AC487
Application Note
RTG4 FPGA: Temperature Monitor using LM99 Temperature Sensor

Microsemi
a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

51900487. 2.0 9/19

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the current publication.

1.1 Revision 2.0

Temperature sensor part number was corrected from TMP464 to TMP461-SP.

1.2 Revision 1.0

The first publication of this document.

2 Temperature Monitor Using LM99 Temperature Sensor

Microsemi RTG4™ radiation tolerant flash-based FPGAs offer significant performance and flexibility advantages to designers working on satellite systems and other equipment required to perform in high radiation environments. RTG4 employs several radiation hardening by design techniques, which help to achieve a high-level of total ionizing dose (TID) hardness. RTG4 also provides built-in Single Event Transient (SET) and Single Event Upset (SEU) mitigation in critical areas to enable the creation of high reliability designs.

This application note shows how the user can configure an external temperature sensor, to perform RTG4 junction temperature measurements across the internal temp diode, and read the sensor status signals using FPGA fabric logic.

In this application note, Texas instruments LM99 temperature sensor, which is available on the *RTG4 development Kit* is interfaced with RTG4 FPGA using I2C protocol. LM99 temperature sensor registers are configured through the I2C interface and RTG4 device core temperature is measured. This reference design does not use any soft processor IP. The RTL logic implements the configuration of LM99 temperature sensor and temperature monitoring.

The LM99 is an 11-bit remote diode temperature sensor with a 2-wire system management bus (SMBus) serial interface. The LM99 accurately measures its own temperature and the temperature of a remote diode-connected transistor such as the 2N3904 or a thermal diode commonly found on graphics processor units (GPU), computer processor units (CPU or other ASICs). For more information about LM99 temperature sensor, see *LM99* info page.

LM99 provides the following set points and status signals:

LM99 set points are:

- Low set point: low temperature alert threshold, asserts ALERT signal if the RTG4 FPGA Core temperature is below this value.
- High set point: high temperature alert threshold, asserts ALERT signal if the RTG4 FPGA Core temperature is above this value.
- T_CRIT set point: critical temperature alert threshold, asserts T_CRIT and ALERT signals if the RTG4 FPGA Core temperature exceeds this value.

LM99 status signals are:

- ALERT: active-Low output signal used as Controller Interrupt or Alert Line.
- T_CRIT: active-Low output signal can be used for system shutdown and also can be used as a status signal. Assertion of this signal is based on the Critical set point value.

These status signals can be monitored and used in the RTG4 FPGA Application. In this reference design, PLL is reset based on the set values of temperature.

The design can be programmed using any of the following options:

- **Using the job file**: To program the device using the job file provided along with the design files, see Appendix: Programming the Devices Using FlashPro Express, page 13.
- **Using Libero SoC**: To program the device using Libero SoC, see Libero Design Flow, page 6.

2.1 Design Requirements

The following table lists the hardware and software required to measure RTG4 temperature monitoring using LM99.

Table 1 • **Resource Requirements**

Requirement	Version
Host PC Operating system	Windows 7 or 10
Hardware	
RTG4 FPGA Development Kit	Rev B
USB A to Mini-B Cable	-
12V, 5A AC Power Adapter and Cords	-
Software	
Libero SoC Design Suite[1]	v12.1
FlashPro Express	V12.1
RTG4 Temperature Monitor GUI (included along with the design files)	-

1. A Libero Platinum license is required to evaluate the design on the RTG4 device.

2.2 Prerequisites

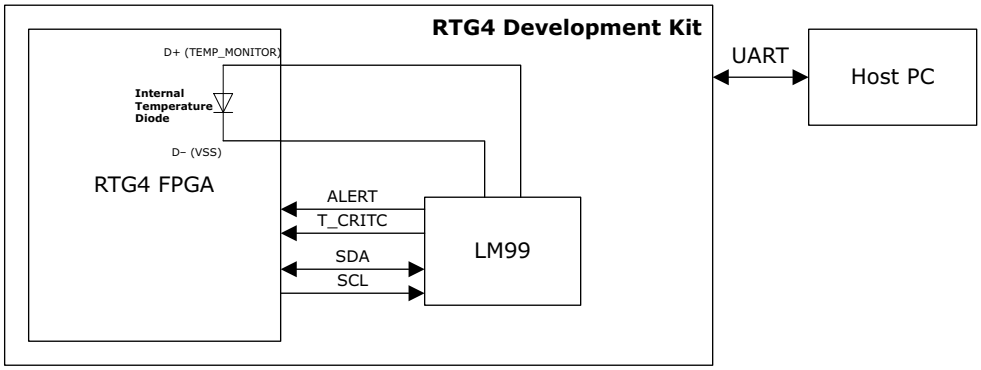
Before you start:

1. For background information about resetting PLL, see *CN19009 RTG4 PLL Lock Stability.*
2. Download the temperature rise window calculator from the following location to arrive at the temperature windows that is required as input for this reference design. *https://www.microsemi.com/document-portal/doc_download/1244318-rtg4-pll-calculator-files-zip*
3. Download the design files from the following location: *https://soc.microsemi.com/download/rsc/?f=rtg4_ac487_liberosocv12p1_df*
4. Download and install Libero SoC v12.1 on the host PC from the following location. *https://www.microsemi.com/product-directory/design-resources/1750-libero-soc#downloads*. The latest versions of ModelSim and Synplify Pro are included in the Libero SoC installation package.

2.3 Demo Design

The following figure shows the RTG4 Temperature Monitor block diagram.

Figure 1 • **Block Diagram**

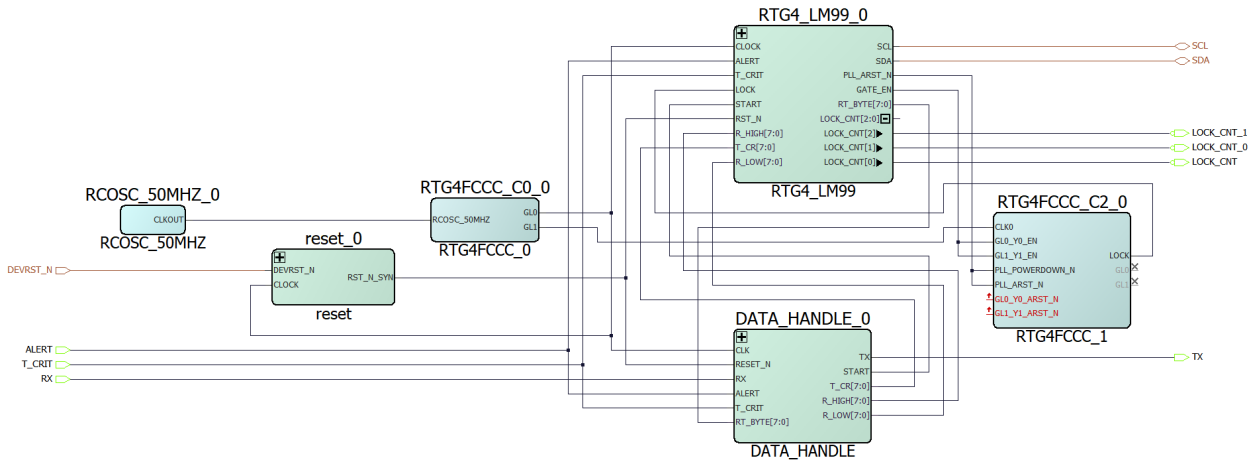


RTG4 FPGA is interfaced with an external LM99 temperature sensor through I2C protocol for monitoring the temperature of the RTG4 Core. Temperature diode points (D+ and D-) and status signals of LM99 are connected to RTG4 FPGA.

2.4 Design Implementation

The following figure shows the top-level Libero design of RTG4-LM99 temperature measurement.

Figure 2 • **Top-Level Libero Design**



2.4.1 RTG4_LM99

This block receives set points from the GUI and configures the LM99 register using I2C. The temperature of RTG4 device core is read from LM99 through I2C and displayed in the GUI. This block also includes the logic to reset the PLL (RTG4FCCC_C2_0) based on the ALERT and T_CRIT status signals of LM99. These signals assert depending on the temperature set point provided by the user. It also resets the PLL if the temperature variation is more than 60 °C. For more information about PLL lock stability, see *CN19009 RTG4 PLL Lock Stability*. The RTL logic in this module can read and write the LM99 register using COREI2C, which is controlled by I2C master RTL logic.

2.4.2 DATA_HANDLE

This block handles data flow between FPGA and GUI Interface for LM99 set point transfers, temperature data and status signals transfers. A COREUART IP configured for 115200 baud rate and a UART interface fsm is used to communicate from FPGA to the GUI in the host PC.

2.4.3 RTG4FCCC_C0_0

This block uses GPDs to get 10 MHz and 25 MHz clock generated from 50 MHz RC oscillator without using PLL.

2.4.4 RTG4FCCC_C2_0

This block uses PLL to generate 100 MHz from 25 MHz clock. The lock signal of this block is monitored by the RTG4_LM99.

2.4.5 Input and Output Signals

The following table lists the input and output signals of the design.

Table 2 • **Input and Output Signals**

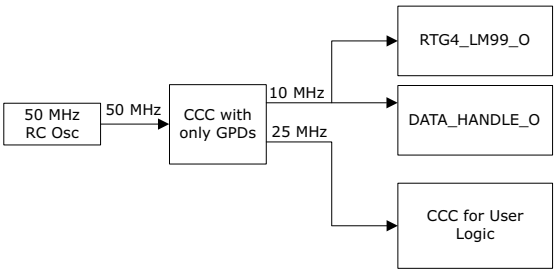
Signals	Direction	Description
DEVRST_N	Input	Device reset
ALERT	Input	Active low status signal coming from LM99. Assertion is based on High and Low set points of LM99.
T_CRIT	Input	Active low status signal coming from LM99. Assertion is based on T_CRIT set point of LM99.
RX	Input	UART RX pin.
SCL	Output	I2C clock pin connected to LM99.
SDA	Output	I2C data pin connected to LM99.
TX	Output	UART TX pin
LOCK_CNT, LOCK_CNT_0, LOCK_CNT_1	Output	Lock signal change counter. Based on deassertion of Lock signal, this counter increments.

2.4.6 Clocking Structure

In this reference design, there are two clock-domains. From 50 MHz RC oscillator, 10 MHz and 25 MHz are generated using RTG4 Clock Conditioning Circuit (CCC) General Purpose Dividers (GPDs) only.

10 MHz is used for GUI interface and LM99 communication. 25 MHz is fed to a CCC PLL that generates 100 MHz clock for user logic. All the blocks (RTG4_LM99_0 and DATA_HANDLE_0) are using synchronous reset.

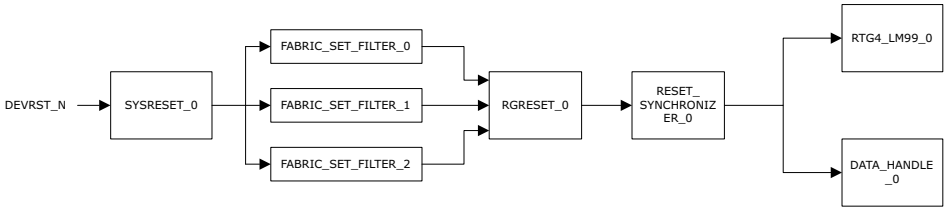
Figure 3 • **Clocking Structure**



2.4.7 Reset Structure

In this reference design, there is only one reset, which is derived form SYSRESET macro which is then synchronized with reset synchronizer block and fed to other blocks.

Figure 4 • **Reset Structure**



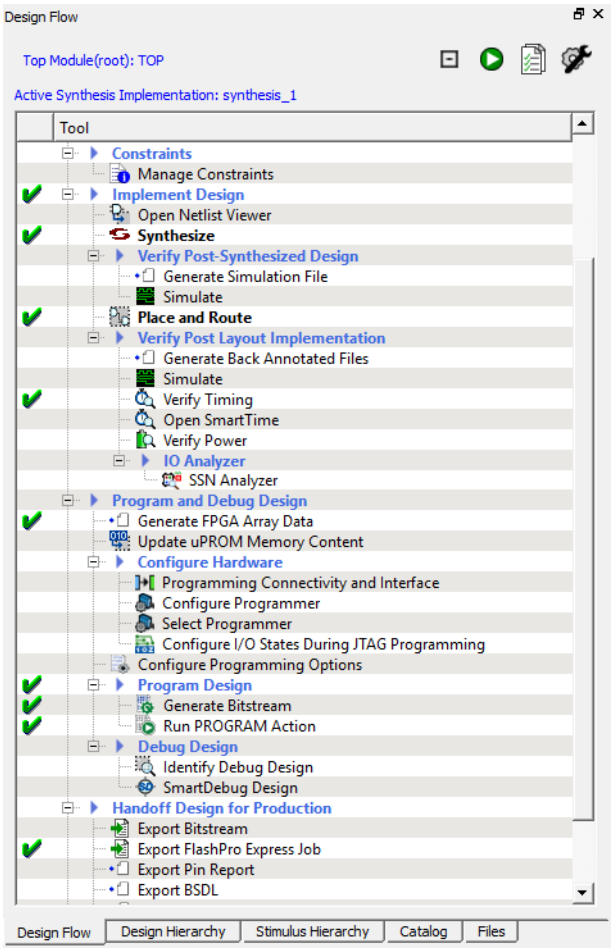
3 Libero Design Flow

This chapter describes the Libero design flow of the demo design. The Libero design flow involves the following steps:

- Synthesize
- Place and route
- Verify Timing
- Generate Bitstream
- Run PROGRAM Action

The following figure shows these options in the Design Flow tab.

Figure 5 • **Libero Design Flow Options**



3.1 Synthesize

To synthesize the design:

1. From the **Design Flow** window, double-click **Synthesize**.

When the synthesis is successful, a green tick mark appears as shown in Figure 5, page 6.

2. Right-click **Synthesize** and select **View Report** to view the synthesis report and log files in the Reports tab.

3.2 Place and Route

From the **Design Flow** window, double-click **Place and Route**. When place and route is successful, a green tick mark appears as shown in Figure 5, page 6.

Right-click **Place and Route** and select **View Report** to view the place and route report and log files in the Reports tab.

3.2.1 Resource Utilization

The following tables list the resource utilization of this design after place and route. These values may vary slightly for different Libero runs, settings, and seed values.

Table 3 • **Resource Utilization**

Type	Used	Total	Percentage
4LUT	954	151824	0.63
DFF	378	151824	0.25
I/O Register	0	2154	0.00
Logic Element	984	151824	0.65

3.3 Verify Timing

To verify timing:

1. From the **Design Flow** window, double-click **Verify Timing**.
2. Right-click Verify Timing and enable all the four corners for timing analysis and constraints coverage report.
3. When the design successfully meets the timing requirements, a green tick mark appears as shown in Figure 5, page 6.
4. Right-click **Verify Timing** and select **View Report** to view the verify timing report and log files in the Reports tab.

3.4 Generate FPGA Array Data

To generate the FPGA array data:

1. Double-click **Generate FPGA Array Data** from the **Design Flow** window.
2. A green tick mark is displayed after the successful generation of the FPGA array data as shown in Figure 5, page 6.

3.5 Generate Bitstream

To generate the bitstream:

1. Double-click **Generate Bitstream** from the **Design Flow** tab.
   When the bitstream is successfully generated, a green tick mark appears as shown in Figure 5, page 6.
2. Right-click **Generate Bitstream** and select **View Report** to view the corresponding log file in the **Reports** tab.

3.6 Run PROGRAM Action

After generating the bitstream, the RTG4 device must be programmed. Follow these steps to program the RTG4 device:

1. Ensure that the following jumper settings are set on the board.

Table 4 • **Jumper Settings**

Jumper	Pin From	Pin To	Comments
J11, J17, J19, J23, J26, J21, J32, and J27	1	2	Default
J16	2	3	Default
J33	1 3	2 4	Default

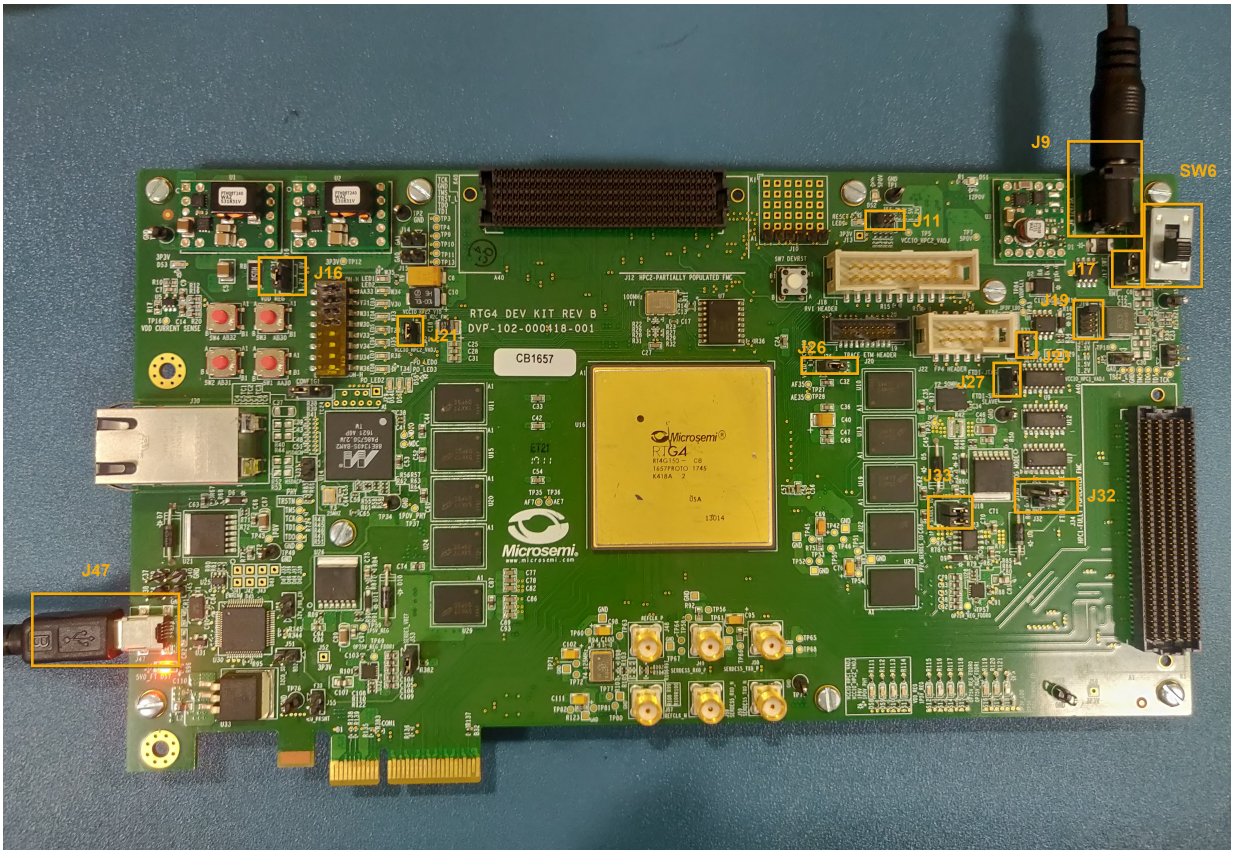
Note: Ensure that power supply switch **SW6** is switched OFF while connecting jumpers on RTG4 development kit.

2. Connect the host PC to the **J47** connector using the USB cable.
3. Connect the power supply to **J9** connector and switch ON the power supply switch, **SW6**.
4. Double-click **Run PROGRAM Action** from the **Libero > Design Flow** tab.

Note: You can also program the device using FlashPro Express. See Appendix: Programming the Devices Using FlashPro Express, page 13.

When the device is programmed successfully, a green tick mark appears as shown Figure 5, page 6. The device is successfully programmed, see Running the Demo, page 9.

Figure 6 • **Board Setup**



4 Running the Demo

This chapter describes how to install and use the GUI to run the temperature monitor demo.

4.1 Installing the GUI

To install the GUI:

1. Extract the contents of the `rtg4_ac487_liberosocv12p1_df.rar` file. From the `rtg4_ac487_liberosocv12p1_df\GUI` folder, double-click the `setup.exe` file.
2. Follow the instructions displayed on the installation wizard.

After successful installation, GUI appears on the Start menu of the host PC desktop.

4.2 Running the Demo

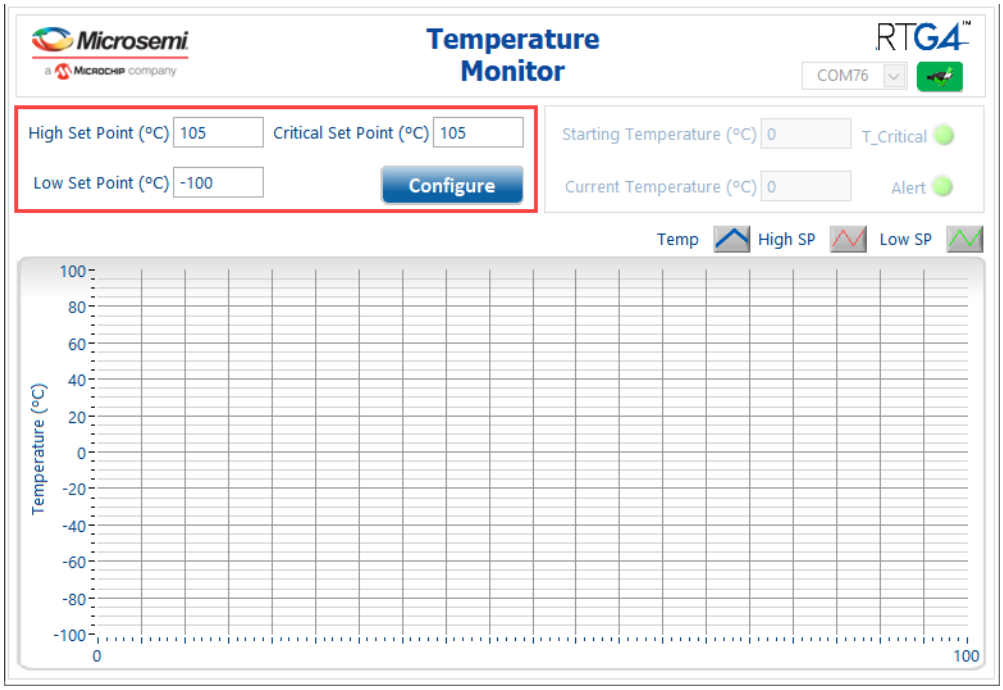
Before you start, ensure that the RTG4 FPGA board is programmed.

1. Open the GUI. It automatically connects to the COM port. If the GUI does not connect automatically, reset the device using switch, **SW7**.
2. Enter the set point values in the GUI.
   Critical set-point values can be obtained from RTG4 PLL Temperature Rise Window calculator. Enter Critical set-point as "Maximum operating junction temp before PLL unlock" from the PLL Temperature Rise Window calculator, as shown in the following figure. Enter Low Set-point and High Set-Point as user defined values based on requirement.

Figure 7 • **PLL Rise Window Calculator Output**

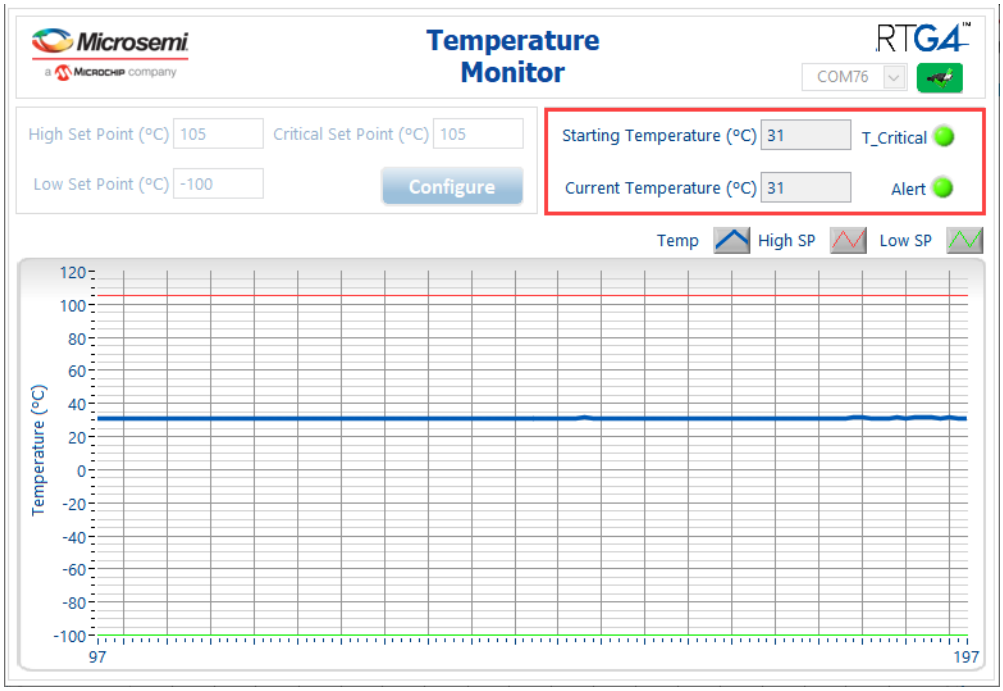
Instructions	Input values in all blue cells Calculated in white cells Results in green cells	
PLL Parameters	***Value***	***Unit***
Reference Clock frequency	25	MHz
Input divider	1	Integer
Lockwindow selection	6000	ppm
VCO frequency	800	MHz
Reset/Power-on VDDPLL	3.3	V
Operating VDDPLL	3.3	V
Reset/Power-on junction temperature	25	°C
PFD frequency (calculated)	25.000	MHz
VDDPLL delta (calculated)	0	mV
Calculated Results		
Minimum temp rise window (Constant VDDPLL voltage)	80	Δ°C
Maximum operating junction temp before PLL unlock (Constant VDDPLL voltag	105	°C
Minimum temp rise window (with VDDPLL Voltage change)	80	Δ°C
Maximum operating junction temp before PLL unlock (with VDDPLL voltage ch	105	°C

Figure 8 • **Entering Set Point Values**



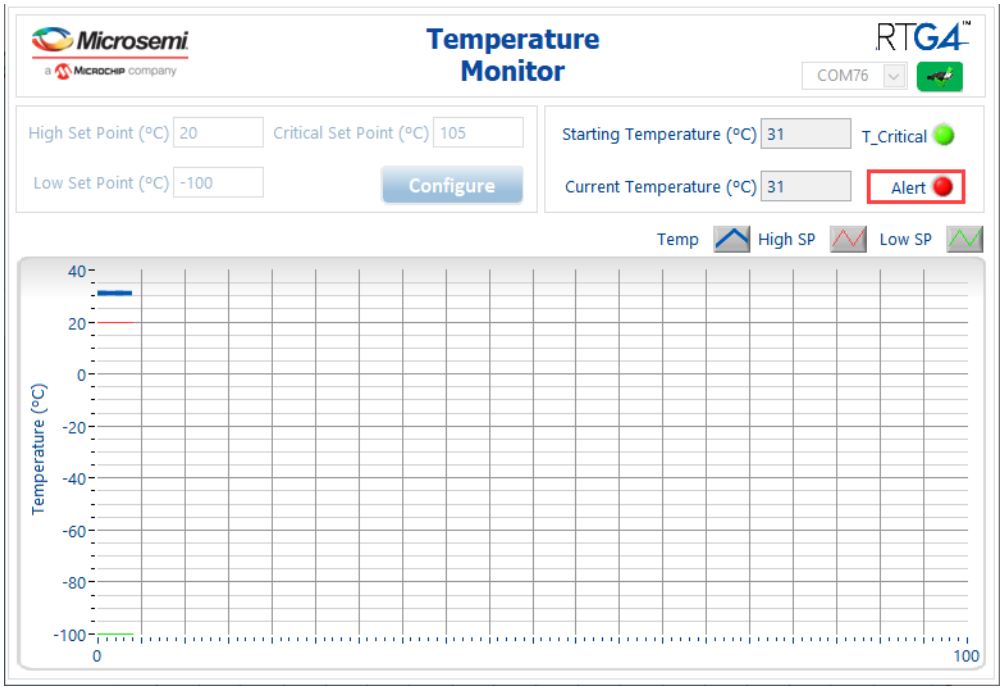
3. Click **Configure**. RTG4 Core temperature and status signals are shown in the following figure.

Figure 9 • **Configuring Core Temperature and Status Signals**



ALERT signal shows red if the core temperature exceeds low or high Set point value or if it exceeds Critical set points value. T_CRIT signal show Reds if the core temperature exceeds Critical set points value. Based on assertion of ALERT or T_CRIT a PLL reset will occur. See the following figure.

Figure 10 • **Alert Assertion**



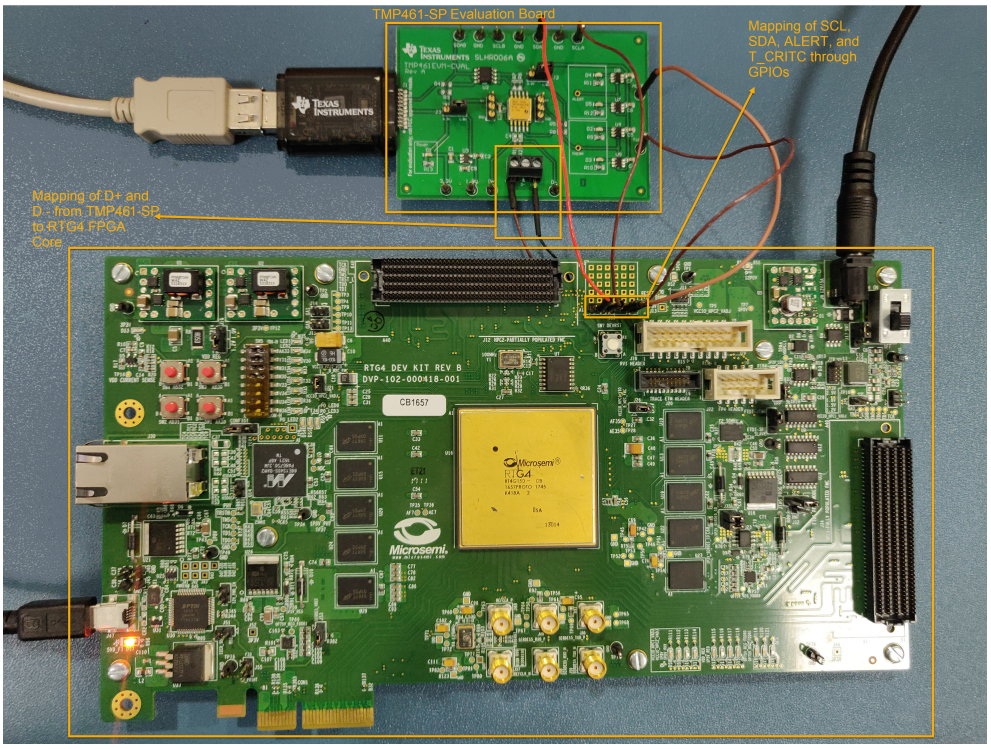
5 Appendix: TMP461-SP Temperature Sensor

TMP461-SP is Radiation tolerant version of the temperature sensor intended for Space applications. TMP461-SP also provides better accuracy. To replace LM99 with TMP461-SP on the RTG4 Dev kit the following re-work is required:

- Hardware changes:
  - Remove LM99 device from the board
  - Remap D+ and D- pins from RTG4 FPGA core to TMP461-SP D+ and D-
- Design changes:
  - Map I2C SDA and SCL pins to any GPIO and connect to TMP461-SP
  - Map the ALERT and TCRIT to any GPIO and connect to TMP461-SP

After performing the above modifications, the reference design provided in this Application Note has also been updated and validated using TMP461-SP mounted on the RTG4 development kit as shown in the following figure.

Figure 11 • **RTG4 FPGA Board Interface with TMP461-SP Evaluation Board**



For more information about TMP461-SP radiation tolerant temperature sensor, see *http://www.ti.com/product/TMP461-SP*.

6 Appendix: Programming the Devices Using FlashPro Express

This section describes how to program the RTG4 device with the .job programming file using FlashPro Express. The .job file is available at the following design files folder location:

```
rtg4_ac487_liberosocv12p1_df\Programming_Job_File
```

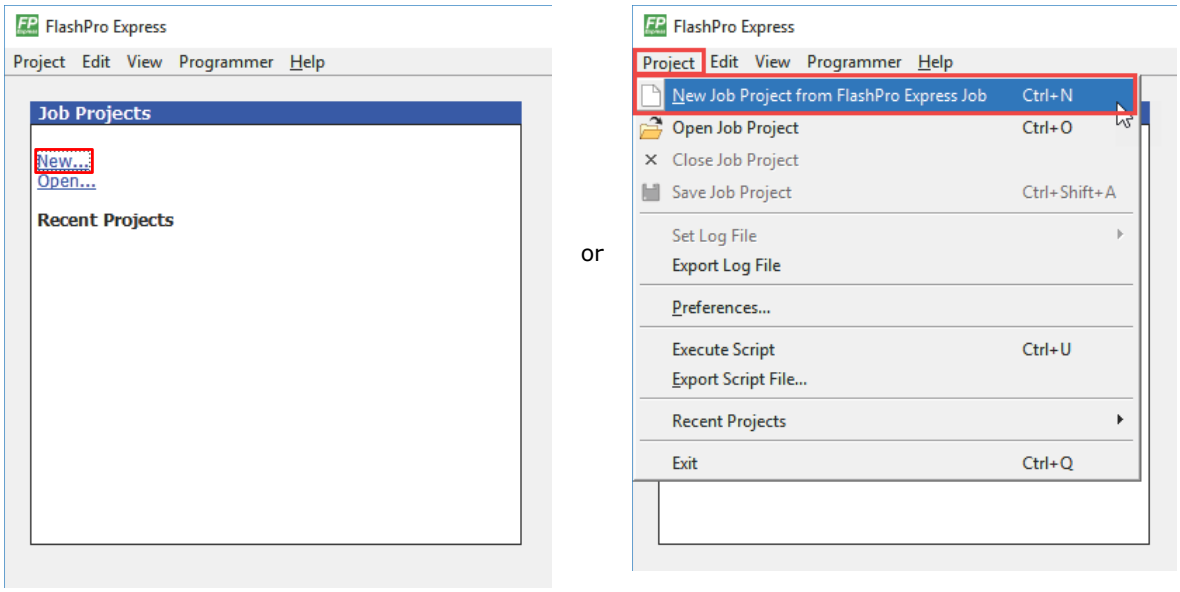
To program the device, complete the following steps:

1. Ensure that the jumper settings on the board are the same as those listed in Table 4, page 8.

Note: The power supply switch must be switched off while making the jumper connections.

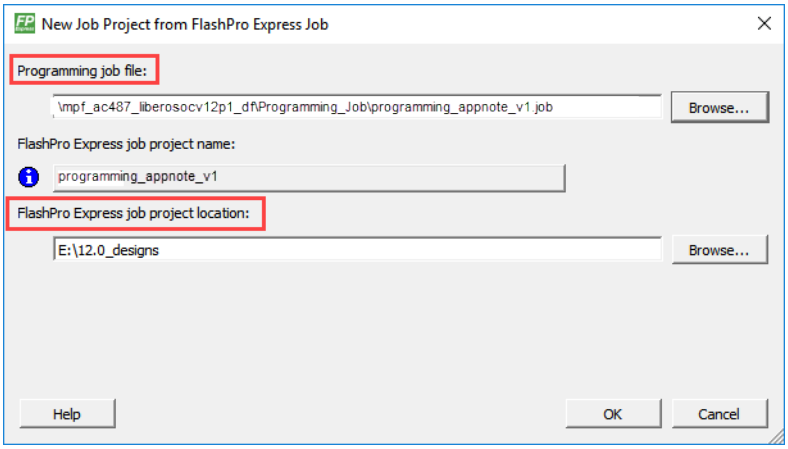
2. Connect the power supply cable to the **J9** connector.
3. Connect the USB cable from the host PC to the **J47** (FTDI port).
4. Power ON the board using the **SW6** slide switch.
5. On the host PC, launch the **FlashPro Express** software.
6. Click **New** or select **New Job Project from FlashPro Express Job** from **Project** menu to create a new job project, as shown in the following figure.

Figure 12 • **FlashPro Express Job Project**



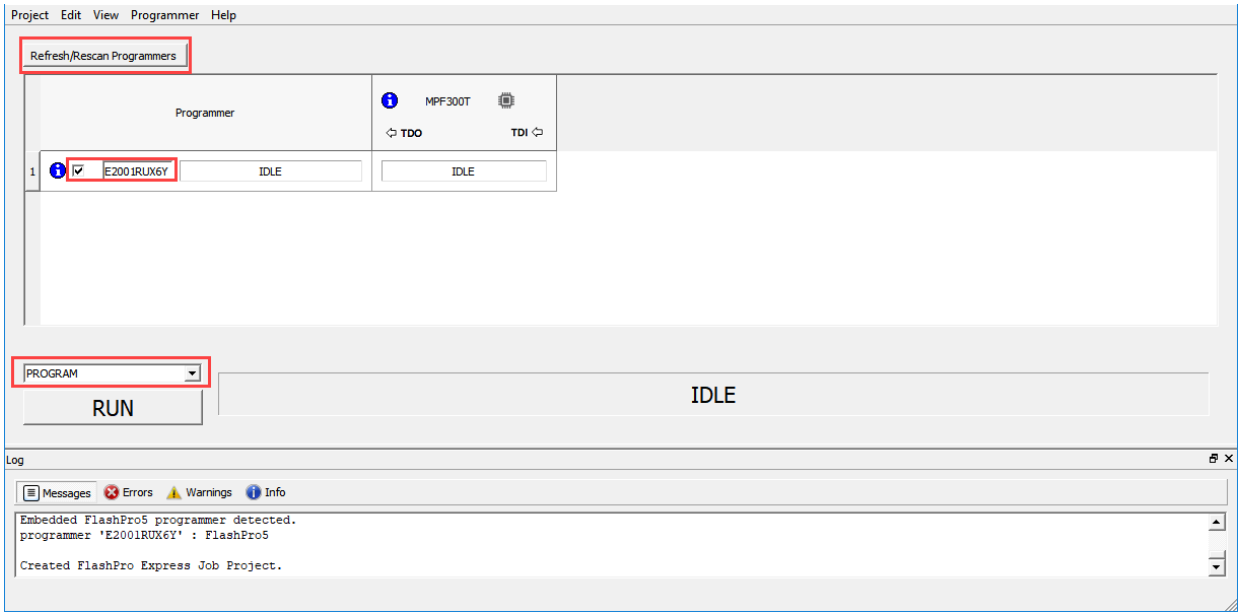
7. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:
   - **Programming job file**: Click **Browse**, and navigate to the location where the .job file is located and select the file. The default location is: *<download_folder>\rtg4_ac487_iberosocv12p1_df\Programming_Job.*
   - **FlashPro Express job project location**: Click **Browse** and navigate to the location where you want to save the project.

Figure 13 • **New Job Project from FlashPro Express Job**



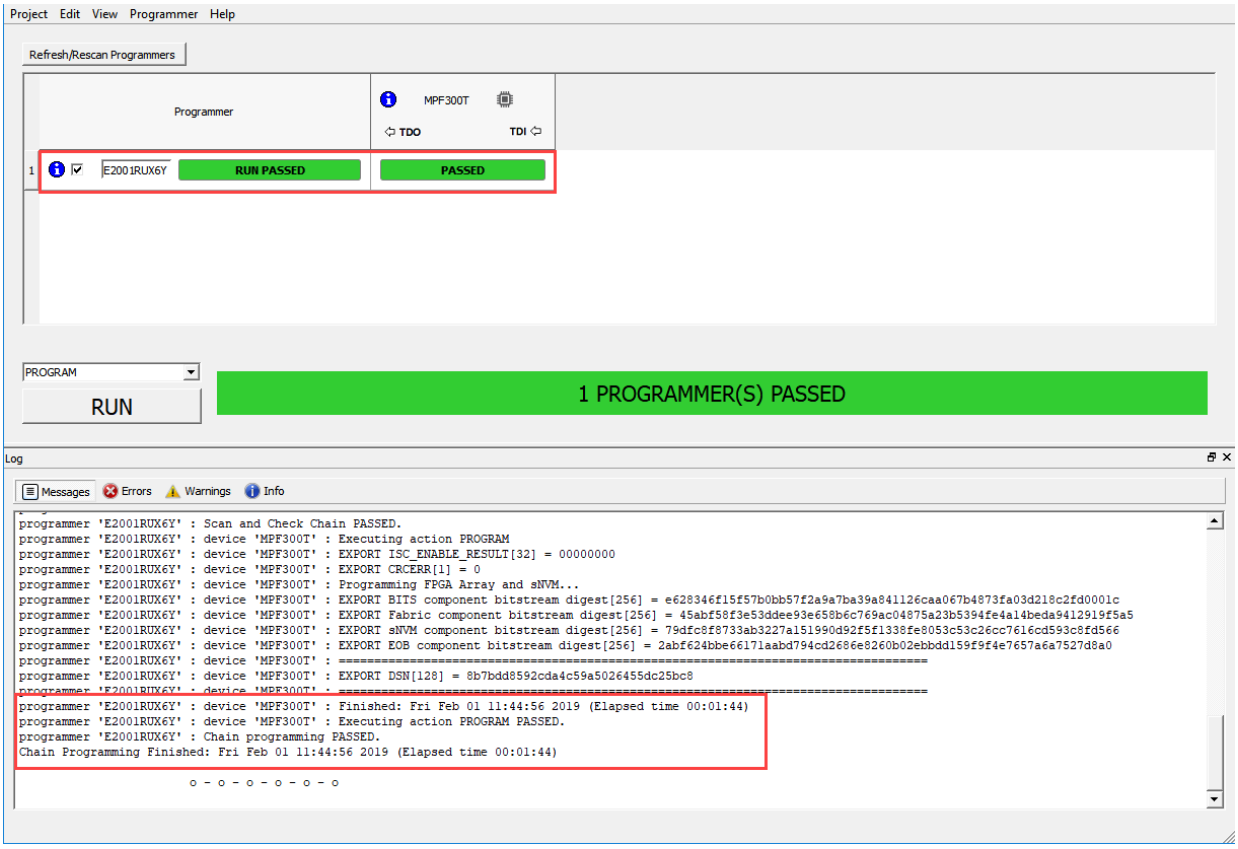
8. Click **OK**. The required programming file is selected and ready to be programmed in the device.
9. The FlashPro Express window appears as shown in the following figure. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/Rescan Programmers**.

Figure 14 • **Programming the Device**



10. Click **RUN** to program the device. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure. To run the demo, see Running the Demo, page 9 section.

Figure 15 • **FlashPro Express—RUN PASSED**



11. Close **FlashPro Express** (**Project** > **Exit**).

7 Appendix: Calibrating LM99 Remote Temperate Data Inaccuracy Due to Remote Diode Non-Ideality

In order to calibrate the errors introduced by non-ideality of remote temperature diode, the temperature sensor can be calibrated by changing offset registers of LM99. User can change the LM99 Remote Temperature Offset High Byte (RTOHB-Address: 0x11) register based on the offset required. Power-on default value of RTOHB is 0. The offset value must be entered in Two's complement format.

For more information, see *LM99 Datasheet*.

8 Appendix: References

This section lists documents that provide more information related to LM99 temperature sensor.

- For more information about Texas instruments LM99 temperature sensor, see *RTG4 development Kit.*
- For more information about LM99 temperature sensor, see *LM99 Product* page.
- For more information about TMP461-SP radiation tolerant temperature sensor, see *TMP461-SP Product* page.