

aFlasher32_V2

User Manual

Version 2.000.00

Contents

1	Getting Started	3
1.1	System requirements	3
1.2	S/W operation.....	3
1.3	Device memory configuration.....	5
2	Program installation	6
2.1	Download of setup program	6
2.2	Execution of Setup program.....	6
3	Operation mode	10
3.1	PC host mode.....	10
3.2	Standalone mode	10
4	Configuration and function of PC GUI.....	11
4.1	GUI configuration	11
4.2	Functions by Menu category	12
4.2.1	Selecting target device	12
4.2.2	Displaying process progress	13
4.2.3	Selecting target memory.....	13
4.2.4	Loading hex files.....	14
4.2.5	Programming in PC mode	16
4.2.6	Process related to Standalone mode operation	18
4.2.7	Displaying buffers loaded in PC	19
5	Output messages	21
5.1	Output messages from the PC GUI	21
5.2	Output messages on aLinkPro.....	22
6	Command process	25
6.1	Overview	25
6.2	Functions and formats of command.....	25
6.2.1	Device programming	26
6.2.2	Device verification	27
6.2.3	Device erase.....	27
6.2.4	Device check for blank	28
6.2.5	Help	28
7	Precautions for programming.....	29
7.1	Precautions for connecting target devices	29
7.2	Precautions for preventing damages	29
	Revision history.....	30

1 Getting Started

1.1 System requirements

aFlasher32 is a program designed under the assumption that it is used in a Microsoft .NET Framework environment.

You can download and install the .NET Framework for free from the Microsoft's website. Before you install, check the following requirements:

- .NET Framework version 4.5 or higher version must be installed.
 - In the environment where the .NET Framework is not installed or a lower version of the .NET Framework is installed, this program does not work.

The OS environment for this program is as follows:

- Microsoft Windows 7 (SP1 must be installed)
- Microsoft Windows 8.x
- Microsoft Windows 10
- Microsoft Windows 11

You can use the USB HID interface to communicate with a dongle. For this communication, use the communication format set in CMSIS-DAP.

1.2 S/W operation

This program supports a function that programs Flash memory embedded in the ABOV's 32-bit device.

The available H/W dongles are aLink and aLinkPro that automatically detect connections. For more information on each H/W, refer to the corresponding H/W's manual.

aLink

- It runs on a host PC and programs a device.
- Built as a low-end, low-cost product, it does not support Standalone mode in which it operates alone without connecting to the PC because it does not have a built-in buffer for device programming.

aLinkPro

- It runs on a host PC and programs a device.
- It supports Standalone mode since it stores what to program on the device in its memory without connecting to the PC.
- To be used as Standalone mode, it should be communicated with all necessary information such as target device information and programming data information using the program's GUI while connected to the PC.

Dongle

The dongles mentioned in this document do not have a protection circuit against constant voltage and overcurrent, so this S/W is not suitable for mass production of integrated products. You are responsible for any issues that arise when you use this S/W for mass production.

Mass production

For mass production, you must use E-PGM+ that ABOV provides or 3rd party programmer.

1.3 Device memory configuration

This program is intended to program the ABOV 32-bit MCU that is based on the ARM Cortex processors. While traditional aFlasher32 targets only Code and Configuration memory, the new aFlasher32 (this S/W) also targets the Data area of the product that contains the Data area.

The aFlasher32 handles the Code and Configuration memory together, and you can set whether to process additional Data areas. Therefore, you can choose whether to program the Code and Configuration memory, only the Data areas, or all areas.

The table below shows the memory components of the product:

Memory	Description
Code	<ul style="list-style-type: none"> • This area stores the binary code that the device executes. • This area consists of Flash memory. • You can erase or write the entire area at once or in blocks. • This area is always selected together with the Configuration because it is accompanied by the configurations required for the device setup, such as Read protection.
Data	<ul style="list-style-type: none"> • This area stores the information that is required for the device execution. • This area may not be included in the device. • This area consists of Flash memory or EEPROM. <ul style="list-style-type: none"> – If this area is configured with Flash memory, you can erase or write all Flash memory at once or in blocks. – If this area is configured with EEPROM, you can also erase or write in bytes.
Configuration	<ul style="list-style-type: none"> • This area stores the initial setup values of H/W required to execute the device. • This area mainly contains Read protection function to protect the values in the Code or Data area.

2 Program installation

2.1 Download of setup program

You can download the Setup program from the ABOV website to install aFlasher32. The file name includes the version of the file and the date it was created, so you can easily identify if the program is up to date.

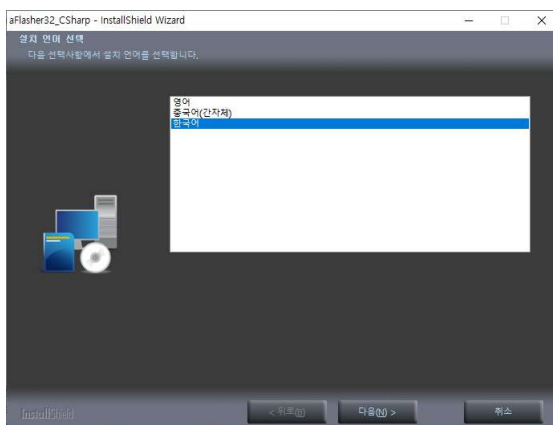
Example)

File name of the Setup program	Setup_aFlasher32_V2.123.45(231217).exe
Purpose of the Setup program	Setup
Target program	aFlasher32
Version of the program	V123.45
Distribution date of the program	231217 → 2023.12.17 (YYYY.MM.DD)

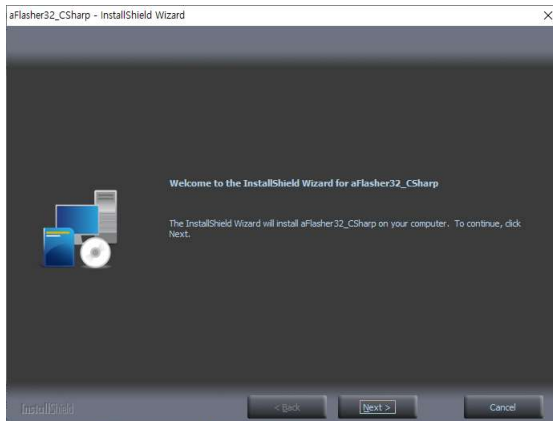
2.2 Execution of Setup program

When you execute the Setup program, it is installed in the following order:

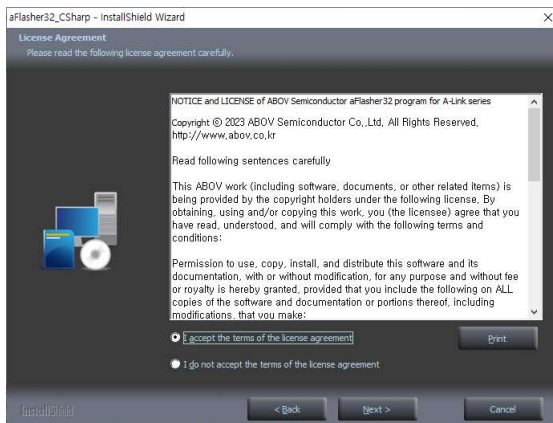
1. In the Setup dialog box, select the language that you want to use.



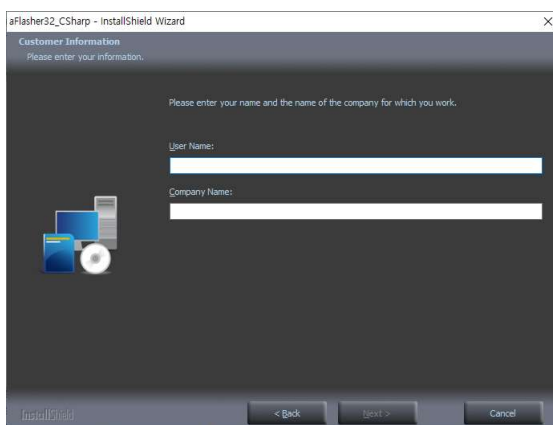
2. Setup greetings are displayed as shown in the figure below.



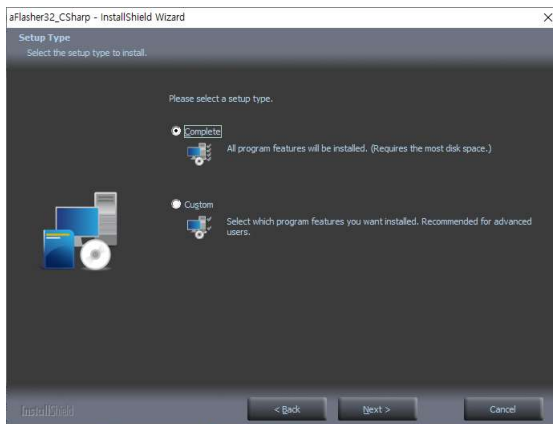
3. In the License Agreement window shown below, choose either Agree or Disagree.



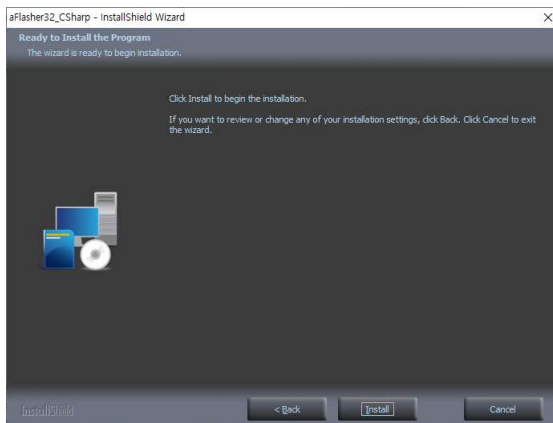
4. Enter user name and company name.



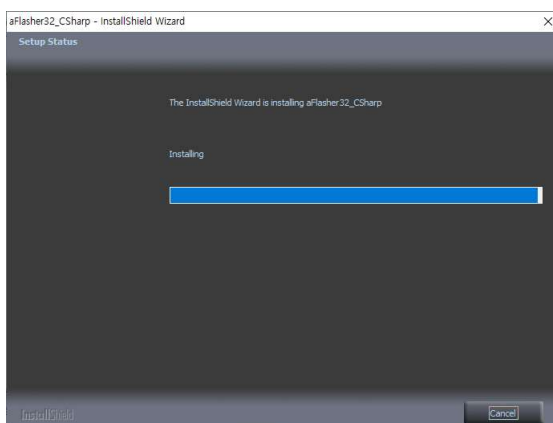
5. In the Setup Type window, choose either Complete or Custom. However, there is no difference between Complete and Custom.



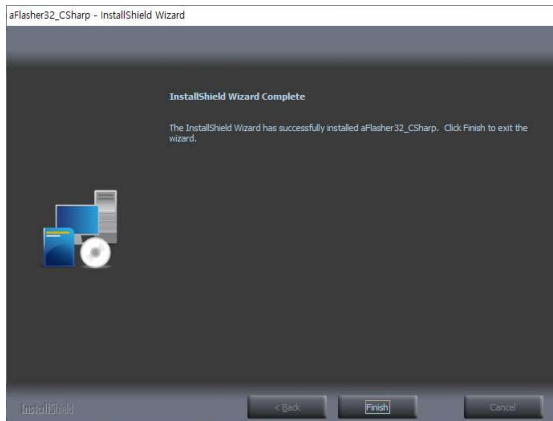
6. Once you have completed entering all information required for installation, the Ready to Install the Program window appears to confirm that you want to install the program.



7. If you click the Install button, the Setup program installation starts. It takes some time to install.



8. The installation complete message is displayed as shown below.



The Setup program is now successfully installed and aFlasher32 is ready for use.

3 Operation mode

3.1 PC host mode

In PC host mode, aFlasher32 performs the operations such as programming, verification, blank checking, and erasing, while connected to the PC.

- PC host mode supports both aLink and aLinkPro.
- This mode reads files in Intel Hex File Format to use hex data.
- This mode does not support binary without address information.
- If the Data to be programmed is blank or filled with the same value, the Data is considered to be abnormal.
- The default setting is to program all areas of the Code, Data, and Config.
 - The device without the Data area is set to exclude the Data program.
 - The device with the internal Data area can be set to program only the Data area.

3.2 Standalone mode

Standalone mode is a mode that allows you to program the target device without connecting to the PC. Therefore, this mode does not apply to the aLink that can be used only when connected to the PC, but only to the aLinkPro. The aLinkPro must include the required information.

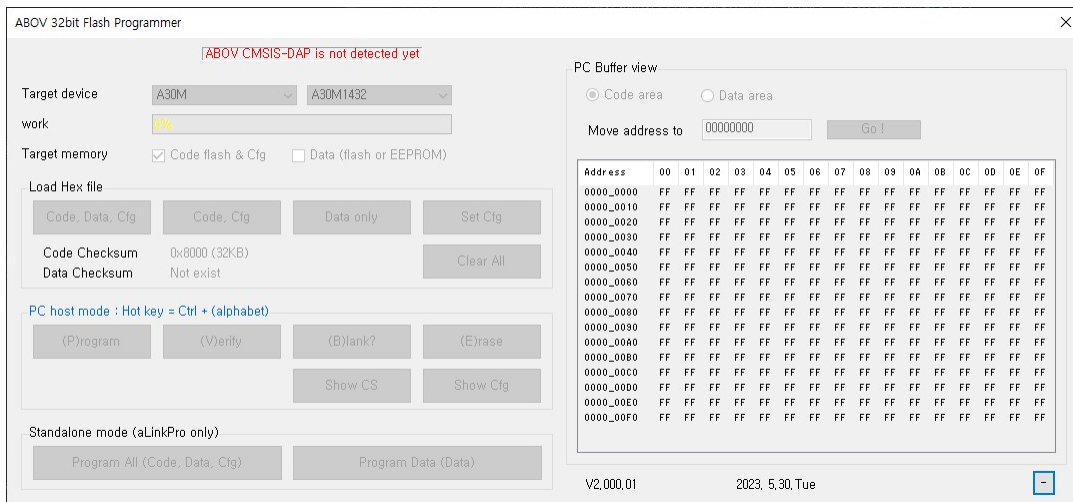
- If the aLinkPro is connected to the PC, it stores all information used to program the target device in its internal memory.
- The firmware operating in Standalone mode, downloader for programming the target device, and the values of Code, Data, and Confirmation of the target device can be stored in the internal memory of the aLinkPro.
- If the Data to program is blank or filled with the same value, the Data is considered to be abnormal.
- The default setting is to program all areas of the Code, Data, and Config.
 - The device without the Data area is set to exclude the Data program.
 - The device with the internal Data area can be set to program only the Data area.
- If you disconnect the PC and aLinkPro that stores the required information, the aLinkPro operates in Standalone mode when you power it off and then on.

4 Configuration and function of PC GUI

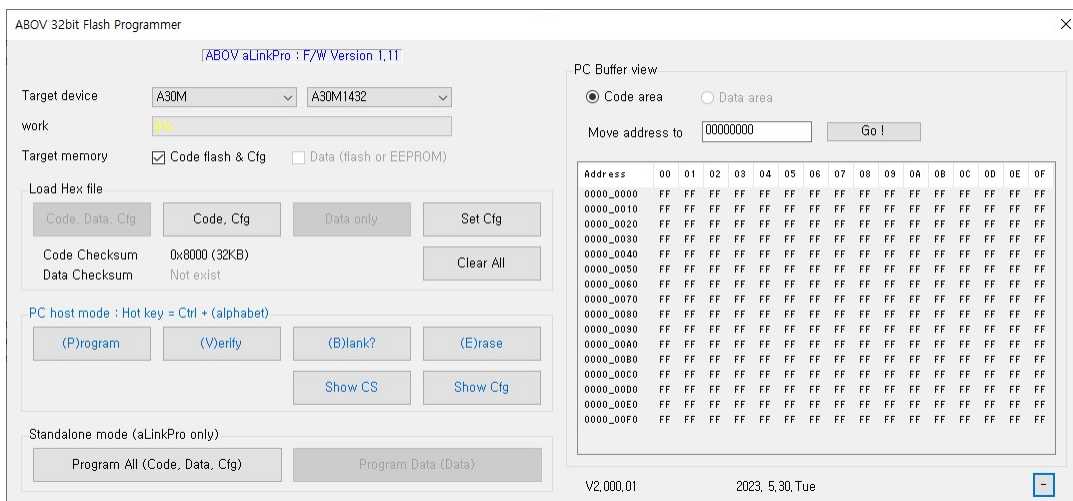
4.1 GUI configuration

When you run the S/W, it checks the connection between the aLink or aLinkPro and the PC to apply the connection state on the GUI.

If the dongle is not connected, all buttons are disabled as shown in the figure below.



If the dongle is connected, the Code flash & Cfg of the target memory item is selected and the associated buttons are enabled.



4.2 Functions by Menu category

The Menu can be categorized as follows:

- Functions to select the target device
- Functions to display the process progress
- Functions to select the target memory
- Functions to load the hex file
- Functions to program in PC mode
- Functions to process in Standalone mode and related operations
- Functions to display buffers loaded in the PC

4.2.1 Selecting target device

Using this menu, you can select the target device to be processed in this S/W.

1. The drop-down list on the left in the figure below selects the series of the device.

2. If you select a device series other than the current device series, all information and buffers loaded in the PC are erased to blank.

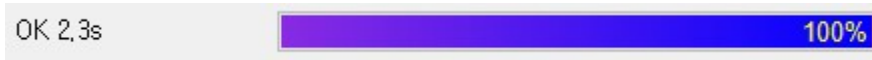
3. The drop-down list on the right in the figure below selects the device belonged to the series selected in 1.

4. If you select a device other than the current device, all information and buffers loaded in the PC are erased to blank.

4.2.2 Displaying process progress

By pressing the buttons related to the device programming, you can display how far that process is going and the associated errors.

- If the process result is 'pass', the OK message and the time consumed for the process are displayed. The progress bar is displayed in blue.



- If the process result is 'fail', the corresponding error message is displayed. The progress bar is changed to red.



4.2.3 Selecting target memory

Using this menu, you can select the memory to process during programming.

The 'Code & Cfg' is selected by default. If you select a device that has an internal Data area, the 'Data (flash or EEPROM)' checkbox will be enabled.

- If you select a device that has no Data memory, the menu appears as follows:

Target device	A31G	A31G111
work		
Target memory	<input checked="" type="checkbox"/> Code flash & Cfg	<input type="checkbox"/> Data (flash or EEPROM)

- If you select a device that has Data memory, the menu appears as follows:

Target device	A31G	A31G226
work		
Target memory	<input checked="" type="checkbox"/> Code flash & Cfg	<input type="checkbox"/> Data (flash or EEPROM)

The items affected by the target memory selection are as follows:

- Load Hex file buttons: It allows to perform the load operation only for the specified memory.
- Programming buttons: It allows to perform the operations such as programming, verification, blank checking, and erasing only for the specified memory.

4.2.4 Loading hex files

The Code and Data, and Cfg buffers have the blank values of the currently selected device as initial values. The 'Cfg' in this document means 'configuration'.

Each buffer does not erase the previous data when it loads a hex file. Instead of erasing, it fills itself with the value loaded later. Therefore, loading multiple hex files results in the same result as filling the buffer with the combination of each file.

The address range of hex files that can be loaded depends on the type of selected target memory.



1. Code, Data, Cfg button

- If you select Code and Data as the target memory, this button is enabled.
- The hex data outside the range of Code, Data, Cfg memory in the selected device is ignored.
- If the hex file includes the Cfg setup value, it is applied together.
- After loading the hex file, a dialog box displaying the Cfg setup state appears. In this dialog box, you can refer to or modify the setup values.

2. Code, Cfg button

- This button is enabled if the Code is included in the target memory.
- The hex data outside the range of Code, Cfg memory in the selected device is ignored.
- If the hex file includes the Cfg setup value, it is applied together.
- After loading the hex file, a dialog box displaying the Cfg setup state appears. In this dialog box, you can refer to or modify the setup values.

3. Data only button

- This button is enabled if the Data is included in the target memory.
- The hex data outside the range of Data memory in the selected device is ignored.
- Although the Cfg setup value is included in the hex file, the hex data is ignored.

4. Set Cfg button

- This button is enabled if the Cfg area is included in the device and the Code is included in the target memory
- A dialog box displaying the Cfg setup state appears. In this dialog box, you can refer to or modify the setup values

5. Clear All button

- The entire area of Code, Data, and Cfg is filled with the blank values of the currently selected device.

6. Code Checksum

- Pressing this button displays the checksum of the value in the code buffer of the currently selected device.

7. Data Checksum

- Pressing this button displays the checksum of the value in the code buffer of the currently selected device.

4.2.5 Programming in PC mode

Using this menu, you can perform the functions related to the internal memory programming of the actual device.

Click the appropriate button or press both the alphabet in parentheses on the button and the control key together to perform the function.



1. Program button

- Using this button, you can program the connected device with the data loaded in the buffer and verify this data on the connected device.
- If Read protection is set to the connected device, it is impossible to erase or write only the specified area. In this case, you can program the selected area after erasing all of Code, Data, and Cfg areas.

2. Verify button

- Using this button, you can verify that the area selected as the target memory of the connected device has the same value as the value loaded in the buffer.
- If Read protection is set to the connected device, it is impossible to perform the verification.

3. Blank button

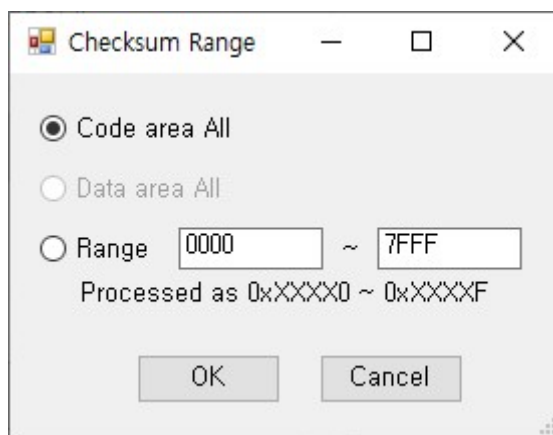
- Using this button, you can verify that the area selected as the target memory of the connected device has the same value as the value loaded in the buffer.
- If Read protection is set to the connected device, it is impossible to perform the verification.

4. Erase button

- Using this button, you can initialize the area selected as the target memory of the connected device to the blank state.
- If Read protection is set to the connected device, the entire area is initialized to the blank state, regardless of the selected area.

5. Show CS button

- Using this button, you can read and display the checksum for an interval specified in the connected device.
- A dialog box shown below appears, where you can select a checksum for the Code, Data, or specific range.



- If you select the Range to set an interval, each address is identified as the format '0xXXXX0 ~ 0xXXXXF'. That is, if you enter 0x1234 as a start address, the device identifies it as 0x1230; if you enter 0x5678 as an end address, the device identifies it as 0x567F.

6. Show Cfg button

- Using this button, you can read and display the configuration of the connected device.

4.2.6 Process related to Standalone mode operation

Standalone mode allows you to program the target device using a dongle separate from the PC. Only the aLinkPro supports Standalone mode because it has its own memory and display capabilities. The aLink does not support this mode and the corresponding button is disabled when the aLink is connected.



1. Program All (Code, Data, Cfg)
 - Pressing this button provides the aLinkPro with all information for programming the target device.
 - When pressing this button, the aLinkPro erases the entire internal memory of the target device and programs the target device with the information located in the internal memory of the aLinkPro.
 - Even if Read protection is enabled in the target device, the aLinkPro erases the target device and continues programming.
2. Program Data (Data)
 - Pressing this button provides the aLinkPro only with data memory information of the target device.
 - If the target device does not have data memory, this button is disabled.
 - The aLinkPro erases only data memory of the target device and programs the data memory of the target device with the information located in the internal memory of the aLinkPro.
 - If Read protection is enabled in the target device, the aLinkPro does not operate with this button.

4.2.7 Displaying buffers loaded in PC

Using this menu, you can display hex data in the buffer loaded in the PC.

PC Buffer view

Code ¹ area Data ² area

Move address to 4

Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000_0000	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_0090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_00A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_00B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_00C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_00D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_00E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0000_00F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

⁵

1. Code area
 - It displays values of the Code buffer on the screen.
2. Data area
 - It displays values of the Data buffer on the screen.
 - If the target device does not have data memory, this button is disabled.
3. Enter address.
 - Enter the start address to display on the screen.
 - The start address to display on the screen consists of 256 units ranging from 0xXX00 to 0xXXFF.
 - For example, if you enter '0x1234' in this field, values in the buffers at the addresses in the range 0x1200 to 0x12FF are displayed.

4. Go!

- It moves to the address you enter and displays the value of the corresponding buffer.

5. Dump window

- It displays the value of the buffer at the specified address.

5 Output messages

5.1 Output messages from the PC GUI

The output messages from the PC GUI are often the same as those displayed on the OLED screen of the aLinkPro. Therefore, the output message from the PC GUI is output in abbreviation form so that it can be displayed on the OLED screen of the aLinkPro that limits the display character length.

The table below shows the abbreviations of the output messages. They are listed alphabetically.

Abbreviation of output message	Meaning
AFLM error	Failed to find the Flash downloader file to transfer to the device.
Blnk Cfg	Checks if the Cfg area of the device is blank.
Blnk Code	Checks if the Code area of the device is blank.
Blnk Data	Checks if the Data area of the device is blank.
Cfg Err	The value of the device's the Cfg area is different from that in the PC buffer.
Code Err	The value of the device's the Code area is different from that in the PC buffer.
Data Err	The value of the device's the Data area is different from that in the PC buffer.
Down FLM	Transfers the Flash downloader program to the device.
Ers Cfg	Erases the Cfg area of the device.
Ers Code	Erases the Code area of the device.
Ers Data	Erases the Data area of the device.
Ers error	Attempted to erase the device and it failed.
F/W error	Failed to find F/W files to transfer to use the aLinkPro as Standalone mode.
Init err	Failed to initialize the device to Program mode.
Locked	Read protection is enabled on the device.
NoBlank Cfg	The Cfg area of the device is not blank.
NoBlank Code	The Code area of the device is not blank.
NoBlank Data	The Data area of the device is not blank.
OK time	The process for the device was successfully completed and the 'time' was spent on that process.
Pgm Cfg	Programs the Cfg area of the device.
Pgm Code	Programs the Code area of the device.

Abbreviation of output message	Meaning
Pgm Data	Programs the Data area of the device.
Read Cfg	Reads the Cfg area of the device.
Unlock error	Attempted to disable Read protection of the device and it failed.
Vfy Cfg	Compares the value in the Cfg area of the device and that in the PC buffer.
Vfy Code	Compares the value in the Code area of the device and that in the PC buffer.
Vfy Data	Compares the value in the Data area of the device and that in the PC buffer.
Vfy FLM	Checks for the Flash downloader program that is transmitted to the device.
VID error	Failed to read the Vendor ID of the target device.

5.2 Output messages on aLinkPro

If the aLinkPro is connected to the PC and controlled using the GUI, the aLinkPro displays a message similar to the output message from the PC GUI. However, if the aLinkPro operates in Standalone mode, it displays the messages shown in the table below.

- When the operation is processed normally, the message in blue characters is displayed on the black background.
- If the error is detected during the operation, the message in black characters is displayed on the blue background.

The table below shows the messages displayed when the aLinkPro operates in Standalone mode. It lists the messages alphabetically.

Message	Meaning
APID x or DPID X	The connected device is a product that has the unintended Cortex series. Example) I intended to program the A31 series but the A33 series is connected.
CPU Halt & Reset	Failed to change to write-allowed mode and accordingly, failed to continue to operate.
devID err! data	Attempted to program the target device set in the aLinkPro and other devices. data is the ID of the currently connected device.
DBG PWR REQ FAIL	Attempted to enter Debug mode to program the device, but it failed.

Message	Meaning
Device name Code:Hex	The Device name and Code checksum that are stored in the aLinkPro Available when the target device does not have Data area. The hex value is the Code checksum.
Device name Data:Hex	The Device name and Data checksum that are stored in the aLinkPro Available when the target device has Data area and mode in which only data is programmed is set on the PC GUI if the aLinkPro operates in Standalone mode. The hex value is the Data checksum.
Device name Hex,Hex	The Device name and Data checksum that are stored in the aLinkPro Available when the target device has Data area and mode in which entire area including code and data is programmed is set on the PC GUI if the aLinkPro operates in Standalone mode. The hex value is the Code and Data checksum.
Download ABOV FLM	Transfers the Flash algorithm to the device for programming the device.
Down err ABOV FLM	Transferred the Flash algorithm to the device for programming the device. But the device denied receiving it.
Erase... Wait	Erasing the device.
Erase X address	Attempted to erase the device but it failed.
HaltTarget OK / No	Attempts to stop the target device for programming it, <ul style="list-style-type: none"> • If the target device stops, OK is displayed. • If the target device does not stop, No is displayed.
Invalid Hex data	All data values to program are the same value. Example) All data is filled with '0x00'.
LOCKED Do again !	Read protection is enabled on the device. Try again.
PGM cfg	Programs the Cfg area of the device.
PGM cfgX address	Attempted to program the Cfg area of the device but it failed.

Message	Meaning
PGM code Start	Starts programming Code area.
PGM code Address	Programming the Code area of the displayed address.
PGM data Start	Starts programming Data area.
PGM data Address	Programming the Data area of the displayed address.
PGM X address	Error occurs while programming the device.
VFY cfg	Checks if the Cfg values were written in the device.
VFG cfgX address	Programmed the device with Cfg but the written value is different from the original.
VFY code	Checks if the programmed value in Code area and the original data are the same and verifies that they are written correctly.
VFY data	Checks if the programmed value in Data area and the original data are the same and verifies that they are written correctly.
VFY X address	The value written to the device is different from the original.

6 Command process

6.1 Overview

Command process is a special mode in which you can enter commands on the prompt of the command window or use the functions of this program in the program that you develop separately, without clicking each button on the PC GUI.

For example, if you enter programming options in a certain device name or hex file name, it calls dl program and then closes itself automatically after processing the command.

6.2 Functions and formats of command

Assuming that this program's name is SWName, this section describes functions and formats of each command. The actual file name is 'aFlasher32_CSharp.exe' or can be different.

Command input system

- The commands must follow this program's name. First calling this program allows the command to be identified and processed.
- The command starts with '/' and is not case sensitive.

The table below shows the commands.

Command	Operation
/P	Program device
/V	Verify device
/E	Erase all Flash in device
/B	Blank check
/?	Help

Directives

- Directives must follow the command.
- The directive starts with '-' and is not case sensitive.
- Depending on the command, the directives in association with the command vary.
- Each directive is not affected by the input order.

The table below shows the directives.

Directives	Operation
-d	Target device name
-f	Hex file name
-o	Option (Cfg) value of target device

6.2.1 Device programming

This function programs the target device's Flash memory with the value in the hex file.

Command structure

- SWName /P -d deviceName -f hexFileName -o optValues...

Input statement	Description
SWName	This program's name
/P	Command (Program)
-d devName	Target device's name
-f hexFileName	Hex file name
-o optionValues...	Option (Cfg) value of the target device If the Cfg. is included in the hex file or is not required, this can be omitted.

Example use of this command

- SWName /P -d A31G112 -f test,hex-o 12 0x34
 - This command programs the A31G112 device with the values of test.hex file.
 - The option is available for the decimal and hexadecimal numbers.
 - The hexadecimal numbers must start with '0x'.

6.2.2 Device verification

This function compares the value of the target device's Flash memory and the value in the hex file.

Command structure

- SWName /V -d deviceName -f hexFileName -o optValues...

Input statement	Description
SWName	This program's name
/V	Command (Verification)
-d devName	Target device name
-f hexFileName	Hex file name
-o optionValues...	Option (Cfg) value of the target device If the Cfg. is included in the hex file or is not required, this can be omitted.

Example use of this command

- SWName /V -d A31G112 -f test.hex -o 12 0x34
 - This command compares the value in the test.hex file with the device named A31G112.
 - The option is available for the decimal and hexadecimal numbers.
 - The hexadecimal numbers must start with '0x'.

6.2.3 Device erase

This function erases the entire value in the target device's Flash memory.

Command structure

- SWName /E -d deviceName

Input statement	Description
SWName	This program's name
/E	Command (Erase)

Example use of this command

- SWName /E -d A31G112
 - This command erases the entire value in Flash memory of the device named A31G112.

6.2.4 Device check for blank

This function checks if the target device's Flash memory is completely erased.

Command structure

- SWName /B -d deviceName

Input statement	Description
SWName	This program's name
/B	Command (Blank check)

Example use of this command

- SWName /B -d A31G112
 - This command checks if the Flash memory of the device named A31G112 is completely erased.

6.2.5 Help

This function shows how to use a command.

Command structure

- SWName /?

Input statement	Description
SWName	This program's name
/?	Command (Help)

Example use of this command

- SWName /?

7 Precautions for programming

7.1 Precautions for connecting target devices

Signal lines

- The aLink or aLinkPro is connected via the target system and SWD interface. The lines connecting it are VCC, VSS, SWCLK, and SWDIO.
- Although the nReset of the SWD interface line is not mandatory, it is recommended to connect it to easily control and program the device.
- Boot pin:
If the device performs a user program, it is highly possible for the SWD pin cannot be used for debugging process because it operates as a GPIO pin intended by the user. In this case, the target device cannot be connected to the aLink or aLinkPro. Therefore, it is recommended that you set the Boot pin as Boot mode and then connect the dongle under the condition in which the device is not performing a user program.

7.2 Precautions for preventing damages

Power

- Whether used as PC mode or Standalone mode, the aLink and aLinkPro are always connected to the target system's power source. This can cause damage to the dongle if a problem occurs on the power supply of the target system. In severe cases, power of the target system comes in through the USB line and damages the PC. To prevent such damage, use the aLink or aLinkPro added with isolation circuits.
- Hot plug:
The aLink and aLinkPro do not contain protection circuits for Hot plugs. Therefore, if you connect a dongle while the target system is turned on, they may be damaged.

Mass production

- **Because the aFlasher32 is designed as a low-cost product without a protection circuit, it is not suitable for mass production except for programming prototypes.**
 - If you want to mass-produce, it is recommended to use a dedicated programmer.
 - Example programmer: ABOV's universal programmer EPGM+

Revision history

Revision	Date	Notes
2.000.00	April, 2023	Initial release
2.000.01	May, 2023	Upload to homepage

Korea

Regional Office, Seoul

R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200

Fax: +82-2-508-6903

www.abovsemi.com

Domestic Sales Manager

Tel: +82-2-2193-2206

Fax: +82-2-508-6903

Email: sales_kr@abov.co.kr

HQ, Ochang

R&D, QA, and Test Center
93, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do, 28126, Korea

Tel: +82-43-219-5200

Fax: +82-43-217-3534

www.abovsemi.com

Global Sales Manager

Tel: +82-2-2193-2281

Fax: +82-2-508-6903

Email: sales_gl@abov.co.kr

China Sales Manager

Tel: +86-755-8287-2205

Fax: +86-755-8287-2204

Email: sales_cn@abov.co.kr

ABOV Disclaimer

IMPORTANT NOTICE – PLEASE READ CAREFULLY

ABOV Semiconductor ("ABOV") reserves the right to make changes, corrections, enhancements, modifications, and improvements to ABOV products and/or to this document at any time without notice. ABOV does not give warranties as to the accuracy or completeness of the information included herein. Purchasers should obtain the latest relevant information of ABOV products before placing orders. Purchasers are entirely responsible for the choice, selection, and use of ABOV products and ABOV assumes no liability for application assistance or the design of purchasers' products. No license, express or implied, to any intellectual property rights is granted by ABOV herein. ABOV disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of ABOV products in such unauthorized applications. ABOV and the ABOV logo are trademarks of ABOV. All other product or service names are the property of their respective owners. Information in this document supersedes and replaces the information previously supplied in any former versions of this document.

© 2023 ABOV Semiconductor – All rights reserved