

# IOC Application for BPM Data-acquisition System

Yong HU [huyong1979@gmail.com](mailto:huyong1979@gmail.com)

## 1 OVERVIEW

The BPM data-acquisition (DAQ) system of KEK Linac uses Tektronix DPO7104 oscilloscopes as digitizers and signal processors to measure BPMs' signals and calculate beam positions and beam charges. The BPM DAQ software, which was a non-EPICS and windows-program, has been upgraded to be based on EPICS that makes the DPO oscilloscope act as Channel Access Server. This new DAQ system is expected to satisfy the requirement of fast measuring beam-position and beam-charge when 50Hz beam-mode switching of Linac is operated in near future.

The EPICS-based BPM DAQ software (the IOC name is "montekIOC") implements the device support of two kinds of EPICS records. One is waveform record which contains the values of all BPMs' parameters (beam position, beam charge, etc.) in one DPO oscilloscope. The other is stringout record which is used to adjust the oscilloscope's settings (vertical range, horizontal delay, etc.)

## 2 IMPLEMENTATION

The new DAQ software (montekIOC) is based on and evolved from the old one (montekdpo). The basic principles for implementing montekIOC are:

- 1) Keep using shared memory programming technique. All data related to BPM are available in shared memory which is read and written by the IOC. This makes montekIOC compatible with other BPM-related application programs (shmsend.exe, uspwmd.exe, etc.) which are also base-on shared memory.
- 2) To get data from shared memory efficiently, waveform record, instead of ai record, is adopted. And subArray record is used to extract each element from waveform record.
- 3) Ten beam modes are available in operation of KEK Linac. Independent waveform record and subArray records are prepared for each beam mode. And the EPICS database files (.db) are automatically generated by a program.
- 4) DPO7104 oscilloscope can perform single-sequence acquisition (ACQ: STOPAFTER SEQ) or continual acquisition (ACQ: STOPAFTER RUNSTOP). To improve the IOC's updating rate (50Hz), continual acquisition is preferable.
- 5) Since the DPO oscilloscope runs in continual acquisition mode, it's difficult to make montekIOC synchronized with extern trigger signal. So, the IOC is triggered by an EPICS record which is generated by the event timing system (EVG module) via Channel Access.

### 2.1 Source files

Except the file 'montekdpo.cpp', all other files of the old DAQ software are kept in montekIOC. The postfix '.cpp' is changed to '.c' and some modifications are made on the file 'tek.c'. Source files of montekIOC are shown in table 1.

Table 1 Source files of montekIOC

File name	main functions	Note
cmprepare.c	1) cmprepare(); 2) datasize();	prepare shared memory; get oscilloscope's setting parameters from 'teksp.dat' file
tek.c	1) tek_open(); 2)tek_init(); 3) tek_verset(); 4) tek_acq(); .....	operate DPO oscilloscope: open tekVisa, set vertical range, acquire waveform data, etc.
spmeas3.c	1)spread(); 2)spmeas(); 3)spcalc();	measure BPM electrode signals and calculate beam positions and beam charge
spwmerr.c	1)errend(); 2) errmsg();	handle errors
drvMontek.c	1)setup_osc(): setup oscilloscope 2)get_pos_cur(): position and charge	driver support routines for DPO7104; initialize DPO oscilloscope.
devWfMontek.c	1)init_wf_record(); 2) read_wf_montek()	device support of waveform record; get BPMs' positions, beam charges, peak-peak voltages, etc.
devStringoutMontek.c	1) init_stringout_record(); 2) write_stringout_montek()	device support of stringout record; set oscilloscope's vertical range, horizontal delay, etc.

## 2.2 Driver and device support of montekIOC

The file 'drvMontek.c' is for driver support, whose main task is to initialize the IOC (create shared memory, open tekVisa session, set oscilloscope, etc.) by init\_montek(). Additionally, two important functions are implemented. One is setup\_osc() which is used to adjust oscilloscope's settings. The other is get\_pos\_cur() which is used to get beam positions and beam charges and called by 'devWfMontek.c'. setup\_osc() calls many tekVisa-related functions including tek\_verset(), tek\_horset(), tek\_dlyset(), etc. get\_pos\_cur() calls three functions including tek\_acq(), spmeas() and spcalc().

The file 'devWfMontek.c' is for device support of waveform record which is the core of montekIOC. Two device support routines are implemented. One is init\_wf\_record() whose main task is to allocate memory space ('paraData') for BPM-related data. The other is very important for montekIOC and it's read\_wf\_montek() whose task is to make beam positions and beam charges available in waveform record. Once montekIOC gets beam-mode value from event timing system, read\_wf\_montek() will start to be processed and to perform following operations: 1)set oscilloscope by setup\_osc() only when Linac operators change oscilloscope's vertical range, delay time, .etc; 2)select oscilloscope's data source (CH3 and CH4 are dedicated for KEKB e+); 3) get the data of beam positions and beam charges by get\_pos\_cur(); 4)calculate the average and standard deviation values of beam positions and charges; 5) copy all BPM-related data to 'paraData' memory space; 6) The last step: copy 'paraData' to the memory space of waveform record.

The file 'devStringoutMontek.c' is for device support of stringout record whose main task is to adjust oscilloscope's vertical range of CH1,2 and CH3,4, horizontal delay, etc.

### 2.3 EPICS naming convention

montekIOC uses five kinds of EPICS records which include event, calcout, waveform, subArray and stringout. The EPICS naming convention for these records are shown in table 2.

Table 2 EPICS naming convention

record type	naming convention(example)	note
event	LiBM:montekName:MODE (LiBM:montek3A1:MODE)	One event record is used to get beam-mode value from event timing record "LiEV:sa".
calcout	LiBM:montekName:MODE:beam-mode LiBM:montek3A1:MODE:KBE	Ten calcout records are used to choose which waveform record to be processed according to beam-mode.
waveform	LiBM:montekName:WAVE:beam-mode LiBM:montek3A1:WAVE:KBE	Ten waveform records are used for ten different beam-mode operations.
subArray	LiBM:BPMname_bunch:parameter:beam-mode LiBM:SP_31_4_1:ISNGL(or IAVE, ISGM):KBE	It's used to extract each element of waveform record. 260 (2*13*10) subArray records for one BPM.
stringout	LiBM:montekName:settings LiBM:montek3A1: VER(or DLY, SP, VER_34)	Four stringout records are used to change oscilloscope's vertical range of CH1,2 and CH3,4, horizontal delay, etc

### 2.4 auto-generating database

A lot of EPICS records need to be prepared. The numbers of subArray records for one BPM can be calculated as following: 13 (beam-parameters: x, y, i, average, sigma, peak-peak voltages) \* 10 (beam modes) \* 2 (bunches) = 260. Each oscilloscope (montekIOC) measures 2~6 BPMs. So, if the oscilloscope controls 6 BPMs, 1560 subArray records are needed.

Preparing so many records manually is very difficult. A simple windows-program (autoConfigure.exe, the source file is autoConfigure.cpp) is developed to auto-generate database files (.db) and startup file (st.cmd).

When the oscilloscope is rebooted, it can automatically update files, start 'shmsend.exe' and 'uspwmd.exe', configure montekIOC, generate .db and st.cmd files and finally start the IOC by using some .bat files (Startup\_montek.bat, start\_monsoft.bat, config\_startIOC.bat, etc.).

## 3 USAGE

Using montekIOC software is easy. The software package is available on Linac servers: [\\orange\montek](#) or [\\eggplant\public\BPM](#) . Find the 'epics' folder and copy the whole folder (it contains 'base-3.14.8.2' and 'montekIOC') to Local Disk (C) of PDO7104 oscilloscope. Then reboot the DPO. The montekIOC will automatically run.

In the future, if 'shmsend.exe' and 'uspwmd.exe' are not used, comment out the corresponding commands in 'start\_monsoft.bat'.

'cygwin.bat' can be used to recompile (use 3.80 version of make.exe) the source codes of montekIOC. It's better to install Microsoft Visual Studio and Cygwin in DPO oscilloscope.

#### 4 CONCLUSION

Many methods and techniques for implementing montekIOC were tested, such as waveform record VS ai record, continual acquisition VS single-sequence acquisition, single-thread programming VS multi-thread programming.

The best combination is waveform record + continual acquisition + single-thread programming.

Performance tests on montekIOC were carried out. The test results show that montekIOC can work.