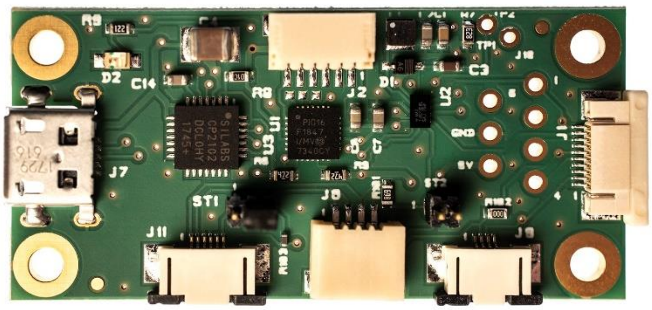
User Guide

USB-M Flexiboard User Guide



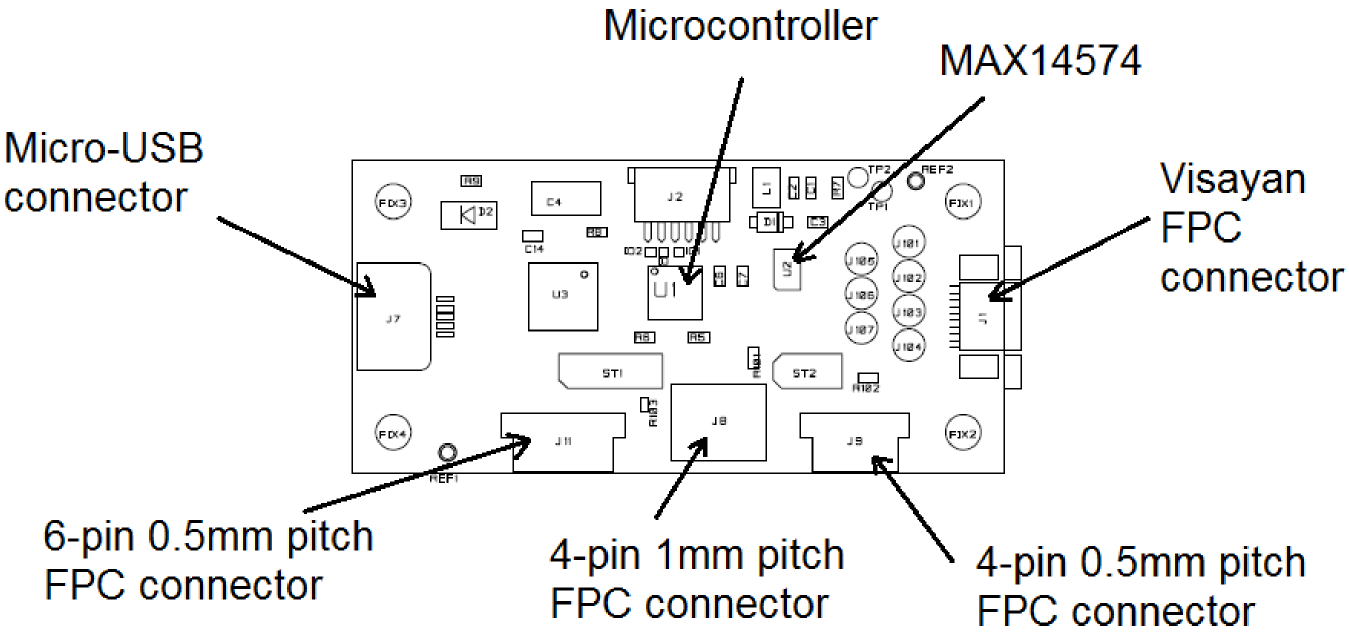
Introduction

The USB-M Flexiboard is a compact development board featuring a USB connector, a microcontroller, a MAX14574 driver and FPC connectors that allow controlling from a computer A-Series, C-S-Series and Visayan lenses. This document explains how to use the USB-M Flexiboard.

Contents

1. Hardware description

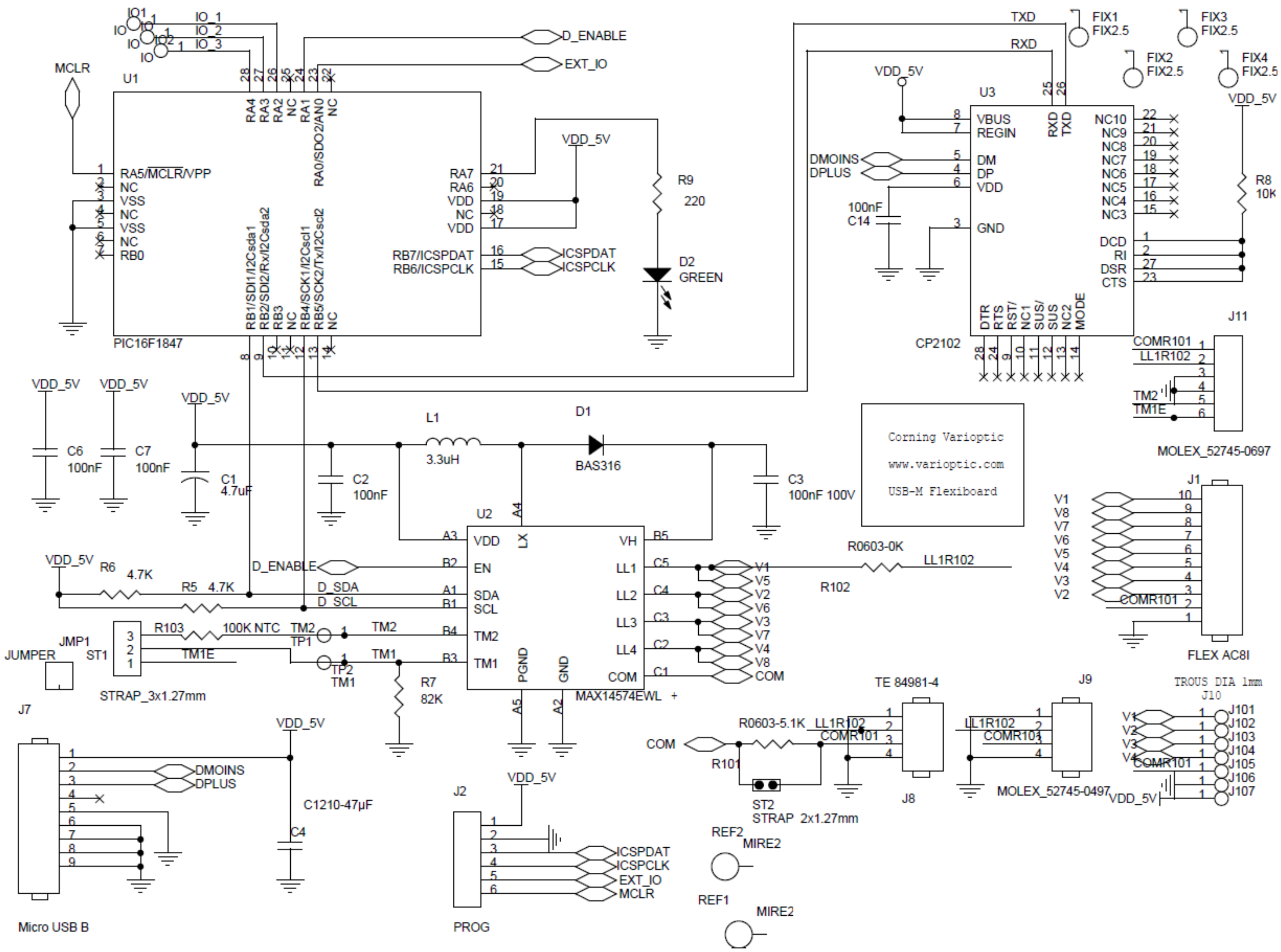
1.1. General description of the board



Board dimensions: 48 x 23 x 8 mm

Weight: 5.4 gr

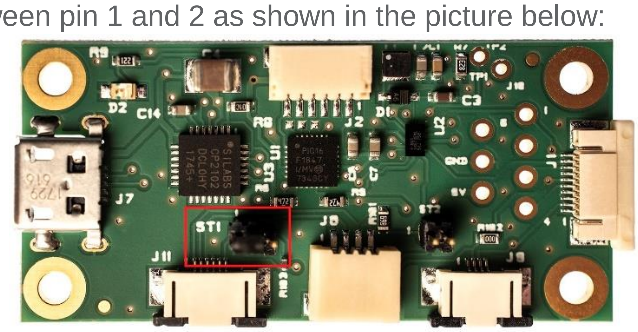
1.2. Board schematic



1.3. Jumper position description

Two jumpers are present on the board:

- Jumper located in ST1 allows customer to use either the thermistor populated 6 pin FPC (i.e FPC-A-10, FPC-A-14 or FPC-A-16, etc…) or the thermistor located on the board. To use thermistor on the flex, please set the jumper between pin 1 and 2 as shown in the picture below:



To use thermistor located on the board, please set the jumper between pin 2 and 3.

- Jumper located in ST2 allows customer to use or bypass serial 5kΩ resistor on the COM output of the Maxim driver. When the jumper is plugged in, the resistor is bypassed. As a reminder this resistor is recommended when driving A-39N0 lens.

2. Bill of material

Item	Quantity	Reference	Part number	PCB Footprint
1	1	C1	4.7uF	603
2	1	C2, C6, C7, C14	100nF	603
3	1	C3	100nF 100V	603
4	1	C4	C1210-47uF	1210
5	1	D1	BAS316	SOD323
6	1	D2	GREEN LED	805
7	4	FIX1, FIX2, FIX3, FIX4	FIX2.5mm	
8	3	IO1, IO2, IO	IO	
9	2	JMP1	JUMPER HARWIN M50-1900005	
10	1	J1	TE CONNECTIVITY 1-1734592-0	
11	1	J2	JST SM04B-SRSS-TB(LF)(SN)	
12	1	J7	Micro USB	
13	1	J8	TE 84981-4	
14	1	J9	MOLEX 52745-0497	
15	1	J11	MOLEX 52745-0697	
16	7	J101, J102, J103, J104, J105, J106, J107	Holes 1mm	
17	1	L1	3.3uH	2x19
18	2	REF1,REF2	MIRE2	
19	2	R5,R6	4.7K	603
20	1	R7	82K	603
21	1	R8	10K	603
22	1	R9	220	603
23	1	R101	R0603-5.1K	603
24	1	R102	R0603-0K	603
25	1	R103	100K NTC	
26	1	ST1	STRAP_3x1.27mm	
27	1	ST2	STRAP_2x1.27mm	
28	1	TP1	TM2	2x19
29	1	TP2	TM1	2x19
30	1	U1	PIC16F1847	UQFN
31	1	U2	MAX14574EWL	5X3WLP
32	1	U3	CP2102	28QFN

3. Use of the USB-M Flexiboard to drive the lens

3.1. In a windows environment

A very simple way to use the USB-M Flexiboard is to install the FocusLab software. When installing FocusLab software ComCasp.dll is copied on your computer. This library sends commands to the board via a virtual com port. This note is not taking care of USB<->UART conversion which is done via Silicon Labs driver Virtual COM port:

http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpdrivers.aspx

As of February 2019, when using Windows 10, the driver is not automatically installed by the operating system when plugging USB-M Flexiboard. Please refer to MAAN - C-COM Board, USB-M Drivboard and Flexiboard installation tip with Windows 10 to install the driver

Using Windows 8 or 7, the USB-UART conversion driver is installed automatically when plugging the USB-M Flexiboard.

3.2. In non-windows environment

In non-windows environment, for example Linux, either the user can use a virtual COM port driver (if any available); either it can build a complete link, thanks to below register map and protocol definition.

4. Register map

Register	ADDR	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Focus_LSB	0x00	R/W	Foc_07	Foc_06	Foc_05	Foc_04	Foc_03	Foc_02	Foc_01	Foc_00
Focus_MSB	0x01	R/W	Foc_15	Foc_14	Foc_13	Foc_12	Foc_11	Foc_10	Foc_09	Foc_08
Control	0x02	R/W	x	x	x	x	x	x	x	EEPROM
Mode	0x03	R/W	x	x	x	x	x	x	Analog	Stand_By
Reserved	0x04	R	x	x	x	x	x	x	x	x
SW version	0x05	R	VER_B7	VER_B6	VER_B5	VER_B4	VER_B3	VER_B2	VER_B1	VER_B0
Reserved	0x06	R	x	x	x	x	x	x	x	x
Reserved	0x07	R	x	x	x	x	x	x	x	x
Reserved	0x08	R	x	x	x	x	x	x	x	x
Reserved	0x09	R	x	x	x	x	x	x	x	x
Fault_Reg	0x0A	R	x	x	x	x	x	D_OverT	D_NoRes	D_OverL

5. Register definition

Field name	Bit	Description
Focus_LSB		*address: 0x00*
Foc [7:0]	[7:0]	Low byte of focus value
Focus_MSB		*address: 0x01*
Foc [15:8]	[7:0]	High byte of focus value. The focus value is a 16 bit integer corresponding to the following V_{rms} value : Code 0x0000 = 24Vrms … Code 0xB3B0 = 70Vrms V_{rms} = N x 0.001 + 24 (in Volts) where N = code from 0x 0000 to 0x B3B0 The driver output is updated only after each writing of the Focus_MSB register.
Control		*address: 0x02*
EEPROM	B0	Setting this bit saves the content of the registers from address 0 to 3 into the EEPROM. After the next power up of the module, the saved register will be loaded. This bit is automatically cleared after EEPROM writing.
Mode		*address: 0x03*
Standby	B0	Setting this bit puts the module in low power mode. The driver will stop generating high voltage and the current consumption will decrease (around 5mA).
Analog	B1	Analog mode bit: if set, the driver will be controlled by the analog input. If cleared, the driver will be controlled by the Focus_LSB and Focus_MSB registers. By default this bit is set and the module is in Analog mode. Writing to the Focus_MSB register automatically clears this bit.
SW Version		*address: 0x05*
VER [7..0]	[7..0]	Software version: this register indicates the version of the USB-M firmware.
Fault_Reg		*address: 0x0A*
D_OverL	B0	This bit, if set, indicates that the driver is overloaded (Vh could not reach 70V).
D_NoRes	B1	This bit, if set, indicates that the driver is no more responding to I2C request
D_OverT	B2	This bit, if set, indicates that the driver is in thermal shutdown

6. UART protocol (RS232)

6.1. Hardware settings

- 57,600 bauds
- No parity
- Data: 8 bits
- Stop: 1 bit

6.2. Writing frame

STX	0x37	Add	Nb_data	Data_1	Data_2	…	Data_n	CRC

- STX = 0x02
- 0x37 = Write command
- Add = Address of first register to be written (if more than one register; the address will be automatically incremented)
- Nb_data = number of registers to be written (12 max)
- Data_1 to n = register value
- CRC = 1 byte sum of all bytes (STX, CDE, ADD, DATA)

<u>Example</u>:

0x02	0x37	0x03	0x01	0xFF	0x3C

- Response of the board if transmission is successful:

STX	0x37	ACK	CRC

With ACK = 0x06

- Response of the board if transmission is not successful:

STX	0x37	NACK	0x06

With NACK = 0x15

In this case the application should send the frame again

6.3. Reading frame

STX	**0x38**	**Add**	**Nb_data**	**CRC**

- STX = 0x02
- 0x38 = Read command
- Add = Address of first register to be read (if more than one register; the address will be automatically incremented)
- Nb_data = number of registers to be read
- CRC = 1 byte sum of all bytes (STX, CDE, ADD, DATA)

Example:

0x02	0x38	0x03	0x01	0x3E

- Response of the board if transmission is successful:

STX	0x38	Data_1	Data_2	…	Data_n	0x3E

- Response of the board if transmission is not successful:

STX	0x38	NACK	CRC