Programmer's Reference

Digital Gamma Finder (DGF)

Differences between software version 2.63 (Pixie-4, Pixie-500) and version 4.xx (Pixie-4 Express, Pixie-500 Express)

Version 4.30, December 2016

XIA LLC

31057 Genstar Road
Hayward, CA 94544 USA

Phone: (510) 401-5760; Fax: (510) 401-5761
http://www.xia.com

Table of Contents

1 Overview

This reference should be read in parallel to the Pixie-4 programmer's manual. It only lists the differences to the older (non-express) software version 2.xx; it is not a full description.

Quick summary: Key changes when upgrading from Pixie-4 to Pixie-4 Express

LabView or other custom user interfaces calling the Pixie API functions should be changed as follows:

- Increase the boot file name array to 16, and specify valid filenames (files must be present) for the 10 entries shown in table 3.2
- Increase the Module_Parameter_Names array to 128 x 17 elements
- Update module and channel parameters according to table 3.5
- Recreate settings files by making changes to the "default" files provided.
- Specify serial numbers, not slot numbers for each module
- Select chassis type (MAX_NUMBER_MODULES) "62" for PXI Express chassis
- Start list mode runs with tun type 0x400 or 0x401
- When calling Pixie_Acquire data, always specify the lower 12 bits (e.g. use 0x4400 to poll progress on a 0x400 run) and a valid filename
- For list mode runs, periodically call runtask 0x4400 to receive an update of the **total** number of spills (2MB each) written to file by the interrupt service routine. (Older code used one spill and/or returned the incremental number of spills since last function call)
- While list mode file processing routines (0x700#) return equivalent data, it should be understood that
  - the data format in the file has changed
  - the data is organized into one file per module, one channel per event
  - runtask 0x7008 i) takes an optional input of a coincidence window to search for simultaneous events from other channel, ii) returns the ADC rate in MHz so that the user interface can display waveforms in correct time scale, and iii) returns many values as full 32 bit numbers where formerly only 16 bit were used.
  - List mode file processing run task 0x7021 processes a file in the new Pixie-4 Express format (.b##) and saves the data in Pixie-4 (.bin) format.
- Third party list mode file parsing routines have to be updated for the new data format (unless the translater routine 0x7021 is used).

2 Pixie API

Pixie_Get_Par_Idx and **Pixie_Get_Par_Name** are new functions employed to assist in using **Pixie_User_Par_IO** when the Pixie C library is linked dynamically to the user's application (.dll library). In this case the names of the user-controlled variables and their indexes in arrays are not accessible to the user's application. The user, therefore, can inquire the library regarding a particular parameter of interest either through an index or a string. As an alternative, the list of names in globals.c can be used.

Pixie_Hand_Down_Names

When downloading boot file names with the ALL_FILES keyword, *Names* is a string array which has N_BOOT_FILES elements. In version 4.xx, N_BOOT_FILES is defined as **16** (formerly 7) and the order of names changed somewhat. See table 3.2 for details

Module_Parameter_Names can currently hold **128** names (formerly 64)

Pixie_Boot_System

no change

Pixie_User_Par_IO

A number of parameters have been added, others have been removed. See table 3.5 for details

Table 2.1: The Combination of User_Par_Name and User_Par_Values.

User_Par_Type	User_Par_Values		
	Name	Size	Data Type
SYSTEM	System_Parameter_Values	64	Double precision
MODULE	Module_Parameter_Values	128×17	Double precision
CHANNEL	Channel_Parameter_Values	64×17×4	Double precision

Pixie_Get_Par_Name

new function

Pixie_Get_Par_Idx

new function

Pixie_Acquire_Data

List mode run types 0x400-0x401 have been added as run type choices (see user manual for a description). Lower 12 bits of current run type should be included in all function calls, e.g. use 0x4400 to poll progress on a 0x400 run in a Pixie-4 Express, use 0x3103 to stop a 0x103 run in a Pixie-4. The string variable *file_name* should always be specified with a valid file name.

Task 0x0085 has been added for a faster adjustment of DC offsets in the DSP (not for Pixie-4)

The sequence of run types called during a list mode run has been simplified:

For the Pixie-4, *Run_Type* = 0x4### or 0x40FF is used to check if the run is stopped, which implies data is ready for readout to file. The top level program then (i) calls *Run_Type* = 0x6### to save data, (ii) increments the spill count by one, and (iii) calls *Run_Type* = 0x2### to resume the run or (iv) ends data acquisition if the requested number of spills is reached.

For the Pixie-500 Express and Pixie-4 Express, *Run_Type* = 0x4400 or 0x4401 are used instead to check if data is ready and save data to file if ready. In interrupt mode, the polling routine really only updates the **total** number of spills written to file, the readout is handled by the interrupt service routine. The return value informs the top level program how many spills have been saved. The top level program only has to end the run once the the requested number of spills is reached – with the important change to use the return value as total number of spills, not as an increment.

Task 0x7008 uses UserData[3] as a new input/output parameter, and returns the ADC digitization rate in UserData[1].

List mode file processing run task 0x7020 has been added to perform data error checks and save corrected data in a new file. List mode file processing run task 0x7021 has been added to process a .b## file and save the data in (Pixie-4) .bin format. Legacy run type 0x7003 is not supported for .b## files.

Pixie_Set_Current_ModChan

no change

Pixie_Buffer_IO

no change

Options for Compiling Pixie API

The option to compile the Pixie API as a static library has been removed. Sample C programs use the DLL.

There are two new compiler switches (options) that has to be set by the makefiles or project files:

- WINDRIVER_API, if "defined", enables those sections of the code related to the PCIe I/O for the Pixie-500 Express. If compiling purely for the Pixie-4 using the PLX PCI interface, this option can be undefined
- WI64, if "defined" enables code specific for 64 bit Windows. Currently, there is no real difference in the source code itself (only linked libraries, which are handled by the compiler), except that the C library built number's highest two digit are changed from "0x32" to "0x64"

The Pixie-4 C library is ANSI C compatible as much as possible, but a number of non-standard functions are used. These are:

- Sleep. Throughout the code, the function *Pixie_Sleep* is used to wait for a time (in ms). In utilities.c, this function is implemented using the Windows API function *Sleep*. For Linux, *usleep* is used.
-

Note that the Pixie software uses version 6.5 of the PLX drivers and version 11.2 of WinDriver.

3 Control Pixie Modules via C program

3.1 Initializing

The number and order of the boot file names has been changed, see table 3.2. Pixie-500 modules (no Express) are not supported in this software. The API expects 10 files to be present, even if they are not used for the particular module type currently used – all defined entries except the Pixie-500 files (index 7, 10).

Some module types, in particular the Pixie-4 Express, come in several variants. The files for two major Pixie-4 Express variants, 16/125 and 14/500, are listed in index 0 and index 8. For other variants, the default name has to be changed to the appropriate variant file name.

Table 3.2: File Names in All_Files.

Index	Default File Name	Note
0	C:\XIA\Pixie\Firmware\P4e_**16_125**_vdo.bin	FPGA configuration (Pixie-4 Express) **numbers indicate ADC bits and rate**
1	C:\XIA\Pixie\Firmware\syspixie_revC.bin	Communication FPGA configuration (Pixie-4 Rev C, D, E)
2	C:\XIA\Pixie\Firmware\pixie.bin	Signal processing FPGA configuration (Pixie-4)
3	C:\XIA\Pixie\DSP\PXIcode.bin	DSP executable binary code for the Pixie-4
4	C:\XIA\Pixie\Configuration\default.set	Settings file
5	C:\XIA\Pixie\DSP\P500e.var	File of DSP I/O variable names Do not use the var file from Pixie-4 DSP!
6	C:\XIA\Pixie\DSP\PXIcode.lst	File of DSP memory variable names for the Pixie-4
7	C:\XIA\Pixie\Firmware\syspixie_RevC.bin unused	Communication FPGA configuration (Pixie-500 Rev B)
8	C:\XIA\Pixie\Firmware\P4e_14_500_vdo.bin	FPGA configuration (Pixie-4 Express 14/500)
9	C:\XIA\Pixie\Firmware\p500e_zdt.bin	FPGA configuration (Pixie-500 Express)
10	C:\XIA\Pixie\DSP\P500code.bin unused	DSP executable binary code for the Pixie-500
11	C:\XIA\Pixie\DSP\P500e.ldr	DSP executable binary code for the Pixie-500 Express
12	C:\XIA\Pixie\DSP\P500e.lst	File of DSP memory variable names for the Pixie-500 Express
13-15	–	unused/reserved

Table 3.3 lists the System_Parameter_Values global variables and the definition of their allowed values.

Table 3.3: Initialization of Module_Global_Values.

Module_Global_Names	Module_Global_ Values Default Value	Definition
NUMBER_MODULES	2	The total number of Pixie-4 modules present Range: 1.. PRESET_MAX_MODULES
OFFLINE_ANALYSIS	0	0 for online analysis, 1 for offline analysis (no I/O with modules)
AUTO_PROCESSLMDATA	0	0 to not writing results in a .dat file while parsing list mode output data 1 to write standard .dat file during parsing 2 to write extended dt2 file during parsing 3 to write extended dt3 file during parsing
MAX_NUMBER_MODULES	7	Select chassis type 7 = 4,5,6,8-slot chassis 13 = 14-slot XIA 6U chassis 17 = 14,18-slot NI 3U chassis 62 = 8-slot PXI/PXIe NI chassis 1062Q use 62 for any PXI Express chassis
KEEP_CW	1	0 Channel COINC_DELAY is set by user. 1 Channel COINC_DELAY is automatically adjusted to compensate for difference in energy filter length Note: not required for Pixie-500 Express and Pixie-4 Express
SLOT_WAVE[0]	110	Module 0 is s/n 110 (use physical slot number for Pixie-4)
SLOT_WAVE[1]	111	Module 1 is s/n 111
SLOT_WAVE[...]	...	Module ... is s/n ...

3.2 Setting DSP variables

In version 4.xx, there are 512 DSP variables (previously 416).

Table 3.5a: Descriptions of System Global Variables

System_Parameter_Names	I/O Type	Unit	Corresponding DSP Variables
NUMBER_MODULES	Read/Write	N/A	N/A
OFFLINE_ANALYSIS	Read/Write	N/A	N/A
AUTO_PROCESSLMDATA	Read/Write	N/A	N/A
C_LIBRARY_RELEASE	Read only	N/A	N/A
C_LIBRARY_BUILD	Read only	N/A	N/A
KEEP_CW	Read/Write	N/A	N/A
SLOT_WAVE	Read/Write	N/A	N/A
SLOT_WAVE is followed by up to 16 slot entries. Entries are addressed as SLOT_WAVE[N]			

Table 3.5b: Descriptions of Module Global Variables in Pixie-4.
~~Removed variables are crossed out,~~ **New ones are bold**

Module_Parameter_Names	I/O Type	Unit	Corresponding DSP Variables
MODULE_NUMBER	Read only	N/A	MODNUM
MODULE_CSRA	Read/Write	N/A	MODCSRA
MODULE_CSRB	Read/Write	N/A	MODCSRB
~~MODULE_FORMAT~~	~~Read/Write~~	~~N/A~~	~~MODFORMAT~~
C_CONTROL	**Read/Write**	**N/A**	**CCONTROL**
MAX_EVENTS	Read/Write	N/A	MAXEVENTS
COINCIDENCE_PATTERN	Read/Write	N/A	COINCPATTERN
ACTUAL_COINCIDENCE_WAIT	Read/Write	ns	COINCWAIT
MIN_COINCIDENCE_WAIT	Read only	ticks	COINCWAIT
SYNCH_WAIT	Read/Write	N/A	SYNCHWAIT
IN_SYNCH	Read/Write	N/A	INSYNCH
RUN_TYPE	Write only	N/A	RUNTASK
FILTER_RANGE	Read/Write	N/A	FILTERRANGE
MODULEPATTERN	Read/Write	N/A	MODULEPATTERN
NNSHAREPATTERN	Read/Write	N/A	NNSHAREPATTERN
DBLBUFCSR	Read/Write	N/A	DBLBUFCSR
MODULE_CSRC	Read/Write	N/A	MODCSRC
BUFFER_HEAD_LENGTH	Read only	N/A	BUFHEADLEN
EVENT_HEAD_LENGTH	Read only	N/A	EVENTHEADLEN
CHANNEL_HEAD_LENGTH	Read only	N/A	CHANHEADLEN
~~OUTPUT_BUFFER_LENGTH~~	~~Read only~~	~~N/A~~	~~LOUTBUFFER~~
NUMBER_EVENTS	Read only	N/A	NUMEVENTSX, A, B
RUN_TIME	Read only	s	RUNTIMEX, A, B, C
EVENT_RATE	Read only	cps	NUMEVENTSX, A, B RUNTIMEX, A, B, C
TOTAL_TIME	Read only	s	TOTALTIMEX, A, B, C
BOARD_VERSION	Read only	N/A	<Hardware PROM>
SERIAL_NUMBER	Read only	N/A	<Hardware PROM>
DSP_RELEASE	Read only	N/A	DSPRELEASE
DSP_BUILD	Read only	N/A	DSPBUILD
FIPPI_ID	Read only	N/A	FIPPIID
SYSTEM_ID	Read only	N/A	HARDWAREID

XET_DELAY	Read/Write	N/A	XETDELAY
PDM_MASKA	Read/Write	N/A	PDMMASKA
PDM_MASKB	Read/Write	N/A	PDMMASKB
PDM_MASKC	Read/Write	N/A	PDMMASKC
USER_IN 0..15	**Read/Write**	**N/A**	**USERIN 0..15**
USER_OUT 0..15	**Read Only**	**N/A**	**USEROUT 0..15**
ADC_RATE	**Read Only**	**MHz**	**<Hardware PROM>**
ADC_BITS	**Read Only**	**N/A**	**<Hardware PROM>**

Table 3.5c: Descriptions of Channel Global Variables in Pixie-4.

Channel_Parameter_Names	I/O Type	Unit	Corresponding DSP Variables
CHANNEL_CSRA	Read/Write	N/A	CHANCSRA
CHANNEL_CSRB	Read/Write	N/A	CHANCSRB
ENERGY_RISETIME	Read/Write	µs	SLOWLENGTH
ENERGY_FLATTOP	Read/Write	µs	SLOWGAP
TRIGGER_RISETIME	Read/Write	µs	FASTLENGTH
TRIGGER_FLATTOP	Read/Write	µs	FASTGAP
TRIGGER_THRESHOLD	Read/Write	N/A	FASTTHRESH
VGAIN	Read/Write	V/V	GAINDAC
VOFFSET	Read/Write	V	TRACKDAC
TRACE_LENGTH	Read/Write	µs	TRACELENGTH
TRACE_DELAY	Read/Write	µs	USERDELAY
COINC_DELAY	Read/Write	µs	COINCDELAY, RESETDELAY
PSA_START	Read/Write	µs	PSAOFFSET
PSA_END	Read/Write	µs	PSAOFFSET , PSALENGTH
~~EMIN~~	~~Read/Write~~	~~N/A~~	~~ENERGYLOW~~
BINFACTOR	Read/Write	N/A	LOG2EBIN
TAU	Read/Write	µs	PREAMPTAUA, PREAMPTAUB
BLCUT	Read/Write	N/A	BLCUT
XDT	Read/Write	N/A	XWAIT, XAVG
BASELINE_PERCENT	Read/Write	N/A	BASELINEPERCENT
CFD_THRESHOLD	Read/Write	N/A	CFDTHR
INTEGRATOR	Read/Write	N/A	INTEGRATOR
CHANNEL_CSRC	Read/Write	N/A	CHANCSRC
GATE_WINDOW	Read/Write	µs	GATEWINDOW
GATE_DELAY	Read/Write	µs	GATEDELAY
BLAVG	Read/Write	N/A	LOG2BWEIGHT
COUNT_TIME	Read only	s	**COUNTTIMEX, A, B, C**
INPUT_COUNT_RATE	Read only	cps	FASTPEAKSX, A, B, C COUNTTIMEX, A, B, C, FTDTX, A, B, C,
FAST_PEAKS	Read only	N/A	FASTPEAKSX, A, B, C
OUTPUT_COUNT_RATE	Read only	cps	NOUTX,A,B, COUNTTIMEX, A, B, C
NOUT	Read only	N/A	NOUTX,A,B,
GATE_RATE	Read only	cps	GCOUNTX, A, B COUNTTIMEX, A, B, C
GATE_COUNTS	Read only	N/A	GCOUNTX,A,B
FTDT	Read only	s	FTDTX, A. B, C
SFDT	Read only	s	SFDTX, A, B, C

GDT	Read only	s	GDTX, A, B ,C
CURRENT_ICR	Read only	cps	ICR
CURRENT_OORF	Read only	%	OORF
PSM_GAIN_AVG	reserved	N/A	
PSM_GAIN_AVG_LEN	reserved	N/A	
PSM_TEMP_AVG	reserved	N/A	
PSM_TEMP_AVG_LEN	reserved	N/A	
PSM_GAIN_CORR	reserved	N/A	
QDC0_LENGTH	Read/Write	samples	QDC0LENGTHX
QDC1_LENGTH	Read/Write	samples	QDC0DELAYX
QDC0_DELAY	Read/Write	samples	QDC1LENGTHX
QDC1_DELAY	Read/Write	samples	QDC1DELAYX
NPPI	Read only	N/A	NPPIX, A, B COUNTTIMEX, A, B, C
PASS_PILEUP_RATE	Read only	cps	NPPIX, A, B

3.3 Access spectrum memory or list mode data

3.3.1 Access spectrum memory

No changes

3.3.2 Access list mode data

In the Pixie-4, there are two data buffers to choose from: the DSP's local I/O buffer (8K 16-bit words), and a section of the external memory (128K 32-bit words). Bit 1 in the variable MODCSRA selects which data buffer is filled during the run:

- If the external memory is chosen to hold the output data, the local buffer is transferred to the external memory when it has been filled. Then the run resumes automatically, without interference from the host, until 32 local buffers have been transferred. The data can then be read from external memory in a fast block read starting from location 0x00020000.
- If the local buffer is chosen, the run stops when the local buffer is filled. The data has to be read out from local memory.

With any data buffer, you can do any number of runs in a row. The first run would be started as a NEW run. This clears all histograms and run statistics in the memory. Once the data has been read out, you can RESUME running. Each RESUME run will acquire another either 32 or 1 8K buffers of data, depending on which buffer has been chosen. In a RESUME run the histogram memory is kept intact and you can accumulate spectra over many runs. The example code shown below illustrates this.

In the Pixie-4 and Pixie-500 Express, there is only one data buffer, the 256 MB SDRAM FIFO. Runs do not stop and resume; data is continuously added to the SDRAM and transferred to a 2MB buffer on the host PC. The host has to check periodically if the 2MB buffer is filled and write the data to file. This is normally handled by an interrupt routine initiated by the module, and the top level interface only has to read the number of buffers stored as updated by the interrupt routine. There still is an option for the top level interface to poll status and write each individual buffer, which however is much slower in the Igor based Pixie Viewer.

There usually is still some data left in the buffer at the moment when a run is stopped by timeout or "number of spills reached" from the top level interface. This data is read out as part of the run stop routine. The total number of buffers is therefore somewhat larger than the requested value.

To distinguish list mode files from different module types, the following information can be used.

- File Names:
  Pixie-4 files are named xxx.bin.
  Pixie-4 Express and Pixie-500 Express files are named xxx.b## (##=module number)

- "RunFormat" value (word 2 of the file)
  For Pixie-4 the value is 0x2000 plus the runtask. The sample rate is 75 MHz.
  For Pixie-500, the value is 0x4000 plus the runtask. The sample rate is 500 MHz
  For Pixie-500 Express, the value is the runtask. The sample rate is 500 MHz
  For Pixie-4 Express, the value is the runtask. The sample rate is 125-500 MHz, details can be extracted from the BoardVersion value (word 7 of the file).

An Example Code Illustrating How to Access List Mode Data (Pixie-4)

```
S32 retval;
U8 direction, modnum, channum;
U32 User_Data[131072]; // an array for holding the MCA spectrum data of
                       // 4 channels
U16 k, Nruns;
char *DataFile = {"C:\\XIA\\Pixie4\\PulseShape\\Data.bin"};

direction = 0;   // download
modnum = 0;      // Module #0
channum = 0;     // Channel #0
Nruns = 10;      // 10 repeated list mode runs
k = 0;           // initialize counter

// start a general list mode run
retval = Pixie_Acquire_Data(0x1100, User_Data, DataFile, modnum);
  if( retval < 0 // Error handling

do
{
  // wait until run has ended
  while( ! Pixie_Acquire_Data(0x4100, User_Data, DataFile, modnum) ) {;}

  // read out the list mode data and save it to a file
  retval = Pixie_Acquire_Data(0x6100, User_Data, DataFile, modnum);
  if( retval < 0 // Error handling

  k ++;
  if(k > Nruns)
  {
    break;
  }

  // issue RESUME RUN command
  retval = Pixie_Acquire_Data(0x2100, User_Data, DataFile, modnum);
    if( retval < 0 // Error handling

}while(1);

// read out the MCA spectrum and put it to array User_Data
retval = Pixie_Acquire_Data(0x9001, User_Data, DataFile, modnum);
  if( retval < 0 // Error handling
```

An Example Code Illustrating How to Access List Mode Data (Pixie-4 Express and Pixie-500 Express)

```
S32 retval;
U8 direction, modnum, channum;
U32 User_Data[131072]; // an array for holding the MCA spectrum data of
                       // 4 channels
U16 k, Nruns;
char *DataFile = {"C:\\XIA\\Pixie4\\PulseShape\\Data.bin"};

direction = 0;   // download
modnum = 0;      // Module #0
channum = 0;     // Channel #0
Nruns = 10;      // 10 spills


// start a general list mode run
retval = Pixie_Acquire_Data(0x1400, User_Data, DataFile, modnum);
  if( retval < 0 // Error handling

// check for data ready, if so it is saved to file
while(  Nruns>retval)
  {retval = Pixie_Acquire_Data(0x4400,User_Data,DataFile,modnum) }

// stop list mode run, save remaining data to file
retval = Pixie_Acquire_Data(0x3400, User_Data, DataFile, modnum);
  if( retval < 0 // Error handling


// read out the MCA spectrum and put it to array User_Data
retval = Pixie_Acquire_Data(0x9001, User_Data, DataFile, modnum);
  if( retval < 0 // Error handling
```

Processing list mode data to extract the number of events or to view individual events is unchanged on the top level (runtasks 0x700#). Internally, the code distinguishes between Pixie-4 .bin files and Pixie4/500 Express .b## files. The latter are also checked (and marked) for data transmission errors. In task 0x7008, argument 3 of the data exchange wave specifies a window to search for coincident events near the requested one in the other 3 channels. Argument 3 is then overwritten with the trace length of channel 0. Argument 1 is overwritten with the ADC rate in MHz as extracted from the file, which can be used to set the time scale in a display of the waveform.

4 User Accessible DSP Variables

General changes are as follows:

- The data memory space, as seen by the host computer, starts at 0x4000 in the Pixie-4 and at 0 in the Pixie-4 Express and Pixie-500 Express.
- DSP output parameters have been increased to 256 (was 160) for all module types.
- Independent .var files are generated for each DSP type. Even though all modules have the same variable map, do not use .var files from the Pixie-4 DSP compilation.

4.1 Module input parameters

MODCSRA: The Module Control and Status Register A

Bits 1,3,4,5,6,7,9,10,12,13,14: Ignored for Pixie-4 Express and Pixie-500 Express

Bit 11: Bypass SDRAM (use smaller/faster in-FPGA buffer instead)
Ignored for Pixie-4 and Pixie-500 Express

MODFORMAT: Removed**.**

RUNTASK: This variable tells the Pixie module what kind of run to start in response to a run start request. Seven run tasks are currently supported.

The variable was previously defined as write only. Now it is read/write and a write is "broadcast" to all modules. Starting a control task does not change the run task.

RunTask	Mode	Trace Capture	CHAN HEADLEN
0	Slow control run	N/A	N/A
256 (0x100)	Standard list mode (P4)	Yes	9
257 (0x101)	Compressed list mode (P4)	Yes	9
258 (0x102)	Compressed list mode (P4)	Yes	4
259 (0x103)	Compressed list mode (P4)	Yes	2
769 (0x301)	MCA mode	No	N/A
1024 (0x400)	Standard list mode (P500e/P4e)	Yes	32
1025 (0x401)	Text list mode (P500e/P4e)	No	32

CONTROLTASK:

See section 4.5 for a list of acceptable values. Some are new

MAXEVENTS:

The parameter is ignored in runtypes 0x301 and 0x400-0x403.

COINCWAIT: Duration of the coincidence time window in clock cycles.

For the Pixie-4, the window is **from** the first event validation **until** the final hit pattern is latched. For the Pixie-4 Express and Pixie-500 Express the window is **around** the fast trigger at the rising edge of the pulse. The KeepCW and CoincDelay parameters are ignored.

Clock cycles are 13.3ns for both Pixie-4, 8ns for Pixie-500 Express and Pixie-4 Express)

Constraints**:** COINCWAIT >= 1
COINCWAIT <= 65531 for Pixie-4 and Pixie-500
COINCWAIT <= min(127, minEFT)
for Pixie-500 Express and Pixie-4 Express

minEFT is the smallest energy filter time of the four channels,
(SLOWLENGTH+SLOWGAP)*2^ FILTERRANGE

INSYNCH: For the Pixie-4, INSYNCH is automatically set to 1 by the DSP after run start. This options should normally be only set for the first run to preserve time correlation across runs.

For the Pixie-500 Express and Pixie-4 Express, INSYNCH is unchanged by the DSP. Module timers are synchronized at every DSP reboot. This option should therefore only be set if all runs should start with a time stamp near zero.

RESUME:

Ignored for Pixie-500 Express and Pixie-4 Express

NNSHAREPATTERN:

Currently not implemented for the Pixie-500 Express and Pixie-4 Express

DBLBUFCSR:

Not recommended for Pixie-4. Ignored for Pixie-500 Express and Pixie-4 Express

CCONTROL: A register containing several bits to specify options for the C library execution. This variable is part of the DSP parameters so that it will be saved with all the others in the .set file, but it is not used in the DSP.

Bit 0-3: Reserved.

Bit 4: If set, print debug messages during the boot process.

Bit 5: If set, print detailed error messages during error check of the list mode data.

Bit 6: If set, print additional debug messages during error check of the list mode data.

Bit 7: If set, print other debug messages that are normally suppressed.

Bit 8: If cleared, run list mode data acquisitions in interrupt mode. The interrupts automatically write the list mode data to file when a buffer in PC memory is filled by the DMA data transfer. Runtask 0x440# called by the top level program (e.g. Pixie Viewer) does nothing more than return the current total number of buffers written to file. Runtask 0x440# can be executed quite infrequently.
If set, run list mode data acquisition in polling mode. No interrupts are active. Runtask 0x440# called by the top level program checks for list mode data to be ready to be written to file, and if so it performs the write. Fast, frequent execution of runtask 0x440# is essential to avoid readout dead time.
Ignored for Pixie-4

Bit 9: If set, check list mode data for errors before writing to file. This bit must be set for Run type 0x401. Ignored for Pixie-4

Bit 10-15: Reserved.

Igor controls: Table entry in the *Run Chassis Setup* panel

C global C_CONTROL

ControlTask to apply change: n/a

XETDELAY:

Currently not implemented for the Pixie-500 Express and Pixie-4 Express

XDATLENGTH:

Removed

4.2 Channel input variables

All channel-0 variables end with "0", channel-1 variables end with "1", etc. In the following explanations the numerical suffix has been removed. Thus, e.g., CHANCSRA0 becomes CHANCSRA, etc.

CHANCSRA: The control and status register bits switch on/off various aspects of the Pixie-4 operation.

Bits 3, 10, 13 ignored for Pixie-4 Express and Pixie-500 Express

Bit 12 ignored for Pixie-500 Express

Bit 14 “always” on for Pixie-4 Express and Pixie-500 Express

CHANCSRC: Control and status register C.

Bits 2, 4, 7, 8, 9, 11, 12 ignored for Pixie-4 Express and Pixie-500 Express

Bits 0, 1, 10 ignored for Pixie-500 Express

Bits 13 ignored for Pixie-4 and Pixie-500 Express

SGA: The index of the relay combinations of the switchable gain amplifier.

For the Pixie-4, the analog SGA gain is G = (1+Rf/Rg)/2 with

Rf = 2150 - 120*((SGA & 0x1)>0) - 270*((SGA &0x2)>0) - 560*((SGA &0x4)>0)

Rg = 1320 - 100*((SGA & 0x10)>0) - 300*((SGA & 0x20)>0)- 820*((SGA & 0x40)>0)

The C library computes the closest SGA setting for a given voltage gain VGAIN and adjusts the parameter DIGGAIN to compensate differences between VGAIN and the gain from SGA up to ±10%.

For the Pixie-500 Express, only SGA bit zero is active and it changes the gain from 1 to 3.

For the Pixie-4 Express, only SGA bit 0..2 are active and it changes the gain as follows. In the future, this may depend on the hardware version, for example high rate digitizers may have fewer gain options.

SGA = 0	gain = 1.6
SGA = 1	gain = 6.7
SGA = 2	gain = 2.4
SGA = 3	gain = 9.9
SGA = 4	gain = 3.5
SGA = 5	gain = 14.7
SGA = 6	gain = 5.4
SGA = 7	gain = 22.6

SLOWLENGTH:
SLOWGAP:
FASTLENGTH:
FASTGAP:

dt = 13.3ns for Pixie-4, 8ns for Pixie-4 Express and Pixie-500 Express

PEAKSAMPLE: Removed

MAXWIDTH:

dt = 13.3ns for Pixie-4, 8ns for Pixie-4 Express and Pixie-500 Express

PAFLENGTH: Removed

TRIGGERDELAY: Removed

COINCDELAY:

In the Pixie-4, this variable is used to delay (individually for each channel) the start of the coincidence window. In the Pixie-4 Express, this variable is used to delay the incoming ADC data. Currently not implemented for Pixie-500 Express

The Igor control and the C global are in units of ns, the DSP variable is in units of delay cycles (13.3ns for Pixie-4, 4*2ns for the Pixie-4 Express with 500 MHz ADCs, 2*8ns for the Pixie-4 Express with 125 MHz ADCs).

Constraints**:** COINCDELAY >= 0
COINCDELAY <= 65533 (for Pixie-4)
COINCDELAY <= 126 (for Pixie-4 Express)

RESETDELAY:

The default value is max(29, CD+4) and should normally not be changed by the user. CD is the value of COINCDELAY in processing clock ticks:
CD = COINCDELAY for the Pixie-4

Ignored for Pixie-500 Express and Pixie-4 Express

TRACELENGTH:

dt = 13.3ns for Pixie-4, 2ns for the Pixie-500 Express and Pixie-4 Express (500), 8ns for the Pixie-4 Express (125)

TRACELENGTH must be a multiple of 32 for the Pixie-500 Express and Pixie-4 Express

TLMAX = 1024 for Pixie-4

TLMAX = min(4096, EFT) for Pixie-500 Express and Pixie-4 Express, where EFT is the energy filter time of the channel, (SLOWLENGTH+SLOWGAP)*2^ FILTERRANGE

USERDELAY:

Constraints**:** USERDELAY >= 0
USERDELAY <= TRACELENGTH for Pixie-4
USERDELAY <= 2047 for Pixie-500 Express

dt = 13.3ns for Pixie-4, 2ns for Pixie-500 Express, 8ns for the Pixie-4 Express

USERDELAY must be a multiple of 4 for the Pixie-4 Express and the Pixie-500 Express

FTPWIDTH: Removed

GATEDELAY, GATEWINDOW:

Not yet implemented for Pixie-500 Express, maximum for GATEDELAY is 255 for Pixie-4 and 127 for Pixie-4 Express

XWAIT:

for Pixie-4

ΔT = XWAIT *13.3ns

Constraints: XWAIT >= 4
XWAIT <= 65533
If XWAIT >12, it is limited to XWAIT = 13+ N*5

If XWAIT > 12, a filter is implemented during the acquisition to return each data point as the average over (XWAIT-3)/5 samples.

for Pixie-500 Express and Pixie-4 Express

ΔT = XWAIT *5 *13.3ns

Constraints: XWAIT must be a power of 2 between 1 and 8192

If XWAIT > 1, a filter is implemented during the acquisition to return each data point as the average over XWAIT samples.

XAVG:

for Pixie-4

For a given dT (in μs), calculate the integer intdt = dT/0.0133
If intdt > 13, XAVG = floor(65536/((intdt-3)/5))
If intdt < 13, XAVG = 65535.

for Pixie-500 Express and Pixie-4 Express

XAVG = $\log_2$ XWAIT

ENERGYLOW: Removed

PSAOFFSET, PSALENGTH:

dt = 13.3ns for Pixie-4

Currently not used for Pixie-500 Express and Pixie-4 Express

QDC#DELAY, QDC#LENGTH: These two parameters specify the position and length of two "QDC" sums on the rising edge of a pulse, computed by the FPGA in real time. (#=0,1) The position is relative to the rising edge trigger point. The units are in ADC samples. Results are placed in the "User PSA" fields in the list mode data.

Constraints**:** QDC#LENGTH >= 4
QDC# LENGTH <= 120
QDC#LENGTH must be multiple of 4
QDC# DELAY >= 0
QDC# DELAY <= 380
QDC1DELAY + QDC1LENGTH
> QDC0DELAY + QDC0LENGTH
The lowest 2 bits of QDC1DELAY and QDC0DELAY must be the same.

Ignored for Pixie-4, not yet implemented in Pixie-4 Express

CFDREG: Removed

INTEGRATOR: This variable controls the energy reconstruction in the DSP.

INTEGRATOR = 0: normal trapezoidal filtering

INTEGRATOR = 1: use gap sum only; good for scintillator signals

INTEGRATOR = 2: ignore gap sum; pulse height=leading sum – trailing sum; good for step-like pulses.

For Pixie-4: INTEGRATOR = 3,4,5: same as 1, but multiply energy by 2, 4, or 8

For Pixie-4 Express and Pixie-500 Express, the resulting pulse height value is scaled with the length of gap sum (INTEGRATOR = 1) or leading sum (INTEGRATOR =2). In addition, the result is multiplied with the decay time tau, which is here used as an arbitrary scaling factor. If tau is set to 1, the pulse height will always fall into the visible range of the spectrum.

PREAMPTAUA, PREAMPTAUB: In integrator mode (INTEGRATOR >0), the decay time correction is meaningless. In the Pixie-4 Express and Pixie-500 Express, PREAMPTAU is therefore used as an arbitrary scaling factor for the computed pulse height in integrator mode. In the Pixie-4, PREAMPTAU is ingnored in integrator mode

4.3 Module output parameters

We now show the output variables, again beginning with module variables and continuing afterwards with the channel variables. The output data block begins at the address 0x4100. Note, however, that this address could change. The output data block comprises of 160 words; 1 block of 32 is reserved for module data; 4 blocks of 32 words each hold channel data.

DECIMATION: This variable is a copy of the input parameter FILTERRANGE. It is copied as an output parameter for backwards compatibility

REALTIMEA, REALTIMEB: The main purpose of this parameter is to see that the DSP is active, not for timing. RealTimeA is unused. Time units are 13.33ns.

RUNTIMEX, RUNTIMEA, RUNTIMEB, RUNTIMEC:

Increased to 64 possible bits. Currently use only 48.

TOTALTIMEX, TOTALTIMEA, TOTALTIMEB, TOTALTIMEC:

Increased to 64 possible bits. Currently use only 48.

GSLTTIMEA, GSLTTIMEB, GSLTTIMEC: Removed.

NUMEVENTSX, NUMEVENTSA, NUMEVENTSB:

Increased to 48 bits.

BUFHEADLEN: Currently, BUFHEADLEN is 6 in runtask 0x10# and 32 in runtask 0x400.

EVENTHEADLEN: Currently, EVENTHEADLEN is 3 in runtask 0x10# and 0 in runtask 0x400

CHANHEADLEN: CHANHEADLEN varies between 2 and 32 words depending on the run type (see RUNTASK).

AOUTBUFFER: Address of the IO data buffer. The addresses are generated by the assembler/linker when creating the executable. On power up the DSP code makes these values accessible to the user. Note that the addresses may change with every new compilation. Therefore your code should never assume to find any given buffer at a fixed address.

LOUTBUFFER: Removed. The length of the IO data buffer is always 8192.

HARDWAREID: ID of the system FPGA configuration.

HARDVARIANT: Removed

FIFOLENGTH: Length of the onboard FIFOs, measured in storage locations.

FIPPIID: ID of the FiPPI FPGA configuration

FIPPIVARIANT: Removed

INTRFCID: Unused

INTRFCVARIANT: Removed

DSPRELEASE: DSP software release number

DSPBUILD: DSP software build number

4.4 Channel output parameters

The following channel variables contain run statistics. Again the variable names carry the channel number as a suffix. For example the COUNTTIME words for channel 2 are COUNTTIMEX2, COUNTTIMEA2, COUNTTIMEB2, COUNTTIMEC2. Channel numbers run from 0 to 3.

COUNTTIMEX, COUNTTIMEA, COUNTTIMEB, COUNTTIMEC:
Formerly called LIVETIME.
Increased counters to 64 bits. Currently 48 bits are used.
dt = 13.3ns *16 for Pixie-4, 8ns *32 for Pixie-4 Express and Pixie-500 Express

FASTPEAKSX, FASTPEAKSA, FASTPEAKSB:
Increased to 48 bits

FTDTX, FTDTA, FTDTB, FTDTC:
Increased to 64 bits. Currently 48 bits are used.
dt = 13.3ns *16 for Pixie-4, 8ns *32 for Pixie-4 Express and Pixie-500 Express

SFDTX, SFDTA, SFDTB, SFDTC:
Increased to 64 bits. Currently 48 bits are used.
dt = 13.3ns *16 for Pixie-4, 8ns *32 for Pixie-4 Express and Pixie-500 Express

GCOUNTX, GCOUNTA, GCOUNTB:

Increased to 48 bits
Not yet implemented in Pixie-4 Express and Pixie-500 Express

NOUTX, NOUTA, NOUTB:

Increased to 48 bits

GDTX, GDTA, GDTB, GDTC:

Increased to 64 bits. Currently 48 bits are used.
dt = 13.3ns *16 for Pixie-4, 8ns *32 for Pixie-4 Express and Pixie-500 Express
Not yet implemented in Pixie-4 Express and Pixie-500 Express

NPPIX, NPPIA, NPPIB:

New counter counting pulses that passed pileup inspection. Useful for dead time correction in cases where pulses are removed by gate or coincidence conditions.

4.5 Control Tasks

Control Task 0x1: **Connect inputs**
Pixie-4 only

Control Task 0x2: **Disconnect inputs**
Pixie-4 only

Control Task 0x3: **Ramp offset DAC**
Control Task 0x3 is still available for backwards compatibility, but Task 0x83 or 0x85 (faster, not for Pixie-4) should be used to adjust offsets. The C library redirects a call to Pixie_Acquire_Data with runtask 0x3 to runtask 0x83. Call task 0x20 to avoid the redirect.

This task is used for calibrating the offset DAC. For each channel the offset DAC is incremented in 2048 equal-size steps. At each DAC setting the DC-offset is determined and written into the list mode buffer. At the end of the task the list mode buffer holds the following data. Its 8192 words are divided up equally amongst the four channels. Data for channel 0 occupy the lowest 2048 words, followed by data for channel 1, etc. The first entry for each channel's data block is for a DAC value of 0, the last entry is for a DAC value of 65504. In between entries the DAC value is incremented in steps of 32. On exit, the task restores the offset DAC values to the values they had on entry.

Control Task 0x4: **Untriggered Traces**

Control Task 0x4 is still available for backwards compatibility, but Tasks 0x84 (traces from 4 channels) or 0x27 (traces from a single channel) should be used instead. The C library redirects a call to Pixie_Acquire_Data with runtask 0x4 to runtask 0x84

For the Pixie-4, this task returns 8192 ADC samples for the channel specified in CHANNUM. The results are written to the 8192 words long I/O buffer.
For the Pixie-500 Express and Pixie-4 Express, this task returns 8192 ADC samples for each channel. The results are transferred to PC memory using the DMA process.

ControlTask 0x6: **Measure Baselines**

Control Task 0x2A is recommended instead for Pixie-4 Express and Pixie-500 Express.

This routine is used to collect baseline values. Currently, the DSP collects six words, B0L, B0H, B1L, B1H, time stamp, and ADC value, for each baseline. 1365 baselines are collected until the 8192-word I/O buffer is almost completely filled. The host computer can then read the I/O buffer and calculate the baseline according to the formula:

Baseline = (B1-B0* exp(-XP)) * Bnorm

with

B1 = (B1L+B1H*65536)
B0 = (B0L+B0H*65536)
XP = (SLOWLENGTH+SLOWGAP) * 2^{FR} / (CLK*TAU)
Bnorm = 2^{-9} / SLOWLENGTH for FR >=2 (Pixie-4)
= 2^{-8} / SLOWLENGTH for FR =1 (Pixie-4)
= 2^{-7} / SLOWLENGTH for FR =0 (Pixie-4)
= $2^{-(FR)}$/ SLOWLENGTH for FR <6 (P4e, P500e)
= $2^{-(FR-2)}$/ SLOWLENGTH for FR >=6 (P4e, P500e)
TAU = PREAMPTAUA+ PREAMPTAUB/65536
CLK = 75 for Pixie-4, 125 for Pixie-4 Express and Pixie-500 Express
FR = FILTERRANGE

Baseline values can then be statistically analyzed to determine the standard deviation associated with the averaged baseline value and to set the BLCUT.
BLCUT should be about 4 times the standard deviation. Baseline values can also be plotted against time stamp or ADC value to explore the

detector performance. BLCUT should be set to zero while running ControlTask 6.

Control Task 0xD (13): **Find DC offset**

This task calibrates the offset DAC. The complete function is implemented in the DSP. Normally called as a subroutine to task 0x85. On exit, the DACs are set to the value matching the request from BASELINEPERCENT.
Since the offset measurement has to take the preamplifier offset into account, this measurement must be made with the preamplifier connected to the input of the Pixie module.

Control Task 0x16 (22):Test EM write (Pixie-4 only)

This routine is used to write a test pattern from the DSP into the external memory (testing list mode data transfers). The data written is as follows:

Word (16bit)	Value	Notes
0	8002	Works as buffer length
1	MODNUM	Can be used to identify a module by writing MODNUM through CAMAC and reading the EM through USB.
2	0xAAAA	
3	0x5555	
4	0xCCCC	
5	0x3333	
6	0x1111	
7	0xEEEE	
8	0x8888	
9	0x7777	
10-8001	Repeat above words 2-9 for 999 times	
8002-8104	103, MODNUM, 25x (0x8888, 0x8888, 0x7777,0x7777), 0x8888 (testing odd sized buffer transfers)	
8105-8207	103, MODNUM, 25x (0xCCCC, 0xCCCC, 0x3333,0x3333), 0xCCCC (testing odd sized buffer transfers)	

Control Task 0x20: **Ramp offset DAC**

This task is used for calibrating the offset DAC. For each channel the offset DAC is incremented in 2048 equal-size steps. At each DAC setting the DC-offset is determined and written into the list mode buffer. At the end of the task the list mode buffer holds the following data. Its 8192 words are divided up equally amongst the four channels. Data for channel 0 occupy the lowest 2048 words, followed by data for channel 1, etc. The first entry for each channel's data block is for a DAC value of 0, the last entry is for a DAC value of 65504. In between entries the DAC value is incremented in steps of 32. On exit, the task restores the offset DAC values to the values they had on entry.

Control Task 0x27: Untriggered Traces (single channel)

This task returns 8192 ADC samples for the channel specified in CHANNUM. The results are written to the 8192 words long I/O buffer. The XWAIT variable determines the time between successive ADC samples. In the Pixie Viewer XWAIT can be adjusted through the dT variable in the Oscilloscope panel. Use this function to check if the offset adjustment was successful.

Control Task 0x28 (40): **Find BLCUT (in DSP)**

This task measures incoming baselines and determines their distribution around DC. The distribution width is reported in HOSTIO0-3 for channel 0-3. This value can be used as the BLCUT parameter to remove "bad baselines" from the running baseline average. A value of 65535 is reported for failure to determine a valid distribution.
Since the baseline measurement has to take the preamplifier noise into account, this measurement must be made with the preamplifier connected to the input of the Pixie module.

ControlTask 0x2A (42): Collect Complete Baselines
(Pixie-4 Express and Pixie-500 Express only)

This routine is used to collect baseline values. Currently, the DSP collects 13 words for each baseline:

B0L, B0H, B1L, B1H, BGL, BGH,	(raw baseline sums)
BL_DSP_L, BL_DSP_H	(baselines computed by DSP)
DC_DSP	(DC offset computed by DSP)
BL_DSP_AVGL, BL_DSP_AVGH	(baselines averages computed by DSP)
time stamp, ADC value	

630 baselines are collected until the 8192-word I/O buffer is almost completely filled. The host computer can then read the I/O buffer and calculate the baseline according to the formula:

Baseline = (B1-B0* exp(-XP)) * Bnorm

with

B1 = (B1L+B1H*65536)
B0 = (B0L+B0H*65536)
XP = (SLOWLENGTH+SLOWGAP) * 2^{FR} / (CLK*TAU)
Bnorm = $2^{-(FR)}$/ SLOWLENGTH for FR <6 (P4e, P500e)
= $2^{-(FR-2)}$/ SLOWLENGTH for FR >=6 (P4e, P500e)
TAU = PREAMPTAUA+ PREAMPTAUB/65536
CLK = 125 for Pixie-4 Express and Pixie-500 Express
FR = FILTERRANGE

Baseline values can then be statistically analyzed to determine the standard deviation associated with the averaged baseline value and to set the BLCUT. BLCUT should be about 4 times the standard deviation. Baseline values can also be plotted against time stamp or ADC value to explore the detector performance.

Control Task 0x85: Adjust offset for all modules (DSP)
(Pixie-4 Express and Pixie-500 Express only)

This routine is used to adjust DC offsets to the target value BASELINEPERCENT for all channels and modules. The C library will call a task in the DSP for each channel and module that finds the target DC offset. This is much faster for Pixie-4 Express and Pixie-500 Express than task 0x83

5 Appendix A — User supplied DSP code Pixie-4

unchanged

6 Appendix B — User supplied DSP code Pixie-4 Express and Pixie-500 Express

New section, please see full manual

7 Appendix C — User supplied Igor code

unchanged