

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Pentek 6509* at our website: *Click HERE*

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.7 Graychip Control Registers (continued)

Table 3-16: Model 6509 - Graychip Registers - VMEbus Addressing (R/W @ VME_base+. . .)

Register	Rec. #1	Rec. #2	Rec. #3	Rec. #4	Rec. #5	Rec. #6	Rec. #7	Rec. #8	All Recs*
FB0	0x000B 8000	0x000B 8040	0x000B 8080	0x000B 80C0	0x000B 8100	0x000B 8140	0x000B 8180	0x000B 81C0	0x000B 8200
FB1	0x000B 8004	0x000B 8044	0x000B 8084	0x000B 80C4	0x000B 8104	0x000B 8144	0x000B 8184	0x000B 81C4	0x000B 8204
FB2	0x000B 8008	0x000B 8048	0x000B 8088	0x000B 80C8	0x000B 8108	0x000B 8148	0x000B 8188	0x000B 81C8	0x000B 8208
FB3	0x000B 800C	0x000B 804C	0x000B 808C	0x000B 80CC	0x000B 810C	0x000B 814C	0x000B 818C	0x000B 81CC	0x000B 820C
SMR	0x000B 8010	0x000B 8050	0x000B 8090	0x000B 80D0	0x000B 8110	0x000B 8150	0x000B 8190	0x000B 81D0	0x000B 8210
OMR	0x000B 8014	0x000B 8054	0x000B 8094	0x000B 80D4	0x000B 8114	0x000B 8154	0x000B 8194	0x000B 81D4	0x000B 8214
FMR	0x000B 8018	0x000B 8058	0x000B 8098	0x000B 80D8	0x000B 8118	0x000B 8158	0x000B 8198	0x000B 81D8	0x000B 8218
GCR	0x000B 801C	0x000B 805C	0x000B 809C	0x000B 80DC	0x000B 811C	0x000B 815C	0x000B 819C	0x000B 81DC	0x000B 821C
DC0	0x000B 8020	0x000B 8060	0x000B 80A0	0x000B 80E0	0x000B 8120	0x000B 8160	0x000B 81A0	0x000B 81E0	0x000B 8220
DC1	0x000B 8024	0x000B 8064	0x000B 80A4	0x000B 80E4	0x000B 8124	0x000B 8164	0x000B 81A4	0x000B 81E4	0x000B 8224
OSR	0x000B 8028	0x000B 8068	0x000B 80A8	0x000B 80E8	0x000B 8128	0x000B 8168	0x000B 81A8	0x000B 81E8	0x000B 8228
CSR	0x000B 802C	0x000B 806C	0x000B 80AC	0x000B 80EC	0x000B 812C	0x000B 816C	0x000B 81AC	0x000B 81EC	0x000B 822C
IO0	0x000B 8030	0x000B 8070	0x000B 80B0	0x000B 80F0	0x000B 8130	0x000B 8170	0x000B 81B0	0x000B 81F0	0x000B 8230
IO1	0x000B 8034	0x000B 8074	0x000B 80B4	0x000B 80F4	0x000B 8134	0x000B 8174	0x000B 81B4	0x000B 81F4	0x000B 8234
QO0	0x000B 8038	0x000B 8078	0x000B 80B8	0x000B 80F8	0x000B 8138	0x000B 8178	0x000B 81B8	0x000B 81F8	0x000B 8238
QO1	0x000B 803C	0x000B 807C	0x000B 80BC	0x000B 80FC	0x000B 813C	0x000B 817C	0x000B 81BC	0x000B 81FC	0x000B 823C
These addresses are Write Only									

Table 3-17: Model 6509 - Graychip Registers - ëC40 Addressing (R/W @ . . .)

Register	Rec. #1	Rec. #2	Rec. #3	Rec. #4	Rec. #5	Rec. #6	Rec. #7	Rec. #8	All Recs*
FB0	0x8002 E000	0x8002 E010	0x8002 E020	0x8002 E030	0x8002 E040	0x8002 E050	0x8002 E060	0x8002 E070	0x8002 E080
FB1	0x8002 E001	0x8002 E011	0x8002 E021	0x8002 E031	0x8002 E041	0x8002 E051	0x8002 E061	0x8002 E071	0x8002 E081
FB2	0x8002 E002	0x8002 E012	0x8002 E022	0x8002 E032	0x8002 E042	0x8002 E052	0x8002 E062	0x8002 E072	0x8002 E082
FB3	0x8002 E003	0x8002 E013	0x8002 E023	0x8002 E033	0x8002 E043	0x8002 E053	0x8002 E063	0x8002 E073	0x8002 E083
SMR	0x8002 E004	0x8002 E014	0x8002 E024	0x8002 E034	0x8002 E044	0x8002 E054	0x8002 E064	0x8002 E074	0x8002 E084
OMR	0x8002 E005	0x8002 E015	0x8002 E025	0x8002 E035	0x8002 E045	0x8002 E055	0x8002 E065	0x8002 E075	0x8002 E085
FMR	0x8002 E006	0x8002 E016	0x8002 E026	0x8002 E036	0x8002 E046	0x8002 E056	0x8002 E066	0x8002 E076	0x8002 E086
GCR	0x8002 E007	0x8002 E017	0x8002 E027	0x8002 E037	0x8002 E047	0x8002 E057	0x8002 E067	0x8002 E077	0x8002 E087
DC0	0x8002 E008	0x8002 E018	0x8002 E028	0x8002 E038	0x8002 E048	0x8002 E058	0x8002 E068	0x8002 E078	0x8002 E088
DC1	0x8002 E009	0x8002 E019	0x8002 E029	0x8002 E039	0x8002 E049	0x8002 E059	0x8002 E069	0x8002 E079	0x8002 E089
OSR	0x8002 E00A	0x8002 E01A	0x8002 E02A	0x8002 E03A	0x8002 E04A	0x8002 E05A	0x8002 E06A	0x8002 E07A	0x8002 E08A
CSR	0x8002 E00B	0x8002 E01B	0x8002 E02B	0x8002 E03B	0x8002 E04B	0x8002 E05B	0x8002 E06B	0x8002 E07B	0x8002 E08B
IO0	0x8002 E00C	0x8002 E01C	0x8002 E02C	0x8002 E03C	0x8002 E04C	0x8002 E05C	0x8002 E06C	0x8002 E07C	0x8002 E08C
IO1	0x8002 E00D	0x8002 E01D	0x8002 E02D	0x8002 E03D	0x8002 E04D	0x8002 E05D	0x8002 E06D	0x8002 E07D	0x8002 E08D
QO0	0x8002 E00E	0x8002 E01E	0x8002 E02E	0x8002 E03E	0x8002 E04E	0x8002 E05E	0x8002 E06E	0x8002 E07E	0x8002 E08E
QO1	0x8002 E00F	0x8002 E01F	0x8002 E02F	0x8002 E03F	0x8002 E04F	0x8002 E05F	0x8002 E06F	0x8002 E07F	0x8002 E08F
These addresses are Write Only									

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.8 Input Channel Control Registers - R/W @ Mode_Ch_A_Ctl and Tag_En_Ch_B_Ctl (D7 - D0)

The input channel control functions in the Model 6509 are divided between two registers. The lower eight bits of the Mode Select / Channel A Control Register provides input control functions for Input Channel A, and the lower eight bits of the Tag Enable / Channel B Control Register provide the identical functions for Input Channel B. The bit structure of the lower eight bits of these two registers is identical. The functions of the Input Channel control bits in each of these registers will be discussed in the subsections below. Both of these registers are accessible to both the 'C40 processor and VMEbus masters. Table 3-19 and Table 3-20, which define the bit structure of both registers, will follow these bit descriptions (see the page after next), and the subsequent sections will describe the functions of the upper eight bits of these two registers.

The design of the ECL SYNC drivers used in the Model 6509 requires the use of jumpers JB1 through JB4 to disable the Master operation of SYNC (SS) and Accumulator SYNC (AS) on the input channel. Disabling the Master operation allows the selected input channel to function properly as a sync slave. The associated input channel Master SYNC terminators should also be removed from their sockets to avoid double termination. As shown in Figure 2-3, resistor network holders are provided on the mezzanine board. These holders serve only as place holders and do not affect the operation of the termination. The table below shows the placement of the jumpers.

Table 3-18: Model 6509 - Sync Mode Jumper Settings

Jumper	Setting	Description
Input Channel A is the Master Sync		
JB1	1-2	Enabled
JB2	1-2	Enabled
JB3	2-3	Disabled
JB4	2-3	Disabled
RN25	Remove from board	RN26 provides termination
Input Channel B is the Master Sync		
JB1	2-3	Disabled
JB2	2-3	Disabled
JB3	1-2	Enabled
JB4	1-2	Enabled
RN26	Removed from board	RN25 provides termination

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.8 Input Channel Control Registers (continued)

3.5.8.1 User (OVLD) LED Bits - D7

These bits control the state of the Front Panel OVLD (red) LED for each input channel. Setting one of these bits to the logic '1' state will turn the OVLD LED for the associated channel ON. Clearing the bit to the logic '0' state turns the associated LED OFF. These LEDs were initially intended as indicators for the threshold detector function (which explains the front panel designation) but are now completely under software control and may be lit at the programmer's discretion.

3.5.8.2 Sync Output Enable Bits - D3

When one of these bits is set to the logic '1' state, the AS and SS lines on the corresponding channel's input connector will function as outputs. Clear this bit in the associated channel for these lines to function as inputs.

These bits, in tandem with the Sync Input Source Bits (see below) allow multiple channels to be synchronized simultaneously in a Master/Slave scenario. Program the "master" channel's source bit for Front Panel (external hardware) sync, and enable the sync output for that channel. The "slave" channels are programmed for external (hardware) sync and for the sync lines as inputs. The Sync lines for all channels used in this manner must be externally connected. When the "master" channel's Software Sync Bits (see Section 3.5.8.4) are toggled, all the connected "slave" channels will be synchronized to the "master".

3.5.8.3 Sync Input Source Bits - D2

These bits determine whether the SS and AS signals for the associated channel come from an external, hardware source (via the Front Panel Input Connector) or are generated internally, in software. To select the Software (internal) path for the sync signals in a given channel, set the associated bit to the logic '1' state. When the Software path is selected, the Software SS bit (see Section 3.5.8.4, on the next page) for a given channel is toggled to reset the digital filters in that channel's receivers. The Software AS bit (see Section 3.5.8.5) is toggled, in this configuration, to reset the phase accumulators on that channel's receivers. Clear the associated bit to the logic '0' state to select the Front Panel (external hardware) sync paths for a given channel's receivers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.8 Input Channel Control Registers (continued)

3.5.8.4 Software System Sync (SS) Bits - D1

When the Sync Source bit (see Section 3.5.8.3, above) for a given channel is set to the logic '1' state (i. e., when the software sync path is selected), and this bit is cleared to the logic '0' state, the digital filter sections in all receivers driven by the associated input channel will be cleared and initialized. When software synchronization is selected, this bit must be in the logic '1' state for the receivers driven by the associated channel to acquire and down-convert data. The software reset function should be implemented by toggling this bit (i. e., clear this bit to '0', then set it back to '1').

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.8 Input Channel Control Registers (continued)

3.5.8.5 Software Accumulator Sync (AS) Bits - D0

When the Sync Source Bit for a given channel (see Section 3.5.8.3, on page 53) selects the Software path for the sync signals, this bit is used as the AS signal, which clears and initializes the phase accumulators in the receiver chips driven by the channel of interest. When the Software Sync is selected, these bits must be in the logic '1' state for a given channel's receivers to acquire and down-convert data. To synchronize a channel's receivers in this manner, toggle this bit for the given channel (i. e., clear the bit to the logic '0' state, then set it back to the logic '1' state).

NOTE: A bit in the Graychip's Sync Mode register allows the SS signal to perform both the system and accumulator sync functions. See Appendix B for further details.

Table 3-19: Model 6509 - Mode Select / Channel A Control Register - R/W @ Mode_Ch_A_Ctl

Bit #	D15	D14	D13	D12	D11	D10	D9	D8
Bit Name	Receiver_8 _Mode	Receiver_7 _Mode	Receiver_6 _Mode	Receiver_5 _Mode	Receiver_4 _Mode	Receiver_3 _Mode	Receiver_2 _Mode	Receiver_1 _Mode
Bit Function	1 = Real Only 0 = Complex							
Bit #	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Bit Name	User_ OVLD_LED	Not Used			Sync_Out _Enable	Sync_In_ Source	Software _SS	Software _AS
Bit Function	1 = ON 0 = OFF	Not Applicable			1 = Enabled 0 = Disabled	1 = Software 0=Front Panel	1=Not Asserted 0 = Asserted	1=Not Asserted 0 = Asserted

Table 3-20: Model 6509 - Tag Enable / Channel B Control Register - R/W @ Tag_En_Ch_B_Ctl

Bit #	D15	D14	D13	D12	D11	D10	D9	D8
Bit Name	Receiver_8 _Tag_Enable	Receiver_7 _Tag_Enable	Receiver_6 _Tag_Enable	Receiver_5 _Tag_Enable	Receiver_4 _Tag_Enable	Receiver_3 _Tag_Enable	Receiver_2 _Tag_Enable	Receiver_1 _Tag_Enable
Bit Function	1 = Tag on LSB 0 = Data on LSB							
Bit #	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Bit Name	User_ OVLD_LED	Not Used			Sync_Out _Enable	Sync_In_ Source	Software _SS	Software _AS
Bit Function	1 = ON 0 = OFF	Not Applicable			1 = Enabled 0 = Disabled	1 = Software 0=Front Panel	1=Not Asserted 0 = Asserted	1=Not Asserted 0 = Asserted

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.9 Receiver Mode Select Register - R/W @ Mode_Ch_A_Ctl (D15 - D8)

The programming of the upper eight bits of the Mode Select / Channel A Control Register depends upon how the internal Control Registers within the Graychip receivers are programmed. If a receiver is programmed to operate in Real-Only Mode, the corresponding bit in this field must be set to the logic '1' state, to allow the receiver's FIFO to accept data only from the 'I' output. If the receiver is operating in Complex Mode, The corresponding bit must be cleared to the logic '0' state, to allow the FIFO to take data from both the 'I' and 'Q' outputs of the receiver. This register is accessible to both the 'C40 processor and VMEbus masters. Table 3-19, on the previous page, tells which bit is associated with which receiver chip.

3.5.10 Tag Enable Register - R/ W @ Tag_En_Ch_B_Ctl (D15 - D8)

The feature controlled by the upper eight bits of the Tag Enable / Channel B Control Register allows each 32-bit word read from the Model 6509's FIFOs to be individually identified as containing In-phase (I) or Quadrature (Q) samples. When one of these bits is set to the logic '1' state, a '0' is substituted for the LSB of each 32-bit 'Q' word, and a '1' is substituted for the LSB of each 32-bit 'I' word. This presents a small trade-off in precision, because the LSB of each 16-bit sample will be lost to the Tag bit. To disable the Tagging function and allow all 16 bits from all receivers to be read out, leave these bits cleared to their default logic '0' states. This register is accessible to both the 'C40 processor and VMEbus masters. Table 3-20, on the previous page, shows which bit enables Tag mode for which receiver. Table 3-21, below, shows examples of Tagged data in the relevant formats.

Table 3-21: Model 6509 - Comm Port Data Packing and Ordering Formats (Tagging Enabled)

Packed, Receiver Ordered Mode, Complex Data				
Sample #	**D31 ---------- D17**	**D16**	**D15 ---------- D1**	**D0**
1	(MSB) Even Receiver, Q Sample 1 (2SB)	0	(MSB) Odd Receiver, Q Sample 1 (2SB)	0
2	(MSB) Even Receiver, I Sample 1 (2SB)	1	(MSB) Odd Receiver, I Sample 1 (2SB)	1
3	(MSB) Even Receiver, Q Sample 2 (2SB)	0	(MSB) Odd Receiver, Q Sample 2 (2SB)	0
4	(MSB) Even Receiver, I Sample 2 (2SB)	1	(MSB) Odd Receiver, I Sample 2 (2SB)	1
Non-Packed, Receiver Ordered Mode, Complex Data				
Sample #	**D31 ---------- D17**	**D16**	**D15 ---------- D0**	
1	(MSB) Selected Receiver, Q Sample 1 (2SB)	0	Indeterminate	
2	(MSB) Selected Receiver, I Sample 1 (2SB)	1	Indeterminate	
3	(MSB) Selected Receiver, Q Sample 2 (2SB)	0	Indeterminate	
4	(MSB) Selected Receiver, I Sample 2 (2SB)	1	Indeterminate	
Packed, I/Q Ordered Mode, Complex Data				
Sample #	**D31 ---------- D17**	**D16**	**D15 ---------- D1**	**D0**
1	(MSB) Selected Receiver, I Sample 1 (2SB)	1	(MSB) Selected Receiver, Q Sample 1 (2SB)	0
2	(MSB) Selected Receiver, I Sample 2 (2SB)	1	(MSB) Selected Receiver, Q Sample 2 (2SB)	0
3	(MSB) Selected Receiver, I Sample 3 (2SB)	1	(MSB) Selected Receiver, Q Sample 3 (2SB)	0
4	(MSB) Selected Receiver, I Sample 4 (2SB)	1	(MSB) Selected Receiver, Q Sample 4 (2SB)	0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.11 Input Multiplex Register - R/W @ In_Mux_Reg

Each receiver on the Model 6509 can select either the Channel A or Channel B input connector at the source of its input data. Channel A is selected for a given receiver when the associated bit in this register is cleared to the logic '0' state, and Channel B is selected when the bit is set to the logic '1' state. The lower eight bits of this 16-bit register are reserved, and should be written with zeros and masked when read. This register is accessible to both the 'C40 processor and VMEbus masters. Table 3-22, below, shows which bit is used to select the input channel for which receiver.

Table 3-22: Model 6509 - Input Multiplex Register - R/W @ In_Mux_Reg

Bit #	D15	D14	D13	D12	D11	D10	D9	D8
Bit Name	Receiver _8_Input	Receiver _7_Input	Receiver _6_Input	Receiver _5_Input	Receiver _4_Input	Receiver _3_Input	Receiver _2_Input	Receiver _1_Input
Bit Function	1 = Channel B 0 = Channel A							
Bit #	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Bit Function	Reserved - Write ë0í, mask when reading							

3.5.12 Threshold Detection Registers - R/W @ Thr_Det_A & Thr_Det_B

These registers determine the conditions that will set the Threshold Detector flags in Status Register #1 (D20 & D16 @ Stat_Reg_1, see Section 3.5.4.3). Those flags, in turn, can be enabled to produce Threshold Detector Interrupts by setting the appropriate bits in Interrupt Mask Register #1 to the logic '1' state (D23 - D16 @ Int_Mask_1, see Section 3.5.2.1). The register at symbolic address Thr_Det_A is used for the receivers driven by Input Channel A, and the one at Thr_Det_B is used for those receivers driven by Channel B. Both of these registers are accessible to both the 'C40 processor and VMEbus masters. The Input Multiplex Register (see Section 3.5.11, above) determines which receivers are driven by which input channel.

The contents of the Threshold Detection Registers are summarized in Table 3-23, at the top of the next page. The subsections beginning below the table describe the functions of the registers' bits.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.12 Threshold Detection Registers (continued)

Table 3-23: Model 6509 - Threshold Detection Registers - R/W @ Thr_Det_A & Thr_Det_B

Bit #	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	Detector _Enable	Threshold _Level_1	Threshold _Level_0	Not Used		Crossing_ Count_2	Crossing_ Count_1	Crossing_ Count_0
Bit Function	1 = Enable 0 = Disable*	See Table 3-24, below		Not Applicxable		See Table 3-25, next page		
* - Negative edge resets counter								

3.5.12.1 Threshold Detector Enable Bit - D7

When this bit is cleared to the logic '0' state, the threshold detection function is disabled. The low-going transition of this bit (from logic'1' to logic '0') resets the Threshold Crossing Counter (see Section 3.5.12.3) to the zero count. To enable the threshold detection function and release the Threshold Crossing Counter, set this bit to the logic '1' state. The counter reset function should be implemented as a toggle (i. e., clear this bit, then set it).

3.5.12.2 Threshold Level Field - Bits D6 & D5

Bits D6 & D5 of the Threshold Detection Registers are used to select the data input level that will be recognized as the threshold, according to Table 3-24, below. The Threshold Crossing Counter will be incremented **for each data sample** on the channel of interest that is either greater than or equal to the positive threshold data value, or less than or equal to the negative threshold data value.

Table 3-24: Model 6509 - Threshold Level Bits

D6 (Lev_1)	D5 (Lev_0)	Threshold Setting	Positive Data Value (Hex)	Negative Data Value (Hex)
0	0	± 1/2 of Full Scale	≥ 0x0400	≤ 0x0BFF
0	1	± 5/8 of Full Scale	≥ 0x0500	≤ 0x0AFF
1	0	± 3/4 of Full Scale	≥ 0x0600	≤ 0x09FF
1	1	± 7/8 of Full Scale	≥ 0x0700	≤ 0x08FF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.12 Threshold Detection Registers (continued)

3.5.12.3 Crossing Count Field - Bits D2 through D0

Bits D2 - D0 of the Threshold Detection Registers are used to select the value in the Threshold Crossing Counter that will set the Threshold Detection flags in Status Register #1 to the logic '1' state, for those receivers driven by the input channel of interest. These status flags can generate VME interrupts if enabled in Interrupt Mask Register #1 (see Section 3.5.4.3 and Section 3.5.2.1). Increasing binary values in this 3-bit field result in increasing odd power-of 2 counter values setting the flags, according to Table 3-25, below.

Table 3-25: Model 6509 - Threshold Crossing Count Value To Set Status Flag

D2	D1	D0	Count
0	0	0	2 (2^1)
0	0	1	8 (2^3)
0	1	0	32 (2^5)
0	1	1	128 (2^7)
1	0	0	512 (2^9)
1	0	1	2048 (2^{11})
1	1	0	8192 (2^{13})
1	1	1	32768 (2^{15})

3.5.13 Receiver Output Data FIFOs - R. O. @ FIFO_1 through FIFO_8

Output data from each digital receiver on the Model 6509 is stored in a separate FIFO register. Data can be read from these eight FIFOs over the VMEbus, and by the on-board 'C40, as well as over the Comm Port interface by off-card 'C40-type devices. The contents of the Comm Port Output Control Register do not affect the format of data read directly from the FIFOs (except that direct readout is disabled if the Comm Ports are enabled). Data packing is unavailable in this mode (i. e., 32-bit reads from the FIFOs will only yield a single, 16-bit data sample). The Tag function, however is available from the FIFOs. An attempt to read an empty FIFO will cause wait states to be asserted on the device mastering the read cycle.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.13 Receiver Output Data FIFOs (continued)

Note that FIFO readout may be accomplished in either of two ways. All reads from a given FIFO may be performed at a single address, with each read producing the next available FIFO sample. Alternatively, block transfers can be used, by incrementing addresses within the given FIFO's address range. The address range for each FIFO is 8 kWords (16 kBytes). Again, each read operation will produce the next available sample from the FIFO. Please refer to Table 3-7, on page 35, for the address range allotted to each receiver's FIFO.

NOTE: Because the FIFO address range is 8 kWords, units equipped with Option 021 (16 kSample FIFOs) each FIFO will require 2 block transfer operations of 8 kWords each to empty if full.

Data read from the FIFOs is always ordered by receiver, never packed, and may be tagged or untagged. For a complete description of data ordering modes, see Section 3.5.7.2 and Table 3-14. For a complete description of data tagging, see Section 3.5.10 and Table 3-21. The formats available when the FIFOs are read directly are shown in Table 3-26, below.

Table 3-26: Model 6509 - Direct FIFO Readout Formats

Untagged, Real Data Only																
Sample #	D31														D16	D15 - D0
1	(MSB)	Real Sample 1													(LSB)	Indeterminate
2	(MSB)	Real Sample 2													(LSB)	Indeterminate
3	(MSB)	Real Sample 3													(LSB)	Indeterminate
4	(MSB)	Real Sample 4													(LSB)	Indeterminate
Untagged, Complex Data																
Sample #	D31														D16	D15 - D0
1	(MSB)	Q Sample 1													(LSB)	Indeterminate
2	(MSB)	I Sample 1													(LSB)	Indeterminate
3	(MSB)	Q Sample 2													(LSB)	Indeterminate
4	(MSB)	I Sample 2													(LSB)	Indeterminate
Tagged, Complex Data																
Sample #	D31													D17	D16	D15 - D0
1	(MSB)	Q Sample 1												(LSB)	0	Indeterminate
2	(MSB)	I Sample 1												(LSB)	1	Indeterminate
3	(MSB)	Q Sample 2												(LSB)	0	Indeterminate
4	(MSB)	I Sample 2												(LSB)	1	Indeterminate

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 ëC40 Global and VMEbus A32/A24 Slave Memory Map (continued)

3.5.14 TMS320C40 IACK Vector Address - W. O. @ ëC40 Address 0x8004 0000

The VMEbus can issue an interrupt to the Model 6509's 'C40 Processor by writing to address A16_base+0x08. Writing this address will assert an interrupt at the 'C40's IIOF1 input, and set the Pending_C40_Interrupt Bit (D15 at address A16_base+0x10) to the logic '1' state.

The 'C40 should acknowledge the IIOF1 interrupt by writing arbitrary data (the data is unused) to Global bus address 0x8004 0000. This will clear the Pending_C40_Interrupt bit in the VMEbus Status Register (A16_base+0x10) to the logic '0' state. Note that an A32 VME master can also access this location, at address A32_base+0x0010 0000, but the usefulness of such an access is questionable, to say the least. No A24 access to this resource is provided.

3.5.15 TMS320C40 Interrupt to VMEbus - W. O. @ ëC40 Address 0x8005 0000

Upon receiving one of the FIFO-related, Overload or Clock Timeout interrupts covered in Section 3.5.2 and Section 3.5.3, the 'C40 processor on the 6509 may respond by passing the interrupt to the VMEbus. The 'C40 processor interrupts VME by writing arbitrary data (the data is unused) to Global bus address 0x8005 0000. The interrupt is asserted on the IRQ Level selected by the IRQ_Level_Select field in the VMEbus Control Register (bits D10 - D8 at the A16_base address - see Section 3.4.1.5 and Table 3-3). When this interrupt is generated, the MSB (D31, the VME Interrupt Status Bit) of Status Register #1 (see Section 3.5.4.1) is set to the logic '1' (true) state. The VMEbus IACK cycle clears the status bit back to the logic '0' (false) state. The 6509's IACK Vector Address is stored in the 8 LSBs of the VMEbus Control Register (see Table 3-3 and Section 3.4.1.6).

Note that an A32 VME master can also access this location, at address A32_base+0x0014 0000, but the usefulness of such an access is highly questionable. No A24 access to this resource is provided.

3.5.16 Global Flash EEPROM - R/W @ Flash

A 512 kByte bank of Flash EEPROM is available to both the 6509's 'C40 Processor and A32 VMEbus masters (no A24 access to this resource is provided). This memory region is arranged as byte data located on longword addresses. In other words, one byte of data may be read from or written to the low byte (D0 - D7) of each longword address (i. e., every fourth byte address). When shipped, each Model 6509 contains firmware in Sectors 1 & 2. Customized application code may be stored in this non-volatile memory bank, to facilitate the creation of standalone signal processing systems. However, if you want to write your own code to perform the functions provided by the firmware, you must first erase the Flash EEPROM prior to writing any custom code to the EEPROM. The ëCí program **`erase65.c`**, located in Appendix A, can be used to erase the EEPROM.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6 TMS320C40 Local Bus Resources

The Local Bus interface of the Model 6509's TMS320C40 processor contains two banks of memory, a 32 kByte (8 k-Longword) EEPROM and a 512 kByte (128 k-Longword) SRAM. Table 3-27, below, shows the 'C40's Local Bus memory map. The subsections that follow the table describe the resources.

Table 3-27: Model 6509 - TMS320C40 Local Bus Memory Map

Address Range	Size	Access	Resource
0x0030 0000 - 0x0030 7FFF	8k x 32	R. O.	Boot EEPROM
0x4000 0000 - 0x4001 FFFF	128k x 32	R / W	Local SRAM

3.6.1 Boot EEPROM - R. O. from 0x0030 0000 to 0x0030 7FFF

This 8 k-Longword (32 kByte) EEPROM stores the factory supplied boot code for the 'C40 processor on the Model 6509. The 'C40 boots from this EEPROM by default. If 6509 is reset after the D14 bit in the VMEbus Control Register (at the A16_base address) has been set to the logic '1' state, the 'C40 will boot from the Comm Port interface instead of this EEPROM.

3.6.2 Local SRAM - R/W from 0x4000 0000 to 0x4001 FFFF

This 128k-longword (512 kByte) bank of SRAM can be accessed only by the 'C40 processor on the 6509.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 4: 65xx Series Firmware Command Set

4.1 Controlling the Pentek 65xx series Digital Receiver

The Pentek 65xx series Digital Receivers contain firmware which can be used to inital-ize the digital receivers and control the data acquistion cycle. Firmware commands can be initiated over the œC40 communication ports or through the VME Interface. The 65xx series has four Input Comm Ports, labeled in pairs as 1&2, 3&4, 5&6, and 7&8, which can be used to setup and control the digital receivers. Any Comm Port can be used to setup any or all receivers.

4.1.1 Firmware Command Format and Structure

Each firmware command contains a command word, a mask specifying input channel or receiver, and an optional set of parameters. Macros are provided for building the command word. Example macros are provided for each firm-ware command definition starting in Section 4.3. The command word, **`P65XX_CMD_WORD`**, is defined in **`65xxcmd.h`** and contains cmd, handle, wait_flag, data_fmt, and spare fields. Figure 4-1, below, shows the **`P65XX_CMD_WORD`** definition as it appears in **`65xxcmd.h`**.

P65XX_CMD_WORD Bits				
D31Ö D23	D22Ö D21	D20	D19Ö D12	D11Ö D0
spare	data_fmt	wait_flag	handle	cmd
Input Channel or Receiver Mask (optional)				
Parameter 1 (optional)				
Parameter N (optional)				

Figure 4-1: 65XX Firmware Command Word Defined in 65xxcmd.h

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.1 Firmware Command Format and Structure (continued)

4.1.2 `cmd` Field

The **`cmd`** field contains the actual executable command. All commands are listed in Section 4.3 starting on page 66.

4.1.3 `handle` Field

The **`handle`** field specifies the connection affected by the command. Each receiver will usually have a separate connection handle.

4.1.4 `wait` Flag

The **`wait`** Flag field specifies that the processor sending the command is waiting for a reply. A reply consists of a command completion status and any return data. When sending commands over the Comm Ports and the wait flag is set to 0, return status and data are not returned to the caller. When the wait flag is set using the VME Interface, a VME Interrupt is generated indicating the command is complete.

4.1.5 `data_fmt`

The **`data_fmt`** field indicates the data format of all floating point parameter values associated with the command. The two parameters affected by this field are the LO (Local Oscillator Frequency) and the Input Clock Rate. All other parameters ignore this field. There are three possible values for this field; **`P65XX_DEFAULT_TYPE`**, **`P65XXX_IEEE_FLOAT_TYPE`**, and **`P65XX_INTEGER_TYPE`**.

4.1.5.1 `P65XX_DEFAULT_TYPE`

The ëC40ís Twos Complement Floating Point Format is the default type of all floating point parameters.

4.1.5.2 `P65XX_IEEE_FLOAT_TYPE`

When specified, the ëC40 converts a 32 bit IEEE Float to a ëC40 Float. It is necessary to use this parameter when using IEEE Std. 754 Single Precision floating format since it is not compatible with the ëC40 floating point format.

4.1.5.3 `P65XX_INTEGER_TYPE`

When floating point data types arenít available, a 32-bit integer value can be specified by using this parameter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.1 Firmware Command Format and Structure (continued)

4.1.6 `Input Channel / Receiver Mask`

The **`Input Channel or Receiver Mask`** specifies one or more input channels or receivers affected by this command.

The following are constants defined in **`65xxcmd.h`**:

```
P65XX_RECEIVER_1        P65XX_INPUT_A
P65XX_RECEIVER_2        P65XX_INPUT_B
P65XX_RECEIVER_3        P65XX_INPUT_C
P65XX_RECEIVER_4        P65XX_INPUT_D
P65XX_RECEIVER_5
P65XX_RECEIVER_6
P65XX_RECEIVER_7
P65XX_RECEIVER_8
```

4.1.7 Optional Parameters

Some commands also contain a list of additional parameters which may need to be specified when using a specific command. Optional parameters are listed in the command definition.

4.2 Disabling the Model 65XX Series Firmware

The firmware on the Model 6509 can be disabled either temporarily or permanently. The temporary solution allows you to disable the firmware, but requires you to disable it each time the reciever is powered on.

4.2.1 Temporary Solution

Setup the Model 6509ís A24/A32 base address, then write the value **`0xc33c 3cc3`** to address **`A24/A32_base+0x0`**. After writing this value you must reset the Model 6509. This can be done by writing **`0x800`** to the VME-Bus Control Register, **`A16 _base+0x0`**. This process must be repeated upon each power-up cycle.

4.2.2 4.2.2 Permanent Solution

The permanent solution for disabling the firmware involves erasing the FLASH memory that contains the firmware. The program for doing this can be obtained from Pentek's Applications Engineering department.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands

This section contains all 65XX Firmware commands. Commands specific to the Pentek Model 6508 are listed in Section 4.6. Each description of the Firmware command set includes an example code fragment showing how the command is used. However, the commands do not include the complete code needed to make the call work. The setup and initialization of arrays and pointers, for example, are not included in every example macro. The code fragment shown below, shows everything that needs to be included each time a command is used.

```
unsigned long handle;                  /* Not included in future examples */
unsigned long status;                  /* Not included in future examples */
unsigned long numRData;                /* Not included in future examples */
unsigned long returnData[2];           /* Not included in future examples */
unsigned long cmdParams[2];            /* Not included in future examples */

#define CMD_COMM_PORT 2  /* Local Communication Port for sending commands */
```

4.3.1 P65XX_OPEN_CONNECTION

Command	**P65XX_OPEN_CONNECTION**
Input Parameters	None
Return Data	Handle Identifier for this command.
Description	Opens a connection to one or more of the receivers. A connection usually needs to be established for each receiver.

Example

```
status = P65XXSendCommandWait (CMD_COMM_PORT,

P65XX_BUILD_CMD_WAIT(P65XX_OPEN_CONNECTION,0),
                     0, (void *)NULL, 0, &numRdata, returnData);
handle=*returnData;
```

Notes:	**A connection is opened and a handle is returned.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.2 P65XX_CLOSE_CONNECTION

Command: **P65XX_CLOSE_CONNECTION**

Input Parameters: Handle Identifies the connection to close.

Return Data: None

Description: Closes an existing connection.

Example:

```
status = P65XXSendCommandWait (CMD_COMM_PORT,

P65XX_BUILD_CMD_WAIT(P65XX_CLOSE_CONNECTION,
                     handle), 0, (void *)NULL, 0, NULL,NULL);
```

Notes: **The open connection corresponding to this connection is closed.**

4.3.3 P65XX_CLOSE _ALL_CONNECTIONS

Command: **P65XX_CLOSE_ALL_CONNECTIONS**

Input Parameters: None

Return Data: None

Description: Closes all connections

Example:

```
Status = P65XXSendCommandWait(CMD_COMM_PORT,
          P65XX_BUILD_CMD_WAIT(P65XX_CLOSE_ALL_CONNECTIONS,0),
          0, (void *)NULL, 0, NULL, NULL);
```

Notes: **All connections are closed.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.4 P65XX_RESET_REGS

Command: **P65XX_RESET_REGS**

Input Parameters: None

Return Data: None

Description: Sets all control registers to a default state.

Example:

```
Status = P65XXP65XXSendCommandWait(CMD_COMM_PORT,
                    P65XX_BUILD_CMD_WAIT(P65XX_RESET_REGS,0),0,
                    (void *)NULL, 0, NULL, NULL);
```

Notes: **All control registers are reset to their default state.**

4.3.5 P65XX_INIT_RCVRS

Command: **P65XX_INIT_RCVRS**

Input Parameters:

Handle	Identifies the open connection.
Data Fmt	Specifies the data format for all floating point parameters.
Receiver mask	Identifies the receivers to be initialized.
Setup Parameter Structure	The **P65XX_RCVR_PARAMS** data structure contains the setup parameters. Each 65XX Model has a unique data structure specified in **65XX.h**. XX is the specific model receiver, such as **6509.h**

Return Data: None

Description: Specifies all setup parameters for one or more receivers.

Example:

```
p6509_Rcvr_Params,

Status = P65XXSendCommandWait(CMD_COMM_PORT,
      P65XX_BUILD_CMD_WAIT(P65XX_INIT_RCVRS,handle),
      SIZEOF(P65XX_RCVR_PARAMS), (void *)&p65XXRcvrParams,
      P65XX_RECEIVER_1, NULL, NULL);
```

Notes: **Receiver 1 is initalized with the settings specified in P65XX_RCVR_PARAMS structure.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.6 P65XX_INIT_INPUT_CHANNELS

Command:	**P65XX_INIT_INPUT_CHANNELS**	
Input Parameters:	Input Mask	Identifies the input channels to be initialized.
	Setup Parameter Structure	The **P65XX_INPUT_PARAMS** data structure contains the setup parameters. Each 65XX Model has a unique data structure specified in 65XX.h. XX is the specific model receiver, such as 6509.h
Return Data:	None	
Description:	Specifies all setup parameters for the Input Channels.	

Example:

```
P6509_INPUT_PARAMS P65xxInputParms;
STATUS = P65XXSendCommandWait(CMD_COMM_PORT,
                P65XX_BUILD_CMD_WAIT(P65XX_INIT_INPUT_CHANNELS,
                0), sizeof(P65XX_INPUT_PARAMS),
                (void *)&P65XXInputParams, P65XX_INPUT_B, NULL,
                NULL);
```

Notes: **Input channel B is initalized with the settings specified in P65XX_Input_Params structure.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.7 P65XX_SET_LO

Command: `P65XX_SET_LO`

Input Parameters:

Handle	Identifies the open connection.
Data fmt	Specifies the data format.
Receiver Mask	Identifies the receivers to be modified.
LO	Local Oscillator frequency specified in floating point.

Return Data: None

Description: Sets the local oscillator frequency for one or more receivers. The LO is dependent on the Input Clock Rate going into the receiver.

Example:

```
cmdParams[0] = 24.0e6;

status = P65XXSendCommandWait (CMD_COMM_PORT,
           P65XX_BUILD_CMD_WAIT(P65XX_SET_LO, handle), 1,
           (void *)cmdParams, P65XX_RECEIVER_2, NULL, NULL);
```

Notes: **The local oscillator frequency for receiver 2 was set to 24 MHz. By performing an OR operation on the receiver masks, you can set up multiple receivers in a single command. For example;**

```
... (void *) cmdParams,
    P65XX_RECEIVER_2   P65XX_RECEIVER_4, ...
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.8 P65XX_SET_DECIMATION

Command: **P65XX_SET_DECIMATION**

Input Parameters:

Handle	Identifies the open connection.
Receiver Mask	Identifies the receivers to be modified.
Decimation Rate	Decimation Rate is dependent on Data Mode. The 6508, 6509, and 6510 have the following range of values: Real Mode Min: 32 Max: 32768 Cmplx Mode Min: 64 Max: 65536

Return Data: None

Description: Sets the decimation rate frequency for one or more receivers.

Example:

```
cmdParams[0] = 2400;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_SET_DECIMATION,
            handle), 1, (void *)cmdParams, P65XX_RECEIVER_2,
            NULL, NULL);
```

Notes: **The decimation rate of receiver 2 was set to 2400. Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.9 P65XX_SET_INP_CLK_RATE

Command: P65XX_SET_INP_CLK_RATE

Input Parameters:

Handle	Identifies the open connection.
Data fmt	Specifies the data format.
Receiver Mask	Identifies the receivers to be modified.
Input Clock Rate	Input Clock Rate specified in floating point.

Return Data: None

Description: Specifies the input clock rate into the digital receiver for one or more receivers. The LO is adjusted in the digital receiver to reflect the changes in the input clock rate.

Example:

```
cmdParams[0] = 50.0e6;

status = P65XXSendCommandWait (CMD_COMM_PORT,
           P65XX_BUILD_CMD_WAIT(P65XX_SET_INP_CLK_RATE,
           handle), 1, (void *)cmdParams, P65XX_RECEIVER_6,
           NULL, NULL);
```

Notes: **An input clock rate of 50 MHz for receiver 6 was set. Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.10 P65XX_FLIP_SPECTRUM

Command:	**P65XX_FLIP_SPECTRUM**	
Input Parameters:	Handle	Identifies the open connection.
	Receiver Mask	Identifies the receivers to be modified.
	Flip Spectrum	Flip Output Spectrum: **P65XX_FLIP_OFF** **P65XX_FLIP_ON**
Return Data:	None	
Description:	Inverts the output spectrum for one or more receivers. For addtional information, refer to the Harris Chip appendix, where this is refered to as Spectral Reverse (Control Register 4, Bit D0).	

Example:

```
cmdParams[0] = P65XX_FLIP_ON

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_FLIP_SPECTRUM, handle),
            1, (void *)cmdParams, P65XX_RECEIVER_6, NULL,
            NULL);
```

Notes:	**Output spectrum for receiver 6 is inverted.** **Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.11 P65XX_OFFSET_SPECTRUM

Command:	**P65XX_OFFSET_SPECTRUM**	
Input Parameters:	Handle	Identifies the open connection.
	Receiver Mask	Identifies the receivers to be modified.
	Offset Spectrum	Offset Output Spectrum: **P65XX_OFFSET_OFF** **P65XX_OFFSET_ON**
Return Data:	None	
Description:	Offsets the output spectrum for one or more receivers. For addtional information, refer to the Harris Chip appendix, where this is refered to as Up Convert (Control Register 4, Bit D32). Up Convert is only applicable when using Complex Data.	

Example:

```
cmdParams[0] = P65XX_OFFSET_OFF;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_OFFSET_SPECTRUM,
            handle), 1,(void *)cmdParams, P65XX_RECEIVER_8,
            NULL, NULL);
```

Notes:	**Offset Spectrum for receiver 8 is it toggled off. Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.12 P65XX_SET_DATA_MODE

Command:	**P65XX_SET_DATA_MODE**	
Input Parameters:	Handle	Identifies the open connection.
	Receiver Mask	Identifies the receivers to be modified.
	Data Mode	Output Data Mode: **P65XX_COMPLEX** **P65XX_REAL**
Return Data:	None	
Description:	Specifies the output data format for one or more receivers.	

Example:

```
cmdParams[0] = P65XX_COMPLEX;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_SET_DATA_MODE, handle),
            1, (void *)cmdParams, P65XX_RECEIVER_8, NULL,
            NULL);
```

Notes: **Receiver 8 is set to use complex data mode. Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.13 P65XX_SET_DATA_SRC

Command:	**P65XX_SET_DATA_SRC**	
Input Parameters:	Handle	Identifies the open connection.
	Receiver Mask	Identifies the receivers to be modified.
	Data Source	Input Data Source: **P65XX_INPUT_A** **P65XX_INPUT_B** **P65XX_INPUT_C** **P65XX_INPUT_D**
Return Data:	None	
Description:	Specifies the input data source for one or more receivers.	

Example:

```
cmdParams[0] = P65XX_INPUT_B;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_SET_DATA_SRC, handle),
            1, (void *)cmdParams, P65XX_RECEIVER_5, NULL,
            NULL);
```

Notes:	**Receiver 5 is setup to use input source B.** **Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.14 P65XX_SET_OUTPUT_SRC

Command: **P65XX_SET_OUTPUT_SRC**

Input Parameters:

Handle	Identifies the open connection.
Receiver Mask	Identifies the receivers to be modified.
Output Source	Output Source: **P65XX_FIFO** **P65XX_COMM_PORT**

Return Data: None

Description: Specifies the destination of the data for one or more receivers.

Example:

```
cmdParams[0] = P65XX_COMM_PORT;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_SET_OUTPUT_SRC,
            handle), 1, (void *)cmdParams, P65XX_RECEIVER_3 |
            P65XX_RECEIVER_4, NULL, NULL);
```

Notes: **Receiver 3 and 4 send all output to the front panel comm ports. Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.15 P65XX_SET_PACKING_MODE

Command: **P65XX_SET_PACKING_MODE**

Input Parameters:

Handle	Identifies the open connection.
Receiver Mask	Identifies the receivers to be modified.
Packing Mode	Packing Mode: **P65XX_PACKED** **P65XX_UNPACKED**

Return Data: None

Description: Specifies the output format for data being transfered over Comm Ports for one or more receivers. Data read directly from FIFO is always unpacked.

Example:

```
cmdParams[0] = P65XX_UNPACKED

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_SET_PACKING_MODE,
            handle), 1, (void *)cmdParams, P65XX_RECEIVER_2,
            NULL, NULL);
```

Notes: **Receiver 2 sends output to the front panel comm ports in unpacked format. Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.16 P65XX_SET_DATA_ORDER

Command: **P65XX_SET_DATA_ORDER**

Input Parameters:

Handle	Identifies the open connection.
Receiver Mask	Identifies the receivers to be modified.
Data Order	Real Mode: **P65XX_RECEIVER_ORDER** **P65XX_TIME_ORDER** Complex Mode: **P65XX_RECEIVER_ORDER** **P65XX_IQ_ORDER**

Return Data: None

Description: Specifies the data order for data being transfered over Comm Ports for one or more receivers.

Example:

```
cmdParams[0] = P65XX_IQ_ORDER;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_SET_DATA_ORDER,
            handle), 1, (void *)cmdParams, P65XX_RECEIVER_7,
            NULL, NULL);
```

Notes: **Receiver 7 will receive complex mode data in IQ order. Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.17 P65XX_SELECT_RECEIVER

Command:	**P65XX_SELECT_RECEIVER**	
Input Parameters:	Handle	Identifies the open connection.
	Receiver Mask	Identifies the receivers to be modified.
	Receiver	Selected Receiver: **P65XX_EVEN_RECEIVER** for receivers 2, 4, 6, 8 **P65XX_ODD_RECEIVER** for receivers 1, 3, 5, 7
Return Data:	None	
Description:	Specifies the receivers to be sent over the Comm Port when data is Packed IQ Data, Packed Time Ordered Data, or Unpacked Receiver Ordered data.	

Example:

```
cmdParams[0] = P65XX_EVEN_RECEIVER;

status = P65XXSendCommandWait (CMD_COMM_PORT,
           P65XX_BUILD_CMD_WAIT(P65XX_SELECT_RECEIVER,
           handle), 1, (void *)cmdParams, P65XX_RECEIVER_4,
           NULL, NULL);
```

Notes: Receiver 4 data is output to COMM PORT 3 & 4 out.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.18 P65XX_TAG_ENABLE

Command:	**P65XX_TAG_ENABLE**	
Input Parameters:	Handle	Identifies the open connection.
	Receiver Mask	Identifies the receivers to be modified.
	Receiver	Selected Receiver: **P65XX_TAG_OFF** **P65XX_TAG_ON**
Return Data:	None	
Description:	Specifies wheter the LSB of the data is a data bit or an indicator of I/Q data for one or more receivers.	

Example:

```
cmdParams[0] = P65XX_TAG_ON;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_TAG_ENABLE, handle), 1,
            (void *)cmdParams, P65XX_RECEIVER_3, NULL, NULL);
```

Notes: **Turns on the I/Q TAG bit for receiver 3.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.19 P65XX_SET_THRESHOLD_DETECTION

Command: **P65XX_SET_THRESHOLD_DETECTION**

Input Parameters:

Input Mask	Identifies the input channels affected.
Threshold Level/Count	The Threshold Level and Count are specified in a single 32-bit word.

Threshold Levels:

P65XX_THRSHLD_DETECT_OFF
P65XX_ONE_HALF_FSCALE
P65XX_FIVE_EIGHTS_FSCALE
P65XX_THREE_FOURTHS_FSCALE
P65XX_SEVEN_EIGHTS_FSCALE

Threshold Counts:

P65XX_CROSS_CNT_2
P65XX_CROSS_CNT_8
P65XX_CROSS_CNT_32
P65XX_CROSS_CNT_128
P65XX_CROSS_CNT_512
P65XX_CROSS_CNT_2028
P65XX_CROSS_CNT_8192
P65XX_CROSS_CNT_32768

Return Data: None

Description: Specifies the data input level recognized as the threshold and the number of times the threshold is crossed before the threshold detection flag is set for one or more of the input channels.

Example:

```
cmdparam[0] = P65XX_THREE_FOURTHS_FSCALE | P65XX_CROSS_CNT_512;

STATUS=P65XXSEND_COMMAND_WAIT(CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT
            (P65XX_SET_THRESHOLD_DETECTION, 0), 1,
            (void *)cmdParams, P65XX_INPUT_B, NULL, NULL);
```

Notes: **The threshold detection flag will be set when channel Bís input level crosses the æ FSCALE Threshold 512 times.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.20 P65XX_RELEASE_FIFO

Command: **P65XX_RELEASE_FIFO**

Input Parameters:

Handle	Identifies the open connection
Receiver Mask	Identifies the receivers to be released

Return Data: None

Description: Specifies to release the FIFO for one or more receivers. FIFO release is dependent on the method used to read the data. When reading from the FIFO, the FIFOs are released independently. When reading data through the front panel Comm Port outputs, the FIFOs are released in pairs.

NOTE: This command only places the FIFO into the released state, and does not reset the FIFO. To reset FIFO see the information in **P65XX_RESET_FIFO**, or **P65XX_FLUSH_FIFO**.

Example:

```
status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_RELEASE_FIFO, handle),
            0, (void *)NULL, P65XX_RECEIVER_2 |
            P65XX_RECEIVER_4, NULL, NULL);
```

Notes: **FIFOs are released on receivers 2 and 4.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.21 P65XX_RESET_FIFO

Command: **P65XX_RESET_FIFO**

Input Parameters:

Handle	Identifies the open connection
Receiver Mask	Identifies the receivers to be reset

Return Data: None

Description: Specifies to reset the FIFO for one or more receivers. FIFO reset is dependent on the method used to read the data. When reading from the FIFO, the FIFOs are reset independently. When reading data through the front panel Comm Port outputs, the FIFOs are reset in pairs.

NOTE: This command only places the FIFO into the reset state, and does not release the FIFO. To release FIFO see the information in **P65XX_RELEASE_FIFO**, or **P65XX_FLUSH_FIFO**.

Example:

```
status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_RESET_FIFO, handle), 0,
            (void *)NULL, P65XX_RECEIVER_1 |
            P65XX_RECEIVER_3, NULL, NULL);
```

Notes: **FIFO is reset on receivers 1 and 3.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.22 P65XX_FLUSH_FIFO

Command: **P65XX_FLUSH_FIFO**

Input Parameters:

Handle	Identifies the open connection
Receiver Mask	Identifies the receivers to be released

Return Data: None

Description: Specifies to reset and release the FIFO for one or more receivers. FIFO flushing is dependent on the method used to read the data. When reading from the FIFO, the FIFOs are flushed independently. When reading data through the front panel Comm Port outputs, the FIFOs are flushed in pairs.

NOTE: Combines **P65XX_RELEASE_FIFO** and **P65XX_RESET_FIFO** into a single command.

Example:

```
status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_FLUSH_FIFO, handle), 0,
            (void *)NULL, P65XX_RECEIVER_5, NULL, NULL);
```

Notes: **FIFO is reset and released on receiver 5.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.23 P65XX_GET_RCVR_STATUS

Command: **P65XX_GET_RCVR_STATUS**

Input Parameters: None

Return Data: Receiver Status Register #1
Receiver Status Register #2

Description: Returns the current value of Receiver Status Register 1 and Receiver Status Register 2.

Example:

```
status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_GET_RCVR_STATUS, 0), 0,
            (void *)NULL, NULL, &numRData, returnData);
```

Notes: **Status Register 1 status is contained in returnData [0].**
Status register 2 status is contained in returnData [1].

4.3.24 P65XX_GET_RCVR_DATA

Command: **P65XX_GET_RCVR_DATA**

Input Parameters: None

Receiver Mask	Identifies the receivers to be read.
Communications Port	Specifies the 6C40 Comm Port indicating the destination of the data. Values 2 through 5 are valid.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.24 P65XX_GET_RCVR_DATA (continued)

Command: **P65XX_GET_RCVR_DATA** (continued)

Input Parameters: Data Type/Latency Specifies the data type and if it is buffered or output immediately. The Type and the Latency are specified in a single 32-bit word.

Data Types:

P65XX_REAL_DATA
Output Real Data

P65XX_COMPLEX_DATA
Output Complex Data

Latency Types:

P65XX_BUFFERED_DATA
Data is output to the Comm Port after the FIFO is half full.

P65XX_NON_BUFFERED_DATA
Data is output to the Comm Port as soon as it is available in the FIFO.

Return Data: None

Description: Multiplexes data from specified receivers and outputs the data to a single é€40 Comm Port.

NOTE: The format of the data sent through the Comm Ports is specified in Table 4-1 on the next page.

Example:

```
cmdParams [0]= 5  /* 65XX Boards 7 & 8 IN Comm Port */
cmdParams [1]= P65XX_COMPLEX_DATA | P65XX_NONBUFFERED_DATA;

status=P65XX_SEND_COMMAND_WAIT(CMD_COMM_PORT,
                P65XX_BUILD_CMD_WAIT(P65XX_GET_RCVR_DATA, 0),
                2, (void *)cmdParams, P65XX_RECEIVER_1 |
                P65XX_RECEIVER_2 |P65XX_RECEIVER_3 |
                P65XX_RECEIVER_4, NULL, NULL);
```

Notes: **Complex data from receivers 1, 2, 3, and 4 are output to Comm Port 5 (labled 7 & 8 in) as soon as itís available.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.24 P65XX_GET_RCVR_DATA (continued)

Table 4-1: P65XX_GET_RCVR_DATA Comm Port Data Format				
Data Mode	**Receiver #**	**D31...D0**		**Notes**
Real	1	I		In Real Mode, data is right justified and sign extended.
	2	I		
	...N	I		
Data Mode	**Receiver #**	**D31...D16**	**D15...D0**	**Notes**
Complex	1	I	Q	
	2	I	Q	
	...N	I	Q	

4.3.25 P65XX_STOP_RCVR DATA

Command: P65XX_STOP_RCVR_DATA

Input Parameters: none

Return Data: none

Description: Aborts the data stream specified by the P65XX_GET_RCVR_DATA command.

Example:

```
status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_STOP_RCVR_DATA, 0), 0,
            (void *)NULL, NULL, NULL, NULL);
```

Notes:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.26 P65XX_DISABLE_FIRMWARE

Command: `P65XX_DISABLE_FIRMWARE`

Input Parameters: none

Return Data: none

Description: Erases the boot sector in flash memory. When the board is reset, the firmware is no longer loaded.

Example:

```
status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_DISABLE_FIRMWARE, 0), 0,
            (void *)NULL, NULL, NULL, NULL);
```

Notes:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.27 P65XX_MODIFY_INTR_MASK

Command: **P65XX_MODIFY_INTR_MASK**

Input Parameters: Interrupt Command — Specifies whether the interrupt should be enabled/disabled, the method of notification, and if the interrupt should be level or edge triggered. Combine this command with following constant to achieve the desired operation:

P65XX_DISABLE_INTERRUPT
Disables the interrupts specified in the second parameter.

P65XX_ENABLE_INTERRUPT
Enables the interrupts specified in the second parameter.

P65XX_GEN_CP_2_OUTPUT
Output Status Word to Comm Port 2 when interrupt condition occurs. (Labled 1&2 IN)

P65XX_GEN_CP_3_OUTPUT
Output Status Word to Comm Port 3 when interrupt condition occurs. (Labled 3&4 IN)

P65XX_GEN_CP_4_OUTPUT
Output Status Word to Comm Port 4 when interrupt condition occurs. (Labled 5&6 IN)

P65XX_GEN_CP_5_OUTPUT
Output Status Word to Comm Port 5 when interrupt condition occurs. (Labled 7&8 IN)

P65XX_GEN_VME_INTR
Generates a VME Bus Interrupt when interrupt condition occurs.

P65XX_EDGE_TRIG_INTR
Interrupts are edge triggered.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.27 P65XX_MODIFY_INTR_MASK (continued)

Command: **P65XX_MODIFY_INTR_MASK** (continued)

Input Parameters:

Interrupt Command (continued)

P65XX_LEVEL_TRIG_INTR
Interrupts are level triggered.

P65XX_MASK_ON_OCCURRENCE
Mask Interrupts after it occurs.

Interrupt Conditions — Defines the interrupts which are enabled or disabled.

P65XX_FIFO_NOT_EMPTY_1
P65XX_FIFO_NOT_EMPTY_2
P65XX_FIFO_NOT_EMPTY_3
P65XX_FIFO_NOT_EMPTY_4
P65XX_FIFO_NOT_EMPTY_5
P65XX_FIFO_NOT_EMPTY_6
P65XX_FIFO_NOT_EMPTY_7
P65XX_FIFO_NOT_EMPTY_8
P65XX_FIFO_HALF_FULL_1
P65XX_FIFO_HALF_FULL_2
P65XX_FIFO_HALF_FULL_3
P65XX_FIFO_HALF_FULL_4
P65XX_FIFO_HALF_FULL_5
P65XX_FIFO_HALF_FULL_6
P65XX_FIFO_HALF_FULL_7
P65XX_FIFO_HALF_FULL_8
P65XX_FIFO_FULL_1
P65XX_FIFO_FULL_2
P65XX_FIFO_FULL_3
P65XX_FIFO_FULL_4
P65XX_FIFO_FULL_5
P65XX_FIFO_FULL_6
P65XX_FIFO_FULL_7
P65XX_FIFO_FULL_8
P65XX_CLOCK_NOT_READY_A
P65XX_CLOCK_NOT_READY_B
P65XX_CLOCK_NOT_READY_C
P65XX_CLOCK_NOT_READY_D
P65XX_THRSHLD_DETECT_A
P65XX_THRSHLD_DETECT_B
P65XX_THRSHLD_DETECT_C
P65XX_THRSHLD_DETECT_D

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.27 P65XX_MODIFY_INTR_MASK (continued)

Command: **P65XX_MODIFY_INTR_MASK** (continued)

Description Enables/Disables the interrupt mask for detecting FIFO Not Empty, FIFO Half Full, FIFO Full, Threshold Detection, and Clock Not Ready. This command also specifies the method of notification. One method of notification is to have the firmware generate a VME Bus Interrupt when a condition occurs. The other method is to have the firmware write the status register associated with the interrupting condition to a specified Comm Port when a condition occurs.

IMPORTANT: Level Triggered Interrupts which are not masked on occurrence can put the â€C40 in an infinite interrupt loop. For this reason, it is recommended to mask all Level Triggered Interrupts on occurrence. When in an infinite interrupt loop, the â€C40 may not respond to Comm Port Commands.

Example:

```
cmdParams[0] = P65XX_GEN_VME_INTR;
cmdParams[1] = P65XX_CLOCK_NOT_READY;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_MODIFY_INTR_MASK, 0), 2,
            (void *)cmdParams, NULL, NULL, NULL);
```

Notes: **Generates a VMEBus error when clock A is not in a ready state.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.28 P65XX_LOOK

Command: **P65XX_LOOK**

Input Parameters: 65XX Address

Return Data: Contents of 65XX Address.

Description: Reads a location of the 65XX.

Example:

```
cmdParams[0] = 0x80000000;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_LOOK, 0), 1,
            (void *)cmdParams, NULL, &numRData, returnData);
```

Notes: **returnData contains the contents of the address 0x8000 0000.**

4.3.29 P65XX_SET

Command: **P65XX_SET**

Input Parameters: 65XX Address

Data Value

Return Data: none

Description: Sets a location on the 65XX.

Example:

```
cmdParams [0] = 0x40005000;
cmdParams [1] = 0x12345678;

status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_SET, 0), 2,
            (void *)cmdParams, NULL, &numRData, returnData);
```

Notes: **Location 0x4000 5000 is set to 0x1234 5678.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 65XX Firmware Commands (continued)

4.3.30 P65XX_GET_REVISION

Command: `P65XX_GET_REVISION`

Input Parameters: none

Return Data:

Revision	This value is the 65XX model in Bit D16 - D31 and the firmware revision in bits D0 - D15. For Example, 6509000a = Model 6509 Firmware Revision A.

Description: Gets the current version of the firmware. The firmware revision can also be viewed at the first location in Global SRAM or at the VME Base Address in A24/A32 Space.

Example:

```
status = P65XXSendCommandWait (CMD_COMM_PORT,
            P65XX_BUILD_CMD_WAIT(P65XX_GET_REVISION, 0), 0,
            (void *)NULL, NULL, &numRData, returnData);
```

Notes: ***returnData contains the current revision of the receiver.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 ëC40 Communication Port Specific Commands

The following two routines contained in **p65xxcmd.c** provide a way to send commands over the ëC40 Comm Port.

4.4.1 P65XXSendCommand

Command: **P65XXSendCommand**

Description: Sends a command to the 65XX board and immediately returns to the caller. No return status or data is sent to the caller.

4.4.2 P65XXSendCommandWait

Command: **P65XXSendCommandWait**

Description: Sends a command to the 65XX board and waits for return status and data before returning to the caller.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.5 VMEBus Specific Commands

Another method to send commands to the 65XX is using the VME interface. This is accomplished by writing the command and it's parameters into the 65XX Dual Port RAM and interrupting the ëC40. The firmware on the ëC40 processes the command and then clears the command location. The return status is stored in the next location followed by any return data.

Table 4-2: Commands & Addresses for the VMEBus

Description	Address
Command Address	VMEbase +0xC000
Input Channel/Receiver Mask	VMEBase + 0xC004
Parameter (n)	VMEBase + 0xC004 + (0x4 * N)
Examples:	Parameter 1 = VMEBase + 0xC008 Parameter 2 = VMEBase + 0xC00C

After the command completes:

Table 4-3: Sending Commands to the VMEBus

Description	Address	Notes
Command Address	VMEbase +0xC000 = 0	
Num Return Data/Status	VMEBase + 0xC004	The Num Return Data/Status Word contains the return status of the command in the D0..D15 Bits and the number of return data in D16..D32.
Return Data (n)	VMEBase + 0xC004 + (0x4 * N)	
Examples:	Return Data 1 = VMEBase + 0xC008 Return Data 2 = VMEBase + 0xC00C	

Return status can be set to; **`P65XX_COMMAND_COMPLETE`**, **`P65XX_COMMAND_ERROR`**, **`P65XX_PARAM_ERROR`**, or **`P65XX_OPEN_ERROR`**. These commands are described in the subsections that follow.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.5 VMEBus Specific Commands (continued)

4.5.1 P65XX_COMMAND_COMPLETE

Command: P65XX_COMMAND_COMPLETE

Description: Command completed successfully

4.5.2 P65XX_COMMAND_ERROR

Command: P65XX_COMMAND_ERROR

Description: Command not recognized

4.5.3 P65XX_PARAM_ERROR

Command: P65XX_PARAM_ERROR

Description: Input parameters error

4.5.4 P65XX_OPEN_ERROR

Command: P65XX_OPEN_ERROR

Description: Error opening connection.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 Model 6509 Specific Firmware Commands

4.6.1 P6509_SET_FILTER_RSP

Command: `P6509_SET_FILTER_RSP`

Input Parameters:

Handle	Identifies the open connection.
Receiver mask	Identifies the receivers to be modified

Filter Response	Filter Response:
	`P6509_80_PERCENT`
	`P6509_90_PERCENT`

Return Data: none

Description: Specifies the Filter Response setting for one or more receivers. This command is only applicable for the Model 6509.

Example:

```
cmdParam[0] = P6509_80_PERCENT

status = P65XXSendCommandWait (CMD_COMM_PORT,
           P65XX_BUILD_CMD_WAIT(P6509_SET_FILTER_RSP,
           handle), 2, (void *)cmdParams, P65XX_RECEIVER_3 |
           P65XX_RECEIVER_4, NULL, NULL);
```

Notes: **Filter Response for receiver 3 and 4 is set to 80 percent. Supports multiple receivers using a single command.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 Model 6509 Specific Firmware Commands (continued)

4.6.2 P6509_SET_INPUT_SYNC_MODE

Command: **P6509_SET_INPUT_SYNC_MODE**

Input Parameters:

Input Channel Mask	Identifies the inputs to be modified.
Input Sync Mode	The setting of this word is identical to bits D0 - D3 in the Mode Select/Channel A and Tag Enable/Channel B Control registers. Combine with the following contants to achieve the desired operation. **P6509_SW_AS_ON** **P6509_SW_AS_OFF** **P6509_SW_SS_ON** **P6509_SW_SS_OFF** **P6509_FP_SYNC_SRC** **P6509_SW_SYNC_SRC** **P6509_SYNC_OUT_DIS** **P6509_SYNC_OUT_ENA**

Return Data: none

Description: Specifies the Input Sync Source and Software Accumulator/System Sync Settings for one or more input channels. This command is only applicable for the Model 6509.

Example:

```
cmdParam[0] P6509_SW_SYNC_SRC | P6509_SW_SS_OFF;

status = P65XXSendCommandWait (CMD_COMM_PORT,
           P65XX_BUILD_CMD_WAIT(P6509_SET_INPUT_SYNC_MODE,
           0), 1, (void *)cmdParams, P65xx_INPUT_A |
           P65xx_INPUT_B, NULL, NULL);
```

Notes: **Input channels A& B SS signal is de-asserted.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 Model 6509 Specific Firmware Commands (continued)

4.6.3 P6509_SET_RCVR_SYNC_MODE

Command: **P65XX_SET_RCVR_SYNC_MODE**

Input Parameters:

Handle	Identifies the open connection.
Receiver mask	Identifies the receivers to be modified
Receiver Sync Mode	This setting is identical to Sync Mode Register in the GC1011A. Combine the following bits for desired operation: **P6509_GS_SS_OFF** **P6509_GS_AS_ON** **P6509_GS_AS_MUX** **P6509_GC_LD_FREQ** **P6509_GC_AS_FREQ** **P6509_GC_SS_DIAG** **P6509_GC_SYNC_TEST** **P6509_GC_SS_MUX**

Return Data: none

Description: Specifies the setting of the GC1011A Sync Mode Register for one of more receivers. Refer to Appendix B for a description of this register. This command is only applicable for the Model 6509.

Example:

```
cmdParams [0] =P6509_GC_AS_FREQ | P6509_GC_AS_ON |
               P6509_GC_AS_MUX | P6509_LD_FREQ;

status = P65XXSendCommandWait (CMD_COMM_PORT,
           P65XX_BUILD_CMD_WAIT(P6509_SET_RCVR_SYNC_MODE,
           handle), 1, (void *)cmdParams, P65xx_Receiver_3,
           NULL, NULL);
```

Notes: **Receiver 3 is set up to use the SS signal to sync the Graychip internal resources. This is necessary when synchronizing multiple Graychips.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP