Application Report
SLDA052–November 2018

PGA450-Q1 software development guide

Bala Ravi

ABSTRACT

The Texas Instruments PGA450-Q1 evaluation module (EVM) allows users to evaluate the operation and performance of the PGA450-Q1 fully integrated system-on-chip analog front-end for ultrasonic sensing. This application report introduces basic device settings to start capturing return echoes within minutes of unboxing the EVM. The EVM is equivalent to that of an end-product form factor, allowing users to easily integrate PGA450-Q1-based sensor module into a real system without the need for additional prototype. To modify the source code that comes with firmware installer, Keil C51 Development tool can be used. Since the GUI source code is not available to the public, use the PGA450 Energia library and code examples for master controller implementation. The PGA450Q1EVM-S is programmed to operate from OTP memory, this application report shows how you can customize the firmware for different commands. It includes a software development guide and examples for UART and LIN demonstration. The system requirements are:

- Evaluation Module
- TI-GER Board
- Software (GUI)
- Arm Keil µvision IDE Software
- Windows 7 or later

Contents

List of Figures

List of Tables

Copyright © 2018, Texas Instruments Incorporated

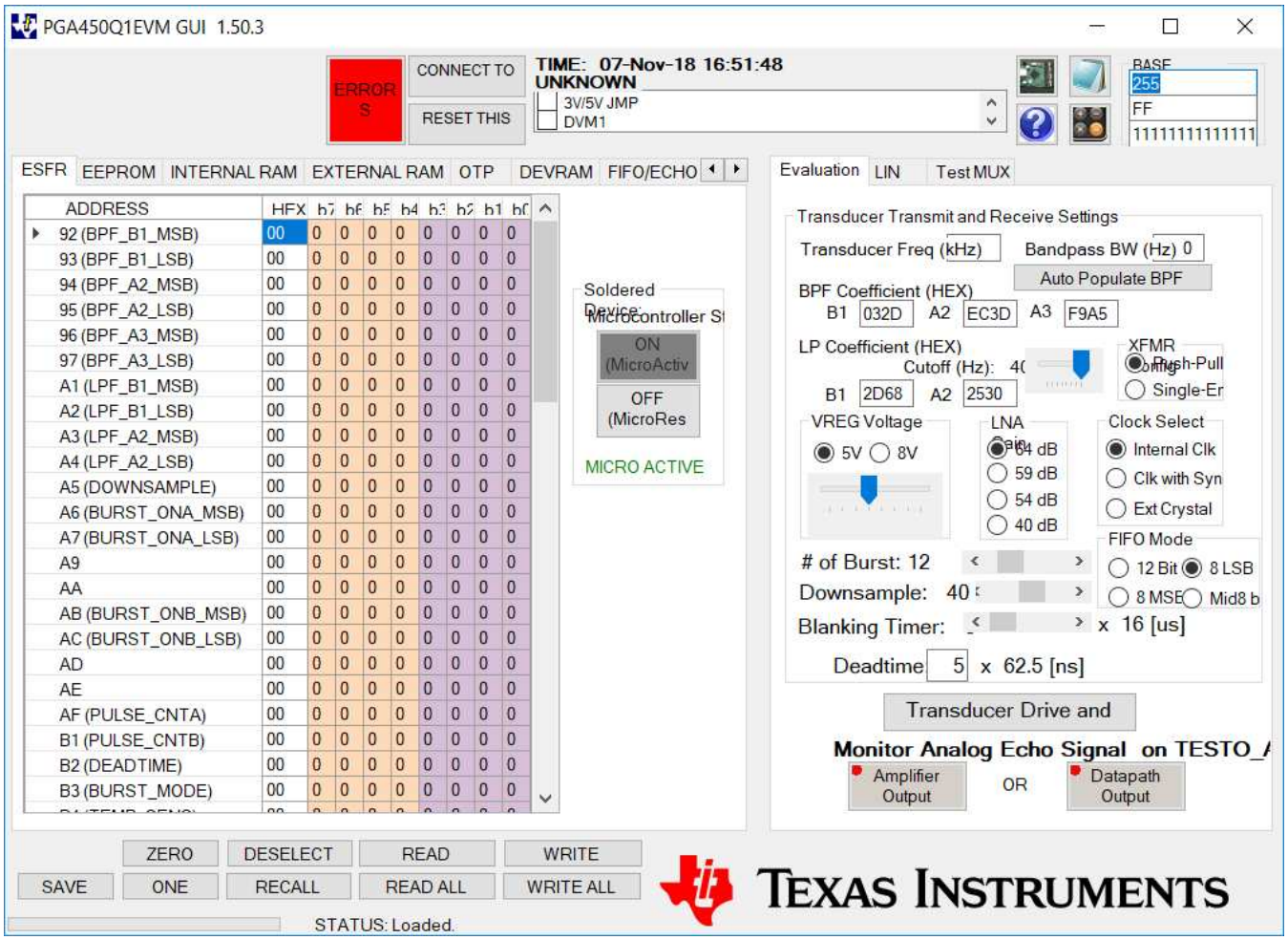
Trademarks

Windows is a registered trademark of others.
All other trademarks are the property of their respective owners.
is a registered trademark of ~others.

1 Resources

Register settings for PGA450Q1 can be configured using the following EVM GUI. The EVM GUI can be downloaded at http://www.ti.com/lit/zip/tidcab3.

Figure 1. Slave Example for PGA450-Q1 Using the EVM GUI



There are seven threshold values. “COMMAND 6” identifies the range across the 768 points of FIFO where the threshold level changes. The level is an 8-bit value from 0-255. For example, “COMMAND 6” of UART reports the threshold values and is decoded as follows:

- "09 FF 20 84 40 74 60 54 80 4C A0 3C E0 34 19 1E"
- Byte1 = x09 = from FIFO 0 to 9, the threshold levelInit is Byte2 (255) (based on EEPROM addr 0x09 as initial ignore count defaulted to 9. Ignore count means threshold level is set to 255)
- Byte2 = xFF = 255d level

Copyright © 2018, Texas Instruments Incorporated

- Byte3 = x20 = from FIFO 10 to 32, the threshold level0 is Byte4 (132)
- Byte4 = x84 = 132d level
- Byte5 = x40 = from FIFO 33 to 64, the threshold level1 is Byte6 (116)
- Byte6 = x74 = 64d level
- Byte7 = x60 = from FIFO 65 to 96, the threshold level2 is Byte8 (84)
- Byte8 = x54 = 84d level
- Byte9 = x80 = from FIFO 97 to 128, the threshold level3 is Byte10 (76)
- Byte10 = x4C = 76d level
- Byte11 = xA0 = from FIFO 129 to 160, the threshold level4 is Byte12 (60)
- Byte12 = x3C = 60d
- Byte13 = xE0 = from FIFO 161 to 224, the threshold level5 and level6 is Byte14 (52) (In this example, level5 and level6 are the same values, lumped into a single return value, not the double interval of 224 - 160 = 64, which is the typical interval of 32.)
- Byte14 = x34 = 52d
- Byte15 = x19 = from FIFO 225 to the end of the FIFO, the threshold level7 is Byte16 (30)
- Byte16 = x1E = 30d

1.1 GUI/Programming Tools

1.1.1 EVM Overview

EVM-S: Small Form-Factor EVM

The PGA450Q1EVM-S is a fully-assembled PCB design for real-world evaluation of the PGA450-Q1 ultrasonic-sensor signal-conditioner device, an ultrasonic transducer, and step-up transformer. This EVM can be integrated with systems for general object detection and distance measurement through air. For example, it can be used in automotive park-assist, level sensing in tanks, collision avoidance for autonomous robotics, and unmanned aerial vehicle landing assist. This small form-factor EVM is intended to act as an ultrasonic module alternative to the full-scale PGA450Q1EVM. TI recommends that you first begin an evaluation with the PGA450Q1EVM before transitioning to an evaluation with the PGA450Q1EVM-S.

TI-GER Board and IDE

To modify the source code made available through the PGA450Q1EVM firmware installer, download the Keil C51 Development Tool for all 8051 devices, which includes the µVision IDE, which is necessary to open and edit the PGA450-Q1 project file. Keil products use a license management system, so without a current license, the product runs as a lite or evaluation edition with a few limitations.

Master Controller Example

See the *PGA450-Q1 EVM user's guide* for a detailed description of the hardware of the EVM and GUI. Since the GUI source code is not available to the public, use the PGA450 Energia Library and code example as an example for master controller implementation and programming guidelines.

Copyright © 2018, Texas Instruments Incorporated

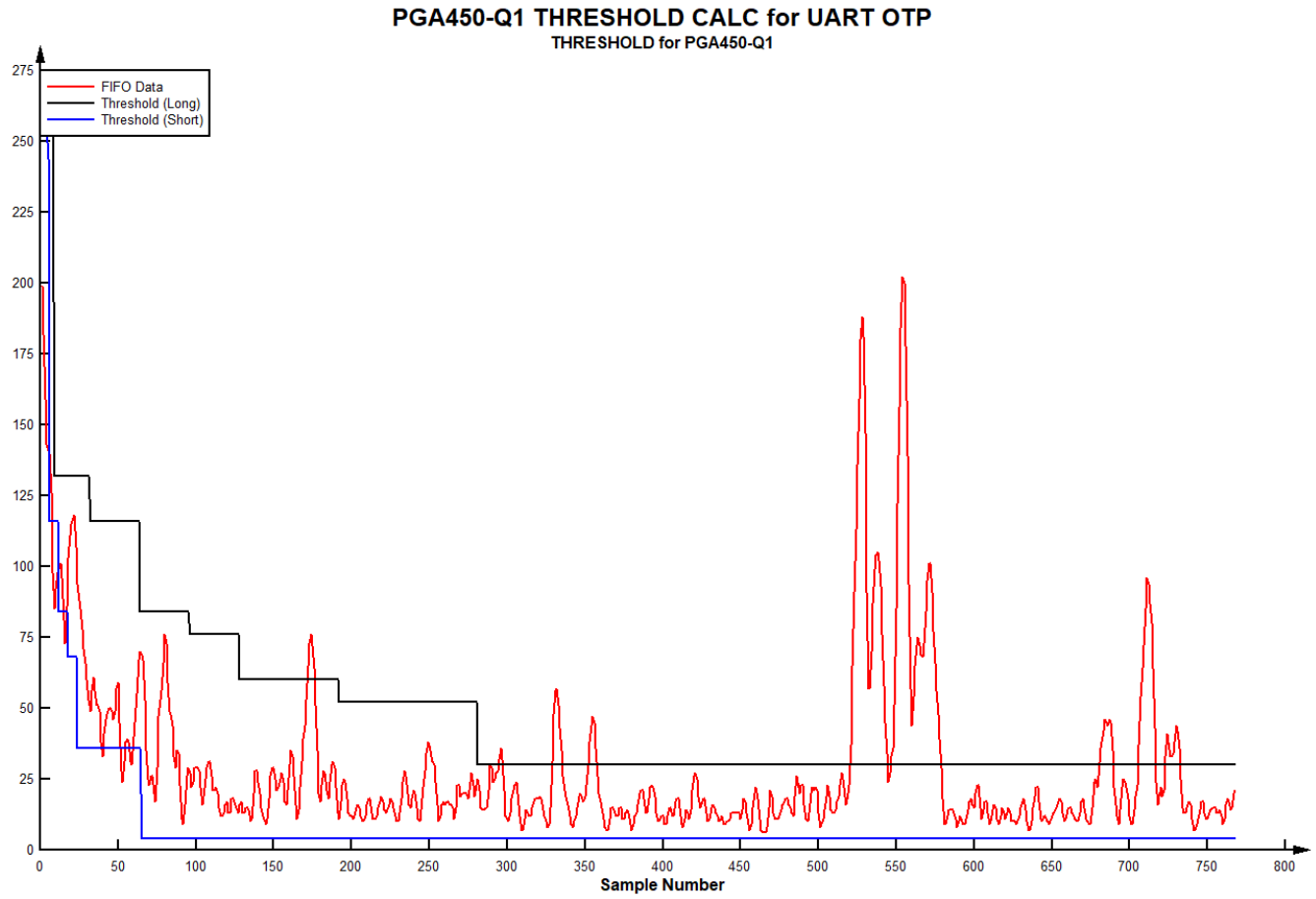
Figure 2. Energia Code for PGA450-Q1 EVM UART Demonstration

Other Tools - Excel

To visualize this process, download the PGA450-Q1 threshold mapper tool from the Ultrasonic FAQ page to see how the EEPROM threshold settings impact FIFO object detection. Figure 3 shows the plot of threshold mapper for PGA450. Access the tool through https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/PGA450_2D00_Q1-THRESHOLD-CALC-for-UART-OTP-EXAMPLE.xlsx.

Copyright © 2018, Texas Instruments Incorporated

Figure 3. Plot of EEPROM to Threshold Mapper for PGA450-Q1



2 Hardware Setup / Connect Procedure

2.1 Programming the PGA450-Q1 DEVRAM or OTP Memory

The PGA450Q1EVM-S memory is programmed to operate from OTP memory, meaning that the PGA450-Q1 device is permanently programmed with a set of predefined commands. If you prefer to customize or modify the firmware for different commands, the PGA450-Q1 device must be replaced with a pristine PGA450-Q1 device. Use the following steps to program the device for DEVRAM or OTP memory.

1. Connect a 12-V system supply voltage to the EVM at J2-1 (MAIN) and connect the SPI pins of the EVM at J3 to the SPI pins on the TI-GER board.

Copyright © 2018, Texas Instruments Incorporated

Figure 4. TI-GER Board SPI and UART Connections

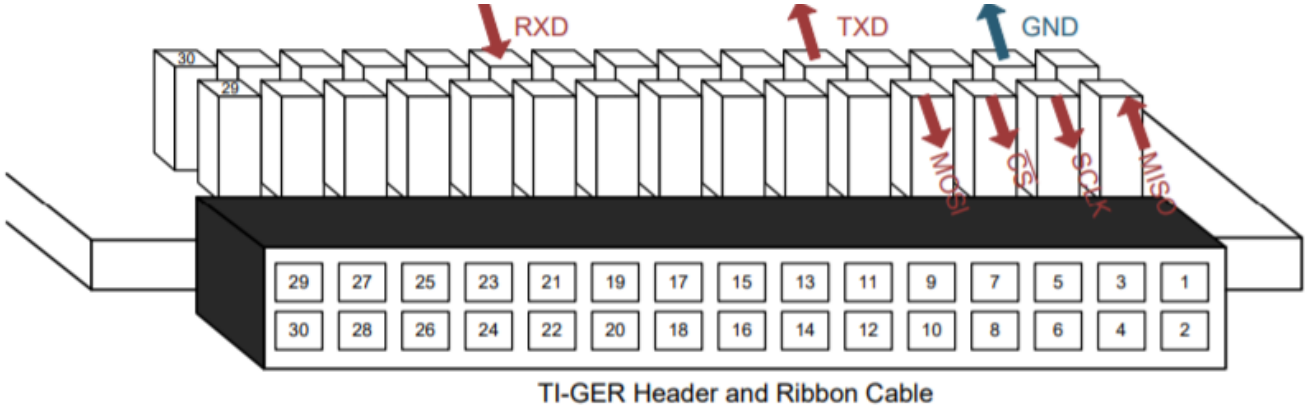


Table 1. TI-GER to PGA450Q1EVM-S Connections Connection

Connection	TI-GER Pin	EVM Pin
SPI-MISO	1	J3-1 (SDO)
SPI-SCLK	3	J3-3 (SCLK)
SPI-CS	5	J3-4 (CS)
SPI-MOSI	7	J3-2 (SDI)
UART-TXD	10	J4-3 (RXD)
UART-RXD	20	J4-4 (TXD)
GND	4	J2-4 (GND)

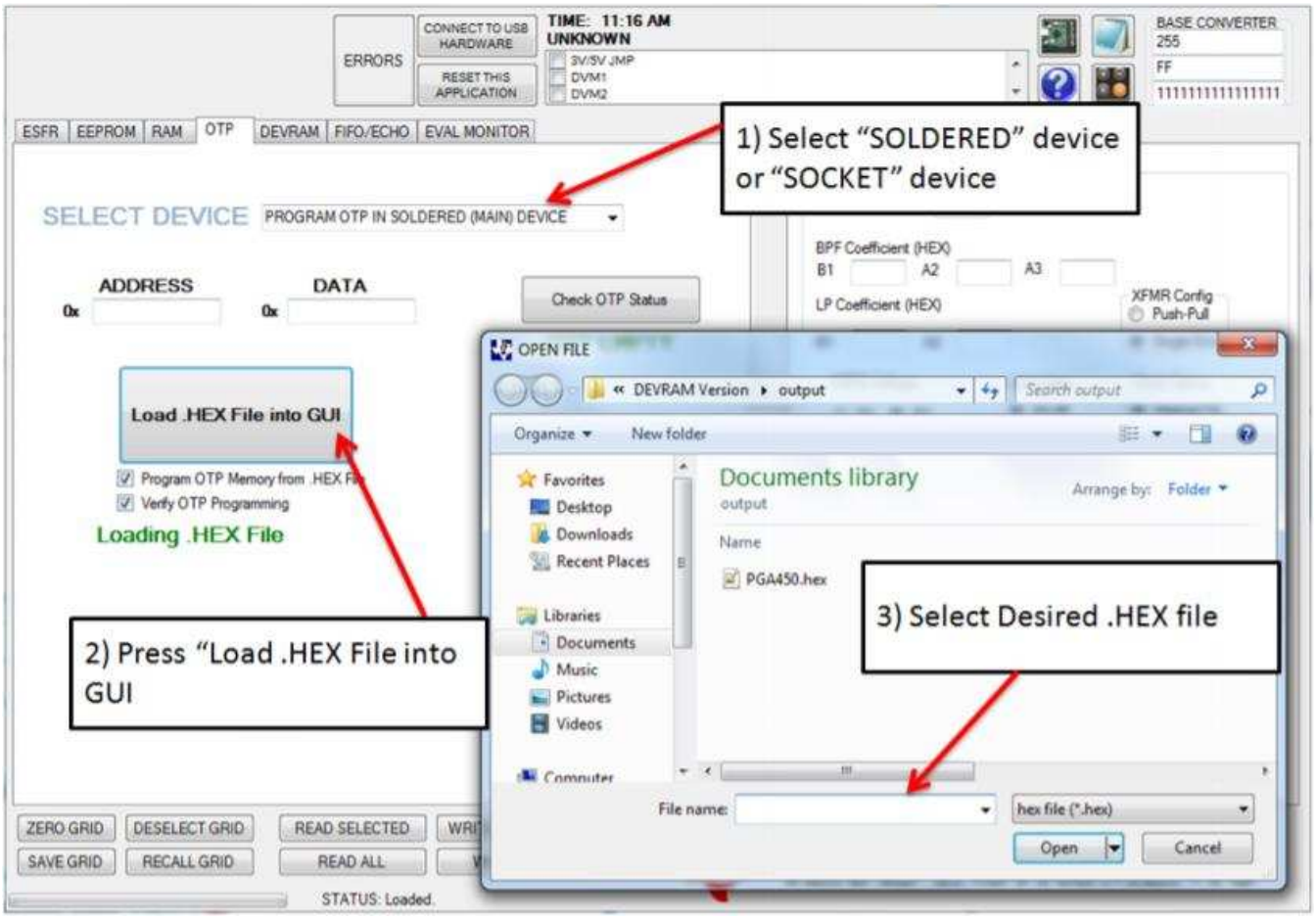
2. Connect the TI-GER board to the PC.
3. Open the PGA450Q1EVM GUI.
4. Click the OFF (Micro Reset) button on the ESFR tab.
5. Click the READ ALL button to read all registers on the ESFR tab. Use register B4 (TEMP_SENS) as an indicator to ensure the device is operating and communicating properly through SPI. Proper communication can be verified if the data of the register reads a value other than 0x00 or 0xFF.
6. Under the OTP tab, click “Check OTP Status” button. If the PGA450-Q1 device has not been previously OTP programmed, the status displays OTP Empty.
7. Connect the 8-V supply to the VPROG_OTP pin on the sensor to program the OTP memory, or to program the DEVRAM memory for the first time, or if the OTP status displays “OTP Empty”.
8. Go to the DEVRAM tab when programming the DEVRAM memory.
9. Check the program OTP memory box in the GUI if you are using a pristine IC that has never been programmed (OTP status displays “OTP Empty”). This option programs both the OTP and DEVRAM memory. The OTP memory is programmed with a long-jump statement to redirect the firmware to load into and be run from the DEVRAM memory.
10. Go to the OTP tab if you are programming OTP memory for production release or a permanently coded PGA450-Q1 device.
11. Click Load .HEX file into GUI, and locate the appropriate OTP-based .HEX file. This is explained visually in the next section. Programming and verification of the device occurs automatically.
12. Disconnect the VPROG_OTP supply voltage (if applicable) when the device is verified. Do not disconnect system supply voltage if DEVRAM has been programmed because the DEVRAM memory clears when the PGA450-Q1 device is power cycled.

Copyright © 2018, Texas Instruments Incorporated

2.2 Programming Tools for DUT

1. Load .HEX file into GUI. The Load .HEX File into GUI button is used to load the contents of a .HEX file into the GUI RAM for use with other operations. When the button is pressed, a second window opens to help you locate and open the desired .HEX file on the PC. See Figure 5 for an example of this operation.

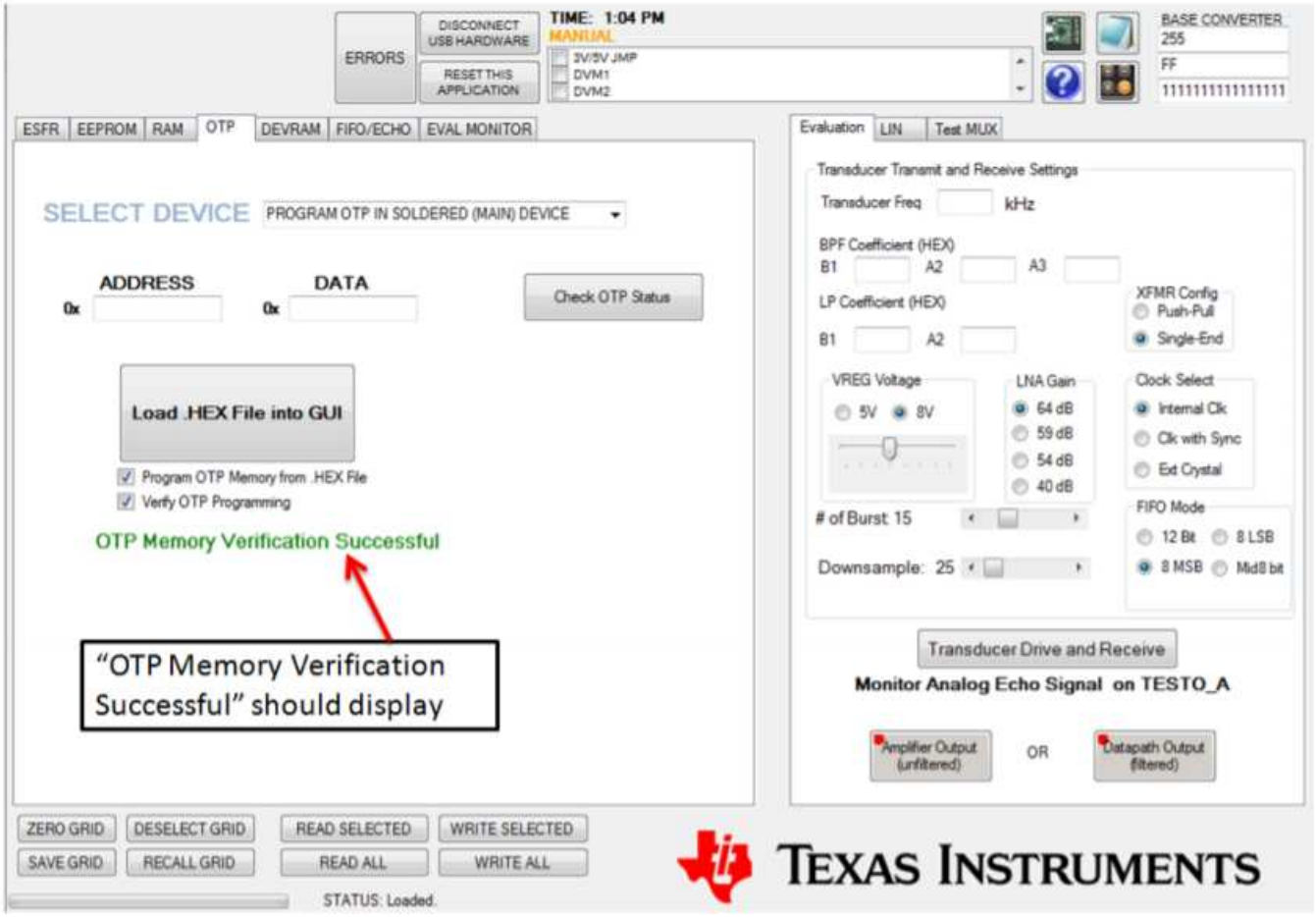
Figure 5. Loading a .HEX File into the GUI



2. Program OTP Memory from .HEX file. If the Program OTP Memory from .HEX File check box is checked (default) when the .HEX file was loaded into the GUI, the OTP memory is programmed with the contents of the .HEX file.
3. Verify OTP Programming. If the Verify OTP Programming button is also checked (default), then after the OTP memory is finished programming, the GUI reads the contents of the OTP memory through SPI and verifies against the .HEX file. Figure 6 shows the message that comes up if the OTP memory matches the contents of the .HEX file.

Copyright © 2018, Texas Instruments Incorporated

Figure 6. OTP Memory Successful Programming Verification

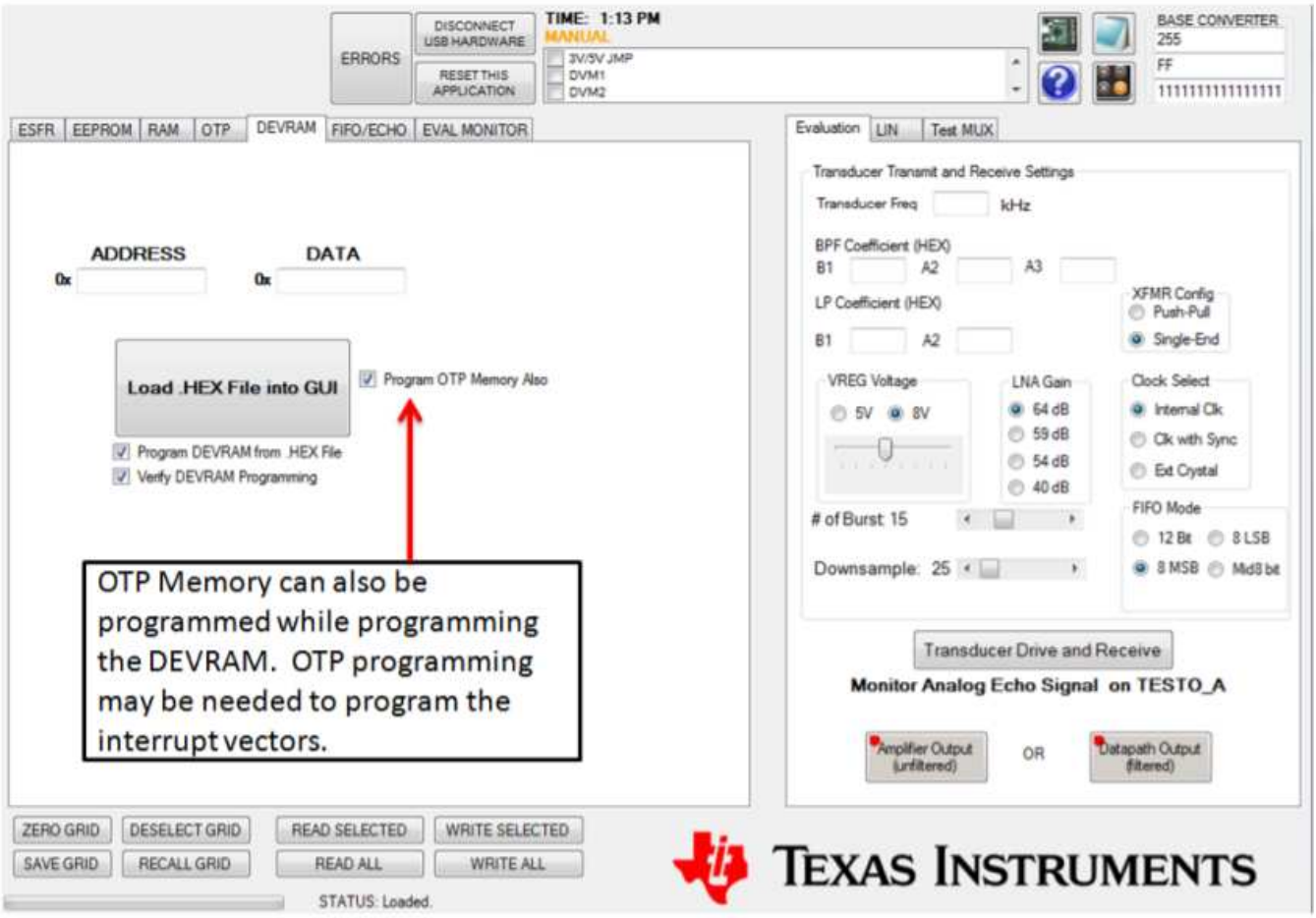


4. Press the Check OTP Status button to verify what is currently programmed into OTP. The three possible results are:
   - Programmed to Jump to DEVRAM: The jump to DEVRAM statement has been programmed into the OTP. This means that programs loaded into the DEVRAM are executed.
   - OTP Empty: Nothing has been programmed in the OTP.
   - Programmed: The OTP has been programmed with something other than the jump to DEVRAM statement.

   The DEVRAM tab is set up only for individual register read/write, without the use of the grid. When this tab is displayed, the READ SELECTED / READ ALL and WRITE SELECTED / WRITE ALL buttons perform the same operation, respectively. The DEVRAM tab also contains buttons used to load a .HEX 8051 program file into the 8051 MCU in the PGA450-Q1 device. The process of loading the .HEX file into the DEVRAM is identical to that of OTP. For a pristine IC that has never been programmed before to run the software from the DEVRAM (OTP Status reads "OTP Empty"), the OTP memory must be programmed with some specific instructions to redirect the 8051 to DEVRAM. This must only be done once.
5. Check the "Program OTP Memory Also" button, and the GUI programs the OTP with this jump statement and program the DEVRAM with the selected .HEX file.

Copyright © 2018, Texas Instruments Incorporated

Figure 7. OTP Memory Can Be Programmed While Programming Development RAM



Socket for Programming OTP

The PGA450Q1EVM runs from the PGA450-Q1 device that is soldered to the board. The EVM provides a footprint for a socket to enable programming the OTP in devices that are for customer-board use. The socket is not populated by default on the EVM. The part number for the recommended socket is OTS-28-0.65-01. The GUI then can be used to select the target PGA450-Q1 device when programming OTP (the two options are the soldered device or the device in the socket). More details of how to do this are described in the OTP section.

2.3 Programming the PGA450-Q1 EEPROM

The PGA450Q1EVM-S is preloaded with the EEPROM values listed in Table 1. These instructions are provided if you prefer to modify the EEPROM using the PGA450Q1EVM GUI rather than the corresponding UART command. Use the following steps to program the device for EEPROM memory.

1. Assuming the MCU status is still set to “MICRO IS IN RESET”, go to the EEPROM tab, and enter the register values for addresses 0x00 through 0x1F as listed in Table 2. The values for 0x00 to 0x1F are the threshold values used in software. TI recommends that you begin with the example values provided, and then adjust these values as necessary. The value at address 0x1F is the sensor address.
2. Click the WRITE SELECTED button. To ensure the threshold values are written correctly, click the “READ ALL” button.
3. Click the PROGRAM EEPROM button to store the threshold values.
4. Click the Reload button. To ensure the threshold values have been written correctly and retained, click the “READ ALL” button. Figure 8 shows what the EEPROM table looks like.

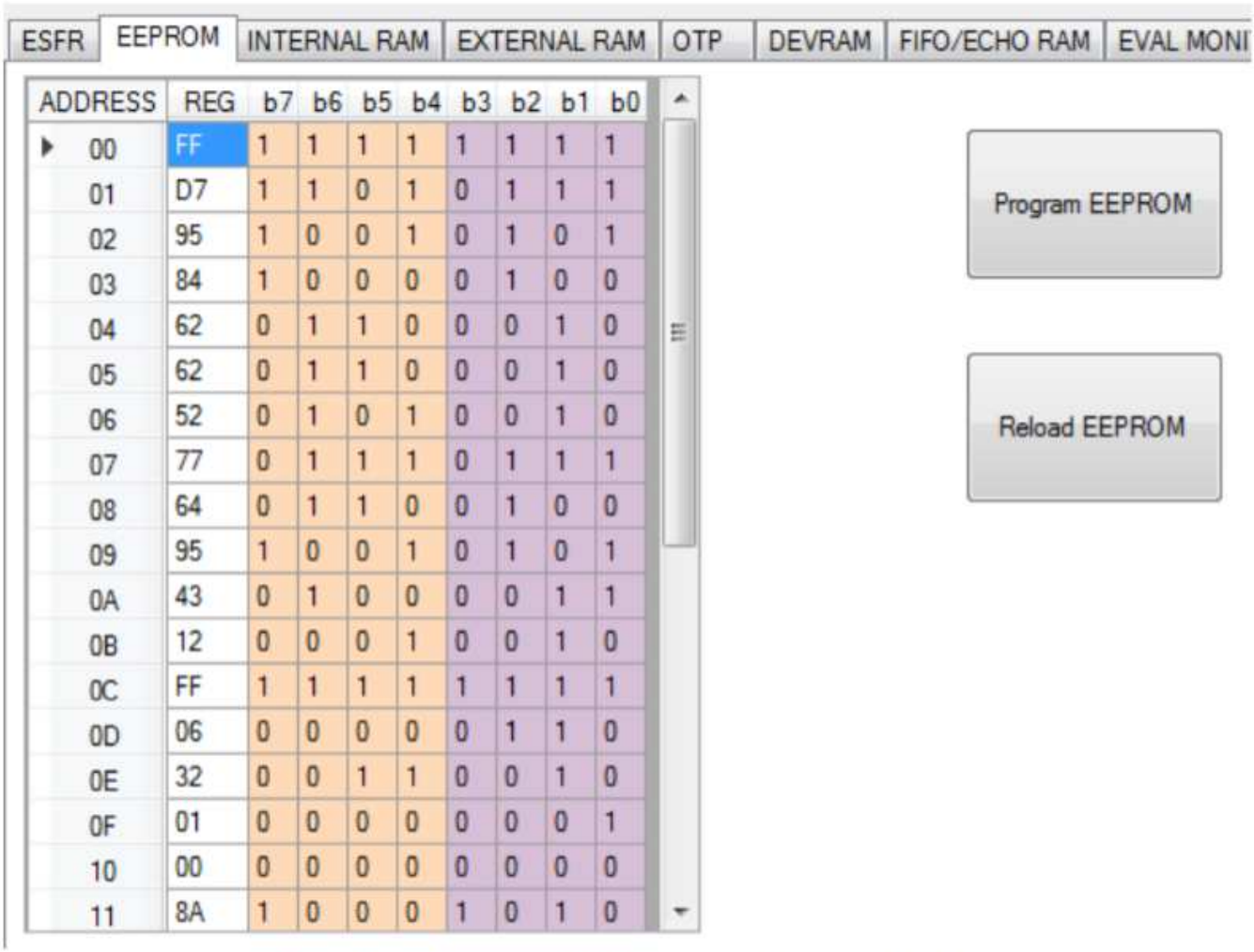
Copyright © 2018, Texas Instruments Incorporated

Table 2. EEPROM Register Map Values

Register Address	Register Value	Description
0x00	0xFF	Threshold level 0
0x01	0xD7	Threshold level 1
0x02	0x95	Threshold level 2
0x03	0x84	Threshold level 3
0x04	0x62	Threshold level 4
0x05	0x62	Threshold level 5
0x06	0x52	Threshold level 6
0x07	0x78	Where in FIFO to apply threshold level 7 defined at EEPROM address 0x08 from to the end of the FIFO. Long mode (most significant hex) multiplied by 40 [interval 40 to 600]. Short mode (least significant hex) multiplied by 8 [interval 8 to 120].
0x08	0x64	Threshold level 7, the fixed level of threshold to end of FIFO. Long mode and short mode multiplied by 5 and 1 respectively.
0x09	0x95	Threshold ignores count from beginning of FIFO for long (most significant hex) and short (least significant hex) modes respectively.
0x0A	0x43	1-6 change interval of the threshold level across FIFO for long mode (most significant hex) multiplied by 8 [interval 8 to 120]. Short mode (least significant hex) multiplied by 2 [interval 2 to 15].
0x0B	0x12	PULSE_CNTA
0x0C	0xFF	BLANKING_TIMER
0x0D	0x06	FIFO_CTRL
0x0E	0x32	DOWNSAMPLE
0x0F	0x01	CONTROL_1
0x10	0x00	BURST_MODE
0x11	0x8A	BURST_ONA_LSB
0x12	0x8A	BURST_OFFA_LSB
0x13	0x05	DEADTIME
0x14	0x09	SAT_DEGLITCH
0x15	0x03	BPF_B1_MSB
0x16	0x2D	BPF_B1_LSB
0x17	0xEC	BPF_A2_MSB
0x18	0x3D	BPF_A2_LSB
0x19	0xF9	BPF_A3_MSB
0x1A	0xA5	BPF_A3_LSB
0x1B	0x35	LPF_B1_MSB
0x1C	0xDD	LPF_B1_LSB
0x1D	0x14	LPF_A2_MSB
0x1E	0x46	LPF_A2_LSB
0x1F	0x01	Sensor address

Copyright © 2018, Texas Instruments Incorporated

Figure 8. EEPROM Registers Programmed Correctly



ADDRESS	REG	b7	b6	b5	b4	b3	b2	b1	b0
00	FF	1	1	1	1	1	1	1	1
01	D7	1	1	0	1	0	1	1	1
02	95	1	0	0	1	0	1	0	1
03	84	1	0	0	0	0	1	0	0
04	62	0	1	1	0	0	0	1	0
05	62	0	1	1	0	0	0	1	0
06	52	0	1	0	1	0	0	1	0
07	77	0	1	1	1	0	1	1	1
08	64	0	1	1	0	0	1	0	0
09	95	1	0	0	1	0	1	0	1
0A	43	0	1	0	0	0	0	1	1
0B	12	0	0	0	1	0	0	1	0
0C	FF	1	1	1	1	1	1	1	1
0D	06	0	0	0	0	0	1	1	0
0E	32	0	0	1	1	0	0	1	0
0F	01	0	0	0	0	0	0	0	1
10	00	0	0	0	0	0	0	0	0
11	8A	1	0	0	0	1	0	1	0

(1) The *Program EEPROM* button programs the EEPROM cache into the EEPROM memory. The values displayed in the GUI are transferred to the cache before being programmed.

(2) The *Reload EEPROM* button copies the EEPROM memory into the EEPROM cache. The values in the GUI are then updated.

3 Create/Load Code

3.1 PGA450-Q1 Software Development Guide for the UART and LIN Demo 8051 MCU Firmware

The PGA450-Q1 has an integrated 8051 8-bit microcontroller, which is able to run its firmware based program from the internal DEVRAM or OTP memory available on the device. The 8051 is automatically enabled upon power-up to immediately start running the firmware from OTP memory. If the firmware is loaded to the OTP memory, then the 8051 does not require any additional instructions to run the firmware. However, if the firmware is loaded into the DEVRAM memory, then you must first reprogram the firmware to DEVRAM memory upon power-up. DEVRAM memory is not able to retain the firmware when power-cycled, but offers the flexibility of modifying the firmware for experimentation.

The orderable TIDA-00151 Automotive Ultrasonic Sensor Interface IC for Park Assist or Blind Spot Detection Systems reference design (or PGA450Q1EVM-S) is programmed to run from OTP memory, and enables the UART interface for evaluation purposes. The LIN interface is not enabled in this version of the EVM-S, but can be enabled on a new device by modifying the demo firmware.

Copyright © 2018, Texas Instruments Incorporated

The demo firmware is a Keil uVision project composed of the following files:

Table 3. Demo Firmware Files in Keil uVision Project

File Name	Description
pga450_main.c	The main function is where the device first enters the firmware program and runs indefinitely by waiting on interrupt commands to execute any additional functions. The Initialization() function is the first function called at startup, and is only called once to configure the PGA450-Q1 ESFRs, UART port, timers, and interrupt routines. The other commands defined in the _main.c file are the detailed functions for the predefined UART and LIN commands, and are only executed upon receiving a valid input from the master controller.
pga450_init.c	The Initialization() function configures the following in the listed order: 1. Initialize the UART, GPIO, and LIN ports. 2. The PGA450-Q1 special function registers are initialized to prepare the device for ultrasonic data collection when using the Murata MA58MF14-7N transducer. 3. The down sample rate is set to the maximum to enable long range object detection up to approximately 7m. 4. The LIN port is enabled with an enhanced CRC. 5. Timer0 is enabled as a free running timer for accurate time-of-flight data collection. 6. Timer1 is enabled for 8-bit serial UART communication at 19.2kbps. 7. Interrupts are enabled for the serial UART port and the LIN port. 8. Reload EEPROM contents to EEPROM buffer. Once the Initialization() function is completed, the program returns to the main while loop to wait for an incoming UART or LIN command interrupt.

Copyright © 2018, Texas Instruments Incorporated

Table 3. Demo Firmware Files in Keil uVision Project (continued)

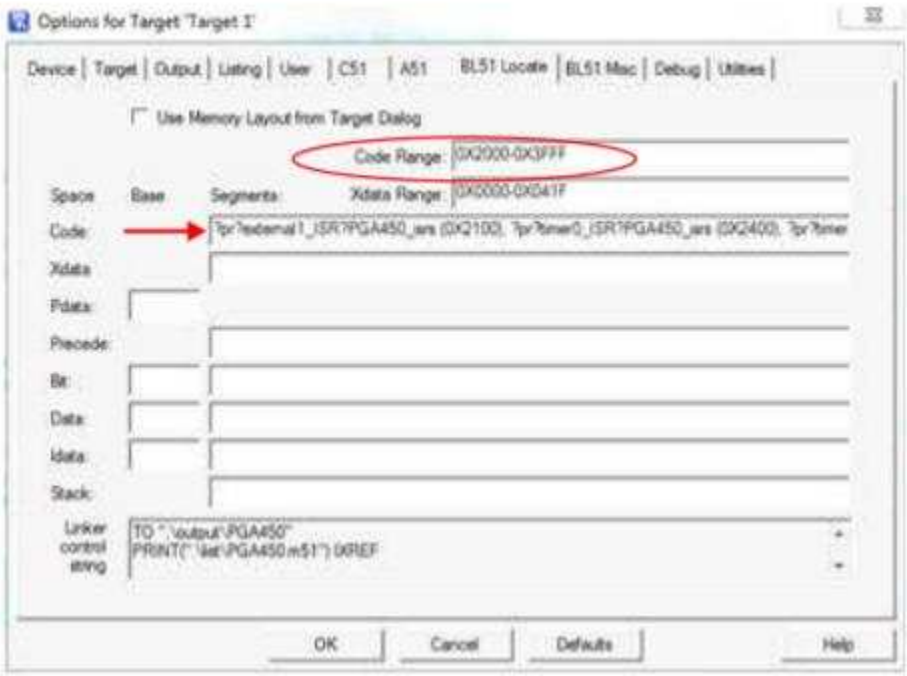
pga450_isrs.c	The Interrupt Service Routines enabled during the Initialization() function configures the device to monitor the UART and LIN ports for incoming commands from a master controller. In the event that an UART command is received, the serial_ISR() first checks if the incoming bytes from the master are valid. Every predefined command requires that the break_byte=0x00 (first byte), the sync_byte=0x55 (second byte), and the addr_nibble (LSB nibble of third byte) be equal to the value at EEPROM address 0x1F. If any of these checks are violated, the serial_ISR() function aborts itself. If the received data check is successful, the third byte is further parsed to check which of the eight pre-defined commands is to be run by referring to the MSB nibble of the third byte. The available demo commands include: • Command 0 - Test UART communication. • Command 1 - Trigger a short or long distance burst and capture with predefined drive/filter settings, burst, capture, and compare. • Command 2 - Read the latest instance of the threshold comparison results. • Command 3 - Update an EEPROM value. • Command 4 - Read all FIFO data. • Command 5 - Burst, capture, and compare based on the custom EEPROM configuration. • Command 6 - Report threshold time and level values. • Command 7 - Not Used. Reserved for custom function. Each command expects a certain number of bytes to further process the command in the “pga450_main.c” file. Refer to the “PGA450Q1EVM-S User's Guide and TIDA-00151 UART Demo Instructional User’s Guide” for details on each command’s input. When a valid command is received, the 8051 executes the command in detail on the “_main.c” file, and returns data (when applicable) to the master in these functions. The same order of events of receiving valid input from the master and retuning data is true for the LIN interface.
pga450_vars.c	The variables file configures pointers to the FIFO data, external RAM, and EEPROM memories accessed by the 8051. Additional arrays are defined for the UART and LIN input and output data.
pga450.h	This header file maps the external special function register (ESFR) names to a numerical address of the PGA450-Q1 device. The 8051 specific register names of the special function register (SFR) and bit registers are also mapped for ease of use throughout the project.
pga450_vars.h	This header file maps the 8051 pins for UART, LIN, and GPIO operation. The external, and external memory locations are defined, and the PGA450-Q1 specific functions such as Initialization() and commandX() are declared.
STARTUP.A51	When compiling the uVision project for the PGA450-Q1, this file determines if the generated output file targets the PGA450-Q1 DEVRAM or OTP memory. Refer to “PGA450Q1EVM User's Guide” for details on configuring this file for the correct memory type.

Copyright © 2018, Texas Instruments Incorporated

3.2 Keil µVision Settings for Programming Firmware to the PGA450-Q1 DEVRAM or OTP Memory

3.2.1 Setup

Figure 9. DEVRAM Target Options

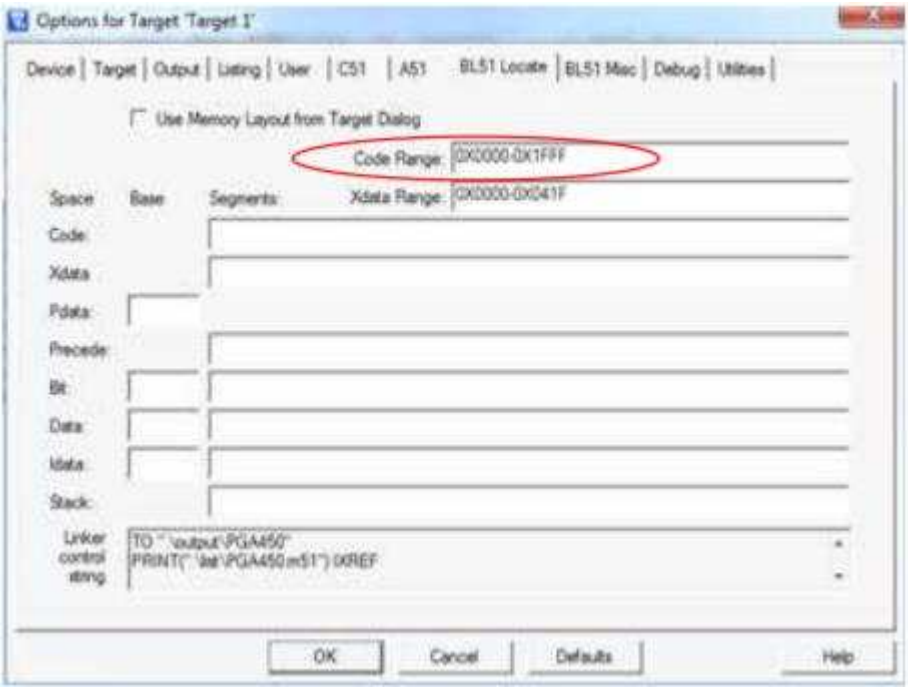


Follow these steps to Program to DEVRAM:

1. Change the code range to the DEVRAM memory space.
   a. Right click on Target 1 in the project window, and select the option for Target. Figure 9 shows an example of this step.
   b. Go to the BL51 Locate tab, and modify the Code Range to go from 0x2000–0x3FFF.
2. Copy the following to the code box:
   ?pr?external1_ISR?PGA450_isrs (0X2100),
   ?pr?timer0_ISR?PGA450_isrs (0X2400),
   ?pr?timer1_ISR?PGA450_isrs (0X2800),
   ?pr?serial_ISR?PGA450_isrs (0X2C00),
   ?pr?linPID_ISR?PGA450_isrs (0X3000),
   ?pr?linSciRxData_ISR?PGA450_isrs (0X3400),
   ?pr?linSciTxData_ISR?PGA450_isrs (0X3800),
   ?pr?external0_ISR?PGA450_isrs (0X3900),
   ?pr?linSync_ISR?PGA450_isrs (0X3D00)
3. Comment out the OTP section in STARTUP.A51, and uncomment the OTP section A. Figure 10 shows an example of this step.

Copyright © 2018, Texas Instruments Incorporated

Figure 10. OTP Target Options

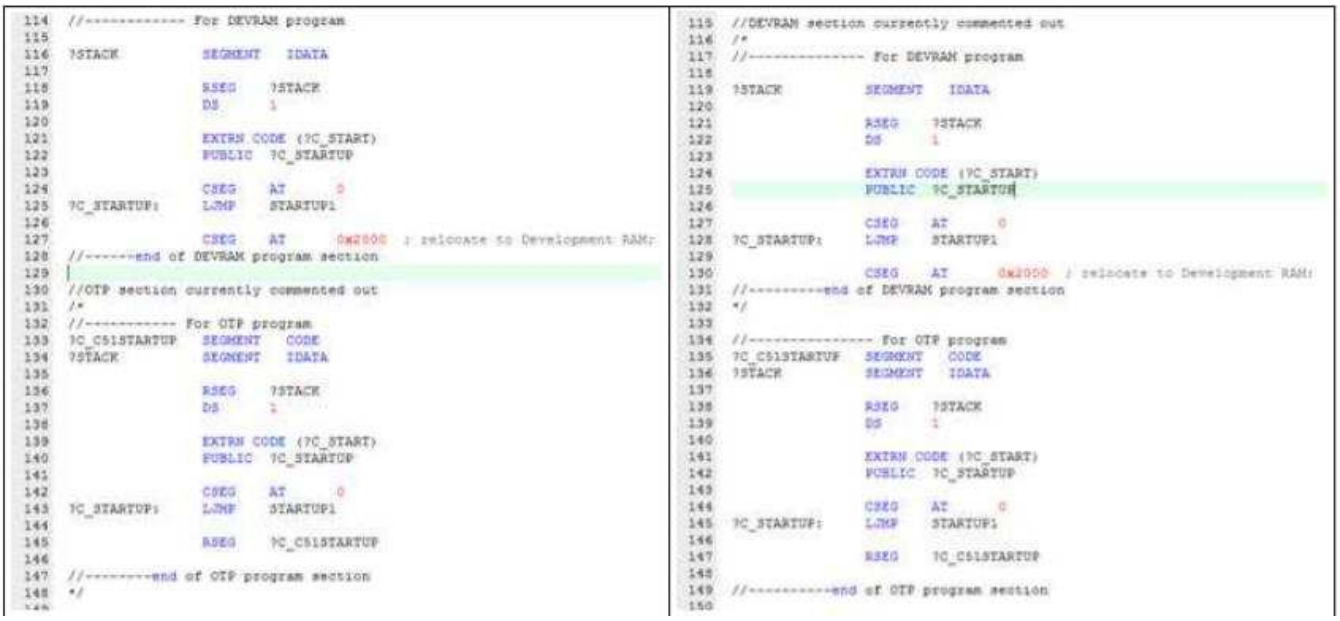


Follow these steps to program the OTP:

1. Change the code range to the OTP memory space.
   a. Right click on Target 1 in the project window and select Options for Target.
   b. Go to the BL51 Locate tab and modify the Code Range to go from 0x0000–0x1FFF.
2. Delete everything in the Code box.
3. Comment out the DEVRAM section in STARTUP.A51, and uncomment the OTP section. Figure 11 shows an example of this step.
4. Build the PGA450.uvproj to generate the custom .HEX file used to program the internal 8051 core. The LIN Demonstration using PGA450Q1EVM Firmware Rev 2.1 provides instructions on how to upload the .HEX file using the EVM GUI.

Copyright © 2018, Texas Instruments Incorporated

Figure 11. DEVRAM STARTUP.A51 Example (Left). OTP STARTUP.A51 Example (Right)



```
//------------ For DEVRAM program

?STACK          SEGMENT   IDATA

                RSEG    ?STACK
                DS      1

                EXTRN CODE (?C_START)
                PUBLIC  ?C_STARTUP

                CSEG    AT      0
?C_STARTUP:     LJMP    STARTUP1

                CSEG    AT      0x2000  ; relocate to Development RAM:
//------end of DEVRAM program section

//OTP section currently commented out
/*
//----------- For OTP program
?C_C51STARTUP   SEGMENT   CODE
?STACK          SEGMENT   IDATA

                RSEG    ?STACK
                DS      1

                EXTRN CODE (?C_START)
                PUBLIC  ?C_STARTUP

                CSEG    AT      0
?C_STARTUP:     LJMP    STARTUP1

                RSEG    ?C_C51STARTUP

//---------end of OTP program section
*/
```

```
//DEVRAM section currently commented out
/*
//--------------- For DEVRAM program

?STACK          SEGMENT   IDATA

                RSEG    ?STACK
                DS      1

                EXTRN CODE (?C_START)
                PUBLIC  ?C_STARTUP

                CSEG    AT      0
?C_STARTUP:     LJMP    STARTUP1

                CSEG    AT      0x2000  ; relocate to Development RAM:
//----------end of DEVRAM program section
*/

//--------------- For OTP program
?C_C51STARTUP   SEGMENT   CODE
?STACK          SEGMENT   IDATA

                RSEG    ?STACK
                DS      1

                EXTRN CODE (?C_START)
                PUBLIC  ?C_STARTUP

                CSEG    AT      0
?C_STARTUP:     LJMP    STARTUP1

                RSEG    ?C_C51STARTUP

//----------end of OTP program section
```

4 Run Code and Interpret Results

4.1 Predefined Codes

4.1.1 UART Command Listing

The following is a list of predefined commands made available on the default DEVRAM and OTP firmware provided for the PGA450Q1EVM-S.

Command 0 - Test UART communication. If communication is working correctly, the PGA450-Q1 returns a value of 0x12 0x23. Example hex entry: 0x00, 0x55, 0x01, 0x00. where:

- [0] 0x00: break field
- [1] 0x55: synchronization field
- [2] 0x01: 0 = command 0. 1 = sensor address set in EEPROM address 0x31
- [3] 0x00: ignored but required checksum

Command 1—Trigger a short or long distance burst and capture with hard-coded drive and receive settings. Also reads first instance of threshold crossing for closest object detected. Example hex entry: 0x00, 0x55, 0x11, 0x02, 0x00. Where:

- [0] 0x00: break field
- [1] 0x55: synchronization field
- [2] 0x11: 1 = command 1. 1 = sensor address set in EEPROM address 0x31
- [3] 0x02: 00 = listen. 01 = short. 02 = long
- [4] 0x00: ignored but required checksum

Command 2—Read first instance of threshold crossing for closest object detected. Example hex entry: 0x00, 0x55, 0x21, 0x00. Where:

- [0] 0x00: break field
- [1] 0x55: synchronization field
- [2] 0x21: 2 = command 2. 1 = sensor address set in EEPROM address 0x31
- [3] 0x00: ignored but required checksum

Copyright © 2018, Texas Instruments Incorporated

Command 3 - Update an EEPROM value. Example hex entry: 0x00, 0x55, 0x31, 0x00, 0xFF, 0x00. Where:

- [0] 0x00: break field
- [1] 0x55: synchronization field
- [2] 0x31: 3 = command 3. 1 = sensor address set in EEPROM address 0x31
- [3] 0x00: 00 = EEPROM address
- [4] 0xFF: FF = EEPROM data
- [5] 0x00: ignored but required checksum

Command 4 - Read all 768 bytes of FIFO data. Example hex entry: 0x00, 0x55, 0x41, 0x00 . Where:

- [0] 0x00: break field
- [1] 0x55: synchronization field
- [2] 0x41: 4 = command 4. 1 = sensor address set in EEPROM address 0x31
- [3] 0x00: ignored but required checksum

Command 5 - Burst and capture ultrasonic profile based on EEPROM configuration of drive and receive settings. Example hex entry: 0x00, 0x55, 0x51, 0x02, 0x00. Where:

- [0] 0x00: break field
- [1] 0x55: synchronization field
- [2] 0x51: 5 = command 5. 1 = sensor address set in EEPROM address 0x31
- [3] 0x02: 0 = listen. 1 = short. 2 = long
- [4] 0x00: ignored but required checksum

Command 6 - Threshold values report for selected mode of operation (short or long). Example hex entry: 0x00, 0x55, 0x61, 0x02, 0x00. Where:

- [0] 0x00: break field
- [1] 0x55: synchronization field
- [2] 0x51: 5 = command 5. 1 = sensor address set in EEPROM address 0x31
- [3] 0x02: 1 = short. 2 = long
- [4] 0x00: ignored but required checksum

“COMMAND 6” identifies the range across the 768 points of FIFO where the threshold level changes. The level is an 8-bit value from 0-255. For example, “COMMAND 6” of UART reports the threshold values and is decoded as follows:

- "09 FF 20 84 40 74 60 54 80 4C A0 3C E0 34 19 1E"
- Byte1 = x09 = from FIFO 0 to 9, the threshold levelInit is Byte2 (255) (based on EEPROM addr 0x09 as initial ignore count defaulted to 9. Ignore count means threshold level is set to 255)
- Byte2 = xFF = 255d level
- Byte3 = x20 = from FIFO 10 to 32, the threshold level0 is Byte4 (132)
- Byte4 = x84 = 132d level
- Byte5 = x40 = from FIFO 33 to 64, the threshold level1 is Byte6 (116)
- Byte6 = x74 = 64d level
- Byte7 = x60 = from FIFO 65 to 96, the threshold level2 is Byte8 (84)
- Byte8 = x54 = 84d level
- Byte9 = x80 = from FIFO 97 to 128, the threshold level3 is Byte10 (76)
- Byte10 = x4C = 76d level
- Byte11 = xA0 = from FIFO 129 to 160, the threshold level4 is Byte12 (60)
- Byte12 = x3C = 60d
- Byte13 = xE0 = from FIFO 161 to 224, the threshold level5 and level6 is Byte14 (52) (In this example, level5 and level6 are the same values, lumped into a single return value, not the double interval of 224 - 160 = 64, which is the typical interval of 32.)

Copyright © 2018, Texas Instruments Incorporated

- Byte14 = x34 = 52d
- Byte15 = x19 = from FIFO 225 to the end of the FIFO, the threshold level7 is Byte16 (30)
- Byte16 = x1E = 30d

Command 7 - Not used in example firmware. Reserved for custom user defined function. Example hex entry: 0x00, 0x55, 0x71, 0x00. Where:

- [0] 0x00: break field
- [1] 0x55: synchronization field
- [2] 0x71: 7 = command 7. 1 = sensor address set in EEPROM address 0x31
- [3] 0x00: ignored but required checksum

4.1.2 UART Demonstration

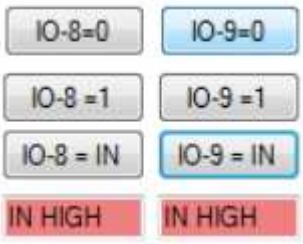
Testing the UART Connection with TI-GER Board. Use the steps that follow to test the UART connection with the TI-GER board.

1. If the user has not disconnected the system power, ensure that the MCU is activated again by clicking the ON (MicroActive) on the ESFR tab.
2. In the top-right of the PGA450Q1EVM GUI, click the Direct TI-GER Control button. This button does not contain text; it is just an image (see the red circle in Figure 12).

Figure 12. Direct TI-GER Control Button

3. Under the GPIO tab, clicking the IO-8 = IN and IO-9 = IN buttons to ensure that IO-8 and IO9 are set as IN HIGH (see Figure 13).

Figure 13. GPIO Tab



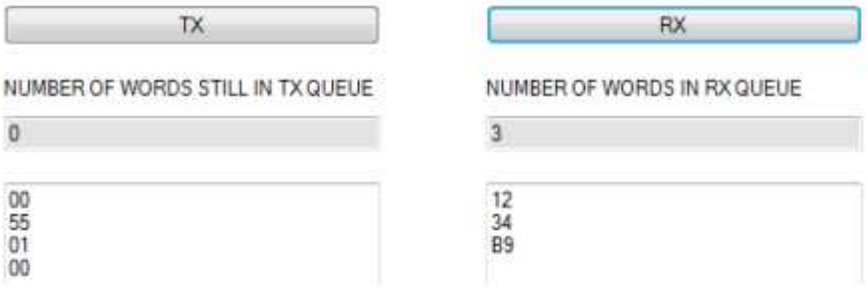
4. Connect the UART pins of the sensor to the UART pins of the TI-GER board (see Figure 4).
   - Pin 10 (TxD) on the TI-GER board goes to RXD on the Sensor.
   - Pin 20 (RxD) on the TI-GER board goes to TXD on the Sensor.
5. On the UART tab, under the UART CONTROL SETTING section, select SETUP #9.
6. In the BAUD RATE box in the lower right corner of the GUI, enter 19200.
7. On the UART TEST tab, select SETUP #9.
8. Click the “OPEN UART MODULE” button.
9. Click the “CHANGE BAUD RATE” button, which should display 19200 and 1-bit = 52.1 μs.
10. Click the “CHECK FOR ERRORS” to ensure there no errors occurred. If an error occurred, one of the boxes is red, at which point the user must click the “CLOSE MODULE IMMEDIATELY” and “CLOSE

Copyright © 2018, Texas Instruments Incorporated

UART MODULE" buttons. Then repeat the setup, beginning at Step 8.

11. Write the values shown in Figure 14 in the TX box. Each byte must be typed on a separate row. The description of the TX data packet is as follows:
    - 00: is the break field
    - 55: is the synchronization field
    - 01: 0 is the command and 1 is the sensor address. Ensure that the sensor address matches what was programmed in the EEPROM address 0x07 from the previous steps.
    - 00: is the checksum field; however, this data is discarded, so the user may enter any value here.

Figure 14. No Response in the RX Box



12. Click the RX button to display the response of the sensor. The description of the RX data packet is as follows:
    - 12: is dummy byte 1 and has no special meaning.
    - 34: is dummy byte 2 and has no special meaning.
    - B9: is the checksum from byte 1 and byte 2.

    The function that calculates the checksum is described as follows: Table 4 provides a checksum calculation of four bytes shown. If the frame has four data bytes of the protected identifier and three data bytes, the calculation is the same. The data = 0x4A, 0x55, 0x93, 0xE5.

Table 4. An Example of Checksum Calculation

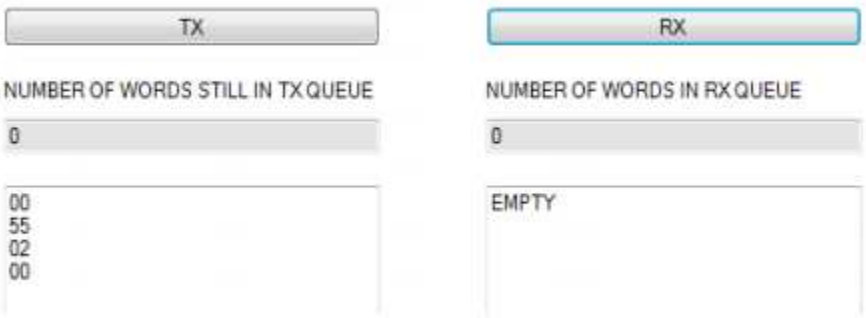
Action	Hex	Carry	D7	D6	D5	D4	D3	D2	D1	D0
0x4A	0x4A		0	1	0	0	1	0	1	0
+0x55 = Add Carry	0x9F	0	1	0	0	1	1	1	1	1
	0x9F		1	0	0	1	1	1	1	1
+0x93 = Add Carry	0x132	1		0	1	1	0	0	1	0
	0x33			0	1	1	0	0	1	1
+0xE5 = Add Carry	0x118	1	0	0	0	1	1	0	0	0
	0x19		0	0	0	1	1	0	0	1
Invert	0xE6		1	1	1	0	0	1	1	0
0x19 + 0.xE6 =	0xFF		1	1	1	1	1	1	1	1

Copyright © 2018, Texas Instruments Incorporated

The resulting sum is 0x19. Inversion yields the final result of checksum = 0xE6. The receiving node can check the consistency of the received frame by using the same addition mechanism. When the received checksum (0xE6) is added to the intermediate result (0x19), the sum is 0xFF.

The previously described TX command is called command 0 – Sensor Check Command. If the sensor is address 0x01 and you attempt to communicate with a sensor using a different address, the sensor 0x01 gives no response. Figure 15 shows that no response was received as displayed in the RX box because the command 0 is sent to sensor address 0x02.

Figure 15. No Response in the RX Box



4.1.3 LIN Demonstration

To learn more about LIN demonstration using internal 8051 MCU firmware of the PGA450-Q1 device please refer to the following application report: LIN Demonstration using PGA450Q1EVM Firmware Rev 2.1 (Rev. A).

Copyright © 2018, Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated