IIA Technical Report Series, No.2, pp.1-19, 2010 Report No:IIA-TRS-0210

Data acquisition electronics for Stellar Photo-Polarimeter.

Srinivasulu,G.

This report should be read along with the report titled “ A stellar Photo-Polarimeter”. This report describes the concept of the electronics design and implementation of PMT pulse counting , rotation of Half wave plate, chopper spinning and shutter control and host computer communication.

Contents

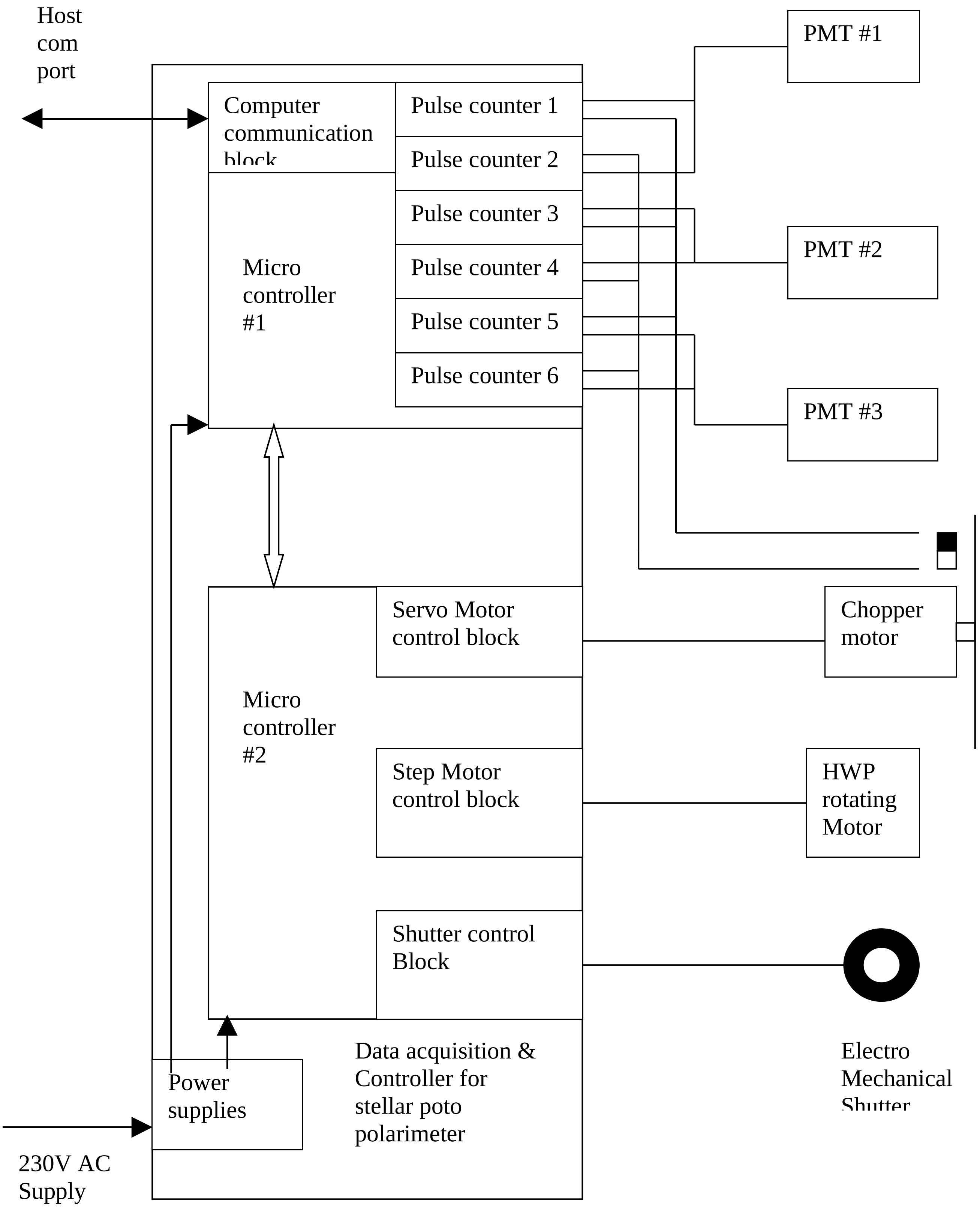
1. Introduction

The data acquisition electronics for the stellar photo-polarimeter instrument has been developed in the institute electronics laboratory. The electronics controller unit acts as an interface between the data acquire host computer and various sub parts of the polarimeter. The controller accepts the instructions from the host computer in form of commands, interprets them and executes the task at lower levels to maintain the required timing sequence in acquiring the light polarization data in the form of photon counts.

A concise command set is generated for the data acquisition and control of sub systems and to check the proper operation of the instrument. The command communication between the host computer and the controller is established based on commonly available serial port (comport) on the desk top personal computers (PCs). The serial port communication is simple in operation and reduces the additional hardware requirement on the computer in the form of bus based IO boards. The serial port comes with built-in drivers in all operating systems. Therefore, the data acquisition and analysis programs can be developed on any operating system of the computer.

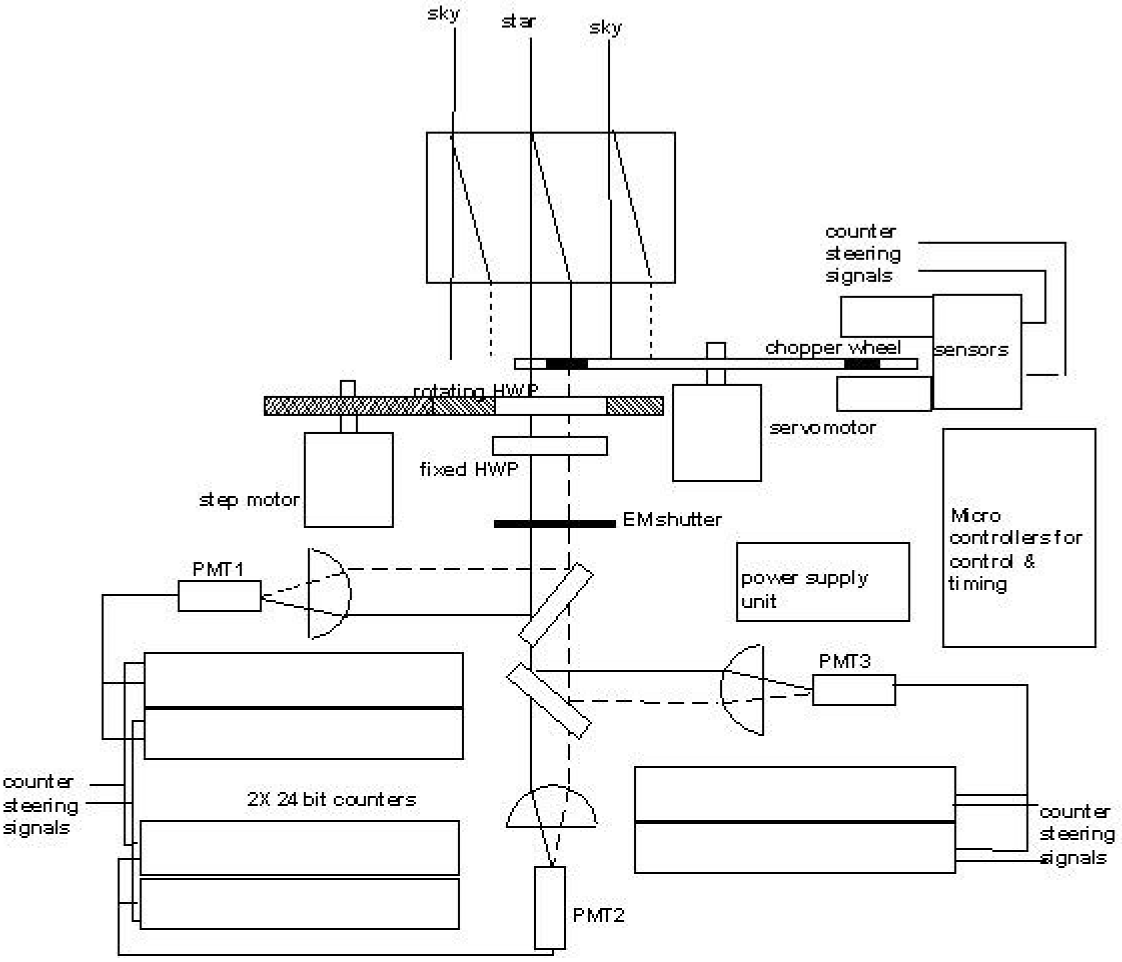
The polarimeter control electronics is a stand alone unit and it consists of a couple of microcontrollers, six (3X2) PMT pulse counters, an electro-mechanical shutter operation block, a stepper motor control block, a servo motor control block and associated power supplies. The unit works in a slave mode, wherein it initiates actions only on receiving the command from host computer and goes into standby mode after completion of the present command execution. The list of operating commands are described in the following pages. The pulse counters are designed for 24 bit length binary counting with input steering mechanism. The shutter controller is to control an electro-mechanical shutter employed for the PMT protection from intense light. The servo controller controls the spinning velocity of the chopper wheel, attached DC motor. The velocity can be set from the host computer.The step motor is employed for the accurate rotation of Half-Wave Plate (HWP).

2. Block diagram of the controller



3. Principle of Operation

The data acquisition electronics block diagram in association with ray diagram of the instrument is shown below.



Data acquisition electronics block diagram.

This technical report is not meant for concept explanation or design description of the polarimeter instrument. However, it is easier to explain the data acquisition electronics part if the instrument working principle is known. In this connection the working principle of the instrument is explained in a few sentences in electronics point of view. When the polarized light ray from the telescope passes through the optical system as shown in the block diagram, the ray splits into ordinary and extra-ordinary rays. When these rays are passed through a set of half wave plates, the intensity of the ray varies with half wave plates orientation angle. These intensity values are measured using Photo Multiplier Tubes (PMT). Therefore the PMT pulse rate is a measure of the ray intensity valve falling the PMT cathode at that instant (for more description of the working principle of this instrument, refer our stellar photo polarimeter report).

The sequence of tasks performed for one data acquisition cycle is explained here. As explained earlier the controller does not execute any task unless host computer initiates them. The following tasks are a sequence of commands sent from the host computer. The data acquisition begins by setting the Chopper wheel rps (revolutions per second) The chopper wheel, which sends the ordinary and extra ordinary rays alternately to the PMT cathode, starts spinning at this rpm. Normally, it is set at 100 rps or 6000 rpm. The rpm is maintained very accurately by the servo control mechanism. There are two sets of sector slots on the chopper wheel, which gives the effective chopping rate at 200 Hz. The dwelling time of each ray in its slot is detected by a pair of sensors and it gives the steering information to counters as to which ray is currently in the slot. Accordingly the corresponding counter opens for PMT counts if master gate is enabled. Then HWP brought to an arbitrary reference by issuing an other command. This is achieved with a step motor whose step angle is 1.8 degrees or 200 steps per rotation. Once the HWP is referenced, the counters are cleared. Then the count corresponding to the number of chopper wheel rotations, called integrations, is loaded into another counter. Then the counter's master gate is enabled. The steering electronics looks for the opening edge of the ordinary ray channel and starts the counting into the corresponding counter. At the closing edge of that channel the counter is closed for counting. Then the steering electronics looks for the opening edge of the extraordinary ray slot then opens the counts into its other counter pair and decrement the integration count. This process repeats for the given integration number. This way the PMT pulses corresponding to ordinary and extraordinary rays are co added in separate counters for given number of integrations. The same operation goes on with the other two PMTs simultaneously. During this time host computer keep monitoring the progress through the communication commands. On completion of integration host issues an other command to read the counters. Then HWP is moved to the next position and repeats the operation for one full circle. At the end the count data analyzed by the host and decides to go next set of data if needed. The additional tasks performed by the controller are to open the shutter during exposure time and close it during other times, reporting the working status of sub parts to the host on request.

4. Command set

Command structure for controlling the instrument and acquiring data by the host computer is described in the following pages. The serial communication from host computer can be established with instrument controller by the following com port (RS232 serial port) settings:

- Baud rate : 9600
- Stop bits : 1
- Parity : N
- Hand shake : no hand shake controls
-

1.Echo check:

1a). Echo1 : $11 *($ indicates hexadecimal value*)

Description: An ASCII character sent followed by the above command to the controller from computer’s comm. port returns the same character. This command is useful to check the integrity of the communication link between computer and controller.

Format: <Command > < ASCII character >.

Return: < same ASCIIcharacter >

Ex: send $11 “A” through serial port as a 2bytes string

“A” will be received back if communication is properly established.

1b). Echo 2: $12

Description: An ASCII character sent followed by the above command to the controller from computer’s comm. port returns the next character. This command is useful to check the integrity of the communication link between computer and controller.

Format: <Command > < ASCII character >.

Return: < Next ASCIIcharacter >

Ex: send $11 “A” through serial port as a 2bytes string

“B” will be received back if communication is properly established.

2. Instrument check :

2a). HWP rotation test: $21

Description: The command checks the rotation of Half Wave Plate and returns the status.

Format: <Command >

Return: < status byte >

status byte = O if HWP rotation is OK

status byte = N if HWP rotation is NOT OK.

2b). Chopper spinning test: $22

Description: The command checks the spinning of chopper and returns the status.

Format: <Command >

Return: < status byte >

status byte = O if spinning

status byte = N if not spinning

2c). Shutter operation test: $24

Description: The command operates the electro-mechanical shutter for ten times and sends the acknowledgement.

Format: <Command >

Return: < status byte >

status byte = O operated

3. Shutter operation.

3a). Shutter Open: $A1

Description: The command opens the shutter.

Format: <Command >

Return: none

3b). Shutter Close: $A2

Description: The command closes the shutter.

Format: <Command >

Return: none

4. Clear counts:

4a). Clear counts of PMT1: $31.

Description: The command clears the ordinary and extraordinary ray counts of PMT1. **Format:** <Command >

Return: none

4b). Clear counts of PMT2: $32.

Description: The command clears the ordinary and extraordinary ray counts of PMT2. **Format:** <Command >

Return: none

4c). Clear counts of PMT3: $34.

Description: The command clears the ordinary and extraordinary ray counts of PMT3. **Format:** <Command >

Return: none

4d). Clear counts of all PMTs: $38.

Description: The command clears the ordinary and extraordinary ray counts of PMT1,PMT2 and PMT3.

Format: <Command >

Return: none

5.Start count:

5a). Start count of PMT1: $41.

Description: The command starts the count on ordinary and extraordinary ray channels of PMT1.

Format: <Command >

Return: none

5b). Start count of PM2: $42.

Description: The command starts the count on ordinary and extraordinary ray channels of PMT2.

Format: <Command >

Return: none

5c). Start count of PMT3: $44.

Description: The command starts the count on ordinary and extraordinary ray channels of PMT3.

Format: <Command >

Return: none

5d). Start count of PMT 1,2&3: $48.

Description: The command starts the count on ordinary and extraordinary ray channels of PMT1,PMT2 and PMT3.

Format: <Command >

Return: none

6. Stop counts :

6a). Stop count of PMT1: $51.

Description: The command stops the count on ordinary and extraordinary ray channels of PMT1.

Format: <Command >

Return: none

6b). Stop count of PMT2: $52.

Description: The command stops the count on ordinary and extraordinary ray channels of PMT2.

Format: <Command >

Return: none

6c). Stop count of PMT3: $54.

Description: The command stops the count on ordinary and extraordinary ray channels of PMT3.

Format: <Command >

Return: none

6d). Stop count of PMT1,2&3: $58.

Description: The command stops the count on ordinary and extraordinary ray channels of PMT2.

Format: <Command >

Return: none

7). Read counts: $60

Description: The command reads the counts of all channels and send them to the host(computer) in the order shown below.

Format: <Command >

Return: PMT counts of 3X 2 channels

Format of count data from controller:

```
<counter1 ordinary ray msb>            // most significant byte
<counter1 ordinary ray nsb>            // next significant byte
<counter1 ordinary ray lsb>            // least significant byte
<counter1 extra-ordinary ray msb>  // most significant byte
<counter1 extra-ordinary ray nsb>   // next significant byte
<counter1 extra-ordinary ray lsb>   // least significant byte
<counter2 ordinary ray msb>          // most significant byte
<counter2 ordinary ray nsb>          // next significant byte
<counter2 ordinary ray lsb>          // least significant byte
<counter2 extra-ordinary ray msb> // most significant byte
<counter2 extra-ordinary ray nsb>  // next significant byte
<counter2 extra-ordinary ray lsb>   // least significant byte
<counter3 ordinary ray msb>          // most significant byte
```

<counter3 ordinary ray nsb> // next significant byte
<counter3 ordinary ray lsb> // least significant byte
<counter3 extra-ordinary ray msb> // most significant byte
<counter3 extra-ordinary ray nsb> // next significant byte
<counter3 extra-ordinary ray lsb> // least significant byte

8). End of integration: $81

Description: The command checks counter1 (PMT1) integration is in progress or integration is complete.

Format: <Command >

Return: ASCII character P if the integration is in progress /
ASCII character C if the integration is completed.

9). Set chopper rps: $72

Description: The command sets the chopper RPS (revolutions per second).

Format: <Command > < rps>

Return: none

rps range : $1..$FF

10). Move HWP to reference position: $C0

Description: The command moves the HWP to its reference position.

Format: <Command >

Return: ASCII character R when the HWP comes to reference position

11). Move HWP to a given Position

11a). HWP to new Position in Clockwise Direction: $B1

Description: The command moves the HWP to its new position in the clock wise direction from its present position by given number of steps. The step angle is 1.8 degrees.

Format: <Command ><steps>

Return: ASCII character M after moving the HWP to new position.

steps range : $1..$FF

11b). HWP to new Position in counterclockwise Direction: $B2

Description: The command moves the HWP to its new position in the counterclockwise direction from its present position by given number of steps. The step angle is 0.9degrees.

Format: <Command ><steps>

Return: none

steps range : $1..$FF

12. Set Integrations : $D0

Description: The command sets the integration number.

Format: <Command >< integration number high byte >< integration number low byte >

Return: none

Integration number range : $1 .. $FFFF

13. Acquire data : $E0

Description: The command starts acquiring the counts data from reference position of HWP. On completion of number of integrations, the data is sent to host .Then moves the HWP to next position and start integrating the counts. The process repeats till the completion of one rotation of HWP. If stop command is not issued, the above process repeats it self.

Format: <Command >

Return: 3X2 channel data as shown below.

Format of count data from controller:

```
<counter1 ordinary ray msb>           // most significant byte
<counter1 ordinary ray nsb>           // next significant byte
<counter1 ordinary ray lsb>           // least significant byte
<counter1 extra-ordinary ray msb>  // most significant byte
<counter1 extra-ordinary ray nsb>   // next significant byte
<counter1 extra-ordinary ray lsb>   // least significant byte
<counter2 ordinary ray msb>         // most significant byte
<counter2 ordinary ray nsb>         // next significant byte
<counter2 ordinary ray lsb>         // least significant byte
<counter2 extra-ordinary ray msb> // most significant byte
<counter2 extra-ordinary ray nsb>  // next significant byte
<counter2 extra-ordinary ray lsb>   // least significant byte
<counter3 ordinary ray msb>         // most significant byte
<counter3 ordinary ray nsb>         // next significant byte
<counter3 ordinary ray lsb>         // least significant byte
<counter3 extra-ordinary ray msb> // most significant byte
<counter3 extra-ordinary ray nsb>  // next significant byte
<counter3 extra-ordinary ray lsb>  // least significant byte
```

5.Description of control electronics

The controller operations are performed by two microcontrollers. One microcontroller looks after the host communication and counters management and other microcontroller is dedicated for controls of the servo motor, step motor and shutter operation. These micro controllers are 8 bit PIC16F877A with 8kb program memory. The built in UART is used for host communication by translating the signals to RS232 levels. The communication between host and controller operates at pre fixed baud rate of 9600 bauds. The PMT signals are passed through Pulse Amplifier discriminator (PAD) to get a TTL compatible pluses. The PAD is a small stand alone unit fixed close to the PMT output. PAD pulsed are connected to the counters through low loss cables and the power to the PAD unit is supplied from the controller along with signal cable. The three sets of counters, meant for three PMTs are 24 bit long binary counters. The 24bit counter is built on two ICs. First 8bit counter, 74F289 is a fast counter with 10ns response time. This counter is integrated with Intel 8254 16 bit counters to make the system compact and programmable. One set of counters built on a separate PCB. All 3 sets of counters are sitting on the common data bus of the microcontroller. The motor control microcontroller built on an other PCB sits on the common bus along with the counters. The communication microcontroller passes the commands to the second controller on the bus if the commands meant for motors or shutter. Then the second microcontroller interpret them executes them and gives acknoledgement. The chopper motor is a DC motor with built in incremental encoder and processor. The built in processor maintains the velocity at set point. The set point can changed through the communication port. The HWP rotator is a step motor from SLOSYN with 200 steps per rotation. The translator circuit is built with L297 & L298 integrated circuits. The electrically operated shutter is from UNIBLITZ and it needs dual volts for operation. The open/ close operations are timed by the micro controller. Additionally, there are LED indicators provided to indicate some crucial parameters on the instrument panel.. These lamps can be switched off optionally in case darkness in required.

6. Present status

The electronics circuits are designed and made into five PCBs and a back plane. The components and motors are procured. The PCBs are loaded with components and checked for static power problems. Programs for the micro controllers are written, in assembly language using Microchip IDE, for command interpretation, execution and for error reporting on one microcontroller. On the second microcontroller programs are written for step motor control servomotor control shutter control programs are written. Initial working of the microcontrollers are debugged from PC using terminal software. Presently the integration with host computer is going on in the laboratory.

7. Acknowledgements

We deeply acknowledge Prof. Hassan, director of the institute, Mr. Sundararajan, HOD, Electronics Division and other senior people of the institute who encouraged and given constructive advices time to time. We thank Mr. K. Ravi, Mr. M.R. Somasekar, Mr. Sagayanathan, Mr. P.Janakiram and other colleagues of the electronics division for the help provided by them in bringing up the instrument to this stage.