

STARTER KIT
FOR ST620x, ST621x and ST622x MCUs

HARDWARE FEATURES

- Immediate evaluation of all ST620x, ST621x and ST622x devices, with stand-alone demonstration routines.
- Simulation and debugging within the user's real application environment.
- In-socket programming of all DIL OTP and EPROM ST620x, ST621x and ST622x devices.
- In-circuit programming of all DIL and SO OTP and EPROM ST620x, ST621x and ST622x devices directly on the user's application board.

SOFTWARE FEATURES

- Software simulation, including I/O read/write.
- Assembler, Linker and Debugger.
- In-socket OTP and EPROM programming utilities.
- In-circuit OTP and EPROM programming utilities
- Application examples and demonstrations



Table of Contents

ST622x-KIT	1
1 INTRODUCTION	3
1.1 Where to go from here...	6
2 THE STARTER KIT HARDWARE	7
2.1 The ST6 Microcontrollers	7
2.2 The Starter Kit Board	7
2.3 Oscillator	10
2.4 Reset Button	10
2.5 + And - Buttons	10
2.6 LED Indicators	11
2.7 Resistance trimmer	11
2.8 Temperature Control Circuit	12
2.9 RS-232 Interface	13
2.10 Demonstration Selector Jumpers	14
3 INSTALLING THE STARTER KIT	15
3.1 Hardware and Software Requirements	15
3.2 Connecting the Power Supply	15
3.3 Installing the Software	15
4 RUNNING THE DEMOS	16
4.1 What the Demos Do	16
4.1.1 Demo 1 - Button/LED Indicator Control	16
4.1.2 Demo 2 - Voltage trimming and LED level indication	16
4.1.3 Demo 3 - Temperature Control	17
4.1.4 Demo 4 - RS-232 Communications	18
4.1.5 Demo 5 - Fuzzy Logic Heater Control	18
4.2 Running the Demonstration Programs	18
5 RUNNING THE ST6-REALIZER DEMO	21
6 CONNECTING EXTERNAL RESOURCES TO THE STARTER KIT BOARD	22
7 USING THE STARTER KIT BOARD AS A HARDWARE SIMULATOR	23
7.1 The Data Transmission Driver	24
7.2 Technical Limitations	24
7.3 Error Messages	26
7.4 Troubleshooting	26
8 PROGRAMMING ST6 MICROCONTROLLERS	27
8.1 Programming Signals	27
8.2 Setting Up the Starter Kit Board	28
8.3 In-Circuit Programming	30
8.3.1 Application Board Connections	30
8.4 Setting Up the Starter Kit Board for In-Circuit Programming	32
9 RUNNING YOUR OWN PROGRAM ON THE STARTER KIT BOARD	34
10 HARDWARE INFORMATION	36
10.1 Parts List	36
10.2 Starter Kit Board Schematic	37

1 INTRODUCTION

The ST622x Starter Kit provides you with all you need to start designing, developing and evaluating programs for ST620x, ST621x and ST622x microcontrollers immediately.

The ST622x Starter Kit includes:

- The ST6 assembler and linker, AST6 and LST6.
- A demonstration of the ST6 application development tool, ST6-Realizer.
- The ST6 Windows debugger, WGDB6.
- The Windows ST6 microcontroller programmer, Epromer.
- The ST6 Starter Kit board, which serves as a demonstration board and low-cost debugging tool.
- Some demonstration programs that show how ST6 microcontrollers use the Starter Kit board resources.
- Some example programs for the ST622x family.
- Two ST62E25, two ST62E20 and one ST62E01 microcontrollers.
- A complete set of paper documentation and online help.

The demonstration programs, that come pre-loaded on an ST62E25 microcontroller, show how the powerful features of ST6 microcontrollers operate in a real environment. The demonstration programs use the hardware resources provided on the Starter Kit board, which include reset and data control buttons, LED indicators, a resistance trimmer, temperature control circuit and an RS-232 interface.

Using the ST6 assembler and linker, AST6 and LST6, you can assemble and link ST6 programs. The AST6/LST6 User Manual will guide you through the steps of developing, assembling and linking programs for the ST6. The Starter Kit software includes a set of example programs of typical ST6 applications. These are installed in the directory **C: \ST622x\examples**.

For a fast-track solution for developing bug-free programs for the ST6, without the hassle of writing assembler code, try out the ST6-Realizer demonstration.

Once you have developed your ST6 program, you can use the Windows-based ST6 program debugger, WGDB6/SIMULATOR, together with the Starter Kit board, as a low-cost but powerful debugging tool. WGDB6 includes an ST6 simulator, that simulates the execution of your program, and uses the ST6 that is plugged into the Starter Kit board to emulate all transactions that are performed with the data space. Thus, using the Starter Kit board with WGDB6, you can view how the microcontroller peripherals behave when your program is executed. WGDB6 includes powerful debugging features, such as source-level debugging, instruction and conditional memory access breakpoints and selective trace recording. The WGDB6 Get-

Introduction

ting Started manual and online help will lead you through the debugging process using WGDB6.

When your program is ready, Epromer provides you with an easy-to-use Windows interface, which lets you prepare executable code, then write it to the ST6 microcontroller that is plugged into either one of the DIL sockets on the Starter Kit board, or your own in-circuit application board that is connected to the Starter Kit board.

To observe and evaluate the consequences of your program on the resources it controls, you can run it on an ST6 microcontroller that is plugged into the Starter Kit board. If it controls a resource that is not included on the Starter Kit board, you can connect your own resource to the board.

Instructions for use - Warning

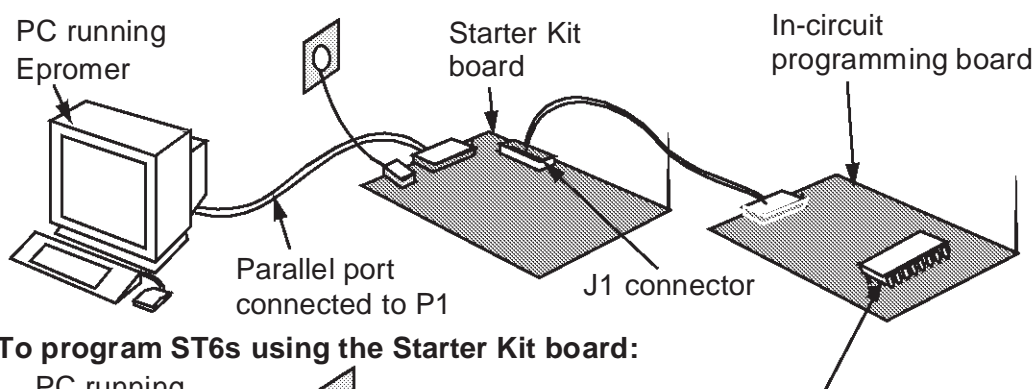
This product conforms with the 89/336/EEC directive; it also complies with the EN55022 emissions standard for ITE, as well as with generic 50082-1 immunity standards.

The product is a **Class A apparatus**. In a residential environment this device may cause radioelectrical disturbances which may require that the user adopt appropriate precautions.

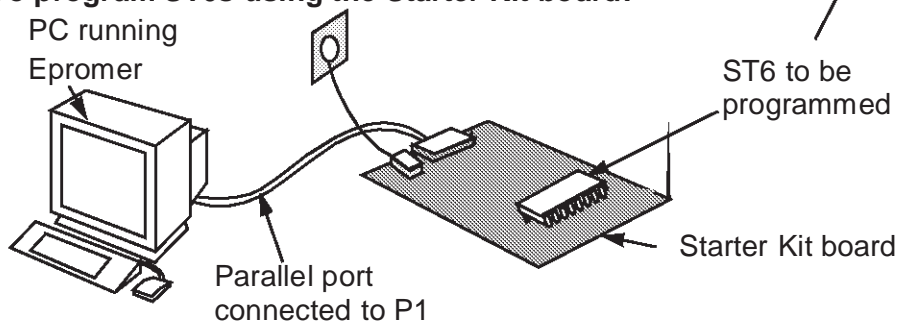
The product is not contained in an outer casing, and cannot therefore be immune against electrostatic discharge (ESD): **it should therefore only be handled at static safe work stations**.

The following diagram summarises the possible uses of the Starter Kit board and the hardware setup required for each one.

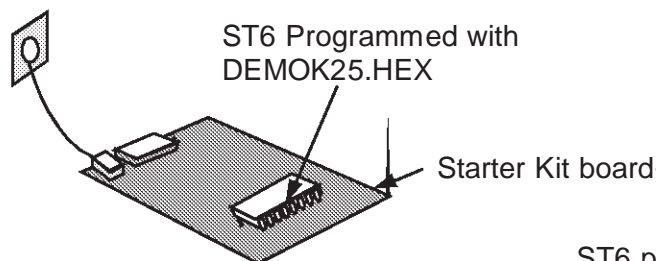
To program ST6s on your own in-circuit programming board:



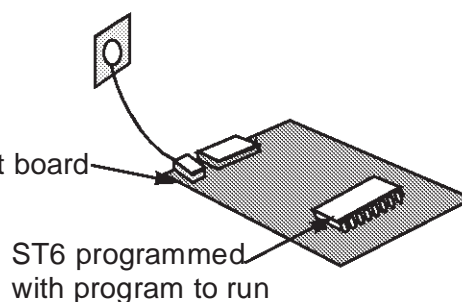
To program ST6s using the Starter Kit board:



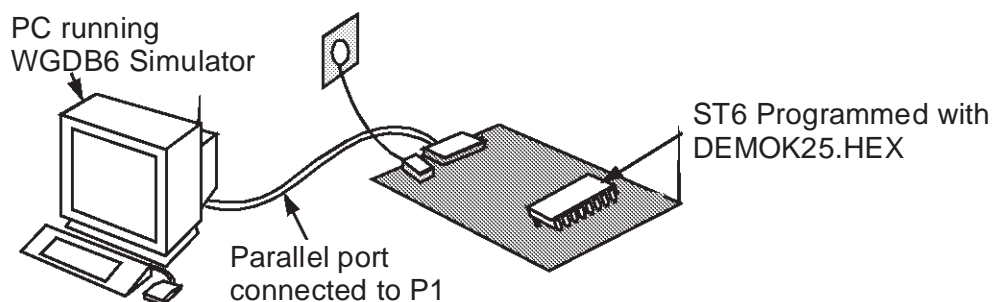
To run the demonstrations:



To run your own program:



To use the Starter Kit board as a software simulator:



1.1 Where to go from here...

The following table directs you to where you should look for further information about using the ST6 Starter Kit.

To:	Refer to:
Find out about the Starter Kit board and ST6 microcontrollers provided with the kit.	"The Starter Kit Hardware" on page 7 of this book.
Install the Starter Kit software, and connect the power supply to the board.	"Installing the Starter Kit" on page 15 of this book.
Install and run the ST6-Realizer demonstration software.	"Running the ST6-Realizer Demo" on page 21 of this book.
Find out what the demonstration applications do, and run them.	"Running the Demos" on page 16 of this book.
Learn how to develop source code for AST6 and LST6.	"AST6/LST6 Assembler and Linker for the ST6 Family - User Manual".
Prepare the Starter Kit board for use as an ST6 hardware simulator with WGDB6.	"Using The Starter Kit Board as a Hardware Simulator" on page 23 of this book.
Install WGDB6, and learn how to use it for debugging your programs.	"WGDB6 Debugger for the ST6 Family - Getting Started".
Prepare the Starter Kit board for programming ST6 microcontrollers using Epromer.	"Programming ST6 Microcontrollers" on page 27 of this book.
Prepare the Starter Kit board for connecting your own in-circuit programming board.	"In-Circuit Programming" on page 30 of this book.
Learn how to use Epromer for programming ST6 microcontrollers.	The Epromer online help.
Connect your own hardware resource to the Starter Kit board.	"Connecting External Resources to the Starter Kit Board" on page 22 of this book.
Run your own program on an ST6 using the Starter Kit board.	"Running Your Own program on the Starter Kit Board" on page 34.

2 THE STARTER KIT HARDWARE

This section describes the ST6 microcontrollers and the Starter Kit board that come with the ST6 Starter Kit. A full schematic of the Starter Kit board is provided in “Hardware Information” on page 36.

2.1 The ST6 Microcontrollers

The Starter Kit includes the following microcontrollers:

- Two ST62E25, Two ST62E20, One ST62E01

The ST62E25 microcontroller labelled DEMOK25 is pre-loaded with the code **DEMOK25.HEX**, which includes the demonstration programs (see “Running the Demos” on page 16), as well as the communications protocol program, that enables you to use the Starter Kit board as a simulator (see “Using The Starter Kit Board as a Hardware Simulator” on page 23). The other ST6 microcontrollers are blank.

The file Demok25.hex is provided on the diskette labelled “ST622x Starter Kit“, so that if you erase it from the ST62E25, you can re-program it following the instructions given in “Programming ST6 Microcontrollers” on page 27.

2.2 The Starter Kit Board

The Starter Kit board includes the following resources:

- Reset and data control buttons.
- LED indicators.
- Resistance trimmer.
- Temperature control circuit.
- RS-232 interface.
- Demonstration program selector jumpers.

It comes with its own power supply unit that can be plugged into an AC mains source, or a DC source with the following characteristics:

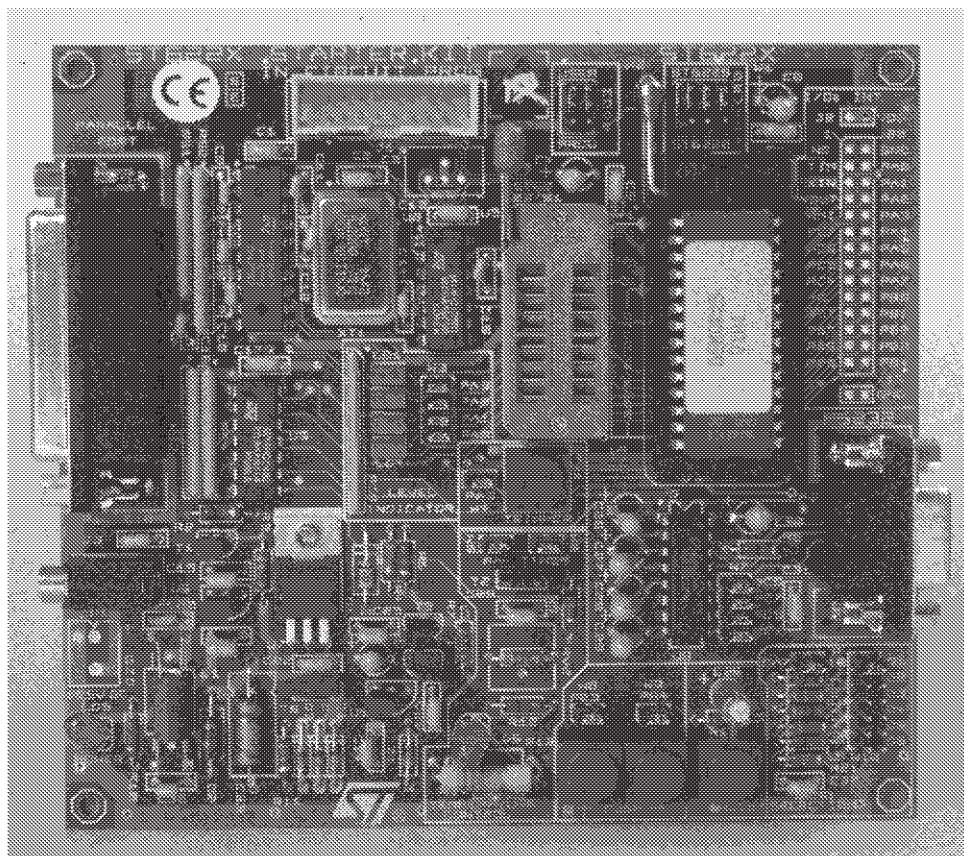
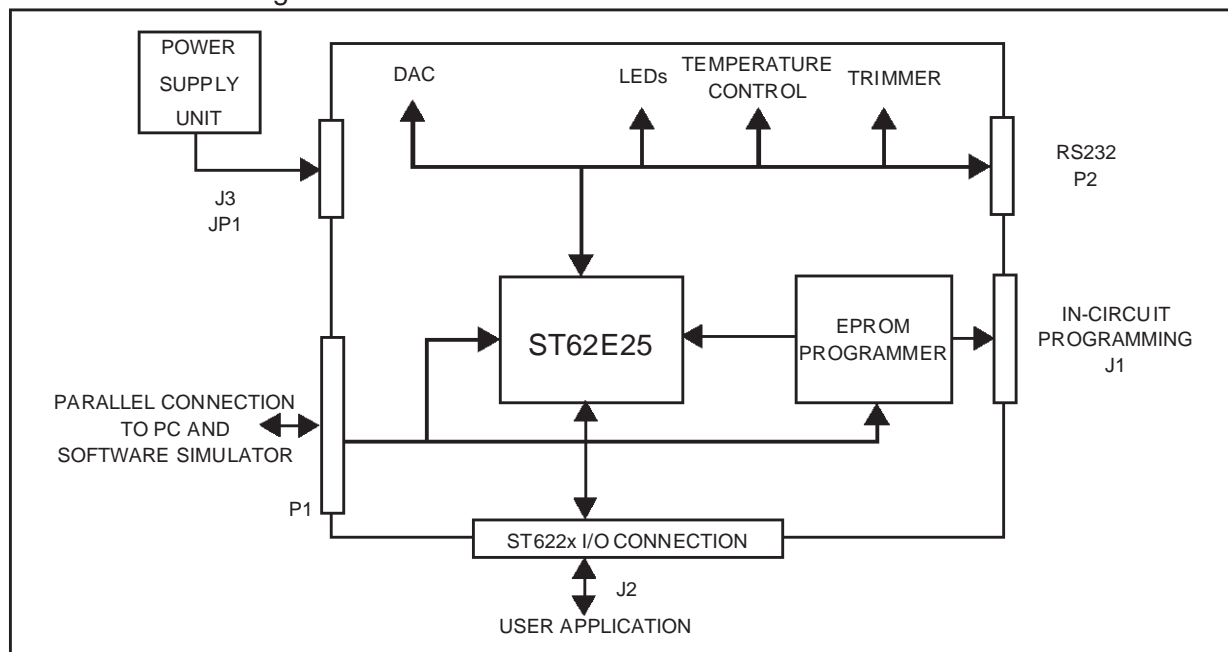
- Voltage: 16V min./20V max., Current: 100 mA min.

It includes the following connectors:

- A parallel port connector (P1) for connection to the host PC when it is used as a hardware simulator or for programming.
- A remote resource I/O interface connector (J2) to which you can connect your own hardware resource.
- An RS-232 connector, which you can use for observing RS-232 communication control using an ST6.
- A connector for your own in-circuit ST6 programming board. See “Application Board Connections” on page 30 for further details.

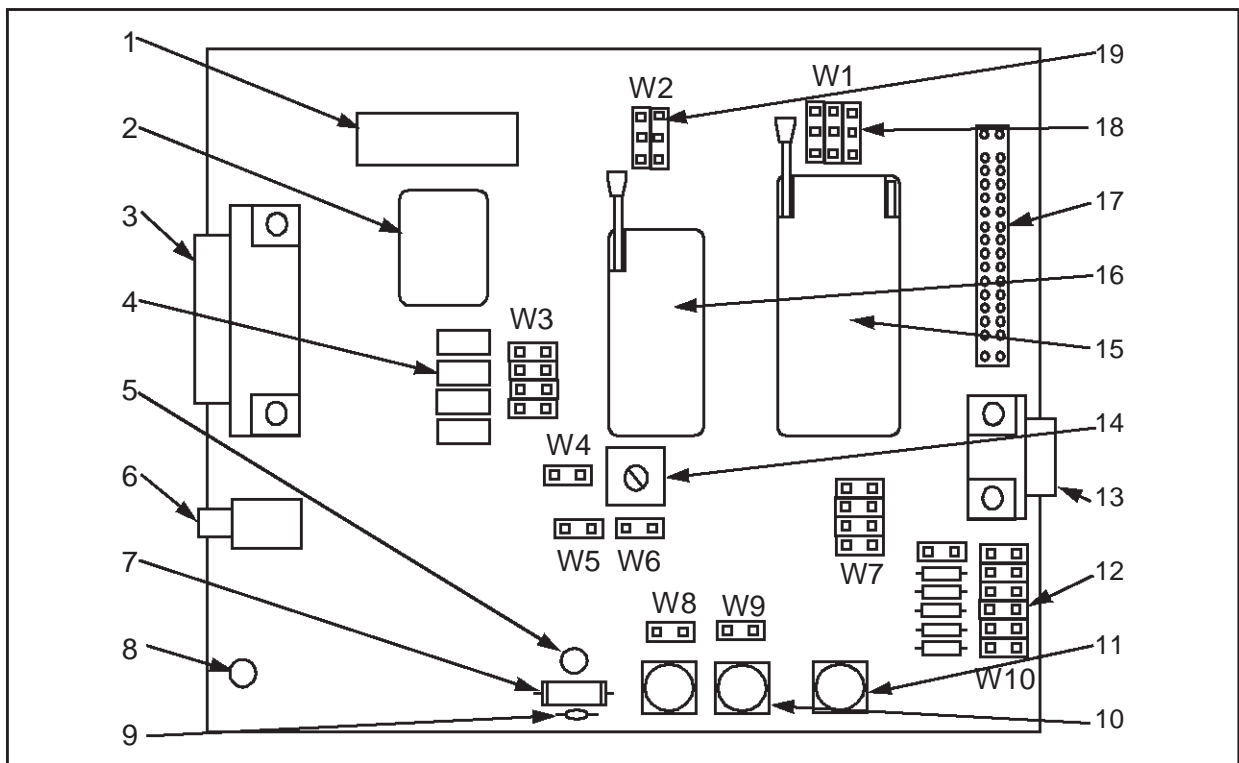
The Starter Kit Hardware

Below is a block diagram of the Starter Kit board:



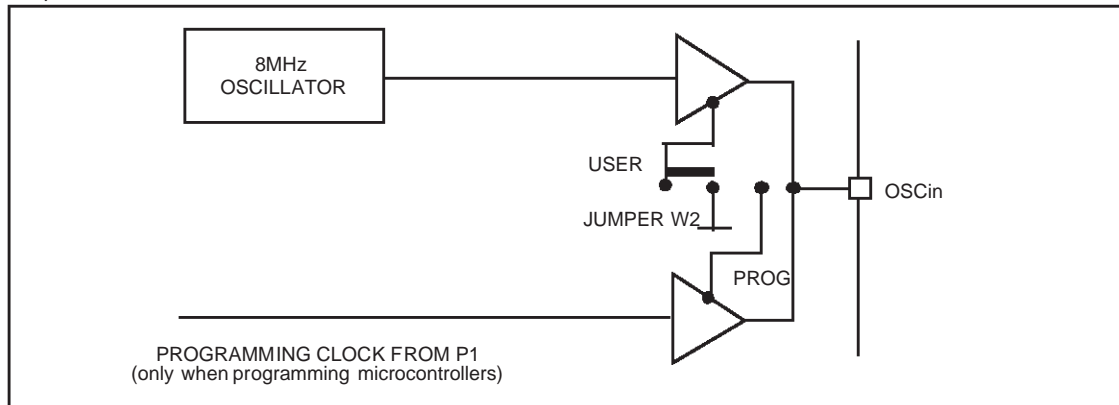
The following diagram shows the layout of the Starter Kit board.

- | | |
|--|---|
| 1 In-circuit programming connector (J1). | 11 RESET button. |
| 2 8 Mhz oscillator. | 12 Demonstration routine selector. |
| 3 PC connector P1. | 13 RS232 interface circuit and connector. |
| 4 4 LEDs. | 14 10 K Ω trimmer. |
| 5 Heater resistor LED indicator LD6. | 15 DIL 20-28 ZIF MCU socket. |
| 6 Power supply JACK connector J3. | 16 DIL 16 ZIF MCU socket. |
| 7 Heater resistor. | 17 Remote resource I/O interface connector J2. |
| 8 Power supply LED indicator LD5. | 18 "ST6220" or "ST6225" device selection jumpers W1. |
| 9 Thermistor. | 19 "Programming" or "User" operating mode selection jumpers W2. |
| 10 "+" and "-" buttons. | |



2.3 Oscillator

An oscillator feeds the ST6225 OSCIN input with an 8 MHz clock signal. You can disconnect the oscillator by removing the two jumpers W2 (marked 19 on the Starter Kit board diagram on page 9).



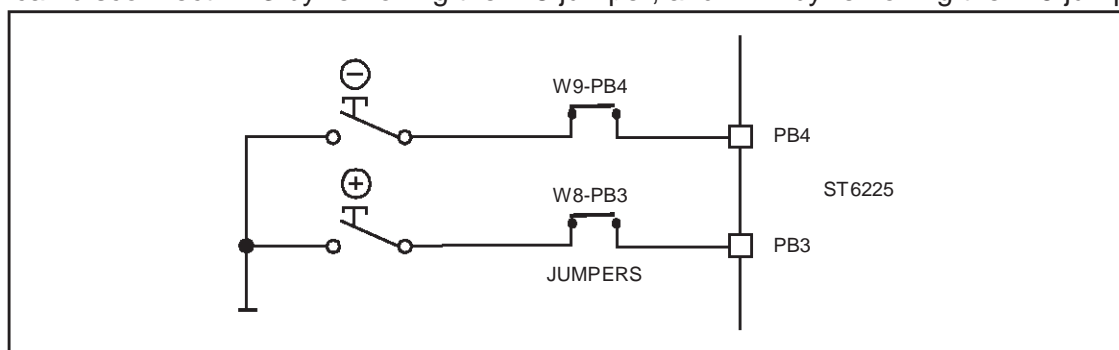
2.4 Reset Button

This activates the ST6225 RESET input when pressed. A power-on reset circuit is also provided.

2.5 + And - Buttons

These are connected to PB3 and PB4 pins on the microcontroller respectively. They drive the PB3 and PB4 inputs down to GND when pressed.

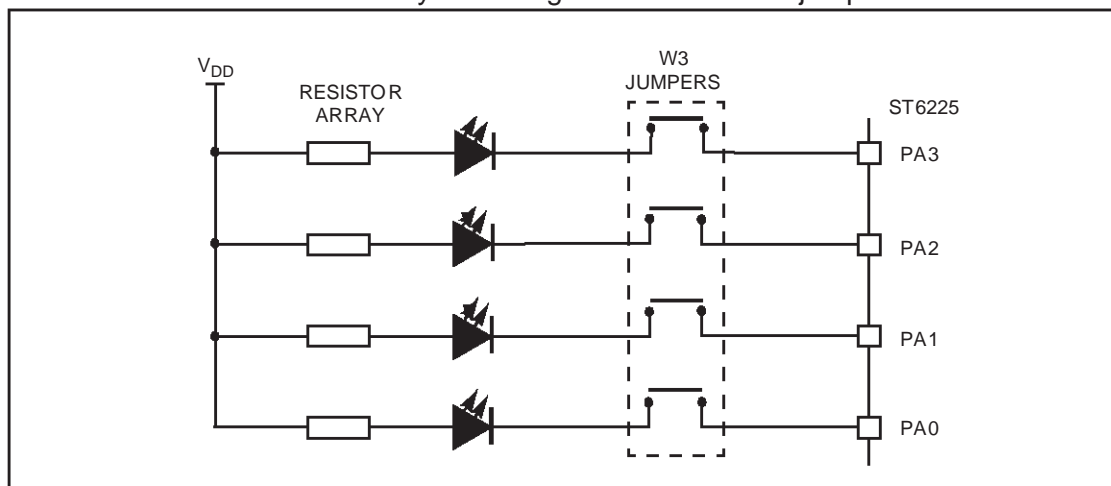
You can disconnect PB3 by removing the W8 jumper, and PB4 by removing the W9 jumper.



2.6 LED Indicators

Four LEDs are used for level indication in the demonstration routines. They are connected to the ST6225 pins: PA0 to PA4 (when programmed as outputs) to demonstrate direct LED-driving capability.

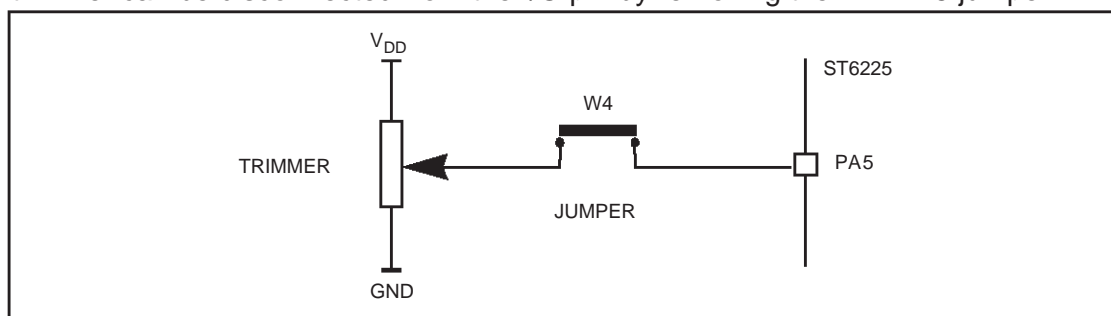
The 4 LEDs can be disconnected by removing the W3-PA0 to 3 jumpers.



2.7 Resistance trimmer

A 10 K Ω resistance trimmer feeds the ST6225 PA5 I/O pin (when programmed as an A/D Converter input) with a variable voltage (0 to 5V DC). It is used for A/D conversion demonstration/evaluation.

The trimmer can be disconnected from the I/O pin by removing the W4-PA5 jumper.



2.8 Temperature Control Circuit

This circuit demonstrates temperature control, using the on-chip A/D converter.

A Heater resistor circuit (150Ω , 1 Watt) is driven by the TIMER output of the ST6225 microcontroller. The heat level can be controlled by varying the duty cycle of a PWM signal present on the TIMER output.

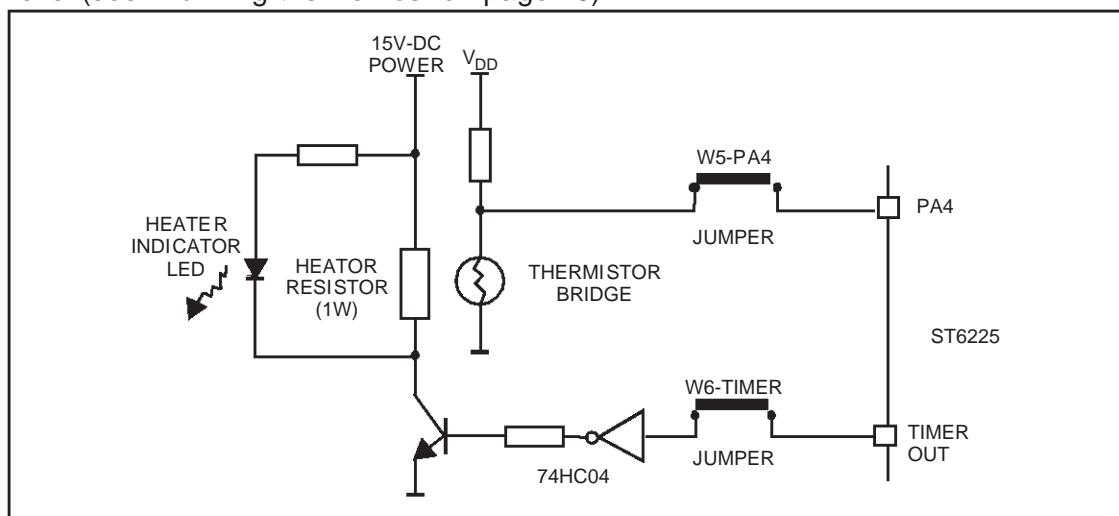
The Heater circuit can be disconnected from the TIMER output pin by removing the W6-TIMER jumper.

A thermistor bridge (Negative Temperature Coefficient) is connected to the PA4 I/O pin, which is programmed as an A/D converter input. The voltage value at the A/D converter input decreases as the thermistor temperature increases.

The Thermistor circuit can be disconnected from PA4 by removing the W5 jumper.

The thermistor is located in close proximity to the Heater resistor on the Starter Kit board, so that it can be used as a sensor to detect the temperature of the resistor body.

Demonstration 6 shows how this circuit is used as a feedback loop that controls the temperature level (see “Running the Demos” on page 16).



2.9 RS-232 Interface

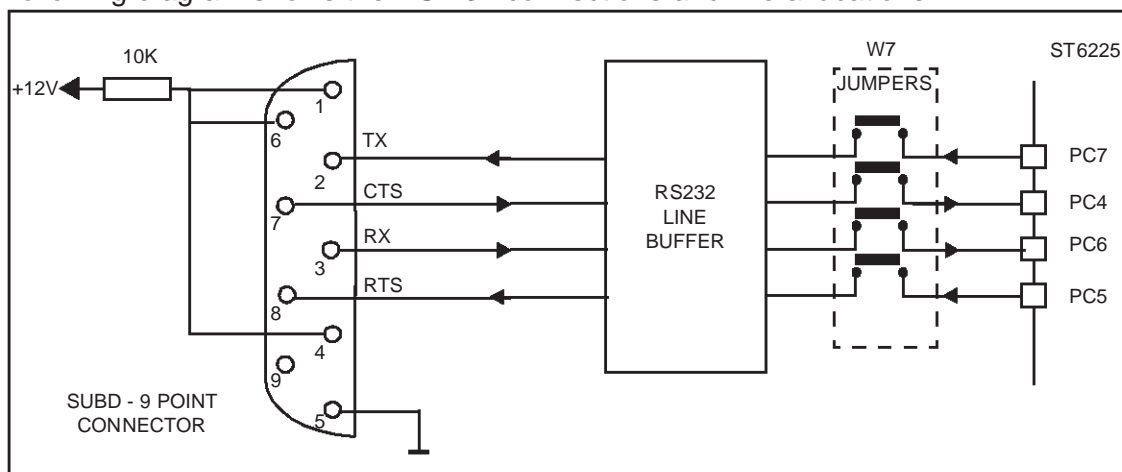
The RS-232 interface enables you to communicate with the pre-programmed ST6225 micro-controller provided with the Starter Kit. It includes an RS-232 buffer circuit that is connected to a standard PC-compatible RS-232 SUBD-9 connector.

The following table lists the RS-232/ST6 pin connections:

Signal Name	SUBD-9 Pin	ST6 Pin
Data Transmission (TX)	2	PC7
Data Reception (RX)	3	PC6
Request to Send (RTS)	8	PC5
Clear to Send (CTS)	7	PC4

You can disconnect these by removing the corresponding jumpers from W7.

The following diagram shows the RS-232 connections and line allocations:



TX, RX, CTS and RTS are defined so that the board is used as a slave. To use the board as a master, swap both the RX and TX and the CTS and RTS pin connections on the cable.

2.10 Demonstration Selector Jumpers

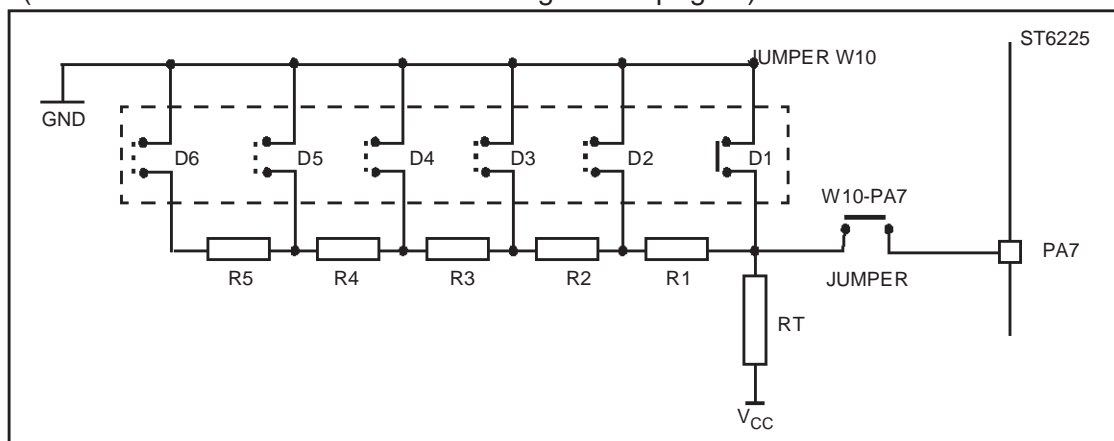
The demonstration selector is made up of a 6-position jumper W10 (marked 12 on the Starter Kit board diagram on page 9), with each jumper connected to a resistor. Each resistor generates a different voltage.

After reset, the voltage value generated by the resistor whose jumper is installed is sent to PA3 on the ST6225. PA3 is programmed as an A/D converter. The program installed on the ST6225 uses the input from PA3 to select the appropriate demo. The following table lists the voltage values generated by each resistor:

Resistor value:	Theoretical voltage value:
RT: 10 K Ω	No JUMPER: 5 V
R1: 680 Ω	JUMPER-D1: 0 V
R2: 750 Ω	JUMPER-D2: 333 mV
R3: 820 Ω	JUMPER-D3: 666 mV
R4: 1 K Ω	JUMPER-D4: 999 mV
R5: 1.2 K Ω	JUMPER-D5: 1.332 V

The same principle can be used for keyboard decoding. For a complete example of this, refer to the SGS-Thomson application note: "AN431: Using ST6 Analog Inputs for Multiple Key Decoding".

You can disconnect the demonstration selector from PA7 by removing the PA7 jumper from W10 (marked 12 on the Starter Kit board diagram on page 9).



3 INSTALLING THE STARTER KIT

3.1 Hardware and Software Requirements

To be able to install and run the ST6 Starter Kit, you need a PC with:

- A 3 1/2" Floppy Disk Drive
- 1.5 Mbytes free memory space
- A free Centronics compatible parallel port connector
- MS-WindowsTM 3.11, NT or 95.

3.2 Connecting the Power Supply

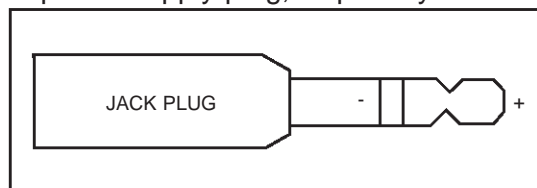
If you have AC mains supply, connect the Jack plug on the power supply cable provided to the J3 input socket, then connect the mains plug to a mains source.

If you have DC mains supply, connect the male plug on the power supply cable provided to the J3 input socket, then connect the mains plug to a mains source with the following characteristics:

- Voltage: 16V min./20V max.
- Current: 100 mA min.

NOTE: To avoid a short circuit, always connect the power input cable to the starter kit board before connecting it to a mains power supply.

If you use your own 3.5 mm power supply plug, its polarity must be as follows:



3.3 Installing the Software

- 1 Place the installation diskette provided into your floppy disk drive.
- 2 In Windows Explorer or File Manager, view the contents of the diskette, then double-click the **Setup** icon.
- 3 Follow the instructions as they appear on screen.

4 RUNNING THE DEMOS

This section describes the demonstration programs that are provided with the Starter Kit and explains how to run them.

4.1 What the Demos Do

The following paragraphs describe the demos that come pre-loaded with the ST6 Starter Kit demos. See “Running the Demonstration Programs” on page 18 below for details on how to select and run a demo.

The source files of these demos are provided with the Starter Kit software in the file **C:\ST622X\SK622XL\DEMOK25.ASM**.

4.1.1 Demo 1 - Button/LED Indicator Control

1 Initialises the pins as follows:

This pin:	Is initialised as:
PB3	Input with pull-up and interrupt. Connected to + button.
PB4	Input with pull-up and interrupt. Connected to - button.
PA0 to PA3	Push-pull outputs. Connected to the four LEDs (as marked on board).

2 Performs a task relating to your actions:

When you do this:	The program does this:
Press the + button	Jumps to an interrupt subroutine that lights up each led in turn, from LD4 to LD1.
Press the - button	Jumps to an interrupt subroutine that lights up each led in turn, from LD1 to LD4.

4.1.2 Demo 2 - Voltage trimming and LED level indication

1 Initialises the pins as follows:

This pin:	Is initialised as:
PA5	Analog input. Connected to the trimmer.
PA0 to PA3	Push-pull outputs. Connected to the four LEDs (as marked on board).

- 2 Reads the A/D converter data register, and turns on a LED according to the Voltage value input by the trimmer:

This voltage: Turns this LED on:

0 to 1	none
1 to 2	LD1
2 to 3	LD2
3 to 4	LD3
4 to 5	LD4

Adjusting the voltage trimmer (marked 9 on the Starter Kit board diagram on page 9) turns on the appropriate LED. Pressing the + or - buttons has the same effect as increasing or decreasing the voltage.

4.1.3 Demo 3 - Temperature Control

- 1 Initialises the pins as follows:

This pin:	Is initialised as:
PA4	Analog input. Connected to the thermistor circuit.
TIMER	Push-pull output. Connected to the Heater Resistor circuit.
PB3	Input with pull-up and interrupt. Connected to + button.
PA0 to PA3	Push-pull outputs. Connected to the four LEDs (as marked on board).

- 2 Reads and stores the A/D converter data register value. This value indicates the temperature at reset.
- 3 Reads and stores the A/D converter data register value at regular intervals. If this value exceeds the value that was stored at reset, a LED is turned on indicating the difference between the two values. The higher the difference is between the stored value and the read value, the higher LED number is turned on (roughly in steps of $LD(n+1)$ for each additional degree difference).

You can either increase the temperature by touching the thermistor (marked 14 on the Starter Kit board diagram on page 9) or pressing the + button.

4.1.4 Demo 4 - RS-232 Communications

This demonstration shows how an RS-232 communication line buffer can be managed using an ST6225 microcontroller.

To run this demonstration:

- 1 Connect the RS-232 connector on the Starter Kit board to a serial port on your PC using the RS-232 cable provided.
- 2 On the host PC, in MS-DOS, execute the program: **PC232_25** (this is supplied on the diskette marked "ST622x Starter Kit").
- 3 Follow the instructions as they appear on screen.

4.1.5 Demo 5 - Fuzzy Logic Heater Control

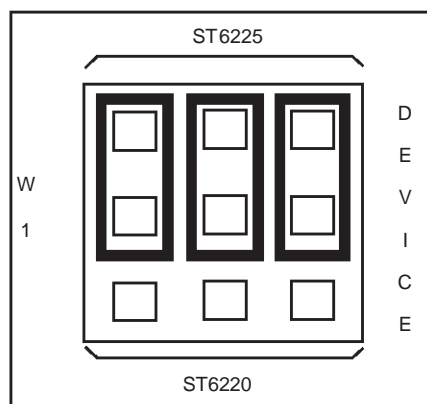
This demonstration shows how a heater control application can be developed using a Fuzzy Logic description. The application was developed using fuzzyTECH explorer development tool. For more information about this demonstration, execute the program **Fuzzykit.exe** from MS-DOS (this is supplied on the diskette marked "ST622x Starter Kit").

4.2 Running the Demonstration Programs

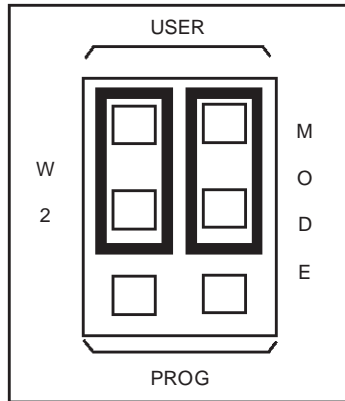
The ST62E25 microcontroller labelled DEMOK25 is programmed with the demonstration software. If this software has been erased from the microcontroller, you can reprogram it from the file **DEMOK25.HEX** (this is supplied on the diskette marked "ST622x Starter Kit"). For details of how to program microcontrollers refer to "Programming ST6 Microcontrollers" on page 27.

To run the demonstrations:

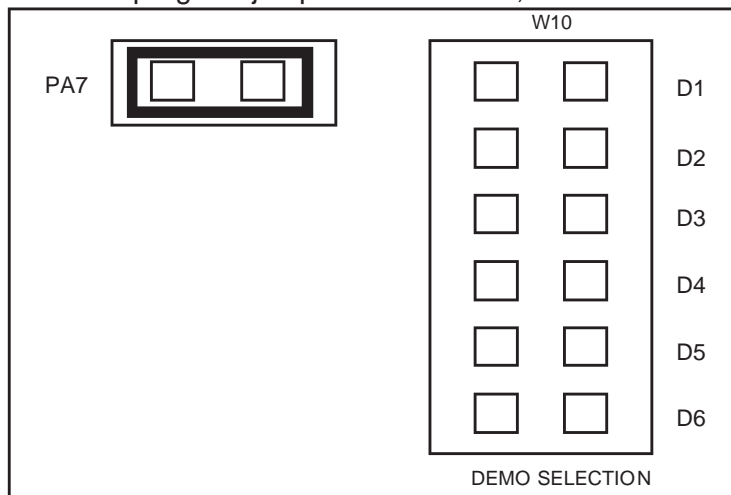
- 1 Power down the Starter Kit board.
- 2 Make sure that the pre-programmed ST62E25 is plugged into the DIL connector, and that the DEVICE jumpers W1 (marked 18 on the Starter Kit board diagram on page 9) are set to ST6225 as shown in the following diagram:



- 3 Select the USER mode using the jumpers marked W2 (marked 19 on the Starter Kit board diagram on page 9), as shown in the diagram below:

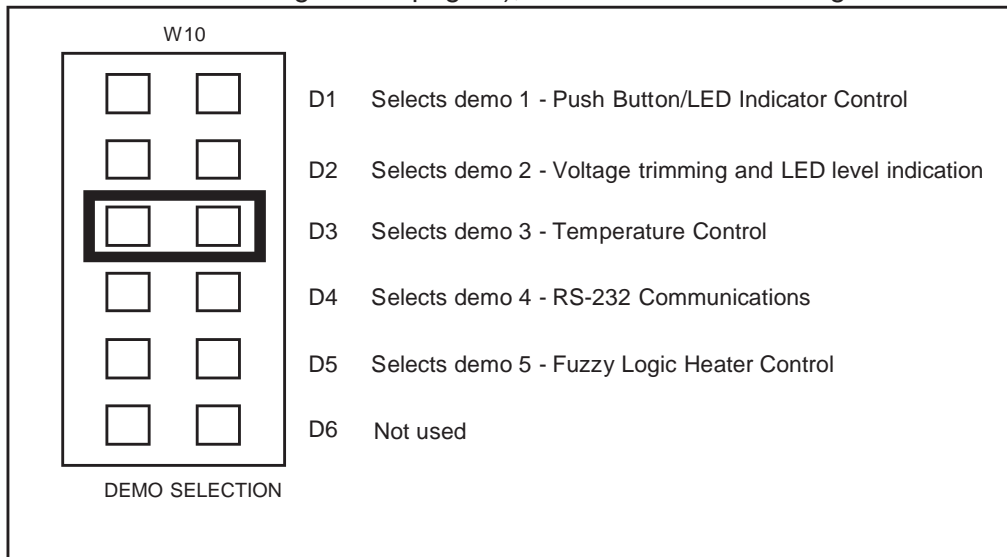


- 4 Disconnect the cable from the parallel port (P1) connection, if it is connected.
- 5 Power up the Starter Kit board.
- 6 Install the demonstration program jumper marked PA7, as shown in the diagram below:



Running the Demos

- 7** Select the demo you want to run, by installing the appropriate jumper on W10 (marked 12 on the Starter Kit board diagram on page 9), as indicated on the diagram below:



For example, in the above diagram demo 3 is selected.

- 8** Press the reset button.

The selected demo is now ran.

To run a different demo, repeat steps 7 and 8.

5 RUNNING THE ST6-REALIZER DEMO

ST6-Realizer is a Windows CASE tool that enables you to develop high-quality applications for the ST6 family of microcontrollers. Using ST6-Realizer, you design your application by drawing symbols and wiring them together. Each symbol represents a process, such as adding two values, and is linked to an ST62 assembly code macro. The wires represent the flow of data, and are linked to variables and constants. You can attach attributes to symbols and wires, in order to attach extra characteristics to them. For example, attaching an attribute of type UINT to a wire defines its value capacity to that of an unsigned integer (0 to 65536).

When using ST6-Realizer, you design your application in schemes. A scheme is like a sheet of paper on which you place symbols and draw wires. Each application is designed on one scheme.

To install the ST6-Realizer demo:

- 1 Place the diskette labelled "ST6-Realizer Demo disk" into your floppy disk drive.
- 2 In Windows Explorer or File Manager, browse the contents of the diskette, and double-click the **Install** icon.

To run the ST6-Realizer demo:

If you are using Windows 3.x, double-click the ST6-Realizer demo icon in the ST6 Tools group.

If you are using Windows 95, click **Start, Programs, ST6 Tools**, then the **ST6-Realizer demo**.

6 CONNECTING EXTERNAL RESOURCES TO THE STARTER KIT BOARD

You can connect your own external resources to the pre-programmed ST62E25 to debug or evaluate your programs, using the connector J2 (marked 17 on the Starter Kit board diagram on page 9).

To be able to connect your own resources to the Starter Kit board, you must disconnect the resources that are already connected to the ST62E25, to avoid external resource/Starter Kit board resource conflicts.

The following table lists the Starter Kit board resources and the corresponding J2 connections, and indicates the jumper that disconnects each resource.

On-board resource	Jumper to disconnect	ST6225 I/O	PIN	PIN	ST6225 I/O	Jumper to disconnect	On-board resource
+5V Supply	-	VDD	1	28	VSS	-	GND
Heater resis. control	W6-TIMER	TIMER	2	27	PA0	W3-PA0	LED Indicator
8MHz OSC.	W2	OSCIN	3	26	PA1	W3-PA1	LED Indicator
none	-	OSCO	4	25	PA2	W3-PA2	LED Indicator
System tasks (Simulator link)	(*)	NMI	5	24	PA3	W3-PA3	LED Indicator
RS232 Driver TX Output	W7-PC7	PC7	6	23	PA4	W5-PA4	Thermistor
RS232 Driver RX Input	W7-PC6	PC6	7	22	PA5	W4-PA5	Trimmer
RS232 Driver RTS Output	W7-PC5	PC5	8	21	PA6	-	none
RS232 Driver CTS Input	W7-PC4	PC4	9	20	PA7	W10-PA7	Routine selector
System tasks (Programming)	W2	VPP	10	19	PB0	(*)	System task (Simul. link)
Reset Button	-	RESET/	11	18	PB1	(*)	System task (Simul. link)
System tasks (Programming)	W2	PB7	12	17	PB2	(*)	System task (Simul. link)
System tasks (Programming)	W2	PB6	13	16	PB3	W8-PB3	+ Button
System tasks (Programming)	W2	PB5	14	15	PB4	W9-PB4	- Button

(*) This is not available if the Starter Kit board is connected to a host PC.

NOTE: Some of the signals on the J2 connector are used during ST6 programming, thus you must disconnect any external resource that is connected to J2 before using the Starter Kit board for programming.

7 USING THE STARTER KIT BOARD AS A HARDWARE SIMULATOR

WGDB6, the ST6 debugger that runs under Windows, lets you test your programs without having to program the EPROM of your target ST6. Depending how much information you want, and how close to real life you want your test environment to be, you can use WGDB6 in one of three ways:

- As a software simulator. If you use WGDB6 as a simulator, you need not attach any additional hardware to your PC. The ST6 simulator program, that comes with WGDB6 and is run when you run WGDB6/Simulator, simulates the execution of your program, letting you step through the code and see what happens as the program runs. WGDB6 simulator includes Wave Form Editor, which simulates the output of the pins on your target ST6 in relation to inputs that you define, enabling you to see how its peripherals react to the inputs they receive.
- With an ST6 hardware emulator. Emulators are hardware systems that act as your target microcontroller, at the same time capturing detailed information, such as which areas of memory are accessed by the program and what happens when they are accessed. In this case, WGDB6/Emulator provides an interface between the emulator and your PC, displaying data captured by the emulator and letting you implement the WGDB6 features in the emulator, such as software or hardware breakpoints.
- With the Starter Kit board as a hardware simulator. This is a cross between the above two. The WGDB6 software simulator simulates the execution of your program, but each time the data space is accessed, it accesses that of the ST6 that is plugged into your Starter Kit board. Thus, using the Starter Kit board with WGDB6, you can view how the real microcontroller peripherals behave when your program is executed.

This section describes the third option, how to use the Starter Kit board as a hardware simulator.

You can use the Starter Kit board to emulate any ST620x, ST621x or ST622x microcontroller. Note, however that you must use the pre-programmed ST62E25 microcontroller, labelled DEMOK25 supplied with the kit for hardware simulation. Thus, when simulating programs designed for other microcontrollers, make sure that you do not use resources that are not available on the microcontroller your application is designed for.

7.1 The Data Transmission Driver

Data is transferred between the simulated peripheral registers and the ST620x, ST621x and ST622x registers via the host PC's parallel port. The DEMOK25.HEX program, with which the ST62E25 microcontroller that is plugged into the Starter Kit board must be loaded includes the transmission driver.

The data transfer driver uses the following bits:

PC parallel port	ST6225	WGDB6 Use
D2	PB2	Synchronisation
D3	PB1	Write data to MCU
D4	RESET	Hardware reset of peripherals
D6	NMI	Initiates data transfer
SDOP	PB0	Read data from MCU

7.2 Technical Limitations

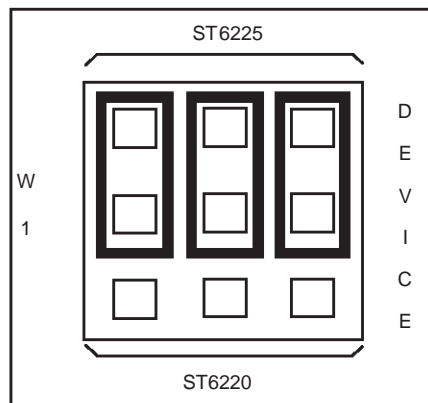
The Starter Kit board has the following limitations when used with WGDB6 as a hardware simulator:

- Real-time program execution is not supported.
- Resetting the ST6225 by power on, pressing the Reset button or external reset does not reset the simulated ST6 core. To perform a complete simulated reset, use the WGDB6 reset command instead.
- Interrupts sent by the ST62E25 microcontroller are not supported by the WGDB6 simulator.
- The pins: NMI, PB0, PB1 and PB2 on the ST62E25 microcontroller are used for communications with the host PC, and are thus not available for simulation.

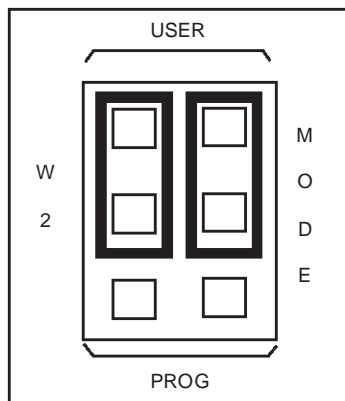
NOTE: Do not connect any external resources to the corresponding J2 connector pins when using the Starter Kit board as a peripheral emulator.

To use the Starter Kit board as a hardware simulator:

- 1 Power down the Starter Kit board.
- 2 Make sure that the ST62E25 labelled DEMOK25 is plugged into the DIL28 socket, and that the DEVICE jumpers W1 (marked 18 on the Starter Kit board diagram on page 9) are set to ST6225 as shown in the following diagram:



- 3 Select the USER mode using the jumpers marked W2 (marked 19 on the Starter Kit board diagram on page 9), as shown in the diagram below:



- 4 Connect the Parallel port P1 on the Starter Kit board to a spare parallel port on your PC using the cable provided with the Starter Kit.
- 5 Power up the Starter Kit board.

To run WGDB6:

- If you are using Windows 95, click the **Start** button, point to **Programs**, then **ST6 Tools**, then click **WGDB6/Simulator**.
- If you are using Windows 3.x, double-click the appropriate **WGDB6/Simulator** icon in the **ST6 Tools** program group.

Refer to “WGDB6 Debugger for the ST6 Family - Getting Started” in the “ST6 Software Development Tools” manual for full instructions on how to use WGDB6.

7.3 Error Messages

The following table lists the error messages you may encounter when using WGDB6 with the Starter Kit board:

Error message	Description
Error 116 Port B protected when using board.	This means that WGDB6 tried to access the PORT B registers. These are used for communications with the board.
Error 117 Communication error with ST620x/ST621x/ST622x board.	This means that a problem occurred during communications between the host PC and the board. Perform the checks listed below.

7.4 Troubleshooting

If there is a communications problem between WGDB6 and the Starter kit board, the title “WGDB6 Simulator” appears in the WGDB6 title bar. In this case, you should check the following:

- That the Starter Kit board is correctly powered up.
- That the parallel port cable is correctly connected.
- That the device jumpers (W2) are in the USER position.
- That the device type selection jumpers (W1) are in the ST6225 position.
- That an ST6225 is plugged into the Starter Kit board, and it is programmed with DEMOK25.HEX.

8 PROGRAMMING ST6 MICROCONTROLLERS

You can use the Starter Kit board, in conjunction with the program Epromer, to program ST620x, ST621x or ST622x microcontrollers. You can also perform in-circuit programming of ST620x, ST621x or ST622x OTP/EPROM microcontrollers using your own board, connected to the Starter Kit board via the connector J1 (marked 1 on the Starter Kit board diagram on page 9).

8.1 Programming Signals

The following table shows the programming signals and states and their corresponding pin numbers.

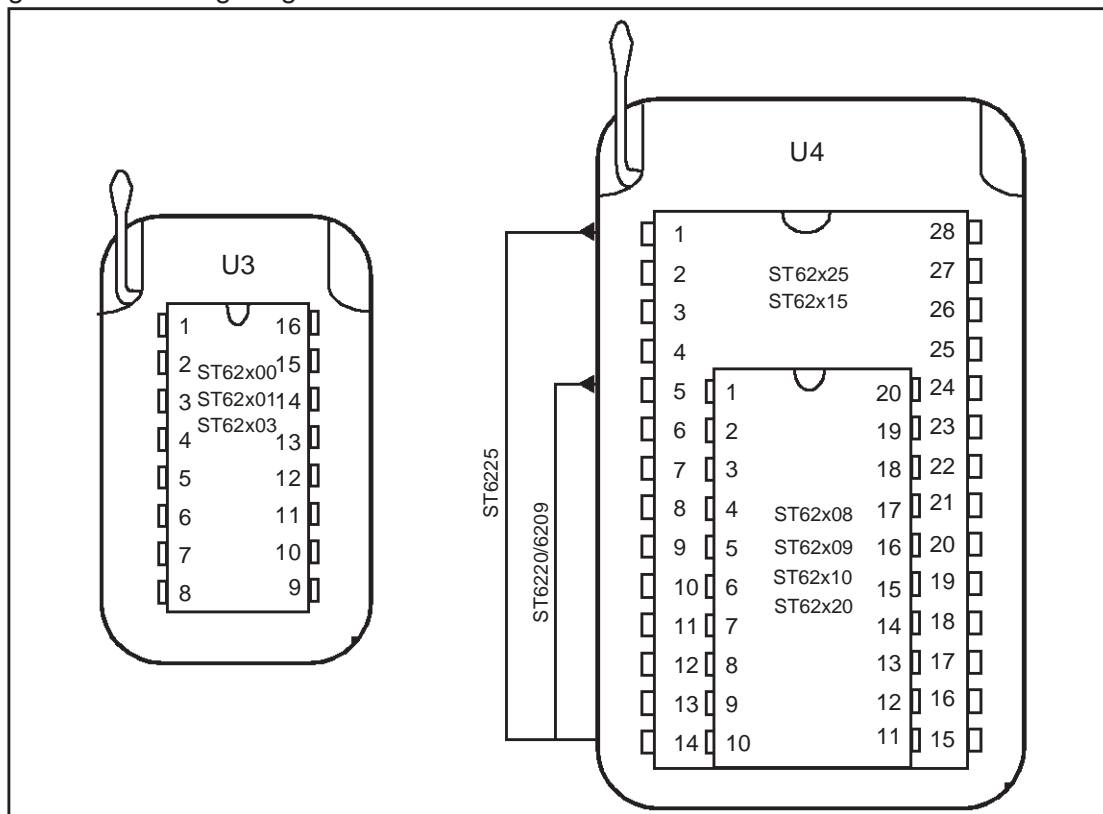
Programming signals	Programming states	MCU DIL28		MCU DIL20		MCU DIL16	
		MCU pin	J2 connector	MCU pin	J2 connector	MCU pin	J2 connector
PB7	High	Pin 12	Pin 12	Pin 8	Pin 12	Pin 7	Pin 12
PB6	High/Low	Pin 13	Pin 13	Pin 9	Pin 13	Pin 8	Pin 13
PB5	High/Low	Pin 14	Pin 14	Pin 10	Pin 14	Pin 9	Pin 14
TEST	5V/12V	Pin 10	NC	Pin 6	Pin 10	Pin 5	Pin 10
OSCIN	High/Low	Pin 3	Pin 3	Pin 14	Pin 7	Pin 2	Pin 7
RESET	High/Low	Pin 11	Pin 11	Pin 16	Pin 11	Pin 6	Pin 11
PB3	High	Pin 16	Pin 16	Pin 12	Pin 16	NC	NC

NOTE: Some of the signals on the J2 connector are used during ST6 programming, thus if you have connected an external resource to J2, you must disconnect it before using the Starter Kit board for programming.

This section describes how to set up the Starter Kit board for programming microcontrollers, and lists the connection requirements for in-circuit application boards.

8.2 Setting Up the Starter Kit Board

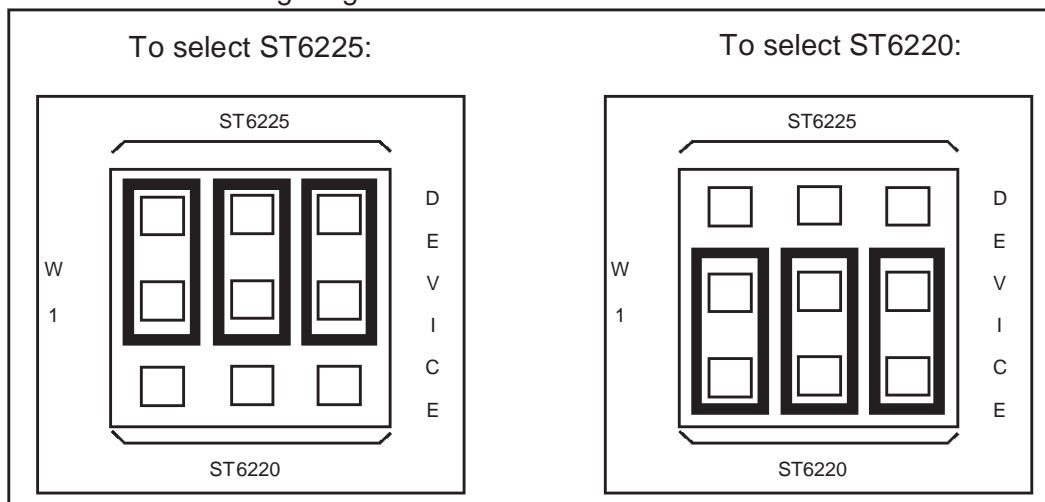
- 1 Power down the Starter Kit board.
- 2 Plug the ST6 microcontroller you want to program into the U3 or U4 ZIF connector, according to the following diagram:



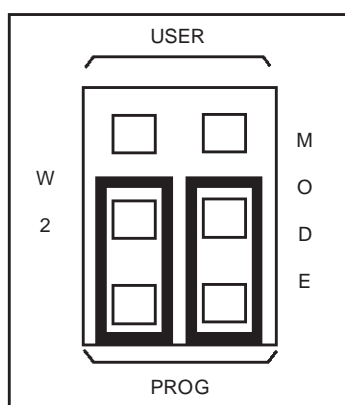
- 3 Set the DEVICE jumpers W1 (marked 18 on the Starter Kit board diagram on page 9) for the microcontroller that you want to program according to the following table:

Device Type	W1 position
ST6200	ST6220
ST6201	ST6220
ST6203	ST6220
ST6208	ST6220
ST6209	ST6220
ST6210	ST6220
ST6220	ST6220
ST6215	ST6225
ST6225	ST6225

As shown in the following diagram:



- 4 Select the PROG mode using the jumpers marked W2 (marked 19 on the Starter Kit board diagram on page 9), as shown in the diagram below:



- 5 Connect the Parallel port P1 on the Starter Kit board to a spare parallel port on your PC using the cable provided with the starter kit.
- 6 Power up the Starter Kit board.

You can now use Epromer to program the microcontroller that is plugged into the Starter Kit board.

NOTE: Epromer does not work under Windows NT.

To run Epromer from Windows 3.x, double-click the Epromer icon in the ST6 Tools group.

To run Epromer from Windows 95, click **Start, Programs, ST6 Tools**, then **Epromer**.

For instructions on how to operate Epromer, click **Help** in the Epromer main window.

8.3 In-Circuit Programming

You can perform in-circuit programming of ST620x, ST621x or ST622x OTP/EPROM microcontrollers using your own board, connected to the Starter Kit board via the connector J1 (marked 1 on the Starter Kit board diagram on page 9).

8.3.1 Application Board Connections

The following paragraphs specify the connection requirements between your application board and the Starter Kit board.

The application board must have a suitable 16-way connector (8x2 header HE10) to be connected via a 16-way cable to connector J1 (marked 1 on the Starter Kit board diagram on page 9) on the Starter Kit board.

The following table shows the required pin connections:

ST620x, ST621x and ST622x Pins	Connector Pins
PB6	Pin 1
PB5	Pin 3
OSCin	Pin 5
PB7	Pin 7
RESET	Pin 9
VPP/TM	Pin 13
VDD	Pin 14, 16
VSS	Pin 2, 4, 6, 8, 10

V_{DD}

Use of the **V_{DD}** connection is optional, depending on whether the application board supply can or cannot be disconnected. If the application board supply is disconnected, you can supply it through pins 14 and 16 of the connector, as long as the total load current does not exceed 100 mA, and the capacitive load is less than 50 µF.

If the application board has its own power supply, its voltage must be set to 5V, so that logic levels are compatible with those of the Starter Kit board.

OSCin

Synchronises the programming operations using a clock generated by the programming tool. OSCin is located on the application board, and must be directly connected to Pin 5 on the 16-way connector. No isolation is needed as long as a quartz crystal or ceramic resonator is used in the application. If an external clock generator is used in the application, it must be disconnected during in-circuit programming.

RESET

Controls the programming mode entry. To prevent signal level contention, RESET must be directly connected to Pin 9 on the 16-way connector, and must be isolated from other nodes on the application board. Any direct connection to V_{DD} , V_{SS} or an output must be avoided. This pin can be connected to a CMOS input, a 2 K Ω pull-up, a 10 K Ω pull-down or left open (Internal pull-up). The capacitive load of the RESET pin should not exceed 1 μ F.

Pins 1 and 7 on the 16-way connector are used to establish communications between the programming tool and the microcontroller.

To prevent signal-level contention, Pins 1 and 7 must be directly connected to PB6 and PB7 on the 16-way connector, and must be isolated from other nodes on the application board. Any direct connection to V_{DD} , V_{SS} or an output must be avoided. These pins may be connected to a CMOS input, a 2 K Ω pull-up, a 10 K Ω pull-down or left open (Internal pull-up).

If **pin 3** on the 16-pin connector is connected to the target device, the same applies. Connection to pin 3 is not necessary if a high voltage level is guaranteed by the board design.

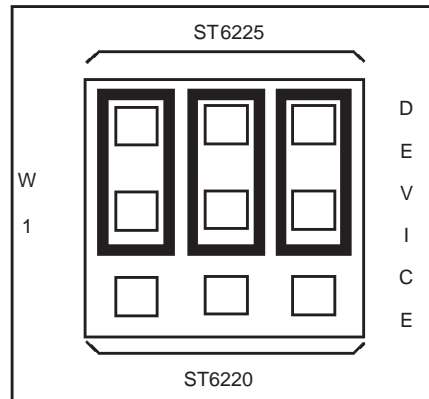
Some **I/O pins** are not connected to the 16-way connector and must be set to a high level during programming. This is normally achieved by the RESET signal sent by the programming tool through the 16-way cable, setting the I/O pins as inputs with an internal 300 K Ω pull-up. To keep these I/O lines high, direct connection of these pins to GND or to any other signal at low level (even temporarily) must be avoided. Only connections to another CMOS input, to an external pull-up or a 10 M Ω pull-down is allowed.

The signals on PB3 and PB5 (if not directly biased through pin 3 of the 16-way connector) must be kept at a high voltage level.

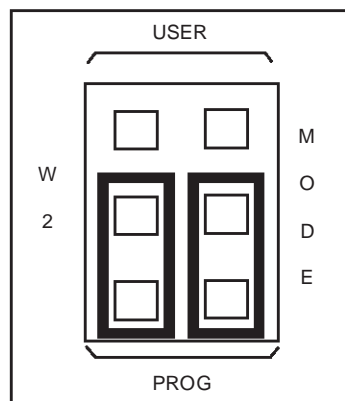
The **Vpp/TM** pin must not be directly connected to GND/ V_{SS} on the application board, to avoid any conflict with the programming voltage provided by the programming tool via pin 13 on the connector. This pin should be pulled down by a resistor with minimum value of 10 K Ω . You must add a 100 nF ceramic capacitor between Vpp/Test and V_{SS} .

8.4 Setting Up the Starter Kit Board for In-Circuit Programming

- 1 Power down the Starter Kit board.
- 2 Set the DEVICE jumpers W1 (marked 18 on the Starter Kit board diagram on page 9) to ST6225, as shown in the following diagram:



- 3 Select the PROG mode using the jumpers marked W2 (marked 19 on the Starter Kit board diagram on page 9), as shown in the diagram below:



- 4 Connect the Parallel port P1 on the Starter Kit board to a spare parallel port on your PC using the cable provided with the starter kit.
- 5 Connect your application board to the connector J1 (marked 1 on the Starter Kit board diagram on page 9) on the Starter Kit board.
- 6 Power up your Starter Kit board.

You can now use Epromer to program the microcontroller that is plugged into the Starter Kit board.

NOTE: Epromer does not work under Windows NT.

To run Epromer from Windows 3.x, double-click the Epromer icon in the ST6 Tools group.

To run Epromer from Windows 95, click **Start, Programs, ST6 Tools**, then **Epromer**.

For instructions on how to operate Epromer, click **Help** in the Epromer main window.

NOTE: If your application board is not powered by the Starter Kit, you must connect it to a 5V DC power supply before you start programming.

9 RUNNING YOUR OWN PROGRAM ON THE STARTER KIT BOARD

You can run your own programs on the Starter Kit board, using any of the Starter Kit resources:

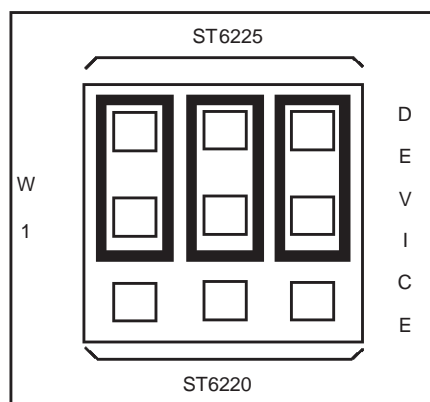
- 8 Mhz oscillator
- 10 Kohm trimmer
- + and - buttons
- Thermistor bridge
- Heater resistor control circuit

NOTE: You can only run applications on the Starter kit board using ST6225 microcontrollers. If your application is designed for another microcontroller, you must change its port definitions to match those of the ST6225.

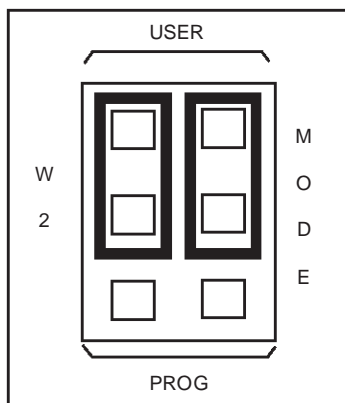
You can also use your own hardware resource by connecting it to the connector J2 (see “Connecting External Resources to the Starter Kit Board” on page 22).

To run your own program on the Starter Kit board:

- 1 Power down the Starter Kit board.
- 2 Program the microcontroller with the application you want to run following the instructions given in “Programming ST6 Microcontrollers” on page 27.
- 3 Make sure that the programmed microcontroller is plugged into the DIL connector, and that the DEVICE jumpers W1 (marked 18 on the Starter Kit board diagram on page 9) are set to ST6225, as shown in the following diagram:



- 4 Select USER mode using the jumpers marked W2 (marked 19 on the Starter Kit board diagram on page 9), as shown in the diagram below:



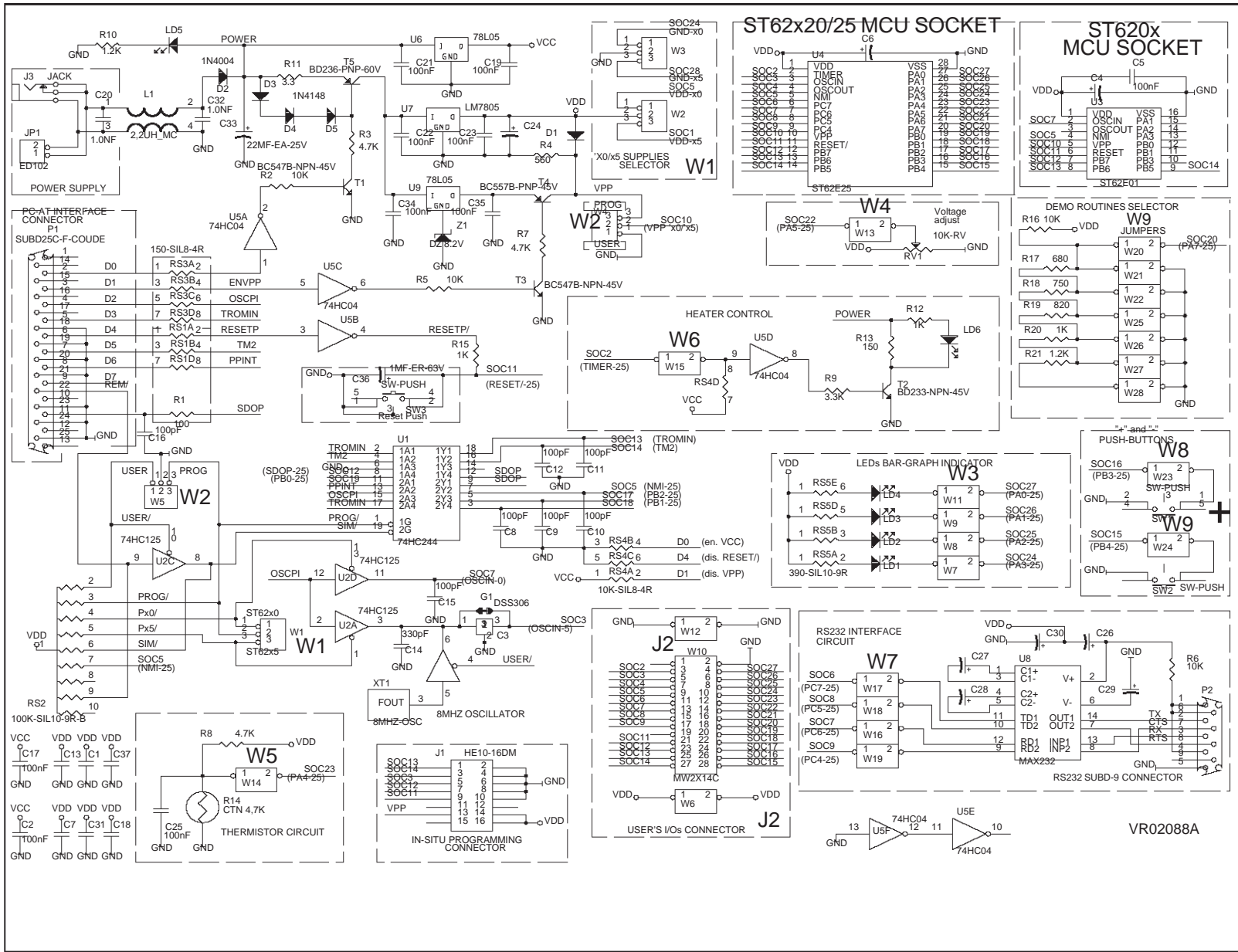
- 5 Disconnect the cable from the parallel port (P1) connection, if it is connected
- 6 Disconnect the demonstration program selector by removing the jumper marked PA7 in the Demonstration Selector circuit marked 12 in the Starter Kit board diagram on page 9.
- 7 If you are using your own hardware resources connected to J2 (marked 17 on the Starter Kit board diagram on page 9), disconnect any Starter Kit board resources that use the same pins, following the instructions given in “Connecting External Resources to the Starter Kit Board” on page 22.
- 8 Power up the Starter Kit board.

10 HARDWARE INFORMATION

10.1 Parts List

Item	Quantity	Reference	Description
1	16	C1,C2,C5,C7,C13,C17,C18,C19,C21,C22,C23,C25,C31,C34,C35,C37	100nF
2	1	C3	DSS306
3	8	C4,C6,C24,C26,C27,C28,C29,C30	100MF-CT-16V
4	7	C8,C9,C10,C11,C12,C15,C16	100pF
5	1	C14	330pF
6	2	C20,C32	1.0NF
7	1	C33	22MF-EA-25V
8	1	C36	1MF-ER-63V
9	1	D1	BYV 10-20 SCHO
10	1	D2	1N4004
11	3	D3,D4,D5	1N4148
12	1	G1	SOLDER BRIDGE
13	1	J1	HE10-16DM
14	1	J3	JACK
15	1	JP1	ED102
16	1	L1	2,2UH_MC
17	4	LD1,LD2,LD3,LD4	LED-RED-RECT
18	2	LD5,LD6	LED-RED-5MM
19	1	P1	SUBD25C-F-ANGLE
20	1	P2	SUBD9C-F-ANGLE
21	3	R3,R7,R8	4.7K
22	1	R4	560R
23	4	R2,R5,R6,R16	10K
24	1	R9	3.3K
25	2	R10,R21	1.2K

10.2 Starter Kit Board Schematic



Notes:

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics 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 SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of SGS-THOMSON Microelectronics.

©1997 SGS-THOMSON Microelectronics - All rights reserved.

Purchase of I²C Components by SGS-THOMSON Microelectronics conveys a license under the Philips I²C Patent. Rights to use these components in an I²C system is granted provided that the system conforms to the I²C Standard Specification as defined by Philips.

SGS-THOMSON Microelectronics Group of Companies

Australia - Brazil - Canada - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands
Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.