LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	**LIGO-T070152-00 - D**	7/29/07

CONFIGURATION EEPROM

Josh Myers, Paul Schwinberg, Daniel Sigg

Distribution of this draft:

all

This is an internal working note
of the LIGO Project.

LIGO Hanford Observatory
P.O. Box 159
Richland, WA 99352
Phone (509) 372-8106
FAX (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (504) 686-3100
FAX (504) 686-7189
E-mail: info@ligo.caltech.edu

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS NW17-161
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

Table of Contents

1 Introduction and Overview

This note describes the serial configuration EEPROM which is used on some LIGO electronics boards as an identify and configuration memory. It contains configuration information to describe board number, revision and serial number. On boards with an FPGA it may also contain initialization data such as an ethernet address. Furthermore. it can contain additional information as needed.

2 Hardware

We are using a standard EEPROM with serial SPI interface. The preferred device is an Atmel AT25256A (Digi-Key AT25256A-10PU-1.8-ND) which can contain up to 32kByte of data and supports a clock rate of up to 10MHz at 3.3V. This device can be powered with any supply between 1.8V and 5V. We are using a DIP8 package with a socket, so that the EEPROM can be programed externally. A typical snippet of a schematics can be seen in Figure 1.

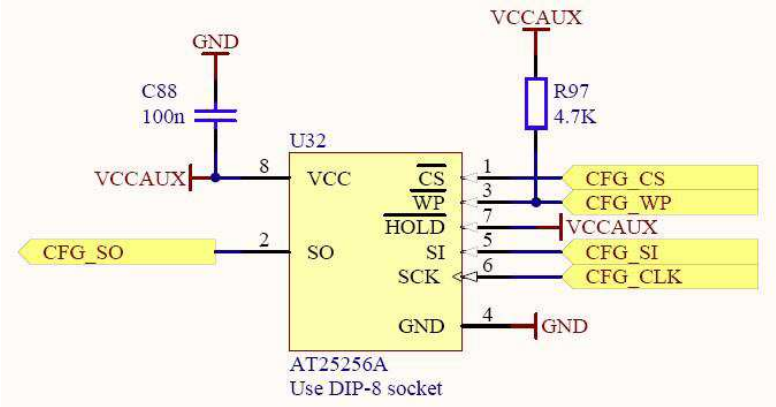

Figure 1: Typical schematics snippet with a configuration EEPROM.

3 Configuration Layout

The layout of the configuration space is shown in Figure 2. There space is divided into 3 areas: an identification area which contains tagged ASCII data, a board space which can be used by the board for storing binary values and a initialization area which is used by FPGA boards to initialize some of its internal registers such as an ethernet address.

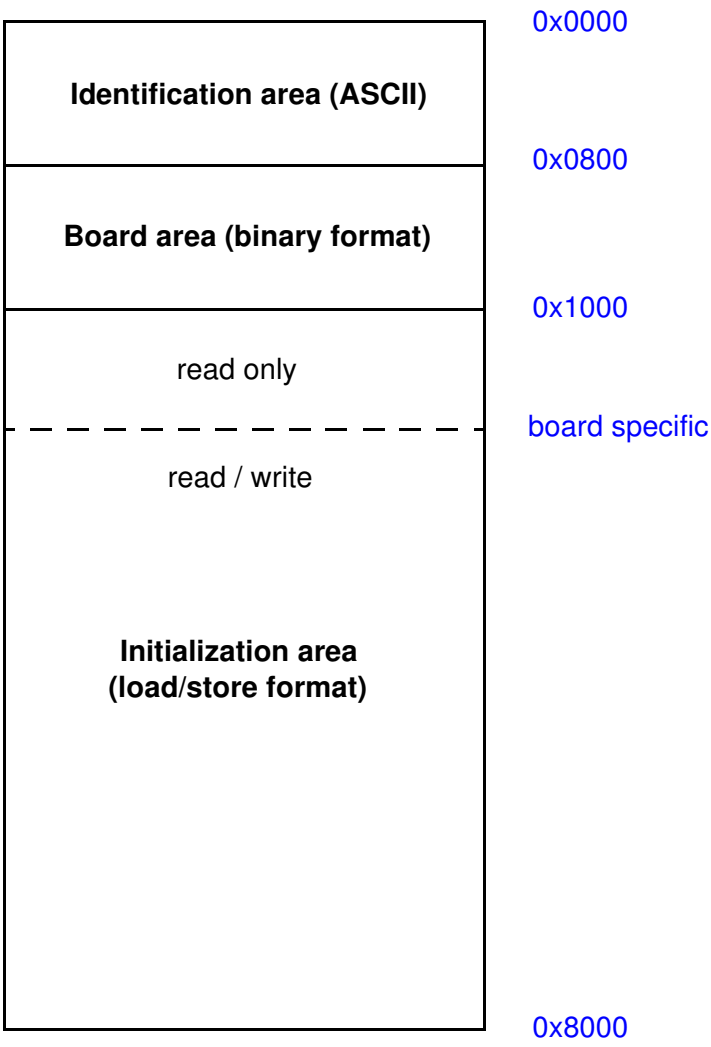

Figure 2: Layout of the configuration space in the EEPROM.

3.1 IDENTIFICATION AREA

The identification area uses a tagged ASCII format similar to XML to store information about the board. It is intended to be interpreted by a back-end computer. The format is a simply a series of <TAG>Value</TAG> constructs. Table 1 lists the predefined XML tags. Additional tags can be defined as needed.

A board which deploys an FPGA may use some of these tags for internal purpose. This is particular important for parameters such as an ethernet MAC address which has to be unique to

Table 1: Predefined XML tags for the identification block.

Tag	Value	Description
<BRD>	070173	Drawing number of the board as submitted to the DCC (6 digits, no 'D' letter)
<REV>	2	Board revision, a single number
<SRV>	1	Optional for boards with a sub revision, number
<BSN>	0023	Board serial number, up to 9 digits
<EA0>	0123456789AB	Ethernet address for EMAC 0
<EA1>	0123456789AB	Ethernet address for EMAC 1
<EA2>	0123456789AB	Ethernet address for EMAC 2
<EA3>	0123456789AB	Ethernet address for EMAC 3

each board. Parameters which are read internally must have an XML tag with exactly three letters. The value will be read as a hex (or BCD) number and stored in a 64 bit field. This means up to 16 digits can be decoded. Illegale characters and leading digits exceeding the 16 digit limit are ignored. XML tags which are not read internally can have more or less than three letters and their value can be any ASCII string. They are only meant to be interpreted by a back end computer or a user.

3.2 Board Area

The configuration stored in the board area is board specific and can be of any format. This is typically only read by the board itself.

3.3 Initialization Area

The initialization area is used by boards with an FPGA to save initialization settings. It contains specific board initialization information such as an ethernet address or which channels are read-out by an ADC board. Some initialization information can be declared read only. For example, an ethernet address would probably be read only and has to be assigned once, whereas the channel configuration of an ADC board may be allowed to change. The format of this area is the same as the payload of a configuration frame as described by T060212. The payload of a configuration frame consists of a set of read and write instructions followed by optional data words. For the initialization area only write instructions make sense; the result of a read will be ignored.

4 Programming

Special utilities and hardware setups will be provided to program a configuration EEPROM. For boards with an initialization area which also implement an ethernet port, the writable part of the initialization space can typically be loaded through ethernet.

5 RECORD KEEPING

For each board the EEPROM configuration is unique, since it contains the DCC number of the board as well as its serial number. It may also contain an unique ethernet address. This means swapping EEPROMs between different boards is strictly disallowed. Since a board repair may require the reprogramming of the EEPROM, each configuration has to be archived in CVS or an equivalent archival system on a board-by-board basis. If there is a writable initialization area, it should also be backed up in case the original board has to be swapped with a new one.

For boards with ethernet connections one or multiple unique ethernet addresses have to be assigned. Special care has to be taken to avoid the same address to be used multiple times. A record of the currently used ethernet addresses has be available in the archive.