Read-out of ATLAS radiation monitors in the Inner Detector

This document describes the communication protocol and procedures with active radiation monitors in ATLAS inner detector.

Prepared by:	*Checked by:*	*Approved by:*
Gregor Kramberger, J. Stefan Institute **Igor Mandić, J. Stefan Institute**		

Distribution List

History of Changes

Rev. No.	*Date*	*Pages*	*Description of changes*

0 Introduction

This document describes the readout procedure of ATLAS radiation monitors in the Inner Detector. There are two types of **r**adiation **m**onitor **s**ensor **b**oards (RMSB), one for Inner Detector (ID-RMSB) and the other for the rest of the ATLAS (ATLAS-RMSB). There is a conceptual difference in the read out of both types. The difference originates from the design of the RMSBs and by the requirement that ELMBs serving monitors outside of the ID must run standard ELMB firmware. Details about readout procedures for monitors outside of the ID is described in a separate document.

Details about radiation sensors and about the design of RMSBs are described in reference [1].

1 ID-RMSB

There are 14 RMSBs in the ID. The read-out of all ID radiation monitors requires 7 ELMBs, which are mounted in 7 readout boxes. The scheme of connections of ELMB, DAC and PP (Patch panel) boards inside of each box is shown in Fig. 1. The readout boxes containing ELMB, DAC and PP boards (Fig. 1.) are installed on 6 PP2 locations (at one location there are 2 boxes) in the ATLAS detector.

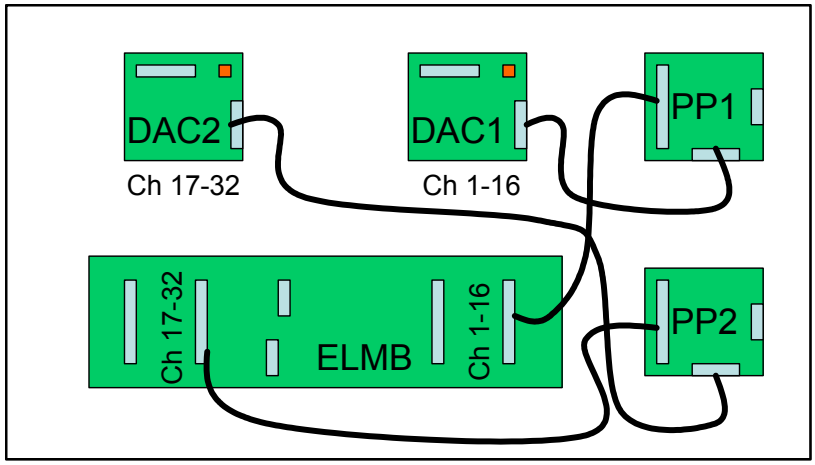

Fig. 1: Scheme of connections in Inner Detector readout box

They are connected to CAN-POWER crate in USA15 with 6 standard can-bus cables and with cables which supply power to DACs. Each RMSB is connected to its own patch panel (PP).

A RMSB hosts: **Temp sensor** (NTC), **2 P-i-N** diodes of high and low sensitivity, **3 RadFETs** (low, medium, high sensitivity), **two DMILL test structures** and **25 µm thick epitaxial silicon pad detector.** At the back side of RMSB there is a **resistive heater.** More detail about RMSBs and radiation sensors can be found in reference [1].

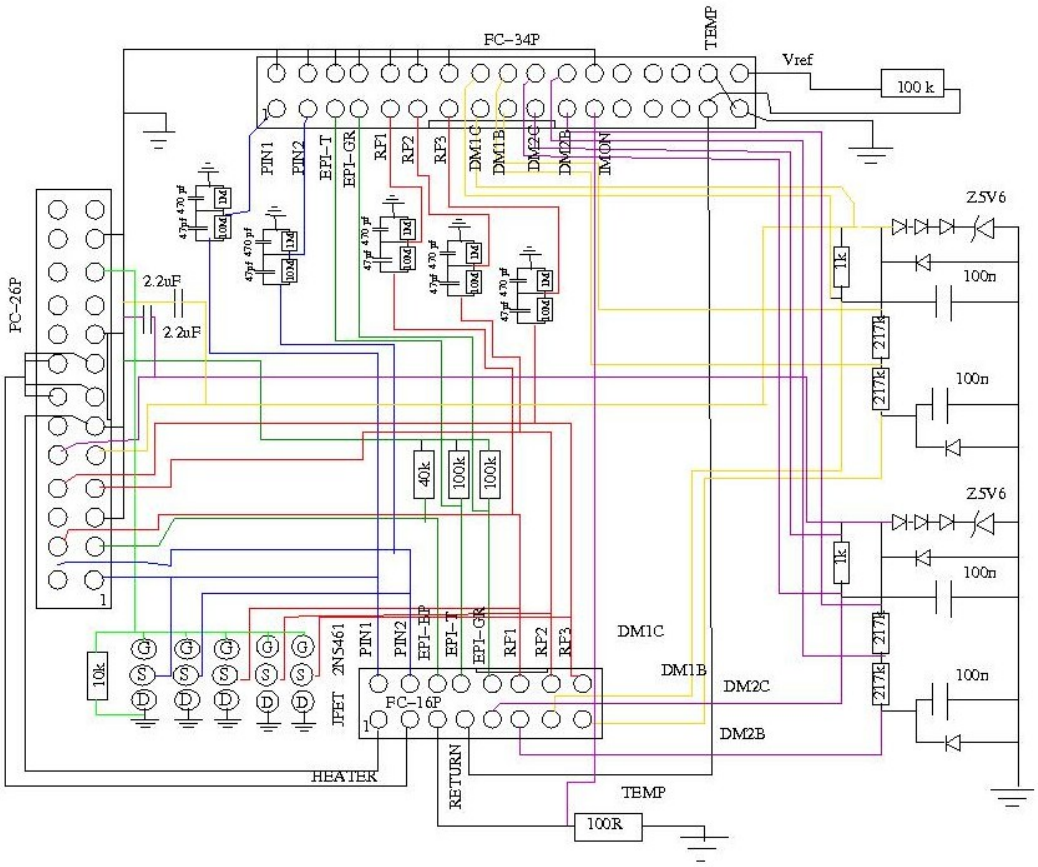

Fig. 2: Electrical scheme of Inner Detector patch panel

Electrical scheme of the PP is shown in Fig. 2. The PP hosts different circuits with the following functionalities:

- Switch: this are JFET transistors (2N5461) used to short the connections of sensors during irradiation which is most of the time. Frequency of readout should be such that the time when the JFETs are disconnected is short compared to irradiation time. The switch is OFF i.e. sensors are shorted when there is no voltage on their bias resistor.
- Temperature measurements: NTC is biased using the 2.5 V reference output of ELMB board. 100 kΩ resistor connected in series with the NTC defines the constant current of 25 μA in the NTC.
- Attenuators: voltage on radfet and p-i-n diodes are measured with ELMB ADCs via 10:1 attenuators
- Current measurement: there is a 100Ω resistor on the return line. By measuring voltage on this resistor the current in the sensors during readout is controlled.
- DMILL – circuitry for over-voltage and ESD protection of DMILL transistors together with measurement resistors. Measurement of voltage on 1 kΩ resistor gives the sum of collector and base current and form measurement of voltage on 217 kΩ resistor base current is determined
- HEATER – four DAC channels are used together to drive the hater
- EPI- one bias (40 kΩ) and two (one for bulk current and one for guard ring current) measurement resistors (100 kΩ)

1.1 Readout of sensors

Readout of the ID-RMSB sensors required changes in the ELMB software. The original ELMB code was upgraded to include the functionalities needed. The readout architecture is made in standard way using an

OPC server to communicate with ELMB over the can-bus. The following steps must be done to set-up and read sensors:

Step 1. There are two different DAC types that can be connected to ELMB. In order to select the right type for our setup the object dictionary at address 2500 sub-index 1 should be set to 1 (Maxim 525), by using SDO command with ELMB in pre-operational mode. The server configuration file should include a line:

DACType = ELMB_3F 2500 2 IO VT_UI1 - look in the Table for description

Step 2. The number of sensors connected to RMSBs and consequently to each ELMB is set to the object in the OD at address 2700 sub-index 0 (SDO command). The OPC server line needed is

RMCh = ELMB_3F 2700 0 IO VT_UI2

At the same time the number of channels used for standard ADC readout in PDO3 mode should be set to 0 – no readout.

Step 3. Each sensor requires one DAC and two ADCs.

- For P-I-N diodes the current of 1 mA is enforced by setting proper DAC channel and the voltage drop over the diode is readout on ADC1. The enforced current is read out as a voltage drop over 100 Ω resistor on ADC2. The conversion of voltage drop to NIEL is explained in section 3.
- For RADFETs the same procedure as for PIN is applied. The enforced current is 100 μA for LAAS radfet and 160 uA for REM radfets.
- For DMILL transistors; collector current of I_c = 10 μA is enforced. Summ of collector current I_c and base current I_b is measured as voltage drop over the 1 kΩ resistor on ADC1 and I_b as voltage drop over the 217 kΩ resistor on ADC2.
- For epi diodes the bias voltage is provided with DAC by forcing current through the 40 kΩ resistor in parallel to diode. Reverse leakage current and the guard ring current are measured as voltage drops over 100 kΩ resitors on ADC1 and ADC2.
- Temperature is measured in a standard way on one ADC channel.

In the Table the DAC and ADC channels corresponding to the given sensor in the RMSB are listed. Since this connections are hardwired in the RMSB they are stored as default values. The DAC values and attenuation factors are also shown.

RMSB #	Sensor	DAC ch.	DAC value	ADC 1	ADC 2	Att. ADC1	Att. ADC2
0	**T**	-	-	15	-	1	1
	PIN1	0	190	0	11	11	1
	PIN2	1	192	1	11	11	1
	RF1	3	201	4	11	11	1
	RF2	4	265	5	11	11	1
	RF3	5	265	6	11	11	1
	DMILL1	6	23	7	8	1	1
	DMILL2	7	23	9	10	1	1
	EPI	2	4000	2	3	1	1
1	**T**	-	-	15	-	1	1
	PIN1	0	190	0	27	11	1
	PIN2	1	192	1	27	11	1
	RF1	3	201	4	27	11	1
	RF2	4	265	5	27	11	1

	RF3	5	265	6	27	11	1
	DMILL1	6	23	7	14	1	1
	DMILL2	7	23	9	26	1	1
	EPI	2	4000	2	19	1	1

The readout procedure was programmed in the ELMB firmware. For this purpose the objects in the OD starting at 2700 sub-index 1 to 32 were added. Each data/object at this address has 4 bytes (UI4 – unsigned int of 4 bytes), which are coded in the following way.

bit: 0-11 dac value
bit: 12-17 adc1 channel
bit: 18-23 adc2 channel
bit: 24-28 dac channel
bit: 29-31 status

status bits (29,30,31) mean
0 0 0 = 0 ; T sensor
0 0 1 = 1 ; PIN
0 1 0 = 2 ; RADFET
0 1 1 = 3 ; DMILL
1 0 0 = 4 ; EPI-SI
1 0 1 = 5 ; PIN-No curr(ADC1)
1 1 0 = 6 ; RadFET-No curr(ADC1)
1 1 1 = 7 ; not used

The setup of the channels is done by using SDO commands. The opc server configuration file should therefore include lines looking like

RSP0 = ELMB_3F 2700 1 IO VT_UI4
RSP1 = ELMB_3F 2700 2 IO VT_UI4
…

The channel is set up by writing the configuration (4 bytes) to this objects.
The C code used for coding and decoding the values can be found below.

```
unsigned int RADMONConfigWrite(short DAC,char adc1, char adc2,char DACch,char status)
{
// return value that should be written to corresponding RSP
  char buff[200];
  unsigned int outval=0;
  unsigned char *td;
  td=(unsigned char *)&outval;
  outval=DAC;   td[1]|=adc1<<4; td[2]=adc1>>4;  td[2]|=adc2<<2;  td[3]=DACch;   td[3]=td[3]|(status<<5);
  return outval;
}

void RADMONConfigRead(unsigned int val)
{
//the variables are filled with values decoded from the 4 bytes returned frrom ELMB
char adc1, adc2,status,DACch;
short DAC;
unsigned char *Conf=(unsigned char *)&val;
unsigned char *dacv=(unsigned char *)&dac_val;

  //dac value
  DAC[0]=Conf[0];  DAC[1]=Conf[1] & 15;
  //adc11,2 channel number
```

```
 adc11=(Conf[1] >> 4 ) | ((Conf[2] << 4) & 48)  ;
 adc2=Conf[2] >> 2 ;
 //get dac ch
 DACch=Conf[3] & 31;
 //get status
 status=Conf[3] >> 5;
}
```

Step 4. As each sensor type requires its own time sequence of applying bias current, waiting and then reading, this should be programmed. The object added for this purpose starts at address 2701 (see table of added OD object in appendix)

The sequence – time delays (set DAC, read ADCs) for each sensor type is coded in the 4 byte word. For each sensor the delay is expressed in units of 10 ms. Therefore, delay [ms] = (0-255) * 10 ms. The minimum delay is 0 ms and maximum (all bits set to 1) is 2550 ms.

Bits 0-7 (byte 0); delay for PiN;
Bits 8-15 (byte 1); delay for RadFETs;
Bits 16-23 (byte 2); delay for DMILLs;
Bits 24-31 (byte 3); delay for epitaxial;

Again, to the opc server config a line should be included which enables communication with ELMB and set the delays:

RMdelay = ELMB_3F 2701 1 IO VT_UI4

It is recommended to use 500 ms delay for PIN, 1000 ms for RadFET, DMILL and epitaxial pad detectors.

The C++ coding for the object is done with

```
unsigned int SetDelay(unsigned char pin,unsigned char rf,unsigned char dmill,unsigned char epi)
{
//return value should be sent to ELMB with SDO, the pin, rf,dmill and epi variables refer to delays
 unsigned int time;
 ptime=(unsigned char *)&time;
 ptime[0]=pin; ptime[1]=rf; ptime[2]=dmill; ptime[3]=epi;
 return time;
}
```

Step 5. To enable biasing of sensor, the switch which shorts sensor contacts during the irradiation must be open first. When the readout is done with *SYNC* this is automatically done. The selection of DAC channel which opens/closes the switch is made in the object dictionary address 2701 sub-index 0 (SDO command). This line should be include to server configuration file:

RMsw = ELMB_3F 2701 0 IO VT_UI4

One DAC channel is used to open and close switches on each Patch Panel. In present configuration a maximum of two DAC boards connected to two Patch Panels, each serving one RMSB, are connected to one ELMB so two DAC channels are used to control the switch. The coding of bits are:

Bit 0-7 (byte 0) ; DAC channel of the RMSB0 switch
Bit 8-15 (byte 0) ; DAC channel of the RMSB1 switch

Step 6. Once all sensors are configured the ELMB must be set to operational mode. The readout is done via PDO4 messages. The sequence is triggered with the SYNC command (in order to suppress ADC readout of standard firmware PDO3 – set Channel number of the standard OPC item to 0). The PDO4 should be configured with the following lines.

[PDO]
4BF = CAN_BUS_1 404 IN 0 20
[PDOItem]
RSM = ELMB_3F 4BF VT_R8 ALL 20

=> for ELMB with address 3F for 32 channels

The PDO items returned are shown after SYNC in the opc server as RSM# (#1-32) contain 8 byte information. The information is coded in the following way:

Bits 0-7 (byte 1); radmon channel;
Bits 8-31 (byte 2,3,4); ADC2
Bits 32-55 (byte 5,6,7); ADC1
Bits 24-31 (byte 8); status byte (bit OR of the status bytes from AD conversion for each ADC)

The C code to decode 8 byte message from the ELMB (PDO4 written to RSMF# and RSME#)

```
void PDO4_Decode(unsigned char *val) {
char channel_no;
unsigned char *adc1,*adc2;
adc1=(unsigned char *) &adc_val1;
adc2=(unsigned char *) &adc_val2;
channel_no=val[0]; conversion_status=val[7];
for(int i=0;i<3;i++)
  { adc1[i]=val[5-i];  adc2[i]=val[2-i]; }
 adc1[3]='\0';  adc2[3]='\0';
}
```

1.2 HEATERS

The aim of temperature control is to stabilize temperature of RMSBs at T = 20°C ± 2 °C and by that assure: smaller measurement error, owing to smaller temperature correction needed, and controlled annealing of the sensors.

Electrical current enforced through the resistive back side (320 Ω) of a Inner Detector RMSB heats the ceramic support of radiation sensors. The power (P) of the heater is controlled by the current from 4 DAC channels connected together. Power needed to rise the temperature of the hybrid by 1°C is 0.017 W and this relation is valid over wide range of temperatures ($\Delta T = 0.017$ W/K $\cdot P$). It was found that time constant of temperature settling is around 10 min, which sets the time scale of temperature control actions. Temperature control software (PID control) was added to ELMB firmware. It uses information from temperature measurement and it sets the current in the DAC channels which power the heater to reach the desired temperature.

The weights of the proportional (P), integral (I) and differentiating part (D) of the PID control are all steered by the objects in the OD (see appendix A). The heater works in the following sequence:

1. Temperature is measured and stored in the circular buffer; although only one temperature measurement is needed prior to the most recent measurement the buffer can be up to 50 points deep. The interval between the measurements/action (new value of heater current) in seconds is set in the object dictionary (address 2703 sub-index 3).
2. The action (in this case the DAC current enforced) is calculated using a PID equation

$$DAC = \frac{d(Ts - Tm)}{dt} \cdot \tau_d + \tau_i \cdot \int (Ts - Tm)\, dt + Gain \cdot (Ts - Tm)$$

The constants are τ_d, and τ_i weight the response of the differentiating and integrating part while *Gain* weights the response to the temperature difference. It was found in the system-tests that integration part is enough to stabilize the temperature within 0.3°C and it is therefore recommended to use only I. In this case td and Gain are set to 0. The typical values for Time=20 s, τ_i=0.1 ADCcount/(s K).

3. The action is taken in such a way that the required DAC response - power is shared by all four channels equally (e.g. 20 mA current needed is split into each DAC giving 5 mA). To prevent instantaneous large currents the maximum increase of current in given step is limited to 250 DAC counts.
4. After the action is taken the count down to new action begins.

The setup of the heater is done with a set of objects starting at OD address 2703 and 2704. The list of objects at different sub-addresses given in the OD dictionary table can be found in the appendix. To access them from OPC server the lines are needed in the configuration file

HEAT_Ch = ELMB_3F 2703 0 IO VT_UI2
HEAT_Adc = ELMB_3F 2703 1 IO VT_UI2
HEAT_DAC = ELMB_3F 2703 2 IO VT_UI4
HEAT_Time = ELMB_3F 2703 3 IO VT_UI2
HEAT_SetT = ELMB_3F 2703 4 IO VT_R4
HEAT_Gain = ELMB_3F 2703 5 IO VT_R4
HEAT_TauI = ELMB_3F 2703 6 IO VT_R4
HEAT_TauD = ELMB_3F 2703 7 IO VT_R4
HEAT_VAL = ELMB_3F 2703 9 IN VT_R4
HEAT_Mode = ELMB_3F 2703 8 IO VT_UI2

There are 4 DAC channels connected to each heater are able to provide 80 mA. The mapping can be found here

Heater DAC 1	8
Heater DAC 2	9
Heater DAC 3	10
Heater DAC 4	11

The circular buffer which is filled with temperature measurements can be accessed by objects at OD address 2704. The counter counting the time to next measurements is at sub-index 0, while measured temperatures can be accessed by sub-index>0. The sub-index>50 are for measurements for second heater.

APPENDIX A – Modifications required by ID-RADMON to object dictionary of ELMB

Index	Sub index	Description	data	Att.	Default	Comment
2700	0	Number of sensor channels	UI4	RW	0	Number of sensor channels
2700	1	Sensor 1				Setup of sensor: Bits 0-7 (byte 1) ; radmon channel Bits 8-31 (byte 2,3,4) ; ADC1 Bits 32-55 (byte 5,6,7) ; ADC1 Bits 24-31 (byte 3); status byte
2700	…	Sensor 2	…	…	…	…
2701	0	RMSB switch	UI4	RW	0	Byte 1; switch RMSB 1 Byte 2; switch RMSB 2
2701	1	RMSB sensor delays	UI4	RW	0	Byte 1; delay for pin; Byte 2; delay for radfets; Byte 3; delay for dmill Byte 4; delay for epitaxial
2701	2	Mode	UI2	RW	0	Selection of the readout mode (not used at the moment)
2702	1					Reading the sensor with an SDO command. Due to 4 byte restriction, only one ADC can be readout.
2703	0	Number of heaters			0	Sets the number of heaters and at the same time turns on those heaters. If set to 1 sets heater #1 on. If set to 2 – heater of RMSB#1 and RMSB#2 are on.
2703	1	ADC channel number of the T sensors	UI2	RW	0	Sets the ADC channel number of the temperature measurement of the first heater.
2703	2	DAC ch. Numbers. Up to 4 heaters can be connected	UI4	RW		Byte 1; 1st DAC channel of Heater 1 Byte 2; 2nd DAC channel of Heater 1 Byte 3; 3rd DAC channel of Heater 1 Byte 4; 4th DAC channel of Heater 1
2703	3	Time interval between the action – temperature measurement [s]	UI2	RW		The value gives effectively the frequency of the T measurements and actions taken by PID controller.
2703	4	Temperature set	UI4	RW		Set temperature [°C] – please note that the data format is UI4 although the data is actually float;
2703	5	Gain	UI4	RW	0	Gain of proportional part of PID (in most cases set to 0)
2703	6	TauI	UI4	RW		Weight for integration part of PID (should not be 0)
2703	7	TauD	UI4	RW		Weight for differential part of PID (should be 0)
2703	8	Mode				Determines what is filled in the readout buffer; 0-Tm; 1-action; 2-integral
2703	9	Value read - set by Mode	UI4	R		According to Mode the value of different variables is read.
2703	10	ADC channel number of the T sensors	UI2	RW	0	Sets the ADC channel number of the temperature measurement of the second heater.
2703	11	…	...	…	…	The same structure as for heater 1 is repeated.
2704	0	Heater depth				
2704	1	Tmeasured [0]	UI4			First measurement in the circular buffer
2704	2	Tmeasured [1]	UI4			
2704	3	…	…	…	…	…

References

[1] G. Kramberger et al., ***Conceptual Design and Functional Specification of ATLAS Radiation Monitor,*** ATL-IC-ES-0017, RADMON-Concept-V2.0.

[2] ELMB firmware.

[3] F. Ravotti, PhD thesis, CERN 2006.