

User's Manual

78K0R - Save It!

Demonstration Kit

Document No. U19663EE1V0UM00 Date Published January 2009

© NEC Electronics (Europe) GmbH

- The information in this document is current as of June, 2008. 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 customerdesignated "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

CAUTION

This is a Test- and Measurement equipment with possibility to be significantly altered by user through hardware enhancements/modifications and/or test or application software. Thus, with respect to Council Directive 89/336/EEC (Directive on compliance with the EMC protection requirements), this equipment has no autonomous function. Consequently this equipment is not marked by the CE-symbol.

EEDT-ST-005-10

CAUTION

This equipment should be handled like a CMOS semiconductor device. The user must take all precautions to avoid build-up of static electricity while working with this equipment. All test and measurement tool including the workbench must be grounded. The user/operator must be grounded using the wrist strap. The connectors and/or device pins should not be touched with bare hands.

EEDT-ST-004-10



For customers in the European Union only

Redemption of Waste Electrical and Electronic Equipment (WEEE) in accordance with legal regulations applicable in the European Union only: This equipment (including all accessories) is not intended for household use. After use the equipment cannot be disposed of as household waste. NEC Electronics (Europe) GmbH offers to take back the equipment. All you need to do is register at http://www.eu.necel.com/weee

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 Inc. (U.S.)

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 03 0 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: 040-244 58 45 Fax: 040-244 45 80

Branch Sweden

Taeby, Sweden Tel: 08-63 80 820 Fax: 08-63 80 388

United Kingdom Branch

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

NEC Electronics Hong Kong Ltd. Hong Kong Tel: 2886-9318

Tel: 2886-9318 Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.

Seoul Branch Seoul, Korea Tel: 02-528-0303 Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.

Singapore Tel: 65-6253-8311 Fax: 65-6250-3583

NEC Electronics Taiwan Ltd.

Taipei, Taiwan Tel: 02-2719-2377 Fax: 02-2719-5951

NEC do Brasil S.A.

Electron Devices Division Guarulhos, Brasil Tel: 55-11-6465-6810 Fax: 55-11-6465-6829

Revision History

Date	Revision	Chapter	Description
2009-03-16	V1.00		First release

Table of Contents

1.	Introduction	10
1.1	Package contents	10
1.2	Features	
1.3	System requirements	
1.4	Trademarks	11
2.	78K0R – Save It! Components	12
2.1	Operation mode settings	12
2.2	SW1	
2.3	SW2	13
2.4	SW3	13
2.5	SW4 (Reset switch)	
2.6	SW5 (INTP0)	
2.7	SW6 (INTP2)	
2.8	JP1	
2.9	JP2	
2.10	Q4 (illuminance sensor)	
2.11	U10, U12 (7 segment LED)	
2.12	LED1 (power LED)	
2.13	Universal area	
2.13		-
	78K0R – Save It! system configuration	
3.1	78K0R – Save It!	
3.2	Host computer	
3.3	Power supply via USB interface	17
4.	78K0R – Save It! installation and operation	18
4.1	Getting started	18
4.2	CD-ROM contents	
5.	Hardware installation	19
6.	Software installation	20
6.1	TK-78K0RKE3L demonstration program installation	20
6.2	IAR Systems Embedded Workbench for 78K installation	
6.3	IAR Systems visualSTATE Installation Device file package of 78K0R/Kx3-L for IAR Embedded Workbench installation	22
6.4 6.5	WriteEZ4 installation	
	USB Driver Installation	
6.6		
6.6.1 6.6.2		
6.0.2	Confirmation of USB Driver Installation	
0.7		29

78K0R - Save It!

7.	Demo application using the TK-78K0R/KE3L demonstration program	30
8.	Applilet EZ PL	31
8.1	Overview	
8.2 8.3	Features Applilet EZ PL demo application	
9.	Flash Programmer WriteEZ4	33
9.		33
9.1	Device Setup	
9.2	Using WriteEZ4	. 34
10.	Erasing the flash memory (using the WriteEZ4)	35
11.	IAR Embedded Workbench 78K0R – Save It! sample project	36
11.1		
		. 36
11.2	Loading the Save It! sample project	. 36
11.2 11.3	Loading the Save It! sample project	. 36 . 41
11.2 11.3 11.3	 Loading the Save It! sample project Running the application Count mode 	. 36 . 41 . 41
11.2 11.3	 Loading the Save It! sample project Running the application Count mode Blink mode 	. 36 . 41 . 41 . 42
11.2 11.3 11.3 11.3	 Loading the Save It! sample project Running the application Count mode Blink mode 	36 41 41 42 42
11.2 11.3 11.3 11.3 11.3 11.3	Loading the Save It! sample project Running the application Count mode Blink mode Blink mode Spinning mode IAR visualSTATE 78KOR – Save It! sample project	36 41 41 42 42 42
11.2 11.3 11.3 11.3 11.3	Loading the Save It! sample project Running the application Count mode Blink mode Blink mode Spinning mode IAR visualSTATE 78KOR – Save It! sample project Hardware setup	• 36 • 41 • 41 • 42 • 42 • 42 • 43
 11.2 11.3 11.3 11.3 12.1 12.2 12.3 	Loading the Save It! sample project Running the application Count mode Blink mode Blink mode Spinning mode IAR visualSTATE 78KOR – Save It! sample project Hardware setup Functionality of the sample project Running the sample project	. 36 . 41 . 42 . 42 . 42 . 42 . 43 . 43 . 43 . 44
 11.2 11.3 11.3 11.3 12.1 12.2 12.3 	Loading the Save It! sample project Running the application Count mode Blink mode Blink mode Solution Spinning mode IAR visualSTATE 78KOR – Save It! sample project Hardware setup Functionality of the sample project	. 36 . 41 . 42 . 42 . 42 . 42 . 43 . 43 . 43 . 44
 11.2 11.3 11.3 11.3 12.1 12.2 12.3 	Loading the Save It! sample project Running the application Count mode Blink mode Blink mode Spinning mode IAR visualSTATE 78KOR – Save It! sample project Hardware setup Functionality of the sample project Running the sample project	. 36 . 41 . 42 . 42 . 42 . 42 . 43 . 43 . 43 . 44 . 46
 11.2 11.3 11.3 11.3 12.1 12.2 12.3 12.4 	Loading the Save It! sample project Running the application 1 Count mode 2 Blink mode 3 Spinning mode IAR visualSTATE 78KOR – Save It! sample project Hardware setup Functionality of the sample project Running the sample project Running the sample project Debug the 78KOR – Save It! visualSTATE sample project Cables	. 36 . 41 . 42 . 42 . 42 . 43 . 43 . 43 . 44 . 46 . 47

List of Figures

Figure 1: 78K0R – Save It! components	12
Figure 2: 78K0R – Save It! 7 segment LEDs structure	15
Figure 3: 78K0R – Save It! 7 segment LED port pin connection	15
Figure 4: 78K0R – Save It! MiniCube2 connector	16
Figure 5: 78K0R – Save It! Minicube2 connection solder bridges	16
Figure 6: 78K0R- Save It! system configuration	17
Figure 7: 78K0R – Save It! CD-ROM contents	
Figure 8: 78K0R – Save It! CRROM autorun.exe	20
Figure 9: IAR Systems Installation screen	
Figure 10: Found New Hardware Wizard (Windows 2000)	
Figure 11: Search Method (Windows 2000)	
Figure 12: Driver File Location (Windows 2000)	
Figure 13: Address Specification 1 (Windows 2000)	
Figure 14: Address Specification 2 (Windows 2000)	
Figure 15: Address Specification 3 (Windows 2000)	
Figure 16: Driver File Search (Windows 2000)	
Figure 17: USB Driver Installation Completion (Windows 2000)	
Figure 18: Found New Hardware Wizard 1 (Windows XP)	
Figure 19: Found New Hardware Wizard 2 (Windows XP)	
Figure 20: Search Location Specification 1 (Windows XP)	
Figure 21: Search Location Specification 2 (Windows XP)	28
Figure 22: Windows XP Logo Testing (Windows XP)	
Figure 23: USB Driver Installation Completion (Windows XP)	29
Figure 24: Windows Device Manager	29
Figure 25: TK-78K0RKE3L demonstration program	
Figure 26: Applilet EZ PL main window	
Figure 27: WriteEZ4 Startup	33
Figure 28: WriteEZ4 Device Setup Dialogue	
Figure 29: WriteEZ4 Device Menu	34
Figure 30: WriteEZ4 open .prm file	
Figure 31: WriteEZ4 flash erase successful	35
Figure 32: IAR Embedded Workbench Startup window	36
Figure 33: IAR Embedded Workbench Main IDE window	37
Figure 34: IAR Embedded Workbench Debugger Options	38
Figure 35: Embedded Workbench Linker Configuration	
Figure 36: IAR Embedded Workbench MINICUBE2 settings	40
Figure 37: IAR C-SPY debugger	41
Figure 38: IAR visualSTATE navigator	44
Figure 39: IAR visualSTATE designer	
Figure 40: IAR Embedded Workbench visualSTATE sample project	46
Figure 41: USB interface cable (Mini-B type)	47
Figure 42: TK-78K0RKE3L schematics 1/3	49
Figure 43: TK-78K0RKE3L schematics 2/3	50
Figure 44: TK-78K0RKE3L schematics 3/3	51

List of Tables

Table 1: 78K0R – Save It! Switch and Jumper settings1	2
Table 2: SW1 modes1	
Table 3: SW2 modes1	3
Table 4: SW3 modes	3
Table 5: JP1 settings1	4
Table 6: JP2 settings1	4
Table 7: 7 segment LED port values	5
Table 8: Applilet EZ PL demo Switch settings	2
Table 9: WriteEZ4 action buttons	4
Table 10: IAR EWB 78K0R – Save It! sample project hardware setup	6
Table 11: IAR VS 78K0R - Save It! sample project hardware setup	3

1. Introduction

The 78K0R - *Save It!* demonstration Kit shows the system development possibilities using a NEC low power consumption microcontroller of the 78K0R/Kx3-L microcontroller family.

A preprogrammed sample project, which is written to work with the **TK-78K0R demo** PC graphical user interface, provided on the CD-ROM, shows the current consumption in the different operating modes of the micro-controller.

Furthermore an IAR Embedded Workbench for 78K, IAR visualSTATE and Applilet EZ PL sample projects for the 78K0R/Kx3-L can be found on the CD-ROM.

1.1 Package contents

- TK-78K0R/KE3L board
- USB cable
- CD-ROM containing the TK78K0R/KE3L GUI, Applilet EZ PL, an evaluation copy of the IAR Embedded Workbench for 78K with 4Kbyte code size limitation and IAR visualSTATE

Please verify that you have received all parts listed in the package contents list attached to the *78K0R* - *Save It!* demonstration kit package. If any part is missing or seems to be damaged, please contact the dealer from whom you received your *78K0R* - *Save It!*.

1.2 Features

- NEC Electronics µPD78F1009 low power consumption general purpose NEC 16-bit 78K0R MCU
- NEC Electronics µPD78F0730 MCU with on-chip USB interface
- 2 latched 7 segment LEDs
- Low current consumption modes demo application with GUI for the Host PC
- Applilet EZ PL graphical based, logic expression design tool, that directly creates the regarding object code
- On-Board debug function (TK-78K0 debugging) The 78K0R Save It! supports an On-Board debug function by using the IAR C-SPY debugger without a need of additional debug hardware. It allows FLASH downloading and standard debug functions like code execution, single stepping, breakpoints, memory manipulation etc.
- Power supply by USB interface or via external power supply
- The IAR Embedded Workbench for 78K and the IAR C-SPY debugger / simulator are included. These packages are restricted in such that maximum program code size is limited to 4 Kbyte.
- IAR visualSTATE demo, a package of integrated tools for developing, testing, and implementing embedded applications based on state chart diagrams.
- Full documentation is included for the NEC 78K078F1009 microcontroller, NEC 78K0 78F0730 microcontroller, IAR Systems Embedded Workbench and IAR Systems C-SPY debugger / simulator.

The *78K0R* – *Save It!* is not intended for code development. NEC does not allow and does not support in any way any attempt to use *78K0R* – *Save It!* in a commercial or technical product.

1.3 System requirements

HOST PC	A PC supporting Windows 2000, Windows XP or Windows Vista is required for the IAR Systems Embedded Workbench demo-version. A Pentium processor with at least 1 GHz CPU performance, with at least 256 Mbytes of RAM, allowing you to fully utilize and take advantage of the product features. 500 Mbytes of free disk space and an additional 10 Mbytes of free disk space on the Windows system drive.
Host interface	A web browser and Adobe Acrobat Reader to be able to access all the product documentation. USB interface that enables communication based on USB (Ver1.1 or later)

Note: Updates of the IAR Embedded Workbench for 78K, documentation and/or utilities for 78K0R SAVE IT!, if available, may be downloaded from the NEC WEB page(s) at http://www.eu.necel.com/78K0RSAVEIT

1.4 Trademarks

IAR Embedded Workbench, visualSTATE and C-SPY are registered trademarks of IAR Systems AB. Microsoft and Windows are registered trademarks of Microsoft Corporation. Adobe and Acrobat Reader are registered trademarks of Adobe Systems Incorporated.

All other product names are trademarks or registered trademarks of their respective owners.

2. 78K0R - Save It! Components

The *TK-78K0R/KE3L* board is equipped with USB-connector and with several connectors in order to be connected to host computers, FLASH programmer, MiniCube2 or any external target hardware.



Figure 1: 78K0R – Save It! components

2.1 Operation mode settings

Caused by the different operating modes the 78K0R –Save It! supports different switches and jumpers are available on the board.

The table below shows the 3 different modes and the regarding switch / jumper setting for each mode.

Switch / Jumper	Demonstration mode	Debugging / Writing mode	PC communication mode
SW1	DEMO	Debug	KOUSB
SW2	DEMO	EXT	KOUSB
SW3.1 – SW3.3	OFF	ON	OFF
SW3.4 – SW3.8	Unused	Any	Any
JP1	Short	Short	Short
JP2	1-2 and 3-4 shorted	1-3 shorted	1-3 shorted

Table 1: 78K0R – Save It! Switch and Jumper settings

2.2 SW1

SW1 is a 3 position switch to set following modes.

Position	Mode
Left	Demo
middle	KOUSB
Right	Debug

Table 2: SW1 modes

2.3 SW2

SW2 is a 3 position switch to set following modes.

Position	Mode
Left	Demo
middle	KOUSB
Right	EXT

Table 3: SW2 modes

2.4 SW3

SW3 is an 8 position DIP switch with following functions

Bit	Mode
1	Mode selection (see Table 1)
2	Mode selection (see Table 1)
3	Mode selection (see Table 1)
4	Connected to microcontroller port pin P33
5	Connected to microcontroller port pin P42
6	Connected to microcontroller port pin P43
7	Connected to microcontroller port pin P77
8	Connected to microcontroller port pin P76

Table 4: SW3 modes

Note: Make sure to set the regarding pull-up resistor option registers (PUx) of the microcontroller if the Pins P33, P42, P43, P76 and P77. Please refer to the <u>device user's manual</u> of the 78K0R/Kx3-L for detailed information about the pull-up resistor option register settings.

2.5 SW4 (Reset switch)

SW4 is connected to the external reset pin of the μ PD78F1009 microcontroller. External reset can be triggered by pushing.

2.6 SW5 (INTP0)

SW5 is connected to the "P120/INTP0/EXLVI" pin of the μ PD78F1009 microcontroller. It can be used as external interrupt source.

Note: Make sure to set up the microcontroller internal pull-up resistor option register (PU12) correctly when using this external switch. Please refer to the <u>device user's manual</u> of the 78K0R/Kx3-L for detailed information about the pull-up resistor option register settings.

2.7 SW6 (INTP2)

SW6 is connected to the "P32/SCK10/SCL10/INTP2" pin of the μ PD78F1009 microcontroller. It can be used as external interrupt source.

Note: Make sure to set up the microcontroller internal pull-up resistor option register (PU12) correctly when using this external switch. Please refer to the <u>device user's manual</u> of the 78K0R/Kx3-L for detailed information about the pull-up resistor option register settings.

2.8 JP1

JP1 is a switch jumper to select the power supply.

Position	Function
Short	USB power supply connected (USB1)
Open	Power supply via CN1 or CN2

Table 5: JP1 settings

2.9 JP2

JP2 is a jumper field to establish a current consumption measurement

Jumper settings	setting
1-2 and 3-4 short	Demonstration mode
1-3 short	Debugging/Writing and PC communication mode
any other	restricted

Table 6: JP2 settings

2.10 Q4 (illuminance sensor)

The illuminance sensor Q4 is connected to the P80/CMP0P/INTP3/PGAI pin of the μ PD78F1009 microcontroller. When lightening up the voltage gains up and when darkening it goes down. So the gain amplifier function of the microcontroller can be used to read the illuminance. For further information to set up the registers of the microcontroller to use the gain amplifier functionality please refer to the <u>device user's manual</u>.

2.11 U10, U12 (7 segment LED)

U10 and U12 are two 7 segment LEDs. They can be set through the latches U11 and U13. The data channels of the latch are connected to channel P20 to P27 and the clock pins to P00 (U10) and P01 (U12) of the of the μ PD78F1009 microcontroller.





NFC

Figure 2: 78K0R – Save It! 7 segment LEDs structure

Figure 3: 78K0R – Save It! 7 segment LED port pin connection

In respect to the pin connection the following table shows the port values to be written to show the numbers from 0 to 9 on the 7 segment LEDs.

Digital number	Port value (hex)	Digital number	Port value (hex)
0	0xC0	5	0x92
1	0xF9	6	0x83
2	0xA4	7	0xF8
3	0xB0	8	0x80
4	0x99	9	0x98

Table 7: 7 segment LED port values

2.12 LED1 (power LED)

LED1 is the power LED, lights up when power is supplied via USB, CN1 or CN2 to the 78K0R - Save It!

2.13 Universal area

The Universal area of the *78K0R* – *Save It!* is a free for the customer to use mounting area. Furthermore there is the possibility to mount a MiniCube2 connector to the Universal area

2.13.1 Minicube2 connector

If a MiniCube2 shall be connected to the *78K0R* – *Save It!* starter kit a 16-pin connector has to be mounted to the universal area as shown below.



Figure 4: 78K0R – Save It! MiniCube2 connector

Furthermore some solder bridges have to be shorted to connect the connector to the target device.



Figure 5: 78K0R – Save It! Minicube2 connection solder bridges

3. 78K0R - Save It! system configuration

The 78K0R – Save It! starter kit system configuration is given in the diagram below



Figure 6: 78K0R- Save It! system configuration

3.1 78K0R – Save It!

The *78K0R* – *Save It!* is a demonstration kit for the 78F1009 16-bit low power consumption microcontroller of the 78K0R family. The demonstration board is connected to the host system via USB interface cable. The host system may be used for On-Chip debugging by using the IAR C-SPY debugger and to allow execution of application programs on *78K0R* – *Save It!* starter kit.

78K0R – Save It! runs the microcontroller at 20 MHz operating speed in normal operation mode.

3.2 Host computer

The USB host interface enables communication to the *78K0R* – *Save It!* board. The μ PD78F0730 78K0 8-Bit microcontroller with on-chip USB interface and the NEC virtual UART driver allows application software to access the μ PD78F1009 device in the same way as it would access a standard RS232 interface. The NEC virtual UART driver appears to the windows system as an extra Com Port, in addition to any existing hardware Com Ports.

3.3 Power supply via USB interface

The *TK-78F0730* board is powered by the USB interface. Optional the power supply can be applied via a JACK connector which has to be mounted to CN3.

4. 78K0R - Save It! installation and operation

4.1 Getting started

The *78K0R* – *Save It!* is delivered with two PC based user programs. The TK-78K0RKE3L demonstration program is designed to measure the current consumption, the CPU frequency and the actual illuminance (measured by an illuminance sensor). The Applilet EZ PL is an easy to use GUI based program to build up small logical circuits which are automatically converted into an IAR Embedded workbench or NEC PM+ project. They can be compiled and downloaded to the target hardware directly from within the Applilet EZ PL; therefore it is necessary that the chosen Workbench Suite is correctly installed on the Host PC.

4.2 CD-ROM contents

NET 78K0R - Save It!	CD-ROM ROOT
C Acrobat	- Acrobat Reader for 32Bit Windows OS
Applilet	- Applilet EZ PL
Device file package	 µPD78F8024 Device file package
Doc	- Documentation
Contract dotnet	- Microsoft .NET package
Driver	- TK-78K0RKE3L driver files
IAR Systems	 IAR Embedded Workbench for 78K IAR visualSTATE
Samples	- Example projects for the <i>78K0R</i> - <i>Save It!</i> Starter Kit
TK-78K0R_KE3L_Demo	 TK-78K0RKE3L current consumption demonstration program
WriteEZ4	 Flash Programmer WriteEZ4 incl. PRM file for µPD78F1009

Figure 7: 78K0R - Save It! CD-ROM contents

5. Hardware installation

After unpacking the *78K0R* – *Save It!* demonstration kit, connect the board via connector CN1 to your host computer using the provided USB interface cable. When *TK-78K0R/KE3L* board is connected, the USB driver needs to be installed on the host machine. Please refer to the following <u>USB Driver Installation</u>.

6. Software installation

The 78K0R – Save It! package comes with the following software packages:

- TK-78K0R/KE3L demonstration program
- IAR Systems Embedded Workbench for 78K 4Kbyte code size limited, including C compiler, assembler, linker, librarian and IAR C-SPY debugger / simulator
- IAR Systems Visual State time limited demo program
- Device file package of µPD78F1009 for IAR Embedded Workbench for 78K
- WriteEZ4 flash programmer software including the PRM file for µPD78F1009



Figure 8: 78K0R – Save It! CRROM autorun.exe

6.1 TK-78K0RKE3L demonstration program installation

Before the TK-78K0RKE3L demonstration program can be installed on the host PC it is mandatory that the .NET Framework is installed on the host PC. For further information about installing .NET Framework please refer to the TK-78K0R_DEMO_GUI.pdf available on the CD delivered with the 78K0R – Save It! package in the "\Doc" folder.

78K0R – Save It!



To install the TK-78K0RKE3L demonstration program just press the regarding button from the Autorun of the CD-ROM provided within the *78K0R – Save it!* package. The setup dialogues will guide you through the installation process. The installation can also be started by executing the TK-78K0R_KE3L_Demo.msi in the directory "\TK-78K0R_KE3L Demo" of the CDROM.

6.2 IAR Systems Embedded Workbench for 78K installation

To install the IAR Systems Embedded Workbench for 78K including C-SPY debugger / simulator press the regarding button from the Autorun of the CDROM provided within the 78KOR – Save It! package. The installation can also be started by executing the Autorun.exe program in the directory "\IAR Systems\" of the CDROM.

When running the autorun.exe the following screen appears.

Welcome to IAR Systems	● 日本語 (Japanese)
IAR Embedded Workbench [®] 4K Kickstart for NEC 78K	
Install IAR Embedded Workbench [®]	
O Install visualSTATE [●]	
htm Release notes	
View the product documentation	
Install Adobe Acrobat Reader	
Explore the CD	
	SYSTEMS

Figure 9: IAR Systems Installation screen

To install the IAR Embedded Workbench for 78K just press the regarding button "Install IAR Embedded Workbench ®". The setup dialogues will guide you through the installation process. For further information about the IAR Embedded Workbench installation refer to the InstallationGuide_IAR_EWB.ENU.pdf in the directory "\IAR Systems\" of the CDROM.

Note: Please make sure to install the 78K0R/Kx3-L device file package for the IAR Embedded Workbench, as explained below, before using the IAR Embedded Workbench with the TK-78K0RKE3L board.

6.3 IAR Systems visualSTATE Installation

To install the IAR Systems visualSTATE press the regarding button from the Autorun of the CDROM provided within the *78KOR* – *Save It!* package. The installation can also be started by executing the Autorun.exe program in the directory "\IAR Systems\" of the CDROM. The IAR Systems Installation screen will appear and the installation of IAR visualSTATE can be started by pressing the "Install visualSTATE ®" button. The setup dialogues will guide you through the installation process. For further information about the IAR visualSTATE installation refer to the "InstallationGuide_IAR_VS.ENU.pdf" in the directory "\IAR Systems\" of the CDROM.

6.4 Device file package of 78K0R/Kx3-L for IAR Embedded Workbench installation

To install the device file package of the 78K0R/Kx3-L for the IAR Embedded Workbench just press the regarding button from the Autorun of the CD-ROM provided within the *78K0R – Save It!* package. The setup dialogues will guide you through the installation process. The installation can also be started by executing the dev_install.exe in the directory "\Device file package" of the CDROM.

6.5 WriteEZ4 installation

To install the WriteEZ4 just press the regarding button from the Autorun of the CD-ROM provided within the 78KOR – Save It! package. The setup dialogues will guide you through the installation process. The installation can also be started by executing the WriteEZ4_V102_EE.exe in the directory "\WRITEEZ4" of the CDROM.

6.6 USB Driver Installation

In order to use the TK-78K0RKE3L board, the USB driver needs to be installed on the host machine. Install the driver according to the following procedure:

Installation on Windows 2000	Page 22
Installation on Windows XP	Page 26

6.6.1 Installation on Windows 2000

1. When the *TK-78K0RKE3L board* is connected with the host machine, the board is recognized by <Plug and Play>, and the wizard for finding new hardware is started. Click Next>.



Figure 10: Found New Hardware Wizard (Windows 2000)

2. Following the window below is displayed. So, check that "Search for a suitable driver ..." is selected, then click Next>.

	Found New Hardware Wizard				
	Install Hardware Device Drivers A device driver is a software program that enables a hardware device to work with an operating system.				
	This wizard will complete the installation for this device:				
Check that "Search fo suitable driver" is selec					
	installation click Next.				
	What do you want the wizard to do?				
	Search for a suitable driver for my device (recommended) Display a list of the known drivers for this device so that I can choose a specific driver	lick			
	< <u>B</u> ack Next > Cancel				

Figure 11: Search Method (Windows 2000)

3. Check the "Specify a location" check box only, then click Next>.



Figure 12: Driver File Location (Windows 2000)

Locate File				<u>? ×</u>
Look jn:	🔁 Driver	•	🗢 🗈 💣 🌆 •	
History Desktop My Documents My Computer	Loca Loca Loca Loca Loca NEC 78KC	nputer loppy (A:) I Disk (C:) I Disk (D:) I Disk (E:) I Disk (F:) I Disk (F:)		
My Network P	File <u>n</u> ame: Files of <u>type</u> :	MQB2SALL.inf Setup Information (*.inf)	•	<u>O</u> pen Cancel
		Locate to "CDROM:\\	Driver"	

4. Locate to the folder "CDROM:\Driver".

Figure 13: Address Specification 1 (Windows 2000)

5. The setup information file "MQB2ALL.inf" is automatic selected, then click Open to proceed within driver installation.

Locate File					? ×
Look <u>i</u> n:	🔁 Driver		•	🗢 🗈 💣 🎫	
	MQB2SALL				
History					
Desktop					
<u></u>					
My Documents					
My Computer					
	File <u>n</u> ame:	MQB2SALL		•	<u>O</u> pen
My Network P	Files of <u>type</u> :	Setup Information (*.inf)		7	Cancel

NE

Figure 14: Address Specification 2 (Windows 2000)

6. After the location of the USB driver has been specified click OK to proceed.

Found New	v Hardware Wizard	×	
-	Insert the manufacturer's installation disk into the drive selected, and then click OK.	OK Cancel	Click.
	Copy manufacturer's files from:		
	I:\Driver	Browse	

Figure 15: Address Specification 3 (Windows 2000)

7. Click Next>



8. Click Finish to complete the installation of the USB driver.



Figure 17: USB Driver Installation Completion (Windows 2000)

6.6.2 Installation on Windows XP

1. When the *TK-78K0R/KE3L board* is connected with the host machine, the board is recognized by Plug and Play, and the wizard for finding new hardware is started. At first the hardware wizard will ask if windows should search on the windows update web, check "No, not this time" and then click Next>.



Figure 18: Found New Hardware Wizard 1 (Windows XP)

2. Check that "Install from a list or specific location (Advanced)" is selected, then click Next>.

NEC

	Found New Hardwa	vare Wizard	
Check that "Ins	tall from a list or	This wizard helps you install software for: USB Device	
specific" is s			Click.
		< <u>B</u> ack <u>N</u> ext > Cancel	

Figure 19: Found New Hardware Wizard 2 (Windows XP)

3. Check that "Search for the best driver in these locations." is selected. Select the "Include this location in the search:" check box and then click <u>Browse</u>.



Figure 20: Search Location Specification 1 (Windows XP)

4. Locate the folder "C CDROM:\Driver" and click OK.



Figure 21: Search Location Specification 2 (Windows XP)

5. As shown below, "NEC Electronics Starter Kit Virtual UART has not passed Windows Logo testing to verify its compatibility with Windows XP." is displayed. Click Continue Anyway.



Figure 22: Windows XP Logo Testing (Windows XP)

6. After the installation of the USB driver is completed the window below is displayed. Click Finish to close the hardware wizard.



Figure 23: USB Driver Installation Completion (Windows XP)

6.7 Confirmation of USB Driver Installation

After installing the USB driver, check that the driver has been installed normally, according to the procedure below. When using the Save It! board the "NEC Electronics Starter Kit Virtual UART" should be present like in the figure below.

Please check in the Windows "Device Manager" within the Windows Properties ("Hardware" tab), that the driver is installed normally.



Figure 24: Windows Device Manager

7. Demo application using the TK-78K0R/KE3L demonstration program

The Low-Power-Consumption Demonstration GUI for the *78K0R* – *Save It!* is Windows software that can be used to check the operation of the 16-bit microcontroller 78K0R/KE3-L mounted on the included demonstration kit board *TK-78K0R/KE3L* made by TESSERA Technology Inc. by performing a simplified display of the features and power of the microcontroller.

The Demo GUI allows the user to check the on-chip oscillator of the KE3-L, the power consumption, and the optical sensor mounted on the TK Board, and to easily switch among the full-speed mode to standby mode. The current in a user program can also be measured.



Figure 25: TK-78K0RKE3L demonstration program

Note: Values displayed on the GUI

Reference values are displayed on the GUI. Results that do not satisfy the device specifications might be displayed depending on the resistor for detecting current and the resonator or measuring the frequency used on the board. Use the values only as a reference for design and evaluation. Use calibrated measuring equipment if precise measurement is required.

For further information about usage and functionality of the tool please refer to the *TK-78K0RKE3L* demonstration program User's manual (TK-78K0R_DEMO_GUI.pdf). The manual can be found on the 78K0R – Save *It!* CDROM in the "\Doc" folder.

8. Applilet EZ PL

8.1 Overview

Applilet EZ PL is a software tool that can be used to automatically generate microcontroller software by using only its GUI (graphical user interface) when installed on a host.

By using Applilet EZ PL, the basic operations of systems using a microcontroller can be learned and evaluated without knowledge of complex programming languages.

🖸 Applilet EZ PL [- Save-It.loc				
<u>File E</u> dit <u>V</u> iew <u>R</u> un						
New Open Save	Print View	it Copy Paste	Undo Start St	op Build		
	Input					Output
74 series Common	DipSW DipSW3-4)	Output
	DipSW DipSW3-5					
	DipSW DipSW3-6					
	DipSW DipSW3-7					
→ (P) (P)→	DIPSW DipSW3-8				
N.C.	Push Button SW5					Digital out
5-2-	Push Button SW6	<u> 5 </u>		ᡚ᠊ᡚ᠊ᡚ᠊ᡚ	Ď	7seg Segment
555						
						JUARTO out
						UART1 out
Comp						Step Moter
FF Cnt						
BCnt Timer						Music Scale
Delay Clock						
+ <mark>8</mark> .8 + ^{Reset}						
K Build Simulate	1					
	e Heip					
<						
						, ROM Capacity : 14Kbyte

Figure 26: Applilet EZ PL main window

8.2 Features

 Automatic generation of object code for the 78K0/Kx2, 78K0S/Kx1+, 78K0R/KE3-L, and 78K0/KC2-L

Applilet EZ PL can be used to automatically generate object code (*.hex) and a source file, which can be written to the flash memory in the 78K0/Kx2, 78K0S/Kx1+, 78K0R/KE3-L, and 78K0/KC2-L, by assigning GUI panels, which represent the various functions of a microcontroller, to the input, processing, or output block area.



Therefore, software can be generated without special knowledge of programming languages supported by the 78K0/Kx2, 78K0S/Kx1+, 78K0R/KE3-L, or 78K0/KC2-L.

• Simulation

Applilet EZ PL can be used to easily simulate and check the operations of the target system. The operation can be checked on the PC without the target system by using this simulation function.

 Automatic generation of a project file for the integrated development environment A *.prj project file for the integrated development environment PM+, made by NEC, or a *.eww project file for the integrated development environment IAR Embedded Workbench, made by IAR

Systems, is automatically generated at the same time as the object code (*.hex). By using this project file, the source file (*.c or *.h) generated by Applilet EZ PL can be easily reedited or recompiled in the integrated development environment of each company. [Note]

Note: Only a source file supported by the integrated development environment PM+, made by NEC, is generated for a low-pin-count microcontroller (78K0S/Kx1+).

Evaluation board

By using Applilet EZ PL with an evaluation board, created software can be easily checked.

For further information please refer to the Applilet EZ PL User's manual (U17656EJ6V0UM00.pdf) available on the 78KOR – Save It! CD-ROM in the "\Doc" directory.

8.3 Applilet EZ PL demo application

The now further described demonstration application can be found on the *78KOR* – *Save It!* CD-ROM in the "\samples\Applilet EZ PL startup sample" directory.

The Applilet EZ PL demonstration project is an easy logical circuit where the both 7 segment LEDs of the *TK-78K0RKE3L* board are toggled, with the set up toggling rate of 0.5sec, depending on the switch settings of the board (shown in <u>Figure 26</u>). To let both 7 segment LEDs toggle it is mandatory that the switches SW3.4, SW3.6 and SW3.8 are set to **ON**, switches SW3.5 and SW3.7 to **OFF** and the switches SW5 and SW6 are not pressed. If any of these settings are changed the LEDs will stop toggling.

Switch Channel	Position
SW3.4	ON
SW3.5	OFF
SW3.6	ON
SW3.7	OFF
SW3.8	ON
SW5 + SW6	OPEN

Table 8: Applilet EZ PL demo Switch settings

9. Flash Programmer WriteEZ4

WriteEZ4 is flash programming software to flash hex files to the related device. For installation information refer to the chapter <u>WriteEZ4 installation</u>.

🗟 WriteEZ4	
<u>File D</u> evice <u>V</u> iew <u>H</u> elp	
/P 🎮 🗔 🖏 🗡 🏶 💸	
>> FlashOpenning Flash Open OK >> ParameterFile Reading Success Read ParameterFile. >> LoadFile Reading Success read HEX file.]	Device Name : Firm Version : ExtCode : Vendor : Parameter file Name : 78F1009 Format : 0419 Version : V1.00 Processor Ver. 0200 Load file Name DEM0_FIRM_KE3L.HE Date : Date : 2008/11/09 09:47:12 Chksum : 60DCh Area : O000000-00FFFFh Connection to device Port : COM4 Speed 115200bps Range Chip Freq. : Internal-OSC Multiply : 1.00 Nultiply : 1.00
<	
Ready	NUM

Figure 27: WriteEZ4 Startup

9.1 Device Setup

To provide all necessary information about the device to be programmed, only the corresponding flash parameter file must be loaded. The parameter file (*.prm) for the μ PD78F1009 is located on the CDROM, in the same folder as the WriteEZ3 setup file. Please use the menu "**Device -> Setup...**" to open the following dialogue and the button "**PRM File Read**" to select the parameter file.

🗟 Device Setup				
Standard Advance				
Parameter file 78F1009.prm	PRM File Read			
Host connection Supply os	scillator			
Port COM4 - Frequen	Internal-OSC MHz			
Speed 115200bps V Multiply	rate 1.00			
Operation Mode				
Chip Start 000				
C Block End 063				
C Area 🗖 Show Addres				
Target Reset Message				
	OK Cancel			

Figure 28: WriteEZ4 Device Setup Dialogue

Please check that the correct host communication port is selected. The used communication port can be seen in the <u>Windows Device Manager</u>.

9.2 Using WriteEZ4

After a successful device selection the internal flash memory can be blank-checked, erased, programmed or verified. WriteEZ4 can be controlled either by menu or by buttons



Figure 29: WriteEZ4 Device Menu



Table 9: WriteEZ4 action buttons

WriteEZ4 supports Intel-Hex and Motorola S-record file formats as input file.

10. Erasing the flash memory (using the WriteEZ4)

If the preprogrammed software is written to the flash it is mandatory to erase the flash of the device before it is possible to use the Applilet EZ or the IAR Embedded Workbench with the *TK-78K0RKE3L* board. To erase the flash please perform following steps.

- 1. Make sure that the *TK-78K0RKE3L* board is in the Debugging/Writing mode. (<u>78K0R Save</u> <u>It! Switch and Jumper settings</u>)
- If not already connected, connect the TK-78K0RKE3L board via USB1 connector to the target device.
- 3. Run the WriteEZ4 flash programming software.
- 4. Select the right .prm file as explained in chapter <u>9.1 Device Setup</u>. The regarding 78F1009.prm file can be found on the CDROM in the folder "CDROM\WRITEEZ4\PRM files"

Open			? 🛛
Look in: ն	PRM files	- 🗢 主	💣 🎟 •
🗖 78F1009.p	rm		
File <u>n</u> ame:	78F1009.prm		<u>O</u> pen
Files of <u>t</u> ype:	PRM Files(*.PRM)	•	Cancel

Figure 30: WriteEZ4 open .prm file

- 5. Click the **erase button**
- 6. After finishing the flash erase following message shall apply



Figure 31: WriteEZ4 flash erase successful

If so the device flash is successfully erased.

11. IAR Embedded Workbench 78K0R - Save It! sample project

The IAR Embedded Workbench IDE is a very powerful Integrated Development Environment that allows you to develop and manage a complete embedded application project.

The now described project can be found on the *78KOR* – *Save It!* CR-ROM in the "\samples\TK-78KORKE3L startup sample" folder.

To run/debug the project with the IAR C-SPY debugger it is necessary to follow the steps described below in this chapter.

For further information about the IAR Embedded Workbench ant the included functionalities please refer to the regarding User's manuals available on the – Save It! CR-ROM in the "\IAR Systems\doc" folder.

Note: Make sure that the Flash memory is erased properly before first download the project to the target device. Refer to the chapter Erasing the flash memory (using the WriteEZ4).

11.1 Hardware setup

To run the program it is necessary to set up the TK-78K0RKE3L board in the following way.

Switch / Jumper	Mode position	
SW1	Debug	
SW2	EXT	
SW3.1 – SW3.3	ON	
SW3.4 – SW3.8	Any	
JP1	Short	
JP2	1-3 shorted	

Table 10: IAR EWB 78K0R - Save It! sample project hardware setup

11.2 Loading the Save It! sample project

After starting the IAR Embedded Workbench for 78K the Startup window pops up. In this window you are able to choose the **Open existing workspace** button.

Embedde	d Workbench Startup	X	
	Create new project in current workspace	e	
	Add existing project to current workspace		
	Open existing workspace		
	Example applications		
Recent workspaces:			
		Open	
C Do not show this window at startup			
		Cancel	

Figure 32: IAR Embedded Workbench Startup window
Then locate the sample project on the CD and open the SAMPLE-TK-78KORKE3L.eww workspace file. Afterwards the IDE should look similar like below.

🔀 IAR Embedded Workbench IDE		
<u>File E</u> dit <u>V</u> iew <u>P</u> roject <u>T</u> ools <u>W</u> indow <u>H</u> elp		
D 🚔 🖬 🕼 🚑 👗 🖻 💼 🗠 က	_ イ ≻ ≒ 注 図 ▷ ≁ @ 卧 時 聯 隊 ይ	<u>ي</u> ا
Workspace ×	main.c	
Debug	50 unsigned char g start;	
Files 😤 🛤	51 unsigned char g_led = 0;	<u> </u>
□ □ SAMPLE-TK-78K0RKE3 ✓	52 unsigned char g_count = 0;	
	53 unsigned char g_led_s = 10;	
H → ⊞ mit_user.c *	54 55	
	56 /*	
Hand Hand Hand Hand Hand Hand Hand Hand	57 **	
⊢⊞ 🗑 port_user.c *	58 **	
Here 🛗 rtc.c 🔹 🔹	59 ** Abstract:	
Hand Hand Hand Hand Hand Hand Hand Hand	60 ** This function implements main function.	
Hand System.c *	61 **	
+ system_user.c *	62 ** Parameters: 63 ** None	
→ ⊞ Bysteminit.c ・	63 ** NOLE 64 **	
Hand Battimer.c ★	65 ** Returns:	
├── Dutput *	66 ** None	
	67 **	
	68 **	
	69 */	
	70 void main(void) 71 {	
	72 g_start = 2;	
	73 while (10)	
	74 {	
	75	
	76 g_led_s = 0;	
	77 while(g_start == 0){	/* LED blink
	78 }	•
SAMPLE-TK-78K0RKE3L		
× Messages	*	File
in a souges		110
TO A CONTRACT OF		
		>
Ready	Errors 0, Warnings 0	NUM M

Figure 33: IAR Embedded Workbench Main IDE window

To verify that the right debugging device is chosen please open the project options window by clicking **Project** \rightarrow **Options** and then open the tab **Debugger**.

Note: Although the On-board debug interface is used, the MINICUBE C-Spy driver must be selected instead of the standard driver TK-78K used for other starter kits. If the debug session via MINICUBE2 and the OCD1 connectors shall be started, also the MINICUBE C-Spy driver must be selected.

The corresponding COM port where the *TK-78K0R/KE3L board* is connected to the host PC will be detected automatically by the IAR C-SPY debugger.

Options for node "SA	MPLE-TK-78KORKE3L"
Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Factory Settings Setup Extra Options Plugins Driver MINICUBE QB-78KOMINI, QB-78KOSxxxMINI, QB-MINI2 Imain Imain Setup macros Imain Device descriptions Imain Override default Imain
	OK Cancel

Figure 34: IAR Embedded Workbench Debugger Options

The next step is to check that the correct linker settings are set. This can be done in the "Linker" category as shown below. Select the "Config" tab and check that the linker command file "Ink78f1009.xcl" is selected. This file is used by the linker and contains information on where to place the different sections of code, data and constants that may be used within the demo project:

Options for node "SA	MPLE-TK-78KORKE3L"
Category: General Options C/C++ Compiler Assembler visualSTATE Coder Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Factory Settings Output Extra Output #define Diagnostics Linker command file Qverride default \$PROJ_DIR\$\[nk78f1009.xc] Override default program entry Entry label program_start C Defined by application Search paths: (one per line) \$TOOLKIT_DIR\$\LIB\ Raw binary image File: Symbol: Segment: Align:
	OK Cancel

Figure 35: Embedded Workbench Linker Configuration

Now after everything has been setup correctly it's time to compile and link the demonstration project. Close the Options menu and select "Rebuild All" from the "Project" menu. If the project is compiled and linked without errors or warnings it can now be downloaded to the *78KOR Save It! board* and debugged.

To start the IAR C-SPY debugger select the option "Debug" from the "Project" menu or press the () "Debugger" button.

When first starting the debugger connection, after building the project, it is necessary to set up the right connection settings. Please take care that the following settings are chosen:

- ID code = FFFFFFFFFFFFFFFFFFFFF
- Main clock = External 20.00 MHz
- Sub clock = External 32.768KHz
- Target connect = TOOL0+TOOL1

MINICUBE2 Hardware Setup fo	r 78KOR (78F100964)	
ID code FFFFFFFFFFFFFFFFFFFF Erase flash before next ID c	Time unit nsec	OK Cancel
Main clock Clock board External System 20.00 MHz	Sub clock C Clock board External System 32.768	Default Fail-safe break View setup
Flash programming Permit Not Permit	Target power off Permit Not Permit	Target connect TOOL0+TOOL1
Pin mask	Peripheral break A (timer) B (serial etc.)	C Connect C Not Connect
Memory map Start address: Length: 0x0 960	Type:	Add
0x00000 - 0x0FFFF Internal ROM 0xFF300 - 0xFFEFF Internal RAM		Remove Remove All

Figure 36: IAR Embedded Workbench MINICUBE2 settings

Now the debugger is started and the demo project is downloaded to the *78K0R Save It! board*. The progress of downloading is indicated by blue dots in the MINICUBE Emulator window. Please note that downloading of larger executables may take some time.

After the download was completed all debug features of IAR C-SPY debugger are available, i.e. Single Stepping, Step Over/-In/-Out, Go-Execution, Breakpoints, Register / Memory view etc.

To get more details on the debugger configuration and capabilities please refer to the "78K IAR Embedded Workbench IDE User Guide" of the IAR installation.



Figure 37: IAR C-SPY debugger

11.3 Running the application

After downloading the project to the device the debugger window opens and, if not changed by the user, the program runs to the first instruction of the main() function. When reaching this point the necessary clock initialization is already executed.

This sample project makes use of different timer interrupts to control the 7 segment LEDs available on the TK-78K0RKE3L board. So this project gives a short overview about timer initialization and timer interrupts usage of one timer array unit (TAU) channel as well as the real time clock (RTC) peripheral.

11.3.1 Count mode

When resuming execution of the program, after the break when entering the main function, the controller starts to count up from 0 to 49 on the 7 segment LEDs. This is controlled by the TAU channel 0 which is initialized as an interval timer. Furthermore every time the interrupt occurs and the value is counted up the interval time increases. During this stage both external interrupts INTP0 and INTP2 are disabled.

11.3.2 Blink mode

After the counter reached its final value of 49 the used timer interrupt is masked and the RTC interrupt will be unmasked and is used for the next stage. Furthermore the INTPO is unmasked to be able to leave this stage whenever the user wants to.

The only function of this stage is to let the final counter value blink in a frequency of 2Hz controlled by the RTC interrupt. To leave this stage only SW5 has to be pressed once.

11.3.3 Spinning mode

When entering the Spinning mode the RTC interrupt and the INTP0 will be masked. In exchange the RTCI Interrupt, which occurs every 31.25ms and the INTP2 are unmask. The function of the Spinning mode is to spin 1 active segment around the 2 7segment LEDs. So every time a RTCI interrupt occurs the active segment is stepped clockwise to the next segment on the outer ring of both 7 segment LEDs. To leave this mode the SW6 has to be pressed to trigger an external interrupt. When leaving the spinning mode the <u>Count</u> mode is entered again.

12. IAR visualSTATE 78K0R – Save It! sample project

IAR visualSTATE is a Windows-based software package of integrated tools for developing, testing, and implementing embedded applications based on state-chart diagrams. It includes a graphical design environment, test tools, a code generator, and a documentation facility.

IAR visualSTATE has been developed in accordance with the Unified Modeling Language notation (UML).

For more information please refer to the UserGuide.pdf included to the *78KOR*—*Save It!* CD-ROM in the "\doc\visualSTATE" folder or to the <u>visualSTATE homepage</u>

The in the following described sample project can be found on the *78K0R* - *Save It*! CD-ROM in the "\samples\IAR visualSTATE sample project" folder.

12.1 Hardware setup

To run the program it is necessary to set up the TK-78K0RKE3L board in the following way.

Switch / Jumper	Mode position
SW1	Debug
SW2	EXT
SW3.1 – SW3.3	ON
SW3.4 – SW3.8	Any
JP1	Short
JP2	1-3 shorted

Table 11: IAR VS 78K0R - Save It! sample project hardware setup

12.2 Functionality of the sample project

The example implements a reaction timing measurement application that measures your reaction time a fixed number of times and then displays the average reaction time in hundredths of a second. The application

works in the following way:

- When the application is started it displays two '-' characters on the display and waits for a press on any of the two buttons.
- A timer with a fairly random interval is now started and when the timer times out a '[' or ']' symbol is displayed on the left or right digit respectively. A new timer is also started for the measurement of your reaction time.
- You should now as fast as possible press the left button if the '[' is shown or the right button if the ']' is shown. The time for you to react is read from the timer and added to a running sum.
- After a button press the process is repeated until 10 trials has been performed. Your average reaction time is now displayed and you can press a button to get back to the initial state and restart the measurement if you want to.

The state machine is implemented with 3 input events (button presses and an expired timer) and 10 small C functions for the interface to the hardware. There are 5 states to keep track of what is currently going on in the application. This is not the only plausible state machine implementation for this application. An alternative

could for example be to have separate, parallel and communicating state machines for time measurement and display updates.

Further details on the sample project can be found in the index.html file available in the 78KOR – Save It! CD-ROM "\samples\visualSTATE\doc" folder.

12.3 Running the sample project

To perform changes on the visualSTATE parts open the visualSTATE workspace file within the "\samples\IAR visualSTATE sample project\vS" folder of the CD-ROM. After loading the workspace the window should look like shown below.



Figure 38: IAR visualSTATE navigator

If the functionality or the settings of the sample shall be changed open the visualSTATE Designer.



Figure 39: IAR visualSTATE designer

To run the project on the TK-78K0RKE3L board locate the IAR Embedded Workbench workspace file within the "\samples\IAR visualSTATE sample project\EW" folder of the CD-ROM.

<pre>Be Ext yew Protect Look Window Beb C Comparison Protect Look Beb Pr</pre>	🔏 IAR Embedded Workbench IDE			
Visite Visite Debug-CSpylink I Files I Image: Second Stress St	<u>File E</u> dit <u>V</u> iew Project <u>T</u> ools <u>W</u> inde			
Debug CSpyLink Files Standburde "Macrodriver.h" Sinclude "macrodriver.h" Sinclude "Sacodriver.h" Sinclude "X3 SoftwareTimers.h" Sinclude "X3 SoftwareTimers.h" Sinclude "X3 SoftwareTimers.h" Sinclude "Karstoner.h" Sinclude "Karstoner.h"	D 🛩 🖬 🕼 🎒 🕺 🖪 🛍	N N 💽 🖉 🦻 🦕 🦎 💆 🔝 🗩 🖓 🎒 🔛 🖓 🧶 🥨		
Debug-CSpyLink Hes Carbon Construction of the second	Workspace 🛛 🗙	S. ActionFunctions c		•
Files ? #SAMPLE-T ? #Sinclude "userdefine.h" #include "userdefine.h" # Sinclude "inv3.Soft # SoftwareTi				-
INES Image: Sample:				
<pre>Function = for the first first</pre>				
<pre>Seture Eventuation of Status Seture Set</pre>				
G # SimpleE 6 # include "timer.h" G > SoftwareTi 7 # include "ReactionSEMLibB.h" G > SiftwareTi 6 G > SiftwareTi 9 G > SiftwareTi 7 # include "ReactionData.h" G > SiftwareTi 10 # include "ReactionAction.h" G > SiftwareTi 12 # endif G > SiftwareTi 13 G > SiftwareTi 14 # include "ReactionAction.h" H > Brown 13 H > Dott 13 H > Dott. 14 # const unsigned char TIMER_ID_TIME_BETWEN_TESTS = 0; H > Dott.user.c 14 H > Dott.user.c 15 I > P > Op.bit.no0 = 1; 19 > 0p.bit.no0 = 1; I > P > Op.bit.no0 = 1; 20 > 000; I > P > Op.bit.no0 = 1; 23 I > P > Op.bit.no0 = 1; 29 > 0p.bit.no0 = 1; I > P > Op.bit.no0 = 1; 30 P O_Dit.no1 = 1; I > P > Op.bit.no0 = 1; 30 P O_Dit.no1 = 1; I > P > Op.bit.no1 = 1; 31 P 0 = 0; I > P > Op.bit.no1 = 1; 31 P 0 = 0; I > P > Op.bit.no1 = 1; 31 P 0 = 0; I > P > Op.bit.no1 = 1; 32) I > O =		5 #include "io78f1009_64.h"		
Image: Soft		6 #include "timer.h"		
<pre>vSCoder // vscoder // vscode //</pre>		7 #include "ReactionSEMLibB.h"		
Image: Construct of the second sec				
<pre>Image: Constant in the second of the se</pre>	- I VSCoder			
<pre>12 #endif 13 14 const unsigned char TIMER_ID_TIME_BETWEN_TESTS = 0; 15 16 VS_VOID DisplayInitialMessage (VS_VOID) 17 { 18 port_user.c 19 popt 19 po_bit.no0 = 1; 20 po_bit.no1 = 1; 21 p0 = 0x00; 22 } 19 Immer_user.c 21 p0 = 0x00; 23 24 25 VS_VOID DisplayClear (VS_VOID) 26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.no1 = 1; 31 P0 = 0; 32 } 3AMPLE_TK.78KORKE3_</pre>				
<pre>13 13 14 const unsigned char TIMER_ID_TIME_BETWEN_TESTS = 0; 13 14 const unsigned char TIMER_ID_TIME_BETWEN_TESTS = 0; 15 15 16 VS_VOID DisplayInitialMessage (VS_VOID) 17 { 18 P2 = 0xBF; /* '-' */ 19 P0_bit.no0 = 1; 20 P0_bit.no1 = 1; 21 P0 = 0x00; 22 } 19 Ovtput 24 25 VS_VOID DisplayClear (VS_VOID) 24 25 VS_VOID DisplayClear (VS_VOID) 26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.no1 = 1; 31 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no1 = 1; 31 P0 = 0; 32 } 5AMPLE-TK-78KORKE3L 13 14 14 const unsigned char TIMER_ID_TIME_BETWEN_TESTS = 0; 13 14 14 const unsigned char TIMER_ID_TIME_BETWEN_TESTS = 0; 15 15 14 const unsigned char TIMER_ID_TIME_BETWEN_TESTS = 0; 15 15 16 VS_VOID DisplayInitialMessage (VS_VOID) 17 { 18 P2 = 0xBF; /* '-' */ 19 P0_bit.no1 = 1; 20 P0_bit.no1 = 1; 21 P0 = 0; 23 24 25 VS_VOID DisplayClear (VS_VOID) 26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no1 = 1; 30 P0_bit.no1 = 1; 31 P0 = 0; 32 } </pre>	🕀 🖽 cstartup.s26			
<pre>14 const unsigned char TIMER_ID_TIME_BETWEN_TESTS = 0; 15 15 16 VS_VOID DisplayInitialMessage (VS_VOID) 17 { 18 P2 = 0xBF; /* '-' */ 19 P0_bit.no0 = 1; 20 P0_bit.no1 = 1; 21 P0 = 0x00; 22) 19 WS_Action 21 P0 = 0x00; 22 23 24 24 25 VS_VOID DisplayClear (VS_VOID) 26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no1 = 1; 31 P0 = 0; 32} SAMPLE-TK-78KORKE3L </pre>				
<pre>15 15 16 VS_VOID DisplayInitialMessage (VS_VOID) 17 { 18 P2 = 0xBF; /* '-' */ 19 P0_bit.no0 = 1; 20 P0_bit.no1 = 1; 21 P0 = 0x00; 22 } 19 Wimer_user.c 22 } 23 24 25 Output 25 SAMPL 26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no1 = 1; 31 P0 = 0; 32 } 34 P0 = 0; 35 P0_bit.no1 = 1; 31 P0 = 0; 32 } 35 P0_bit.no1 = 1; 31 P0 = 0; 32 } 35 P0_bit.no1 = 1; 31 P0 = 0; 32 } 35 P0_bit.no1 = 1; 31 P0 = 0; 32 } </pre>				
<pre>16 Wp OrtC user.C 16 VS_VOID DisplayInitialMessage (VS_VOID) 17 (18 P2 = 0xBF; /* '-' */ 19 P0_bit.no0 = 1; 20 P0_bit.no1 = 1; 21 P0 = 0x00; 22 P0_bit.no1 = 1; 22 P0_bit.no1 = 1; 23 Output 24 25 VS_VOID DisplayClear (VS_VOID) 26 (27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.no1 = 1; 31 P0 = 0; 32 P0_bit.no1 = 1; 32 P0_bit.</pre>				
Image: System construction 17 { Image: System us 18 P2 = 0xBF; /* '-' */ Image: System us 19 P0_bit.no0 = 1; Image: System us 19 P0_bit.no1 = 1; Image: System us.c. 10 P0_bit.no1 = 1; Image: System us.c. 10 P0_bit.no1 = 1; Image: System us.c. 10 P0_bit.no1 = 1; Image: System us.c. 12 P0 = 0x00; Image: System us.c. 12 P0 = 0x00; Image: System us.c. 12 P0 = 0; Image: System us.c.	Here in port.c			
<pre>18 system_us 18 P2 = 0xBF; /* '-' */ 19 P0_bit.no0 = 1; 20 P0_bit.no1 = 1; 20 P0_bit.no1 = 1; 21 P0 = 0x00; 22} 23 C Output 24 25 VS_VOID DisplayClear (VS_VOID) 26 (27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.no0 = 1; 31 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no1 = 1; 31 P0 = 0; 32 } SAMPLE-TK-78KORKE3L</pre>				
□ ■ system_us 19 P0_bit.no0 = 1; □ □ □ timer.c 20 P0_bit.no1 = 1; □ □ □ VS_Action 23 □ □ Output 24 □ □ SAMPL 25 (27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no1 = 1; 30 P0_bit.no1 = 1; 31 P0 = 0; 32 P0_bit.no1 = 1; 31 P0 = 0; 32 ?				
□ Bitwer.c 20 P0_bit.nol = 1; □ Bitwer_user.c 21 P0 = 0x00; □ Dutput 23 □ Output 24 □ BiSAMPL 26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.nol = 1; 30 P0_bit.nol = 1; 30 P0_bit.nol = 1; 30 P0_bit.nol = 1; 31 P0 = 0; 32 } SAMPLE-TK.78KORKE31 fo				
→ B timer.c 21 P0 = 0x00; → B timer_user.c 22; → D type 23 → Output 24 → B SAMPL 25 VS_VOID DisplayClear (VS_VOID) ≥ 6 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.no1 = 1; 31 P0 = 0; 32; j				
H imer_user.c 22) H imer_user.c 23 Control 23 Control 24 Control 25 VS_VOID DisplayClear (VS_VOID) 26 (27 PO = 0; 28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.no1 = 1; 31 PO = 0; 31 P0 = 0; 32 }				
□ Output 24 □ □ □ SAMPL 25 VS_VOID DisplayClear (VS_VOID) 26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.no1 = 1; 31 P0 = 0; 31 P0 = 0; 32 }				
SAMPL 25 VS_VOID DisplayClear (VS_VOID) 26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.nol = 1; 31 P0 = 0; 31 P0 = 0; 32 }	⊞ 🛗 VS_Action	23		
26 { 27 P0 = 0; 28 P2 = 0xFF; 29 P0 bit.no0 = 1; 30 P0 bit.no1 = 1; 31 P0 = 0; 32 } SAMPLE-TK-78KORKE3L	🖵 📮 Output	24		
27 P0 = 0; 28 P2 = 0xFF; 29 P0_bit.nol = 1; 30 P0_bit.nol = 1; 31 P0 = 0; 32} SAMPLE-TK-78KORKE3L	He B SAMPL	25 VS_VOID DisplayClear (VS_VOID)		
28 P2 = 0xFF; 29 P0_bit.no0 = 1; 30 P0_bit.no1 = 1; 31 P0 = 0; 32.} fro 4				
29 P0_bit.no0 = 1; 30 P0_bit.nol = 1; 31 P0 = 0; 32} fo ◀				
30 P0_bit.nol = 1; 31 P0 = 0; 32} fo 4				
31 P0 = 0; 32) f0 <				
SAMPLE-TK-78KORKE3L				
SAMPLE-TK-78KORKE3L				
Messages		t0 •		•
	Messages			F
	moodgoo			
	<			
Errors 0, Warnings 0 Ln 1, Col 1 NUM			DU UT	

Figure 40: IAR Embedded Workbench visualSTATE sample project

When loading the visualSTATE sample project all necessary settings to run the application on the TK-78K0RKE3L board are already set. So the user just has to download the application to the target device.

The application can be downloaded by click Project \rightarrow Debug.

Note: Make sure to erase the memory of the target device properly before the first download of the visualSTATE sample project by using the WriteEZ4 as described in <u>chapter 10</u>.

12.4 Debug the 78K0R – Save It! visualSTATE sample project

After downloading the application to the target device all from the C-SPY debugger already known debug features are available. Furthermore more visualSTATE specific Debug features can be used like an actual State view or a graphical animation of the states and state switches. For further information about the visualSTATE C-SPY plug-in please refer to the regarding chapter in the C-SPYLinkUserGuide.pdf which is available in the "\doc\visualSTATE" folder.

13. Cables

13.1 USB interface cable (Mini-B type)



Figure 41: USB interface cable (Mini-B type)

14. Schematics



Figure 42: TK-78K0RKE3L schematics 1/3



Figure 43: TK-78K0RKE3L schematics 2/3



Figure 44: TK-78K0RKE3L schematics 3/3

[MEMO]