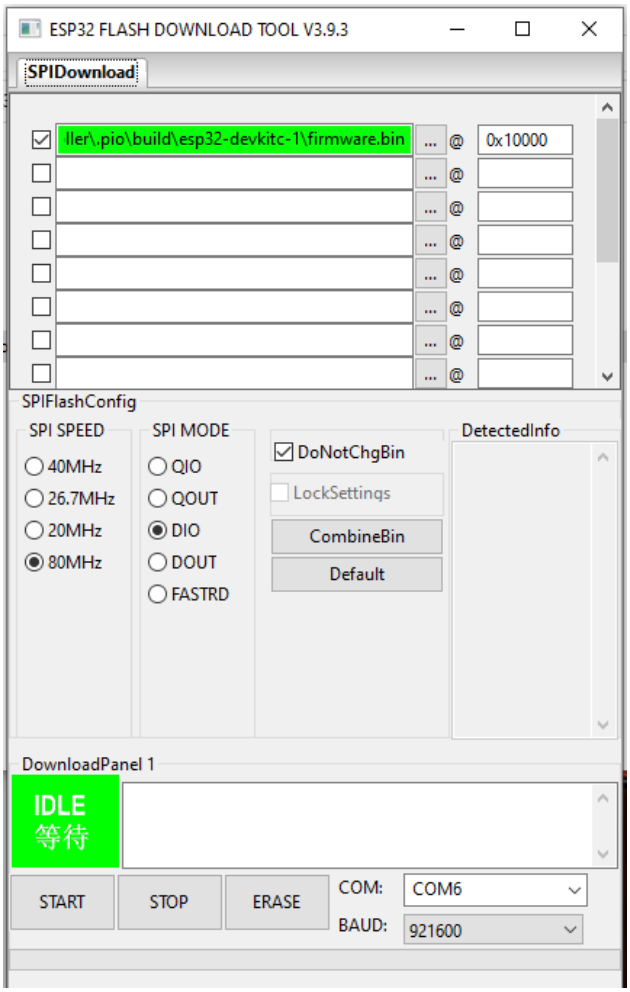


Figure 13 Espressif Download tool - main window

Launch Master

In row 1 select the path for the .bin file to be downloaded and set the address to 0x10000. Tick the tickbox to the left.

This will only change the application code for the device, which is fine since we are not aiming to change the bootloader or any other data.

Ensure the other settings are as shown above, except the COM port must be set to that for your particular case.

Once done with these settings, press START (bottom left) to begin.

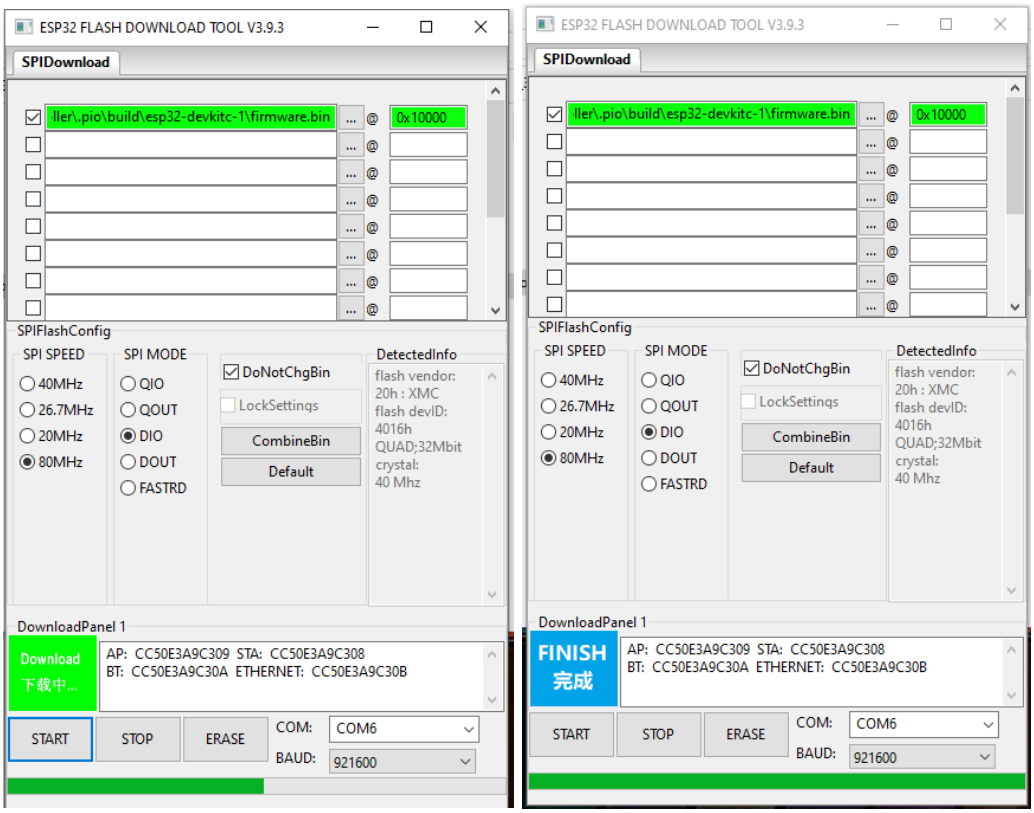


Figure 14 Progress of Espressif flash tool during writing

Once you see Finish, the process is complete.

It is possible the device will not automatically reset even after closing the Flash tool.

You can either remove the USB cable and power cycle the device OR open up a terminal program which should force it to reset.

7.2 Pad Slave

Plug the Raspberry PI Pico board into a free USB port of the host computer using a micro USB cable.

NOTE: As the PICO is mounted to a board without relief around the USB connector, most connectors will not engage without modification (ie removal of plastic on the underside of the over-molded connector).

After plugging the Pico board into the host computer, the board appears as a Flash drive. If it does not, then unplug the board from the USB port. Hold down the BOOTSEL switch found on top of the board and plug the board back into the USB port. After three seconds, release the BOOTSEL switch.

Copy the firmware file (filename.uf2) onto the drive. It will disappear and the Pad slave will start running.

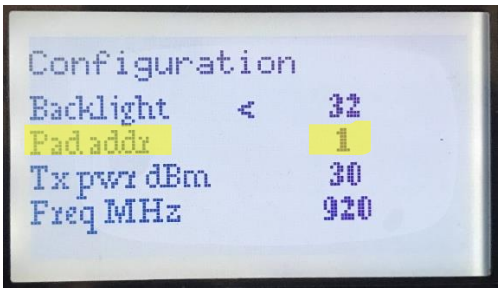
The firmware is now updated.

8 Quick start guide

1. Install antenna onto Launch controller

Photo required

2. Locate Pad Slave, install antenna & power up without ignitors connected
3. Power up Launch controller and ensure Pad ID is set to same as Pad slave of interest.



4. Open Diagnostics screen on Launch Controller
    a. Ensure that Packet success > 95%

Photo required

5. Select Launch Control screen
6. Confirm LC battery at least 3.7V at start of session (safe to operate down to 3.0V)
7. Confirm Pad battery at least 12V

Photo required

8. Power down Pad slave
9. Connect ignitor(s)
10. Move away from rocket and then power up Pad slave
11. Move back to behind launch control line
12. Arm Launch controller
13. Confirm continuity

Photo required

14. Fire after countdown and ensuring safety