GANG

PROGRAMMER

For the ST6 Family

USER MANUAL

Release 2.0

AUGUST 2000

USE IN LIFE SUPPORT DEVICES OR SYSTEMS MUST BE EXPRESSLY AUTHORIZED.

STMicroelectronics PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF STMicroelectronics. As used herein:

1. Life support devices or systems are those which (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided with the product, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can reasonably be expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Gang Programmer for the ST6 Family

FOREWORD

The gang programmer is a tool designed for programming up to 10 EPROM or OTP microcontrollers of the ST6 family

It runs in either remote or standalone (local) mode.

In **Remote mode**, the gang programmer is connected to a compatible PC through an RS232 serial interface. Object code in either .S19 or Intel .hex format is read from disk files to program the device. The menu-driven software provides access to PROGRAM, VERIFY, BLANK CHECK, READ, and many other utility functions.

In **Stand-alone mode**, the microcontrollers can be programmed by simple key operation directly from a master microcontroller or a master EPROM device.

Two coloured light-emitting diodes (leds) are used for end-of-operation reporting: success, error, or defective device.

The programmer can also perform blank check tests and verify tests to check the quality of the programmed devices.

Table of Contents

Table of Contents

1 INTRODUCTION

1.1 OVERVIEW

The gang programmer is organised in two separate parts:

- A base unit module common to all ST6 microcontrollers, controlling operation mode and providing the RS232C serial link for remote mode.
- A dedicated module with an interface board providing the programming voltage levels and a socket adaptor board for the specific pinout of each MCU in the ST6 family.

For a list of available gang programmers, “Appendix C” on page 28.

1.2 EQUIPMENT

1.2.1 Components

The equipment comprises:

- 1 ST6 gang programmer
- 1 serial communication cable 25M/25F
- 1 adapter 9 pins/25 pins for PC/AT
- 1 blank EPROM
- 1 ”MCU ON CD” CD-ROM

1.2.2 Front Panel

The front panel comprises:

- 10 sockets dedicated to the slaves (devices to be programmed) numbered 1 through 10
- 2 coloured LEDs associated with each slave
- 1 reference MCU socket where a master CPU can be inserted
- 1 reference REPROM 32-pin DIL socket where a REPROM memory can be inserted. You can select the size of this device by operating the rotary switch located nearby.
- 1 POWER FAIL LED on when a default is detected on power supply
- 1 BUSY LED indicating that the gang programmer is running
- 7 (or 8 for MCUs supporting E2PROM) keys allowing the selection of operation mode and function to perform: OTP, LOCK, BLANK, VERIF, PROG, REMOTE, START and EEP for device supporting E2PROM.

1.3 DEVICE INSTALLATION

Place the devices into their socket with the erasure window on the top and pin 1 matching the mark on the board.

CAUTION: Placing any device into a socket without care may damage the device or the board. Never insert or remove devices when supplied. Devices are supplied only during read or write operations.

2 ANTISTATIC REQUIREMENTS

2.1 TESTERS AND TOOLS

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

2.2 ANTISTATIC EQUIPMENT

An antistatic equipment should comprise:

- A conductive table top, made of steel or clean aluminium or covered by an antistatic surface (superficial resistivity equal to or higher than 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 worktable.

It is mandatory to wear an antistatic wrist or ankle strap, 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 worktable must be free of any non antistatic plastic objects.

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

3 ST62EXX COMPATIBILITY CONSIDERATIONS

ST620X/1X/2X/5X/6X C revision devices feature new options which are available through one or two option bytes, depending on the device.

To avoid unpredictable device behaviour, option bytes must be programmed. The Gang Programmer automatically manages option byte programming, whatever the mode, remote or stand-alone, based on the device sales type (e.g. ST62E20C) or a jumper setting.

However, you will have to check carefully the revision letter (B or C) of the devices to be programmed. If the devices (reference device and target device) do not belong to the same revision family, the programming process could produce wrong results.

4 REMOTE MODE

4.1 PC CONNECTION

In remote mode the gang programmer is connected to a PC through the serial RS232 serial interface. The programming process is menu-driven.

4.2 HARDWARE INSTALLATION

Proceed as follows:

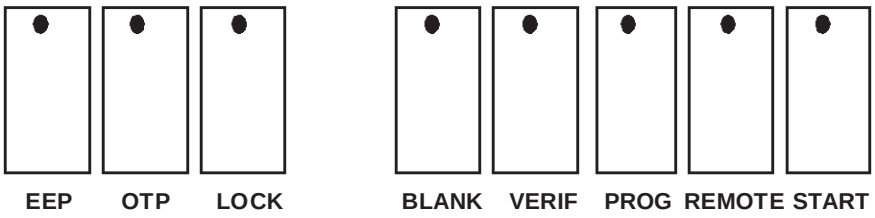
1 Power off the PC computer and the gang programmer,

2 Connect the PC serial interface port COM1 or COM2 to the RS232C connector located on the rear panel using the interface cable,

3 Power on the PC and the gang programmer, The

4 Select the remote mode by pressing the REMOTE key on the gang programmer keypad. The associated led on the REMOTE key lights up.

GANG PROGRAMMER KEYPAD



4.3 SOFTWARE INSTALLATION

The Gang Programmer comes with a CD-ROM that contains the ST6 ToolChain. To install the Windows Epromer programmer interface for starter kits, EPBs and gang programmers, follow these steps:

1 Insert the “MCU ON CD” CD-ROM into your CD-ROM drive. The CD-ROM’s autorun feature will open up a welcome screen on your PC.

If the autorun feature does not work, use the Windows File Manager or Windows Explorer to browse to the CD-ROM’s root folder, and double-click on Welcome.exe.

2 Select Install Your Development Tools from the list of options. A new screen will appear listing the different families of STMicroelectronics MCUs.

3 Use your mouse to place the cursor over the ST6 Tools option. Choose ST Tools and ST6 Toolchain from the lists that appear.

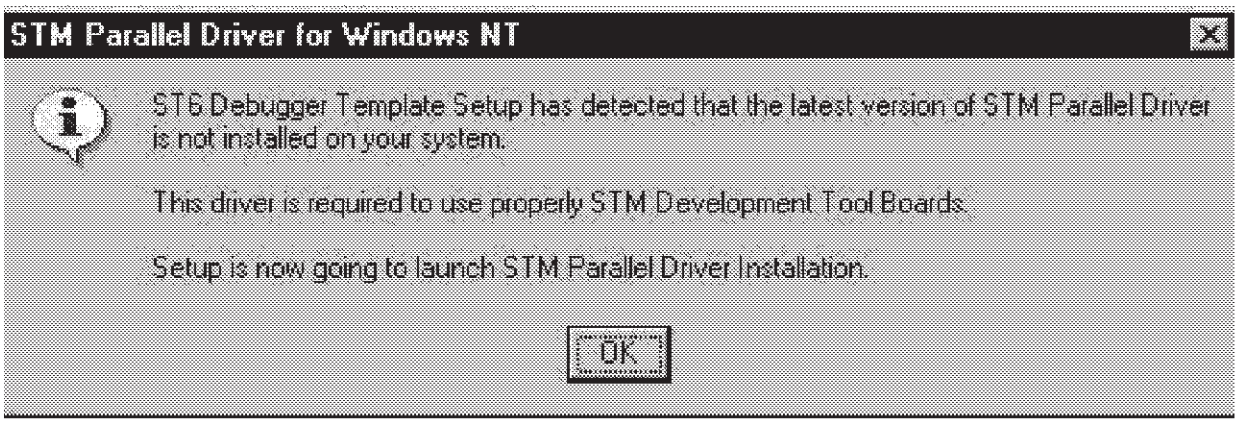
4 The install wizard will be launched. Follow the instructions that appear on the screen.

5 Select the program folder that the software will be installed to and click on the next button.

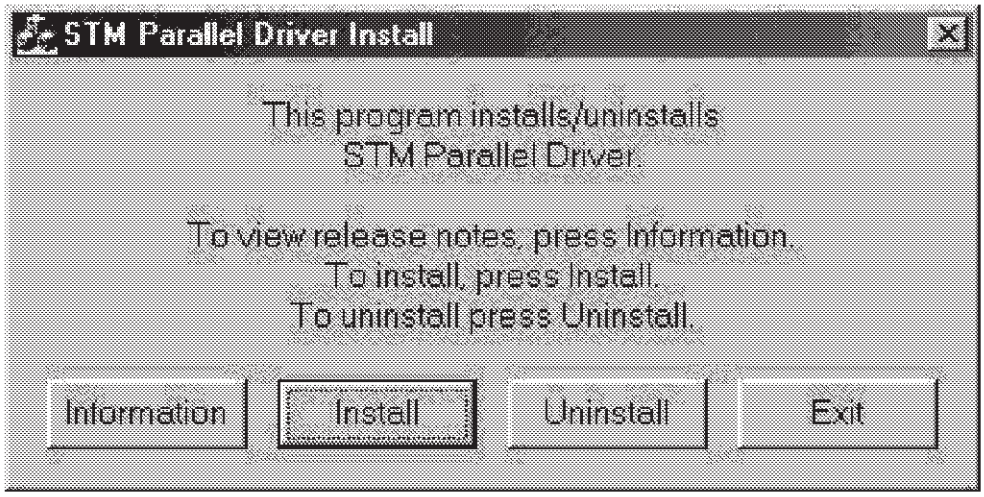
6 You will be prompted to choose the package you wish to install. To install the complete ST6 Toolchain select all the components. To install only the programmer interface select the Windows Epromer component.

7 Follow the instructions that appear on your screen. According to the selected components, you will be prompted to select the communication port you wish to connect the emulator and the starter kit to.

8 If you are installing the ST6 Toolchain on a Windows NT platform, you must install the Windows NT parallel port driver supplied on the CD-ROM. A window pops up if you have not already installed this driver (parstm.sys).



9 Click the OK button. Then the following window appears:



10 Click Install. The following window appears:

11 Click the OK button

The installation is now complete. You can choose to read the Release Notes, then click the Finish button.

To start the Windows Epromer, just click the corresponding icon in the cascading menus. For the first use, it is mandatory to configure the software through the menu Configure>Configure Epromer. You will have to select the Epromer Hardware (EPB, Gang Programmer, ...), the device to be programmed and the communication port to be used.

For more information on Windows Epromer utilisation refer to the Windows Epromer help file (Winee.hlp).

The latest version of the ST6 Toolchain can be downloaded from our web site: http://mcu.st.com.

To proceed to an update of the existing configuration, just click the Configure Software radio button, and follow the advices.

4.4 RUNNING WINDOWS EPROMER WITH GANG PROGRAMMERS

Windows Epromer is a programming tool working with gang programmers as well as EPROM programmer boards (EPBs) or starter kits. However, when used with a gang programmer, Windows Epromer is subject to some limitations due to the hardware specificities of the board.

4.4.1 Read Operations

- Read operations are only performed on the device placed on the MCU REFERENCE socket
- It is available on all space defined and selected

4.4.2 Verifying, Blank Checking, Programming Considerations

Verifying, blank checking and programming are performed on the slave devices.

The operation stops in either case:

- it was successfully completed for at least one slave device,

or:

- it failed for all the detected slave devices.

In case of failure for at least one slave device, a detailed report is produced.

Also, the programming process is automatically preceded by a blank checking operation. During all this blank checking phase, the Windows Epromer cannot manage the progress bar.

On the EPROM space, blank checking is performed on the EPROM and OPTION spaces.

Note: This operation may require an appreciable amount of time, and during this period, only the blinking of the BUSY LED will let you informed that some work is progressing.

4.5 MEMORY SPACE MANAGEMENT

4.5.1 MCU Memory Spaces

ST6xxxx microcontrollers have three types of memory spaces, all of them programmable. They are:

- The EPROM space, to program code and static data,
- The EEPROM space that can be used to save data supposed to be updated during program execution.
- The OSD space, to program the characters for OSD.

The Windows Epromer software manages ST6 EPROM, EEPROM and OSD spaces in a special way, considering each of the spaces a continuous space from address 0 up to an address depending on the specified microcontroller.

4.5.2 EPROM Space Mapping

The EPROM size differs from one ST6 microcontroller to the other, but there is a constant matching between the Windows Epromer linear memory mapping and the physical ST6 page address structure, as shown hereafter:

Table 1. EPROM Mapping

Epromer Address	Device Address	Device Page
0080h-07FFh	080h-07FFh	page 0
0800h 0F9Fh	0800h 0F9Fh	page 1 static
0FF0h-FF7h	0FF0h-0FF7h	page 1 static
0FFCh-FFFh	0FFCh-0FFFh	page 1 static
1010h-17FFh	0010h-07FFh	page 2
1810h-1FFFh	0010-07FFh	page 3

Note: Some areas in the microcontroller EPROM space cannot be used by your application programs. They are reserved for production test purposes. As a consequence, when producing program code, you must carefully specify the correct memory mapping for the devices you program. Fore more information refer to the data sheet for the considered device.

4.5.3 EEPROM Space Mapping

The EEPROM space of the controllers is generally organised in banks of 64 bytes. Application programs can only access one bank at a time. The Windows Epromer manages EEPROM spaces as continuous address ranges whose value depends on the ST6 microcontroller.

You will find hereafter EEPROM space mappings for some MCUs:

Table 2. EEPROM Mapping for ST62XXX MCUs

Epromer Address	EEPROM Bank	DRBR Register Value
0-3Fh	Page 0	01h
40h-7Fh	Page 1	02h

Table 3. EEPROM Mapping for ST63E73,T73 MCUs

Epromer Address	EEPROM Bank	DRBR Register Value
0-3Fh	Page 0	01h
40h-7Fh	Page 1	02h
80h-BFh	Page 2	03h
C0h-FFh	Page 3	81h
100h-13Fh	Page 4	82h
140h-17Fh	Page 5	83h
180h-1BFh	DDC page 0	10h
1C0h-1FFh	DDC page 1	20h

Table 4. EEPROM Mapping for ST63E78,T78,E85,T85,E87,T87,E88,T88,E89,T89 MCUs

Epromer Address	EEPROM Bank	DRBR Register Value
0-3Fh	Page 0	01h
40h-7Fh	Page 1	02h
80h-BFh	Page 2	03h
C0h-FFh	Page 3	81h
100h-13Fh	Page 4	82h
140h-17Fh	Page 5	83h

5 STAND-ALONE MODE

5.1 METHOD OF OPERATION

The gang programmer enables you to perform several distinct operations on your microcontrollers. Depending on your needs, you may blank check, verify, or program your microcontrollers. For this, when operating in stand-alone mode, proceed as follows:

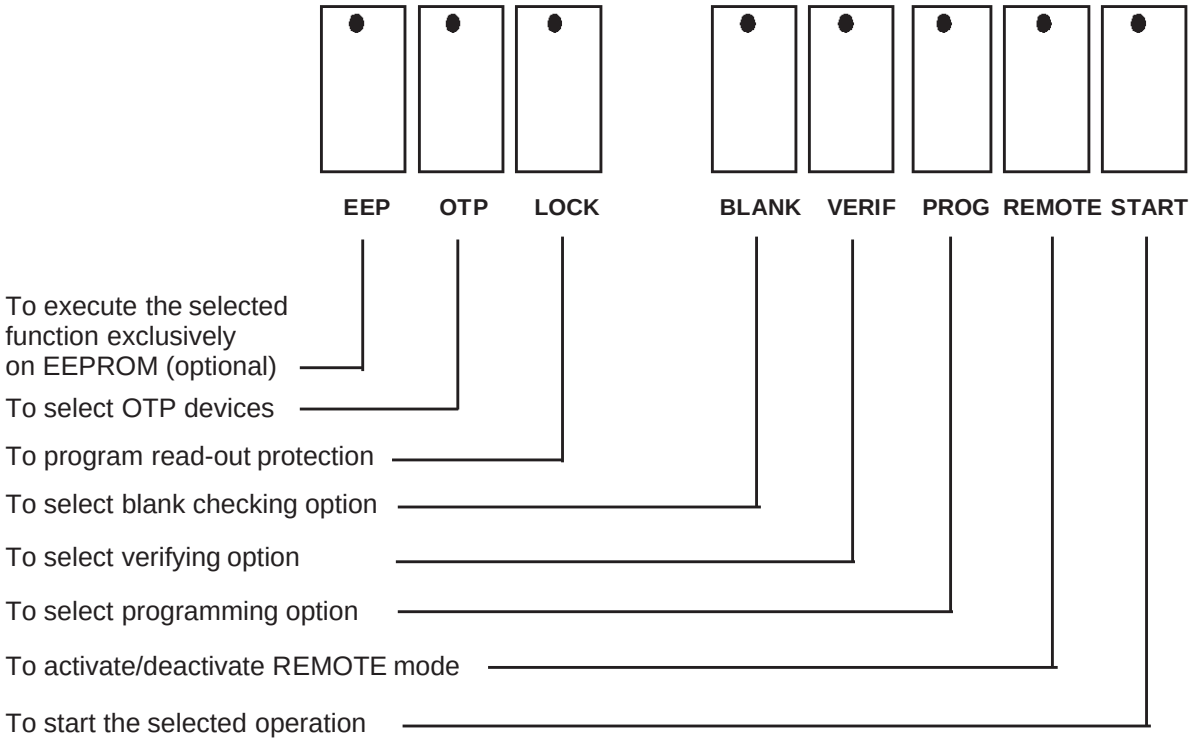
1 Select the stand-alone mode for your gang programmer.

2 Configure your gang programmer,

3 Proceed to the desired operation: blank checking, programming, or verifying.

Seven (or eight) keys in the gang programmer keypad will help you execute these functions:

GANG PROGRAMMER KEYPAD



5.2 SELECTING THE STAND-ALONE MODE

This mode is selected by default when powering on the gang programmer. You may also switch between REMOTE and STAND-ALONE modes by pressing the REMOTE key which operates thus as a toggle.

5.3 CONFIGURING THE GANG PROGRAMMER

Configuring the gang programmer consists of specifying various options that would fully describe the device programming environment. So, you will have to specify:

- The device types, if necessary, depending on the original factory settings of your gang,
- The device versions with respect to the programmed workspace,
- The reference component from which the programming is to be performed,

5.3.1 Device Type Selection

When shipped, the gang is configured with factory settings supporting the devices listed for each gang in the Appendix C. If for some reason, you want to change the type of the device to be programmed (or blank checked, or verified) by your gang, you will have to set some jumpers on the interface board. For this, proceed as follows:

1 Power off the gang programmer,

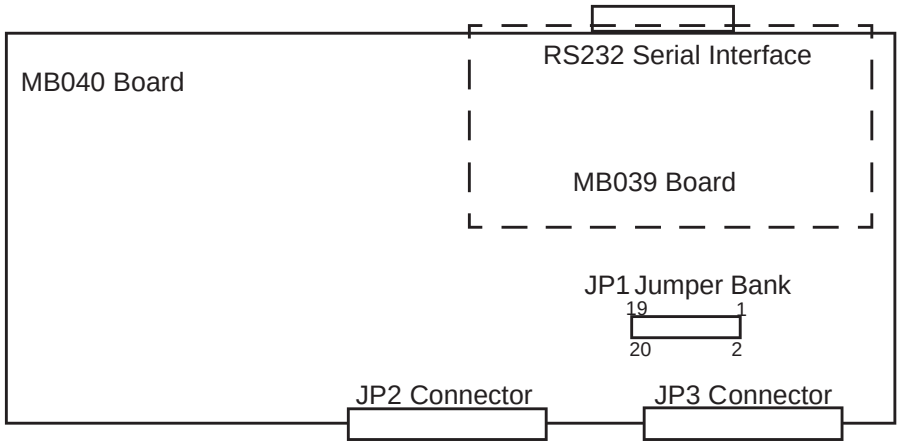
2 Remove the four side screw of the top module,

3 Open the gang programmer,

4 Remove the screws that hold the interface board through the sockets,

5 Unplug and extract the interface board,

6 Set the appropriate jumpers on the JP1 jumper bank.



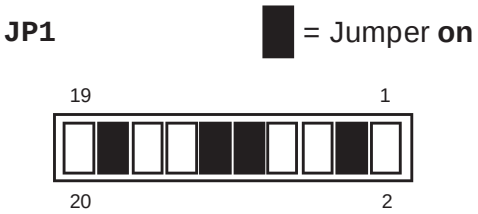
7 Plug the interface board into the sockets and screw it on again

8 Close the gang programmer and screw the top module on again

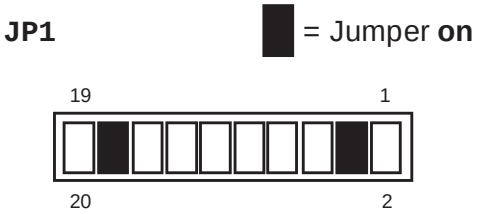
Setting the JP1 Jumpers

To program the following devices:

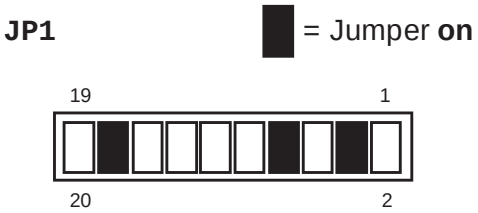
ST62T00
ST62T03
ST62T08
ST62T09



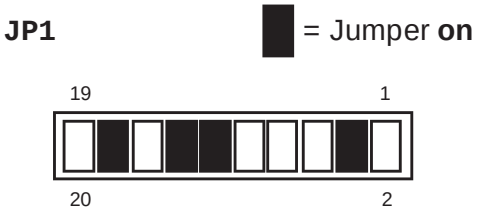
ST62E01/T01
ST62E10/T10
ST62E15/T15



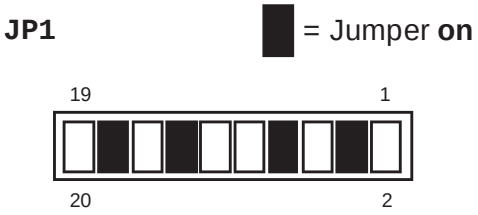
ST62E20/T20
ST62E25/T25



ST62E18T18
ST62E28/T28



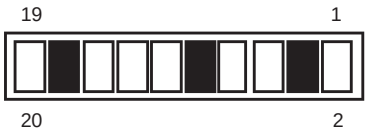
ST62E30/T30
ST62E32/T32



ST62E40/T40

ST62E80/T80

JP1 = Jumper **on**



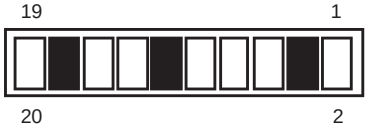
ST62E42/T42

ST62E85/T85

JP1 = Jumper **on**

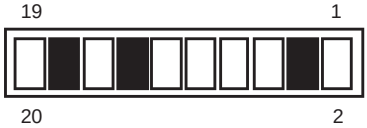
ST62E45/T45

JP1 = Jumper **on**



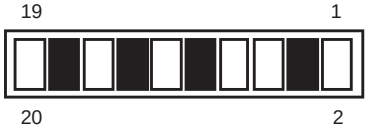
ST62T53

JP1 = Jumper **on**



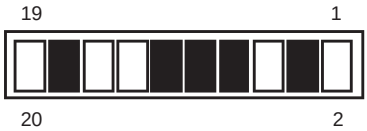
ST62T55

JP1 = Jumper **on**



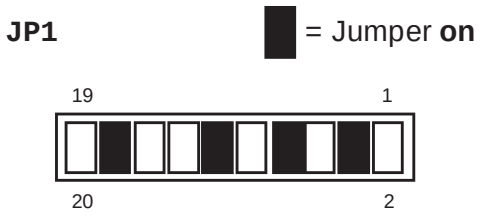
ST62T63

JP1 = Jumper **on**

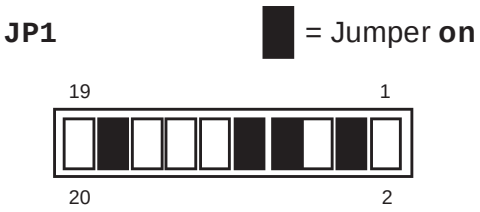


ST62E60/T60

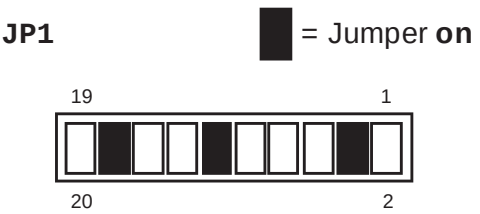
ST62E65/T65



ST63E73/T73

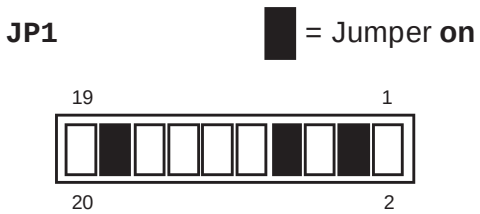


ST63E78/T78



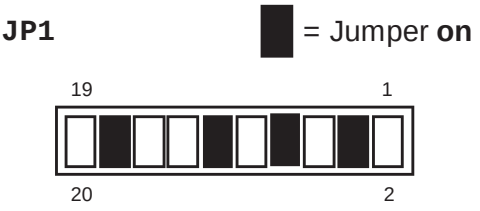
ST63E85/T85

ST63E87/T87



ST63E88/T88

ST63E89/T89



Note: Setting the jumper 3-4 causes a verifying operation to be automatically executed after the programming process. This jumper is set in the default configuration (factory setting).

5.3.2 Device Version Selection

Select the version of the device (OTP, EPROM, EEPROM):

To select **OTP**, press the OTP key on the gang programmer keypad (**EPROM** is selected by default). When OTP is selected (key light is on), to return to the EPROM selection, press the OTP key again.

To select **EEPROM**, press the EEP key (optional).

5.3.3 Reference Selection

The gang programmer is mainly designed for programming the ST6xxxx microcontrollers from a reference MCU. However, a reference REPROM device containing the code to be programmed can be used for programming slave devices.

This reference REPROM must contain all bytes related to EPROM, EEPROM, OPTION byte and OSD if these features are present inside the devices to be programmed. The reference REPROM code mapping should be as follows:

Table 5. Reference REPROM Code Mapping

code	Reprom addresses
MCU EPROM code	from address *0* to address *romsize-1*
MCU Option LSB byte	address *romsize*
MCU Option MSB byte	address *romsize + 1*
MCU E2prom code	from address *romsize+0x100* to address *romsize+0x100 +e2promsize* -1
MCU OSD code	from address *romsize+0x300* to address *romsize+0x300 +osdsize* - 1

where *romsize* is the size in bytes of the ROM and *e2promsize is* the size in bytes of the EEPROM of the device to be programmed.

Example 1:

To program an **ST62E65C** microcontroller with a reference REPROM, place into the reference REPROM:

- the EPROM code from address 0x0080 to 0x0FFF (4Kbytes of ROM)
- the LSB Byte of the Option Byte at address 0x1000
- the MSB Byte of the Option Byte at address 0x1001
- the EEprom code from address 0x1100 to 0x0117F (128 bytes)

Example 2:

To program an **ST63E88** microcontroller with a reference REPROM, place into the reference REPROM:

- the EPROM code from address 0x0080 to 0x4FFF (20Kbytes of ROM)
- the Option Byte at address 0x5000
- the EEprom code from address 0x5100 to 0x527F (384 bytes)
- the OSD code from address 0x5300 to 0x6DFF

Note: For the Option bytes, the reserved bits are required to be set or clear. So, the user has to check the state of this bits in the datasheet of the device to be programmed, and include them in the Options byte value.

When using a reference REPROM, the REPROM size must be specified by the means of the rotary switch located nearby the 32-pin DIL socket.

3 positions are available:

- 2764/27128 for 8K x8 and 16K x8 EPROMs
- 27256 for 32K x8 EPROMs
- 27512/271000 for 64K x 8 and 128K x8 EPROMs

The selection between the two types of reference is automatic. The gang programmer first checks whether a reference MCU exists and then checks whether a reference REPROM exists.

5.4 BLANK CHECKING YOUR MICROCONTROLLERS

The Blank Check is performed on the Eprom space, the Option bytes and the OSD space.

You may want to check whether the devices you're about to program have not some code on them. For this, proceed as follows:

1 Remove (if any) the reference EPROM from its socket,

2 Insert into the sockets the ST6xExx MCUs to be checked,

3 Select the device version (OTP/EPROM) via the OTP key,

4 Select the BLANK operation by pressing the BLANK key,

5 Validate the function by pressing the START key.

The blank checking process begins, while the red BUSY LED starts flashing.

The LED final status reflects the result of the operation:

When lit, a green socket LED reports a successful end of operation.

When lit, a red socket LED reports an error.

No LED on for a device means that the device is not detected.

Blinking LEDs indicate technology contention between reference and/or target chips.

Note: :You cannot blank check EEPROMs.

5.5 Programming your Microcontrollers

To program your microcontrollers, proceed as follows:

1 Make sure that you have inserted into the appropriate socket either the reference MCU or the reference EPROM; also, make sure that the jumpers are properly set in the JP1 jumper bank.

2 Insert into the slave sockets the ST6XEXX MCUs to be programmed,

3 If you want to activate the read protect option, press the LOCK key,

4 Select the PROG operation by pressing the PROG key,

5 Validate the operation by pressing the START key.

The programming process begins while the red BUSY LED starts flashing.

The LED final status reflects the result of the operation:

When lit, a green socket LED reports a successful end of operation.

When lit, a red socket LED reports an error.

No LED on for a device means that the device is not detected.

Blinking LEDs indicate technology contention between reference and/or target chips.

ALL LEDS on means that the reference device is defective, absent, or protected against reading.

Note: If the EEP key is pressed, only the EEPROM space will be programmed and verified. If not, every space (EPROM, Option byte(s), EEPROM and OSD) will be programmed and verified.

Setting the jumper 3-4 causes a verifying operation to be automatically executed after the programming process. This jumper is set in the default configuration (factory setting).

5.6 VERIFYING YOUR MICROCONTROLLERS

To verify your microcontrollers, proceed as follows:

1 Make sure that you have inserted into the appropriate socket either the reference MCU or the reference EPROM; also, make sure that the jumpers are properly set in the JP1 jumper bank.

2 Insert into the slave sockets the ST6XEXX MCUs to be verified,

3 Select the VERIF operation by pressing the VERIF key,

4 Validate the operation by pressing the START key.

The verifying process begins while the red BUSY LED starts flashing.

The LED final status reflects the result of the operation:

When lit, a green socket LED reports a successful end of operation.

When lit, a red socket LED reports an error.

No LED on for a device means that the device is not detected.

Blinking LEDs indicate technology contention between reference and/or target chips.

ALL LEDS on means that the reference device is defective, absent, or protected against reading.

Note: If the EEP key is pressed, only the EEPROM space will be verified. If not, every space (EPROM, Option byte(s), EEPROM and OSD) will be verified.

Setting the jumper 3-4 causes a verifying operation to be automatically executed after the programming process. This jumper is set in the default configuration (factory setting).

APPENDIX A

ABBREVIATIONS

EPROM: Erasable Programmable Read-Only Memory (can be programming then erased using UV light).

EEPROM: Electrically Erasable Programmable Read-Only Memory

OSD: On Screen Display

MCU: MicroController Unit

OTP: One-Time Programmable memory (once programmed, cannot be erased).

APPENDIX B

PROGRAMMING SUPPLIES VOLTAGE CHECKING

VCC Supply Voltage:4.80 to 5.20 Volts

VPP Supply Voltage:13 +/- 0.25 Volts

These voltages do not require any calibration but can be checked using the VCHECK6.EXE file. To perform the Programming Voltage Checking proceed as follows:

- Power off the PC and the gang programmer.
- Connect the COM1 or COM2 serial interface port of the PC to the RS232 connector located on the rear panel of the gang programmer.
- Remove all devices from the sockets.
- Power on the PC and the gang programmer.
- Select the REMOTE mode by pressing the REMOTE key. The REMOTE key led lights up.
- Double click on VCHECK6.EXE to run the VCHECK6 program. This program is by default located in the directory **c:\program files\stm\st6toolchain\epromer**.
- Using a multimeter (cal. 20 V), check the VCC and VPP voltages referenced to GND on the dedicated pins of each socket; values depends on the MCU (see table below).
- If a voltage is out of range, it is advisable to send the gang programmer for repairing. Any maintenance operation should be performed by qualified STMicroelectronics service personnel only.

Table 6. 67 [[[[6XSSOLHV 3LQ $VVLJQPHQW

MCU	Package	VCC pin	VPP pin	GND pin
ST62T08 ST62T09 ST62E10/T10 ST62E18/T18 ST62E20/T20	DIP20 SO20	1	6	20
ST62E15/T15 ST62E28/T28 ST62E25/T25	DIP28 SO28	1	10	28
ST62E40/T40	PQFP80	23	18	24
ST62E42/T42	PQFP64	17	16	18
ST62E45/T45	PQFP52	12	11	13
ST62T53 ST62T63 ST62E60/T60 ST62E93/T93	DIP20 SO20	9	3	10

MCU	Package	VCC pin	VPP pin	GND pin
ST62E65/T65 ST62E55/T55 ST62E94/T94	DIP28 SO28	11	3	12
ST62E80/T80	PQFP100	40	27	41
ST62E85/T85	PQFP80	30	21	31
ST63E73/T73	SDIP42	42	29	21
ST63E78/T78 ST63E85/T85 ST63E87/T87 ST63E88/T88 ST63E89/T89	SDIP42	42	30	21

APPENDIX C

CURRENT ST6 GANG PROGRAMMERS

Table 7. ST62E0X Gang Programmer (GP/DIP,GP/SO)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62T00/T00C/T03/T03C	DIP16/SO16	1K	-	X
ST62E01/E01C/T01/T01C	DIP16/SO16	2K	-	

Table 8. ST62E10 Gang Programmer (GP/DIP,GP/SO)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62T08/T08C/T09/T09C	DIP20/SO20	1K	-	
ST62E10/E10C/T10/T10C	DIP20/SO20	2K	-	X
ST62E18C/T18C	DIP20/SO20	8K	-	
ST62E20/E20C/T20/T20C	DIP20/SO20	4K	-	

Table 9. ST62E15 Gang Programmer (GP/DIP,GP/SO)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E15/E15C/T15/T15C	DIP28/SO28	2K	-	X
ST62E25/E25C/T25/T25C	DIP28/SO28	4K	-	
ST62E28C/T28C	DIP28/SO28	8K	-	

Table 10. ST62E30 Gang Programmer (GP/DIP,GP/SO)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E30B/T30B	DIP28/SO28	8K	128	X

Table 11. ST62E32 Gang Programmer (GP/DIP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E32B/T32B	SDIP42	8K	128	X

Table 12. ST62E40 Gang Programmer (GP/QFP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E40/E40B/T40/T40B	QFP64	8K	128	X

Table 13. ST62E42 Gang Programmer (GP/QFP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E42/E42B/T42/T42B	QFP64	8K	-	X

Table 14. ST62E45 Gang Programmer (GP/QFP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E45/E45B/T45/T45B	QFP52	4K	128	X

Table 15. ST62E60 Gang Programmer (GP/DIP,GP/SO)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62T53B ST62T53C	DIP20/SO20	2K	-	
ST62E60B/T60B ST62E60C/T60C	DIP20/SO20	4K	128	X
ST62T63B ST62T63C	DIP20/SO20	2K	64	

Table 16. ST62E65 Gang Programmer (GP/DIP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E55B/T55B ST62E55C/T52C	DIP28/SO28	4K	128	
ST62E65B/T65B ST62E65C/T65C	DIP28/SO28	4K	128	X

Table 17. ST62E80 Gang Programmer (GP/QFP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E80/T80/E81/T81	QFP100	8K	128	X

Table 18. ST62E85 Gang Programmer (GP/QFP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST62E85/T85	QFP80	8K	-	X

Table 19. ST63E73 Gang Programmer (GP/DIP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST63E73/T73	SDIP42	16K	512	X

Table 20. ST63E87 Gang Programmer (GP/DIP)

Supported devices	Package	EPROM	EEPROM	Factory Configuration
ST63E85/T85/E87/T87	SDIP42	20K	384	X
ST63E78/T78	SDIP42	16K	384	
ST63E88/T88/E89/T89	SDIP42	20K	384	

Index

Index

Notes: