SGS-THOMSON
MICROELECTRONICS

ST623x-KIT

STARTER KIT
FOR ST6230 and ST6232 MCUs

HARDWARE FEATURES

- Immediate evaluation of ST6230 and ST6232 devices, with stand-alone demonstration routines.
- Simulation and debugging within the user's real application environment.
- In-socket programming of all OTP and EPROM ST6230 (DIP28) and ST6232 (SDIP42) devices.
- In-circuit programming of all OTP and EPROM ST6230 and ST6232 devices directly on the user's application board (all packages).

SOFTWARE FEATURES

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

Table of Contents

1 INTRODUCTION

The ST623x Starter Kit provides you with all you need to start designing, developing and evaluating programs for ST6230 and ST6232 microcontrollers immediately.

The ST623x 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.
- One ST6230 and one ST6232 microcontroller.
- A complete set of paper documentation and online help.

The demonstration programs, that come pre-loaded on the ST6232 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 voltage control oscillator, three 7-segment displays, 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: \ST623x \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 Getting 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.

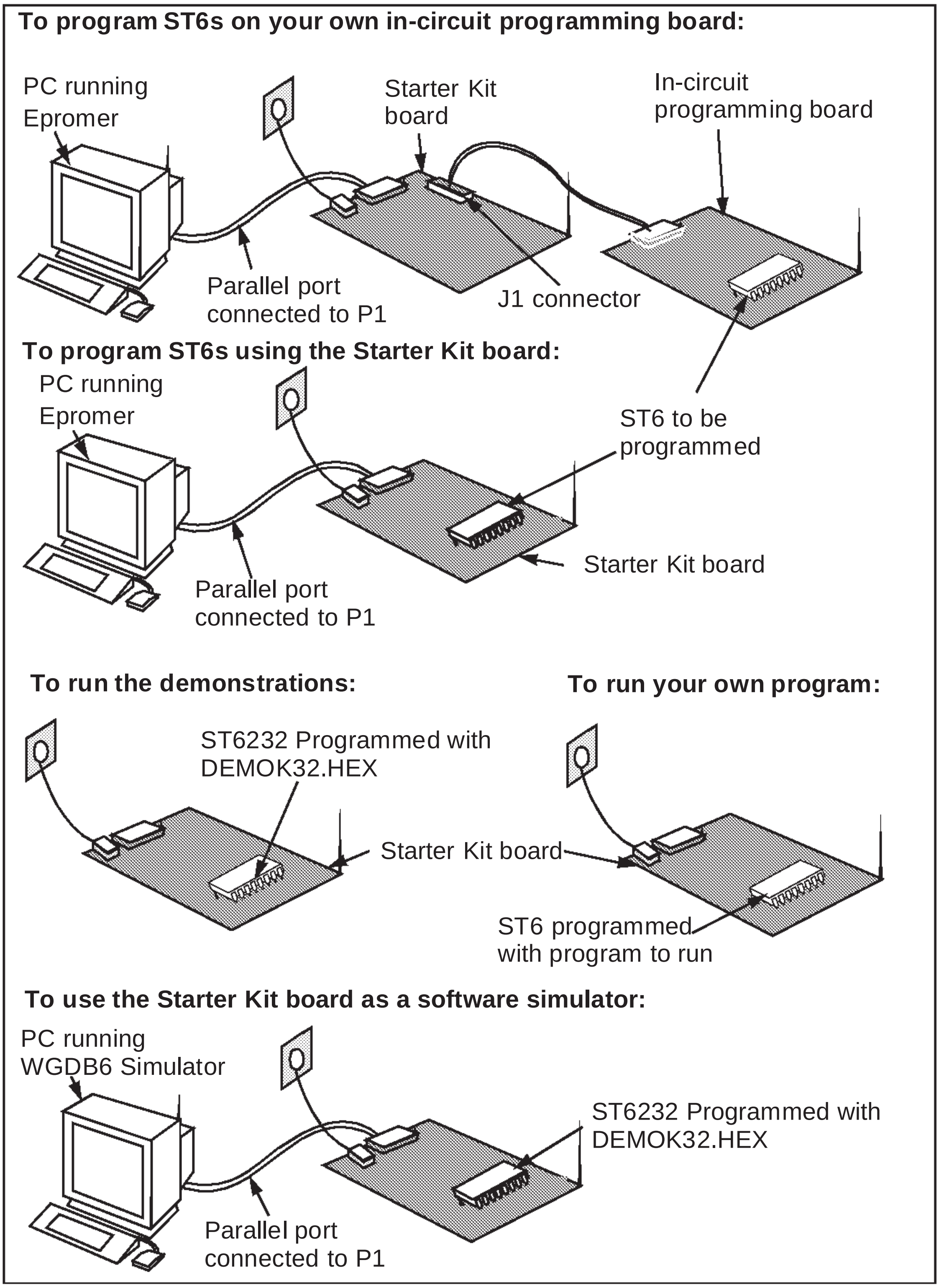
Instructions for use - Warning

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:
PC running Epromer
Starter Kit board
In-circuit programming board
Parallel port connected to P1
J1 connector
To program ST6s using the Starter Kit board:
PC running Epromer
ST6 to be programmed
Starter Kit board
Parallel port connected to P1
To run the demonstrations:
To run your own program:
ST6232 Programmed with DEMOK32.HEX
Starter Kit board
ST6 programmed with program to run
To use the Starter Kit board as a software simulator:
PC running WGDB6 Simulator
ST6232 Programmed with DEMOK32.HEX
Parallel port connected to P1

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 16 of this book.
Install and run the ST6-Realizer demonstration software.	"Running the ST6-Realizer Demo" on page 22 of this book.
Find out what the demonstration applications do, and run them.	"Running the Demos" on page 17 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 25 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 29 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 23 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 33.

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 35.

2.1 The ST6 Microcontrollers

The Starter Kit includes the following microcontrollers:

- One ST6230, One ST6232

The ST6232 microcontroller is pre-loaded with the code **DEMOK32.HEX**, which includes the demonstration programs (see “Running the Demos” on page 17), 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 25). The other ST6 microcontrollers are blank.

The file Demok32.hex is provided on the diskette labelled “ST623x Starter Kit“, so that if you erase it from the ST6232, you can re-program it following the instructions given in “Programming ST6 Microcontrollers” on page 29.

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.
- Three 7-segment displays.
- A voltage control oscillator.

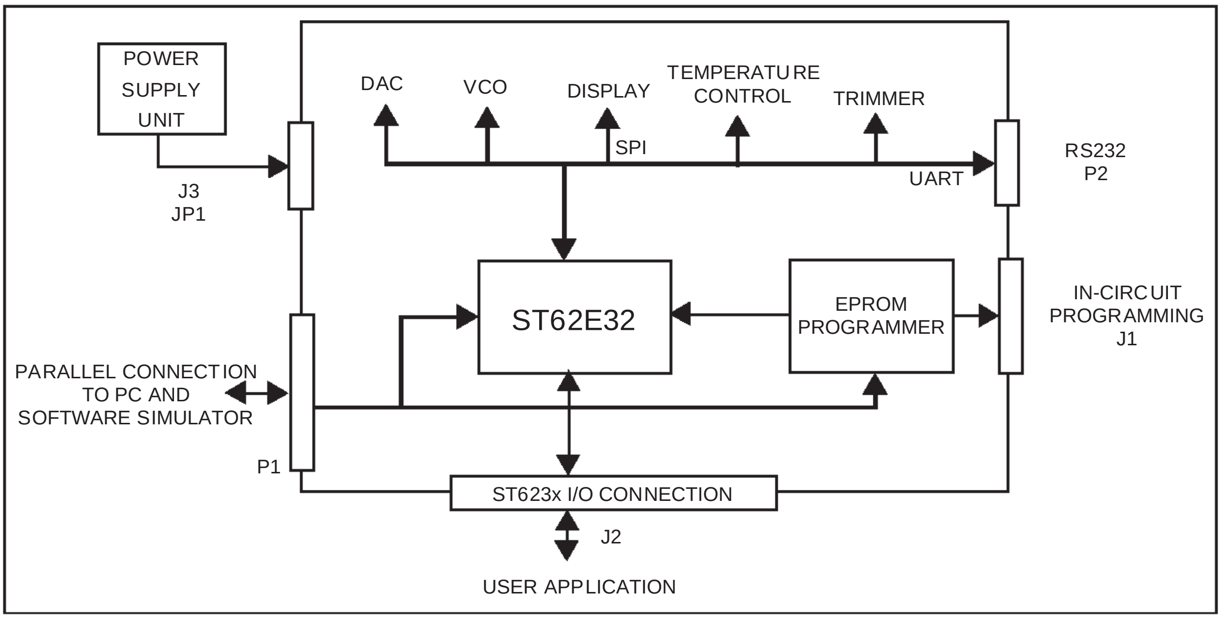
It comes with its own power supply unit that can be plugged into an AC mains source, or a DC source with the following characteristics:

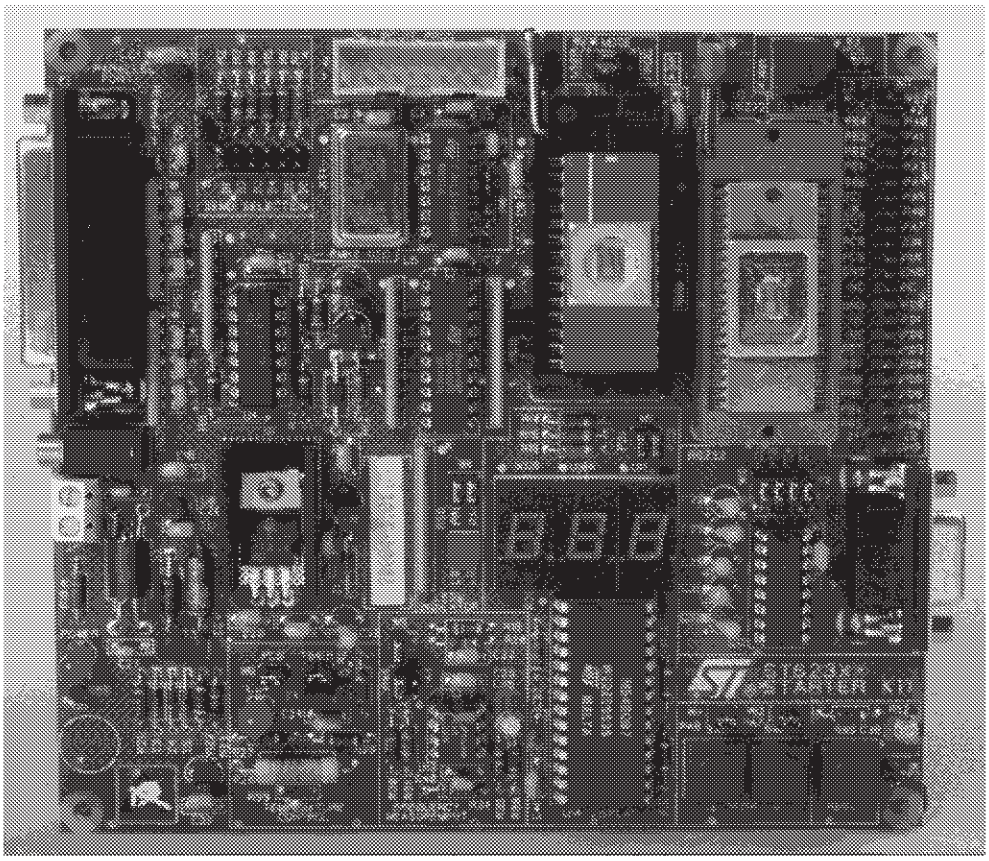
- Voltage: 16V min./20V max., Current: 100 mA min.

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.

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).

2 Demonstration routine selector.

3 8 Mhz oscillator.

4 PC connector P1.

5 LEDs.

6 Voltage control oscillator.

7 Power supply JACK connector J3.

8 Heater resistor LED indicator LD4.

9 Power supply LED indicator LD5.

10 Heater resistor.

11 Thermistor.

12 RESET button.

13 “+” and “-” buttons.

14 RS232 interface circuit and connector.

15 7-segment displays.

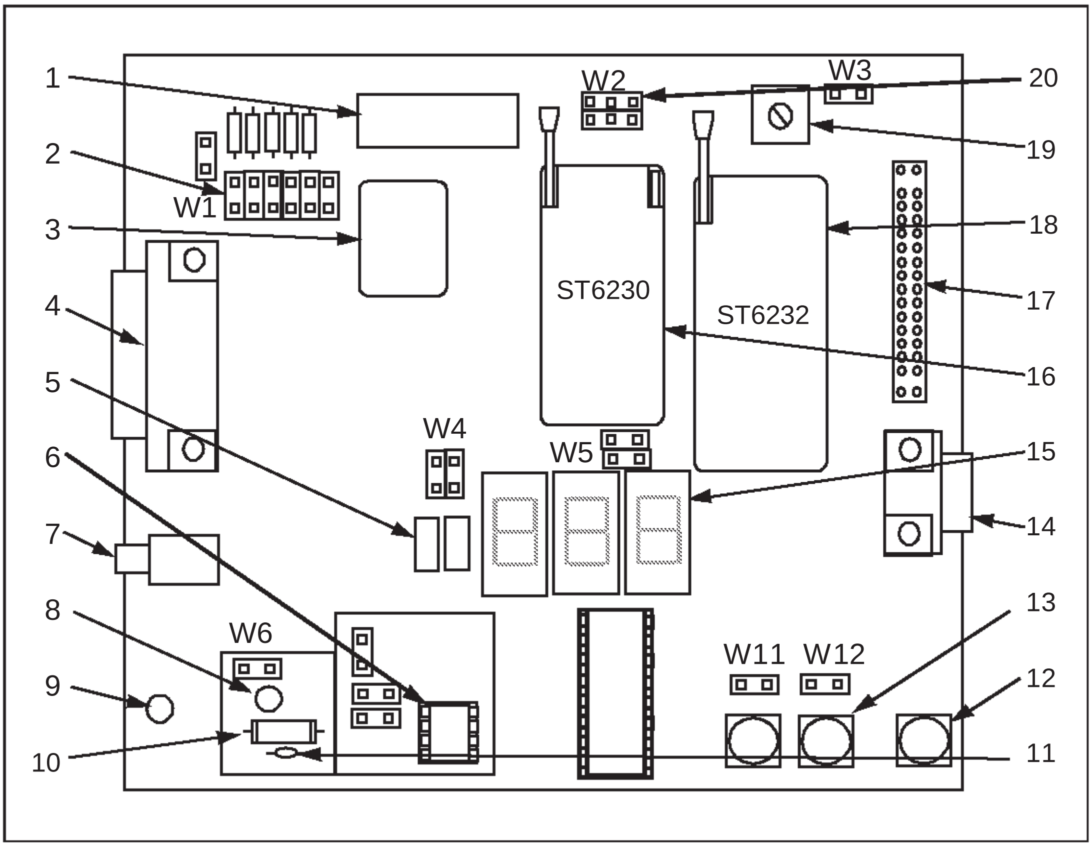
16 DIP 28 ZIF MCU socket.

17 Remote resource I/O interface connector J2.

18 SDIP 42 ZIF MCU socket.

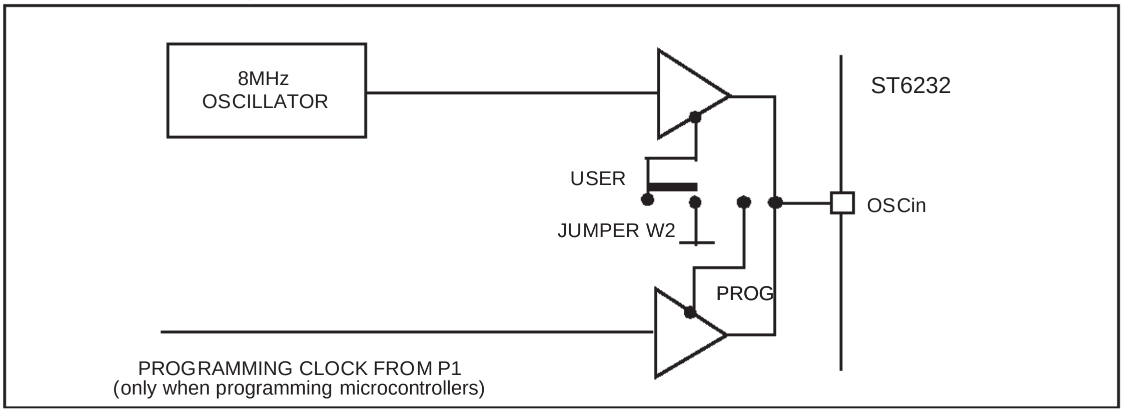
19 10 KΩ trimmer.

20 “Programming” or “User” operating mode selection jumpers W2.



2.3 Oscillator

An oscillator feeds the ST6232 OSCIN input with an 8 MHz clock signal. You can disconnect the oscillator by removing the two jumpers W2 (marked 20 on the Starter Kit board diagram on page 9).


8MHz
OSCILLATOR
ST6232
USER
JUMPER W2
OSCin
PROG
PROGRAMMING CLOCK FROM P1
(only when programming microcontrollers)

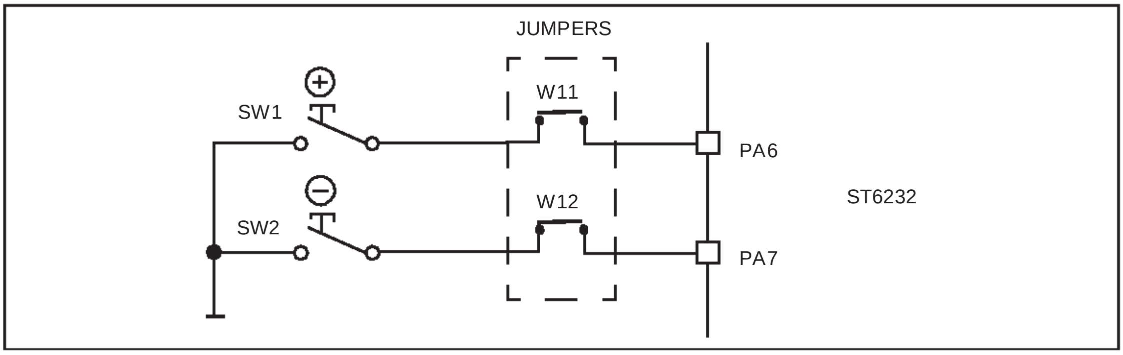
2.4 Reset Button

This activates the ST6232 RESET input when pressed. A power-on reset circuit is also provided.

2.5 + And - Buttons

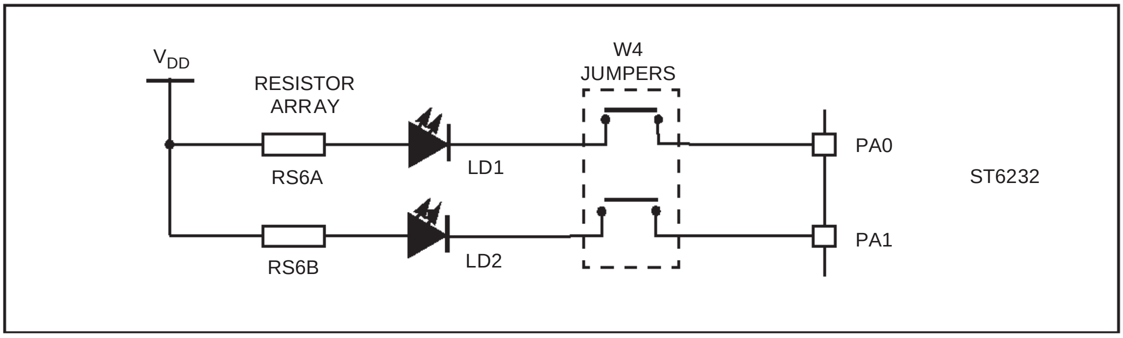
These are connected to PA6 and PA7 pins on the ST6232 respectively (when programmed as inputs). They drive the PA6 and PA7 inputs down to GND when pressed. They increment the three 7-segment displays in Demo 1 (see “Demo 1 - Button/LED Indicator Control” on page 17).

You can disconnect PA6 by removing the W11 jumper, and PA7 by removing the W12 jumper.



2.6 LED Indicators

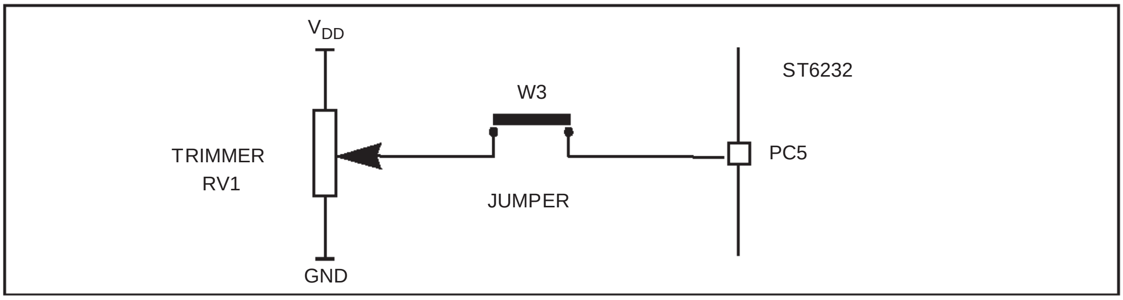
Two LEDs are used for level indication in the demonstration routines. They are connected to the ST6232 PA0 and PA1 pins (when programmed as outputs). They are used during the RESET operation. The 2 LEDs can be disconnected by removing the W4-PA0 and 1 jumpers.



2.7 Resistance trimmer

A 10 KΩ resistance trimmer feeds the ST6232 PC5 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. This process id demonstrated by Demo 2 (see “Demo 2 - Voltage trimming and LED level indication” on page 17).

The trimmer can be disconnected from the I/O pin by removing the W3-PC5 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 ST6232 microcontroller. The heat level can be controlled by varying the duty cycle of a PWM signal present on the TIMER output.

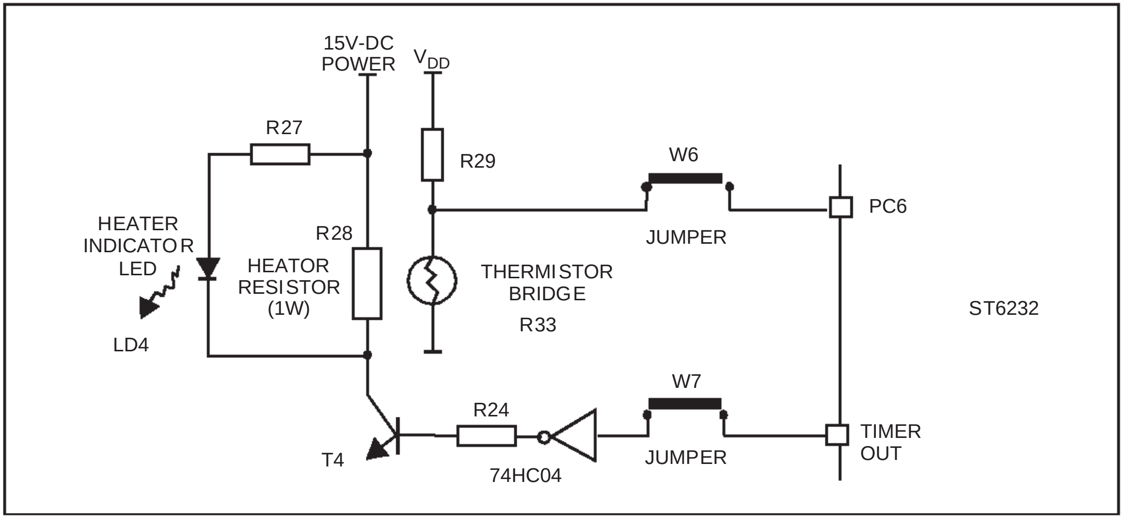
The Heater circuit can be disconnected from the TIMER output pin by removing the W7-TIMER jumper.

A thermistor bridge (Negative Temperature Coefficient) is connected to the PC6 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 PC6 by removing the W6 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 “Demo 3 - Temperature Control” on page 18).



2.9 RS-232 Interface

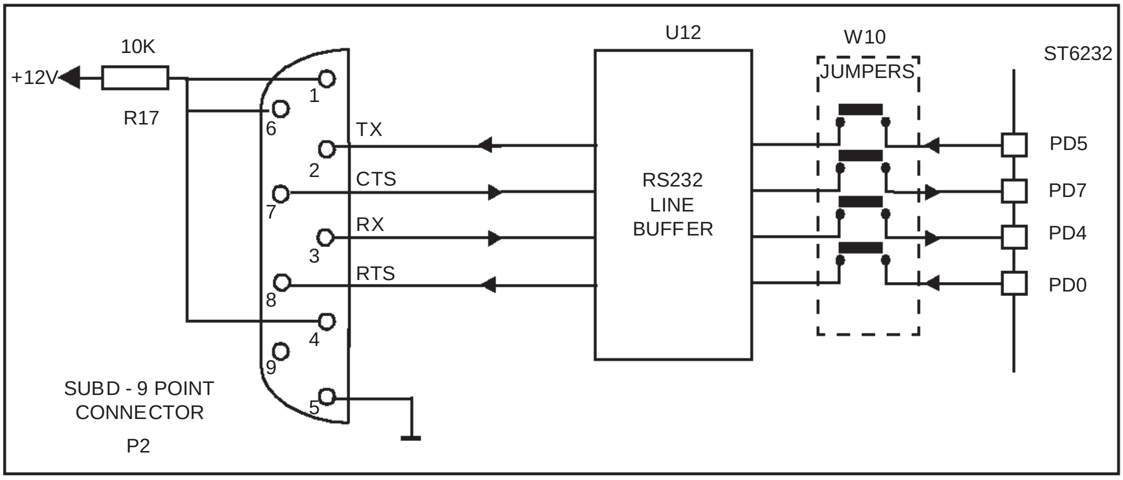
The RS-232 interface enables you to communicate with the pre-programmed ST6232 microcontroller provided with the Starter Kit. It includes an RS-232 buffer circuit that is connected to a standard PC-compatible RS-232 SUBD-9 connector.

The following table lists the RS-232/ST6232 pin connections:

Signal Name	SUBD-9 Pin	ST6 Pin
Data Transmission (TX)	2	PD5
Data Reception (RX)	3	PD4
Request to Send (RTS)	8	PD0
Clear to Send (CTS)	7	PD7

You can disconnect these by removing the corresponding jumpers from W10.

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

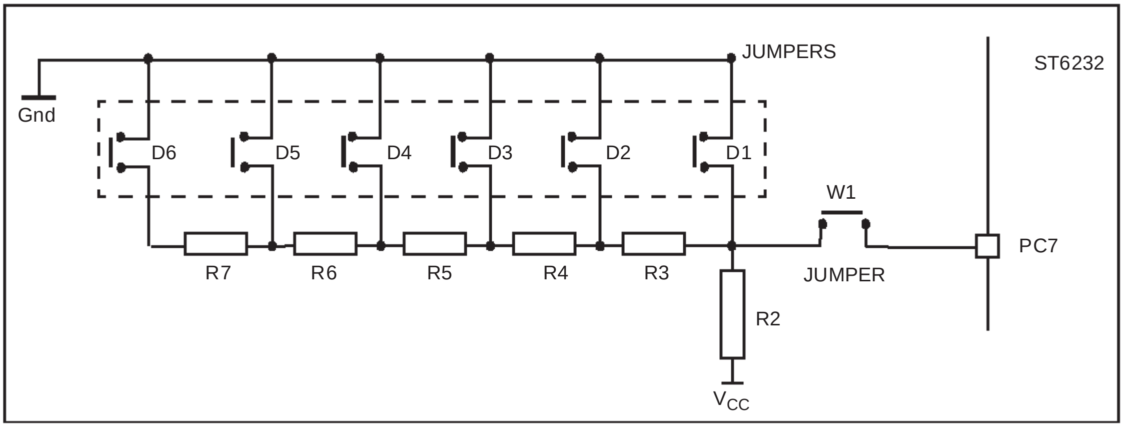
The demonstration selector is made up of a 6-position jumper W1 (marked 2 on the Starter Kit board diagram on page 9), with each jumper connected to a resistor. Each resistor generates a different voltage.

After reset, the voltage value generated by the resistor whose jumper is installed is sent to PC7 on the ST6232. PC7 is programmed as an A/D converter. The program installed on the ST6232 uses the input from PC7 to select the appropriate demo. The following table lists the voltage values generated by each resistor:

Resistor value:	Theoretical voltage value:
R2: 10 KΩ	No JUMPER: 5 V
R3: 680 Ω	JUMPER-D1: 0 V
R4: 750 Ω	JUMPER-D2: 333 mV
R5: 820 Ω	JUMPER-D3: 666 mV
R6: 1 KΩ	JUMPER-D4: 999 mV
R7: 1.2 KΩ	JUMPER-D5: 1.332 V
	JUMPER-D6: 1.539 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 PC7 by removing the PC7 jumper from W1 (marked 2 on the Starter Kit board diagram on page 9).



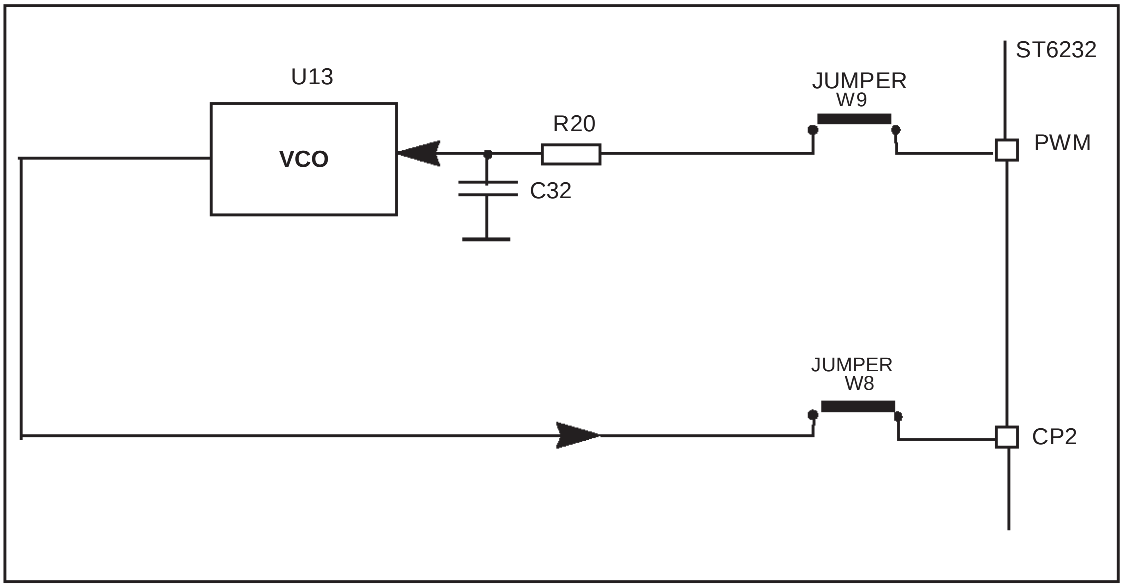
2.11 Frequency Measurement

This application simulates the servo-control of a motor's speed using a regulation loop.

The 16-bit Autoreload Timer output PWM generates a frequency of about 15Khz. You can increase or decrease the duty cycle of this frequency using the "+" and "-" buttons. This frequency is then filtered to produce a proportional DC voltage of between 0 and 5 Volts, which attacks the voltage control oscillator (marked 6 on the Starter Kit board diagram on page 9). The voltage control oscillator produces a frequency at its output in the interval of 50Hz-1Khz. The 16-bit Autoreload Timer measures this frequency through its input port CP2 with a frequency precision of 1/100. The result is displayed on the three 7-segment displays.

Demo 5 demonstrated this process (see "Demo 5 - Frequency Measurement" on page 19).

The Frequency Measurement circuit can be disconnected from the CP2 input and the PWM output by removing the corresponding W8 and W9 jumpers.



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-Windows^{TM} 3.11, NT or 95.

3.2 Connecting the Power Supply

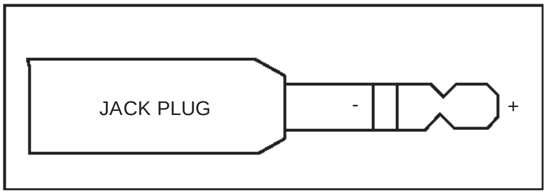
If you have AC mains supply, connect the Jack plug on the power supply cable provided to the J3 input socket, then connect the mains plug to a mains source.

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 19 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:\ST623X\SK623XLI\DEMOK32.ASM**.

4.1.1 Demo 1 - Button/LED Indicator Control

1 Initialises the pins as follows:

This pin:	Is initialised as:
PA6	Input with pull-up and interrupt. Connected to + button.
PA7	Input with pull-up and interrupt. Connected to - button.
PD1 and PD3	Outputs, connected to the LED display driver for the 7-segment displays.

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 adds one to the number displayed on the 7-segment displays.
Press the - button	Jumps to an interrupt subroutine that subtracts one to the number displayed on the 7-segment displays.

4.1.2 Demo 2 - Voltage trimming and LED level indication

1 Initialises the pins as follows:

This pin:	Is initialised as:
PC5	Analog input. Connected to the trimmer RV1.
PD1 and PD3	Outputs, connected to the LED display driver for the 7-segment displays.

2 Reads the A/D converter data register, and displays the voltage on the 7-segment displays.

Adjusting the voltage trimmer (marked 19 on the Starter Kit board diagram on page 9) displays the appropriate voltage output by the trimmer on the 7-segment displays. 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:
PC6	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.
PD1 and PD3	Outputs, connected to the LED display driver for the 7-segment displays.

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, and displays the difference between the initial temperature and the last-read temperature in °C.

You can either increase the temperature by touching the thermistor (marked 11 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_32** (this is supplied on the diskette marked “ST623x Starter Kit “).

3 Follow the instructions as they appear on screen.

4.1.5 Demo 5 - Frequency Measurement

1 Initialises the pins as follows:

This pin:	Is initialised as:
PWM	Connected to the input of the Voltage Control Oscillator through an R-C filter) as output (ART16 peripheral). The PWM output is initialised with a frequency of about 15Khz and a duty cycle of 50%.
CP2	Connected to the input of the Voltage Control Oscillator through an R-C filter) as input (ART16 peripheral).
PA6	Input with pull-up and interrupt. Connected to + button.
PA7	Input with pull-up and interrupt. Connected to - button.
PD1 and PD3	Outputs, connected to the LED display driver for the 7-segment displays.

2 Reads the frequency at the output of the voltage control oscillator using the CP2 inputs and displays the frequency value on the three 7-segment displays.

You can adjust the frequency displayed (between 50Hz and 1Khz) by pressing the “+” and “-” buttons. This adjusts the duty cycle in steps of 15Khz.

4.1.6 Demo 6 - 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 “ST623x Starter Kit “).

4.2 Running the Demonstration Programs

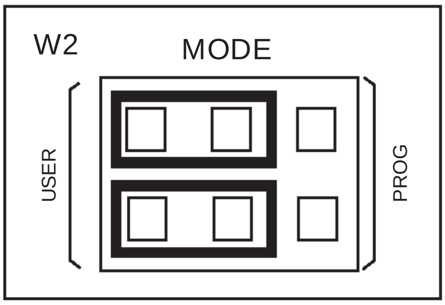
The ST6232 microcontroller labelled DEMOK32 is programmed with the demonstration software. If this software has been erased from the microcontroller, you can reprogram it from the file **DEMOK32.HEX** (this is supplied on the diskette marked “ST623x Starter Kit“). For details of how to program microcontrollers refer to “Programming ST6 Microcontrollers” on page 29.

To run the demonstrations:

1 Power down the Starter Kit board.

2 Make sure that the pre-programmed ST6232 is plugged into the SDIP 42 ZIF MCU socket.

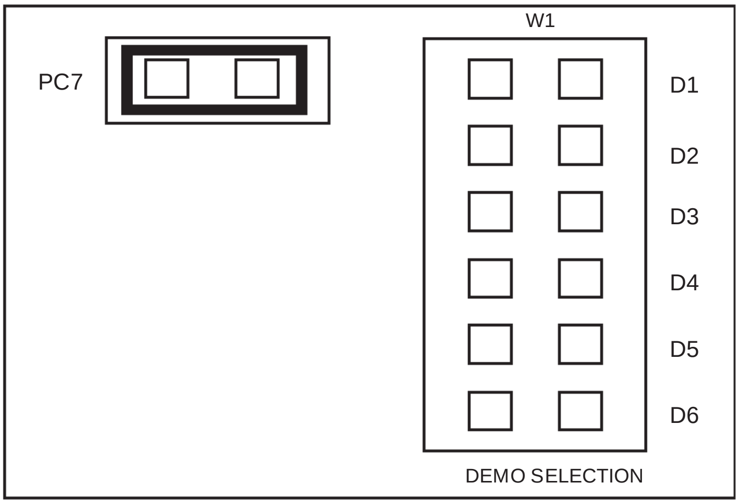
3 Select the USER mode using the jumpers marked W2 (marked 20 on the Starter Kit board diagram on page 9), as shown in the diagram below:



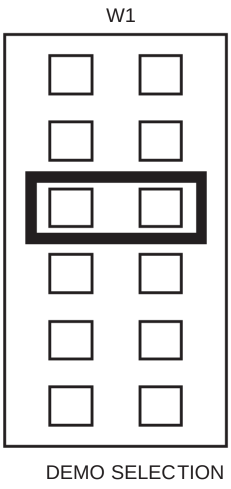
4 Disconnect the cable from the parallel port (P1) connection, if it is connected.

5 Power up the Starter Kit board.

6 Install the demonstration program jumper marked PC7, as shown in the diagram below:



7 Select the demo you want to run, by installing the appropriate jumper on W1 (marked 2 on the Starter Kit board diagram on page 9), as indicated on the diagram below:



D1 Selects demo 1 - Push Button/LED Indicator Control

D2 Selects demo 2 - Voltage trimming and LED level indication

D3 Selects demo 3 - Temperature Control

D4 Selects demo 4 - RS-232 Communications

D5 Selects demo 5 -Frequency Measurement

D6 Selects demo 6 - Fuzzy Logic Heater Control

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 ST6232 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 ST6232, 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	ST6232 I/O	PIN	PIN	ST6232 I/O	JUMPER	On-board resource
none	-	PE4	1	42	PA0	W4-	LED Indicator
none	-	PE3	2	41	PA1	W4-	LED Indicator
System tasks (Simulator link)	(*)	PE2	3	40	PA2	-	none
System tasks (Simulator link)	(*)	PE1	4	39	PA3	W9	ART16-PWM
System tasks (Simulator link)	(*)	PE0	5	38	PA4	-	none
8Mhz OSC.	W2	OSCIN	6	37	PA5	W8	ART16-CP2
none	-	OSCO	7	36	PA6	W11	Push-but.(+)
Routine selector	W1	PC7	8	35	PA7	W12	Push-but.(-)
Thermistor	W6	PC6	9	34	TIMER	W7	Heater resis. control
Trimmer	W3	PC5	10	33	NMI	(*)	System task (Simul. link)
GND	-	VSS	11	32	AVSS	-	GND
+5V Supply	-	VDD	12	31	AVDD	-	+5V Supply
GND	-	VSS	13	30	PD0	W10	RS232 Driver RTS Input
+5V Supply	-	VDD	14	29	PD1	W5	SPI-Scl
System tasks (Programming)	W2	VPP	15	28	PD2	-	none
Push-button Power-on	-	/RESET	16	27	PD3	W5	SPI-Sout
none	-	PB7	17	26	PD4	W10	RS232 Driver RX Input
System tasks (Programming)	W2	PB6	18	25	PD5	W10	RS232 Driver TX Input
System tasks (Programming)	W2	PB5	19	24	PD6	W2	System tasks (Programming)

On-board resource	JUMPER	ST6232 I/O	PIN	PIN	ST6232 I/O	JUMPER	On-board resource
System tasks (Programming)	W2	PB4	20	23	PD7	W10	RS232 Driver CTS Input
none	-	PB3	21	22	PB0	-	none

(*) 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 ST6230 or ST6232 microcontroller. Note, however that you must use the pre-programmed ST6232 microcontroller, labelled DEMOK32 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 ST6230 or ST6232 registers via the host PC's parallel port. The DEMOK32.HEX program, with which the ST6232 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	ST6232	SIMULATOR USAGE
D2	PE2	Synchronisation
D3	PE1	Write data to MCU
D4	RESET	Hardware reset of peripherals
D6	NMI	Initiates data transfer
SDOP	PE0	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 ST6232 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 ST6232 microcontroller are not supported by the WGDB6 simulator.
- The pins: NMI, PE0, PE1 and PE2 on the ST6232 microcontroller are used for communications with the host PC, and are thus not available for simulation.

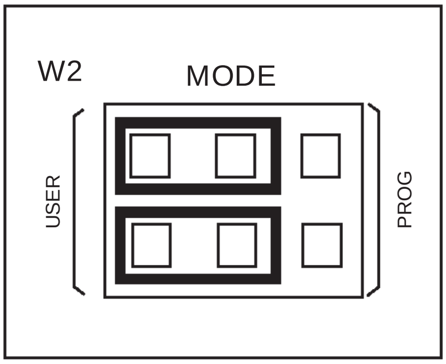
NOTE: Do not connect any external resources to the corresponding J2 connector pins when using the Starter Kit board as a peripheral emulator.

To use the Starter Kit board as a hardware simulator:

1 Power down the Starter Kit board.

2 Make sure that the pre-programmed ST6232 is plugged into the SDIP 42 ZIF MCU socket.

3 Select the USER mode using the jumpers marked W2 (marked 20 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 E protected when using board.	This means that WGDB6 tried to access the PORT E registers. These are used for communications with the board.
Error 117 Communication error with ST6230/ST6232 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 (W1) are in the USER position.
- That an ST6232 is plugged into the Starter Kit board, and it is programmed with DEMOK32.HEX.

8 PROGRAMMING ST6 MICROCONTROLLERS

You can use the Starter Kit board, in conjunction with the program Epromer, to program ST6230 or ST6232 microcontrollers. You can also perform in-circuit programming of ST6230 or ST6232 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).

NOTE: The PB4, PB5, PB6, PD2, OSCIN, OSCOUT and RESET pins are used to perform programming operations. If these pins are connected to an external resource via J2, you must either disconnect them (see “Connecting External Resources to the Starter Kit Board” on page 23) or set them to high impedance state during programming operations.

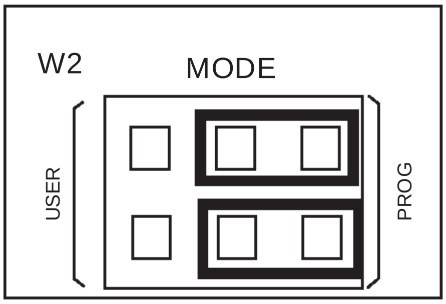
This section describes how to set up the Starter Kit board for programming microcontrollers, and lists the connection requirements for in-circuit application boards.

8.1 Setting Up the Starter Kit Board

1 Power down the Starter Kit board.

2 Plug the ST6 microcontroller you want to program into the appropriate MCU socket on the Starter Kit board.

3 Select the PROG mode using the jumpers marked W2 (marked 20 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.

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.

8.2 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).

8.2.1 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:

ST6230 and ST6232	Connector Pins
PB5	Pin 1
PB4	Pin 3
OSCin	Pin 5
PB6	Pin 7
RESET	Pin 9
PD6	Pin 11
VPP/TM	Pin 13
VDD	Pin 14, 16
VSS	Pin 2, 4, 8

V_{DD}

Use of the $\mathbf{V_{DD}}$ 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 an 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 KΩ pull-up, a 10 KOhm pull-down or left open (Internal pull-up). The capacitive load of the RESET pin should not exceed 1 µF.

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 PB5 and PB6 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 KΩ pull-up, a 10 KOhm 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.

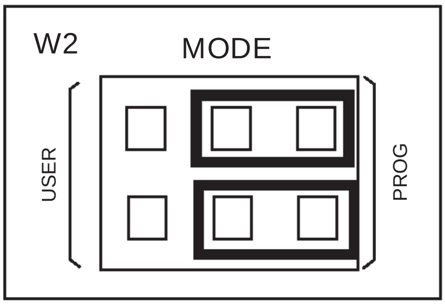
Some **I/O pins** are not connected to the 16-way connector and must be set to a high level during programming. This is normally achieved by the RESET signal sent by the programming tool through the 16-way cable, setting the I/O pins as inputs with an internal 300 KΩ pull-up. To keep these I/O lines 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 MΩ pull-down is allowed.

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 KΩ. You must add a 100 nF ceramic capacitor between Vpp/Test and $V_{SS.}$

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

1 Power down the Starter Kit board.

2 Select the PROG mode using the jumpers marked W2 (marked 20 on the Starter Kit board diagram on page 9), as shown in the diagram below:



3 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.

4 Connect your application board to the connector J1 (marked 1 on the Starter Kit board diagram on page 9) on the Starter Kit board.

5 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:

- The ST6230 or ST6232 microcontroller.
- Reset and data control buttons.
- LED indicators.
- Resistance trimmer.
- Temperature control circuit.
- RS-232 interface.
- Demonstration program selector jumpers.
- Three 7-segment displays.
- A voltage control oscillator.

NOTE: You can only run applications on the Starter kit board using ST6232 microcontrollers. If your application is designed for another microcontroller, you must change its port definitions to match those of the ST6232.

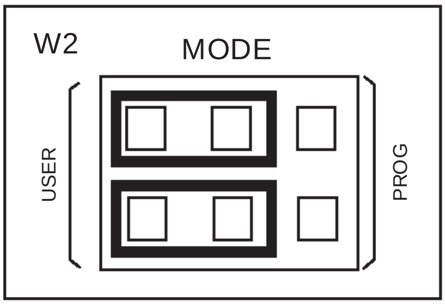
You can also use your own hardware resource by connecting it to the connector J2 (see “Connecting External Resources to the Starter Kit Board” on page 23).

To run your own program on the Starter Kit board:

1 Power down the Starter Kit board.

2 Program an ST6232 microcontroller with the application you want to run following the instructions given in “Programming ST6 Microcontrollers” on page 29, and plug it into the SIDP42 ZIF MCU socket.

3 Select the USER mode using the jumpers marked W2 (marked 20 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 Disconnect the demonstration program selector by removing the jumper marked PC7 in the Demonstration Selector circuit marked 2 in the Starter Kit board diagram on page 9.

6 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 23.

7 Power up the Starter Kit board.

10 HARDWARE INFORMATION

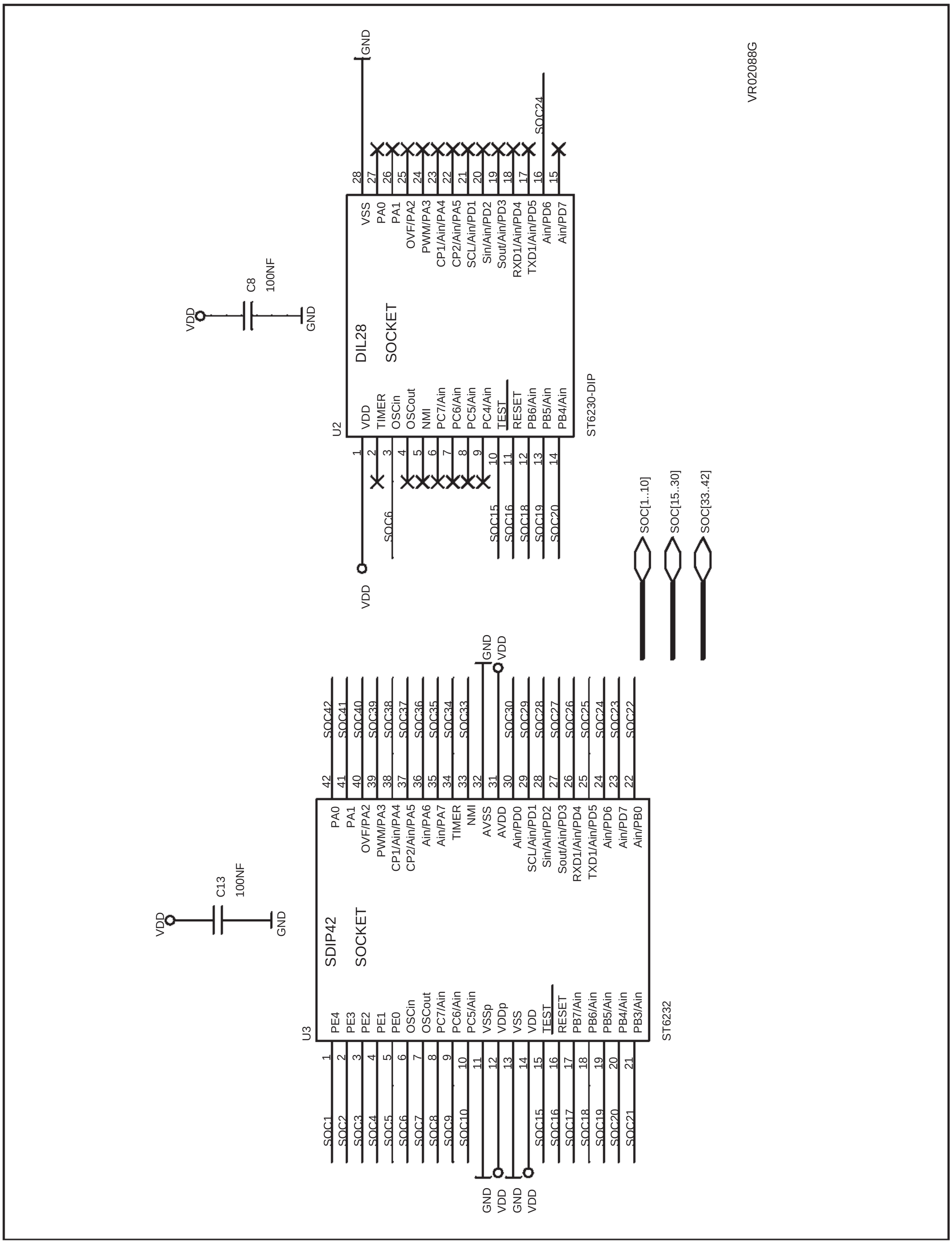
10.1 Parts List

Item	Quantity	Reference	Description
1	20	C1,C2,C3,C4,C8,C11,C12,C13,C18,C19,C20,C21,C26,C28,C29,C32,C34, C35,C39,C41	100nF
2	1	C40	47nF
3	6	C22,C23,C25,C30,C31,C33	10MF-CT-16V
4	8	C5,C6,C7,C9,C10,C14,C15,C16	100pF
5	1	C14	330pF
6	3	C17,C27,C37	1.0NF
7	1	C24	22MF-EA-25V
8	2	C36,C38	1MF-ER-63V
9	1	D1	BYV 10-20 SCHO
10	1	D2	1N4004
11	3	D3,D4,D5	1N4148
12	1	J1	HE10-16DM
13	1	J3	JACK
14	1	JP1	ED102
15	1	L1	2,2UH_MC
16	2	LD2,LD3	LED-RED-RECT
17	2	LD1,LD4	LED-RED-5MM
18	1	P1	SUBD25C-F-ANGLE
19	1	P2	SUBD9C-F-ANGLE
20	2	RS1,RS3	150-SIL8-4R
21	1	RS5	100K-SIL10-9R-B
22	1	RS2	1K-SIL8-4R
23	1	RS4	10K-SIL10-9R-B
24	1	RS6	390-SIL10-9R
25	1	RV1	10K-RV
26	2	R7,R18	1.2K
27	1	R21	3.3R
28	6	R10,R19,R22,R26,R29,R37	4.7K
29	1	R13	560R
30	5	R6,R11,R12,R27,R32	1K

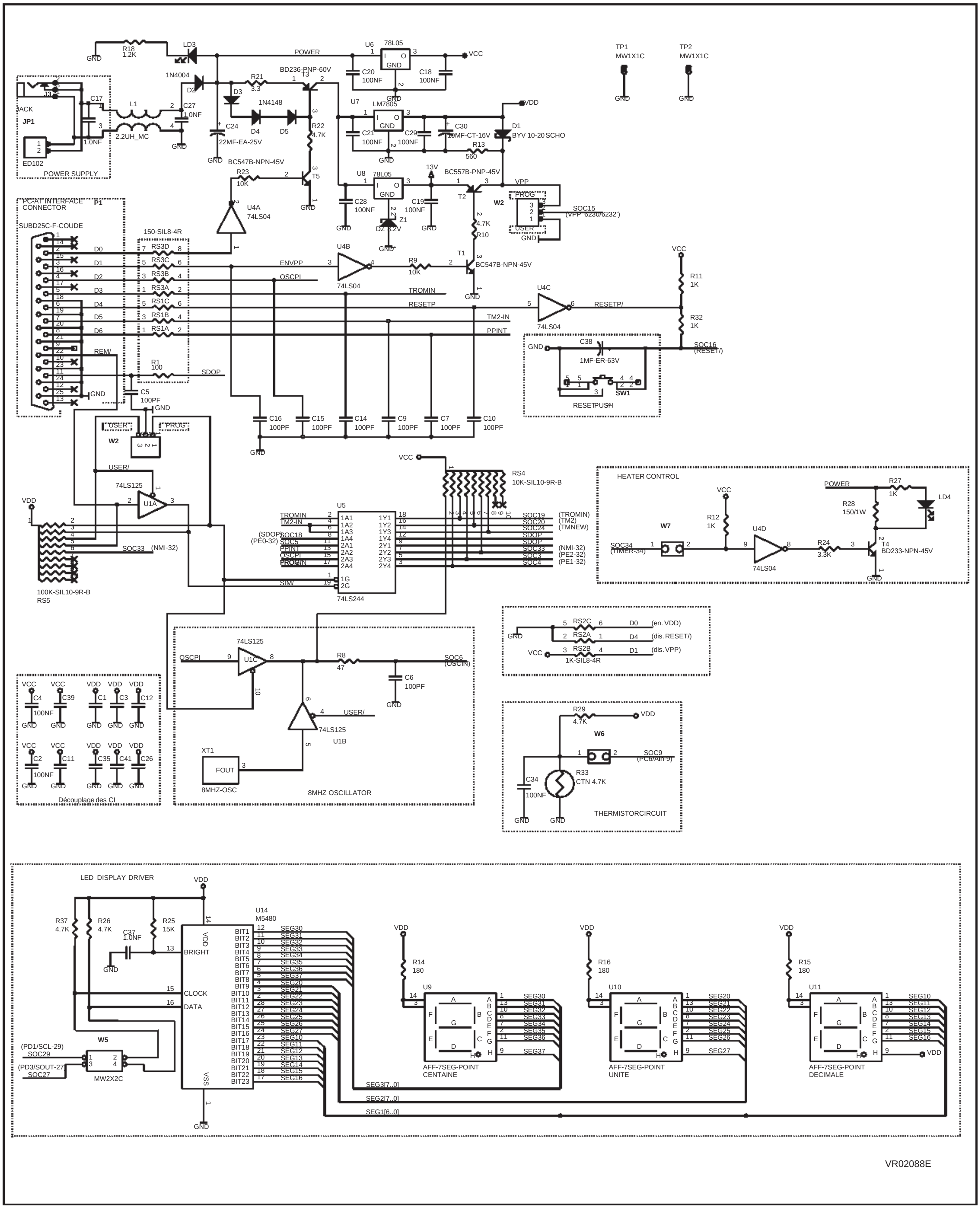
PARTS LIST (Cont'd)

Item	Quantity	Reference	Description
31	5	R2,R9,R17,R23,R30	10K
32	1	R1	100R
33	2	R20,R35	100K
34	1	R34	6.8K
35	1	R31	18K
36	2	R8,R36	47R
37	1	R28	150R/1W
38	1	R33	CTN 4.7K
39	1	R24	3.3K
40	1	R25	15K
41	3	R14,R15,R16	180R
42	1	R3	680
43	1	R4	750R
44	1	R5	820R
45	3	SW1,SW2,SW3	PUSH-BUTTON
46	1	T3	BD236-PNP-60V
47	2	T1,T5	BC547B-NPN-45V
48	1	T2	BC557B-PNP-45V
49	1	T4	BD233-NPN-45V
50	2	U6,U8	78L05
51	1	U7	LM7805
52	1	U4	74LS04
53	1	U13	LM331
54	1	U1	74LS125
55	1	U5	74LS244
56	1	U12	MAX232
57	1	U14	M5480
58	3	U9,U10,U11	AFF-7SEG-POINT
59	1	XT1	8MHZ-OSC
60	1	Z1	DZ 8.2V

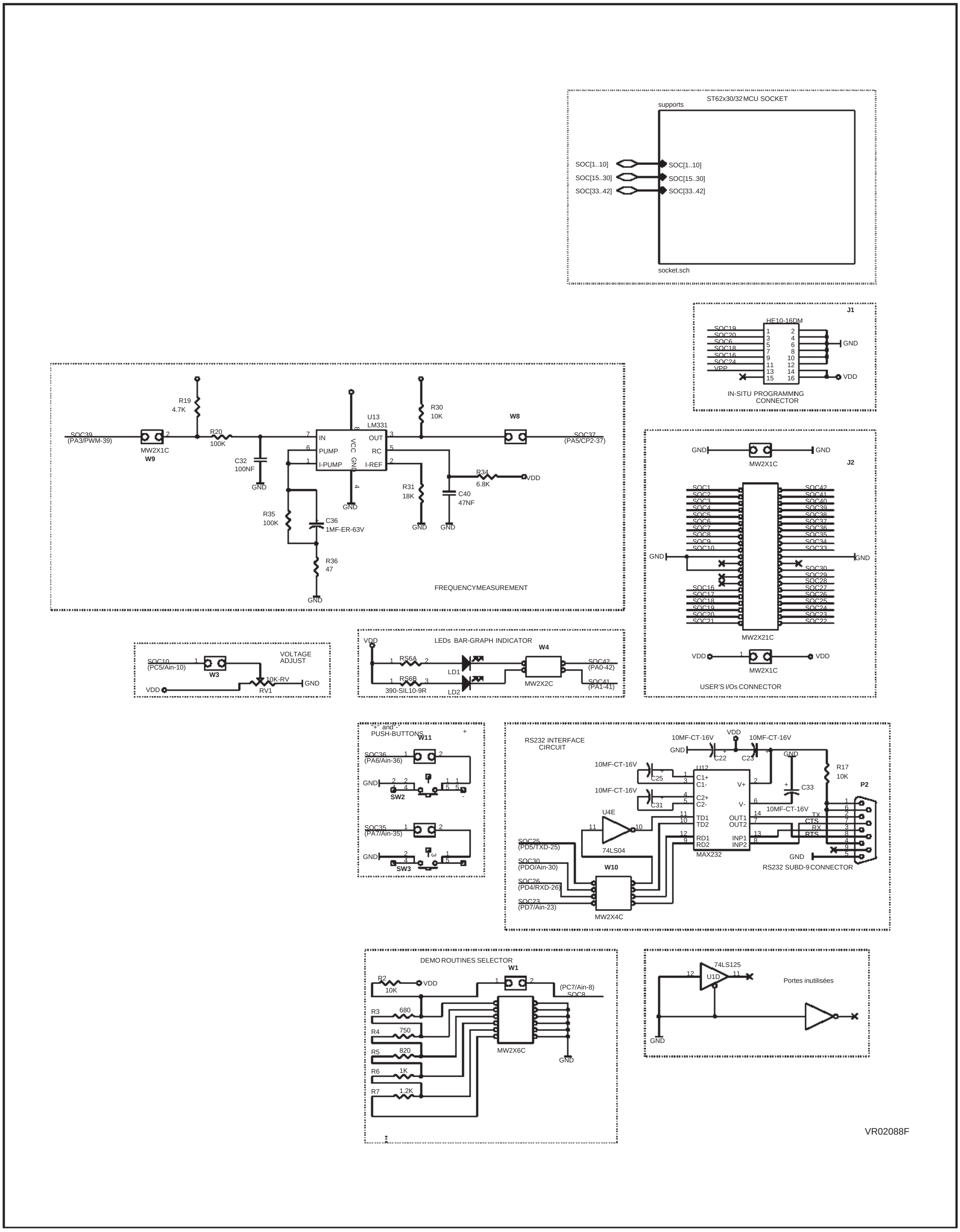
10.2 Starter Kit Board Schematic



Board Schematic (Part 1)



Board Schematic (Part 2)



Notes:

THE EUROPEAN QUALITY
AWARD
WINNER '97

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.

Printed in France by Imprimerie AGL

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 - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands - Singapore Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.

CHLORINE FREE

Recycled and chlorine free paper