

I2C I/O Expander Integrated In Low-Cost MSP430 MCU



Introduction

Many applications require simple I/Os functions such as blinking multiple LEDs; however, there may be not enough general-purpose I/O pins for the host microcontroller (MCU) or processor to perform these tasks. The inter-integrated circuit (I²C) interface enables serial communication between the host and the MSP430™ MCU, which can act as an I/O expander for I²C communication. In this example, the MSP430 MCU is an I²C slave using the enhanced serial communication interface (eUSCI) module to receive commands from the master to control 8 general-purpose I/O pins.

The following functions can be expanded:

- I²C interface to expand with 8 simple I/O pins
- Set I/O output value (port, group, or bit)
- Read I/O input value supported only by GUI (port, group, or bit)

Note

This example can be used with any MSP430 LaunchPad™ development kit with the required MCU peripherals. For migrating pinouts and peripherals, see the device-specific data sheet.

Implementation

Figure 1 shows the communication between the MSP430FR2433 MCU and the host controller through I²C or the PC GUI through UART (backchannel USB-UART on the LaunchPad kit).

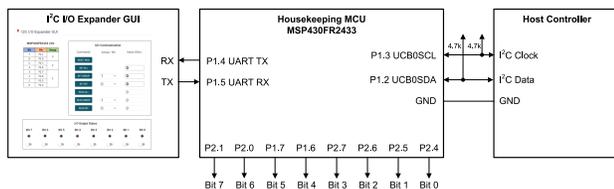


Figure 1. Implementation Overview

The host controller can write data to the eUSCI_B0 of MSP430FR2433 MCU through the I²C bus, using serial data (SDA) and serial clock (SCL). There are 8 pins on the housekeeping MCU to expand the I²C commands.

The host processor I²C configuration is:

- Address slave with 7-bit address
- Single master environment
- Slave address is 0x48

The I²C communication is achieved by the correct slave address and a 3-byte message. The first byte is the command, the second byte selects the group or bit number, and the third byte is the data. The MSP430FR2433 device outputs the data to or reads the data from the mapped I/O pins defined in the software. Figure 2 shows the pinout and connections on the MSP430FR2433 LaunchPad development board.

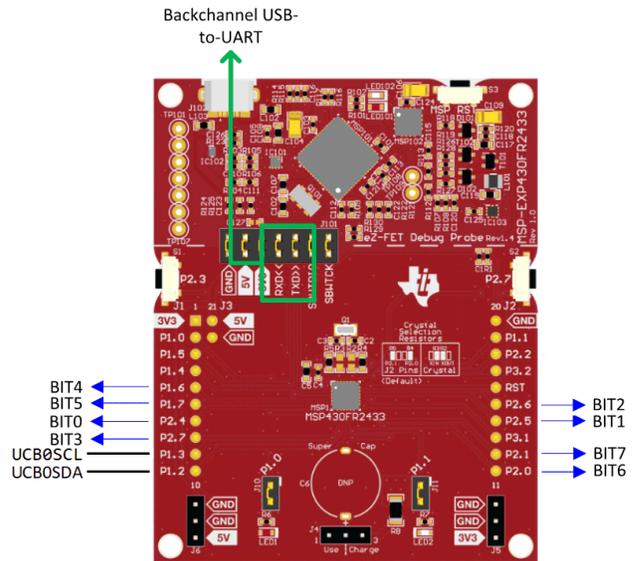


Figure 2. MSP430FR2433 LaunchPad Board and Connections

In the 3-byte message package, the first byte is the **Host Command**, which tells the housekeeping MCU what operation should be done for the pins. Command options are Reset All (0), Set All (1), Set Group (2), Set Bit (3), Read All (4), Read Group (5), and Read Bit (6). Table 1 lists the 8-bit host command values.

Table 1. Host Command: Valid Value 0-6 in 1 Byte

Host Command	Value (Dec)	Function
Reset MCU	0	Reset housekeeping MCU; set all bit 0 to bit 7 I/Os to 0; initialize GUI
Set All	1	Set housekeeping MCU all bit 0 to bit 7 I/Os based on "Data Value" command

Table 1. Host Command: Valid Value 0-6 in 1 Byte (continued)

Host Command	Value (Dec)	Function
Set Group	2	Set selected 1-3 group I/Os based on "Data Value" command
Set Bit	3	Set selected bit 0 to bit 7 IO based on "Data Value" command
Read All	4	Read out all bit 0 to bit 7 I/Os; supported only by GUI
Read Group	5	Read out selected 1-3 group I/Os; supported only by GUI
Read Bit	6	Read out selected bit 0 to bit 7 I/Os; supported only by GUI
	Other value	No meaning

In the 3-byte message package, the second byte is the **Data Index**. The 8-bit data index value determines which group or bit number to write data to when using a Set Group, Set Bit, Read Group, or Read Bit command. The 8-bit I/O port defined for the MSP430FR2433 example code divides the port into 3 groups: group 1 (bits 3-0), group 2 (bits 5-4), and group 3 (bits 7-6). For a Reset All, Set All, or Read All command, this byte is a "don't care". [Table 2](#) lists the 8-bit Data Index values.

Table 2. Data Index: Valid Value 0-7 in 1 Byte

Data Index	Function
0	Configure BIT 0
1	Configure BIT 1 / Group 1
2	Configure BIT 2 / Group 2
3	Configure BIT 3 / Group 3
4	Configure BIT 4
5	Configure BIT 5
6	Configure BIT 6
7	Configure BIT 7
Other data	No meaning

In the 3-byte message package, the last byte is the **Data Value**. This is the 8-bit data value to be output to the I/O port, specified group, or specified bit. For a Set command (1, 2, or 3), the data in bit 7 to bit 0 represents the 8-bit value to be sent to the selected output. For the Reset All command or any Read command, the 8-bit data is don't care. [Table 3](#) lists the 8-bit data values.

Table 3. Data Value: Valid Value 0-255 in 1 Byte

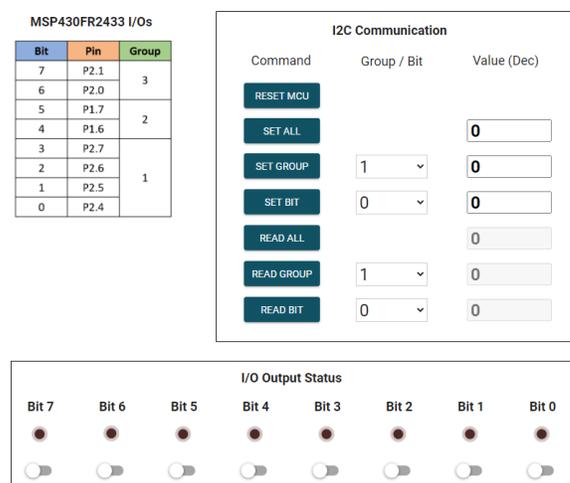
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset MCU	X	X	X	X	X	X	X	X
Set All	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
Set Group	X	X	X	X	0/1	0/1	0/1	0/1
Set Bit	X	X	X	X	X	X	X	0/1
Read All	X	X	X	X	X	X	X	X
Read Group	X	X	X	X	X	X	X	X
Read Bit	X	X	X	X	X	X	X	X

Performance

The host controller sends the host command and data to the MSP430FR2433 MCU using a specified bit rate. The bit rate in the I²C I/O expander code example is approximately 100 kbps.

The time for action of MSP430FR2433 MCU depends on the CPU clock frequency and the low-power mode (LPM) setting of the device. The following test results used the default 1-MHz CPU clock frequency and LPM0 for standby. The action time can be optimized by using a higher CPU clock frequency, which may increase code size to configure the CPU clock, as well as a lower low-power mode to let the CPU wake up from LPM mode more quickly.

I²C I/O Expander GUI


Figure 3. I²C I/O Expander GUI

[Figure 3](#) shows the GUI, which supports the following features:

- Monitor the initiated I²C transactions and update the status of the MSP430FR2433 I/O pins with a virtual LED array
- Act as the host application to expand the I/O pins over backchannel USB-to-UART.

When using the GUI as a monitor, it updates the data values to the associated command and group

or bit value initiated over the 3-byte I²C transaction. Furthermore, if a Set command is issued, it updates the virtual LED array with the values specified from the data byte of the transaction. For instance, if a 3-byte I²C transaction is 0x03, 0x02, 0x01, then a "Set Bit" command is issued, bit 2 is the specified bit, and it writes a value of "1". P2.6 goes high, the GUI updates the "Group/Bit" value to "2", and the "Data" value updates to "1" for the "Set Bit" row. Additionally, the LED and switch for bit 2 switch to the "ON" state.

Alternatively, the GUI can serve as the host application using backchannel USB-to-UART. The user can click on a Command button to implement Reset MCU, Set, or Read command for all, a group, or a bit. Ensure the "Group/Bit" and "Data" values are set by pressing "Enter" key before initiating a command. Additionally, switches are available below the LEDs to individually set or clear bits. To initiate the same command as before, update the "Group/Bit" value to "2" and the "Data" value to "1" in "Set Bit" row, then click on the "Set Bit" command to initiate the command. P2.6 goes high and the bit 2 LED and switch to the "ON" state.

Figure 4 shows a code flowchart to help with understanding of the software. An I²C transaction or GUI UART command triggers an interrupt to set or read the I/O values and update the GUI.

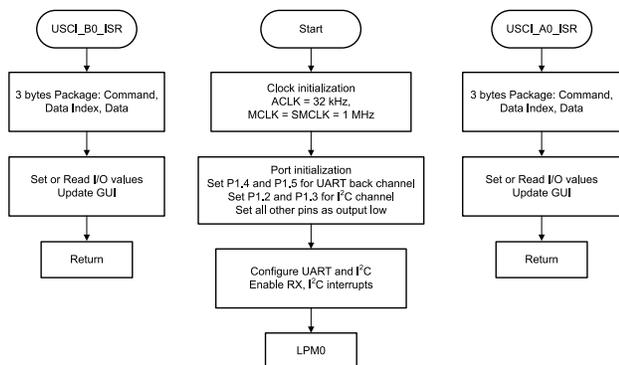


Figure 4. Software Flowchart

To Get Started

1. Watch the "[I²C I/O Expander](#)" training video, which shows how to use the serial peripheral interface to expand I²C data from a host controller into the secondary I/O ports.
2. [Order an MSP430FR2433 LaunchPad kit](#) to evaluate the I²C I/O expander example code.
3. Download and test this example with the [I²C I/O expander example GUI](#), where you can monitor I²C transactions from a host controller to the MSP430FR2433 or send commands to the expanded I/O port.
4. [Evaluate the I²C I/O expander example code](#) for the MSP430FR2433 LaunchPad kit.

Device Recommendations

Part Number	Key Features
MSP430FR2433	16KB FRAM, 4KB SRAM, 10-bit ADC, UART/SPI/I ² C, Timer
MSP430FR2422	8KB FRAM, 2KB SRAM, 10-bit ADC, UART/SPI/I ² C, Timer

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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, regulatory 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](#) 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.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated