

The PMT-block Test Bench Developed in Pisa

D. Costanzo, T. Del Prete, E. Mazzoni, G.F. Pagani,
R. Paoletti and D. Rizzi

*Dipartimento di Fisica, Università di Pisa
and
Istituto Nazionale di Fisica Nucleare, Sezione di Pisa, Italy*



ATL-TILECAL-2000-012

11/08/2000

Abstract

A PMT-block test bench has been developed in Pisa. We have built and put into operation a system which allows a fast measurement of the main properties of a complete PMT-block (light mixer + PMT + divider + 3-in-1 card). In this report we describe in detail the system, both hardware and software, and the protocol used for the PMT-blocks characterization. We also report the results obtained with this bench in the test of a first batch of 80 PMT-blocks used to instrument the Module 0 for the 1999 test beam.

Contents

1	Introduction	2
2	Mechanics and Optics	4
3	Electronics	5
4	Protocol	7
4.1	Introduction	7
4.2	CIS	8
4.3	Fast Pulse	8
4.4	Slow Integrator	9
4.5	Data Analysis and Decision	10
4.6	Sequence of operations	10
5	Software	11
6	First Test	13
7	Conclusions	17
A	The DAQ software	18
	References	23

1 Introduction

The PMT-block [1] is the device used in the ATLAS Tile Calorimeter to convert light into electric signals. It is composed of a photomultiplier tube (Fig. 1), a light mixer (the interface between the PMT and the fiber bundle), a high voltage divider, and a 3-in-1 card which is the interface with the electronics read-out.

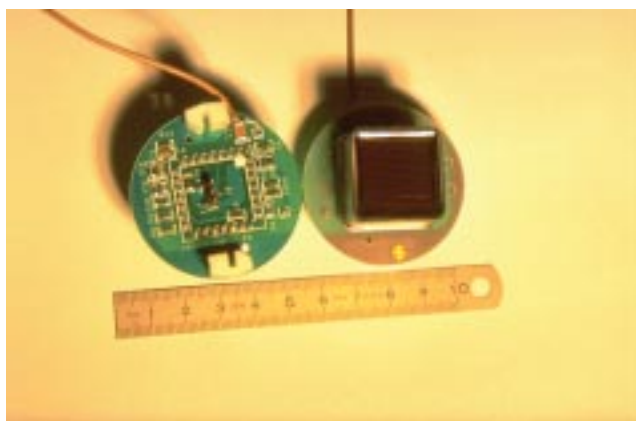


Figure 1: A picture of the Hamamatsu R7877 and its divider.

The PMT is assembled inside a soft iron and a mu-metal cylinders which provide adequate magnetic shielding. Fig. 2 shows the PMT-block assembled and the individual pieces.

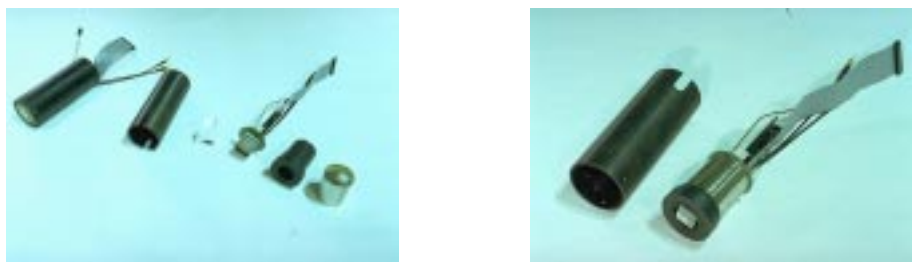


Figure 2: The PMT-block. a) at left the individual pieces are shown and b) at right the PMT-block is fully assembled.

The PMT-blocks will be housed in *super-drawers* which provide the mechanical support to PMTs and electronic boards, supply high voltages to the PMT dividers and low voltages to the 3-in-1 cards, convey control and calibration signals to 3-in-1 cards. Super-drawers also house the electronics to process signals from the 3-in-1 shaper and integrator.

All the components assembled in a PMT-block will undergo a Quality Control procedure; however it is necessary to check the functionality of the PMT-block as a whole, before it is inserted in a super-drawer. This test bench is intended to check each specific functionality of a PMT-block, while no attempt is made to characterize the individual components as specific test benches are dedicated to this task. The result of this test bench will be either *Full Functionality* or identification of the parts which are not working properly with a brief description of the problem.

The PMT-block test bench has been developed to be:

- *affordable* – Commercial electronics can be very expensive and should be avoided, when possible. On the other side a lot of the electronics used for the test-beams is no longer needed and can be recycled;
- *easy to use* – The system should be operated also by non expert operators, therefore the system must be very intuitive and easy to operate;
- *stable* – Several months of work will be needed to characterize all the TileCal PMT-blocks. Moreover, it is planned to have the test bench working during all the experiment life time to detect possible PMT-block failures. This implies that the test bench must operate in stable conditions for about 15 years;
- *portable* – After the initial development, the test bench will be moved and integrated in the framework of the final site where the actual test will take place. The final platform used for the DAQ system can be different from the one used at the development stage (see Tab. 1). We decided to write the DAQ in ANSI-C code to minimize the efforts in moving the system to another platform and to reduce the costs that another institution would bear if a commercial DAQ system (e.g. LabView) was used;
- *versatile* – The PMT-block test bench does not have a final protocol yet, so the system has to be easily modifiable and versatile enough to accomplish any change in the protocol itself. This characteristic turned out to be very useful since the same system is going to be used in Pisa as the DAQ system for PMT characterization.

Keeping this in mind, we split the problem into three different parts:

1. the hardware used in this test bench;
2. the protocol used for the characterization of the PMT-block test performed in 1999;

- the DAQ software, which is rather independent from this specific test bench.

In this note we describe in detail each item and we report on the first test done in summer 1999 using a prototype of the system.

The PMT-block test bench has now been moved to Athens University where the test of all the blocks used in TileCal will be carried out.

2 Mechanics and Optics

Fig. 3 shows a simplified scheme of the test bench. Up to eight PMT-blocks can be housed in special cradles inside a light tight box (Fig. 4 and Fig. 5). Blocks are connected to the High Voltage distribution, to signal and to control cables which are routed through the *mother board* as in the experiment. The connection between PMT-blocks and the mother board is provided by a special patch panel (Fig. 5).

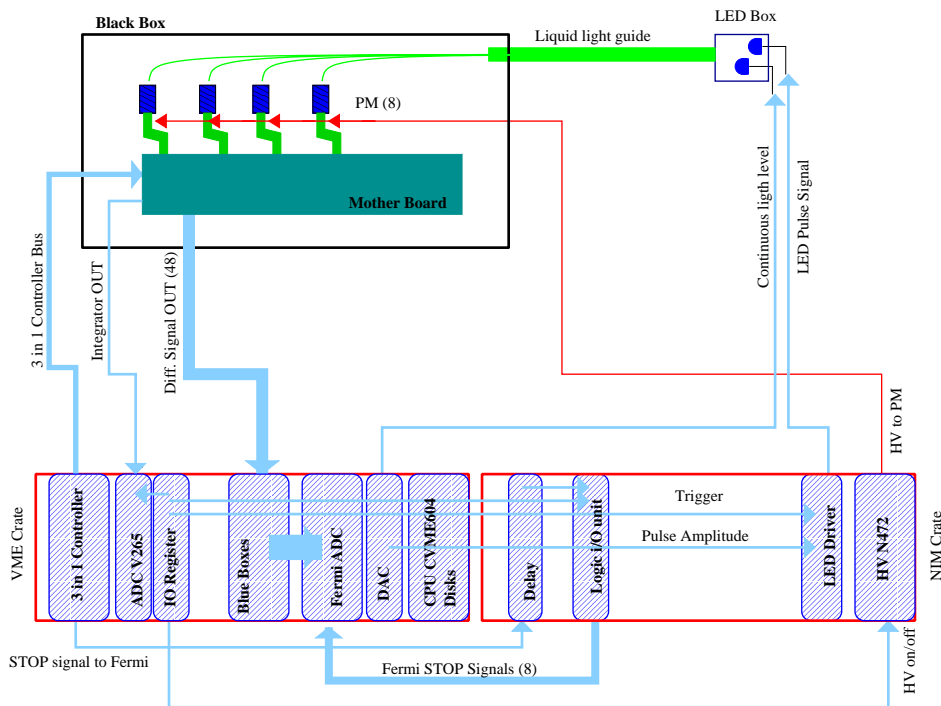


Figure 3: *Diagram of the Test Bench.*

The PMT's are lighted by two LEDs, one operated in DC and the second in pulsed mode. The LEDs are housed in a black plastic cylinder and face a



Figure 4: Overall view of the test bench. The black box, closed, at the right. At the left the electronics for the DAQ.

liquid fiber (Fig. 6) which is connected, inside the light box, to a fiber bundle in order to distribute the light to each PMT (Fig. 6). The amount of light that each PMT receives has been roughly equalized among the eight PMTs using optical filters.

3 Electronics

The test bench comprises control and readout electronics located in a NIM and a VME crate. The sequence of operations is controlled by a Cestia CVME604 CPU located in the VME crate. Fig. 3 shows a simplified scheme of the test bench:

- A VME I/O register (CAEN V262) is used to switch ON and OFF the PMTs high voltages and to trigger the LED (pulse mode). To avoid the use of a programmable HV system we decided to use the same high voltage (800 V) for all the channels;

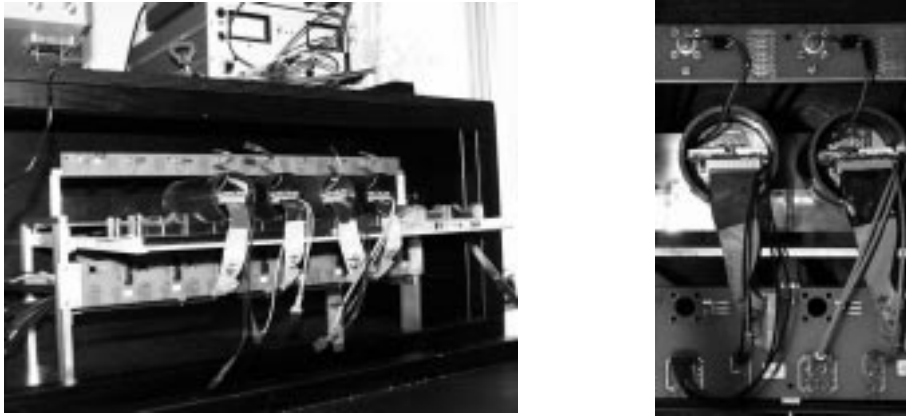


Figure 5: *The picture on the left shows the PMT-blocks mounted inside the black box. The picture on the right shows a detail of the connection of PMT-blocks to two intermediate boards used for distributing signals, HV and controls.*

- The pulsed LED driver is a NIM home-made circuit, which produces a fast (20-30 ns) signal whose amplitude (0, +60 V) is modulated by a DC level. Two home made VME Digital to Analog converters are used to control the pulsed LED light and to switch ON the DC operated LED to the desired light intensity;
- A special VME unit (the 3-in-1 controller) – developed by the Chicago group – is used to set the mode of operation of the 3-in-1 cards and to provide the charge injection for linearity checks;
- The fast output signals are bipolar and need to be converted to single ended signals before they can be sent to the ADC. This is provided by *blue boxes* (home-made circuit by Stockohlm used in test beam data tacking) whose output feeds eight FERMI ADC (CAEN V571). The STOP signal to the FERMI is generated by the VME I/O unit;
- The signals from the *integrator* is routed to ADC (CAEN V265) with a $1 \mu\text{s}$ gate signal generated by the VME I/O unit.

The whole DAQ system developed in Pisa is presented in Fig. 7, a complete list of the electronics and hardware used in the test bench is presented in Tab. 1.

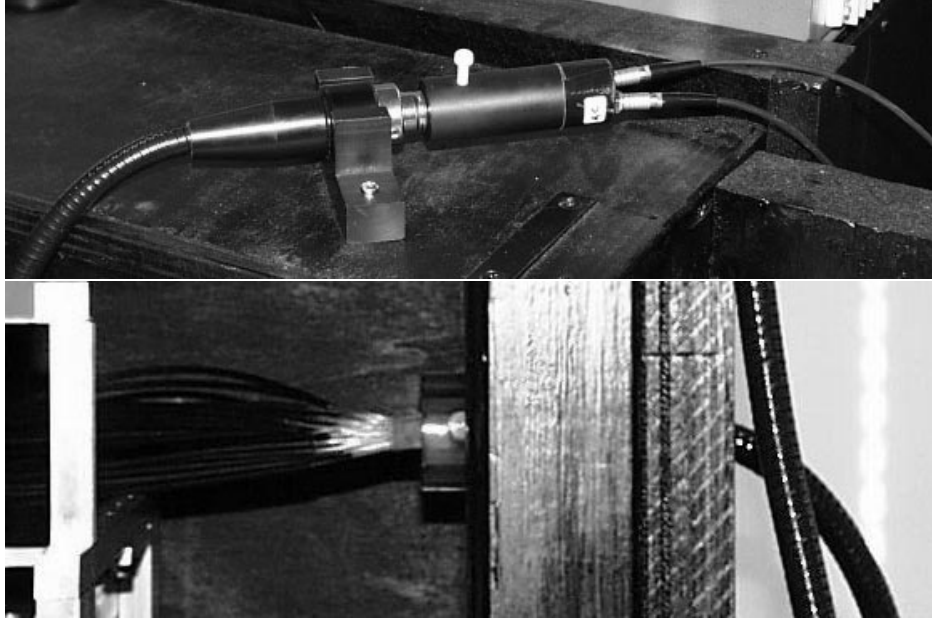


Figure 6: *The picture on the bottom shows a detail of the coupling of the liquid fiber to the fiber bundle. The fiber bundle is made of 32 clear fibers which are distributed to each PMT. The picture on the top shows a detail of the light source system. Two LEDs, one operated in DC and the other in pulsed mode are mounted inside a black cylinder and coupled to a liquid fiber to take the light into the black box.*

4 Protocol

4.1 Introduction

The PMT-block is an assembly of three main parts: a light mixer, a PMT with its divider and a 3-in-1 card.

The 3-in-1 cards form the front end of the electronics chain. They provide 3 basic functions: a fast pulse shaping of the PMT output, a slow integrator circuit for measuring the current from the Cesium calibration and the minimum bias events, and a charge injection calibration system. These three functions are controlled by an on-board logic driven by a remote controller. The aim of this test bench is to check that all these functions are properly working.

We divide the test in three parts accordingly to the PMT-block working mode: charge injection (CIS), data taking operation (fast pulse) and calibration data taking (slow integrator). For all these parts a test procedure has

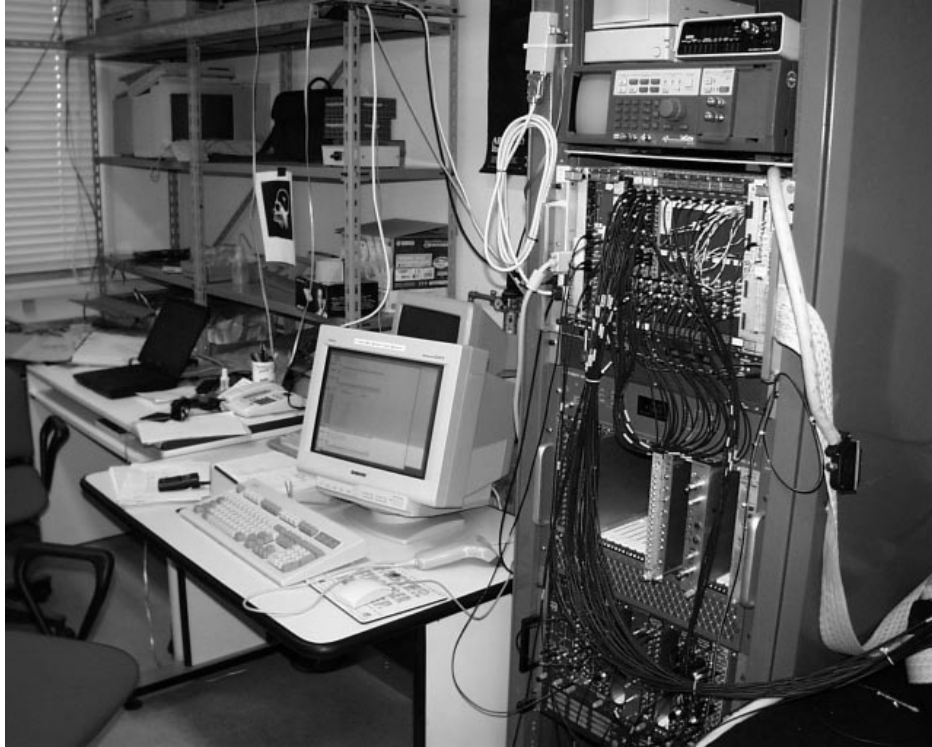


Figure 7: Overall view of the DAQ system.

been implemented.

4.2 CIS

During the Charge Injection System test the high voltage and light system are OFF and the 3-in-1 card is set in the charge injection configuration. Different charges have been injected and the outputs from the low, high gain and trigger channels are recorded. At the end of the test linearity plots (ADC vs. injected charge) are fit with a straight line. The gain of each channel (the slope of straight line) is computed and stored on disk.

4.3 Fast Pulse

During this part of the test the high voltage is ON and light system is operated in pulsed mode. In this case we mimic the real experimental situation by flushing the PMTs with a fast blue LED. Three different light levels are used to simulate: a) the signals released in the calorimeter by a minimum ionizing

Component	Model	Qt	Available from
Liquid guide		1	Pisa
Black box		1	Pisa
Light case		1	Pisa
HV patch panel		1	Pisa
Signal patch panel		1	Pisa
Crate VME		1	
Crate NIM		1	
CPU	Cetia CVME 604	1	
I/O register	CAEN V262	1	
ADC q-int.	CAEN V265	1	
diff./se adapters		3	CERN
Fermi ADC	CAEN V571	8	CERN
clock fan out		1	Chicago
3in1 ctrl		1	Chicago
DAC		2	Pisa
LED driver		1	Pisa
NIM/TTL adapter		1	Pisa
Fan IN Fan OUT		1	Pisa
HV system 4 ch	CAEN N472	1	
LV power supply		3	
Cables		100	Pisa
1/4 mother board		1	Chicago

Table 1: *List of the hardware components of the test bench.*

particle, b) a signal falling in the overlap region of the two gain channels, and c) a 1 TeV jet. Around 5000 events for each PMT-block are recorded. The program performs the pulse recognition and pedestal subtraction on-line during the train of pulses recorded by the V571 flash ADC. Two histograms for each PMT are filled: one for the low gain and the other for the high gain. The average value of each histogram and its r.m.s. are written on disk.

4.4 Slow Integrator

During the Slow Integrator test the high voltage is ON and the light system is operated in DC mode. Three different levels of DC light are used to simulate: a) pile-up events, b) signal from Cs source and c) a level of light in between. The output current for each level is measured and stored on disk.

4.5 Data Analysis and Decision

The data analysis is performed on-line at the same time as the data taking and, at the end of the test, the PMT-block relevant information is available in an ASCII file.

The aim of this test is to check the functionality of the PMT-blocks without performing their full characterization. The measured values are checked to be inside a confidence range and the decision to accept/reject each block is taken. This operation is performed on-line by the DAQ system and a screen message indicates to the operator which PMT-blocks are to be rejected. The confidence ranges used are summarized in Tab. 2.

	Low Gain	Hi Gain	Trigger Sum ¹
CIS (gain)	0.8–1.2	52–78	–
Fast Pulse (average)	> 250	> 150	–
Slow int.	> 10 + increasing path		

Table 2: *Confidence ranges used for the PMT-block acceptance.*

At the end of each measurement the ASCII file is converted to an EXCEL one to perform the statistics and the database functions.

4.6 Sequence of operations

The sequence of operations for the test of a PMT-block batch is the following:

1. Put 8 PMT-block into the black box (manual);
2. Connect the PMT-block cables to the patch panel (manual);
3. Collect the PMT-block information and initialize the database. This part has been performed manually, however it will be automatic when each block will be labeled with a bar-code;
4. Perform the CIS test and store the results on disk (automatic);
5. Perform the fast pulse test and store the results on disk (automatic);
6. Perform the slow integrator test and store the results on disk (automatic);

¹In the summer 1999 test we do not test this part of the PMT-block due to missing electronics.

7. Check the results and decide to accept/reject the PMT-block (automatic);
8. Remove the PMT-blocks and put aside those to be rejected (manual).

A complete test of a batch of 8 PMT-blocks take about 10 minutes from step 1 to 8.

5 Software

The guidelines followed in the development of this test bench have been stated in the introduction. The software has to fulfill all these requirements i.e. it has to be affordable, easy to use, stable, portable and versatile.

To fulfill all these requirements we developed the software with a layer philosophy. The operations to be performed on the VME data bus are two: write and read data from the bus. A given board is interfaced by a small library which supplies the basic set of routines to operate the board itself.

The layer structure of our software is the following:

- *kernel* – This is the part of the code that interfaces to the menu, data and file handling and deals with error management. It takes care of the VME bus initialization and read/write functions. This is the only platform dependent part of the code and must be written accordingly to the hardware used to interface the VME bus;
- *libraries* – Each VME board has its own library of functions that are responsible for initialization, control and data acquisition. If a new board is added a new library has to be written. The debugging is easy as only the basic functions have to be checked while more complex functions can be obtained by suitable scripts. At this level the software is platform independent and can be moved from one system to another without changing the code if the system supplies an ANSI-C compliant compiler;
- *scripts* – This is the part dealing with the implementation of a specific test. The user takes advantage of the hardware functions without worrying how operations are performed and has only to concentrate on the measurements. This part is, of course, platform independent.

The software has been developed on a Cestia PowerPC board (CVME 604) with LynxOS 2.4 operating system. An ANSI-C compiler is supplied.

Every hardware board is activated by a set of commands available from a sub menu in the DAQ program. These routines are used for simple operations like to set the number of boards present in the crate, set the VME address to access them and the basic read/write functions.

The DAQ system is essentially a command line driven program, with simple operations available from menus and sub menus that perform specific actions. Multiple commands can be written on the same line and the program takes care of splitting them in individual items. These commands can be grouped in macros that the program is able to execute. Macros can be called and nested and complicated sequences of commands are easily obtained thus providing a very high level of flexibility. Any kind of test bench can be implemented without too much trouble provided the right macros are written.

To handle the data coming from the VME boards we have included in the system a histogram handling package à la HBOOK [2] and a data handling package à la SIGMA [3] to perform simple mathematical operations and a file handling package. A preliminary on-line data analysis can be performed and data can be written on disk for an accurate data analysis to be done off-line.

The software has been checked on several platforms. Table 3 summarizes the code status. In the Appendix we show an example of the scripts used in the test.

Platform	Op. Sys.	Compiled	Running
PowerPC	LynxOS	x	x
PowerPC	MacOS	x	
Motorola 68xxx	OS/9	x	x
Motorola 68xxx	MacOS	x	x
Risc	AIX	x	
Alpha	OpenVMS	x	
Intel	Linux	x	

Table 3: *Platforms and operating systems on which the software has been tested, compiled and running. Not on every platform/OS the software can be used simply because it not possible to access the VME bus, however the same code can be debugged by using appropriate simulation routines.*

6 First Test

In summer 1999 we received in Pisa 80 PMT-blocks to be tested for complete functionality. At that time the test bench was not yet complete and only four PMT-blocks could be tested at the same time. Part of the electronics was still missing, in particular some of the *Blue Boxes* and half of the FERMI. It was nevertheless an unique occasion to test the whole procedure, check the time needed to test each PMT-block, test the reliability of the software and its flexibility to cope with unexpected needs, and finally setup a skeleton of the database.

The experience was very interesting. Of course many details of the test were not foreseen and we changed (as usual at the very last moment) several times the protocol of the test both to improve the interface with the operator and to change some of the test procedure.

The basic libraries of the DAQ (that part specific to each VME modules) were fully debugged in advance and were not modified at this stage. We had just to change many times the scripts. However, since they needed not to be compiled, the setup of the macros was extremely fast and friendly.

Fig. 8 shows the EXCEL sheet summarizing the measurements. Each row contains the characterization of a PMT block. The first two columns are the slope and the intercept of a linear fit to the CIS test in low gain. The data in the second column should be around 1. The intercept is less important for the functionality of the 3-in-1 card. One should worry only if, as for *lab051*, it is too large. The same is true for the high gain test. Here the slope should be around 65. Fig. 9 shows the distributions of the slopes for low and high gain.

The *Pulse Low* and *Pulse High* columns show the mean and the r.m.s. of the distributions. Here the HV is on and the PMT is excited by a flash of light. Since no optimization was made to equalize the PMT gain and the amount of light in each channel, we should expect that the PMT-blocks response would depend on their position on the bench. In some cases the PMT is saturated and this can be identified by the large mean and small r.m.s.. Fig. 10 shows the distribution of the means.

Finally the integrator output is measured for several levels of light (last three columns). Fig. 11 shows their distributions.

Label	Low Gain	High Gain	Pulse Low	Pulse High	Trigger	Integrator							
lab001	1.12	6.09	63.55	30.21	610.76	28.64	924.05	75.45	21.76	1.24	48.00	79.00	192.00
lab002	0.96	5.38	68.34	34.12	982.73	0.57	982.43	0.43	44.31	2.29	122.00	162.00	295.00
lab003	1.09	5.37	61.31	81.03	431.78	20.48	706.88	94.19	15.68	1.24	39.00	44.00	124.00
lab004	1.13	6.36	66.96	33.30	461.86	20.26	747.75	99.97	16.76	1.40	48.00	57.00	201.00
lab005	1.10	5.45	64.77	30.88	644.95	29.92	932.91	69.99	22.90	1.21	56.00	60.00	225.00
lab006	0.89	38.87	67.60	29.91	564.67	27.66	963.07	40.83	22.26	1.75	71.00	88.00	294.00
lab007	1.10	5.80	65.78	30.01	170.92	8.94	269.17	35.76	6.24	0.90	31.00	32.00	56.00
lab008	1.10	5.89	63.75	29.95	560.28	27.62	846.17	144.39	54.89	170.28	45.00	69.00	217.00
lab009	1.11	5.69	64.81	28.88	535.20	25.29	893.63	74.00	18.75	1.36	37.00	64.00	161.00
lab010	0.97	5.15	70.12	39.29	982.81	0.52	982.21	0.81	52.05	1.96	104.00	173.00	294.00
lab011	1.10	5.82	65.94	30.85	440.34	18.45	753.20	92.03	15.77	1.28	41.00	46.00	110.00
lab012	1.08	38.68	66.69	32.59	505.42	21.09	857.51	80.38	18.46	1.86	55.00	70.00	185.00
lab013	1.13	6.16	65.70	30.15	737.54	38.05	970.29	34.13	25.69	1.39	51.00	99.00	223.00
lab014	0.97	5.99	69.45	32.08	968.12	24.41	982.43	0.73	38.37	2.35	102.00	148.00	291.00
lab015	1.10	5.53	65.81	31.90	329.78	16.48	508.67	57.93	11.87	1.17	44.00	42.00	94.00
lab016	1.10	6.60	66.54	31.61	488.95	74.51	725.90	74.67	17.98	1.80	56.00	89.00	220.00
lab017	1.13	5.80	67.02	29.35	353.33	17.03	557.15	69.77	12.52	1.05	44.00	60.00	141.00
lab018	0.98	5.41	69.07	32.51	982.77	0.00	982.53	0.75	52.75	1.84	135.00	216.00	294.00
lab019	1.16	7.00	69.63	35.24	226.42	11.91	355.23	44.15	8.29	1.21	22.00	43.00	84.00
lab020	1.12	6.03	66.05	30.50	419.28	22.11	660.42	68.12	15.36	1.42	60.00	63.00	191.00
lab021	1.14	5.80	66.68	30.43	981.08	0.66	982.62	0.54	40.52	2.67	83.00	129.00	300.00
lab022	0.96	5.60	68.10	32.55	478.26	25.74	883.69	82.23	17.96	1.27	56.00	90.00	193.00
lab023	1.10	5.64	68.00	31.14	207.09	10.67	335.48	40.01	7.38	1.08	32.00	42.00	77.00
lab024	1.13	6.31	66.47	30.53	981.36	0.19	980.89	1.60	40.38	2.25	105.00	126.00	299.00
lab025	1.06	5.22	60.39	28.98	648.09	31.66	952.66	54.76	23.48	1.57	47.00	88.00	194.00
lab026	0.95	5.51	67.22	31.40	982.71	0.49	982.30	0.79	45.58	2.35	125.00	150.00	288.00
lab027	1.10	6.45	65.49	31.53	456.04	77.94	723.95	77.50	15.94	1.12	45.00	62.00	136.00
lab028	1.10	5.89	65.81	30.70	654.71	30.80	961.31	45.69	23.39	1.54	69.00	87.00	293.00
lab028	1.16	6.54	67.37	31.62	642.83	29.61	949.04	61.90	22.76	1.54	71.00	94.00	197.00
lab029	0.94	5.61	68.75	31.84	808.24	39.81	982.41	0.59	29.34	1.88	91.00	134.00	290.00
lab030	1.12	6.33	63.96	81.81	215.04	11.24	341.48	45.93	7.76	1.17	87.00	59.00	97.00
lab031	1.13	6.72	67.23	30.46	824.02	41.21	981.34	0.78	29.76	1.71	76.00	111.00	292.00
lab032	1.10	5.56	62.08	28.01	439.80	20.44	688.70	62.48	15.62	1.24	63.00	93.00	175.00
lab033	0.94	5.27	68.33	31.28	353.43	14.91	660.34	67.27	12.48	1.21	52.00	64.00	194.00
lab034	1.09	5.54	66.53	30.77	180.09	7.24	299.18	40.56	6.30	0.96	35.00	72.00	73.00
lab035	1.10	19.10	65.92	29.58	446.12	20.22	716.81	77.08	35.00	135.27	70.00	76.00	204.00
lab036	1.09	5.57	63.37	28.81	500.76	21.78	770.51	93.33	17.96	1.29	40.00	50.00	175.00
lab037	0.91	4.71	65.94	31.04	566.31	23.10	969.41	35.98	21.28	1.48	63.00	98.00	287.00
lab038	1.09	6.04	64.73	29.25	301.48	12.73	482.54	59.89	2.16	0.67	49.00	44.00	89.00
lab039	1.11	5.89	64.89	30.08	308.44	12.07	484.80	54.79	11.06	1.34	58.00	56.00	174.00
lab040	1.08	5.95	61.74	29.66	541.55	28.20	790.72	85.35	19.44	1.46	55.00	74.00	175.00
lab041	0.93	5.22	66.75	31.10	405.78	22.53	710.18	79.93	15.05	1.07	32.00	80.00	182.00
lab042	1.08	6.14	65.27	31.57	504.88	27.38	765.36	93.67	17.98	1.26	40.00	55.00	153.00
lab043	1.11	6.20	66.63	31.99	519.00	70.02	744.28	93.24	18.37	1.53	49.00	61.00	190.00
lab044	1.07	5.62	61.42	27.79	252.32	13.10	365.32	55.36	9.21	0.89	30.00	54.00	96.00
lab045	0.94	5.56	67.35	31.19	426.45	20.16	754.37	69.84	16.08	1.24	49.00	71.00	199.00
lab046	1.09	6.17	64.38	31.80	198.48	9.67	305.00	101.70	7.04	0.93	85.00	18.00	85.00
lab047	1.09	6.31	65.97	30.59	281.20	13.33	419.58	43.20	10.37	1.19	45.00	49.00	143.00
lab048	1.08	5.60	63.79	29.83	598.52	32.68	844.54	106.30	21.58	1.45	53.00	67.00	206.00
lab049	0.91	5.55	64.94	30.16	543.48	28.36	886.05	72.57	20.83	1.47	54.00	95.00	292.00
lab050	1.07	6.32	66.90	32.46	486.05	23.38	710.03	97.47	17.30	1.31	52.00	60.00	143.00
lab051	1.00	52.15	63.55	30.74	346.92	17.11	483.60	56.09	12.62	1.14	65.00	77.00	249.00
lab052	1.06	5.35	60.01	28.17	368.20	17.98	531.56	76.88	13.35	0.91	44.00	62.00	134.00
lab053	0.91	19.20	66.25	30.61	920.61	48.87	982.49	0.19	34.04	1.62	82.00	144.00	289.00
lab054	1.09	6.59	66.69	32.88	409.09	21.56	594.38	69.56	14.54	1.05	12.00	57.00	131.00
lab055	1.08	19.44	65.25	30.98	434.80	20.40	632.25	101.19	15.64	1.30	52.00	70.00	219.00
lab056	1.08	4.89	60.34	28.47	332.39	14.11	426.52	48.25	11.92	0.98	29.00	60.00	139.00
lab057	0.92	4.40	65.33	30.06	621.36	28.61	939.68	59.50	23.86	1.92	60.00	116.00	292.00
lab058	1.07	4.86	64.67	30.07	359.77	89.96	457.49	62.67	12.54	1.07	29.00	56.00	113.00
lab059	1.07	5.12	63.57	28.76	189.93	8.79	247.38	29.46	6.86	1.23	31.00	64.00	102.00
lab060	1.04	4.65	59.49	32.82	568.50	27.87	758.17	95.64	20.22	1.12	50.00	98.00	223.00
lab061	0.93	5.05	65.81	30.44	982.60	0.29	982.19	0.47	49.08	1.95	181.00	197.00	306.00
lab062	1.04	5.13	63.36	28.22	423.19	18.93	591.96	95.72	14.90	1.31	49.00	84.00	117.00
lab063	1.07	5.34	63.89	28.53	433.61	19.56	594.73	67.61	34.72	135.28	53.00	89.00	210.00

Figure 8: EXCEL summary of the measurement. Each line is a PMT-block.

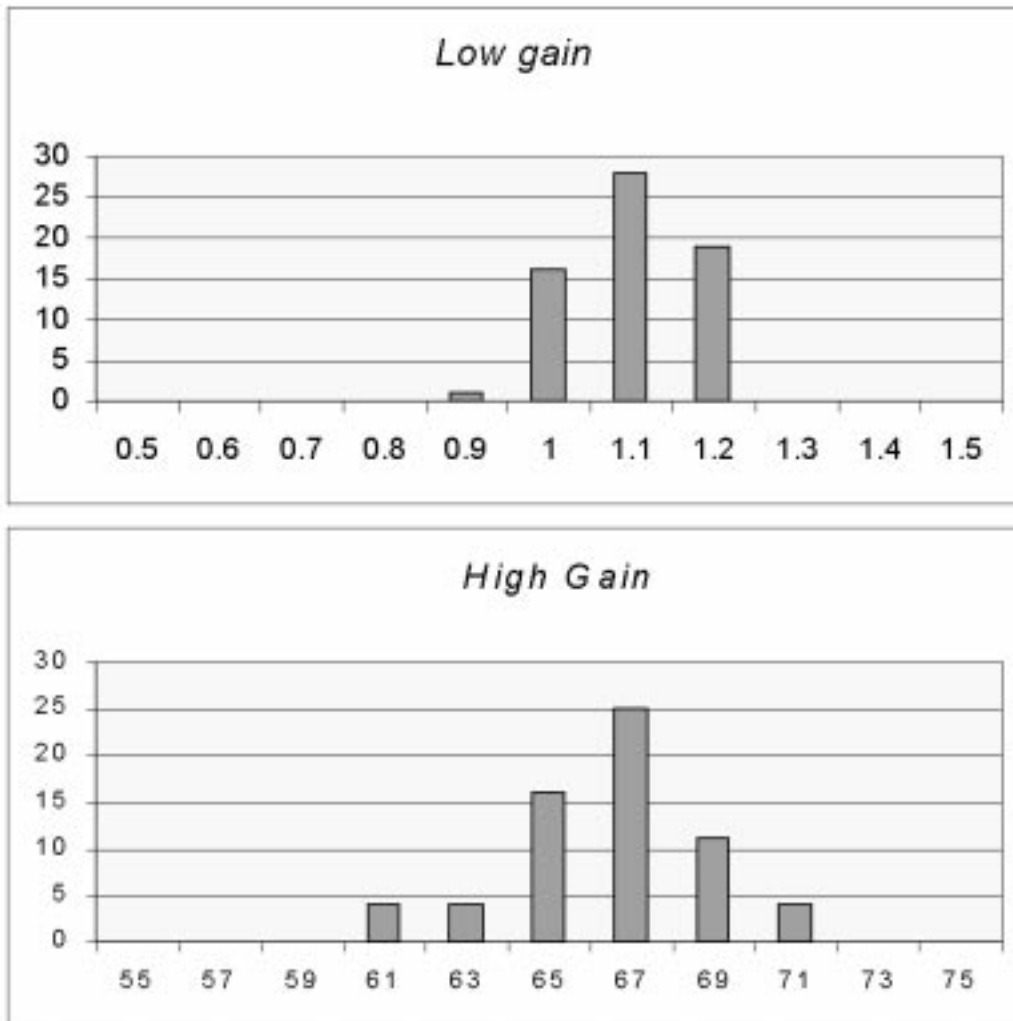


Figure 9: *Distribution of the low and high gain for the CIS test.*

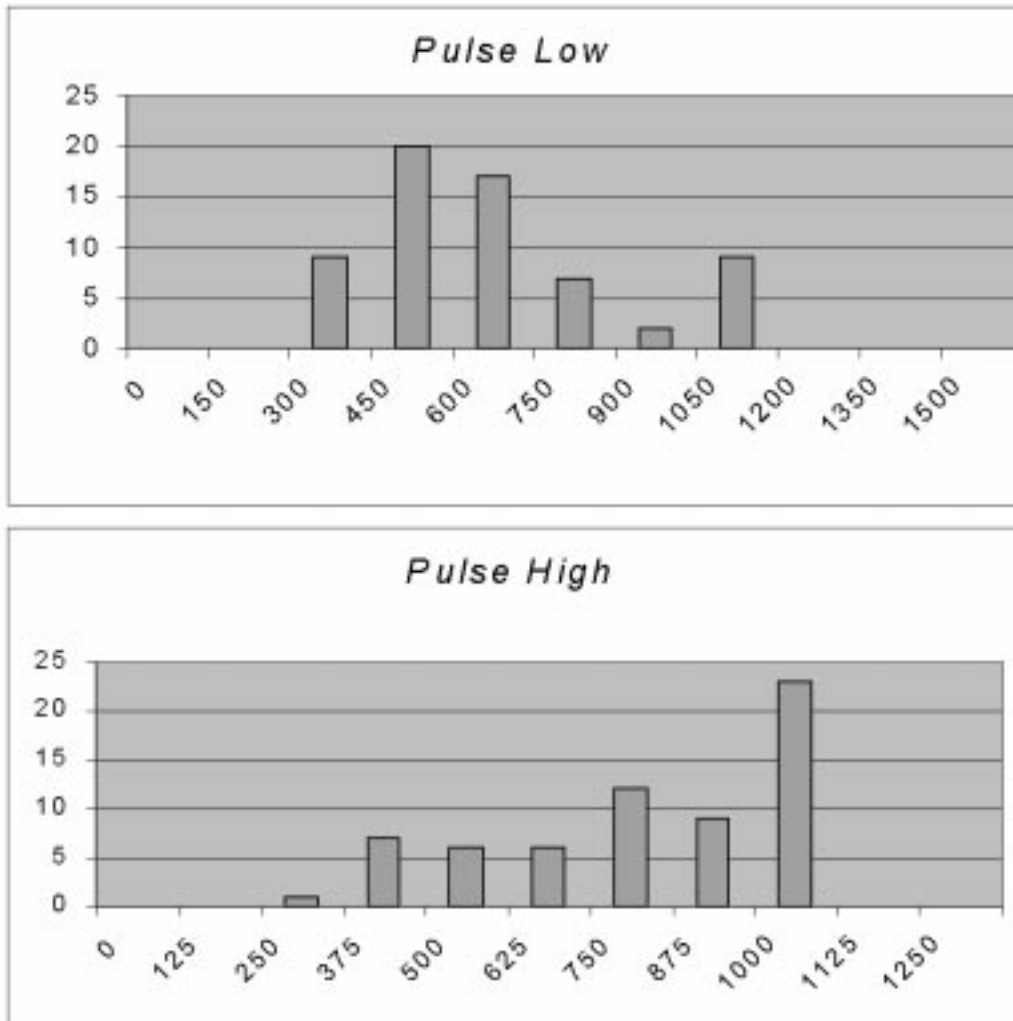


Figure 10: *Distribution of the PMT-block outputs in pulsed mode for the low and high intensity pulses.*

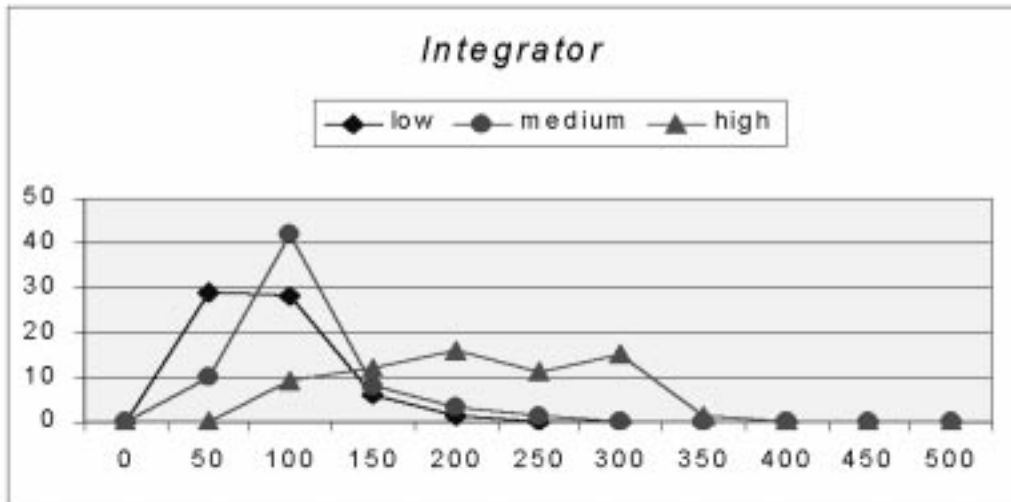


Figure 11: *Distribution of charge from the integrator for the three different levels of light.*

7 Conclusions

A test bench for the PMT-blocks to be used in TileCal has been developed in Pisa. With this test the functionality of all the PMT-blocks will be checked before the final insertion inside the super-drawers. Eventual assembly/electronics problems can be spotted in a fast way. Eighty PMT-blocks have been tested in summer 1999 and a first protocol has been defined.

The whole DAQ is made of basic routines, specific of each electronic instrument housed in the VME crate and a structure which permits high level calls from non compiled code. The whole system is nicely designed in such a way that scripts can call each other or be called manually by the operator. Facility exists for linear fits to histograms, histogram manipulation, mathematical operations, as in *SIGMA* in *PAW*.

This software has been successfully tested in the setup of the PMT-block test bench and it will implement the DAQ in the more demanding tests and characterization of photomultipliers which will start in September 2000.

A The DAQ software

In this section we show some examples of how the scripts used in the Data Acquisition for these tests look like.

The scripts are ordered in a hierarchy. The top level scripts organize the sequence of the different tests. The first script is the *test.uic*:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
! Macro Name: test.uic !
!!
! Last Modified: May 30th 2000 !
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
! CALLED BY: user input !
!!
! CALLS: test_cis.uic !
! test_pulse.uic !
! test_int.uic !
! test_merge.uic !
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
! This is the macro starting the PMT block test !
! !
! Run this one with the command @test.uic in vmetest !
! to perform a complete test. !
! Every newly implemented test part should be called !
! from here. !
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
message "*****\n"
message "* The PMT Block Test Bench * \n"
message "* Version 1.0 May 30th 2000 * \n"
message "*****\n"
!
! CIS Test
!
@test_cis.uic
!
! Pulsed Test
!
@test_pulse.uic
!
! Integrator Test
!
@test_int.uic
!
! Summarize results on test.dat
!
@test_merge.uic
message "Test is Finished\n"
message "RENAME the output file\n"
```

The script consists of several calls to other scripts. The first is the script *test_cis.uic* which is listed below:

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!   Macro Name: test_cis.uic
!
!   Last Modified: May 30th 2000
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!   CALLED BY: test.uic
!
!   CALLS:   cis_fast.uic
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!   This is the macro for Charge Injection calibration
! !
!   The real work is done in cis_fast.uic
!   This macro is only the final interface.
! !
!   call @test_cis.uic from vmetest to perform only
!   CIS test (for debugging purposes)
! !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

message "*****\n"
message "* CIS calibrations *\n"
message "*****\n"
!
!   Call to cis_fast.uic
!
@cis_fast.uic
!
! Print the results obtained for online monitoring
!
message @gain_low1.dat message "\n"
message @gain_low2.dat message "\n"
message @gain_low3.dat message "\n"
message @gain_low4.dat message "\n"
message @gain_low5.dat message "\n"
message @gain_low6.dat message "\n"
message @gain_low7.dat message "\n"
message @gain_low8.dat message "\n"
message @gain_high1.dat message "\n"
message @gain_high2.dat message "\n"
message @gain_high3.dat message "\n"
message @gain_high4.dat message "\n"
message @gain_high5.dat message "\n"
message @gain_high6.dat message "\n"
message @gain_high7.dat message "\n"
message @gain_high8.dat message "\n"

```

which, in turns calls *cis_fast.uic*:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!   Macro Name: cis_fast.uic
!
!   Last Modified: May 31st 2000
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!   CALLED BY: test_cis.uic
!
!   CALLS: cis_hist_book.uic
!           cis_init2.uic
! cis_hist_reset.uic
! cis_event_low.uic
! cis_hist_fill_low.uic
! cis_stack.uic
! cis_init1.uic
! cis_event_high.uic
! cis_hist_fill_high.uic
!
!   VARIABLES: X0 --> injected charge
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!   This is the macro for Charge Injection calibration
!   Version using the fast Fermi reading to save cpu
!   time.
!!
!   COMMENT: The fast fermi reading returns as a result
!   the peak value subtracted by the average read
!   of the first 5 fermi samplings.
!!
!   The real cis work is done here and divided into
!   submacros each doing simple actions.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Book histos used for the cis test
!
@cis_hist_book.uic
!
! Initialize 3 in 1 controller for charge injection
! Each of the 8 tubes that can be connected to the mother board
! in the black box are initialized by a call to cis_init2.uic
!

3in1
    @cis_init2.uic
    zone 0 sector 0 tube 47 address
    . . . . .

return
!
! Reset and initialize Fermi modules.
!

Fermi
select 1 reset sample_write 30 delay_write 0
. . . . .
return
!
! Low gain CIS calibration: Injected charged is 0, 200, 400, 600, 800
```

```

!
! Comments on the 0 case for low gain. For the other cases refer to them
!
message 0\n
wait 1
sigma < 0 > X0 ret
!
! Acquire CIS events. Loop 10 times to reduce stat error. This can be
! changed, but 10 seems ok.
!
loop 10 @cis_event_low.uic
!
! The mean value obtained from the 10 loops is stored in another histo
! (1111 for pmt 1) as a function of the injected value. A linear fit will
! be performed on this histo to evaluate the gain.
!
@cis_hist_fill_low.uic
!
! Other Change injected values:
!
.....
!
! Put results into files. One for each pmt block.
! As before comments for the first tube.
!
!
! open output file
!
sigma
fopen gain_low1.dat
!
! write tube number
!
      fwrite "1\t"
!
! 1.249 is used to convert the value to a gain using the low blue module
!
      < 1.249 > x6 pull
!
! Linear fit to the histo storing the mean values vs. charge injected
!
fit 1111 @cis_stack.uic fclose
!
! same as before for the other tubes
!

return

!
! High gain calibration. Everything is as before. Refer to the low gain
! case
!

```

This last script consists of several calls to ancillary scripts to manage histograms. The most relevant call is the one to *FERMI*. This routine performs basic I/O operation on the VME bus. This routine is coded in C and, part of it is listed below:

```
/*
```

```

* This command performs a read to the Fermi V571 samples
* and returns the difference between the maximum sampling value
* and the mean of the first 5 samplings
*
* This is a replacement of the READ command to speed up the test
*
*/

if(strcmp(verb,"FASTREAD")==0) /* User command is FASTREAD */
{
/*
* Get the qualifiers associated with the FASTREAD command
* Debug : 1 Debug is on / 0 Debug is off
* Channel: number of channel to be read (1,2,3)
*/

    if((status = get_qualifier("PMT_Fermi_Menu", "FASTREAD", "CHANNEL",&nch))==0){}
    if((status = get_qualifier("PMT_Fermi_Menu", "FASTREAD", "DEBUG", &deb))==0){}
/*
* arr[256] is used to store the Fermi samplings information. It is first
* set to zero and then filled up by a call to Fermi_wait_for_trigger
*/

    uzero(arr,256);
    err=Fermi_wait_for_trigger( nch, arr);
/*
* Detect eventual errors
*/
    if(err>0)
    {
        error_logger( "Fermi_wait_for_trigger", ERWARN, "Timeout error");
        return;
    }
    else
    {
/*
* Debug if it is requested
*/
        if(deb>0) printf("Fermi %ld Channel=%ld\n", this_fermi, nch);
/*
* average is the average of the first 5 samplings used as a pedestal
*/
        average=(float)arr[0]+arr[1]+arr[2]+arr[3]+arr[4];
        average /= 5.;
/*
* Look for the maximum voltage in the 256 Fermi samplings
*/
        vmax= 0.;
        for(i=0;i<sample;i++)
        {
            if(deb>0)printf("%d %d\n",i,*(arr+i));
            if((float)arr[i]>vmax)vmax=(float)*(arr+i);
        }
/*
* Push into the BMenu stack the relevant information, i.e. the
* difference between the maximum and the average of the first 5 samplings
*/
        pushStack(vmax-average);
    }
}

```

References

- [1] ATLAS Collaboration, *TILE Calorimeter Technical Design Report*, CERN/LHCC/96-42, 15 December 1996
- [2] R. Brun, *HBOOK users guide*, Program Library Y250. CERN, 1992.
- [3] R. Brun, O. Couet, C. Vandoni and P. Zanmarini, *PAW users guide*, Program Library Q121. CERN, 1991.