GANG PROGRAMMER USER MANUAL

1 INTRODUCTION

1.1 OVERVIEW

Hardware Features:

– Program the ST9XEXX and ST9XEXXX MCUs in EPROM or OTP version.

– Standalone and PC driven modes.

SOFTWARE FEATURES

– PC driven mode : menu driven software

Intel hex (or Motorola S19) file format

Description:

The gang programmer is designed for programming up to 10 EPROM or OTP microcontrollers of the ST9Xxxx family. It runs either in standalone (local) mode or remote mode under control of a DOS compatible PC.

The gang programmer is organized in 2 separate parts:

A base unit module, common to all ST9Xxxx microcontrollers, controllinoperation mode and providing the RS232C serial link for remote mode.

A dedicated module containing an interface board providing the programming voltage levels and a socket adaptor board for the specific pinout of each MCU in ST9Xxxx family. (See annex 4 for more details on gang programmer adaptors).

In standalone mode, the microcontrollers are programmed in a simple key operation directly from a master microcontroller or a master EPROM device.

Two coloured LEDs indicate the operational pass or fail and possibly defective devices.

The gang programmer also may perform empty checks and verify tests ensuring the quality of programmed devices.

In Remote mode, the gang programmer is connected to a DOS compatible PC through an RS232 serial channel. Object code in either S19 or Intel hex format is read from disk files to program the devices. The menu driven software also offers VERIFY, BLANK CHECK, READ master and other utility functions.

Ref : DOC-ST90EXX-GP Rev. 3.01

This is preliminary information on a new product in development or undergoing evaluation. Details are subject to change without notice.

1.2 EQUIPMENT SUPPLIED

- 1 ST9XExx/Exxx gang programmer.

- 1 serial communication cable 25M/25F.

- 1 adapter 9 pins/25 pins for PC/AT.

- 1 blank EPROM.

- 1 floppy 3"1/2 containing the software.

Front panel description:

On the front panel are located :

– 10 sockets dedicated to the slaves (devices to program) numbered from 1 to 10.

– 2 coloured LEDs associated to each slave.

– 1 socket repered MCU REFERENCE in which a master MCU can be inserted.

– 1 socket in 32 pin DIL package repered EPROM REFERENCE in which an EPROM memory can be inserted. The size of this device is selectable thanks to a rotary switch.

– 1 POWER FAIL LED illuminating when a default is detected on power supply.

– 1 BUSY LED indicating that the gang programmer is running.

– 7 keys allowing the selection of operation mode and function to perform.

1.3 STARTUP PROCEDURE

– Power on the gang programmer thanks to the power ON/OFF switch located on the rear panel. During 2 secondes the POWER FAIL LED must illuminate waiting for a good stabilization of power supply.

– Then select the operation mode (local or remote) thanks to the REMOTE key. When pressing this key you can toggle between the 2 modes. When the key LED illuminates the REMOTE mode is selected, otherwise it is the stand-alone mode.

2 WORK ENVIRONMENT

2.1 TESTERS AND EQUIPMENTS

All tester, equipment and tools used at any production step or for any manipulation of semiconductor devices must have its shield connected to GROUND.

2.2 WORK STATION

The antistatic work station is composed of:

– A conductive table top, made of steel or clean aluminium or covered by an antistatic surface (superficial resistivity equal or superior to 0.5 Megohm/cm2), grounded through a ground cable (conductive cable from protected equipment to ground isolated through a 1 Megohm resistor placed in series).

– An antistatic floor covering grounded through a conductive ground cable (with serial resistor between 0.9 and 1.5 Megohm).

2.3 MANIPULATION OF FINISH GOODS

Manipulation of finish goods must be made at a grounded work station (see 2.1 and 2.2).

It is mandatory to wear (either on wrist or ankle) an antistatic bracelet, connected to the antistatic floor covering or to the grounded equipment.

It is mandatory to wear antistatic gloves or finger coats.

Nylon clothing is prohibited during manipulation of parts.

The work station must be free of any non antistatic plastic objects.

The wearing of the antistatic bracelet must be controlled every day.

3 REMOTE MODE

3.1 HARDWARE INSTALLATION

The following steps must be performed.

– Power off the PC computer and the gang programmer.

– Using the interface cable connect the serial interface port COM1 or COM2 of the PC computer to the RS232C connector located on the rear panel.

– Power on the PC and the gang programmer.

– Select the REMOTE mode by pressing the REMOTE key. The key led must illuminate to indicate the remote mode is selected.

3.2 SOFTWARE INSTALLATION

First of all do a backup of the source disk, and store it at a safety place.

It's possible to work on the floppy disk but it's better to install the sofware on the hard disk to increase speed execution following next operations :

– Insert the supplied disk in the default disk drive. (Usually drive A)

– From the DOS prompt change to the desired working directory.

– Type the following command: COPY A:*.*

The different files provided with the ST9XEXX gang programmer are the following :

- ST9GP.BAT

This is a batch file which calls the different graphics utilities and the main program

- ST9GANG.EXE

This the main program.

- ST9GANG.DEV

This file contains the list of the existing ST9X family members including their names, their different available programming addresses ranges.

- ST9GANG.LBR

This file contains the different screens used by the main program.

- BEG.COM, AFFICHE.COM, END.COM which are utility files.

The different files provided with the ST9XEXXX gang programmer are the following :

- ST9XGP.BAT

This is a batch file which calls the different graphics utilities and the main program

- ST9XGANG.EXE

This the main program.

- ST9XGANG.DEV

This file contains the list of the existing ST9X family members including their names, their different available programming addresses ranges.

- ST9XGANG.LBR

This file contains the different screens used by the main program.

- BEG.COM, AFFICHE.COM, END.COM which are utility files.

All these files must be in the same directory.

3.3 DEFAULT PATH AND FILE EXTENSION

The default path for all the sofware is the path from which the program was called.

If the desired file is in another directory the user has to type the entirely access path.

The default file extension is **.HEX** and the default file type supported is the INTEL HEX.

Nevertheless the MOTOROLA S19 format is also usable when reading a file.

3.4 PROGRAM EXECUTION

For starting the software enter the command :

ST9GP <cr> for ST9XEXX gang programmer

ST9XGP <cr> for ST9XEXXX gang programmer

First a title page will be displayed. After pressing any key, select the serial communication port which is connected to the gang.

Then, after the next key pressing, the main menu will be displayed on screen.

You can now select the desired function. Details of the functions are described on the following pages.

3.5 SELECTING AND EXITING A COMMAND

To select a command, the user can either move the cursor (inverted video bar) to the desired function with the SPACE BAR or an arrow key and press <cr>, or type the uppercase letter corresponding to the command in the menu.

Each time the cursor moves a message corresponding to the selected function appears at the bottom left of the screen.

At any time a function can be aborted by pressing **ESC** key.

3.6 FUNCTIONS DESCRIPTIONS

3.6.1 3 TYPE

This function allows the user to select the device type to be programmed.

The selection is made by moving the inverted video bar until the desired device name is reached and by pressing <cr>.

Differents informations are selected in the same time :

– minimum and maximum addresses memory range.

– version of microcontroller (OTP / Eprom).

– programming memory areas available.

– availability and size of EEPROM inside the chip.

These informations have been read from the file <file>.DEV in startup phase of remote software. (See setup file syntax in annex 2).

These setup data are set by STMicroelectronics for each family member and should not be changed by customers.

Without this file or if it is corrupted the software will be interrupted.

Notes :

– The device that has been choosen is displayed in the bottom right hand corner of screen.

– When selecting a device the default memory space selected is the EPROM space. The active memory space is marked ON in the bottom right hand corner of the screen. (The inactive is marked OFF).

– Until this function is complete, pratically all the other commands are unreachable.

3.6.2 DEV

This function allows the user to display and verify the specifications of the selected device. Hitting any key exits the function.

3.6.3 COM

This function allows the user to change the selected serial communication port in case of a mistake when starting the program.

Move the cursor to the desired port and hit <cr> to validate.

By default the selected port is COM1.

3.6.4 LOAD

This function loads the data from the selected file into the PC memory buffer.

The user is prompted for the desired filename.

If no path or file extension are provided, the default will be used.

Data will be read from the file until either the memory buffer is full, or the end of file is reached.The checksum of data will then be displayed.

If the file doesn't exist, a message will appear on screen and a new filename is prompted for.

If any problems are encountered during LOAD, an error message will be displayed at the bottom of the screen. (See error message description in annex 1 for more details)

3.6.5 RAM

This command allows, by the mean of a second menu, the user to examine and modify the content of the memory buffer.

The RAM functions are described later in this manual. (RAM menu section)

3.6.6 FILE

This function writes the data in the memory buffer to the disk using the supplied filename.

This file is stored in an INTEL HEX format.

If a file with the same name already exists, the user will be prompted to confirm erasure of the previous file.

If any problems are encountered, an error will appear at the bottom of the screen.

3.6.7 PRG

The selected memory SPACE (either EPROM of EEPROM) of the selected device will be programmed using the data stored in the memory buffer. The user will be prompt to 'overprogram' if the device is already programmed. Then the device will be programmed.

After programming each byte, the content of the device will be verified against the memory buffer byte to confirm the successfull programming.

After a correct verify operation a 'success' message will appear. If an error of programming is encountered a 'fail' message will appear on screen indicating the failing byte address.

The result message is detailed for each device present onto sockets.

If any problem is encountered an error message is displayed at the bottom of screen. (See annex 1 for more details).

More over during all the operation the led BUSY is flashing, and the programming result is also displayed on the LEDs of the gang programmer.

3.6.8 VERI

This function compares the contents of the selected memory SPACE of the device against the data contained in the memory buffer.

Verify is interrupted if any error is detected, and an error message is displayed indicating the error address location. (See error message description in annex 1 for more details).

Like for the PRG function the result is also indicated on the LEDS.

3.6.9 LOCK

This function allows the user to protect the device against future reading.

After selecting this function, the user is prompted to confirm the protection and then informed of the result of protection.

Note that, after the protection is implemented, all the functions performed on these devices will fail.

The result of protection is diplayed in the same way as the result of programming.

3.6.10 BLK

This command performs a blank checking of the device EPROM memory. The command is available only if an EPROM type device and the EPROM memory space are currently selected. EEPROM memory blank checking is not supported and actually does not make sense since the EEPROM space is systematically erased prior to programming.

A 'fail' or 'success' message will indicate the device state to the user.

(See annex 1 for more details on error message).

Like for the PRG function the result is also indicated on the LEDS.

3.6.11 READ

This function reads the content of the device and stores its content into the memory buffer.

After the complete operation, the checksum of data will be displayed. If any problem is encountered an error message will appear on screen.

3.6.12 SPACE

This command allows the user to select the space, EPROM or EEPROM, to be accessed.

The functions of the programmer software READ, PROG, VERIF and LOAD operate on the selected space and all the parameters used by these functions are relevant to the selected space.

Note : The selected space is marked 'ON' in the bottom right corner of the screen.

3.6.13 EXIT

This command allows the user to exit from the program to the operating system.

3.7 RAM EDIT MENU

The access to this menu is performed by the RAM function in the main menu.

It allows examination and modification of the content of the memory buffer which contains the programming data.

An area of 128 bytes beginning at the minimum memory address (as defined in the setup file <file>.DEV) is displayed on the screen.

3.7.1 ADDR

This command allows the user to display a new memory buffer area from the desired address.

Any address between the minimum and maximum device memory can be entered (in hexadecimal).

In no case the memory display will exceed the maximum device address specified in the setup file <file>.DEV .

3.7.2 EDIT

This command allows the user to modify the buffer content.

At start of function the cursor is positioned on the first address area.

Then the user can enter new data values in hexadecimal. The ASCII field will be updated at the same time as data will be modified.

Arrows keys may be used to access any location on screen.

ESCAPE key is used to exit this command to the menu.

3.7.3 FILL

A buffer area can be initialized to arbitrary hexadecimal value between two pre-selected address limits.

The user will be prompted for the start address, the end address, the hex value and a confirmation for the previous data.

3.7.4 PGUP

This command displays the previous page of 128 bytes of the memory buffer.

3.7.5 PGDN

This command displays the next page of 128 bytes of the memory buffer.

3.7.6 EXIT

This command allows the user to exit from the RAM menu and return to the main menu.

4 STAND-ALONE MODE

This mode is selected by default when powering on the gang programmer. (Nevertheless the user can return from the REMOTE mode by pressing the REMOTE key).

The remote key led must be turned off to indicate stand-alone mode selection.

4.1 STAND-ALONE MODE CONFIGURATION

-OTP key

For any operation the microcontroller version (OTP or EPROM) must be set before starting. This selection is done by the OTP key. By pressing this key you can toggle between these two versions.

When the key led illuminates the OTP version is selected otherwise the EPROM version is selected.

This selection is mandatory to distinguish between the two existing versions of devices in which the available program spaces are different.

In OTP version the top 64 bytes are reserved by STMicroelectronics for testing purposes (See component data sheets).

-EEP key

When pressing the EEP key operations will be performed ONLY ON EEPROM memory. When this key is not activated operations are performed both on EPROM and EEPROM spaces.

-LOCK key

Before programming the user must indicate if he wants to protect (or not) the devices against future reading.

If required that will be done automatically at the end of the programming phase.
his option is selectable or not thanks the LOCK key. When the key led illuminates the devices will be protected.
Once protected, the devices behaviour on the gang programmer has no more signification.
So be careful of the LOCK key status before starting a programming.

4.2 REFERENCE SELECTION

The gang programmer is mainly designed for programming the ST9XExx microcontrollers from a reference MCU.

Nevertheless a reference EPROM device containing the code to be programmed can be used for programming the slaves. This EPROM must contain all bytes related to EPROM, EEPROM and OPTION bytes if these features are present inside the devices to program.

The selection between the two masters is done automatically by the gang which first checks if a MCU reference is present and after checks for a present EPROM memory.

So the master having priority is the MCU reference.

When using an EPROM reference the user must select the corresponding size by the mean of the rotary size located nearby the 32 pin DIL socket. (See annex III for Eprom positionning on socket).

3 positions are available :

– 2764 / 27128 for EPROM 8K x8 and 16K x8.

– 27256 for EPROM 32K x8.

– 27512 /271000 for EPROM 64K x8 and 128K x8.

4.3 PROGRAMMING ST9XEXX/EXXX MICROCONTROLLERS

For programming a device the user must follow the operations described below :

– Insert the reference MCU (or EPROM) including user's program.

– Insert into the sockets the MCUs to program.

– Select the MCU version (OTP/EPROM) thanks to the OTP key.

– Select (or remove) the E2PROM ONLY OPERATION thanks the E2PROM key.

– Select (or remove) the protection option thanks the LOCK key.

– Select the PROG function by pressing the PROG key.

– Validate the function by pressing the START key.

During programming the BUSY LED is flashing.

At the end of the programming phase, a verification is automatically performed on all the devices, even those successfully programmed.

When programming is performed the result is displayed on the LEDs associated to each socket according the following code :

– GREEN LED for a successfull function .

– RED LED for a failing function.

– NO LED associated with a present device means that the MCU is not detected:

--->Check if the device is correctly inserted into the socket. If it is correct,the device is certainly defective.

– ALL SLAVES LEDS means that the reference device is not present, defective or protected.

4.4 VERIFYING ST9XEXX/EXXX MICROCONTROLLERS

For verifying a device the user must follow the operations described below :

– Insert the reference MCU (or EPROM) including user's program.

– Insert into the sockets the MCUs to verify.

– Select the MCU version (OTP/EPROM) thanks to the OTP key.

– Select (or remove) the E2PROM ONLY OPERATION thanks the E2PROM key.

– Select the VERIF function by pressing the VERIF key.

– Validate the function by pressing the START key.

During verifying the BUSY LED is flashing.

When verifying is performed the result is displayed on the LEDs associated to each socket according the following code :

– GREEN LED for a successfull function .

– RED LED for a failing function.

– NO LED associated with a present device means that the MCU is not detected:

--->Check if the device is correctly inserted into the socket. If it is correct,the device is certainly defective.

– ALL SLAVES LEDS means that the reference device is not present, defective or protected.

4.5 BLANK CHECKING ST9XEXX/EXXX MICROCONTROLLERS

For checking if the devices are not programmed the user must follow the operation described below :

– Insert into the sockets the MCUs to check.

– Select the MCU version (OTP/EPROM) thanks to the OTP key.

– Select the BLANK function by pressing the BLANK key.

– Validate the function by pressing the START key.

E2prom blank checking has no real signification, thus the blank check is performed on EPROM space only, even if the EEP key is ON.

During blank checking the BUSY LED is flashing.

When blank checking is performed the result is displayed on the LEDs associated to each socket according the following code :

-GREEN LED for a successfull function .

-RED LED for a failing function.

-NO LED associated with a present device means that the MCU is not detected:

--->Check if the device is correctly inserted into the socket. If it is correct,the device is certainly defective.

-ALL SLAVES LEDS means that the reference device is not present, defective or protected.

5 ANNEXES

5.1 ANNEX I : ERROR MESSAGES

Device not yet selected. This message means that you attemp to execute a command before selecting a device. This is required to set different parameters such as the maximum EPROM addresses.

File not yet loaded. This message appears when the commands which operate on data in PC memory buffer are called before loading data from an hexadecimal file.

No hexadecimal file. The selected file does not match either the INTEL HEX or the MOTOROLA S19 format.

Checksum error. The data contained in the hexadecimal file are corrupted. You must verify the file content.

File content exceeds EPROM size. The data file content exceeds the device MCU EPROM capacity or does not match the available memory areas.

Address > Eprom size. The specified address is greater than the maximum Eprom address.

Start address > End address.The specified starting address is greater than the end address.

Access to non available memory range. Access has been attempted to a memory range which is not in the EPROM space of the device.

No answer from gang programmer. The gang programmer does not respond to the PC. Check if the serial cable is well connected, the remote mode well selected, the right serial port is selected on the IBM-PC and in any case restart the gang programmer.

Transmission error! Check the ligne. There was a problem during data transfert. Restart the gang programmer.

Writing disk error. This error could be caused by no space left on the disk.

Attempt to write to a write protected disk. The destination disk is write-protected.

Attempt to access to an unknown unit. The specified destination is unknown to DOS system.

Drive not ready. No disk is present or detected in drive.

Drive error. General failure during disk operation.

Setup file not found. In startup phase the remote software did not find the setup data file in the working directory.

5.2 ANNEX II : SETUP FILE SYNTAX

The file content specifying the devices and their available memory areas must respect the following syntaxe :

- **For an EPROM version :**

– Name of device :	ST90E40
– EEPROM header	<E2prom addresses>
– Min. and Max. memory address in hexa :	0000 01FF
– Memory header :	<Eprom addresses>
– Min. and Max. memory address in hexa :	0000 3FFF
– Min. and Max. address of first area :	0000 3FFF
– Device separator	*

For an OTP version :

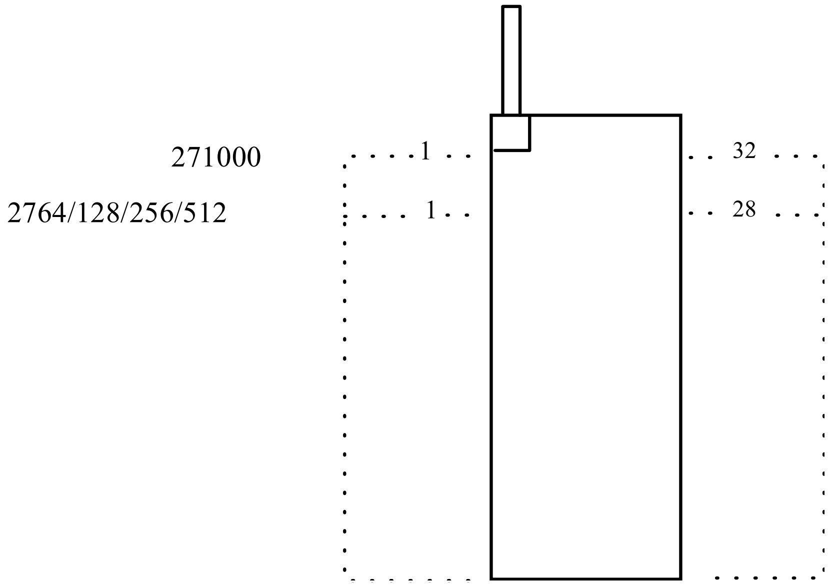
– Name of device :	ST90T40
– EEPROM header	<E2prom addresses>
– Min. and Max. memory address in hexa :	0000 01FF

– Memory header : <OTP addresses>

– Min. and Max. memory address in hexa : 0000 3FFF

– Min. and Max. address of first area : 0000 3FBF

Eprom reference location onto socket



Device separator*

Take care to respect the following points if you modify the file content :

– The different header must not be modified.

– The additional areas must be in increasing order.

– The min and max addresses of these areas must not exceed the min and max memory addresses of the device.

5.3 ANNEX III : EPROM REFERENCE

Eprom mapping

The content of the EPROM used as reference for programming the microcontroller must be exactly the same as the target microcontroller.

It means that the code used for programming the EPROM must be located inside the EPROM at the actual addresses of the EPROM inside the chip to program.

It is the same for the EEPROM content when it is available in the chip.

Hereafter is given an example for the ST90E40 microcontroller. It has to be extrapolated for other microcontroller regarding their own mapping.

MCU FeatureEPROM location

EPROM START	0000
EPROM END	3FFF
EEPROM START	4100
EEPROM END	42FF

Note: In the case where the mapping of the EPROM is split in different areas, this is taken into account by the gang programmer itself. Therefore do not take care of reserved areas inside the reference EPROM device.

5.4 ANNEX IV

5.4.1 Current ST9XEXX gang programmers description

SALES-TYPE	PACKAGE	SUPPORTED DEVICES
ST90E4X-GP/LCC68	PLCC 68 CLCC 68	ST90E30 ST90T30 ST90E40 ST90T40
ST90E4X-GP/QFP80	PQFP 80	ST90E30 ST90T30 ST90E40 ST90T40
ST90E31-GP/DIP48	DIP 48	ST90E31 ST90T31
ST90E26-GP/DIP48	DIP 48	ST90E26 ST90T26
ST90E27-GP/DIP40	DIP 40	ST90E27 ST90T27
ST90E28-GP/LCC44	LCC 44	ST90E28 ST90T28
ST90E58-GP/LCC84	LCC84	ST90E58 ST90T58
ST92E93-GP/DIP42	Shrink DIP 42	ST92E93 ST92T93
ST92E94-GP/DIP42	Shrink DIP 42	ST92E91 ST92T91 ST92E94 ST92T94
ST92E94-GP/DIP56	Shrink DIP 56	ST92E91 ST92T91 ST92E94 ST92T94
ST92E96-GP/DIP56	Shrink DIP 56	ST92E96 ST92T96
The firmware available for these gang programmer are : 222, 223, 224, 225, 226, 227 .		

5.4.2 Current ST9XEXXX gang programmers description

SALES-TYPE	PACKAGE	SUPPORTED DEVICES
ST90E158-GP/LCC84	PLCC 84	ST90E158 ST90T158
The firmware available for these gang programmer are : 222A, 223, 224, 225, 226, 227A .		

In order to support different kind of devices, the socket board of the gang programmer can be simply exchanged.

Nevertheless, it is mandatory to set some jumpers on the interface board in order to update the firmware regarding the kind of device to program.

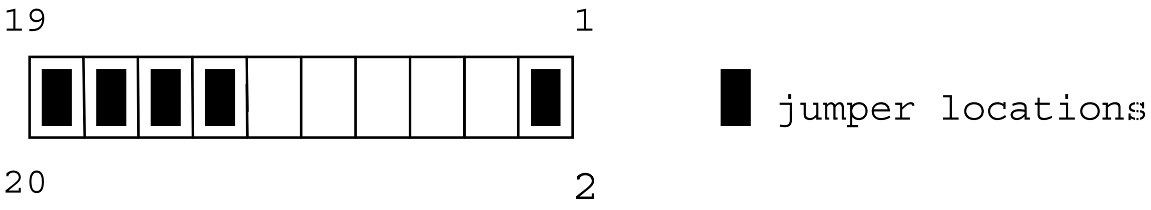
To do this modification the user must use the following operations:

- Power off the gang programmer.
- Unfast the four screws on the top module.
- Open the gang programmer.
- Remove the nuts maintaining together the interface board and the socket board located inside the top module.
- Extract the interface board.
- SET THE JUMPERS ON CONNECTOR JP1 ACCORDING THE DESCRIPTION OF THE NEXT PAGE.
- Connect the interface board to the socket board and screw the nuts.
- Close the gang programmer and screw back the top module to the bottom one.

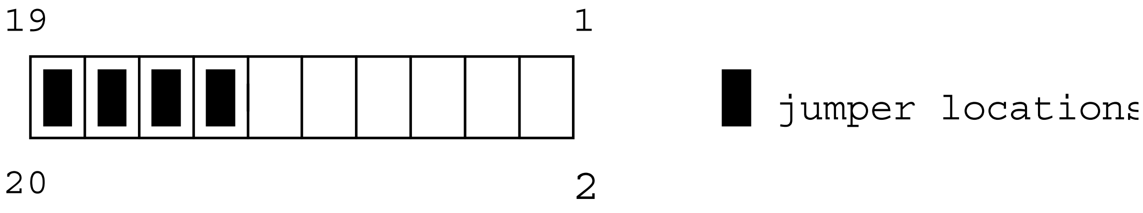
5.4.3 Jumpers locations for current ST9XEXX devices

The connector JP1 of the interface board must be configured according to the particular circuit to be used as follow:

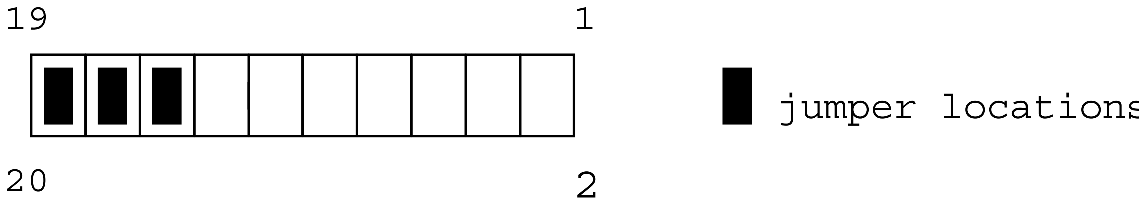
- For the components ST90E26/T26/E27/T27



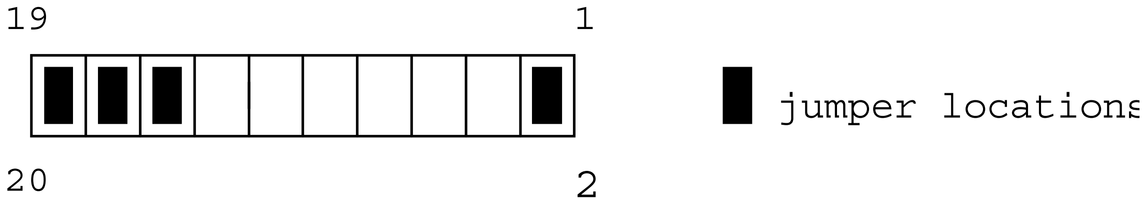
- For the components ST90E28/T28



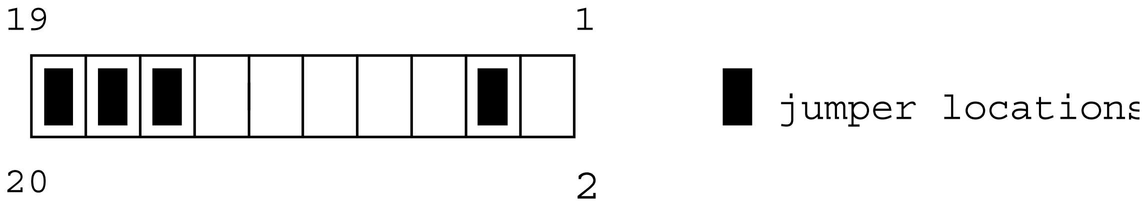
– For the components ST90E30/T30



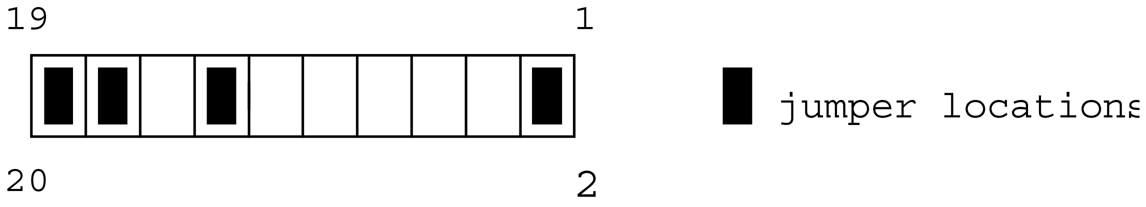
– For the components ST90E31/T31



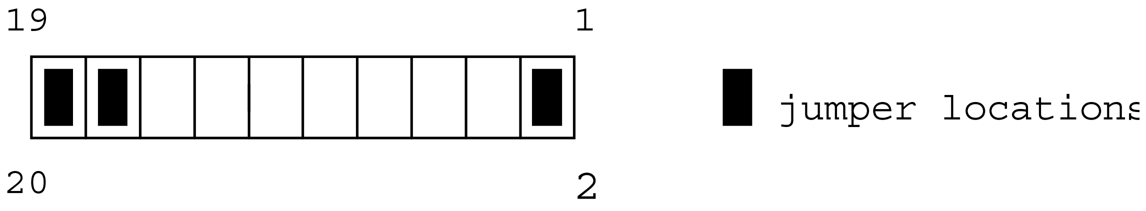
– For the components ST90ET40



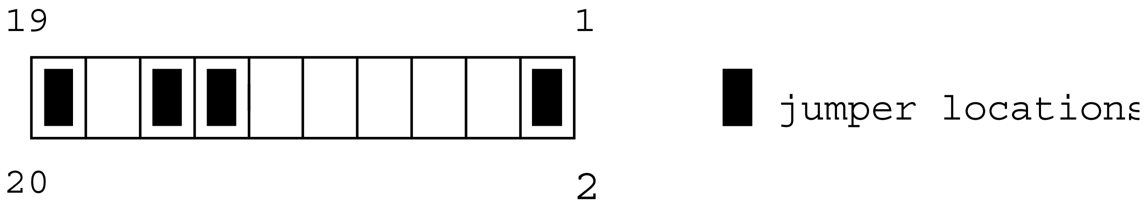
– For the components ST92E91/T91/E94/T94



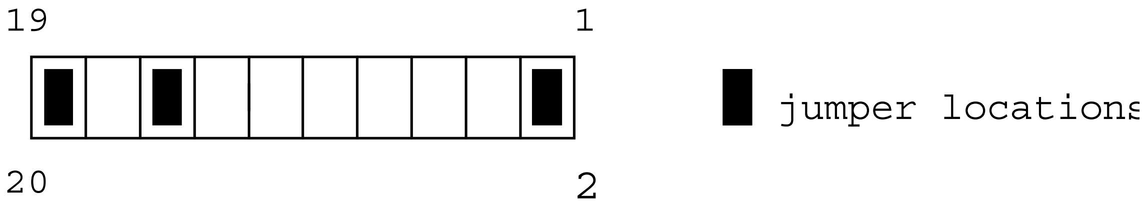
– For the components ST92E93/T93



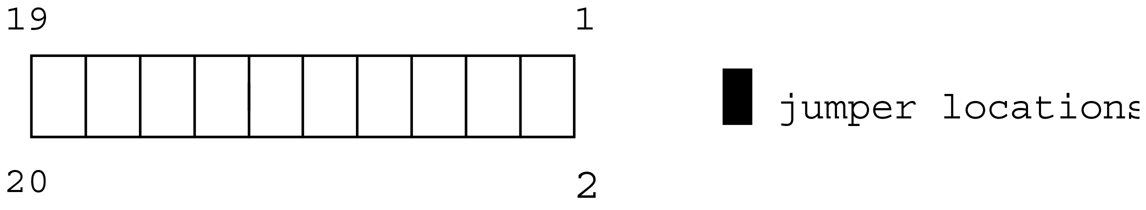
– For the components ST90E58/T58



– For the components ST92E96/T96



– For the components ST90E158/T158



THE SOFTWARE INCLUDED IN THIS NOTE IS FOR GUIDANCE ONLY. STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM USE OF THE SOFTWARE.

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of STMicroelectronics.

STMicroelectronics Group of Companies
Australia - Brazil - Canada - China - France - Germany - Italy - Japan - Korea - Malaysia - Malta - Mexico - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.