

# MCC 128

## 16-bit Voltage Measurement DAQ HAT for Raspberry Pi®



The MCC 128 is a 16-bit, high-speed voltage measurement DAQ HAT. The MCC 128 is shown here connected to a Raspberry Pi (not included).

### Features

- 8 SE/4 DIFF 16-bit voltage inputs
- Single-ended and differential input modes
- 100 kS/s max sample rate (320 kS/s aggregate for stacked boards)
- Multiple input ranges
- Onboard sample buffers allow high-speed acquisition
- External scan clock
- 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 128 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 128 HAT provides 8 SE/4 DIFF analog inputs for voltage measurements. Up to eight MCC HATs can be stacked onto one Raspberry Pi.

### Raspberry Pi Interface

The MCC 128 header plugs into the 40-pin general purpose I/O (GPIO) connector on a user-supplied Raspberry Pi. The MCC 128 was tested for use with all Raspberry Pi models with the 40-pin GPIO connector.

### 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 128 provides 8 single-ended or 4 differential analog inputs. The input range is selectable for  $\pm 10$  V,  $\pm 5$  V,  $\pm 2$  V, or  $\pm 1$  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 edge or level sensitive, rising or falling edge, or high or low level.

### Power

The MCC 128 is powered with 5 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.

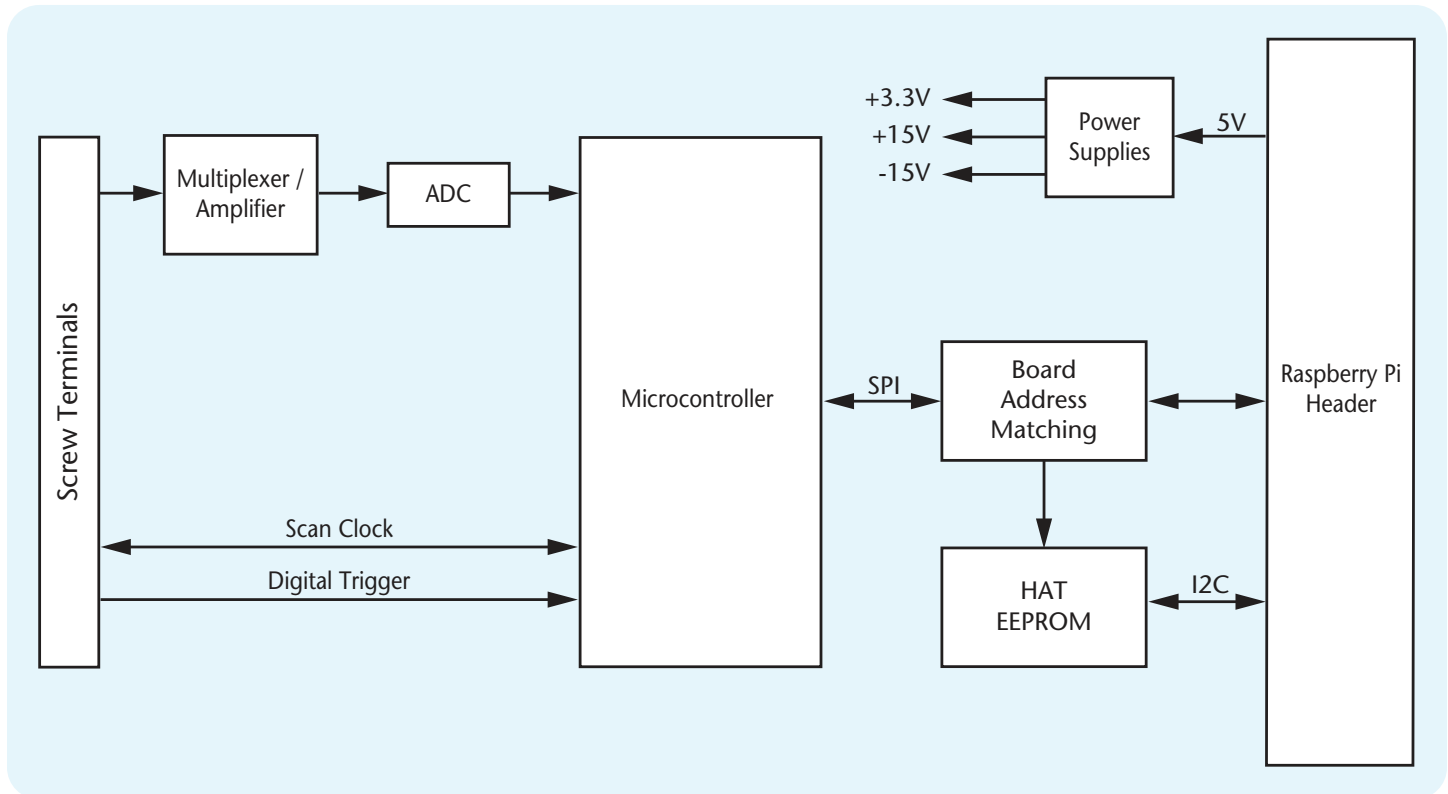
The MCC DAQ HAT Library supports operation with multiple MCC DAQ HATs running concurrently.

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

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

# MCC 128

## Block Diagram



### OEM Version

The MCC 128-OEM is designed with (unpopulated) header connectors instead of screw terminals, and is functionally equivalent to the standard version. The MCC 128-OEM accepts 1×6 and 1×10 0.1 in. spacing header connectors.

### 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 128

## 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 MCC 128 HATs using the external clock and trigger options. The continuous\_scan example is shown here. Source code is included.

```
File Edit Tabs Help
pi@janet_pi:~$ cd daqhats_dev/examples/c/mcc128/continuous_scan
pi@janet_pi:~/daqhats_dev/examples/c/mcc128/continuous_scan $ ./continuous_scan

Selected MCC 128 device at address 0

MCC 128 continuous scan example
Functions demonstrated:
mcc128_a_in_scan_start
mcc128_a_in_scan_read
mcc128_a_in_scan_stop
mcc128_a_in_mode_write
mcc128_a_in_range_write
Input mode: SINGLE_ENDED
Input range: +/- 10 V
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
477           113565    7.48128 V  6.38712 V  10.73094 V  9.92967 V
```

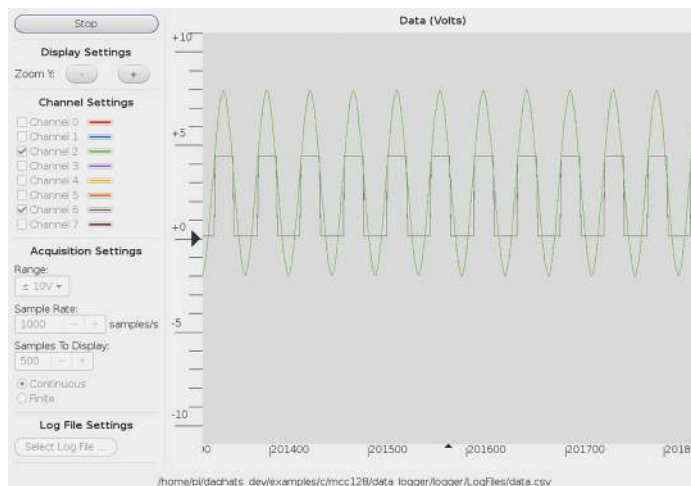
Display the value of each input channel in a terminal window

#### User Interface

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

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

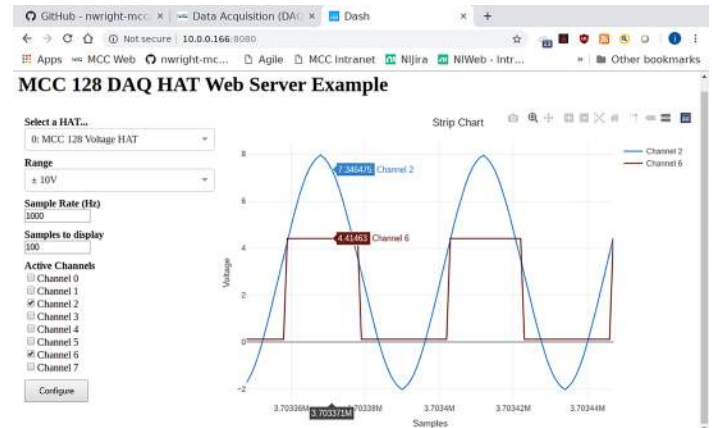
The datalogger example shows how to acquire data from the MCC 128, display the data on a strip chart, and log the data to a CSV file. This example can be run from the terminal.



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).

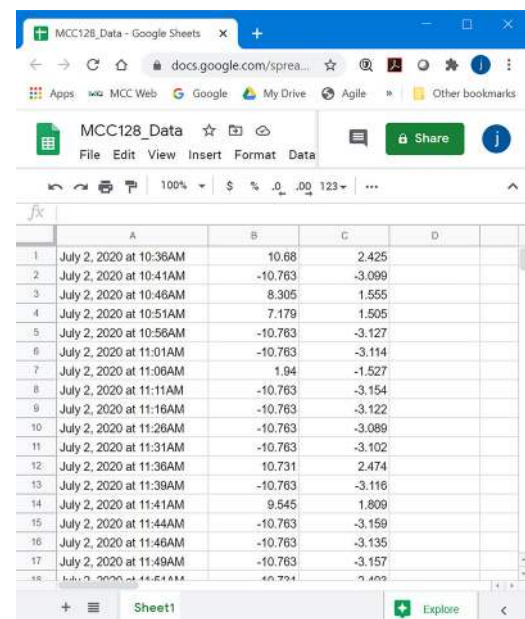


Configure options and view strip chart data from your browser

#### 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 128 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

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: 16 bits

#### Input modes

Single ended (CHx – AGND)

Differential (CHxH – CHxL)

Number of channels: 8 single-ended or 4 differential; software selectable

Input voltage ranges:  $\pm 10$  V,  $\pm 5$  V,  $\pm 2$  V,  $\pm 1$  V

#### Absolute maximum input voltage

CHx relative to AGND:  $\pm 30$  V max (power on),  $\pm 20$  V max (power off)

Input impedance:  $>1$  G $\Omega$

Input bias current:  $\pm 200$  pA, typ

INL:  $\pm 1.8$  LSB

DNL: 16 bits no missing codes

CMRR, DC to 5 kHz

$\pm 10$  V range: 56 dB

$\pm 5$  V range: 57 dB

$\pm 2$  V range: 61 dB

$\pm 1$  V range: 65 dB

Input bandwidth, small signal (-3 dB): 220 kHz

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

Crosstalk: Adjacent channels, DC to 10 kHz: -70 dB

Input coupling: DC

Recommended warm up time: 1 minute min

#### Sampling rate, hardware paced

Internal scan clock: 1 S/s to 100 kS/s, software-selectable

External scan clock: 100 kS/s max

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

Conversion time, per channel: 9.8  $\mu$ s

Scan clock source: Internal scan clock, External scan clock input on terminal CLK

Channel queue: Up to eight unique, ascending channels in a single range and mode

Throughput (dependent on the load on the Raspberry Pi processor. The highest throughput may be achieved by using a Raspberry Pi 3 B+.)

#### Raspberry Pi<sup>®</sup> 2 / 3 / 4

Single board: 100 kS/s max

Multiple boards: Up to 320 kS/s aggregate

#### Raspberry Pi A+ / B+

Single board: Up to 100 kS/s

Multiple boards: Up to 100 kS/s aggregate

### Accuracy

Analog input DC voltage measurement accuracy		
Range	At full scale (typ at 25 °C)	At full scale (maximum over temperature)
SE $\pm 10$ V	$\pm 6$ mV	$\pm 34$ mV
SE $\pm 5$ V	$\pm 3$ mV	$\pm 17$ mV
SE $\pm 2$ V	$\pm 1.2$ mV	$\pm 7$ mV
SE $\pm 1$ V	$\pm 600$ $\mu$ V	$\pm 3.4$ mV
DIFF $\pm 10$ V	$\pm 14$ mV	$\pm 42$ mV
DIFF $\pm 5$ V	$\pm 6.5$ mV	$\pm 21$ mV
DIFF $\pm 2$ V	$\pm 2.1$ mV	$\pm 8$ mV
DIFF $\pm 1$ V	$\pm 900$ $\mu$ V	$\pm 3.7$ mV

### Noise Performance

Range	Vrms
$\pm 10$ V	350 $\mu$ V
$\pm 5$ V	220 $\mu$ V
$\pm 2$ V	150 $\mu$ V
$\pm 1$ V	100 $\mu$ V

For the peak to peak noise distribution test, the input channel is in single-ended mode connected to AGND at the input terminal block, and 100,000 samples are acquired at the maximum throughput. The performance is the same for single-ended and differential mode

### 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, 100 k $\Omega$  pull-down to ground

Input high voltage threshold: 1.3 V min

Input low voltage threshold: 1.5 V max

Input hysteresis: 0.4 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, 100 k $\Omega$  pull-down to ground

Input high voltage threshold: 1.3 V min

Input low voltage threshold: 1.5 V max

Input hysteresis: 0.4 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 = -100  $\mu$ A), 2.4 V min (IOH = -4 mA)

Output low voltage: 0.1 V max (IOL = 100  $\mu$ A), 0.4 V max (IOL = 4 mA)

Output current:  $\pm 4$  mA max

### Memory

Data FIFO: 7 K (73,728) analog input samples

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

### Power

Supply current, 5 V supply:

Typical: 85 mA

Maximum: 135 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)

GPIO16, 20 (Reset, IRQ)

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

# MCC 128

## Ordering



### Hardware

Part No.	Description
MCC 128	16-bit, 8 channel voltage measurement DAQ HAT. Raspberry Pi model with the 40-pin GPIO connector required.
MCC 128-OEM	16-bit, 8 channel voltage measurement DAQ HAT with (unpopulated) header connectors instead of screw terminals. Raspberry Pi model with the 40-pin GPIO connector 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> .