

# Interfacing FlashRunner 2.0 with TI\_UCD devices

## Supported Protocol

- I2C

## Supported Memories

- Program Flash
- Code Flash

## ROM Mode and Program Mode

ROM Mode is the state in which the device is when no user code is programmed. In this state, the boot ROM is executed, and the device responds to boot ROM commands at the slave address 0xB. Program Mode is the state in which the device is when it is already programmed or when the command RUN is used. Its slave address and its behaviour will be dictated by the user code. In this state, only the commands PMBUS\_WRITE and PMBUS\_READ can be used.

A device in Program Mode can be brought back to ROM Mode (and so it can be re-programmed) by the driver during the CONNECT command, if the parameter KEEP\_PROGRAM\_MODE is set to "NO". To maintain the Program Mode of a device, set the parameter KEEP\_PROGRAM\_MODE to "YES". When the device is in Program Mode, the driver will use the protocol frequency given with the command PROGRAM\_MODE\_CLK. If it is not set, the communication frequency will be lowered to 1/4 of the frequency used for commands in ROM Mode.

## DevDatas

- **CHECKSUM\_SIZE**: the checksums handled by these devices can either be 4bytes or 8bytes long. The 2 checksum functions are:
  - 4bytes checksum: sum each byte of the memory section, storing the result in 32 bits
  - 8bytes checksum: sum each word (4 bytes) and store the result in 64 bits

## Additional Parameters

### KEEP\_PROGRAM\_MODE

: Setting this parameter to "YES", the driver will not disable the Program Mode of the device, if it is found in this mode. When the device is kept in Program Mode, the standard programming commands (MASSERASE, PROGRAM, READ ...) cannot be used, as they require the device to be in ROM Mode. Only the commands PMBUS\_WRITE and PMBUS\_READ can be used in this mode.

### DEVICE\_ADDRESS

: slave address of the device when in Program Mode. Use this parameter if the I2C line has more than one slave. If this parameter is not set, the driver will search for every possible slave address and select the first one it finds.

## PROGRAM\_MODE\_CLK

: Protocol frequency when the device is in Program Mode. It should be lower than the normal protocol frequency specified with the parameter PROTCCLK (which is used in ROM Mode).

## Algorithm Commands

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

- F: Program Flash.
- E: Code Flash.
- C: Entire chip (only for 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 and checks the connection to the Boot ROM of the device.

If the device is already programmed, it is possible that its Boot ROM is disabled. If no device in ROM Mode is found, the programmer will try to find a device in Program Mode.

## MASSERASE

**MASSERASE** <C|F|E>, or **BLANKCHECK** <F|E> <start\_addr> <size>

Erases the selected memory. MASSERASE C erases both Flash memories.

## BLANKCHECK

**BLANKCHECK** <F|E>, or **BLANKCHECK** <F|E> <start\_addr> <size>

Checks that the selected memory is blank (all to 0xFF)

## 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>, or **VERIFY** <F|E> <start\_addr> <size>

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

## READ

**READ** <F|E>, or **READ** <F|E> <start\_addr> <size>

Reads the selected memory, printing it into the terminal.

The OTP Options will also be printed in the log file using the OTP\_OPTIONS structure

## DUMP

**DUMP** <F|E>, or **DUMP** <F|E> <start\_addr> <size>

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

## RUN

**RUN**, or **RUN** <seconds>

Start execution of Flash code. If the flash code of the device is valid, the device will go in Program Mode. If the parameter is provided, the driver will wait for the specified time before starting the next command

## PMBUS\_WRITE

**PMBUS\_WRITE** <command> <data1> ...

Transmits a generic PMBUS byte sequence to the device.  
Additional header bytes (Ex: block size) must be added as data bytes

The PEC byte is automatically added.

The data will be transmitted LSB first

The data length is automatically calculated from the number of data entries and their format

Examples:

- writing **PMBUS\_WRITE** 0xAA will only transmit the command byte 0xAA
- writing **PMBUS\_WRITE** 0xAA 0x11 will transmit the command byte 0xAA and the data byte 0x11
- writing **PMBUS\_WRITE** 0xAA 0x0011 will transmit the command byte 0xAA and the data bytes 0x11 and 0x00
- writing **PMBUS\_WRITE** 0xAA 0x00 0x11 will transmit the command byte 0xAA and the data bytes 0x00 and 0x11

## PMBUS\_READ

**PMBUS\_READ** <command> <data\_len>, or **PMBUS\_READ** <command>

Receives a generic PMBUS byte sequence from the device, printing it to the terminal.

If the <data\_len> parameter is not specified, the driver will assume that the first received byte is the data length.

The PEC byte is not counted inside <data\_len>, nor is it printed to the terminal.

## DISCONNECT

Powers-off the device.