The GLAST experiment at SLAC

The Power Distribution Unit

Electronics group

Programming ICD specification

Document Version:	1.2
Document Issue:	2
Document Edition:	English
Document Status:	Under release control
Document ID:	LAT-TD-01543
Document Date:	July 14, 2004

Stanford Linear Accelerator Center (SLAC)
2575 Sandhill Road
Menlo Park California, 94025 USA

Abstract

A conceptual design of the external interface for the PDU (Power Distribution Unit).

Hardware compatibility

This document assumes the following hardware revisions:

PDU: Version TBD

Intended audience

This document is intended principally as a guide for the *developer* and *users* of the PDU Power Distribution Unit. Users include:

- Developers of the sub-system electronics which interface with the PDU
- Developers of Flight-Software
- Developers of I&T (Integration and Test) based systems

All readers of this document are expected to be familiar with the concepts described in [1] and indeed, most the references described within.

Conventions used in this document

Certain special typographical conventions are used in this document. They are documented here for the convenience of the reader:

- Field names are shown in bold and italics (*e.g.*, ***respond*** or ***parity***).
- Acronyms are shown in small caps (*e.g.*, SLAC or TEM).
- Hardware signal or register names are shown in Courier bold (*e.g.*, `RIGHT_FIRST` or `LAYER_MASK_1`)

References

1 “LAT Inter-module Communications - A reference manual,” Michael Huffer, LAT-TD-00606.

2 “LAT Power Distribution Unit Conceptual Design and Electrical ICD,” Gunther Haller and Patrick Young, LAT-TD-01743.

3 “LAT GASU Specification and Electrical ICD,” Gunther Haller, LAT-SS-01544.

4 Data sheet for the “MAXIM +2.7V, low-power, 2 channel 108ksps, Serial 12-Bit ADCs in 8-Pin uMax,” hhttp://www.maxim-ic.com.

Note: For additional resources, refer to the LAT Electronics, DAQ Critical Design Requirements List. On the LAT Electronics, Data Acquisition & Instrument Flight Software page (http://www-glast.slac.stanford.edu/Elec_DAQ/Elec_DAQ_home.htm), click Hardware and then click List of all documents.

Document Control Sheet

Table 1 Document Control Sheet

Document	**Title:**	The Power Distribution Unit Programming ICD specification		
	Version:	1.2		
	Issue:	2		
	Edition:	English		
	ID:	LAT-TD-01543		
	Status:	Under release control		
	Created:	February 9, 2002		
	Date:	July 14, 2004		
	Access:	V:\GLAST\Electronics\Design Documents\PDU\v1.2\Frontmatter.fm		
	Keywords:	Power Distribution Unit PDU		
Tools	**DTP System:**	Adobe FrameMaker	**Version:**	6.0
	Layout Template:	Software Documentation Layout Templates	**Version:**	V2.0 - 5 July 1999
	Content Template:	--	**Version:**	--
Authorship	**Coordinator:**	Michael Huffer		
	Written by:	Michael Huffer		

Table 2 Approval sheet

Name	Title	Signature	Date
Gunther Haller	LAT Chief Electronics Engineer		
JJ Russell	Flight Software Lead		

Document Status Sheet

Table 3 Document Status Sheet

Title:	The Power Distribution Unit Programming ICD specification		
ID:	LAT-TD-01543		
Version	**Issue**	**Date**	**Reason for change**
1.0	1	5/05/2003	Initial draft
1.1	1	5/07/2003	First release for public consumption
1.1	3	3/05/2004	Updated fonts. Labelled issue 3 to avoid confusion with version 1.1, issue 1, which was labelled issue 2.
1.2	1	5/06/2004	Edits from Jana including typos and total revision of the environmental monitoring registers.
1.2	2	6/10/04	Updated references and PDF TOC.

Table of Contents

Abstract . . . 3

Hardware compatibility . . . 3

Intended audience . . . 3

Conventions used in this document . . . 3

References . . . 4

Document Control Sheet . . . 5

Document Status Sheet . . . 6

List of Figures . . . 9

List of Tables . . . 11

Chapter 1
Principles of operation . . . 13
- 1.1 Overview . . . 13

Chapter 2
Registers . . . 15
- 2.1 Introduction . . . 15
- 2.2 Conventions . . . 15
- 2.3 PDU controller registers . . . 16
  - 2.3.1 Configuration register . . . 16
    - 2.3.1.1 Version ID . . . 17
  - 2.3.2 Address register . . . 17
  - 2.3.3 Command/Response statistics register . . . 18

2.4 Power Management . . . 18
2.4.1 Selective set and clear of register fields . . . 18
2.4.2 Power Management of the Crates . . . 19
2.4.3 Power Management of the TEMS . . . 20
2.4.4 Power Management of the ACD . . . 21
2.5 Monitor register . . . 22
2.6 Environmental monitoring . . . 23
2.6.1 ADCS_00-06 register . . . 24
2.6.2 ADCS_07-13 register . . . 25
2.6.3 ADCS_14-19 register . . . 25
2.6.4 Groups of Environmental quantities . . . 26

Chapter 3
Commanding . . . 35
3.1 Overview . . . 35
3.1.1 Conventions . . . 35
3.2 The PDU's access descriptor . . . 36
3.3 Accessing the controller . . . 37
3.3.1 Dataless commands . . . 37
3.3.2 Load commands . . . 38
3.3.3 Read commands . . . 38
3.4 Accessing the Environmental monitoring block . . . 39
3.4.1 Dataless commands . . . 39
3.4.2 Load commands . . . 39
3.4.3 Read commands . . . 39

List of Figures

Figure 1 p. 16 Configuration register

Figure 2 p. 17 Structure of revision register or field

Figure 3 p. 18 Address register

Figure 4 p. 18 The Command/Response statistics register

Figure 5 p. 19 Power supply switching for the crates

Figure 6 p. 21 Power management register for the TEMs

Figure 7 p. 21 Power supply switching for the ACD

Figure 8 p. 22 Environmental monitor register

Figure 9 p. 25 Environmental monitoring result register (00-06)

Figure 10 p. 25 Environmental monitoring result register (07-13)

Figure 11 p. 26 Environmental monitoring result register (14-19)

Figure 12 p. 35 Hierarchy of target types

Figure 13 p. 36 PDU access descriptor

Figure 14 p. 37 Access descriptor for the controller's dataless commands

Figure 15 p. 38 Access descriptor for the controller's register load commands

Figure 16 p. 38 Payload for the controller's register load commands

Figure 17 p. 38 Access descriptor for the controller's register read commands

Figure 18 p. 39 Response to a register read command of the controller

Figure 19 p. 39 Access descriptor for register read commands of the environmental monitoring block

Figure 20 p. 40 Response to a register read of the environmental monitoring block

List of Tables

Table 1	p. 5	Document Control Sheet
Table 2	p. 5	Approval sheet
Table 3	p. 6	Document Status Sheet
Table 4	p. 16	The PDU control registers
Table 5	p. 17	Usage of the type field of the revision register
Table 6	p. 23	Environmental monitoring registers
Table 7	p. 27	Group zero (0) Environmental quantities
Table 8	p. 28	Group one (1) Environmental quantities
Table 9	p. 29	Group two (2) Environmental quantities
Table 10	p. 30	Group three (3) Environmental quantities
Table 11	p. 31	Group four (4) Environmental quantities
Table 12	p. 32	Group five (5) Environmental quantities
Table 13	p. 33	Group six (6) Environmental quantities
Table 14	p. 34	Group seven (7) Environmental quantities
Table 15	p. 36	Block numbers of the PDU
Table 16	p. 37	The controller’s dataless commands

Chapter 1

Principles of operation

1.1 Overview

The PDU (Power Distribution Unit) processes the 28 volt feeds from the spacecraft and both switches and distributes this power to various DAQ assemblies within the LAT. These include the GASU, the TEM (Tower Electronics Module), ACD front-end electronics, and the EPU (Event Processing Unit) crates. In addition, various temperatures and voltages around the LAT are digitized and monitored. For more details see [2].

Chapter 2

Registers

2.1 Introduction

The PDU contains a series of *registers* for configuration and management. These registers are divided into two groups:

- PDU control
- Environmental monitoring

Within each group, registers have a common length to allow their access through broadcast operations. The access model for these registers is through the LAT common Command/Response Protocol. This protocol is specified generically in [1]. The specific implementation of this protocol for the PDU is described in Chapter 3.

2.2 Conventions

Certain conventions apply to the fields within a register. These conventions fit into one of three classes:

Not defined: Undefined fields are identified as Must Be Zero (MBZ) and are illustrated *grayed out*. An MBZ field will:

— Read back as zero

— Ignore writes

— Reset to zero

Read/Write: A *Reset* will set a read/write field to zero.

Read-only: Read-only fields are illustrated *lightly* grayed-out along with their intended value. Any read-only field will:

— Ignore writes

— Reset to zero, unless otherwise documented

Any field used as a boolean has a width of one bit. A value of one (1) is used to indicate its *set* or *true* sense and a value of zero (0) to indicate its *clear* or *false* sense. Field numbering for registers is such that zero (0) corresponds to a register's Least Significant Bit (LSB) and thirty-one (decimal) corresponds to a register's Most Significant Bit (MSB). Register addresses are specified in *hexadecimal* unless otherwise noted.

2.3 PDU controller registers

This section incorporates all the registers whose configuration has a global effect over all the functional blocks of the PDU.

Table 4 The PDU control registers

Name	Address	Description
CONFIGURATION	00	Configuration and setup
ADDRESS	01	LATp node address
C/R_STATISTICS	02	Command/response statistics
CRATES	03	Power control for EPU crates
TEMS	04	Power control for TEMs
ACD	05	Power for the FREE boards of the ACD
MONITOR	06	Selects environmental monitoring group and starts acquisition
Total	**7**	

2.3.1 Configuration register

In general, this register allows defeating those features of the PDU which in the normal course of operation would always be enabled. This functionality is present only to allow *testing* of those features and to perhaps recover from single-point failure. Great care should be exercised in using any other than the default values for this register.

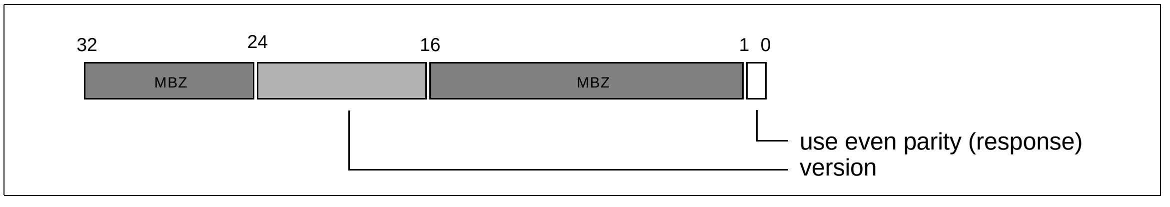

Figure 1 Configuration register

use even parity (response): Determines whether the parity generated by the PDU when transmitting *response* packets is *odd* or *even*. If the field is *clear*, *odd* parity is generated. If the field is *set*, *even* parity is generated. Note: This field is intended to be used to test whether the response receiver will in fact detect parity errors. In normal operation this field should be always be *clear*.

version: Specifies the hardware revision level of the PDU. The structure of this field is defined in Figure 2. Note: This field is *read-only*.

2.3.1.1 Version ID

The fields of this register are somewhat self-explanatory with the exception of the ***type*** field. This field is intended to differentiate both the context in which the module was implemented and how it was intended to be used. The values for this field are defined in Table 5.

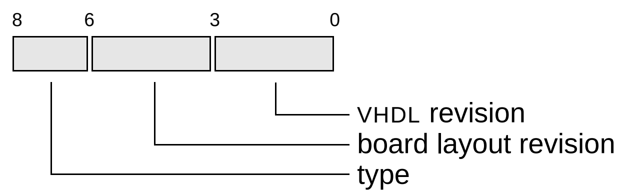

Figure 2 Structure of revision register or field

Table 5 Usage of the type field of the revision register

Value[1]	Description
`00`	Software emulation/engineering model
`01`	Engineering model
`10`	Qualification model
`11`	Flight model

1. In binary

2.3.2 Address register

This register is used to specify the PDU's node address on the Command/Response fabric. (See [1].) Note that all nodes on the Command/Response fabric must have a unique value. This register allows for definition of *only* the address's lower five bits. As the PDU is a *slave* on the Command/Response fabric, the high order bit is an implied *zero* (0).

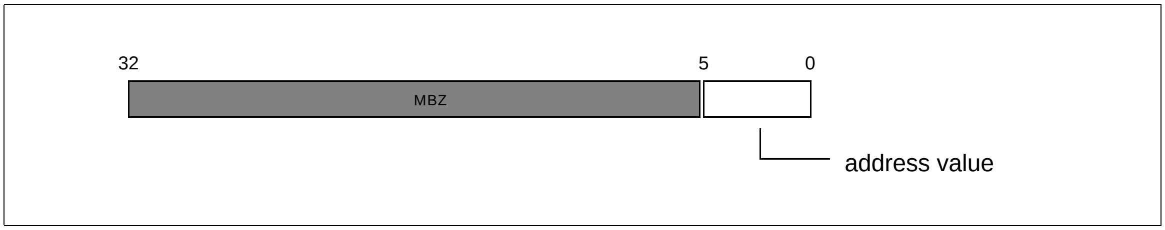

Figure 3 Address register

2.3.3 Command/Response statistics register

The PDU is a node on the Command/Response fabric. As such, it is obligated to keep statistics on both the commands it receives and the responses it transmits. (See [1].) These statistics are accessed in the register illustrated in Figure 4. Note that when the register is written, the value to be written will be ignored and the register set to *zero*.

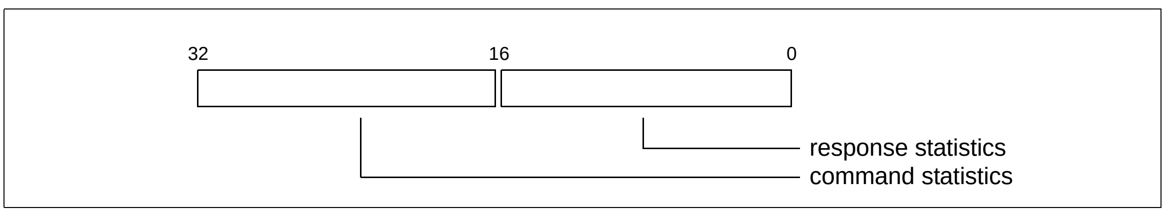

Figure 4 The Command/Response statistics register

response statistics: The packet statistics for the *outgoing response* wire. See [1] for a description of the structure of this field.

command statistics: The packet statistics for the *incoming command* wire. See [1] for a description of the structure of this field.

2.4 Power Management

2.4.1 Selective set and clear of register fields

The following two registers are somewhat peculiar, in that the specified interface is trying to maximize two goals:

- Minimize the number of round trips to power the instrument
- Allow for write access to individual fields without the attendant horrors of dealing with the necessity of indivisible read-modify-write

These goals are realized by having for each field of a register a corresponding ***write enable*** field. Only if the write enable field is set, is the value of a field actually allowed to change. The corresponding ***write enable*** field is always found by shifting the field offset, *left* `10`[1] places.

For example, assume the ***EPU2 on*** field of the register illustrated in Figure 5 is to be changed, leaving all its other fields unchanged. The bit offset of this field is `3`. Shifting this offset left sixteen (decimal) places locates the field's ***write enable*** field at bit-offset `13` (a mask of `40000`). This field must be *set*, in order to either turn on or off the power to EPU_2. For example, writing the value `40004` to the register will turn *on* the EPU_2 crate. Writing the value `40000` will turn *off* the EPU_2 crate. All the other fields of the register will remain unchanged. Note: This behaviour only applies to the register when it is *written*. If the register is *read*, the ***write enable*** fields will return *zero*.

2.4.2 Power Management of the Crates

This register switches the 28 volt power supplies of the *three* EPU (Event Processing Unit) crates. In order to mitigate against single-point-failure, each one of these supplies has *two* DC-to-DC converters. One converter is called the *primary* converter and the other the *redundant* converter. Only one of these converters can be active at any one time. This register also allows for switching between primary and redundant converters. The structure of this register is illustrated in Figure 5:

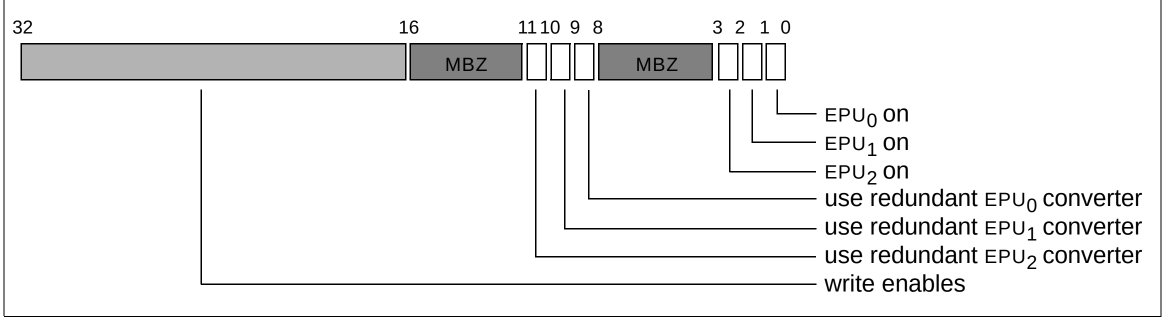

Figure 5 Power supply switching for the crates

EPU (0) on: The LAT has three EPU crates. This field determines whether the PDU supplies 28 volts to the *first*, or EPU_0, crate. If this field is *set*, the switch is closed and 28 volts is supplied to the crate. If the field is *clear*, the switch is open and 28 volts is *not* supplied to the crate.

EPU (1) on: The LAT has three EPU crates. This field determines whether the PDU supplies 28 volts to the *second*, or EPU_1, crate. If this field is *set*, the switch is closed and 28 volts is supplied to the crate. If the field is *clear*, the switch is open and 28 volts is *not* supplied to the crate.

1. All values in *hexadecimal*, unless noted

EPU (2) on: The LAT has three EPU crates. This field determines whether the PDU supplies 28 volts to the *third*, or EPU_2, crate. If this field is *set*, the switch is closed and 28 volts is supplied to the crate. If the field is *clear*, the switch is open and 28 volts is *not* supplied to the crate.

use redundant EPU (0) converter: In order to mitigate against single-point-failure, each EPU crate has *two* DC-to-DC converters. One converter is called the *primary* converter and the other the *redundant* converter. Only one of these converters can be active at any one time. This field provides for selection of either converter of the *first*, or EPU_0, crate. If the field is *clear*, the primary converter is used. If the field is *set*, the redundant converter is used.

use redundant EPU (1) converter: In order to mitigate against single-point-failure, each EPU crate has *two* DC-to-DC converters. One converter is called the *primary* converter and the other the *redundant* converter. Only one of these converters can be active at any one time. This field provides for selection of either converter of the *second*, or EPU_1, crate. If the field is *clear*, the primary converter is used. If the field is *set*, the redundant converter is used.

use redundant EPU (2) converter: In order to mitigate against single-point-failure, each EPU crate has *two* DC-to-DC converters. One converter is called the *primary* converter and the other the *redundant* converter. Only one of these converters can be active at any one time. This field provides for selection of either converter of the *third*, or EPU_2, crate. If the field is *clear*, the primary converter is used. If the field is *set*, the redundant converter is used.

write enables: This field is only applicable on *write* access. Its usage is described in Section 2.4.1. On *read* access, this field returns *zero*.

2.4.3 Power Management of the TEMS

This register switches the 28 volt supplies to the sixteen towers of the LAT. The electronics for each tower is represented by the TEM (Tower Electronics Module). The structure of this register is illustrated in Figure 6. Power switching for each TEM is represented by a single bit field. If the field for the corresponding TEM is *set*, the switch is closed and 28 volts is supplied to the TEM. If the field is *clear*, the switch is open and 28 volts is *not* supplied to the TEM.

The ***write enable*** fields are applicable only on *write* access. Its usage is described in Section 2.4.1. On *read* access, this field returns *zero*.

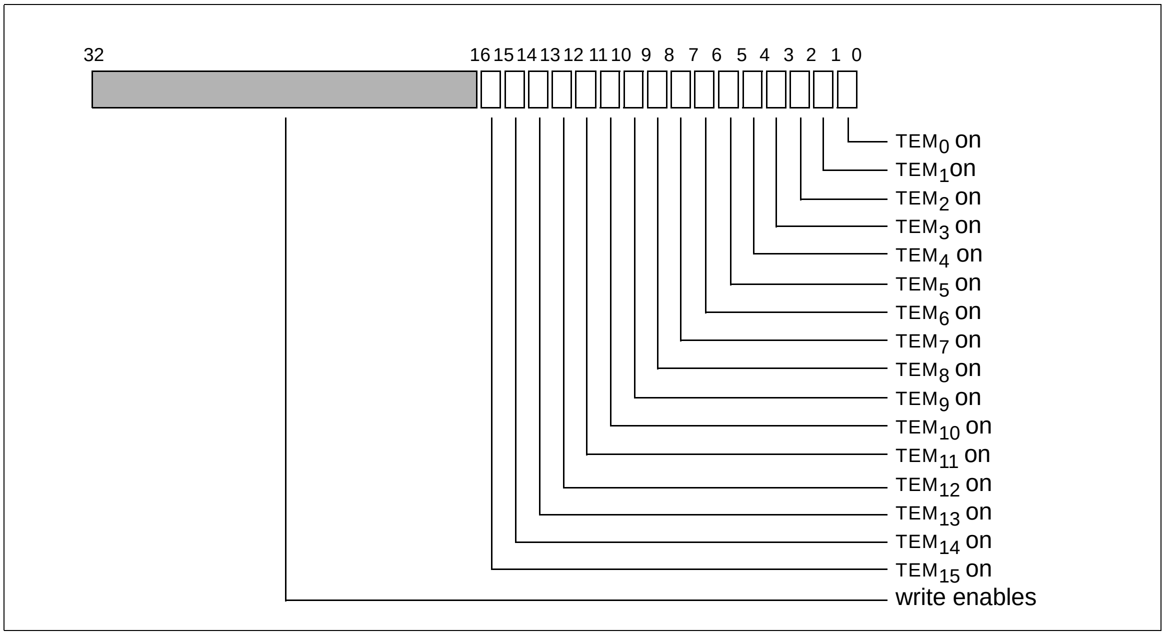

Figure 6 Power management register for the TEMS

2.4.4 Power Management of the ACD

Power for the 12 FREE boards of the ACD is supplied through a DC-to-DC converter located within the GASU box. (See [3].) In order to mitigate against single-point-failure, the GASU actually contains *two* power supplies. One is called the *primary* supply and the other the *redundant* supply. Only one of the two supplies is active at any one point in time. Within each supply (primary or redundant), are two *converters*. One is called the *primary* converter and the other the *redundant* converter. Again, only one of the two converters is active at any one point in time. Thus, the GASU contains a total of *four* DC-to-DC converters (for the ACD); however, only one of the four is used at any one time. This register is used to both switch the 28 volts to the appropriate supply and to select within the supply which of its two converters will be used. The structure of this register is illustrated in Figure 7:

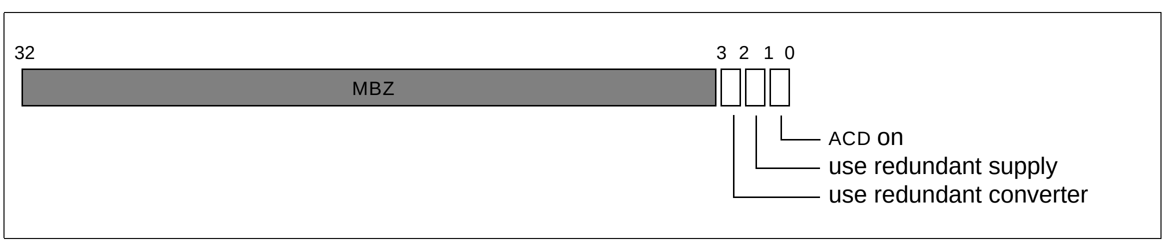

Figure 7 Power supply switching for the ACD

ACD on: This field determines whether the PDU supplies 28 volts to the selected supply (as explained in the field below) on the GASU used to power the 12 FREE boards of the ACD. If this field is *set*, the switch is closed and 28 volts is supplied to the ACD. If the field is *clear*, the switch is open and 28 volts is *not* supplied to the ACD.

use redundant supply: In order to mitigate against single-point-failure, each GASU has *two* power supplies used to power the ACD. One supply is called the *primary* and the other the *redundant*. Only one of these supplies can be active at any one time. This field provides for selection of one of the two supplies. If the field is *clear*, the primary supply is used. If the field is *set*, the redundant supply is used.

use redundant converter: In order to mitigate against single-point-failure, each supply of the GASU has *two* DC-to-DC converters. The selection of the supply is determined by the field described previously. This field controls the selection of the converter within a supply. One converter is called the *primary* converter and the other the *redundant* converter. Only one of these converters can be active at any one time. This field provides for selection of either converter. If the field is *clear*, the primary converter is used. If the field is *set*, the redundant converter is used.

2.5 Monitor register

This register is used to initiate the acquisition of the environmental quantities monitored by the PDU. Its usage is discussed in the section below. (See Section 2.6.)

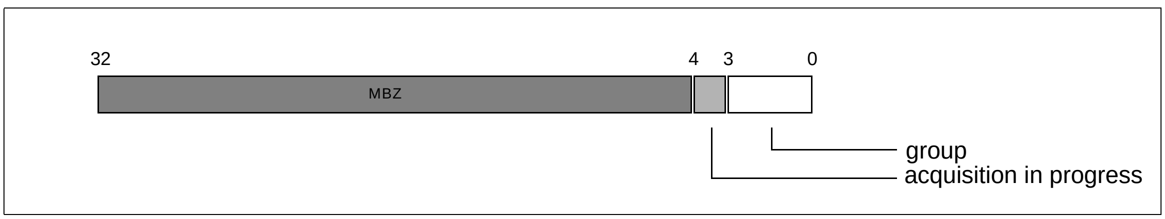

Figure 8 Environmental monitor register

group: This field specifies the group whose twenty different quantities are to be acquired. The group is specified by a value from zero (0) to seven (7), where the group number corresponds to one of the tables discussed in Section 2.6.4. Writing to this field (assuming an acquisition cycle is not currently in progress) will initiate an acquisition cycle as described in Section 2.6.

acquisition in progress: This field reflects whether an acquisition cycle is currently in progress. If the field is *clear*, a cycle is *not* on-going. If this field is *set*, an acquisition cycle is currently on-going. This field is *read-only*.

2.6 Environmental monitoring

The PDU provides the capability to monitor various environmental quantities over the entire LAT. There are a total of 160 quantities the PDU monitors and these quantities include:

- Voltages of the DC-to-DC converters used in the 3 EPU crates
- Voltages of the DC-to-DC converters used in the 16 tower power supplies
- Temperatures within the 3 EPU crates
- Temperatures within the 16 tower power supplies
- Temperatures on the PCB (Printed Circuit Board) of the 16 Tower Electronics Modules (TEM)
- Temperatures on the LAT mechanical structure

In order to acquire (digitally) these environmental quantities, the PDU contains twenty, individual, 12-bit ADCs. The ADC is a commercial part, the MAXIM 145. (See [4].) Thus, the PDU can, on request, acquire and store simultaneously twenty different quantities. However, these cannot be any arbitrary 20 different values out of the collection of 160. Instead, the PDU organizes these 160 quantities into 8 sets of 20 quantities apiece. Each set is called a *group*. The members of each group are fixed and are specified in Section 2.6.4.

The PDU provides a register which allows the user to specify which particular group to acquire (discussed in Section 2.5) and then after conversion is complete, a set of *three* registers which will contain the 20 digitized quantities corresponding to the members of the specified group. Three registers are necessary as the Command/Response protocol (with its maximum, one cell sized response) would not allow all twenty quantities to be returned in a single read. So these quantities are split over three registers; the first two registers contain seven members each of a group and the last register contains six members of a group. These three registers are contained in their own block, called the ***environmental monitoring*** block as enumerated in Table 6. This block has two unique features:

- All its registers are 96-bits (12 bytes) long.
- All its registers are read-only.

Table 6 Environmental monitoring registers

Name	Address	Description
`ADCS_00-06`	`00`	Conversion results for the *first* 7 members of a specified group
`ADCS_07-13`	`01`	Conversion results for the *second* 7 members of a specified group
`ADCS_14-19`	`02`	Conversion results for the *last* 6 members of a specified group
Total	**3**	

To actually employ the monitoring capability of the PDU is a three step procedure:

i. Determine the group that is to be acquired by writing a value to the ***group*** field of the MONITOR register. (See Section 2.5.) The value corresponds to the group whose data is to be acquired. This register contains an ***acquisition in progress*** flag. If this flag is *set*, an acquisition cycle is in progress; if this flag is *clear*, an acquisition cycle can be started. The act of writing this register starts an acquisition cycle. Once started, the PDU will flip this flag to its *set* state, indicating acquisition is in progress. If an acquisition is in progress when the register is written, the write is ignored. Note that the ***acquisition in progress*** flag is *read-only*—its state is completely controlled by the PDU.

ii. Wait for the acquisition cycle to complete. (The acquisition time is roughly 2804 system clocks, or 140.2 micro-seconds). When the conversion is complete, the ***acquisition in progress*** field will be *clear*.

iii. Obtain the twenty acquired values by reading the three ADC registers. These registers are described in the sections below. The first register contains the first seven acquired values for the specified group, the second register, the next seven, and the third register, the last six values. The acquired values will persist until either the module is reset or another acquisition cycle is requested. Note that each register contains a copy of the MONITOR register in order to determine whether its values are valid. Note also that these registers are *read-only*.

To illustrate, assume, for example, the current value of the digital (3.3 volt) supply for TEM_{14} should be acquired. As this quantity is a member of group *two*, the MONITOR register would be written with a value of 2. (See Table 9.) Once conversion is complete, its value would be returned in register ADCS_00_06 (discussed in Section 2.6.1), within the field labelled ADC_{01}.

2.6.1 ADCS_00-06 register

This register contains the *first* seven digitized quantities of the group specified by the acquisition cycle requested by writing the MONITOR register described within Section 2.5. The register contains seven 12-bit fields, each field corresponding to a digitized value. The register also contains a ***monitor*** field starting at bit offset 84. This field is a copy of the MONITOR register illustrated in Figure 8 and reflects the state of the current acquisition cycle. Thus, when the acquisition cycle is complete, this field will have its ***acquisition in progress*** field *cleared* and will contain the group corresponding to the acquired quantities. If the register is read while conversion is on-going, the ***acquisition in progress*** field will be *set*. In such a case, the read value has no physical significance. Note: This register is *read-only*.

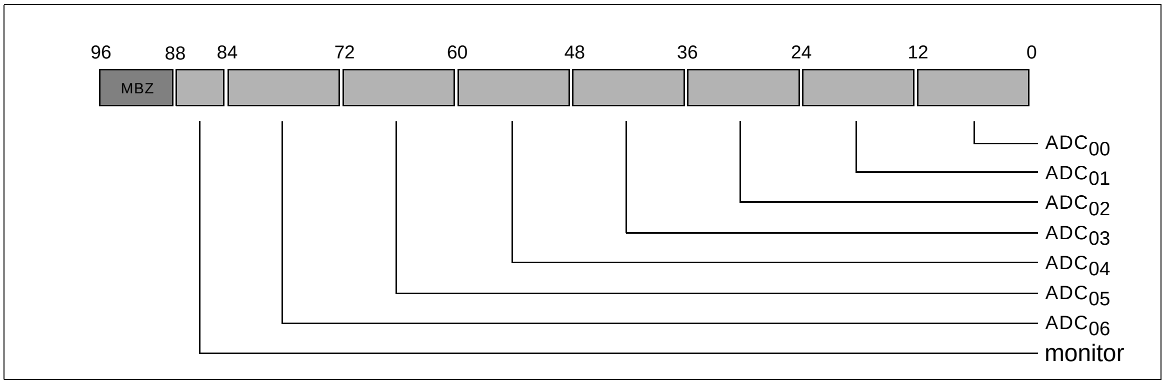

Figure 9 Environmental monitoring result register (00-06)

2.6.2 ADCS_07-13 register

This register contains the *second* seven digitized quantities of the group specified by the acquisition cycle requested by writing the `MONITOR` register described within Section 2.5. The register contains seven 12-bit fields, each field corresponding to a digitized value. The register also contains a ***monitor*** field starting at bit offset 84. This field is a copy of the `MONITOR` register illustrated in Figure 8 and reflects the state of the current acquisition cycle. Thus, when the acquisition cycle is complete, this field will have its ***acquisition in progress*** field *cleared* and will contain the group corresponding to the acquired quantities. If the register is read while conversion is on-going, the ***acquisition in progress*** field will be *set*. In such a case, the read value has no physical significance. Note: This register is *read-only*.

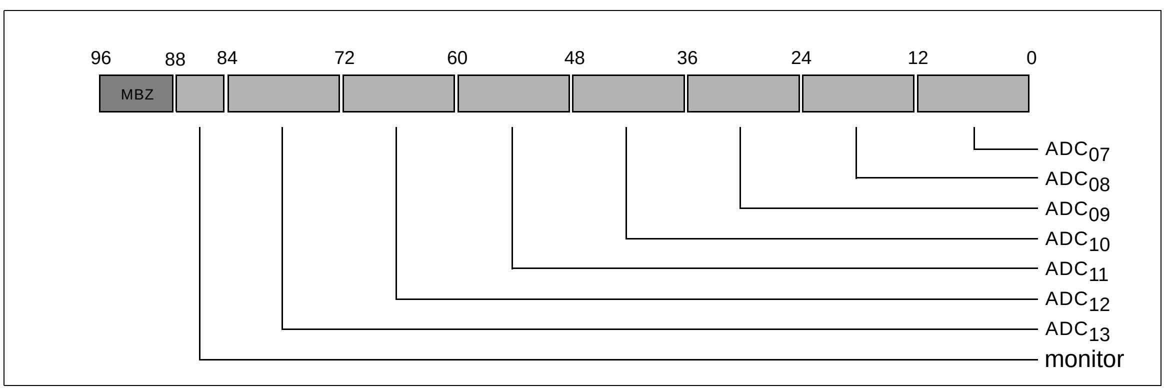

Figure 10 Environmental monitoring result register (07-13)

2.6.3 ADCS_14-19 register

This register contains the *last* six digitized quantities of the group specified by the acquisition cycle requested by writing the `MONITOR` register described within Section 2.5. The register contains six 12-bit fields, each field corresponding to a digitized value. The register also contains a ***monitor*** field starting at bit offset 84. This field is a copy of the `MONITOR` register illustrated in Figure 8 and reflects the state of the current acquisition cycle. Thus, when the acquisition cycle is complete, this field will have its ***acquisition in progress*** field *cleared* and will contain the group corresponding to the acquired quantities. If the register is read while conversion is on-going, the ***acquisition in progress*** field will be *set*. In such a case, the read value has no physical significance. Note: This register is *read-only*.

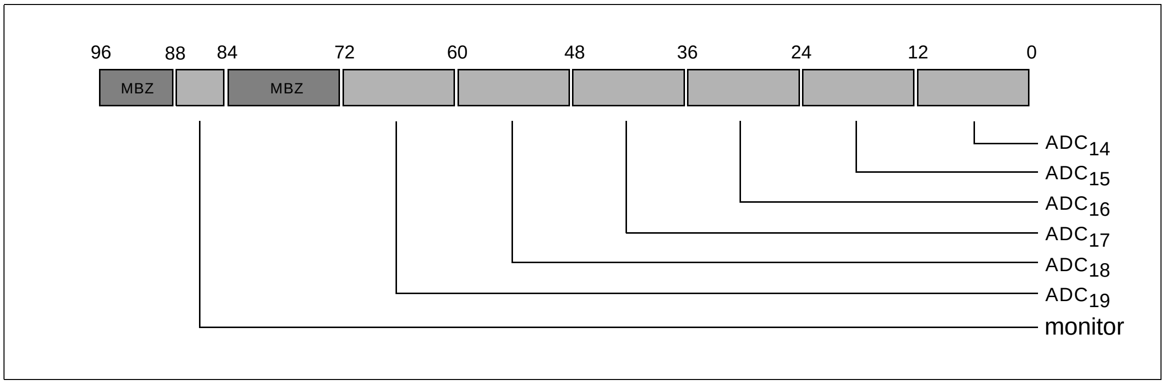

Figure 11 Environmental monitoring result register (14-19)

2.6.4 Groups of Environmental quantities

Table 7 Group *zero* (0) Environmental quantities

Name	Measures:	Conversion constant	ADC #
Spare Voltage (00)	Voltage		00
TEM_{12} Digital (3.3)	Voltage		01
TEM_2 Digital (3.3)	Voltage		02
TEM_8 PCB	Temperature		03
TEM_{12} PCB	Temperature		04
EPU_2	Temperature		05
EPU_0	Temperature		06
TEM_6 PCB	Temperature		07
+*y* VCHP-DSHP Interface (04)	Temperature		08
Spare RTD (00)	Temperature		09
Spare RTD (01)	Temperature		10
Spare RTD (02)	Temperature		11
Spare RTD (03)	Temperature		12
Spare (01)	Temperature		13
+*y* Grid-Radiator Interface (00)	Temperature		14
-*y* Grid-Radiator Interface (00)	Temperature		15
Grid (00)	Temperature		16
Grid (03)	Temperature		17
Grid (06)	Temperature		18
Grid (09)	Temperature		19
Total			**20**

Table 8 Group *one* (1) Environmental quantities

Name	Measures:	Conversion constant	ADC #
Spare Voltage (01)	Voltage		00
TEM_{13} Digital (3.3)	Voltage		01
TEM_{3} Digital (3.3)	Voltage		02
TEM_{9} PCB	Temperature		03
TEM_{13} PCB	Temperature		04
Spare (00)	Temperature		05
EPU_{1}	Temperature		06
TEM_{7} PCB	Temperature		07
+*y* VCHP-DSHP Interface (05)	Temperature		08
+*y* Radiator Antifreeze Heaters (00)	Temperature		09
ACD Composite Shell (00)	Temperature		10
-*y* Radiator Antifreeze Heaters (00)	Temperature		11
ACD Composite Shell (01)	Temperature		12
Spare (02)	Temperature		13
+*y* Grid-Radiator Interface (01)	Temperature		14
-*y* Grid-Radiator Interface (01)	Temperature		15
Grid (01)	Temperature		16
Grid (04)	Temperature		17
Grid (07)	Temperature		18
Grid (10)	Temperature		19
Total			**20**

Table 9 Group *two* (2) Environmental quantities

Name	Measures:	Conversion constant	ADC #
Spare Voltage (02)	Voltage		00
TEM_{14} Digital (3.3)	Voltage		01
TEM_{4} Digital (3.3)	Voltage		02
TEM_{10} PCB	Temperature		03
TEM_{14} PCB	Temperature		04
TEM_{0} PCB	Temperature		05
TEM_{3} PCB	Temperature		06
TEM_{6} Power Supply	Temperature		07
-*y* VCHP-DSHP Interface (00)	Temperature		08
+*y* Radiator (00)	Temperature		09
+*y* VCHP Reservoir Heaters (00)	Temperature		10
-*y* Radiator (00)	Temperature		11
-*y* VCHP Reservoir Heaters (00)	Temperature		12
ACD PMT rail (00)	Temperature		13
+*y* VCHP-XLHP (Radiator-X-LAT) Interface (00)	Temperature		14
-*y* VCHP-XLHP (Radiator-X-LAT) Interface (00)	Temperature		15
Grid (02)	Temperature		16
Grid (05)	Temperature		17
Grid (08)	Temperature		18
Grid (11)	Temperature		19
Total			**20**

Table 10 Group *three* (3) Environmental quantities

Name	Measures:	Conversion constant	ADC #
Spare Voltage (03)	Voltage		00
TEM_{15} Digital (3.3)	Voltage		01
TEM_{5} Digital (3.3)	Voltage		02
TEM_{11} PCB	Temperature		03
TEM_{15} PCB	Temperature		04
TEM_{1} PCB	Temperature		05
TEM_{4} PCB	Temperature		06
TEM_{1} Power Supply	Temperature		07
-*y* VCHP-DSHP Interface (01)	Temperature		08
+*y* Radiator (01)	Temperature		09
+*y* VCHP Reservoir Heaters (01)	Temperature		10
-*y* Radiator (01)	Temperature		11
-*y* VCHP Reservoir Heaters (01)	Temperature		12
ACD PMT rail (01)	Temperature		13
+*y* VCHP-XLHP (Radiator-X-LAT) Interface (01)	Temperature		14
-*y* VCHP-XLHP (Radiator-X-LAT) Interface (01)	Temperature		15
Calorimeter Baseplate (00)	Temperature		16
Calorimeter Baseplate (04)	Temperature		17
Calorimeter Baseplate (08)	Temperature		18
Calorimeter Baseplate (12)	Temperature		19
Total			**20**

Table 11 Group *four* (4) Environmental quantities

Name	Measures:	Conversion constant	ADC #
TEM_8 Digital (3.3)	Voltage		00
EPU_2 Digital (3.3/5 sum)	Voltage		01
TEM_6 Digital (3.3)	Voltage		02
TEM_8 Power Supply	Temperature		03
TEM_{12} Power Supply	Temperature		04
TEM_2 PCB	Temperature		05
TEM_5 PCB	Temperature		06
+*y* VCHP-DSHP Interface (00)	Temperature		07
-*y* VCHP-DSHP Interface (02)	Temperature		08
+*y* Radiator (02)	Temperature		09
+*y* VCHP Reservoir Heaters (02)	Temperature		10
-*y* Radiator (02)	Temperature		11
-*y* VCHP Reservoir Heaters (02)	Temperature		12
ACD PMT rail (02)	Temperature		13
+*y* VCHP-XLHP (Radiator-X-LAT) Interface (02)	Temperature		14
-*y* VCHP-XLHP (Radiator-X-LAT) Interface (02)	Temperature		15
Calorimeter Baseplate (01)	Temperature		16
Calorimeter Baseplate (05)	Temperature		17
Calorimeter Baseplate (09)	Temperature		18
Calorimeter Baseplate (13)	Temperature		19
Total			**20**

Table 12 Group *five* (5) Environmental quantities

Name	Measures:	Conversion constant	ADC #
TEM_9 Digital (3.3)	Voltage		00
Spare Voltage (04)	Voltage		01
TEM_7 Digital (3.3)	Voltage		02
TEM_9 Power Supply	Temperature		03
TEM_{13} Power Supply	Temperature		04
TEM_0 Power Supply	Temperature		05
TEM_3 Power Supply	Temperature		06
+*y* VCHP-DSHP Interface (01)	Temperature		07
-*y* VCHP-DSHP Interface (03)	Temperature		08
+*y* Radiator (03)	Temperature		09
+*y* VCHP Reservoir Heaters (03)	Temperature		10
-*y* Radiator (03)	Temperature		11
-*y* VCHP Reservoir Heaters (03)	Temperature		12
ACD PMT rail (03)	Temperature		13
+*y* VCHP-XLHP (Radiator-X-LAT) Interface (03)	Temperature		14
-*y* VCHP-XLHP (Radiator-X-LAT) Interface (03)	Temperature		15
Calorimeter Baseplate (02)	Temperature		16
Calorimeter Baseplate (06)	Temperature		17
Calorimeter Baseplate (10)	Temperature		18
Calorimeter Baseplate (14)	Temperature		19
Total			**20**

Table 13 Group *six* (6) Environmental quantities

Name	Measures:	Conversion constant	ADC #
TEM_{10} Digital (3.3)	Voltage		00
TEM_{0} Digital (3.3)	Voltage		01
EPU_{0} Digital (3.3/5 sum)	Voltage		02
TEM_{10} Power Supply	Temperature		03
TEM_{14} Power Supply	Temperature		04
TEM_{1} Power Supply	Temperature		05
TEM_{4} Power Supply	Temperature		06
+*y* VCHP-DSHP Interface (02)	Temperature		07
-*y* VCHP-DSHP Interface (04)	Temperature		08
+*y* Radiator (04)	Temperature		09
+*y* VCHP Reservoir Heaters (04)	Temperature		10
-*y* Radiator (04)	Temperature		11
-*y* VCHP Reservoir Heaters (04)	Temperature		12
ACD BEA-Grid Interface (00)	Temperature		13
+*y* VCHP-XLHP (Radiator-X-LAT) Interface (04)	Temperature		14
-*y* VCHP-XLHP (Radiator-X-LAT) Interface (04)	Temperature		15
Calorimeter Baseplate (03)	Temperature		16
Calorimeter Baseplate (07)	Temperature		17
Calorimeter Baseplate (11)	Temperature		18
Calorimeter Baseplate (15)	Temperature		19
Total			**20**

Table 14 Group *seven* (7) Environmental quantities

Name	Measures:	Conversion constant	ADC #
TEM_{11} Digital (3.3)	Voltage		00
TEM_{1} Digital (3.3)	Voltage		01
EPU_{1} Digital (3.3/5 sum)	Voltage		02
TEM_{11} Power Supply	Temperature		03
TEM_{15} Power Supply	Temperature		04
TEM_{2} Power Supply	Temperature		05
TEM_{5} Power Supply	Temperature		06
+*y* VCHP-DSHP Interface (03)	Temperature		07
-*y* VCHP-DSHP Interface (05)	Temperature		08
+*y* Radiator (05)	Temperature		09
+*y* VCHP Reservoir Heaters (05)	Temperature		10
-*y* Radiator (05)	Temperature		11
-*y* VCHP Reservoir Heaters (05)	Temperature		12
ACD BEA-Grid Interface (01)	Temperature		13
+*y* VCHP-XLHP (Radiator-X-LAT) Interface (05)	Temperature		14
-*y* VCHP-XLHP (Radiator-X-LAT) Interface (05)	Temperature		15
Spare (03)	Temperature		16
Spare (04)	Temperature		17
Spare (05)	Temperature		18
Spare (06)	Temperature		19
Total			**20**

Chapter 3
Commanding

3.1 Overview

This chapter describes the remote protocol necessary to access both the registers[1] and functional blocks of the PDU. It follows the Command/Response Protocol discussed in [1]. The registers of the PDU are organized into a hierarchy of *two* blocks as illustrated in Figure 12:

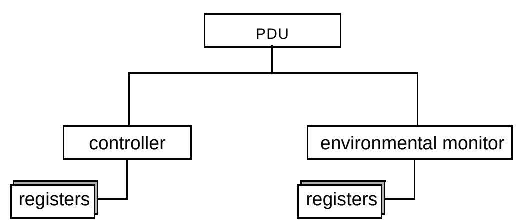

Figure 12 Hierarchy of target types

3.1.1 Conventions

All data structures described in this chapter are from the perspective of being "on-the-wire." Therefore, the left-most field in any description is transmitted *first*, or is considered to be transmitted on the *zeroth* clock. Fields are numbered from the beginning of the ***packet header*** described in [1].

1. Enumerated and described in Chapter 2

3.2 The PDU's access descriptor

Directly following the LATp header of any received command packet is a fixed size, 16-bit structure, called the PDU's ***access descriptor***. This descriptor is a parameterization of the access rules required to address the functional blocks and registers of the PDU. Its structure is illustrated in Figure 13:

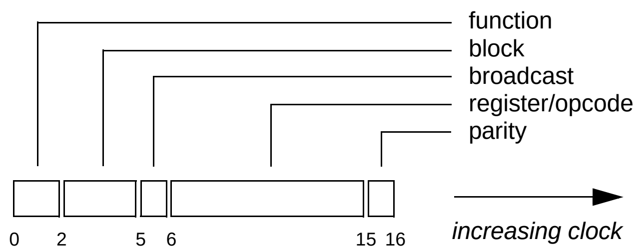

Figure 13 PDU access descriptor

function: Enumerates what *type* of access is required of the target by the command, for example, whether the command will either *read* or *write* the specified register. The valid enumerations for this field are described in [1].

block: This field enumerates which of the two blocks of the PDU are to be accessed. The correspondence between block type and number is enumerated in Table 15.

broadcast: Determines how the ***register/opcode*** field is interpreted. If this field is *false*, the ***register/opcode*** field is used to determine which register to access. If this field is *true*, the ***register/opcode*** field is ignored and the access descriptor is applied to *all* the registers of the type specified by the ***block*** field. Note: A broadcast operation is *not* permitted if the ***function*** field specifies a `read` operation.

register/opcode: If the ***function*** field has a value of either `read` or `load`, this field contains the *number* of the register to be accessed. If the function is `dataless`, this field determines the *type* of dataless access.

parity: The *odd* parity value over the entire ***access descriptor***.

Table 15 Block numbers of the PDU

Name	Number
`controller`	0
`environmental monitor`	1

3.3 Accessing the controller

An enumeration and description of the registers of the controller block may be found in Section 2.3, "PDU controller registers". All registers of the controller are 32-bits in length.

3.3.1 Dataless commands

Table 16 The controller's dataless commands

Name	Opcode	Description
`NOP`	0	No operation
`RESET`	1	Hard reset of the PDU
Total	**3**	

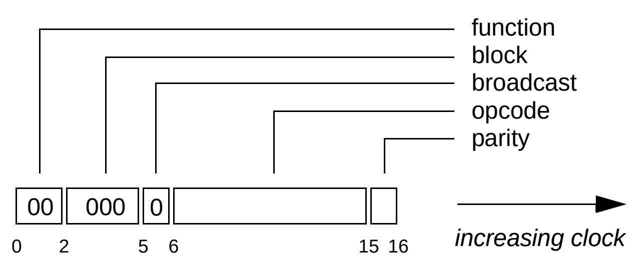

Figure 14 Access descriptor for the controller's dataless commands

`Dataless` functions do *not* require a payload. As a `dataless` function requires no response, the ***respond*** field of the packet is set to *false*.

3.3.2 Load commands

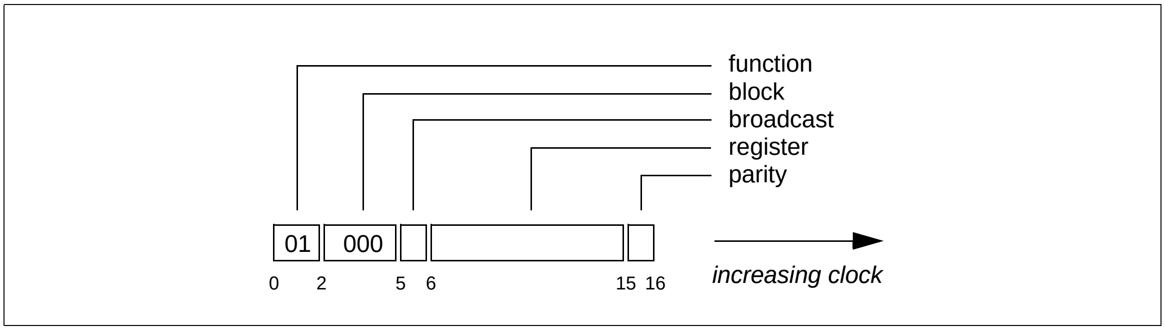

Figure 15 Access descriptor for the controller's register load commands

All registers of the controller are 32 bits long. Consequently, all `Load` functions require a 32-bit payload. The format of this payload is illustrated in Figure 16. As a `Load` function does not require a response, the ***respond*** field of the packet is set to *false*.

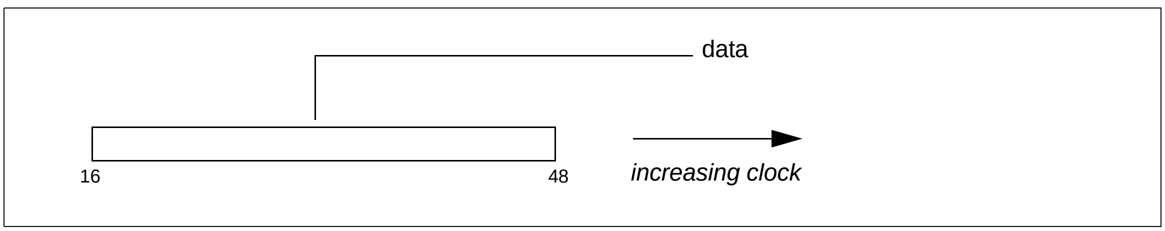

Figure 16 Payload for the controller's register load commands

3.3.3 Read commands

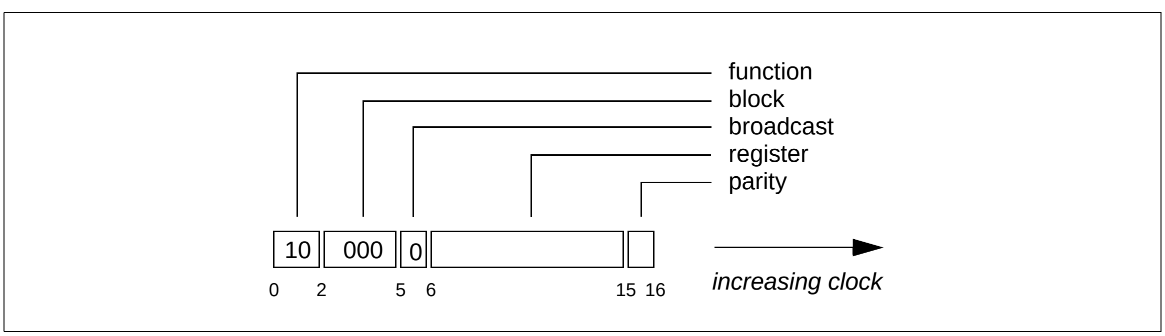

Figure 17 Access descriptor for the controller's register read commands

`Read` functions require *no* payload. The value of the register read is returned as a response. As these `reads` *do* generate a response, the command packet's ***respond*** field is set to *true*. The format of that response is illustrated in Figure 18.

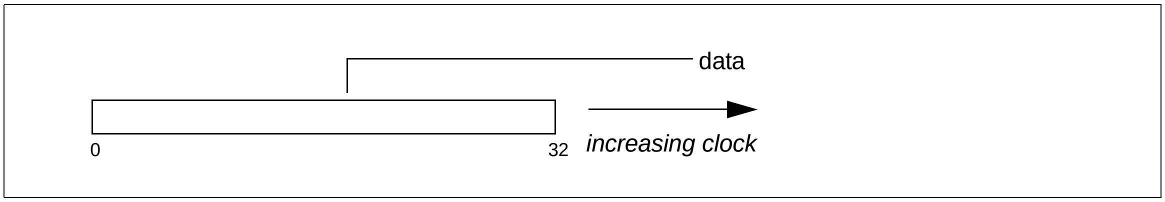

Figure 18 Response to a register read command of the controller

3.4 Accessing the Environmental monitoring block

An enumeration and description of the registers of the environmental block may be found in Section 2.6, "Environmental monitoring". All registers of this block are 112-bits in length.

3.4.1 Dataless commands

None.

3.4.2 Load commands

Not permitted.

3.4.3 Read commands

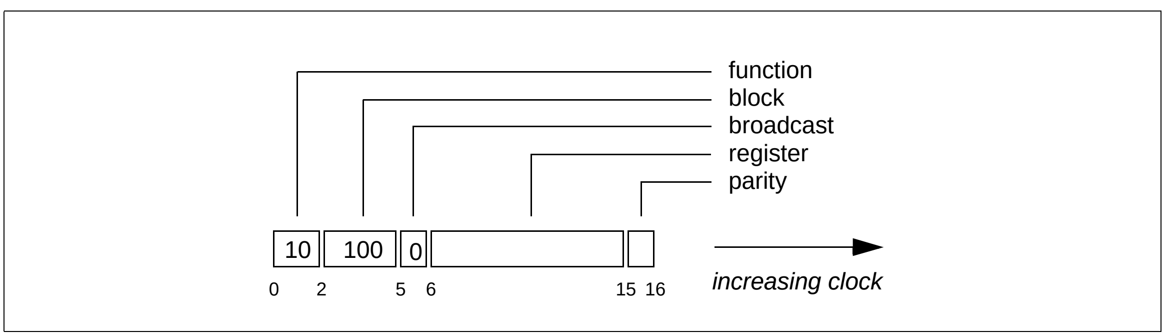

Figure 19 Access descriptor for register read commands of the environmental monitoring block

`Read` functions require *no* payload. The value of the register read is returned as a response. As these `reads` *do* generate a response, the command packet's ***respond*** field is set to *true*. The format of that response is illustrated in Figure 20.

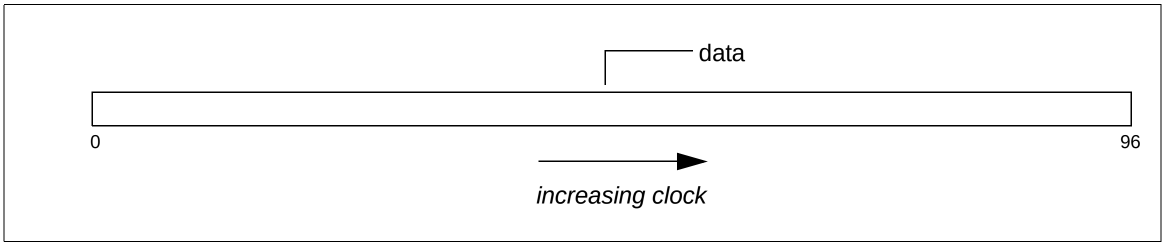

Figure 20 Response to a register read of the environmental monitoring block