PM0083
Programming manual

How to program STM8T143 OTP options

1 Introduction

The STM8T143 family can be used in the default (un-configured) state. However, for some applications, certain one-time-programmable (OTP) options must be set.

This document describes the hardware and software interfaces needed to program the user selectable options on the STM8T143 devices.

Related documentation

- STM8T143 datasheet
- ST-TSLINK user manual (UM0795)

Contents

2 Hardware interface

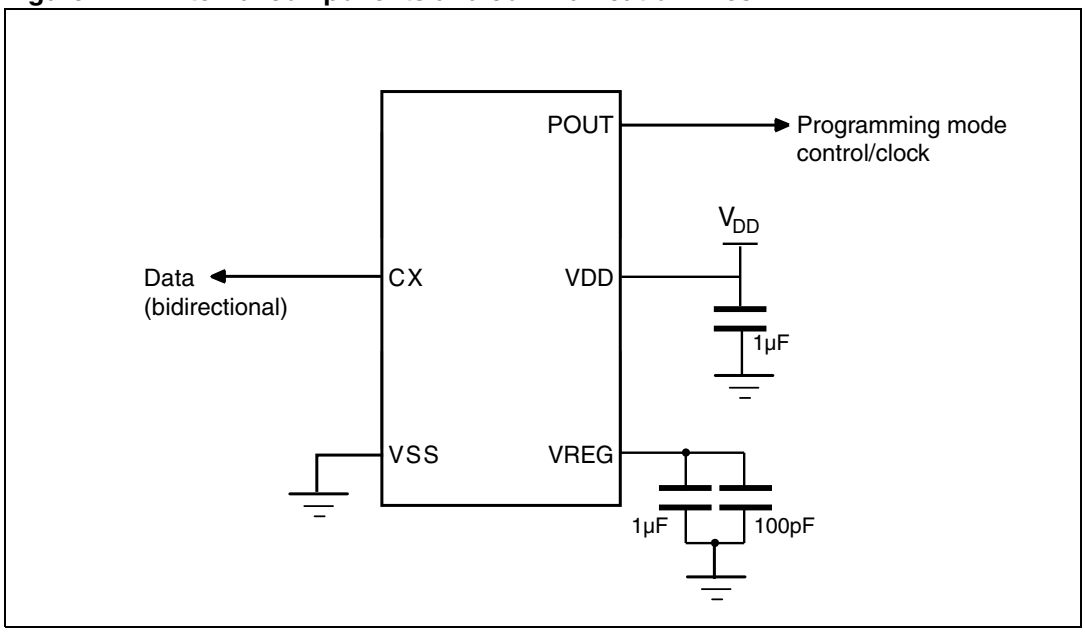
This section provides information on the hardware interface used to program the OTP options. For detailed information on other aspects of the device please refer to the STM8T143 datasheet.

2.1 External components

The STM8T143 has an internal voltage regulator that needs an external capacitor on the VREG pin to function correctly. For a cleaner external supply a 1µF capacitor and a ceramic 100 pF capacitor between VREG & VSS is also recommended. It is recommended to place a 1 µF capacitor between VDD and VSS.

Note: The VREG capacitors should be placed as close as possible to the VREG & VSS pins of the STM8T143*.*

Figure 1. External components and communication lines



2.2 Communication lines

- CX: Data (bi-directional)
- POUT: Programming mode entry clock and serial communication clock

2.3 Voltage levels for option programming and reading

The following table gives the voltage specifications. It is recommended to use the typical values when designing a programming tool.

Table 1. Voltages for reading or programming

VDD pin	Min	Typ	Max	Units
Programming mode entry	2.8	5.0	5.5	V
Option byte programming	6.0	6.2	6.4	
Option byte reading	2.8	5.0	5.5	

Caution: All I/Os must have the same voltage level as VDD.

During option programming, the supply voltage of the STM8T143 is raised to 6.2 V. It is recommended to make sure that the I/Os of the programmer can operate at 6.2 V.

2.4 In-circuit programming

When programming the device directly on the application board, the communication lines must be directly connected to the device (not through CX resistor for example).

Furthermore, in case of In Circuit Programming, the programming tools user manual must contain the maximum load current permitted to drive the application without disturbing the programming signals.

3 Software interface

3.1 Command overview

The following programming mode serial commands are used in the procedures described in this section.

Table 2. List of commands

Command	Description
0x30	Read an indirect addressed byte from RAM.
0x40	Write an indirect addressed byte to RAM (writes data to a RAM address of the device during the next two serial command communication cycles.)
0x60	Program the specified option byte.
0xF0	Request acknowledge.

3.2 Write a byte to the STM8T143

Prerequisite: POUT = HIGH and CX must be configured as input on the programming tool before calling the write procedure.

Table 3. Write procedure

Step	Action	Description
1.	Wait for CX = HIGH	Wait for device to pull CX high to indicate that it is ready to receive.
2.	POUT = LOW and wait 50 µs	Pull POUT low as part of handshaking protocol.
3.	POUT = HIGH	Pull POUT high before clocking out the data to the device.
4.	Configure CX as output and wait 50 µs	Programming tool takes control of data on CX.
5.	Clock out data to device (8 bits)	Clock out 8 data bits on CX at a clock frequency of 8.8 kHz on POUT. (Clock on POUT ten times faster than it was during programming mode entry.). Data on CX must be set up while clock on POUT is low. Most significant bit transmitted first. End sequence with POUT high.
6.	POUT = LOW	Pull POUT low after 8th bit.
7.	Configure CX as input and wait 50 µs	Programming tool releases control of data on CX and waits 50 µs.
8.	POUT = HIGH	Pull POUT high to end the writing sequence.

3.3 Read a byte from the STM8T143

Prerequisite: POUT = HIGH and CX must be configured as input on the programming tool before calling the read procedure.

Table 4. Read procedure

Step	Action	Description
1.	Wait for CX = HIGH	Wait for device to pull CX high to indicate that it is ready to transmit.
2.	POUT = LOW	Pull POUT low as part of the handshaking protocol.
3.	Wait for CX = LOW	Check if device responds to handshake with a low signal.
4.	POUT = HIGH	Pull POUT high before clocking in the data from the device.
5.	Clock in data from device (8 bits)	Clock in 8 data bits on CX at a clock frequency of 8.8 kHz on POUT. (Clock on POUT is ten times faster than it was during programming mode entry.) The most significant bit is received first. Data on CX becomes valid some time after the clock on POUT goes low and is valid by the time the clock goes high. The first seven bits keep their status until the next falling edge on the clock, but the last bit changes its state extremely quickly after receiving its rising clock edge. Therefore it is advised to sample the data just before the clock rising edge.

Note: The STM8T143 always assumes by default that it will receive a byte. It only transmits a byte if the previous command instructed it do so.

3.4 Programming mode entry

When the STM8T143 boots up after power-on, it checks for a programming mode entry sequence on POUT. If the programming mode entry sequence is recognized on POUT, it enters programming mode. If the programming mode entry sequence is not recognized, then it will enter functional mode.

Table 5. Programming mode entry procedure

Step	Action	Description
1.	Set up I/Os	CX output (Data), POUT output (Clock).
2.	All outputs = LOW	Prevent device from being powered through any of the I/Os.
3.	Wait 200 ms	Wait for all capacitors to discharge after previous test (if any).
4.	Device power ON	VDD = 5 V
5.	Change CX to an input	Configure CX as input on the programming tool so that STM8T143 can take control of CX.
6.	Wait for at least 4 ms but not more than 7 ms	Wait for device to start boot sequence and VREG to stabilize.
7.	POUT - clock 8 pulses. See *Figure 2*.	The timing on POUT is important. This timing is taken from the center frequency of the STM8T143. If the frequency is changed, or not according to specification (for example due to tampering with tuning options), then the timing needs to be scaled accordingly.
8.	POUT = HIGH	Pull POUT high to indicate to the device that the programming tool is ready to send data.
9.	Wait 560 µs	Wait for the device to respond.
10.	Wait for CX = HIGH	Check if the STM8T143 has pulled CX high. If CX is not high, then it did not enter programming mode. If CX is high then there is a good probability that it entered programming mode.
11.	Write 0xF0	Write the 0xF0 command to request an acknowledge.
12.	Read	Read the reply byte.
13.	Check if reply = 0xA5	If the result is 0xA5 then confirmation has been received from the device that it is actually in programming mode.

Figure 2. Programming mode entry clock on POUT - showing the 8 pulses

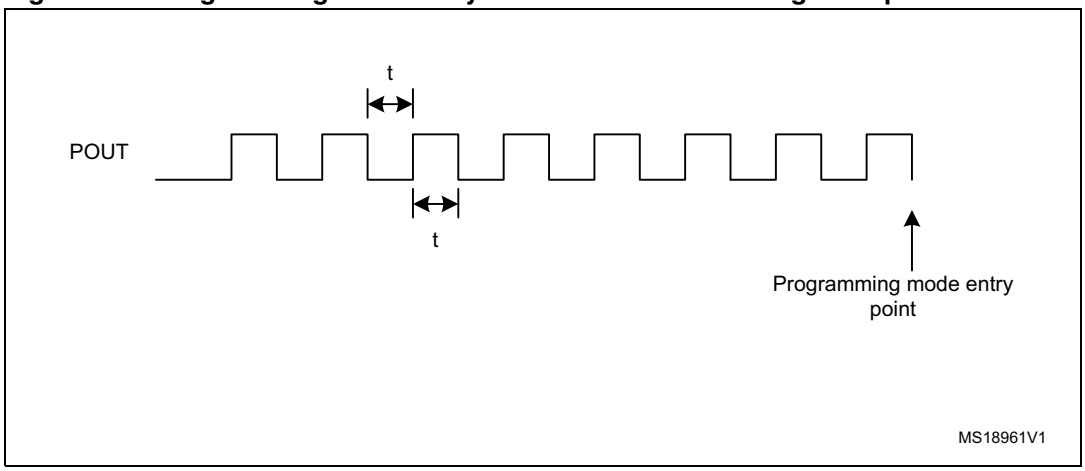


Table 6. Clock timing

	Min	Typ	Max	Unit
f_{POUT}	854	880	906	Hz
t_{POUT}	1.17	1.136	1.103	ms

3.5 Read the status of the option bytes

3.5.1 Programming mode entry

Follow the steps described in *Section 3.4: Programming mode entry on page 7*.

3.5.2 Option read

Option read checks the option status by writing a data byte to a specific RAM address of the STM8T143, and then reading the option status returned by the STM8T143. Data is written to the STM8T143 by giving it a command first, then sending it the address and finally sending it the data.

Table 7. Option read procedure

Step	Action	Description
1.	Write 0x40 to STM8T143	0x40 is the command to be given before writing data.
2.	Write 0x0A to STM8T143	0x0A is the address to write data to for option status enquiry.
3.	Write 0x1i to STM8T143 (i = 0, 1 or 2)	Send the option byte number (i) ORed with instruction 0x10.
4.	Write 0x40 to STM8T143	0x40 is the command to be given before writing data.
5.	Write 0x0A to STM8T143	0x0A is the address to write data to for option status enquiry.
6.	Write option byte #	Send the option byte number to the STM8T143.
7.	Write 0x30 to STM8T143	0x30 is the command to be given before reading data from a RAM address of the STM8T143.
8.	Write 0x0B to STM8T143	0x0B is the option data address.
9.	Read from STM8T143	Shift "dummy" data (0x00) out to STM8T143, simultaneously shifting data in from STM8T143.
10.	Repeat steps 1 to 9 for option bytes 1 and 2.	Each option byte's status (for 8 options) is read in the same manner. **Note:** Option bytes are read in the following order: byte0, byte1 and byte2.
11.	STM8T143 power OFF	VDD = 0 V

3.6 Verify the status of the option bytes

3.6.1 Programming mode entry

Follow the steps described in *Section 3.4: Programming mode entry on page 7*.

3.6.2 'Option write verify'

Option write verify is a more stringent test to check the option status. It is good practice to perform both *Option read* and *Option write verify* and compare the results in order to prove that the options have been programmed correctly.

Table 8. Option write verify procedure

Step	Action	Description
1.	Write 0x40	0x40 is the command to be given before writing data.
2.	Write 0x0A	0x0A is the address to write data to for option status enquiry.
3.	Write 0x9i (i = 0, 1 or 2)	Send the option byte number (i) ORed with instruction 0x90.
4.	Write 0x40	0x40 is the command to be given before writing data.
5.	Write 0x0A	0x0A is the address to write data to for option status enquiry.
6.	Write option byte #	Send the option byte number.
7.	Write 0x30	0x30 is the command to be given before reading data from a RAM address of the STM8T143.
8.	Write 0x0B	0x0B is the option data address.
9.	Read from device	Read option data (one byte) for this option byte.
10.	Repeat steps 1 to 9 for option bytes 1 and 2.	Each option byte's status (for 8 options) is read in the same manner. **Note**: Option bytes are read in the following order: byte0, byte1 and byte2.
11.	Device power OFF	VDD = 0 V

3.7 Programming the option bytes

3.7.1 Programming mode entry

Follow the steps described in *Section 3.4: Programming mode entry on page 7*.

3.7.2 Option set-up and programming

Options are programmed one byte at a time.

Table 9. Option programming procedure

Step	Action	Description
1.	Write 0x40	0x40 is the command to be given before writing data.
2.	Write 0x0F	0x0F is the option loop address.
3.	Write 0x04	Write number of program steps (4).
4.	Write 0x30	0x30 is the command to be given before reading data.
5.	Write 0x0F	0x0F is the option loop address.
6.	Read	Read data
7.	Reply = 0x04?	If the result is 4 (number of program steps written in step 3) then the device has accepted the data.
8.	Write 0x60	Issue *option program* command.
9.	Write option byte	Write data to the selected option byte.
10.	VDD = 6.2 V	Raise supply voltage to 6.2 V
11.	Write 0x2i (i = 0, 1 or 2)	Send the option byte number (i) ORed with instruction 0x20.
12.	Wait 40 ms	Wait 40 ms for option programming.
13.	Write 0xAi(i = 0, 1 or 2)	Send the option byte number (i) ORed with instruction 0xA0.
14.	Wait 40 ms	Wait 40 ms for option programming.
15.	Write 0xEi (i = 0, 1 or 2)	Send the option byte number (i) ORed with instruction 0xE0.
16.	Wait 40 ms	Wait 40 ms for option programming.
17.	Write 0x0i (i = 0, 1 or 2)	Send only the option byte number.
18.	VDD = 5.0 V	Lower supply voltage to normal operating voltage again.
19.	Repeat steps 1 to 18 for option bytes 1 and 2.	Each option byte's status (for 8 options) is written in the same manner. **Note**: Option bytes are programmed in the following order: byte0, byte1 and byte2.
20.	Device power OFF	VDD = 0 V

4 Revision history

Table 10. Document revision history

Date	Revision	Changes
07-Apr-2011	1	Initial release
20-May-2011	2	Replaced VREG by VDD and in 11 other places thoughout document. Updated *Section 3.5* to support option byte 3.