

**Physics Division -- *Fast* Electronics Group**

**User Software for EFADC Compton Polarimeter**

**May 2021**

**Hai Dong**

#  Introduction

 This document describes the User Software for the EFADC Compton Polarimeter. The software is written in C++ and requires Visual Studio Community to be installed on the computer. Visual Studio Community is free and the instruction to install is described in Visual Studio Community Installation Section. The User Software configures the EFADC250, establishes Ethernet Connection between the host computer and the EFADC, and interacts with the user via command line. Figure 1 shows the user interface

#  Installing Visual Studio Community

* + Go to <https://my.visualstudio.com/Benefits?wt.mc_id=o~msft~vscom~nav&campaign=o~msft~vscom~nav>
	+ Click on Download button on Visual Studio Community screen
	+ Run vs\_community\_\_1529215591.1588620714.exe as administrator
	+ Click on .Net desktop development. Make sure the options are as shown below.
	+ Click on Desktop Development with C++. Make sure the options are as shown below

Net desktop development options:



Desktop Development with C++ options:



#  Setting Up Software to Run alone

* + Make a run directory for your project
	+ Copy “RunInit.txt” to this directory
		1. This file specifies the run parameters. It is read and send to EFADC as part of initialization. User can modify the run parameter in this file and rerun the program to apply the changes.
	+ Copy “PlayBack.txt” to this directory
		1. This file specifies the data for the playback memory. It is read and send to EFADC as part of initialization. User can modify the data and issue ‘pb‘ command at command prompt >>/
	+ Copy CompPol.exe to this directory
	+ Run CompPol.exe as administrator
		1. The software establish connections with host computer. UDP is used to send command and read configuration and status registers. TCPIP is used to receive process data.
		2. Configure all 16 ADC IC and DAC IC.
		3. Read “RunInit.txt” and send to EFADC
		4. Read “PlayBack.txt” and send to EFADC.
		5. The “number of Ethernet byte to collect” is default to 100. User can change this with ‘pk’ command.
		6. Display Run Parameters
		7. Display User Input Menu
	+ User can change the Run Parameters with command.
	+ Hit ‘c’ to process and collect data
		1. The program issues “Collect On” command to EFADC
		2. The program collect at least “number of Ethernet byte to collect” and write to CollectData.TXT.
		3. The program issues “Collect Off” command to EFADC
		4. EFADC stop processing data and freeze time stamp
		5. User can hit ‘c’ to start collecting again. Time stamp will tick again.
		6. To reset trigger number and time stamp, use ‘tc’ command
	+ Hit ‘q’ to terminate the program. This closes connection to host and CollectData.Txt
	+ DO NOT starts program until “CollectData.bin” is renamed since the data is erased when file is opened.

# Setting Up Software to Run with Visual Studio Community

* Make a run directory for your project
* Unzips “CompPol” to this directory
* Run Visual Studio Community program,
* In Visual Community open project 🡪 CompPol.sln
* Start 🡪 Start Without Debug
	+ 1. The software establish connections with host computer. UDP is used to send command and read configuration and status registers. TCPIP is used to receive process data.
		2. Configure all 16 ADC IC and DAC IC.
		3. Read “RunInit.txt” and send to EFADC
		4. Read “PlayBack.txt” and send to EFADC.
		5. The “number of Ethernet byte to collect” is default to 100. User can change this with ‘pk’ command.
		6. Display Run Parameters
		7. Display User Input Menu
* User can change the Run Parameters with command.
* Hit ‘c’ to process and collect data
	+ 1. The program issues “Collect On” command to EFADC
		2. The program collect at least “number of Ethernet byte to collect” and write to CollectData.TXT.
		3. The program issues “Collect Off” command to EFADC
		4. EFADC stop processing data and freeze time stamp
		5. User can hit ‘c’ to start collecting again. Time stamp will tick again.
		6. To reset trigger number and time stamp, use ‘tc’ command
* Hit ‘q’ to terminate the program. This closes connection to host and CollectData.Txt
* DO NOT starts program until “CollectData.bin” is renamed since the data is erased when file is opened.

#  Description of User Interface

 User can change run parameters, start collection, change playback data, read and displays configuration and status registers. For commands that change run parameters, the program prompts and checks for appropriate values before sending to EFADC.

Figure 1: User Interface



* + da: Set DAC output to adjust the pedestal. Table 1 shows the DAC count verse the ADC count with no inputs. The software initialize all DAC to 3150.
	+ This group set the even number of sample to process.
		1. S1: Set number of BCM samples to sum or to write out.
		2. S2: Set number of PMT samples to sum or to write out
		3. S3: Set number of Pocket Cell samples to sum or to write out. Remember first sample and every subsequent 10th samples.
		4. S4: Set number of Helicity samples to sum or to write out. Remember first sample and every subsequent 10th samples.
		5. S5: Set number of T\_Settle samples to sum or to write out. Remember first sample and every subsequent 10th samples.
	+ This group set the number of samples before (NSB) sample that cross threshold to be part of S1,S2,S3,S4, or S5.
		1. nb: NSB for BCM
		2. np: NSB for PMT
		3. nc: NSB for Pocket Cell
		4. nh: NSB for Helicity
		5. nt: NSB for T Cell
	+ This group set ADC count (threshold) a number of samples (NSAT) must cross to create a trigger.
		1. tb: Threshold for BCM (TETBCM)
		2. tp: Threshold for PMT (TETPMT)
	+ ns: Number of samples of BCM or PMT has to be greater (NSAT) than TETBCM and TETPMT to create a trigger.
	+ tm: 1 entry turn on test mode. When test mode is on, samples are from Play Back memory. The Play Back memory outputs data at .5 second interval. When test mode is off, samples are from EFADC front panel Lemo connectors.
	+ ts: 0 selects BCM trigger to start data processing. 1 selects PMT trigger to start data processing.
	+ tc: timestamp clear. 1 will clear time stamp and trigger number. Not persistence (clear line goes hi and then lo)
	+ rr: Read and display configuration and status registers. Calculate and display FPGA die temperature. Display ADC channels 0-4.
	+ mo: 0 selects Semi-Int mode. 1 selects Sample mode
	+ ps: Set the number of trigger NOT to be processed.
	+ pk: Set the Minimum number of Ethernet bytes to collect. See “Number of Bytes Contains in Each Trigger” section to determine how many triggers will be collected
	+ c: Turn Collect on to collect Minimum number of Ethernet bytes set by pk command
	+ q: Quit the program.

ADC Input Channel Mapping:

 ADC0 🡪 BCM

 ADC1 🡪 PMT

 ADC2 🡪 Pocket Cell

 ADC3 🡪 Helicity

 ADC4 🡪 T Settle

#  Read Out Data Format

Read Out Data Format:

Sample mode

* Header Word (one 32 bits)
* Trigger Time (two 32 bits)
* BCM samples (S1/2 32 bits)
* PMT samples (S2/2 32 bits)
* Pockels Cell samples (S3/2 + 1 32 bits)
* Helicity samples (S4/2 + 1 32 bits)
* T-Settle samples (S5/2 + 1 32 bits)
* Event Trailer (one 32 bits)
* ------------------------------------------------------------

Total 32-bits Words = [(S1+S2+S3+S4+S5)/2] + 4 + 3 32-bits words every 12.8 uS

Total Bytes = Total 32-bits Words \* 4

Semi-Int mode

* Header Word (one 32 bits)
* Trigger Time (two 32 bits)
* BCM sum (1 32 bits)
* PMT sum (1 32 bits)
* Pockels Cell sum (1 32 bits)
* Helicity sum (1 32 bits)
* T-Settle sum (1 32 bits)
* Event Trailer (one 32 bits)
* ------------------------------------------------------------

Total 32-bits Words = 9 32-bits words every 12.8 uS

Total Bytes = 36

#  TABLE 1: DAC Values verse ADC Count

**ADC COUNT in Dec (Bit 13 is Overflow) with No Inputs**

Board Serial Number 0001

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DAC Count | ADC 0 | ADC 1 | ADC 2 | ADC 3 | ADC 4 |
| 4095 | 4096 | 4096 | 4096 | 4096 | 4096 |
| 4000 | 4096 | 4096 | 4096 | 4096 | 4096 |
| 3500 | 4096 | 4096 | 4096 | 4096 | 4096 |
| 3100 | 353 | 380 | 391 | 412 | 374 |
| 3000 | 489 | 532 | 526 | 553 | 509 |
| 2900 | 626 | 669 | 659 | 694 | 643 |
| 2800 | 766 | 802 | 791 | 820 | 776 |
| 2700 | 899 | 940 | 927 | 952 | 910 |
| 2600 | 1036 | 1074 | 1059 | 1084 | 1045 |
| 2500 | 1175 | 1207 | 1194 | 1217 | 1179 |
| 2400 | 1305 | 1344 | 1326 | 1352 | 1314 |
| 2300 | 1445 | 1481 | 1462 | 1484 | 1448 |
| 2200 | 1583 | 1614 | 1596 | 1619 | 1589 |
| 2100 | 1720 | 1751 | 1727 | 1751 | 1720 |
| 2000 | 1853 | 1889 | 1863 | 1884 | 1852 |
| 1900 | 1990 | 2023 | 1998 | 2017 | 1986 |
| 1800 | 2124 | 2159 | 2132 | 2151 | 2120 |
| 1700 | 2265 | 2293 | 2265 | 2278 | 2253 |
| 1600 | 2396 | 2430 | 2400 | 2415 | 2390 |
| 1500 | 2536 | 2567 | 2532 | 2548 | 2526 |
| 1400 | 2672 | 2565 | 2532 | 2552 | 2524 |
| 1300 | 2810 | 2839 | 2797 | 2816 | 2792 |
| 1200 | 2947 | 2972 | 2933 | 2948 | 2928 |
| 1100 | 3083 | 3108 | 3068 | 3080 | 3061 |
| 1000 | 3216 | 3242 | 3202 | 3214 | 3200 |
| 900 | 3365 | 3369 | 3335 | 3352 | 3332 |
| 800 | 3489 | 3512 | 3471 | 3480 | 3466 |
| 700 | 3629 | 3651 | 3601 | 3615 | 3600 |
| 600 | 3770 | 3783 | 3735 | 3748 | 3733 |
| 500 | 3902 | 3915 | 3867 | 3882 | 3868 |
| 400 | 4033 | 4054 | 4005 | 4010 | 4003 |
| 300 | 8191 | 8191 | 8191 | 8191 | 8191 |
| 200 | 8191 | 8191 | 8191 | 8191 | 8191 |
| 100 | 8191 | 8191 | 8191 | 8191 | 8191 |

# TCPIP Data Format from EFADC

**Event Header** (2) – indicates the start an event.

 (31) = 1

 (30 – 27) = 2

 (26 – 0) = trigger number

**Trigger Time** (3) – time of trigger occurrence relative to the most recent global reset. Time in the ADC data processing chip is measured by a 48-bit counter that is clocked by the 250 MHz system clock. The six bytes of the trigger time

 Time = TA TB TC TD TE TF

are reported in two words (Type Defining + Type Continuation).

Word 1:

 (31) = 1

 (30 – 27) = 3

 (26 – 24) = TC bits 2 – 0 (duplicated in Word 2)

 (23 – 16) = TD

 (15 – 8) = TE

 (7 – 0) = TF

Word 2:

 (31) = 0

 (30 – 24) = reserved (read as 0)

 (23 – 16) = TA

 (15 – 8) = TB

 (7 – 0) = TC

**Window Raw Data** (4) – raw ADC data samples for the trigger window. The first word identifies the channel number and window width. Multiple continuation words contain two samples each. The earlier sample is stored in the most significant half of the continuation word. Strict time ordering of the samples is maintained in the order of the continuation words. A *sample not valid* flag bit 13 will be set when PTW+1 is odd.

Word 1:

 (31) = 1

 (30 – 27) = 4

 (26 – 23) = channel number (0 – 15)

 (22 – 12) = reserved (read as 0)

 (8 – 0) = number of samples

Words 2 - N:

 (31) = 0

 (30) = reserved (read as 0)

 (29) = sample x not valid

 (28 – 16) = ADC sample x (includes overflow bit)

 (15 – 14) = reserved (read as 0)

 (13) = sample x + 1 not valid

 (12 – 0) = ADC sample x + 1 (includes overflow bit)

**Pulse Parameters** (9) – computed pulse parameters for detected pulses in a channel. The first word identifies the channel number, event number within the block, and pedestal information for the window. Multiple continuation word *pairs* contain information about the pulses detected. For a channel with hits detected:

Word 1 : Channel ID and Integral of pulse

 (31) = 1

 (30 – 27) = 9

 (26 – 23) = channel number (0 – 15)

 (22) = 0

 (21) = One or more samples is overflow = 0x1FFF

 (20) = One or more sample is underflow = 0x1000

 (19 – 0) = 20-bit sum of raw samples that constitute the pulse data set

**Event Trailer:** Indicate the end of an event.

 EVENT\_TRAILER = X"E8000000";

# Example of TCPIP Packet

 Samples From Play Back (as shown in PlayBack.text section)

Mode 0(Semi-Int) ; NSB = 0; S1=S2=32; S3=S4=S5=4; NSB=0

C

ADC 1

ET

ADC 4

ADC 0

Time Stamp

Event Header

ADC 2

ADC 3

Mode 1(Semi-Int) ; NSB = 0; S1=S2=32; S3=S4=S5=4; NSB=0

ET

Not

Valid

samples

ADC 2

Header

ADC 3

Samples

ADC 2

Samples

ADC 0 header

Time Stamp

Event Header

ADC 0

Samples

ADC 1

header

ADC 1

Samples

ADC 3

Samples

ADC 3

Header

ADC 4

Header

# Example of RunInit.txt

0 NSAT

1 RunMode

0 TestMode

0 ResetTimeTrigNum

0 SelEthernetPara

1 TrigSel

500 TETBCM

502 TETPMT

2 NSB\_BCM

32 S1\_BCM

10 NSB\_PMT

34 S2\_PMT

2 NSB\_Pockels\_Cell

4 S3\_Pockels\_Cell

6 NSB\_Helicity

8 S4\_Helicity

10 NSB\_T\_Settle

12 S5\_T\_Settle

0 PreScale

 .

#  PlayBack.txt Used For “Example of TCPIP Packet”

**10 201 202 203 204 513 600 1000 4000 1100 600 500 224 224 225 226 223 225 230 193 205 208 207 207 197 211 206 206 204 205 208 205**

**10 211 212 213 214 513 600 1000 4001 1100 600 500 224 224 225 226 223 225 230 193 205 208 207 207 197 211 206 206 204 205 208 205**

**10 221 222 223 224 513 600 1000 4002 1100 600 500 224 224 225 226 223 225 230 193 205 208 207 207 197 211 206 206 204 205 208 205**

**10 231 232 233 234 513 600 1000 4003 1100 600 500 224 224 225 227 223 225 230 193 205 208 207 207 197 211 206 206 204 205 208 205**

**10 241 242 243 244 513 600 1000 4004 1100 600 500 224 224 225 228 223 225 230 193 205 208 207 207 197 211 206 206 204 205 208 205**

**10 251 252 253 254 205 206 207 208 209 210 211 511 600 1000 2005 700 500 418 207 206 205 204 203 202 201 200 199 198 197 196 195**

**10 261 262 263 204 205 206 207 208 209 210 211 212 511 600 1000 2006 700 500 419 207 206 205 204 203 202 201 200 199 198 197 196**

**10 271 272 273 204 205 206 207 208 209 219 211 212 213 511 600 1000 2007 700 500 420 207 206 205 204 203 202 201 200 199 198 197**

**10 281 282 283 204 205 206 207 208 209 219 211 212 213 214 511 600 1000 2008 700 500 421 207 206 205 204 203 202 201 200 199 198**

**10 291 292 293 204 205 206 207 208 209 219 211 212 213 214 215 511 600 1000 2009 700 500 422 207 206 205 204 203 202 201 200 199**

**10 301 302 303 204 205 206 207 208 209 219 211 212 213 214 215 216 511 600 1000 2010 700 500 423 207 206 205 204 203 202 201 210**

**10 311 312 313 204 205 206 207 208 209 219 211 212 213 214 215 216 217 511 600 1000 2011 700 500 424 207 206 205 204 203 202 211**

**10 321 322 323 204 205 206 207 208 209 219 211 212 213 214 215 216 217 218 511 600 1000 2012 700 500 425 207 206 205 204 203 212**

**10 331 332 333 204 205 206 207 208 209 219 211 212 213 214 215 216 217 218 219 511 600 1000 2013 700 500 426 207 206 205 204 213**

**10 341 342 343 204 205 206 207 208 209 219 211 212 213 214 215 216 217 218 219 220 511 600 1000 2014 700 500 427 207 206 205 214**

 **10 351 352 353 204 205 206 207 208 209 219 211 212 213 214 215 216 217 218 219 220 221 511 600 1000 2015 700 500 428 207 206 215**