Interfacing FlashRunner 2.0 with AVR8 devices

Supported Protocols

• **SPI**: protocol supported on every device. Check section SPI Frequencies

• **JTAG**: protocol not supported on every device. This protocol is not affected by the internal frequency of the device, and thus it can reach higher frequencies. It is recommended to use the JTAG protocol instead of SPI, unless the device does not support JTAG programming, or if the JTAG pins are not accessible on your board

Supported Memories

• **Flash**: main memory of the AVR devices. It is word aligned, and so each address corresponds to 2 bytes each. Keep this in mind when converting your source files to frb

• **EEPROM** data memory of the devices. It is not necessary to erase it before re-programming it, but it is strongly suggested to erase it anyway. Some devices have the EESAVE fuse bit, that prevents the EEPROM to be erased during a chip erase procedure (command MASSERASE C). In that case, be sure to set the correct value for this bit before erasing the device. In the frb files, the data for EEPROM must be offset to address 0x800000. Official tools might offset this memory to address 0x810000, but for retro-compatibility, we must keep the "incorrect" offset. To alleviate this issue, the FRB manager will automatically remap address 0x810000 (and following) to address 0x800000 (and following), so that the user does not need to manually remap the EEPROM blocks while creating the FRB file.

• **FUSE BITS** configuration bits of the device, that might change the behaviour of the device. After being programmed, they do not take effect until the device is reset.

The most notable fuses, at least for what concerns the programming procedures, are Fuse bits (and Lock bits) are usually included in common source files, but they cannot be specified inside the frb. They can instead be programmed and verified with the commands PROG_FUSE, PROG_LOCK, VERIFY_FUSE and VERIFY_LOCK, where the values of each byte is specified in the command itself. While converting your firmware to frb, you will need to remove the data blocks of the Fuse (and Lock) bits, and instead specify the data directly on the commands PROG_FUSE and VERIFY_FUSE.

- CKSEL: selects the clock source of the device. If your device does not have an external oscillator connected and you set the source to be an external clock, you might not be able to communicate again with the device.
- EESAVE: if 0, EEPROM memory will not be erased during a Chip erase
- JTAGEN and SPIEN: enable or disable JTAG and SPI communications. They cannot be set on their respective protocols (i.e. JTAGEN cannot be modified if you are programming the device in JTAG).

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

- CKDIV8: divides the source clock by 8. To speed up the SPI protocol, the driver will automatically try to disable this fuse.

- **LOCK BITS** byte that controls the lock features of the device. A protected device can only be unlocked with a chip erase procedure (command MASSERASE C). As for fuse bits, they cannot be specified inside the frb file. You can add their programming using the commands PROG_LOCK and VERIFY_LOCK

Algorithm Commands

The letter after the command names specifies on which memory region the command is to be executed:

- F: Flash
- E: EEPROM
- C: entire chip (used only for the command MASSERASE)

Many commands support the parameters <start_addr> and <size>, that specify to execute the command, respectively, from which address and for how many addresses.

CONNECT

Powers-on the device, enters programming mode and checks the connection.
Warning: some devices do not have a device ID, and thus there is not a reliable way to know if the device is correctly connected to the FlashRunner2.0. In that case, the programming cycle could fail during the commands BLANKCHECK or VERIFY.

MASSERASE

MASSERASE <C>

Erases the Flash memory. If the fuse bit EESAVE is set, the EEPROM memory will be erased as well. If you want to also erase the EEPROM, be sure to first set the EESAVE fuse bit

BLANKCHECK

BLANKCHECK <F|E>, or BLANKCHECK <F|E> <start_addr> <size>

Checks that the selected memory is clear.

PROGRAM

PROGRAM <F|E>, or PROGRAM <F|E> <start_addr> <size>

Programs the selected memory using the data loaded into the FRB file.

VERIFY

VERIFY <F|E> R, or VERIFY <F|E> R <start_addr> <size>

Verifies that the data written into the selected memory corresponds to the data of the FRB file.

PROG_FUSE

PROG_FUSE <FLB|FHB|EFB> <value>

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

Programs the selected fuse byte (FLB: fuse low byte, FHB: fuse high byte, EFB: extended fuse byte) with the specified value

VERIFY_FUSE

VERIFY_FUSE <FLB|FHB|EFB> <value>

verifies that the selected fuse byte was programmed with the specified value.

PROG_LOCK

PROG_LOCK LKB <value>

Programs the lock bits with the value specified to control the lock features of the device. A protected device then can only be unlocked with a chip erase procedure.

VERIFY_LOCK

VERIFY_LOCK LKB <value>

Verifies the value programmed in the lock bits.

READ

READ <F|E>, or READ <F|E> <start_addr> <size>

Reads the selected memory, printing it into the communications log of the workbench.

DUMP

DUMP <F|E>, or DUMP <F|E> <start_addr> <size>

Reads the selected memory, dumping it into the dumping file.

READ

READ <FUSE|LOCK|SIGNATURE>

Reads the selected byte(s).

READ <CALIBRATION> <Calibration Address/Frequency>

Reads the calibration value of the oscillator corresponding to the selected byte address (or frequency).

SAVE_CALIBR_VALUE

SAVE_CALIBR_VALUE <Target Memory> <Target Address> <Oscillator Frequency/Address>

Reads the factory calibration value of the internal oscillator, and writes it on the specified target memory and address. Some devices have more than one oscillator, each running at different frequencies. So, you can decide which calibration value to read/write using the Oscillator Frequency parameter, by specifying the frequency of the desired oscillator. Specifying the calibration address (instead of the calibration frequency) is also supported.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

CALIBRATE_RC

CALIBRATE_RC <Target Memory> <Target Address>

Executes the RC calibration procedure to find the calibration value that better approximates the expected clock frequency of the internal oscillator. This value is then stored in the dynamic memory of FlashRunner2.0, so that it will then be programmed (on the selected target memory and address) during the PROGRAM command, alongside the rest of the frb.
This command is currently not supported on every device. Contact SMH support if you need this functionality on a currently non-supported device.

RUN

RUN <seconds>

Releases the device from programming mode, so that the programmed code can be executed. A delay (in seconds) can be specified.

DISCONNECT

Powers-off the device.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

SMH Technologies®

SMH Technologies S.r.l.

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com