Introduction

This user's manual provides instructions on using the *CondUIt software* to program and communicate with the FS140X/FS160X/FS100X/FS1412/FS1525 chip through the FaradaySemi I2C InterfaceBoard. The manual uses FS1525 as an example.

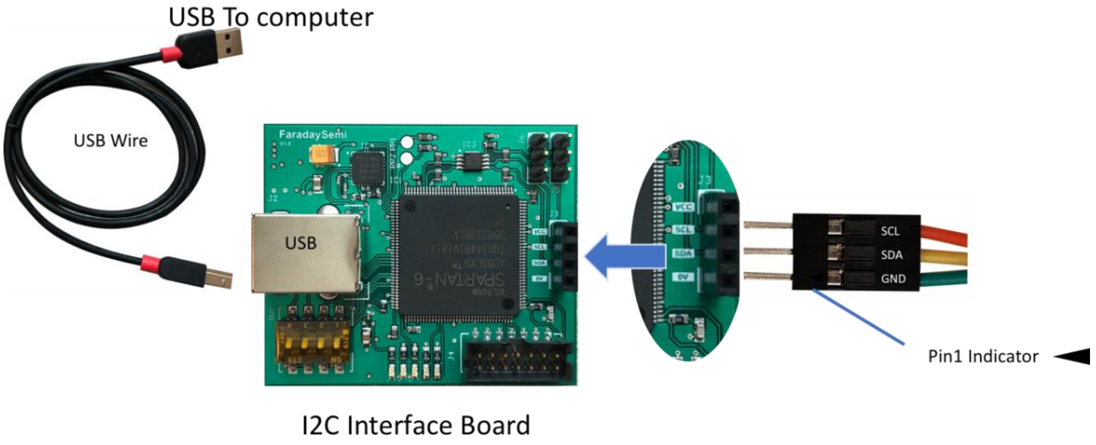
The software and complementary hardware, read and write directly to the circuit's registers using I2C.

System Requirements

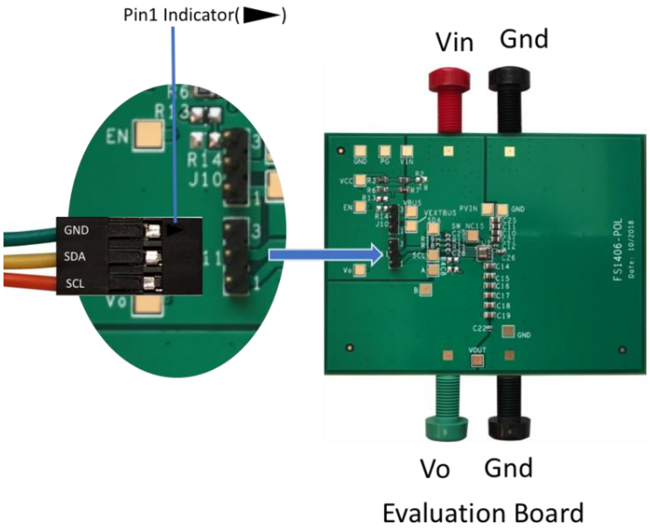
- Computer installed with Windows 7 or higher.
- 2GB Of RAM
- USB Port Access
- FaradaySemi I2C interface board kit

Software and Hardware Setup

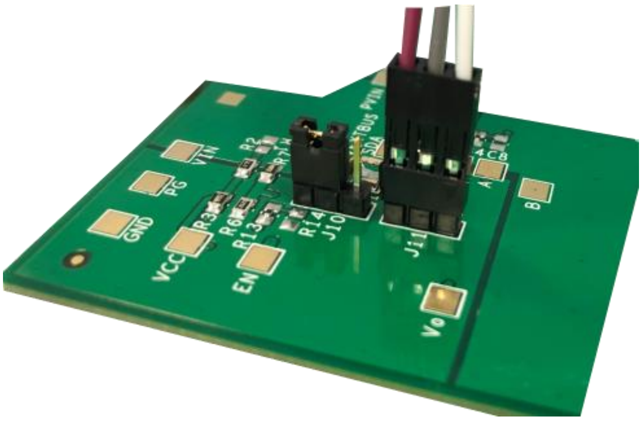
1. Download and install Silicon Labs CP210x USB to UART Bridge VCP Driver.
2. Extract the given GUI zip file – xyz.zip.
3. Connect the I2C interface board to the evaluation board by using jumper cables as shown in Fig 1.
4. Connect I2C interface board to computer using USB Cable as shown in Fig 1.
5. Bus Voltage
   a. The default setup uses **VCC as bus voltage**, jumper shorts pins 1 and 2 of J10 as shown in Fig 2.a.
   b. If using **external bus voltage**, place shorting jumper across pins 2 and 3 of J10 and attach external Bus Voltage to test point "VEXTBUS." Show in in Fig 2.b.



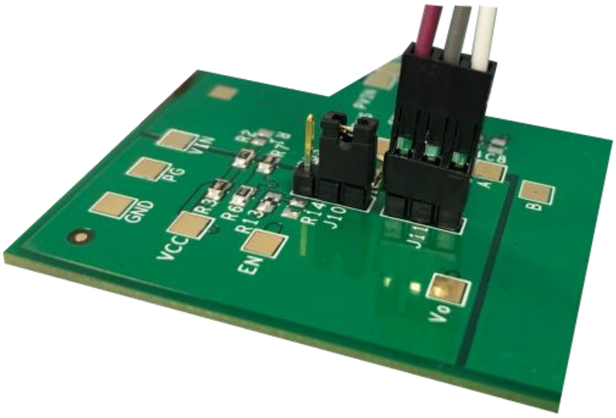
Patent Protected: US 9,729,059 B1; US 10,193,442 B2



(Fig 1.) I2C interface board connections



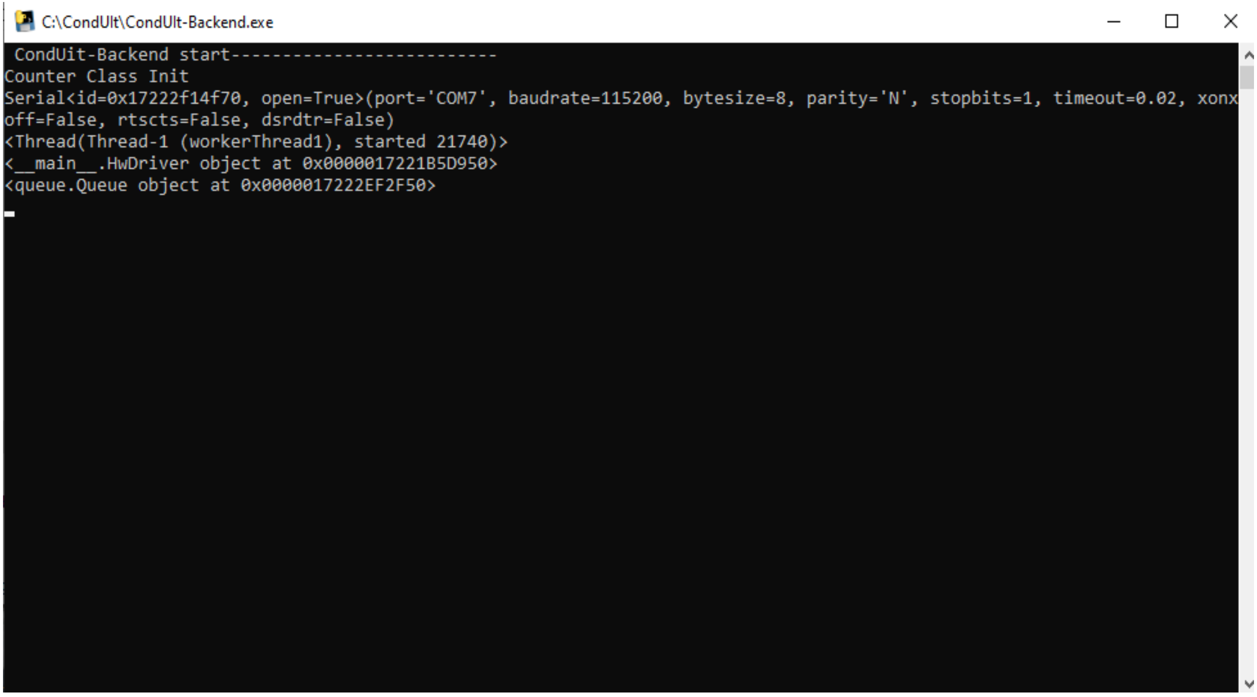
(Fig 2.a) Jumper across Pins 3 and 1 of J10. Uses VCC voltage as Bus Voltage.



(Fig 2.b) Jumper across Pins 2 and 1 of J10. Uses VEXTBUS as Bus Voltage.

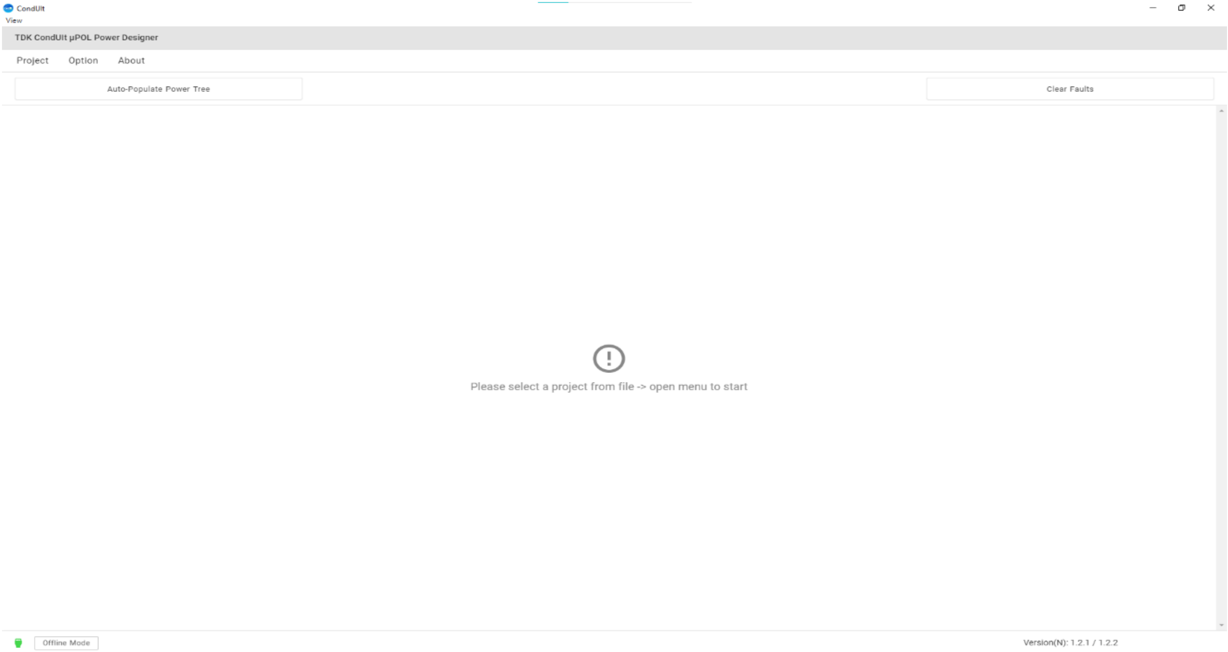
Launching the GUI

1. Go to C:\Conduit (the default installation folder) and double-click CondUIt-Backend.exe to launch the backend. Wait until the backend is ready as indicated by the highlighted line below.



```
 CondUit-Backend start-------------------------
Counter Class Init
Serial<id=0x17222f14f70, open=True>(port='COM7', baudrate=115200, bytesize=8, parity='N', stopbits=1, timeout=0.02, xonx
off=False, rtscts=False, dsrdtr=False)
<Thread(Thread-1 (workerThread1), started 21740)>
<__main__.HwDriver object at 0x0000017221B5D950>
<queue.Queue object at 0x0000017222EF2F50>
```

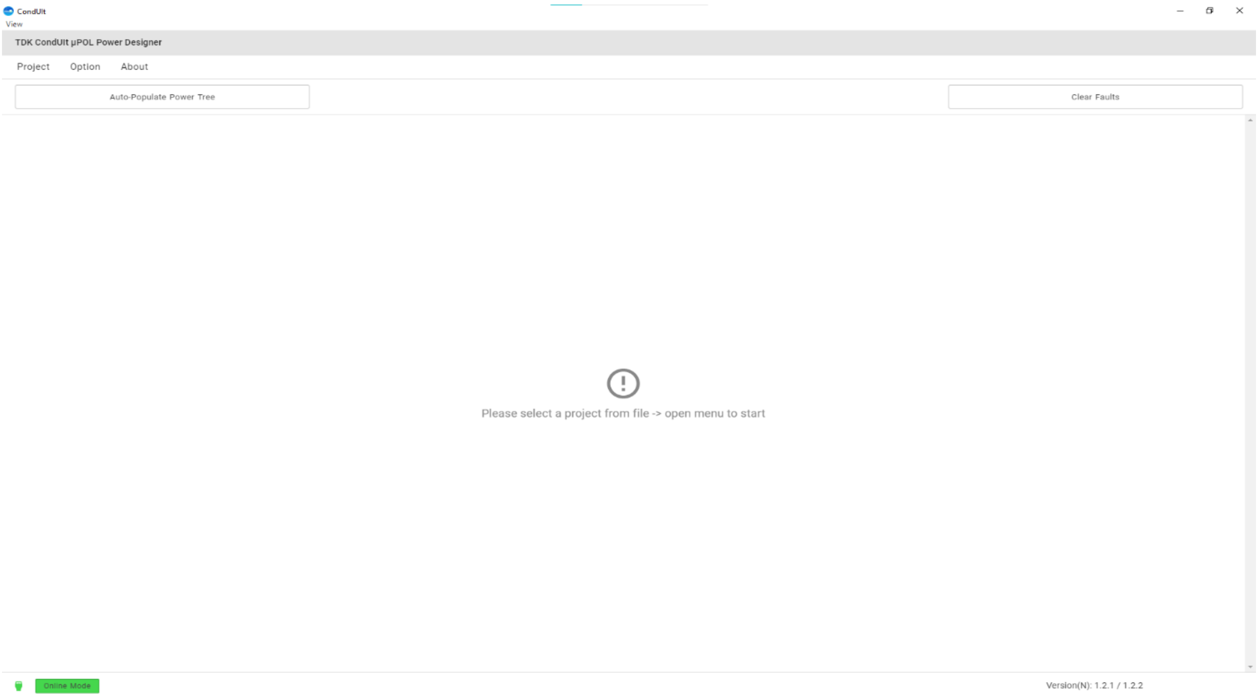
2. From C:\Conduit, double click the CondUIt application to launch the GUI. The GUI will launch in “offline” mode as shown in the highlighted area below, which means that it can be used without connecting the dongle to a FS1525 device/demo board. The green symbol indicates that the dongle status is good.



Please select a project from file -> open menu to start

NOTE: A future release of CondUIT will allow launching the backend as well as front-end with a single click

3. At this stage a project can be built in “Offline” mode, but this user guide covers the operation when a demo board is connected to the dongle and powered up. This is called the “Online” mode. The steps for offline mode are identical. To switch to online mode, click on Offline mode and it will toggle to say Online mode, in green.

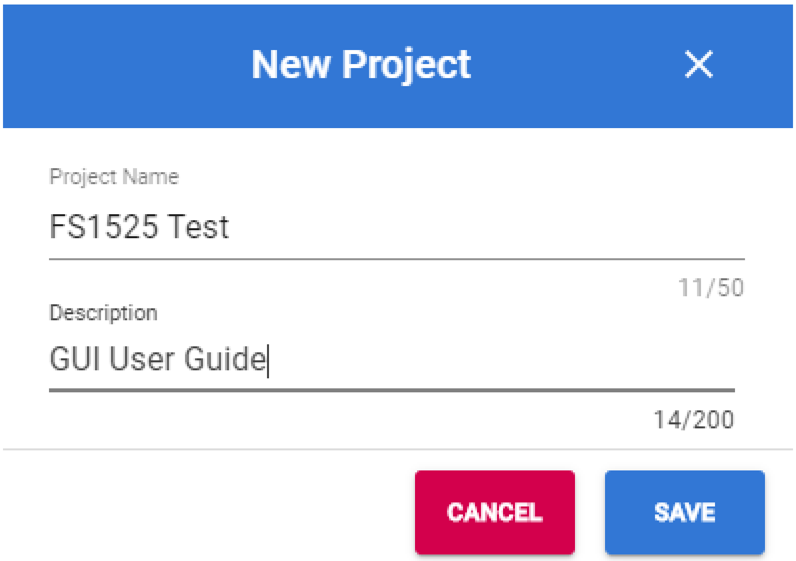


Please select a project from file -> open menu to start

Version(N): 1.2.1 / 1.2.2

Creating and working with a Project

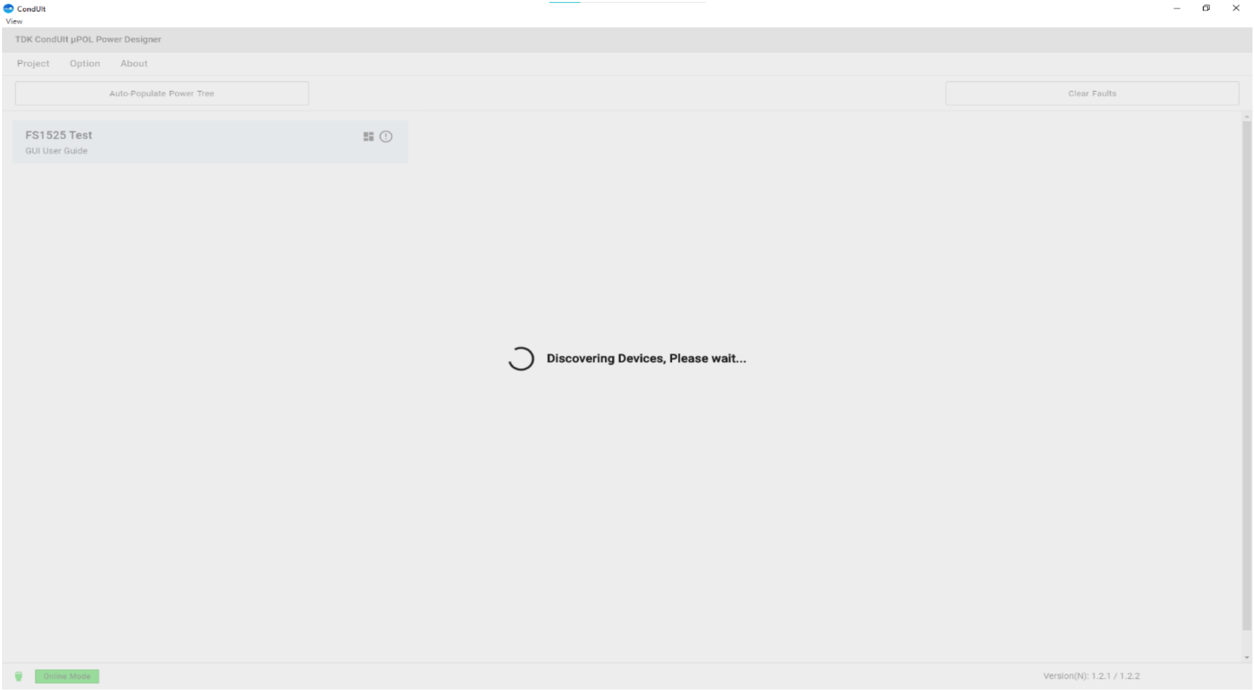
1. Go to Project>Create to create a new project. After filling out Project Name and Description, click the SAVE button to continue.



Project Name
FS1525 Test
11/50

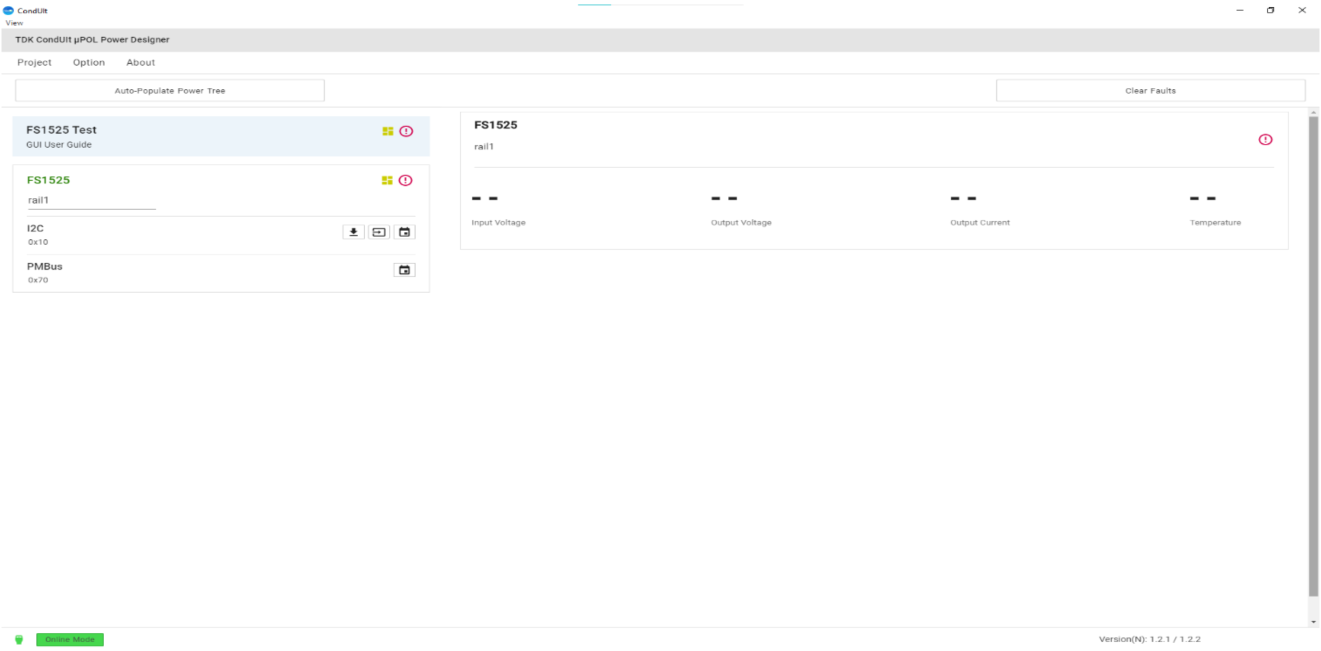
Description
GUI User Guide
14/200

2. Click on the Auto-populate Power Tree button to scan the bus for all i2c/PMBus devices. When a device is found, it will automatically populate in the project power tree. Device discovery scans for nearly 127 addresses with some wait states, so this step can take up to 2-3 minutes.

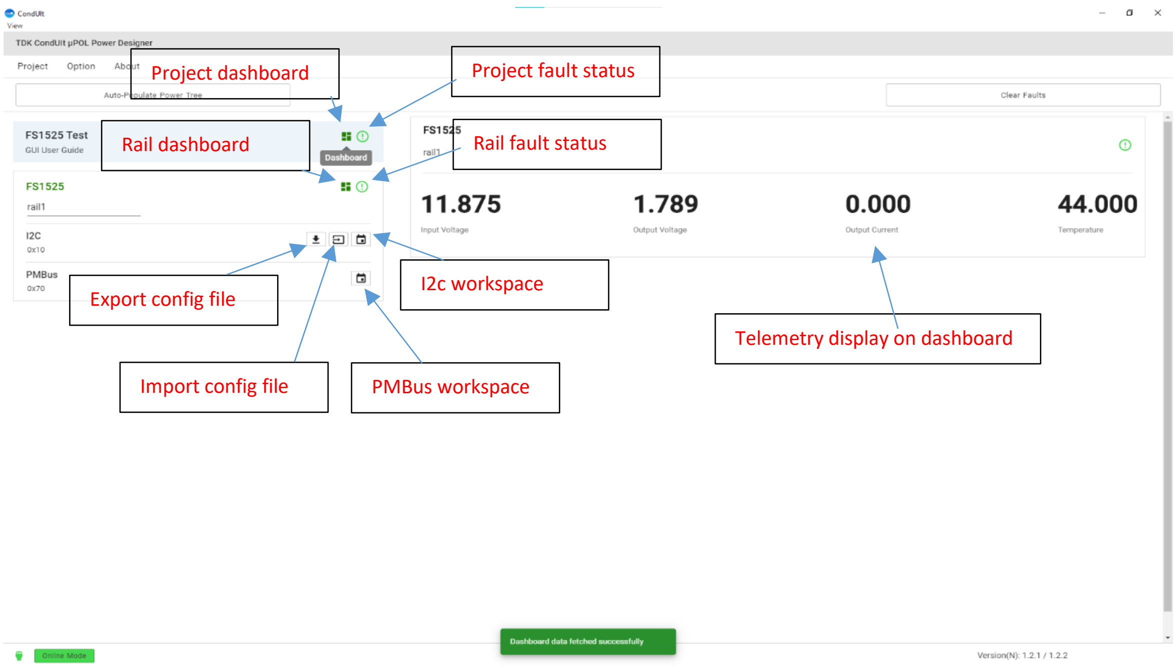


Discovering Devices, Please wait...

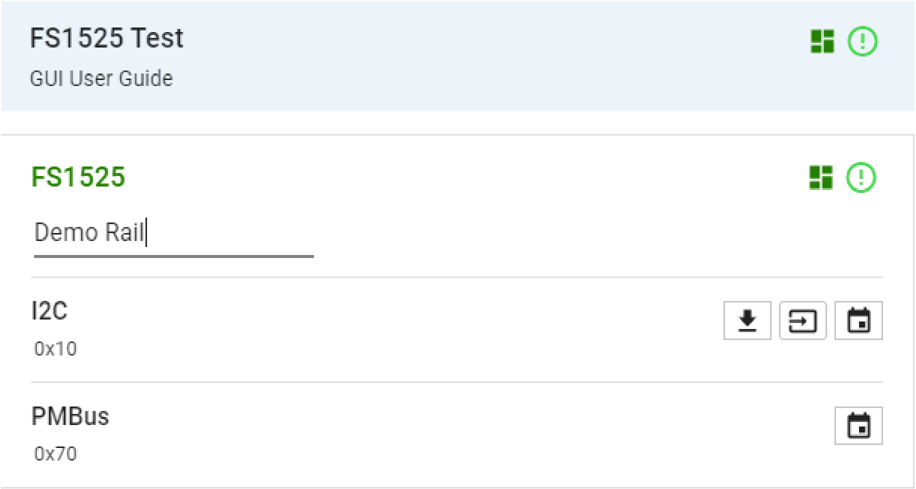
3. After the Auto-populate process is complete, the display will show the power tree enumerated with all the devices that responded to the scan, including their i2c and PMBus addresses.



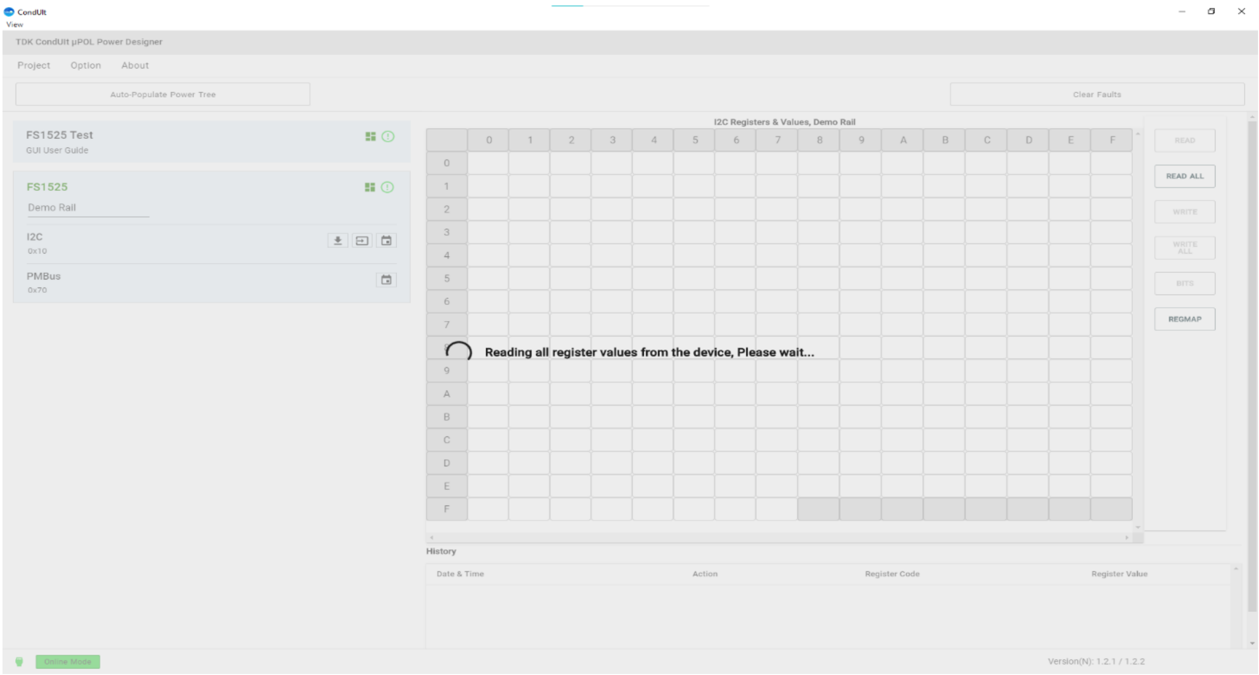
At this stage, the Telemetry display dashboard is not populated and fault indicator is red. Clicking on Project Dashboard brings up the system values and clicking on Fault status gives the current status of Faults of the device.



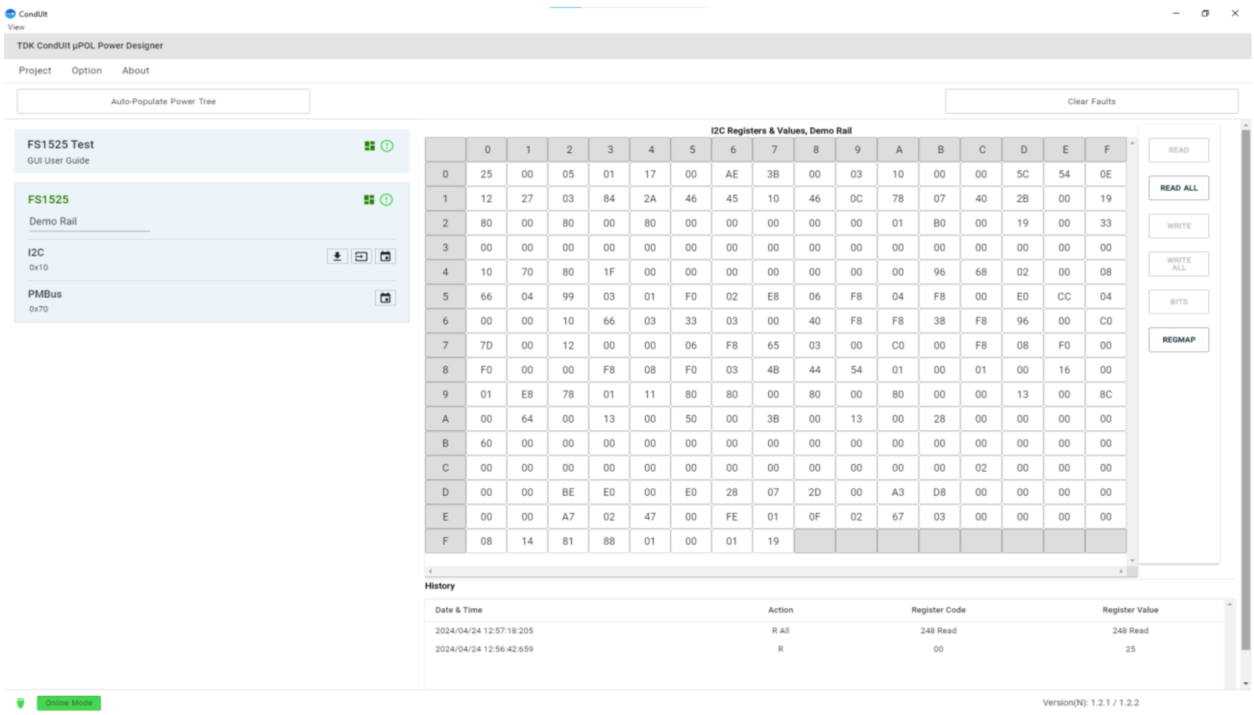
4. It is recommended to provide meaningful names to the rail (s). This is particularly useful for a multi-rail project and is done simply by clicking and typing over the default Rail name.



5. At this stage, the user can read all the i2c registers using the Read All Registers button.



Reading all register values from the device, Please wait...



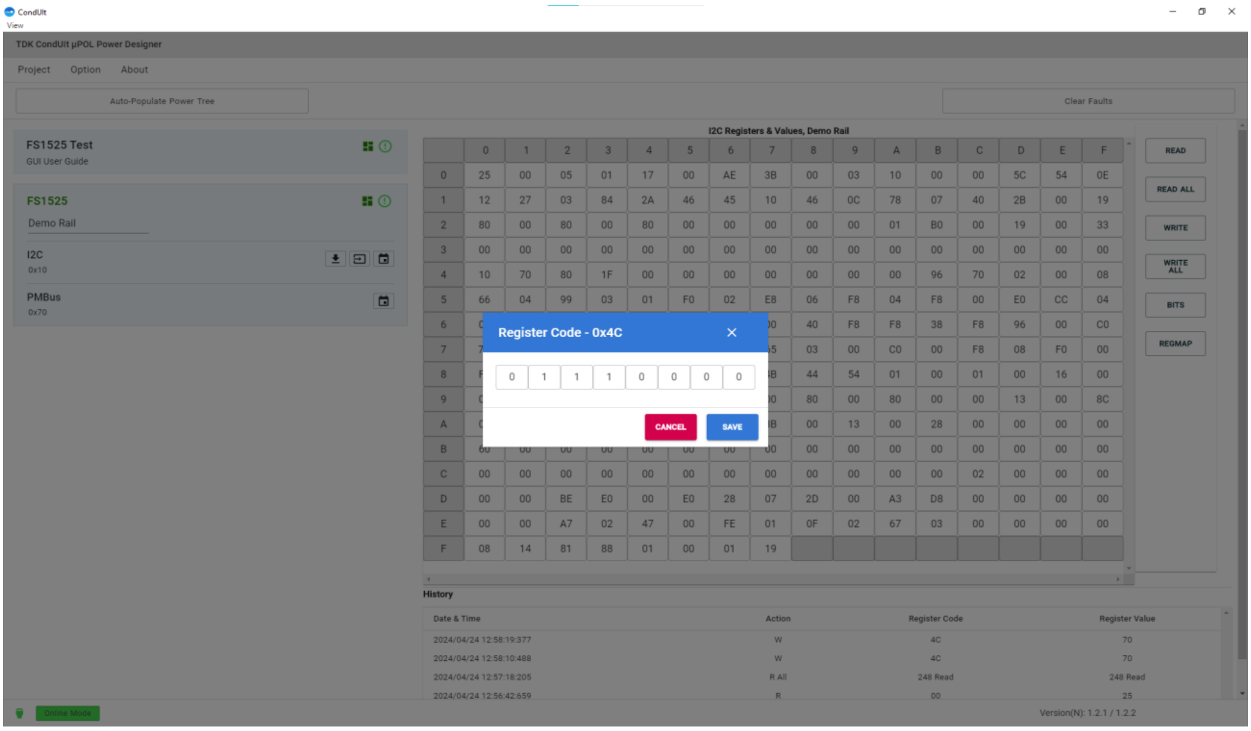
I2C Registers & Values, Demo Rail

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	25	00	05	01	17	00	AE	3B	00	03	10	00	00	5C	54	0E
1	12	27	03	84	2A	46	45	10	46	0C	78	07	40	2B	00	19
2	80	00	80	00	80	00	00	00	00	00	01	B0	00	19	00	33
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	10	70	80	1F	00	00	00	00	00	00	00	96	68	02	00	08
5	66	04	99	03	01	F0	02	E8	06	F8	04	F8	00	E0	CC	04
6	00	00	10	66	03	33	03	00	40	F8	F8	38	F8	96	00	C0
7	7D	00	12	00	00	06	F8	65	03	00	C0	00	F8	08	F0	00
8	F0	00	00	F8	08	F0	03	4B	44	54	01	00	01	00	16	00
9	01	E8	78	01	11	80	80	00	80	00	80	00	00	13	00	8C
A	00	64	00	13	00	50	00	3B	00	13	00	28	00	00	00	00
B	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00
D	00	00	BE	E0	00	E0	28	07	2D	00	A3	D8	00	00	00	00
E	00	00	A7	02	47	00	FE	01	0F	02	67	03	00	00	00	00
F	08	14	81	88	01	00	01	19								

History

Date & Time	Action	Register Code	Register Value
2024/04/24 12:57:18:205	R All	248 Read	248 Read
2024/04/24 12:56:42:659	R	00	25

6. Exporting a configuration file:
   This step illustrates the export/save of a register configuration file in text format once any changes have been made to any of the registers. In this illustration, to keep it simple, only registers 4C and 4D are written using the WRITE ALL button. Reg 0x4D is written with a value of 0x02 and Reg 0x4C is written with a value of 0x70. Note that single registers may be written using the WRITE button. The BITs button is useful to change individual bit values within a register when different bits or groups of bits in each 8-bit register serve different functions.



I2C Registers & Values, Demo Rail

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	25	00	05	01	17	00	AE	3B	00	03	10	00	00	5C	54	0E
1	12	27	03	84	2A	46	45	10	46	0C	78	07	40	2B	00	19
2	80	00	80	00	80	00	00	00	00	00	01	B0	00	19	00	33
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	10	70	80	1F	00	00	00	00	00	00	00	96	70	02	00	08
5	66	04	99	03	01	F0	02	E8	06	F8	04	F8	00	E0	CC	04
6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	40	F8	F8	38	F8	96	00	C0
7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	03	00	C0	00	F8	08	F0	00
8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	44	54	01	00	01	00	16	00
9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	80	00	80	00	00	13	00	8C
A	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	00	13	00	28	00	00	00	00
B	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00
D	00	00	BE	E0	00	E0	28	07	2D	00	A3	D8	00	00	00	00
E	00	00	A7	02	47	00	FE	01	0F	02	67	03	00	00	00	00
F	08	14	81	88	01	00	01	19								

Register Code - 0x4C

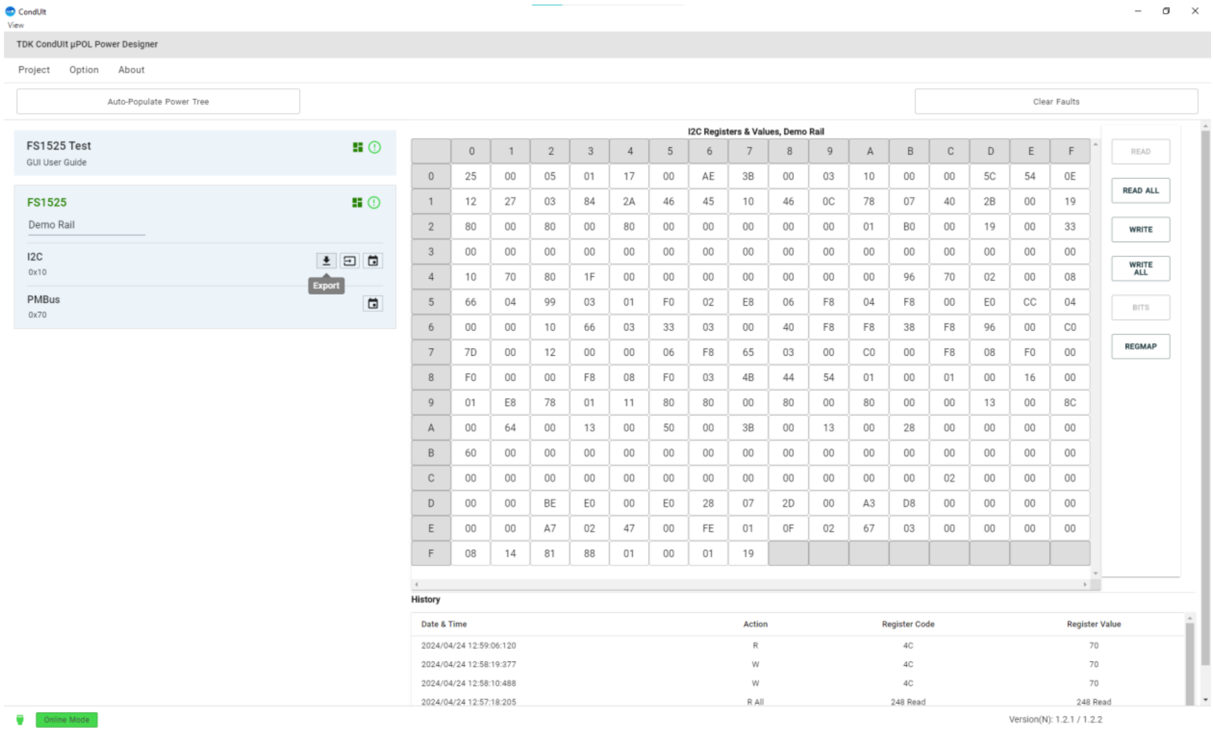
0	1	1	1	0	0	0	0

CANCEL SAVE

History

Date & Time	Action	Register Code	Register Value
2024/04/24 12:58:19:377	W	4C	70
2024/04/24 12:58:10:488	W	4C	70
2024/04/24 12:57:18:205	R All	248 Read	248 Read
2024/04/24 12:56:42:659	R	00	25

Before clicking on Export Symbol, Go to Options/Expert Mode, enter password: expertgui. Click on Export and all the values will be saved as a configuration file.

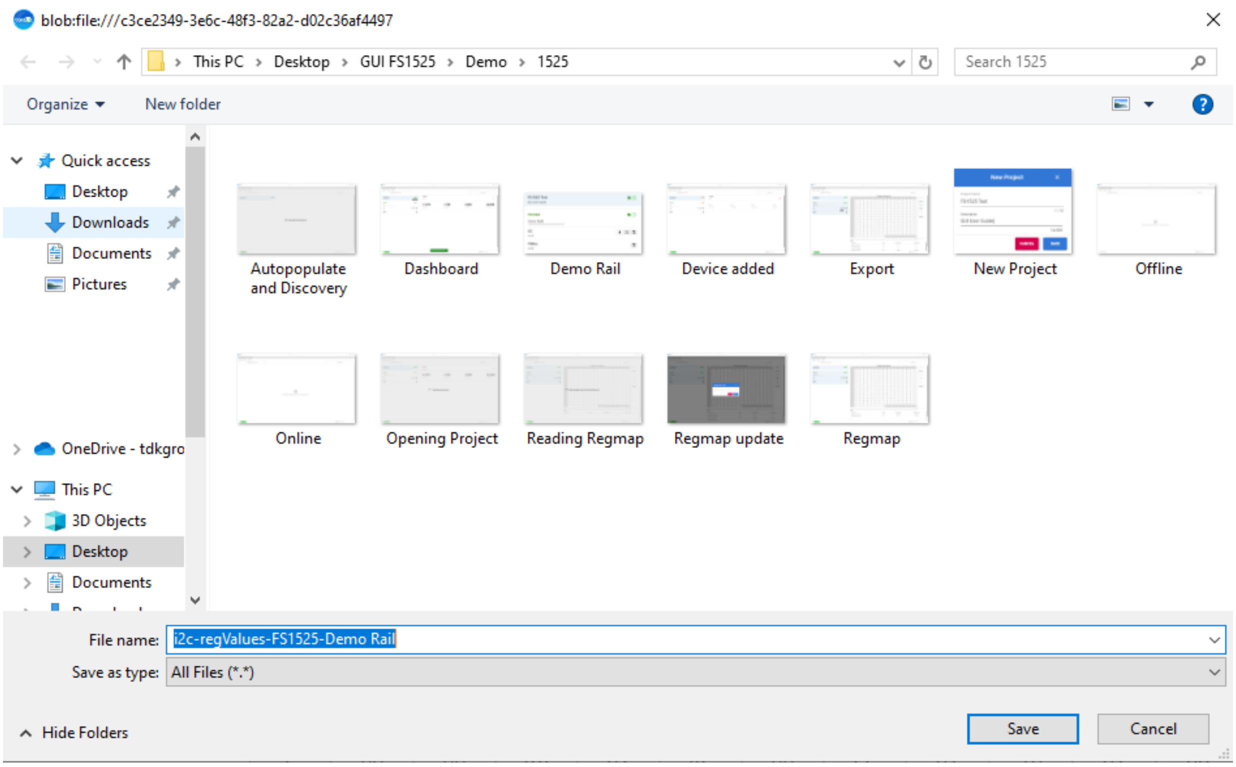


I2C Registers & Values, Demo Rail

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	25	00	05	01	17	00	AE	3B	00	03	10	00	00	5C	54	0E
1	12	27	03	84	2A	46	45	10	46	0C	78	07	40	2B	00	19
2	80	00	80	00	80	00	00	00	00	00	01	B0	00	19	00	33
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	10	70	80	1F	00	00	00	00	00	00	00	96	70	02	00	08
5	66	04	99	03	01	F0	02	E8	06	F8	04	F8	00	E0	CC	04
6	00	00	10	66	03	33	03	00	40	F8	F8	38	F8	96	00	C0
7	7D	00	12	00	00	06	F8	65	03	00	C0	00	F8	08	F0	00
8	F0	00	00	F8	08	F0	03	4B	44	54	01	00	01	00	16	00
9	01	E8	78	01	11	80	80	00	80	00	80	00	00	13	00	8C
A	00	64	00	13	00	50	00	3B	00	13	00	28	00	00	00	00
B	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00
D	00	00	BE	E0	00	E0	28	07	2D	00	A3	D8	00	00	00	00
E	00	00	A7	02	47	00	FE	01	0F	02	67	03	00	00	00	00
F	08	14	81	88	01	00	01	19								

History

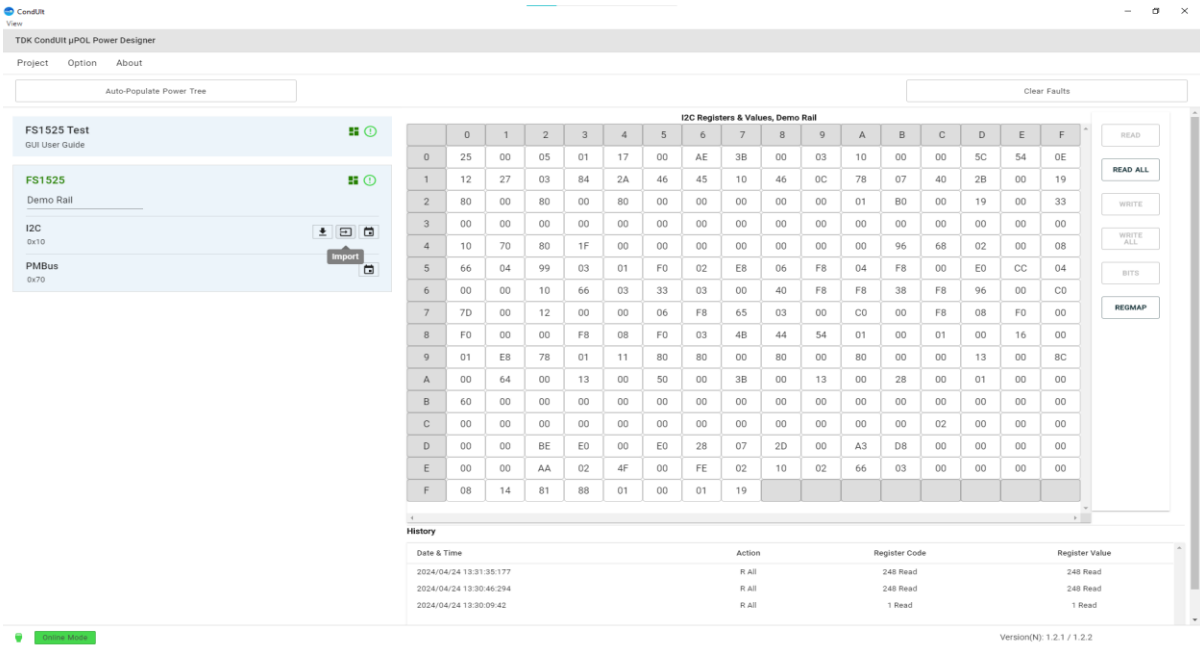
Date & Time	Action	Register Code	Register Value
2024/04/24 12:59:06:120	R	4C	70
2024/04/24 12:58:19:377	W	4C	70
2024/04/24 12:58:10:488	W	4C	70
2024/04/24 12:57:18:205	R All	248 Read	248 Read



File name: i2c-regValues-FS1525-Demo Rail

Save as type: All Files (*.*)

7. Importing a text configuration file into the registers:
   By clicking the import button, the user can browse and select the configuration file they wish to import into the design/project.

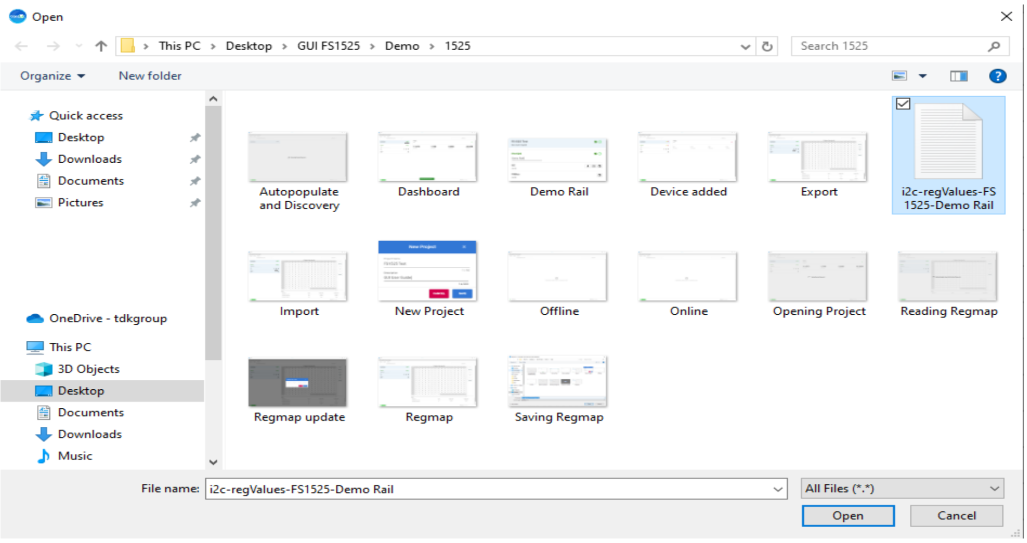


I2C Registers & Values, Demo Rail

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	25	00	05	01	17	00	AE	3B	00	03	10	00	00	5C	54	0E
1	12	27	03	84	2A	46	45	10	46	0C	78	07	40	2B	00	19
2	80	00	80	00	80	00	00	00	00	00	01	B0	00	19	00	33
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	10	70	80	1F	00	00	00	00	00	00	00	96	68	02	00	08
5	66	04	99	03	01	F0	02	E8	06	F8	04	F8	00	E0	CC	04
6	00	00	10	66	03	33	03	00	40	F8	F8	38	F8	96	00	C0
7	7D	00	12	00	00	06	F8	65	03	00	C0	00	F8	08	F0	00
8	F0	00	00	F8	08	F0	03	4B	44	54	01	00	01	00	16	00
9	01	E8	78	01	11	80	80	00	80	00	80	00	00	13	00	8C
A	00	64	00	13	00	50	00	3B	00	13	00	28	00	01	00	00
B	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00
D	00	00	BE	E0	00	E0	28	07	2D	00	A3	D8	00	00	00	00
E	00	00	AA	02	4F	00	FE	02	10	02	66	03	00	00	00	00
F	08	14	81	88	01	00	01	19								

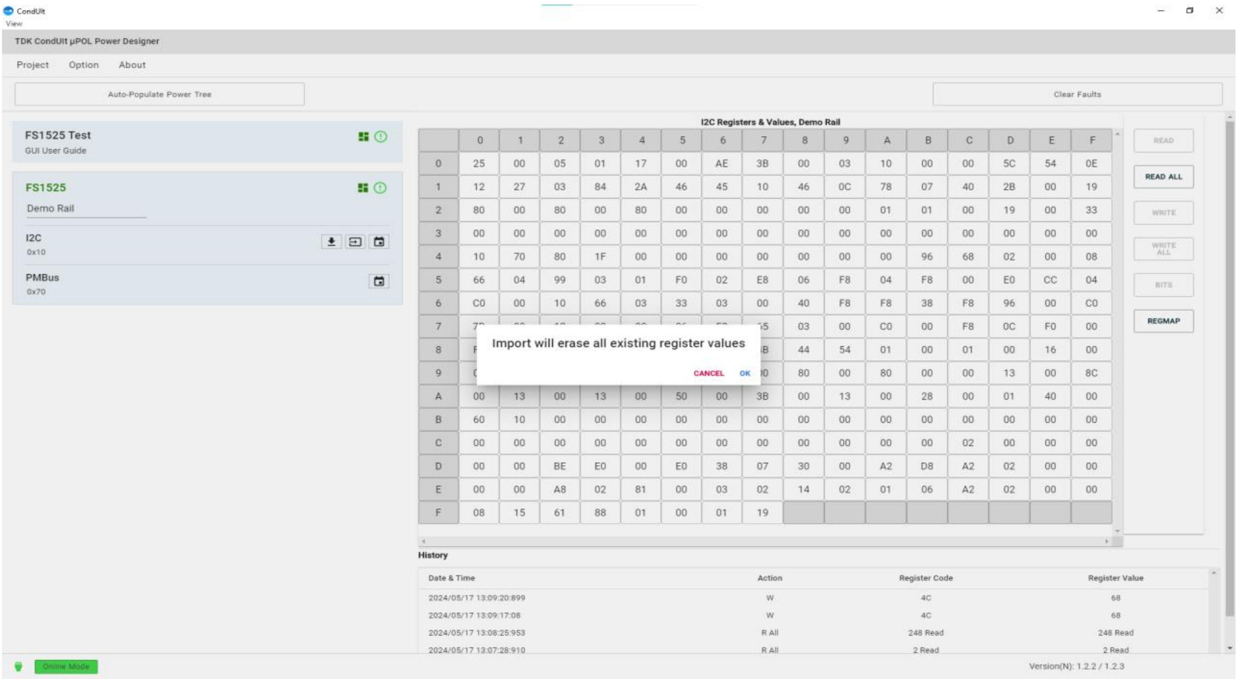
History

Date & Time	Action	Register Code	Register Value
2024/04/24 13:31:35:177	R All	248 Read	248 Read
2024/04/24 13:30:46:294	R All	248 Read	248 Read
2024/04/24 13:30:09:42	R All	1 Read	1 Read



File name: i2c-regValues-FS1525-Demo Rail

All Files (*.*)



FS1525 Test
GUI User Guide

FS1525
Demo Rail

I2C
0x10

PMBus
0x70

I2C Registers & Values, Demo Rail

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	25	00	05	01	17	00	AE	3B	00	03	10	00	00	5C	54	0E
1	12	27	03	84	2A	46	45	10	46	0C	78	07	40	2B	00	19
2	80	00	80	00	80	00	00	00	00	00	01	01	00	19	00	33
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	10	70	80	1F	00	00	00	00	00	00	00	96	68	02	00	08
5	66	04	99	03	01	F0	02	E8	06	F8	04	F8	00	E0	CC	04
6	C0	00	10	66	03	33	03	00	40	F8	F8	38	F8	96	00	C0
7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	03	00	C0	00	F8	0C	F0	00
8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	44	54	01	00	01	00	16	00
9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	80	00	80	00	00	13	00	8C
A	00	13	00	13	00	50	00	3B	00	13	00	28	00	01	40	00
B	60	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00
D	00	00	BE	E0	00	E0	38	07	30	00	A2	D8	A2	02	00	00
E	00	00	A8	02	81	00	03	02	14	02	01	06	A2	02	00	00
F	08	15	61	88	01	00	01	19								

Import will erase all existing register values

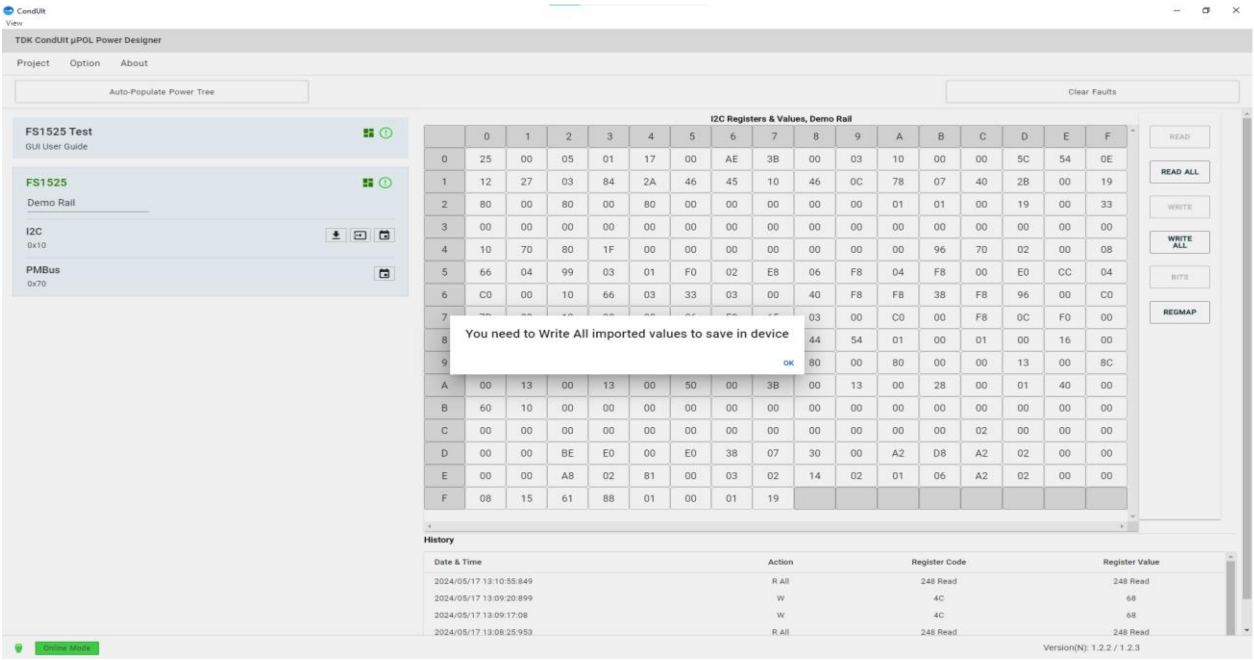
CANCEL OK

History

Date & Time	Action	Register Code	Register Value
2024/05/17 13:09:20:899	W	4C	68
2024/05/17 13:09:17:08	W	4C	68
2024/05/17 13:08:25:953	R All	248 Read	248 Read
2024/05/17 13:07:28:910	R All	2 Read	2 Read

Version(N): 1.2.2 / 1.2.3

Click OK



FS1525 Test
GUI User Guide

FS1525
Demo Rail

I2C
0x10

PMBus
0x70

I2C Registers & Values, Demo Rail

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	25	00	05	01	17	00	AE	3B	00	03	10	00	00	5C	54	0E
1	12	27	03	84	2A	46	45	10	46	0C	78	07	40	2B	00	19
2	80	00	80	00	80	00	00	00	00	00	01	01	00	19	00	33
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	10	70	80	1F	00	00	00	00	00	00	00	96	70	02	00	08
5	66	04	99	03	01	F0	02	E8	06	F8	04	F8	00	E0	CC	04
6	C0	00	10	66	03	33	03	00	40	F8	F8	38	F8	96	00	C0
7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	03	00	C0	00	F8	0C	F0	00
8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	44	54	01	00	01	00	16	00
9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	80	00	80	00	00	13	00	8C
A	00	13	00	13	00	50	00	3B	00	13	00	28	00	01	40	00
B	60	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00
D	00	00	BE	E0	00	E0	38	07	30	00	A2	D8	A2	02	00	00
E	00	00	A8	02	81	00	03	02	14	02	01	06	A2	02	00	00
F	08	15	61	88	01	00	01	19								

You need to Write All imported values to save in device

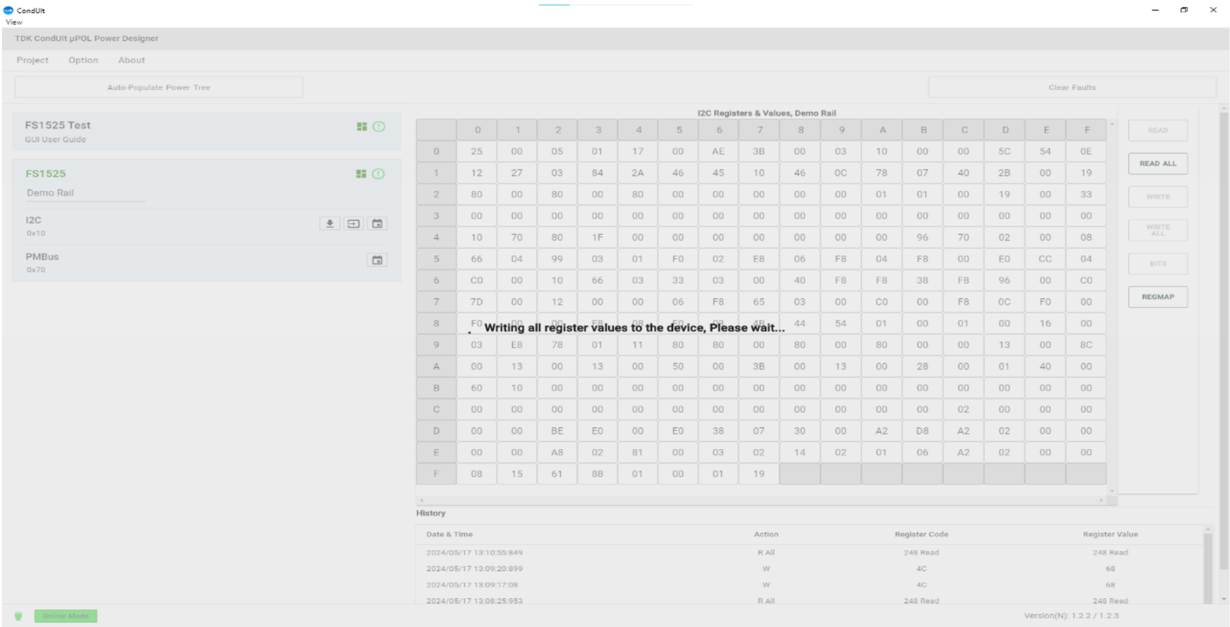
OK

History

Date & Time	Action	Register Code	Register Value
2024/05/17 13:10:55:849	R All	248 Read	248 Read
2024/05/17 13:09:20:899	W	4C	68
2024/05/17 13:09:17:08	W	4C	68
2024/05/17 13:08:25:953	R All	248 Read	248 Read

Version(N): 1.2.2 / 1.2.3

Click OK. Note that at this point the configuration values are part of the project but have not yet been written to the device registers. To do so, click WRITE ALL.



I2C Registers & Values, Demo Rail

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	25	00	05	01	17	00	AE	3B	00	03	10	00	00	5C	54	0E
1	12	27	03	84	2A	46	45	10	46	0C	78	07	40	2B	00	19
2	80	00	80	00	80	00	00	00	00	00	01	01	00	19	00	33
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	10	70	80	1F	00	00	00	00	00	00	00	96	70	02	00	08
5	66	04	99	03	01	F0	02	E8	06	F8	04	F8	00	E0	CC	04
6	C0	00	10	66	03	33	03	00	40	F8	F8	38	F8	96	00	C0
7	7D	00	12	00	00	06	F8	65	03	00	C0	00	F8	0C	F0	00
8	F0	00	00	E8	08	F0	03	4B	44	54	01	00	01	00	16	00
9	03	E8	78	01	11	80	80	00	80	00	80	00	00	13	00	8C
A	00	13	00	13	00	50	00	3B	00	13	00	28	00	01	40	00
B	60	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00
D	00	00	BE	E0	00	E0	38	07	30	00	A2	D8	A2	02	00	00
E	00	00	A8	02	81	00	03	02	14	02	01	06	A2	02	00	00
F	08	15	61	88	01	00	01	19								

Writing all register values to the device, Please wait...

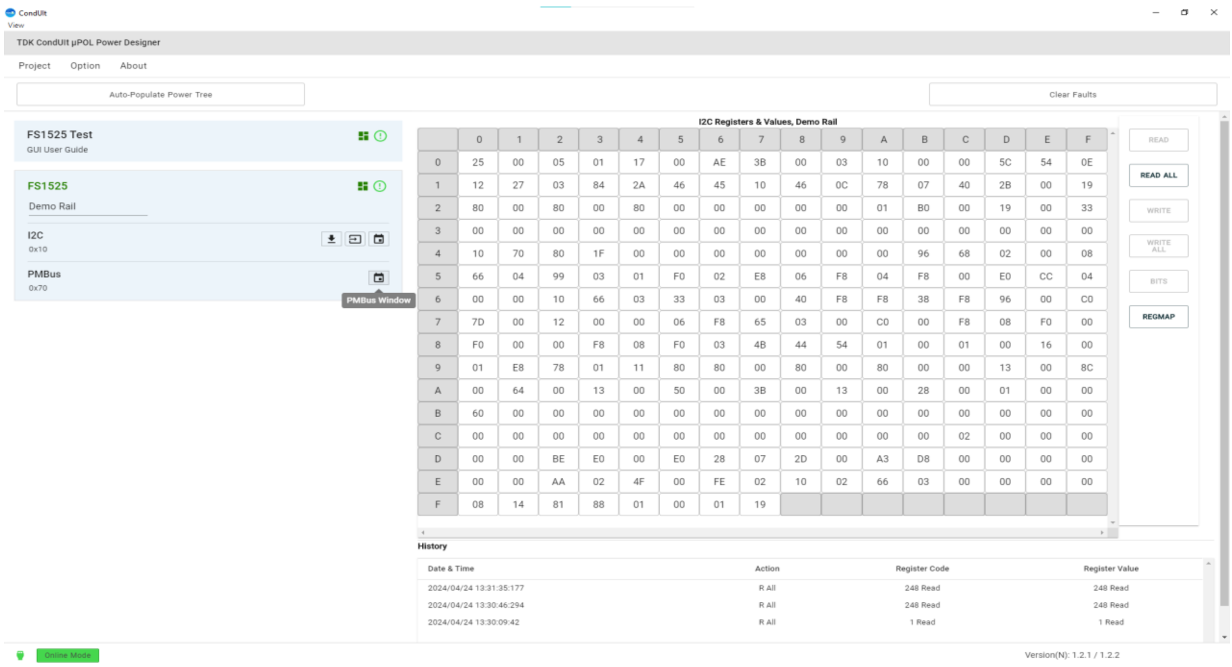
History

Date & Time	Action	Register Code	Register Value
2024/05/17 13:10:55:849	R All	248 Read	248 Read
2024/05/17 13:09:20:899	W	4C	68
2024/05/17 13:09:17:08	W	4C	68
2024/05/17 13:08:25:953	R All	248 Read	248 Read

NOTE:

Note that any changes so far are still volatile and the device will revert to its default values if Vin/Vcc are cycled. To permanently store the values to the 10 times programmable OTP memory, please follow the steps in the appendix. In a future release of CondUIt this process will be made simpler by addition of a single Burn OTP button.

8. PMBus work area: Click the PMBus workspace button to bring up the PMBus work area.

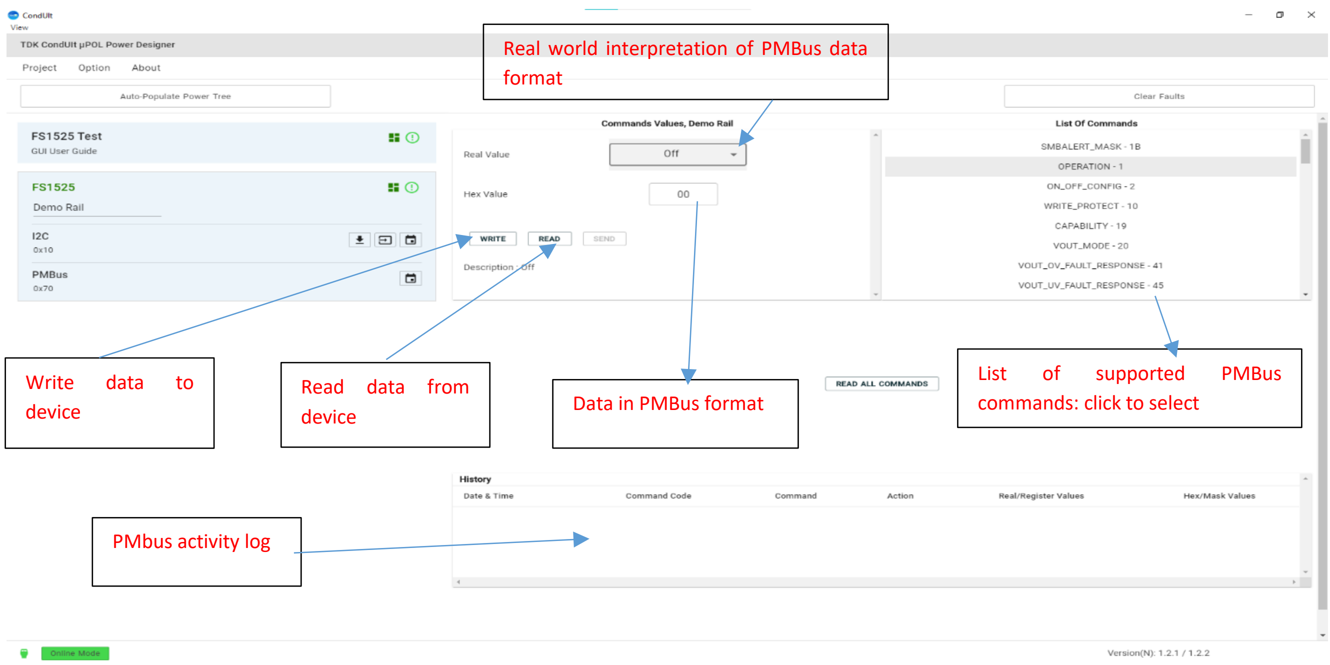


I2C Registers & Values, Demo Rail

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	25	00	05	01	17	00	AE	3B	00	03	10	00	00	5C	54	0E
1	12	27	03	84	2A	46	45	10	46	0C	78	07	40	2B	00	19
2	80	00	80	00	80	00	00	00	00	00	01	B0	00	19	00	33
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	10	70	80	1F	00	00	00	00	00	00	00	96	68	02	00	08
5	66	04	99	03	01	F0	02	E8	06	F8	04	F8	00	E0	CC	04
6	00	00	10	66	03	33	03	00	40	F8	F8	38	F8	96	00	C0
7	7D	00	12	00	00	06	F8	65	03	00	C0	00	F8	08	F0	00
8	F0	00	00	F8	08	F0	03	4B	44	54	01	00	01	00	16	00
9	01	E8	78	01	11	80	80	00	80	00	80	00	00	13	00	8C
A	00	64	00	13	00	50	00	3B	00	13	00	28	00	01	00	00
B	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00
D	00	00	BE	E0	00	E0	28	07	2D	00	A3	D8	00	00	00	00
E	00	00	AA	02	4F	00	FE	02	10	02	66	03	00	00	00	00
F	08	14	81	88	01	00	01	19								

History

Date & Time	Action	Register Code	Register Value
2024/04/24 13:31:35:177	R All	248 Read	248 Read
2024/04/24 13:30:46:294	R All	248 Read	248 Read
2024/04/24 13:30:09:42	R All	1 Read	1 Read



Commands Values, Demo Rail

Real Value: Off

Hex Value: 00

Description : Off

List Of Commands

- SMBALERT_MASK - 1B
- OPERATION - 1
- ON_OFF_CONFIG - 2
- WRITE_PROTECT - 10
- CAPABILITY - 19
- VOUT_MODE - 20
- VOUT_OV_FAULT_RESPONSE - 41
- VOUT_UV_FAULT_RESPONSE - 45

History

Date & Time	Command Code	Command	Action	Real/Register Values	Hex/Mask Values

Real world interpretation of PMBus data format

Write data to device

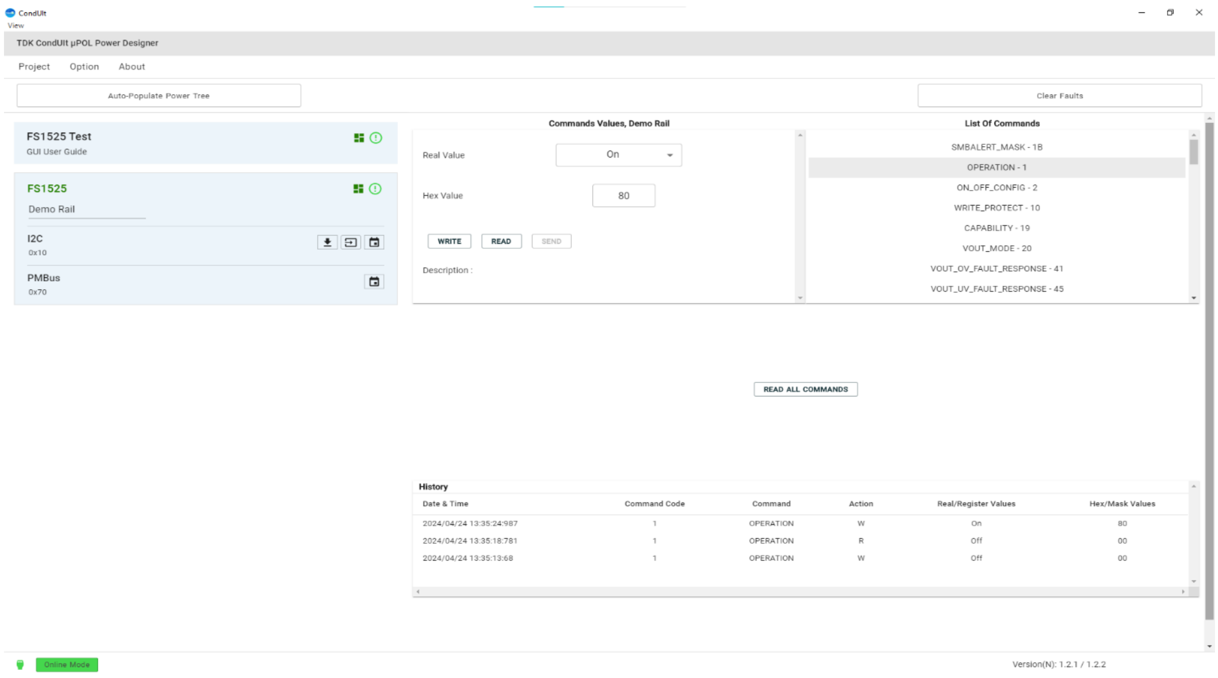
Read data from device

Data in PMBus format

List of supported PMBus commands: click to select

PMbus activity log

See an illustration of command history below:



Commands Values, Demo Rail

Real Value: On

Hex Value: 80

Description :

List Of Commands

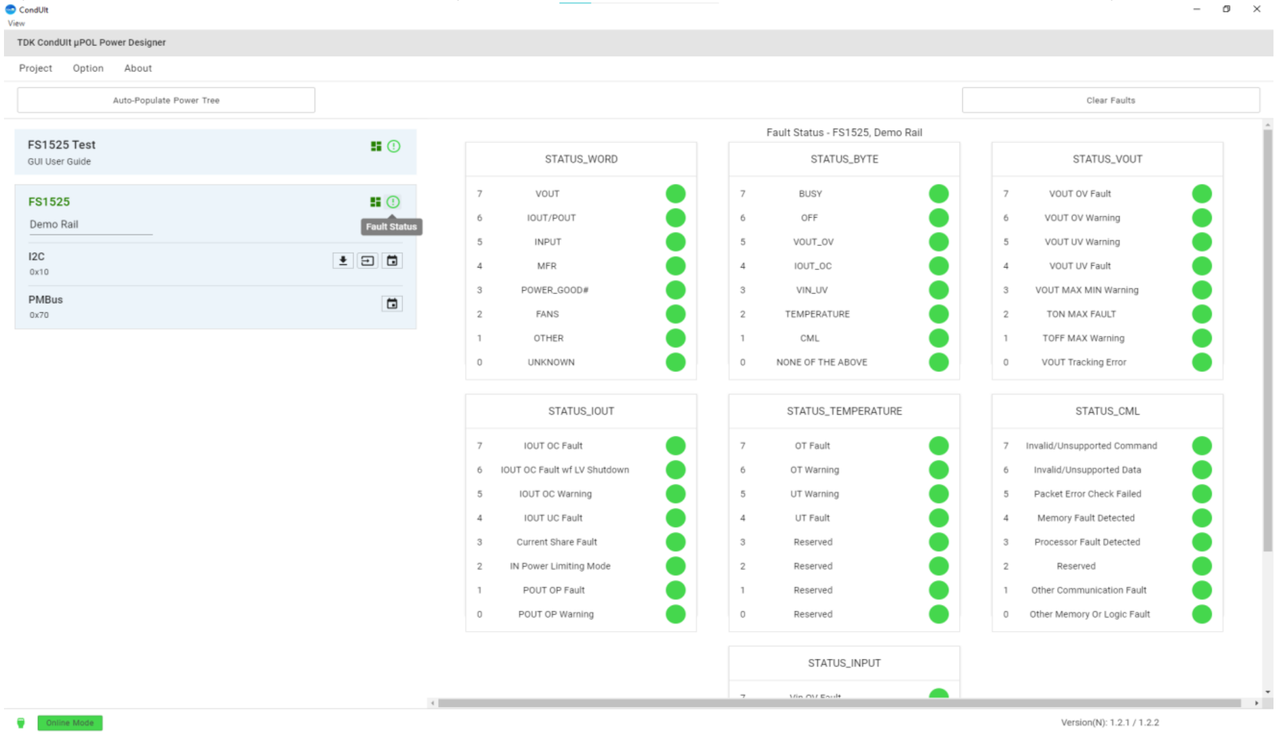
- SMBALERT_MASK - 1B
- OPERATION - 1
- ON_OFF_CONFIG - 2
- WRITE_PROTECT - 10
- CAPABILITY - 19
- VOUT_MODE - 20
- VOUT_OV_FAULT_RESPONSE - 41
- VOUT_UV_FAULT_RESPONSE - 45

History

Date & Time	Command Code	Command	Action	Real/Register Values	Hex/Mask Values
2024/04/24 13:35:24:987	1	OPERATION	W	On	80
2024/04/24 13:35:18:781	1	OPERATION	R	Off	00
2024/04/24 13:35:13:68	1	OPERATION	W	Off	00

Version(N): 1.2.1 / 1.2.2

9. Device FAULT Status : There is a dedicated area in GUI indicating any Faults that happen during the device operation. This is an example of FS1525 Fault Status without any faults -



Fault Status - FS1525, Demo Rail

STATUS_WORD		
7	VOUT	●
6	IOUT/POUT	●
5	INPUT	●
4	MFR	●
3	POWER_GOOD#	●
2	FANS	●
1	OTHER	●
0	UNKNOWN	●

STATUS_BYTE		
7	BUSY	●
6	OFF	●
5	VOUT_OV	●
4	IOUT_OC	●
3	VIN_UV	●
2	TEMPERATURE	●
1	CML	●
0	NONE OF THE ABOVE	●

STATUS_VOUT		
7	VOUT OV Fault	●
6	VOUT OV Warning	●
5	VOUT UV Warning	●
4	VOUT UV Fault	●
3	VOUT MAX MIN Warning	●
2	TON MAX FAULT	●
1	TOFF MAX Warning	●
0	VOUT Tracking Error	●

STATUS_IOUT		
7	IOUT OC Fault	●
6	IOUT OC Fault wf LV Shutdown	●
5	IOUT OC Warning	●
4	IOUT UC Fault	●
3	Current Share Fault	●
2	IN Power Limiting Mode	●
1	POUT OP Fault	●
0	POUT OP Warning	●

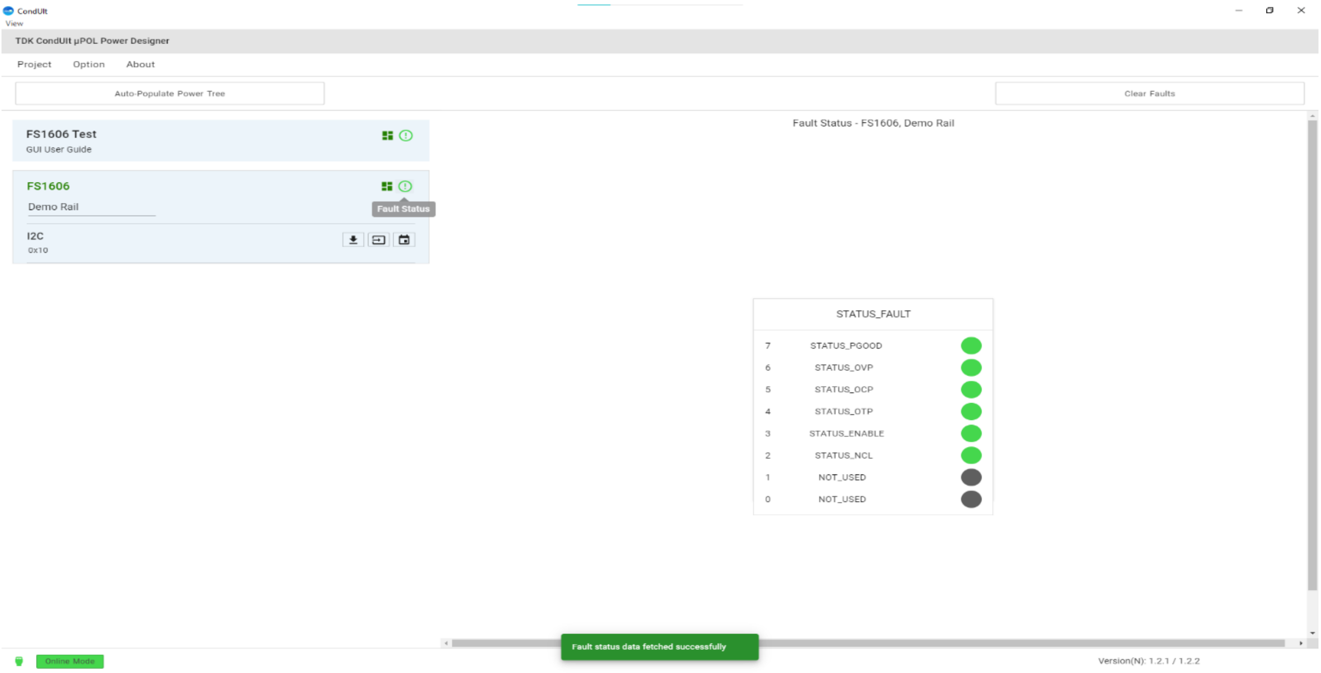
STATUS_TEMPERATURE		
7	OT Fault	●
6	OT Warning	●
5	UT Warning	●
4	UT Fault	●
3	Reserved	●
2	Reserved	●
1	Reserved	●
0	Reserved	●

STATUS_CML		
7	Invalid/Unsupported Command	●
6	Invalid/Unsupported Data	●
5	Packet Error Check Failed	●
4	Memory Fault Detected	●
3	Processor Fault Detected	●
2	Reserved	●
1	Other Communication Fault	●
0	Other Memory Or Logic Fault	●

STATUS_INPUT		
7	Vin OV Fault	●

Version(N): 1.2.1 / 1.2.2

For FS160X/FS100X/FS140X, there is no PMBUS functionality, and the fault status page is much simpler –

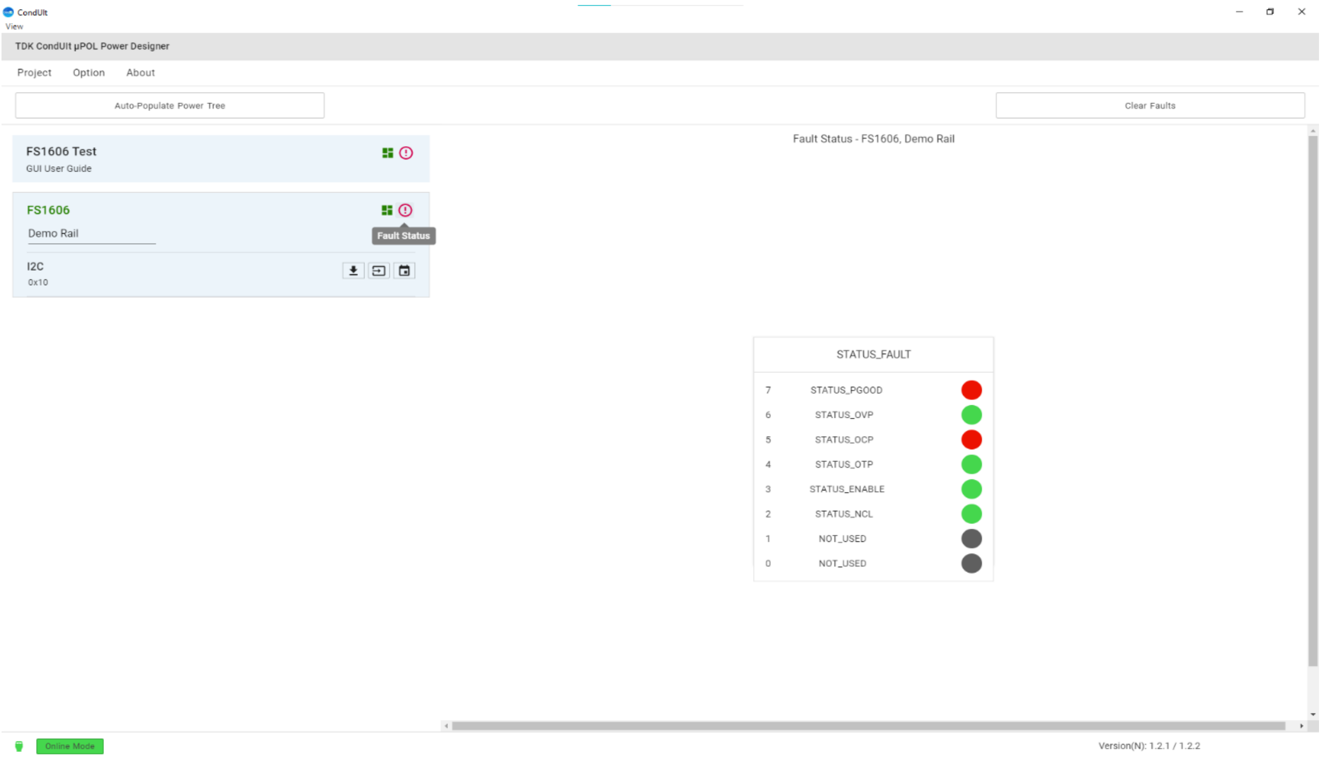


Fault Status - FS1606, Demo Rail

STATUS_FAULT		
7	STATUS_PGOOD	🟢
6	STATUS_OVP	🟢
5	STATUS_OCP	🟢
4	STATUS_OTP	🟢
3	STATUS_ENABLE	🟢
2	STATUS_NCL	🟢
1	NOT_USED	⚫
0	NOT_USED	⚫

Fault status data fetched successfully

This is an example of FS160X in OCP –

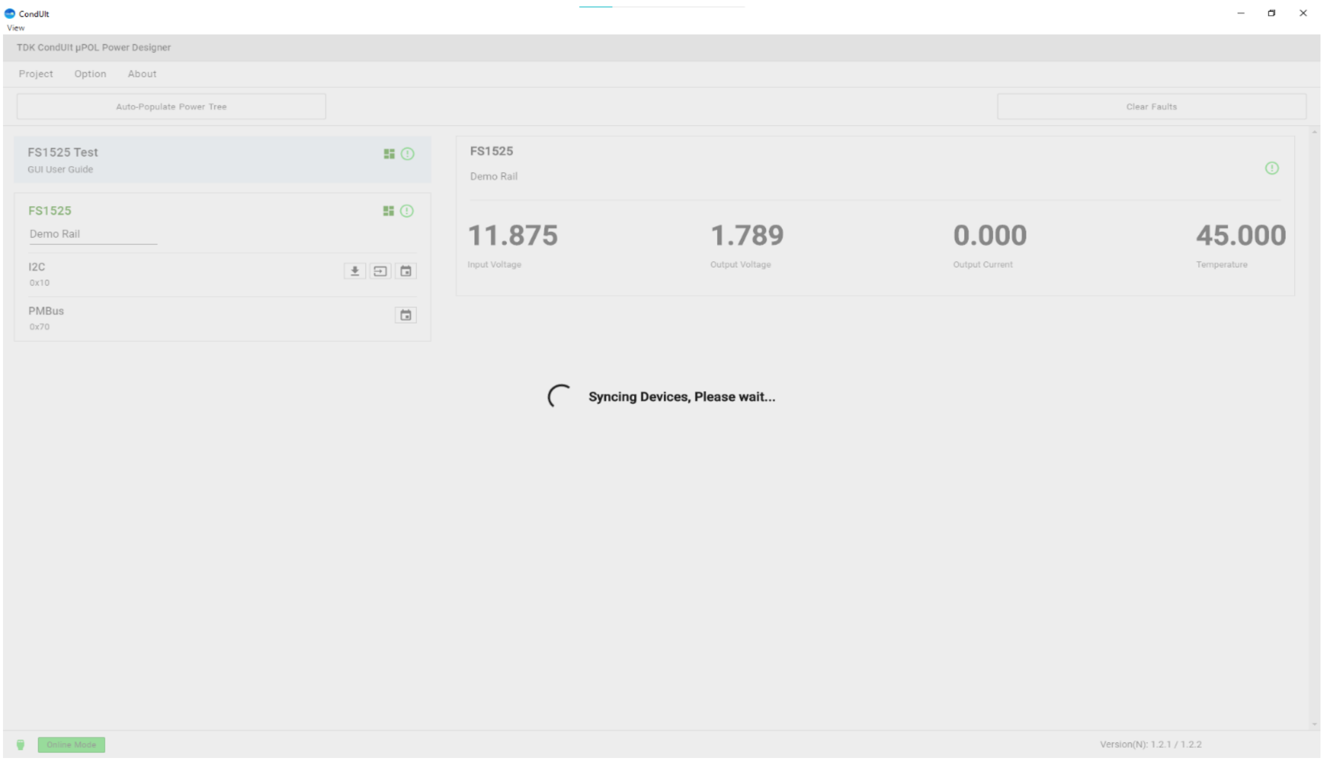


Fault Status - FS1606, Demo Rail

STATUS_FAULT		
7	STATUS_PGOOD	🔴
6	STATUS_OVP	🟢
5	STATUS_OCP	🔴
4	STATUS_OTP	🟢
3	STATUS_ENABLE	🟢
2	STATUS_NCL	🟢
1	NOT_USED	⚫
0	NOT_USED	⚫

NOTE – The fault bits for FS160X/100X/FS140X are stickied, and removing the Fault won't change the flag from Red to Green. Also, the Clear Faults tab on top right is tied to PMBUS and thus only works for FS1525/FS1412. As such, to reset the bit, En Cycling needs to be done. (NCL needs Power Cycling)

10. Opening an already created Project.
    Go to Project>Open and select the project that needs to be accessed. Once selected, the following screen pops up –

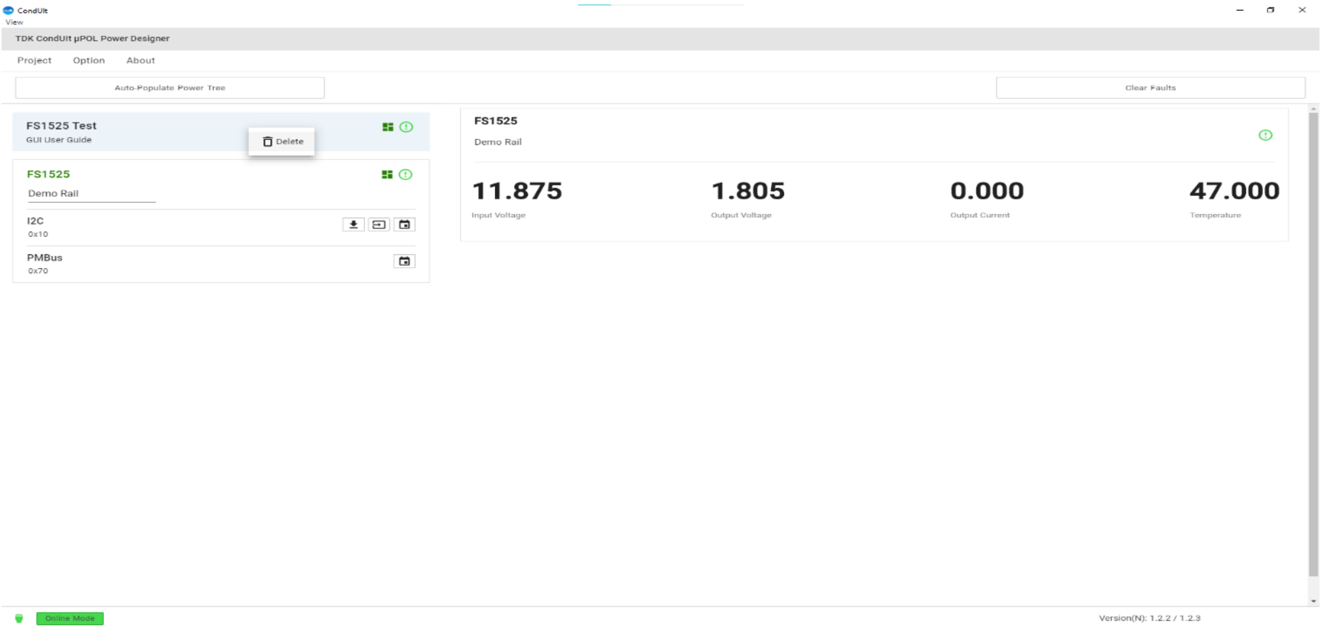


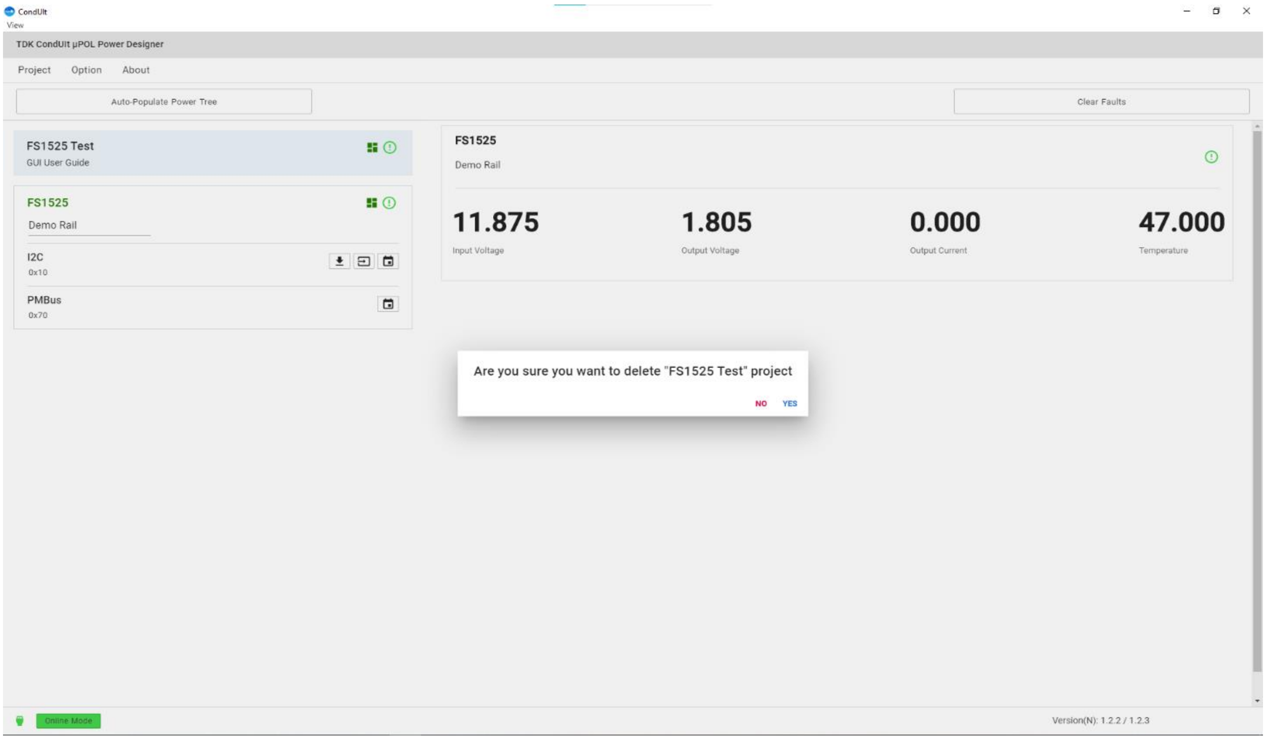
11.875	1.789	0.000	45.000
Input Voltage	Output Voltage	Output Current	Temperature

Syncing Devices, Please wait...

After Syncing is complete, the project will be ready for use.

11. Deleting a Project
    Right Click on the Dashboard Panel and Delete icon will come up.





Are you sure you want to delete "FS1525 Test" project

APPENDIX:

INSTRUCTIONS to burn OTP for FS1412

1) Once all registers are written to desired values, read all the values into a configuration file as (register, data) pairs

2) Read Reg 0x92 to check number of available user banks

Reg 0x92[7:4]	Number of writes left
0001	9
0010	8
0011	7
0100	6
0101	5
0110	4
0111	3
1000	2
1001	1
1010	0

3) Apply 7.5V+/-0.25V to Vin pin

4) If number of writes left > 0, write 0x02 to reg 0x89
5) Read reg 0x93. If bit [1] is 1, the write to OTP succeeded. If this bit is 0, the write failed.
6) If successful, cycle Vin.
7) Verify step:
   Read all regs, compare with the values in configuration file, and verify that they Match
8) If steps 5 or 7 fail, retry steps 1 to 4.
9) If steps 5 or 7 fail again, debug.

INSTRUCTIONS to burn OTP for FS160X/FS100X

1) Once all registers are written to desired values, read all the values into a configuration file as (register, data) pairs

2) Read Reg 0x20 to check number of available banks

Reg 0x20[5:0]	Number of writes left
001001	3
010010	2
011011	1
100100	0

3) Apply 7.5V+/-0.25V to Vin pin

4) If number of writes left > 0, write 0x15 to reg 0x2B, make sure bit[1] of reg 0x1D is 1 (turn on OTP clock)

5) Read reg 0x21. If bit [0] is 1, the write to OTP succeeded. If this bit is 0, the write failed.

6) If successful, cycle Vin.

7) Verify step:

   Read all regs, compare with the values in configuration file, and verify that they Match

10) If steps 5 or 7 fail, retry steps 1 to 4.

11) If steps 5 or 7 fail again, debug.

INSTRUCTIONS to burn OTP for FS1525/FS1515

1) Once all registers are written to desired values, read all the values into a configuration file as (register, data) pairs

2) Read Reg 0xF2 to check number of available user banks

Reg 0xF2[7:4]	Number of writes left
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A(10)

3) Apply 7.5V+/-0.25V to Vin pin

4) If number of writes left > 0, write 0x02 to reg 0xC9

5) Read reg 0xF3. If bit [1] is 1, the write to OTP succeeded. If this bit is 0, the write failed.

6) If successful, cycle Vin.

7) Verify step:
Read all regs, compare with the values in configuration file, and verify that they Match

8) If steps 5 or 7 fail, retry steps 1 to 4.

9) If steps 5 or 7 fail again, debug.

INSTRUCTIONS to burn OTP for FS140X

1) Once all registers are written to desired values, read all the values into a configuration file as (register, data) pairs

2) Read Reg 0x20 to check number of available banks

Reg 0x20[5:0]	Number of writes left
001001	3
010010	2
011011	1
100100	0

3) **Apply 7.5V+/-0.25V to Vin pin**
4) If number of writes left > 0, write 0x02 to reg 0x1D, make sure bit[1] of reg 0x1D is 1 (turn on OTP clock)
5) Read reg 0x21. If bit [1] is 1, the write to OTP succeeded. If this bit is 0, the write failed.
6) If successful, cycle Vin.
7) Verify step:

   Read all regs, compare with the values in configuration file, and verify that they Match
8) If steps 5 or 7 fail, retry steps 1 to 4.
9) If steps 5 or 7 fail again, debug.