

# **IO-Flasher – STM8**

## **User Manual**

**Rev. 1.2**

**Date: 07 March 2019**

**[www.iottrue.com](http://www.iottrue.com)**

**[info@iotue.com](mailto:info@iotue.com)**

# **1. Introduction to IO-FLASHER Series**

## **1.1 IO-FLASHER Overview**

IO-FLASHER is the series or stand-alone programmer solution from IoTrue Micro-solutions. The IO-FLASHERS are designed to fulfil the demands of a production environment. It is a Stand-Alone Programmer that can be operated manually without the host PC. Optionally it can be use with host PC. In manual Stand-Alone Programmer mode, the unit is operated using buttons. LED or LCD is available on the programmer to help the operator with optimized indication or menu. IO-FLASHER can communicate with Host PC over USB interface. IO-Flasher comes in many variants to meet the industry requirement of low cost and reliable programming solution. Below are the members of the IoTrue IO-Flasher series:

- ❖ IO-FLASHER – STM8 (Available)
- ❖ IO-FLASHER PRO – STM8 (Available)
- ❖ IO-FLASHER PRO – STM8/ISOL (On Request)
- ❖ IO-GANG FLASHER – STM8 (On Request)
- ❖ IO-FLASHER PRO – STM32 (Coming Soon)

### 1.1.1 IO-Flasher – STM8

IO-FLASHER – STM8 is a programming tool for STM8 series microcontroller with on-chip FLASH, EEPROM and Option Byte memory with STM8 core. IO-FLASHER – STM8 is designed to flash the target device in stand alone mode or host controller mode.

#### Features of IO-Flasher STM8

- ❖ Standard Programming Connector: - Standard 4-Pin SWIM connector compatible with STMicroelectronics STM8 Eval board programming connectors.
- ❖ Offline Mode: - Stand-alone SWIM programmer (Once set up, IO-Flasher can be controlled without the use of PC program). Programming starts with simply button press.
- ❖ Online Mode: - Programming can be done directly from IO-FLASHER GUI.
- ❖ Dual Mode: - While IO-FLASHER is configured as offline mode. Its can be still used in online mode for same or different firmware code.
- ❖ Programmable Power Output: - No power supply required, Powered through USB. Target board can be power from IO-FLATHER. Output voltage can be adjusted from 0 Volt to ~5Volt depending on selected target device (up to 200mA to target with overload protection).
- ❖ Target Power Sensing: - If target is self-powered or powered from IO-FLASHER, then it can be measured by the IO-FLASHER.
- ❖ In-built Protection: - All external port protected with ESD, Short circuit and over current. Isolation is available in Isolated version of IO-FLASHER (e.g. IO-FLASHER - STM8/ISOL)
- ❖ Programming Options: - Have many programming options to program Flash, EEPROM, Option memory with auto lock feature.
- ❖ Programming Indication: - IO-Flashers has LED indication for ready, Programming complete and error state
- ❖ Supported devices: - Support for all STM8 devices
  - STM8S
  - STM8L
  - STM8AF
  - STM8AL
- ❖ IO-FLASHER GUI: - Highly professional GUI with simple and user-friendly interface.



Figure 1

### 1.1.2 IO-Flasher PRO– STM8

IO-FLASHER PRO is the advance version of IO-FLASHER having **LCD DISPLAY**. It has all the features of IO-FLASHER – STM8 with below features in addition.

#### Features of IO-Flasher PRO - STM8

- ❖ Standard Programming Connector: - Standard 4-Pin SWIM connector compatible with STMicroelectronics STM8 Eval board programming connectors.
- ❖ Offline Mode: - Stand-alone SWIM programmer (Once set up, Flasher can be controlled without the use of PC program). Programming starts with simply button press.
- ❖ Online Mode: - Programming can be done directly from IO-FLASHER GUI.
- ❖ Limit Counter: - Limit counter can set to stop offline programming after counter reached.
- ❖ Serial Number Programming: - Any serial number can be programmed to target MCU with automatic incrementing.
- ❖ Password protection: - You can protect you programmer with a password to protect it from erase or rewriting. This will also prevent the Limit counter status.
- ❖ LCD Display: - IO-Flasher PRO– STM8 has interactive OLED LCD Display
- ❖ Program Menu: - It stores all project settings from GUI to work independently without GUI. This Menu shows all settings done from GUI.
- ❖ Interactive programming: - LCD display Programming status, Number of chips programmed. Error codes and meaning on that error code.
- ❖ Dual Mode: - While IO-FLASHER is configured as offline mode. Its can be still used in online mode for same or different firmware code.
- ❖ Programmable Power Output: - No power supply required, Powered through USB. Target board can be power from IO-FLATHER. Output voltage can be adjusted from 0 Volt to ~5Volt depending on selected target device (up to 200mA to target with overload protection).
- ❖ Target Power Sensing: - If target is self-powered or powered from IO-FLASHER, then it can be measured by the IO-FLASHER.
- ❖ In-built Protection: - All external port protected with ESD, Short circuit and over current. Isolation is available in Isolated version of IO-FLASHER (e.g. IO-FLASHER -STM8/ISOL)
- ❖ Programming Options: - Have many programming options to program Flash, EEPROM, Option memory with auto lock feature.
- ❖ Supported devices: - Support for all STM8 devices
- ❖ STM8S, STM8L, STM8AF, STM8AL
- ❖ IO-FLASHER GUI: - Highly professional GUI with simple and user-friendly interface.



**Figure 2**

### **1.1.3 IO-Flasher PRO– STM8/ISOL (On Request)**

IO-Flasher PRO– STM8/ISOL is the Isolated version of IO-Flasher PRO– STM8. SWIM connector is galvanic isolated from the target board. Which makes its more robust.

Please contact at [info@iotrue.com](mailto:info@iotrue.com) if you need this programmer on priority basis.

### **1.1.4 IO-GANG FLASHER (On Request)**

IO-GANG Flasher is a gang programmer for STM8. This is a on request product that can be customized as per customer need. Customer can choose isolated/Non isolated version of gang programmer. Customer can also choose number of parallel socket (2, 4, 6 or 8) to program many STM8 device parallely.

Please contact at [info@iotrue.com](mailto:info@iotrue.com) if you need this programmer on priority basis.

### **1.1.5 General Information**

IO-Flasher can be operated in slave mode from PC with **IO-Flasher GUI** or can be operated in standalone mode. The latest version of IO-FLASHER GUI can be downloaded from below link:

**<http://www.iotrue.com/downloads>**

**Power Supply:** IO-FLASHER can be powered from HOST PC over USB while working in online mode. Or can also be powered up with Battery power bank, USB charger via USB while working in standalone mode.

## **2. IO-FLASHER- STM8 family Specification**

### **2.1. Supported Operating systems: -**

IO-FLASHER GUI is supported by below operating systems:

- Microsoft Windows XP
- Microsoft Windows XP x64
- Microsoft Windows Vista
- Microsoft Windows Vista x64
- Windows 7
- Windows 7 x64
- Windows 8
- Windows 8 x64
- Windows 10
- Windows 10 x64

### **2.2. Operating Temperate:**

Operating range: +5°C ... +60°C

### **2.3. Electrical Characteristics:**

- Input rating on USB : 5Volt +/- 1Volt.
- Supply Voltage to Target from IO-FLASHER : 2Volt to approx. 5Volt
- Supply Current to Target from IO-FLASHER : Maximum 200mA
-

### **3. IO-Flasher GUI:**

IO-FLASHER GUI is a most sophisticated and user-friendly programming tool which enable the user with error free programming on the target device.

#### **3.1. IO-FLASHER GUI Features:**

IO-FLASHER has below feature and advantages:

- Encrypted Project creation with password protection.
- Saves User HEX/S19/BIN file in project itself in encrypted form.
- Hex File Viewer and Editor
- Protect Hex data from accidental change.
- Display checksum of original and modifies file separately.
- Display Flash, EEPROM and Option bytes in different tabs.
- Easy to handle option byte in Option byte TAB.
- Separate configuration window and Configuration view.
- Log view to show all programming activity.

#### **3.2. Downloading IO-FLASHER GUI:**

The latest version of IO-FLASHER GUI can be downloaded from below link:

**<http://www.iotrue.com/downloads>**

### 3.3. Installing IO-FLASHER GUI:

Click on setup file available is downloaded software from above link. You will get the below window

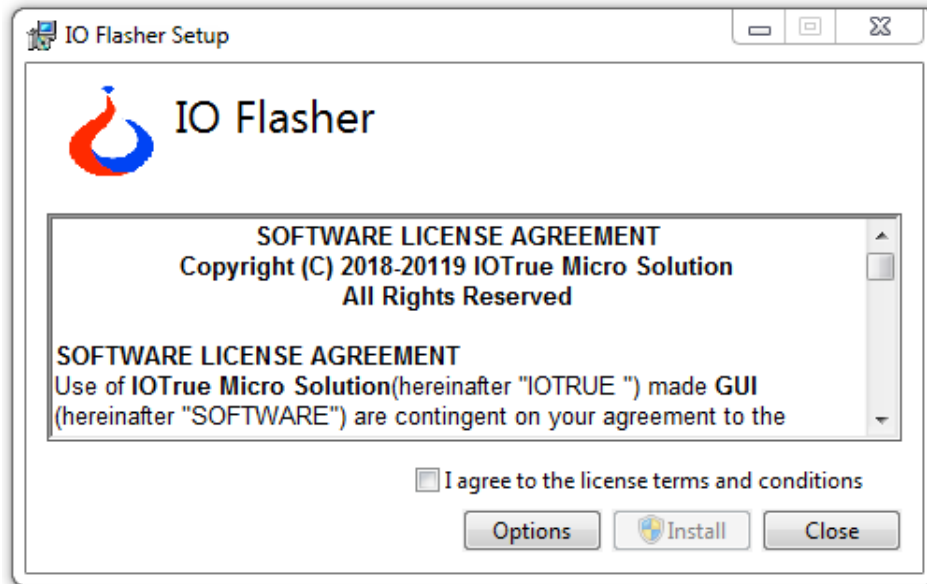


Figure 3

Read the Licence agreement carefully and check mark on "I agree to the term and conditions" if you are agree to it.

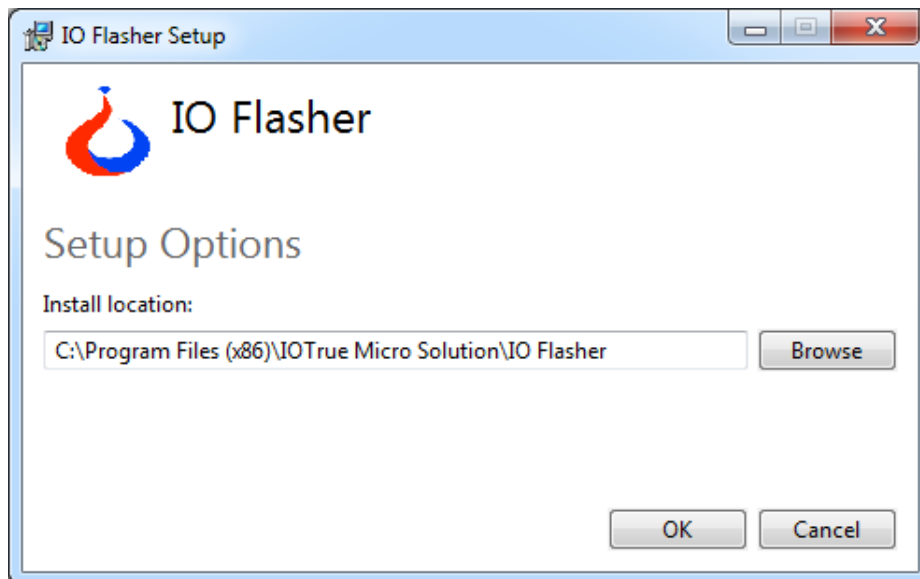


Figure 4

You can also change the Installation directory by clicking on **Option** button



### 3.4. IO-FLASHER GUI Overview:

Once installation is done, you can open IO-FLASHER GUI from short-cut on desktop or from start menu -> IoTure -> IO-FLASHER in your desktop PC or laptop.

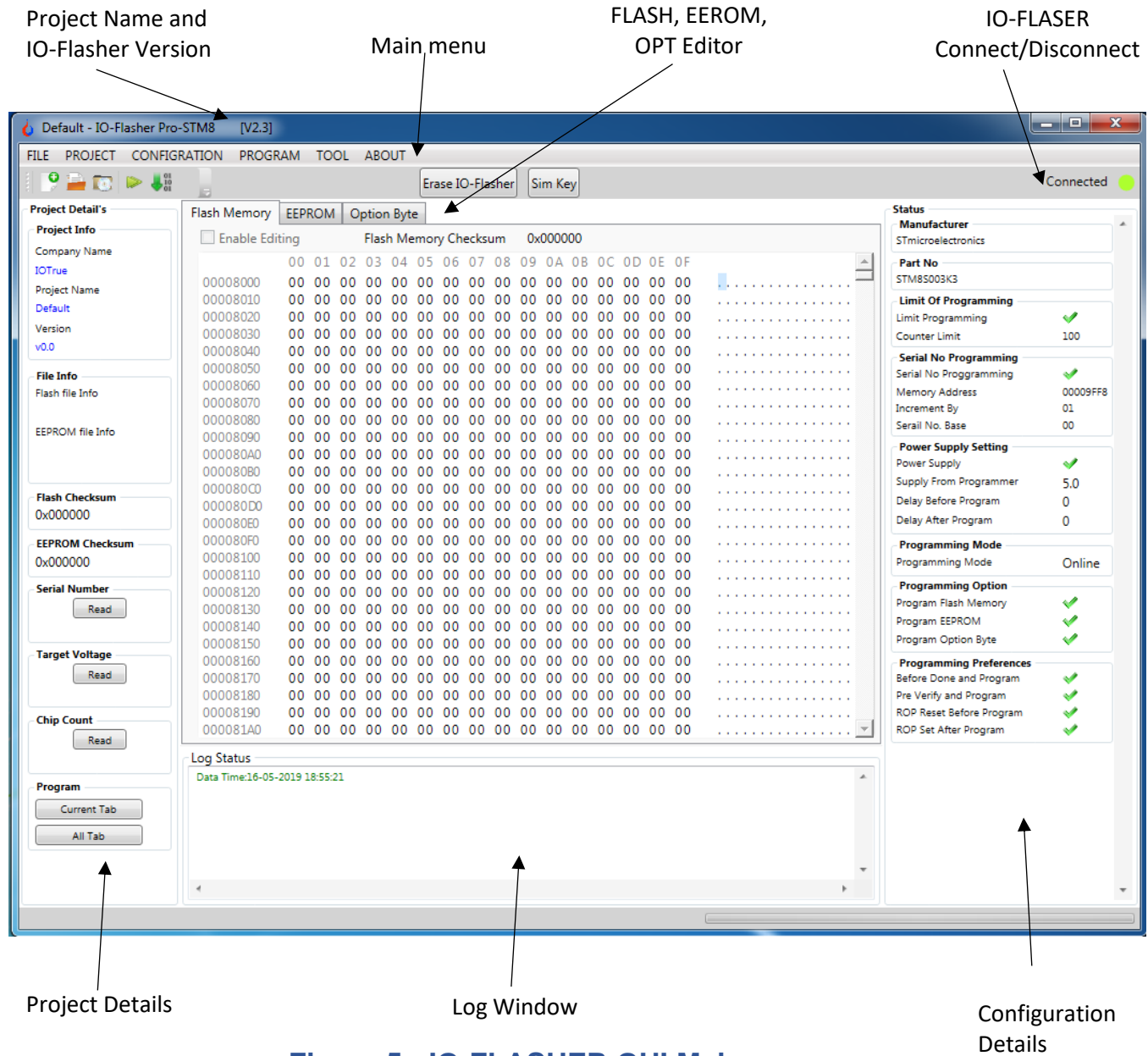


Figure 5: IO-FLASHER GUI Main page

### 3.5. Project Creation (Optional):

Project creation is optional step.

Creating project help you to eliminate need of opening the '.Hex', '.S19', '.Bin' file again and again. Project can save your '.Hex', '.S19', '.Bin' file data in project file itself.

Project will be encrypted and **password protected (AES-256)** so that any unauthorised person cannot open your project.

You can create project from project-> New project

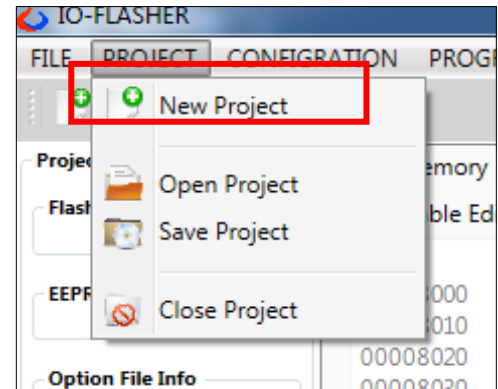


Figure 6

Once you click on new project below tab will open.

- Give your company name. This name will also be displayed on the programmer LCD. Maximum length of the company name is 21 characters. (Optional)
- Give your project name maximum 21 Character long
- Give project version maximum 6 character long
- Project location anywhere
- You can optionally enable security to encrypt you project with your password. Maximum length of the password can be 16 character in any combination
- Click on create after filling the enquired field.

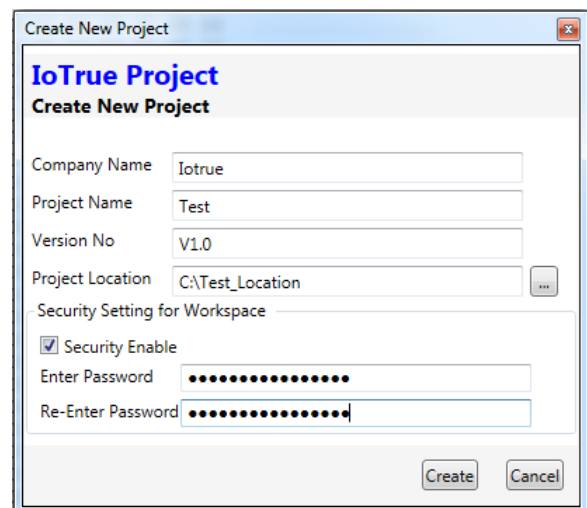
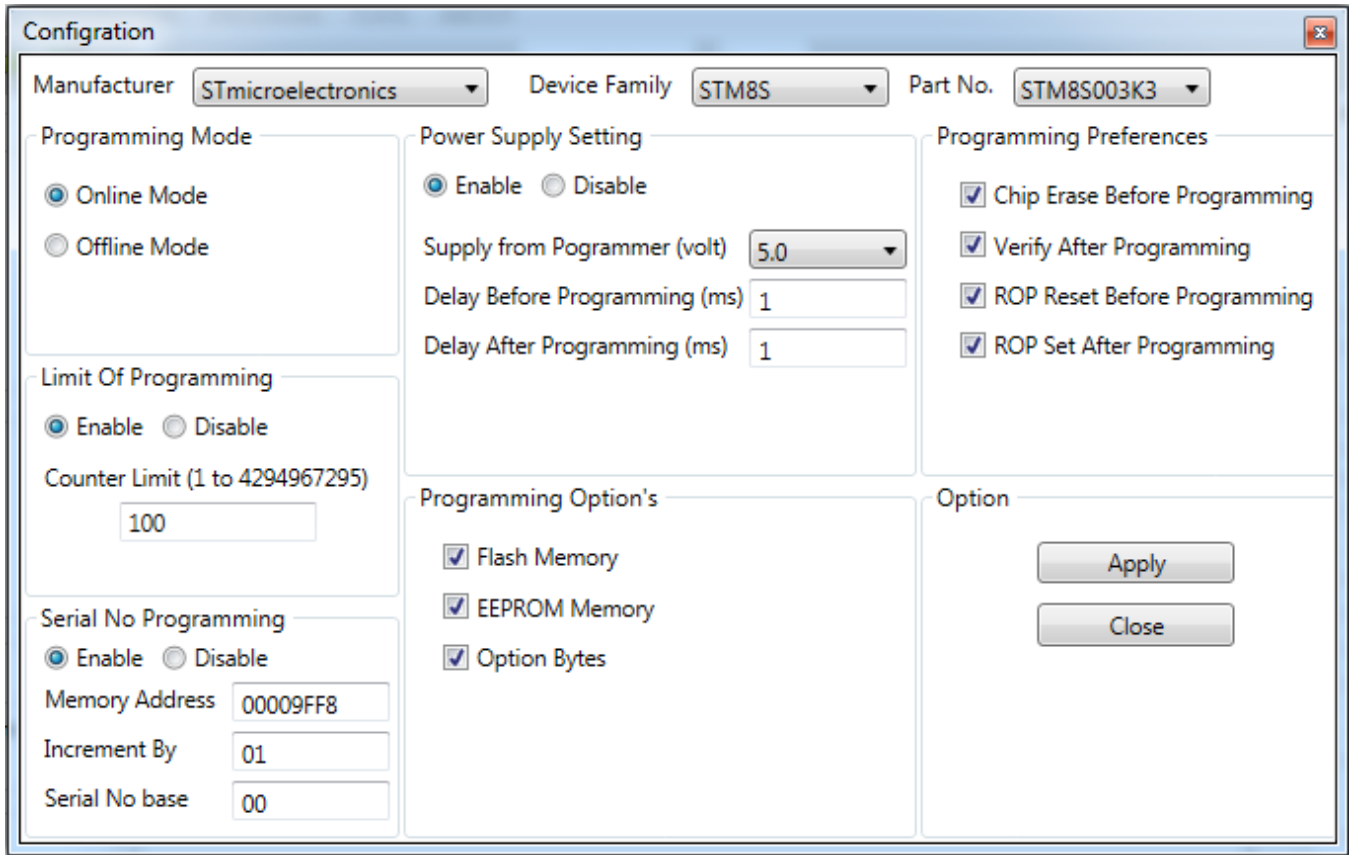


Figure 7

**Note: - Please note that if you forgot the password once enabled, you will not able to Reprogram/Erase the IO-Flasher. This feature has been implemented for Security reason.**

**In this scenario you need to send the IO-Flasher to IoTrue Micro Solutions to factory reset the IO-Flasher.**

**3.5.1 Configuration window:** - Now configuration window will pop-up automatically. Check the below screenshot of the configuration window.



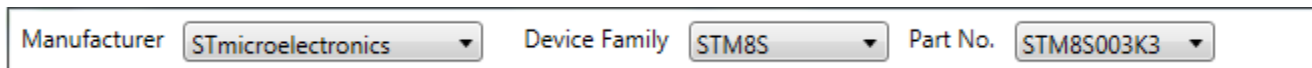
**Figure 8**

You have below option in the configuration window to select:

**Manufacturer Name:** - Display the MCU chip manufacturer name.

**Device Family:** - Select the microcontroller (MCU) family.

**Part No:** - Select the desired microcontroller part number to program.



**Figure 9**

### 3.5.2 Programming Mode: - IO-Flasher support two programming mode.

Offline Mode: - This mode also called stand-alone programming mode. In this mode the selected '.Hex', '.S19', '.Bin' file will be stored in the secure internal flash of the programmer. Connection to target board is not required in this mode once IO-Flasher programmed in offline mode.

Data Stored in IO-Flasher can not be read back to protect the firmware IP. Only configuration data can be read IO-Flasher with a valid password (if enabled).

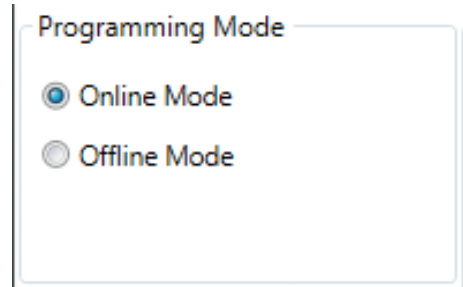


Figure 10

Online Mode: - In this mode the selected MCU will be directly program with the FLASH/EEPROM/OPT data selected by the user. User has open the '.Hex', '.S19', '.Bin' file in the respected tab. In this mode '.Hex', '.S19', '.Bin' file will not stored in the internal memory of the IO-FLASHER.

### 3.5.3 Power Supply Settings: -

- When you program the target, IO-FLASHER first checks itself, if target is already powered up by external power supply. Depending upon the external power supply is being used or not, you have below two option to select one of them.

Power Supply Enable: - If your target is not supplied from external power supply then, you can select Power Supply Enable option. Now you have option to select below option to select

- Supply Voltage: Select the voltage level that you want to feed to your target board from IO-Flasher. Target voltage from programmer cannot be selected beyond device operating range to protect the device.
- Delay Before Programming: - This delay will be in mSec. This delay will be inserted between button press and actual programming starting. This delay will be dependent on the capacitance of you target board. So you need to select this delay according to the raising time of supply voltage on your target board.
- Delay After Programming: - This delay will be in mSec. This delay will be inserted between programming complete and power supply off from target. This delay can be useful if you want to do some functional test after programming.

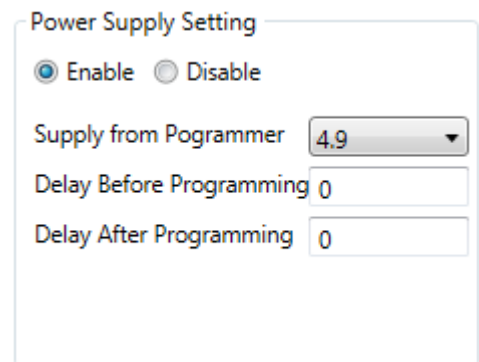


Figure 11

Power Supply Disable: - If your target board is already powered with external power supply then you should select this option.

*Note:* - Although IO-Flasher has the protection against over current and short-circuit. But it is recommended to disable the power feed from programmer if your target board has its own power.

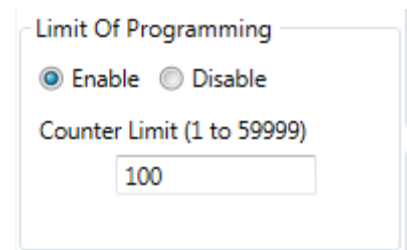
- You can also read the target power supply by clicking on Target voltage button.



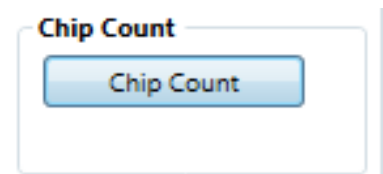
**Figure 12**

### 3.5.4 Limit programming: -

- You can limit the IO-FLASHER to limit the number of target board to be programmed. Once the counter will reach to define range. IO-Flasher will stop programming in Offline (Standalone) mode.
- This will help you to monitor that how many target board has been programmed using IO-Flasher.
- To protect your password theft the counter cannot be changed separately.
- You can also read the number of target board program by clicking on



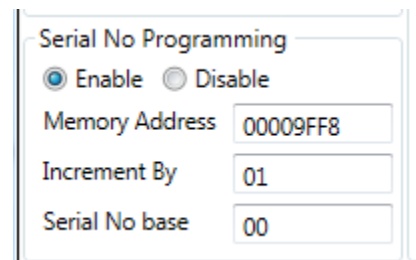
**Figure 13**



**Figure 14**

### 3.5.5 Serial No. programming: -

- If you want to store serial number to your individual target board then you can enable serial number programming option. You need to provide below inputs to use the option. You can have maximum 8 byte long serial number



**Figure 12**

Memory Address – Here you can provide the FLASH memory address. Wherever you want to save the serial number. By default, it takes the end address of the flash memory e.g if you are using SM8S003K3P6 which has 8KB of flash. Then

Start address of Flash memory will be = 0x008000  
 End address of Flash memory will be = 0x008000 + 8KB -1  
 = 0x008000 + 0x2000 -1  
 = 0x00A000 - 1  
 = 0x009FFF  
 Now start address of serial No will be = 0x009FFF - 8 + 1  
 = 0x009FF8

Serial No. Base: - Here you can give the starting of the serial number. IO-FLASHER will automatically increment the serial number depending upon serial No. increment By option (Explain below in section xxxxxx)

e.g: - if Serial Number start from 0x1100000000000000

Increment By option selected as 1

The serial number will be 0x1100000000000000, 0x1100000000000001,  
 0x1100000000000002, 0x1100000000000003, 0x1100000000000004,  
 0x1100000000000005.....and so on

Increment By – If you are using multiple programmer for production than you can use Increment by option for serial number.

For Example, if you are using 3 programmer A, B, C.

A Serial base = 0x1100000000000000

B Serial base = 0x1100000000000001

C Serial base = 0x1100000000000002

Increment By option should be selected as 3

Now serial number programmed by A will be 0x1100000000000000,  
 0x1100000000000003, 0x1100000000000006, 0x1100000000000009,  
 0x110000000000000B ..... and so on

serial number programmed by B will be 0x1100000000000001, 0x1100000000000004,  
 0x1100000000000007, 0x110000000000000A, 0x110000000000000C ..... and so on

serial number programmed by C will be 0x1100000000000002, 0x1100000000000005,  
 0x1100000000000008, 0x110000000000000B, 0x110000000000000D ..... and so on

So, each target board will have separate serial number as below.

0x1100000000000000,                    0x1100000000000001,                    0x1100000000000002,  
0x1100000000000003,                    0x1100000000000004,                    0x1100000000000005,  
0x1100000000000006,                    0x1100000000000007,                    0x1100000000000008,  
0x1100000000000009..... and so on.

- You can also read the serial number by clicking on Serial No button

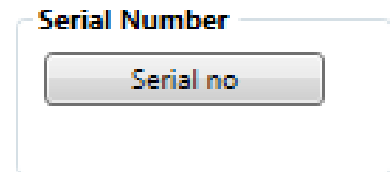


Figure 13

### 3.5.6 Programming Options: -

- In this setting you can select which Memory section you want to program during Online or Offline (Standalone) mode.
- For Example in Fig 14 EEPROM Memory is not selected so it will not be program in Chip.

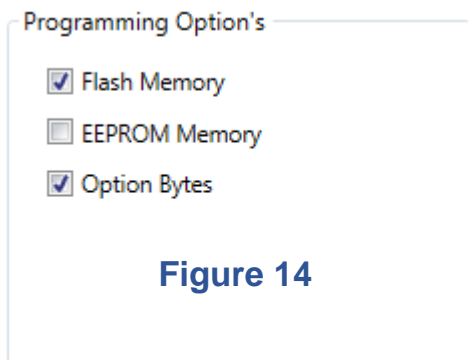


Figure 14

### 3.5.7 Programming Preferences: -

- Chip Erase Before programming: - Selected Chip will be erased first before programming.
- Program and Verify: - Code will be verified if it has been correctly written on the respected memory area.
- ROP reset before programming: - Programmer will check if Readout Protection (ROP) or PCROP has been set on the connected chip. It will automatically clear the ROP bit or PCROP bit first before programming
- ROP Set After Programming: - Readout Protection (ROP) will be automatically set by the programmer to protect your code inside chip.

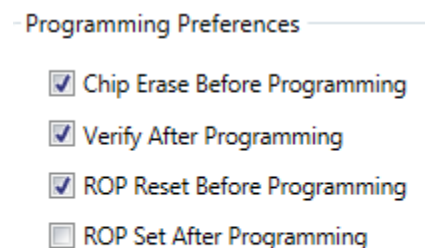


Figure 15

**3.6 Configuration Status window:** - This window will show all the option that has been selected in configuration . It is recommended to review these settings before start of programming to avoid any user mistake.

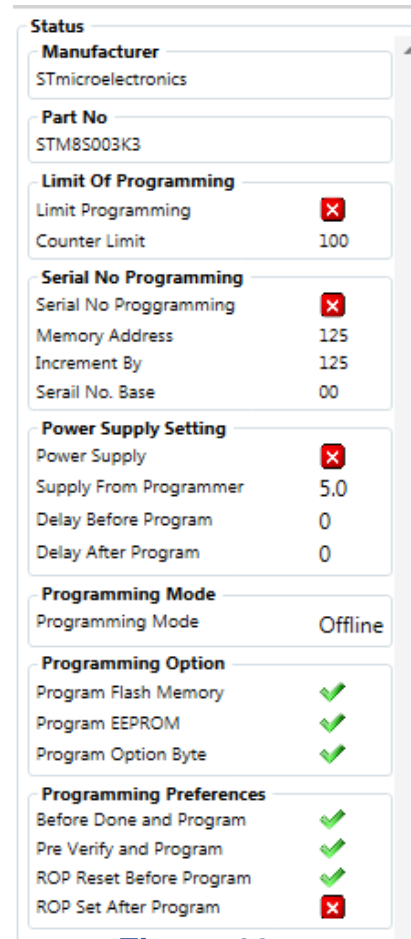


Figure 16

**3.7 Opening Flash and EEPROM files:** -

User can open the .Hex, .S19, .Bin file any time in hex editor. Hex editor will show the selected file in read only mode y default

To open the Flash File for selected device, Go to File->Open-> Flash File

To open the EEPROM File for selected device, Go to File->Open-> EEPROM File

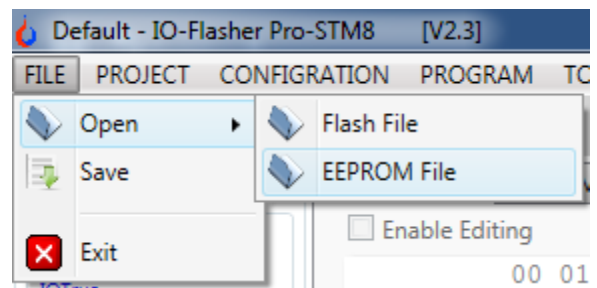
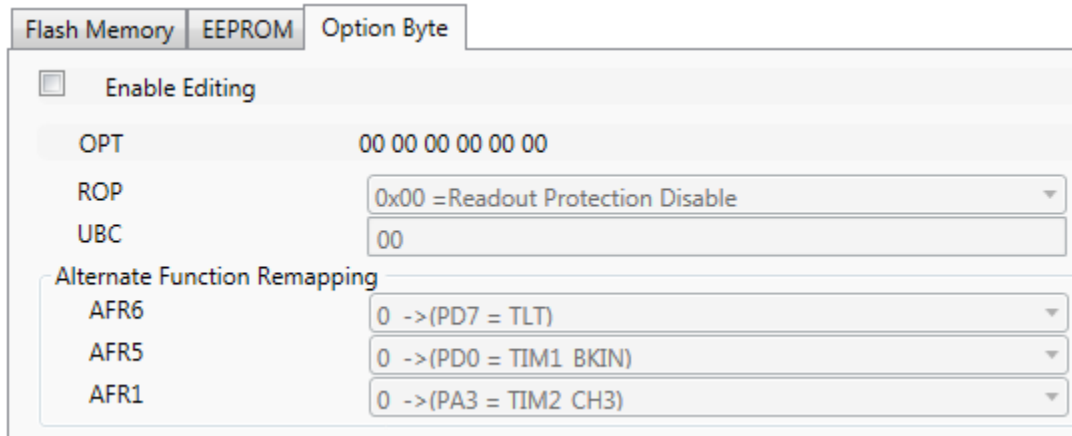


Figure 17



**3.8 Setting Up Option Byte:** - IO-FLASHER GUI has an interactive view for Option byte selection. If you have selected the option byte once for a given project, then you don't need to reselect the option byte again because all you setting will be saved in the encrypted project itself.

*Note: You need to check on "Enable Editing" option to edit the option bytes.*

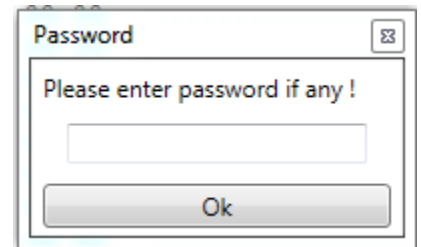


**Figure 18**

**3.9 Erase IO-Flasher:** - Clicking on this button will erase the data/code stored in IO-Flasher.

One you click on this button IO-Flasher will prompt you to enter the password for authentication.

If you enabled the password in the program stored in the IO-Flasher then you need to enter the same password to Erase the IO-Flasher.

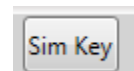


**Figure 19**

**Note: - Please note that if you forgot the password once enabled, you will not able to Reprogram/Erase the IO-Flasher. This feature has been implemented for Security reason.**

**In this scenario you need to send the IO-Flasher to IoTrue Micro Solutions to factory reset the IO-Flasher.**

**3.10 Sim Key:** - Sim Key is the simulation key. Clicking on this button will simulate the physical button pressed on IO-Flasher- STM8.

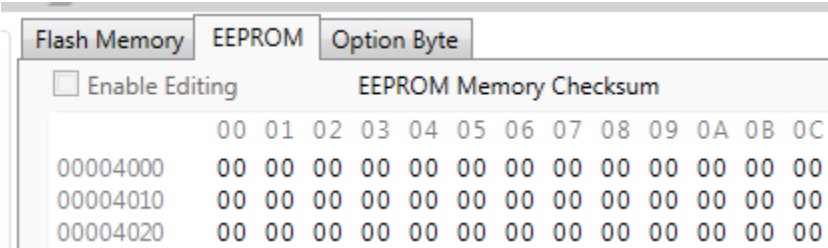


**Figure 20**

## 4. Target MCU programming:

**Online programming:** - Once you done with the configuration and file opening (Flash/EEPROM files) and Option byte selection. You can program your device by clicking on program current tab or program all tab.

Program current tab will program the tab which have selected out of Flash Memory , EEPROM and Option byte. For example in below image EEPROM tab is selected so programmer will program only EEPROM tab while clicking on program current tab button.



The screenshot shows the IO-Flasher software interface with three tabs: Flash Memory, EEPROM, and Option Byte. The EEPROM tab is selected. Below the tabs, there is a checkbox for 'Enable Editing' and a section titled 'EEPROM Memory Checksum'. The checksum table has 13 columns labeled 00 through 0C and three rows of data.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C
00004000	00	00	00	00	00	00	00	00	00	00	00	00	00
00004010	00	00	00	00	00	00	00	00	00	00	00	00	00
00004020	00	00	00	00	00	00	00	00	00	00	00	00	00

**Figure 21**

Program All tab will program all the tab data (Flash Memory , EEPROM and Option byte) depending on your selection in configuration-> Programming option

**Note:** Good thing is that, while using Online programming Mode. It will not affect the offline data stored in the programmer. To know about offline mode read below section

**Offline (Standalone) mode:** - Once you done will the configuration and file opening (Flash/EEPROM files) and Option byte selection. You can Flash all data in IO-Flasher programmer if Offline mode is selected.

While Flashing data in offline mode there is no need to connect the chip with IO-Flasher.

Once IO-FLASHER is programmer in offline mode you can disconnect it from host PC. And you can program your chip without IO-FLASHER GUI. By simply providing power to IO-FLASHER USB from and USB adaptor or USB charger or USB power bank or any PC.

Programming Chip in Offline (Standalone) mode: - You can program the chip by simply pressing the button on IO-FLASHER.

**IO-FLASHER basic version:** - In IO-FLASHER basic version you have three LED to indicate the status on programmer

**GREEN LED:** - Indicates that programmer is ready to program next chip.

**RED LED** – Indicate that there is any error during programming. Press button again to clear the error state of programmer.

**BLUE LED:** - Indicates (while blinking) that IO-FLASHER is programming the chip and busy right now. Please do not press the button again if this LED is Blinking.



**Figure 22**

If you are getting RED LED indication again and again than use IO-FLASHER GUI in monitoring mode to check the error code and its meaning. (Not available IO-Flasher GUI V2.0)

**IO-FLASHER PRO:** - In IO-FLASHER PRO you have LCD which display all indication, error code and its meaning during programming. Below are many screens showing useful information



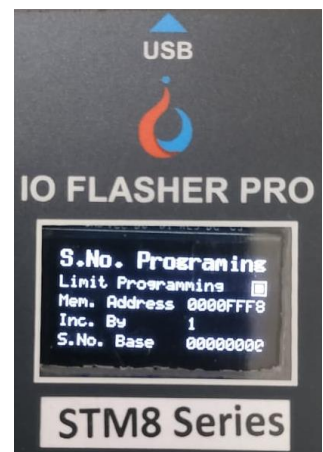
**Figure 23(a)**



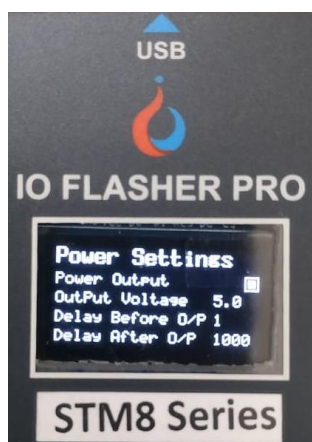
**Figure 23(b)**



**Figure 23(c)**



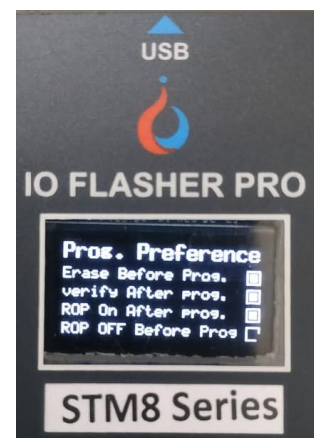
**Figure 23(a)**



**Figure 21(e)**



**Figure 21(f)**



**Figure 21(g)**

## 5. Error Description: -

ERROR 81 CAPTURE\_COMMAND\_MISMATCH= 0x81,  
ERROR 82 CAPTURE\_ADDRESS\_MISMATCH,  
ERROR 83 SWIM\_NACK\_RECEIVED,  
ERROR 84 SWIM\_PARITY\_ERROR\_RECEIVED,  
ERROR 85 SWIM\_CAPTURE\_FAIL,  
ERROR 86 NACK\_OR\_PARITY\_ERROR,  
ERROR 87 SWIM\_ENTRY\_FAILED,  
ERROR 88 SWIM\_RESET\_FAILED,  
ERROR 89 SWIM\_WRIT\_ADR\_WR\_FAILED,  
ERROR 8A SWIM\_CPU\_NOT\_STALLED,  
ERROR 8B SWIM\_RD\_DM\_CSR2\_RD\_FAILED,  
ERROR 8C SWIM\_CLK\_SWIMCCR\_RD\_FAILED,  
ERROR 8D SWIM\_SYNC\_FRAM\_NOT\_RECEIVED,  
ERROR 8E SWIM\_SYNC\_FRAM\_NOT\_RECEIVED2,  
ERROR 8F SWIM\_WRIT\_ADR1\_FAILED,  
ERROR 90 SWIM\_WRIT\_ADR2\_FAILED,  
ERROR 91 SWIM\_WRIT\_CKDIVR\_FAILED,  
ERROR 92 SWIM\_RD\_DEVICE\_ID\_FAILED,  
ERROR 93 SWIM\_RD\_IAPSR\_FAILED,  
ERROR 94 SWIM\_DEVICE\_ROP\_ENABLED,  
ERROR 95 SWIM\_DEVICE\_UBC\_ENABLED,  
ERROR 96 SWIM\_ROP\_CHECK\_FAILED,  
ERROR 97 SWIM\_RD\_OPT0\_ROP\_FAILED,  
ERROR 98 SWIM\_RD\_OPT1\_UBC\_FAILED,  
ERROR 99 SWIM\_RD\_NOPT1\_UBC\_FAILED,  
ERROR 9A SWIM\_EXIT\_ERROR\_AFTR\_ROP\_CLEAR,  
ERROR 9B TVDD\_OUT\_OF\_RANGE,  
ERROR 9C TARGET\_NOT\_POWERED,  
ERROR 9D TARGET\_OVER\_CURRENT  
ERROR 9E PROG\_FLASH\_ERASE\_FAIL, // Fail to ERASE programmer internal Flash  
memory . ERROR  
ERROR 9F CONF\_WRITE2FLASH\_FAILED, // Fail to PROGRAM programmer internal  
Flash ERROR memory with configuration data.  
ERROR A0 NO\_CONF\_DATA\_FOUND\_IN\_FLASH,  
ERROR A1 CODE\_DATA\_WRITE2FLASH\_FAILED,  
ERROR A2 OPT\_BYTE\_WRITE2FLASH\_FAILED,  
ERROR A3 NO\_CODE\_DATA\_FOUND\_IN\_FLASH,  
ERROR A4 FLASH\_MEMORY\_NOT\_SELECTED\_4\_PROGRAMMING,  
ERROR A5 EEPROM\_MEMORY\_NOT\_SELECTED\_4\_PROGRAMMING,  
ERROR A6 OPT\_MEMORY\_NOT\_SELECTED\_4\_PROGRAMMING,

ERROR A7 MEMORY\_ADDRESS\_OUT\_OF\_RANGE, // Received Memory is out of range  
from ERROR selected MCU memry range.  
ERROR A8 COUNTER\_REACHED\_LIMIT, // if Limit programming counter enabled and  
counter ERROR reached to its limit  
ERROR A9 PASSWORD\_MISMATCH,  
ERROR AA READ\_FLASH\_IAPSR\_CMD\_ERROR,  
ERROR AB BLOCK\_STD\_PROG\_TIME\_CMD\_ERROR,  
ERROR AC BLOCK\_UNLOCK\_FLASH\_KEY1\_CMD\_ERROR,  
ERROR AD BLOCK\_UNLOCK\_FLASH\_KEY2\_CMD\_ERROR,  
ERROR AE BLOCK\_UNLOCK\_EEPROM\_KEY1\_CMD\_ERROR,  
ERROR AF BLOCK\_UNLOCK\_EEPROM\_KEY2\_CMD\_ERROR,  
ERROR B0 BLOCK\_FLASH\_EEPROM\_LOCKED\_ERROR,  
ERROR B1 BLOCK\_CHEK\_IAPSR\_DUL\_PUL\_CMD\_ERROR,  
ERROR B2 BLOCK\_WRITE\_FLASH\_CR2\_CMD\_ERROR,  
ERROR B3 BLOCK\_STALL\_CPU\_CMD\_ERROR, // STM8 core not stalled.  
ERROR B4 WRITE\_BLOCK\_DATA\_CMD\_ERROR, // Block write commomd error  
ERROR B5 BLOCK\_IAPSR\_HVOFF\_ERROR, // HVOFF Flag not set after block  
programming ERROR  
ERROR B6 BLOCK\_IAPSR\_CLEAR\_CMD\_ERROR, // HVOFF and EOP Read commond  
fail( ERROR this commong required for HVOFF and EOP flag clearing)  
ERROR B7 DATA\_VERIFICATION\_FAILED,  
ERROR B8 DATA\_VERIFICATION\_CMD\_FAILED,  
ERROR B9 BYTE\_UNLOCK\_FLASH\_KEY1\_ERROR,  
ERROR BA BYTE\_UNLOCK\_FLASH\_KEY2\_ERROR,  
ERROR BB BYTE\_UNLOCK\_EEPROM\_KEY1\_ERROR,  
ERROR BC BYTE\_UNLOCK\_EEPROM\_KEY2\_ERROR,  
ERROR BD BYTE\_FLASH\_EEPROM\_LOCKED\_ERROR,  
ERROR BE BYTE\_CHEK\_IAPSR\_DUL\_PUL\_ERROR,  
ERROR BF BYTE\_WRITE\_FLASH\_CR2\_ERROR,  
ERROR C0 BYTE\_WRITE\_DATA\_BYTE\_ERROR,  
ERROR C1 BYTE\_CHEK\_IAPSR\_EOP\_ERROR,  
ERROR C2 EXIT\_RD\_IAPSR\_ERROR,  
ERROR C3 EXIT\_CSR\_ADR\_ERROR,  
ERROR C4 EXIT\_SRST\_FAILED  
ERROR C5 OPT\_BYTE\_ERASE\_ERROR

## **Hardware Warranty and Repairs: -**

IoTrue Micro Solutions thoroughly tests all hardware before shipping to end users and distributors. However, occasionally a user may receive a piece of hardware that is not working right out of the box. This may be due to damage during shipping or storage. In this situation, the customer should contact IoTrue Micro Solutions and request a replacement within 30 days after purchasing the product. IoTrue Micro Solutions will issue a no-cost exchange for a new item which is determined to be not working when it is shipped from IoTrue Micro Solutions.

Our technical experts will analyse if it is a manufacturing defect or shipment damage. IoTrue Micro Solutions will replace or repair without charge to the customer and will also bear the second shipment cost. IoTrue Micro Solutions's warranty does not cover damage incurred through mishandling or misuse of the product by customer.

IoTrue Micro Solutions provides **1-Year support warranty** after purchasing the product. During the support warranty IoTrue Micro Solutions will provide the off-site free updates on the firmware in any new device(IC) launched in the same product series. During support warranty, IoTrue Micro Solutions will also provide the paid repairing of the product if product is damaged due to mishandling or misuse of the product. After one year of purchase date, no updates will be provided and paid repairing will be subjected to availability of the spare parts/components as design may be changed and firmware version may be outdated. This will solely IoTrue Micro Solutions decision to provide support or not, on the product older than one year.

If you have purchased a product from IoTrue Micro Solutions and it is in need of repair please call us to speak with a technical expert and he will guide you for the testing of the product and further steps. You may need to ship the product to IoTrue Micro Solutions on your own cost. If the damage is due mishandling or misuse of the product, the client is responsible for covering all costs associated with the repair and shipment cost. We will provide the estimate the cost of the repair. In some cases, when the product has expensive parts in need of repair, IoTrue Micro Solutions may suggest replacing the old product for a new product at a discounted price.

## **Disclaimer**

Specifications written in this document are believed to be accurate, but are not guaranteed to be entirely free of error. IoTrue Micro Solutions, Inc. reserves the right to make changes, corrections, enhancements, modifications, and improvements to IoTrue Micro Solutions products and/or to this document at any time without notice herein to improve reliability, function, or design. Please make sure your manual is the latest edition. IoTrue Micro Solutions, Inc. does not assume any liability arising out of the application or use of any product or circuit described herein. Purchasers are solely responsible for the choice, selection, and use of IoTrue Micro Solutions products and IoTrue Micro Solutions assumes no liability for application assistance or the design of Purchasers' products. In no event will IoTrue Micro Solutions be liable for additional damages, including any lost profits, lost savings or other incidental or consequential damages arising out of the use or inability to use these programs, even if . IoTrue Micro Solutions has been advised of the possibility of such damage.

By using this software or product, you accept all the terms of this agreement.

## **Copyright notice**

This software and accompanying documentation are protected by Copyright law and also by International Treaty provisions. You may not extract portions of this manual or modify the PDF file in any way without the prior written

permission of the manufacturer. All the software described in this document is copyrighted by IoTTrue Micro Solutions, Inc. Copyright notices have been included in the software. IoTTrue Micro Solutions authorizes you to make archival copies of the software and documentation for the sole purpose of back-up and protecting your investment from loss. Under no circumstances may you copy this software or documentation for the purpose of distribution to others. Under no conditions may you remove the copyright notices from this software or documentation.

## **Trademarks**

Names mentioned in this manual may be trademarks of their respective companies.

Brand and product names are trademarks or registered trademarks of their respective holders.

## **Contact address**

IoTTrue Micro-Solutions  
C-242, Heera Enclave, United Paradise, Shahveri,  
Sector-4, Greater Noida West,  
Gautam Buddha Nagar,  
Uttar Pradesh 201009  
Mob: +91 99 7174 2727  
Email: [Info@iotrue.com](mailto:Info@iotrue.com)  
Website: <https://www.iotrue.com>