Chipsmall

Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

ASCII command interface to Flasher V2.00

Introduction

Since firmware version 1.84, Flasher 4, Flasher 4R1, Flasher 5 and Flasher ST7 support a simple ASCII command interface via RS232.
This document describes the ASCII command interface of Flasher firmware version 2.00
Once set up using Flasher PC program, Flasher can be driven by any application or just a simple terminal using ASCII commands.

Every known command is acknowledged by Flasher and then executed.
After command execution, Flasher sends an ASCII reply message.
If an unknown command is received, Flasher responds with #NACK

General command and reply message format

Any ASCII command has to start with the start delimiter #.
Any ASCII command has to end with simple carriage return (ASCII code 13)
Commands can be sent upper or lower case.
Reply messages from Flasher are sent back using the same format. All characters are upper case in reply messages.

Communication port settings

Flasher is driven via RS232 serial port with the following interface settings:
8 data bits,
ODD parity
1 stop bit
at 115200 baud.

Commands to Flasher

The following commands are supported by current version of Flasher firmware:

#AUTO

The Auto command behaves exactly as the start button or external remote control input.
- Flasher starts clearing target, if "automatic clear before program" option was set.
- Flasher programs target CPU
- Flasher verifies target CPU
- Flasher verifies its internal CRC to validate the programming operation.

Finally, Flasher responds with
- **#OK** if no error occurred
- **#ERRxxx** if any error occurred during operation. xxx represents the errorcode, normally replied to Flasher PC program. The #ERRxxx message may be followed by an additional error text.

During execution of the AUTO command, Flasher automatically sends "status" messages via RS232 to reflect the state of execution. Typically during execution of #AUTO command, Flasher will reply the following sequence of messages:
#ACK
#STATUS:START CMD
#STATUS:CONNECTING
#STATUS:CLEARING
#STATUS:WRITING
#STATUS:VERIFYING
#STATUS:VERIFY FLASHER CRC
#STATUS:READY
#OK

#AUTO NOINFO

This command may be used instead of AUTO, if no status messages from Flasher should be sent during execution.
The command ends with **#OK** or **#ERRxxx**

#BLANK

This command can be sent to perform a blank check over all selected target Flash sectors.
Flasher will reply the following sequence of messages:
#ACK
#STATUS:START CMD
#STATUS:CONNECTING
#STATUS:RESET
#STATUS:CONNECTING
#STATUS:BLANK CHECK
#STATUS:READY
#OK
in case of blank check failure, Flasher will respond with
#ERR004:BLANK CHECK FAILED.

#CLEAR

This command can be sent to clear all selected target Flash sectors.
Flasher will reply the following sequence of messages:
#ACK
#STATUS:START CMD
#STATUS:CONNECTING
#STATUS:RESET
#STATUS:CONNECTING
#STATUS:CLEARING
#STATUS:READY
#OK
in case of an erase failure, Flasher will respond with
#ERR001:ERASE FAILED.
or an other error message if any communication error occured.

#CLEARALL

This command can be used with Flasher ST7 to remove the read out protection of ST7 target CPUs.
The read out protection is removed and the whole target Flash area is erased.

#START

This command can be sent to release Flashers target interface. All signals from Flasher to target will be set into high-Z mode, reset of target will be released. May be used to start target application program.
Flasher will reply the following sequence of messages:
#ACK
#STATUS:START CMD
#STATUS:READY
#OK

#CRC

This command is useful to identify or verify Flashers internal data which should be programmed to target CPU.
Flasher responds with the CRC of its current memory content:
* **#CRC:xxxx** if data is present. xxxx is a 4 digit HEX number.
* **#CRC:----** if no valid data is present in Flashers memory

#STATUS

This command can be sent any time, even during other command execution. Flasher responds with its current state. All defined state messages are described under "Flasher reply messages from Flasher" later on in this manual.

#PROGRAM

This command can used instead of #AUTO to program a target without final verification.
Whether the target is cleared before programming depends on options which were set during configuration using the Flasher PC program.

#VERIFY

This command can used to verify the target Flash content against the data stored in Flasher.

#RESULT

This command can be sent any time, even during other command execution. Flasher responds with the last result of a previous executed command, or the result of a running command.

#CANCEL

This command can be sent to abort a running program. It may take a while until the current program is actually canceled.
Flasher will respond with
#ERR007:CANCELD.

Reply messages from Flasher

The reply messages from Flasher follow the same data format as commands.
Any reply message starts with ASCII start delimiter #.
Any reply message ends with simple carriage return (ASCCI code 13).

The following reply messages from Flasher are defined:

#ACK

Flasher replies with #ACK message on reception of any defined command before the command itself is executed.

#NACK

Flasher replies with #NACK, if an undefined command was received.

#OK

Flasher replies with #OK, if a command other then #CRC, #STATUS or #RESULT was executed and ended with no error.

#CRC:xxxx

Flasher replies with #CRC: after calculating the CRC of its internal data memory. The CRC xxxx is sent back as four digit HEX number.
If Flasher does not contain valid data, Flasher replies with **#CRC:----**.

#STATUS:

Flasher replies with its current state.
The following status messages are currently defined:

Message	Description
#STATUS:READY	Flasher is ready to receive a new command.
#STATUS:INITIALIZING	Flasher performs self check and internal init.
#STATUS:CLEARING	Flasher is clearing target CPU.
#STATUS:WRITING	Flasher is writing (programming) target CPU.
#STATUS:READING	Flasher reads target CPU.
#STATUS:VERIFYING	Flasher verifies target CPU.
#STATUS:BLANK CHECK	Flasher performs blank check of target CPU.
#STATUS:IDLE	Flasher is currently idle, but current command is not completed yet.
#STATUS:START CMD	Flasher starts execution of a received command.
#STATUS:ERASE-VERIFY	Flasher performs an Erase Verify operation. Required for some CPUs only.
#STATUS:VERIFY FLASHER CRC	After programming and verifying target, Flasher verifies its internal data memory by recalculating and verifying the CRC over all data.
#STATUS:CLEAR ALL	Flasher executes "Clear Readout Protection" on ST7 CPUs.
#STATUS:CONNECTING	Flasher initializes connection to target CPU.

#ERRxxx

If any command other then **#CRC**, **#STATUS** or **#RESULT** was terminated with an error, Flasher cancels the command and replies with an error message instead of #OK message.
The three digit error number xxx is the same error code number which would normally be sent back to Flasher PC program. The error code numbers are described in the Flasher user manual.
Some error codes may be followed by colon and an additional error text.
For example:
#ERR007:CANCELED.