Table of Contents

README

最新版本号 **Version:2018.05.12**

程序下载请访问 https://github.com/wuhongyi/PKUXIADAQ

网页版说明书请访问 http://wuhongyi.cn/PKUXIADAQ/

markdown版本说明书请访问
README/

本地网页版说明书请访问
docs/

pdf版本说明书请访问
README.pdf

- 对本获取程序有任何的意见及建议(功能添加及改进)，欢迎给吴鸿毅(wuhongyi@qq.com)发邮件。
- 我们将会尽快完善软件的中英文使用说明书，当前基本都是以通过操作演示讲解软件的使用为主。

本说明书仅适用于 **XIA LLC Pixie-16**

本程序由北京大学实验核物理组开发。

技术指导:

- 李智焕
- 谭辉(XIA LLC)

软件主要开发者:

- 吴鸿毅(wuhongyi@qq.com)
- 李晶(lijinger02@126.com)

说明书主要撰写者:

- 吴鸿毅
- 王翔
- 罗迪雯

本程序的开发得到以下单位的支持：

- XIA LLC
- 中国科学院近代物理研究所(IMP)
- 中国原子能科学研究院(CIAE)
- 香港大学(HKU)
- 山东大学(威海)(SDU)
- ...

本程序适用于 XIA Pixie16 系列采集卡，支持100/250/500MHz采集卡(具体支持型号可查看图形软件中的File->About)，最大支持8个机箱同步运行，即1600+路信号同时采集。本程序包要求使用 **CERN ROOT6** 版本。要求采用 **1920x1080** 及以上分辨率显示屏。

本程序的设计兼容 100/250/500 MHz 采集卡的混合使用，只需在 cfgPixie16.txt 添加各类采样率采集卡的固件位置即可，程序在线能够自动识别采集卡类型并加载相应的固件。当前我们只有100/250MHz 14bit的采集卡，因此默认可运行该类型的采集卡，如要支持其它类型，请联系 XIA LLC 获取对应固件或者联系吴鸿毅(wuhongyi@qq.com)。

用户使用程序包中包含以下文件/文件夹:

- Decode(将原始二进制数据转为ROOT)
- docs(使用说明书，网页版)
- firmware(固件)
  - firmware/firmware.md(历史各版本固件说明)
- GUI(图形软件)
- MakeEvent(事件重构程序，可选)
- NOGUI(非图形软件。新版本升级中，暂时不可用)
- OnlineStattics(在线监视程序)
- parset(参数设置文件)
- PlxSdk.tar.gz(Plx9054驱动)
- README(说明书md版)
- README.md(github首页介绍)
- README.pdf(pdf版本说明书)
- software(pixie16驱动API，非官方标准，已经被吴鸿毅修改)
- TestTool(开发者测试工具，用户不需要！！！)

程序安装

本程序安装要求

- CERN ROOT 6
  - GCC >= 4.8
- FFTW3

本程序测试过的系统包括Scientific Linux 7.2/7.3/7.4

软件安装步骤

- 删除个人目录下的老版本PKUXIADAQ文件夹
- 将本程序包解压缩到个人目录中($HOME)
- 设置环境变量
- 编译Plx9054驱动
- 编译pixie16驱动API(该API被吴鸿毅修改过，非官方标准驱动)
- 编译图形化获取软件
- 编译在线监视程序
- 编译数据转换程序
- 编译事件重构程序(可选)

```
##设置环境变量

#在 .bashrc 文件中添加
export PLX_SDK_DIR=$HOME/PKUXIADAQ/PlxSdk

# 将 PKUXIADAQ.tar.gz(或者PKUXIADAQ-master.tar.gz) 放置到 /home 下的个人目录下，即 ~/ 位置
tar -zxvf PKUXIADAQ.tar.gz  #解压缩
或者
tar -zxvf PKUXIADAQ-master.tar.gz
mv PKUXIADAQ-master  PKUXIADAQ

#得到 PKUXIADAQ 目录
```

```
##编译Plx9054驱动

#打开新终端
cd ~
cd PKUXIADAQ/
rm -rf PlxSdk #删除可能存在的未删除驱动，如果没有该目录则不用执行该行命令
tar -zxvf PlxSdk.tar.gz
cd PlxSdk/PlxApi/
make clean
```

```
make
#成功后你将会看到 Library "Library/PlxApi.a" built successfully

cd ../Samples/ApiTest/
make clean
make
#成功后你将会看到 Application "App/ApiTest" built successfully

cd ../../Driver/
./builddriver 9054

#成功后你将会看到 Driver "Plx9054/Plx9054.ko" built sucessfully
```

```
##编译pixie16

cd ~
cd PKUXIADAQ/software/
make clean
make
#只要没报错，并且该文件夹内生成libPixie16App.a libPixie16Sys.a
```

```
##编译图形化获取软件

#修改设置参数
cd ~
cd PKUXIADAQ/parset/
#修改cfgPixie16.txt文件。
#其中CrateID 后面的数值表示机箱编号，时值允许0-15。如果单机箱则随意设置(一般就采用默认的0)，如果多个机箱同步运行务必让每个机箱的该编号设置未不同的数值。
#SettingPars 后面为参数设置文件，写入要采用的参数配置文件即可。
#ModuleSlot 后面第一个数值表示插件个数，如果有3个插件则为3。之后的数字未为每个插件在机箱的插槽位置（插槽位置从2开始计数），有三个插件则之后分别为2 3 4。
#参数 ModuleSampingRate与ModuleBits 只对离线模式生效，当主界面采用Offline模式初始化时则读取该参数。

#修改Run.config文件，该文件中第一行为原始数据存放路径，第二行为文件名。
#修改RunNumber文件，该文件中的数值为运行的run number。

cd ~
cd PKUXIADAQ/GUI/
make clean
make
```

```
##编译在线监视程序

cd ~
cd PKUXIADAQ/OnlineStattics/
```

```
#修改 PixieOnline.config 文件中的参数
#第一行为获取数据文件存放路径
#第二行为获取文件名

make clean
make
```

```
## 编译数据转换程序

cd ~
cd PKUXIADAQ/Decode/
#修改 UserDefine.hh，按照程序中的说明修改即可

make clean
make
```

```
## 编译事件重构程序

cd ~
cd PKUXIADAQ/MakeEvent/
#修改 UserDefine.hh，按照程序中的说明修改即可

make clean
make
```

程序使用说明

- 开机机箱后重启电脑(电脑必须晚于机箱开启)
- 开启机箱后ROOT权限下加载Plx9054驱动
- 正常获取

```
## ROOT权限下加载Plx9054驱动
cd ~
cd PKUXIADAQ/PlxSdk/Bin/
su #输入ROOT密码
./Plx_load 9054
#将会看到加载成功的提示
exit  #退出ROOT权限
```

```
##启动图形界面程序

cd ~
cd ~/PKUXIADAQ/GUI
```

```
./pku
#将会弹出图形化界面
#可选择 Online/Offline Mode 然后按 Boot 初始化
#等待初始化成功后，可修改输出数据文件路径，文件名，run number。按 Complete 按钮确认。
#此时 LSRunStart 按钮变为可操作。即可开始按Start，之后第二次按即为Stop。
#Online Statistics选项选择表示发送在线统计
#Update Energy Monitor每选择一次则从插件内部读取一次能谱信息并发送给在线程序（频繁选择会影响获取）
```

```
##启动在线监视程序

cd ~
cd PKUXIADAQ/OnlineStattics/
./online
#将会弹出图形化界面
#查看上面的原始数据文件夹路径、文件名是否正确。按 Complete 确认。
#按 RunStart开始启动监视，每3秒更新一次每路的输入率、输出率。(开启机箱后第一次启用该程序需要在获取开启之后)
#监视界面右下角有对写入硬盘使用量的监视。

#EnergyMonitor页面用来查看能谱。由于插件内部寄存器大小限制，该能谱与实际能谱道址范围存在差别。
```

```
##执行数据转换程序

cd ~
cd PKUXIADAQ/Decode/

#在上一轮获取结束之后，我们便可将上一轮数据转为ROOT文件
./decode xxx
#xxx 表示 Run Number
```

Guide

- 这里需要介绍跳线接法
- 原始数据定义

Decode

GUI

配置好 **parset** 内参数文件 进入 GUI 目录，执行以下命令即可弹出主控制界面

```
./pku
```

主控制界面


PKU XIA Pixie-16 DAQ
File
UV_Setup
Expert
Monitor
Offline
List Mode Run
Online Mode
Boot
Setup
File Path:
/home/wuhongyi/data/
File Name:
data
Run Num:
1061
Complete
Control
Online Statistics
LSRunStart
Update Energy Monitor
Information
Version: 2018.05.12
System not booted
Last run number: 1060

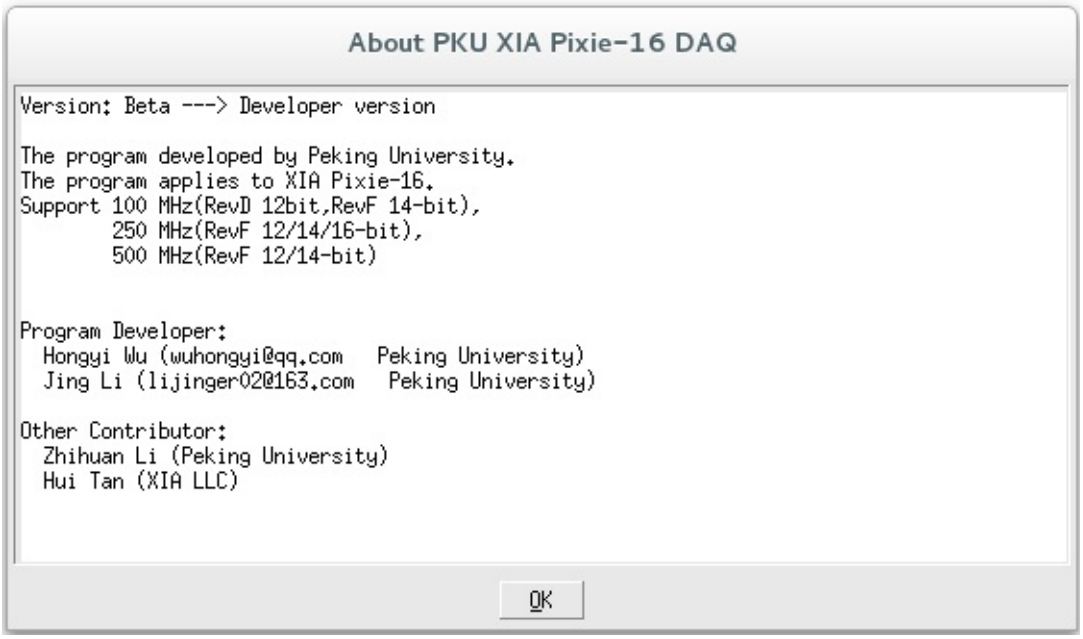
主界面最上方是File、UV_Setup、Expert、Monitor、Offiline五个下拉栏。里面的子菜单如下所示：

- File
  - Exit
  - About
- UV_Setup
  - Base Setup
  - Trigger Filter
  - Energy
  - CFD
  - QDC
  - Decimation
  - Copy Pars
  - Save2File
- Expert
  - Module Variables
  - CSRA
  - Logic Set
- Monitor
  - Hist & XDT
  - Trace & Baseline
- Offiline
  - Adjust Par
  - Simulation(暂未实现)

File 下拉栏

本下拉栏内容没有实际用途。

About

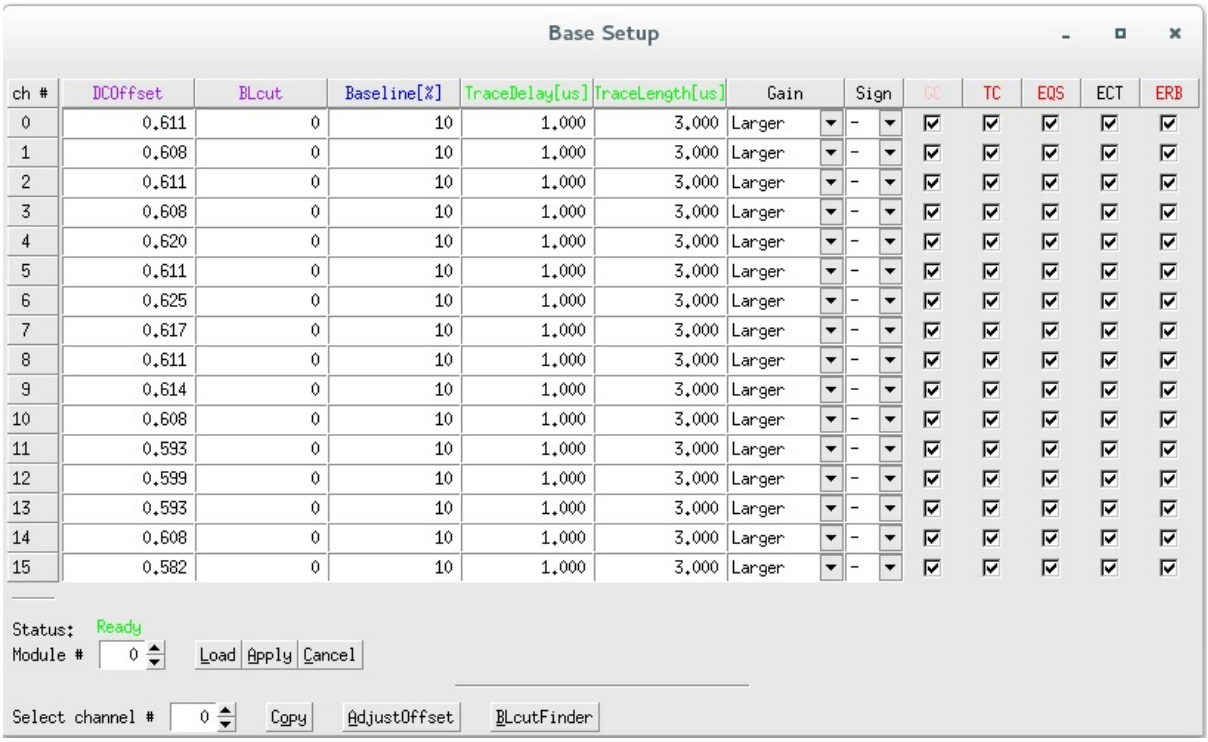


软件开发者介绍。之后将会添加主程序基本操作说明。

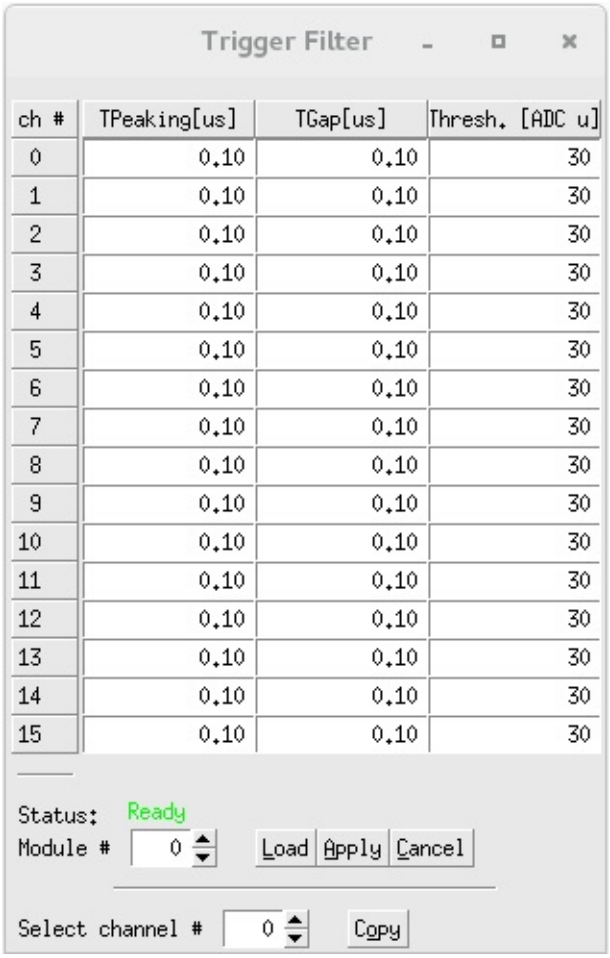
UV_Setup 下拉栏

本下拉栏中调节内容为基础内容，任何使用Pixie16获取系统的人员都应该熟悉并掌握其调节技巧。

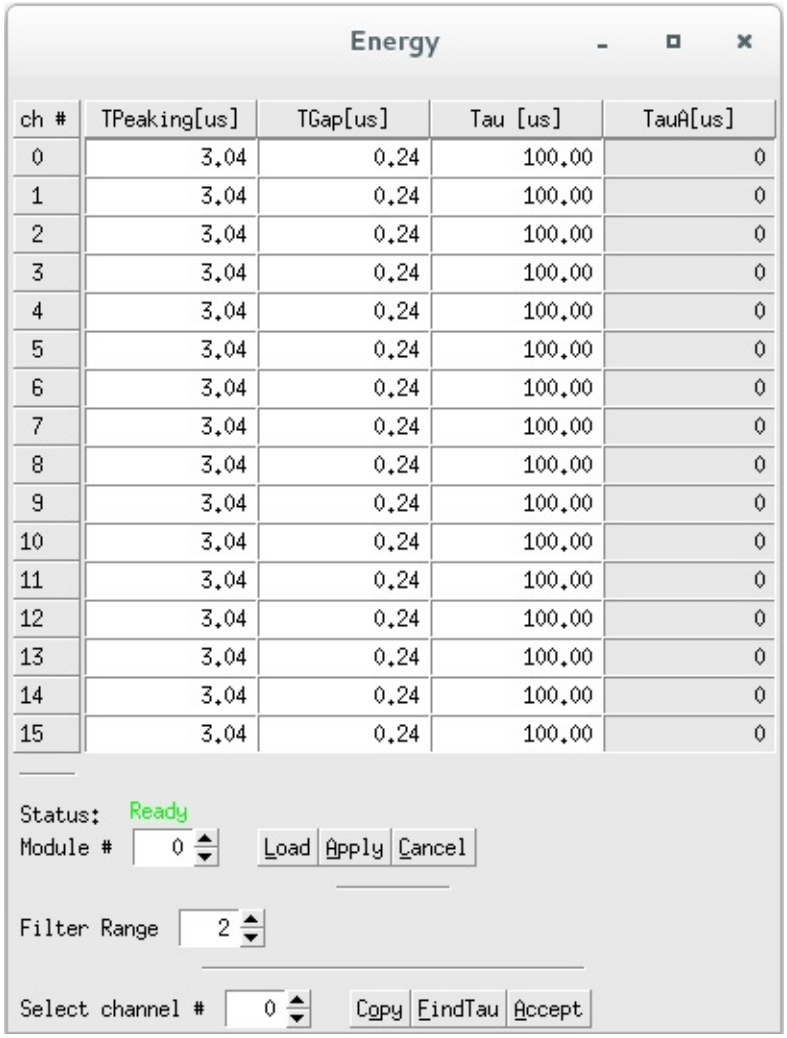
Base Setup



Trigger Filter



Energy


Energy
ch # TPeaking[us] TGap[us] Tau [us] TauA[us]
0 3.04 0.24 100.00 0
1 3.04 0.24 100.00 0
2 3.04 0.24 100.00 0
3 3.04 0.24 100.00 0
4 3.04 0.24 100.00 0
5 3.04 0.24 100.00 0
6 3.04 0.24 100.00 0
7 3.04 0.24 100.00 0
8 3.04 0.24 100.00 0
9 3.04 0.24 100.00 0
10 3.04 0.24 100.00 0
11 3.04 0.24 100.00 0
12 3.04 0.24 100.00 0
13 3.04 0.24 100.00 0
14 3.04 0.24 100.00 0
15 3.04 0.24 100.00 0
Status: Ready
Module # 0
Load
Apply
Cancel
Filter Range 2
Select channel # 0
Copy
FindTau
Accept

CFD

Cfd Par.

ch #	CFD Delay[us]	CFD Frac[0-7]	CFD Thre
0	0.10	2.00	120.00
1	0.10	2.00	120.00
2	0.10	2.00	120.00
3	0.10	2.00	120.00
4	0.10	2.00	120.00
5	0.10	2.00	120.00
6	0.10	2.00	120.00
7	0.10	2.00	120.00
8	0.10	2.00	120.00
9	0.10	2.00	120.00
10	0.10	2.00	120.00
11	0.10	2.00	120.00
12	0.10	2.00	120.00
13	0.10	2.00	120.00
14	0.10	2.00	120.00
15	0.10	2.00	120.00

Status: Ready
Module # 0 Load Apply Cancel

Select channel # 0 Copy

QDC

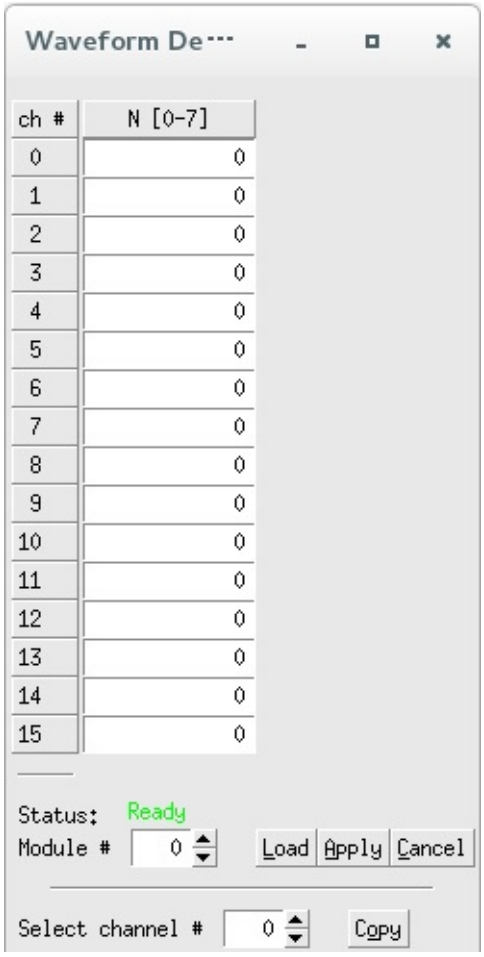
Qdc Par.

ch #	QDC len0[us]	QDC len1[us]	QDC len2[us]	QDC len3[us]	QDC len4[us]	QDC len5[us]	QDC len6[us]	QDC len7[us]
0	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
1	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
2	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
3	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
4	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
5	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
6	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
7	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
8	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
9	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
10	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
11	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
12	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
13	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
14	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13
15	0.30	0.63	0.88	1.13	1.38	1.63	1.88	2.13

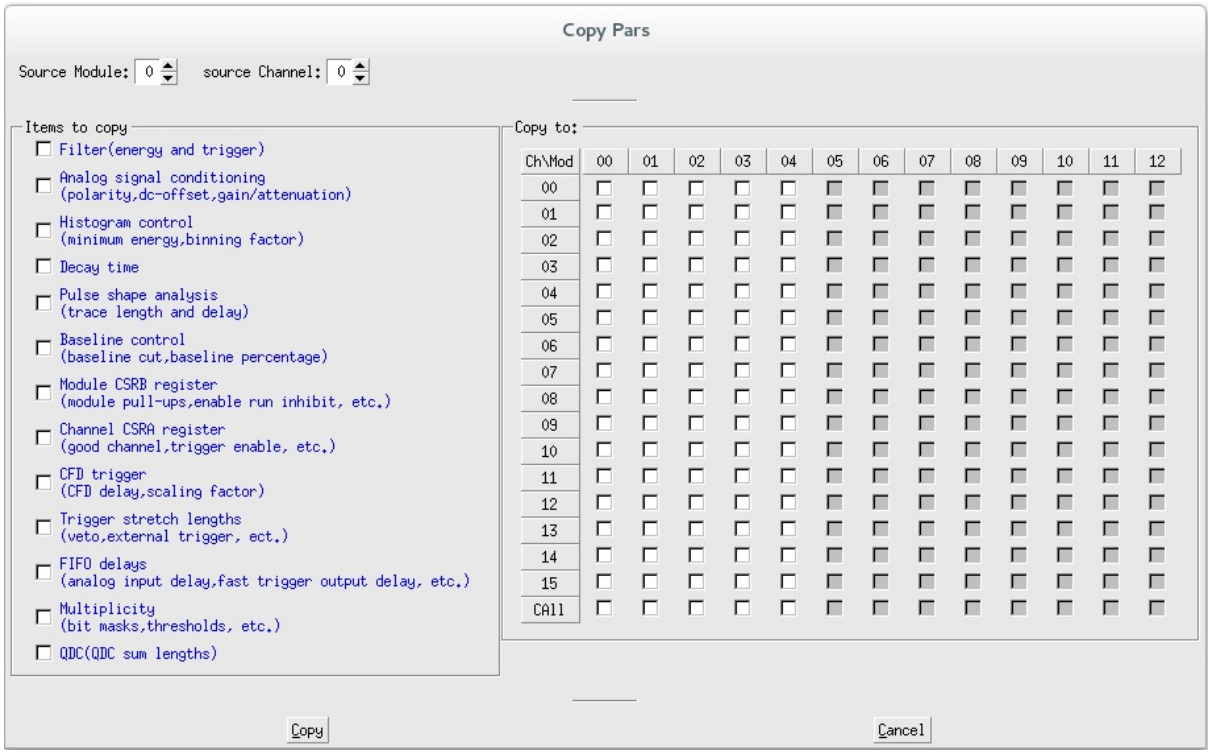
Status: Ready
Module # 0 Load Apply Cancel

Select channel # 0 Copy

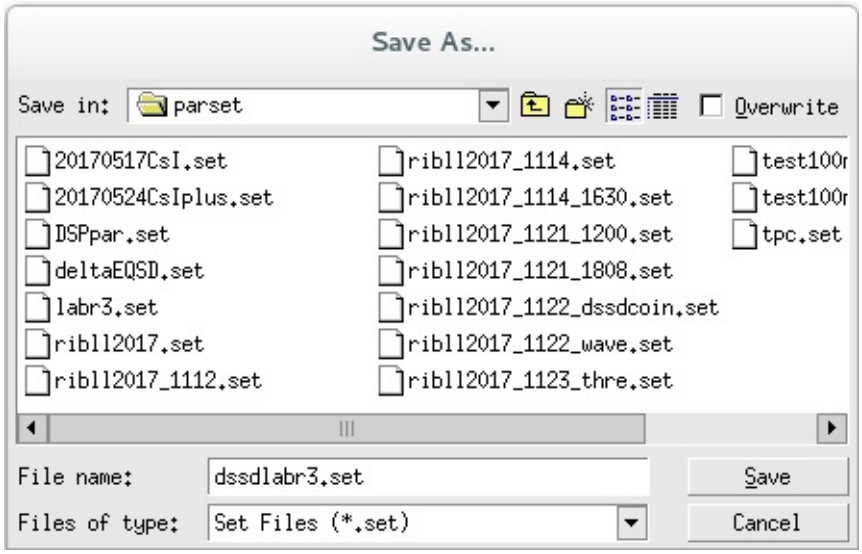
Decimation



Copy Pars



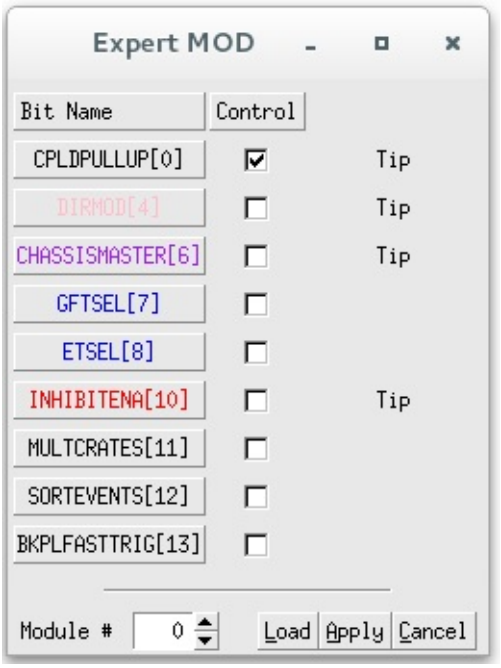
Save2File



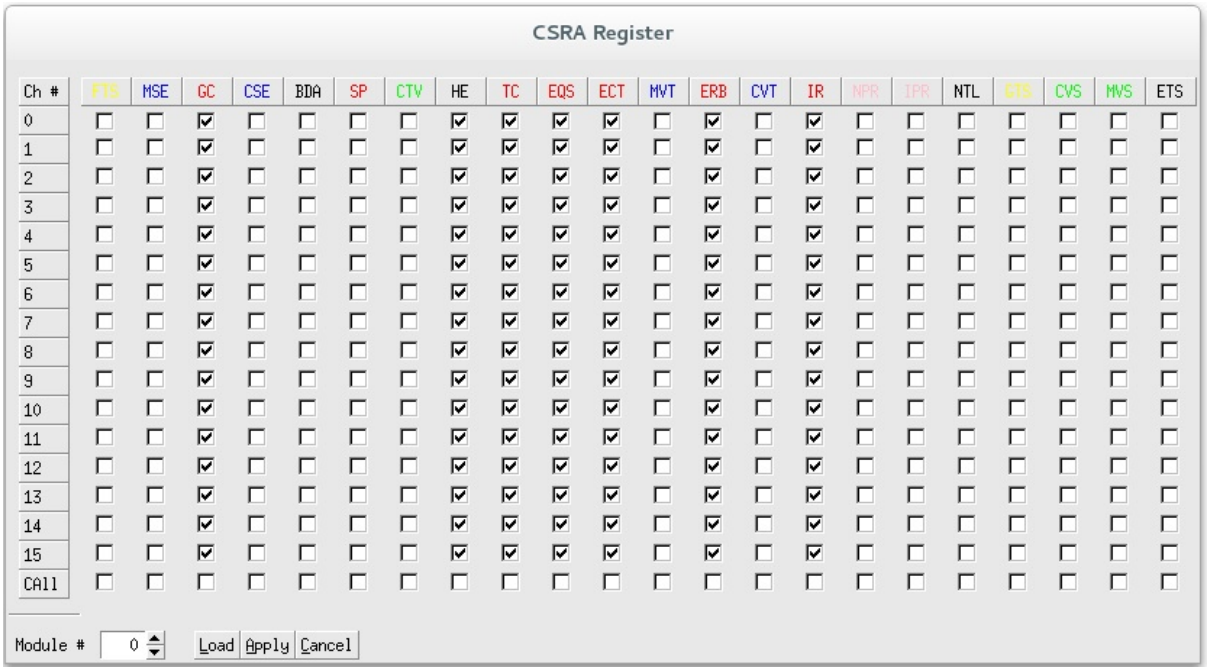
Expert 下拉栏

本下拉栏中调节内容为高阶内容，需要对获取逻辑有一定基础的人学习掌握。

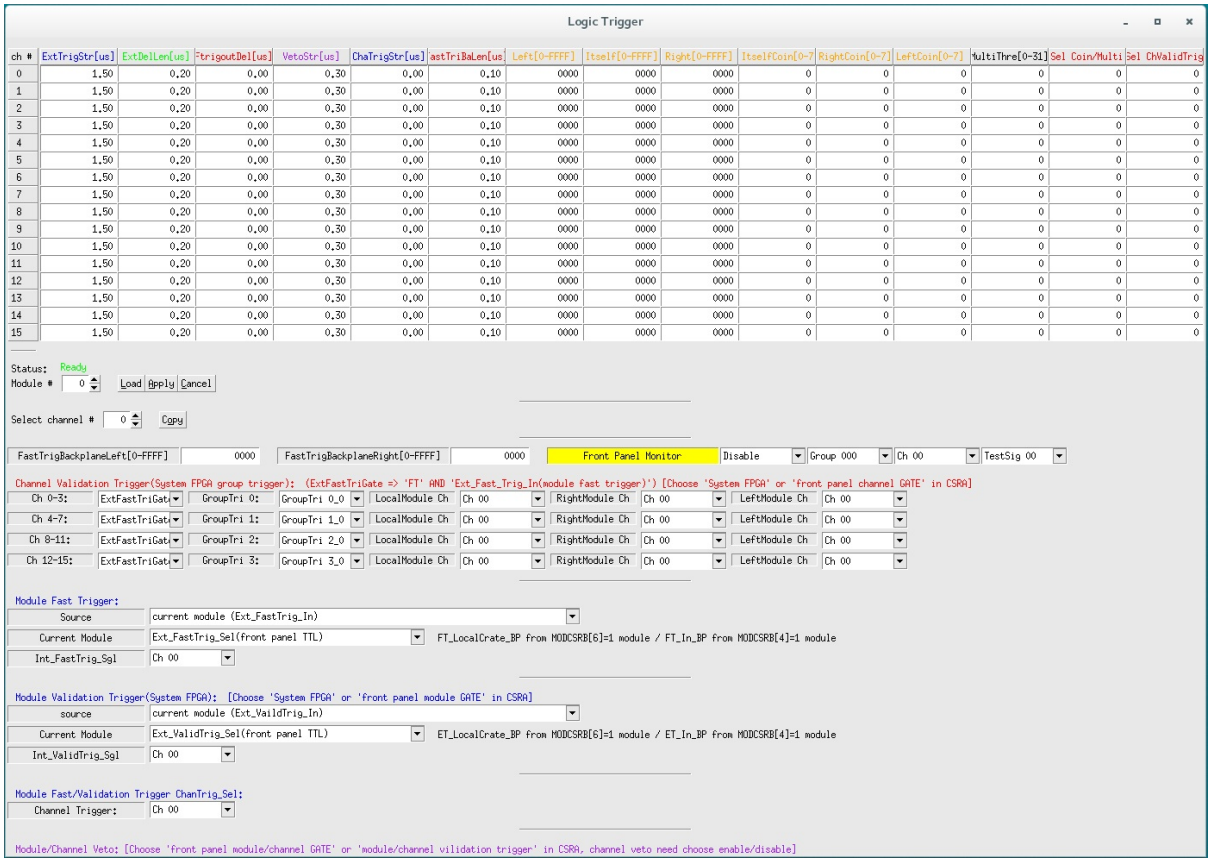
Module Variables



CSRA



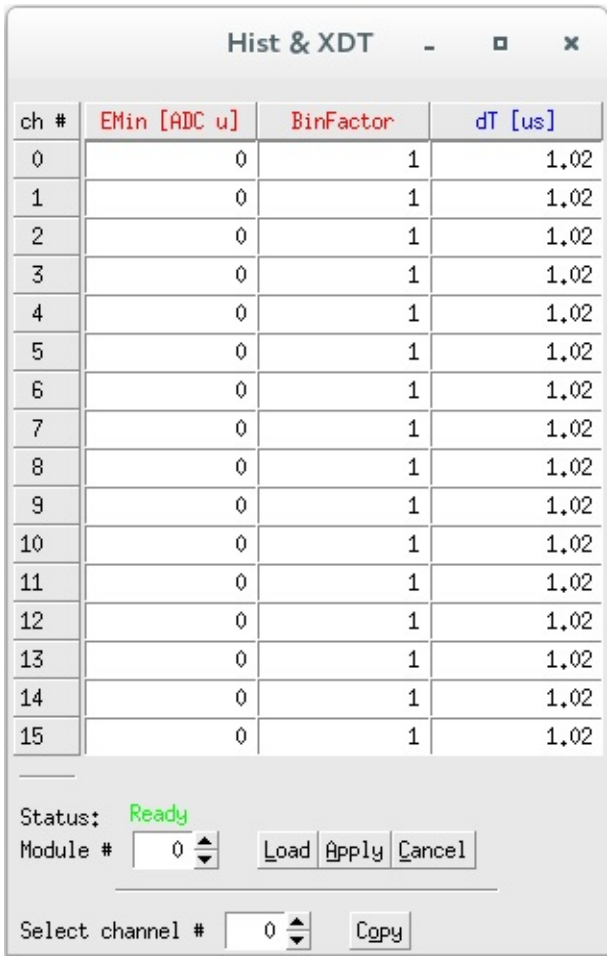
Logic Set



Monitor 下拉栏

本下拉栏中调节内容为监视波形噪声水平、基线分布等。

Hist & XDT



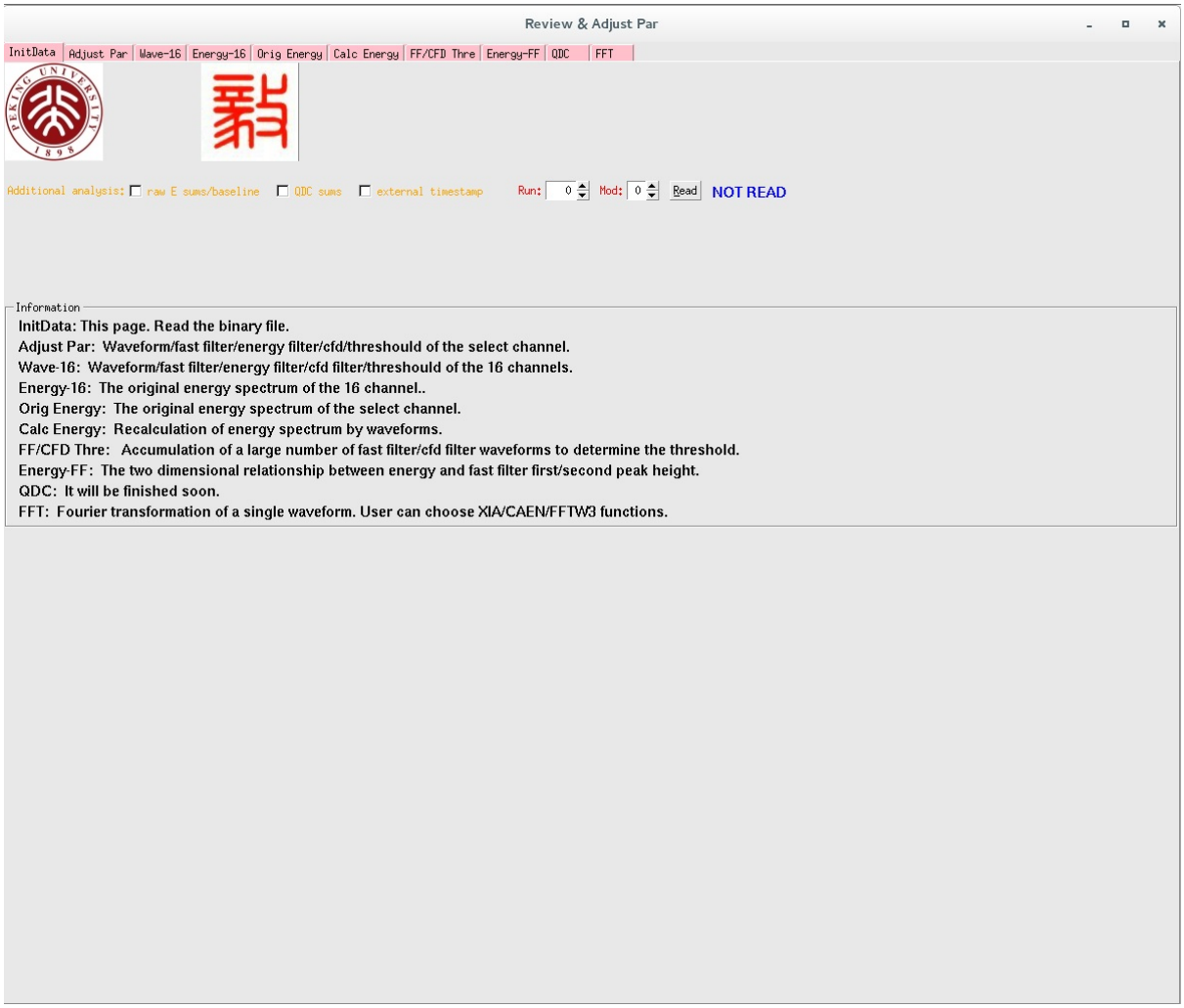
Trace & Baseline

待补充

Offiline 下拉栏

本下拉栏中为离线参数优化调节。

Adjust Par



Simulation

通过模型产生不同类型探测器的波形，辅助使用者学习参数优化调节的。

OnlineStattics

MakeEvent

Front Panel

本页内容待更新。。。

LVDS I/O port(J101)

普通网口

当前标准固件中还没定义

J155（letter"A"）

输出信号

3.3V I/O port，输出阻抗是 50 欧

A2、D1 地 A1、B1、B2、C1、C2、D2 输出信号

J151-J154

输入信号

differential LVDS signals

- 16 channel gate input
- 1 module gate input
- 1 not use current

J151-J155

输入信号

single-ended TTL external input signals

FI0、FI2、FI3、FI4、FI6、FI7

这是什么输入信号？external trigger？clock signals？

从前面板输出监视来看，fast trigger stretch length(FastTrigBackLen) 实际值为16路中最大的值。

前面板监视信号中，左边的矩形波是fast trigger，其宽度表示时间。 右边的信号表示Extern delay(ExternDelayLen) 表示为了等待trigger而延迟采集的时间

group 000

FTRIG_DELAY 采集延迟时间 只要fast filter 过阈值就会产生 FTRIG_VAL 基本同上，有效采集时候才有信号 GLBETRIG_CE stretched external global validation trigger CHANETRIG_CE stretched channel validation trigger ，开启CSRA bit13 。采集延迟必须在这个时间窗内才能采集到 看到的信号实际起始位置在 100 ns，意味着System FPGA 处理时间需要 100 ns ？ 信号宽度由 ChanTrigStretch 控制。

选择 CSRA bit0 bit18 FTRIG_DELAY 总会多 100 ns

开发者指南

本章节介绍 Pixie16 开发中使用的Pixie-16 API 函数及获取程序的基本原理。

XIA API

```
PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16InitSystem (
    unsigned short NumModules,    // total number of Pixie16 modules in the system
    unsigned short *PXISlotMap,   // an array containing the PXI slot number for ea
ch pixie16 module
    unsigned short OfflineMode ); // specify if the system is in offline mode

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ExitSystem (
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadModuleInfo (
    unsigned short ModNum,        // module number
    unsigned short *ModRev,       // returned module revision
    unsigned int   *ModSerNum,    // returned module serial number
    unsigned short *ModADCBits,   // returned module ADC bits
    unsigned short *ModADCMSPS ); // returned module ADC sampling rate

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16BootModule (
    char *ComFPGAConfigFile,      // name of communications FPGA configuration file
    char *SPFPGAConfigFile,       // name of signal processing FPGA configuration f
ile
    char *TrigFPGAConfigFile,     // name of trigger FPGA configuration file
    char *DSPCodeFile,            // name of executable code file for digital signa
l processor (DSP)
    char *DSPParFile,             // name of DSP parameter file
    char *DSPVarFile,             // name of DSP variable names file
    unsigned short ModNum,        // pixie module number
    unsigned short BootPattern ); // boot pattern bit mask

// Acquire ADC traces in single or multiple modules
// Use this function to acquire ADC traces from Pixie-16 modules. Specify the modul
e using ModNum. If ModNum is set to be less than the total number of modules in the
 system, only the module specified by ModNum will have its ADC traces acquired. But
 if ModNum is equal to the total number of modules in the system, then all modules
in the system will have their ADC traces acquired.
// After the successful return of this function, the DSP's internal memory will be
filled with ADC trace data. A user's application software should then call another
function Pixie16ReadSglChanADCTrace to read the ADC trace data out to the host comp
uter, channel by channel.
PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16AcquireADCTrace (
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadSglChanADCTrace (
    unsigned short *Trace_Buffer, // trace data
    unsigned int Trace_Length,    // trace length
    unsigned short ModNum,        // module number
    unsigned short ChanNum );     // channel number
```

```
PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16IMbufferIO (
    unsigned int *Buffer,         // buffer data
    unsigned int NumWords,        // number of buffer words to read or write
    unsigned int Address,         // buffer address
    unsigned short Direction,     // I/O direction
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16EMbufferIO (
    unsigned int *Buffer,         // buffer data
    unsigned int NumWords,        // number of buffer words to read or write
    unsigned int Address,         // buffer address
    unsigned short Direction,     // I/O direction
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16StartListModeRun (
    unsigned short ModNum,        // module number
    unsigned short RunType,       // run type
    unsigned short mode );        // run mode

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16StartHistogramRun (
    unsigned short ModNum,        // module number
    unsigned short mode );        // run mode

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16CheckRunStatus (
    unsigned short ModNum );      // Pixie module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16EndRun (
    unsigned short ModNum );      // Pixie module number

PIXIE16APP_EXPORT double PIXIE16APP_API Pixie16ComputeInputCountRate (
    unsigned int   *Statistics,
    unsigned short ModNum,
    unsigned short ChanNum );

PIXIE16APP_EXPORT double PIXIE16APP_API Pixie16ComputeOutputCountRate (
    unsigned int   *Statistics,
    unsigned short ModNum,
    unsigned short ChanNum );

PIXIE16APP_EXPORT double PIXIE16APP_API Pixie16ComputeLiveTime (
    unsigned int   *Statistics,
    unsigned short ModNum,
    unsigned short ChanNum );

PIXIE16APP_EXPORT double PIXIE16APP_API Pixie16ComputeProcessedEvents (
    unsigned int   *Statistics,
    unsigned short ModNum );

PIXIE16APP_EXPORT double PIXIE16APP_API Pixie16ComputeRealTime (
    unsigned int   *Statistics,
```

```
    unsigned short ModNum );

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16complexFFT (
    double *data,
    unsigned int length );

PIXIE16APP_EXPORT unsigned short PIXIE16APP_API APP16_TstBit (
    unsigned short bit,
    unsigned short value );

PIXIE16APP_EXPORT unsigned short PIXIE16APP_API APP16_SetBit (
    unsigned short bit,
    unsigned short value );

PIXIE16APP_EXPORT unsigned short PIXIE16APP_API APP16_ClrBit (
    unsigned short bit,
    unsigned short value );

PIXIE16APP_EXPORT unsigned int PIXIE16APP_API APP32_SetBit (
    unsigned short bit,
    unsigned int   value );

PIXIE16APP_EXPORT unsigned int PIXIE16APP_API APP32_ClrBit (
    unsigned short bit,
    unsigned int   value );

PIXIE16APP_EXPORT unsigned int PIXIE16APP_API APP32_TstBit (
    unsigned short bit,
    unsigned int   value );

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16SetDACs (
    unsigned short ModNum );

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ProgramFippi (
    unsigned short ModNum );

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16AdjustOffsets (
    unsigned short ModNum );

// Acquire baselines from a module
PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16AcquireBaselines (
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadSglChanBaselines (
    double *Baselines,            // returned baselines values
    double *TimeStamps,           // time stamp for each baseline value
    unsigned short NumBases,      // number of baseline values to read
    unsigned short ModNum,        // module number
    unsigned short ChanNum );     // channel number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16RampOffsetDACs (
```

```
    double *DCValues,             // returned DC offset values
    unsigned short NumDCVals,     // number of DC values to read
    unsigned short ModNum );

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ControlTaskRun (
    unsigned short ModNum,        // Pixie module number
    unsigned short ControlTask,   // Control task number
    unsigned int   Max_Poll );    // Timeout control in unit of ms for control task
 run

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16BLcutFinder (
    unsigned short ModNum,        // Pixie module number
    unsigned short ChanNum,       // Pixie channel number
    unsigned int   *BLcut );      // BLcut return value

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16TauFinder (
    unsigned short ModNum,        // Pixie module number
    double         *Tau );        // 16 returned Tau values, in 衽

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16WriteSglModPar (
    char *ModParName,             // the name of the module parameter
    unsigned int   ModParData,    // the module parameter value to be written to th
e module
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadSglModPar (
    char *ModParName,             // the name of the module parameter
    unsigned int   *ModParData,   // the module parameter value to be read from the
 module
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16WriteSglChanPar (
    char *ChanParName,            // the name of the channel parameter
    double ChanParData,           // the channel parameter value to be written to t
he module
    unsigned short ModNum,        // module number
    unsigned short ChanNum );     // channel number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadSglChanPar (
    char *ChanParName,            // the name of the channel parameter
    double *ChanParData,          // the channel parameter value to be read from th
e module
    unsigned short ModNum,        // module number
    unsigned short ChanNum );     // channel number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadHistogramFromModule (
    unsigned int   *Histogram,    // histogram data
    unsigned int   NumWords,      // number of words to be read out
    unsigned short ModNum,        // module number
    unsigned short ChanNum);      // channel number
```

```
PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadStatisticsFromModule (
    unsigned int   *Statistics,   // run statistics data
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16SaveHistogramToFile (
    char *FileName,               // histogram data file name
    unsigned short ModNum);       // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16GetModuleEvents (
    char *FileName,               // the list mode data file name (with complete pa
th)
    unsigned int *ModuleEvents ); // receives number of events for each module

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16GetEventsInfo (
    char *FileName,                  // the list mode data file name (with complet
e path)
    unsigned int   *EventInformation, // to hold event information
    unsigned short ModuleNumber);     // the module whose events are to be retrieved


PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadListModeTrace (
    char *FileName,               // list mode data file name
    unsigned short *Trace_Data,   // list mode trace data (16-bit words)
    unsigned short NumWords,      // number of 16-bit words to be read out
    unsigned int   FileLocation); // the location of the trace in the file

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadHistogramFromFile (
    char  *FileName,              // the histogram data file name (with complete pa
th)
    unsigned int *Histogram,      // histogram data
    unsigned int NumWords,        // number of words to be read out
    unsigned short ModNum,        // module number
    unsigned short ChanNum);      // channel number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16SaveDSPParametersToFile (
    char  *FileName );            // the DSP parameters file name (with complete pa
th)

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16LoadDSPParametersFromFile (
    char  *FileName );            // the DSP parameters file name (with complete pa
th)

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16CopyDSPParameters (
    unsigned short BitMask,             // copy items bit mask
    unsigned short SourceModule,        // source module
    unsigned short SourceChannel,       // source channel
    unsigned short *DestinationMask );  // the destination module and channel bit m
ask

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadMSGFile (
    char *ReturnMsgStr );
```

```
PIXIE16APP_EXPORT unsigned int PIXIE16APP_API Decimal2IEEEFloating(double DecimalNu
mber);

PIXIE16APP_EXPORT double PIXIE16APP_API IEEEFloating2Decimal(unsigned int IEEEFloat
ingNumber);

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16SaveExternalFIFODataToFile (
    char *FileName,               // list mode data file name
    unsigned int   *nFIFOWords,   // number of words read from external FIFO
    unsigned short ModNum,        // module number
    unsigned short EndOfRunRead); // indicator whether this is the end of run read

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16RegisterIO (
    unsigned short ModNum,        // the Pixie module to communicate to
    unsigned int   address,       // register address
    unsigned short direction,     // either MOD_READ or MOD_WRITE
    unsigned int   *value );      // holds or receives the data

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadCSR (
    unsigned short ModNum,
    unsigned int   *CSR );

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16WriteCSR (
    unsigned short ModNum,
    unsigned int   CSR );

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16CheckExternalFIFOStatus (
    unsigned int   *nFIFOWords,
    unsigned short ModNum );

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ReadDataFromExternalFIFO (
    unsigned int   *ExtFIFO_Data, // To receive the external FIFO data
    unsigned int   nFIFOWords,    // number of words to read from external FIFO
    unsigned short ModNum );      // module number

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ComputeFastFiltersOffline (
    char           *FileName,          // the list mode data file name (with comple
te path)
    unsigned short ModuleNumber,       // the module whose events are to be analyzed

    unsigned short ChannelNumber,      // the channel whose events are to be analyz
ed
    unsigned int   FileLocation,       // the location of the trace in the file
    unsigned short RcdTraceLength,     // recorded trace length
    unsigned short *RcdTrace,          // recorded trace
    double         *fastfilter,        // fast filter response
    double         *cfd );             // cfd response

PIXIE16APP_EXPORT int PIXIE16APP_API Pixie16ComputeSlowFiltersOffline (
    char           *FileName,          // the list mode data file name (with comple
```

```
te path)
    unsigned short ModuleNumber,       // the module whose events are to be analyzed

    unsigned short ChannelNumber,      // the channel whose events are to be analyz
ed
    unsigned int   FileLocation,       // the location of the trace in the file
    unsigned short RcdTraceLength,     // recorded trace length
    unsigned short *RcdTrace,          // recorded trace
    double         *slowfilter );      // slow filter response

// Add by Hongyi Wu
PIXIE16APP_EXPORT int PIXIE16APP_API HongyiWuPixie16ComputeSlowFiltersOffline (
    char           *FileName,          // the list mode data file name (with comple
te path)
    unsigned short ModuleNumber,       // the module whose events are to be analyzed

    unsigned short ChannelNumber,      // the channel whose events are to be analyz
ed
    unsigned int   FileLocation,       // the location of the trace in the file
    unsigned short RcdTraceLength,     // recorded trace length
    unsigned short *RcdTrace,          // recorded trace
    double         *slowfilter,        // slow filter response
    unsigned int   bl,
    double         sl,
    double         sg,
    double         tau,
    int            sfr,
    int            pointtobl );
```