ST6 FAMILY

EPROM PROGRAMMER

USER MANUAL

Release 2.0

AUGUST 2000

EPROM Programmer for the ST6 Family

FOREWORD

The ST6xxxx EPROM programmer is a tool designed for programming EPROM or OTP micro-controllers of the ST6 family.

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

In **Remote mode**, the EPROM programmer is connected to a DOS-compatible PC through an RS232 serial interface or through a parallel interface. Object code in Intel .hex format is read from disk files to program the device. The menu-driven software provides access to PRO-GRAM, 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

1.1.1 EPROM Programmer Types

EPROM programmers equipped with a *parallel interface* can be used only in *remote mode*. This class include the following EPBs:

- ST62E3X

- ST62E8X

- ST63E73

EPROM programmers equipped with a *serial interface* can be used in both *remote* and *stand-alone modes*.

This class include the following EPBs:

- ST62E2XC

- ST62E4X

- ST62E6X

- ST63E1XX

In both modes, the customer can protect the devices against further reading.

1.2 EQUIPMENT

The supplied equipment comprises:

1.2.1 Serial EPROM PROGRAMMER

- 1 ST6 remote programmer

- 1 25-pin serial communication cable

- 1 adapter 9/25 for PC/AT

- 1 blank EPROM

- 1 "MCU ON CD" CD-ROM

1.2.2 Parallel EPROM PROGRAMMER

- 1 ST6 remote programmer

- 1 parallel communication cable

- 1 "Plug in" power supply pack

- 1 "MCU ON CD" CD-ROM

1.3 POWER SUPPLY

The board can be supplied from the integrated power supply provided with the board, or from an external +15V DC / 0.5A power supply.

When using an external power supply, take care of the polarities marked nearby the two-point connector.

1.4 DEVICE INSTALLATION

Place the device into the relevant zero insertion force socket with the erasure window on the top and pin 1 matching the mark on the board.

CAUTION: Placing the 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 semi-conductor 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 ST62E2XC & ST62E6X 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 EPB 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 EPROM programmer is connected to a PC through the appropriate inter-face, serial or parallel, depending on the EPROM programmer type. The programming process is menu-driven.

4.2 HARDWARE INSTALLATION

4.2.1 EPROM Programmer Board with Serial Interface

Proceed as follows:

1 Power off the PC computer and the EPROM programmer,

2 Configure the board in remote mode by placing a jumper at the REMOTE position in the W1 jumper bank, depending on the type of EPB:
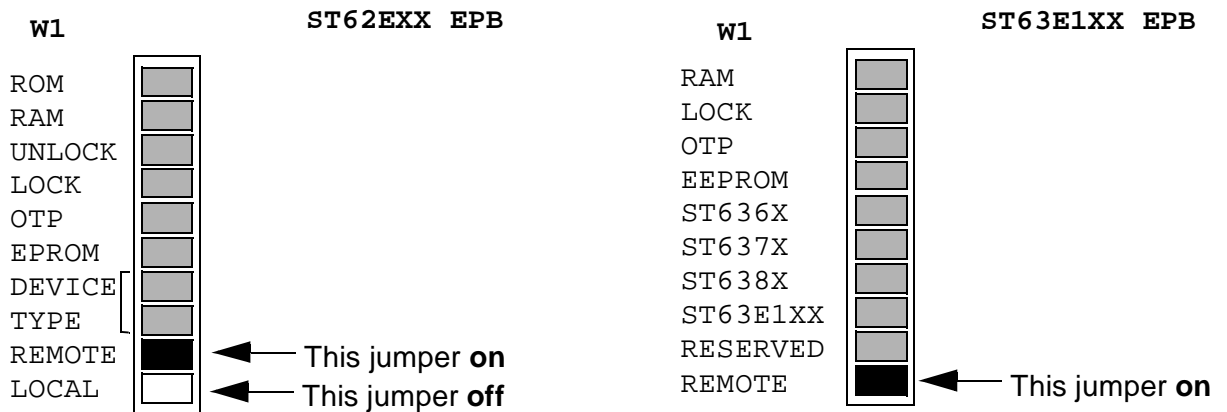
Note: EPBs are shipped with the REMOTE jumper **on**.

3 Connect the serial interface PC port COM1 or COM2 to the J3 board connector using the serial interface cable,

4 Power on the PC and the EPROM programmer board,

5 The REMOTE LED (LD1) and the BUSY LED (LD2) light up.

4.2.2 EPROM Programmer Board with Parallel Interface

Proceed as follows:

1 Power off the PC computer and the EPROM programmer,

2 Using the parallel interface cable, connect the parallel port of the PC (LPT1 or LPT2) to the P1 board connector,

CAUTION: Connecting the P1 board connector to a serial port of the PC could damage the board.

3 Power on the PC and the EPROM programmer board,

4 The POWER LED (LD1) lights up, showing that the board is supplied.

Notes: The supplied interface cable has been tested in order to operate properly on most PCs. Do not use any other cable, especially if it is longer than the one provided by STMicro-electronics: the board may not operate properly.

The cable should be connected directly to the DB-25 female connector of the PC parallel port. This connector is similar to the one installed on the board. Do not insert any additional cables or switchboxes between the PC and the board: a malfunctioning of the board may result.

If a dongle is mounted on the PC parallel port, it should not interfere with the programming board. Should you notice a dysfunctioning of the board, remove the dongle and restart the installation procedure.

4.3 SOFTWARE INSTALLATION

The EPB 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.

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

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.

4.4 MEMORY SPACE MANAGEMENT

4.4.1 MCU Memory Spaces

ST6xxxx microcontrollers have two types of memory spaces, both programmable. They are:

• The EPROM space, for program code and static data,

• The EEPROM space that can be used to save data supposed to be updated during program execution.

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

4.4.2 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
0-7FFh	0-7FFh	page 0
800h-FFFh	800-FFFh	page 1 static
1000h-17FFh	0-7FFh	page 2
1800h-1FFFh	0-7FFh	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.

4.4.3 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,E71,T71 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 ST63E85,T85,E87,T87,E69,T69 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

4.5 IN-CIRCUIT PROGRAMMING

4.5.1 Direct MCU Programming

Parallel EPBs provides you also with an IN-CIRCUIT programming capability. This enables you to program microcontrollers directly in their application boards. A special connector (J3) is reserved for this usage.

4.5.2 J3 In-Circuit Connector Pin Assignment

An in-circuit EPROM programming is possible, if the relevant pins of the J3 connector (8x2 HE10) located on the EPB board are connected to the programming pins of the MCU on user's application board.

For these connections, a 16-pin connector must be installed on the application board

Only a few signals of the 16-pins cable are used. They are listed below including their connection to the MCU and their I/O characteristics (seen from the programming tool point of view).

Table 5. J3 Signal Characteristics for MCU Connection

J3 pin	I/O	ST62E3X	ST62E8X	ST63E73
pin 1	O	PB5	PC6	PC5
pin 3	O	not used	PC5	not used
pin 5	O	OSCIN	OSCIN	OSCIN
pin 7	I	PB6	PC7	PC2
pin 9	O	RESET/	RESET/	RESET/
pin 11	I	not used	not used	not used
pin 13	O	VPP/TM	VPP/TM	TEST
pin 14	I	PB4	VDD	VDD
pin 16	I	PD6	VSS	VSS
pin 14,16	power	VDD		
pin 2,4,8	power	VSS		

The following conditions must be fulfilled:

- Pin 1 must not be connected to any other output. A pull-up resistor of 2K min. and a pull-down resistor of 10 k min. are allowed.

- Pin5 must not be connected to a clock generator output. A quartz or a ceramic resonator are allowed.

- Pin 7 must not be connected to any other output.   A pull-up resistor of 2K min. and a pull-down resistor of 10 k min. are allowed.

- Pin 9 must be opened or pulled up by a resistor of 2k min. The capacities load should not exceed 1μF.

- Pin 13 VPP/TM (RESET) on application must not be connected directly to GND, but should be pulled down by a resistor of 10K min. It is mandatory to add a ceramic capacitor of 100nF between VPP/TM and VSS.

- EPROMer pin for VSS must be connected to the MCU VSS pins.

• Epromer pin VDD connection is optional (and not recommended). If the chip is supplied by the application power, the voltage must be 5V to avoid excessive current through the CMOS input protection diodes. If the chip is supplied by the EPROMer, the total load current should not exceeds 100mA and the capacitive load must be lower than 50 pF.

4.5.3 In-Circuit Programming Procedure

Proceed as follows:

1 Connect the EPB to the PC parallel port,

2 Verify that no devices are plugged into the sockets,

3 Connect the EPB to the application board under power off condition,

4 Power on the application board (+5V),

5 Click on the **winee icon** to run the Windows Epromer program,

6 Follow the instructions on your screen.

5 STAND-ALONE MODE

5.1 METHOD OF OPERATION

Only EPROM programmers equipped with a serial interface can operate in stand-alone mode. They are:

ST62E2XC,

ST62E4X,

ST62E6X,

and **ST63E1XX** EPBs.

EPBs enable you to perform several distinct operations on your microcontroller. Depending on your needs, you may blank check, verify, or program your microcontroller. For this, when operating in stand-alone mode, proceed as follows:

1 Select the stand-alone mode for your EPB,

2 Configure your EPB,

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

5.2 SELECTING THE STAND-ALONE MODE

To select the stand-alone mode, place (or remove) a jumper at the appropriate location in the jumper bank of your EPB.

Note: EPBs are shipped with the REMOTE jumper **on**.

The operation to be performed depends on the type of your EPB:

For more information on EPB general layout, see "Appendix B" on page 34.
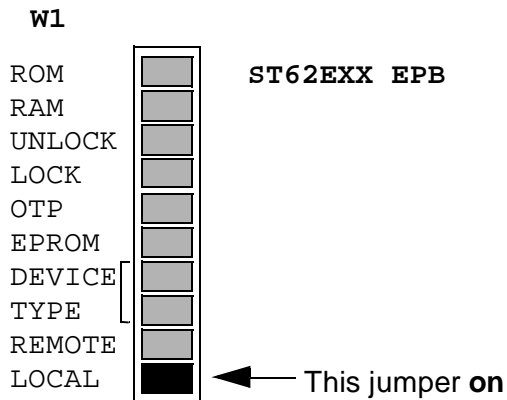
For ST62EXX EPBs, set the LOCAL jumper in the W1 jumper bank.

```
W1
ROM
RAM
UNLOCK
LOCK
OTP
EPROM
DEVICE
TYPE
REMOTE
LOCAL        ◄──── This jumper on
```

ST62EXX EPB

```
        W1
RAM
LOCK                   ST63E1XX EPB
OTP
EEPROM
ST636X
ST637X
ST638X
ST63E1XX
RESERVED
REMOTE        ◄────────  This jumper off
```

For ST63E1XX EPBs, remove the REMOTE jumper in the W1 jumper bank.
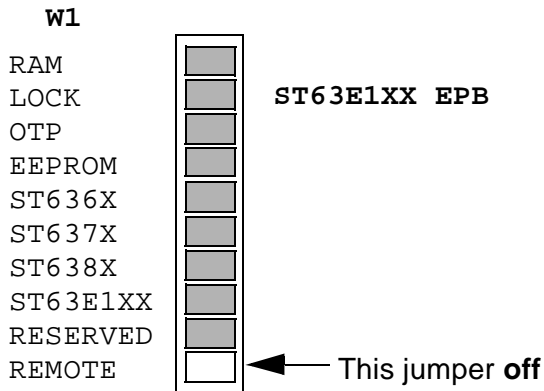
5.3 CONFIGURING THE EPB

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

1 The device type,

2 The version of the device with respect to the programmed workspace,

3 The reference component from which the programming is to be performed,

5.3.1 Device Type Selection

In the W1 jumper bank select the device type to be blank checked, programmed, or verified, by setting the appropriate jumper.

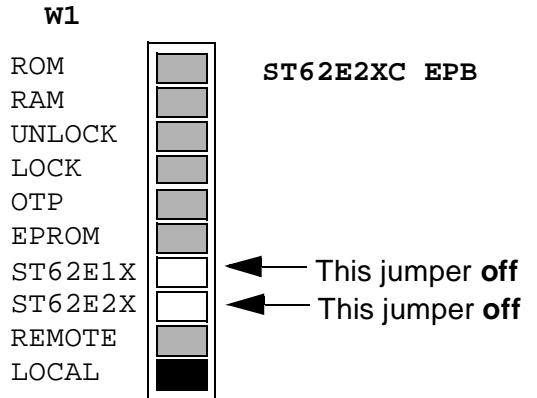**Note:** EPBs are shipped with both DEVICE TYPE jumpers on.

For more information on the devices supported by the various EPBs, see "Appendix C" on page 36
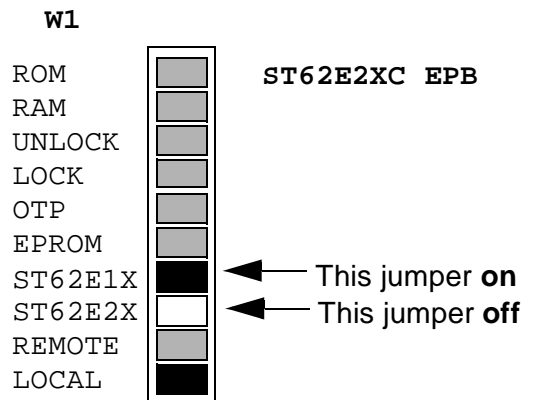
ST62EXX EPB FAMILY

ST62E2XC EPB

To program the following devices:

ST62T00/T00C

ST62T03/T03C

ST62T08/T08C

ST62T09/T09C

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	
EPROM	
ST62E1X	← This jumper **off**
ST62E2X	← This jumper **off**
REMOTE	
LOCAL	

ST62E2XC EPB

To program the following devices:

ST62E01/E01C(*)

ST62T01/T01C

ST62E10/E10C

ST62T10/T10C

ST62E15/E15C

ST62T15/T15C

(*): For this device type, a jumper must also be set on the OTP position

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	
EPROM	
ST62E1X	← This jumper **on**
ST62E2X	← This jumper **off**
REMOTE	
LOCAL	

ST62E2XC EPB

To program the following devices:

ST62E20/E20C

ST62T20/T20C

ST62E25/E25C

ST62T25/T25C

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	
EPROM	
ST62E1X	← This jumper **off**
ST62E2X	← This jumper **on**
REMOTE	
LOCAL	
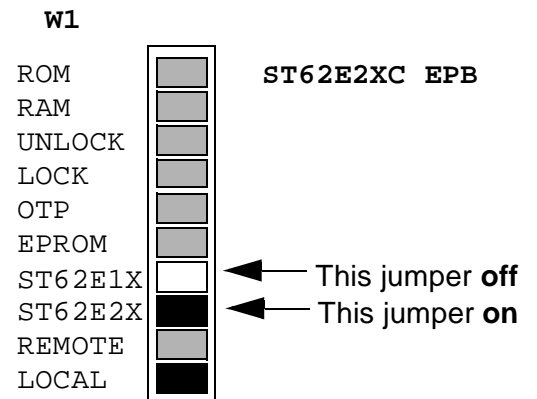
ST62E2XC EPB

ST62E2XC EPB (cont'd)

To program the following devices:

ST62E29C/T29C

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	← This jumper **off**
EPROM	← This jumper **off**
ST62E1X	← This jumper **on**
ST62E2X	← This jumper **on**
REMOTE	
LOCAL	

ST62E2XC EPB
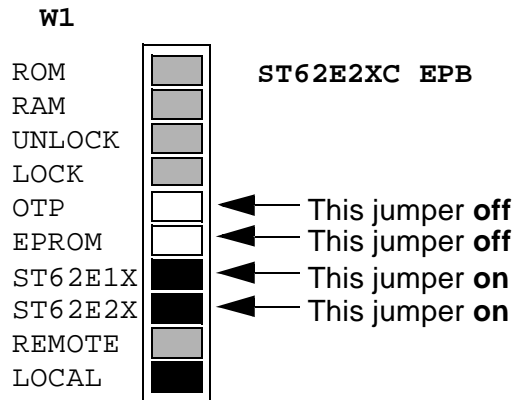
To program the following devices:

ST62E05C/T05C

ST62E18C/T18C

ST62E28C/T28C

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	← This jumper **on**
EPROM	
ST62E1X	← This jumper **on**
ST62E2X	← This jumper **on**
REMOTE	
LOCAL	

ST62E2XC EPB
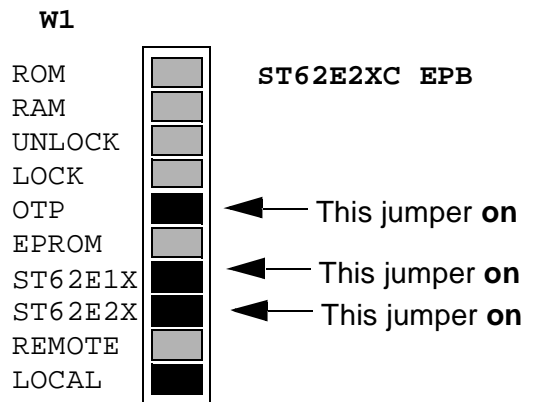
ST62E4X EPB

To program the following devices:

ST62E40/E40B

ST62T40/T40B

ST62E42/E42B

ST62T42/T42B

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	
EPROM	
4KBYTES	← This jumper **off**
8KBYTES	← This jumper **on**
REMOTE	
LOCAL	
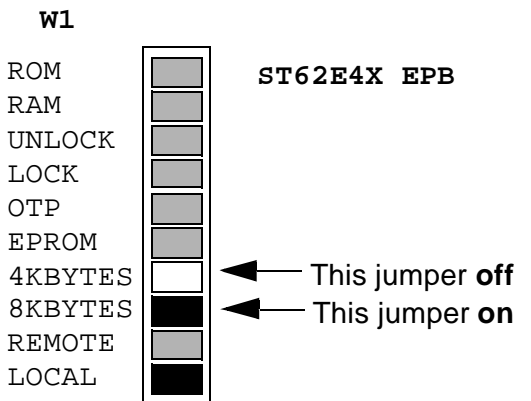
ST62E4X EPB

ST62E4X EPB (cont'd)

To program the following devices:

ST62E45/E45B

ST62T45/T45B

ST62T46/T46B

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	
EPROM	
4KBYTES	◄——— This jumper **on**
8KBYTES	◄——— This jumper **off**
REMOTE	
LOCAL	

ST62E4X EPB

ST62E6X EPB

To program the following devices:

ST62T52B/T52C

ST62T53B/T53C

ST62T62B/T62C

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	
EPROM	
R1	◄——— This jumper **off**
R2	◄——— This jumper **off**
REMOTE	
LOCAL	
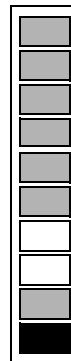
ST62E6X EPB

To program the following device:

ST62T55B/T55C

W1

ROM	
RAM	
UNLOCK	
LOCK	
OTP	
EPROM	
R1	◄——— This jumper **off**
R2	◄——— This jumper **on**
REMOTE	
LOCAL	
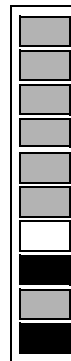
ST62E6X EPB

ST62E6X EPB (cont'd)

W1

To program the following device:

ST62T63B/T63C

```
ROM
RAM
UNLOCK
LOCK
OTP
EPROM
R1        ◄──── This jumper on
R2        ◄──── This jumper off
REMOTE
LOCAL
```

ST62E6X EPB

W1
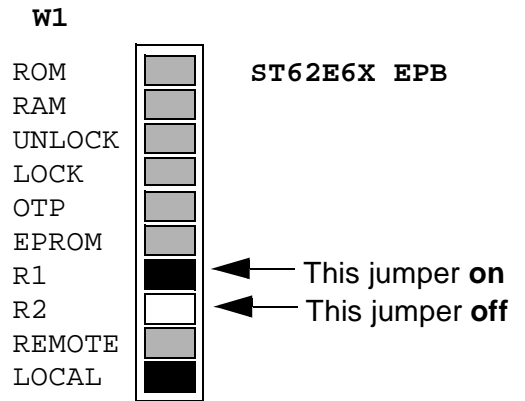
To program the following devices:

ST62E60B/T60B/E60C/T60C

ST62E65B/T65B/E65C/T65C

```
ROM
RAM
UNLOCK
LOCK
OTP
EPROM
R1        ◄──── This jumper on
R2        ◄──── This jumper on
REMOTE
LOCAL
```

ST62E6X EPB
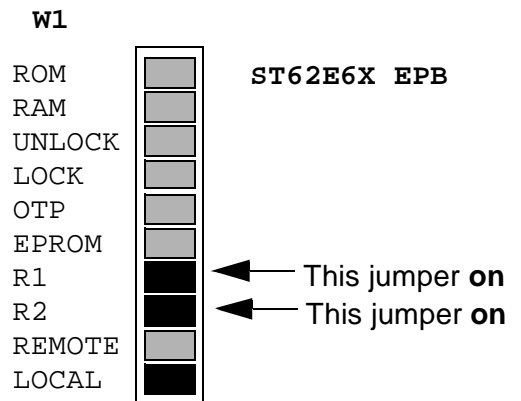
ST63EXXX EPB FAMILY

ST63E1XX EPB

To program the following devices:

ST63E69

ST63T69

```
              W1
         ┌────────┐
RAM      │ ▓▓▓▓▓▓ │
LOCK     │ ▓▓▓▓▓▓ │   ST63E1XX EPB
OTP      │ ▓▓▓▓▓▓ │
EEPROM   │ ▓▓▓▓▓▓ │
ST636X   │ ██████ │ ◄─────  This jumper on
ST637X   │        │
ST638X   │        │
ST63E1XX │        │
RESERVED │ ▓▓▓▓▓▓ │
REMOTE   │        │
         └────────┘
```
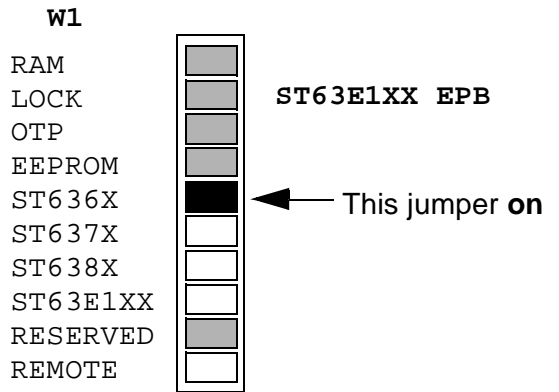
To program the following devices:

ST63E76

ST63T76

ST63E78

ST63T78

```
              W1
         ┌────────┐
RAM      │ ▓▓▓▓▓▓ │
LOCK     │ ▓▓▓▓▓▓ │   ST63E1XX EPB
OTP      │ ▓▓▓▓▓▓ │
EEPROM   │ ▓▓▓▓▓▓ │
ST636X   │        │
ST637X   │ ██████ │ ◄─────  This jumper on
ST638X   │        │
ST63E1XX │ ██████ │ ◄─────  This jumper on
RESERVED │ ▓▓▓▓▓▓ │
REMOTE   │        │
         └────────┘
```
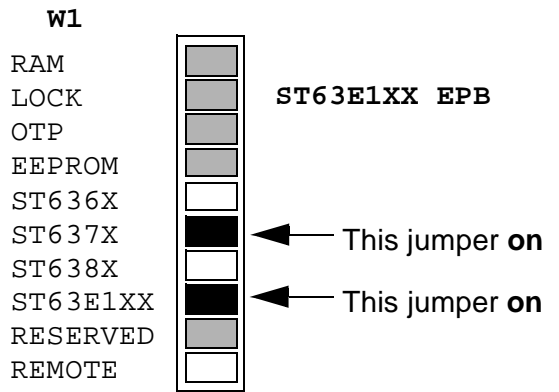
To program the following devices:

ST63E77

ST63T77

```
              W1
         ┌────────┐
RAM      │ ▓▓▓▓▓▓ │
LOCK     │ ▓▓▓▓▓▓ │   ST63E1XX EPB
OTP      │ ▓▓▓▓▓▓ │
EEPROM   │ ▓▓▓▓▓▓ │
ST636X   │        │
ST637X   │ ██████ │ ◄─────  This jumper on
ST638X   │        │
ST63E1XX │        │
RESERVED │ ▓▓▓▓▓▓ │
REMOTE   │        │
         └────────┘
```
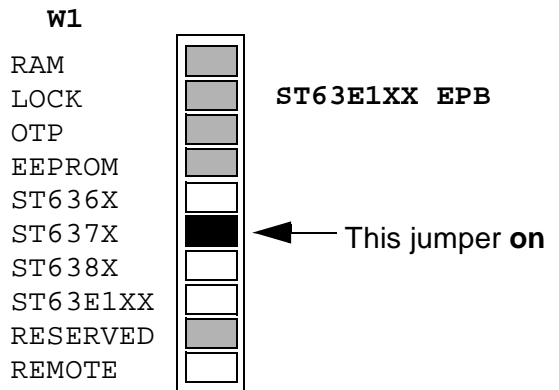
ST63E1XX EPB (cont'd)

To program the following devices:

ST63E85

ST63T85

ST63E87

ST63T87

```
                    W1
RAM
LOCK                         ST63E1XX EPB
OTP
EEPROM
ST636X
ST637X
ST638X            ◀──── This jumper on
ST63E1XX
RESERVED
REMOTE
```
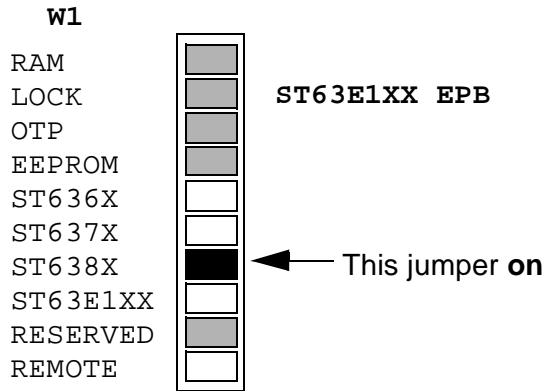
To program the following devices:

ST63E156

ST63T156

```
                    W1
RAM
LOCK                         ST63E1XX EPB
OTP
EEPROM
ST636X
ST637X
ST638X
ST63E1XX          ◀──── This jumper on
RESERVED
REMOTE
```
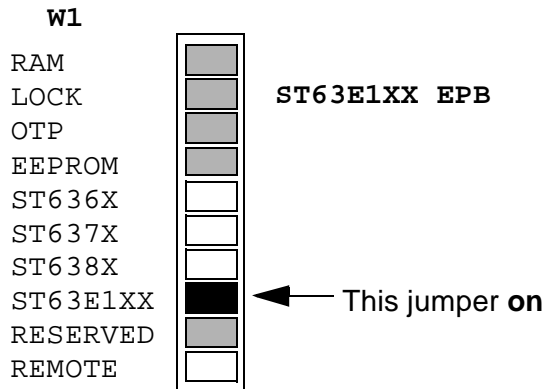
5.3.2 Device Version Selection

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

Note

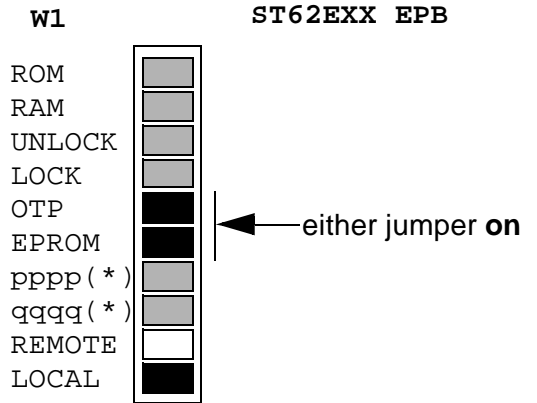EPBs are shipped with the EPROM jumper **on**.

ST62EXX EPB:

Set the appropriate jumper:

OTP version

or

EPROM version

(*) pppp is ST62E1X, 4KBYTES, or R1,

and qqqq is ST62E2X, 8KBYTES, or R2,

depending on the type of ST62EXX EPB.

```
    W1              ST62EXX EPB
 ROM        [   ]
 RAM        [   ]
 UNLOCK     [   ]
 LOCK       [   ]
 OTP        [███]
 EPROM      [███]  ◄─── either jumper on
 pppp(*)    [   ]
 qqqq(*)    [   ]
 REMOTE     [   ]
 LOCAL      [███]
```
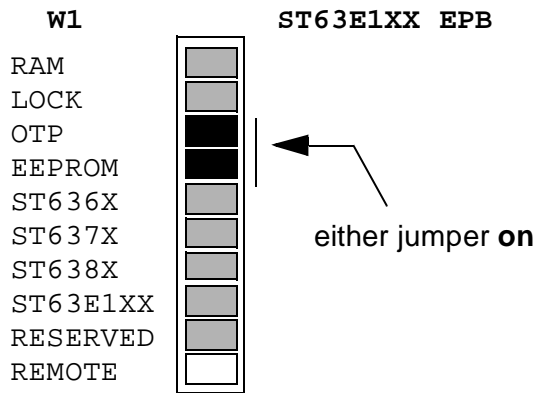
ST63E1XX EPB:

OTP or EPROM version

or

EEPROM version

When the OTP jumper is set, the device is assumed to be an OTP device. Otherwise it is considered an EPROM device. When the EEPROM jumper is set, the EEPROM space is exclusively selected.

```
    W1              ST63E1XX EPB
 RAM       [   ]
 LOCK      [   ]
 OTP       [███]
 EEPROM    [███]  ◄───
 ST636X    [   ]         either jumper on
 ST637X    [   ]
 ST638X    [   ]
 ST63E1XX  [   ]
 RESERVED  [   ]
 REMOTE    [   ]
```

5.3.3 Reference Selection

In stand-alone mode, microcontrollers are programmed via a simple key operation directly from a master microcontroller (**Reference MCU**) or from a master EPROM device (**Reference REPROM**).
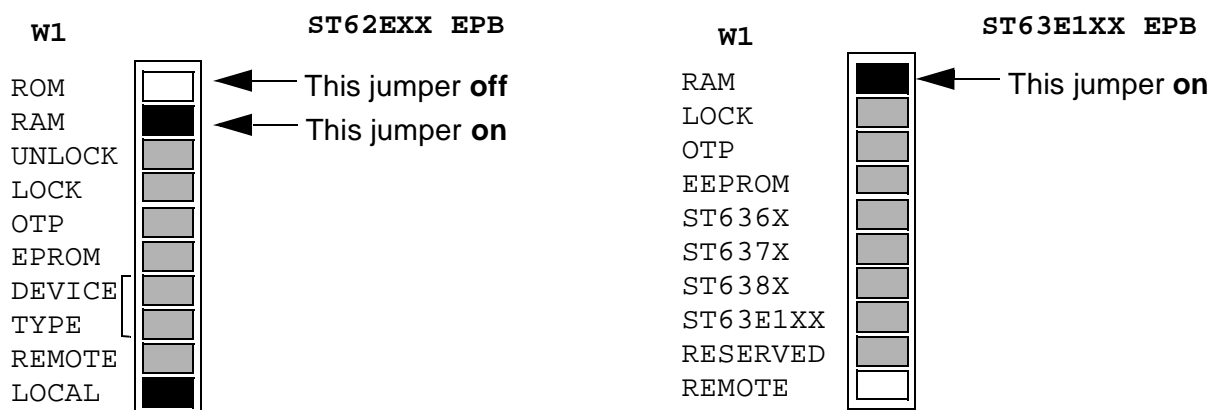
5.3.3.1 Using a Reference MCU

Proceed as follows:

1 Set a jumper on the **RAM** position in the W1 jumper bank:

For more information on EPB general layout, see "Appendix B" on page 34.

Note: EPBs are shipped with the RAM jumper on.

```
    W1           ST62EXX EPB
ROM       [ ]  ◄──  This jumper off
RAM       [■]  ◄──  This jumper on
UNLOCK    [ ]
LOCK      [ ]
OTP       [ ]
EPROM     [ ]
DEVICE    [ ]
TYPE      [ ]
REMOTE    [ ]
LOCAL     [■]
```

```
    W1           ST63E1XX EPB
RAM       [■]  ◄──  This jumper on
LOCK      [ ]
OTP       [ ]
EEPROM    [ ]
ST636X    [ ]
ST637X    [ ]
ST638X    [ ]
ST63E1XX  [ ]
RESERVED  [ ]
REMOTE    [ ]
```

2 Insert the reference MCU containing the program into the appropriate socket:

3 Check that the jumpers on W1 match the device specifications.

4 Set the MASTER/TARGET switch on the MASTER position:

5 Validate your choice by pressing the START switch.

The reference MCU is being read and its contents are stored into the board memory. When present, the EPROM, EEPROM, and OPTION byte are stored in the meantime.

Note that while the reading operation progresses, the entire EPB memory contents are automatically verified against the reference MCU contents.

The reading process begins while the red BUSY LED (LD2) starts flashing.

The LED final status reflects the result of the operation:

When lit, the green OK LED (LD4) reports a successful end of operation.

When lit, the red ERROR LED (LD3) reports an error.

ALL LEDS on means that the reference device is protected against reading.

Once the reference MCU is read and until a new read operation is performed or a power down occurs, the memory contents remain available for programming or verifying as many devices as required. See "Programming your Microcontroller" on page 30, and "Verifying your Micro-controller" on page 32.

5.3.3.2 Using a Reference EPROM

The EPB 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, and OPTION byte if these features are present inside the devices to be programmed. The reference RE-PROM code mapping should be as follows:

Table 6. Reference REPROM Code Mapping

code	addresses
MCU EPROM code	from address *0* to address *romsize-1*
MCU Option byte:	address *romsize*
MCU E2prom code:	from address *romsize+0x100* to address *romsize+0x100 +e2promsize* -1

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

Example 1:

To program an **ST62E25C** microcontroller with a reference REPROM, place in the reference EPROM:

- the EPROM code from address 0x0800 to 0x0FFF (2 Kbytes of ROM)
- the Option Byte at address 0x1000
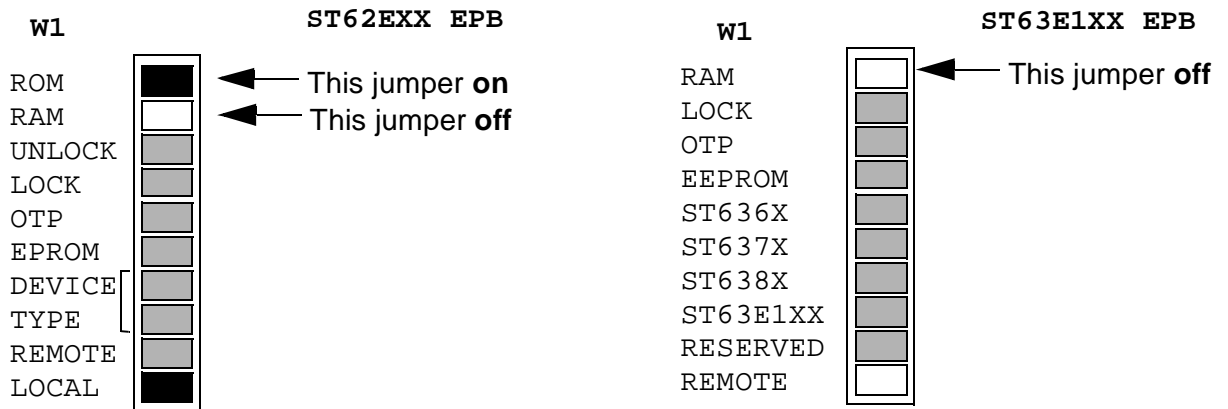
Example 2:

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

- the EPROM code from address 0x0000 to 0x0FFF (4 Kbytes of ROM)
- the LSB byte of Option Byte at address 0x1000
- the MSB byte of the Option Byte at address 0x1001

• the EEPROM code from address 0x1100 to 0x117F (128 bytes of EEPROM)

To set up a programming environment using a reference REPROM, proceed as follows:

1 Specify that you want to use a reference EPROM by setting (or removing) the appropriate jumper in the W1 jumper bank, depending on the type of EPB:

```
        W1           ST62EXX EPB                W1            ST63E1XX EPB
      ROM    ███  ◄──── This jumper on      RAM     □  ◄──── This jumper off
      RAM    □    ◄──── This jumper off     LOCK    ▓
      UNLOCK ▓                              OTP     ▓
      LOCK   ▓                              EEPROM  ▓
      OTP    ▓                              ST636X  ▓
      EPROM  ▓                              ST637X  ▓
      DEVICE ▓                              ST638X  ▓
      TYPE   ▓                              ST63E1XX ▓
      REMOTE ▓                              RESERVED ▓
      LOCAL  ███                            REMOTE  □
```

2 Select a reference EPROM, by setting the appropriate jumper combination in the W2 jumper bank:

```
              W2              W2              W2


EPROM Ref.:   27128           27256           27512
              2764
```
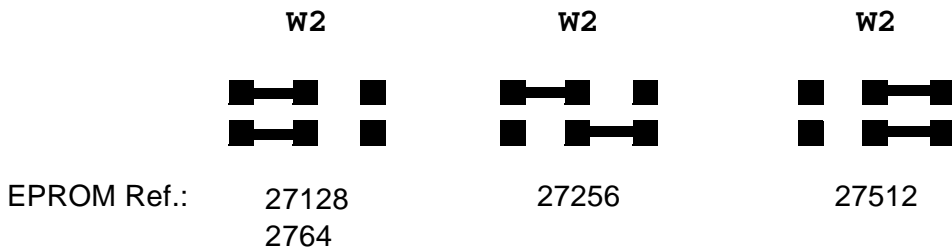
Note: Boards are configured by default for a 27256-type EPROM.

5.4 BLANK CHECKING YOUR MICROCONTROLLER

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

1 Select the reference EPROM as the reference component,

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

3 Insert into the socket the ST6xxxx MCU to be checked,

4 Set the MASTER/TARGET switch on the TARGET position:

5 Set the PROG/VERIF switch on the VERIF position:

6 Validate your settings by pressing the START key.

The blank checking process begins while the red BUSY LED (LD2) starts flashing.

The LED final status reflects the result of the operation:

When lit, the green OK LED (LD4) reports a successful end of operation.

When lit, the red ERROR LED (LD3) reports an error.

ALL LEDS on means that the device is protected against reading.

5.5 PROGRAMMING YOUR MICROCONTROLLER

To program your microcontroller, proceed as follows:

1 If you are using a Reference MCU, make sure that you have read the appropriate reference MCU beforehand, and that the appropriate code has been stored in the memory of your EPB, or
If you are using a reference REPROM, insert it in the appropriate socket, depending on the type of EPB:

In any case, make sure that the jumpers are properly set in the W1 and W2 jumper banks.

2 Insert the microcontroller to be programmed into the appropriate socket, depending on the type of EPB.

3 If you want to protect the programmed device against further reading, you must specify your choice before you start the programming procedure. The protection will take effect at the completion of the procedure. A protected device cannot be used for programming or verifying another device; this is therefore the best method to protect your software against piracy.
To select the read protect option, set the appropriate jumper in the W1 jumper bank, depending on the type of EPB:

```
     W1              ST62EXX EPB                W1            ST63E1XX EPB

    ROM         [      ]                       RAM        [      ]
    RAM         [      ]                       LOCK       [ ████ ]  ◄──── This jumper on
    UNLOCK      [      ]                        OTP       [      ]
    LOCK        [ ████ ]  ◄──── This jumper on EEPROM    [      ]
    OTP         [      ]                       ST636X     [      ]
    EPROM       [      ]                       ST637X     [      ]
    DEVICE      [      ]                       ST638X     [      ]
    TYPE        [      ]                       ST63E1XX   [      ]
    REMOTE      [      ]                       RESERVED   [      ]
    LOCAL       [ ████ ]                       REMOTE     [      ]
```
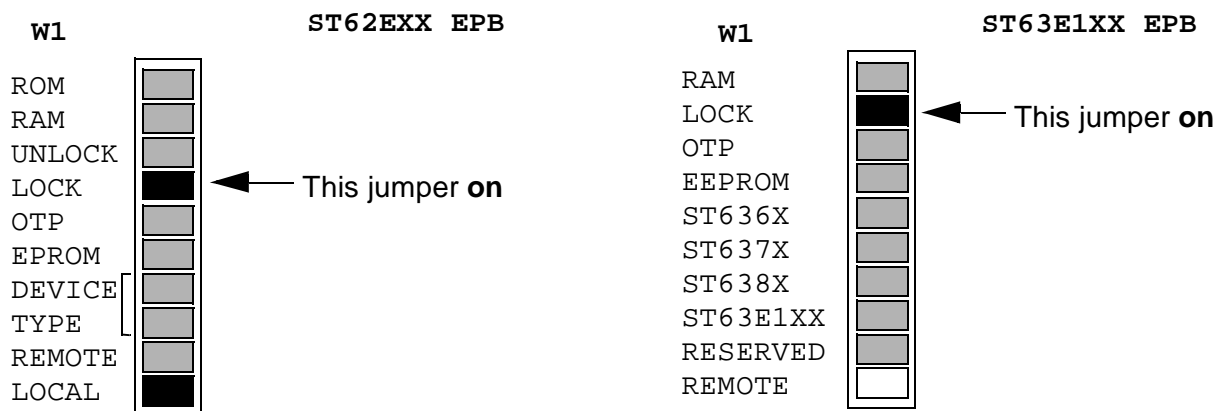
To prevent the device from being ever read-protected, set the UNLOCK jumper. This function is available for devices with the OPTION byte (ST62E2XX only).

Note: EPBs are shipped with the UNLOCK jumper on.

4 Set the MASTER/TARGET switch on the TARGET position.

5 Set the PROG/VERIF switch on the PROG position.

6 Validate your settings by pressing the START switch.

The programming process begins while the red BUSY LED (LD2) starts flashing.

The LED final status reflects the result of the operation:

 When lit, the green OK LED (LD4) reports a successful end of operation.

 When lit, the red ERROR LED (LD3) reports an error.

You can program another device without stopping the power supply: just replace the programmed microcontroller by a new blank one and push the START switch.

5.6 VERIFYING YOUR MICROCONTROLLER

To verify your microcontroller, proceed as follows:

1 If you are using a Reference MCU, make sure that you have read the appropriate reference MCU beforehand, and that the appropriate code has been stored in the memory of your EPB, or

If you are using a reference EPROM, insert it in the appropriate socket, depending on the type of EPB:

2 Insert the microcontroller to be programmed into the appropriate socket, depending on the type of EPB:

3 Set the MASTER/TARGET switch on the TARGET position,

4 Set the PROG/VERIF switch on the VERIF position,

5 Validate your settings by pressing the START switch.

The verifying process begins while the red BUSY LED (LD2) starts flashing.

The LED final status reflects the result of the operation:

When lit, the green OK LED (LD4) reports a successful end of operation.

When lit, the red ERROR LED (LD3) reports an error.

APPENDIX A

ABBREVIATIONS

EPB: EPROM Programming Board

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

EEPROM: Electrically Erasable Programmable Read-Only Memory

MCU: MicroController Unit

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

APPENDIX B

ST6XXXX EPROM PROGRAMMING BOARD LAYOUT

ST62E2X/1X/0X EPB:

RS232 SERIAL INTERFACE

J3

W1

JUMPER BANK

TARGET DEVICE
SOCKETS

ST62E00/01

ST62E15/E25

ST62E10/E20

W2

REFERENCE SOCKET

IC20

LEDS

REMOTE BUSY ERROR OK

SWITCHES

VERIF TARGET START

ST62E5X/6X EPB:

RS232 SERIAL INTERFACE

J1

W1

JUMPER BANK

ST62E55/65   ST62E53/60/63   ST62E52/62

TARGET DEVICE
SOCKETS

W2

REFERENCE SOCKET

U20

LEDS

REMOTE BUSY ERROR OK

SWITCHES

VERIF TARGET START

ST62E4X EPB:

RS232 SERIAL INTERFACE

J1

W1

JUMPER BANK

TARGET DEVICE
SOCKETS

ST62E45

ST62E42

W2

REFERENCE SOCKET

ST62E40

U18

LEDS

SWITCHES

ST62E46B

REMOTE BUSY ERROR OK

VERIF TARGET START

ST63E1XX EPB:

RS232 SERIAL INTERFACE

J3

W1

IN-CIRCUIT
PROGRAMMING
CONNECTOR

JUMPER BANK

W2

IC14

REFERENCE SOCKET

TARGET DEVICE
SOCKETS

LEDS

SWITCHES

ST63140
ST63142

ST63126
ST63156

ST636X
ST637X
ST638X

ST6369

REMOTE BUSY ERROR OK

VERIF TARGET START

APPENDIX C

CURRENT ST6 EPROM PROGRAMMING BOARDS

For more information on EPB general layout, "Appendix B" on page 34

Table 7. ST62E2XC EPB (serial interface)

Supported devices	Package	EPROM	EEPROM
ST62T00/T00C/T03/T03C	DIP16/SO16	1K	-
ST62E01/E01C/T01/T01C	DIP16/SO16	2K	-
ST62T05C/E05C	DIP16	6K	
ST62T08/T08C/T09/T09C	DIP20/SO20	1K	-
ST62E10/E10C/T10/T10C	DIP20/SO20	2K	-
ST62E15/E15C/T25/T25C	DIP28/SO28	2K	-
ST62T18C/E18C	DIP20	6K	
ST62E20/E20C/T20/T20C	DIP20/SO20	4K	-
ST62E25/E25C/T25/T25C	DIP28/SO28	4K	-
ST62T28C/E28C	DIP28	6K	
ST62T29C/E29C	DIP28	8K	

Table 8. ST62E3X EPB (parallel interface)

Supported devices	Package	EPROM	EEPROM
ST62E30B/T30B	DIP28/SO28	8K	128
ST62E32B/T32B	SDIP42	8K	128
ST62E35B/T35B	QFP52	8K	128

Table 9. ST62E4XB EPB (serial interface)

Supported devices	Package	EPROM	EEPROM
ST62E40/E40B/T40/T40B	QFP80	8K	128
ST62E42/E42B/T42/T42B	QFP64	8K	-
ST62E45/E45B/T45/T45B	QFP52	4K	128
ST62E46B/T46B	SDIP56	4K	128

Table 10. ST62E6XB EPB (serial interface)

Supported devices	Package	EPROM	EEPROM
ST62T52B/T52C	DIP20/SO20	2K	-
ST62T53B/T53C	DIP28/SO28	2K	-
ST62T55B/T55C	DIP28/SO28	4K	-
ST62E60B/T60B/E60C/T60C	DIP20/SO20	4K	128
ST62T62B/T62C	DIP16/SO16	2K	64
ST62T63B/63C	DIP20/SO20	2K	64
ST62E65B/T65B/E65C/T65C	DIP28/SO28	4K	128

Index

Index

Notes: