

OCCIE MEMS CONTROLLER USER GUIDE



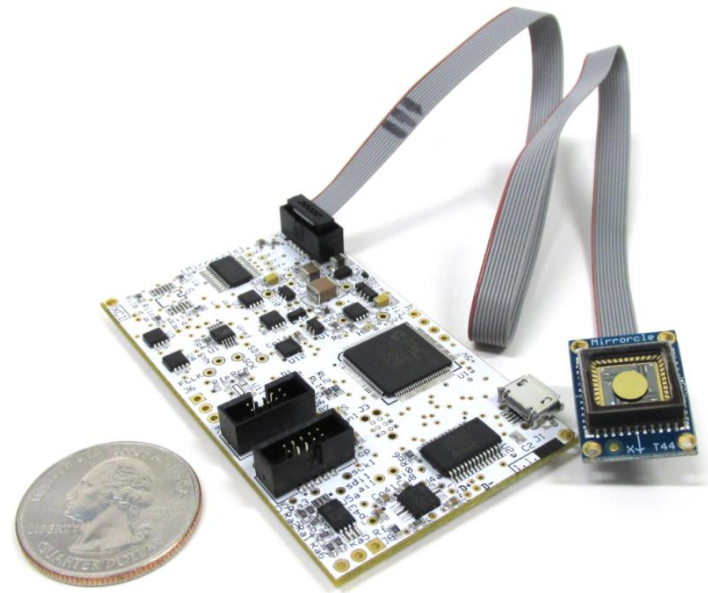
Last Revised:
Feb 1st, 2018

Dr. Veljko Milanovic

Abhishek Kasturi

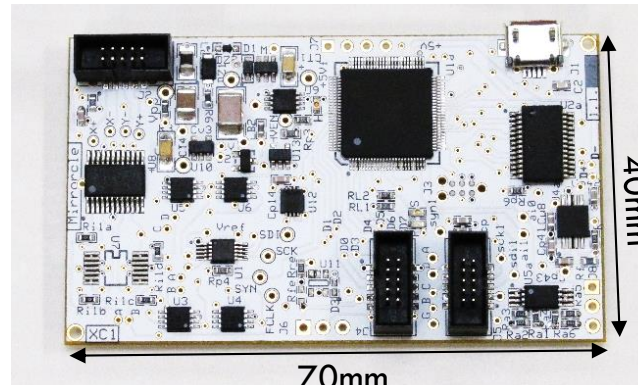
Table of Contents

- Overview and Features
- Control & Connections of the Controller
- Various Ways of Control
- Terminal Commands
- Connector Pinouts
- Embedded MEMS Driver

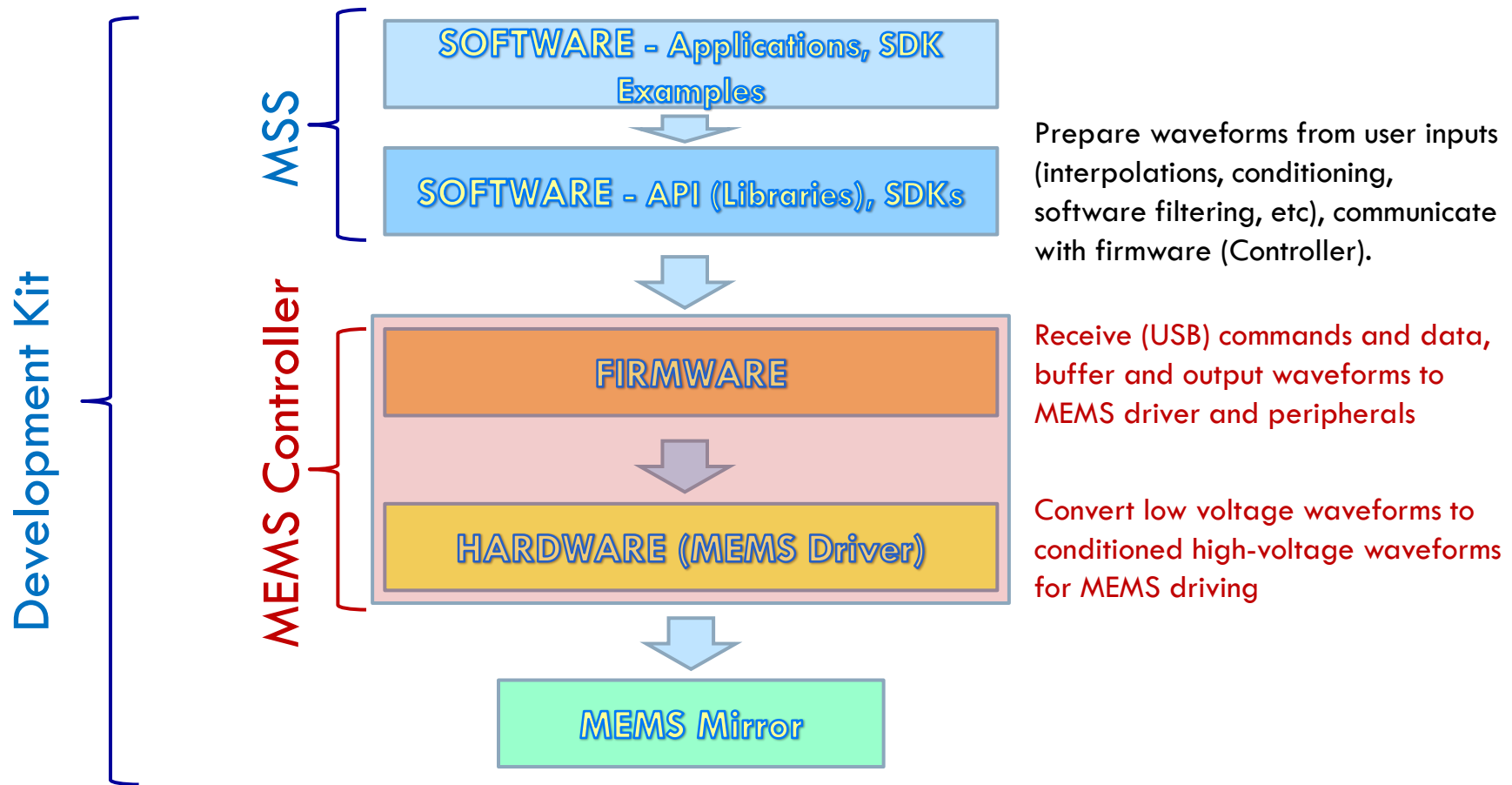


MEMS Controller OCCIE – Overview

- OCCIE 1.x is a Mirrorcle-designed controller with a fast Microchip PIC32MZ MCU. It provides USB communication, integrated MEMS driver for driving MEMS devices, synchronization with other systems, and programmable hardware filtering to avoid overshoot and ringing. It is a compact, OEM version of the USB-SL MZ Controller for volume production applications.
- Controller is USB bus-powered and provides power for the MCU, MEMS driver, laser driver and some low-power peripherals. System can therefore run solely on a laptop powered by its battery. Interfacing and controlling the board is provided by C++ API for Windows applications, by Matlab API, or by LabView API, as shown in many SDK examples and executable programs included with the Mirrorcle Software Suite.
- **Targeted Applications:** Entertainment, MEMS Vector Display, etc.



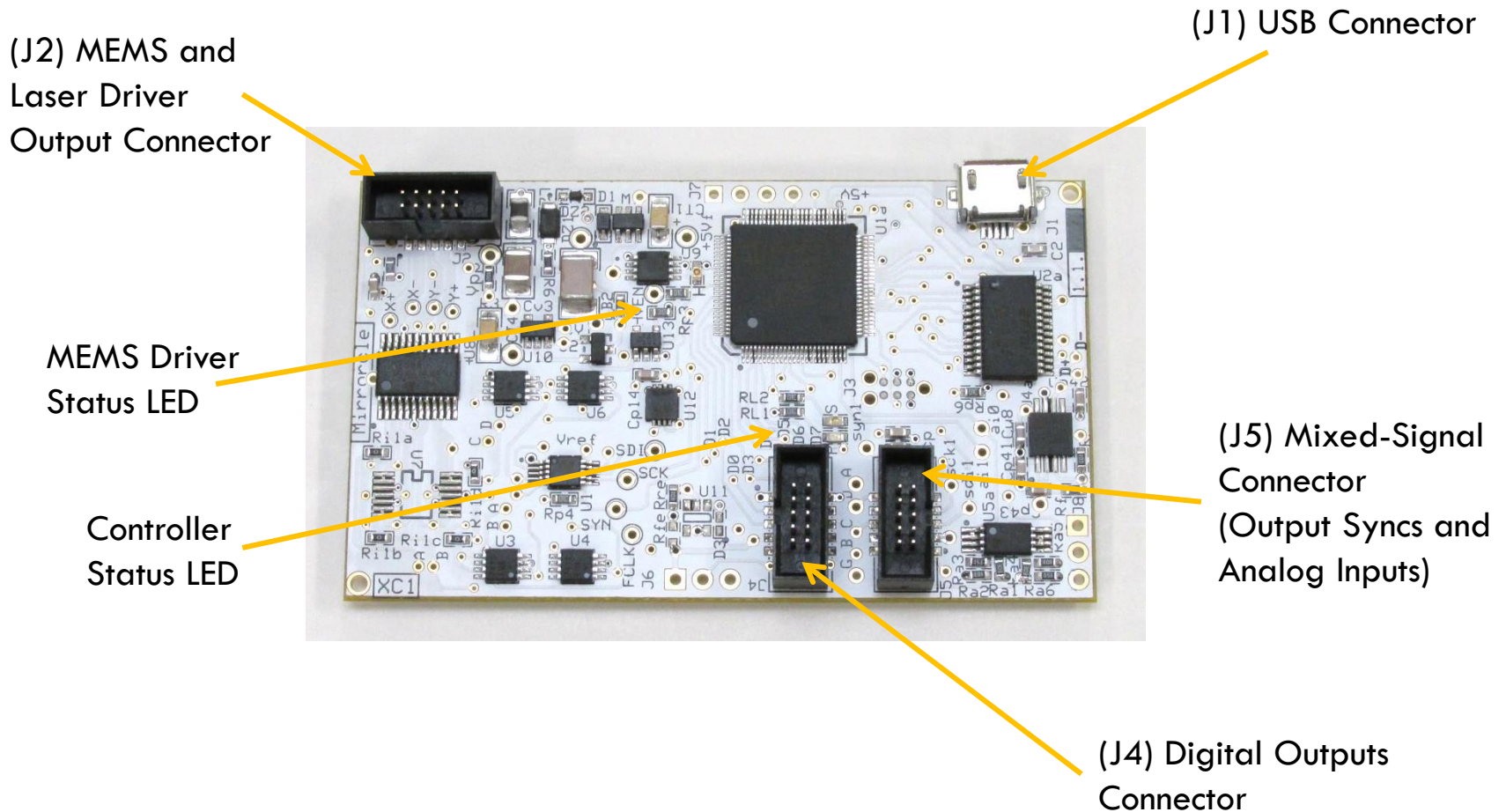
Layers of Abstraction in MEMS Mirror Control



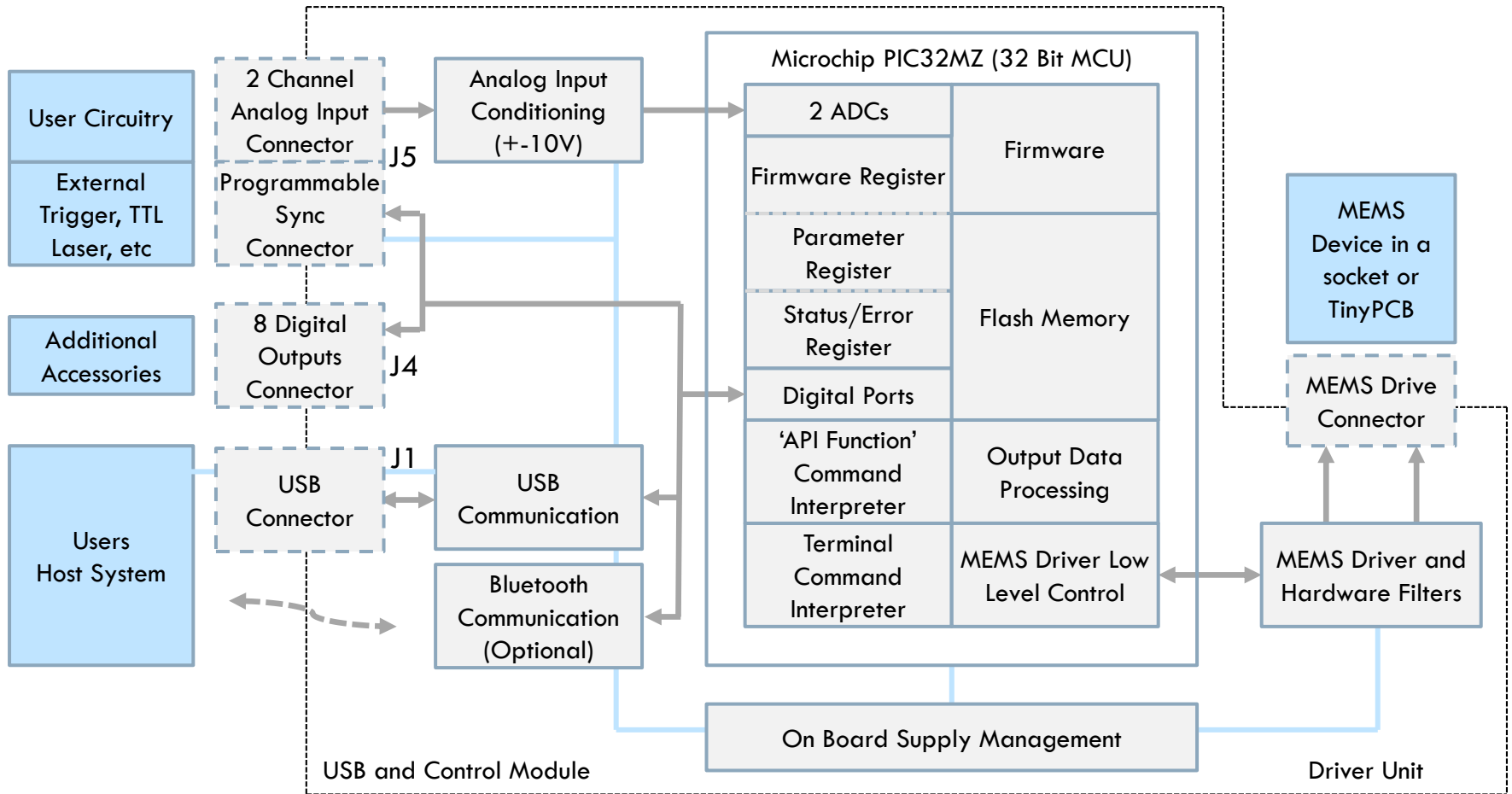
Features

- 4x (16-bit) Analog Outputs for X and Y axis control of MEMS mirrors
- Programmable hardware-based low-pass filters and protection circuitry to reduce the chances of device damage
 - Output bandwidth from 50Hz to 25kHz (governed by programmable filters)
- 8x Correlated Digital Outputs (3.3V) for controlling other components or systems
- Sample Rates up to 120,000 samples per second (120 kSPS)
- 500kB of Onboard RAM allows up to 100,000 samples to be stored
- 2x (12-bit) Analog Inputs with +/- 10V input range
- Flash Memory allows storage of settings and data for stand-alone operation (no PC)
- Sync Port for synchronization with additional Controllers or driving of laser peripherals
- USB Powered, with 750mW Power Consumption
- Small Footprint of 40mm x 70mm x 9.5mm
- Integrated laser driver for EaZy 1.1 folded cells on MEMS Output Connector
- IN DEVELOPMENT: Wireless option allows battery-run wireless operation over Bluetooth 16 bit outputs for X and Y axis control

Status LEDs & Connections

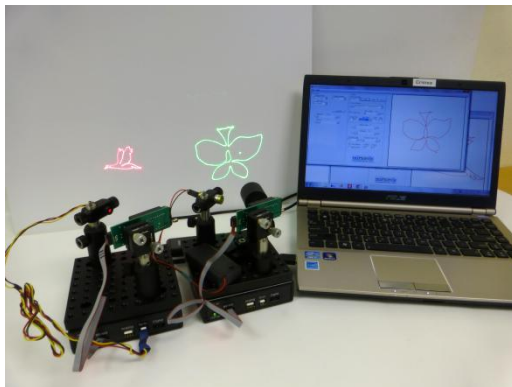


Block Diagram



Multi-Platform Compatibility

- Several ways of communicating with the OCCIE Controller are available, whether by USB interface or by Bluetooth interface with a PC or an Android or Linux device.



Various Ways to Control

APIs in several languages



- Comprehensive APIs for generation of content (MEMS positions and correlated digital outputs), for control and streaming of the content, reading of analog inputs, synchronization with additional Controllers or peripherals, tracking, etc.
 - C++ SDK (Windows and Linux)
 - LabVIEW and Matlab SDKs
 - Java (Android) SDK
- Easy-to-use GUI and Console Applications such as MirrorcleDraw, MirrorcleLinearRaster, MirrorcleTrack, for various platforms (previous slide)

Serial Terminal Commands

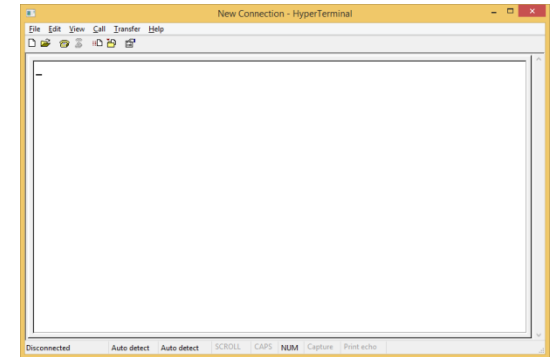
- Serial port terminal commands are available for more basic MEMS mirror control without the use of the Windows, Android, Linux APIs.

USB Supply, Status LEDs & Wireless Option

- Please Note that the power supply (+5VDC) and fast communication is handled via the USB interface. To ensure proper operation, use the provided USB cable directly plugged into the PC instead of a USB hub (too long or not conforming cables could cause malfunction).
- Ensure proper USB or battery power supply. The USB voltage level has to be in spec (at least 4.85 V). If a USB port or a battery does not provide adequate supply, the Mirrorcle MEMS Controller card may respond to software commands but may not fully enable the MEMS Driver.
- The green status LED will light up after approx. 4 seconds when the device is ready. It will flash during communication. The red LED indicates the switched on MEMS Controller.
- With the optional Wireless Add-On, Bluetooth is used to communicate with the Mirrorcle MEMS Controller with almost all Android devices by using the powerful Android SDK. When used in wireless mode, power should be supplied to the USB input port from either a computer's USB port or a 5VDC battery.
- For communicating with SoCs e.g. Arduino or Raspberry Pi, its possible to change the USRAT/Bluetooth interface baudrate of the processor between 115200 and 460800 Baud.

How to use Terminal Mode

- Please consider this mode is dedicated for experienced users only.
- Install Terminal Application (e.g. HyperTerminal, PuTTY)
- Connect the terminal application to the proper COM port.
- Enable/unlock the UART/terminal mode by entering '\$MTI\$'.
- Switch on displaying your commands (ECHO) by 'MTI+EO\n\r'.
- Ensure proper hardware settings from their representative datasheets with 'MTI+SetVbias 70\n\r', 'MTI+SetVdifferenceMax 100\n\r', and 'MTI+SetHardwareFilterBw 300\n\r'.
- Enable the MEMS Mirror driver with 'MTI+EnableDevice\n\r'.
- Manipulate the output by steering the mirror: 'MTI+GotoDevicePosition 0.45 -1.00 255\n\r'.
- Before quitting the terminal application, disable the MEMS driver by 'MTI+DisableDevice\n\r' and exit/lock the terminal/UART mode by the 'MTI+Exit\n\r' command.
- Please note that there are also short commands available.
- Type MTI+? For a complete list of available terminal commands.



Terminal Communication Table

Prefix	Command	Suffix (Return)	Short Command	Description	Reply	Reply Suffix	Example Terminal Input
	\$MTI\$	\n or \r		Enter Terminal Command Mode	MTI-Device [name] Ready in command mode	\n	\$MTI\$\n
MTI+	Exit	\n or \r	EX	Exit Terminal Command Mode	MTI-Device Exit command mode	\n	MTI+EX\n
MTI+	EnableDevice	\n or \r	EN	Enable MEMS Driver	MTI-OK	\n	MTI+EN\n
MTI+	DisableDevice	\n or \r	DI	Disable MEMS Driver	MTI-OK	\n	MTI+DI\n
MTI+	EnableDigitalOutput	\n or \r	ED	Enable Digital Output	MTI-OK	\n	MTI+ED\n
MTI+	DisableDigitalOutput	\n or \r	DD	Disable Digital Output	MTI-OK	\n	MTI+DD\n
MTI+	EnableModulation	\n or \r	EM	Enable Modulation for PD measurement	MTI-OK	\n	MTI+EM\n
MTI+	DisableModulation	\n or \r	DM	Enable Modulation for PD measurement	MTI-OK	\n	MTI+DM\n
MTI+	GoToDevicePosition	\n or \r	GT	Mirror XY Position and Digital output M, Send 3 values: float float byte	MTI-OK	\n	MTI+GT 0.5 -.22 13\n
MTI+	GetAnalogInputValue	\n or \r	GA	Replies with AIO and AI1 values as floats, samples of both analog input channels	MTI-AIO:float AI1:float	\n	MTI+GA\n
MTI+	GetPhotoDetectorValue	\n or \r	PD	Replies with AIO value - AIO value halfway through sample (laser is on half of sample and off half of sample) - Must be in SyncMode 2	MTI-AIO:float	\n	MTI+PD\n
MTI+	EchoOn	\n or \r	EO	Echo mode on (displays keyboard input)	MTI-OK	\n	MTI+EO\n
MTI+	EchoOff	\n or \r	EF	Echo mode off	MTI-OK	\n	MTI+EF\n
MTI+	DisableConfirm	\n or \r	DC	Disable confirm reply after commands (only MTI-OK)	MTI-OK	\n	MTI+DC\n
MTI+	EnableConfirm	\n or \r	EC	Enable confirm reply after commands	MTI-OK	\n	MTI+EC\n
MTI+	SetVdifferenceMax	\n or \r	VD	Sets Maximum Voltage Difference, number between 0-200	MTI-OK	\n	MTI+VD 80\n
MTI+	SetHardwareFilterBw	\n or \r	BW	Set Hardware Filter, number between 50-15000	MTI-OK	\n	MTI+BW 200\n
MTI+	SetVbias	\n or \r	VB	Set Bias Voltage, number between 0-100	MTI-OK	\n	MTI+VB 70\n
MTI+	SetSyncMode	\n or \r	SM	Set SyncMode, number between 0-6	MTI-OK	\n	MTI+SM 2\n
MTI+	GetDeviceParams	\n or \r	GP	Get device parameters	MTI-[list of device parameters]	\n	MTI+GP\n
MTI+	Help	\n or \r	?	Lists all terminal commands	MTI-[list of all commands]	\n	MTI+?\n

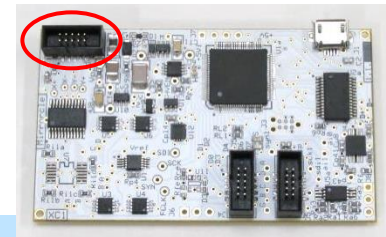
Description

Previous command was invalid
Previous send data was invalid

Error Replies

MTI-ERR InvalidCommand
MTI-ERR InvalidCommandData

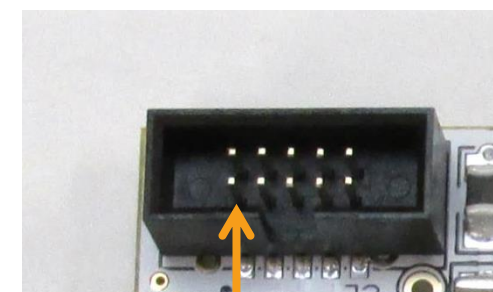
*Please note: MTI-OK reply can be disabled by DisableConfirm command



J1: MEMS Output Connector Pinout

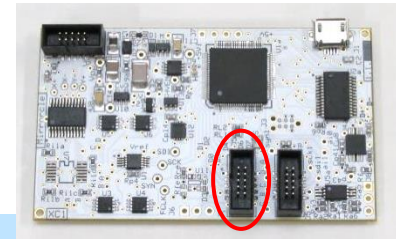
- 4x High Voltage Analog Output Channels
 - Voltage Range: 0V to 200V
 - DAC Resolution: 16-Bit
 - Sample Rate: up to 120ksps+ (API sample rate setting)

MEMS Connector: 10-Pin Header		
J5-Pin	Name	Description
1	HV_A (X+)	MEMS Channel X+
2	GND	Ground
3	HV_B (X-)	MEMS Channel X-
4	GND	Ground
5	HV_C (Y-)	MEMS Channel Y-
6	GND	Ground
7	HV_D (Y+)	MEMS Channel Y+
8	GND	Ground
9	LD+	Laser Diode (Anode)
10	LD-	Laser Diode (Cathode)



Pin 1

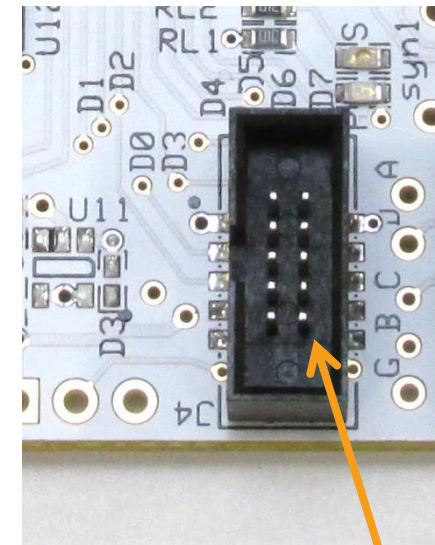
Header Digikey Part #:
1175-1629-ND



J4: Digital Output Connector Pinout

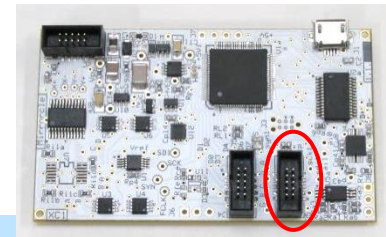
- The Digital Output connector has 8 digital outputs that are synchronous with the MEMS Driver sample output. (It enables to trigger accessories like cameras, lasers, etc.)
- Pin 1 can supply +3.3V, with a maximum current output of 25mA.
- Pins DOut0 – DOut8 can source 3mA of current per channel.

DOut Connector: 10 - Pin Header		
J8-Pin	Name	Description
1	+3.3V	+3.3V Supply, limited to 25mA
2	GND	Ground
3	DOut0	Digital Output Pin DOut0
4	DOut1	Digital Output Pin DOut1
5	DOut2	Digital Output Pin DOut2
6	DOut3	Digital Output Pin DOut3
7	DOut4	Digital Output Pin DOut4
8	DOut5	Digital Output Pin DOut5
9	DOut6	Digital Output Pin DOut6
10	DOut7	Digital Output Pin DOut7



Pin 1

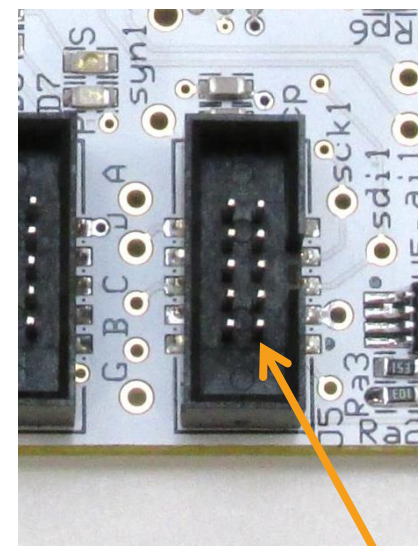
Header Digikey Part #:
1175-1629-ND



J5: Host Connector Pinout

- The Host connector has 3 digitally configurable synchronization pins as well as 2 analog inputs, and a 5V supply input for standalone applications not using the USB port.
- Note that the Sync pins are still under development and not available for the user

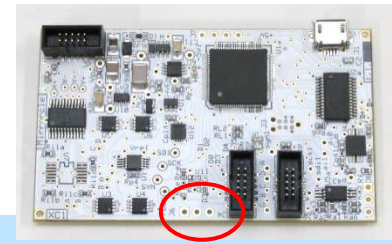
DOut Connector: 10 - Pin Header		
J8-Pin	Name	Description
1	AIN0	Analog Input Channel 0
2	AIN1	Analog Input Channel 1
3	+5VDC	+5V Supply
4	GND	Ground
5	SYNC0	Digital Synchronization Pin 0
6	NC	No Connection
7	SYNC1	Digital Synchronization Pin 1
8	SYNC2	Digital Synchronization Pin 2
9	NC	No Connection
10	NC	No Connection



Pin 1

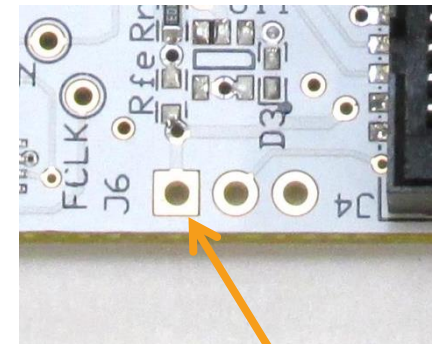
Header Digikey Part #:
1175-1629-ND

J6: Synchronization Connector Pinout



- The Synchronization port controls accessories e.g. Laser, Camera, etc.
- The port's middle Sync-pin can be set via software as an output to send start trigger, sample clocks.
- This port's middle Sync-pin can be set via software as an input to receive external sample clock or start triggers.
- The default setting for this pin is a direct copy of DOut0.

Sync Connector: 3 - Pin Header		
J6-Pin	Name	Description
1	+3.3V	+3.3VDC, 60mA
2	SNYC	Pin Function set with MTISync Parameter
3	Gnd	Ground



Pin 1

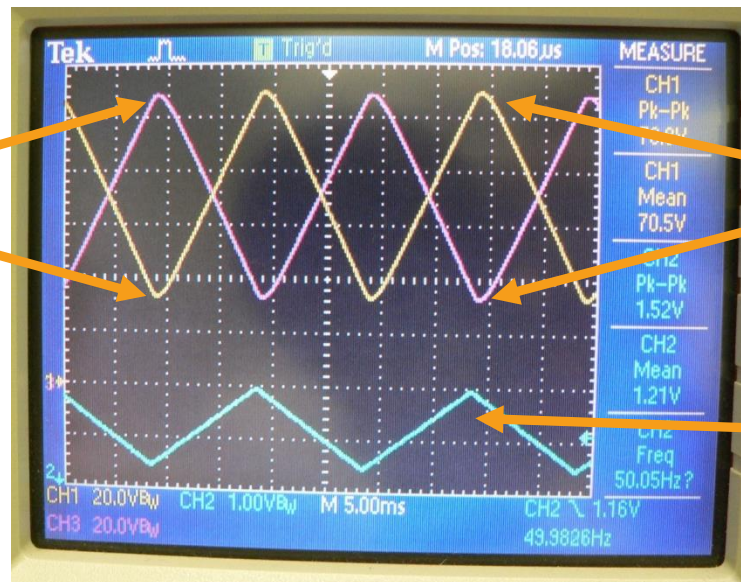
- **The SYNC pin is limited to sourcing / sinking 3mA of current – depending on it's operating mode of output or input.**

Header Digikey Part #: A19451-ND
Header not populated by default

Embedded MEMS Driver

- Embedded, sophisticated MEMS driver generates biased differential quad (BDQ) channel high voltage control signals
- Driver Bandwidth is governed by hardware filters set in software – settings from 50Hz to 50kHz are accepted.
- Output Voltage Range (each channel): 0V - 160V
- BDQ driving provides linearization and smooth driving of MEMS mirrors over all four quadrants (bi-directional on both axes).

HV_B is high
HV_A is low
Mirror rotates in
X- direction



HV_A is high
HV_B is low
Mirror rotates in
X+ direction

Resulting Mirror
X-axis Tilt

Laser Driver

- Pins 9 and 10 of J2 provide current-based laser driving which responds to commands from the waveforms OCCIE is streaming
- Default configuration is for red laser diodes with $\sim 60\text{mA}$ of current when the DOut0 pin is HIGH (the modulation “M” sample is >0) and $\sim 0\text{mA}$ when DOut0 is LOW
- Pin 9 can be connected directly (via short connection of few cm) to an ANODE of a laser diode such as the HL63603TG
- Pin 10 can connected directly (via short connection of few cm) to a CATHODE of the same laser diode
- With a modification on the board, the driver can drive a 520nm laser or a 450nm laser

Thank You for Choosing

If you have any further questions or suggestions please email us:

support@mirrorcletech.com

