5.0 Data Flash Programming Protocol

The PowerLAN Controller features a unique Data Flash Programming (DFP) protocol that leverages the well known transactions of SMBus to provide an efficient method of data transfer. The DFP can be used to configure the PowerLAN Controller without the use of the bqWizard application or API.

5.1 Notation for SMBUS transactions

This document provides many examples of actual SMBUS transactions. The transactions are displayed in the following format.

#	S/P	Addr	Read/Write	Data	PEC
1	SP	0B	Write Transaction	00 1A C0	88
2	S	0B	Write Transaction	0	
3	SP	0B	Read Transaction	00 14	A1

The examples assume a SMBUS base address of 0x0B. Note that a typical read operation is shown as two separate parts. The first part has the command and the second part the actual read. This is also indicated by the S/P column which shows Start (S) and (SP) Stop signals. The reads will have a repeated Start. Also note that although the base address is 0x0B, the actual address sent over SMBUS will actually be 0x16 or 0x17 depending on the transaction type (read or write). The example above demonstrates writing 0xC01A to Manufacturer Access() (0x00) followed by a reading back 0x1400 from the same address. Values are shown in hexadecimal and the PEC is as defined by the SMBUS standard.

5.2 Enabling DFP Access

In order to utilize the DFP, the part must be placed in the DFP communications mode. This is accomplished by read/writes to the SBDATA ManufactureAccess() (0x00) command. The Procedure is as follows

Unlock the pack by writing 0x44BB to 0x00

Check system flags to ensure pack is unlocked and application is ok. This is done by writing 0xC01A to 0x00 followed by a read of 0x00. Bits 12 and 10 should be set indicating the application is running correctly with a valid CRC, bit 9 should be cleared indicating the pack is unlocked. Ignore other bits.

System Flags

Bit	7	6	5	4	3	2	1	0
Contents								
Bit	15	14	13	12	11	10	9	8
Contents				App Good		CRC Good	Pack Locked	

1. Place part in DFP mode by writing 0x55AA to 0x00.

2. Read 0x00 and verify response of 0x55AA. If the response is not 0x55AA the operation has failed, and the steps must be repeated.

Example:

#	S/P	Addr	Record	Data	PEC
1	SP	0B	Write Transaction	00 BB 44	10
2	SP	0B	Write Transaction	00 1A C0	88
3	S	0B	Write Transaction	0	
4	SP	0B	Read Transaction	00 14	A1
5	SP	0B	Write Transaction	00 AA 55	25
6	S	0B	Write Transaction	0	
7	SP	0B	Read Transaction	AA 55	FB

In this example, the write of 0x44BB to the Manufacturer Access location in line 1 is used to unlock the pack. This is followed in line 2 by writing 0xC01A to the same location as a request to read the System Flags. This is read in lines 3 and 4. The response of 0x1400 has the bits set indicating that the Application and data CRCs are correct and shows that the pack is unlocked. In line 5, the write of 0x55AA enables the file access format and this is verified by the read in lines 6 and 7.

Note: After approximately 1 minute of inactivity, the part will automatically revert back to standard SBDATA mode. The part can be held in DFP mode during long periods of inactivity by sending a heartbeat command by writing 0x55AA to 0x00.

5.3 DFP Operation

Once DFP mode is enabled, four commands (**Request**, **Status**, **Read**, & **Write**) are used for communications. The procedure for a standard transaction is as follows.

1. A **Request** command is sent to the part seeking permission for the transaction.
2. The **Status** command is used to determine that the request is permitted.
3. Upon acceptance of the request, a **Read** or **Write** command is issued.

Each of these commands will be described below with examples in the next section.

Request (0x3C)

The request command is used to initiate and request permission for a transfer. It is a write with the following elements:

Addr - Pack address. In all the examples, this will be 0x0B and since this is a write, it will appear on the bus as 0x17. That is, the address shifted up is 0x16 and this becomes 0x17 with the addition of the write bit.

CMD - Command code. For the Request command, this will always be 0x3C

LEN- Length of request.

R/W - Read/Write flag. This will be a 0 for a read request and a 1 for write request

File - This is the internal file number.

Start – this is the starting element (chunk) number.

Count – this is the number of elements (chunks) to be read or written.

Transaction Format:

Byte	1	2	3	4	5	6	7	8	9
Contents	Addr	CMD	LEN	R/W	File		Start		Count

Status (0x3D)

The status command is used to determine the system status and to determine if a request has been granted. It is a write of the command followed by a read with the following elements

Addr - Pack address.

CMD - Command code. For the Status command, this will always be **0x3D**

Response – This is a 16 bit reply from the PL116. The possible meanings are shown in the table below. Non published response should be considered an error.

Status	Meaning
0x0000	OK – The access requested is granted. The last command passed
0x00FF	Busy – the system is busy with a prior command
0x0001	The access level must be 1 or higher to permit the request
0x0002	The access level must be 2 or higher to permit the request
0x0003	The access level must be 3 or higher to permit the request
0x0004	The file number requested does not exist
0x0005	The request cannot be performed
0x0006	The size of the request is too large
0x0007	Firmware download failed
0x0008	Firmware download successful
0x0009	The start address of the request is invalid
0x000A	An illegal command was received

Transaction Format:

Byte	1	2	3	4
Contents	Addr	CMD	Response	

Note: When a BUSY 0x00FF response is received, it indicates the part is busy processing the previous request. Continue to issue Status commands until a response of 0x0000 is received.

Write (0x3E)

The Write command is a write with the following elements:

Addr - Pack address.

CMD - Command code**. For the Status command, this will always be 0x3E**

Count- Length of data to follow in bytes.

Data – This is the data being sent to the system.

Transaction Format:

Byte	1	2	3	4…..N
Contents	Addr	CMD	Count	Data…

Read (0x3F)

The Read command is a write followed by a read with the following elements:

Addr - Pack address.

CMD - Command code **For the Status command, this will always be 0x3F**

Count – This is a byte count of the number of bytes to be read

Data – The data sent back from the system

Transaction Format:

Byte	1	2	3	4…..N
Contents	Addr	CMD	Count	Data…

6.0 Setting the Wired Bit Using DFP

By definition, The Wired bit should be set when the pack is assembled and it is safe to access the bq78PL11x's data flash memory. If the Wired bit is not set then the bq78PL11x inhibits writes to data flash memory. Since the Wired bit itself resides in data flash memory it cannot be written directly if it's Unwired. We need to first tell the device that we intend to set the bit; this is the first step of the process. Once the command is sent requesting to enable writing of the Wired bit, we must then immediately follow with the actual write to set the Wired bit; this is the second step of the process.

Enabling Wired bit access

This part assumes that you have already enabled DFP. Note: If the contents of the 'Algorithm Enable' register (which contains the Wired bit information) is unknown, we must first read this register to ensure we preserve the other bits.

Optional - Read Algorithm Enable register

#	S/P	Addr	Record	Data	PEC
1	SP	0B	Write Transaction	3C 06 00 09 49 00 01 00	xx
2	S	0B	Write Transaction	3D	

3	SP	0B	Read Transaction	00 00	9A
4	S	0B	Write Transaction	3F	xx
5	SP	0B	Read Transaction	02 06 80	xx

In line 1 a **Request** is sent to read 0x0001 chunk from file 0x09 starting at chunk 0x0049 which is the location or this particular register. This is followed by a **Status** read in line 2 and 3 in which permission to read the data is verified by a 0x0000 response from the part. Lines 4 & 5 comprise the **Read** request. The read is a comprised of a write of the 0x3F command, followed by a read of the length and data. In this example the value of Algorithm Enable is 0x8006. *Note: Data is read and written in LSB – MSB*

Step 1 - Sending the command to request data flash write access

#	S/P	Addr	Record	Data	PEC
1	SP	0B	Write Transaction	3C 06 01 1B 00 00 02 00	xx
2	S	0B	Write Transaction	3D	
3	SP	0B	Read Transaction	00 00	9A
4	SP	0B	Write Transaction	3E 04 0D 00 00 00	xx
5	S	0B	Write Transaction	3D	xx
6	SP	0B	Read Transaction	00 00	9A

In line 1 a **Request** is sent to write 0x0002 chunks to file 0x1B starting at chunk 0x0000. This is followed by a **Status** read in line 2 and 3 in which permission to write the data is verified by a 0x0000 response from the part. Line 4 is the **Write** of 0x04 bytes which contain the command data. Lines 5 & 6 is another **Status** read to verify the write was successful.

Step 2 - Writing Algorithm Enable register to enable the Wired bit

#	S/P	Addr	Record	Data	PEC
1	SP	0B	Write Transaction	3C 06 01 09 49 00 01 00	xx
2	S	0B	Write Transaction	3D	
3	SP	0B	Read Transaction	00 00	9A
4	SP	0B	Write Transaction	3E 02 0E 80	xx
5	S	0B	Write Transaction	3D	xx
6	SP	0B	Read Transaction	00 00	9A

In line 1 a **Request** is sent to write 0x0001 chunk to file 0x09 starting at chunk 0x0049. This is followed by a **Status** read in line 2 and 3 in which permission to write the data is verified by a 0x0000 response from the part. Line 4 is the **Write** of 0x02 bytes which contain the new value of Algorithm Enable with Wired set. In this example the new value is 0x800E (See Ex. 1). Lines 5 & 6 is another **Status** read to verify the write was successful.

Ex. 1
const WIRED_BIT = 0x8
Algorithm Enable = Algorithm Enable | WIRED_BIT
0x800E = 0x8006 | 0x08