ST7 FAMILY

ST7MDT20 EVALUATION BOARD

EXERCISE MANUAL

Release 1.8

ST7 FAMILY

ST7MDT20 EVAL BOARD

EXERCISE MANUAL

July 2002

REF- DOC-ST7MDT20-EVx

ST7 MDT20 Evaluation Board

Exercise Manual for the ST7 MDT20 Eval Board

INTRODUCTION

This document is intended for people getting started with ST7 microcontrollers or who are writing ST7 Assembly or C Code for the first time. It gives practical guidelines for performing the exercises using the ST7 MDT20 Eval Board and allows users to get to know ST7 peripherals and software tools.

The "MCU ON CD" CD-ROM containing the tutorial exercises is included in this package. The tutorial exercise installation procedure will create all the resource files (located in the INIT directory) and source files for the solutions (located in the RESULT directory) on the PC hard disk drive, as well as the batch files required to compile, assemble and link. This manual and the associated exercises are also available on ST Internet site (http://mcu.st.com). To simplify the use of the Hiware C exercises, select the default configuration when installing the software.

To install the tutorial exercises, insert the MCUonCD into the disk driver, the autorun program is then executed. Click on "Install your Development Tools", then click on "ST7 Tools", finally click on "Tutorial exercises".

To help explain the exercises, the MDT20 Evaluation Board may be used. This evaluation board is described in the following sections.

Two MDT20 evaluation boards are available:

■ ST7MDT20-EVY/xx: that's the one which was existing under the previous sales type: ST7MDT2-TRAIN/xx with the socket Yamaichi (IC149) mounted on it (refer to 1.3 Connection). You have to use the DB200 adaptor to be able to put the emulator (EMU2B) probe (with the DB389 adaptor) or a device (with the DB379 adaptor). The "xx" stands for the power supply type (EU, UK or US).

■ ST7MDT20-EVC/xx: that's the new board with the CAB socket mounted on it, avoiding to use adaptors to work in emulation mode or device mode. The CAB socket is also delivered with the EMU3 emulators supporting the TQFP64 14*14 package.

Rev. 1.8

Table of Contents

Table of Contents

1 ST7 MDT20 EVAL BOARD DESCRIPTION

The ST7 MDT20 Eval Board has been designed to support either an emulator probe or an ST7 device (FLASH, ROM) in a TQFP package. To do this, a reset system and a 16-MHz crystal are connected to dedicated pins.

1.1 OVERVIEW

This board can be used without any modifications to test and verify all the exercises given in this manual. Additional hardware functions can be implemented by installing components in the wrapping zone.

For more details refer to the schematic drawing in the appendix.

Figure 1. Overview of the ST7 MDT20 Eval Board
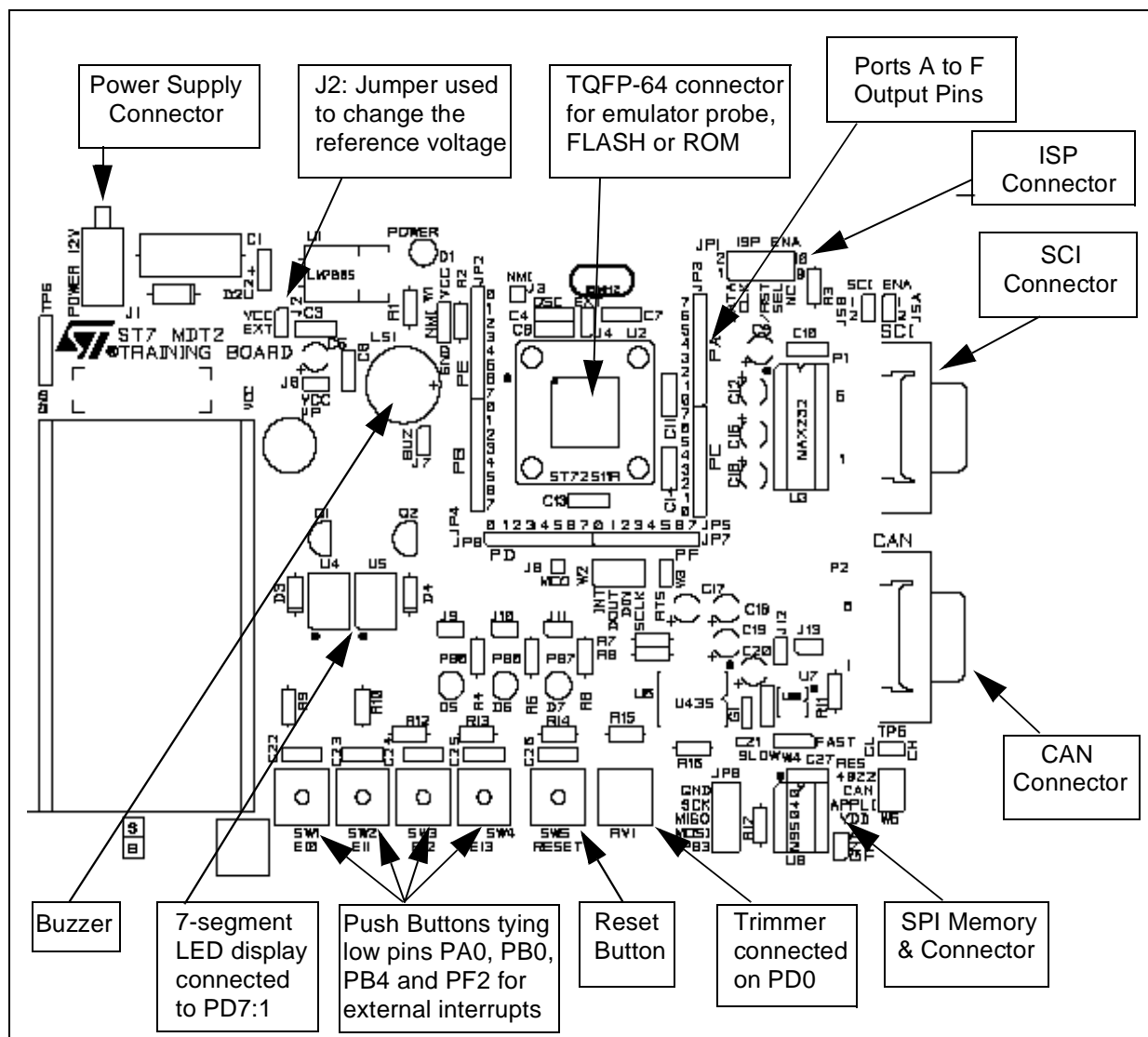
1.2 POWER SUPPLY

This board is supplied with an AC/DC converter and a 5-volt regulator (V_{CC} = 5 V). The reference voltage can be changed by removing jumper J2 and connecting the new reference voltage as shown in Figure 2.

Figure 2. How to Change the Board Reference Voltage
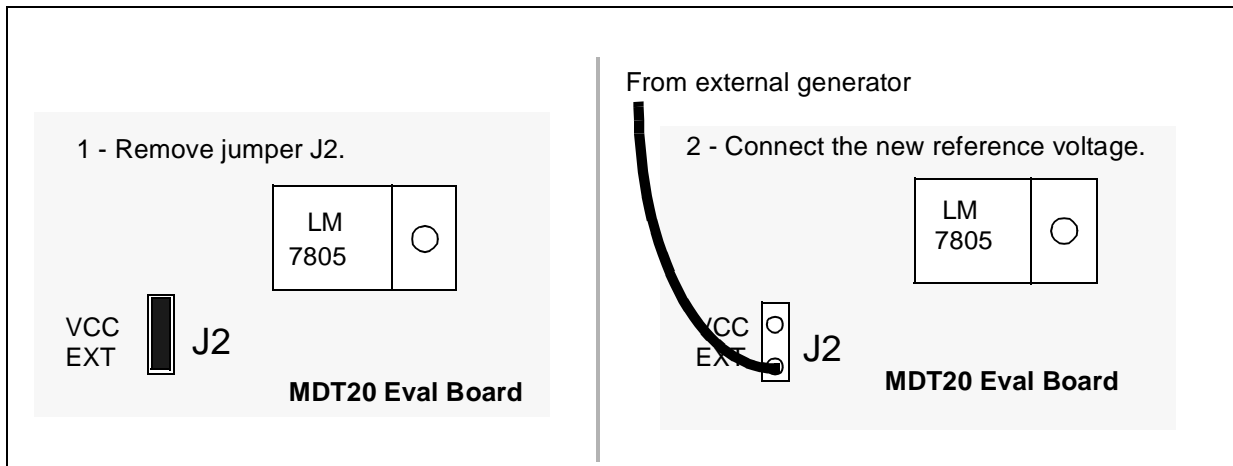
When the emulator probe is connected, the LEDs can be activated (when bits PB7:5 are set) and the port signals will be displayed, even if the board is not powered-on. However, the board must be supplied with 5 Volts (pins V_{CC} & GND) in order for the ADC and push-button switch to be operational.

1.3 CONNECTION

1.3.1 ST7MDT20-EVY/xx (Yamaichi socket)

This board is delivered with a Yamaichi socket (IC149). All connectors required for an emulation strategy are included either in the emulator package (DB407, DB389) or in the development kit package (DB404).

In order to use the MDT20 in Device mode, contact your nearest ST sales office and order the DB379 connector.
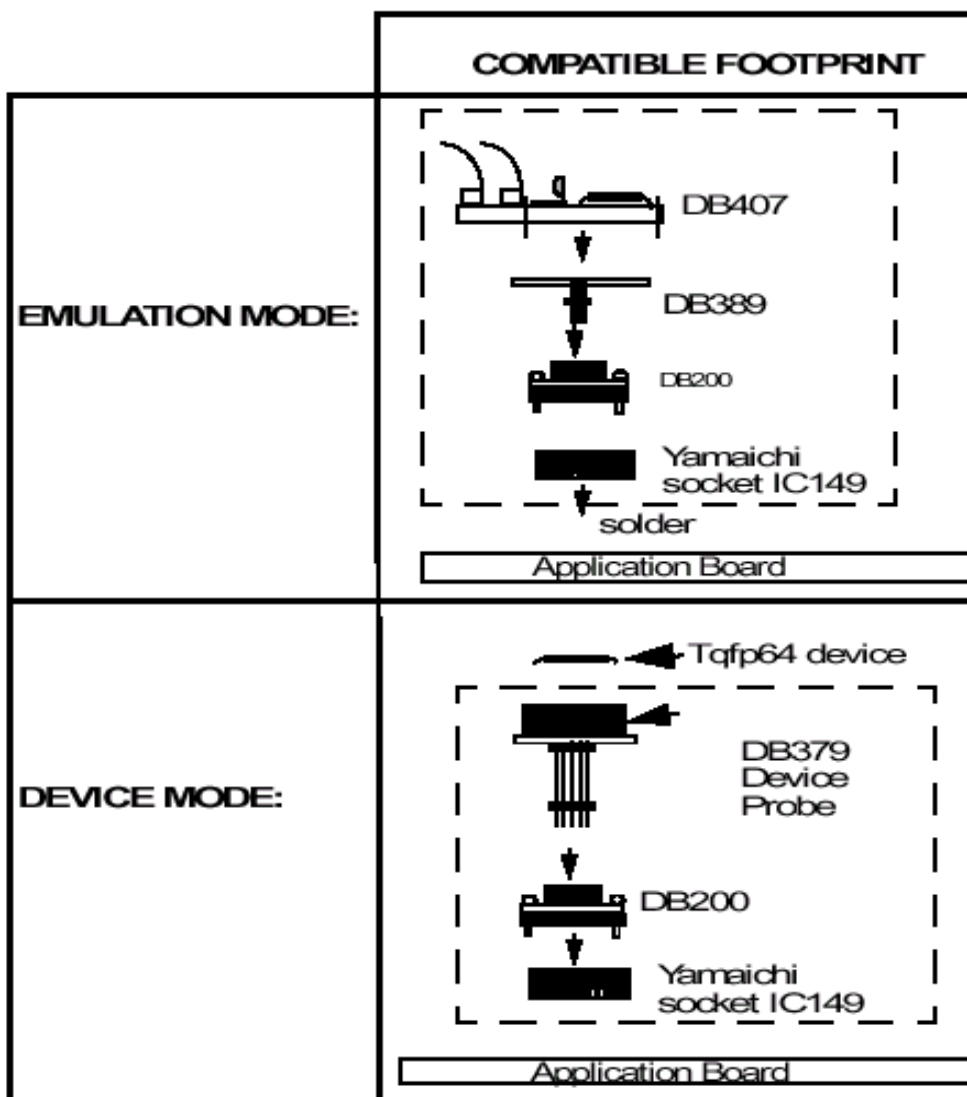
1.3.2 ST7MDT20-EVC/xx (CAB socket)

This board is delivered with a CAB socket. This socket has been chosen for EMU3 generation of emulators because they avoid the user to use different adaptors to work in emulation mode or in device mode. In emulation mode, the user has just to remove the socket top and to plug the emulator probe on the CAB socket mounted on the board. In device mode, the user has just to put the device on the socket mounted on the board and to close it with the socket top provided.

cab

Page 44.15	QFP-Socket Cover	Series Q2101
	QFP-Base Socket	Series Q2001

The QFP-Socket Cover comes with 4 screws M2x6 and the QFP-Base Socket comes with a screwdriver.
The Sockets are JEDEC Pad compatible and compact in design.

recom. pad design
K ≥ (C +1,0)
G ≤ (C - 2D)

Contact
Beryllium copper, nickel 2,5 µm, gold 0,25 µm
Insulator Body
Liquid cristall Polymer (LCP), UL 94 V0
Temperature
Continual operation   -25 to + 85°C
Resistance to solder heat
   Reflow   240°C, 20 Sec.
   Vaper phase   220°C, 60 Sec.

Mechanical life/cycles   100
Rated Voltage   100 V_{RMS}
Contact Resistance   < 70 mΩ
Insulation Resistance   > 5 x 10^8 Ω
Current Carring Capacity   0,5 A
(per contact)

Series Q2101

Series Q2001

Series Q2101
Series Q2001

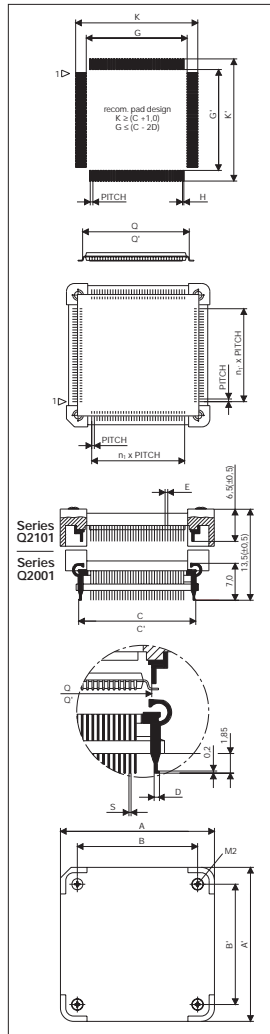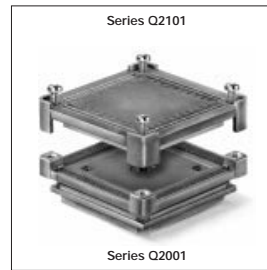| A / A'	B / B'	C / C'	D	E	H	Q / Q'	S	n₁ / n₁'	Pitch	Pins	S=Square / R=Rectangular	Q2001 Part No.	Q2101 Part No.
19,00	12,65	12,1	0,5	0,18	0,2	11,22	0,18	19	0,4	80	S	3303231	3303275
21,00	14,65	14,1	0,5	0,18	0,2	13,20	0,18	24	0,4	100	S	3303232	3303276
23,00	16,60	16,0	0,5	0,18	0,2	15,42	0,18	29	0,4	120	S	3303233	3303277
25,00	18,65	18,1	0,5	0,20	0,2	17,22	0,18	35	0,4	144	S	3303234	3303278
23,55 / 29,55	17,20 / 23,20	16,1 / 22,1	0,5	0,18	0,2	15,40 / 21,40	0,18	31 / 45	0,4	156	R	3303235	3303279
29,00	22,65	22,1	0,5	0,20	0,2	21,20	0,18	43	0,4	176	S	3303236	3303280
37,50	31,15	30,6	0,5	0,18	0,2	30,00	0,18	63	0,4	256	S	3303237	3303281
17,00	10,65	9,1	0,5	0,23	0,25	8,40	0,18	11	0,5	48	S	3303238	3303282
21,00	14,65	14,1	0,5	0,25	0,25	13,40	0,18	19	0,5	80	S	3303239	3303283
23,00	16,60	16,0	0,5	0,23	0,25	15,12	0,18	24	0,5	100	S	3303240	3303284
29,00	22,65	22,05	0,5	0,25	0,25	21,20	0,18	29	0,5	120	S	3303241	3303285
29,00	22,65	22,05	0,5	0,25	0,25	21,80	0,18	29	0,5	120	S	3303241	3303286
24,00 / 30,00	17,65 / 23,65	16,1 / 21,1	0,5	0,25	0,25	15,40 / 21,40	0,18	25 / 37	0,5	128	R	3303243	3303287
27,00	20,65	20,1	0,5	0,25	0,25	19,40	0,18	31	0,5	128	S	3303242	3303288
29,00	22,65	22,05	0,5	0,25	0,25	21,20	0,18	35	0,5	144	S	3303244	3303289
33,00	26,65	26,1	0,5	0,23	0,25	25,24	0,18	39	0,5	160	S	3303245	3303290
33,00	26,65	26,1	0,5	0,23	0,25	25,24	0,18	43	0,5	176	S	3303246	3303291
37,00	30,65	30,1	0,5	0,25	0,25	29,40	0,18	51	0,5	208	S	3303247	3303292
37,00	30,65	30,1	0,5	0,23	0,25	30,00	0,18	51	0,5	208	S	3303247	3303293
41,00	34,65	34,1	0,5	0,25	0,25	33,90	0,18	59	0,5	240	S	3303248	3303294
22,10	15,75	14,6	0,8	0,25	0,35	13,95	0,23	15	0,65	64	S	3303249	3303295
24,10	17,75	16,6	0,8	0,25	0,35	15,80	0,23	19	0,65	80	S	3303250	3303296
24,10	17,75	16,6	0,8	0,25	0,35	15,35	0,23	19	0,65	80	S	3303250	3303297
23,75 / 30,25	17,40 / 23,90	16,25 / 22,75	0,5	0,25	0,35	16,57 / 23,07	0,25	19 / 29	0,65	100	R	3303251	3303298
23,75 / 30,25	17,40 / 23,90	16,25 / 22,75	0,5	0,25	0,35	15,35 / 21,25	0,25	19 / 29	0,65	100	R	3303251	3303299
23,75 / 30,25	17,40 / 23,90	16,25 / 22,75	0,5	0,25	0,35	15,91 / 21,85	0,25	19 / 29	0,65	100	R	3303251	3303300
23,75 / 30,25	17,40 / 23,90	16,25 / 22,75	0,5	0,25	0,35	16,77 / 23,07	0,25	19 / 29	0,65	100	R	3303251	3303301
28,10	21,75	20,6	0,8	0,25	0,35	21,00	0,23	24	0,65	100	S	3303252	3303302
38,70	31,70	30,6	0,8	0,25	0,35	30,30	0,23	35	0,65	144	S	3303253	3303303
38,70	31,70	30,6	0,8	0,23	0,35	30,30	0,23	39	0,65	160	S	3303254	3303304
38,70	31,70	30,6	0,8	0,23	0,35	30,30	0,23	41	0,65	168	S	3303255	3303305
19,70	13,35	12,2	0,5	0,5	0,5	11,20	0,25	10	0,8	44	S	3303260	3303311
19,70	13,35	13,3	0,75	0,25	0,5	12,22	0,23	10	0,8	44	S	3303261	3303312
23,70	17,35	17,3	0,75	0,25	0,5	16,20	0,30	15	0,8	64	S	3303256	3303306
23,70	17,35	16,2	0,5	0,25	0,5	15,20	0,25	15	0,8	64	S	3303262	3303307
23,70 / 30,10	17,35 / 23,75	17,3 / 23,3	0,75	0,25	0,5	16,20 / 22,40	0,30	15 / 23	0,8	80	R	3303257	3303308
23,70 / 30,10	17,35 / 23,75	17,3 / 23,3	0,75	0,25	0,5	15,96 / 22,00	0,30	15 / 23	0,8	80	R	3303257	3303309
23,80 / 29,80	17,45 / 23,45	17,2 / 23,2	0,75	0,25	0,6	16,60 / 22,60	0,30	12 / 18	1,0	64	R	3303259	3303310

Please check Dimension Q because several versions are availiable.

1.4 OSCILLATION SYSTEM

The ST7MDT20 Eval Board is designed to be used with an on-board 16-MHz crystal. But another clock signal can be used by connecting the corresponding oscillator to the OSCIN pin. To do this, remove jumper J4 and connect the external clock signal generator to the OSCIN pin as shown in Figure 3.

Figure 3. How to Change the Oscillator Frequency
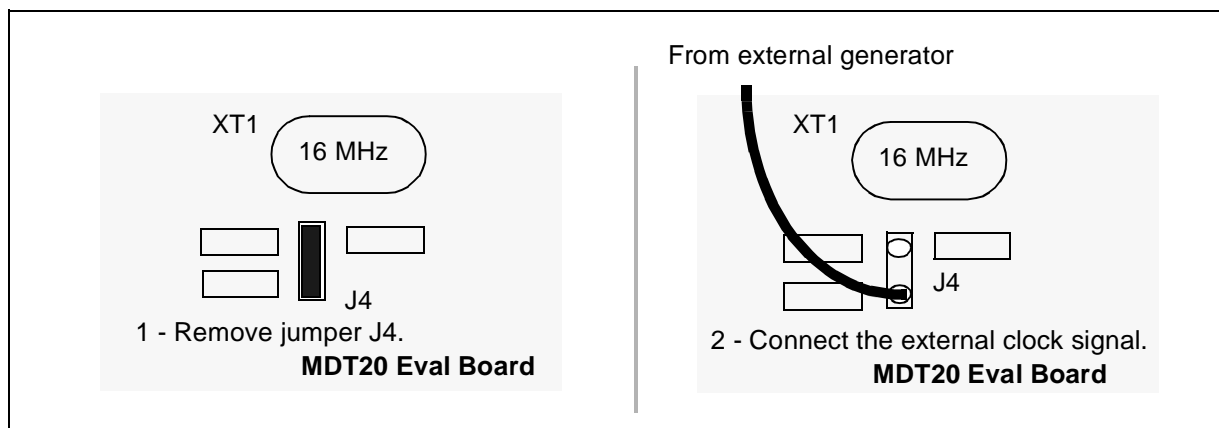
1.5 CAN DEVICE COMMUNICATION

A CAN transceiver (L9615 or U435) is connected to the microcontroller CAN pin (pins PE2 and PE3). In order to supply power to the CAN transceiver, the "APPLI VDD" jumper must be connected to the W5 connector.

The Rx_CAN pin of the MCU can be disconnected from the transceiver by removing jumper J12 and the CAN speed is selected according to the configuration of jumper W4. A 120-ohm terminating resistor is validated when jumper J13 is connected.

1.6 SPI SERIAL COMMUNICATION

An external EEPROM memory with an SPI serial interface is connected to the SPI pins of the ST7 socket as shown in the following figure. But the serial EEPROM can be disconnected and the SPI bus used for other communications. This hardware configuration is explained in Application Note AN970 "SPI Communication Between ST7 and E2PROM".

Figure 4. How to Use the SPI Connection

Using the embedded SPI EEPROM	Using the SPI Bus
Install all the jumpers on the SPI connector.	Remove the jumpers and use the left-hand side connector.
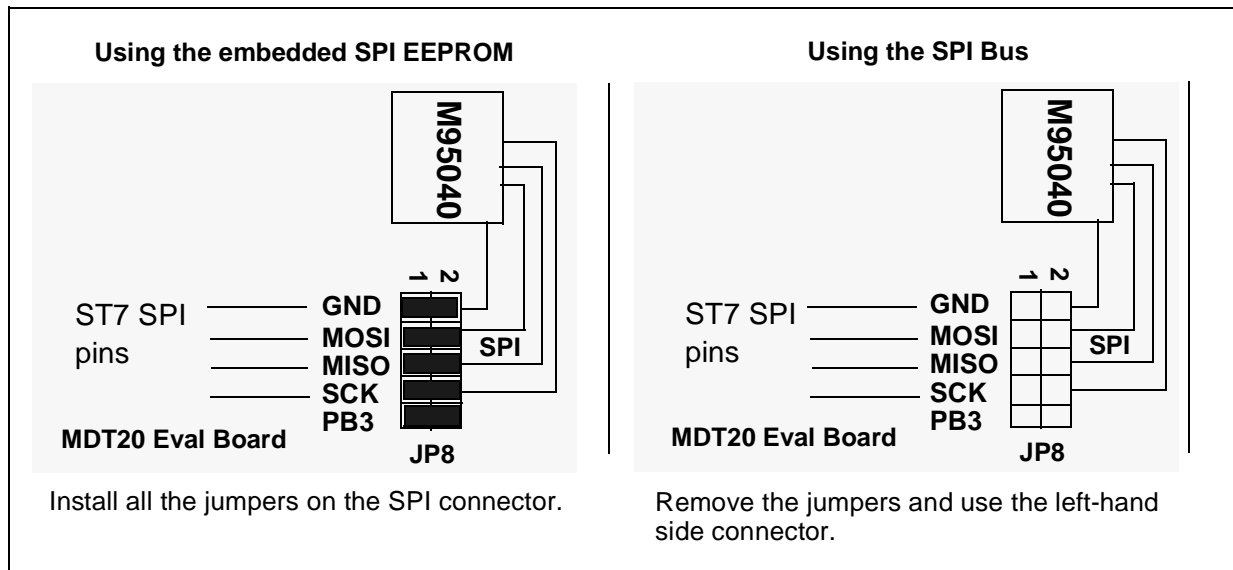
1.7 ISP (IN-SITU PROGRAMMING) CONNECTION

The ISP connector is used to program Flash devices using ISP protocol. If an ST programming tool is being used, connector JP1 must be plugged to the cable provided by the ST programming tool and the board has to be powered-on.

Figure 5. ISP Connections
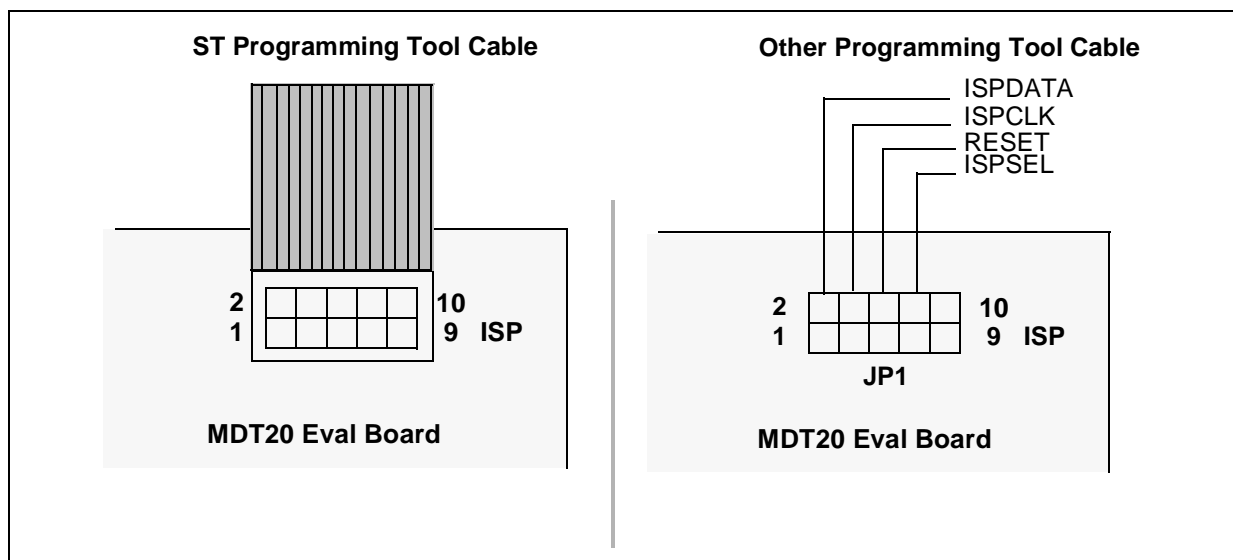
This connector can also be used with a tool other than the one provided by ST. In this case, use the connections as described in Figure 5.

1.8 SCI DEVICE COMMUNICATION

The DB9 connector can be used to link the ST7 MDT20 Eval Board to a PC, for example. A "MAX232" component is used to fit the ST7 TTL output to the line level of a PC serial connector.

1.9 I/O PORT FUNCTIONS

All I/O port pins are accessible through external connectors (connector JP3 for port A, JP4 for port B, JP5 for port C, JP6 for port D, JP2 for port E, JP7 for port F).

1.9.1 Port A

The PA0 pin can be used to generate an external interrupt (EI0). It is connected to the ground through push-button switch SW1.

Pins PA2 and PA3 are used to validate the 7-segment LED displays (useful for multiplexing).

1.9.2 Port B

Pins PB5, PB6, PB7 are connected to 3 LEDs.

Pins PB0 and PB4 can be used to generate an external interrupt (EI2 and EI3). These pins are connected to the ground through push-button switches SW3 and SW4.

Table 1. Port B Alternate Functions (ST72511 only)

PB0	PWM3	PWM Output 3
PB1	PWM2	PWM Output 2
PB2	PWM1	PWM Output 1
PB3	PWM0	PWM Output 0
PB4	ARTCLK	PWM-ART External Clock

1.9.3 Port C
Table 2. Port C Alternate Functions

PC0	OCMP2_B	Timer B Output Compare 2
PC1	OCMP1_B	Timer B Output Compare 1
PC2	ICAP2_B	Timer B Input Capture 2
PC3	ICAP1_B	Timer B Input Capture 1
PC4	MISO	SPI Master In Slave Out Data
PC5	MOSI	SPI Master Out Slave In Data
PC6	SCK	SPI Serial Clock
PC7	SS	SPI Slave Select

1.9.4 Port D

This port is connected to the 7-segment LED which is validated with a '1' logical value on pins PA2 and PA3.

Table 3. Port D Alternate Functions

PD0	AIN0	Analog Input 0
PD1	AIN1	Analog Input 1
PD2	AIN2	Analog Input 2
PD3	AIN3	Analog Input 3
PD4	AIN4	Analog Input 4
PD5	AIN5	Analog Input 5
PD6	AIN6	Analog Input 6
PD7	AIN7	Analog Input 7

1.9.5 Port E

This port is connected to both the SCI transceiver (ST232 or MAX232) and the CAN transceiver (available only on ST72511, ST72512, ST72532), as described in the following table.

Table 4. Port E 0Alternate Functions

PE0	TDO	SCI Transmit Data Out
PE1	RDI	SCI Receive Data In
PE2	CANTX	CAN Transmit Data Output
PE3	CANRX	CAN Receive Data Input

1.9.6 Port F

The PF2 pin can be used to generate an external interrupt (EI1). It is connected to the ground through the push-button switch.

Table 5. Port F Alternate Functions

PF0	CLKOUT	Main Clock Output (Fosc/2)
PF1	BEEP	Beep Signal Output
PF3	OCMP2_A	Timer A Output Compare 2
PF4	OCMP1_A	Timer A Output Compare 1
PF5	ICAP2_A	Timer B Input Capture 2
PF6	ICAP1_A	Timer B Input Capture 1
PF7	EXTCLK_A	Timer A External Clock Source

1.10 JUMPER DESCRIPTION

1.10.1 Jumper J2 - External Voltage Supply

Please refer to Section 1.2.

1.10.2 Jumper J4 - External Oscillator System

Please refer to Section 1.4.

1.10.3 Jumper J5 - SCI Enable/Disable

In order to use the SCI peripheral device, these 2 jumpers must be connected; otherwise, it is recommended that they be removed from the board. Please refer to Figure 4.

1.10.4 Jumper J6 - MCU Voltage Supply

This jumper is dedicated to the MCU power supply only. When it is removed, it can be used, for example, to measure the current consumption of the MCU by connecting a DC-ammeter in parallel.

1.10.5 Jumper J7 - Buzzer Enable/Disable

This jumper enables the buzzer to be used once it is connected to pin PF1.

1.10.6 Jumpers J9, J10, J11 - LED Enable/Disable

These 3 jumpers are used to connect the three LEDs (D5, D6 and D7) to pins PB5, PB6 and PB7. It is recommended that the jumpers be removed from the board if this feature is not used.

1.10.7 ISP Connector

Please refer to Section 1.7.

1.10.8 NMI Pull-down/Pull-up

A pull-down (or, pull-up) is applied on pin NMI if a jumper is connected between pins 2-3 (or, pins 1-2) on the board.

2 SOFTWARE INSTALLATION

The following software packages must be installed in order to do the exercises:

– ST7 Assembly Toolchain,

– STVD7 Debugger,

– C-Compiler (Hiware and Cosmic Evaluation versions).

2.1 ST7 ASSEMBLY TOOLCHAIN

The ST7 Assembly Toolchain can be directly installed on the PC using the "MCU ON CD" CD-ROM (Welcome.exe file) or downloaded from the ST Internet site (http://mcu.st.com).

When the installation is completed, click on the Finish button and then reboot the PC so that that the autoexec.bat modifications can be taken into account.

2.2 STVD7

The STVD7 is the new 32-bit debugger developed by ST. This is a basic IDE (Integrated Development Environment) tool, which is used to edit, build and debug applications in a single environment.

The STVD7 program can be directly installed on the PC using the "MCU ON CD" CD-ROM (Welcome.exe file) or downloaded from the ST Internet site (http://mcu.st.com).

2.3 HIWARE AND COSMIC DEMO VERSIONS

These two third-party companies have developed a C-compiler for use with ST7 microcontrollers. An evaluation version of each compiler can be directly installed on the PC using the "MCU ON CD" CD-ROM (Welcome.exe file) or downloaded from each of their respective Internet sites (http://www.hiware.com and http://www.cosmic-software.com).

2.4 TUTORIAL EXERCISES

The exercises can be installed using the "MCU ON CD" CD-ROM ("Install your Development Tools", "ST7 Tools", "ST Demo Boards" and "ST7 Evaluation Board").

3 TRAINING EXERCISES

3.1 EXERCISE 1: ASSEMBLER SYNTAX

The purpose of the first exercise is to improve your knowledge of ST7 Assembler syntax and the STVD7 environment.

1. Launch the STVD7 Simulator:

■ Configure the Toolchain paths as requested. The ST assembly toolchain path is already entered. For Hiware, the correct path is C:\Hiware\prog and for Cosmic: C:\Program Files\cosmic software\st7eval\cxst7 (if the default paths were selected during the installation procedure). A Browse function is also available to find the correct path.

■ Create a new project:
  – Click on "File, New Workspace" and enter the requested information (see Figure 7.)
  – Click on the "Source Directory" tab in the Workspace window and double click on the directory: the default user directory is selected by default.

2. Modify the files located in the following directory (C:\exercise\init\asm, see Figure 6.) to correct the assembler syntax. Some syntax errors have been introduced by replacing correct directives or lines of code by '?????'.

3. When the file has been corrected, write the compile.bat batch file to assemble, link and create a new *.s19 file (assign a new name to the file). The batch file also generates the S19, SYM, MAP and OBJ files. For further assistance, please refer to the Technical Manual.

4. Then, click on "Project, Build", or directly on the Build shortcut, to build your application.

5. Detected errors are listed in the Output window. Click on the error to display the line of code. Correct the error and then rebuild.

6. To debug, simply click on the blue "D" displayed in the toolbar. Then select "Tools, MCU Configuration" in order to configure the MCU (ST72334N4) and the option byte (Watchdog software,...).

Now, the rest is up to you...

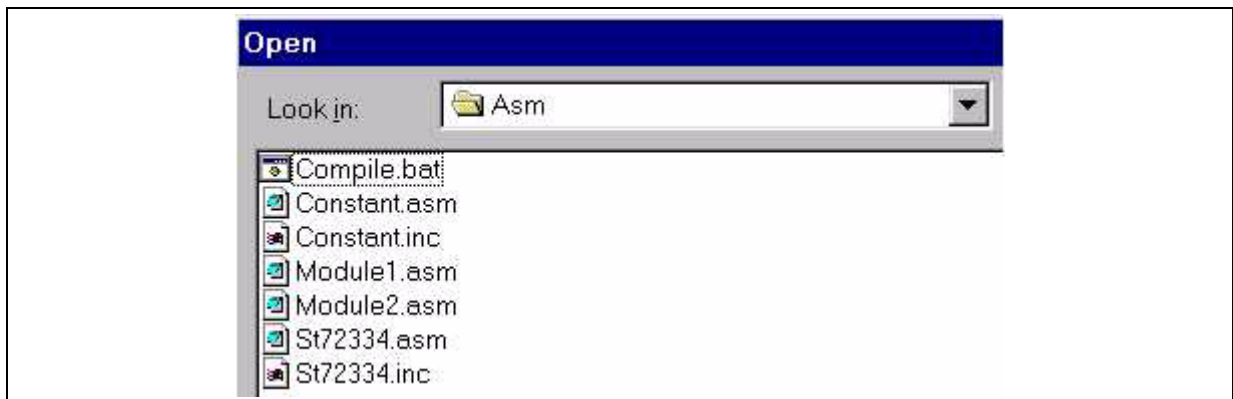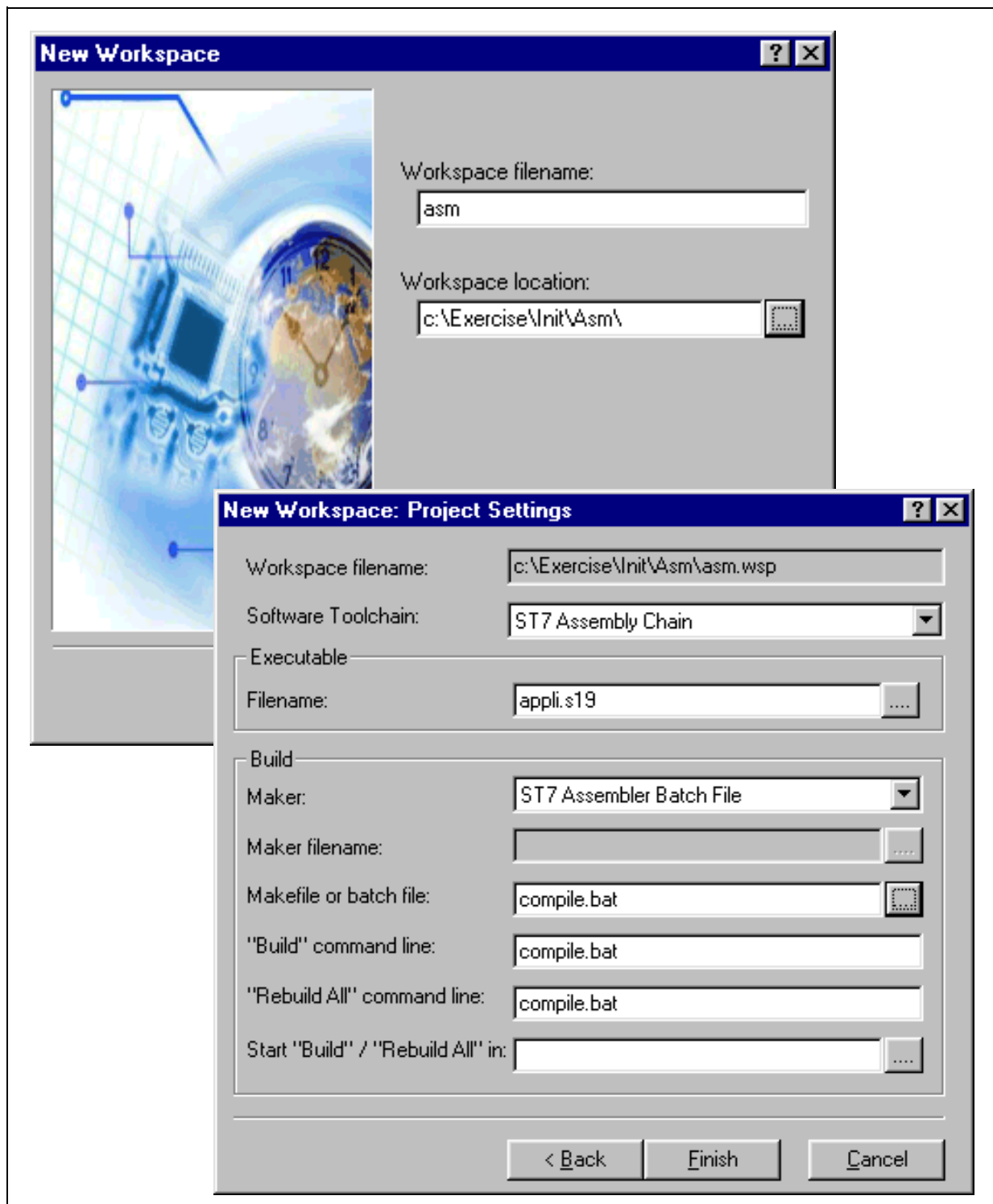**Figure 6. Assembler Syntax Resource Files**

Figure 7. STVD7 New Workspace Project Settings

3.2 EXERCISE 2: USE OF THE 16-BIT TIMER IN ASSEMBLY LANGUAGE

Starting with a similar program frame (New Project: File, New Workspace), design a software program for the ST72334N4 that:

1. Generates a PWM signal on pin PF4 (40-kHz frequency, 20% duty cycle).

2. Generates an interrupt every period and toggles pin PB0 defined as a push-pull output (square wave at 20 kHz).

Refer to the ST72334 datasheet for the correct use of the timer in PWM mode.

Since the MCU uses a 16-MHz crystal, it must be configured in Normal mode and the "Divider by 2" function must be selected for the timer clock in order to have the best accuracy.

■ Write a main program that:
  – Configures pin PF0 as a push-pull output.
  – Enters the calculated value for the pulse length and the period length in TAOC1R and TAOC2R registers using the datasheet formula.
  – Configures the 16-bit timer in PWM mode.
  – Enables the interrupt process and waits for interrupts.
■ Write an interrupt routine that toggles pin PFO and clears interrupt flags ICF1and ICF2.

In this exercise, the entire structure is given, but only the comment lines have been kept. The correct ST7 code must be written by using only the comments.

3. Check the validity of your software program by displaying the signal on pins PF0 and PF4 with an oscilloscope or by using the ST7 Waveform Editor after an STVD7 simulation.

When an application is run on the simulator, a port.out file that contains all the pin changes is automatically generated. It is used to check the operation of the application without using any hardware.

The following timing diagram should be obtained:

Figure 8. PWM Timing Diagram
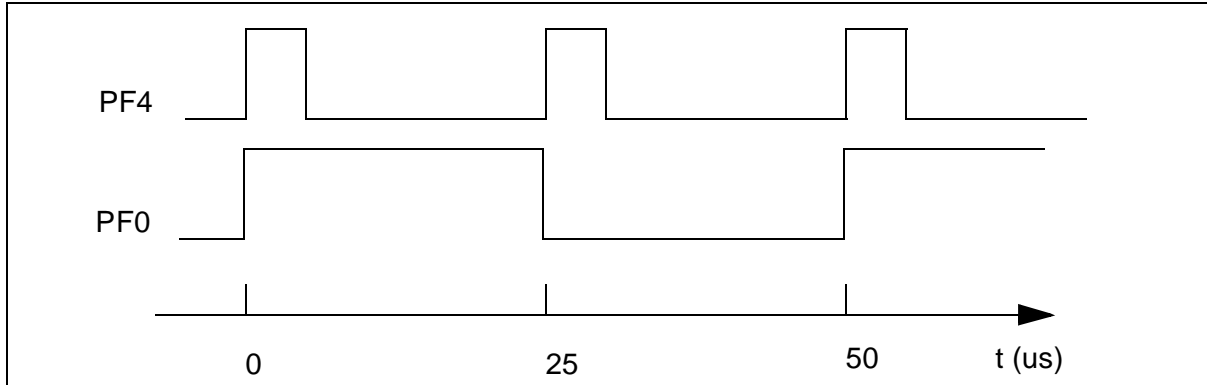
The hardware registers and mapping file (ST72334.asm) are printed in the appendix. The labels of the hardware registers are also listed in this section. The resource files are shown in Figure 9.

Figure 9. Timer Resource Files
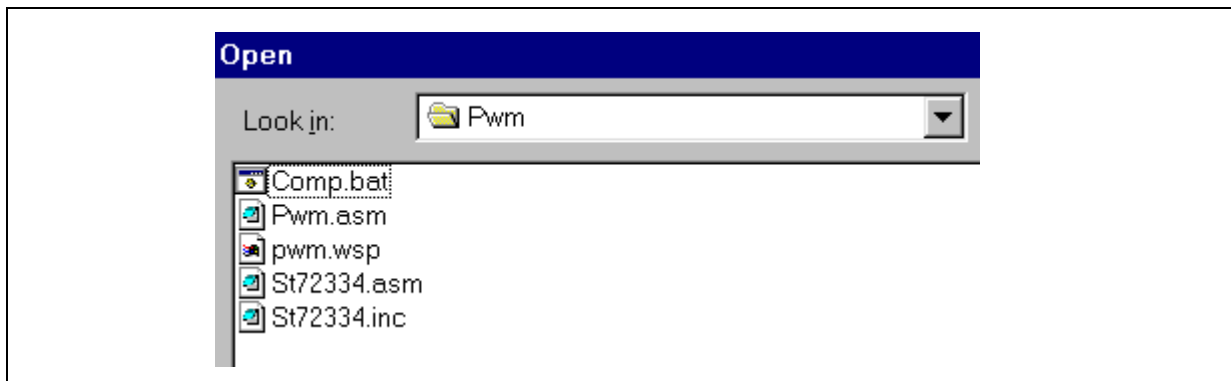
3.3 EXERCISE 3: USE OF THE 16-BIT TIMER IN HIWARE C LANGUAGE

3.3.1 Purpose of the Exercise

The purpose of this exercise is to use the Hiware C Toolchain to write a program that performs the same PWM functions as in the previous exercise, but using the following resources.

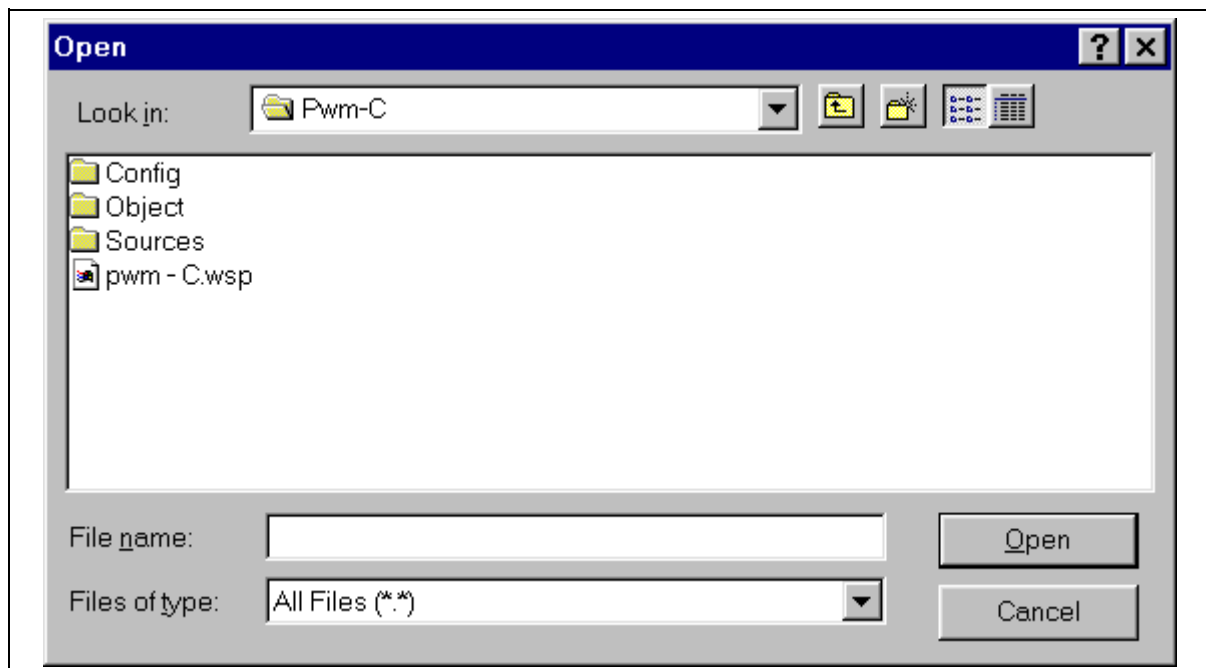**Figure 10. C-Language Timer Resource Files**

The linker parameter file (enviro.prm) and the makefile (enviro.mak) file have been placed in the working directory. If an editor such as Winedit is used in place of the STVD7 program, please refer to Application Note AN 989 'Starting with ST7 Hiware C' available on Internet (http://mcu.st.com. Then select "ST7", "Technical Documentation", "Application Notes".)

The Default.env and Project.ini files are used to configure the Hiware C Toolchain and the text editor. These two files are automatically created by the Hiware setup program in the Configuration dialog box.

All files are ready-for-use, except for the main.c and itenviro.c files. In order to be able to use these files, the following procedure must be completed:

■ In the main.c file, write the instructions that:
  – Configure pin PF0 as a push-pull output.
  – Enter the calculated value for the pulse length and the period length in TAOC1R and TAOC2R registers using the datasheet formula.
  – Configure the 16-bit timer in PWM mode.
 – Enable the interrupt process and wait for interrupts.

■ Write an interrupt function in itenviro.c file that:

– Toggles pin PFO.

– Clears interrupt flags ICF1 and ICF2.

3.3.2 How to Use the Hiware C Demo Program

The Hiware C Compiler demo can be directly installed on the PC using the "MCU ON CD" CD-ROM (Welcome.exe file) or downloaded from their Internet site (http://www.hiware.com).

When the installation is completed, open the HIWARE TOOLS icons in the group of HIWARE demo icons (Start Menu) and click on the first icon displayed in the Hiware tools toolbar.

Figure 11. HiWare Tools Toolbar
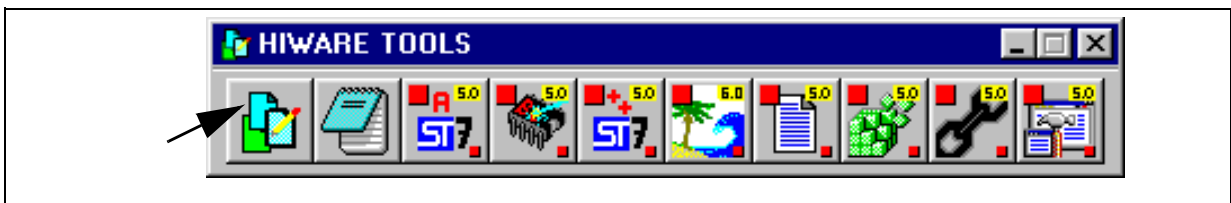
This icon opens the *Configuration* dialog box. Click on *New button* and enter the application directory (C:\exercise\init\C334) and the various tabs (refer to Application Note AN989), then click on *OK*.

This action creates or updates the project.ini and default.env files located in the application directory.

An editor is not required for the Hiware Tool box which is just used to create the default.env and project.ini files.

Once the Hiware configuration parameters are entered, open the STVD7 program (simulator or development kit) and create a new project by specifying the Hiware toolchain in the project settings window.

The correct configuration is listed in the technical manual.

3.4 EXERCISE 4: USING THE ADC

The purpose of this exercise is to write lines of code in C language that are able to read an analog voltage and to show the result in hex value on the two 7-segment LED displays by multiplexing.

The the ST72334 C-environment is located in the C:\exercise\init\ADC directory. Click on the "ADC.wsp" project in this directory and the resource files are displayed as shown in Figure 12.

Figure 12. ADC Resource Files
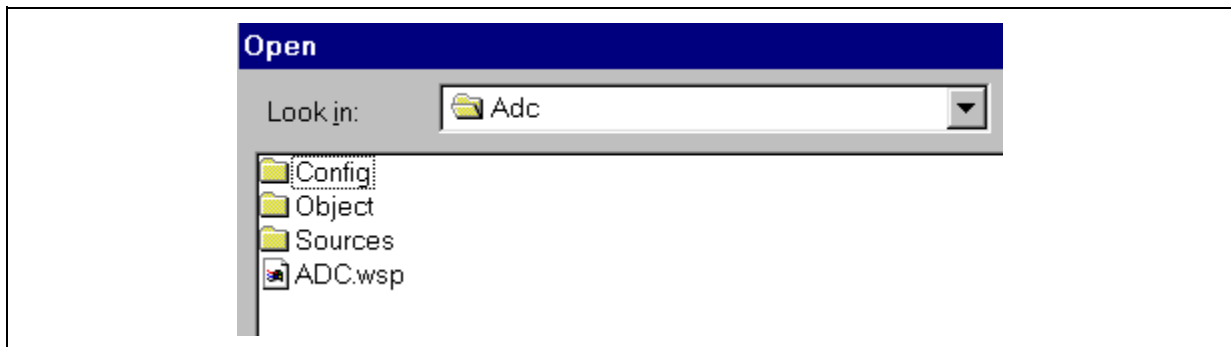
Only the "main.c" program must be completed, according to the explanatory comments.

- Table "tab[16]" contains the configuration for the 7-segment LED displays. When addressing tab[i], the corresponding digit i is accessed when PDDR is pushed.

- Label "lab1" is an infinite loop. It starts a new conversion, waits until it is completed, and store the tens and ones unit value into the i and j registers. Then, it outputs the i and j values on Port D, and decreases a counter so that the result is displayed on the LEDs. Then, it starts again.

- A high level on output PA3 (output PA2, respectively) selects the tens (ones, respectively) value on the dedicated 7-segment LED display.

3.5 EXERCISE 5: USING THE SPI DRIVER
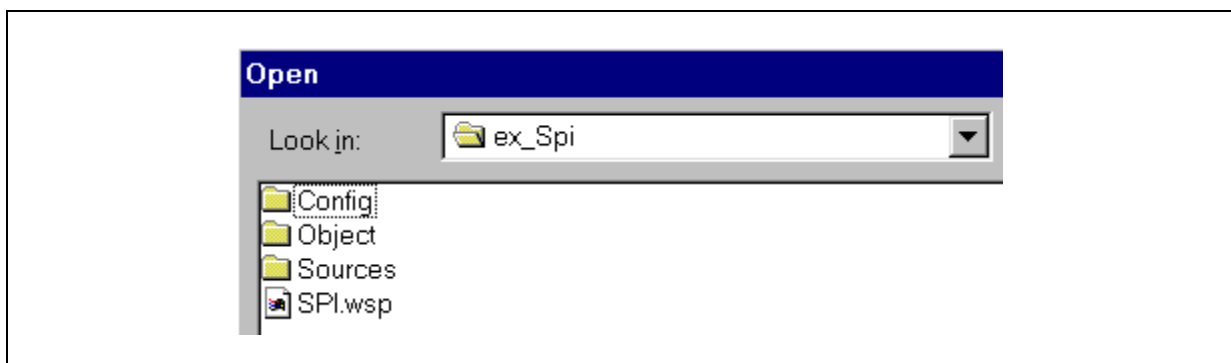
The purpose of this exercise is to use the SPI driver to program the embedded SPI EEPROM.

The structure of the main module and the batch file for efficient HIWARE C language compiling as well as the ST72334 resource directory are located in the SPI directory.

First, open the "SPI.wsp" project file.

The resource files are displayed as shown in Figure 13.

Figure 13. SPI Resource Files

You have to complete the "main.c" and "itenviro.c" programs, according to the explanatory comments.

The first step is to send "value[3]" into the EEPROM, starting at the address 0x00. Each value has to be displayed on the 7-segment LED display. Then, by pressing push-button switches EI1, EI2 or EI3, the previously sent value must be read at the 0x00, 0x01 and 0x02 addresses, and the result must be displayed on the 7-segment LED display in hex value. If the 0x00 EEPROM memory address is read, LED PB5 has to light up; LEDs PB6 or PB7 LED have to light up if the 0x01 or 0x02 memory addresses are read.

3.6 EXERCISE 6: USING THE SCI DRIVER

The purpose of this exercise is to use the SCI driver to communicate with your PC, with the help of the "Hyperterminal" software. The parameters for this software program must be correctly configured before use.

1) Click on "File → Properties" and select the correct COM port.

2) In the "Settings" menu, check the "Windows Keys" and "ANSI Emulation" check boxes.

3) In "File → Properties → Configure", set the following parameters: "Bits per second: 19200", "Data bits: 8", "Parity: None", "Stop bits: 1", "Flow control: None".

The ST72334 environment in HIWARE C language is located in the ex_SCI directory.

First, open the "SCI.wsp" project file.

Figure 14. SCI Resource Files
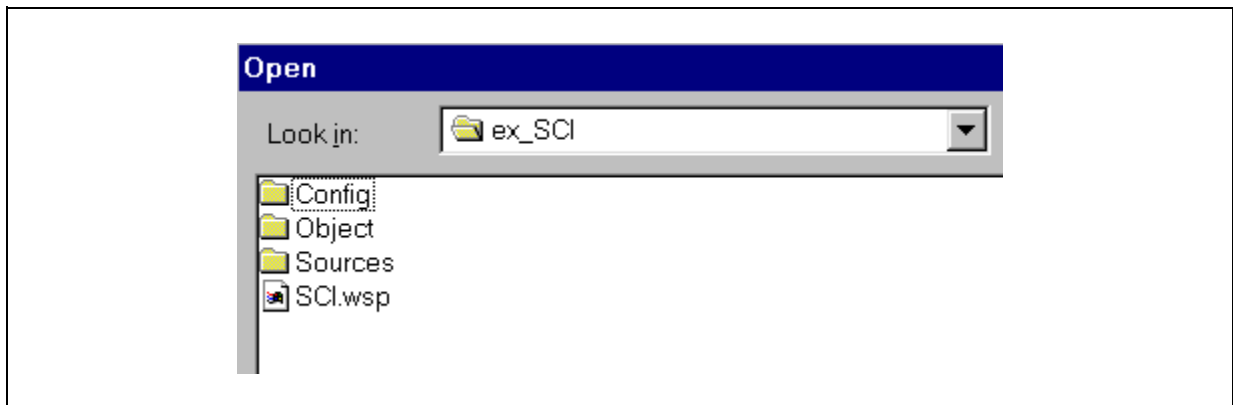
Then, complete the "main.c" program, according to the explanatory comments.

The first step is to send "title[ ]" to the PC. Then, wait for eight characters to be entered before sending it back to the PC. Afterwards, the program starts over, and eight new characters must be entered.

4 APPENDIX 1: ST72334 REGISTER AND MEMORY MAPPING FILE

The following lists the content of the hardware register file: ST72334.asm.

```
st7/
;**********************************************************************
; TITLE:          ST72334.ASM
; AUTHOR:         CMG Microcontroller Application Team
; DESCRIPTION:    ST72334 Register and memory mapping
; Define here the micro RAM & ROM Size according to the sales type:
; #define ST72334N2  1  ;=> 8K ROM, 384 bytes RAM also available for ST72334N2
 #define ST72334N4   1  ;=> 16K ROM, 512 bytes RAM  also available for ST72334N4
;**********************************************************************
;**********************************************************************
BYTES       ; following addresses are 8 bit length
;******************************************************************
  segment byte at 0-71 'periph'
;******************************************************************
;******************************************************************
;              I/O Ports registers
;******************************************************************
.PADR  DS.B  1    ; port A data register
.PADDR DS.B  1    ; port A data direction register
.PAOR  DS.B  1    ; port A option register
    DS.B  1    ; not used

.PCDR  DS.B  1    ; port C data register
.PCDDR DS.B  1    ; port C data direction register
.PCOR  DS.B  1    ; port C option register
    DS.B  1    ; not used

.PBDR  DS.B  1    ; port B data register
.PBDDR DS.B  1    ; port B data direction register
.PBOR  DS.B  1    ; port B option register
    DS.B  1    ; not used

.PEDR  DS.B  1    ; port E data register
.PEDDR DS.B  1    ; port E data direction register
.PEOR  DS.B  1    ; port E option register
    DS.B  1    ; not used

.PDDR  DS.B  1    ; port D data register
.PDDDR DS.B  1    ; port D data direction register
.PDOR  DS.B  1    ; port D option register
    DS.B  1    ; not used

.PFDR  DS.B  1    ; port F data register
.PFDDR DS.B  1    ; port F data direction register
.PFOR  DS.B  1    ; port F option register
    DS.B  1

reserved0 DS.B 8     ; unused
```

```
;***************************************************************************
;              Miscellaneous 1 register
;***************************************************************************
.MISCR1 DS.B   1    ; miscellaneous register 1


;***************************************************************************
;              SPI registers
;***************************************************************************
.SPIDR DS.B   1    ; SPI data register
.SPICR DS.B   1    ; SPI control register
.SPISR DS.B   1    ; SPI status register
reserved1 DS.B 5    ; unused
;***************************************************************************
;              Main Clock Registers
;***************************************************************************
.MCCSR DS.B 1; Main Clock Control/Status register
;***************************************************************************
;              Watchdog Registers
;***************************************************************************
.WDGCR DS.B   1    ; Watchdog control register
.CRSR DS.B 1; clock, reset, supply control/status register
;***************************************************************************
;              EEPROM Register
;***************************************************************************
.EECSR  DS.B   1    ; Data-EEPROM Control/Status register
reserved2 DS.B 4     ; unused
;***************************************************************************
; Timer A registers
;***************************************************************************
.TACR2  DS.B   1    ; timer A control register 2
.TACR1  DS.B   1    ; timer A control register 1
.TASR   DS.B   1    ; timer status register
.TAIC1HR DS.B  1    ; timer A input capture 1 high register
.TAIC1LR DS.B  1    ; timer A input capture 1 low register
.TAOC1HR DS.B  1    ; timer A output compare 1 high register
.TAOC1LR DS.B  1    ; timer A output compare 1 low register
.TACHR  DS.B   1    ; timer A counter high register
.TACLR  DS.B   1    ; timer A counter low register
.TAACHR DS.B   1    ; timer A alternate counter high register
.TAACLR DS.B   1    ; timer A alternate counter low register
.TAIC2HR DS.B  1    ; timer A input capture 2 high register
.TAIC2LR DS.B  1    ; timer A input capture 2 low register
.TAOC2HR DS.B  1    ; timer A output compare 2 high register
.TAOC2LR DS.B  1    ; timer A output compare 2 low register
;***************************************************************************
;              Miscellaneous 2 register
;***************************************************************************
.MISCR2 DS.B   1    ; miscellaneous register 2
;***************************************************************************
; Timer B registers
;***************************************************************************
.TBCR2  DS.B   1    ; timer B control register 2
.TBCR1  DS.B   1    ; timer B control register 1
.TBSR   DS.B   1    ; timer B status register
.TBIC1HR DS.B  1    ; timer B input capture 1 high register
```

```
.TBIC1LR DS.B  1    ; timer B input capture 1 low register
.TBOC1HR DS.B  1    ; timer B output compare 1 high register
.TBOC1LR DS.B  1    ; timer B output compare 1 low register
.TBCHR  DS.B  1    ; timer B counter high register
.TBCLR  DS.B  1    ; timer B counter low register
.TBACHR DS.B  1    ; timer B alternate counter high register
.TBACLR DS.B  1    ; timer B alternate counter low register
.TBIC2HR DS.B  1    ; timer B input capture 2 high register
.TBIC2LR DS.B  1    ; timer B input capture 2 low register
.TBOC2HR DS.B  1    ; timer B output compare 2 high register
.TBOC2LR DS.B  1    ; timer B output compare 2 low register


;***********************************************************************
;          SCI registers
;***********************************************************************
.SCISR DS.B  1    ; sci status register
.SCIDR DS.B  1    ; sci data register
.SCIBRR DS.B  1     ; sci baud rate register
.SCICR1 DS.B  1     ; sci control register 1
.SCICR2 DS.B  1     ; sci control register 2
.SCIERPR DS.B  1      ; sci extended receive prescaler register
     DS.B  1     ; reserved
.SCIETPR DS.B  1      ; sci extended transmit prescaler register
reserved5 DS.B 24    ; unused


;***********************************************************************
;          ADC registers
;***********************************************************************
.ADCDR DS.B  1    ; adc data register
.ADCCSR DS.B  1     ; adc control status register


;***********************************************************************
  segment byte at 80-FF 'ram0' ;Zero Page
;***********************************************************************
WORDS      ; following addresses are 16 bit length


;***********************************************************************
;  for ST72334J2 or ST72334N2 with 8K ROM & 384 bytes RAM

#ifdef ST72334N2
  segment byte at 100-1FF   'stack'
  segment byte at E000-FFDF  'rom' ;
#endif
;***********************************************************************


;***********************************************************************
  segment byte at C00-CFF 'eeprom' ;EEPROM area
;***********************************************************************


;***********************************************************************
;  for ST72334J4 or ST72334N4 with 16K ROM & 512 bytes RAM
;
#ifdef ST72334N4
  segment byte at 100-1FF   'stack'
  segment byte at 200-27F   'ram1'
```
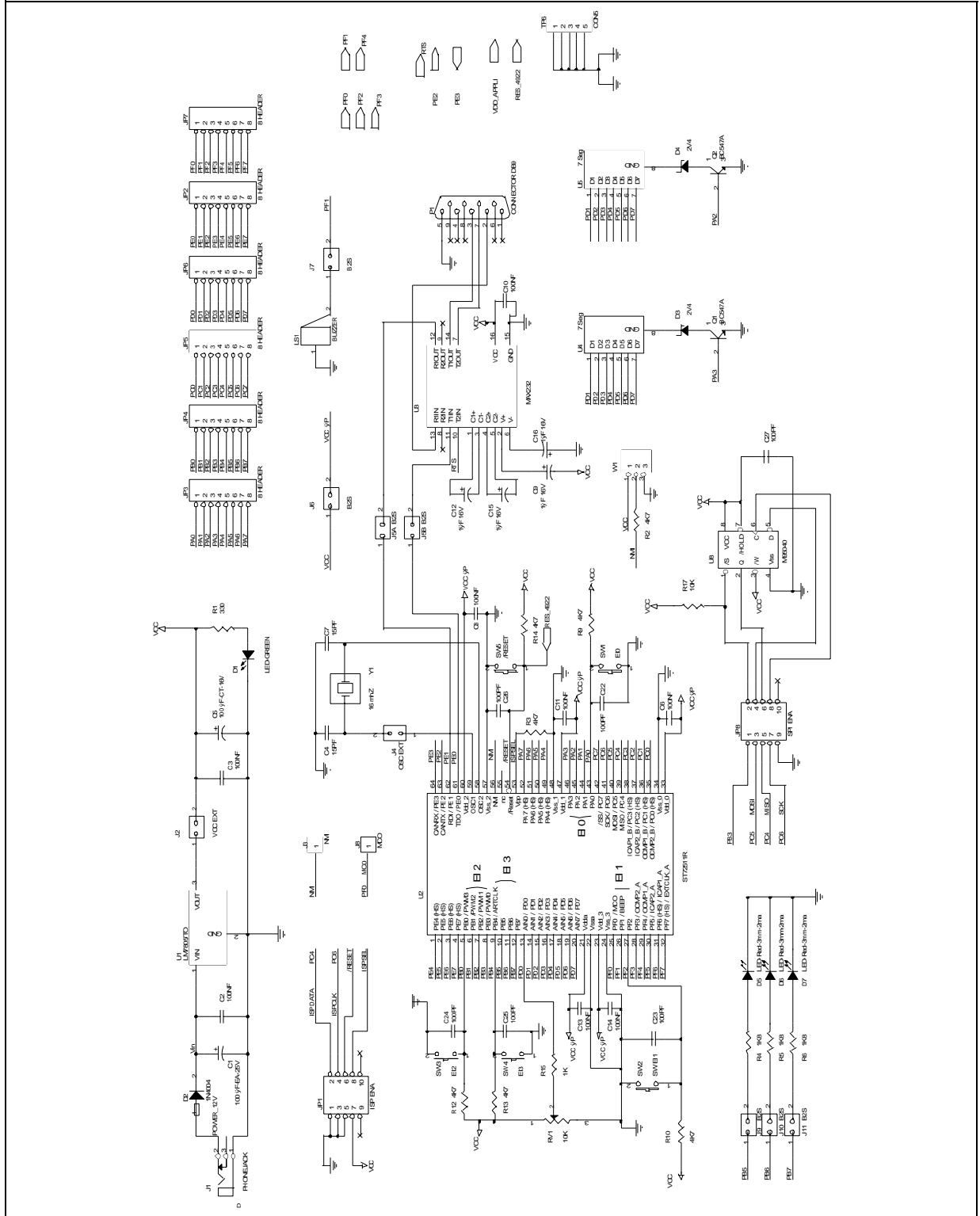
```
  segment byte at C000-FFDF 'rom'
#endif
;**********************************************************************

;**********************************************************************
  segment byte at FFE0-FFFF 'vectit'
;**********************************************************************
end
```

5 APPENDIX 2: SCHEMATIC DIAGRAMS

Figure 15. MDT20 Eval Board (Part 1)

Figure 16. MDT20 Eval Board (Part 2)