WriteNow! Series

Single and Parallel In-System Programmers

Additional Document for User's Manual

Programming Algorithm for Renesas

78K family

AN00010200EN - Rev. 1.00

Introduction

This Application Note provides WriteNow! programming commands' specifications for Renesas 78K family.

WriteNow! Technology allows to program up to 8 devices at once, by reducing, in a drastical way, programming times, costs and system complexity. Algocraft provides support of Renesas 78K family through all of its WriteNow! Series, single and parallel In-System Programmers. Our four WN programmers are: WN-PRG01A (programs 1 device by one), WN-PRG02A (programs 2 devices in parallel), WN-PRG04A (programs 4 devices in parallel), and WN-PRG08A (programs 8 devices in parallel). In addition, we provide external Demultiplexer modules as accessories for the multiple-sites WriteNow! models, able to extend the programming channels up to 32 in a sequential way (8+8+8+8).

Read the WriteNow! User's Manual included in the software before to use the WriteNow! Programmer.

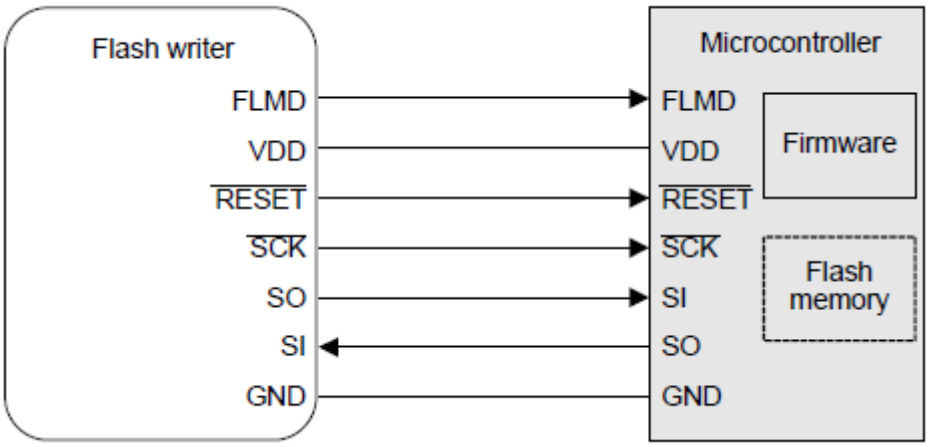
78K Family Overview

The extensive 78K series lineup provides the ideal product for almost every application. For a compact, low-power MCU, choose one of the products in the 8-bit 78K0 and 78K0S Series, and for a controller that offers the performance of a 32-bit MCU but consumes the power of an 8-bit one, look no further than the 16-bit 78K0R.

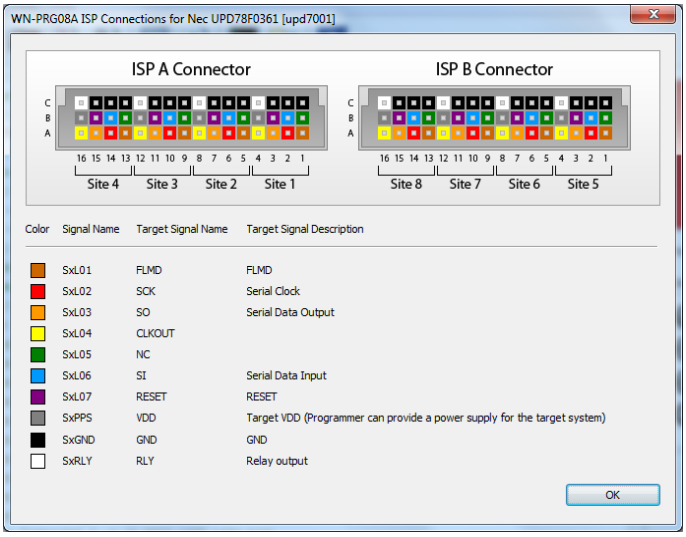
Communication Protocols

The Programmer supports two types of protocols: CSI and UART.

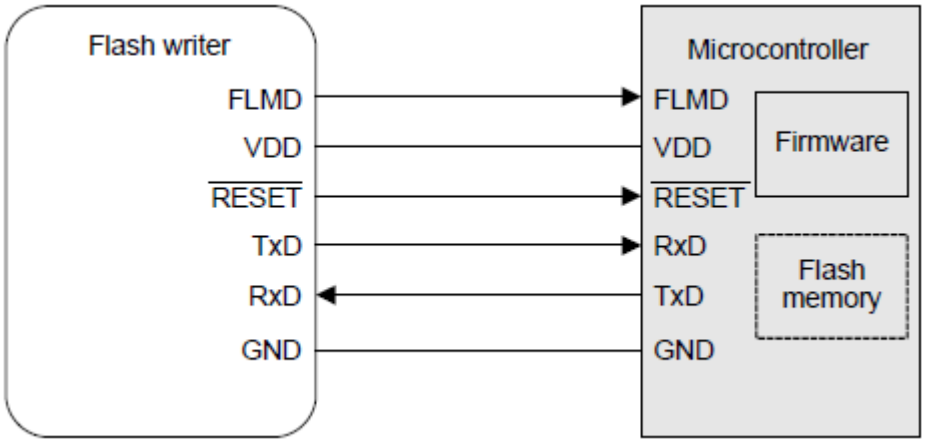
CSI protocol:



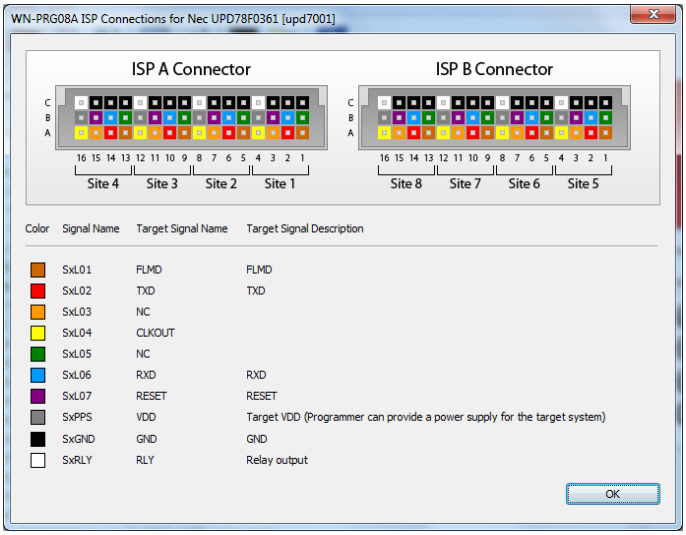
In the CSI protocol, the Programmer is always the master of the communication. For this communication, the SCK, SO, and SI pins are used. Data must be handled in 8-bit units, with the MSB first. In order to view the connections for your selected target device, select Debug > Show ISP Connections from the WriteNow! GUI interface as below:



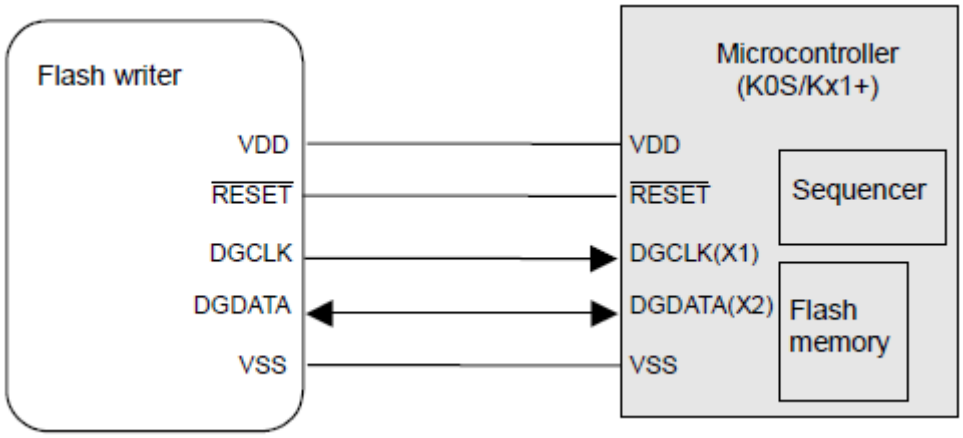
UART protocol (78K0, flow-A type):



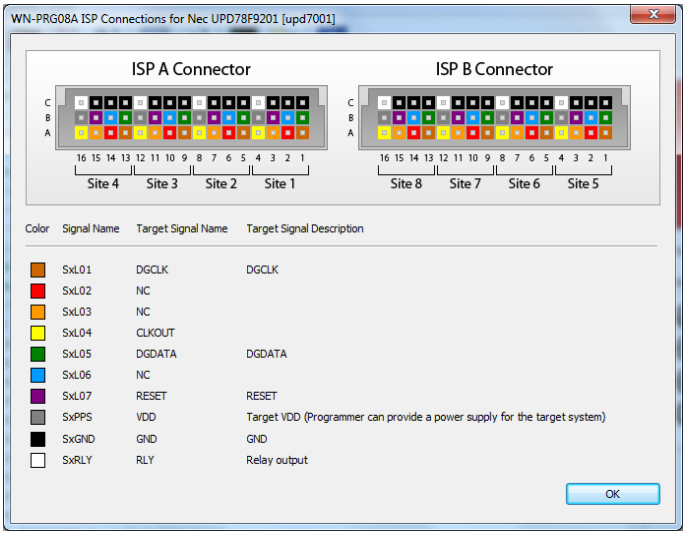
About UART protocol, the Programmer is always the master of the communication. For this communication, the TxD and RxD pins are used. Data must be handled in 8-bit units, with the LSB first. In order to view the connections for your selected target device, select Debug > Show ISP Connections from the WriteNow! GUI interface as below:



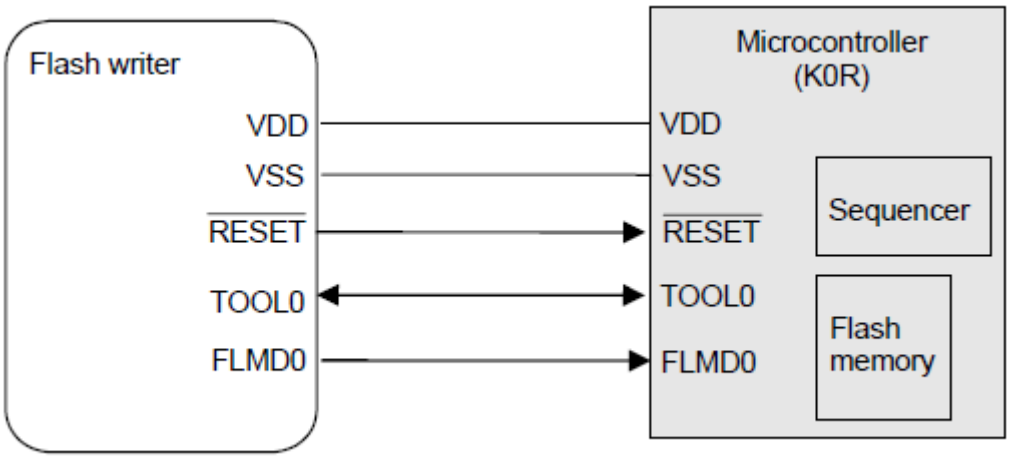
UART protocol (78K0S, Single-wire, flow-B type):



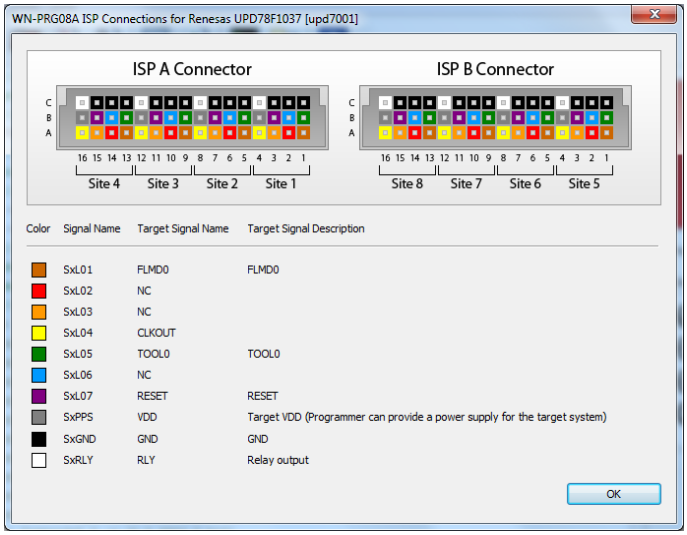
This communication System is exclusively for the 78K0S/Kx1+. The DGDATA pin is used for communication. Data must be handled in 8-bit units, with the LSB first, baudrate at 115200 bps. In order to view the connections for your selected target device, select Debug > Show ISP Connections from the WriteNow! GUI interface as below:



UART protocol (78K0R, Single-wire, flow-C type):



The TOOL0 pin is used for communication. Data must be handled in 8-bit units, with the LSB first. In order to view the connections for your selected target device, select Debug > Show ISP Connections from the WriteNow! GUI interface as below:



Data Formats (Flash Process Flow)

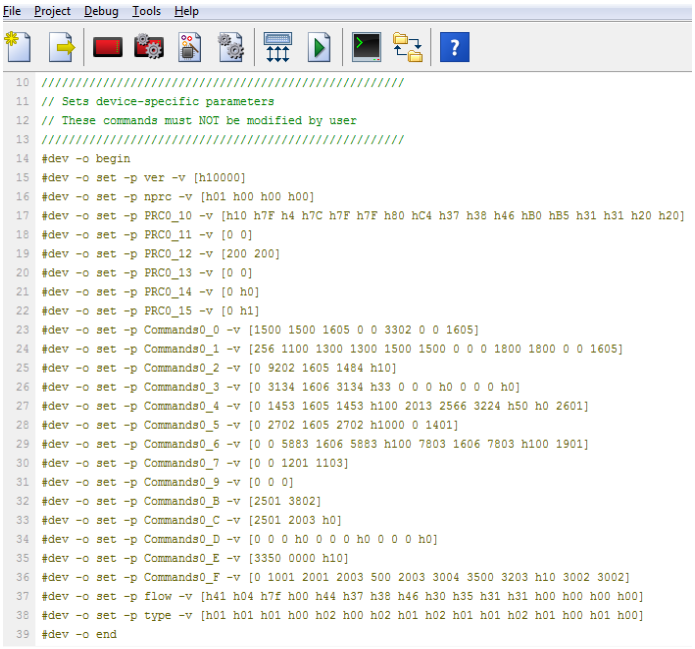
The Renesas 78K family uses different programming flow based on the following table:

Signature (Byte 2-3)	Flash Macro	Device	Flash Process Flow
7F-01	CZ6HSF	78K0	Flow-A
7F-04	MF2	78K0	Flow-A
7F-04	MF2	78K0R	Flow-C
EF-04	MF2	78K0R(2nd)	Flow-C
None	CZ6HSF	78K0S	Flow-B
DF-04	MF2	78K0(2nd)	Flow-D

Each Flash Process Flow differs from each other in command/communication data format and in processing. More information can be found on the Renesas "Single Voltage Flash MCU Standard Specifications for Serial Writer" Manual (Not disclosed to users).

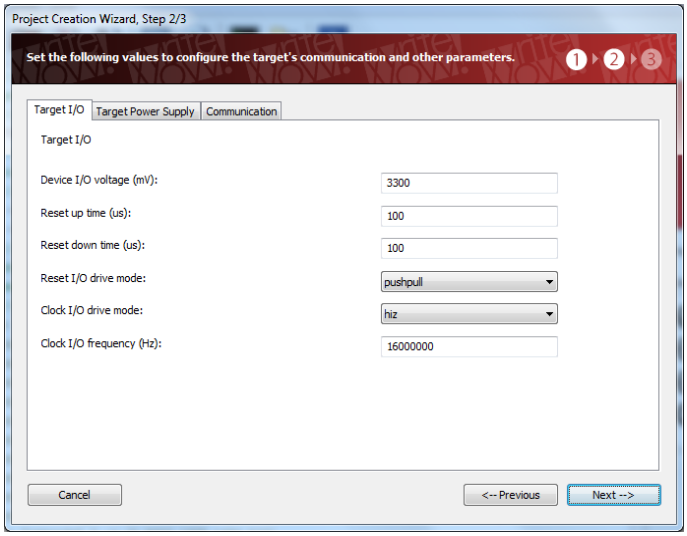
WriteNow! Device Settings (PR5 file)

WriteNow! Programmer uses the settings of the Renesas Parameter File (PR5 file) used by the Renesas PG-FP5 Flash Memory Programmer. The PR5 file contains important information about communication process such as Wait's Parameters for send/receive command and data, Wait's Parameters for blank Check, Erase, Program, Verify and read operations (delay, timeout, retry number), memory map address, signature string configuration, etc.



WriteNow! Project Settings

All device-specific settings, target Parameters and connection values are set by WriteNow! GUI interface (Project Creation Wizard). The Wizard will automatically fill all data with typical values for the selected target device.



The **"Device I/O voltage"** setting specifies the voltage of the ISP lines. You should check the target board schematics, or ask the board developer about this value. The allowed voltage also depends on the selected target device.

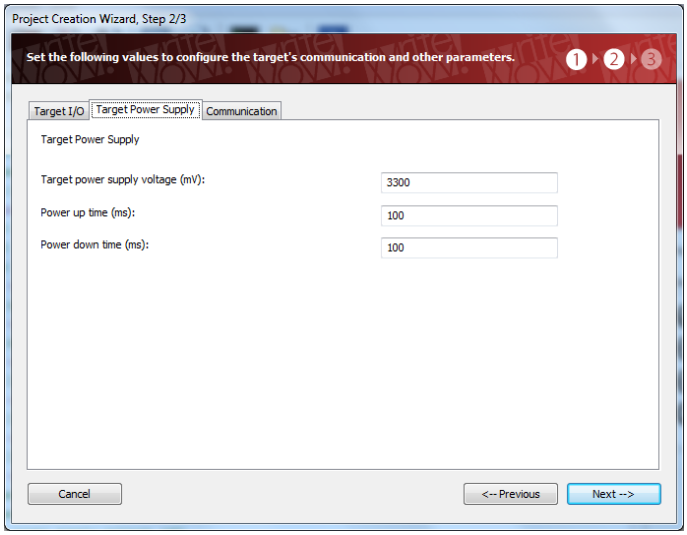
The **"Reset up time (us)"**, **"Reset down time (us)"** and the "**Reset I/O drive mode**" settings grant you to configure the RESET pin for the establishing communication between the programmer and device.

The **"Clock I/O drive mode"** and **"Clock I/O frequency"** settings allow you to decide how the SxL04 ISP line is driven. This line can be used as an auxiliary ISP line

(to provide a clock to the target device), as a generic I/O line, or as a high-impedance output (no electrical driving). We suggest leaving this line floating (HiZ) when not used, in order to decrease the electrical noise on other ISP lines.

The Wizard will automatically fill the commands in the project file (wnp) as below:

```
#conf -o begin
// Target I/O settings
// Target I/O
// Set Device I/O voltage (mV)
#conf -o set -p vddio -v 3300
// Set Reset up time (us)
#conf -o set -p rstup_time -v 100
// Set Reset down time (us)
#conf -o set -p rstdw_time -v 100
// Set Reset I/O drive mode
#conf -o set -p rstio_drive -v pushpull
// Set Clock I/O drive mode
#conf -o set -p clkio_drive -v hiz
// Set Clock I/O frequency (Hz)
#conf -o set -p clkio_freq -v 16000000
```



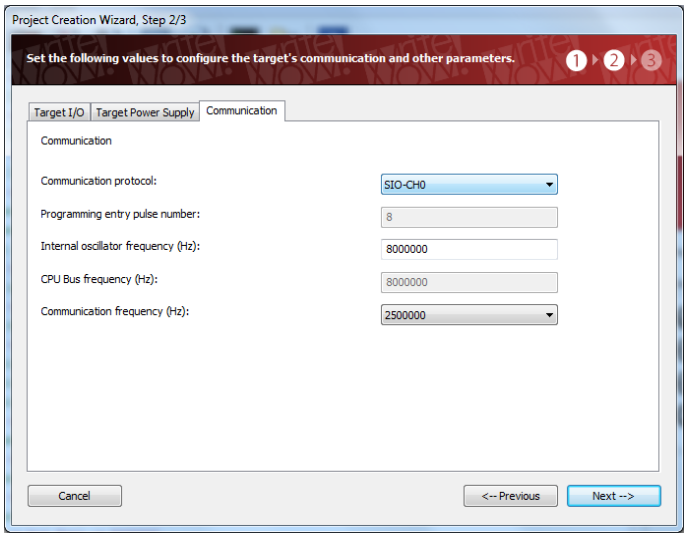
Target Power Supply tab controls the programmable power supply line available for each ISP site.

The voltage of the programmable power supply line ("**Target power supply voltage**" setting) can be in the range 1700mV to 13000mV..

The "**Power up time**" and the "**Power down time**" settings specify the delay between the programmable power supply line turning on / off and subsequent operations.

The Wizard will automatically fill the commands in the project file (wnp) as below:

```
59 // Target Power Supply settings
60 // Target Power Supply
61 // Set Target power supply voltage (mV)
62 #conf -o set -p pps_voltage -v 3300
63 // Set Power up time (ms)
64 #conf -o set -p pps_uptime -v 100
65 // Set Power down time (ms)
66 #conf -o set -p pps_dwtime -v 100
```



Communication tab sets the parameters for the protocol. This windows can change from device to device based on programming process flow.

The "**Communication protocol**" specify the type of protocol, for Renesas 78K family the protocol could be UART or SIO(SPI) protocol.

The "**Programming entry pulse number**" is the number of pulses on FLMD0 pin to set the correct "Microcontroller Operating Mode". By executing the flash firmware, the flash microcontroller reads the number of pulses stored in the built-in timer/counter to decide on the communication system. This settings can not be edited by user.

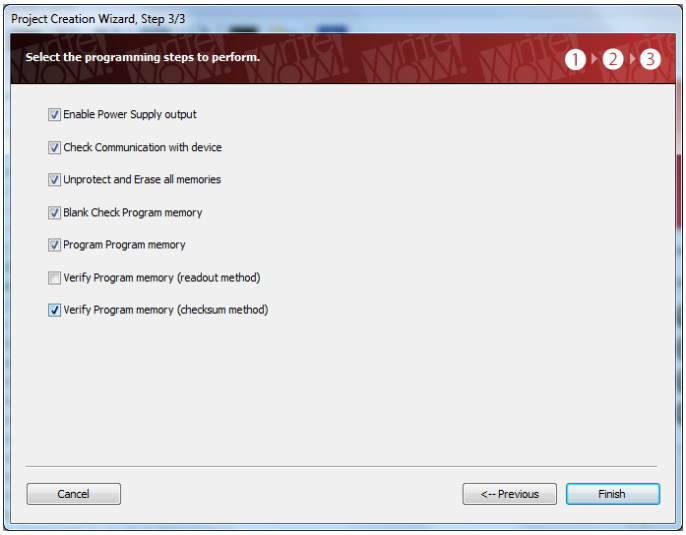
The "**Internal/External Oscillator frequency**" and "CPU bus frequency" specifies the values (in Hz) of the oscillator frequency and CPU bus frequency (not edited).

Finally the "**Communication frequency**" or the "**Baudrate**" set the speed of the communication. Maximum bitrate speed achievable is affected by wiring environment used; refer to Algocraft's WP00010101EN Understanding ISP Wiring white paper for further more information.

The Wizard will automatically fill the commands in the project file (wnp) as below:

```
// Communication settings
// Communication
// Set Communication protocol
#conf -o set -p protocol -v SIO-CH0
// Set Programming entry pulse number
#conf -o set -p entry_pulse_num -v 8
// Set Internal oscillator frequency (Hz)
#conf -o set -p osc_freq -v 8000000
// Set CPU Bus frequency (Hz)
#conf -o set -p bus_freq -v 8000000
// Set Communication frequency (Hz)
#conf -o set -p bitrate -v 2500000
```

In the next step you select which programming operation to perform on the target.



Enable the "**Enable Power Supply output**" option in order to power the board by the WriteNow! SxPPS power line.

The "**Check Communication with device**" option establishes a communication between WriteNow! Programmer and target device. This option should be used during the project setup in order to be sure that a good wiring and electrical parameters has been reached.

The "**Unprotect and Erase all memories**" option removes the device protection and erase the entire memory.

The "**Blank Check Program memory**" option checks if the device memory is blank.

The "**Program Program memory**" option program the device memory.

The "**Verify Program memory (readout method)**" command is used to compare the data, in a specified address range, sent from the programmer with the data read from the device (read level) to confirm whether they match.

The "**Verify Program memory (checkum method)**" command is used to verify the device memory using the checksum method.

```
// Power Supply on
#prog -o cmd -c pps -v on
// Target Connection
#prog -o cmd -c connect
// Unprotect and Erase all memories
#prog -o cmd -c erase -m all
// Blank Check memory
#prog -o cmd -c blankcheck -m flash -t h0000 -l h30000
// Programs memory
#prog -o cmd -c program -m flash -s h0000 -t h0000 -l h30000
// Verifies  memory (checksum method)
#prog -o cmd -c verify -v chks -m flash -s h0000 -t h0000 -l h30000
```

Note: The "**Read Program memory**" command is not available on the Renesas 78K family.

78K0R/Fx3 Data Flash Programming

Data Flash of this family cannot be programmed directly.

Following the Renesas specification, the programming of the Data Flash needs to follow these steps:

- Program the Microcontroller with a special firmware (provide by Renesas)
- Execute the firmware in normal mode
- In this mode, the firmware will copy the data from 0x2000 address to Data Flash area.
- Program the Microcontroller with a second project with the user firmware.

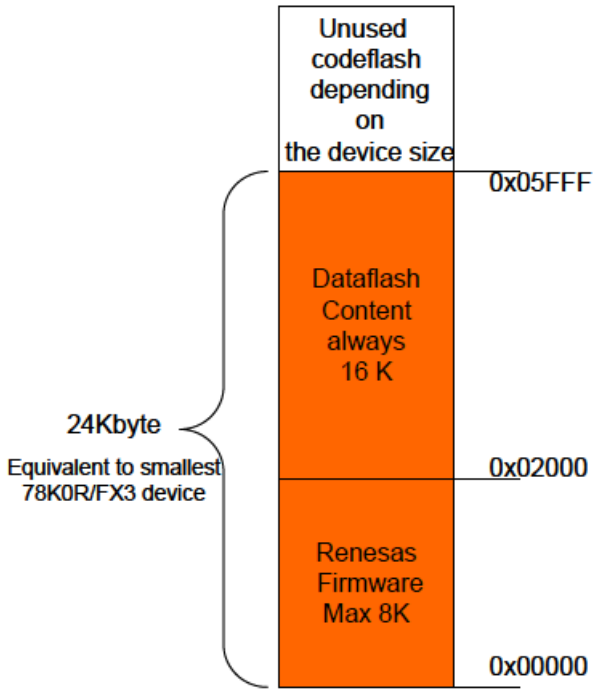
In WriteNow! Project generator GUI follow these steps:

Step1:

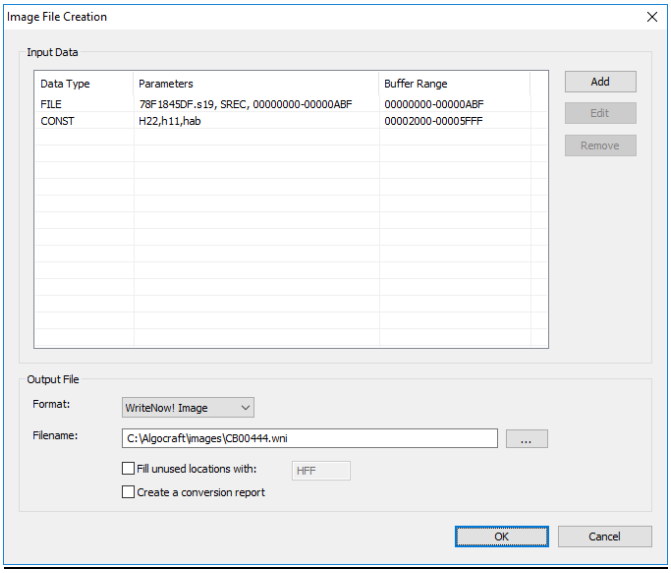
A special .wni file must be created. At first you have to add the right .s19 file selected from \drivers\uk\78F18XXDF list. This file is the Renesas firmware.

Then you have to add the binary file of the DataFlash. The data must be loaded from 0x2000 address, according the following picture:

Mapping of the codefflash area

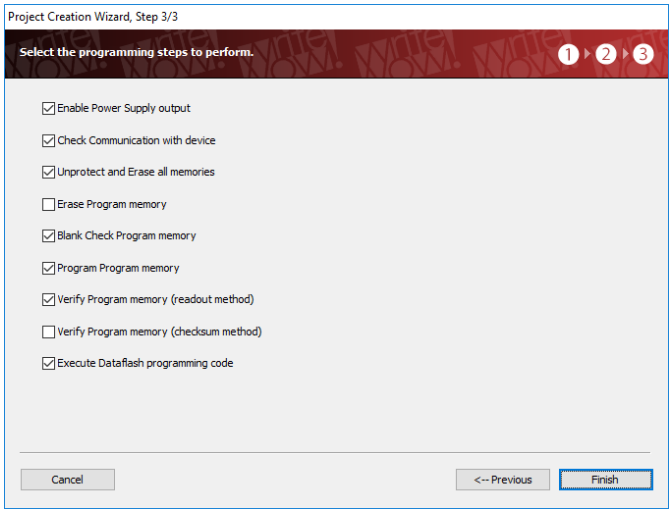


The dataflash content which is stored from address 0x02000 onwards is the pure binary data without ID tags, addresses, and checksum. The Renesas firmware will copy this data one by one to the real dataflash.



Step2: Project settings and programming steps

See the picture below:



It is important to enable the "**Execute DataFlash programming code**" command. This command executes the Microcontroller in normal mode and the firmware will copy the data from 0x2000 address to Data Flash.

Programming Times

The following table shows programming times for selected Renesas 78K family devices. Programming times of your current System may therefore be different from the ones listed here below.

Device	Protocol / Speed	Memory Size	Operations	Times (s)
UPD78F0393	CSI @250000	32KB	E+BC+P+V	7 s
UPD78F0502	CSI @250000	24KB	E+BC+P+V	5,2 s
uPD78F1845	UART @ 2000000	256KB	E+BC+P+V (Checksum)	10,1 s

References

- Algocraft WriteNow! Series User's manual
- Algocraft Understanding ISP Wiring White Paper
- Renesas Specification for Fx3 Dataflash Programming (Confidential)
- Renesas Single Voltage Flash MCU Standard Specifications for Serial Writer (Confidential)
- Renesas PG-FP4 System Specification (Confidential)
- Renesas device-specific User's Manual

Conclusion

Algocraft WriteNow! Programmer's Series represents a professional Solution involved, above all, in the automotive field by many important customers. WriteNow! In System Programming solution has been consolidated for more than 5 years and it keeps on being strengthened thanks to many installations on various In Circuit Test machines, representing a competitive advantage compared to other solutions of the worldwide market.

We work daily to ensure and keep on affirming high standard. **Why?** The continuous improvement, the maintenance of the highest quality and programming standards by adding new devices, by trying to improve our performances, our programming times…are our goal!

How? We are able to develop new algorithms upon request by customers enlarging

our device list, we are capable to better our performances, we try to beat our programming times continuously, challenging at first ourselves!