

# Programming, Testing and Integration of new ADC module (IP330) for SST-1 COIL Power Supply Control System

Dinesh Kumar Sharma, Murtuza M Vora and SST-1 Power System Division  
 Institute for Plasma Research, Bhat, Gandhinagar-382428, Gujarat  
 E-mail: dinesh@ipr.res.in

## Introduction

Power supply control system (PSCS) is the subsystem of SST-1 Coil Power Supply (CPS) system.

PSCS uses VME bus architecture and RTOS VxWorks for the data monitoring and control of SST-1 CPS system. The PSCS uses Analog I/O, Digital I/O module, processor card and timing module for its control and monitoring functions.

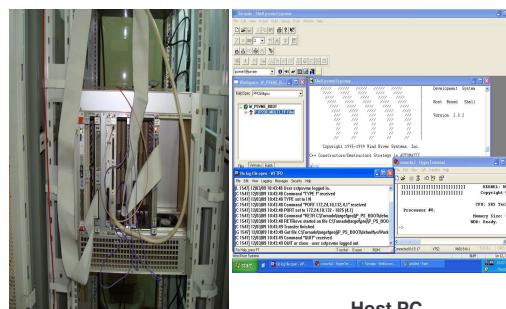
Previously we were using VGD4 ADC module from Pentland Systems for the data acquisition of (voltage and current parameters).

The intermittent behaviour of VGD4 module during ramp up and ramp down was saw toothed type instead of liner profiles. So It was decided to replace with IP 330 ADC module from Acromag Inc.

This paper discusses embedded programming in VxWorks for the IP 330.

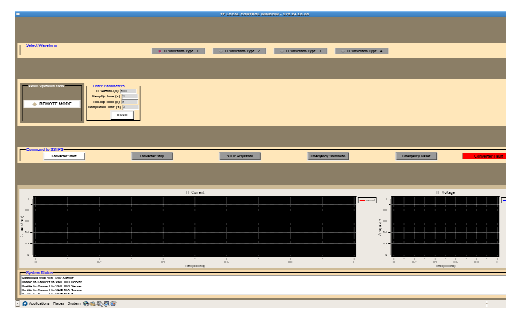
## PSCS

- PSCS uses VME bus architecture
- SBSVG4 (PPC 750) processor module
- Digital I/O module- Acromag AVME 9470
- Analog In – Acromag IP 330 module
- Analog Out module- Acromag IP 230
- RTOS- VxWorks with IDE Tornado
- Programming in C and C++
- GUI developed in TCL/Tk
- Host and GUI PC communicate through client/server mechanism
- It uses GPS receiver for time synchronization
- V/I and I/V converter modules and interface relay logics



PSCS VME System

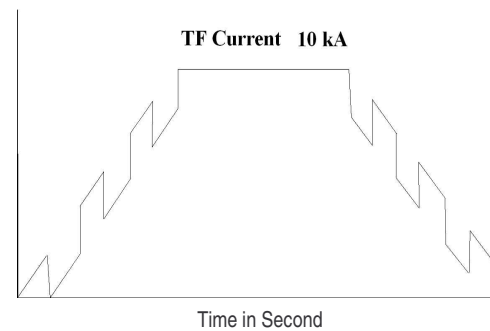
Host PC



GUI

## Intermittent Behaviour of VGD4 module

During ramp up and ramp down the intermittent behaviour of VGD4 module is shown in the graph during spare coil testing

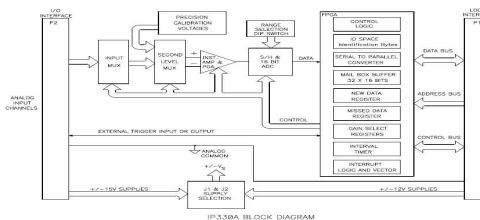


## Key features of AVME 9660 carrier and IP 330 module

- Interface for four IP module
- Provides full IP data & register Access
- LED display simplify debugging
- Front panel 50 pin connector access
- Supports two interrupt channel per IP module

### Key features of IP330 module-

- A/D 16 bit resolution
- 5  $\mu$ sec conversion time
- 16 differential and 32 single ended channels
- Individual channel mail box
- Interrupt upon conversion complete mode
- Programmable control of channel scanning
- User programmable interval time
- Uniform continuous, single and burst scanning mode
- External trigger scan mode
- User programmable scan mode
- Precision on board calibration voltage



## Hardware configuration

- Analog input range selection by jumper setting
- We have chosen  $\pm 10V$  accordingly jumpers 2,5,6,9 are ON and 1,3,4,7,8 are kept OFF
- We are using external power supply  $\pm 15V$  accordingly J1-1&2(IN) and 2&3(OUT) and J2-1&2(IN) and 2&3(OUT)

- The carrier board interfaces with VME bus as 1Kbyte block of address locations in short I/O address space.
- A10 to A15 lines are provided as jumper selectable on the carrier board.
- In our case for IP 330 we have chosen as F400 as the base address of the board.
- Address pins A10 to A15 jumpers are inserted except pin A11 to generate the base address as F400.

## Software configuration

- The board is addressable on 1K byte boundaries in the short I/O A16 address space.
- Carrier board does not have ID prom. It has board status registers.
- 1K byte of memory is composed of blocks of memory for the I/O and ID spaces up to four IP modules.
- Short I/O memory map for IP A module  
 Base address+(0000 to 007F)- IP A I/O space  
 (0080 to 00BF)-IP A ID space  
 (00C0 to 00FF)-carrier board status registers
- 0-255 bytes are assigned for each IP module.
- On the carrier board memory map there are configurable registers- status register, interrupt level register, IP error register, interrupt enable and pending register and interrupt clear register.

## Sample program for IP 330 module

```

//unsigned short* ch1=(unsigned short*)ip330_base+0x42;
volatile char* ip330_scb=(char*)ip330_base+0x11;
volatile char* ip330_ler=(char*)0x0cf4e1;
volatile char* ip330_lr=(char*)0x0cf4c3;
volatile char* ip330_lpr=(char*)0x0cf4e5;
volatile char* ip330_cbr=(char*)0x0cf4c1;
volatile char* ip330_iprr=(char*)0x0cf4c5;
volatile char* ip330_icr=(char*)0x0cf4e5;
volatile char* ip330_csr=(char*)0x0cf4c1;

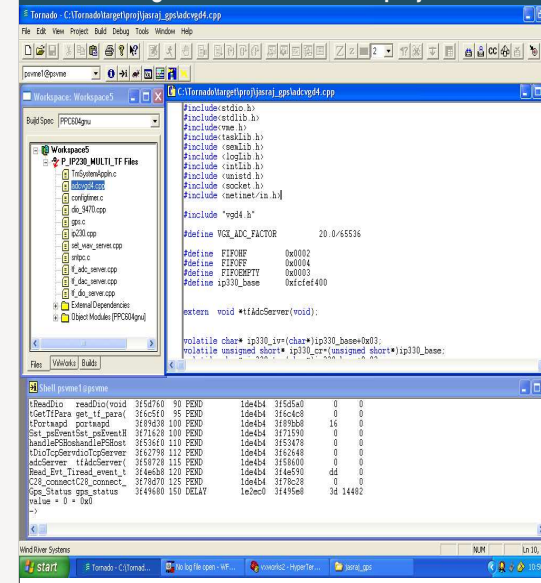
int CCSAdcSockfd=0;
short IP330Data[5];
SEM_ID SemFile;

short ADC_DISABLE_FLAG=0;

/******Initialize TCP Server and set IP330 modes*****
int initAdc_IP330()
{
int tid;
tid=taskSpawn("adcServer",115,0,20000,(FUNCPTR)ifAdcServer,0,0,0,0,0,0,0,0);
setIP330();
}

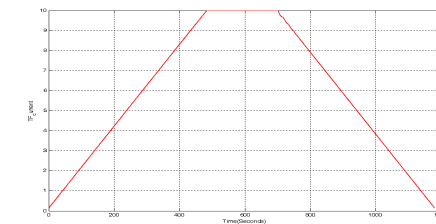
/******Set IP330 in Continuous Acquisition Mode, Bank 0 is enabled *****
int setIP330()
{
*ip330_cr=0x0906// single scan with no interrupts
*ip330_ip=0x28;
*ip330_scv=0x00;
*ip330_acv=0x04;
*ip330_ct=0x04;
//taskDelay(5);
*ip330_gsc0=0x00;
*ip330_gsc1=0x00;
*ip330_gsc2=0x00;
*ip330_gsc3=0x00;
*ip330_gsc4=0x00;
taskDelay(5);
}
    
```

## Integration with Main PSCS projects

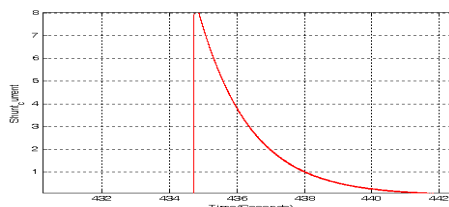
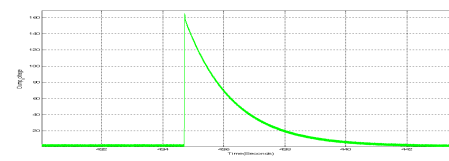
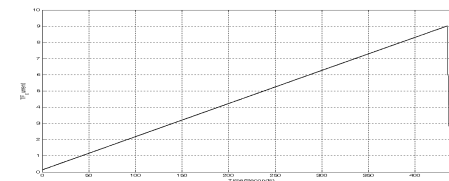


## Results

10kA current in 13<sup>th</sup> TF coil tested on 29/11/2010



TF current, Dump voltage and Shunt current during quench For the 13<sup>th</sup> TF coil tested on 29/11/2010



## Conclusion

IP330 module programming has been completed successfully and now it is the part of PSCS. We have tested 13 TF coil section and acquired TF voltage and current during normal operations and TF current, TF voltage, dump voltage, isolator voltage and shunt current during quench. IP 330 module is being used from March 2010 onwards. The module is behaving as per its specifications without a single failure.

## Reference

- <http://www.vita.com>
- <http://www.acromag.com>
- <http://www.windriver.com>