MAX31732EVKIT#

Four–Channel Temperature Sensor

General Description

The MAX31732 Evaluation kit (EV kit) provides the hardware and software graphical user interface (GUI) necessary to evaluate the MAX31732, a multi-channel temperature sensor that monitors its own local temperature and the temperatures of four remote diode-connected transistors.

The MAX31732 is specified over the -40°C to +125°C operating temperature range and powered up with a 3.3V supply provided by the Micro-USB Type-B cable. It is available in a 4mm x 4mm 24-TQFN package.

The EV kit includes an installed MAX31732, quad external diode-connected transistors, and a MAX32625 PICO board (already installed in U2) to use as a USB-to-SMBus/I2C interface. It connects to the PC through a MAX32625 PICO board and a Micro-USB Type-B cable.

The EV kit is versatile, easy to use, and USB-powered. The Printed Circuit Board (PCB) layout has been carefully designed, assembled, and tested.

A Windows® 7 or higher operating system is required to use the GUI.

Design files for this circuit board are available.

Ordering Information **appears at end of data sheet.**

Quick Start

Required Equipment

- MAX31732EVKIT# Hardware
- Windows 7 or higher
- Micro-USB Type-B cable is included

Required Software GUI

- MAX31732GUI.exe

Procedure

In the following sections, software-related items are identified by bolding. Text in **Bold** refers to items directly from the EV kit software.

Follow the steps below to verify the board:

1. Ensure that all jumpers/shunts (J1–J10) are installed according to the EV kit Board Connections. See *Figure 1*.
2. Set the EV kit hardware on a nonconductive surface to ensure that nothing on the PCB gets shorted together.
3. Before starting the GUI, connect the USB side of the cable to the PC and the Micro-USB side to the EV kit Pico board U2. The power LED (D2) on U2 (PICO board MAX32625) should be slowly flashing green.
4. Visit: *https://www.analog.com/en/products/max31732.html* (Under the Tools & Simulation tab, download the latest version of the MAX31732GUI.exe software. Save the software to a temporary folder and unpack the zip file.) Install the EV kit software on the computer by running the MAX31732GUI.exe program inside the temporary folder.
5. Once the GUI installation is complete, find Analog Devices → MAX31732 GUI in your Windows start menu and run it. The GUI will display a splash screen. Then, a Windows dialog box should appear on your screen, showing a COM port found by the GUI. Click **Connect**. If the COM port does not appear, check your cable connection between the computer and the EV kit Pico board.
6. After these steps, the GUI must launch on your computer.

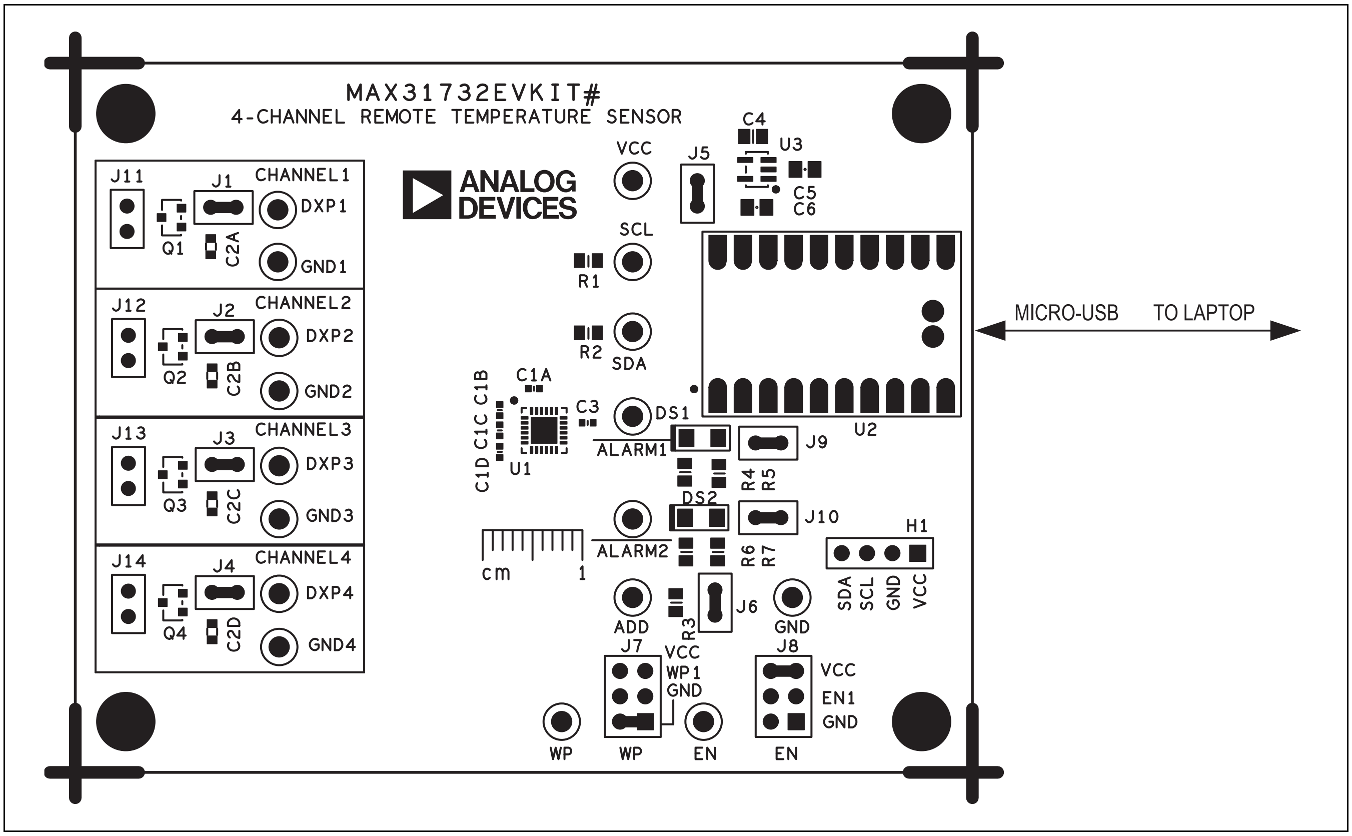


Figure 1. Board Connections

Table 1. Jumpers Default Connections

JUMPER	DEFAULT CONNECTION	FEATURE
J1	Enable Q1 Diode	Connects Remote Diode 1 to DXP1
J2	Enable Q2 Diode	Connects Remote Diode 1 to DXP2
J3	Enable Q3 Diode	Connects Remote Diode 1 to DXP3
J4	Enable Q4 Diode	Connects Remote Diode 1 to DXP4
J5	VDUT	Supplies 3.3V from USB to EV kit
J6	ADD	Connects pin ADD to GND
J7	WP	Connects pin WP to GND
J8	EN	Connects pin EN to VDUT
J9	$\overline{\text{ALARM1}}$	Connects pin $\overline{\text{ALARM1}}$ to PIN0_2
J10	$\overline{\text{ALARM2}}$	Connects pin $\overline{\text{ALARM2}}$ to PIN0_3

Setup and Operation

Once the connection has been successful, the user should see **Connected** at the bottom right corner of the GUI screen.

Ensure none of the green indicators under the Status tab in the GUI are red. They all must be green, as shown in *Figure 2*. The LEDs on the left side show a channel over temperature or under temperature condition, or a diode fault. The LEDs on the right side show what kind of a diode fault is detected. If a diode fault indicator is red, then that channel will be skipped, and the diode temperature corresponding to that channel will not be updated.

Menu and Status Bar

Under the "**File**" menu the user can simply **Exit**.

Under the "**Option**," the user can choose the Polling Rate either in Auto or "100ms-1000ms." If the user needs to record temperatures into a file, click **Log Polling Data to File** and then click **Start Polling** (under the status tab). The data can be saved as a .CSV format file.

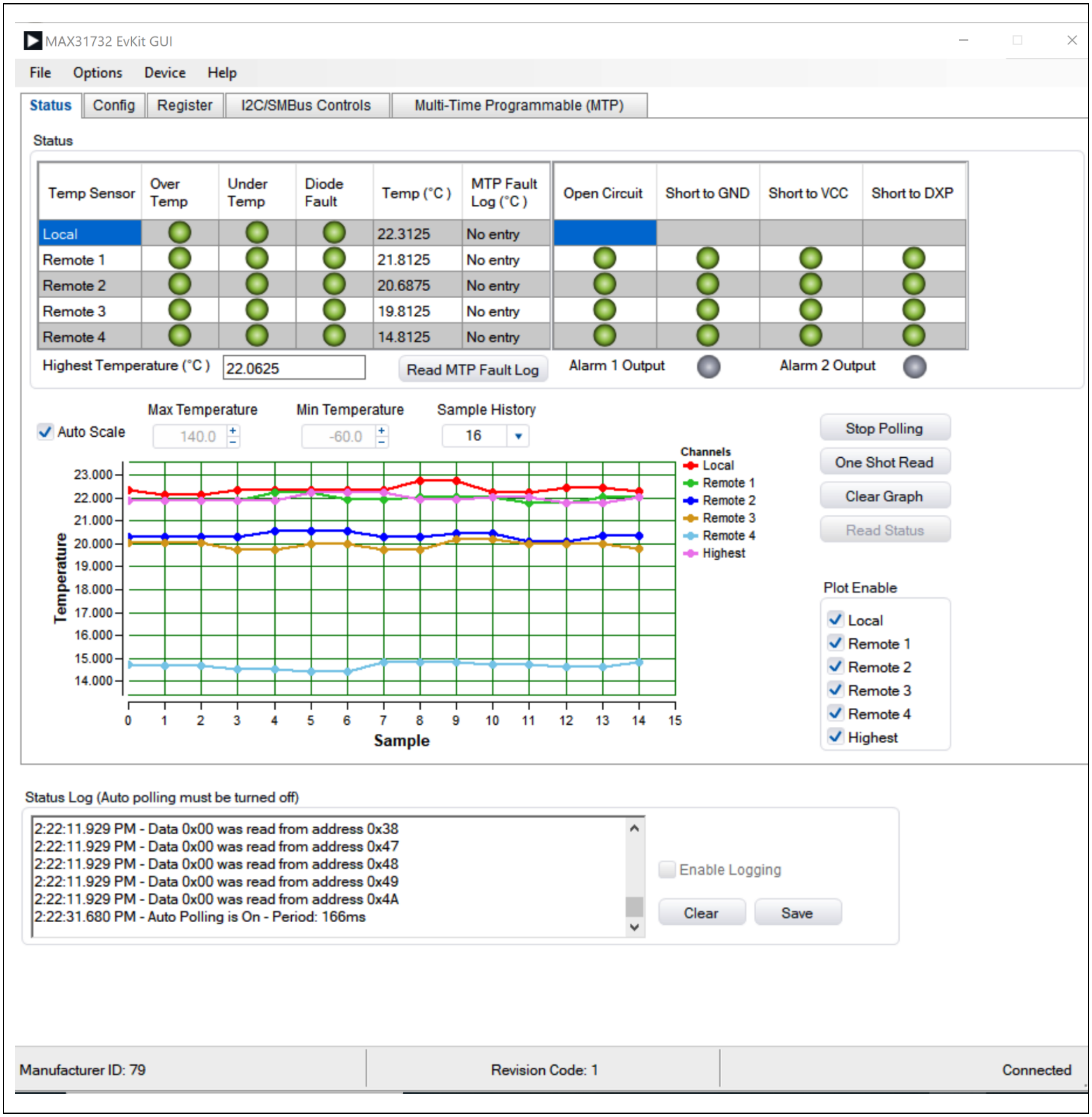
Under "**Device"** menu, the user can choose the preferred I2C clock frequency, as well as additional choices like **Target Device Address** and **Reset Devices** is equivalent to clicking to **POR** on the **Config tab**, which performs a soft-POR function (0x0F, Bit 6).

Status Tab

Under the **Status** tab sheet (*Figure 2*), the user can either click on ***One Shot Read*** or ***Start Polling*** to display all the fault status and temperature data.

This screen provides the user with several options, such as **Auto Scale** or the ability to manually select the **Maximum Temperature** or **Minimum Temperature** on the graph and alter the **Sample History** from 16 to 512. The ***Read Status*** button is used to read the current diode over/under-temp or diode fault status when auto-polling is off. When auto-polling is on (by clicking on the **Start Polling** button), then the current status is read at the **Polling Rate**.

The Temp (°C) column shows the current or last polled temperature. If the Highest Temperature Enabled is selected (Config Tab), then the highest temperature will be shown in the **Highest Temperature (°C) box** in the Status area. The **Read MTP Fault Log** button will query the MTP Fault Logging registers and display whatever is logged there in Celsius. If nothing is logged, then **No entry** will be displayed.



Temp Sensor	Over Temp	Under Temp	Diode Fault	Temp (°C)	MTP Fault Log (°C)	Open Circuit	Short to GND	Short to VCC	Short to DXP
Local	●	●	●	22.3125	No entry				
Remote 1	●	●	●	21.8125	No entry	●	●	●	●
Remote 2	●	●	●	20.6875	No entry	●	●	●	●
Remote 3	●	●	●	19.8125	No entry	●	●	●	●
Remote 4	●	●	●	14.8125	No entry	●	●	●	●

Highest Temperature (°C) 22.0625

Alarm 1 Output ●

Alarm 2 Output ●

☑ Auto Scale

Max Temperature 140.0

Min Temperature -60.0

Sample History 16

☑ Local
☑ Remote 1
☑ Remote 2
☑ Remote 3
☑ Remote 4
☑ Highest

Status Log (Auto polling must be turned off)

```
2:22:11.929 PM - Data 0x00 was read from address 0x38
2:22:11.929 PM - Data 0x00 was read from address 0x47
2:22:11.929 PM - Data 0x00 was read from address 0x48
2:22:11.929 PM - Data 0x00 was read from address 0x49
2:22:11.929 PM - Data 0x00 was read from address 0x4A
2:22:31.680 PM - Auto Polling is On - Period: 166ms
```

☐ Enable Logging

Manufacturer ID: 79

Revision Code: 1

Connected

Figure 2. Status Tab

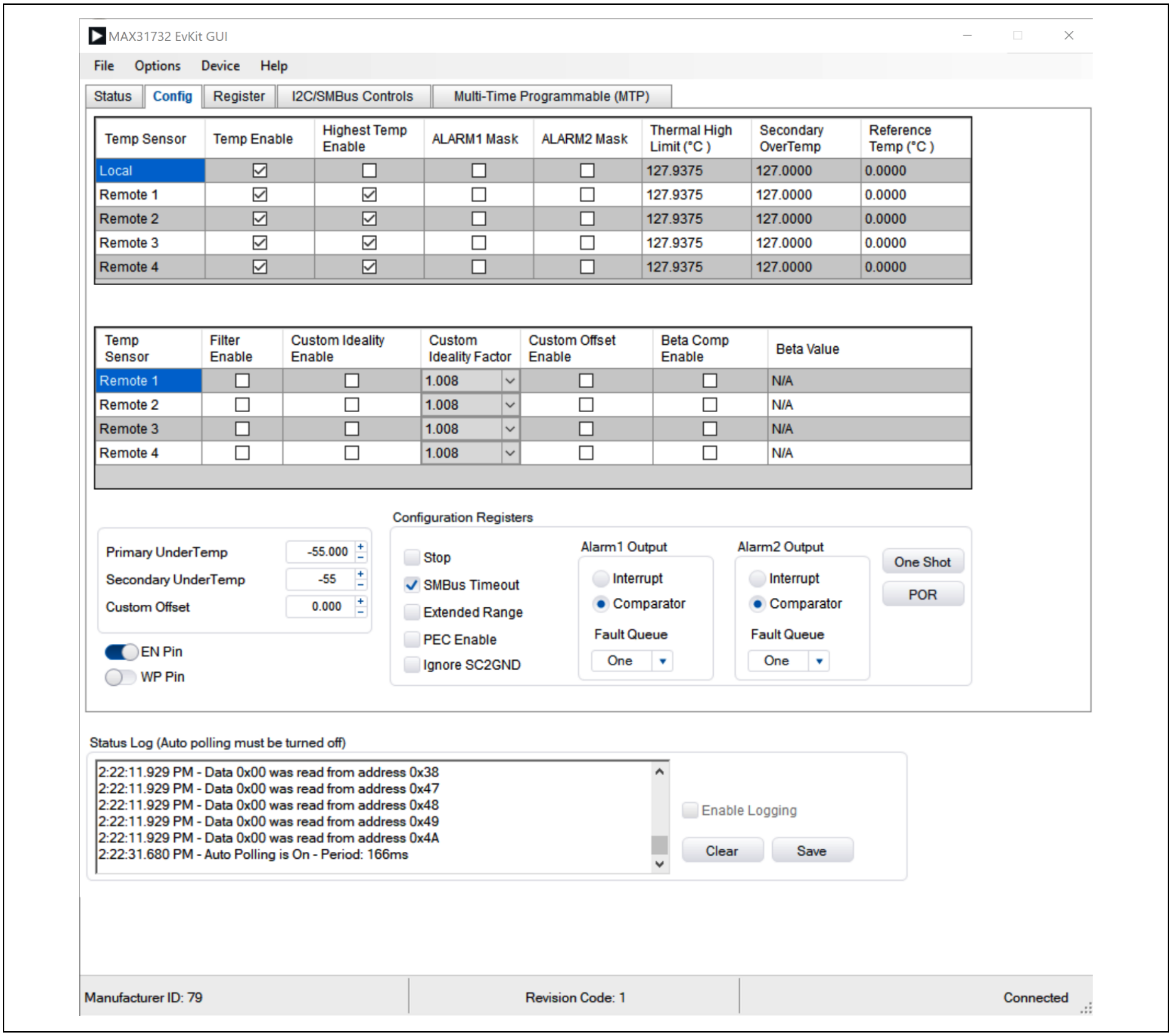
Config Tab

The **Config** tab (*Figure 3*) displays all the current values for the configuration resistors and allows the user to change nearly every configurable setting in the MAX31732. Configuration settings related to the **MTP Fault Log** are located on the **Multi-Time Programmable** (last tab). Check boxes are used for individual bit settings, a pull-down for the Ideality settings, and a floating-point entry for any temperature limits. **Beta Value** is a read-only value and is only shown if **Beta Compensation** is enabled. If the MAX31732 EVKIT, EN, and WP configuration jumpers (J7/J8) are set to allow the GUI to control them (WP and EN), the **EN Pin** and **WP Pin** sliders on the lower left can be used to drive those MAX31732 input pins.

If auto-polling is turned off on the **Status** tab, then the actual I2C commands sent to the MAX31732 over the I2C bus (due to interacting with the controls on this tab) are shown in the **Status Log** at the bottom of the screen.

PEC Enable is not allowed to be written to the MTP using this GUI.

Refer to the *MAX31732* datasheet for more details.



Temp Sensor	Temp Enable	Highest Temp Enable	ALARM1 Mask	ALARM2 Mask	Thermal High Limit (°C)	Secondary OverTemp	Reference Temp (°C)
Local	☑	☐	☐	☐	127.9375	127.0000	0.0000
Remote 1	☑	☑	☐	☐	127.9375	127.0000	0.0000
Remote 2	☑	☑	☐	☐	127.9375	127.0000	0.0000
Remote 3	☑	☑	☐	☐	127.9375	127.0000	0.0000
Remote 4	☑	☑	☐	☐	127.9375	127.0000	0.0000

Temp Sensor	Filter Enable	Custom Ideality Enable	Custom Ideality Factor	Custom Offset Enable	Beta Comp Enable	Beta Value
Remote 1	☐	☐	1.008	☐	☐	N/A
Remote 2	☐	☐	1.008	☐	☐	N/A
Remote 3	☐	☐	1.008	☐	☐	N/A
Remote 4	☐	☐	1.008	☐	☐	N/A

Primary UnderTemp -55.000
Secondary UnderTemp -55
Custom Offset 0.000

EN Pin (on)
WP Pin (off)

Configuration Registers

☐ Stop
☑ SMBus Timeout
☐ Extended Range
☐ PEC Enable
☐ Ignore SC2GND

Alarm1 Output
○ Interrupt
● Comparator
Fault Queue One

Alarm2 Output
○ Interrupt
● Comparator
Fault Queue One

Status Log (Auto polling must be turned off)

```
2:22:11.929 PM - Data 0x00 was read from address 0x38
2:22:11.929 PM - Data 0x00 was read from address 0x47
2:22:11.929 PM - Data 0x00 was read from address 0x48
2:22:11.929 PM - Data 0x00 was read from address 0x49
2:22:11.929 PM - Data 0x00 was read from address 0x4A
2:22:31.680 PM - Auto Polling is On - Period: 166ms
```

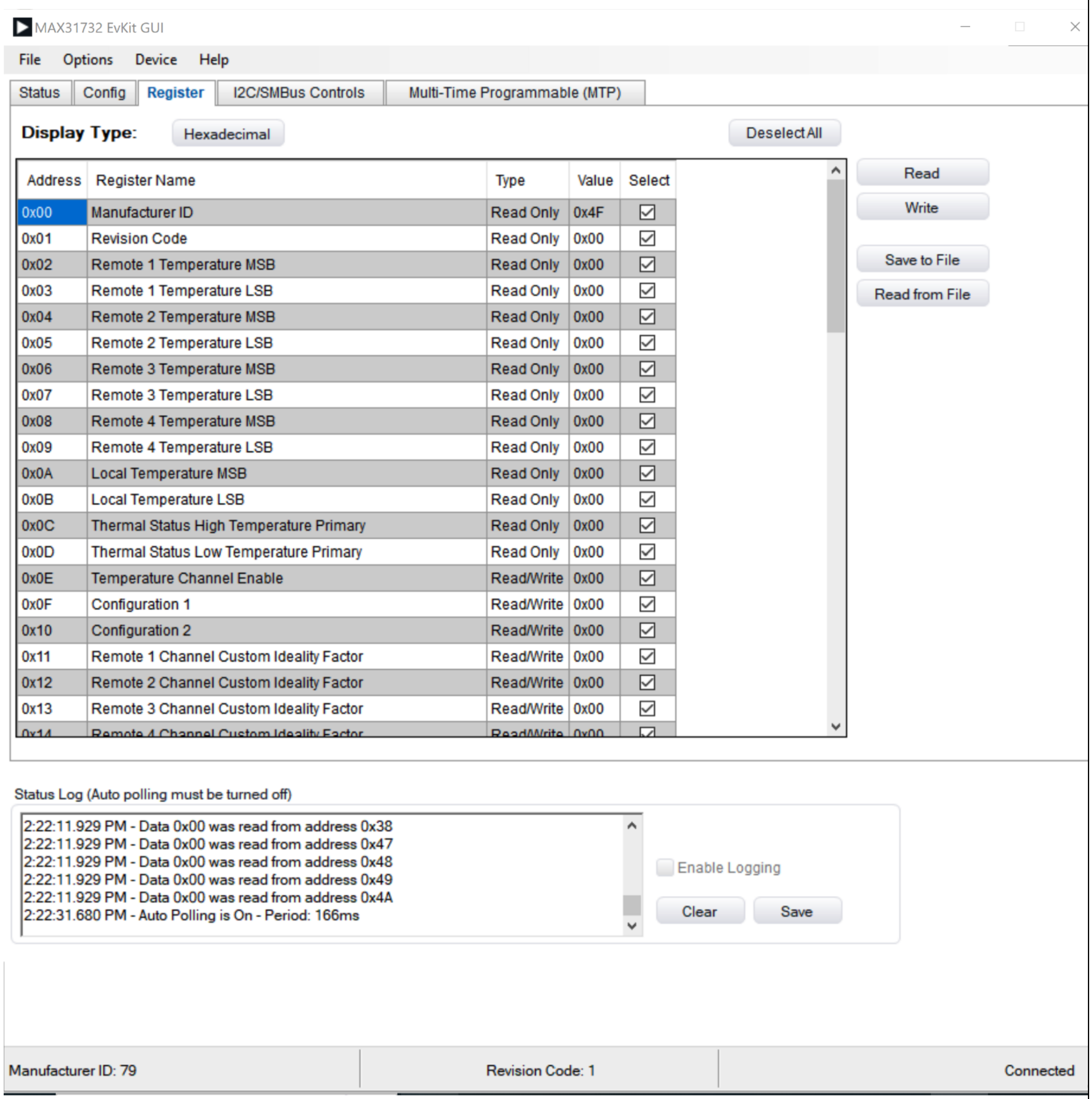
☐ Enable Logging

Manufacturer ID: 79 | Revision Code: 1 | Connected

Figure 3. Config Tab

Register Tab

The Register tab (*Figure 4*) displays the status of all 74 addresses, 0x00 - 0x4A, register names, and their current data in HEX or Decimal. To read register values, use the **Deselect All** or **Sellect All**, as desired, or manually use the **Select** column to individually select RAM addresses, and click on Read button. The user can similarly select those RAM addresses which are writable, edit the **Value** column as desired, and click the **Write** button. The register map can simply be saved to or read from a .CSV Excel file by clicking on the **Save To File** or **Read From File** button.



Display Type: Hexadecimal

Address	Register Name	Type	Value	Select
0x00	Manufacturer ID	Read Only	0x4F	☑
0x01	Revision Code	Read Only	0x00	☑
0x02	Remote 1 Temperature MSB	Read Only	0x00	☑
0x03	Remote 1 Temperature LSB	Read Only	0x00	☑
0x04	Remote 2 Temperature MSB	Read Only	0x00	☑
0x05	Remote 2 Temperature LSB	Read Only	0x00	☑
0x06	Remote 3 Temperature MSB	Read Only	0x00	☑
0x07	Remote 3 Temperature LSB	Read Only	0x00	☑
0x08	Remote 4 Temperature MSB	Read Only	0x00	☑
0x09	Remote 4 Temperature LSB	Read Only	0x00	☑
0x0A	Local Temperature MSB	Read Only	0x00	☑
0x0B	Local Temperature LSB	Read Only	0x00	☑
0x0C	Thermal Status High Temperature Primary	Read Only	0x00	☑
0x0D	Thermal Status Low Temperature Primary	Read Only	0x00	☑
0x0E	Temperature Channel Enable	Read/Write	0x00	☑
0x0F	Configuration 1	Read/Write	0x00	☑
0x10	Configuration 2	Read/Write	0x00	☑
0x11	Remote 1 Channel Custom Ideality Factor	Read/Write	0x00	☑
0x12	Remote 2 Channel Custom Ideality Factor	Read/Write	0x00	☑
0x13	Remote 3 Channel Custom Ideality Factor	Read/Write	0x00	☑
0x14	Remote 4 Channel Custom Ideality Factor	Read/Write	0x00	☑

Status Log (Auto polling must be turned off)

```
2:22:11.929 PM - Data 0x00 was read from address 0x38
2:22:11.929 PM - Data 0x00 was read from address 0x47
2:22:11.929 PM - Data 0x00 was read from address 0x48
2:22:11.929 PM - Data 0x00 was read from address 0x49
2:22:11.929 PM - Data 0x00 was read from address 0x4A
2:22:31.680 PM - Auto Polling is On - Period: 166ms
```

□ Enable Logging

Manufacturer ID: 79

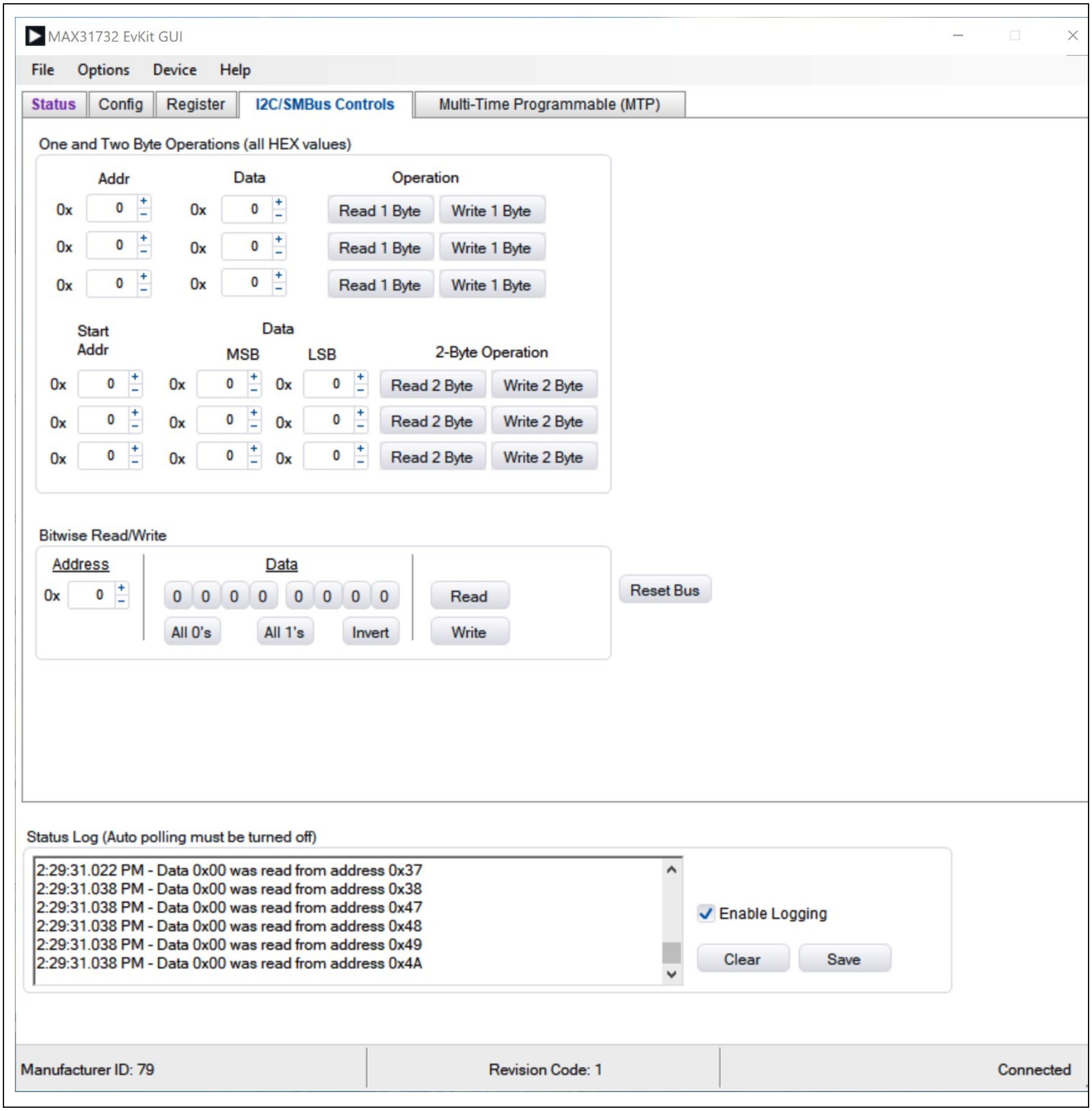
Revision Code: 1

Connected

Figure 4. Register Tab

I2C/SMBus Control Tab

The I2C/SMBus Control Tab (*Figure 5*) allows the user to read and write the registers using HEX values. The **One or Two-Bytes operations** (in HEX) group box contains controls to read/write 1 or 2 Bytes at a time. To read or write a register, enter the desired register address in the **Addr** or **Start Addr** edit box and press the **Read or Write button**. The **Bitwise Read/Write** group box allows the user to read/write the data in binary format. To read or write, enter the register address in the **Address** edit box and press the **Read** or **Write** button. The data bits can be flipped by pressing the bit buttons. The **All 0's, All 1's**, and **Invert** buttons are useful shortcuts to change all the bits at one time.



```
2:29:31.022 PM - Data 0x00 was read from address 0x37
2:29:31.038 PM - Data 0x00 was read from address 0x38
2:29:31.038 PM - Data 0x00 was read from address 0x47
2:29:31.038 PM - Data 0x00 was read from address 0x48
2:29:31.038 PM - Data 0x00 was read from address 0x49
2:29:31.038 PM - Data 0x00 was read from address 0x4A
```

☑ Enable Logging

Manufacturer ID: 79 | Revision Code: 1 | Connected

Figure 5. I^{2}C Control Tab

Multi-Time Programmable (MTP) Tab

The **MTP** tab (*Figure 6*) allows the user to interact with the MTP memory, addresses 0x80 to 0xB9. It shows the register names, and when read, their values are displayed. The user can read all the registers or **Deselect All**, select the desired register, and then click on **Read**. The user can set any of the MTP values except for the MTP fault logging region, 0x82–0x8D.

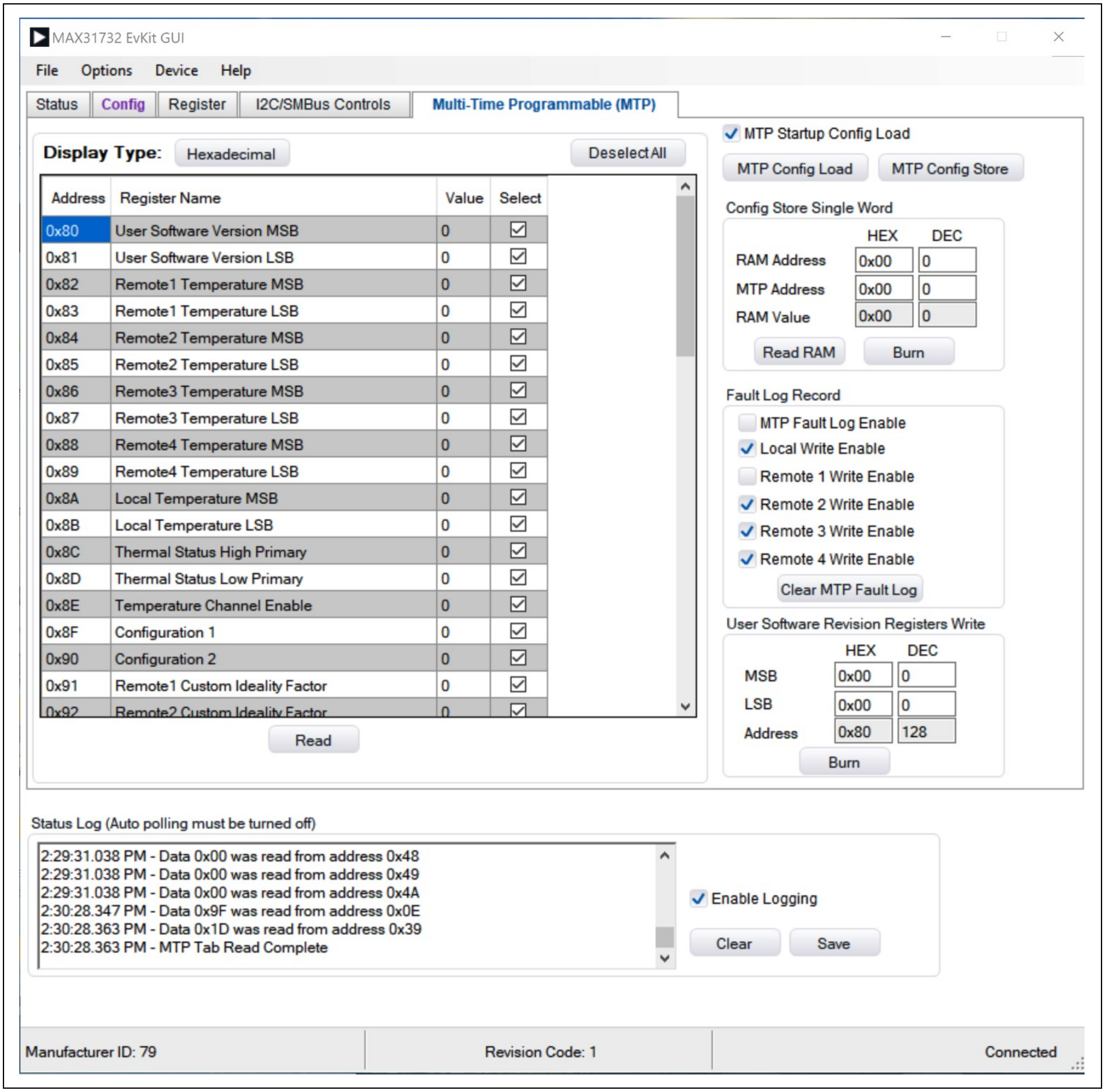
When the chip is powers up, the MPT configuration register values get copied to the RAM memory. Using the **MTP Config Load** button, the user can cause all MTP configuration register values to be loaded into the RAM. Using the **MTP Config Store** button, the user can write the entire range of configuration registers in RAM to be stored in the MTP. Note that only configurable values have an equivalent space in the MTP, i.e., not temperature status, or command bits.

The **Config Store Single Word** tool allows any configuration register to be written to the MTP individually, though at two bytes at a time. To use this tool, the user must change the register in RAM first, then use this tool to write the value from RAM to MTP.

The **User Software Revision Registers Write** tool can be used to write a user software revision code or any other value into MTP address 0x80-0x81.

Using the **Fault Log Record** section, the user can set up which channels to write into the fault logging section of the MTP in the event of an $\overline{\text{ALARM1}}$ condition. The **MTP Fault Log Enable** bit is the controller enable for this function. The user can also clear the fault log using the **Clear MTP Fault Log** button.

Assuming that auto-polling is off in the **Status** Tab, what was sent to the MAX31732 can be seen over the I^{2}C bus in the **Status Log** at the bottom of the screen.



Display Type: Hexadecimal

Address	Register Name	Value	Select
0x80	User Software Version MSB	0	☑
0x81	User Software Version LSB	0	☑
0x82	Remote1 Temperature MSB	0	☑
0x83	Remote1 Temperature LSB	0	☑
0x84	Remote2 Temperature MSB	0	☑
0x85	Remote2 Temperature LSB	0	☑
0x86	Remote3 Temperature MSB	0	☑
0x87	Remote3 Temperature LSB	0	☑
0x88	Remote4 Temperature MSB	0	☑
0x89	Remote4 Temperature LSB	0	☑
0x8A	Local Temperature MSB	0	☑
0x8B	Local Temperature LSB	0	☑
0x8C	Thermal Status High Primary	0	☑
0x8D	Thermal Status Low Primary	0	☑
0x8E	Temperature Channel Enable	0	☑
0x8F	Configuration 1	0	☑
0x90	Configuration 2	0	☑
0x91	Remote1 Custom Ideality Factor	0	☑
0x92	Remote2 Custom Ideality Factor	0	☑

☑ MTP Startup Config Load

Config Store Single Word

	HEX	DEC
RAM Address	0x00	0
MTP Address	0x00	0
RAM Value	0x00	0

Fault Log Record

- ☐ MTP Fault Log Enable
- ☑ Local Write Enable
- ☐ Remote 1 Write Enable
- ☑ Remote 2 Write Enable
- ☑ Remote 3 Write Enable
- ☑ Remote 4 Write Enable

User Software Revision Registers Write

	HEX	DEC
MSB	0x00	0
LSB	0x00	0
Address	0x80	128

Status Log (Auto polling must be turned off)

```
2:29:31.038 PM - Data 0x00 was read from address 0x48
2:29:31.038 PM - Data 0x00 was read from address 0x49
2:29:31.038 PM - Data 0x00 was read from address 0x4A
2:30:28.347 PM - Data 0x9F was read from address 0x0E
2:30:28.363 PM - Data 0x1D was read from address 0x39
2:30:28.363 PM - MTP Tab Read Complete
```

☑ Enable Logging

Manufacturer ID: 79 | Revision Code: 1 | Connected

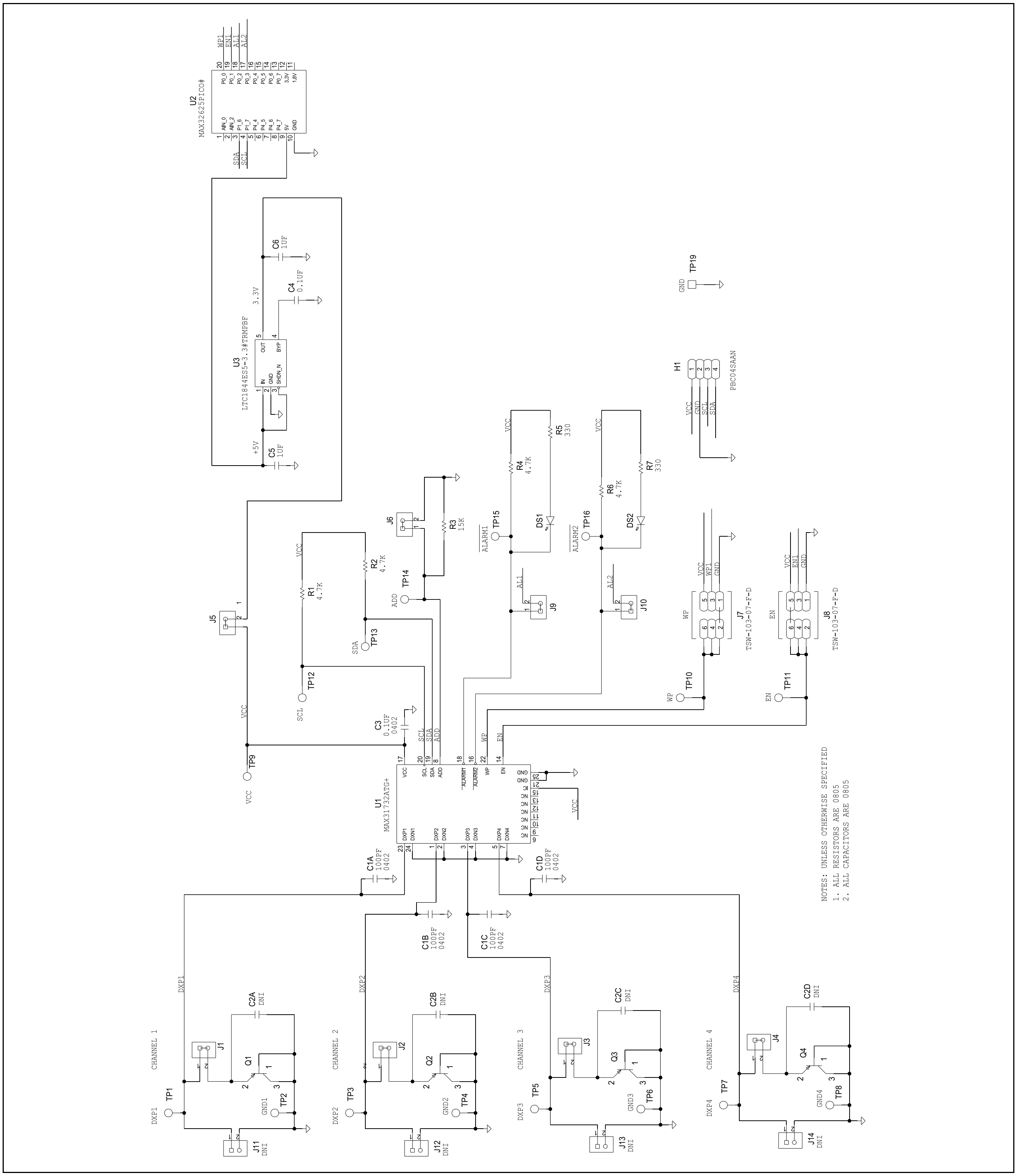
Figure 6. MTP Tab

Ordering Information

PART	TYPE
MAX31372EVKIT#	EV Kit

#Denotes RoHS-compliant.

MAX31732EVKIT# Schematic Diagram



NOTES: UNLESS OTHERWISE SPECIFIED
1. ALL RESISTORS ARE 0805
2. ALL CAPACITORS ARE 0805

Notes

ASSUMED BY ANALOG DEVICES FOR ITS USE, NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THIRD PARTIES THAT MAY RESULT FROM ITS USE. SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE. NO LICENCE, EITHER EXPRESSED OR IMPLIED, IS GRANTED UNDER ANY ADI PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR ANY OTHER ADI INTELLECTUAL PROPERTY RIGHT RELATING TO ANY COMBINATION, MACHINE, OR PROCESS WHICH ADI PRODUCTS ALL INFORMATION CONTAINED HEREIN IS PROVIDED “AS IS” WITHOUT REPRESENTATION OR WARRANTY. NO RESPONSIBILITY IS OR SERVICES ARE USED. TRADEMARKS AND REGISTERED TRADEMARKS ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS.