

NEC

User's Manual

V850ES – Touch It!

**Demonstration kit for the V850ES/JG2 and
NL2432HC22-41K 3.5" TFT Colour LCD Module
with touch screen panel**

[MEMO]

- **The information in this document is current as of May, 2007. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
"Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
"Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
"Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics America Inc.
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 030
Fax: 0211-65 03 1327

Sucursal en España
Madrid, Spain
Tel: 091- 504 27 87
Fax: 091- 504 28 60

Succursale Française
Vélizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

Filiale Italiana
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

Branch The Netherlands
Eindhoven, The Netherlands
Tel: 040-265 40 10
Fax: 040-244 45 80

Tyskland Filial
Taebly, Sweden
Tel: 08-638 7200
Fax: 08-638 7222

United Kingdom Branch
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics (China) Co., Ltd
Beijing, P.R. China
Tel: 010-8235-1155

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Korea Ltd.
Seoul, Korea
Tel: 02-558-3737
Fax: 02-558-5141

NEC Electronics Singapore Pte. Ltd.
Singapore
Tel: 65-6253-8311
Fax: 65-6250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-8175-9600
Fax: 02-8175-9670

Table of Contents

CHAPTER 1	INTRODUCTION	9
CHAPTER 2	KIT CONTENTS	11
CHAPTER 3	RELEVANT DOCUMENTATION	13
CHAPTER 4	QUICK START GUIDE	16
4.1	Using the pre-loaded Introductory Demo.	16
4.1.1	Touch It! welcome screens	16
4.1.2	Introduction screens	17
4.1.3	Calibration and Keyboard Screen	17
4.1.4	Using bitmaps as buttons	18
4.2	Ending the demo	18
CHAPTER 5	HARDWARE SETUP	19
5.1	V850ES/JG2	19
5.2	LEDs	19
5.3	Switches	19
5.4	Connectors	20
5.5	Epson S1D13A05 LCD/USB companion chip	21
5.6	TFT Colour LCD display	21
5.7	TouchScreen panel	22
CHAPTER 6	CONNECTORS	23
6.1	TFT connector J9	23
6.2	5v power connector J2	23
6.3	USB connector J8	23
6.4	Flash Programming connector J4	24
CHAPTER 7	SOFTWARE INSTALLATION	25
7.1	Touch It! Demonstration kit software installation	25
7.2	PC Hardware requirements	25
7.3	Package contents	25
7.4	Software installation	25
7.5	Structure of Software installation	26
CHAPTER 8	TO RUN THE SUPPLEMENTARY DEMOS	28
8.1	Supplementary demos	28
8.2	Application demo	28
8.3	Segger examples	29
CHAPTER 9	STRUCTURE OF THE DEMO CODE	30
9.1	Basic structure of software demo	30
9.2	Segger embOS and emWin	31
9.3	Demo code overview	33
9.4	Main	34

9.4.1	Initialising the V850.....	34
9.4.2	Initialising the TFT display registers.....	35
9.4.3	Initialising the Epson S1D13A05 LCD companion chip.....	35
9.4.3.1	LCDLIN.c.....	35
9.5	OS_Init.....	35
9.5.1	Setting up the LCD touch screen header files.....	35
9.5.1.1	LCDconf.h.....	35
9.5.2	Guiconf.h.....	36
9.5.3	GUITouchConf.h.....	36
9.6	Creating tasks.....	36
9.6.1	MainTask.....	37
9.6.2	User task.....	37
9.6.3	Touch Screen task.....	37
9.6.4	Touchscreen Calibration.....	39
9.7	General overview of introduction demo program.....	39
CHAPTER 10	STARTING THE IAR TOOLS.....	41
10.1	IAR tools.....	41
10.2	Compiler and linker settings.....	42
10.3	Compiling a project.....	43
CHAPTER 11	TO SETUP A NEW IAR PROJECT.....	44
11.1	To create a new IAR project space.....	44
11.1.1	Adding project groups.....	44
11.1.2	Adding files to the project groups.....	45
CHAPTER 12	USING THE MINICUBE2 (OPTIONAL).....	46
12.1	Installing the MINICUBE2 software.....	46
12.1.1	Upgrading the MINICUBE2 firmware.....	46
12.1.2	Installing the QB2 MINICUBE2 Flash programming application.....	46
12.2	Using the MINICUBE2 to programme the Touch It! board (optional).....	47
CHAPTER 13	DEBUGGING THE PROGRAM USING ON-CHIP DEBUG WITH THE MINICUBE2 (OPTIONAL).....	49
13.1.1	XCL Setup.....	49
13.1.2	Disable the UART.....	50
13.1.3	Debug.....	50
CHAPTER 14	COLOUR INDEX.....	51
14.1	Colour mapping between the LCD connector and board.....	51
CHAPTER 15	EPSON S1D13A05 CONFIGURATION SETTING.....	53
CHAPTER 16	BILL OF MATERIALS.....	56
CHAPTER 17	SCHEMATICS.....	59

List of Figures

Figure 1-1 Touch IT! Demonstration kit (without the enclosure)	10
Figure 4-1 USB connection to the Touch It! board	16
Figure 4-2 Touch the opening screen	16
Figure 4-3 Introduction screens.....	17
Figure 4-4 Keyboard and Calibration screens.....	17
Figure 4-5 Bitmap control demo	18
Figure 5-1 Position of switches on Touch It! circuit board.....	20
Figure 5-2 TFT LCD Touch panel display	22
Figure 6-1 PG-FP4 and MINICUBE2 programmers.....	24
Figure 7-1 Software installation structure.....	26
Figure 7-2 Structure of demo sub-directory.....	26
Figure 7-3 Structure of Src sub-directory	27
Figure 8-1 Starting the Application Demo	28
Figure 9-1 Organisation of software files.....	30
Figure 9-2 Overview of main()	33
Figure 9-3 Touch Screen circuit	38
Figure 9-4 Structure of Segger example containing Maintask()	39
Figure 10-1 IAR project overview panel	41
Figure 10-2 IAR Projects Option configuration window.....	42
Figure 12-1 MINICUBE2.....	46
Figure 12-2 MINICUBE2 switch settings	47
Figure 12-3 MINICUBE2 connections	47
Figure 12-4 QB-Programming application and device setup	48
Figure 13-1 Reconfiguring the linker XCL file.....	49
Figure 13-2 IAR Debug option setup for the MINICUBE2.....	50
Figure 15-1 Epson general configuration	53
Figure 15-2 Epson preference configuration	53
Figure 15-3 Epson clock configuration	54
Figure 15-4 Epson panel configuration	54
Figure 15-5 Epson panel power configuration.....	55

List of Tables

Table 1 V850ES manuals.....	13
Table 2 TFT colour LCD module.....	13
Table 3 Segger embOS and emWin documents.....	13
Table 4 Epson S1D13A05 documents.....	14
Table 5 Touch It! Schematics.....	14
Table 6 MINICUBE2 Flash programmer and on-chip debug (not included).....	15
Table 7 Switches on the Touch It! Board.....	19
Table 8 Connectors on TFT V850ES board.....	20
Table 9 Memory model settings.....	32
Table 10 IAR compiler settings.....	43
Table 11 Correct pinning from Epson S1D13A05 to TFT LCD connector.....	51
Table 12 Actual pinning from Epson S1D13A05 to TFT LCD connector.....	52
Table 13 Bill of Materials.....	56

CHAPTER 1 INTRODUCTION

Thank you for choosing NEC Electronics V850ES TFT LCD Touch It! demonstration kit. Whether you are designing a LCD solution for the first time, or you are looking for a solution for your next product, the Touch It! demonstration kit from NEC Electronics can help you see how easy it is to integrate Touch Screen technology into your next design.

If you are designing your next control panel for a domestic appliance, heating controller or portable device, touch panel LCD displays are being used increasingly in products as consumers demand more functionality and usability. NEC Electronics' technology can help you achieve that.

The V850ES – Touch It! demonstration kit provides a complete demonstration and reference design of the NEC Electronics V850ES/JG2 32-bit microcontroller and the NL2432HC22-41K 3.5" TFT Colour LCD Module which includes a touchscreen interface. It also includes demonstrations using trial versions of the embOS Real Time Operating System (RTOS) and emWin embedded graphics technology from Segger Microcontroller Systeme which provides an ideal platform to start your software evaluation and prototype.

The V850ES – Touch It! Demonstration kit will enable you to:-

- run the pre-loaded introductory demo
- compile additional Segger emWin examples based on your own application

The optional MINICUBE2 Flash programmer/on-chip debug device is required to:-

- program prebuilt demonstration programs to the Touch It! board
- program additional Segger emWin examples compiled with the EWW850-KS16 IAR tools
- allows you to make small adaptations to the demonstrations, recompile and execute

The pre-loaded introductory demo program can be used right out of the box on the V850ES – Touch It! demonstration kit when a 5 volt supply is applied to the board via the USB cable.

For information regarding the understanding and use of the embOS and emWin, please refer to the Segger documentation.

Please be aware, the trial versions of the Segger products are only for evaluation. For development and production purposes, the use of embOS and emWin products require licensing from Segger Microcontroller Systeme GmbH. For pricing and licensing details, please contact Segger Microcontroller Systeme directly to discuss this.

www.segger.com

SEgger Microcontroller Systeme GmbH
Heinrich-Hertz-Str. 5
40721 Hilden, Germany
Phone: +49(0)2103-2878-0
Fax: +49(0)2103-2878-28
email: info@segger.com

The purpose of this demonstration kit is to help you quickly understand what components are required and how they can be used when you incorporate a TFT colour LCD display into your next design.

If you wish to flash program additional pre-built programs into the demonstration kit, you will need to use the MINICUBE2 Flash programmer/On-chip debug device (not included with the Touch It! Demonstration kit) which is available from NEC Electronics.

Please refer to the MINICUBE2 installation guides and user's manual listed in Chapter 3 for more information.

The EWV850-KS16 Kick Start version of the V850 IAR Embedded Workbench tools allows you to compile upto 16Kbytes of application code. The project environment is setup to produce a hex file for flash ROM programming.

The MINICUBE2 can also be used for on-chip debug (OCD) with the Touch It! Demonstration kit. Just ensure the correct xcl linker file is used to reserve 2Kbytes of flash and 16 bytes of RAM, and disable any UART resources or interrupts for the ROM monitor.



Figure 1-1 Touch IT! Demonstration kit (without the enclosure)

The kit contains all the necessary hardware and software to quickly setup and run and show a number of demos on the TFT Colour LCD.

The emWin simulation is also included in the CD to allow emWin to be evaluated on the PC platform without the need for a Touch It! Demonstration board. Unzip the emWin_Simulation file and start the executable in the EXE directory. Please refer to the readme.html file and the Software Products->emWin->Simulation section on the Segger website for further information. Please note that Segger licensing terms and agreements apply.

CHAPTER 2 KIT CONTENTS

- Touch It! Demonstration kit board containing the
 - V850ES/JG2 μ PD70F3718 cpu
 - Epson S1D13A05 LCD/USB Companion chip
 - USB connector (connected to S1D13A05 but not used in this demonstration kit)
 - Buttons to toggle LCD backlight
 - TFT connector
 - Programming connector

- NL2432HC22-41K TFT Colour LCD Module
 - 3.5" Super Reflective NLT screen
 - 320 x 240 QVGA
 - Touchscreen panel

- IAR software tools Kick Start edition (limited version to 16Kbytes application code)
- Sample software utilising Segger embOS and emWin software libraries
 - Introductory demo
 - Pre-loaded on the Touch It demonstration kit.
 - Sample Washing machine demo
 - Hex file requires download to the V850ES Flash memory
 - Additional pre-built Segger emWin examples to download
 - Touch_sample – allows you to see the touchscreen cursor values
 - GraphXY – demo showing a combination of graphics, check boxes and buttons.
 - MemDev_demo – demo showing the emWin memory devices option in action to display a fast dynamic graphical display
 - ColorBar demo – shows the TFT LCD colour palette

- emWin simulation
 - GUISimulation executable to show emWin running on a PC platform
 - Also includes sample code which can be compiled with Visual C++ v6.0 or later
- USB cable

CHAPTER 3 RELEVANT DOCUMENTATION

Table 1 V850ES manuals	
U17715EJ1V0UD00.pdf	User Manual for μ PD70F3718 cpu
U17966EE4V0IF00.PDF	MINICUBE2 Setup manual for Flash Programming
U18527EJ1V0UM00.PDF	QB Programmer User Guide
http://www.eu.necel.com/update	Development tools software downloads and updates

Table 2 TFT colour LCD module	
nl2432hc22-41k_DOD-MD-0058_1.pdf	Preliminary Specification for TFT Colour LCD Module
More information available from http://www.eu.necel.com/products/display/index.html http://www.nec-lcd.com/en/technology/nlt_sr_nlt.html http://www.eu.necel.com/_pdf/EPMC-PU-0018-2.0.PDF	

Table 3 Segger embOS and emWin documents	
embOS_Generic.pdf	embOS User and reference manual by Segger Microcontroller Systeme
embOS_V850_IAR.pdf	embOS CPU & compiler specifics for V850 & IAR tools by Segger Microcontroller Systeme
emWinUser.pdf	emWin Graphic Library with Graphic Interface manual by Segger Microcontroller Systeme
More information available from Segger website	www.segger.com

Table 4 Epson S1D13A05 documents	
Technical Manual s1d13a05tm.pdf	S1D13A05 LCD/USB Companion chip Technical manual X40A-Q-001-1
S1D13A05_Hardware_Functional_Spec_Rev_702.pdf Document number X40A-A-001-07	Hardware Functional Specification for Epson S1D13A05 LCD/USB companion chip
x40ag003.pdf	Programming Notes and Examples
S1D13A05 configuration tool CFG	Download from products section of the Epson web site
Epson website	http://www.erd.epson.com/ http://www.epson-electronics.de under Products->Semiconductors->Graphic Controllers -> Standard LCD Controller

Table 5 Touch It! Schematics
Complete Schematic Touch_it v2.0 - 25Mar2007.pdf

Table 6 MINICUBE2 Flash programmer and on-chip debug (not included)	
<p>Goto http://www.eu.necel.com/update select QB-MINI2-EE → MINICUBE2 for 78K0S, 78K0, and 78K0R for latest firmware and select QB-Programmer -> Programmer GUI for MINICUBE2 for QB2 programming installation to obtain the following installation files</p>	
MINICUBE2_Firmware_Vxxx.zip	Latest firmware for MINICUBE2. Uses the Diagnostic tool from MINICUBE2 utilities to install
MINICUBEUtilities_Vx.xx.zip	Installation for MINICUBE2 programs to check on-chip-debug and diagnostic tool to for MINICUBE2 self-check and firmware upgrade.
MQB2ALL_Vxxx.zip	USB driver for QB2 programming application
QBP_Vxxx.zip	QB2 Flash programmer installation
PRM-70F3724_Vxxx.zip	PRM file for QB2 device setup
U18527EJ1V0UM00.pdf	QB-Programmer programming GUI operation
U17966EE4V0IF00.PDF	MINICUBE2 setup manual
U18371EJ1V0UM00.pdf	QB-MINI2 on-chip-debug emulator with programming function
ZUD-CD-06-0044-4-E.pdf ZUD-CD-06-0044-4-E.pdf	MINICUBE2 diagnostics user's manual MINICUBE2 On-chip debug checker user's manual

CHAPTER 4 QUICK START GUIDE

4.1 Using the pre-loaded Introductory Demo.

The Touch It! Demonstration kit is preloaded with an introduction software demo that only requires the kit to be connected to 5v using the DC power cable or via the USB connector. (see Table 8 Connectors on TFT V850ES board for the correct jumper connections). Some of the screens are adapted from Segger's emWin demonstration programs.

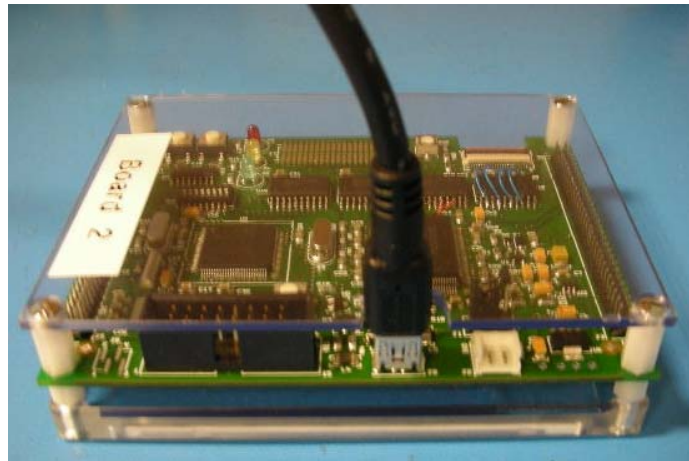


Figure 4-1 USB connection to the Touch It! board

The demo sequence is as follow:-

4.1.1 Touch It! welcome screens

This welcome page first displays the Touch It! handprint.

Tap the handprint and then on the "Next" button shown on the next screen to continue.

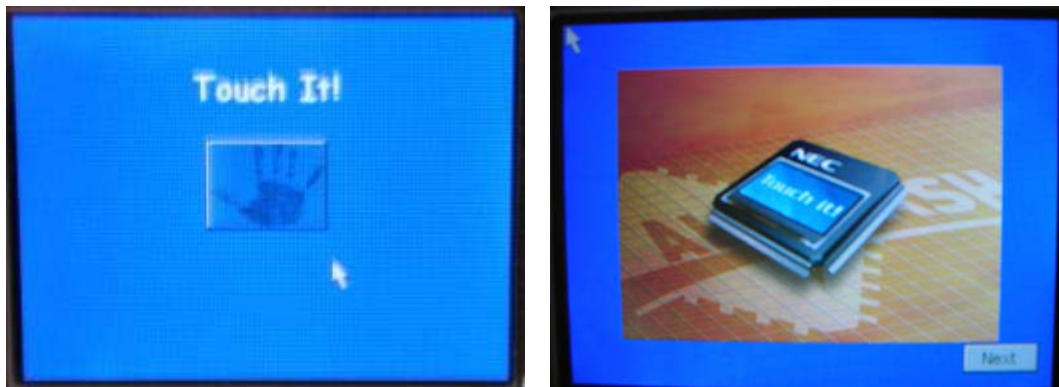


Figure 4-2 Touch the opening screen

4.1.2 Introduction screens

This shows the NEC Electronics introduction screen and a brief slide show of what the Touch It! Demonstration kit can do for you.

The demo will continue to the next screen after 20 seconds or when the “Next” button is pressed,

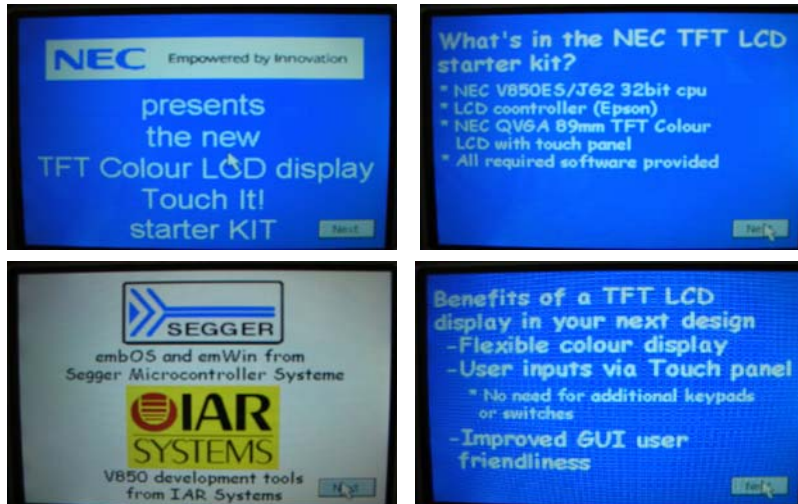


Figure 4-3 Introduction screens

4.1.3 Calibration and Keyboard Screen

After the slide show, a calibration program will run allowing you to recalibrate the touchscreen if necessary. Click the Calibrate button to enter this program.

To test the calibration, click on the test calibration button or click on the Keyboard button to try out the demo as a “QWERTY” keyboard. Click 'ESC' to quit the demo.



Figure 4-4 Keyboard and Calibration screens

4.1.4 Using bitmaps as buttons

The next demo shows bitmaps used as buttons. Click on a bitmap to display a different message at the bottom of the screen. This screen will end after 20 seconds or when “Next” is pressed.

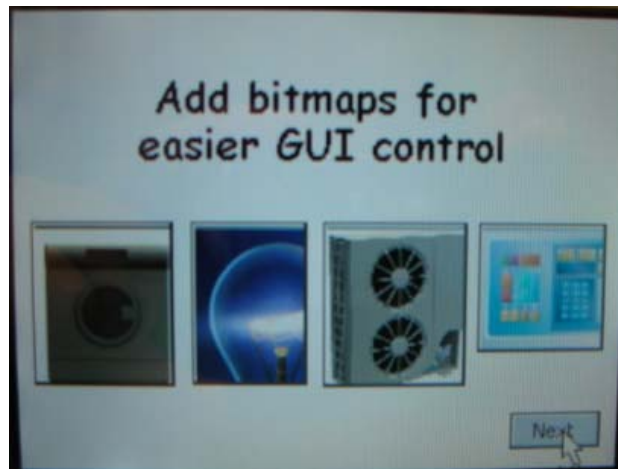


Figure 4-5 Bitmap control demo

4.2 Ending the demo

To end the demo, remove the USB cable from the board.

CHAPTER 5 HARDWARE SETUP

The demonstration kit board incorporates the V850ES/JG2 high performance 32-bit CPU, the Epson LCD driver, the TFT colour LCD display and a number of leds, switches and connectors and the NL2432HC22-41K TFT colour LCD display module by NEC Electronics.

5.1 V850ES/JG2

The V850ES/JG2 is NEC Electronics' high performance 32-bit CPU suited for low power applications. The V850ES/JG2 features the V850ES 32-bit RISC core, 512Kbytes flash memory and 40Kbytes internal RAM. The V850ES/JG2 supports upto 84 input/output ports which also serve as alternative functions to the large feature list of peripherals, such as timers, ADC's, DAC's, serial comms, interrupts. The external bus feature is used to interface to the Epson LCD companion chip which drives the TFT LCD display.

The V850ES/JG2 is driven by an external 5MHz clock which is PLL'ed up to 20MHz.

A 32.768kHz crystal is also connected to the sub-clock inputs to the V850.

For more information, please refer to document listed at the end of Chapter 3.

5.2 LEDs

The red led shows that power is applied to the board.

A yellow and green led are connected to CPU port pin P97 and P98 respectively and can be used as status lights by the demo software.

5.3 Switches

The following switches are available on the board.

Switch	Function
SW1	CPU Reset
SW2	8way DIP used to ground ADC inputs ANI5 to ANI11 if unused
SW3	Input to V850ES P03/INTP0
SW4	Input to V850ES P04/INTP1

Table 7 Switches on the Touch It! Board

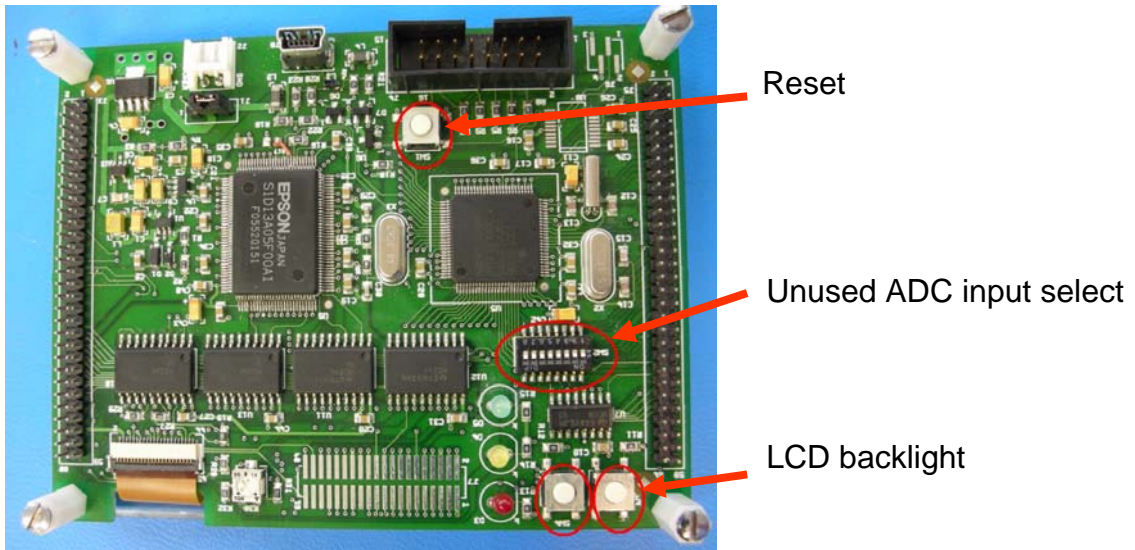


Figure 5-1 Position of switches on Touch It! circuit board

5.4 Connectors

Connector	Function	Comments
J1	5v select	Close pins 2+3 for USB power Close pins 1+2 for power from Conn Socket J2
J2	Connector for 5V power	Close pins 1+2 on J1 to use this
J3	Test pins to CPU pins 1 to 50	
J4	MINICUBE2 and PG-FP4 connector	To program the cpu flash memory
J5	Test pins to CPU pins 51 to 100	
J6	Conn Socket 4	Pin1 RS232 Rx Pin2 RS232 Tx
J7	Conn Socket 20 x 2	LCD signals from CPU
J8	Conn PCB 4	USB connector – can provide 5v supply to stater kit if J1 pins 2+3 closed
J9	Hirose FP23 45 way	TFT connector for LCD ribbon connection

Table 8 Connectors on TFT V850ES board

5.5 Epson S1D13A05 LCD/USB companion chip

The S1D13A05 is a LCD graphics controller that provides the interface between the TFT colour LCD display and the V850ES microcontroller. The S1S13A05 provides a 2-D Hardware Acceleration Engine to improve the drawing functions and includes a SRAM display buffer of 256Kbytes. The S1D13A05 also incorporates a USB controller, but this is not discussed in this manual.

The 20Mhz CLKOUT signal from the V850ES is supplied to the S1D13A05.

A 48Mhz clock is available to the Epson S1D13A05 device, but is not used on the Touch

The S1D13A05 supports a “SwivelView” mode which allows the display to be rotated 0°,90°,180° and 270° on the screen.

Normally the top edge of the TFT colour LCD display is when the flexible connector is orientated on the lefthand side of the display. The demonstration kit uses the Epson LCD driver in SwivelView so that the display is rotated 90° making the edge with the flexible connector the top edge. (refer to LCDConf.h)

The S1D13A05 device drives the LCD display in 16bits per pixel (bpp) mode so approximately 65000 different colours are achievable.

The initialisation functions and parameters are handled by the emWin library functions from Segger Microcontroller Systeme. Please refer to the software section and Segger documentation for further details.

For further information, please refer to the Epson documents listed above or visit www.erd.epson.com

5.6 TFT Colour LCD display

The NL2432HC22-41K TFT colour LCD display module is a 240(h) x 320(v), 3.5” amorphous-silicon (a-Si) TFT active matrix LCD panel with touch panel and a back light. (The normal aspect of the screen is in portrait mode with the flexible connector ribbon on the left side of the panel).

The LCD display utilises [NEC LCD Technologies’ Super Reflective NLT](#) technology to illuminate the display.

Although the display is capable of over 250,000 colours (6 bits per colour), approx 65000 colours are available on this demonstration kit as the Epson S1D13A05 lcd companion IC drives the LCD panel via a 18bit interface.

The demonstration software utilising the Segger emWin graphics library uses 565 colour mode. Images and graphics in the software design are handled in 16bit-mode, i.e. 5green, 6red, 5blue. The colour index is translated to 18-bits before writing to the Epson LCD device.

Please refer to Chapter 13 “Colour Index” for further information.

In order to run the demos in landscape mode, the Epson IC provides a “swivelmode” that can rotate the image round 90°, 180° or 270° counterclockwise. Swivel mode is enabled by setting the relevant values in the registers of the Epson device. Chapter 15 shows the Epson configuration tools that enables the correct register configuration to be derived. This configuration tool can be obtained directly from Epson.

5.7 TouchScreen panel

The touchscreen panel is an integral part of the NL2432HC22-41K TFT Colour LCD display. It consists of a resistive array for the vertical axis and another for the horizontal axis. The resistive array is fed into the V850's 10-bit ADC inputs. The inputs are analysed by software to determine if the screen has been pressed or tapped to display a cursor over a position on the screen or show a button being clicked before

For more details on the TFT colour LCD display with touchscreen panel, please refer to the TFT LCD datasheet listed in Chapter 3.

Caution: The Touch screen display flex ribbon connector is fragile. Take care not to bend or crease the flex ribbon to avoid damage.

Section 9.6.3 briefly describes how the touchscreen is setup to read each respective touchscreen axis.



Figure 5-2 TFT LCD Touch panel display

CHAPTER 6 CONNECTORS

6.1 TFT connector J9

The TFT colour LCD is connected to the board via the 45 way ribbon cable connector by Hirose (FH23-45S-0.3SHW).

For correct operation of the connector, please refer to the Hirose datasheet or visit the Hirose website <http://www.hirose.com> to access the datasheet e58613007.pdf

The ribbon cable is fragile, so please take care when handling the board and ensure that the ribbon is not creased or damaged.

Details of the pin connections can be found in the NEC Electronics TFT colour LCD manual.

6.2 5v power connector J2

A 5 volt source can be connected via J2. Before connecting power to the board, please refer to Chapter 4 for the correct jumper settings for J1.

6.3 USB connector J8

The USB connector is to provide an alternative 5 volt power source to the Touch It! Demonstration kit. Before connecting the USB cable, ensure the jumper settings are correct for J1 in Chapter 4.

The USB functionality can be provided by the Epson S1D13A05 LCD companion chip, but the USB driver is not available at present, therefore this function is not enabled for the Touch It! Demonstration kit.

6.4 Flash Programming connector J4

The 2 x 8 way connector J4 is available to connect flash programming equipment to the Touch It! Demonstration kit.

2 types of flash programming devices can be used.

- MINICUBE2 – (not included with Touch It! Demonstration kit)
- PG-FP4 Universal Flash programmer – (not included with Touch It! Demonstration kit)



Figure 6-1 PG-FP4 and MINICUBE2 programmers

The programming voltage for the V850ES Flash is supplied by the PG-FP4 or by the MINUCBE2 when SW2 is switched to '3', so it is not necessary to apply power via the USB connector or J2.

CHAPTER 7 SOFTWARE INSTALLATION

7.1 Touch It! Demonstration kit software installation

The pre-loaded introductory demonstration kit demo will run on the Touch It! Demo board when 5 volts is applied via the mini-USB connector.

To examine, edit additional software demos or recompile and run the standalone Segger example files, the demonstration kit software includes the EWV850-KS16 Kick Start version of the IAR V850 software development tools (limited to 16Kbytes application code).

7.2 PC Hardware requirements

A PC supporting Windows 98 or later is required for the IAR Systems workbench.

Pentium 166MHz (at least), 64MByte of RAM, 256-colour display (1024 * 768), mouse, CD-ROM drive and 40Mbytes of free hard disk space are required to install the tool packages.

A USB interface (version 1.1 or later) is required to support the programming devices.

7.3 Package contents

Please verify that you have received all the parts listed in the package contents list, document "README.pdf"

7.4 Software installation

The software installation CD will run automatically when inserted into the CD ROM drive.

Click on the required icon on the setup screen to install the documentation, tools and application examples.

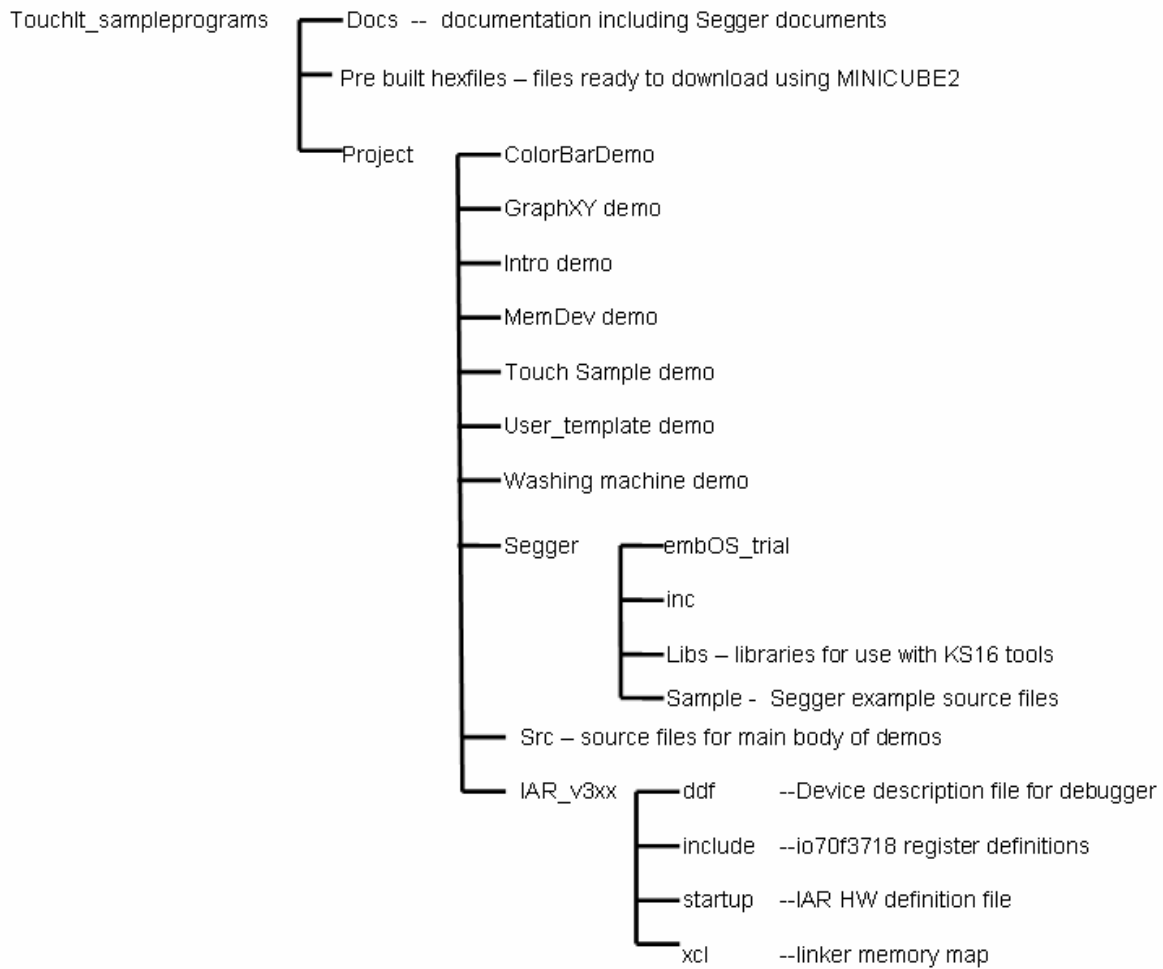


Figure 7-1 Software installation structure

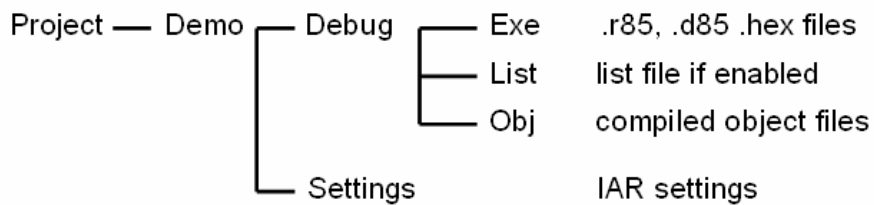


Figure 7-2 Structure of demo sub-directory

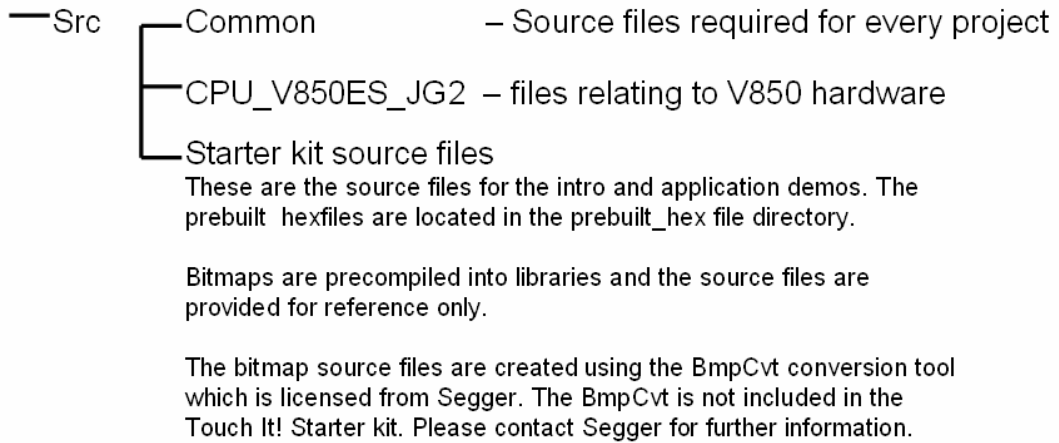


Figure 7-3 Structure of Src sub-directory

CHAPTER 8 TO RUN THE SUPPLEMENTARY DEMOS

8.1 Supplementary demos

Additional pre-built demos are provided with the Touch It! software installation under the \Prebuilt_hexfile directory.

To program these into the flash memory of the V850ES/JG2, you will need to obtain the MINICUBE2 Flash programmer. Please refer to Chapter 12 and associated documentation for additional information.

The supplementary demos consist of 1 application demo and the Segger emWin examples.

8.2 Application demo.

A washing machine application demo is provided on the software installation under the project directory Washingmachine\debug\exe. To view the code, start the IAR EWV850-KS16 tools and open the Touch It! project workspace and then select the TouchItDemo2 project.



Figure 8-1 Starting the Application Demo

8.3 Segger examples.

A sample of Segger examples for emWin are also included in the software installation which can also be compiled and run on the Touch It! demonstration kit.

Please note that not all Segger examples will run if they require other components such as Antialiasing, etc which are not included in this kit. Please contact Segger Microsysteme if you have these requirements for these additional options.

The additional project directories provided are:-

- GraphXY demo – demonstrates a windows based graphical interface
- Touch sample demo – shows the values of the touch screen relative to the display
- ColorBar demo – shows the colour capabilities of the display
- MemDev demo – uses special emWin library with Memory Devices option

The project workspace is already configured with a hex file to program into the demonstration kit.

If you wish to recompile the project from the IAR IDE, click on the “Make” icon, or select the menu option Project->Make or Project->Build All.

To try out other examples, you can use the User_template project which has been created to enable you to easily choose a Segger examples which are included in the directory
\\TouchIt_sampleprograms\\Project\\Segger\\Sample\\Tutorial

To add an example file, click and highlight the Src group in the User_template project. Then add the required file using the Project->Add option.

You can also keep the previous source file in the group but exclude it from the build by selecting Project->Options, then tick the “Exclude from Build” tick box. Next add the new file you wish to include for the compile. Using this method, you can use the group to hold all the source files of interest, but only keep one enabled for the compilation build.

Each example from the Segger tutorial demonstrates the different features of emWin, enabling you to pick the ones you require. The Touch It! demonstration kit provides you with the basic elements to develop a small prototype for your evaluation.

If you need to create a new project, follow the steps in Chapter 11.

CHAPTER 9 STRUCTURE OF THE DEMO CODE

9.1 Basic structure of software demo

This document is not intended to describe how to use embOS and emWin but will describe how the Touch It! Demonstration kit demo code is built on this and what is required to run the code.

To find out more how embOS and emWin works, please go to the Segger Microcontroller System website www.segger.com or contact Segger directly.

The diagram below shows the main software components of the Touch It! Demonstration.

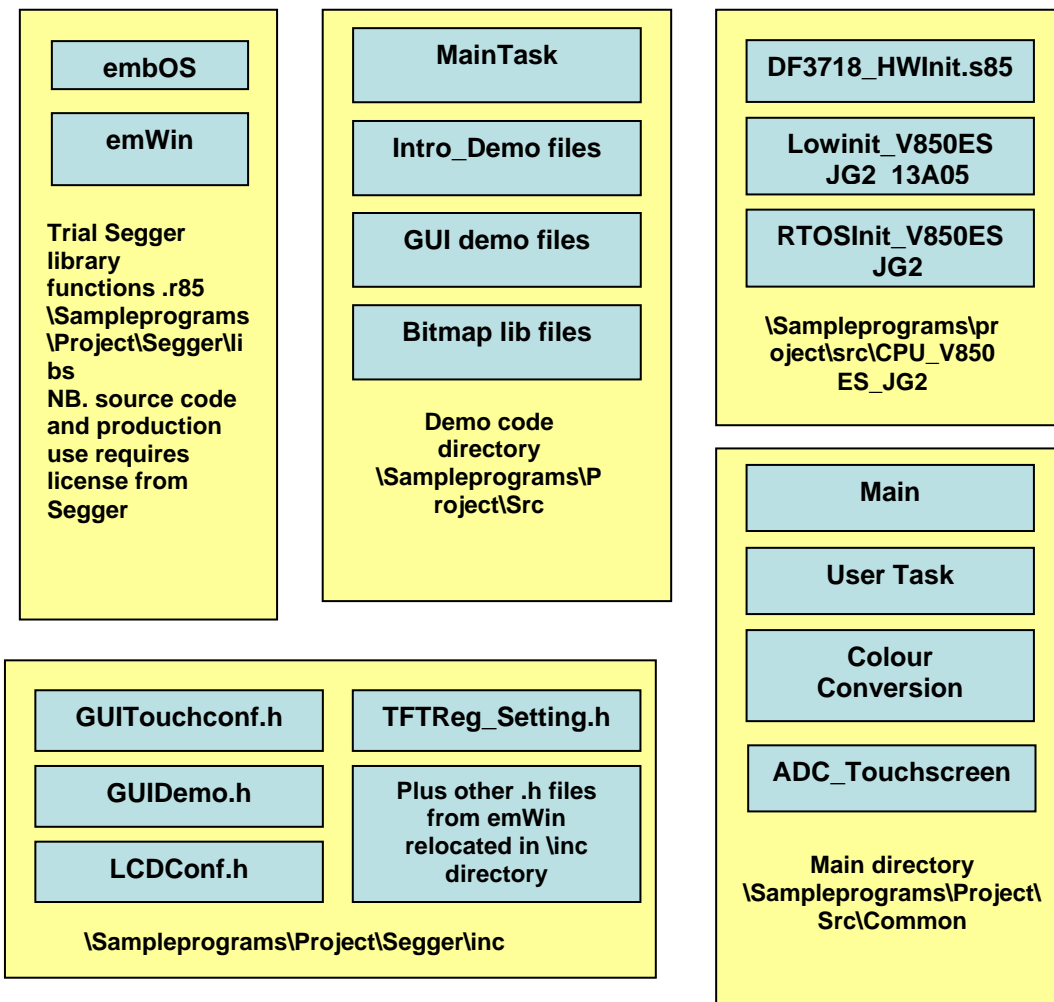


Figure 9-1 Organisation of software files

9.2 Segger embOS and emWin

The TFT Colour LCD demonstration kit software utilises trial library versions of embOS, the efficient Real Time Operating System (RTOS) and emWin graphics library from Segger Microcontroller Systeme GmbH.

These libraries are built to enable applications to be compiled using the special KS16 version of the IAR tools. (Please note that the regular library files in the trial download will not compile with the KS16 IAR tools, and require the full version of the IAR tools which require additional licensing.)

The emWin library included in the Touch It! Demonstration kit also contains trial optionals of Memory Devices, Window Manager and Widgets . If you need to evaluate the Segger products more closely, please contact Segger Microcontroller Systeme directly or visit their website to obtain further information.

Segger Microcontroller Systeme GmbH contact details can be found at:-

<http://www.segger.com/about.html>

For development and production, the use of embOS and emWin products from Segger Microcontroller Systeme GmbH require licensing. For pricing and licensing details, please contact Segger Microcontroller Systeme directly to discuss this.

The trial version of embOS v3.32m is used in this demonstration kit. embOS is included as a trial library and limited to 3 tasks and used as a compiled library specific to the V850ES cpu and IAR compilers.

The embOS library memory model used is shown in Table 9 below

This trial version of embOS for the V850 using the IAR tools can also be formally downloaded from the Segger website <http://www.segger.com/downloads.html>

The use of emWin in this demonstration kit is targeted with the Epson S1D13A05 LCD companion IC to drive the TFT colour LCD display.

Using a RTOS such as embOS enables the software application to be managed as tasks by the OS kernel. Tasks can be given a priority so that higher priority tasks always run before lower priority tasks. The OS also handles the timers and interrupts and a number of API's are provided to enable the application to use these facilities. More information can be found on the Segger website http://www.segger.com/embos_general.html or in the embOS documentation.

Using a graphics library such as emWin provides the software designer with a higher level of abstraction when developing an application. emWin handles all the low level functions required for displaying text, graphics, bitmap handling, widgets, and also functions to create a Windows based application if required. emWin handles the software drivers to the Epson S1D13A05 LCD companion chip and the TFT LCD display to make the development of your next project using the TFT LCD display easier.

In addition, the emWin product also provides support for the touchscreen panel, and enables the software development an easy way to develop touch and click applications on the TFT LCD panel.

For further details on emWin please refer to <http://www.segger.com/emwin.html>

This demonstration kit includes a number of software demos which shows you the potential possibilities you can use to develop your product into

The demo software is designed as individual tasks which runs on embOS and calls the various library functions provided by the emWin graphics library.

Both Segger embOS realtime OS and emWin graphics library are provided as compiled trial libraries. using the following memory models

Memory model	Setting
Short Addressing	On
Data Model	Tiny
Code Model	Normal

Table 9 Memory model settings

Both embOS and emWin libraries are incorporated into the IAR project just like a 'C' or 'assembler' source file in the project pane, but make sure that the same memory models are used if you decide to compile the entire demo code.

In the Touch It! Demonstration kit, embOS is used to manage 3 tasks. Tasks created using embOS are

- the main demo application task – MainTask created in main()
- Touch Screen task _TouchTask – created in TouchScreen_Init which is called by main ()
- User task (for user evaluation purposes)

Each tasks are is setup with a different priority level using the OS_Create API. A higher number in the priority field indicates a higher priority rating for the task.

To use the embOS and emWin features, the application demo code calls the various API functions.

For further details on the operation of embOS or emWin, please refer to the Segger manuals listed in Chapter 3.

The demo example for these applications are provided in this demonstration kit.

9.3 Demo code overview

The demo code has a basic structure called from main().

Main() performs the majority of the initialisation, setups the interrupt enabled before creating the MainTask, User Task, TouchScreen initialisation, before handing over to the OS by calling OS_Start().

Your application software design will follow a similar structure where the entry point for your code will be from MainTask(). You can see this by looking at the Segger examples.

Once the OS starts, the kernel manages the application by running each task in turn until OS_Delay() is called which suspends that task for the time frame required. The OS kernel decides the next task to run until OS_Delay() is called again. When the time period specified by OS_Delay is lapsed, the task is then made ready again to run at the next available sequence.

Essentially, the kernel manages the runtime between MainTask(), User Task and _TaskTouch().

The GUI, LCD and TouchScreen initialisation is handled by the emWin libraries when GUI_Init() is called and is configured according the the 'C' defines in GUIconf.h, LCDconf.h and GUITouchconf.h.

More details will be discussed in the section Configuration Files further down this chapter.

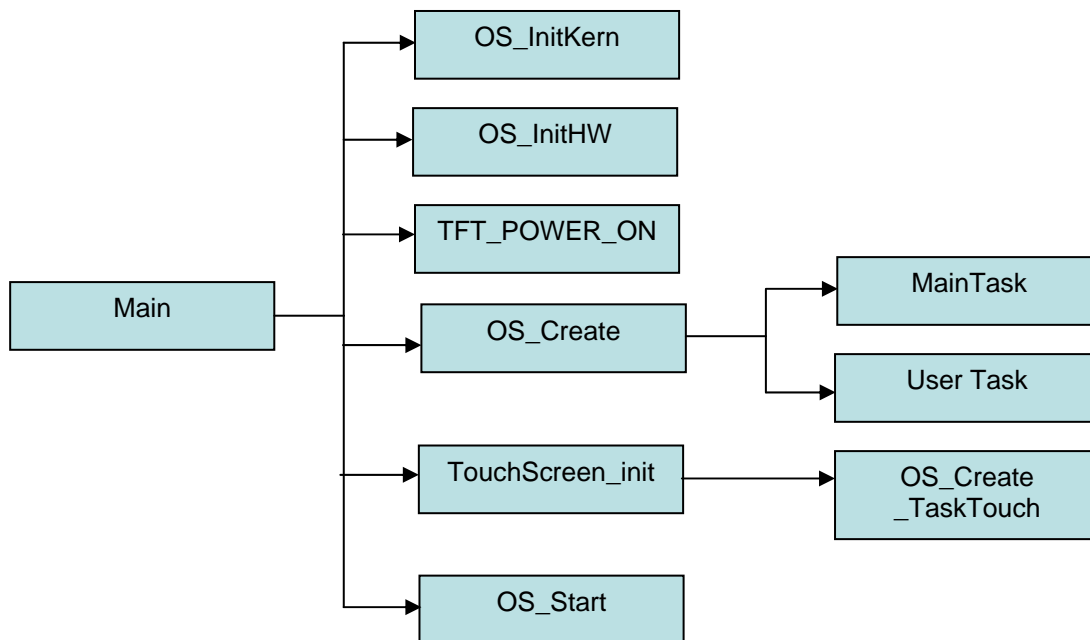


Figure 9-2 Overview of main()

9.4 Main

The functions contained in main perform the following:-

Disable the interrupt prior to hw and OS initialisation. The OS initialisation is performed by a function called `OS_INIT()` in the `RTOSInit_V850ES_JG2.c` file. (`RTOSInit_V850ES_JG2_no_uart.c` is used in the examples so that you may also use the optional MINICUBE2 device).

Create the `MainTask()` task and `User_task()`. `User_task()` is set to a lower priority than `MainTask()`.

Next the touchscreen initialisation routine `TouchScreen_Init()` in `ADC_Touchscreen.c` is called which initialises the ADC inputs used for the touchscreen and also creates the touchscreen task() which is set to a high priority level than `MainTask()`.

Finally, `OS_Start()` is called to enable the embOS kernel to control the application and determine the first task to run.

For the purpose of the Touch It! Demonstration kit, the first task to run will be the TouchScreen task i.e. `_TouchTask()` as this has the highest priority. Once `_TouchTask()` calls the `OS_Delay` function, `_TouchTask()` is suspended for the duration of the delay and control is returned to the embOS kernel. The kernel assesses which task has the next highest priority, in this case, `MainTask()`.

`MainTask()` contains a `while(1)` loop which calls the main body of the demo. This file resides in the IAR project group `Src`. Each demo routine will call an `OS_Delay` or an OS touchscreen key waiting routine. When these OS functions are called, the active task is suspended for the duration of the delay or until a touch screen click has been detected and control is returned to the OS. During these delays, embOS determines if another task which is ready to run. In this case, this will either be the `User_Task()` or `_TouchTask()`.

`User_Task()` runs until it calls `OS_Delay()` and suspends and returns control back to the embOS kernel. If all tasks are now waiting for their `OS_Delay` to expire and are not ready to run, the kernel calls `OS_Idle` until `OS_Delay` expires for one of the tasks which then becomes ready run.

For further information on how embOS operates, please refer to Chapters 2, 3 and 4 in the `embOS_Generic.pdf` users manual.

9.4.1 Initialising the V850

The V850ES is normally initialised in the startup code sequence as described above before any OS code has been run. This is achieved by the `_cstart_low_level_init` function in the file `Lowinit_V850ES_JG2_13A05.c`. Timer P1 is setup to be a PWM output to control the TFT backlight and initially set to a frequency of 2KHz and 100% duty cycle. The dimming control is handled in the TouchScreen task.

9.4.2 Initialising the TFT display registers

Next the TFT LCD hardware registers are configured by calling a macro function “TFT_POWER_ON_COMMAND” (source found in TFTReg_Setting.h). The sequence of data and timeouts required when setting up the TFT registers are detailed in the NECEL TFT LCD user manual.

9.4.3 Initialising the Epson S1D13A05 LCD companion chip

The LCD companion chip is initialised when LCD_Init() is called by the function GUI_Init() in emWin library. The function GUI_Init() in turn is called at the beginning of the first task to use the LCD, in this case, MainTask().

9.4.3.1 LCDLIN.c

This file provides the specific drivers to the Epson LCD companion chip. If a different LCD driver is to be considered in your project, this file will need to be updated before the emWin libraries are able to output to the TFT colour LCD display. The Segger emWin product is able to support other different LCD drivers, but these are not included in the Touch It! Demonstration kit. Please contact Segger for further information.

9.5 OS_Init

In addition to calling the OS_Initkern() function, OS_init() is also called near the start of main() to perform the low level initialisation for the clocks and serial comms ports used by the kernel and OS debug functions. Please refer to the Chapter 17 in the Segger embOS_Generic.pdf for further information.

9.5.1 Setting up the LCD touch screen header files

9.5.1.1 LCDconf.h

In order for the emWin to drive the Epson S1D13A05 LCD companion chip correctly, it needs to be informed which LCD driver chip is being used, the parameters of the LCD screen and the number of bits for the colour resolution per pixel. For further information, please refer to Chapter 27 emWinUser.pdf

```
#define LCD_XSIZE    320 /* X-resolution of LCD, Logical coord. */
#define LCD_YSIZE    240 /* Y-resolution of LCD, Logical coord. */

#define LCD_BITSPERPIXEL 16

#define LCD_CONTROLLER 1304 //
#define LCD_SWAP_BYTE_ORDER 1
#define LCD_SWAP_RB    0

#define LCD_USE_BITBLT    0

#define LCD_FIXEDPALETTE -1 // to enable use of User colour conversion function
```

The next section contains the register values derived from the Epson configuration setup which are used in the initialisation of the device. Please refer to Chapter 15

The final part of the file is the LCD_INIT_CONTROLLER() macro called by the GUI_init and LCD_init functions once the MainTask has started.

Please refer to Chapter 24 of the emWinUser.pdf regarding “**Additional info for the S1D13A05**”.

9.5.2 Guiconf.h

This file defines the GUI features to enable for emWin. For the Touch It! Demonstration kit, GUI support with the OS, touchscreen, window manager and memory device library are enabled.

9.5.3 GUITouchConf.h

This file is required when the emWin library is built and setups up the touchscreen ADC limits inputs for the V850ES to use.

e.g.

```
#define GUI_TOUCH_AD_TOP      330
#define GUI_TOUCH_AD_BOTTOM  890
#define GUI_TOUCH_AD_LEFT    110
#define GUI_TOUCH_AD_RIGHT   760
```

(please note: the values above may differ from the actual numbers required for the hardware)

This file also includes flags to determine if the X and Y axis need to be swapped or mirrored.

The Touch It! Demonstration kit is configured so that the top edge is the TFT side with the FPC flexible connector strip.

9.6 Creating tasks

The Segger trial version of embOS is provided and is limited to a maximum of 3 tasks. The Touch It! demonstration kit utilises 3 tasks which are described below. At least 1 task must be created before OS_Start() is called start running embOS. Tasks are created using the embOS OS_CREATETASK(). Please refer to the embOS_Generic.pdf user manual for further details.

9.6.1 MainTask

This task is created in `main()`. `MainTask()` forms the framework for the Touch It! Demos which are described in the next section 9.7

When you wish to examine other examples in the Segger sample routines, these files also contains a `MainTask()` function. As a result, you can substitute these functions into the demos for further evaluation.

9.6.2 User task

An empty user task is created in `main()` and set at a lower priority than `MainTask()` and is executed every 10mS. This user task is provided for evaluation purposes.

9.6.3 Touch Screen task

The emWin graphics library supports various means of user inputs via keyboard, joysticks and pointing devices. For the purpose of this demonstration kit, only the touch screen functions will be considered.

emWin provides full support for the touchscreen function from reading the touchscreen inputs, calculating the position on the LCD screen to the routines and functions to position a cursor or determine that a button or icon has been pressed or clicked.

As all other functions, the touchscreen also has to be initialised. The default values representing the minimum and maximum ADC readings need to be defined in the `GUITouchconf.h` include file.

These values are used by the emWin library function `GUI_Touch_Driveranalog` (note: this source file is not available in this demonstration kit. Please contact Segger Microsysteme for further details).

To initialise the touchscreen, the function `main()` calls the function `TouchScreen_Init()` in `ADC_Touchscreen.c`.

`TouchScreen_Init()` creates `_TouchTask()` which is setup to be activated every 10mS. `_TouchTask()` enables the emWin `GUI_Touch_Exec()` to be called to service the touch screen functions.

The basic circuit for the touchpanel is shown below:-

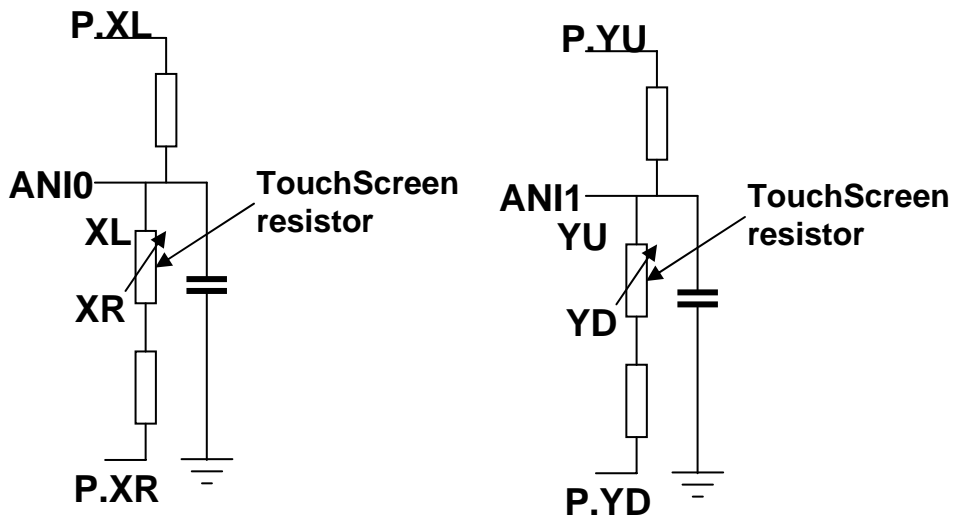
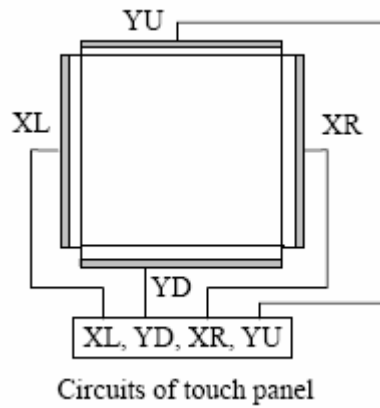


Figure 9-3 Touch Screen circuit

To read the X-axis value of the touchscreen on the V850ES ADC input ANI0, the cpu port lines connected to the X-axis P.XL and P.XR are place in high-impedence input mode and a voltage is applied to the resistive array on the Y-axis by setting the cpu port lines connected to P.YU high in output mode and setting P.YD low in output mode.

To read the value of the Y-axis, the inverse operation is used.

These are handled by the functions in ADC_TouchScreen.c via emWin library call GUI_Touch_Exec.

	Port.XL	Port.XR	Port.YD	Port.YU
To read X axis ANI0	Set port to input	Set port to input	Set port to low ouput	Set port to high output
To read Y axis ANI1	Set port to high output	Set port to low output	Set port to input	Set port to input

Once the touchscreen has been initialised, the emWin touchscreen API calls can be used to detect where the touchscreen is pressed or a button drawn on the display has been clicked, e.g. `GUI_Getket()` or `GUI_TOUCH_Getstate()`.

The LCD backlight dimming is also handled in the Touch Screen task which reads the 2 user switches SW3 and SW4 to change the brightness of the TFT backlight. The PWM duty cycle is changed by writing a new value to the V850ES register TP1CCR1.

9.6.4 Touchscreen Calibration

Although default values for the touchscreen are defined in `GUI_Touchconf.h`, i.e. the minimum and maximum ADC values for the X and Y axis, there will be slight variations due to tolerancing from one touchscreen to another.

As a result, at some point in time, there will be a need to run a calibration program. The Segger emWin library provides such a function and this is included in the demonstration software.

An example calibration demo is included in the default demo program.

9.7 General overview of introduction demo program

Once the hardware, OS and touchscreen have all been initialised in `main()`, the demo calls `MainTask()` in `maintask.c`.

Firstly, the emWin `GUI_Init()` function is called to initialise the LCD and emWin data display structures before calling the main body of `GUIDEMO_main()` in the while (1) loop.

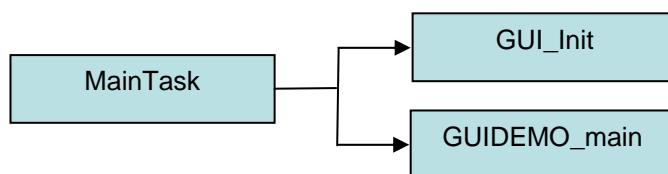


Figure 9-4 Structure of Segger example containing `Maintask()`

Each individual demo as seen in Chapter 4 is called in the for loop of `GUIDEMO_main()` from the array `_TouchIt_apfTest[]`.

The demo functions called are:- TouchIt_intro,

- NECEE_intro,
- TouchIt_Slide1,
- TouchIt_Slide2,
- Display_blockdiagram,
- Display_SeggerIAR_logo,
- TouchIt_Slide3,
- TouchIt_Slide4,
- Keyboard_Calibration,
- Display_Application_Bitmaps,

These demos show how text and values can be written to the display and how bitmap graphics can be shown and also used as bitmapped buttons for use by the touch panel.

The bitmap graphics are embedded in the 'C' code in the form of data array structures which are converted from native bitmaps using the BmpCvt conversion tools from Segger Microsysteme. Further information can be found in Chapter 10 of the emWinUser.pdf document.

Please contact Segger Microsysteme GmbH for more information about licensing the BmpCvt tool and other software products. <http://www.segger.com/software.html>

CHAPTER 10 STARTING THE IAR TOOLS

10.1 IAR tools

Start the IAR Embedded Workbench tools for NEC V850 by clicking on the Windows Start button:-

Start->All Programs->IAR Systems->IAR Embedded Workbench for NEC V850

or click on the icon on the Windows desktop.

To open the demo project workspace, click on File->Open->WorkSpace and then browse to the \Project directory and open the .EWW workspace file.

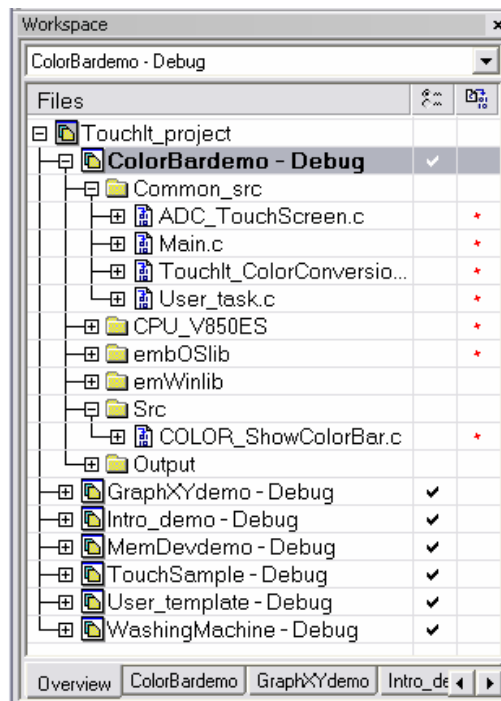


Figure 10-1 IAR project overview panel

10.2 Compiler and linker settings

Once the project is open, confirm the compiler and linker settings by clicking on Projects->Options to open the “Options” configuration window

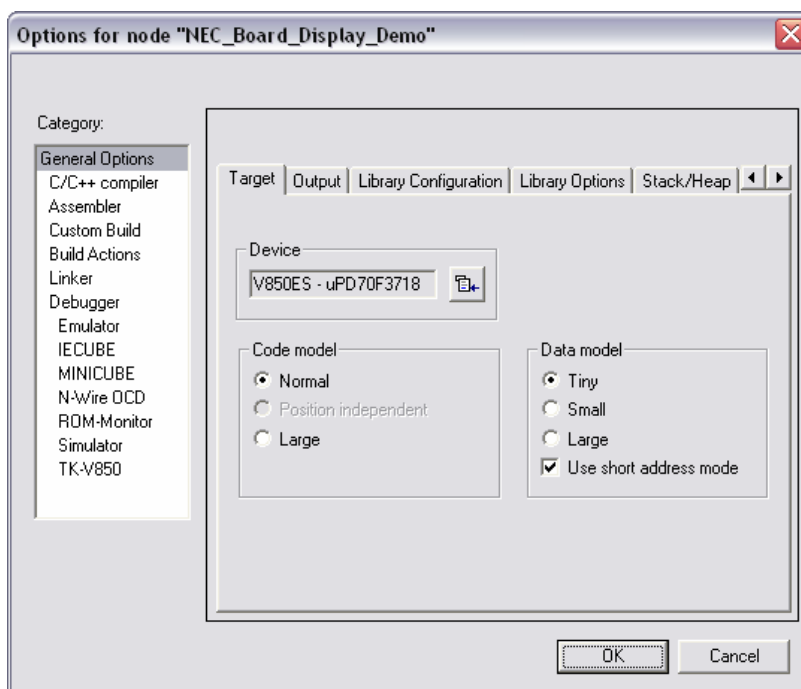


Figure 10-2 IAR Projects Option configuration window

This section lists the key settings for the compiler environment which need to be changed from the default setting. The options which are not listed are left as defaults.

Category	Option	
General Option	Target	Device = V850ES-uPD70F3718 Code Model = Normal Data Model = Tiny + Use Short Address Mode
	Output	Output = Executable
	Library Config	Full DLIB
C/C++ compiler	Optimizations	Optimizations=Size+Medium
C/C++ compiler	Preprocessor	\$PROJ_DIR\$\..\Segger\inc \$PROJ_DIR\$\..\Segger\inc\GUIConf or GUIConf_w_MemDev \$PROJ_DIR\$\..\IAR_v3xx\include \$PROJ_DIR\$\..\Src\IntroductionDemo \$PROJ_DIR\$\..\Segger\inc\GUI\Core \$PROJ_DIR\$\..\Segger\inc\GUI\JPEG

	Defines	\$PROJ_DIR\$..\Segger\inc\GUI\Widget \$PROJ_DIR\$..\Segger\inc\GUI\WM OS_LIBMODE_DP INIT_PLL Swivelview
Linker	Output Extra output	Allow C-Spy specific extra output file Generate extra output file Override default (if a different hex file name is required) Change the filename extension to .hex
	Diagnostics List Config	disable range checks Generate linker listing Override default xcl file \$PROJ_DIR\$..\IAR_v3xx\xc\DF3718_for_MINICUBE2_v2.XCL
	Processing	Fill unused code memory with 0xFF

Table 10 IAR compiler settings

10.3 Compiling a project

First, highlight a project by clicking on the project tab in the Project Window. To rebuild the project, click on Project->Make or Project->Clean and then Project->Rebuild All or click on the 'Make' icon.

If any project settings are changed, it is advantageous to clean and rebuild the entire project to ensure all object files are compiled with the up-to-date settings.

CHAPTER 11 TO SETUP A NEW IAR PROJECT

11.1 To create a new IAR project space

11.1.1 Adding project groups

Open the V850 IAR tools with the supplied "TouchIt_project.eww" workspace. You will see a panel with the tab "overview" on the bottom.

Click on the top PROJECT level to put it into context and then open the menu Project->Create New Project.

In the Create New Project Window, select V850 family + empty project and click "OK".

Next a Save As window will appear. Create a new folder using the "Create New Folder Icon" if required, browse to this and then provide a File Name before clicking "OK".

A new project will appear in the Overview panel.

Now, you need to configure the project space. First click on the + sign by an existing project to expand it. You can see the new groups you need to create.

Next you need to create the same groups as the other projects. To add a new group, right click on the project and scroll to Add->Group. Create the following groups

- Common_src
- CPU_V850ES
- embOSlib
- emWinlib
- Src

11.1.2 Adding files to the project groups

Next you need to add the source files for each group, add the files from the directories shown below.

- CPU_V850ES
 - \Sampleprograms\Project\Src\CPU_V850ES_JG2\DF3718_HWInit.s85
 - \Sampleprograms\Project\Src\CPU_V850ES_JG2\
Lowinit_V850ES_JG2_13A05.c
 - \Sampleprograms\Project\Src\CPU_V850ES_JG2\
RTOSInit_V850ES_JG2_no_UART.c
- embOSlib
 - \Sampleprograms\Project\Segger\libs\osESTNDP.r85
 - \Sampleprograms\Project\Segger\OS_Error.c
- emWinlib
 - \Sampleprograms\Project\Segger\libs\emWinlib_STN.r85
 - Or \Sampleprograms\Project\Segger\libs\emWinlib_MemDev_STN.r85
- Common_src
 - \Sampleprograms\Project\Src\Common\Main.c
 - \Sampleprograms\Project\Src\Common\TouchIt_ColorConversion.c
 - \Sampleprograms\Project\Src\Common\UserTask.c
 - \Sampleprograms\Project\Src\Common\ADC_Touchscreen.c
- Src
 - Add 'C' file from the Segger /Sample/Tutorial directory
 - For example
 \Sampleprograms\Project\Segger\Sample\Sample\Tutorial\2DGL_DrawPolygon.c
 - These examples contain the MainTask() function.

Next, click on the new project so that it is in context. Then open the menu Project->Options and set up the configuration according to the Chapter 10. Click OK when this is done.

CHAPTER 12 USING THE MINICUBE2 (OPTIONAL)



Figure 12-1 MINICUBE2

12.1 Installing the MINICUBE2 software

12.1.1 Upgrading the MINICUBE2 firmware

Before the MINICUBE2 can be used with the V850 CPU, the internal firmware needs to be upgraded if it is before version 4.04. This can be checked on the MINICUBE2 package label.

Download and install the latest MINICUBE2 firmware, MINICUBE2 utilities and the MQB2ALL USB drivers from the locations highlighted in table Table 6.

The MINICUBE2 diagnostics user's manual ZUD-CD-06-0044-4-E.pdf and MINICUBE2 On-chip debug checker user's manual ZUD-CD-06-0044-4-E.pdf is also included in the installation.

When you connect the MINICUBE2 to your PC for the first time via USB, you will be prompted to install a USB driver which can be found in the directory C:\Program Files\NEC Electronics Tools\mqb2all .

Next run the MINICUBE2 diagnostic checker from the Start->All Programs->NEC Electronics Tools->MINICUBE2 Utilities menu. Click on the Firmware tab and browse to the MINICUBE2_Firmware_Vxxx.hex file. Click on the Start button to commence the firmware update.

Refer to U18371EJ1V0UM00.pdf and ZUD-CD-06-0044-4-E.pdf for further details on MINICUBE2 firmware update.

12.1.2 Installing the QB2 MINICUBE2 Flash programming application

Please refer to U18527EJ1V0UM00.PDF QB-Programmer Programming GUI operation Manual.

To obtain the installation files for the MINICUBE2 QB-Programmer and device PRM files, goto <http://www.eu.necel.com/update> and

select QB-Programmer -> Programmer GUI for MINICUBE2

This is also highlighted in Table 6. Additional documentation can be obtained from the tools download website.

Before connecting the MINICUBE2, ensure the 2 switches are set to 3 (power provided by the MINICUBE2) and M2.

Extract and install the QBP installation program and PRM files from the zipfile.

12.2 Using the MINICUBE2 to programme the Touch It! board (optional)

The MINICUBE2 can be used as a flash programmer as well as an On-chip debug (OCD) device.

Before connecting the MINICUBE2, ensure the first switch is set to “M2”. Then set the 2nd switch to “3” to supply the programming voltage to the V850.



Figure 12-2 MINICUBE2 switch settings

To use the MINICUBE2 as a flash programmer, ensure the MINICUBE2 is connected to connector J4 on the Touch It! board using the 2x8 way cable and to the PC using the USB cable.

Ensure the MINICUBE2 USB driver is installed before starting the QBP program.

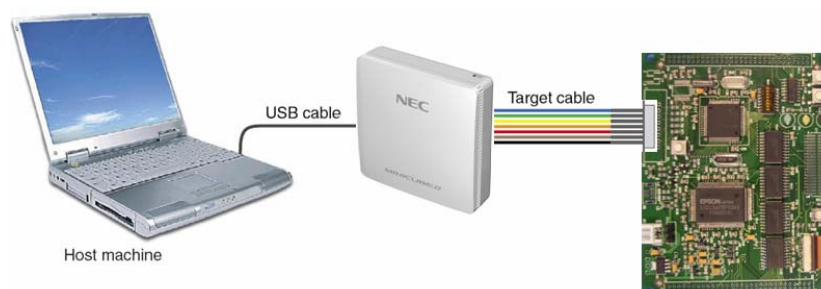


Figure 12-3 MINICUBE2 connections

To start the QBP program, goto Start->All Programs->NEC Electronics Tools->QBP2. Next, install the PRM file for the 70F3718 from the Device->Setup menu. The programmer is ready for use.

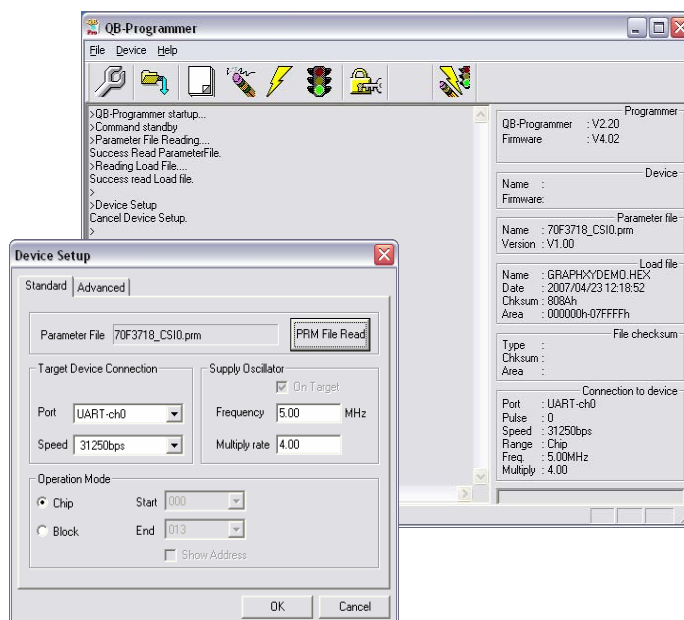


Figure 12-4 QB-Programming application and device setup

To program a hex file to the Touch It! demonstration kit:-

Firstly, select File->Load and browse to the required hex file from the following directories:-

- \Pre_built_hexfile
- \Project\Demo\Debug\Exe if you have recompiled a project

Next, select the Autoprocedure (EPV) Erase, Program and Verify option to start reprogramming process.

Please refer to U18527EJ1V0UM00.pdf QB-Programmer programming GUI operation manual for further details.

CHAPTER 13 DEBUGGING THE PROGRAM USING ON-CHIP DEBUG WITH THE MINICUBE2 (OPTIONAL)

If you have purchased a MINICUBE2 and wish to debug your code running on the Touch It! kit with the V850 IAR kickstart tools, you will need to use the MINICUBE2 using the On-chip debug facility. You will need to ensure the correct linker XCL file is used and UART is disabled in the code as described below before recompiling.

13.1.1 XCL Setup

The IAR linker uses an xcl file to define the memory segmentation for the application. When the MINICUBE2 is used for on-chip debug, it utilises 16 bytes of RAM and 2Kbytes of Flash ROM for ROM monitor purposes. For example, reserve the area between 0xFFFF5000 to 0xFFFF501F in RAM and 0x3F0 to 0x13EF in Flash ROM.

Use the file D3718_MINICUBE2_v3.xcl in the linker config options if the MINICUBE2 is to be used which reserves the memory spaces above for the MINICUBE2.

This file is located in the project \IAR_v3xx\xcl directory.

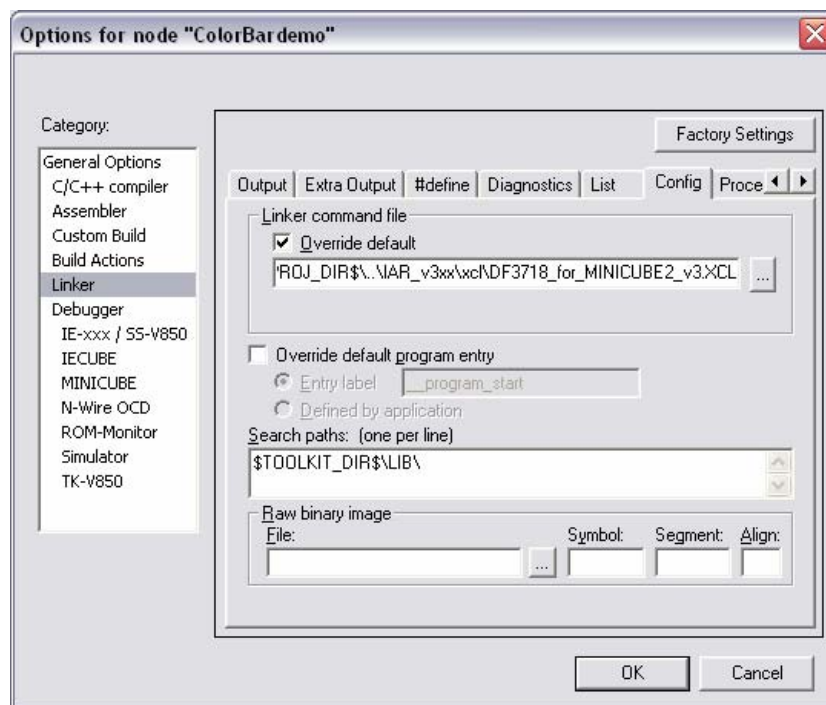


Figure 13-1 Reconfiguring the linker XCL file

13.1.2 Disable the UART

Also, as the MINICUBE2 uses the UARTA RX and TX on port3.0 and port3.1, the application should not use this peripheral during debug using the MINICUBE2. Also, any interrupt routines for the RX and TX channels must not be compiled into the code, otherwise this will conflict with the MINICUBE2.

The UART is normally used by the Segger embOS for the embOS viewer. To disable this feature, the #define OS_UART is set to 1 in the 'C' file RTOSInit_V850ES_JG2_no_UART.c.

13.1.3 Debug

Before you can start debugging, ensure that the MINICUBE2 USB driver is installed and the latest MINICUBE2 firmware is upgraded as described in Chapter 12

Connect the MINICUBE2 according to the manual U18371EJ1 pdf file

Ensure the IAR debug options are selected for the MINICUBE2

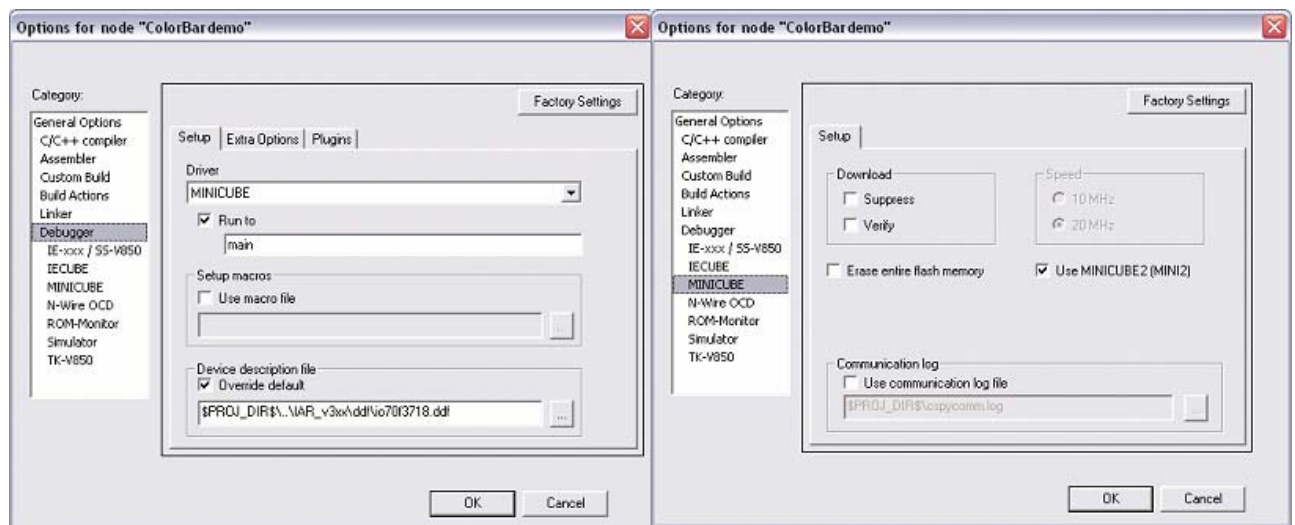


Figure 13-2 IAR Debug option setup for the MINICUBE2

CHAPTER 14 COLOUR INDEX

This section describes a circuit board layout error which is corrected by the software routine contained in the file Support_

14.1 Colour mapping between the LCD connector and board

The mapping for the colour signals for the TFT Colour LCD display can be found in Section 4.5 of the TFT colour LCD specification nl2432hc22-41k_DOD-MD-0058_1.pdf.

You can see that the colour signals include pins 13 to 30 in the order of blue, green, red.

These signals should correspond to the FPDATn signals from the Epson S1D13A05 LCD companion chip. Information relating to the signal mapping can be found in the following Epson document :-

Technical Manual s1d13a05tm.pdf -
Hardware Functional Specification -
Section 4.5 LCD Interface Pin Mapping

Refer to the information in column Color TFT Panel -> Generic TFT -> 18-bit

The correct pinning should be as follows:-

FP_DAT0 (Red5)	->	LCD pin 30 (Red5)
FP_DAT1 (Red4)	->	LCD pin 29 (Red4)
FP_DAT2 (Red3)	->	LCD pin 28 (Red3)
FP_DAT3 (Green5)	->	LCD pin 24 (Green5)
FP_DAT4 (Green4)	->	LCD pin 23 (Green4)
FP_DAT5 (Green3)	->	LCD pin 22 (Green3)
FP_DAT6 (Blue5)	->	LCD pin 18 (Blue5)
FP_DAT7 (Blue4)	->	LCD pin 17 (Blue4)
FP_DAT8 (Blue3)	->	LCD pin 16 (Blue3)
FP_DAT9 (Red2)	->	LCD pin 27 (Red2)
FP_DAT10 (Red1)	->	LCD pin 26 (Red1)
FP_DAT11 (Red0)	->	LCD pin 25 (Red0)
FP_DAT12 (Green2)	->	LCD pin 21 (Green2)
FP_DAT13 (Green1)	->	LCD pin 20 (Green1)
FP_DAT14 (Green0)	->	LCD pin 19 (Green0)
FP_DAT15 (Blue2)	->	LCD pin 15 (Blue2)
FP_DAT16 (Blue1)	->	LCD pin 14 (Blue1)
FP_DAT17 (Blue0)	->	LCD pin 13 (Blue0)

Table 11 Correct pinning from Epson S1D13A05 to TFT LCD connector

However, the board is pinned as follows where Red and Green are inverted:-

FP_DAT0 (Red5)	->	LCD pin 24 (Green5)
FP_DAT1 (Red4)	->	LCD pin 29 (Green4)
FP_DAT2 (Red3)	->	LCD pin 28 (Green3)
FP_DAT3 (Green5)	->	LCD pin 24 (Red5)
FP_DAT4 (Green4)	->	LCD pin 23 (Red4)
FP_DAT5 (Green3)	->	LCD pin 22 (Red3)
FP_DAT6 (Blue5)	->	LCD pin 18 (Blue5)
FP_DAT7 (Blue4)	->	LCD pin 17 (Blue4)
FP_DAT8 (Blue3)	->	LCD pin 16 (Blue3)
FP_DAT9 (Red2)	->	LCD pin 27 (Green2)
FP_DAT10 (Red1)	->	LCD pin 26 (Green1)
FP_DAT11 (Red0)	->	LCD pin 25 (Green0)
FP_DAT12 (Green2)	->	LCD pin 21 (Red2)
FP_DAT13 (Green1)	->	LCD pin 20 (Red1)
FP_DAT14 (Green0)	->	LCD pin 19 (Red0)
FP_DAT15 (Blue2)	->	LCD pin 15 (Blue2)
FP_DAT16 (Blue1)	->	LCD pin 14 (Blue1)
FP_DAT17 (Blue0)	->	LCD pin 13 (Blue0)

Table 12 Actual pinning from Epson S1D13A05 to TFT LCD connector

The Segger libraries provide the low level drivers to handle normal pinout as in Table 11, but when non-standard connections are used, alternative functions need to be used to handle the remapping of colour signals when writing the data to the Epson device. To achieve, this alternative functions

LCD_Color2Index_User and LCD_COLOR LCD_Index2Color_User which reinvert the red and green colour signals.

To enable the Segger libraries to use the custom functions, the 2 defines in LCDconf.h need to be set.

```
#define LCD_FIXEDPALETTE -1 and #define LCD_SWAP_BYTE_ORDER 1
```

The user functions can be found in the routine Support_070402_ColorConversion.c

CHAPTER 15 EPSON S1D13A05 CONFIGURATION SETTING

This chapter contains 5 screenshots showing the settings used for the Epson S1D13A05 initialisation. The tool is obtained from Epson and creates the correct values to use in the initialisation of the device. The output values were saved and transferred to the LCD_INIT_CONTROLLER macro in LCDconf.h

The configuration tool is described in more detail in the Epson document

Epson Document number:- X40A-Q-001-1 Filename:- Technical Manual s1d13a05tm.pdf

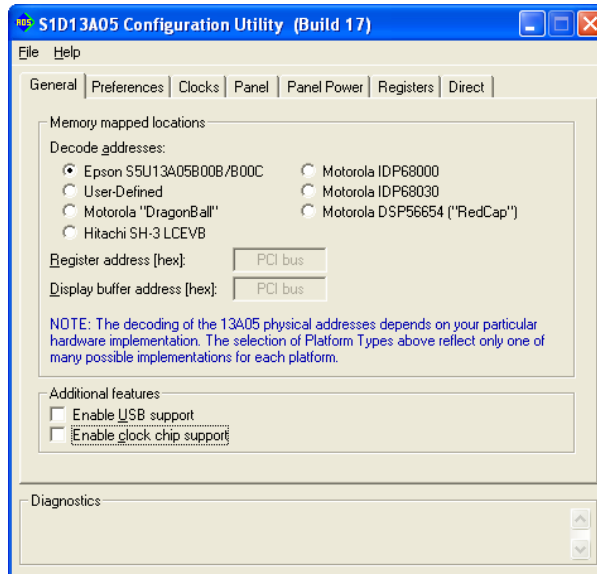


Figure 15-1 Epson general configuration

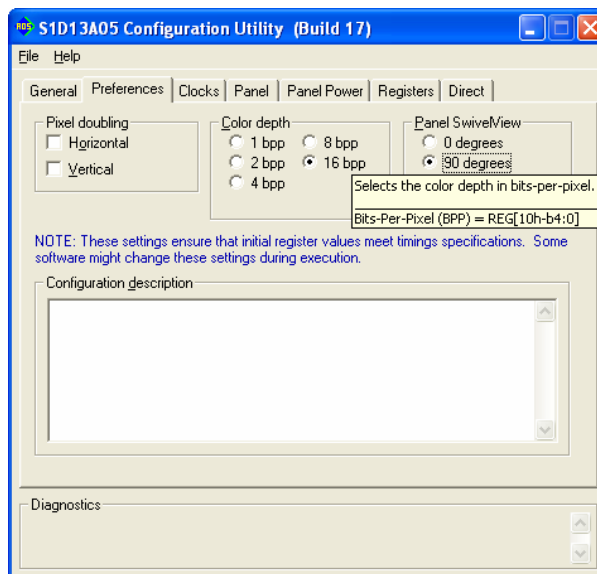


Figure 15-2 Epson preference configuration

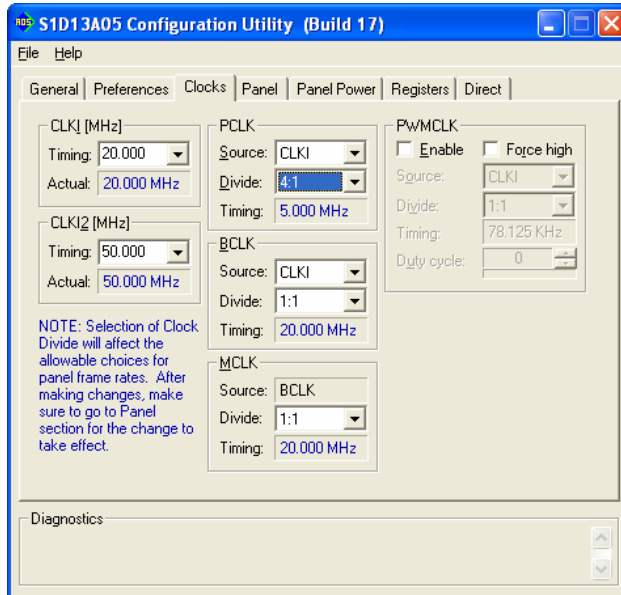


Figure 15-3 Epson clock configuration

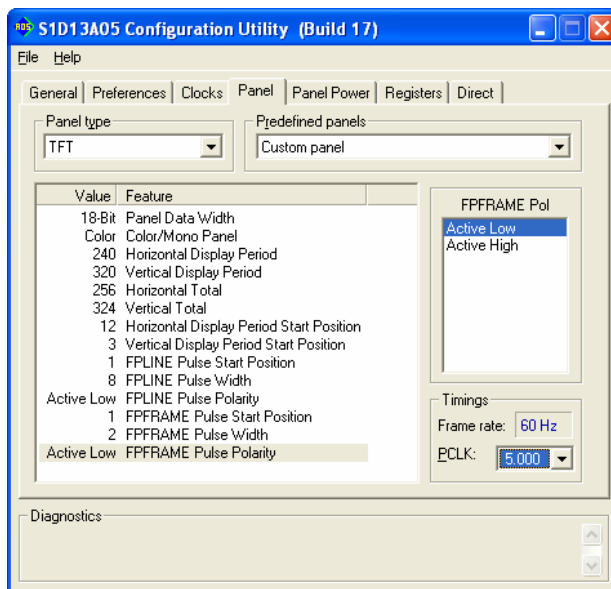


Figure 15-4 Epson panel configuration

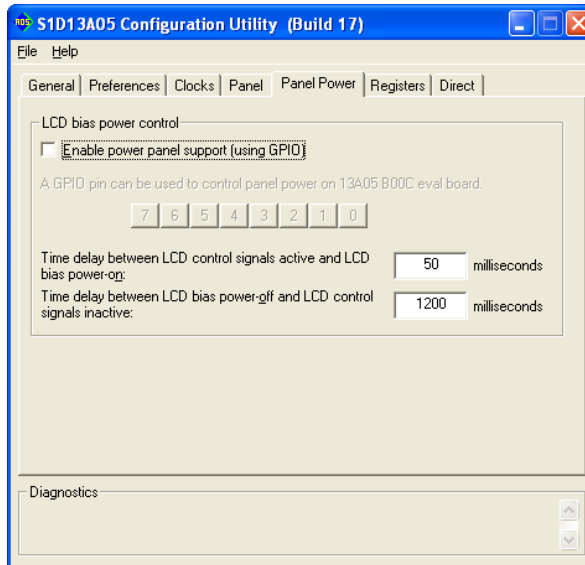


Figure 15-5 Epson panel power configuration

CHAPTER 16 BILL OF MATERIALS

This section lists the components used on the Touch It! Circuit board.

Table 13 Bill of Materials

No.	Quantity /per Board	Reference	Part	Part Type
1	4	C1,C8,C10,C11	4.7u	Type 'A' Tantalum TAJA475K010R
2	1	C2	0.68u	0805 Ceramic 25V minimum
3	2	C3,C4	2.2u	Type 'A' Tantalum TAJA225K010R
4	6	C5,C16,C17,C18,C19,C20,	100n	0805 Ceramic X/R
5	6	C21,C22,C23,C24,C25,C26,		
6	6	C32,C33,C34,C35,C36,C37,		
7	4	C38,C39,C40,C41		
8	6	C27,C28,C31,C44,C45,C46		
9	4	C6,C9,C42,C43	1u	Type 'A' Tantalum TAJA105K016R
10	1	C7	10n	0805 Ceramic B37941X1103K060
11	4	C12,C13,C14,C15	10p	0805 Ceramic B37930K5100J60
12	2	C29,C30	12p	0805 Ceramic 2238 867 15129
13	1	D1	DIODE SCHOTTKY	Small signal Schottky diode MBR0540
14	1	D2	DIODE, ZENER 22V	DIODE, ZENER 22V 5% SOD-323, Current max:200mA BZX384-C22
15	1	D3	LED	Red diffused LED,HLMP- 1700 2mA 3mm leaded
16	1	D4	LED	Yellow diffused LED,HLMP-1719 2mA 3mm leaded
17	1	D5	LED	Green diffused LED,HLMP1790 2mA 3mm leaded
18	2	D6,D7	BAV99/SOT	DIODE, DUAL SOT-23 BAV99
19	1	J1	HEADER 3	Pin header: 1raw 3 way 2mm pitch
20	1	J1	JUMPER SOCKET	2 Ways, Pitch:2mm

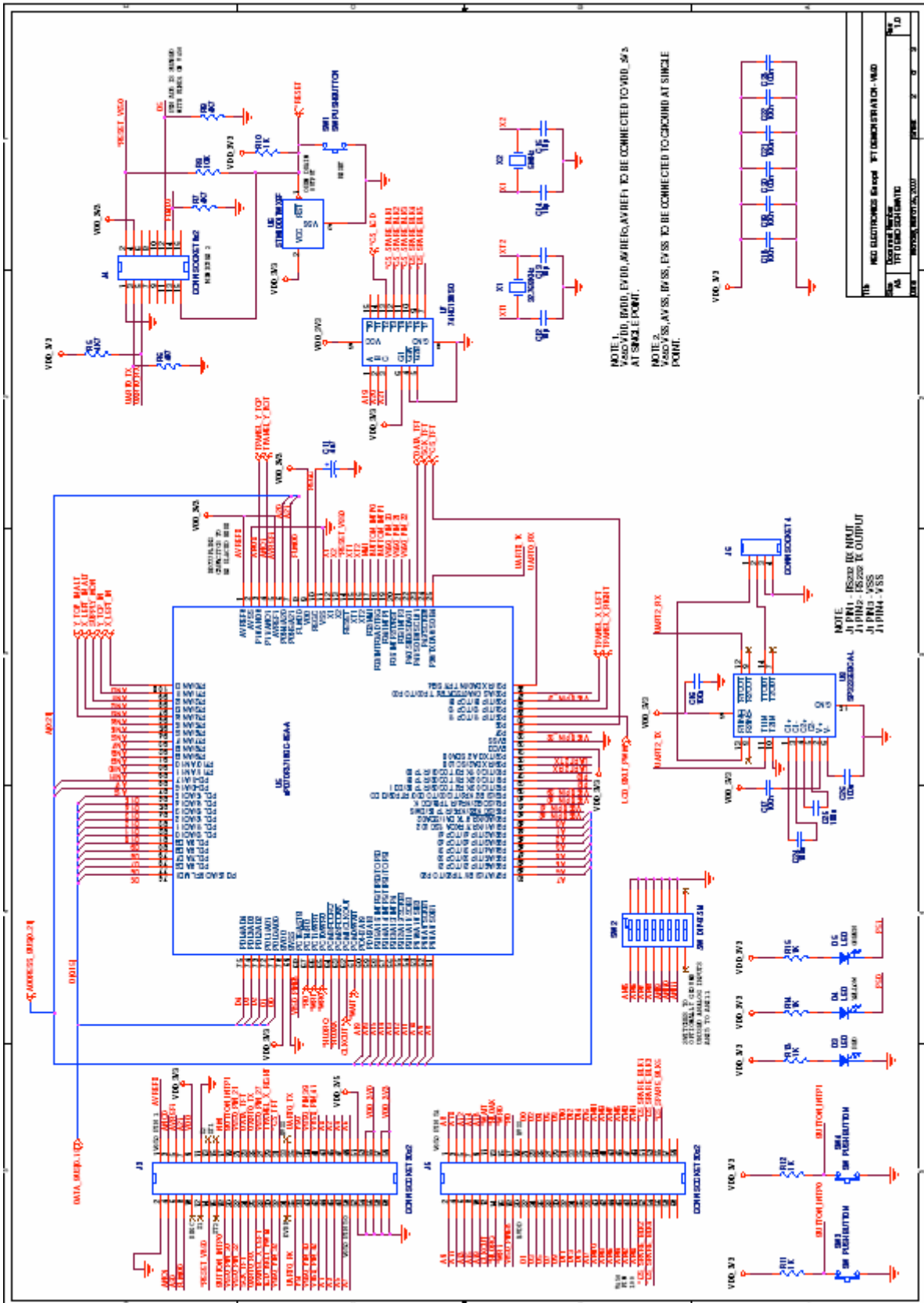
CHAPTER 16 BILL OF MATERIALS

21	1	J2	Connector,PCB,side entry header,2.5mm pitch,2 way	2way 1 row side entry header, S2B-EH (LF) (SN)
22	2	J3,J5	CONN SOCKET 20x2	expension connector 20wayx2row, 2.00mm pitch
23	1	J4	CONN SOCKET 8x2	Connector 8 way x 2 row vertical through hole header
24	1	J6	CONN SOCKET 4	HEADER, VERT 1 ROW 5 WAY; HARWIN M52-040000P0545
25	1	J7	CONN SOCKET 20x2	HEADER, VERT 2 ROW 40 WAY; HARWIN M52-040000S2045
26	1	J8	5-pin mini B Connector	USB mini B SMT Connector,rectangular, through hole, horizontall
27	1	J9	Hirose FP23 45 Way	FH23-45S-0.3SHW(05) Lead type FPC connector
28	2	L2,L4	INDUCTOR FERRITE/SM	Fair-Rite 2512066017Y1
29	1	L1	INDUCTOR, 1210 10UH	EPCOS B82422T1103K
30	1	L3	USB Common Mode Choke	Murata DLW21HN900SQ2
31	1	R1	500R	MC 0.1W 0805 1% 510R
32	1	R2	5R	MC 0.1W 0805 1% 4R7
33	3	R3,R4,R31	47K	MC 0.1W 0805 1% 47K
34	4	R5,R6,R7,R9	4K7	MC 0.1W 0805 1% 4K7
35	3	R8,R16,R17	10K	MC 0.1W 0805 1% 10K
36	6	R10,R11,R12,R13,R14,R15	1K	MC 0.1W 0805 1% 1K
37	1	R18	150K	MC 0.1W 0805 1% 150K
38	1	R19	100R	MC 0.1W 0805 1% 100R
39	1	R29	150R	MC 0.1W 0805 1% 150R
40	4	R20,R21,R27,R28	20R	MC 0.1W 0805 1% 20R
41	2	R22,R23	301K	MC 0.1W 0805 1% 300K
42	1	R24	1K5	MC 0.1W 0805 1% 1K5
43	1	R25	1M	MC 0.1W 0805 1% 1M
44	1	R26	470R	MC 0.1W 0805 1% 470R
45	1	R30	100K Lin	TS53YL 100K
46	1	R32	68K	MC 0.1W 0805 1% 68K
47	3	SW1,SW3,SW4	SW PUSHBUTTON	Std SMD tactile switch B3S-1000
48	1	SW2	SWITCH, DIL 1/2 PITCH SMD 8 WAY	Multicomp MCHDS-08-T
49	1	U1	IC, SM LED DRIVER ADJUSTABLE	Driver ZXLD1937ET5 ZETEX TSOT23-5 pin package

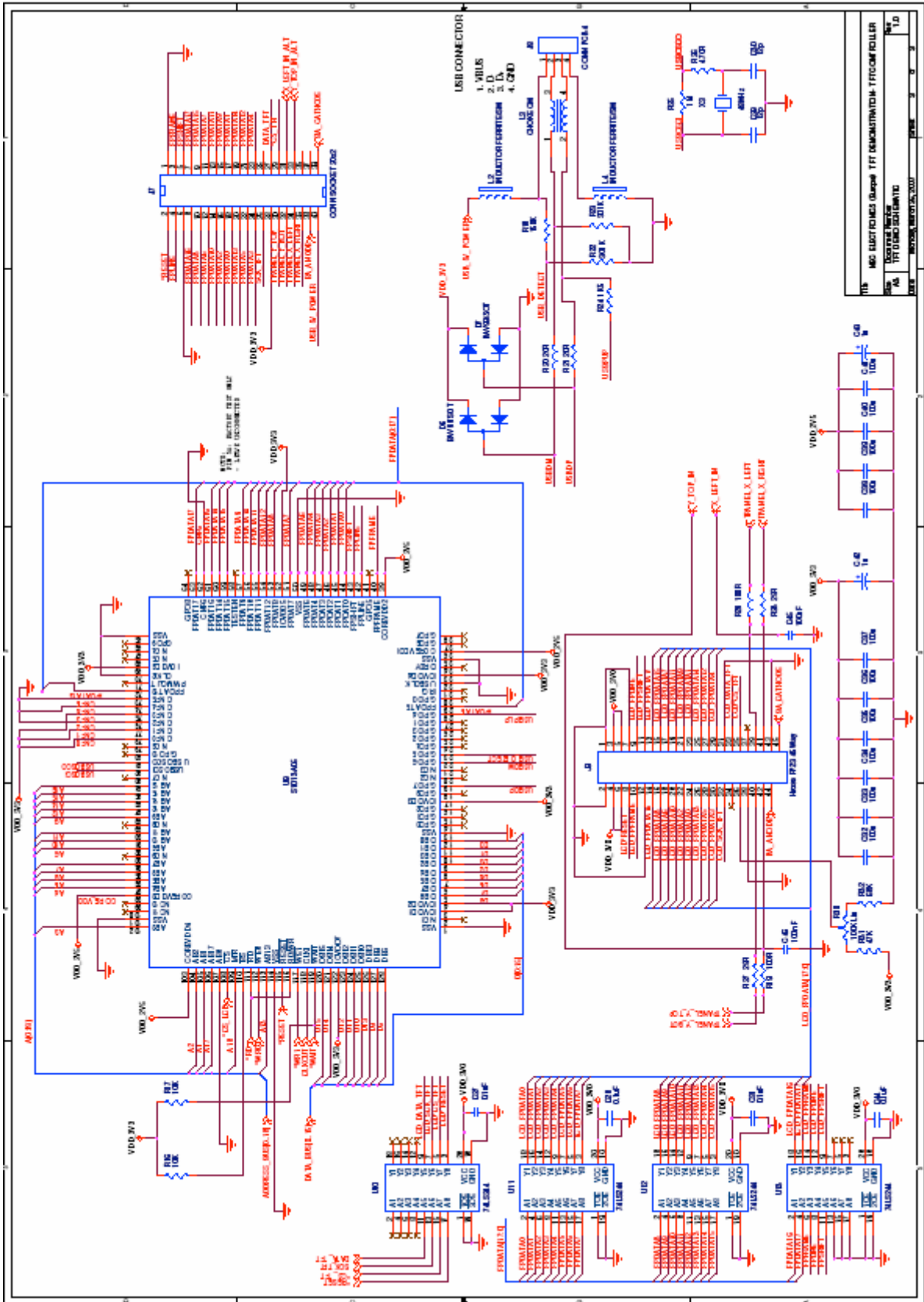
CHAPTER 16 BILL OF MATERIALS

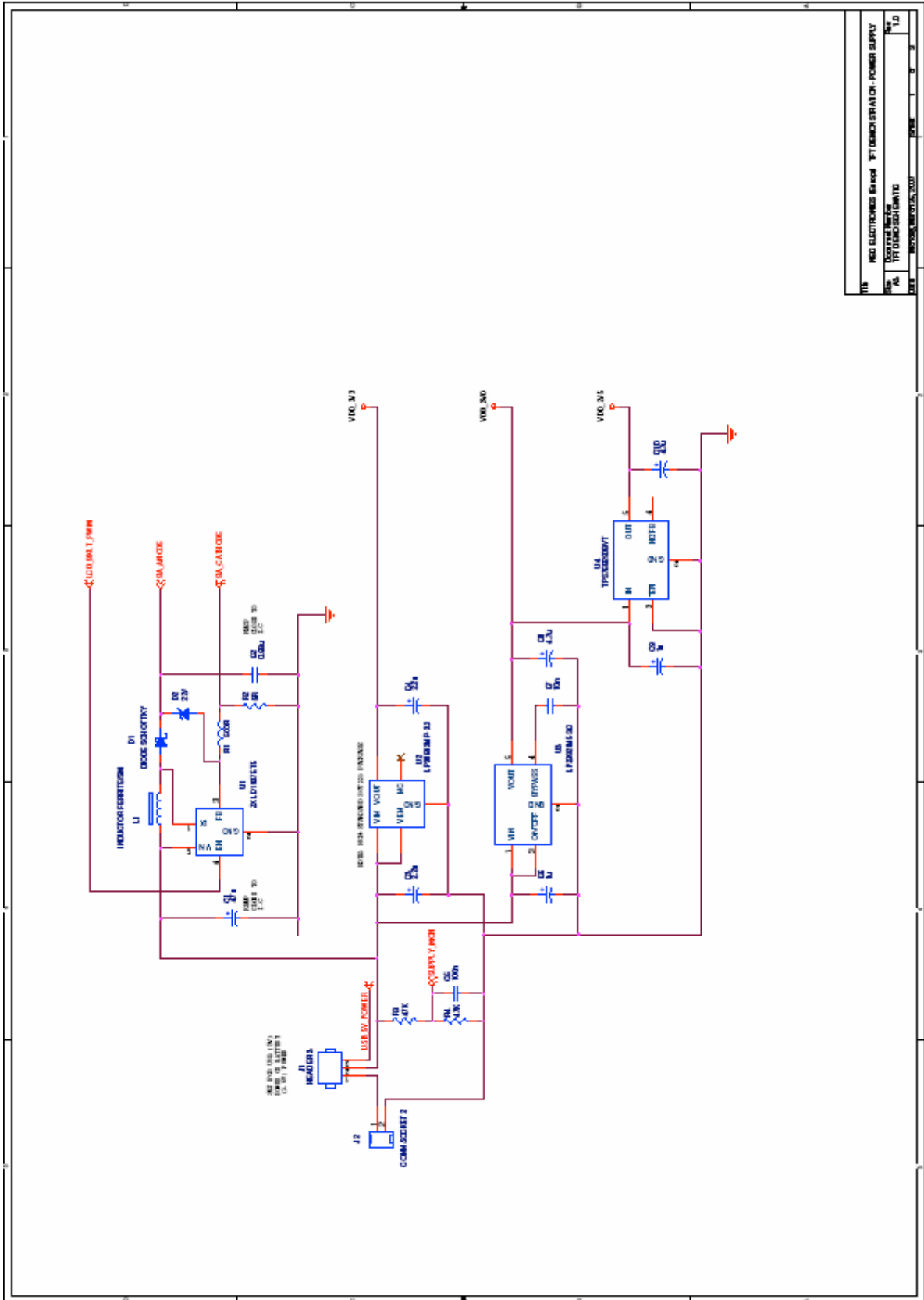
50	1	U2	IC, SM REGULATOR, 500MA LDO	LP38693MP-3.3 (0.5A) SOT223 (NON STANDARD PACKAGE)
52	1	U3	IC, SM REGULATOR LDO +3.0V	LP2982IM5-3.0 (50mA) 5-Lead Small Outline Package (M5)
54	1	U4	IC, SM LDO REG 0.1A 2.5V	TPS76925DBVT (0.1A)n DBV Package
56	1	U5	uPD70F3718GC-8EA-A	uPD70F3718GC-8EA-A 512KByte of flash memory
57	1	U6	IC, SM MPU LOW OD RESET 3.08V	STM1001TWX6F SOT23-3
58	1	U7	IC, SM LOGIC DECODER/DEMULTIPLEXER	74HC138/SO SO16
59	1	U8	IC, SM TRANSCEIVER RS-232 +3.0V TO +5.5V	SP3232EBCA-L SSOP16
60	1	U9	S1D13A05	S1D13A05 LCD Controller 128-pin QFP5 package
61	4	U10,U11,U12,U13	IC, SM LOGIC BUFFER AND LINE DRIVER	SN74HC244DW
62	1	X1	32.768KHz	NC38 watch crystal, through hole 32.768kHz 12.50pF
63	1	X2	Crystal, 5.000000MHZ	Citizen Crystal, through hole HC49US5.000MABJ-UB
64	1	X3	48MHz	Citizen Crystal, through hole HC49US48.000MABJ-UB
65	1	NEC TFT display	NL2432HC22-41K	3.5" NEC TFT Display

CHAPTER 17 SCHEMATICS



CHAPTER 17 SCHEMATICS





FILE	REG ELECTRONICS Group - PT DEMONSTRATION - POWER SUPPLY
DESIGNER	REG ELECTRONICS
DATE	1/11/2006 10:41:00 AM
DESCRIPTION	POWER SUPPLY
DATE	1/11/2006 10:41:00 AM