Applicability:

ADLD25PC, ADLN2000PC, ADLQM67PC, ADLQM87PC, ADLE3800PC and any other future boards using NXP PCA9535 GPIO controller.

Background:

16bit GPIO on ADL Embedded Solutions' CPU boards is accomplished through the use of the NXP PCA9535 GPIO controller. Specifically the PCA9535BS package. The PCA9535 is an I2C/SMBus device attached directly to the PCH chipset of the relevant CPU board and is controlled via the Intel SMBus registers. Refer to the appropriate Intel chipset datasheet for register detail.

The **SMBus I/O base address** for current CPU boards is as follows:

ADL Board	SMBus I/O Base Address
ADLD25PC	0500h
ADLN2000PC	1180h
ADLQM67PC	F040h
ADLQM87PC	F040h
ADLE3800PC	E000h

Key register **offset addresses** are as follows:
04h = Holds PCA9535 Slave address. Enter 40h for the PCA9535 on ADL CPU boards.
03h = PCA9535 command byte pointers (see pg. 6 of the PCA9535 datasheet)
05h = 1st data byte
06h = 2nd data byte
02 = Trigger SMBus transaction

Reference Documents:

Intel SMBus Register section of the appropriate Intel chipset datasheet. The Intel SMBus registers are consistent across chipsets, so if you have one you have them all. For example, see section 18.2 of the Intel QM67 datasheet: http://www.intel.com/content/www/us/en/chipsets/6-chipset-c200-chipset-datasheet.html

NXP PCA9535BS datasheet. Refer to command Byte information on page 6.
http://www.nxp.com/documents/data_sheet/PCA9535_PCA9535C.pdf

ADL Embedded Solutions GmbH, Eiserfelder Straße 316 D-57080 Siegen, Germany P. +49 (0) 271 250 810 0 F. +49 (0) 271 250 810 20
e-mail: sales@adl-europe.com; web: www.adl-europe.com

ADL Embedded Solutions Inc., 4411 Morena Blvd., Suite 101 San Diego, CA 92117-4345 P. +1 858 490-0597 F. +1 858 490-0599
e-mail: sales@adl-usa.com; web: www.adl-usa.com

Pseudo-Code for GPIO Write/Read Using DOS Debug assuming SMBus I/O base address = F040h:

Writing To GPIO Ports:

-iF040 ; Query SMBus host controller status
00 ; if the 1st byte is not zero, clear it by writing the same hex number back into this address.

-oF044,40 ; This is the PCA9535BS slave address
-oF043,06 ; Point to the PCA9535BS Config Port via Command Byte. See page 6 of datasheet.

-oF045,00 ; Configure Port 0 as all outputs by writing all 0's
-oF046,00 ; Configure Port 1 as all outputs by writing all 0's

-oF042,4C ; Trigger the SMBus transaction

-iF040 ; Query the SMBus Status
42 ; A 1 in the 2nd bit location indicates transaction completed and interrupt generated. Note that the interrupt is not hooked up on the ADLQM67/87 but it does need to be cleared before any other SMBus transactions can be initiated.

-oF040,2 ; Clear the interrupt bit

-oF043,02 ; Point to PCA9535BS Output Port 0 via command byte. Note that all Port 0 and Port 1 byte registers (input, output, config) work in pairs. If you write a WORD to the either Port register, the overflow automatically gets written to the other Port register.

-oF045,FF ; Setup all 1's to Port 0
-oF046,FF ; Setup all 1's to Port 1

-oF042,4C ; Trigger the transaction. You should see all 1's on all 16 bits of the GPIO

-iF040 ; Read the Host Status
42

-oF040,2 ; Clear the interrupt bit

ADL Embedded Solutions GmbH, Eiserfelder Straße 316 D-57080 Siegen, Germany P. +49 (0) 271 250 810 0 F. +49 (0) 271 250 810 20
e-mail: sales@adl-europe.com; web: www.adl-europe.com

ADL Embedded Solutions Inc., 4411 Morena Blvd., Suite 101 San Diego, CA 92117-4345 P. +1 858 490-0597 F. +1 858 490-0599
e-mail: sales@adl-usa.com; web: www.adl-usa.com

Reading From GPIO Ports:

Note: Slave read address = 41. See page 6 of PCA0535BS datasheet.

Example:

```
-oF044,41        ; This is the PCA9535BS slave address for read-commands
-oF042,4C        ; Trigger the SMBus transaction
-iF045           ; Read Port 2 data
-iF046           ; Read Port 1 data
```

Note: The above Debug code works just fine if you manually type it in because then the pause after "Trigger" is long enough. For free-running C-code, you must examine the SMBus status busy bit to assure the transaction is complete.

Windows and Linux Drivers:

32bit/64bit Linux and 64bit Windows drivers exist for the ADLQM67PC which have been developed by one of our clients. Unfortunately, these drivers are platform specific and do not work on other platforms. Contact ADL Technical Support for details, support@adl-usa.com.

For Linux environments, some engineers have successfully used Linux ioperm, outb and inb C code. Other utilities may be available, but we don't have any specific recommendations. Sample code may be available through support@adl-usa.com. We would appreciate any feedback to support@adl-usa.com if someone has other ways of manipulating GPIO in Linux environments.

JC Ramirez
Sept. 19, 2014

ADL Embedded Solutions GmbH, Eiserfelder Straße 316 D-57080 Siegen, Germany P. +49 (0) 271 250 810 0 F. +49 (0) 271 250 810 20
e-mail: sales@adl-europe.com; web: www.adl-europe.com

ADL Embedded Solutions Inc., 4411 Morena Blvd., Suite 101 San Diego, CA 92117-4345 P. +1 858 490-0597 F. +1 858 490-0599
e-mail: sales@adl-usa.com; web: www.adl-usa.com