ModatProgrammer

USER GUIDE

Gabdelgaziz Sayfutdinov

11.01.2021

Table of Contents

Introduction

The ModatProgrammer tool is used to program devices over different interfaces. Currently available interfaces are:

- SWI (Single Wire Interface)
- I2C
- RS485
- UART

The ModatProgrammer can be started by using the Executable file:
`modat_programmer_tool.exe`

Available arguments for programming tool

To properly use the `modat_programmer_tool`, the executable file has to be started using additional argument which are explained in more detail in the following sections.

The commands are described using a table, where the table columns have to following meaning. Below is an example for the `interface` argument.

Argument	Value	
--interface	i2c	Required

This means, that the executable file gets the following argument: **`--interface=i2c`**

The arguments are marked with the following flags:

Required	The arguments have to be set for a successful execution
Not required	This argument has to be set depending on the context.

Command call order

Each additional for the `modat_programmer_tool` must follow the following pattern.

1. **System settings**
2. **Interface settings**
3. **Read/write command**

System settings

The following system settings are currently available.

--port	COMx: COM Port to which the ModatProgrammer is connected	Required

Interface settings

Before writing or reading an interface has to be selected. Depending on the interface, additional interface settings have to configured.
Note: To ensure proper functionality the order of the arguments should be kept as given.

SWI interface

--interface	swi	Required

I2C interface

--interface	i2c	Required
--set_device_addr	Device addr of I2C device in HEX e.g. 0x50	Required
--set_device_type	Choose one of the following device types: - eeprom24lc16 - eeprom24lc128 - eeprom24lc256	Required

UART interface

To be implemented …

RS485 interface

To be implemented …

Read/Write command

Write command

--write	Define address at which the data should be written	Required
--data_type	"string" … write a string "modat" … write a modat by filling modat	Required

If as data_type string is chosen, the data can be added with the following argument

--data	Data to be transmitted as string	Required

If the data_type is chosen to be modat, the following arguments have to be filled for the THR1AFE_BO module. Other modules will be added in the future.

--name	Name of module	Required
--sno	Serial number of the module	Required
--revision	Revision of module	Required
--module_comp_version	Module compatibility version	Required
--modat_header_only	`True` or `False`: Only modat header is written	Not required
--major	Major version of module	Required
--minor	Minor version of module	Required

Read command

--read	Define address at which the data should be read	Required
--count	Number of bytes to read	Required

Config file

Instead of having to input the data into the using arguments, some configurations can be set in the `config.ini` file located in the config folder.
All parameters in the config file
The following can be set with the `config.ini` file:

Available sections

System settings

In the SystemSettings you can set the serial PORT.

GPIO settings

With the GPIOSettings you can set different pins/flags such as VCC or UART_DBG_EN pin. Not actively used yet.

Interface settings

With InterfaceSettings you can set the interface, DUT address, etc. For more see user guide.

Modat

In the Modat section you can set modat properties.

Available argument

The config file can be filled with the following arguments.

Argument	Example values	Description
port	COM4	Serial port
interface	swi / i2c	Selected interface for programming
data_type	Modat	Data type for written data
data_addr	0	Address of memory to write to
header_only	True or False	Writing only header of modat/msgpack
name	THR1AFE_BO	Name of module
revision	2	Revision of module
module_comp_version	1	Module compatibility version
major	1	Major version of module
minor	1	Minor version of module

Examples of config files for different modules (e.g. THR1AFE_BO) can be found in the Command references chapter.

Command references

This section contains examples of how to fill the arguments for given modules.

THR1AFE_BO

The THR1AFE_BO Module requires the following commands for writing a modat header to the SWI EEPROM. The data is filled starting from address 0.

Assuming the following data:
Module name: THR1AFE_BO
Revision: 2
Comp Version: 1
Serial number: 123456789

Manual input

The `modat_programmer_tool.exe` has to be executed using the following arguments.

```
C:\modat_programmer_tool.exe --port=COM4 --interface=swi
  --write=0 --data_type=modat --modat_header_only=True
  --name=THR1AFE_BO --sno=123456789 --revision=2
  --module_comp_version=1
```

With Config file

To keep the arguments short, the config file can be used. Below is an example how the config file for the THR1AFE_BO looks like:

```
[SystemSettings]
port: COM4

[GPIOSettings]
# vcc_on_off: True -> not used yet

[InterfaceSettings]
interface: swi
data_type: modat
data_addr: 0x0
header_only: True

# define modat data_addr
[Modat]
name: THR1AFE_BO
revision: 2
module_comp_version: 1

```

The only argument to put into the modat_programmer_tool is the serial number which is results in the following function call.

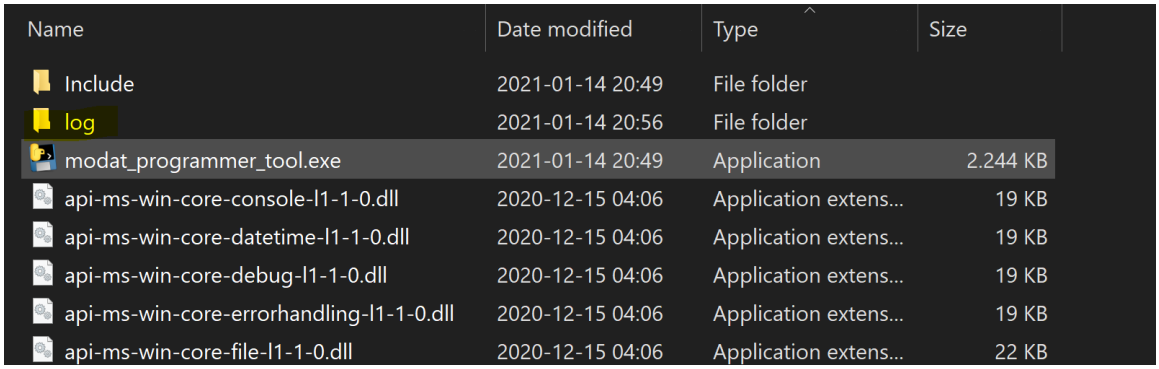
```
C:\modat_programmer_tool.exe --sno=123456789

```

To check the written data, it is possible to read the values from the EEPROM. The size of the written data is fixed to 64 bytes. Therefore, by using the following arguments, the data can be read.

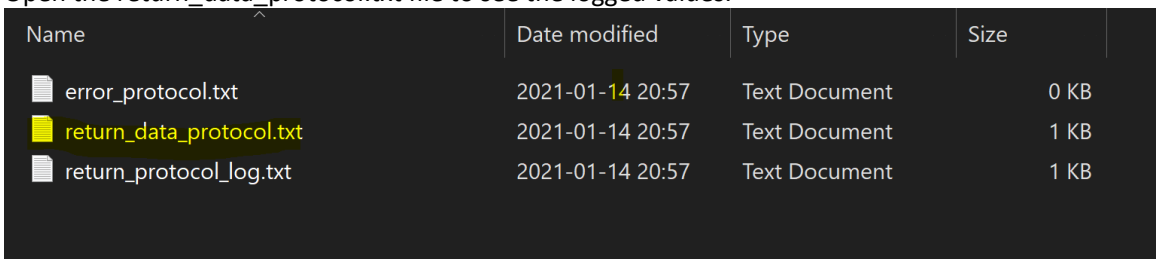
```
C:\modat_programmer_tool.exe --port=COM4 --interface=swi
 --read=0 --count=64
```

The read values can be seen in the log files which are located in the same directory as the `modat_programmer_tool.exe` file in the “log” folder.



Name	Date modified	Type	Size
Include	2021-01-14 20:49	File folder	
log	2021-01-14 20:56	File folder	
modat_programmer_tool.exe	2021-01-14 20:49	Application	2.244 KB
api-ms-win-core-console-l1-1-0.dll	2020-12-15 04:06	Application extens...	19 KB
api-ms-win-core-datetime-l1-1-0.dll	2020-12-15 04:06	Application extens...	19 KB
api-ms-win-core-debug-l1-1-0.dll	2020-12-15 04:06	Application extens...	19 KB
api-ms-win-core-errorhandling-l1-1-0.dll	2020-12-15 04:06	Application extens...	19 KB
api-ms-win-core-file-l1-1-0.dll	2020-12-15 04:06	Application extens...	22 KB

Open the return_data_protocol.txt file to see the logged values.



Name	Date modified	Type	Size
error_protocol.txt	2021-01-14 20:57	Text Document	0 KB
return_data_protocol.txt	2021-01-14 20:57	Text Document	1 KB
return_protocol_log.txt	2021-01-14 20:57	Text Document	1 KB