USE INTERRUPT DRIVEN MODE TO REDESIGN AN IOC FOR DIGITAL POWER SUPPLY AT SSC-LINAC

Shi An, Wei Zhang, Kewei GU, Junqi Wu, Xiaojun Liu, IMP, Lanzhou, China

Abstract

SSC-LINAC control system is based on EPICS architecture. The sub control system of digital power supplies is a kind of IOC send and receive custom command via Ethernet and TCP/IP protocol. The old IOC is designed to use period scan mode IOC, and there are so many digital power supplies, that we can't make sure every connect condition of digital power supply is fine. IOC must wait a long time if one of them can't connect correctly and other digital power supply's PV may also be blocked. An IOC that uses interrupt driven mode to avoid the shortcoming was designed. This will be described in this paper.

INTRODUCTION

SSC is a separated-sector cyclotron. To improve the efficiency of HIRFL, a linear accelerator is considered as a new injector for SSC of HIRFL [1]. The SSC-LINAC control system is based on EPICS. There are many sub systems such as power supply control system, vacuum monitor system and so on. The structure of SSC-LINAC control system is shown in Figure 1.

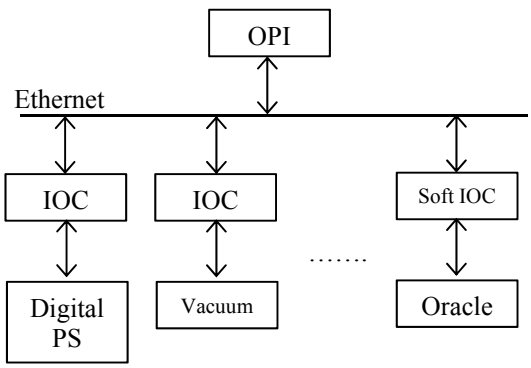

Figure 1: Structure of SSC-LINAC control system.

Almost all of the power supply in the system uses digital power supply. So control interface between IOC and power supply is Ethernet interface. IOC sends and receives predefined string command via the Ethernet with the PS (power supply).

DESCRIPTIONS

This part wills description the old PS IOC design and new design of the IOC, also description implement of the new IOC. Descriptions include two sections and each section is introduced in detail as following. The structure of the PS IOC is shown in Figure 2.

Structure of the Origin PS IOC

The bottom of the structure is digital PS. Above the PS is Ethernet interface and use TCP/IP as low level transfer protocol. In the middle layer is device support of EPICS architecture. The device support has two functions. First function is a periodic that sends a read command string to PS in every one second and receives current state of the PS. The second function is sends a write command string to PS. The top of the structure is record support.

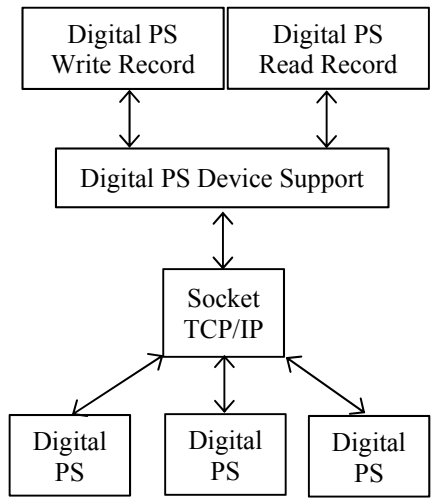

Figure 2: Structure of the PS IOC.

In the real case we create two PVs for every digital PS and there are 78 power supplies in SSC-LINAC. Actually all of PVs are run in one thread in the structure. If just one digital PS is down and all of other digital power supply's PV also be blocked (blocked time depend on the timeout setting). So we must design a new device support to avoid the bad situation.

Design a New Device Support for Digital PS

First switch the Record periodic SCAN ("1 second") to SCAN rate driven from a device specific source ("I/O Intr"). In our case separated thread for every one PS is used.

Second step is writing device support use interrupt model and implement in the following sections.

1. init

```
static long init(int phase)
{
        if(phase==0)
                initHookRegister(&start_ps_thread);
        return 0;
}
```


ISBN 978-3-95450-148-9

Register a hook function at the device support init(int) function.

2. Init hook

```
static void start_ps_thread (initHookState state)
{
   ELLNODE *cur;

   if(state!=initHookAfterInterruptAccept)
                        return;

   for(cur=ellFirst(&allprngs); cur; cur=ellNext(cur))
   {
        struct prngState *priv = CONTAINER(cur,
struct prngState, node);
        priv->generator = epicsThreadMustCreate(
           "psThread",
           epicsThreadPriorityHigh,
           epicsThreadGetStackSize(epicsThreadStack
Small),
           &ps_thread, priv);
   }
}
```

The hook function starts a thread for each of the digital PS.

3. Digital PS thread

```
static void ps_thread(void* raw)
{
           struct prngState* priv=raw;
           char send_buf[16] = {0,};
           char buf[BUFF_SIZE] = {0,};
           char *endptr;

           //init socket
           int sockfd;
           struct sockaddr_in device_addr;
           ……
           //connect to PS
           …...

           while(1)
           {
           //send and receive something from the PS

           epicsMutexMustLock(priv->lock);
           //write useful data to record
           ……
           //queue a request
           scanIoRequest(priv->scan);
           epicsThreadSleep(interval);

           }
}
```

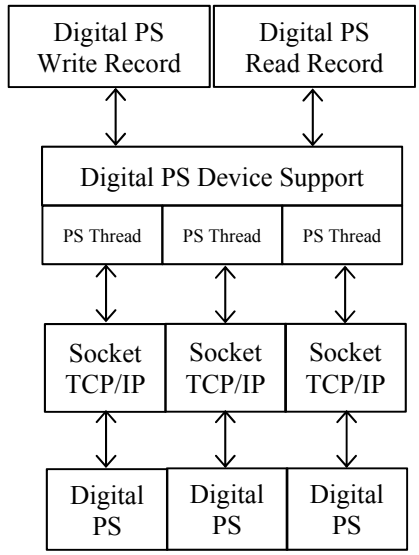

Figure 3: New structure of the PS IOC.

Every PS thread just connect to one PS and when the useful data is ready the function scanIoRequest () to process associated records. Figure 3 shows the structure of the new device support for digital PS.

CONCLUSION

The origin structure of PS IOC uses periodic method. All processing is done on a single thread. So any digital PS connect block will cause the entire IOC to become unresponsive [2].

New design uses separated thread to interact with each one of PS and results will push into record automatically. The new design solves the problem and make some new advantages.

First advantage is more real-time. Digital PS thread could run in a very fast loop and don't affect each other. So the up level records could get the data faster than before.

Second is more stability. One thread crash will not affect other thread, every thread just like run in a sandbox. Whole system become more stability.

REFERENCES

[1] He Yuan, Wang Zhijun, etc. "6-14 Design of Linear Injector for SSC", IMP & HIRFL Annual Report, 2009, P(256-257).

[2] Michael Davidsaver. "Basic EPICS Device Support", https://pubweb.bnl.gov/~mdavidsaver/#doc

ISBN 978-3-95450-148-9