ANY-OUT™ LDO Controlled by I²C™ IO Expander Device

Masashi Nogawa *Power Management Products*

ABSTRACT

Recent low-dropout (LDO) linear voltage regulator devices from Texas Instruments feature a new function termed *ANY-OUT*. With an ANY-OUT LDO, the device output voltage can be programmed by selecting the binary-weighted device pins to either be ground or float. This ANY-OUT method of selecting two options (to ground or to float) has a very good compatibility with well-known digital control methods such as I²C or PMBus™ .

In this document, a PC control example of an ANY-OUT LDO via I²C is demonstrated. This example indicates that the ANY-OUT is a good solution for digital control capability demands in the voltage regulator market.

Contents

1 Materials 2
2 Preparing the I²C IO Expander Board 2
3 Preparing the USB-to-I²C Communication Software on the PC 3
4 System Setup and Operation from a PC 5
5 Summary 11
6 Appendix: List of Released ANY-OUT LDO (as of May 23, 2012) 11

List of Figures

1 Schematic of I²C IO Expander Board 2
2 I²C IO Expander Board 3
3 Picture of USB-TO-GPIO 3
4 Tool Folder Web Page 4
5 Selecting Required File from Sub-Archive Files 4
6 Total System Connection 5
7 Starting GUI Software 5
8 GUI Screen after Sending the First Line of Initialization Code 6
9 GUI Screen after Sending the Second Line of Initialization Code 7
10 GUI Screen after Sending the Third Line of Initialization Code 7
11 GUI Screen with Failure, after Sending the First Line of Initialization Code 7
12 GUI Screen Reading the PG = L Status After Sending Disable Code 10
13 GUI Screen Reading the PG = H Status After Sending Enable (3.3V Output) Code 10

ANY-OUT is a trademark of Texas Instruments.
Microsoft is a registered trademark of Microsoft Corporation.
I²C is a trademark of NXP Semiconductors.
PMBus is a trademark of SMIF, Inc.
All other trademarks are the property of their respective owners.

1 Materials

Throughout this document, the following materials are used.

- One ANY-OUT LDO TPS7A7200RGW on evaluation module: TPS7A7200EVM-718
- One I²C IO expander device: PCA9557PW
- One universal serial bus (USB) to general-purpose input/output (GPIO) adaptor: USB-TO-GPIO
- One blank TSSOP universal (prototyping) board
- One P-type FET
- One USB cable (standard A to mini B)
- Several pin-headers, sockets, and jumper cables
- Two resistors: 3 kΩ × 2
- One bypass capacitor: 1 µF

2 Preparing the I²C IO Expander Board

As a first step, prepare the I²C IO expander board with a PCA9557PW device. Figure 1 is the target schematic. To avoid the confusion of handling similar component names on two boards, the pin headers and sockets are instead referred to as H1 (headers) and S1 and S2 (sockets) on this I²C IO expander board to differentiate from the pin header names on the TPS7A7200EVM-718, which begin with *J* or *JP*.

In Figure 1, H1 is a 2-pin × 5-pin header to a 10-pin ribbon cable that comes with the USB-TO-GPIO. Pins 3, 4, 9, and 10 are used in this document. Pins 3 and 4 provide power supply to the PCA9557 device. Pins 9 and 10 are the clock and data signals of the I²C interface, respectively.

S1 and S2 are sockets to JP1 and JP5, respectively, on the TPS7A7200EVM-718 board. At the S1 socket, a voltage supply is taken from the TPS7A7200EVM-718 with ON and OFF. ON (VCC) and OFF (GND) create a supply that generates a proper EN drive signal (that is, MP1, R1, and R2 form a level shifter). The S2 socket controls the ANY-OUT voltage setting pins. For the ease of connecting two boards, place S1 and S2 in correlation to JP1 and JP5, respectively. Note that J5 is a 7-pin × 2-pin header on the TPS7A7200EVM-718. Furthermore, be aware of what S2 component is selected.

T1 is a test point that is wire connected to the PG signal (TP1) on the TPS7A7200EVM-718. C1 is a bypass capacitor for the PCA9557.

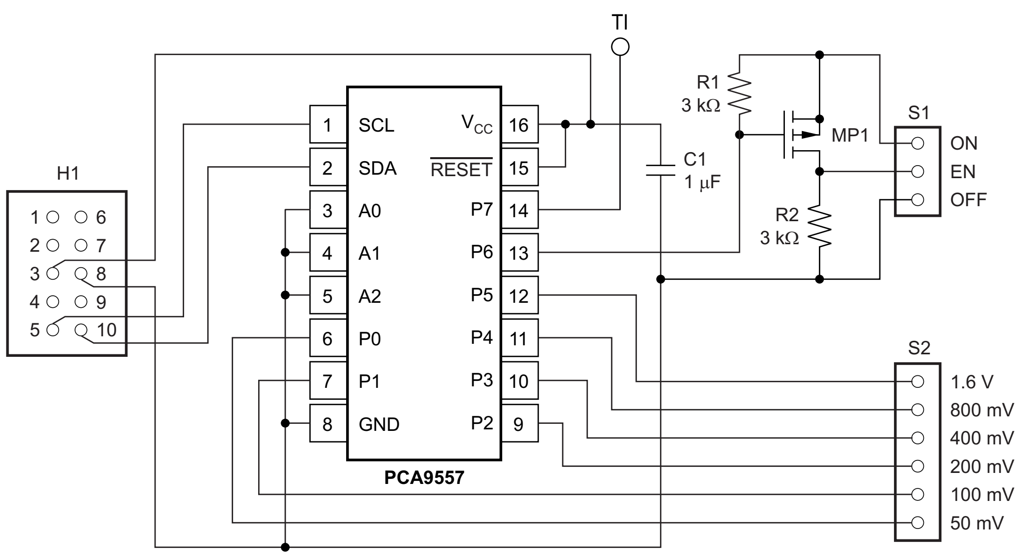

Figure 1. Schematic of I²C IO Expander Board

Figure 2 shows an example of implementing Figure 1. Here, an in-house (not released for public) universal TSSOP board is used. As can be determined from the simple schematic of Figure 1, creating this board takes only a couple of hours.

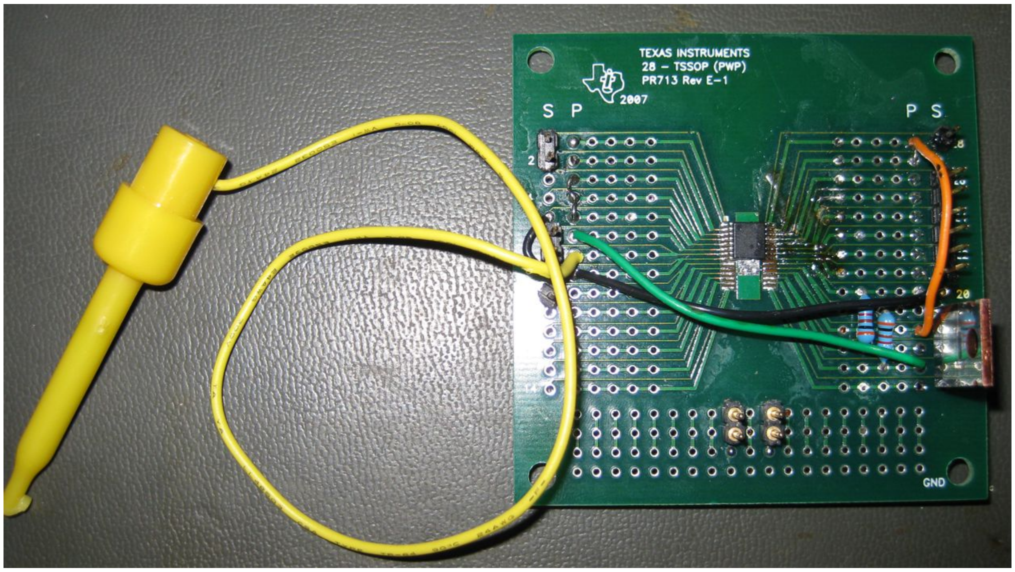

Figure 2. I²C IO Expander Board

3 Preparing the USB-to-I²C Communication Software on the PC

When a USB-TO-GPIO adaptor is obtained (as shown in Figure 3), download the control software from the Tool folder http://www.ti.com/tool/usb-to-gpio by following the *Related Products → TI Software → Reference GUIs and Libraries for Eval and Usage of the USB Interface Adaptor* (see Figure 4). The file *sllc288.zip* is then downloaded. After extracting the ZIP file, there is one *README.txt* file and three sub-ZIP files; only the contents in the *USB Interface Adaptor GUI-v1.10.zip* are required in this document (see Figure 5). To use this software, *Microsoft Framework version 2.0* is required and is available from the Microsoft® web site. Next, extract the *USB Interface Adaptor GUI-v1.10.zip* file.

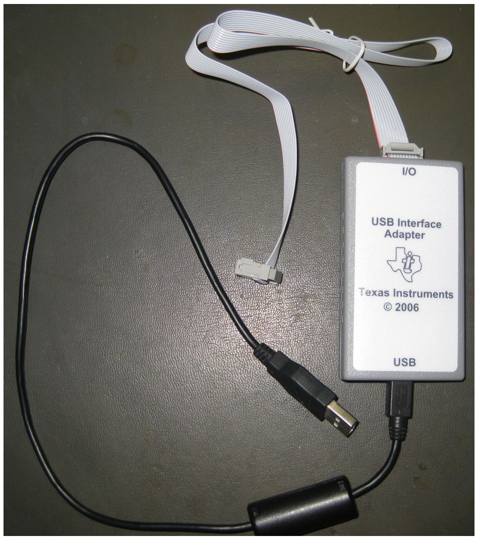

Figure 3. Picture of USB-TO-GPIO

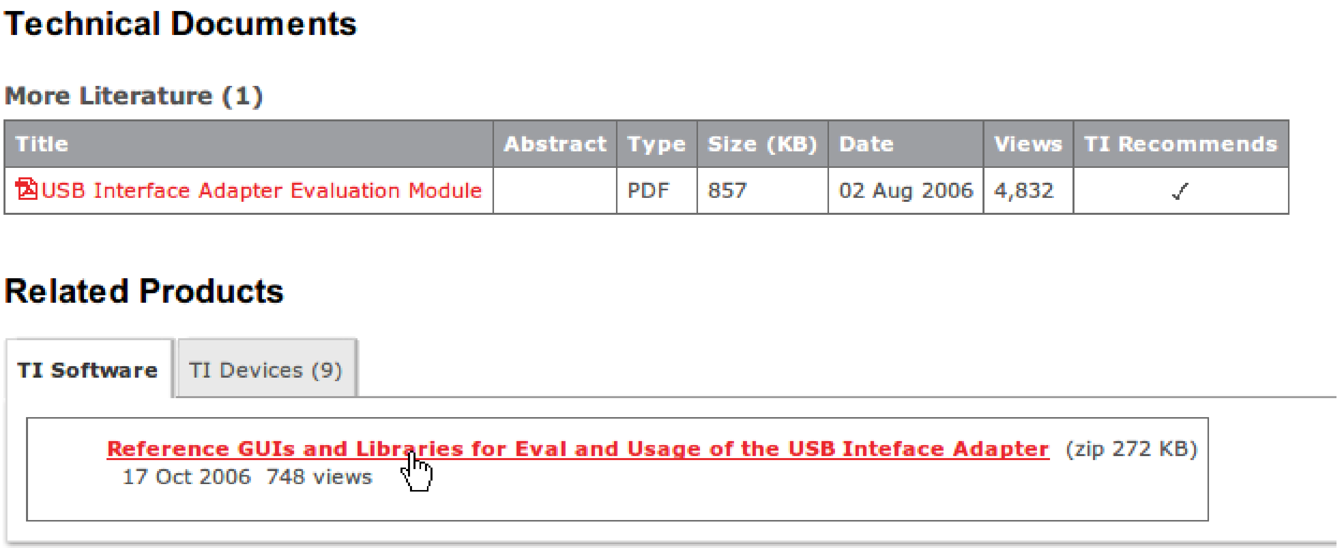

Figure 4. Tool Folder Web Page

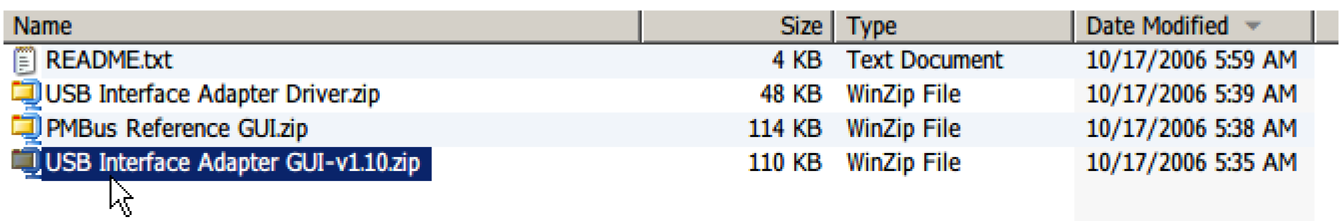

Figure 5. Selecting Required File from Sub-Archive Files

The user guide, found in the same tool folder (see upper half of Figure 4), explains this adaptor and software. Refer to the guide for further information.

4 System Setup and Operation from a PC

4.1 Interfacing the System Together and Start-Up GUI Program

Preparation is now complete and everything is properly connected. Figure 6 shows an example system setup (note that Figure 6 shows a highly-modified TPS7A7200EVM-718). Although not shown in Figure 6, the USB-TO-GPIO adaptor USB connection is going to a host computer. When everything is connected, power up the TPS7A7200EVM-718 board with 4.0 V from a lab supply. Do not forget to connect T1 of the I²C IO expander board to TP1 of the TPS7A7200EVM-718 board by a wire.

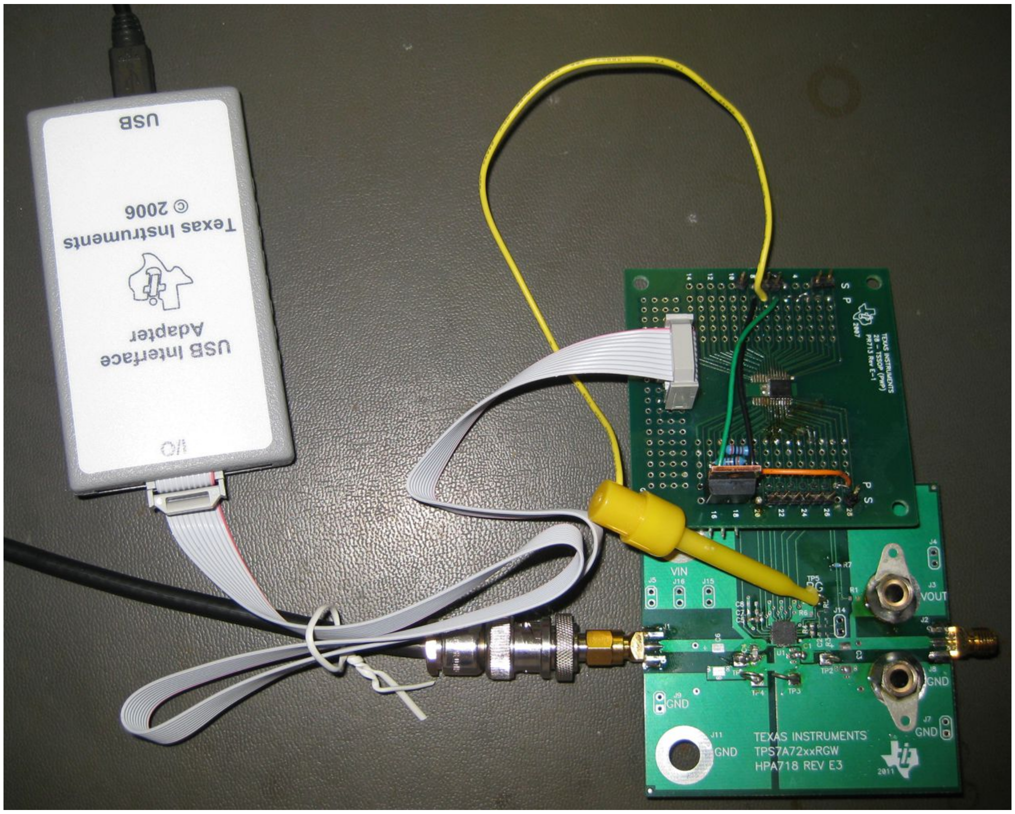

Figure 6. Total System Connection

The TPS7A7200EVM-718 board is now ready to be controlled from the PC. Open the *USB SAA GUI.exe* GUI software (see Figure 7). In this GUI screen, only the *I2C* and *Pull-up Resistors* control sections are required in this document (see Figure 8).

Name	Size	Type	Date Modified
USB SAA GUI.xml	1 KB	XML Document	10/17/2006 5:28 AM
USB SAA GUI.pdb	132 KB	PDB File	10/17/2006 5:28 AM
USB SAA GUI.exe	124 KB	Application	10/17/2006 5:28 AM
USB Adapter Driver.xml	1 KB	XML Document	10/10/2006 2:31 PM
USB Adapter Driver.pdb	196 KB	PDB File	10/10/2006 2:31 PM
USB Adapter Driver.dll	72 KB	Application Extensi...	10/10/2006 2:31 PM
USB SAA GUI.vshost.exe	6 KB	Application	9/23/2005 7:56 AM

Figure 7. Starting GUI Software

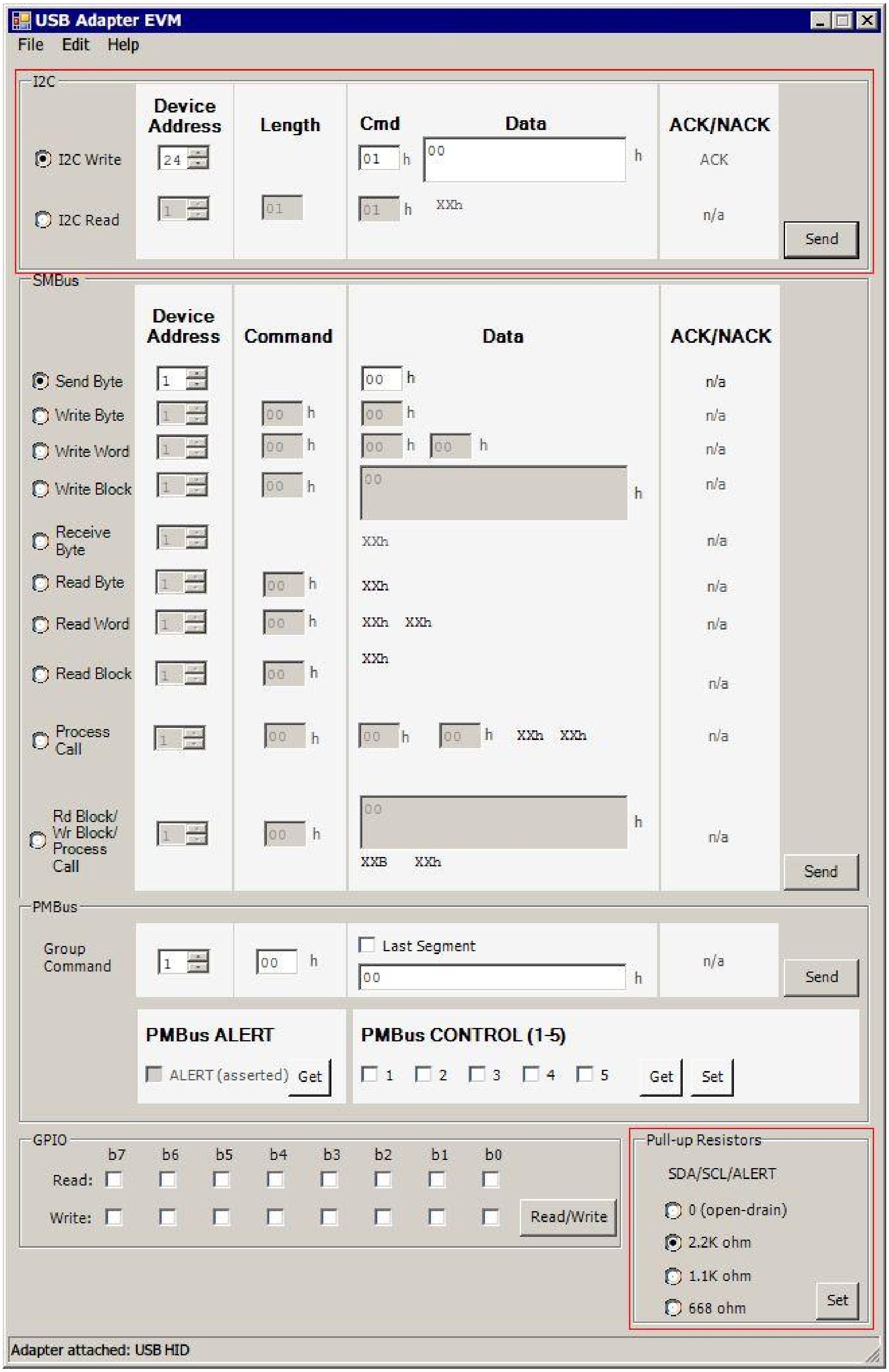

Figure 8. GUI Screen after Sending the First Line of Initialization Code

4.2 Initializing the PCA9557 and the Operation Strategy

The PCA9557 has four control registers. Initialize these registers from the *I2C Write* function of the GUI software by sending each line of Table 1. Input each line of Table 1 in the GUI and click the *Send* button in *I2C* section and repeat this process for all three lines. Figure 8 shows the result of sending the first line, Figure 9 shows the second line, and Figure 10 shows the third line. When the *Send* operation is successful, an ACK string should be displayed under the ACK/NACK column, as shown in Figure 8, Figure 9, and Figure 10. If ERROR is shown in the ACK/NACK column (see Figure 11), check the I^{2}C IO expander board and the connections. Because register (Cmd) 00h is reads values from the PCA9557, initialization is not needed. In Figure 1, the I^{2}C slave address bits are all connected to GND, which selects 24 (decimal) or 18 (hex) as the address when combined with the fixed address bits within the PCA9557 (see page 7 of the PCA9557 data sheet).

Table 1. PCA9557 Initialization Code

DEVICE ADDRESS	LENGTH	CMD	DATA
24	n/a	01h	00h
24	n/a	02h	00h
24	n/a	03h	FFh

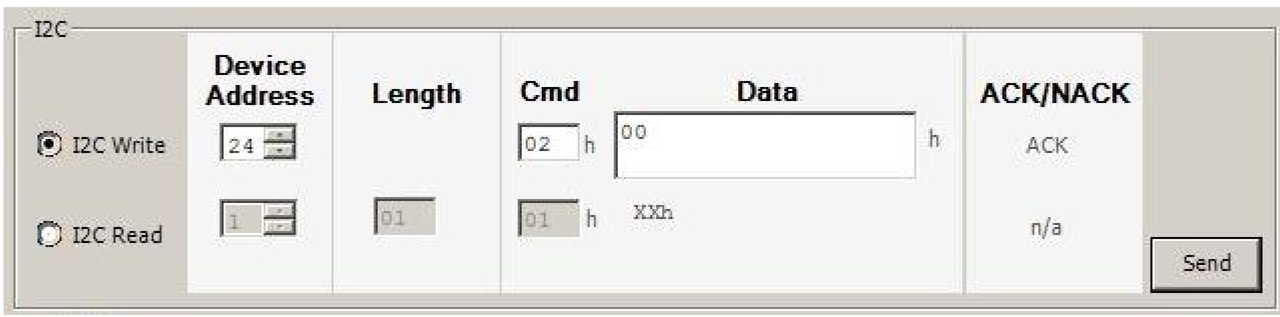

Figure 9. GUI Screen after Sending the Second Line of Initialization Code

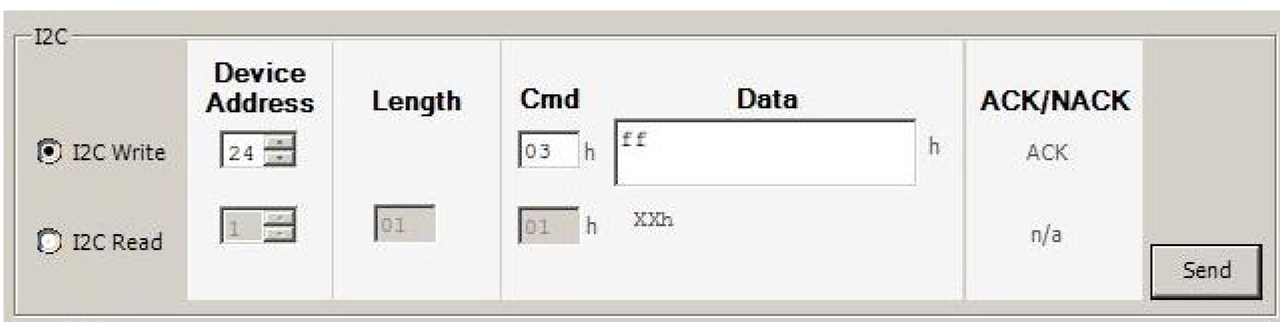

Figure 10. GUI Screen after Sending the Third Line of Initialization Code

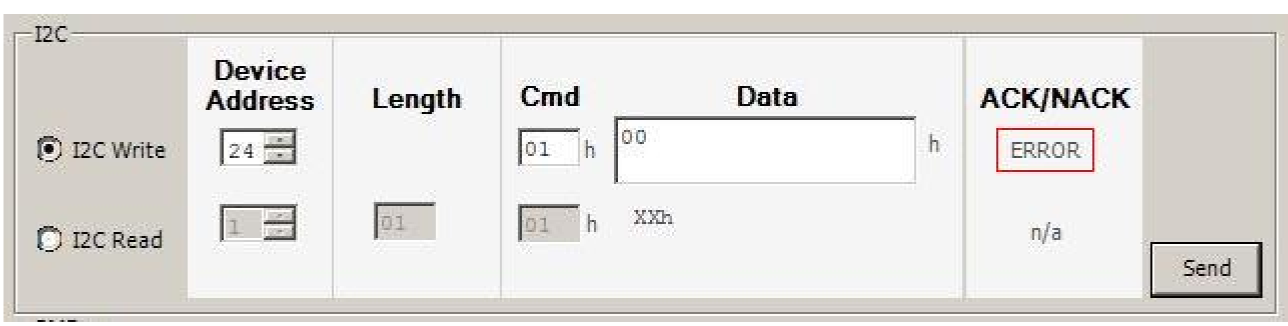

Figure 11. GUI Screen with Failure, after Sending the First Line of Initialization Code

To control the ANY-OUT voltage setting pins of the TPS7A7200EVM-718 at the J5 header, the IO pins (P0 to P5) of the PCA9557 are configured to be either ground or float via the GUI (see Figure 1). Also, the level shifter circuit driving EN requires selecting either ground or float. For this selection, an open-drain output is the most preferable output configuration. However, the PCA9557 device does not have open-drain outputs, except for the P0 terminal. To achieve open-drain equivalent effects at the P1 to P6 terminals, the following operation is used:

- The ground option can be implemented by using P1 to P6 as output terminals with a logic low setting. The low-side output transistor within the P1 to P6 pins drive these pins to GND. This setting is equivalent to an open-drain terminal in on-state.
- The float option can be implemented by using P1 to P6 as input terminals. With the PCA9557, P1 to P6 are configurable as inputs or outputs; when used as an input configuration, P1 to P6 become high impedance. This setting is equivalent to an open-drain terminal in high-impedance state.

With this operation strategy, an always logic low output is required when P1 to P6 are selected as outputs. Therefore, data 00h is written to the Output Port Register in the first line of Table 1. With the third line of Table 1, all IO pins are configured as inputs by writing FFh to the Configuration Register; thus, P1 to P6 become high impedance (floating). After the Table 1 initializes, the TPS7A7200 is disabled (EN = L) and the target output voltage becomes 0.5 V (see Table 1 of the TPS7A7200 data sheet). Afterwards, only toggle the Configuration Register bits (Cmd 03h) to achieve pseudo open-drain outputs at P1 to P6.

The meaning of the second initialization code is explained later in Section 4.4.

4.3 Operation Code Table

Table 2 describes the entire system operating code. As explained in Section 4.2, only changing the Configuration Register bits results in full system operation. Pick up the target output voltage value and send the code (refer to Figure 10 with data specified in Table 2), then the TPS7A7200 device follows the command. When an entire series of operations finishes, send the first line of Table 2 to disable the TPS7A7200 (this line is the same as the third line of Table 1).

Table 2. Operation Code Table

DEVICE ADDRESS	LENGTH	CMD	DATA	TARGET OUTPUT VOLTAGE
24	n/a	03h	FFh	(Disable the TPS7A7200)
24	n/a	03h	BFh	0.50 (V)
24	n/a	03h	BEh	0.55 (V)
24	n/a	03h	BDh	0.60 (V)
24	n/a	03h	BCh	0.65 (V)
24	n/a	03h	BBh	0.70 (V)
24	n/a	03h	BAh	0.75 (V)
24	n/a	03h	B9h	0.80 (V)
24	n/a	03h	B8h	0.85 (V)
24	n/a	03h	B7h	0.90 (V)
24	n/a	03h	B6h	0.95 (V)
24	n/a	03h	B5h	1.00 (V)
24	n/a	03h	B4h	1.05 (V)
24	n/a	03h	B3h	1.10 (V)
24	n/a	03h	B2h	1.15 (V)
24	n/a	03h	B1h	1.20 (V)
24	n/a	03h	B0h	1.25 (V)
24	n/a	03h	AFh	1.30 (V)
24	n/a	03h	AEh	1.35 (V)
24	n/a	03h	ADh	1.40 (V)
24	n/a	03h	ACh	1.45 (V)
24	n/a	03h	ABh	1.50 (V)
24	n/a	03h	AAh	1.55 (V)

Table 2. Operation Code Table (continued)

DEVICE ADDRESS	LENGTH	CMD	DATA	TARGET OUTPUT VOLTAGE
24	n/a	03h	A9h	1.60 (V)
24	n/a	03h	A8h	1.65 (V)
24	n/a	03h	A7h	1.70 (V)
24	n/a	03h	A6h	1.75 (V)
24	n/a	03h	A5h	1.80 (V)
24	n/a	03h	A4h	1.85 (V)
24	n/a	03h	A3h	1.90 (V)
24	n/a	03h	A2h	1.95 (V)
24	n/a	03h	A1h	2.00 (V)
24	n/a	03h	A0h	2.05 (V)
24	n/a	03h	9Fh	2.10 (V)
24	n/a	03h	9Eh	2.15 (V)
24	n/a	03h	9Dh	2.20 (V)
24	n/a	03h	9Ch	2.25 (V)
24	n/a	03h	9Bh	2.30 (V)
24	n/a	03h	9Ah	2.35 (V)
24	n/a	03h	99h	2.40 (V)
24	n/a	03h	98h	2.45 (V)
24	n/a	03h	97h	2.50 (V)
24	n/a	03h	96h	2.55 (V)
24	n/a	03h	95h	2.60 (V)
24	n/a	03h	94h	2.65 (V)
24	n/a	03h	93h	2.70 (V)
24	n/a	03h	92h	2.75 (V)
24	n/a	03h	91h	2.80 (V)
24	n/a	03h	90h	2.85 (V)
24	n/a	03h	8Fh	2.90 (V)
24	n/a	03h	8Eh	2.95 (V)
24	n/a	03h	8Dh	3.00 (V)
24	n/a	03h	8Ch	3.05 (V)
24	n/a	03h	8Bh	3.10 (V)
24	n/a	03h	8Ah	3.15 (V)
24	n/a	03h	89h	3.20 (V)
24	n/a	03h	88h	3.25 (V)
24	n/a	03h	87h	3.30 (V)
24	n/a	03h	86h	3.35 (V)
24	n/a	03h	85h	3.40 (V)
24	n/a	03h	84h	3.45 (V)
24	n/a	03h	83h	3.50 (V)
24	n/a	03h	82h	3.55 (V)
24	n/a	03h	81h	3.60 (V)
24	n/a	03h	80h	3.65 (V)

4.4 Reading the PG Pin Status

This system can read the TPS7A7200 PG pin status through P7 of the PCA9557 IO pin. This means that the system supports a complete LDO remote control and monitor.

To monitor the PG status, use the I^{2}C read function of the GUI software by sending the code in Table 3. The read value is displayed under the data column. The 8-bit data MSB shows the PG status. Figure 12 shows the returned data (40h), which means the MSB is PG = L (logic low); this result is captured after sending the disable code (Cmd = 03h and data = FFh). Figure 13 shows the returned data (80h)), which means the MSB is PG = H (logic high); this result is captured after sending the enable with 3.3-V output code (Cmd = 03h and data = 87h).

With the second initialization code in Table 1, all input bits are configured to return positive logic.

In the Figure 12 condition, all IO P7 to P0 bits are configured as input ports. At P5 to P0, the PCA9557 device reads digital data bits as all logic low from the 0.5-V analog signal, which is the TPS7A7200 device reference voltage. At P6, a pull-up resistor (R1) gives a logic high. At P7, the TPS7A7200 outputs a not-power-good signal that is logic low. Thus, the resulting read value is 40h.

In the Figure 13 condition, P2 to P0 are configured as input ports and, as previously described, the read values are logic low. P6 to P3 are configured as output ports and have a logic low level, meaning that the reading values are all logic low. At P7, assuming the TPS7A7200 board is working normally, the LDO outputs a power-good signal that is logic high. Thus, the resulting reading value is 80h.

Table 3. PCA9557 Initialization Code

Device Address	Length	Cmd	Data
24	1	00 h	n/a

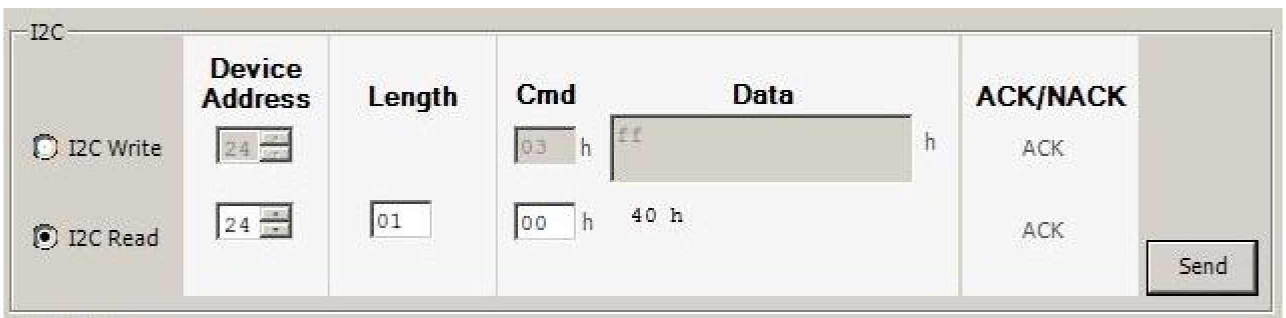

Figure 12. GUI Screen Reading the PG = L Status After Sending Disable Code

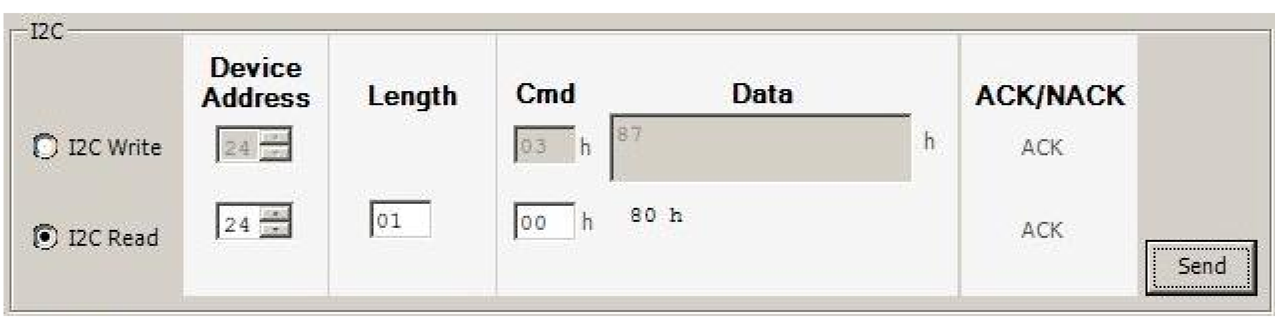

Figure 13. GUI Screen Reading the PG = H Status After Sending Enable (3.3V Output) Code

5 Summary

An ANY-OUT LDO TPS7A7200 can be controlled by the I^2C interface in combination with the I^2C IO expander PCA9557 device. The controller monitors the device PG status through the I^2C connection in addition to controlling the LDO. This example details the capability of an ANY-OUT LDO in digital remote control and monitor demands. Even though the TPS7A7200 does not support I^2C, the example circuit in Figure 1 with the PCA9557 device can upgrade the LDO to an I^2C-compatible solution. Texas Instruments is in the process of releasing more ANY-OUT LDOs.

6 Appendix: List of Released ANY-OUT LDO (as of May 23, 2012)

- TPS7A7100, 1A LDO
- TPS7A7200, 2A LDO
- TPS7A7300, 3A LDO

IMPORTANT NOTICE

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
		TI E2E Community Home Page	e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265