190V Driver Evaluation Board

MEMs Evaluation Driver Board for the Raspberry Pi

Features

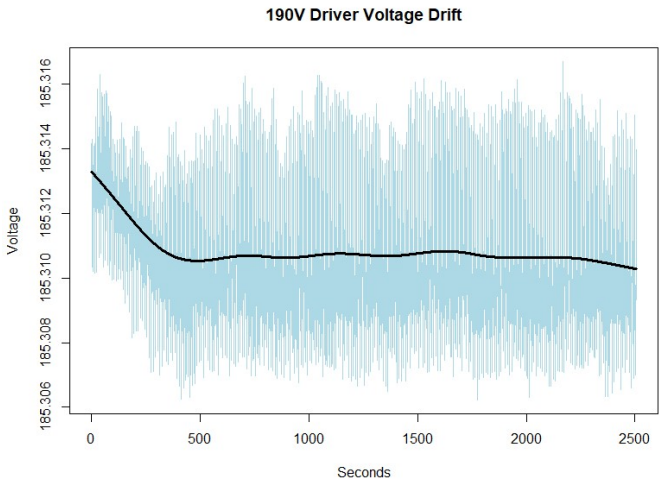
- 4 Channels up to 185V each
- Single 12V power supply needed
- Stable output within 65mV
- Capable of high speed outputs when SPI interface driven by external microcontroller
- Plugs directly onto the Raspberry Pi 2 or 3
- Arbitrary waveform generation
- Updates as fast as 1.6MHz using SPI
- Slew Rate as fast as 12V/uS
- Software source code included

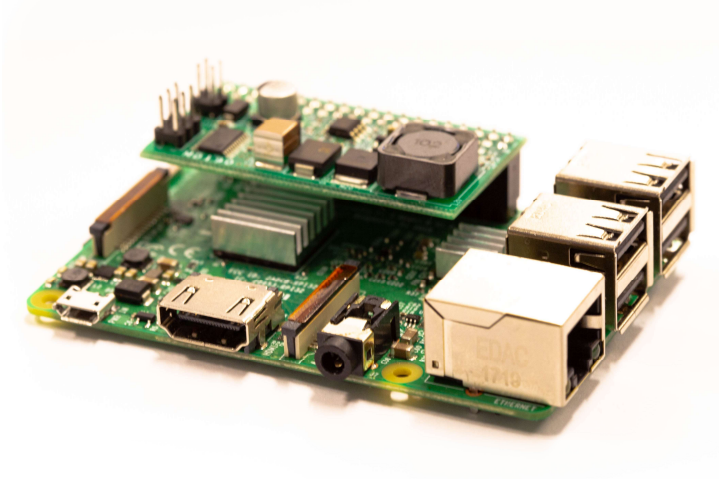
Overview

The 190V MEMs evaluation board is an all in one solution providing four channels of arbitrary waveform generation up to 185V to actuate MEMs mirrors. Typical output from the driver is stable with less than 40mV of noise and less than 85mV of drift.

The board is designed to plug directly onto the Raspberry Pi with zero additional circuitry required. Software and source code for controlling the driver is provided in C++ for higher performance applications. For even higher performance needs, an SPI interface and protocol are provided allows the end user to produce waveforms with much greater resolution.

Typical Noise and Drift

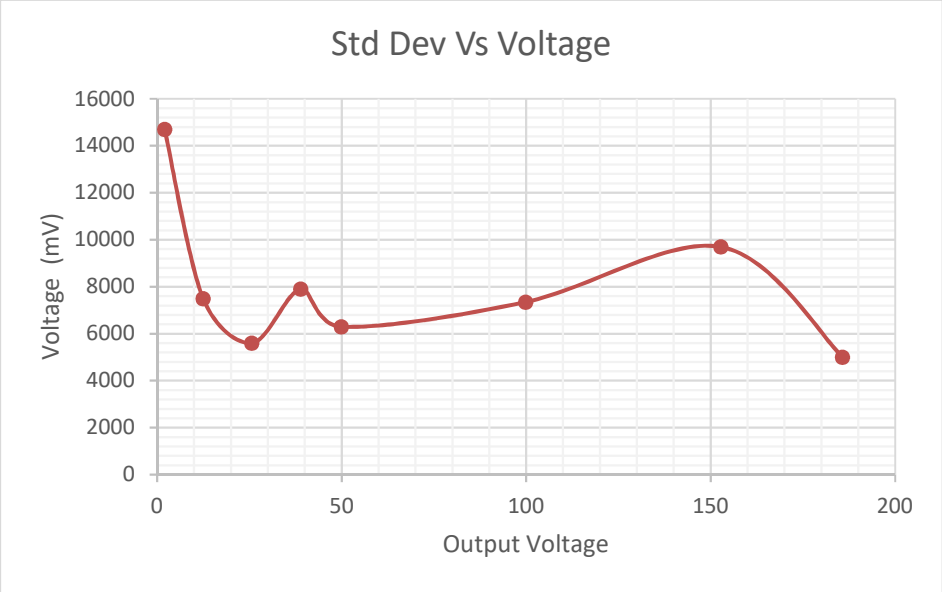




Preciseley Microtechnology Corporation
Innovation Center, 11421 Sask. Dr., Edmonton, Alberta, Canada T6G2M9
Tel: (780) 641 1925 Email: info@preciseley.com

1.0 Performance Characteristics

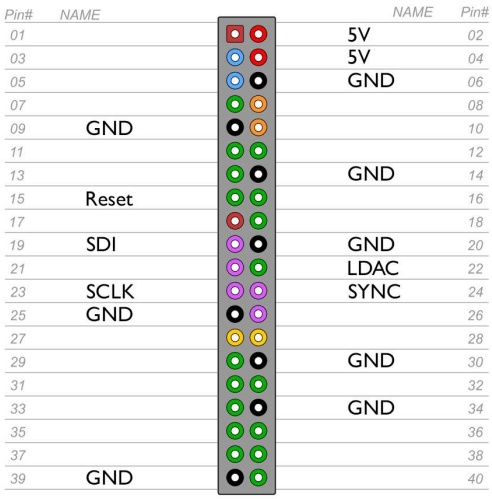
	190V
Max Operating Voltage	185
Slew Rate (V/us)	12/uS
Typical Drift Vpp (mV)	< 100
Typical Noise	40mV
Typical Maximum Sine Wave	
Max Freq	15kHz
Max Amplitude Vpp	185V



2.0 Wiring and Communication

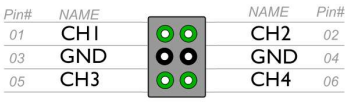
The board provides four output channels and two grounds and an interface to the Raspberry Pi. The pinout of each connector is shown below. Take special note of the pin numbers. Pins 1, 2, 3, and 4 are silk screened on the evaluation board.

Development Board Connector

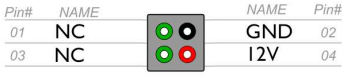


Take special care with the voltage input connector not to reverse polarity as this will damage the board.

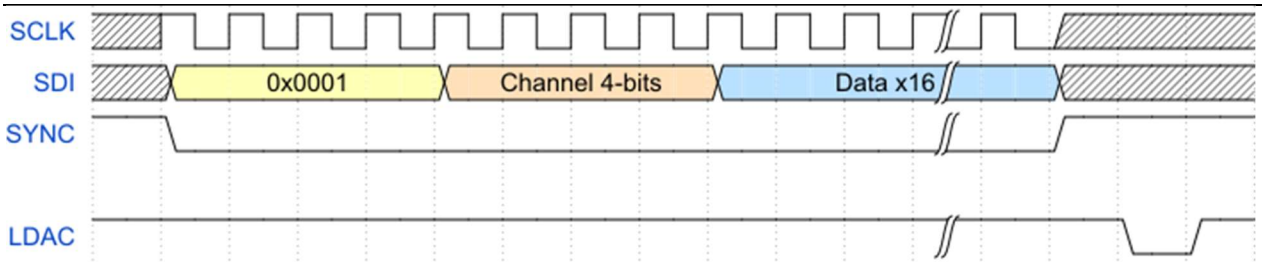
HV Output Pinout



Power Input Pinout



The 190V evaluation board features a 4 wire interface compatible with SPI. The board is capable of communication up to 50MHz for custom application requiring faster update rates. See the figure below for a typical timing diagram. This is provided for you in the sample source code.



3.0 Evaluation Software Usage

Special Note – *Always power the Raspberry Pi before powering the driver board. When shutting down, always power off the driver board first.* ***Failure to do so will result in damage!***

Download the programs and example waveform files to the raspberry pi.

This program was compiled for use on the Raspberry Pi 3 running Raspbian. Normally you will not need to recompile from source code. The program comes compiled already. If you are using a different environment you will need to recompile before running. See section 6.

Open a terminal and change to the directory containing the program and wave files. Start the program with a waveform csv file as a parameter.

```
./csv_driver_190V [insert name of waveform file]
```

Example…

```
./csv_driver_190V arbitrary_waveform_001.csv
```

Troubleshooting…

In order to run the program, you may need to change the permissions of the file. Run this command to allow the program to run.

```
sudo chmod 755 csv_driver_190V
```

4.0 Running Waveforms with the Evaluation Program

The evaluation program accepts a csv file as a parameter. The format of the file is critical and must be followed. Each line has three parameters, the channel, the timing, and the output value. The channel ranges from 1 to 4. The timing is in microseconds. And the output value is a floating point number in the range from 0.0 to 190.0. Each channel MUST be an integer. To put the device into loop state a special row of all zeros (0) made be used.

Waveform File Format

- The file should have three columns
  - Column 1 is either 1,2,3,4 or 0
  - Column 2 is time in Microseconds with no decimal
  - Column 3 is the output voltage. Maximum of 190V. May contain a decimal
  - Be sure to end each line with a comma
- The file should end in .csv
- If the waveform should run as a periodic waveform, put 0,0,0, as the last line
  - If the last line of the file is a channel number, the file will be run one time and end
- The file must be sorted by the time column except for the 0,0,0, row at the end of the file.
- Please check the documentation for the maximum voltage for the mirror being used.
  - Damage to the target device is possible if the voltage rating is exceeded.
  - Please consult the device specifications for maximum voltage of the target device.

When creating a waveform file, the file must be sorted by the microsecond's column so that each row is greater than or equal to the time of the previous row. Channel order does not matter.

(Channel)	(Microseconds)	(Output Voltage)
1,	200,	50.0,
2,	4000,	32.1,
1,	8000,	37.2,
2,	8000,	10.0,
0,	0,	0,

The file format of the csv must have commas (,) at the end of each line.

5.0 Example Waveform

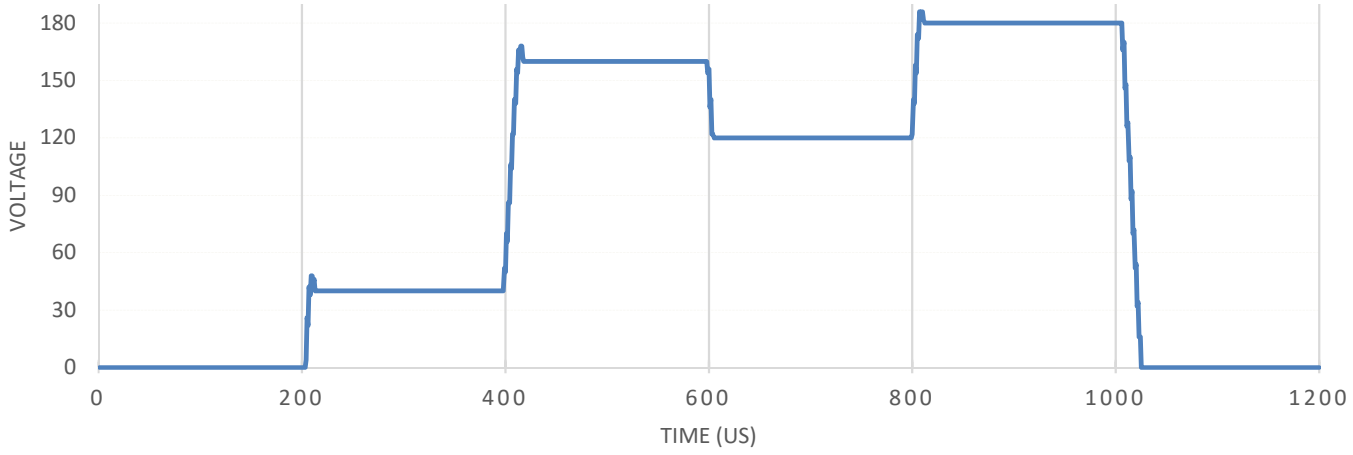
Several example files are included with the example program in the SDK Zip File.

Here is an example file and its generated output. Note that the waveform loop restarts at time 1000. Be sure to include the start point and the end point of the wave you wish to define. In this example, we want to hold at 50V for 200us at the end. The start of the 50V starts at 800us and ends at 1000us. If we did not define 50V at 1000us, the waveform would not be generated predictably.

File Contents of example.csv

```
1,0,0,
1,200,40,
1,400,160,
1,600,120,
1,800,180,
1,1000,180,
0,0,0,
```

Captured Output – Note the pattern restarts at 1000us



For a better estimate of performance, please refer to Section 1.0 Performance Characteristics. For advanced performance and advanced timing, the Raspberry Pi should be replaced with dedicated hardware.

6.0 Source Code and Compiling

You don't need to compile the program to use it. A compiled version for the 190V and 50V drivers is included in the SDK zip file. We recommend using a fresh install of Raspbian on the Pi 3 to avoid issues. If you would like to see the code or would like to customize the program we included the code.

The source code for the evaluation program is included in the SDK Zip File. We recommend using a fresh install of Raspbian on the Pi 3 to ensure a working environment.

For Advanced Users Who Are Interested In Adding Features

To re-compile our evaluation program follow the steps below.

First check that wiringPi is installed by opening a terminal and typing the command:

```
gpio –v
```

This should response with the version of wiring installed. This should be installed with Raspbian. If you are using another flavor of operating system, you'll need to make sure wiringPi is installed.

Compile the code on the raspberry pi with the following command.

```
g++ -D V190 -Wall –o csv_driver_190V csv_driver.cpp –lwiringPi
```

**This command should be hand typed to avoid issues with hidden characters. Copy and Paste of this line can cause issues.