

Interfacing FlashRunner 2.0 with STM8 devices

- Driver Version: 2.11

Algorithm Notes

This algorithm works with SWIM_100M fpga. The protocol name is SWIM.

See the document: *UM0470: STM8 SWIM communication protocol and debug module* for further information.

This FPGA has been created in order to support the parallelism: the main problem with the ported version was that the entry sequence was performed with delays: see the image below.

Due to communication problems related with the ACK sending/receiving, the new fpga structure is based on BDM protocol structure: All the frame sending and receiving are handle by the SWIMCSR register. See the STM_A_HAL_SWIM_StartOperation function. See the *SWIM_100M wiki* for further information.

Swim command summary:

- **ROTF**: Read on the Fly
- **WOTF**: Write on the Fly
- **SRST**: System reset

Every read/write byte or buffer use the functions above: care about the endianness and the buffer casting from 32bit values (from fpga register FPGA_REG_SWIMRX) to 8bit values for (higher level).

After every byte sent by FR2.0, the device return an ACK bit.

After every byte sent by the device, the fpga send an ACK bit.

Important difference from LAN algorithm: **A delay between ACK received and next byte sending has been introduced**. A delay after byte received and ACK send has been introduced. Check the function STM_A_HAL_SWIM_SetBitDur for further information related with the byte structure. Note that the delays has been introduced based on *STM8S Tool*.

Another important difference from LAN algo:

- STM_A_Slow_One_Low_Dur has been changed from 2 to 4
- STM_A_Fast_One_Low_Dur has been changed from 2 to 3

Additional Parameters List:

- **BIT_TYPE** (HIGH/LOW) – Select the Speed Bit Format: HIGH/LOW speed - Default: HIGH
- **PRESC_ENABLE** - Enable/Disable Prescaler

BIT_TYPE:

- **BIT_TYPE HIGH** -> One bit is generated with ten SWIM clock pulses
- **BIT_TYPE LOW** -> One bit is generated with twenty-two SWIM clock pulses

PRESC_ENABLE:

- **PRESC_ENABLE YES** -> Set **Default SWIM communication frequency** based on synchronization command

- `PRESC_ENABLE` NO -> Double the Default SWIM communication frequency - *Device related*

The `OPTION_BYTE` handle has been changed: Once selected the correct memory map structure for O, select the correct values using the `OB_DEF_VAL_0_15` & `OB_DEF_VAL_16_31`. Basically, instead of adding 26char values for every OB address, it has been introduced a string of 16 hex u_char. Use the 'TPCMD DEBUG' command to have a list of *ob_address - ob_default_value*.

`HSI_FREQ` – HSI Frequency - Default: 16000000 = 16MHz, has been removed from driver management: Frame Bit duration are based on frequency obtained from sync command - HSI can be around 16Mhz up to 20MHz.

Alignments:

- **ERASE:** start_addr & size must be `erasePageSize` aligned
- **PROGRAM:** start_addr & size must be `progPageSize` aligned (64 or 128 bytes aligned)

An external pull-up resistor (~330Ω) is required on SWIM line. Pull-up resistor value may be different, compatibly with voltage and current specifications of SWIM pin.

DIE_ID

Only the **STM8S105xx** and **STM8S207xx** part numbers do support the `DIE_ID` read operation. The `DIE_ID` is an unique part number Identifier, as reported in table below:

Part Number	DIE_ID Address	DIE_ID Value
STM8S105xx	0x67F0	0x373942
STM8S207xx	0x67F0	0x373941

Note: `DIE_ID` check cannot be performed with `ROP` (Readout Protection) active.

Algorithm Flow

Commands List

Memory types:

F → Flash

E → Eeprom

O → Option Byte

TPCMD

#TPCMD MASSERASE <F|E>

This function perform a Masserase for F, E or whole memory

#TPCMD ERASE <F|E> <start_addr> <size>

Single Erase for F, E

start_addr & size must be `erasePageSize` aligned

#TPCMD BLANKCHECK <F|E> or **BLANKCHECK** <F|E> <start_addr> <size>

Blankcheck command for Flash, Eeprom - Auto or Manual mode allowed

#TPCMD PROGRAM <F|E|O> or **PROGRAM** <F|E|O> <start_addr> <size>

Program command for F, E, O - Auto or Manual mode allowed

start_addr and size must be progPageSize aligned

#TPCMD VERIFY <F|E|O> <R> or VERIFY <F|E|O> <R> <start_addr> <size>

Verify command for F, E, O - Auto or Manual mode allowed.

R – Normal verify with Read-Out method

start_addr and size must be progPageSize aligned

#TPCMD READ <F|E|O> <start_addr> <size>

Read function for F, E, O – No Address check

#TPCMD DUMP <F|E|O> <start_addr> <size>

Dump function for F, E, O – Address check

#TPCMD RUN or TPCMD RUN <delay(sec)>

Run in User Mode command

#TPCMD ROP_CHECK

This command returns the state of Read-Out Protection.

#TPCMD RESTORE_OB

This command restore the options bytes to their default value.

#TPCMD UNPROTECT

This command remove the ROP and perform a chiperase.

#TPCMD SWIM_RST

Perform a SRST: system reset command

#TPCMD CONNECT

Connect function: Power on, entry and ROP Check

#TPCMD DISCONNECT

Disconnect function: Power off and exit