Fuse board SW interface specification

2024-07-17

1. Introduction

This document specified the human and machines interfaces used to communicate with DefSecIntel Fuse Board firmware.

2. Overview

Fuse boards are used in a system that consists of multiple Fuse boards where one is configured as a master and others are configured as slave devices. Master Fuse board interfaces with the system (e.g. central controller) over Ethernet and with slave Fuse boards over the RS-485 interface.

From a system control point of view the master and slaves appear as a single device where master is the entry point. It means that when a command is sent to master that concerns slaves, the master communicates with slaves in the background. There are a few exceptions where the system can address slave devices separately, such as when reading device identifiers of slaves.

The communication protocol for all functions except firmware upgrade, shall be Modbus. Master device acts as a Modbus TCP server on Ethernet and Modbus master on RS-485. Slave devices act as Modbus slaves on RS-485 and are silent on Ethernet.

Modbus registers shall be designed in a way that they appear to be in master Fuse board, but they are actually synchronized with the registers in slaves.

Firmware shall be the same for master and slave Fuse boards.

Todo: Add overview drawing

3. Functions

3.1. Device configuration

DIP switches shall be used to determine if the Fuse board operates as a master or slave, what is its slave address and how it gets the IP address.

DIP switch	Function
0	Master / Slave
1	Master IP mode (auto / manual)
2	Slave address bit 0
3	Slave address bit 1
4	Slave address bit 2
5	*Reserved*
6	*Reserved*
7	Enforce bootloader mode

In total 8 unique slave addresses can be used.

Firmware reads DIP switch position once during start-up (and power-up). Therefore restart (re-powering) is required after changing DIP switches.

Bootloader mode enforcing is a back-up option for a worst-case scenario when application firmware is invalid and does not react to bootloader mode entry command.

Todo: add picture of DIP switch and explain switch position.

3.2. Device identification

The following identifiers apply for all Fuse board devices – master and slaves. The identifiers shall be readable like Modbus registers. From a system point of view each device shall be separately addressed.

Parameter	Master/Slave	Description
UID	M, S	STM32 unique ID (96-bits)
MAC	M, S	MAC address of Ethernet interface.
Serial number[1]	M, S	Factory programmed serial number.
Hardware revision[1]	M, S	To identify the hardware on which the software is running at.
Bootloader version	M, S	Currently loaded bootloader version.
Firmware version	M, S	Currently loaded application firmware version.
Number of slaves	M	The number of detected/connected slaves.
Number of outputs	M	The number of detected/connected outputs. It is (1 + slave count) x 8

[1] STM32 internal one-time-programmable (OTP) memory shall be used to store serial number and hardware revision. OTP memory is not erasable or re-programmable.

Explanation:

The system shall first ask the master how many slaves there are. Then it can use determined addresses to ask identification parameters of all the slaves.

3.3. Firmware upgrade

A/B boot shall be used. New firmware image can be downloaded while it continues to operate on the existing one. MCU that is in use (STM32H5) has dual-bank Flash which allows to do it. A reset is required after downloading to make the new firmware operational.

Firmware download protocol is TBD.

Idea:
When master firmware has been upgraded, it will automatically upgrade slaves with the same firmware.

3.4. Configuration functions

The following chapters explain the software controllable configuration options. Configuration options shall be read and written like Modbus holding registers.

3.4.1. Master device configuration options

Following configuration options apply for master:

Parameter	Description
IP address (manual)	IP v4 address. Effective only when DIP switch is configured to use manual IP.

3.4.2. Output based configuration options

The following configuration options apply for master and slaves, but they all go through the master. Modbus addressing is output based, not device based. E.g. output 10 is first slave's second output.

Parameter	Description
Start-up state.	Off / On. Applied with a delay after firmware start-up. *Note: electrically outputs are off when Fuse board is unpowered, or firmware has not yet started.*
Start-up delay.	Time in milliseconds after start-up when to change output state to its start-up state.
Forward current limit	0 to +15A limit for each output.
Reverse current limit	0 to -15A limit for each output.
Activation filter time	Current limit ignoring time (milliseconds) when output is turned on. To filter out spikes caused by in-rush current.
Operation filter time	Current limit ignoring time (milliseconds) when output is already on. To filter out spikes that occur during operation.
Recovery mode	Behavior after the current limit is exceeded and output is turned off for device protection. • No recovery, stay off. Off-on sequence or reset command required to turn on again. • Try to automatically turn it on up to 10 times. Apply delay (recovery off time) between retries. After retries, stay off. Controlled off-on or reset required. *Note: must have a limit on retries to increase relay life.*
Recovery off time	Time to wait in milliseconds before trying to turn the output on again.
Reset off time	Time to wait in milliseconds in off state when doing output automatic reset.

Question: Have output under- and overvoltage detection also?

3.5. Operation functions

Operation shall be performed by reading and writing coil registers.

3.5.1. Device (master, slave) operation functions

Function	R/W	Description
Restart	W	Restart the Fuse board. If firmware has been upgraded, it will become effective.
Uptime	R	Get uptime of firmware. Time in seconds since last start-up.

3.5.2. Output based operations

These operations are performed through the master. When an output is addressed that is on the slave (outputs beyond 8), then master forwards the operation to the slave.

Function	R/W	Description
Output on request	W	Turn on output.
Output off request	W	Turn off output.
Output reset request	W	Do automatic off and on sequence of output.
Read output request	R	Returns output requested state (off or on). Reset request is treated as request to turn on.
Read output state	R	Returns output actual state: • Off • Off, but voltage detected at output. • On • On, but no voltage detected (fuse burned?)
Read output state reason	R	Returns latest reason for output state: • Starting-up • Start-up default • Control request • Reset request • Protection (current limit exceeded) • Recovery
Read output fault flags	R	Returns faults that have occurred since device power-up or last fault flags clearing. Flags are: • Reverse current has been exceeded and output has been turned off. • Forward current has been exceeded and output has been turned off. • Voltage detected at turned off output (wiring issue?).

		• Absence of output voltage (fuse burned?) detected.
Reset output fault flags	W	Clear all output fault flags.
Read output voltage	R	Returns output voltage (V).
Read output current	R	Return output current (A).
Read output power	R	Returns output power (W).
Read output energy	R	Return output energy (J) since turning it on.
Read switch	R	Read push-button switch state.
Output LED control	W	Temporary LED control. Color: 1-bit for R, G and B. Time: 5-bits * 500ms (hence 0.5 to 16 seconds). Automatically restores LED normal operation after given time.

3.6. Troubleshooting functions

Ideas:

- Have a telnet port to get devices human readable log output.
    - Master needs to combine its own and slave's logs.
- Store errors and critical faults in device non-volatile memory and have a (Modbus) interface to read them out.