Blue Clover Devices

System Manual

PLT

0.7.1

2020-05-09

Revision History

Revision History	
Revision 0.7.1	2020-05-09
• PLT Signals: Clarify routed vs direct probes	
Revision 0.7.0	2020-04-28
• PLT Signals: Typo fix	
Revision 0.6.9	2020-03-30
• Include PLT Connectors • Report connector: JSON / CSV	
Revision 0.6.5	2019-11-06
No change	
Revision 0.6.4	2019-10-08
No change	
Revision 0.6.3	2019-10-08
• CI/CD Connector: GitHub Actions example	
Revision 0.6.2	2019-10-03
• Update target in YAML example • Expand CI/CD Connector documentation • Debug Mode: Clarify need for Administrator role	
Revision 0.6.1	2019-09-19
• Typo fixes	
Revision 0.6.0	2019-09-09
• Document CI/CD Connector • Document Report Connector	
Revision 0.5.8	2019-08-26
• No Change	
Revision 0.5.7	2019-08-22

- No Change

Revision 0.5.6 2019-08-19

- No Change

Revision 0.5.5 2019-06-02

- Debug keys
- Update screenshots

Revision 0.5.1 2019-04-15

- Document PLT-200A Probes and Signals

Revision 0.5.0 2019-04-01

- No Change

Revision 0.4.7 2019-03-08

- Split off Test Suite Reference

Revision 0.4.5 2019-02-21

- Test Suite Reference: Use PLT-100A UART port name

Revision 0.4.4 2019-02-18

- Operation Guide: update PLT screenshots

Revision 0.4.3 2019-02-14

- Layout updates
- Update Project References
- Operation Guide: Clarify 10-pin, 14-pin and HD78 DUT connections

Revision 0.4.2 2019-02-13

- Update PLTcloud User Manual

Revision 0.4.1 2019-02-06

- Add PLT Operation Guide
- Split off Assembly instructions

Revision 0.4.0 2019-02-05
Update to incorporate PLT
Revision 0.1.0 2016-05-12
Initial draft

Table of Contents

Preface

Document containing instructions for using the PLT system to deploy and execute test suites.

1. PLTcloud User Manual

1.1 Introduction

This document describes the features of the PLT cloud with its different sections. As an example, it focuses on the “IoTSDK Demo” project, that aims to program the ***STM32 Discovery kit IoT node*** with a firmware image integrating the BLE protocol and test it.

1.2 Home

Use the Home page to navigate easily across the different sections of the PLTcloud website.

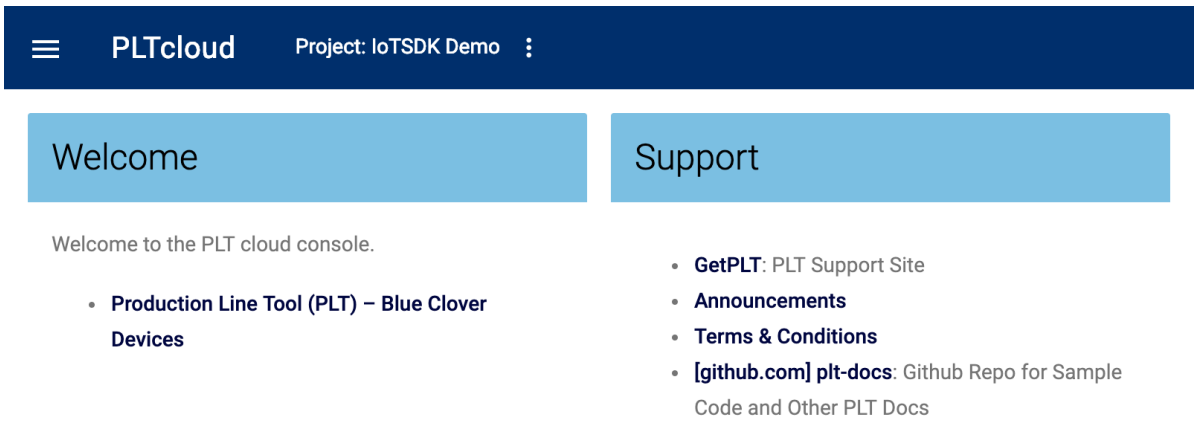

Figure 1.1. Home screen

1.3 PLTs

PLTs are grouped in “Deployment Groups”. These are groups of PLTs that are meant to run the same set of tests. For example, you can have a group of PLTs that you will want to use for In-Circuit (ICT) testing and another group of PLTs that you will want to use for Final Assembly, Test & Pack-out (FATP) testing.

New Deployment Groups can be created in the “PLTs” view.

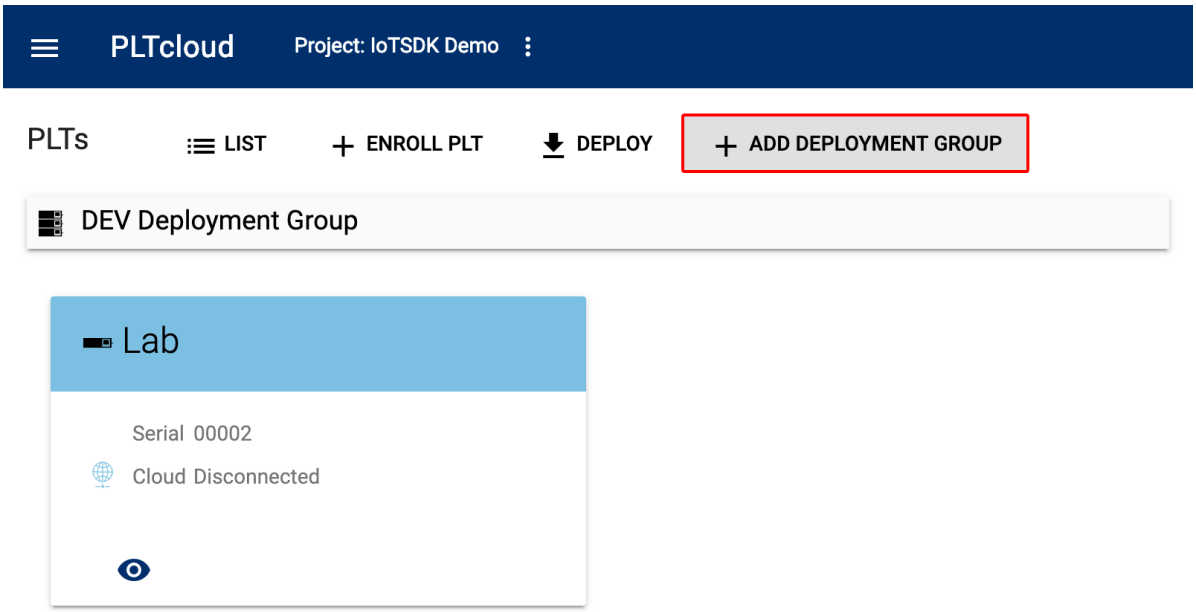

Figure 1.2. PLTs, grouped in Deployment Groups

You can then fill out the name and the description of the group.

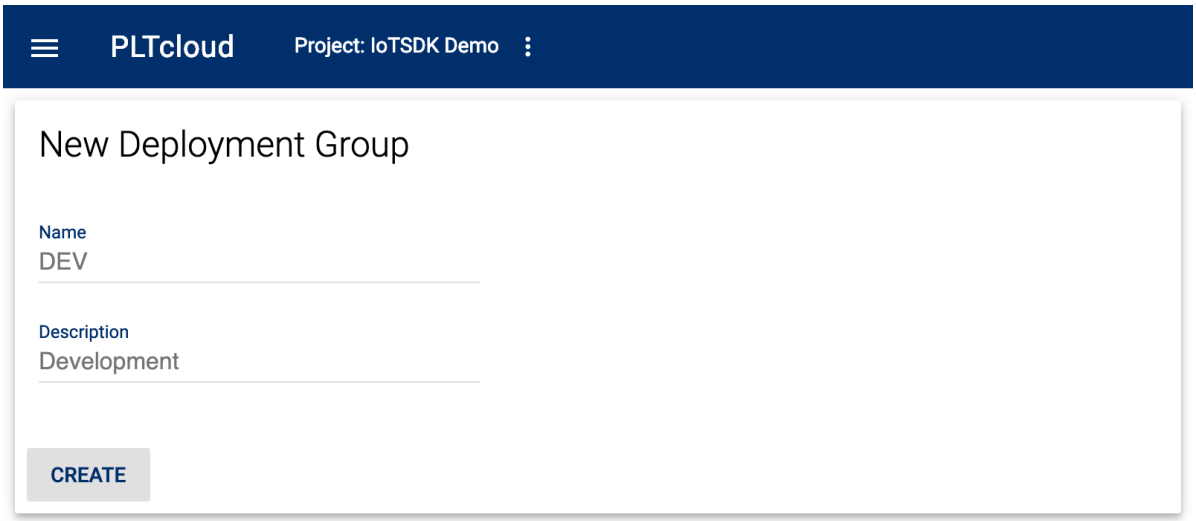

Figure 1.3. New Deployment Group

From the “PLTs” view, detailed information on a PLT can be seen by clicking on the dark-blue eye icon.

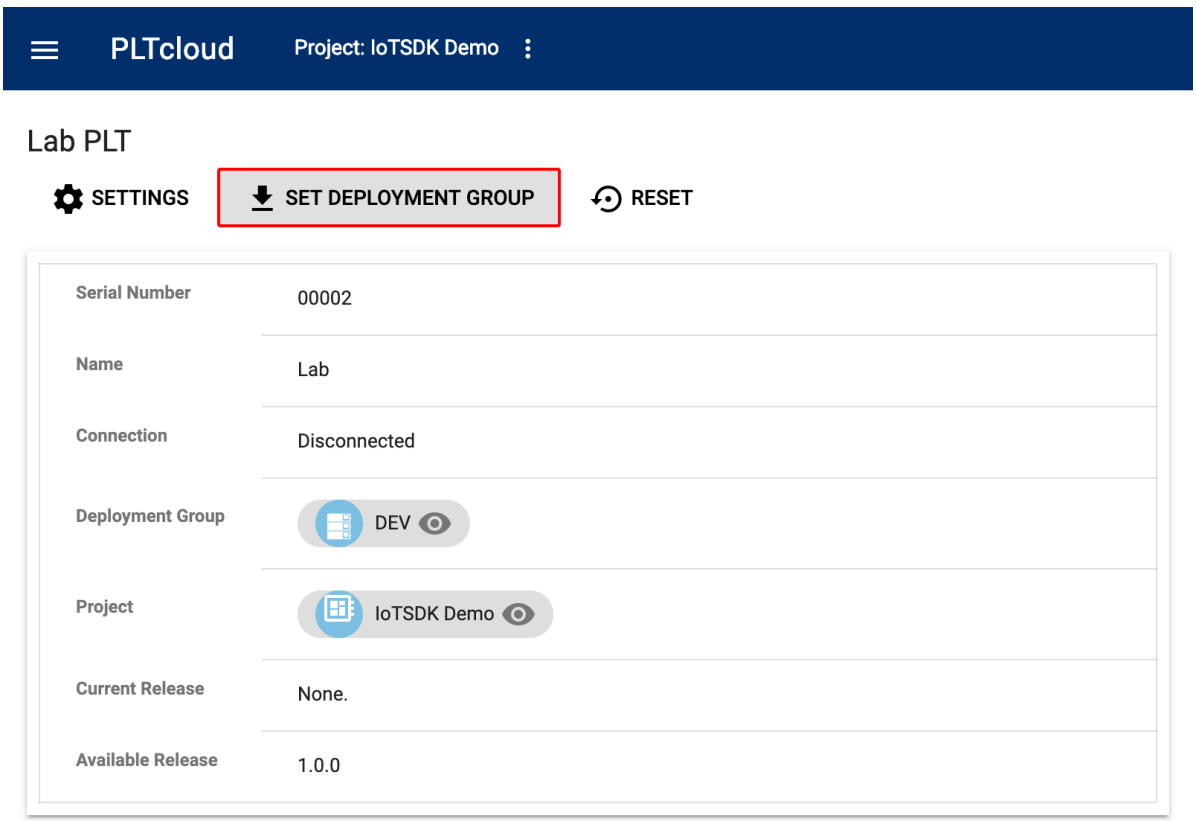

Figure 1.4. PLT Detail View

In the case of the “STM IoT SDK” project, we are deploying to PLTs that are part of the “DEV” group.

1.4 Adding a PLT

On the PLT UI panel, press the Menu button to enter the PLT Menu, and select Enroll.

The PLT will show an enrollment code on the OLED display.

Figure 1.5. Enrollment Code on PLT OLED display

You can enroll your PLT under the “PLTs” section of the PLTcloud website. On the PLTcloud website, use this code as the Activation code in the “Enroll PLT” form.

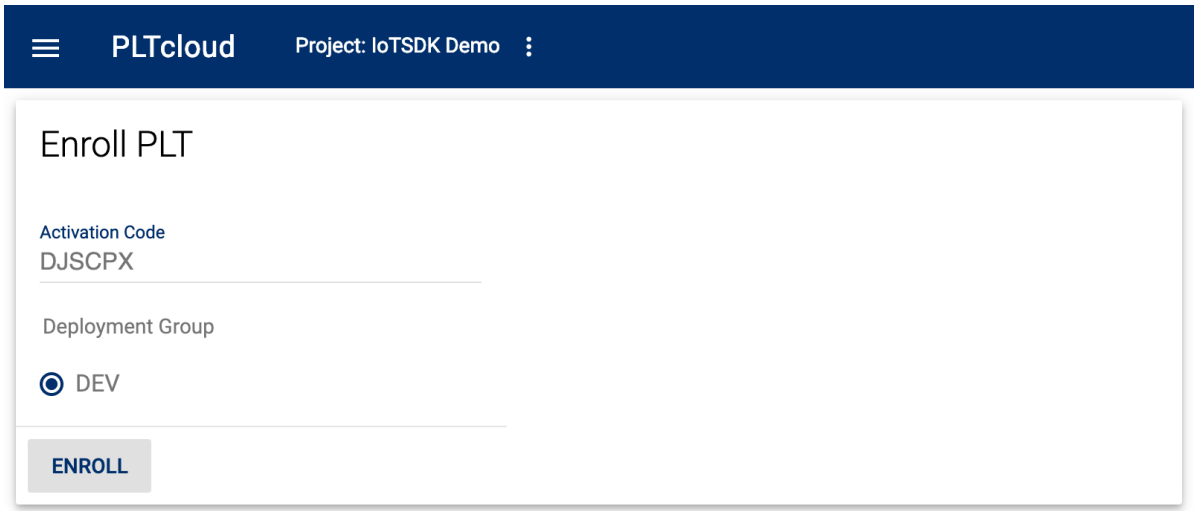

Figure 1.6. PLT Enrollment Form

You can then set the PLT to a specific deployment group by clicking on on “SET DEPLOYMENT GROUP”.

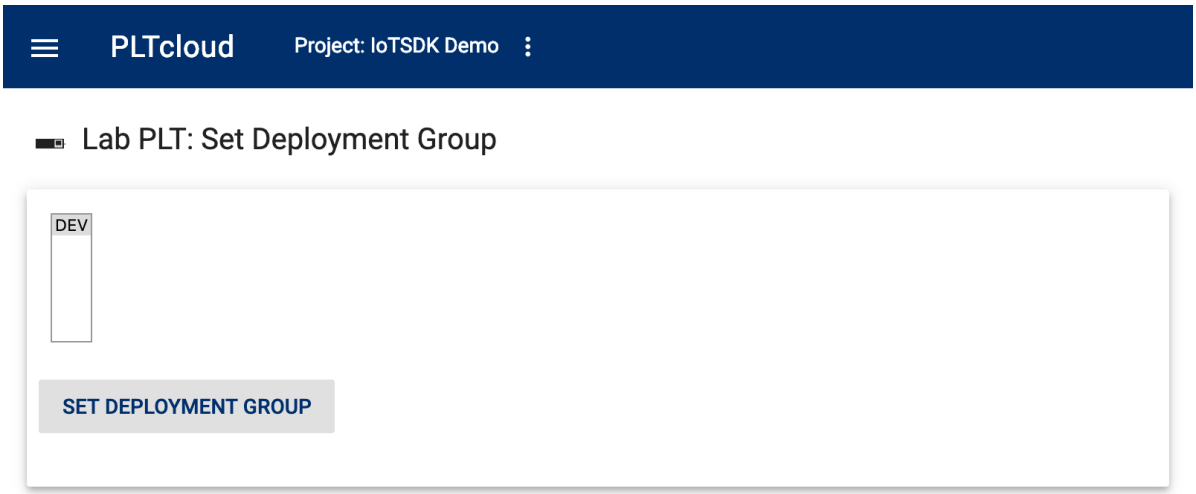

Figure 1.7. Setting Deployment Group for a PLT

For the “STM IoT SDK” project, we set to the “DEV” deployment group.

1.5 PLT Views

You can see a list of all PLTs by pressing “LIST” from the “PLTs” view:

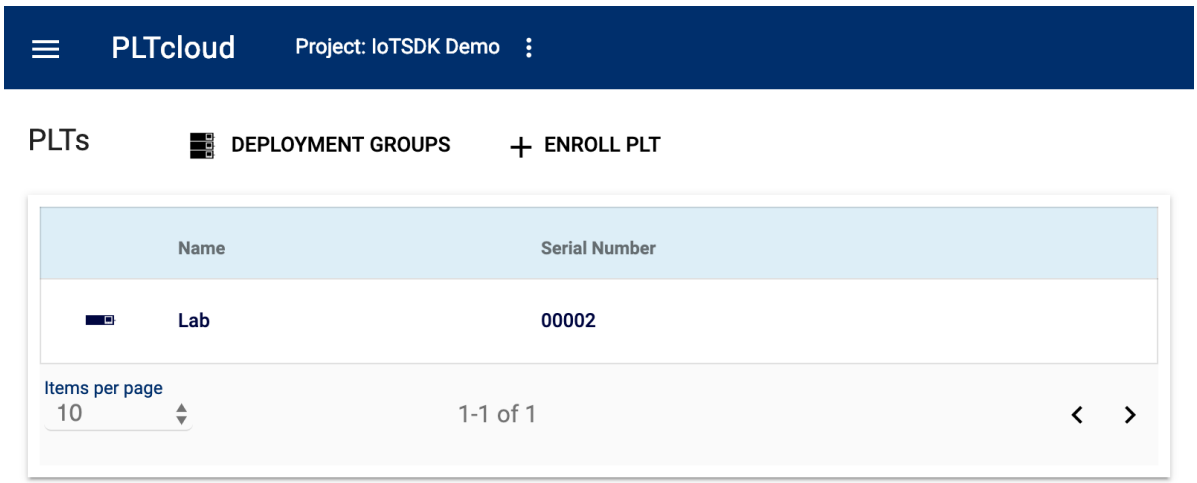

Figure 1.8. PLT list view

Or in grouped by deployment group by clicking on on “DEPLOYMENT GROUPS”.

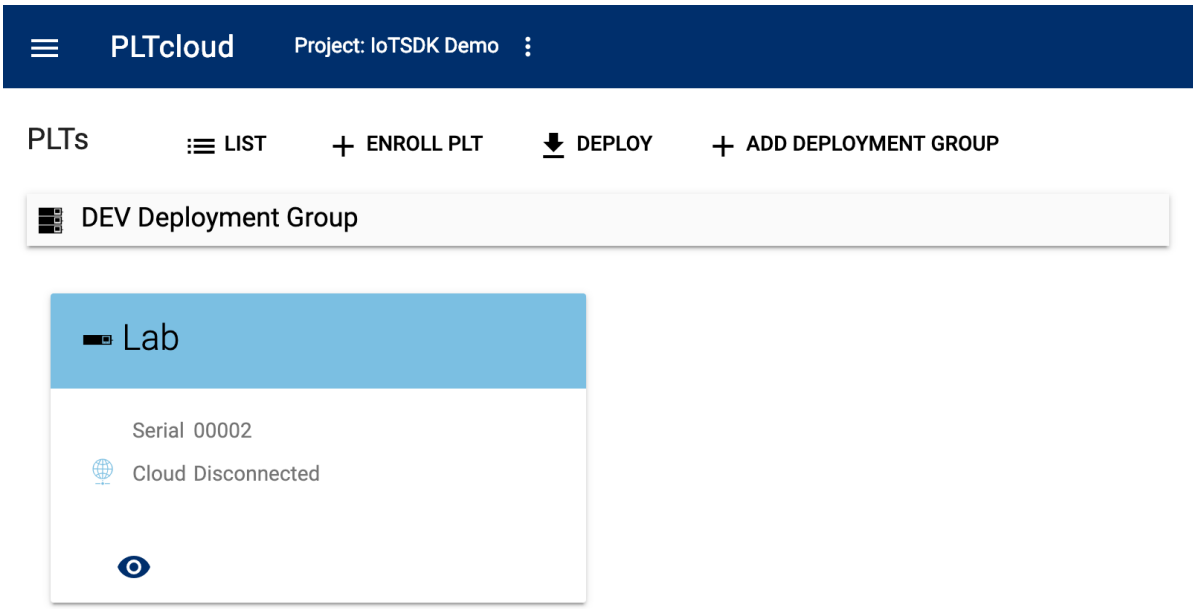

Figure 1.9. PLTs, grouped by Deployment Group

1.6 Create Project

You can create a project or go through the list of projects already created by clicking on the 3 dots in the top app bar:

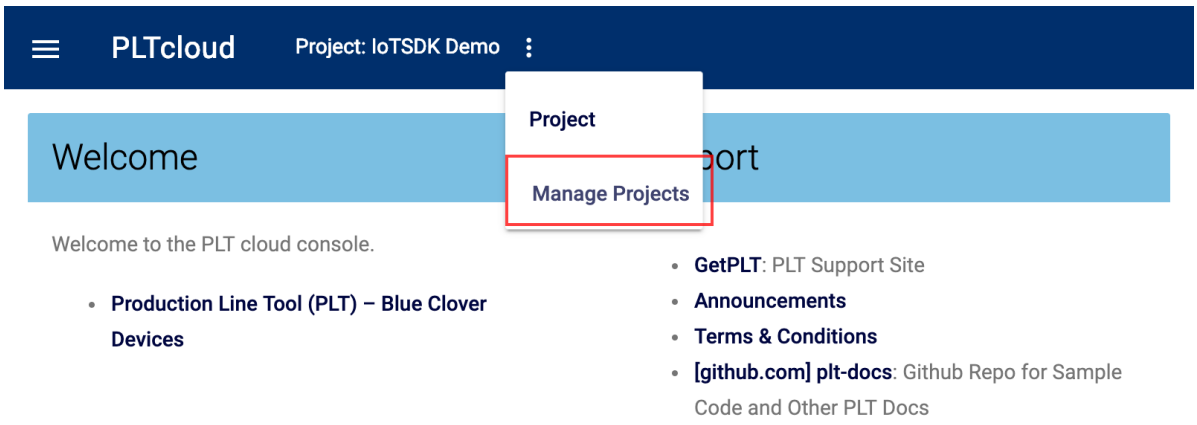

Figure 1.10. Project Menu

The list of projects allows you to switch between them and to select the project you want to work on in PLTcloud. In our case, we selected the “IoTSDK Demo” project.

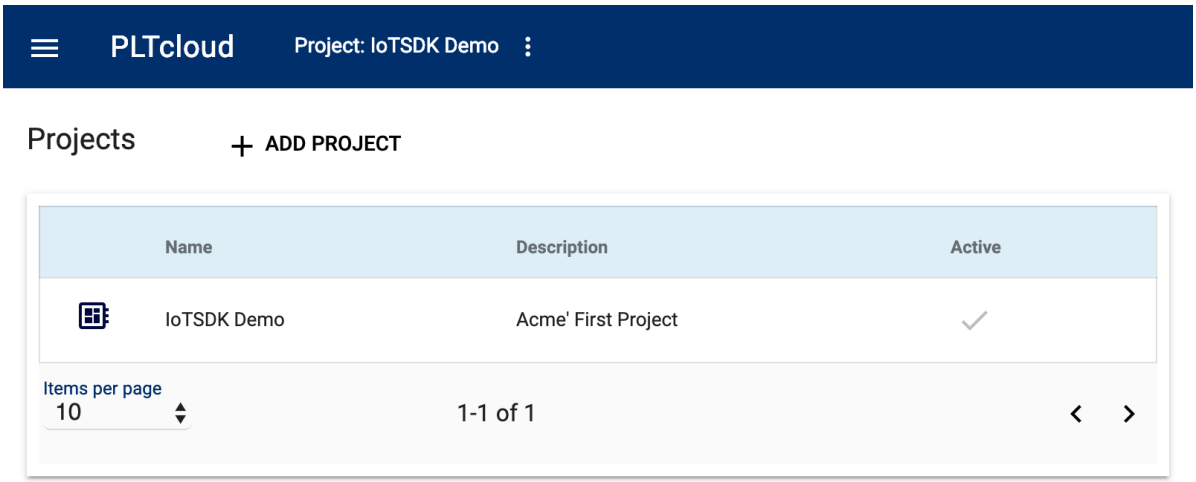

Figure 1.11. Project List

1.7 Project Elements

You can add elements to a project.

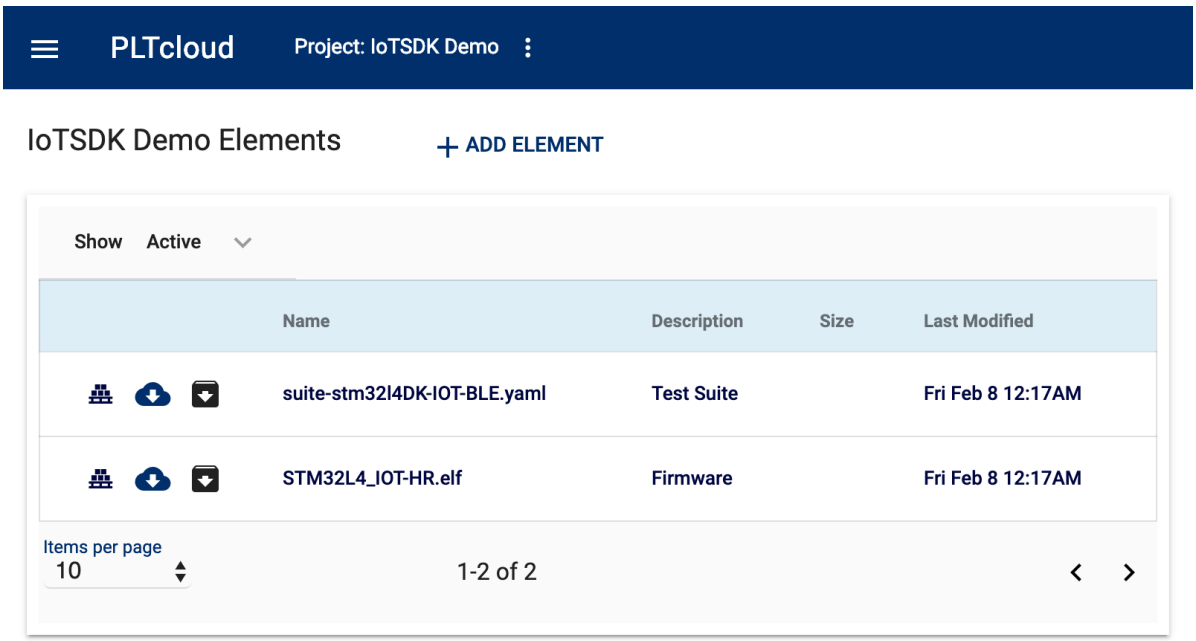

Figure 1.12. Element List

Elements that you might want to add to a project can, for example, be firmware images (`.hex`, `.elf`, `.bin`), or test suite definition files (`.yaml`).

To test and program the STM IoT demo board, we need to upload the firmware and the test plan. The firmware can be uploaded as an ELF file, for example, `STM32L4_IOT_HR.elf`. The test plan is supplied as a YAML file, for example, `suite-stm32l4DK-IOT-BLE.yaml`:

```
title: "LY10-PLT demo: stm32 DK IoT node"
suite:
  - ident: ICT-T1
    title: Erase stm32l4_DevKit with USB ST-Link
    steps:
      - command: erase stm32l4_STLink
  - ident: ICT-T2
    title: Program STM32L4-IOT-BLE stm32l4_DevKit with USB ST-Link
    steps:
      - command: program stm32l4_STLink none,STM32L4_IOT_HR.elf,none
  - ident: ICT-T3
    title: Identify DUT
    steps:
      - command: identify stm32l4_IoT_STLink
  - ident: ICT-T4
    title: BLE discovery
    steps:
      - command: bledis %BLEMAC% 30  # Wait up to 30 seconds for BLE discovery to
complete
```

1.8 Project Releases

You can create releases for your projects.

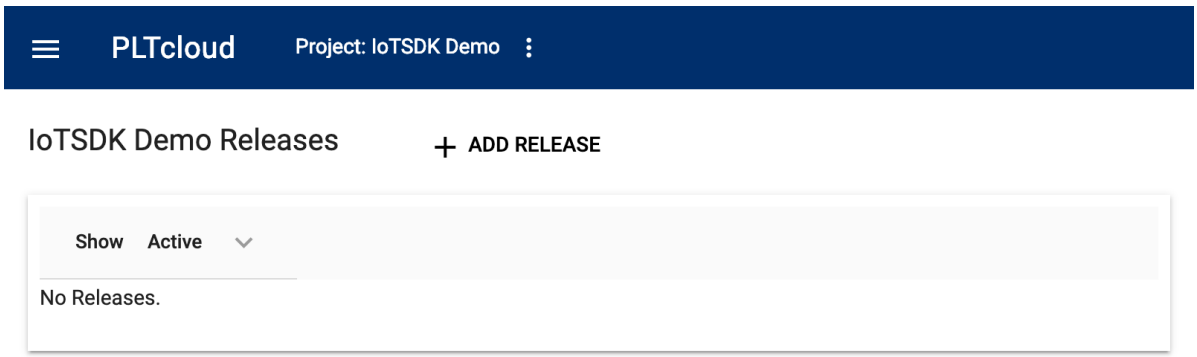

Figure 1.13. Release List (empty)

The elements that you will be able to add to a new release are elements that you will have previously added through the “Elements” section. You can create multiple releases per project.

For example, to create a new release, you would fill the version and description of cases of your release, and select the elements (firmware images and YAML files) you want to be part of your release.

In our case, we need to create a release of the firmware image (`STM32L4_IOT_HR.elf`) and of the YAML file (`suite-stm32l4DK-IOT-BLE.yaml`) for the “IoTSDK Demo” project. These two elements are therefore checked when creating the release that we named “STM32L4 IoT node BLE”.

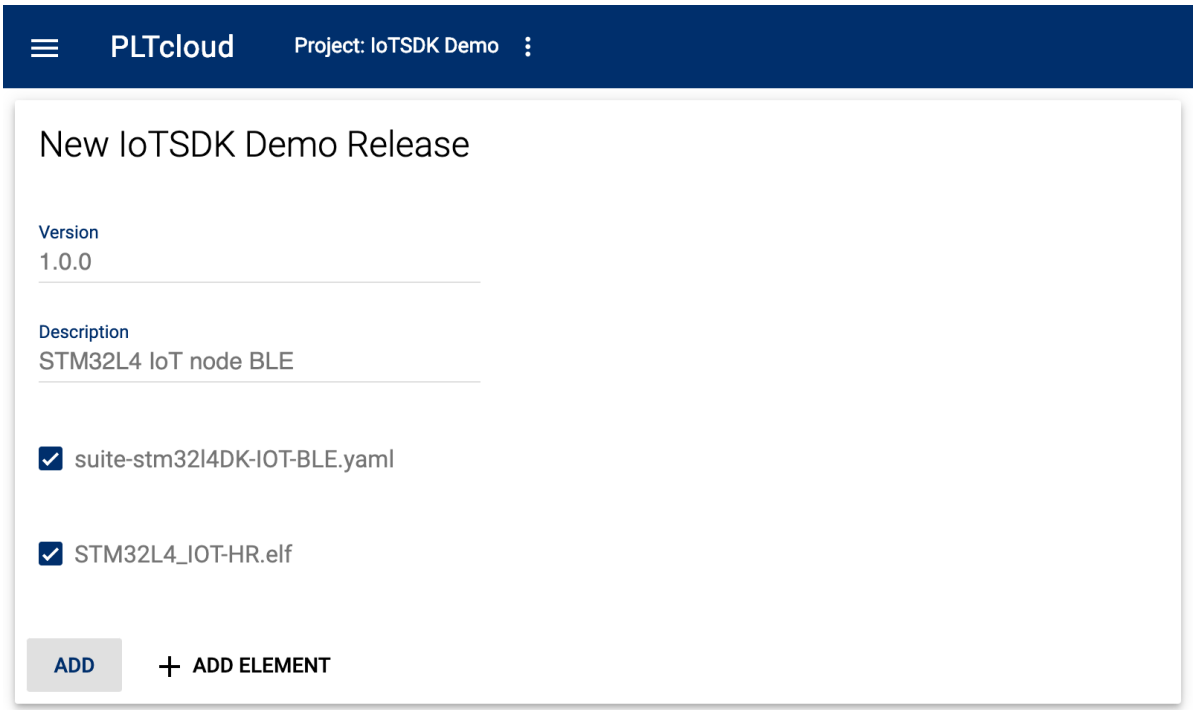

Figure 1.14. New Release

Back to the main release section, you can click on the deployment icon next to the release you want to deploy to be able to update the PLTs part of that deployment group with this release. Here, it has been deployed to the “DEV” deployment group.

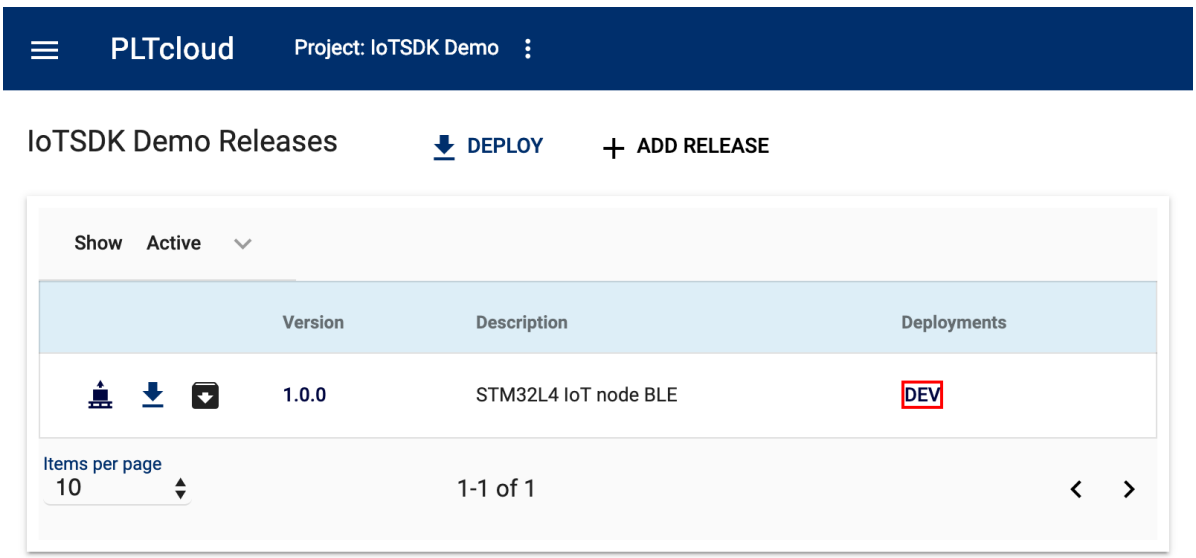

Figure 1.15. Release List

1.9 Project Settings

The **Edit** view can be used to modify basic project settings: - Project Name and Description - Serial Number Allocation scheme (Monotonic / Webhook)

1.9.1 Serial Number Webhook API

To perform number allocation through an external webhook API, select the **Webhook** Serial Number Allocation scheme.

Specify the API endpoint:

- WebHook URL: URL of the API endpoint accepting HTTP POST requests from PLTcloud.
- Authentication Token: Hex-encoded token, used to sign the JSON payload sent to the provided URL

1.9.1.1 Webhook Request payload

The configured webhook endpoint will be sent an HTTP POST request containing a JSON payload like below:

```
{
    "dut_id": "<DUT IDENTIFIER>",
    "ble_mac": "<BLE MAC>",
    "mcu_id": "<MCU ID>",
    "MYCUSTOMKEY": "my custom value"
}
```

For security reasons, you probably want to limit requests to those coming from PLTcloud. This can be accomplished by setting an authentication token, and validate the X-Hub-Signature header sent in the webhook request.

The X-Hub-Signature header contains a SHA1 HMAC hex digest of the webhook payload, using the webhook's authentication token as the key and prefixed with sha1=. The way you verify this signature varies, depending on the language of your code base.

1.9.1.2 Webhook Response

The API endpoint should return something like:

```
{
    "serial_number": "<SERIAL NUMBER>"
}
```

1.9.2 Debug Keys

The **Debug Keys** view allow users with the ***Administrator*** role to manager public SSH keys to control local access to PLTs while in ***Debug Mode***.

The SSH keys for which public keys are attached for a project can be used to upload test plan YAML files and other artifacts using **SFTP**.

Once a debug SSH (.pub) key is set on a project that is deployed to a PLT, the debug mode can be activated by an *Administrator* from the **Reset** section of the PLT view in PLTcloud:

- Reset PLT in Debug Mode
- Wait for PLT to restart in debug mode

While the PLT is in debug mode, an SFTP client (such as sftp or ***WinSCP***) can be used to upload release elements.

```
$ sftp debug@<PLT IP>
debug@<PLT_IP>$ put myTestPlan.yaml
debug@<PLT_IP>$ put firmware.hex
debug@<PLT_IP>$ quit
```

Note

after adding or modifying test plan YAML files, use the on-screen menu to select “Restart” to ensure the new plans are available.

1.10 Reports

When a test is executed on a PLT, the report will be automatically generated and be available under the “Reports” section.

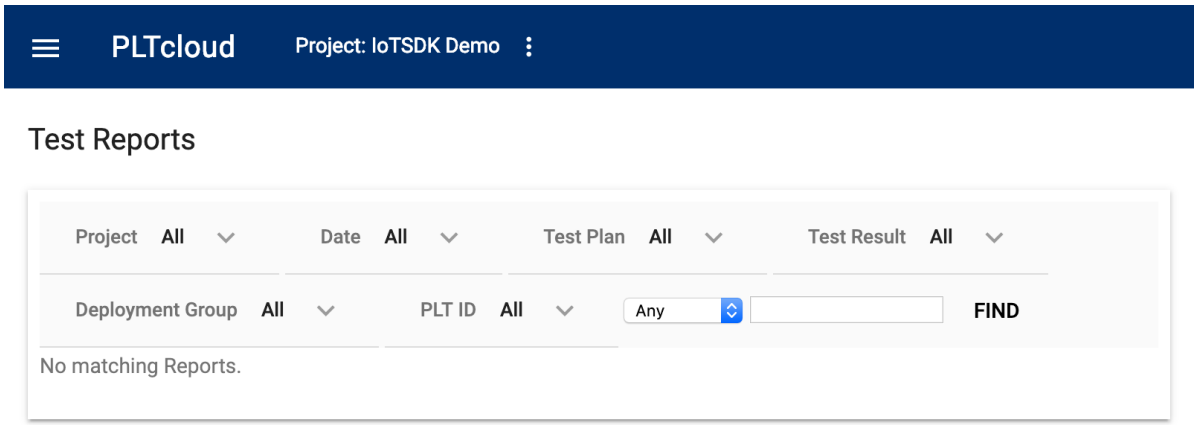

Figure 1.16.

Reports are listed in chronological order, and they will be bold-ed if they were generated less than 5 minutes ago.

2. PLT Operation Guide

2.1 Summary

The Production Line Tool (PLT) executes one or more test plans, for In-Circuit Test (ICT), Final Assembly, Test and Packout (FATP) or other production stations.

With a custom test suite definition, the PLT can also be used for integration testing during development.

It tests interactions over:

- BLE
- UART

During the Test plan, the PLT may also perform firmware programming of:

- RF SoC
- External flash

2.2 Setup

- Connect the Ethernet port to a local network with (outbound) Internet access.
- Connect the AC power socket to AC power (110-240VAC, 50-60Hz)
- (Optional) Connect Zebra label printer to external USB port
- (Optional) Connect Symbol bar code scanner to external USB port
- Use power switch on rear of unit to power on the PLT.

2.3 Usage

2.3.1 DUT Connection

- Use Tag-Connect TC2030-IDC cable to connect 6-pin rear-end IDC SWD socket with the DUT.
- PLT-100A: Use Tag-Connect TC2070-IDC cable to connect 14-pin rear-end IDC Test socket with the DUT.
- PLT-200A: Use Tag-Connect TC2050-IDC cable to connect 10-pin rear-end IDC JTAG socket with the DUT.
- PLT-200A: Use HD78 cable to connect 78-pin rear-end ICT socket with the In-Circuit Test Fixture.

2.3.2 Main Screen

After starting up, the PLT shows the currently selected test plan on the display.

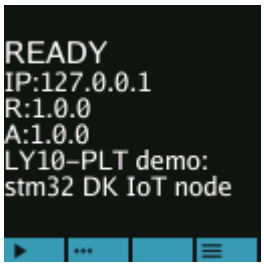

Figure 2.1. Main PLT OLED display

Press button 2 to switch to the correct test plan.

2.3.3 Test Suite Execution

From the main screen, press button #1 to start the test suite.

Figure 2.2. PLT OLED display while tests are passing.

The test suite items will be executed until a failure occurs.

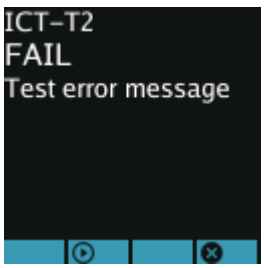

Figure 2.3. PLT OLED display for Test Item failure

When a test item fails, press button 2 to continue with the next item from the test suite.

Press button 4 to abort the test plan.

2.3.4 Operator Tests

Some test steps are not fully automated and require operator intervention.

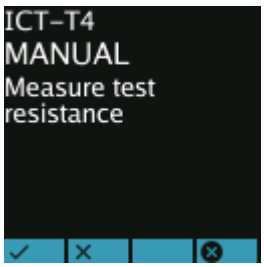

Figure 2.4. PLT OLED display for Operator test item step

2.3.5 Test Suite Completion

Upon completion of the ICT and FATP test plans, a label is printed on a ZPL barcode printer.

2.4 Label Test

To print a test label, first press button #3 from the main screen.

Use button #1 (Up) and button #2 (Down) to highlight the “Label Test” menu entry.

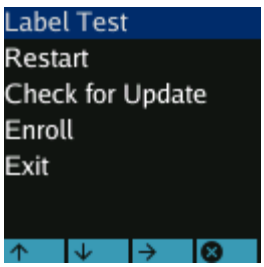

Figure 2.5. Menu on PLT OLED display

Press button #3 (Select) to print a test label.

A test label will be printed on the attached Zebra label printer.

Press button #4 (Exit) to return to the main screen.

2.5 Check for Update

To check for updated PLT firmware, first press button #3 from the main screen.

Use button #1 (Up) and button #2 (Down) to highlight the “Check for Update” menu entry.

Press button #3 (Select) to check for updated PLT firmware label.

If an updated PLT firmware is available, it will be installed.

If no updated PLT firmware is available, a message stating “No Update Available” will be shown.

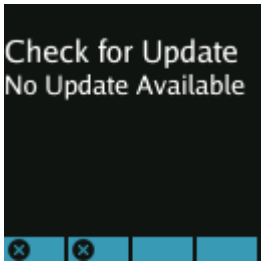

Figure 2.6. Check for Update without available update

3. PLT Probes and Signals

3.1 Summary

The PLT-200A Production Line Tool provides several test probes and signals:

- 8 Power Rails
- 49 Digital Probes
- 45 Analog Probes
- Serial interfaces
    - 1x SWD
    - 1x JTAG
    - 2x UART/SPI
    - 1x CAN

3.2 Probes and Signals

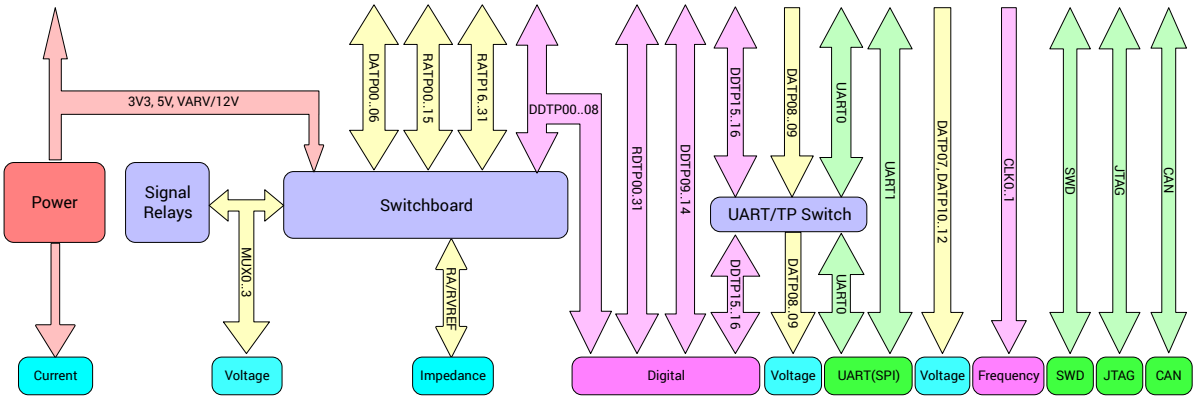

Figure 3.1. PLT-200A Probes and Signals

3.3 Power Rails

- 3V3 (3.3V)
- 5V (5V)
- VARV, 12V (2..12V)
- SWD (3.3V)
- JTAG (3.3V)
- UART0, UART1 (3.3V)

Table 3.1. PLT-200A Power Rails

PLT-200A	Notes	Rear	ICT	Switchboard	Voltage	Current
3V3	3.3V	y	y	y	(Switchboard)	y
5V	5V	y	y	y	(Switchboard)	y
12V/VARV	2..12V	y	y	y	(Switchboard)	y
SWD_VDD	3.3V	y	y	n	n	n
JTAG_VCC	3.3V	y	y	n	n	n
UART0_VDD	3.3V	y	y	n	n	n
UART1_VDD	3.3V	n	y	n	n	n
GND	GND	y	y	y	(Switchboard)	n

3.4 Analog Signals

3.4.1 Analog Probes

- 13 Direct Analog Test Probes (DATP00..DATP12)
    - Connected directly from PPC to PLT
    - DATP00..DATP06 can be switched to the MUX0 and MUX1 channels for shorting, impedance or voltage measurements
    - DATP07..DATP12 are connected directly to the DAQ, allowing voltage measurements
    - DATP08 and DATP09 can be connected to the UART0_RXD and UART0_TXD probes), to allow voltage measurements on those lines.
- 32 Routed Analog Test Probes (RATP00..RATP31)
    - Connected from PPC to PLT through an analog multiplexer on the PPC
    - Only one routed analog test probe from a given ADG bank can be routed at the same time.
    - RATP00..RATP15 are part of the “ADG0 Bank”
    - RATP16..RATP31 are part of the “ADG1 Bank”
    - RATPxx pins can be routed to MUX0 and MUX1 channels for shorting, impedance or voltage measurements

Table 3.2. PLT-200A Analog Test Probes

PLT-200A	Notes	Rear	ICT	Switchboard	Voltage	Current
DATP00		n	n	y	(Switchboard)	n
DATP01		n	n	y	(Switchboard)	n
DATP02		n	n	y	(Switchboard)	n
DATP03		n	n	y	(Switchboard)	n
DATP04		n	n	y	(Switchboard)	n
DATP05		n	n	y	(Switchboard)	n
DATP06		n	n	y	(Switchboard)	n
DATP07		n	n	n	y	n
DATP08	(Alt: UART0_RXD)	y	y	n	y	n
DATP09	(Alt: UART0_TXD)	y	y	n	y	n
DATP10		n	n	n	y	n
DATP11		n	n	n	y	n
DATP12		n	n	n	y	n
RATP00	ADG0 Bank	n	n	y	(Switchboard)	n
RATP01	ADG0 Bank	n	n	y	(Switchboard)	n
RATP02	ADG0 Bank	n	n	y	(Switchboard)	n
RATP03	ADG0 Bank	n	n	y	(Switchboard)	n
RATP04	ADG0 Bank	n	n	y	(Switchboard)	n
RATP05	ADG0 Bank	n	n	y	(Switchboard)	n
RATP06	ADG0 Bank	n	n	y	(Switchboard)	n
RATP07	ADG0 Bank	n	n	y	(Switchboard)	n
RATP08	ADG0 Bank	n	n	y	(Switchboard)	n

PLT-200A	Notes	Rear	ICT	Switchboard	Voltage	Current
RATP09	ADG0 Bank	n	n	y	(Switchboard)	n
RATP10	ADG0 Bank	n	n	y	(Switchboard)	n
RATP11	ADG0 Bank	n	n	y	(Switchboard)	n
RATP12	ADG0 Bank	n	n	y	(Switchboard)	n
RATP13	ADG0 Bank	n	n	y	(Switchboard)	n
RATP14	ADG0 Bank	n	n	y	(Switchboard)	n
RATP15	ADG0 Bank	n	n	y	(Switchboard)	n
RATP16	ADG1 Bank	n	n	y	(Switchboard)	n
RATP17	ADG1 Bank	n	n	y	(Switchboard)	n
RATP18	ADG1 Bank	n	n	y	(Switchboard)	n
RATP19	ADG1 Bank	n	n	y	(Switchboard)	n
RATP20	ADG1 Bank	n	n	y	(Switchboard)	n
RATP21	ADG1 Bank	n	n	y	(Switchboard)	n
RATP22	ADG1 Bank	n	n	y	(Switchboard)	n
RATP23	ADG1 Bank	n	n	y	(Switchboard)	n
RATP24	ADG1 Bank	n	n	y	(Switchboard)	n
RATP25	ADG1 Bank	n	n	y	(Switchboard)	n
RATP26	ADG1 Bank	n	n	y	(Switchboard)	n
RATP27	ADG1 Bank	n	n	y	(Switchboard)	n
RATP28	ADG1 Bank	n	n	y	(Switchboard)	n
RATP29	ADG1 Bank	n	n	y	(Switchboard)	n
RATP30	ADG1 Bank	n	n	y	(Switchboard)	n
RATP31	ADG1 Bank	n	n	y	(Switchboard)	n

3.5 Digital Signals

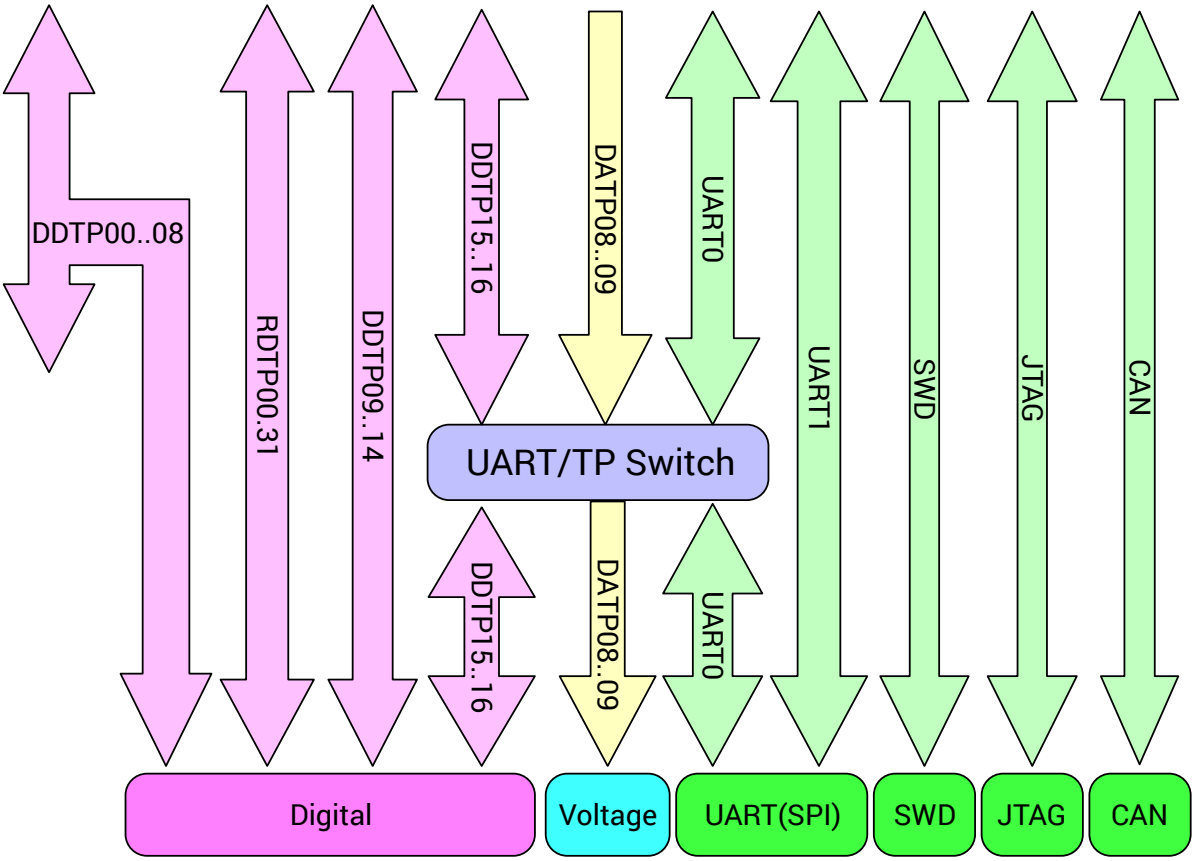

Figure 3.2. PLT-200A Digital Signals

3.5.1 Digital Probes

- 17 Direct Digital Test Probes (DDTP00..DDTP16)
    - Connected directly from PPC to PLT
    - DDTP00..DDTP08 can be switched to the MUX2 and MUX3 channels for shorting
    - Can be configured as inputs (no pull-up) or outputs (low or high)
    - Can sense logic state for probes that are configured as inputs
    - DDTP15 and DDTP16 can be connected to the UART0_CTS and UART0_RTS probes, to allow digital sensing or control of those lines.
- 32 Routed Digital Test Probes (RDTP00..RDTP31)
    - Connected from PPC to PLT through a multiplexer on the PPC
    - Can be configured as inputs (with or without pull-up) or outputs (low or high)
    - Can sense logic state for probes that are configured as inputs

Table 3.3. PLT-200A Digital Test Probes

PLT-200A	Notes	Rear	ICT	Switchboard	Voltage	Current
DDTP00		n	y	y	(Switchboard)	n
DDTP01		n	y	y	(Switchboard)	n
DDTP02		n	y	y	(Switchboard)	n
DDTP03		n	y	y	(Switchboard)	n

PLT-200A	Notes	Rear	ICT	Switchboard	Voltage	Current
DDTP04		n	y	y	(Switchboard)	n
DDTP05		n	y	y	(Switchboard)	n
DDTP06		n	y	y	(Switchboard)	n
DDTP07		n	y	y	(Switchboard)	n
DDTP08		n	y	y	(Switchboard)	n
DDTP09		n	y	n	n	n
DDTP10		n	y	n	n	n
DDTP11		n	y	n	n	n
DDTP12		n	y	n	n	n
DDTP13		n	y	n	n	n
DDTP14		n	y	n	n	n
DDTP15	(Alt: UART0_CTS)	y	y	n	n	n
DDTP16	(Alt: UART0_RTS)	y	y	n	n	n
RDTP00		n	y	n	n	n
RDTP01		n	y	n	n	n
RDTP02		n	y	n	n	n
RDTP03		n	y	n	n	n
RDTP04		n	y	n	n	n
RDTP05		n	y	n	n	n
RDTP06		n	y	n	n	n
RDTP07		n	y	n	n	n
RDTP08		n	y	n	n	n
RDTP09		n	y	n	n	n
RDTP10		n	y	n	n	n
RDTP11		n	n	n	n	n
RDTP12		n	n	n	n	n
RDTP13		n	n	n	n	n
RDTP14		n	n	n	n	n
RDTP15		n	n	n	n	n
RDTP16		n	n	n	n	n
RDTP17		n	n	n	n	n
RDTP18		n	n	n	n	n
RDTP19		n	n	n	n	n
RDTP20		n	n	n	n	n
RDTP21		n	n	n	n	n
RDTP22		n	n	n	n	n
RDTP23		n	n	n	n	n
RDTP24		n	n	n	n	n
RDTP25		n	n	n	n	n
RDTP26		n	n	n	n	n
RDTP27		n	n	n	n	n
RDTP28		n	n	n	n	n
RDTP29		n	n	n	n	n
RDTP30		n	n	n	n	n

PLT-200A	Notes	Rear	ICT	Switchboard	Voltage	Current
RDTP31		n	n	n	n	n

3.5.2 Clock Signals

- CLK0, CLK1

Table 3.4. PLT-200A Clock Signals

PLT-200A	Notes	Rear	ICT	Switchboard	Voltage	Current
CLK0		n	n	n	n	n
CLK1		n	n	n	n	n

3.5.3 Serial Interfaces

- UART0, UART1
- SWD
- JTAG
- CAN

Table 3.5. PLT-200A Serial Interface Signals

PLT-200A	Notes	Rear	ICT	Switchboard	Voltage	Current
UART0_TXD	(Alt: DATP09)	y	y	n	(Alt: DATP09)	n
UART0_RXD	(Alt: DATP08)	y	y	n	(Alt: DATP08)	n
UART0_CTS	(Alt: DDTP15)	y	y	n	n	n
UART0_RTS	(Alt: DDTP16)	y	y	n	n	n
UART1_TXD		n	y	n	n	n
UART1_RXD		n	y	n	n	n
UART1_CTS		n	y	n	n	n
UART1_RTS		n	y	n	n	n
SWD_NRST		y	y	n	n	n
SWD_SWDIO		y	y	n	n	n
SWD_SWCLK		y	y	n	n	n
SWD_SWO		y	y	n	n	n
JTAG_TMS		y	y	n	n	n
JTAG_TCK		y	y	n	n	n
JTAG_TDO		y	y	n	n	n
JTAG_TDI		y	y	n	n	n
JTAG_TRST		y	y	n	n	n
CAN_L		n	y	n	n	n
CAN_H		n	y	n	n	n

3.6 Switchboard

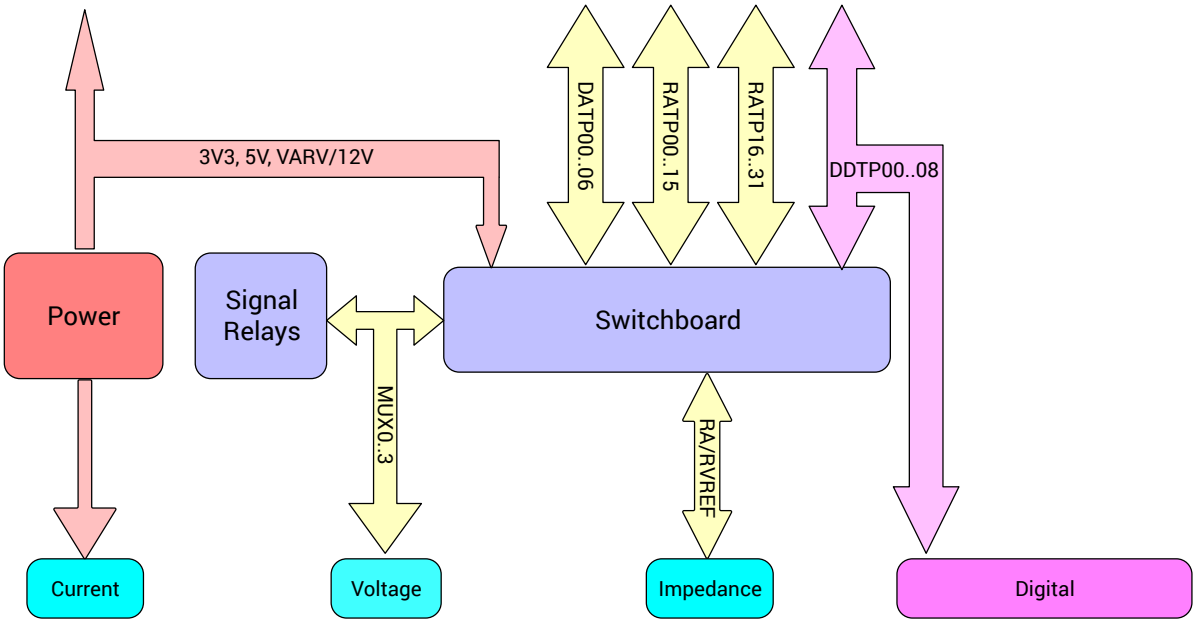

Figure 3.3. PLT-200A Switchboard

The mux command can be used to assign any of the following signals to one of 4 multiplex channels.

signal	MUX	Description
DATP00..DATP06	0, 1	Direct Analog Test Probes
RATP00..RATP31	0, 1	Routed Analog Test Probes
DDTP00..DDTP08	2, 3	Direct Digital Test Probes
RVREF	all	Impedance measurement, reference voltage
RA	all	Impedance measurement, test voltage
GND	all	Ground
3V3OUT	all	3.3V power rail
5VOUT	all	5.0V power rail
VARVDIV	all	VARV power rail, after 1/3 voltage divider

Relays between the different multiplex channels can be controlled using the short command.

```
suite:
- ident: ICT-T1
  title: Setup MUX
  steps:
  - command: mux 0 DATP02
  - command: mux 1 DATP03
  - command: mux 2 RA
  - command: mux 3 RVREF
  - command: short 0 2 set
  - command: short 1 3 set
- ident: ICT-T2
  title: Measure impedance DATP02..DATP03
  steps:
  - command: measure impedance 10-200kOhm
```

4. PLT Connectors

- 1x IEC 320 AC Power inlet (110-240VAC)
- 1x RJ45 Cat6a Ethernet port
- 4x High-Retention USB Type-A socket
- DUT connectors:
    - J2: 78-pin ICT Fixture connection
    - PWR4: 4-pin DUT power connection
    - J11: 10-pin JTAG Probe
    - J8: 6-pin SWD Probe
    - J9: 6-pin UART/Analog Probe

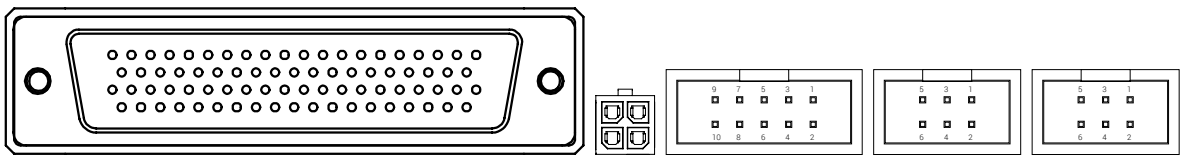

Figure 4.1. DUT connectors on PLT rear: ICT, Power, JTAG, SWD and UART/TP

4.1 DUT UART/Analog Probe connector

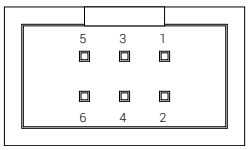

Figure 4.2. Face view of UART0/TP receptacle (J9) on PLT rear

- TC2030-IDC, 6-pin Tag-Connect (UART)
- PLT200A: J9 (CnC Tech 3020-06-0200-00)

Pin	UART Signal	Alt	Description
1	UART0_RXD	DATP08	UART Receive
2	UART0_TXD	DATP09	UART Transmit
3	UART0_CTS#	DDTP15	UART Clear To Send
4	UART0_RTS#	DDTP16	UART Ready To Send
5	UART0_VDD	-	DUT Power
6	GND	-	Ground

4.2 DUT SWD Probe connector

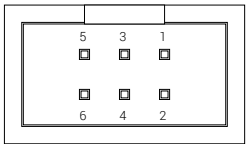

Figure 4.3. Face view of SWD receptacle (J8) on PLT rear

- TC2030-IDC, 6-pin Tag-Connect (SWD)
- PLT200A: J8 (CnC Tech 3020-06-0200-00)

Pin	SWD Signal	Description
1	SWD_VDD	DUT Power
2	SWD_SWDIO	Serial Wire Data I/O
3	SWD_NRST	System Reset
4	SWD_SWCLK	Serial Wire Clock
5	GND	Ground
6	SWD_SWO	Serial Wire Output

4.3 DUT JTAG Probe connector

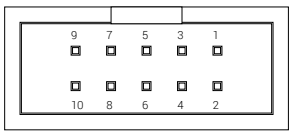

Figure 4.4. Face view of JTAG receptacle (J11) on PLT rear

- TC2050-IDC, 10-pin Tag-Connect
- PLT-200A: J11 (CnC Tech 3020-10-0200-00)

Pin	JTAG Signal	Description
1	+3V3	Debug Rail Voltage
2	JTAG_TMS	Test Mode Select
3	GND	Ground
4	JTAG_TCK	Test Clock Pin
5	JTAG_VCC	DUT Power
6	JTAG_TDO	Test Data Out
7	n.c.	-
8	JTAG_TDI	Test Data In
9	n.c.	-
10	JTAG_TRST	System Reset

4.4 DUT Power Fixture Connection

Connector: Molex Micro-FIT; 430450400

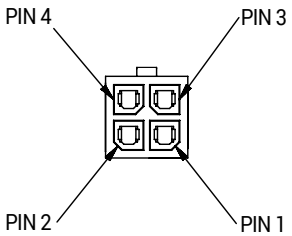

Figure 4.5. Face view of Molex 430450400 Micro-FIT header on PLT rear

Pin	Signal	Description
1	GND	GND
2	5VOUT	5V
3	3V3OUT	3.3V
4	VARVOUT	Variable DC (2V - 12V)

4.5 DUT ICT Fixture Connection

Connector: Harting Elektronik 09565527613 78 Pin D-Sub

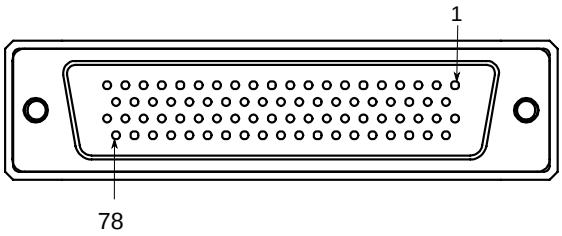

Figure 4.6. Face view of Female HD78 receptacle (J2) on PLT rear

Pin	Signal	Description
1	PPC_CLK01	Dedicated Clock Signal
2	GND	GND
3	PPC_CLK02	Dedicated Clock Signal
4	GND	GND
5	PPC_ADG0_A0	Analog MUX #0: A0
6	PPC_ADG0_A1	Analog MUX #0: A1
7	PPC_ADG0_A2	Analog MUX #0: A2
8	PPC_ADG0_A3	Analog MUX #0: A3
9	PPC_ADG0_EN	Analog MUX #0: Enable
10	GND	GND
11	PPC_ADG0	Analog MUX #0: Analog
12	PPC_ADG1_A0	Analog MUX #1: A0
13	PPC_ADG1_A1	Analog MUX #1: A1
14	PPC_ADG1_A2	Analog MUX #1: A2
15	PPC_ADG1_A3	Analog MUX #1: A3
16	PPC_ADG1_EN	Analog MUX #1: Enable
17	PPC_ADG1	Analog MUX #1: Analog
18	GND	GND
19	PPC_A0	DATP00; Dedicated Analog
20	PPC_A1	DATP01; Dedicated Analog
21	PPC_A2	DATP02; Dedicated Analog
22	PPC_A3	DATP03; Dedicated Analog
23	PPC_A4	DATP04; Dedicated Analog
24	PPC_A5	DATP05; Dedicated Analog
25	PPC_A6	DATP06; Dedicated Analog
26	PPC_A7	DATP07; Dedicated Analog
27	GND	GND
28	PPC_A8	DATP08; Dedicated Analog
29	PPC_A9	DATP09; Dedicated Analog
30	PPC_A10	DATP10; Dedicated Analog
31	PPC_A11	DATP11; Dedicated Analog
32	PPC_A12	DATP12; Dedicated Analog
33	PPC_D12	DDTP12; Digital

Pin	Signal	Description
34	PPC_D13	DDTP13; Digital
35	PPC_D14	DDTP14; Digital
36	GND	GND
37	+3V3	PPC Power supply
38	+12V	PPC Power supply
39	+5V	PPC Power supply
40	PPC_D0	DDTP00; Digital
41	PPC_D1	DDTP01; Digital
42	PPC_D2	DDTP02; Digital
43	PPC_D3	DDTP03; Digital
44	PPC_D4	DDTP04; Digital
45	PPC_D5	DDTP05; Digital
46	PPC_D6	DDTP06; Digital
47	PPC_D7	DDTP07; Digital
48	PPC_D8	DDTP08; Digital
49	PPC_D9	DDTP09; Digital
50	PPC_D10	DDTP10; Digital
51	PPC_D11	DDTP11; Digital
52	GND	GND
53	GND	GND
54	SWD_VDD	SWD: VDD
55	SWD_NRST	SWD: NRST
56	SWD_SWDIO	SWD: SWDIO
57	SWD_SWCLK	SWD: SWCLK
58	SWD_SWO	SWD: SWO
59	JTAG_VCC	JTAG: VCC Debug Rail Voltage
60	CAN_H	CAN: H
61	CAN_L	CAN: L
62	JTAG_TMS	JTAG: TMS Test Mode Select
63	JTAG_TCK	JTAG: TCK Test Clock Pin
64	JTAG_TDO	JTAG: TDO Test Data Out
65	JTAG_TDI	JTAG: TDI Test Data In
66	JTAG_TRST	JTAG: RESET System Reset
67	UART1_VDD	UART1: VDD
68	UART1_CTS#	UART1: Clear To Send
69	UART1_RTS#	UART1: Ready To Send
70	UART1_RXD	UART1: Receive
71	UART1_TXD	UART1: Transmit
72	UART0_VDD	UART0: VDD
73	UART0_CTS#	UART0: Clear To Send (Alt: DDTP15)
74	UART0_RTS#	UART0: Ready To Send (Alt: DDTP16)
75	UART0_RXD	UART0: Receive (Alt: DATP08)
76	UART0_TXD	UART0: Transmit (Alt: DATP09)
77	PPC_SDA	I2C: Data

Pin	Signal	Description
78	PPC_SCL	I2C: Clock

Compatible Cable Assemblies:

- Harting Elektronik CS-DSDHD78MM0-002.5 CABLE ASSY HD78 SHLD GRAY 762MM

5. Report Connector

It's important to not that PLTcloud itself is **not** a data warehouse. PLTcloud serves as a datasource for a data pipeline (or ETL tool) enabling you to replicate data from various sources and consolidate it into a single location.

5.1 Supported Destinations

At this point, PLTcloud supports Amazon S3 buckets as Report Destinations.

Data will not begin replication until you've successfully connected a destination to a project, and at least one test plan is executed for the project.

As part of the Enterprise plan, additional destinations can be supported.

5.2 Amazon S3 Destination

5.2.1 Prerequisites

- An Amazon Web Services (AWS) account. Signing up is free -go to https://aws.amazon.com to create an account.
- Permissions to create and manage S3 buckets in AWS.

5.2.2 Step 1: Create an Amazon S3 Bucket

- Sign into AWS Console
- Click Services on the top-left corner
- Select **S3** under the Storage section
- Click the **+ Create Bucket** button

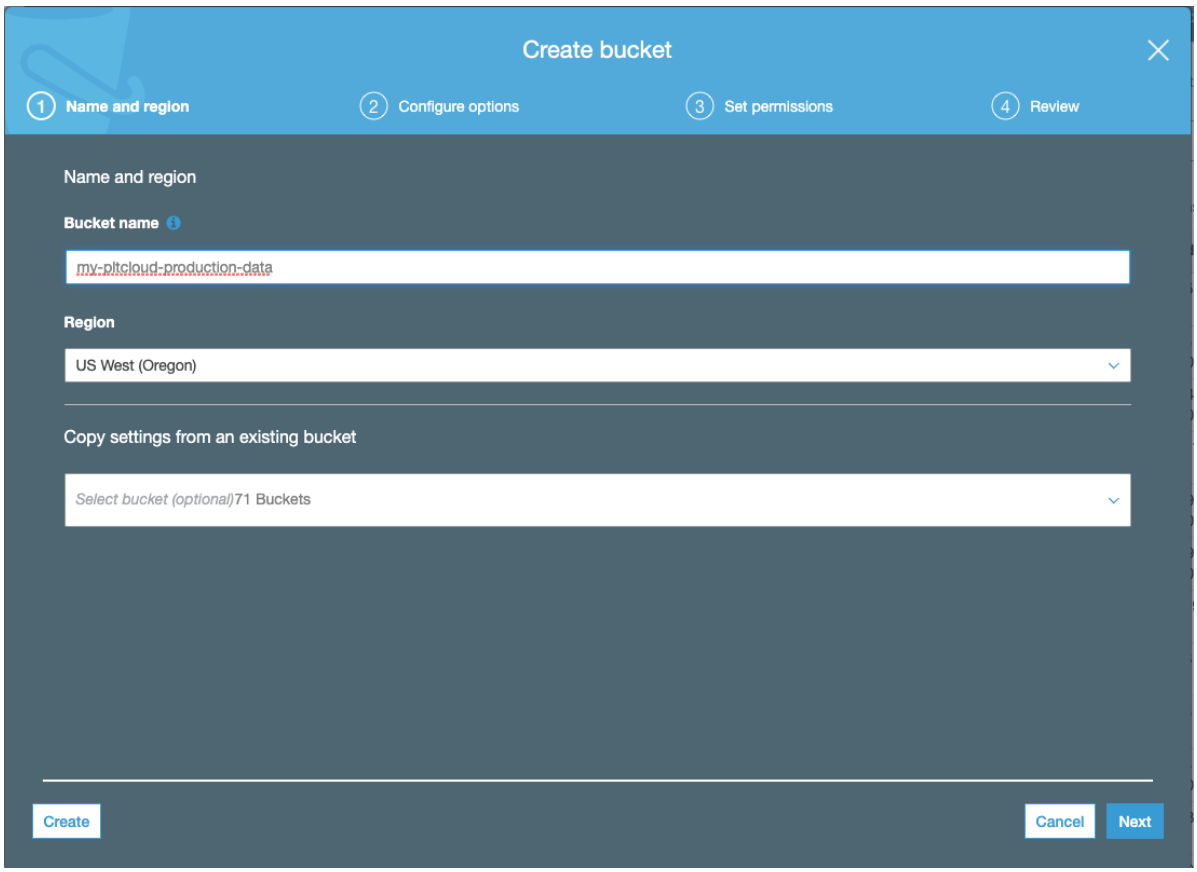

Figure 5.1. AWS S3 Bucket Creation

- On the first screen, Name and region, complete the following:
    - **Bucket name**: Enter a unique DNS-compliant name for the bucket.
    - **Region**: Select the region you want the bucket to be located in.
- Click **Next**
- On the Configure Options screen, leave all default settings as-is, and click **Next**
- On the Set Permissions screen, choose **Block all public access** and click **Next**
- On the Review screen, verify all settings and click the **Create bucket** button.

5.2.3 Step 2: Set the Report Connector Destination in PLTcloud

- If you aren't signed into your PLTcloud account, sign in first.
- Select the Project for which you'd like to set a Report Connector destination.
- Navigate to Project Settings
- Click on the **Report Connector** button
- Click on the **+ Add Destination** button
- Fill in the fields as below:
    - S3 Bucket: Fill in the name of your S3 bucket (without any `s3://` prefix)
    - Region: Fill in the AWS region of your S3 bucket
    - Report Prefix: Optionally fill in a prefix (folder name) for the reports within the bucket
- Select either `csv` or `json` for CSV or JSON reports

5.2.4 Step 3: Grant bucket access

- Navigate to the Report Connector Destination detail page.
- Expand the **S3 Bucket Policy** tab
- Copy the S3 Bucket Policy JSON
- Sign into AWS in another tab, if not currently logged in.
- In AWS Console, navigate to the Permissions tab of the S3 bucket that serves as the Report Connector Destinations.
- In the Permissions tab, click the Bucket Policy button.
- In the Bucket Policy editor, paste the bucket policy code from PLTcloud.
- Click the **Save** button when finished.

At this point, test reports should start to appear in the S3 bucket a few seconds after a test plan associated with the Project is executed on a PLT.

6. Continuous Integration

PLTcloud allows for uploading a release utilizing a command line client. This client can be included in your CI pipeline in order to automatically deploy a firmware build to PLTcloud.

6.1 Supported Environments

The PLTcloud client is distributed as a Debian package and should work on Debian and other distributions supporting “deb” files.

The PLTcloud CLI has been tested with TravisCI and GitHub Actions. Other CI tools can be utilized as long as they run a supported OS and support the installation of custom packages.

6.2 Usage

```
pltcloud -t <TOKEN> -v <VERSION> -p <UUID> -f <GLOB>
```

-t token API Token -f string File specifier -v version Release version -p uuid Project UUID

Files can be specified with patterns such as: `**/prefix*`, `grandparent/**/child?`, `**/parent/*`, or even just `**` (which will include all files and directories recursively).

6.3 Example: Travis CI integration

6.3.1 Prerequisites

- A TravisCI account and Travis CI client
- A git repository containing DUT firmware image and test plans
- A PLTcloud organization, user and project

6.3.2 Step 1: Validate Travis CI build

In order to upload a release, the PLT test plan, and any associated assets such as DUT firmware must be available in a directory. The existing build should create all the necessary files for a release in the install or script life cycle.

For example:

```
sudo: required
language: c
script:
- make dist
```

6.3.3 Step 2: Add section to install PLTcloud CLI

Add a `before_deploy:` section to `.travis.yml` in order to install the CLI tool.

```
before_deploy:
- sudo apt-get update
- sudo apt-get install -y musl
- wget https://download.pltcloud.com/cli/pltcloud_0.3.0_amd64.deb
- sudo dpkg -i pltcloud_0.3.0_amd64.deb
```

6.3.4 Step 3: Configure Project and Token environment variables

- Log in to PLTcloud and select the Project menu item from the project drop-down in the top banner.
- Copy the `UUID` from the project detail page and as an environment variable in `.travis.yml`

  For example:

  ```
  env:
    global:
      PROJECT_UUID=672124b6-9894-11e5-be38-001d42e813fe
  ```

- Select API Tokens from the drop-down menu under the user menu.
- Select Add Release Token, login and copy the Release Upload Token
- Encrypt the token and add it to `.travis.yml` with the command `travis encrypt API_TOKEN=********** --add`

6.3.5 Step 4: Add deployment

- Add deployment section to `.travis.yml`

  ```
  deploy:
    - provider: script
      skip_cleanup: true
      script: pltcloud -t "$API_TOKEN" -f "dist/*" -v "$TRAVIS_TAG" -p
  "$PROJECT_UUID"
      on:
        all_branches: true
        tags: true
  ```

6.4 Example: GitHub Actions

6.4.1 Prerequisites

- A GitHub account and repository containing DUT firmware image and test plans
- Access to GitHub Actions beta
- A PLTcloud organization, user and project

6.4.2 Step 1: Validate firmware build and test plans

In order to upload a release, the PLT test plan, and any associated assets such as DUT firmware must be available in a directory. The existing GitHub action build the firmware and copy test plans to a known directory. For example the Zephyr Action in the Zephyr firmware for demo board project builds the firmware files and copies the test suites to a dist directory.

6.4.3 Step 2: Configure project and token secrets

- Log in to PLTcloud and select the Project menu item from the project drop-down in the top banner.
- Copy the `UUID` from the project detail page and a secret named `PROJECT_UUID` in your GitHub project.
- Select `API Tokens` from the drop-down menu under the user menu in PLTcloud
- Select `Add Release Token`, login and copy the `Release Upload Token`
- Add the release token to GitHub secrets and name it `API_TOKEN`

Step 3: Create an Action to deploy to PLTcloud ==========

- Create a new action folder `pltcloud` in same folder as existing actions, for example `.github/actions/pltcloud`.
- Create a new file named `Dockerfile` in the `pltcloud` action folder with the following contents:

```
FROM debian:stretch

LABEL "com.github.actions.name"="PLTcloud publish"
LABEL "com.github.actions.description"="Publish to PLTcloud"
LABEL "com.github.actions.icon"="package"
LABEL "com.github.actions.color"="blue"

LABEL "repository"="https://github.com/bcdevices/ly10-zephyr-fw"
LABEL "homepage"="https://github.com/bcdevices/ly10-zephyr-fw"
LABEL "maintainer"="Blue Clover Devices"

WORKDIR /usr/src

RUN set -xe \
  && apt-get update \
  && apt-get install -y --no-install-recommends \
  musl \
  ca-certificates \
  bash \
  wget \
 && rm -rf /var/lib/apt/lists/*

RUN wget https://download.pltcloud.com/cli/pltcloud_0.3.0_amd64.deb
RUN dpkg -i pltcloud*_amd64.deb

ADD entrypoint.sh /entrypoint.sh
ENTRYPOINT ["/entrypoint.sh"]
```

- Create a new file named `entrypoint.sh` in the `pltcloud` actions folder with the following contents:

```
#!/bin/bash -l

pltcloud -t "$API_TOKEN" -f "dist/*" -v "${GITHUB_REF:10}" -p "$PROJECT_UUID"
```

- Make sure the `entrypoint.sh` file is executable using `chmod u+x entrypoint.sh`
- Add a step in `.github/workflows/main.yml` beneath step that builds the firmware.

```
- uses: .github/actions/action-pltcloud
  if: contains(github.ref, 'tags')
  env:
    API_TOKEN: ${{ secrets.API_TOKEN }}
    PROJECT_UUID: ${{ secrets.PROJECT_UUID }}
```

Bibliography

Books

Rex Black. ***Managing the Testing Process***. Practical Tools and Techniques for Managing Hardware and Software Testing. Wiley. 3rd Edition. August 11, 2009. ISBN-10: 0470404159. ISBN-13: 978-0470404157.

Andreas Spillner, Tilo Linz, and Hans Schaefer. ***Software Testing Foundations***. A Study Guide for the Certified Tester Exam. Rocky Nook. 3rd Edition. January, 2011. ISBN-13: 978-1-933952-78-9.