

# QSFP\_DD API

## Documentation

## Table of Contents

1. Introduction .....	3
1.1. Acronyms and abbreviations .....	3
2. APIs .....	3
2.1. General Functions .....	3
2.1.1. USB Connection.....	3
▪ ConnectToHost.....	3
▪ Disconnect.....	3
2.1.2. Monitoring .....	3
▪ P3V3_Current_Monitor.....	3
▪ GetVCC .....	4
2.2. QSFP_DD MSA functions.....	4
2.2.1. I2C access .....	4
▪ I2CRead .....	4
▪ I2CWrite .....	4
2.2.2. Alarms and controls signals .....	5
▪ MODPRS .....	5
▪ LPMODE .....	5
▪ RESET .....	5
▪ MODSEL.....	5
▪ IntL .....	5

## 1. Introduction

This document describes the various Application Programming Interface (API) functions for the Multilane QSFP\_DD host board (ML4062). Each function is described with its parameters and return values.

### 1.1. Acronyms and abbreviations

API	Application Programming Interface
DLL	Dynamic Link Library (.dll file)
USB	Universal Serial Bus
I2C	Inter-Integrated Circuit

## 2. APIs

### 2.1. General Functions

#### 2.1.1. USB Connection

##### ▪ ConnectToHost

Description	Opens a USB connection to QSFP_DD Host
Call	<b>bool __stdcall ConnectToHost(UInt16 Instance)</b>
Parameters	UInt16 Instance: USB instance of plugged host
Returns	True or False

##### ▪ Disconnect

Description	Disconnects from CFP2 Host and close open USB connection
Call	<b>bool __stdcall Disconnect(UInt16 Instance)</b>
Parameters	UInt16 Instance: USB instance of plugged host
Returns	True or False

#### 2.1.2. Monitoring

##### ▪ P3V3\_Current\_Monitor

Description	Measures current value on the 3.3V line
Call	<b>bool __stdcall P3V3_Current_Monitor(UInt16 Instance, double* data)</b>
Parameters	UInt16 Instance: USB instance double* data: Current value in mA

<b>Returns</b>	True or False
----------------	---------------

#### ▪ GetVCC

<b>Description</b>	Measures voltage on the VCC1 line
<b>Call</b>	<b>bool __stdcall GetVCC(int Instance, double* Data)</b>
<b>Parameters</b>	UInt16 Instance: USB instance double* Data: VCC1 value in V
<b>Returns</b>	True or False

## 2.2. QSFP\_DD MSA functions

### 2.2.1. I2C access

#### ▪ I2CRead

<b>Description</b>	Reads I2C
<b>Call</b>	<b>bool __stdcall I2CRead(int Instance, BYTE SlaveAddress, BYTE registerAddress, BYTE* ReadBuff, BYTE Length);</b>
<b>Parameters</b>	int instance: USB instance BYTE SlaveAddress: Slave Address which is 0xA0 BYTE registerAddress: Register to read data from BYTE* ReadBuff: Pointer to the data that is read BYTE Length: Number of registers read sequentially (max=32)
<b>Returns</b>	True or False

#### ▪ I2CWrite

<b>Description</b>	Writes I2C
<b>Call</b>	<b>bool __stdcall I2CWrite(int Instance, BYTE SlaveAddress, BYTE Register , BYTE Value);</b>
<b>Parameters</b>	int instance: USB instance BYTE SlaveAddress: Slave Address which is 0xA0 BYTE Register: Register where the data will be written BYTE Value: The value to write on the Register
<b>Returns</b>	True or False

## 2.2.2. Alarms and controls signals

### ▪ MODPRS

<b>Description</b>	Reads the opposite of MODPRS_L QSFP_DD pin to check if the QSFP_DD module is inserted in the Host.
<b>Call</b>	<b>bool __stdcall MODPRS(UInt16 Instance, bool* status)</b>
<b>Parameters</b>	UInt16 Instance: USB instance bool* status: True if HW pin is 0 : module present False if HW pin is 1 : module absent
<b>Returns</b>	True or False

### ▪ LPMODE

<b>Description</b>	Asserts/Deasserts LPMODE
<b>Call</b>	<b>bool __stdcall LPMODE(UInt16 Instance, bool asserted)</b>
<b>Parameters</b>	UInt16 Instance: USB instance bool asserted: True to assert LPMODE False to deassert LPMODE
<b>Returns</b>	True or False

### ▪ RESET

<b>Description</b>	Asserts/Deasserts RESET (inverse of RESET_L pin)
<b>Call</b>	<b>bool __stdcall RESET (UInt16 Instance, bool asserted)</b>
<b>Parameters</b>	UInt16 Instance: USB instance bool asserted: True to assert RESET (RESET_L pin: 0) False to deassert RESET (RESET_L pin: 1)
<b>Returns</b>	True or False

### ▪ MODSEL

<b>Description</b>	Asserts/Deasserts MODSEL (inverse of MODSEL_L pin)
<b>Call</b>	<b>bool __stdcall MODSEL(UInt16 Instance, bool asserted)</b>
<b>Parameters</b>	UInt16 Instance: USB instance bool asserted: True to assert MODSEL (MODSEL_L pin: 0) False to deassert MODSEL (MODSEL_L pin: 1)
<b>Returns</b>	True or False

### ▪ IntL

<b>Description</b>	Reads the opposite INT_L QSFP_DD pin
<b>Call</b>	<b>bool __stdcall IntL(int Instance, bool* status)</b>

Parameters	int Instance: USB instance bool* status: True if HW pin is 1 False if HW pin is 0
Returns	True or False

### North America

48521 Warm Springs Blvd. Suite 310  
Fremont, CA 94539  
USA  
+1 510 573 6388

### Worldwide

Houmal Technology Park  
Askarieh Main Road  
Houmal, Lebanon  
+961 81 794 455

### Asia

14F-5/ Rm.5, 14F., No 295  
Sec.2, Guangfu Rd. East Dist.,  
Hsinchu City 300, Taiwan (R.O.C)  
+886 3 5744 591