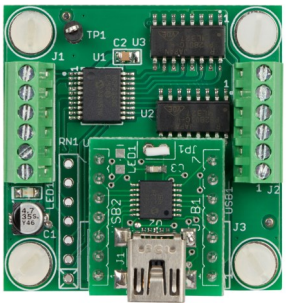
EMEDDED TEST SOLUTIONS

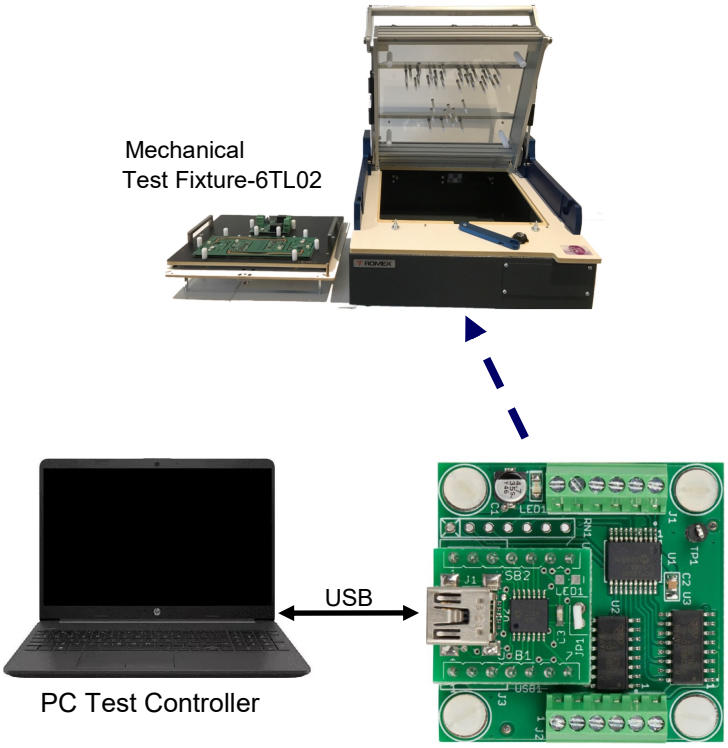
FID-MATE, FIXTURE ID MODULE



- *OI Unique SBS™- Single Board Solutions*
- *8 general purpose optocoupler I/O lines (4 inputs & 4 outputs)*
- *Use to support Test Fixture status lines (DUT_Present & Fixture_Ready inputs)*
- *Built-in test fixture cycle counter*
- *Support Pass/Fail status (front panel LED's)*
- *USB interface, 2.0*
- *Compact size (1.50" x 1.50")*

Quick Start Guide

The sequence of steps on the right provides a simple procedure for getting customers up and running as quickly as possible. The diagram below highlights a typical application for the FID-MATE. The goal of the FID-MATE is to provide access to a collection of test fixture controls and status indicators.



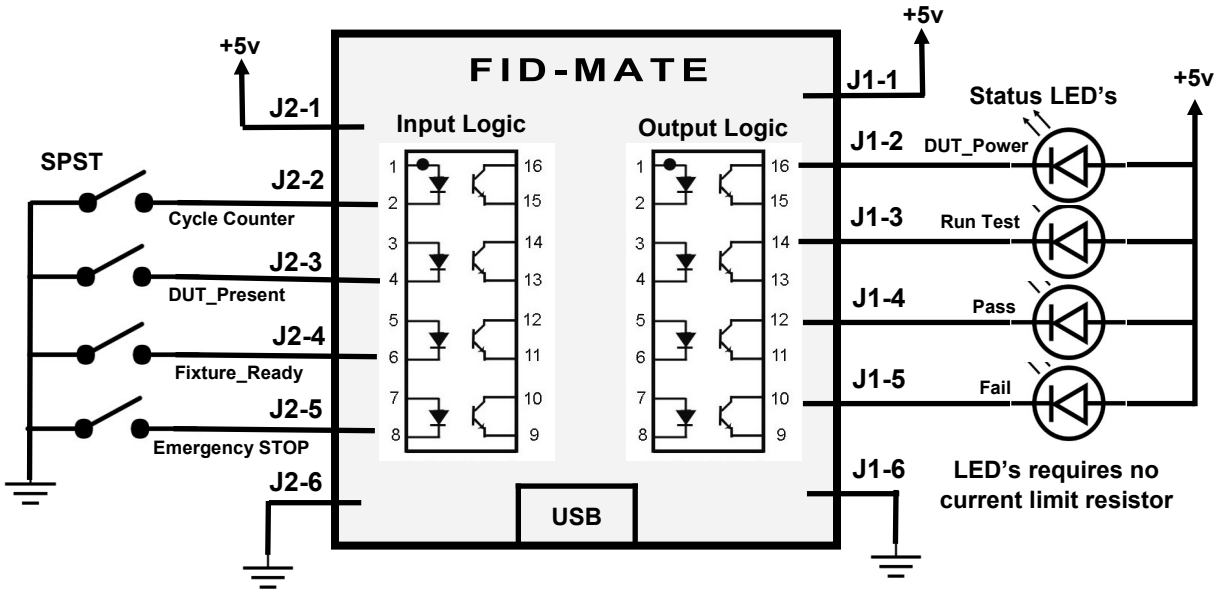
1	Go to the overtoninstruments website, then go to instrument downloads and select the driver for the relevant OS (i.e. windows, mac or linux.
2	Run setup file, CP210xVCPinstaller...
3	Connect the (Oi) instrument module. Connect a USB cable from the PC and to the (Oi) instrument. Verify the power LED is On, on the (Oi) instrument.
4	Check the COM Port status. Go to the Device Manager and select Ports (COM & LPT), to verify the following, 'Silicon Lab CP210x USB to UART Bridge (COM?)'
5	Use the (Oi) Control Panel - GUI. Go to the 'Test Instrument Modules' folder and follow the path to locate the folder that includes your (Oi) instrument. Open the 'Software' folder and locate the 'CVI GUI' folder and select the executable.
6	Use TeraTerm (or an equivalent terminal emulator program) to exercise the (Oi) instrument. Follow the instructions in the product User's Manual.

MARCOM20230926-01 rev01

DFT MG-Products, Rijkevoortsedijk 27A, 5447 BD, Rijkevoort, The Netherlands, info@designedfortest.com www.designedfortest.com

1.0 The Hook-up

The requirements to connect the FID-MATE is illustrated below.



DFT MG-Products, Rijkevoortsedijk 27A, 5447 BD, Rijkevoort, The Netherlands, info@designedfortest.com www.designedfortest.com

2.0 Serial Command Set

To facilitate remote control for the FID-MATE, a USB interface is required. When connected to a host PC, the USB connection appears as a "Virtual Com Port", which establishes a serial data communications link between the two. The default protocol is 19200 baud rate, no parity, 1 stop bit and no flow control. The FID-MATE will respond to a unique set of ASCII serial data commands (listed below). The first three bytes of the command string starts with the prefix **'FM_',** followed by a code that represents the actual command. All commands are upper case sensitive and are terminated with a carriage-return. If the command is valid, the FID-MATE will return either a **'<>'**, or a bracketed result (i.e. **'<11000110>'**. After the command is processed, then a **'→'** byte is returned (this response is a "prompt" to signal the FID-MATE is ready to receive the next command). If the FID-MATE detects an incorrect command then one of three error symbols will be generated, (1) invalid command and a **'><'** is returned, (2) a command that is out-of-limits then a **'>>'** is returned, and (3) a command that is out of sequence then a **'<<'** is returned. In some cases the error symbol will include a bracketed result (i.e. '**>1<'**), which defines a specific error code. Keep in mind, some commands take longer then others to process - so we recommend inserting a small delay between each character sent (perhaps a 1msec or less). This will prevent a buffer over-flow condition. **Please note, the instrument does not support 'backspace' for correcting the input sequence.**

Command	Function	Response	Description	Example
FM_BRn	Set baud rate code	<>	Select 1 of 4 different baud rates by changing -n-code. 0 = 1200, 1 = 2400, 2 = 9600 & 3 = 19200. Baud will remain set. Default code is 3 (19200).	'FM_BR2' Select 2400 baud rate
FM_BR?	Get baud rate code	<?>	Get current baud rate code (-n- is the return code 0 to 3).	
FM_ID?	Get module ID	<?>	Get module current identification, model and version number.	'FM_ID?' <FID-MATE(vl)rev1.0>
FM_MR	Maser Reset	<>	Reset & initialize the module	
FM_DOnb	Set output bit	<>	Set output bit (-n- = bit number 0 to 3) and (-b- = 1 or 0, enable or disable) b0 = DUT_Power (Green) b1 = Run_Test (Yellow) b2 = Pass_Test (Green) b3 = Fail_Test (Red)	'FM_DO21' Set bit number 2 logic 1
FM_DO?n	Get output bit status	<?>	Get output bit status (-n- = bit number 0 to 3) .	'FM_DO?2' Returns <1/0>
FM_DI?	Get input bit status	<?>	Get input 4-bit status (xxxx, bit number 3 to 0) b0 = Cycle_Counter b1 = DUT_Present b2 = Fixture_Ready b3 = EPO	'FM_DI?' Returns <0010>

DFT MG-Products, Rijkevoortsedijk 27A, 5447 BD, Rijkevoort, The Netherlands, info@designedfortest.com www.designedfortest.com

2.0 Serial Command Set, cont.

Command	Function	Response	Description	Example
FM_CC	Clear Cycle Counter	<>	Reset the cycle counter to 0	
FM_CMn	Set cycle counter mode	<>	Set cycle counter mode with -n-code (0 = disable & 1 = enable). Default code is 0, after module reset.	'FM_CM1' Enable counter mode
FM_CM?	Get cycle counter mode status	<?>	Get cycle counter mode status (-b- is the return value, 0 or 1 binary).	'DM_BM?' Returns <1>
FM_OL?	Get cycle counter over limit status	<?>	If cycle counter is greater than 65535 (-b- is the return logic state 0 or 1). A logic high indicates over limit.	'FM_OL?' Returns <1/0>
FM_SDn	Save ID data string	<>	A data string is saved to a address code (–n– number). Data should bracketed '|', at the beginning and end of the data string. 0 = Fixture Part Number (24 bytes) 1 = Fixture Serial Number (24 bytes) 2 = Fixture Asset Number (24 bytes) 3 = Cycle Counter Number (10 bytes) 4 = DUT Name (24 bytes) 5 = DUT Part Number (24 bytes) 6 = DUT Sub Part Number (10 bytes) 7 = DUT Serial Number	'FM_SD1|987654-1234|' Save Fixture Serial Number
FM_RD?n	Retrieve ID data string	<?>	Retrieve the data string from address code (-n– number).	FM_RD?1 <987654-1234> Get Fixture Serial Number
FM_CSn	Clear data string	< >	Clear the data string from address code (-n– number).	FM_CS1 Write a " ", to the Fixture Serial Number string
FM_CD	Clear ID Data	<>	Clear ID data, set to all bytes to " "	
FM_MAnnn	Set the Module Address	<>	The module address a 3 digit code (000 - 255), that allows an external computer to locate a physical module in case the USB Com Port jumps to a different port number. The module address is stored EEPROM, and can be retrieved to at any time.	'FM_MA010' Set to address 10
FM_MA?	Get the Module Address	<?>	Retrieve the module address.	'FM_RM?' Returns <010>

DFT MG-Products, Rijkevoortsedijk 27A, 5447 BD, Rijkevoort, The Netherlands, info@designedfortest.com www.designedfortest.com

2.0 Serial Command Set, cont.

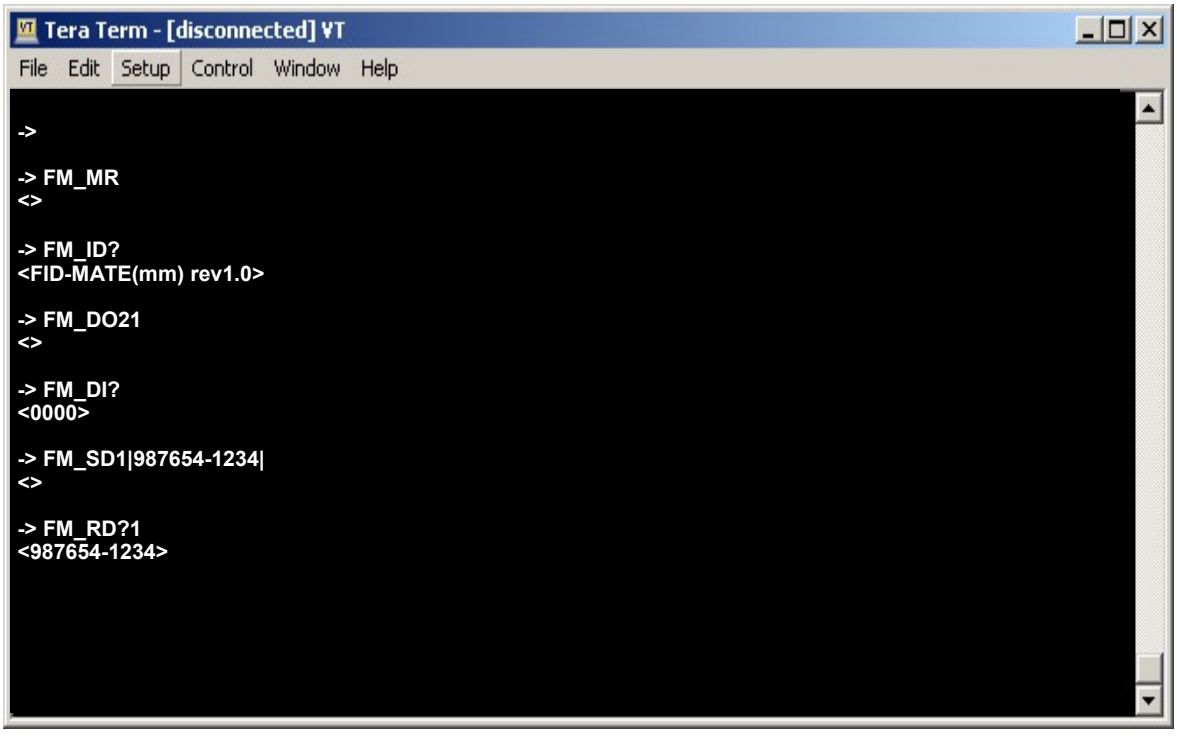
Command	Function	Response	Description	Example
FM_UD	Write User Data	<>	A 32-byte character string is stored in EEPROM, as User Data. The command includes a bracket to define the string length.	'FM_UD|Overton Instruments|
FM_UD?	Get User Data	<?>	Retrieve the User Data.	'FM_UD? Returns <Overton Instruments>
FM_UC	Clear User Data	<>	Clear the 32-byte User Data string stored in EEPROM.	
Error Codes		>0<	In certain instances, the command will generate a error response or code (see below). >0< : Input string buffer overflow	

DFT MG-Products, Rijkevoortsedijk 27A, 5447 BD, Rijkevoort, The Netherlands, info@designedfortest.com www.designedfortest.com

3.0 Programming

Programming the FID-MATE is very simple. When connected to a external PC, the FID-MATE responds to a simple set of ASCII commands. To communicate, just use any programming language that allows access through a serial port, including Visual BASIC, C/C++, LabView, Lab Windows/CVI or Python.

TeraTerm is a “free” open-source, terminal emulator program that is widely used and can be easily downloaded from the internet. TeraTerm emulates different types of computer terminals, from DEC VT100 to DEC VT382. It supports telnet, SSH 1 & 2 and serial port connections. Use the USB cable to connect the PC to the FID-MATE. Run TeraTerm and configure the settings for 19200 bps, 8 data bits, no parity, 1 stop bit and no flow control. TeraTerm will show which COM port the FID-MATE is connected. Press the ‘Enter’ key and the ‘à’ prompt should appear on the screen. The sequence below describes a typical session. First, the module is reset with the ‘FM_MR’ command. Next, the module is queried for its ID number, by sending the ’FM_ID?’ command. Now, we want to set bit2 to high ‘FM_DO21’. Next, we can queried the input bits ‘FM_DI?’. Now, we can store the Fixture Part Number ‘FM-SD1|987654-1234|, And lastly, we check it by sending ‘FM-RD?1’.



DESIGNED FOR TEST
PART OF MG-PRODUCTS

MG-Products B.V.
Rijkevoortsedijk 27A
5447 BD, Rijkevoort (NL)
Phone: +31 (0)485 382 133
www: www.designedfortest.com
E-mail: Info@designedfortest.com