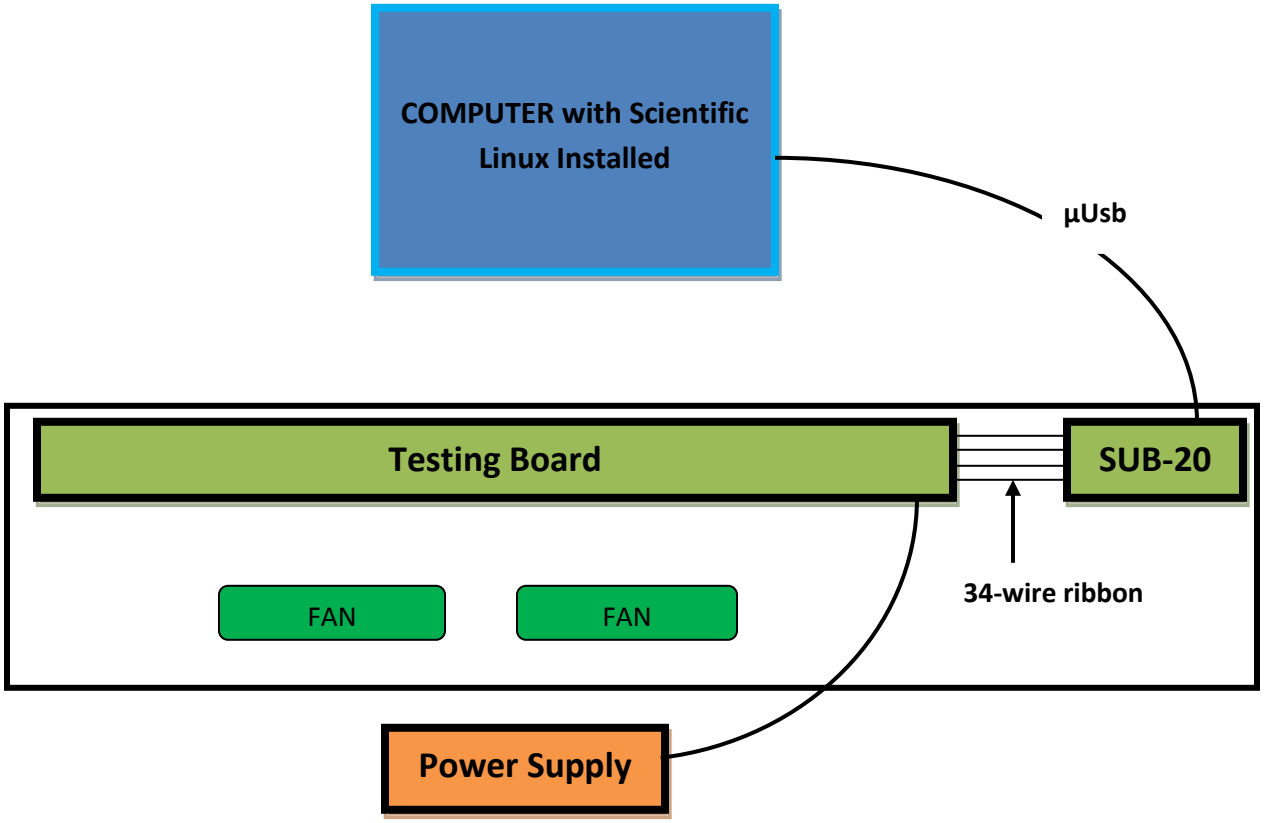
POWER MEZZANINE TESTING

INTRODUCTION

Power Mezzanine Testing program is designed to monitor or test μHTR **Power Mezzanine (PM)** and **Auxiliary Power Mezzanine (APM)** for long term (~20 hour) stability tests. Power mezzanine provide required voltages to various parts of the μHTR(Micro HCAL trigger and readout card). If one of the PM/APM fails during the data collection then It will cause the loss of data. So It is important to check PM/APM stability test for at least 20 hours. this testing program designed to work with SUB-20 module which communicates with test board (having 2 slots for APM and 3 slots for PM) through I2C(Inter Integrated circuit) communication and also have interface with computer through μusb.

During the test power mezzanines are inserted into the appropriate slots on test board and It is powered ON. A test software is then started and a quick test of margin up and margin down mode (setting the output voltage 5% high or low) are conducted followed by a long test of the nominal settings. During each test the output voltages, currents, power, and temperature of the modules are recorded every 10 seconds. At the end of every test the average and extremum values of every quantity are used to evaluate the worthiness of the modules which are recorded in an output database.



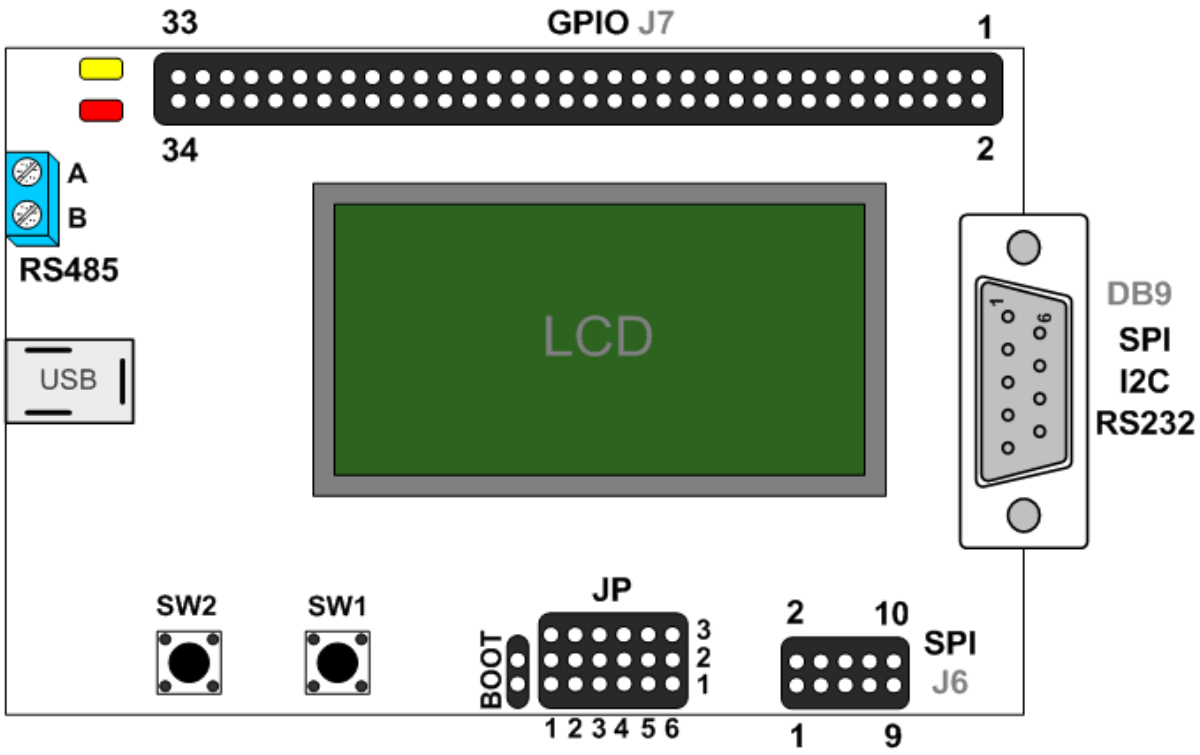
Testing Equipments :

1) SUB-20 Module :

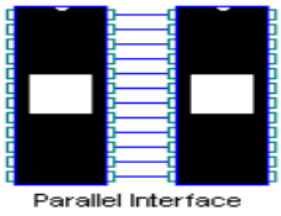
SUB-20 is a versatile and efficient bridge device providing simple interconnect between PC (USB host) and different hardware devices and systems via popular interfaces such as **I2C(Inter Integrated Circuit), SPI** , **MDIO RS232** , **RS485**, **SMBus**, **ModBus**, **IR** and others. SUB-20 is a powerful I/O controller with **32 GPIO(General Purpose Input/Output)**, 8 Analog Inputs, PWM Outputs, Edge Detectors, LCD, Leds and push buttons. SUB-20 is equipped with status Leds, LCD and buttons.

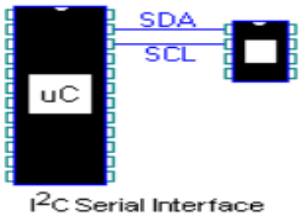
NOTE : We are using only the features marked in blue colored above of SUB-20 module.

Reference : http://www.xdimax.com/sub20/sub20.html



- **I2C :** It is a two wire , low-medium speed communication bus developed by Philips Semiconductors in the early 1980s. It provides a low cost, powerful chip to chip communication between different components which resides on the same circuit board.





Old days :

- chip-to-chip communications used many wires in a parallel interface
- often requiring ICs to have 24, 28, or more pins.
- Many of these pins were used for inter-chip addressing, selection, control, and data transfers
- In a parallel interface, 8 data bits are typically transferred from a sender IC to a receiver IC in a single operation

Now :
● performs chip-to-chip communications using only two wires in a serial interface
● allowing ICs to communicate with fewer pins
● The two wires in the I2C Bus are called **Serial Clock (SCL)** and **Serial Data (SDA)**
● These two wires carry addressing, selection, control, and data, one bit at a time.
● The SDA wire carries the data, while the SCL wire synchronizes the sender and receiver during the transfer.

Some most significant features :

i. Only two bus lines are required
ii. Simple master/slave relationships exist between all components
iii. Each device connected to the bus is software-addressable by a unique address (in our case it provides us a unique mac address for each PM/APM**)**

Reference : http://www.i2c-bus.org/i2c-bus/

- **GPIO :** It is a generic pin on an integrated circuit whose behavior, including whether it is an input or output pin, can be controlled by the user at run time. A GPIO port is a group of GPIO pins (typically 8 GPIO pins here it is 32) arranged in a group, and treated as a single port.
  Pin 1 and 2 of GPIO is used for I2C communication between SUB-20 and testing board. Pin 25 to 32 can be used for collecting information (like temperature, voltage, current) from ADC's of testing board. Pin information of GPIO can be seen into the SUB-20 manual. Manual can be downloaded from the following link.
  http://www.xdimax.com/sub20/doc/sub20-man.pdf

2) Micro usb cable :

It simply connects the SUB-20 Module to PC for interfacing. Also SUB-20 module gets power from PC through micro usb cable.

3) TESTING BOARD :

It consists 5 slots for mounting 3-PM(1V,1V,3.3V) and 2-APM(1.8V, 2.5V). It having 2 ADC's IC to monitor temperature, voltage and current.

4) FANS :

Two fans are mounted on the base board for cooling of PM/APM and resistors embedded into

the test board.

5) POWER SUPPLY :

Power Supply has been used for powering test board and PM/APM inserted into the test board. Output from power supply is : 12 V voltage, 10 A current.

6) RIBBON (34-pin Connector)

It connects test board to SUB-20 module, so that SUB-20 can talk to test board.

SOFTWARE INSTALLATION :

- Download Packages and Installation :
  svn co svn+ssh://USERNAME@svn.cern.ch/reps/cmshcos/trunk/hcalUHTR
  (Note : One need an ACTIVE CERN ACCOUNT AND PASSWORD TO DOWNLOAD THIS)

  cd hcalUHTR/tool/moduleCheckSUB20/
  ./install.sh -i
- After successful installation, It makes an executable file named "**HTR_PowerMezz_Test.exe"** into folder /**moduleCheckSUB20.** Now this executable can be run directly to test PM/APM with options (these options are mentioned below in testing procedure).

PRECAUTIONS taken before stating the test :

1. Before mounting OR unmounting PM/APM from test board MAKE SURE THAT POWER IS TURNED OFF.
2. MAKE SURE THAT ALL PINS OF PM/APM ARE INSERTED PROPERLY INSIDE THE OUNTING BOARD. IF THEY ARE DISPLACED THIS WILL LEAD TO BURNING OF THE PM/APM.
3. MAKE SURE AC IS ON ALL THE TIME .
4. MAKE SURE ATLEAST 2 FANS ARE ON DURING THE TEST.

TESTING PROCEDURE :

1. Make all the connections properly. Connect SUB-20 to test board through 34 pin connector, SUB-20 to PC through micro-usb cable and 12 V power supply to test board.
2. Insert PM/APM into the appropriately labeled slots on the test board.
3. Once all the PM/APM securely seated in their respective slots, the 12 V power supply should be enabled.

4. Now Run the executable file **HTR_PowerMezz_Test.exe** as given below:

4.1. Set path of libusb & libsub using:

source usblibcfg.sh (*)

*this .sh file is already exist into the **moduleCheckSUB20** folder.

Note : If you are not working as super-user(or root) then do the *"su and type the password of root to work as super-user".*

4.2. When testing PM/APM for the first time store some basic info inside the EEPROM of PM/APM as follows:

./uHTR_PowerMezz_Test.exe -a -p -S testing_site_name -N tester_name

4.3. To read information inside the PM/APM EEPROM(Electronic Erasable Programmable Read Only Memory) use:

./uHTR_PowerMezz_Test.exe -r

If the mezzanine hasn't been tested, It will show you the following information:

MAC: 00:04:a3:85:1a:c3
Data format version: 255
Mezz type code: 255
Mezz subtype code: 255
Mezz type (string):
Serial number: 255
Manufacture date:
Manufacture site:
Manufacture tester:
Test release:

4.4. To enable all the PM/APM use:

./uHTR_PowerMezz_Test.exe --enable

All LED on test board will glow.

4.5. To disable all the PM/APM use:

./uHTR_PowerMezz_Test.exe --disable

All LED on test board will stop glowing.

4.6. To run test for 1 min :

./uHTR_PowerMezz_Test.exe -t -U 1 -D 1 -T 1 -L 1

this will run the test for 1 min each of the test type : Margin Up, down, Nominal, High Load.

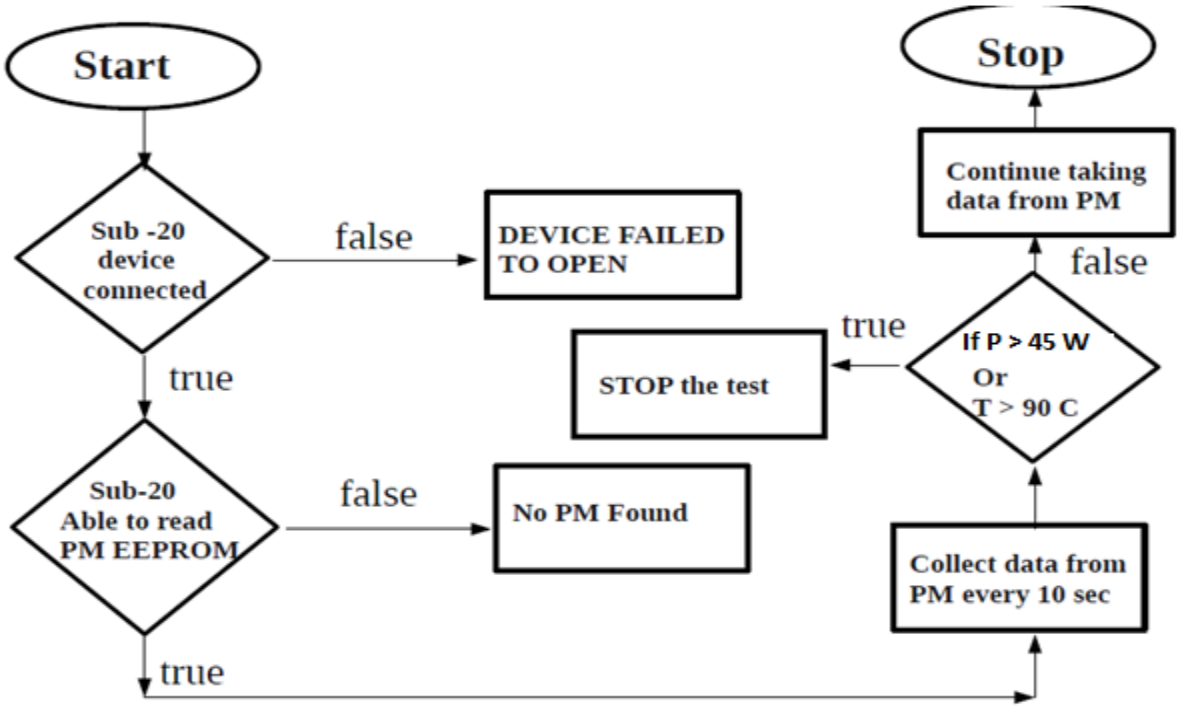
4.7. To run test FULL TEST which takes 0.5+0.5+19+19 hrs do

./uHTR_PowerMezz_Test.exe -t -U 30 -D 30 -T 1140 -L 1140

1140 mins = 19 hrs. Time should be enter in mins. also remember when using more than one option one should strictly follow the following option order:
order is = rapthe:dq:M:U:D:T:L:P:A:S:N:V

After the full test, it will generate a log file for each PM and APM named as PowerMezz-[mac-address] and AuxPowerMezz-[mac-address] respectively. A summary file for each PM and APM will also generate named as PowerMezzSummary.txt and AuxPowerMezzSummary.txt respectively.

FLOW CHART of TESTING PROCEDURE :



Reference : https://twiki.cern.ch/twiki/bin/view/Main/HCALUpgrade

NOTE: 1. Testing series of PM/APM in Delhi University will start from 2001.

2. Serial numbers should be recorded into the SerialNumberRecord file.

3. Put the main output file on Twiki.

Installation of testing software@DU in Ubuntu-12.04 through svn :

•Download package via command on terminal

svn co svn+ssh://USERNAME@svn.cern.ch/reps/cmshcos/trunk/hcalUHTR
(Note : One need an ACTIVE CERN ACCOUNT AND PASSWORD TO DOWNLOAD THIS)

•go to hcalUHTR/tool/moduleCheckSUB20/

•install via **./install.sh -i**

facing Error1
**********Screen 1**********
g++ -c uHTR_PowerMezz_Test.cpp -o uHTR_PowerMezz_Test.o -O3 -Wall -I. -I/home/kavita/hcalUHTR/tool/moduleCheckSUB20/libusb-1.0.9//include/libusb-1.0/ -I/home/kavita/hcalUHTR/tool/moduleCheckSUB20/SUB-20-snap-110319//lib -g -DURPI
In file included from uHTRMezzInterface.h:5:0,
from uHTRPowerMezzInterface.h:4,
from uHTR_PowerMezz_Test.cpp:9:
comInterface.h:43:33: **fatal error: gnublin-api/gnublin.h:** No such file or directory
compilation terminated.
make: *** [uHTR_PowerMezz_Test.o] Error 1
!!!!!!! INSTALL FAILED !!!!!!!

|=====================================|
| run the following once per terminal |
| "source usblibcfg.sh" |
|=====================================|

****
Error says **gnublin-api/nublin.h is missing**

4. Steps to resolve the error
•go to the link
http://gnublin.embedded-projects.net/downloads/

•dowunload->**source** from API/Tools
•extract gnublin-api-master.zip
•copy the required files: gnublin.h and gnublin.cpp
•go inside the folder /home/user/hcalUHTR/tool/moduleCheckSUB20
•make a new folder named “gnublin-api”
•paste those files from step 4 into the folder :gnublin-api”

•then install via command ./install.sh -i in the folder /home/user/hcalUHTR/tool/moduleCheckSUB20
•will face the **error2**

**********Screen 2**********

g++ -c uHTR_PowerMezz_Test.cpp -o uHTR_PowerMezz_Test.o -O3 -Wall -I. -I/home/kavita/hcalUHTR/tool/moduleCheckSUB20/libusb-1.0.9//include/libusb-1.0/ -I/home/kavita/hcalUHTR/tool/moduleCheckSUB20/SUB-20-snap-110319//lib -g -DURPI
uHTR_PowerMezz_Test.cpp:12:28: **fatal error: boost/thread.hpp: No such file or directory**
compilation terminated.
make: *** [uHTR_PowerMezz_Test.o] Error 1
!!!!!!! INSTALL FAILED !!!!!!!

|=====================================|
| run the following once per terminal |
| "source usblibcfg.sh" |
|=====================================|

****
Error says lib boost is missing, **need to install libboost1.48-all-dev**

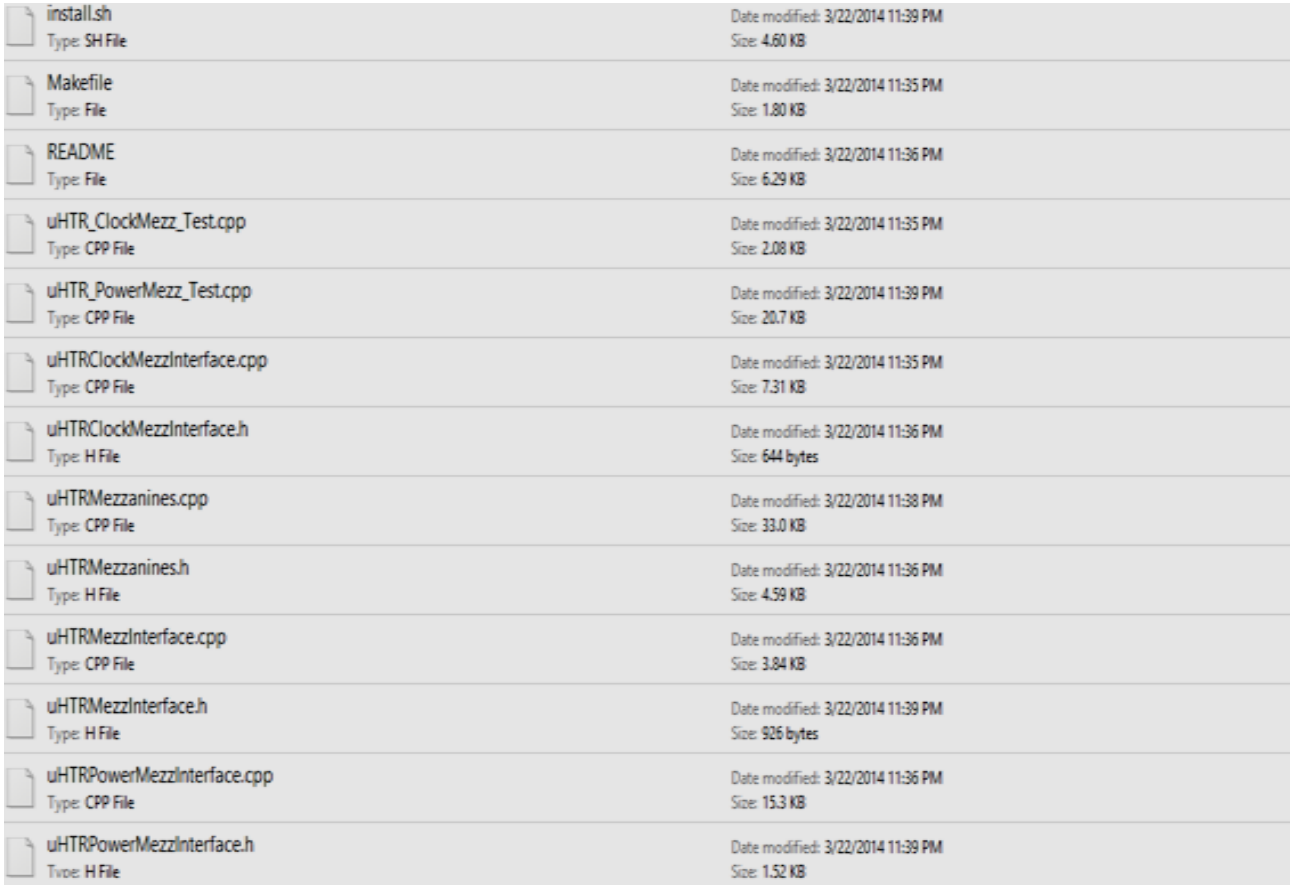
•**steps to resolve error2:**
1. go to terminal and type “sudo apt-get install libboost1.48-all-dev “
2.again run ./install -i in the same folder

You will see successfully compiled screen
**********(Screen 3)**********
g++ -c comInterface.cpp -o comInterface.o -O3 -Wall -I. -I/home/kavita/hcalUHTR/tool/moduleCheckSUB20/libusb-1.0.9//include/libusb-1.0/ -I/home/kavita/hcalUHTR/tool/moduleCheckSUB20/SUB-20-snap-110319//lib -g -DURPI
g++ -c -o gnublin.o gnublin-api/gnublin.cpp
gcc -O2 -I. uHTR_PowerMezz_Test.o uHTRMezzInterface.o uHTRPowerMezzInterface.o uHTRMezzanines.o comInterface.o gnublin.o -lboost_thread -lstdc++ -lm -o **uHTR_PowerMezz_Test.exe**

|=====================================|
| run the following once per terminal |
| "source usblibcfg.sh" |
|=====================================|
****

- While running the command ***"uHTR_PowerMezz_Test.exe -r"*** the executable generated was unable to interface and didn't detect the PM/APM **uHTR_PowerMezz_Test.exe.**

- So to troubleshoot this problem we copied all the source files from the folder **moduleCheckSUB20** brought from SINP Kolkata. After that we replaced them by all the files was present into the folder downloaded through ***svn.*** Then we ran the installation command ***./install.sh -i*** and software installed successfully. Also this software is working well. A snapshot of the folder containing the relevant source files is attached below.



install.sh	Type: SH File	Date modified: 3/22/2014 11:39 PM Size: 4.60 KB
Makefile	Type: File	Date modified: 3/22/2014 11:35 PM Size: 1.80 KB
README	Type: File	Date modified: 3/22/2014 11:36 PM Size: 6.29 KB
uHTR_ClockMezz_Test.cpp	Type: CPP File	Date modified: 3/22/2014 11:35 PM Size: 2.08 KB
uHTR_PowerMezz_Test.cpp	Type: CPP File	Date modified: 3/22/2014 11:39 PM Size: 20.7 KB
uHTRClockMezzInterface.cpp	Type: CPP File	Date modified: 3/22/2014 11:35 PM Size: 7.31 KB
uHTRClockMezzInterface.h	Type: H File	Date modified: 3/22/2014 11:36 PM Size: 644 bytes
uHTRMezzanines.cpp	Type: CPP File	Date modified: 3/22/2014 11:38 PM Size: 33.0 KB
uHTRMezzanines.h	Type: H File	Date modified: 3/22/2014 11:36 PM Size: 4.59 KB
uHTRMezzInterface.cpp	Type: CPP File	Date modified: 3/22/2014 11:36 PM Size: 3.84 KB
uHTRMezzInterface.h	Type: H File	Date modified: 3/22/2014 11:39 PM Size: 926 bytes
uHTRPowerMezzInterface.cpp	Type: CPP File	Date modified: 3/22/2014 11:36 PM Size: 15.3 KB
uHTRPowerMezzInterface.h	Type: H File	Date modified: 3/22/2014 11:39 PM Size: 1.52 KB