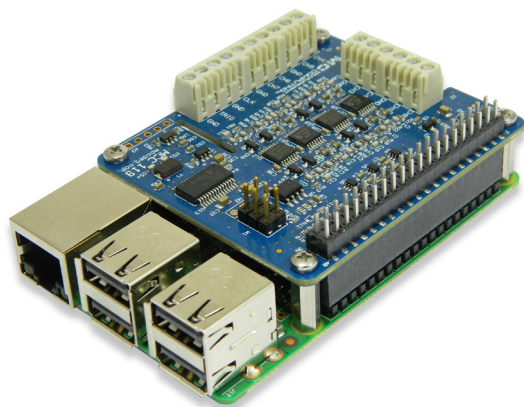


# MCC 118

## Voltage Measurement DAQ HAT for Raspberry Pi®



The MCC 118 is a 12-bit, high-speed 8-channel voltage measurement DAQ HAT. The MCC 118 is shown here connected to a Raspberry Pi (not included).

### Features

- Eight 12-bit voltage inputs
- 100 kS/s max sample rate (320 kS/s aggregate for stacked boards)
- $\pm 10$  V input range
- Onboard sample buffers allow high-speed acquisition
- External scan clock I/O
- External digital trigger input
- Screw terminal connections
- Stack up to eight MCC HATs onto a single Raspberry Pi

### Software

- MCC DAQ HAT Library; available on GitHub

### Supported Operating Systems

- Linux®/Raspbian

### Programming API

- C, C++, Python

### Overview

The MCC 118 is a voltage HAT (Hardware Attached on Top) board designed for use with Raspberry Pi, the most popular single-board computer on the market today.

A HAT is an add-on board with a 40W GPIO (general purpose input/output) connector that conforms to the Raspberry Pi HAT specification.

The MCC 118 HAT provides eight single-ended (SE) analog inputs for voltage measurements. Up to eight MCC HATs can be stacked onto one Raspberry Pi.

### Raspberry Pi Interface

The MCC 118 header plugs into the 40-pin general purpose I/O (GPIO) connector on a user-supplied Raspberry Pi. The MCC 118 was tested for use with Raspberry Pi A+, B+, 2 and 3 models.

### HAT configuration

HAT configuration parameters are stored in an on-board EEPROM that allows the Raspberry Pi to automatically set up the GPIO pins when the HAT is connected.

### Stackable HATs

Up to eight MCC HAT boards can be stacked onto a single Raspberry Pi. Multiple boards can be synchronized using external clock and trigger input options.

Users can mix and match MCC HAT models in the stack.

### Analog Input

The MCC 118 provides eight 12-bit SE analog inputs. The analog input range is fixed at  $\pm 10$  V.

### Sample Rates

- Single-board: max throughput is 100 kS/s.
- Stacked boards: max throughput is 320 kS/s aggregate<sup>1</sup>.

### External Scan Clock

A bidirectional clock I/O pin lets users pace operations with an external clock signal or with the board's internal scan clock. Use software to set the direction.

### Digital Trigger

The external digital trigger input is software-configurable for rising or falling edge, or high or low level.

### Power

The MCC 118 is powered with 3.3 V provided by the Raspberry Pi through the GPIO header connector.

### MCC DAQ HAT Library

The open-source MCC DAQ HAT Library of commands in C/C++ and Python allows users to develop applications on the Raspberry Pi using Linux.

The library is available to download from [GitHub](#). Comprehensive API and hardware [documentation](#) is available.

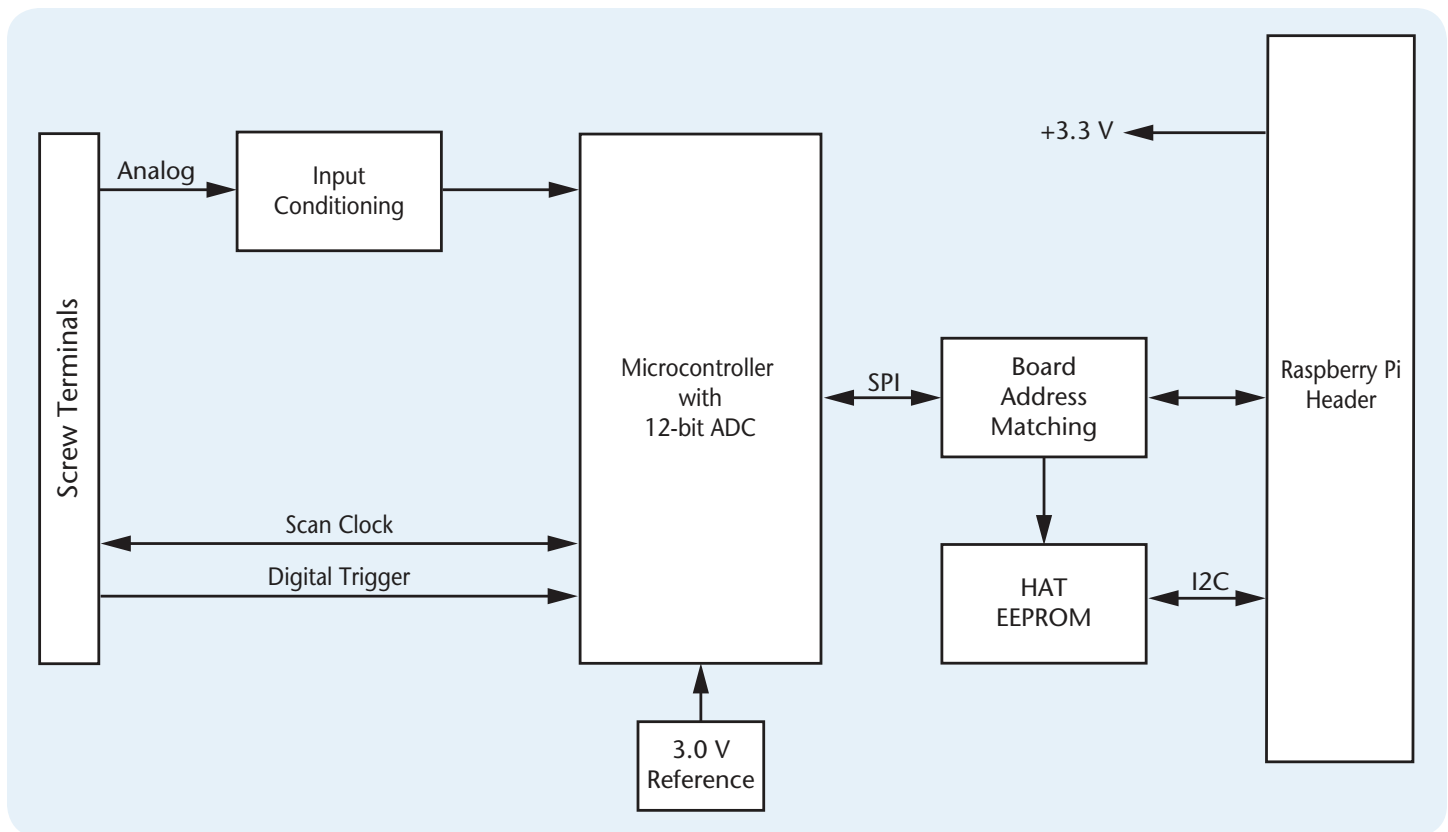
The MCC DAQ HAT Library supports operation with multiple MCC DAQ HATs running concurrently.

Console-based and user interface (UI) example programs are available for each API.

<sup>1</sup> Rate achieved using a Raspberry Pi 3.

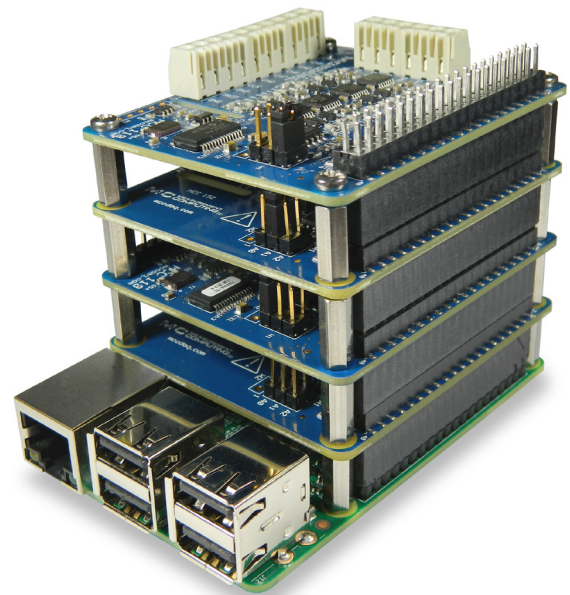
# MCC 118

## Block Diagram



### Stackable!

Connect up to eight MCC DAQ HATs onto a single Raspberry Pi. Onboard jumpers identify each board in the stack. Use an external clock and connect the trigger inputs to synchronize the acquisition.



# MCC 118

## Example Programs



### MCC DAQ HAT Examples

The MCC DAQ HAT Library includes example programs developed in C/C++ and Python that users can run to become familiar with the DAQ HAT library and boards; source code is included.

#### Console-Based (C/C++ and Python)

Console-based examples are provided that demonstrate how to perform continuous and finite scans, trigger an acquisition, and synchronously acquire data from multiple DAQ HATs (source included). The continuous\_scan example is shown here.

```
pi@janet_pi:~/daqhats/examples/c/mcc118/continuous_scan $ ./continuous_scan
Selected MCC118 HAT device at address 0

MCC118 continuous scan example
Functions demonstrated:
  mcc118_a_in_scan_start
  mcc118_a_in_scan_read
  mcc118_a_in_scan_stop
Channels: 0, 1, 2, 3
Requested scan rate: 1000.00
Actual scan rate: 1000.00
Options: OPTS_CONTINUOUS

Press ENTER to continue ...

Internal data buffer size: 40000
Starting scan ... Press ENTER to stop

Samples Read  Scan Count  Channel 0  Channel 1  Channel 2  Channel 3
474           32970     4.39551 V  5.24444 V  1.74411 V  1.74104 V
```

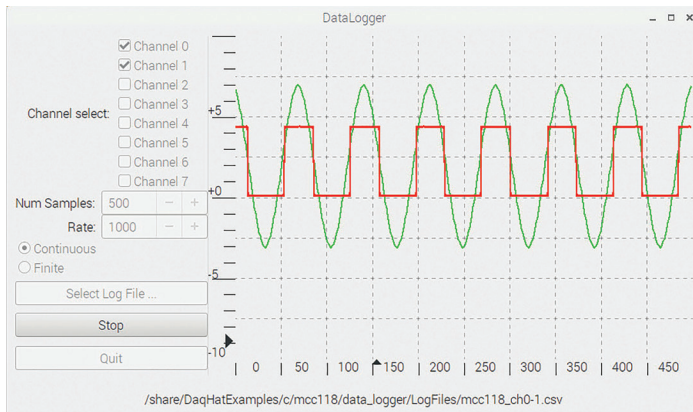
Display the value of each input channel in a terminal window

#### User Interface

Example programs featuring a user interface are provided in different formats. Examples of each are shown here.

#### DataLogger (C/C++)

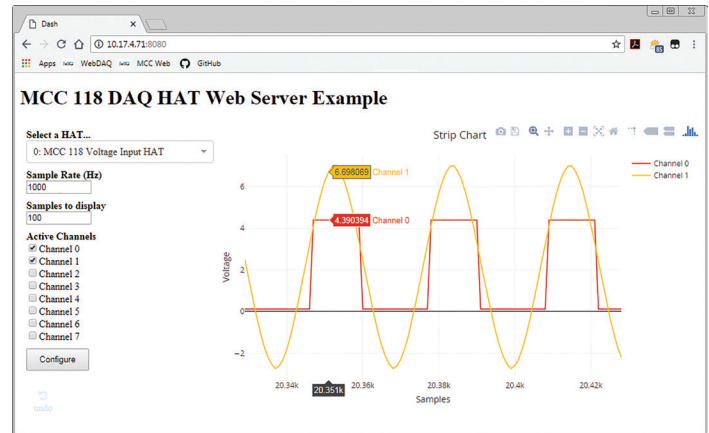
The datalogger example shows how to acquire data from the MCC 118, display the data on a strip chart, and log the data to a CSV file. This example can be run from the terminal or accessed using an IDE such as CodeBlocks or Geany (source included).



Configure options, plot data on a strip chart, and log data to a file

#### Web Server (Python)

The web server example lets users configure acquisition options and view acquired data from a browser window. This example is written for Python (source included).

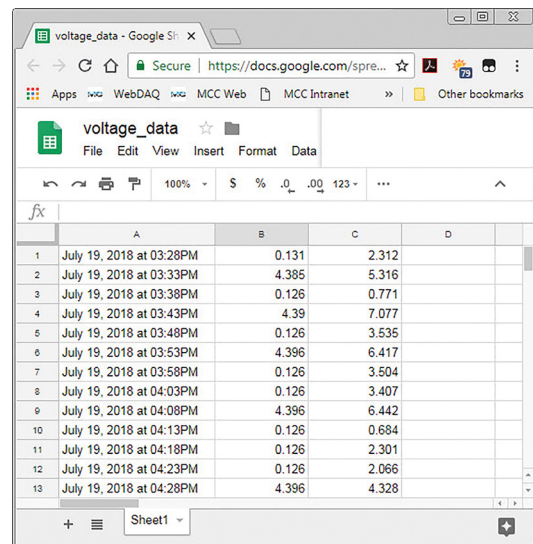


Configure options and view strip chart data from your browser

#### IFTTT Applet (Python)

IFTTT (If This Then That) is a free web-based service that interacts with apps and hardware to automate various functions. The IFTTT DAQ HAT example reads two MCC 118 channels at regular intervals, and writes the data to a Google Sheets spreadsheet.

Users can remotely monitor the spreadsheet from Google Drive. An IFTTT account is required. This example is written for Python (source included).



View logged data on a Google Sheets spreadsheet from your browser

### Specifications

All specifications are subject to change without notice.  
Typical for 25 °C unless otherwise specified.

#### Analog Input

A/D converter type: Successive approximation  
ADC resolution: 12 bits

Number of channels: 8 single-ended

Input voltage range:  $\pm 10$  V

Absolute maximum input voltage

CHx relative to GND:  $\pm 25$  V max (power on or power off)

Input impedance: 1 M $\Omega$  (power on or power off)

Input bias current

10 V input: -12  $\mu$ A

0 V input: 2  $\mu$ A

-10 V input: 12  $\mu$ A

Monotonicity: Guaranteed

Input bandwidth, small signal (-3 dB): 150 kHz

Maximum working voltage

Input range relative to AGND:  $\pm 10.1$  V max

Crosstalk (adjacent channels, DC to 10 kHz): -75 dB

Input coupling: DC

Recommended warm-up time: 1 minute min

Sample rate, hardware paced

Internal scan clock: 0.004 S/s to 100 kS/s, software-selectable

External scan clock: 100 kS/s max

Sampling mode: 1 A/D conversion for each configured channel per clock

Conversion time, per channel: 8  $\mu$ s

Scan clock source

Internal scan clock

External scan clock input on terminal CLK

Channel queue: Up to eight unique, ascending channels

Throughput, Raspberry Pi<sup>®</sup> 2 / 3

Single board: 100 kS/s max

Multiple boards: Up to 320 kS/s aggregate\*

Throughput, Raspberry Pi A+ / B+

Single board: Up to 100 kS/s\*

Multiple boards: Up to 100 kS/s aggregate\*

\*When using multiple boards, throughput depends on the load on the Raspberry Pi processor. The highest throughput may be achieved by using a Raspberry Pi 3.

#### Accuracy

##### Analog Input DC Voltage Measurement Accuracy

Range:  $\pm 10$  V

Gain error (% of reading): 0.098 max

Offset error: 11 mV max

Absolute accuracy at full scale: 20.8 mV

Gain temperature coefficient(% reading/°C): 0.016

Offset temperature coefficient (mV/°C): 0.87

##### Noise Performance

For peak to peak noise distribution, the input channel is connected to AGND at the input terminal block, and 12,000 samples are acquired at the maximum throughput.

Range:  $\pm 10$  V

Counts: 5

LSBrms: 0.76

#### External Digital Trigger

Trigger source: TRIG input

Trigger mode: Software-selectable for rising or falling edge, or high or low level

Trigger latency:

Internal scan clock: 1  $\mu$ s max

External scan clock: 1  $\mu$ s + 1 scan clock cycle max

Trigger pulse width: 125 ns min

Input type: Schmitt trigger, weak pull-down to ground (approximately 10 K)

Input high voltage threshold: 2.64 V min

Input low voltage threshold: 0.66 V max

Input voltage limits: 5.5 V absolute max, -0.5 V absolute min, 0 V recommended min

#### External Scan Clock Input/Output

Terminal name: CLK

Terminal types: Bidirectional, defaults to input when not sampling analog channels

Direction (software-selectable):

Output: Outputs internal scan clock, active on rising edge

Input: Receives scan clock from external source, active on rising edge

Input clock rate: 100 kHz max

Clock pulse width: 400 ns min

Input type: Schmitt trigger, weak pull-down to ground (approximately 10 K), protected with a 150 k $\Omega$  series resistor

Input high voltage threshold: 2.64 V min

Input low voltage threshold: 0.66 V max

Input voltage limits: 5.5 V absolute max, -0.5 V absolute min, 0 V recommended min

Output high voltage: 3.0 V min (IOH = -50  $\mu$ A), 2.65 V min (IOH = -3 mA)

Output low voltage: 0.1 V max (IOL = 50  $\mu$ A), 0.8 V max (IOL = 3 mA)

Output current:  $\pm 3$  mA max

#### Memory

Data FIFO: 7 K (7,168) analog input samples

Non-volatile memory: 4 KB (ID and calibration storage, no user-modifiable memory)

#### Power

Supply current, 3.3 V supply:

Typical: 35 mA

Maximum: 55 mA

#### Interface

Raspberry Pi GPIO pins used:

GPIO 8, GPIO 9, GPIO 10, GPIO 11 (SPI interface)

ID\_SD, ID\_SC (ID EEPROM)

GPIO 12, GPIO 13, GPIO 26, (Board address)

Data interface type: SPI slave device, CEO chip select

SPI mode: 1

SPI clock rate: 10 MHz, max

#### Environment

Operating temperature: 0 °C to 55 °C

Storage temperature: -40 °C to 85 °C max

Relative humidity: 0% to 90% non-condensing

#### Mechanical

Dimensions (L x W x H): 65 x 56.5 x 12 mm (2.56 x 2.22 x 0.47 in.) max

## Order Information

### Hardware

Part No.	Description
MCC 118	12-bit, 8-channel voltage measurement DAQ HAT; Raspberry Pi A+, B+, 2, 3 or later required.

### Software

Part No.	Description
MCC DAQ HAT Library	Open-source library for developing applications in C, C++, and Python on Linux for MCC DAQ HAT hardware. Available for download on GitHub at <a href="https://github.com/mccdaq/daqhats">https://github.com/mccdaq/daqhats</a> .