XDPP1100 Development Environment Introduction

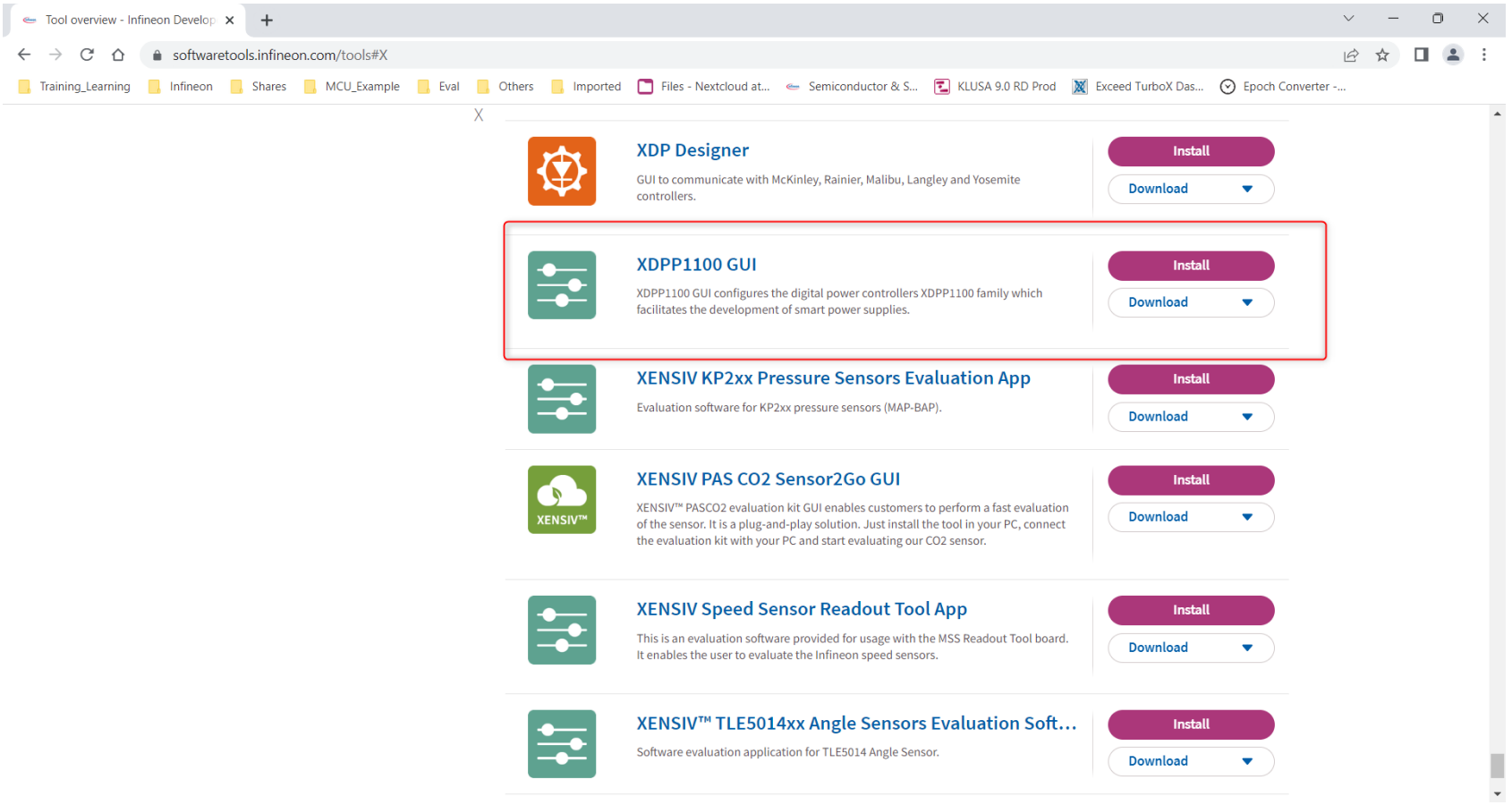
Ivan Seet
29/03/2023

Table of contents

XDPP1100 GUI Installation

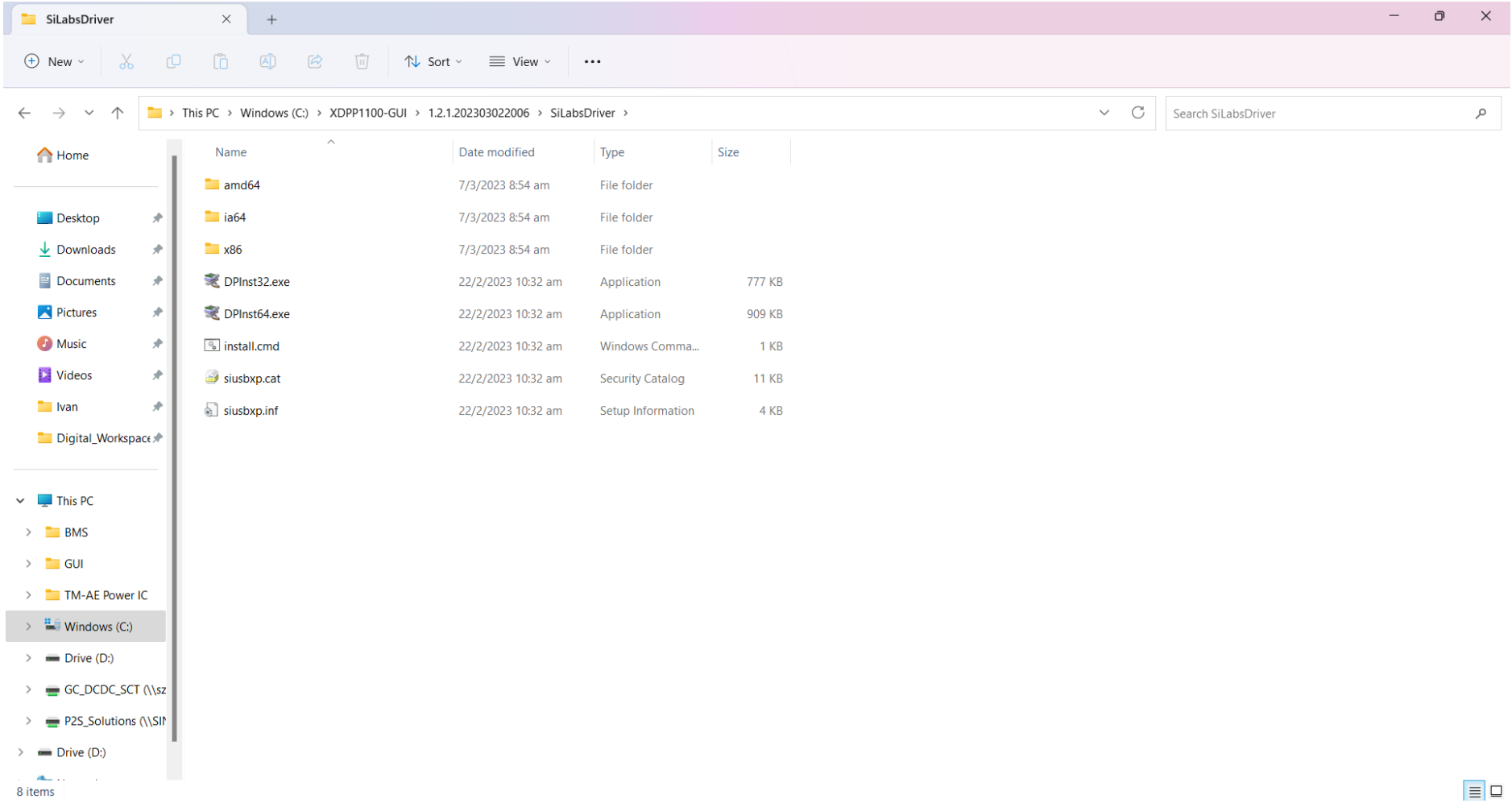
XDPP1100 Installation Package

› Install from Infineon Developer Center
› https://softwaretools.infineon.com/tools



Install USB700A/B Driver

› Click the “install.cmd”/ “DPInst64.exe” in the SiLabsDriver” folder



XDPP1100 Firmware development Environment

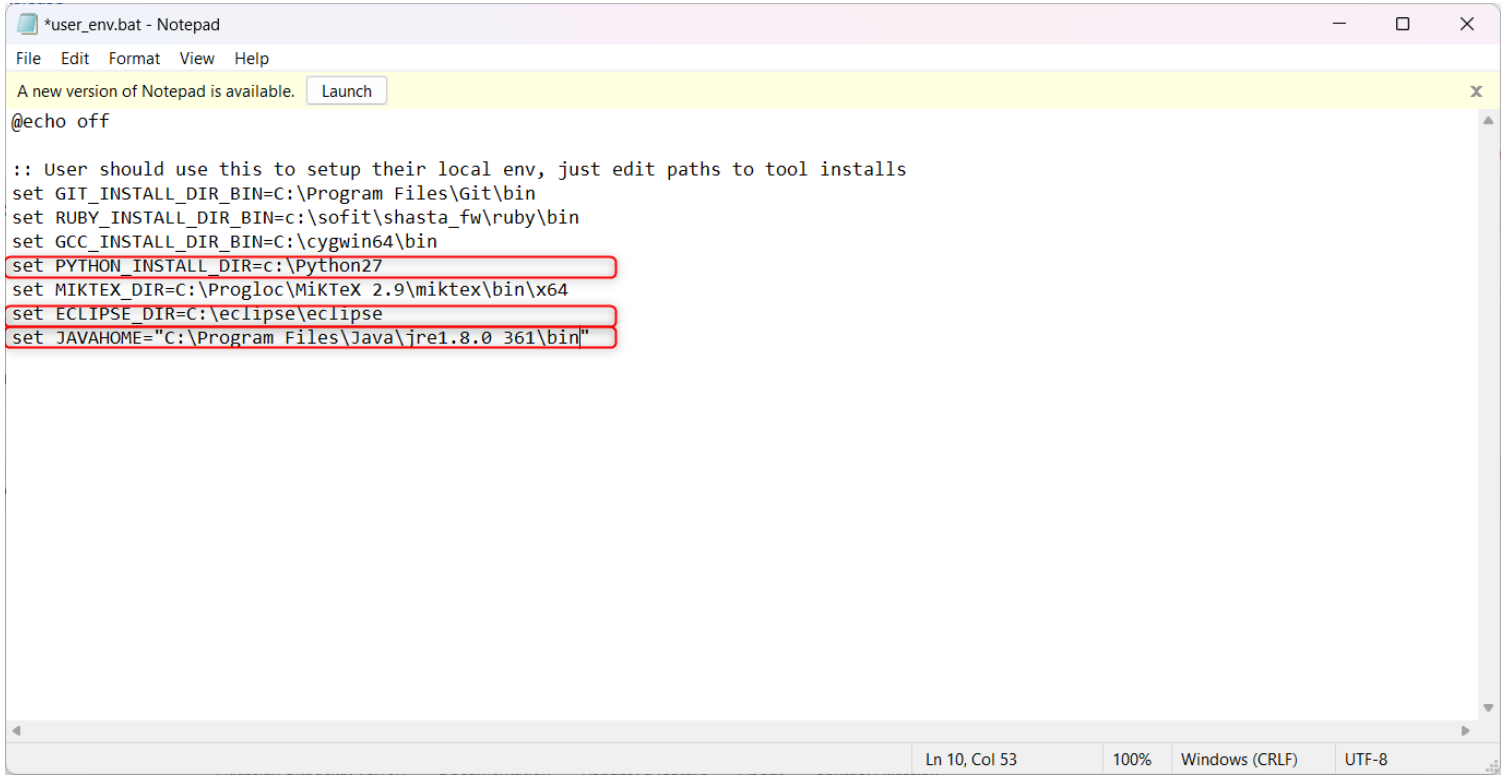
Firmware Environment Installation Procedures

- Clone the repo branch “projects/template.development”
  - https://bitbucket-dmz.infineon.com/projects/PICCR/repos/xdpp1100_fw_release/browse?at=refs%2Fheads%2Fprojects%2Ftemplate.development
- Python 2.7
  - https://www.python.org/downloads/release/python-2710/
- Eclipse(neon 3 version)
  - https://www.eclipse.org/downloads/packages/release/neon/3
- Java (Version 8 Update 341 for Eclipse neon 3 version)
  - https://www.java.com/en/download/windows_ie.jsp
  - Java for Windows Version 8 Update 341

*Note: If other version of eclipse is install, Java version will be different. https://wiki.eclipse.org/Eclipse/Installation

Eclipse Workbench Setup

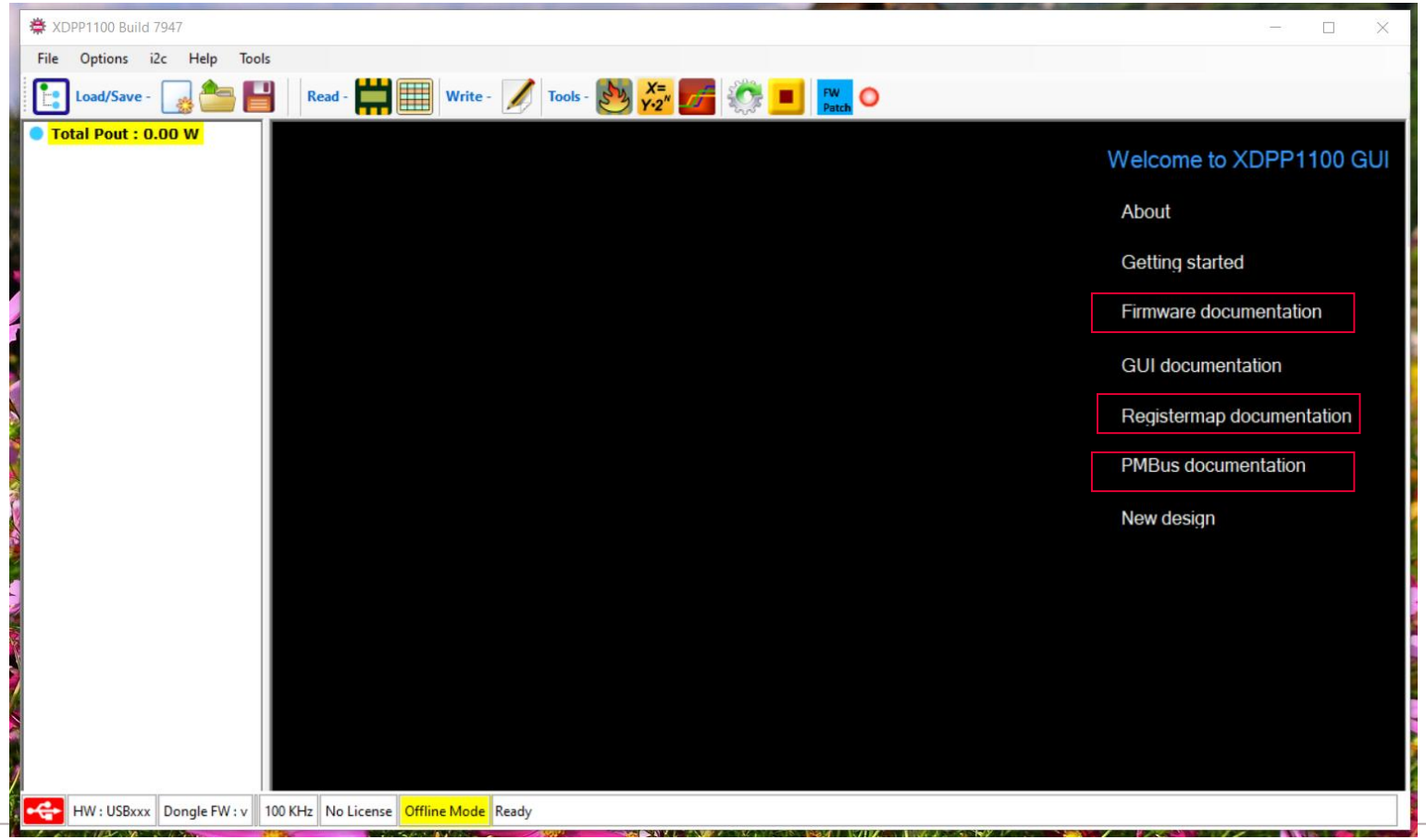
- Start by installing version of java, eclipse, python.
- If using git for firmware repository install this as well.
- Edit C:\XDPP1100\XDPP1100_fw\user_env.bat
- Set paths to required tools in this file:
  - Git, JAVAHOME, PYTHON and ECLIPSE



XDPP1100 GUI Overview

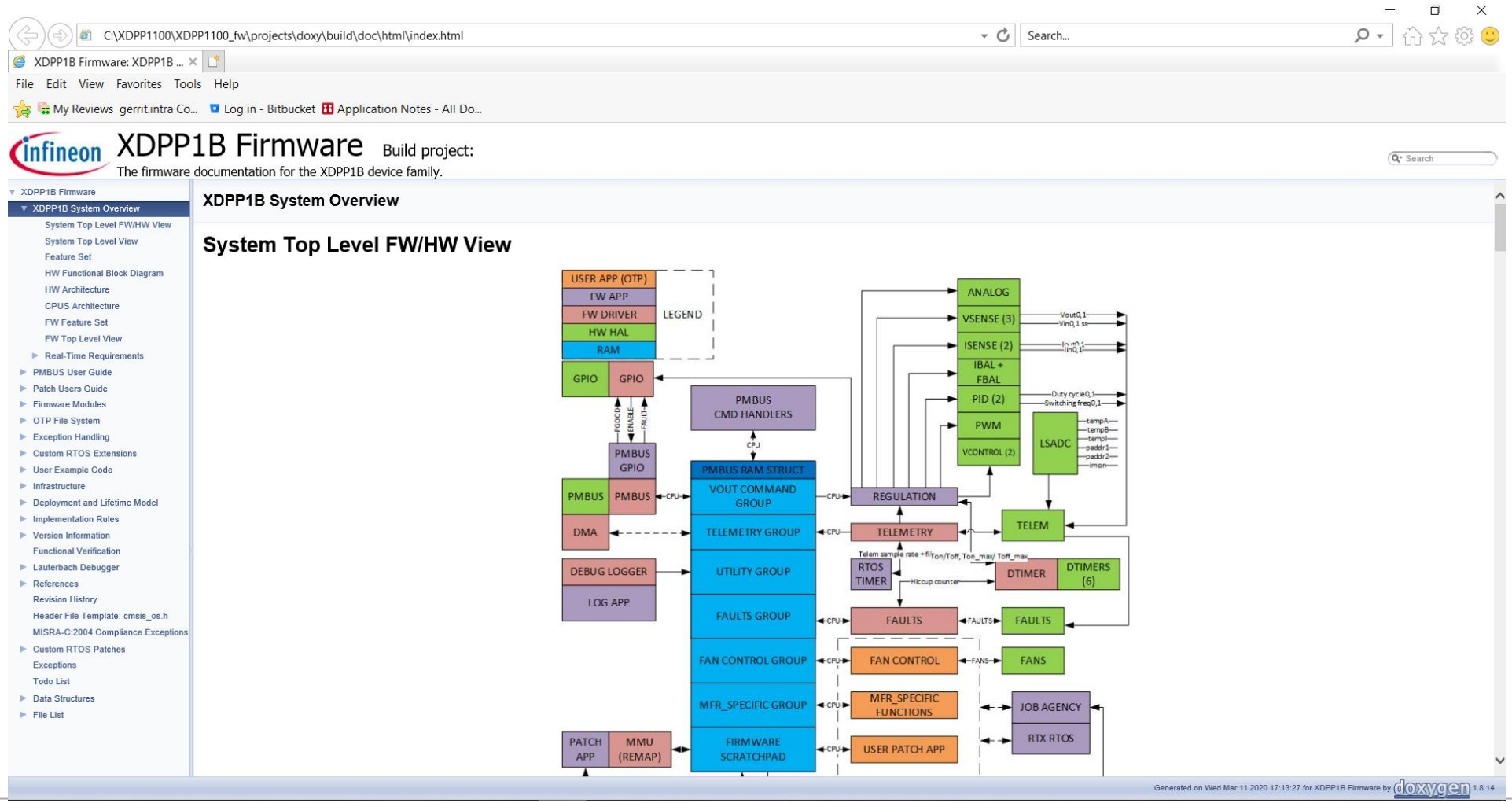
XDPP1100 GUI

- XDPP1100 GUI
  - Firmware documentation, Registermap documentation and PMBus documentation open the html document.



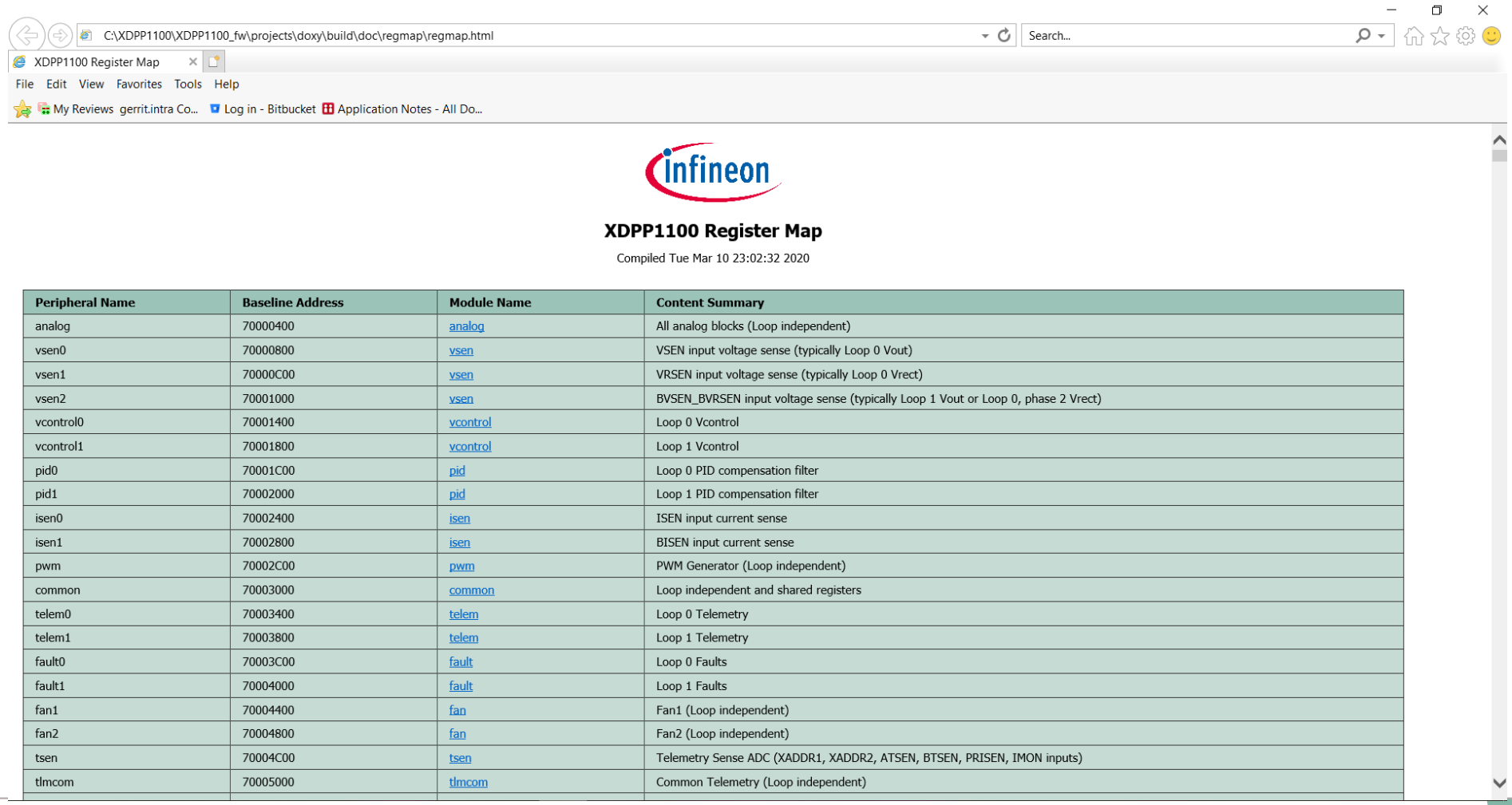
Firmware Documentation

› Firmware details Document



Registermap Documentation

› Register map List



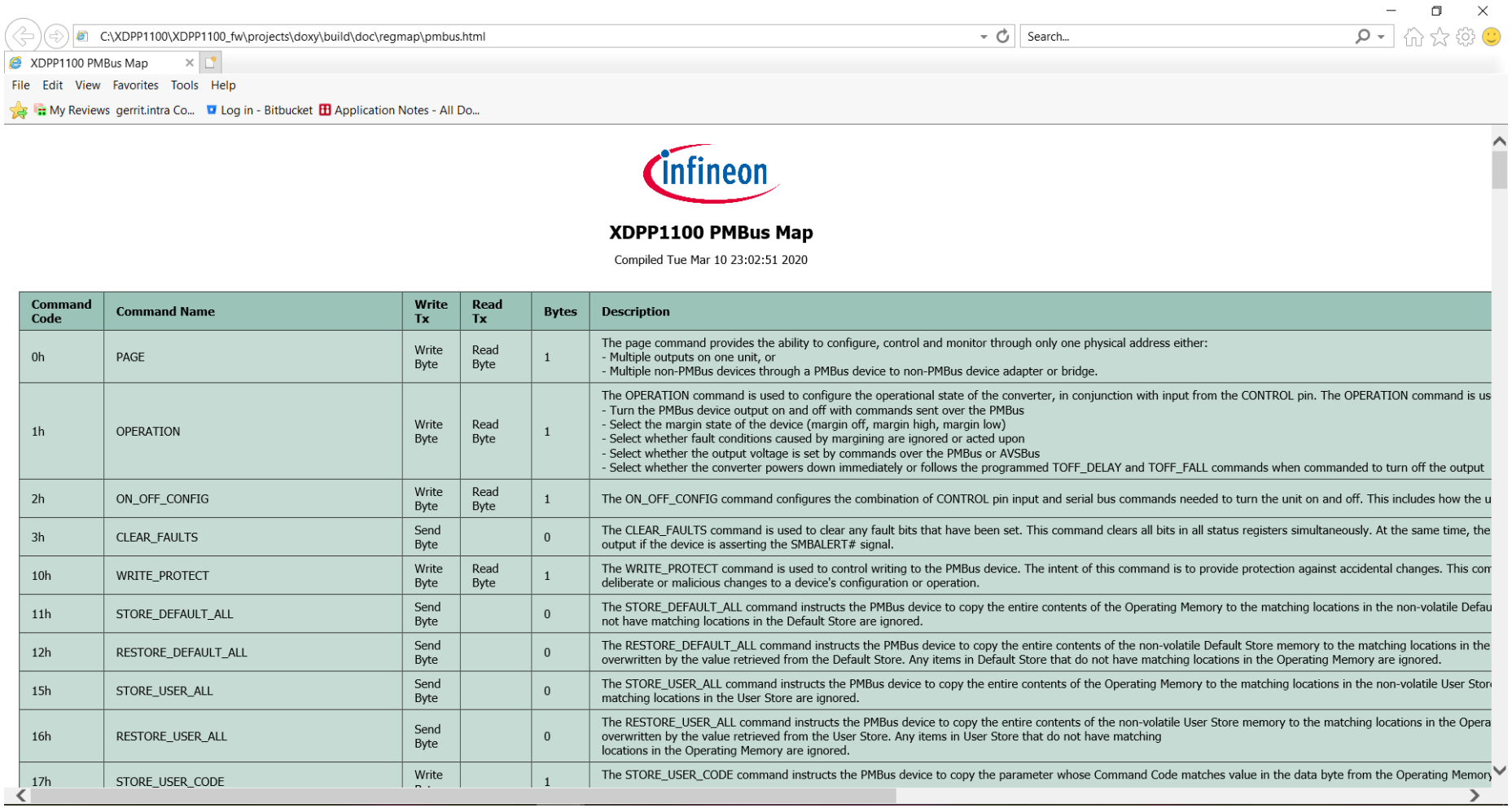
XDPP1100 Register Map

Compiled Tue Mar 10 23:02:32 2020

Peripheral Name	Baseline Address	Module Name	Content Summary
analog	70000400	analog	All analog blocks (Loop independent)
vsen0	70000800	vsen	VSEN input voltage sense (typically Loop 0 Vout)
vsen1	70000C00	vsen	VRSEN input voltage sense (typically Loop 0 Vrect)
vsen2	70001000	vsen	BVSEN_BVRSEN input voltage sense (typically Loop 1 Vout or Loop 0, phase 2 Vrect)
vcontrol0	70001400	vcontrol	Loop 0 Vcontrol
vcontrol1	70001800	vcontrol	Loop 1 Vcontrol
pid0	70001C00	pid	Loop 0 PID compensation filter
pid1	70002000	pid	Loop 1 PID compensation filter
isen0	70002400	isen	ISEN input current sense
isen1	70002800	isen	BISEN input current sense
pwm	70002C00	pwm	PWM Generator (Loop independent)
common	70003000	common	Loop independent and shared registers
telem0	70003400	telem	Loop 0 Telemetry
telem1	70003800	telem	Loop 1 Telemetry
fault0	70003C00	fault	Loop 0 Faults
fault1	70004000	fault	Loop 1 Faults
fan1	70004400	fan	Fan1 (Loop independent)
fan2	70004800	fan	Fan2 (Loop independent)
tsen	70004C00	tsen	Telemetry Sense ADC (XADDR1, XADDR2, ATSEN, BTSEN, PRISEN, IMON inputs)
tlmcom	70005000	tlmcom	Common Telemetry (Loop independent)

PMBus Documentation

› PMBus Commands List



XDPP1100 PMBus Map

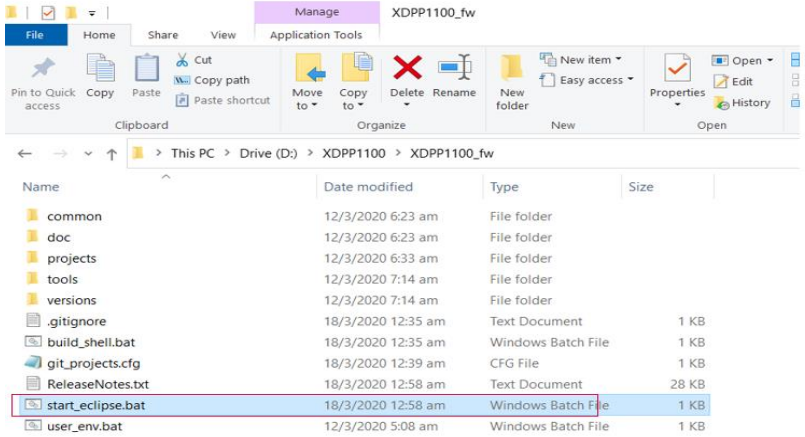
Compiled Tue Mar 10 23:02:51 2020

Command Code	Command Name	Write Tx	Read Tx	Bytes	Description
0h	PAGE	Write Byte	Read Byte	1	The page command provides the ability to configure, control and monitor through only one physical address either: - Multiple outputs on one unit, or - Multiple non-PMBus devices through a PMBus device to non-PMBus device adapter or bridge.
1h	OPERATION	Write Byte	Read Byte	1	The OPERATION command is used to configure the operational state of the converter, in conjunction with input from the CONTROL pin. The OPERATION command is us - Turn the PMBus device output on and off with commands sent over the PMBus - Select the margin state of the device (margin off, margin high, margin low) - Select whether fault conditions caused by margining are ignored or acted upon - Select whether the output voltage is set by commands over the PMBus or AVSBus - Select whether the converter powers down immediately or follows the programmed TOFF_DELAY and TOFF_FALL commands when commanded to turn off the output
2h	ON_OFF_CONFIG	Write Byte	Read Byte	1	The ON_OFF_CONFIG command configures the combination of CONTROL pin input and serial bus commands needed to turn the unit on and off. This includes how the u
3h	CLEAR_FAULTS	Send Byte		0	The CLEAR_FAULTS command is used to clear any fault bits that have been set. This command clears all bits in all status registers simultaneously. At the same time, the output if the device is asserting the SMBALERT# signal.
10h	WRITE_PROTECT	Write Byte	Read Byte	1	The WRITE_PROTECT command is used to control writing to the PMBus device. The intent of this command is to provide protection against accidental changes. This com deliberate or malicious changes to a device's configuration or operation.
11h	STORE_DEFAULT_ALL	Send Byte		0	The STORE_DEFAULT_ALL command instructs the PMBus device to copy the entire contents of the Operating Memory to the matching locations in the non-volatile Defau not have matching locations in the Default Store are ignored.
12h	RESTORE_DEFAULT_ALL	Send Byte		0	The RESTORE_DEFAULT_ALL command instructs the PMBus device to copy the entire contents of the non-volatile Default Store memory to the matching locations in the overwritten by the value retrieved from the Default Store. Any items in Default Store that do not have matching locations in the Operating Memory are ignored.
15h	STORE_USER_ALL	Send Byte		0	The STORE_USER_ALL command instructs the PMBus device to copy the entire contents of the Operating Memory to the matching locations in the non-volatile User Stor matching locations in the User Store are ignored.
16h	RESTORE_USER_ALL	Send Byte		0	The RESTORE_USER_ALL command instructs the PMBus device to copy the entire contents of the non-volatile User Store memory to the matching locations in the Opera overwritten by the value retrieved from the User Store. Any items in User Store that do not have matching locations in the Operating Memory are ignored.
17h	STORE_USER_CODE	Write		1	The STORE_USER_CODE command instructs the PMBus device to copy the parameter whose Command Code matches value in the data byte from the Operating Memory

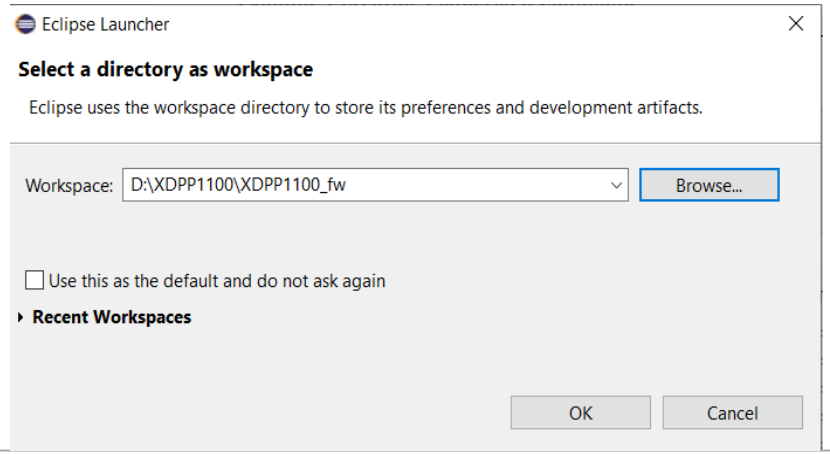
Integrated Development Environment Setup

Start Eclipse Environment

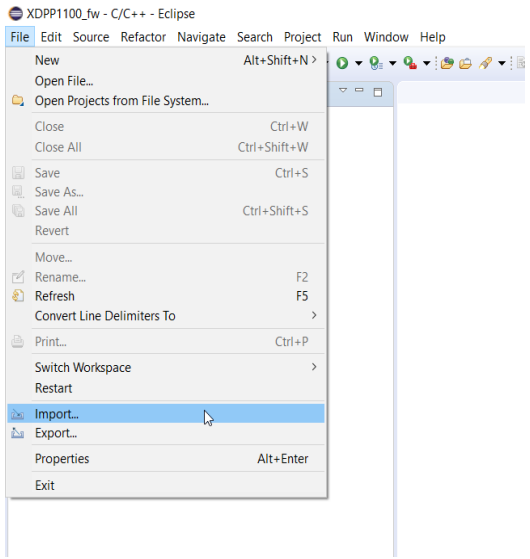
› Start eclipse by double clicking ….\xdpp1100_fw_release\start_eclipse.bat

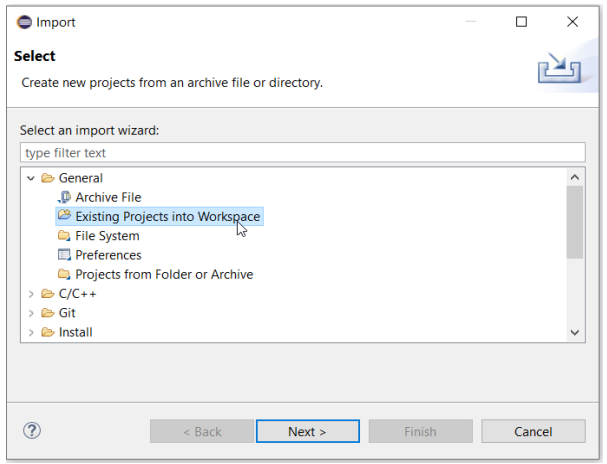


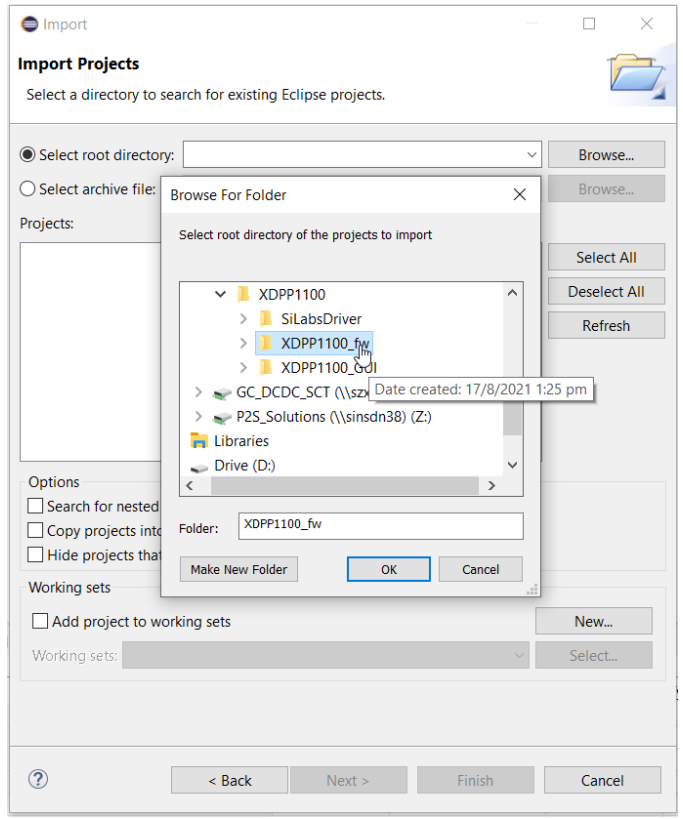
› Select a Workspace. E.g select location XDPP1100_fw

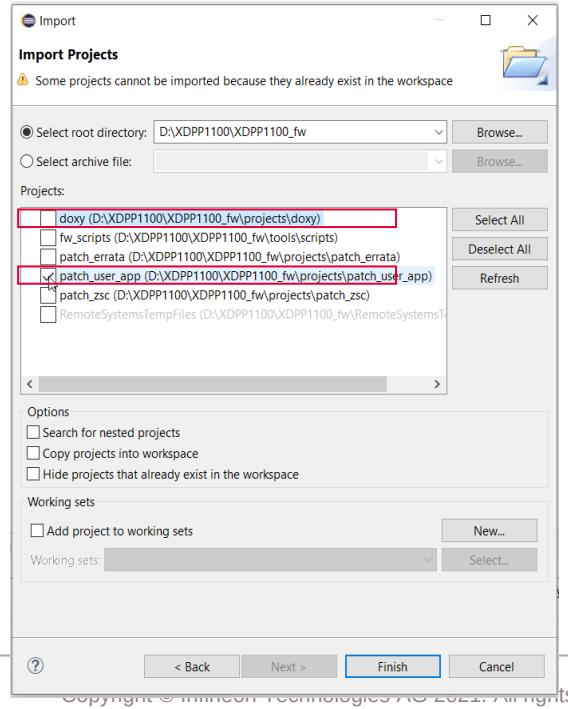


Step 1. Import existing patch_user_app



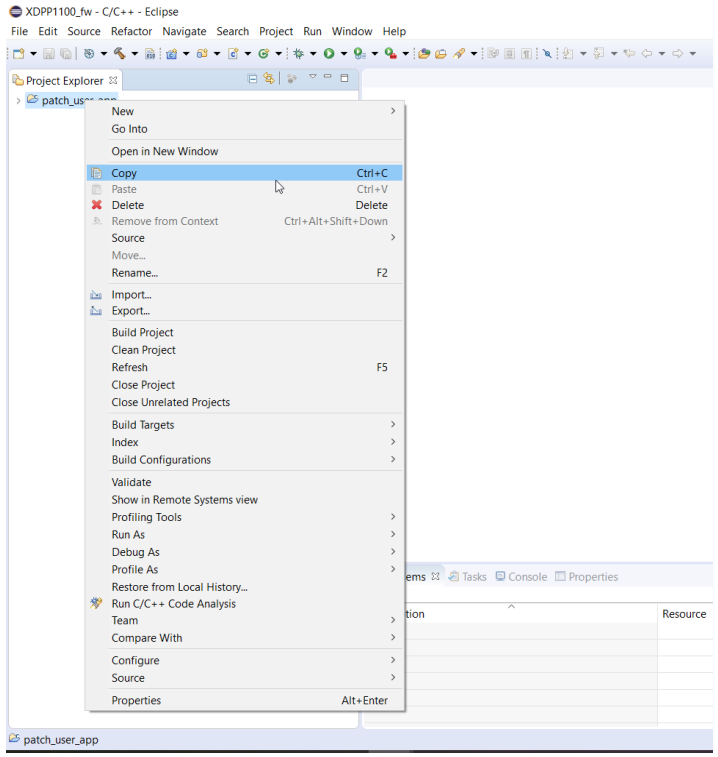


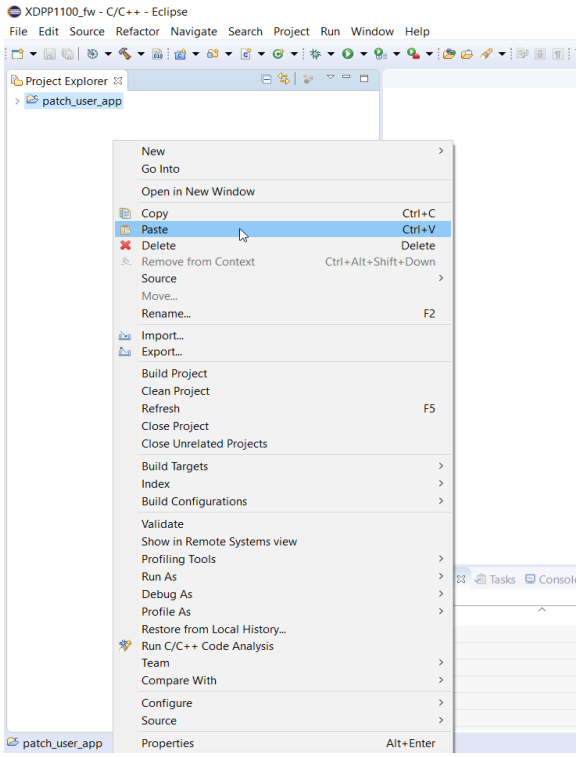


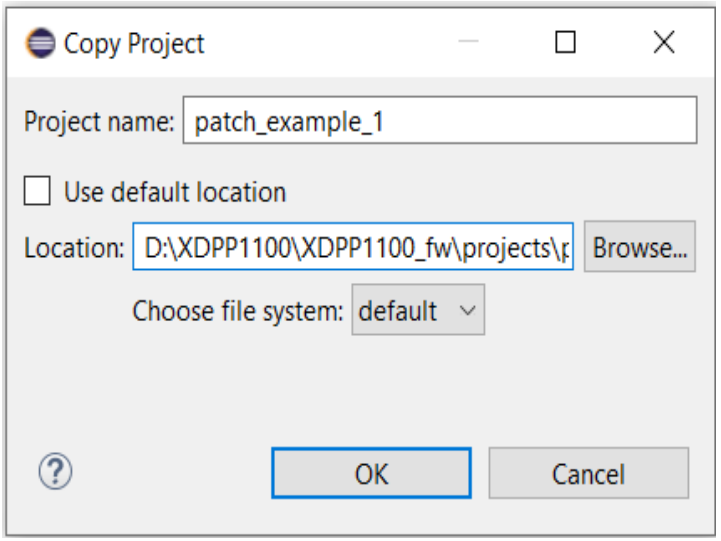


Step 2. Copy patch_user_app and Paste to create new project

› Key the project name (e.g patch_example_1) and select location as ..\XDPP1100\XDPP1100_fw\projects\patch_example_1.

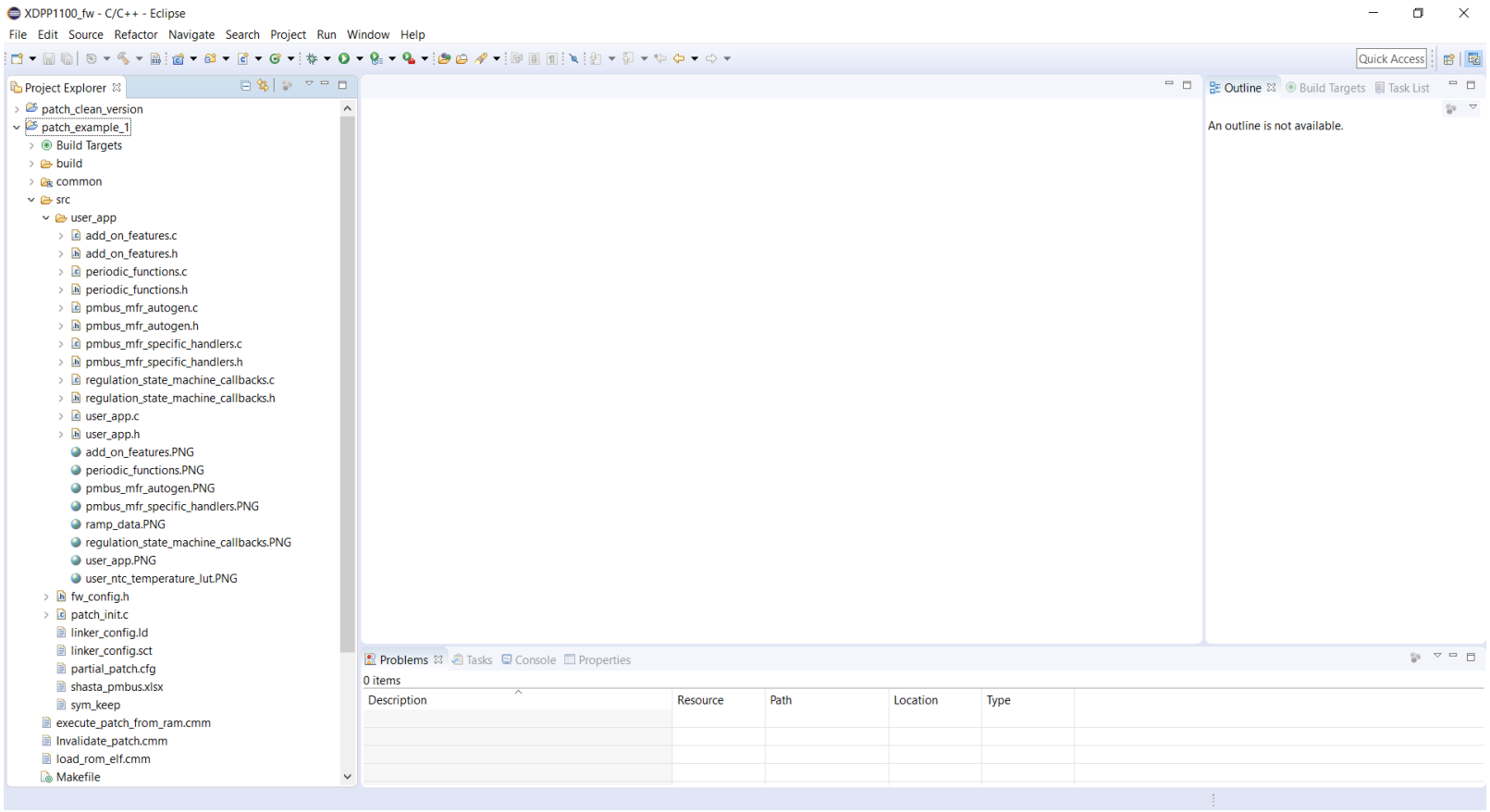






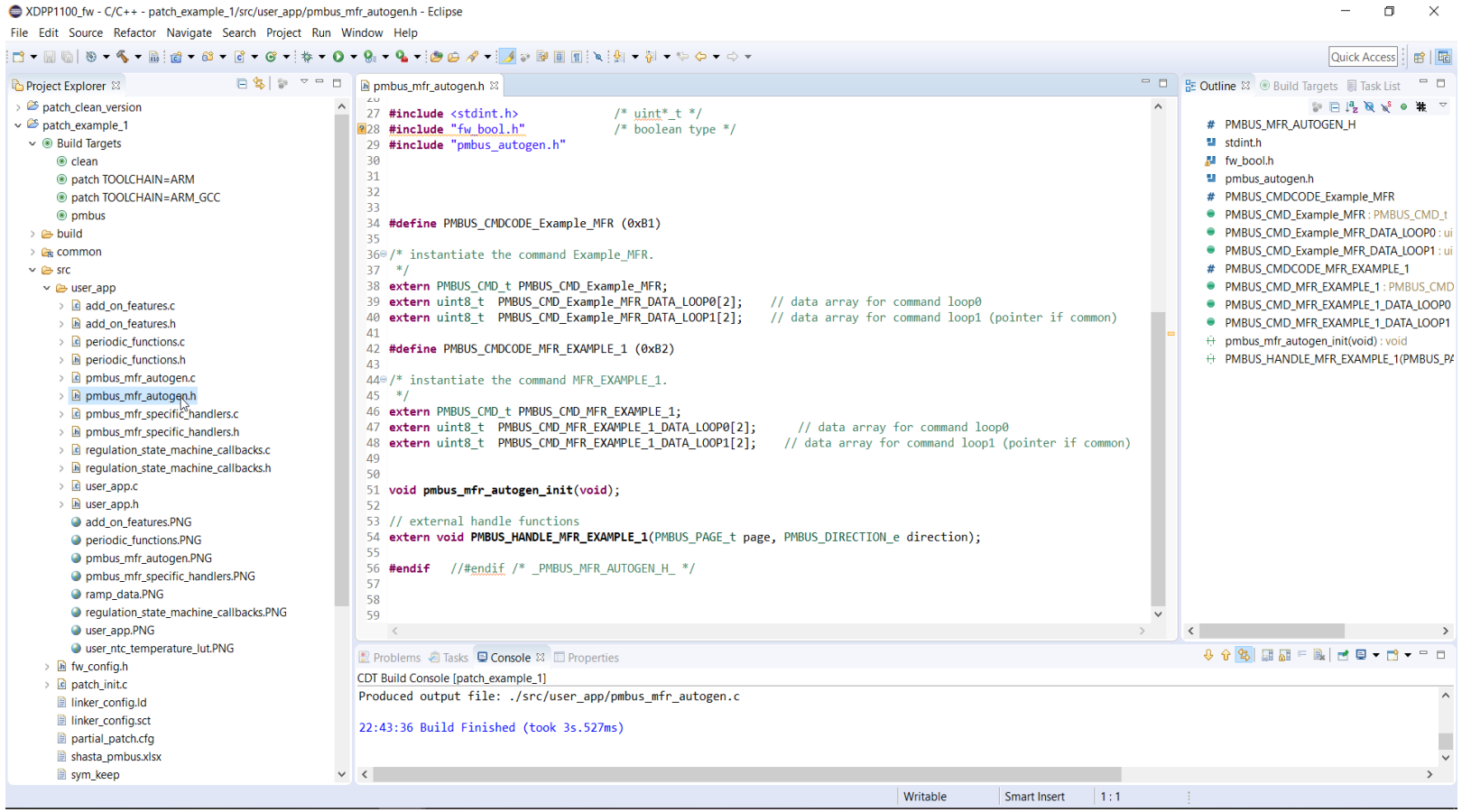
Step 3. New Project Created

- New project is created for new firmware patch development



Step 4. Compile new created project

› Expand the “Build Targets” to clean and compile patch using “patch TOOLCHAIN==ARM_GCC”



Number Format

Q-number Format

- Q-number format enables representation of fractions and decimal points by signed integers, making rational numbers possible in the CPU.
- Q-number format is usually denoted as **Qm.n**, where:
  - **m** represents the **integers**
  - **n** represents the **decimal points/fractions**
- Using Q-number format perspective:
  - **Unsigned integer** representation can be denoted as **Ux.0**
  - **Signed integer** representation can be denoted as **Sx.0**

Format	Bit length	Min. value		Max. value	
		BIN	DEC	BIN	DEC
U8.0	8	0000 0000	0	1111 1111	255
S8.0	8	1000 0000	-128	0111 1111	127

Q-number Format

- Adding decimal points/fractions can be done by setting the value of “n”.
  - Setting **n** to **3** on the above U8.0 and S8.0 examples yields Q-numbers of U8.3 and S8.3, respectively.
  - Adding “n” will also lengthen the binary numbers from 8 to 11, and therefore the new bit length can be calculated as: $Bit\ length\ (N) = m + n$
- In U8.0 and S8.0, every increment of LSB corresponds to an increased value of 1 in DEC value.
  - In U8.3 and S8.3, however, every increment of LSB corresponds to a different value.
    - This property is called “LSB weight”, which can be calculated as: $LSB\ weight = 2{-}(n)$

Format	Bit Length	LSB Weight	Min Value (DEC)	Max Value (DEC)
U8.3	8 + 3 = 11	2^(-3) = 0.125	0	(2(11) - 1) * (2(-3)) = 255.875
S8.3	8 + 3 = 11	2^(-3) = 0.125	(-(2(11 – 1))) * (2(-3)) = -128	(2(11 – 1) - 1) * (2(-3)) = 127.875

Format	Bit Length	LSB Weight	Min Value (BIN)	Max Value (BIN)
U8.3	8 + 3 = 11	2^(-3) = 0.125	0000 0000.000	1111 1111.111
S8.3	8 + 3 = 11	2^(-3) = 0.125	1000 0000.000	0111 1111.111

Example

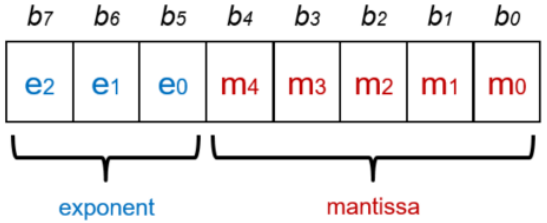
Format	Bit Length	LSB Weight	Min Value (DEC)	Max Value (DEC)
U8.3	8 + 3 = 11	2^(-3) = 0.125	0	255.875
U3.8	8 + 3 = 11	2^(-8) = 0.00390625	0	7.99609375
S8.3	8 + 3 = 11	2^(-3) = 0.125	-128	127.875
S3.8	8 + 3 = 11	2^(-8) = 0.00390625	-4	3.99609375

Format	Bit Length	LSB Weight	Min Value (DEC)	Max Value (DEC)
U8.-3	8 + (-3) = 5	2^(-(-3)) = 8	0	248
U3.-8	3 + (-8) = -5 (Invalid)			
S8.-3	8 + (-3) = 5	2^(-(-3)) = 8	-128	120
S3.-8	3 + (-8) = -5 (Invalid)			

Format	Bit Length	LSB Weight	Min Value (DEC)	Max Value (DEC)
U-8.3	-8 + 3 = -5 (Invalid)			
U-3.8	-3 + 8 = 5	2^(-8) = 0.00390625	0	0.12109375
S-8.3	-8 + 3 = -5 (Invalid)			
S-3.8	-3 + 8 = 5	2^(-8) = 0.00390625	-0.0625	0.05859375

LINEAR11

› Rational Numbers can also be represented in Exponent-Mantissa format



e = Exponent bit-length

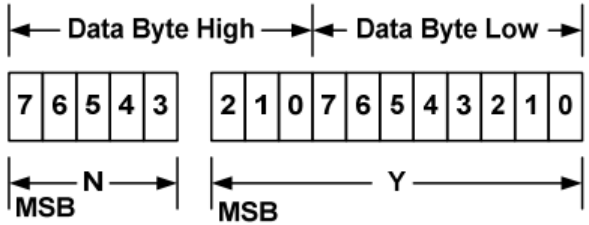
m = Mantissa bit-length

w = Weighting factor

Property	Formula	Representation
Bit Length	e + m	R[(e + m - 1) : 0]
Exponent		R[(e + m - 1) : m]
Mantissa	2m + R[(m - 1) : 0]	R[(m - 1) : 0]
Rational Numbers	Mantissa * 2Exponent * 2w	R[(e + m + 1) : 0]

LINEAR11

› PMBus Linear Data Format is represented as following



› The relation between Y, N and the “real world” value X is:
- $X = Y \cdot 2^N$
- Where X is “read world” value,
- Y is an 11-bit, two’s complement integer (also called Mantissa)
- N is a 5-bit, two’s complement integer (also called Exponent).

Examples

Real World value (X)	Exponent (N)	Exponent (N)	Mantissa (Y)	Mantissa (Y)	16-bit Representation
Real World value (X)	DEC	BIN	DEC	BIN	16-bit Representation
0.5	-1	1 1111	1	000 0000 0001	0xF801
0.5	-9	1 0111	256	001 0000 0000	0xB900
0.5625	-4	1 1100	9	000 0000 1001	0xE009
0.5625	-5	1 1011	18	000 0001 0010	0xD812
-45.375	-3	1 1110	-363	110 1001 0101	0xEE95
-45.375	-4	1 1100	-726	101 0010 1010	0xE52A

API use for Conversion

› API that can be use for conversion

```
// LINEAR11 to Q-number
// This is equivalent to 36.0 in decimal.
uint16_t LIN11_num = 0xE920;

// Get the Mantissa
int32_t qNum_man = LINEAR11_TO_MANTISSA(LIN11_num);
// Get the Exponent
int32_t qNum_exp = LINEAR11_TO_EXPONENT(LIN11_num);

// Shift mantissa based on exponent and get original number
int32_t qNum = SHIFT_EXPONENT(qNum_man, qNum_exp);

printf("%d", qNum); // print out 36
```

```
// Q-Number to Linear11
// qNum is the Q number to be converted.
int32_t qNum = 36;
int8_t exponent = -3;

// Calculate LINEAR11 exponent
uint8_t LIN11_expo = TWOS_COMPLEMENT(5, exponent);
// Calculate LINEAR11 mantissa
int16_t LIN11_mant = qNum << -LIN11_expo;

// Shift mantissa based on exponent and get original number
uint16_t LIN11_num = LIN11_expo << 11 | LIN11_mant & 0x7FF;

printf("%x", LIN11_num); // print out 0xE920
```

Others numbering system

- Some registers that are directly interfaced to the hardware block may already have pre-assigned binary point and LSB weighting.
  - Voltages referenced to the VS ADC's (e.g., internal vout, vrect and vcontrol) have a LSB at 1.25mV. This was chosen to match the VS ADC LSB weight.
  - Other voltages generally place the binary point at 1V to match PMBus.
  - Currents generally place the binary point at 1A to match PMBus.
  - Resistances (e.g., for Droop/loadline) generally place the binary point at 1mOhm to match PMBus.
  - Temperatures generally place the binary point at 1C to match PMBus.
  - Powers generally place the binary point at 1W to match PMBus.
  - Time parameters generally place the binary point based on the HW clock period associated with the parameter. Some variations in binary points are possible such as 5ns, 10ns, 20ns, etc. For some longer time parameters, binary points at 1ms is possible in order to match PMBus.

Useful Link

Useful Web Link

- Public
  - https://www.infineon.com/cms/en/product/power/dc-dc-converters/digital-power-controllers/
  - https://www.infineon.com/dgdl/Infineon-DCDC_Converter_XDP_digital_power_XDPP1100_Firmware-Software-v01_00-EN.zip?fileId=5546d46279cccfdb0179ea3e3550019e&da=t

- myInfineon account (Public with myInfineon account)
  - https://www.infineon.com/cms/en/myInfineon/p/profile/#/productRegistration
  - https://softwaretools.infineon.com/projects/create (Register Demo board for documents)

- myInfineon Collaboration Platform (Required access right from PM)
  - https://myicp.infineon.com/sites/power_management/SitePages/default.aspx

Part of your life. Part of tomorrow.