# EIGHT CHANNEL RTD
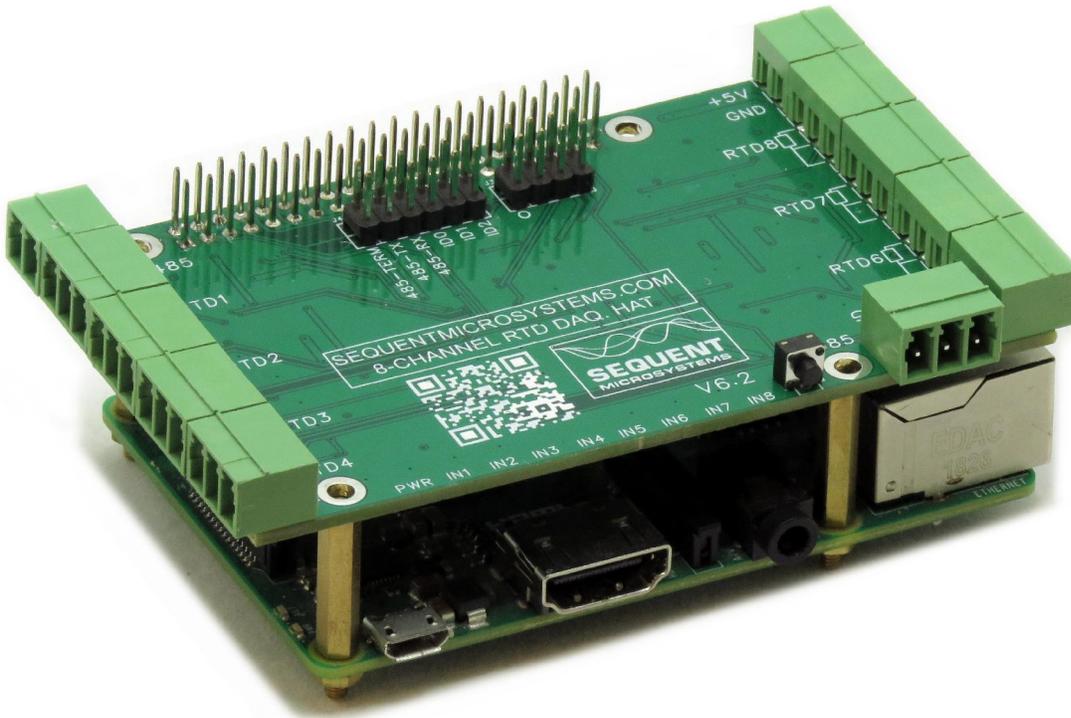
# DATA ACQUISITION CARD FOR RASPBERRY PI

## USER'S GUIDE VERSION 6.2

### SequentMicrosystems.com

# GENERAL DESCRIPTION



The 8-CHANNEL RTD Data Acquisition Card is a stackable expansion card for Raspberry Pi. It is compatible with all Raspberry Pi versions from Zero to 4 and offers a compact and inexpensive solution for reading and storing data from up to 64 RTD-100 temperature sensors.

Pluggable connectors make the RTD card easy to use when multiple cards are stacked up. Programmable LEDs can be activated on a predefined threshold of each input. A general purpose push button can be used to shut down Raspberry Pi without a keyboard and monitor, or to implement any auxiliary function.

Since temperature measurement using RTD is based on resistance, the resistance in the lead wires and connectors must be taken into account when calculating the overall resistance in the system circuit. Using a 3-wire circuit design allows lead wire resistance to be factored out of the overall calculation.

Mechanically, the RTD Card adheres to the Sequent Microsystems Modular Industrial format. It can be installed in the free 3D printable stackable enclosure. All the cards in this format have the same mechanical specifications.

# FEATURES

- 24 bit delta-sigma A/D converters
- Factory accuracy: 0.1%
- Maximum accuracy (through calibration): 0.01%
- Maximum acquisition speed 60 cps.
- Eight layer stackable to 64 RTD channels
- RS485/MODBUS transceiver
- PT100 sensors
- Programmable threshold LEDs on all inputs
- Pluggable connectors
- Reverse polarity protection on power supply connector
- General Purpose Push Button
- On-board hardware watchdog capable of power cycling the Raspberry Pi.
- On-board resettable fuse
- Command Line, Node-RED and Python Drivers
- OpenPLC Module, Modbus RTU

Up to eight 8-Channel RTD cards can be stacked on top of one Raspberry Pi. Each card is equipped with a 32-bit STM processor running at 48MHz. RTD inputs are processed using 24 bit Delta-Sigma converters and can achieve up to 0.01 °C accuracy.

The 8-Channel RTD cards share a serial I2C bus using only two of the Raspberry Pi's GPIO pins to manage all eight cards. This feature leaves the remaining 24 GPIOs available for the user. The card works with all versions of Raspberry Pi, from Zero to 5.

# WHAT IS IN YOUR KIT

1.  8-Channel RTD Data Acquisition Card for Raspberry Pi
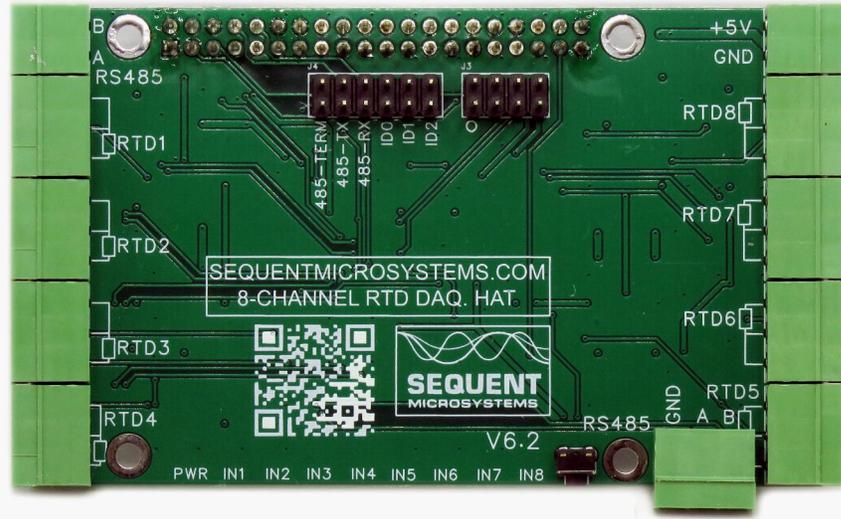


2.  Mounting hardware



    a.  Four M2.5x19mm male-female brass standoffs
    b.  Four M2.5x5mm brass screws
    c.  Four M2.5 brass nuts
3.  Two jumpers.



    You do not need the jumpers when using only one 8-Channel RTD card. See STACK LEVEL JUMPERS section if you plan to use multiple 8-Channel RTD cards.
4.  All the required female mating connectors.
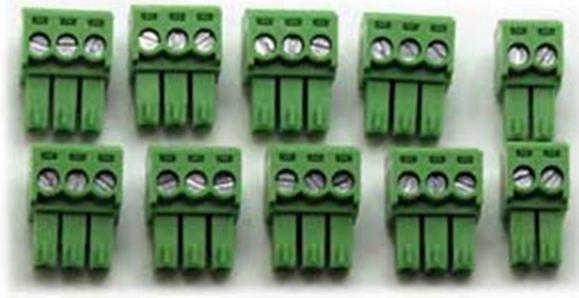
## QUICK START-UP GUIDE

1. Plug your 8-Channel RTD card on top of your Raspberry Pi and power up the system.
2. Enable I2C communication on Raspberry Pi using raspi-config.

3. Install the 8-Channel RTD software from github.com:

   a. ~$ git clone https://github.com/SequentMicrosystems/megartd-rpi.git

   b. ~$ cd /home/pi/megartd-rpi

   c. ~/megartd-rpi$ sudo make install

4. ~/megartd-rpi$ megartd

   The program will respond with a list of available commands.

## BOARD LAYOUT
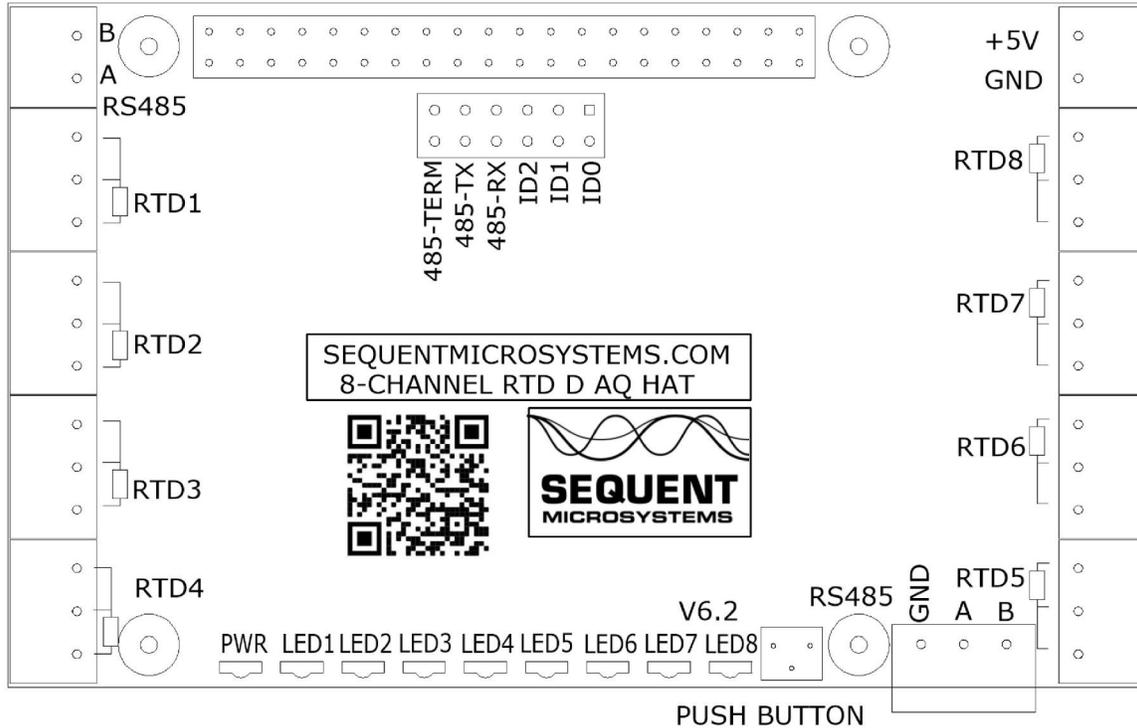


Your 8-Channels RTD Card comes with appropriate mounting hardware. Up to eight cards can be stacked on top of one Raspberry Pi. The power LED indicates when power is applied to the board. Each channel has one LED which can be programmed to turn on at the desired input threshold.

The 3-wire connection to the RTD-100 sensors is indicated on the board's silkscreen. The board can be powered either from Raspberry Pi or from its own +5 pluggable connector. Only one 5V power supply is required for a system.

The three left sections of the J11 jumper are used to select the stack level. The two right sections switch the RS485 port source between Raspberry Pi and local processor.

# STACK LEVEL JUMPERS

The 8-Channel RTD Card is controlled by Raspberry Pi using only the I2C interface. It occupies the address space 0x40 - 0x47. The local address can be configured using the Stack Level Jumpers. Two jumpers are provided for your convenience. A maximum stack of eight cards requires a total of 12 jumpers. The stack level jumpers are the 3 left positions of J11.

Cards can be installed on Raspberry Pi in any order. The 3 position jumper is selecting the stack level of the card, as follows:
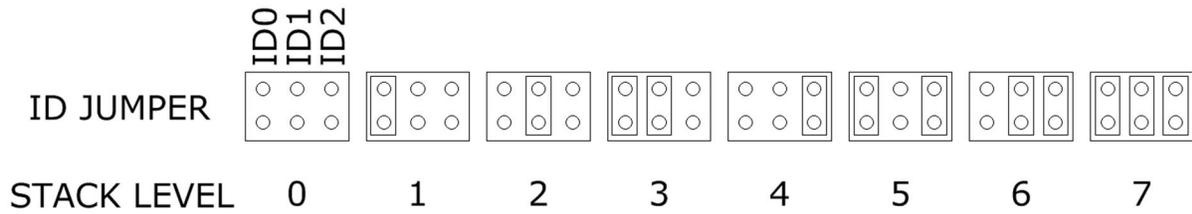
# POWER REQUIREMENTS

The 8-Channel RTD Card requires +5V power, supplied either from the Raspberry Pi expansion bus, or from its own pluggable power connector. A local 3.3V regulator powers the rest of the circuits (See Schematic 9).

<div style="border:1px solid black;">

**We recommend using only one +5V source to power
both the Raspberry Pi and the 8-Channel RTD card.**

</div>

Raspberry Pi 3 current consumption:          250 mA @ +5V (could be as high as 2A)

 8-Channel RTD Card current consumption:      50 mA @ +5V

The connector which powers the RTD card can handle up to 4A and is protected by a 3A resettable fuse. We recommend using a 5V regulated power supply rated at 3A or higher.

The 8-Channel RTD Card can be stacked up to eight levels. A multi-stack configuration can be powered from any of the cards.

# MECHANICAL SPECIFICATIONS

The 8-Channel RTD Card has the mechanical specifications of the Sequent Microsystems Modular Industrial format. In can be installed in the 3D printable stackable enslosure.
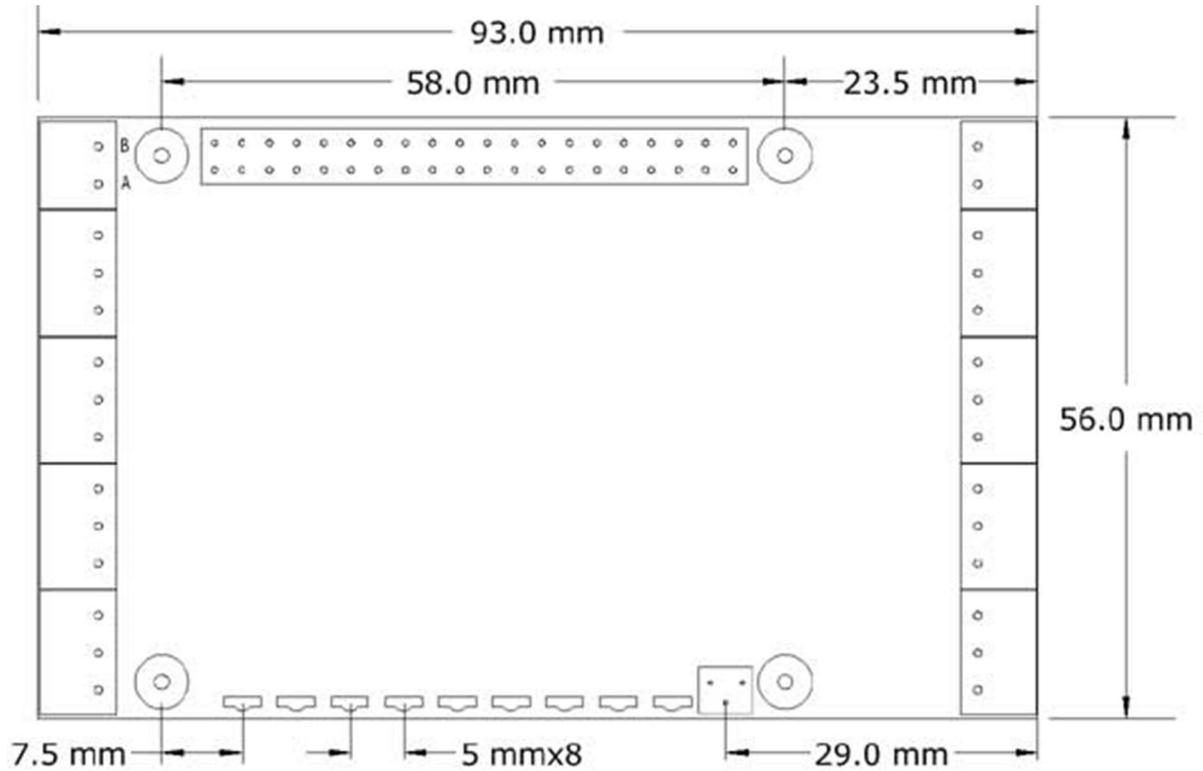


The cardF is compatible with any card manufactured by Sequent Microsystems and can be mounted in any order. You may also mix cards from other vendors, assuming they do not use the same I2C address.

# HARDWARE WATCHDOG

The 8-Channel RTD card contains a built-in hardware watchdog which will guarantee that your mission-critical project will recover and continue running even if Raspberry Pi software hangs up. After power up the watchdog is disabled, and becomes active after it receives the first reset.

The default timeout is 120 seconds. Once activated, if it does not receive a subsequent reset from Raspberry Pi within 2 minutes, the watchdog cuts the power and restores it after 10 seconds.

Raspberry Pi needs to issue a reset command on the I2C port before the timer on the watchdog expires. The timer period after power up and the active timer period can be set from the command line. The number of resets is stored in flash and can be accessed or cleared from the command line. All the watchdog commands are described by the online help function.

## SOFTWARE SETUP

1. Have your Raspberry Pi ready with the latest OS.

2. Enable I2C communication:

    ~$ sudo raspi-config

| | | |
|---|---|---|
| 1. Change User Password | Change password for default user | |
| 2. Network Options | Configure network settings | |
| 3. Boot Options | Configure options for start-up | |
| 4. Localisation Options | Set up language and regional settings to match.. | |
| 5. Interfacing Options | Configure connections to peripherals | |
| 6. Overclock | Configure overclocking for your Pi | |
| 7. Advanced Options | Configure advanced settings | |
| 8. Update | Update this tool to the latest version | |
| 9. About raspi-config | Information about this configuration | |

| | | |
|---|---|---|
| P1 | Camera | Enable/Disable connection to the Raspberry Pi Camera |
| P2 | SSH | Enable/Disable remote command line access to your Pi |
| P3 | VNC | Enable/Disable graphical remote access to your Pi using... |
| P4 | SPI | Enable/Disable automatic loading of SPI kernel module |
| P5 | I2C | Enable/Disable automatic loading of I2C kernel module |
| P6 | Serial | Enable/Disable shell and kernel messages to the serial port |
| P7 | 1-Wire | Enable/Disable one-wire interface |
| P8 | Remote GPIO | Enable/Disable remote access to GPIO pins |

4. Install the megartd software from github.com:

    ~$ git clone https://github.com/SequentMicrosystems/megartd.git

5.    ~$ cd /home/pi/megartd

6.    ~/megartd-rpi$ sudo make install

7.    ~/megartd-rpi$ megartd

The program will respond with a list of available commands.

Type "megartd -h" for online help.

After installing the software, you can update it to the latest version with the commands:

1.    ~$ cd /home/pi/megartd-rpi

2.    ~/megartd-rpi$ git pull

3.    ~/megartd-rpi$ sudo make install

## CALIBRATION

The A/D converter measures the temperature by sending a 1mA current through the RTD-100 sensors and measuring the voltage. The current generator is using a 820 Ohms, 0.1% resistor. Thus, the accuracy of an un-calibrated device is limited to 0.1%.

The user can further calibrate the board using the command line and a precision, 100 Ohms resistor. First, short the input to be calibrated and issue the command:

megartd <id> c[alibrate] <channel> 0

Next, connect to the input to be calibrated a precision 0.01% 100 ohms resistor and issue the command:

megartd <id> c[alibrate] <channel> 100

Any resistor between 0 and 100 ohms can be used, but to achieve best results we recommend using a value as close as possible to the 100 ohms end of scale.

To reset the calibration to the factory value, issue the command

megarts <id> rc[alibrate]