Fuse board SW interface specification v1.2

2025-03-24

1. Introduction

This document specified the human and machines interfaces used to communicate with DefSecIntel Fuse Board firmware version 1.0.0.0.

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.

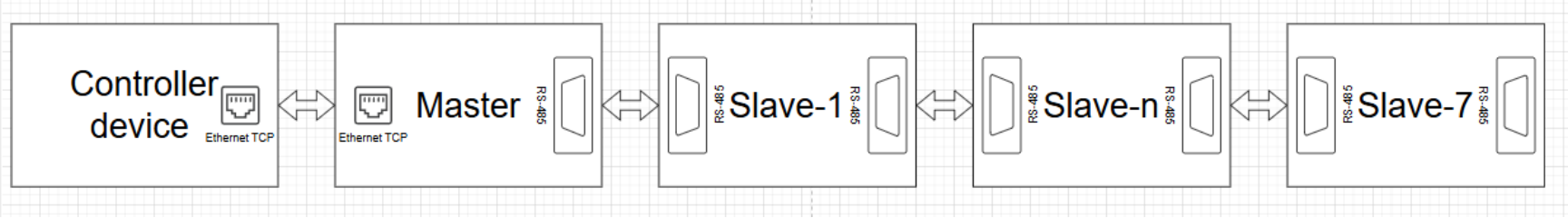
From a system control point of view the master and slaves appear as a single device where master is the entry point. It means that when a command is sent to master that concerns slaves, the master communicates with slaves in the background. There are a few exceptions where the system can address slave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

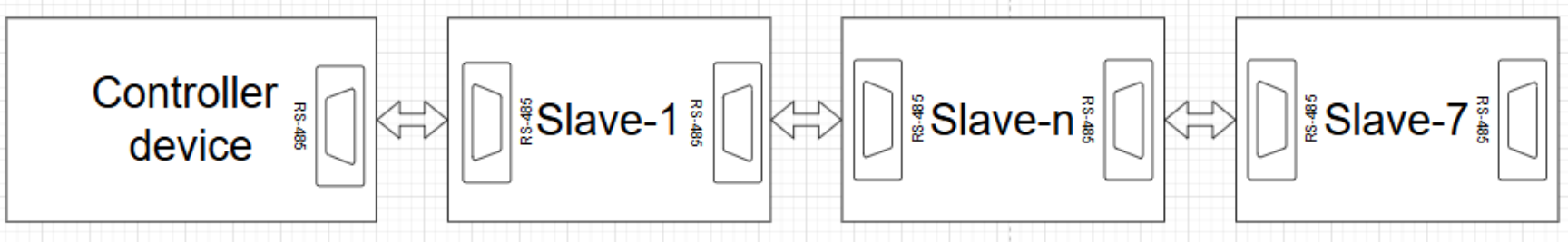
Modbus registers shall be designed in a way that they appear to be in master Fuse board, but they are actually synchronized with the registers in slaves.

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

Controller device connected to slaves via master fuseboard via Ethernet TCP.



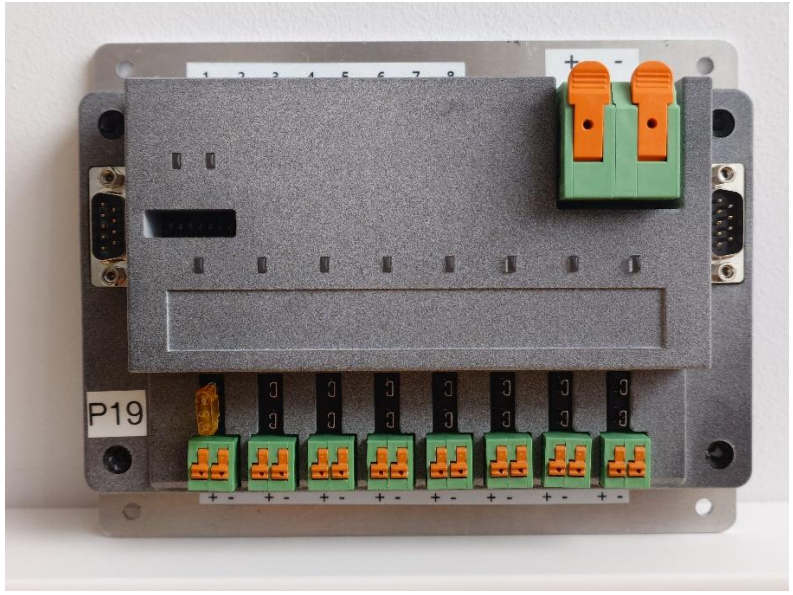
Controller device connected directly via RS-485 to slaves.

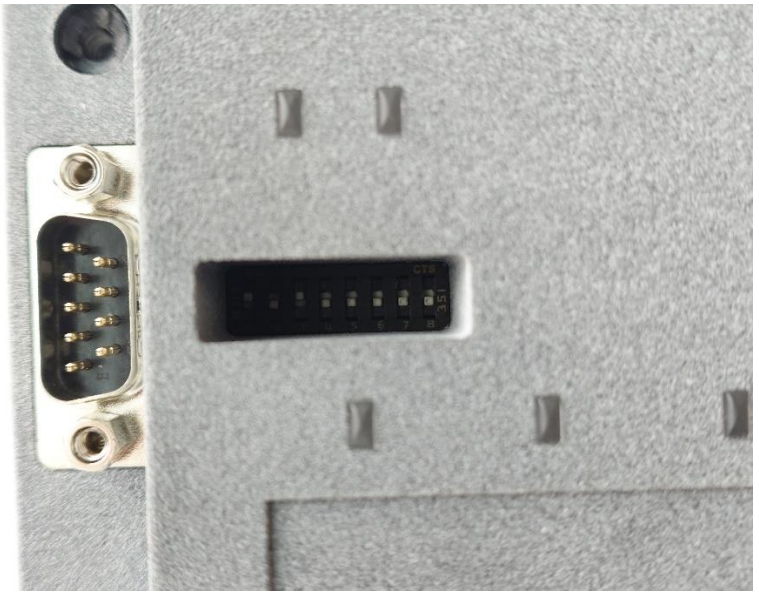


3. Functions

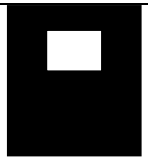
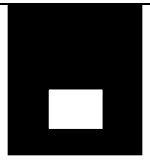
3.1. Device configuration

Primary operational mode configuration is defined with DIP switches on the PCB. DIP switches are placed on the left side of the PCB/device. When the cover is on, they are in a sinked hole:





DIP switches are counted from left to right, from 0 to 7. Their white plastic sliders can be switched up or down.

Switch up:			Switch down:	

DIP switches functions:

Switch:	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Function:	**Mode**	**IP mode**	**Slave count (when mode = master) Slave address (when mode = slave)**			**Reserved**	**Reserved**	**Reserved**
			Bit 2	**Bit 1**	**Bit 0**			
Up:	Master	Static	See table below					
Down:	Slave	DHCP						

Slave count and slave address bits encoded values are:

Switch:		**2**	**3**	**4**
Slave count	**Slave address**	**Bit 2**	**Bit 1**	**Bit 0**
0	1[1]	Down	Down	Down
1	1	Up	Down	Down
2	2	Down	Up	Down
3	3	Up	Up	Down
4	4	Down	Down	Up
5	5	Up	Down	Up
6	6	Down	Up	Up
7	7	Up	Up	Up

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

[1] Address zero is reserved for master only, so in this combination slave address “zero” is 1.

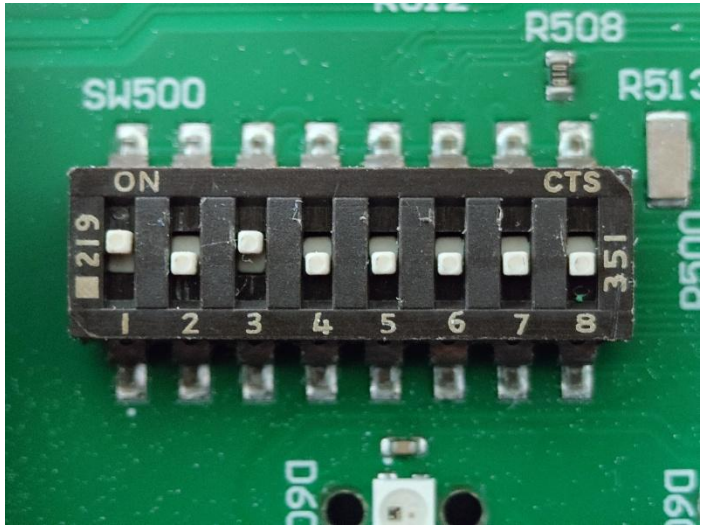


Figure 1 Example of DIP switches set up to be "master" mode, using DHCP for IP and with "Slave Count" being 1

3.2. Modbus addressing

Each device on a Modbus network is assigned a unique Unit ID (ranging from 0 to 255), which is configured using the DIP switches mentioned earlier. Unit ID is used to send data to each device separately.

Modbus uses 16-bit addressing to read or write data to Modbus devices. In this document, all addresses are written in hexadecimal format, following the C language convention, with a 0x prefix (e.g., 0x01FF).

Some addresses include a "P", which represents a port number on the device. The port value is a hexadecimal number ranging from 0 to E (0x0 to 0xE), but it cannot be 0xF due to address limitations. Additionally, the port number must not exceed the actual number of physical ports available on the device.

Addresses for functions were selected according to the Modbus addressing reference. Hex addresses were selected between those address ranges in a way to handle them more easily in code, especially in terms of bits.

3.3. Device identification

The following read-only identifiers apply for all Fuse board devices – master and slaves. The identifiers shall be readable as Modbus holding registers with function 0x03. Each device is separately addressed with device specific Unit ID. The first two identifiers are special and apply for master only.

Parameter	Description	Master/Slave	Address	Quantity	Format
Number of slaves	Number of slaves (0-7)	M	0xA000	1 (0.5)	U8. MSB is zero.
Number of outputs	Number of outputs	M	0xA001	1 (0.5)	U8. MSB is zero.
UID	STM32 unique ID (96-bits)	M, S	0xA002	6	Binary. 12 bytes.
MAC	MAC address of Ethernet interface.	M, S	0xA008	3	Binary. 6 bytes.
Reserved		M, S	0xA00B	5	
Reserved	Future serial number expansion	M, S	0xA010	2	
Serial number[1]	Device unique serial number.	M, S	0xA012	2 (1.5)	24-bit unsigned big-endian integer. MSB is zero.
Hardware revision[1]	To identify the hardware on which the software is running at.	M, S	0xA014	1 (0.5)	U8. MSB is zero.
Reserved			0xA015	11	
Bootloader version	Currently loaded bootloader version. *(reserved)*	M, S	0xA020	8	ASCII, zero delimited
Firmware version	Currently loaded application firmware version.	M, S	0xA028	8	ASCII, zero delimited

[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.

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

3.4. Configuration functions

The following chapters explain the software controllable configuration options. Configuration options shall be read and written like Modbus holding registers. Function 0x03 is to read single holding register, function 0x06 is to write single holding register.

3.4.1. General device configuration options

The following configuration registers apply for master.

Parameter	Description	Address	Format
IP address (manual)	IP v4 address. Effective only when DIP switch is configured to use manual IP.	0xA800	Third (MSB) and fourth (LSB) byte of IP address
		0xA801	First (MSB) and second (LSB) byte of IP address
Reserved	Reserved for future parameters	0xA802 – 0xA80F	Write zeroes

3.4.2. Output based configuration options

The following configuration options apply for master and slaves, but they all go through the master.

Parameter	Description	Address (W)	Format	Unit
Normal state	Relay normal state: NO (normally open) or NC (normally closed). This is determined by hardware.	0xBP00	0 – NO 1 – NC	
Start-up state	Output off / On. Applied with a delay after firmware start-up. *Note: electrically outputs are off (NO) when Fuse board is unpowered, or firmware has not yet started.*	0xBP01	0 - Off 1 - On	
Start-up delay	Time in milliseconds after start-up when to change output state to its start-up state.	0xBP02	U16 big-endian	ms

Forward current limit	0 to +15A limit for each output.	0xBP03	U16 big-endian	mA
Reverse current limit	0 to -15A limit for each output. Limit is expressed in positive numbers.	0xBP04	U16 big-endian (positive value)	mA
Activation filter time	Current limit ignoring time (milliseconds) when output is turned on or off. To filter out spikes caused by in-rush or back-EMF currents.	0xBP05	U16 big-endian	ms
Operation filter time	Current limit ignoring time (milliseconds) when output is already on. To filter out spikes that occur during operation.	0xBP06	U16 big-endian	ms
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.*	0xBP07	0 – No recovery 1 – Retry once 2 – Retry twice … 10 – Retry 10 times	
Recovery off time	Time to wait in milliseconds before trying to turn the output on again.	0xBP08	U16 big-endian	ms
Reset off time	Time to wait in milliseconds in off state when doing output automatic reset.	0xBP09	U16 big-endian	ms
Reserved	Reserved for future parameters	0xBP0A – 0xBP0F	Write zeroes	

Reminder: P is the number of port to be configured

3.5. Operation functions

Operation shall be performed by reading and writing coil registers. Modbus read coil function is 0x01 and write multiple coils function is 0x0f.

3.5.1. Device (master, slave) operation functions

Function	R/W	Description	Address	Quantity	Format
Restart	W	Restart the Fuse board. If firmware has been upgraded meanwhile, it will become effective.	0x2000	8	0x55 – Restart addressed device. 0x66 – Restart all devices (master only).
Config change	W	To control all configuration parameters (holding registers) at once.	0x2001	8	0xAA – Reset configuration (all zeroes). 0xBB – Set current configuration as factory default. 0xCC – Restore factory default configuration. 0xDD – Save current configuration to Flash so it is remembered at next boot.
Reformat /reinit flash	W	To reformat and reinitialize non-volatile flash memory	0x2002	8	0x11 – Make immediate MCU reset and format file system at start-up.

Can be accessed by using function read input registers 0x04

Function	R/W	Description	Address	Quantity	Format
Uptime	R	Get uptime of firmware. Time in seconds since last start-up.	0x9000	2	U32 big-endian. Unit: seconds.

3.5.2. Output based operations

These operations are performed through the master. When an output is addressed that is on the slave, the master forwards the operation to the slave.

Can be accessed by using function read coil 0x01, write coil 0x05 or write multiple coils 0x0f.

Function	R/W	Description	Address	Quantity	Format
Output control	W	Output control request. Allows to set static state Note: In each byte, each bit is one port Each address is one bit	0x001P	1 – 16 (up to port count)	0 – Turn off 1 – Turn on

Can be accessed by using function read input registers 0x04. Multiple addresses can be accessed with one modbus message by setting the first address of the first function to be accessed and then setting quantity for as many addresses as are wanted to be accessed.

Function	R/W	Description	Address	Format
Output state	R	Returns output actual state. Return state 3 (output on, no voltage) could mean that fuse is burned. Return state 2 indicates some wiring problem.	0x8P00	0 – Output is off 1 – Output is on 2 – Output is off, excessive current detected. 3 – Output is on, no voltage detected (fuse problem). 4 – Relay reset ongoing

				5 – Over-/Undervoltage recovery in progress 6 – Recovery attempts exceeded
Output state reason	R	Returns last reason for output being in the state it is.	0x8P01	0 - Starting-up 1 - Start-up default 2 - Control request 3 - Button press 4 - Reset request 5 – Configuration change 6 - Protection (e.g. current limiter) 7 - Recovery (after protection) 8 – Recovery ended (no more attempts)

Output fault flags	R	Returns faults that have occurred since device power-up or last fault flags clearing.	0x8P02	Bitmasks: 1 – Forward current exceeded 2 – Reverse current exceeded 4 – Excessive voltage detected 8 – Absence of voltage detected
Read output voltage	R	Returns output voltage (mV).	0x8P03	U16 big-endian Factor: x1, Unit: mV
Read output current	R	Return output current (mA).	0x8P04	S16 big-endian Factor: x10, Unit: mA
Read output power	R	Returns output power (mW).	0x8P05	S16 big-endian. Factor: x100, Unit: mW
Read output energy	R	Return output energy (J) since turning it on.	0x8P06	U64 big-endian Bits 48-63 Factor: x1, Unit: J
			0x8P07	U64 big-endian Bits 32-47 Factor: x1, Unit: J
			0x8P08	U64 big-endian Bits 16-31 Factor: x1, Unit: J
			0x8P09	U64 big-endian

				Bits 0-15 Factor: x1, Unit: J
Read switch	R	Read push-button switch state.	0x8P0A	0 – Not pressed 1 – Pressed

Reminder: P is the number of port to be configured

Can be accessed with function write multiple coils 0x0f.

Function	R/W	Description	Address	Quantity	Format
Output LED control	W	Temporary LED control. Automatically restores LED normal operation after given time.	0x1P00	24	Byte 0-1: RGB565 color. Byte 2: Time in x0.1s.
Output self-reset request	W	Make an automatic off-on sequence (useful when system asks for self-reset).	0x1P03	8	0 – Normal 1 – Off-on sequence
Output fault flags reset	W	Output fault(s) can be cleared by writing the specific bitmask(s). Fault flags cannot be set.	0x1P04	8	Bitmasks: 1 – Forward current exceeded 2 – Reverse current exceeded 4 – Excessive voltage detected 8 – Absence of voltage detected

3.6. State indication

Each device has two main RGB LEDs and 8 port specific RGB LEDs.

Power indicator (left LED):

State	LEDs
Device off	Stable black
Device on (heartbeat)	Blinking red

Device ethernet connection indicator (right LED):

State	LEDs
Device off	Stable black
Ethernet disconnected	Stable red
Ethernet link up, acquiring IP address	Blinking yellow
Ethernet link up, got IP	Stable blue
Modbus TCP/IP connection established	Stable green

Note: *In the case of slave device, Ethernet stays functional. As Ethernet is not to be used in the case of slave device (using RS485 instead), the indicator staying red is expected and device is functional. This applies for FW version 1.0.0 and it may change in the future version.*

Device ports indicators (LED 1 to 8):

State	LEDs
Output off	Stable black
Output on, has voltage	Stable green
Output on, overcurrent or undercurrent	Stable orange
Output on, no voltage (fuse burned)	Stable red
Output off, excessive voltage	Blinking yellow
Output doing reset	Stable blue
Output in protection mode	Blinking red

3.7. Device configuration levels and boot-up process

During boot-up, the device checks for configurations in the following order:

1. **Live Configuration** – Stored in the file system.
2. **Factory Default Configuration** – Also stored in the file system.
3. **Firmware Default Configuration** – Hardcoded into the firmware.

If a configuration is missing or corrupted at a given level, the next available configuration is used.

3.7.1. Behavior on a blank device

If the device has newly uploaded firmware and the file system has not been formatted, no configurations exist in the file system. In this case, the **firmware default configuration** is used.

3.7.2. Behavior after formatting the file system

After formatting and rebooting, if no configuration files are found in the file system, the device again uses the **firmware default configuration**. However, at this point, the **firmware default configuration is also saved as the live configuration**, ensuring it is used after future reboots.

3.7.3. Managing configurations

Current running configuration can be modified using functions in **section 3.5**. Changes take effect immediately but are **not automatically saved** for use after future reboots.

To make these changes persistent, functions in **section 3.6.2** are to be used to save the **current running configuration** as either the **live configuration** or the **factory default configuration**.

Unlike the live configuration, the **factory default configuration can be restored** at any time to become the current running configuration.

3.7.4. Firmware default configuration parameters

The **firmware default configuration** includes the following predefined values:

- **Network Configuration:**
  - Default IP Address: 192.168.1.100
- **Current Limits:**
  - Forward Current Limit: 15000 mA
  - Reverse Current Limit: 1500 mA
- **Filter Times:**
  - Activation Filter Time: 200 ms
  - Operation Filter Time: 200 ms
- **Device States:**
  - Normal State: 0
  - Startup State: 0
  - Startup Delay: 500 ms
- **Recovery & Reset:**
  - Recovery Mode: 0
  - Recovery Off Time: 1000 ms
  - Reset Off Time: 2000 ms

These values are used when no live or factory default configuration is available.