

User's Manual

78K0R-SPINIT

Single Board Low Voltage Motor Control Starter Kit

µPD78F1235

Document No. U19823EE1V0UM00 Date published May 2009 © NEC Electronics 2009 Printed in Germany

Legal Notes

- The information in this document is current as of May, 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 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).

Notes for CMOS Devices

1. VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

2. HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can result in malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

3. PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and to quickly dissipate it should it occur. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

4. STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

5. POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be

judged separately for each device and according to related specifications governing the device.

6. INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

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 otherlegal issues may also vary from country to country.

NEC Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8668, Japan Tel: 044 4355111 http://www.necel.com/

[America]

U.S.A.

2880 Scott Blvd.

Tel: 408 5886000

NEC Electronics America, Inc.

Santa Clara, CA 95050-2554,

http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH

Arcadiastrasse 10 40472 Düsseldorf, Germany Tel: 0211 65030 http://www.eu.necel.com/

United Kingdom Branch

Cygnus House, Sunrise Parkway Linford Wood, Milton Keynes MK14 6NP, U.K. Tel: 01908 691133

Succursale Française

9, rue Paul Dautier, B.P. 52 78142 Velizy-Villacoublay Cédex France Tel: 01 30675800

Tyskland Filial

Täby Centrum Entrance S (7th floor) 18322 Täby, Sweden Tel: 08 6387200

Filiale Italiana

Via Fabio Filzi, 25/A 20124 Milano, Italy Tel: 02 667541

Branch The Netherlands

Steijgerweg 6 5616 HS Eindhoven, The Netherlands Tel: 040 2654010

[Asia & Oceania]

NEC Electronics (China) Co., Ltd

7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian District, Beijing 100083, P.R.China Tel: 010 82351155 http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.

Room 2511-2512, Bank of China Tower, 200 Yincheng Road Central, Pudong New Area, Shanghai 200120, P.R. China Tel: 021 58885400 http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.

12/F., Cityplaza 4, 12 Taikoo Wan Road, Hong Kong Tel: 2886 9318 http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.

7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C. Tel: 02 27192377

NEC Electronics Singapore Pte. Ltd.

238A Thomson Road, #12-08 Novena Square, Singapore 307684 Tel: 6253 8311 http://www.sg.necel.com/

NEC Electronics Korea Ltd.

11F., Samik Lavied'or Bldg., 720-2, Yeoksam-Dong, Kangnam-Ku, Seoul, 135-080, Korea Tel: 02-558-3737 http://www.kr.necel.com/

Preface

- **Readers** This manual is intended for users who want to understand the functions of the 78K0R-SPINIT for motor control.
- **Purpose** This manual presents the hardware manual of the 78K0R-SPINIT for motor control.
- **Organization** This system specification describes the following sections:
 - Jumper Descriptions
 - Hardware Setup
 - Schematics

Legend Symbols and notation are used as follows:

- Weight in data notation: Left is high order column, right is low order column
- Active low notation: xxx (pin or signal name is over-scored) or /xxx (slash before signal name)
- Memory map address: High order at high stage and low order at low stage
- **Note** Additional remark or tip
- Caution Item deserving extra attention
- **Numeric Notation**
- Binary: xxxx or xxxB
- Decimal: xxxx
- Hexadecimal: xxxxH or 0x xxxx

Prefixes representing powers of 2 (address space, memory capacity):

- K (kilo): 210 = 1024
- M (mega): 220 = 10242 = 1,048,576
- G (giga): 230 = 10243 = 1,073,741,824

Table of Contents

Chap	oter 1	Introduction	10
Chap	oter 2	78K0R-SPINIT Kit Contents	11
Chap	oter 3	Hardware Setup	12
Chap	oter 4	Quick Start Procedure for Motor Operation	and
onap		Control	
4.1	Standal	one Mode	15
4.2	PC Mod	e	16
Chap	oter 5	Drive and Motor Protection	17
Chap	oter 6	GUI Operation	18
• 6.1		• Kit Virtual UART for USB Installation	
6.2	GUI Soft	ware Installation	22
6.3	To use C	GUI Software	23
6.4	Set the	Serial Port Number	25
6.5	Run the	Motor Using Speed Control Mode	26
6.6	Change	the PID Settings	26
6.7	Change	the Setup Settings	28
6.8	Change	the Startup Settings	29
Chap	oter 7	Flash Programming the 78K0R-SPINIT Kit (USB)	31
7.1		SPINIT Switch Settings	
7.2		h the Comm Port	
7.3	Start the	WriteEZ4 Application	33
7.4	WriteEZ	4 Device Setup	34
7.5	WriteEZ	4 Hex File Selection	36
Chap	oter 8	Flash Programming with a Flash Programmer	39
8.1	Downloa	ad the Following Files	39
8.2	Save an	d Uncompress the IAR Workbench Project	40
8.3	Switch S	Settings and connecting the MINICUBE2	40
Chap	oter 9	Using the IAR Embedded Workbench	46
9.1	Softwar	e Installation	46
9.2	78K0R-8	SPINIT Setup for On-chip Debugging Using USB	47
9.3	78K0R-8	SPINIT Setup for On-chip Debugging Using MINICUBE2	48
9.4	IAR Emb	bedded Workbench Start Up	48
9.5	General	Options	51
9.6	Compile	r Options	54
9.7	Assemb	ler Options	57
9.8	Linker C	options	59
9.9	-	ed Debugger Selection	
9.10	Workspa	ace and Project Setup if Not Compatible with the Installed IAR Workbench	1 63

9.11	Build/Rebuild the Project	64
9.12	Debugging	64
Chap	ter 10 Appendix	70

Chapter 1 Introduction

The 78K0R-SPINIT kit is a complete 3-phase motor control evaluation system based on NEC Electronics' µPD78F1235 microcontroller (MCU), a 16-bit application-specific product (ASSP) specifically designed for motor control applications.

The kit contains all of the hardware and software necessary to quickly set up and run a low-voltage brushless DC motor (BLDCM). On-board hardware facilitates easy programming with either the MINICUBE2 On-Chip debugger/programmer or directly through the onboard USB port using a standard USB cable, and source code debugging without the need for additional hardware tools.

The source code and project files are available free of charge and can be downloaded from the *NEC motor control website*

In order to modify the source code the IAR development environment is required and a time limited version can be acquired directly from the IAR website or a code limited trial version is included in the MINICUBE2 On-Chip debugger/programmer or the full IAR development tool can be purchased from NEC or through an NEC Electronics franchised distributor.

Please Note: If you require debugging the 78K0R-SPINIT kit while using the USB serial port for system communications a MINICUBE2 will be required.



Figure 1-1 78K0R-SPINIT Kit Image

Chapter 2 78K0R-SPINIT Kit Contents

The 78K0R-SpinIt KIT should contain the following items:

- MC-78K0R-IE3 single board controller containing the µPD78F1235 ASSP microcontroller
- BLDC motor: Anaheim Automation BLY17S15V8000 (15V)
- FW75550/15 DC power supply
- USB Cable

For information about the electrical characteristics and hardware functions of the μ PD78F1235 microcontroller, refer to 78K0R/IE3 (iPD78F1235) User's Manual (U19163EJ1V0UD00).

For the instruction descriptions, refer to the 78K0R 16-bit Single-Chip Microcontrollers Instruction Manual (U17792EJ4V0UM00).

Chapter 3 Hardware Setup

The kit is shipped with a default jumper configuration to run the motor with the following on-board user interface controls as soon as the motor is connected and power is supplied.

- START/STOP, FORWARD, REVERSE and MODE push-buttons
- SPEED potentiometer
- 7-segment LED

Table 3-	1 DS1	Dip Switch	Settings
----------	-------	-------------------	----------

Switch		USB Debug + Flash Programming	MINICUBE2 + (GUI Operation)
	1-8	ON	ON
D01	2-7	OFF	OFF
DS1	3-6	OFF	ON
	4-5	ON	OFF
JP22	Debug Mode	OFF = USB	ON = MINICUBE2

Table 3-2 Jumper Settings

Jumper	Description	Default Setting
JP5	MINICUBE2 RX - TX link	ON
JP6	1-2 Reset_USB (MINICUBE2) 3-4 Reset_KR (MINICUBE2)	3-4
JP7	1-2 FLMD0_USB (MINICUBE2) 3-4 FLMD0_KR (MINICUBE2)	3-4
JP9	1-2 Comparator1 (Phase U BEMF) 3-4 Hall1	3-4
JP10	1-2 Comparator2 (Phase V BEMF) 3-4 Hall2	3-4
JP11	1-2 Comparator3 (Phase W BEMF) 3-4 Hall3	3-4
JP12	U-Hi FET drive isolate	ON
JP13	V-Lo FET drive isolate	ON
JP14	U-Lo FET drive isolate	ON
JP15	W-Hi FET drive isolate	ON
JP16	V-Hi FET drive isolate	ON
JP17	W-Lo FET drive isolate	ON
JP18	U phase isolate (from onboard FET)	ON
JP19	V phase isolate (from onboard FET)	ON
JP20	W phase isolate (from onboard FET)	ON
JP21	Motor Over current Protection Hardware Circuit	ON
JP22	ON = MINICUBE2 Debug Mode OFF = USB Debug Mode	ON

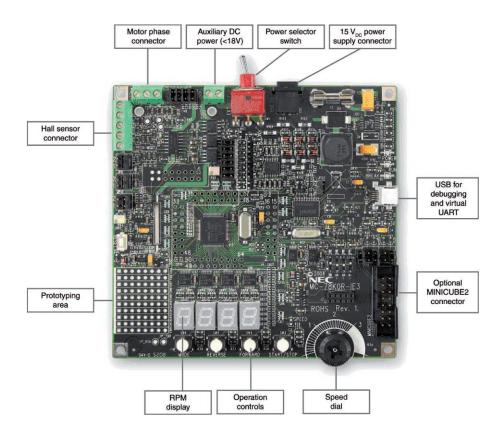


Figure 3-1 78K0R-SPINIT Controller Board Layout

To connect the motor, attach the phase terminals to connector CN1 and the Hall sensor terminals to connector J5, as shown in *Table 3-3* below.

- **Note** Other low-voltage BLDC motors with similar characteristics may be used. Consult the motor specifications and make appropriate changes, including software if necessary based your own assessment.
- Table 3-3Motor Connections

Motor Terminals		78K0R-SpinIt	
Function	Anaheim Automation BLY171S-15V-8000		
Phase U	Yellow	CN1-1	
Phase V	Red	CN1-2	
Phase W	Black	CN1-3	
Hall sensor 1	White	J5-1	
Hall sensor 2	Blue	J5-2	
Hall sensor 3	Green	J5-3	
Hall sensor 5 V _{DC}	Red	J5-4	
Hall sensor GND	Black	J5-5	

Motor terminal connections for Anaheim Automation BLY171S-15V-8000 motor are as shown below.

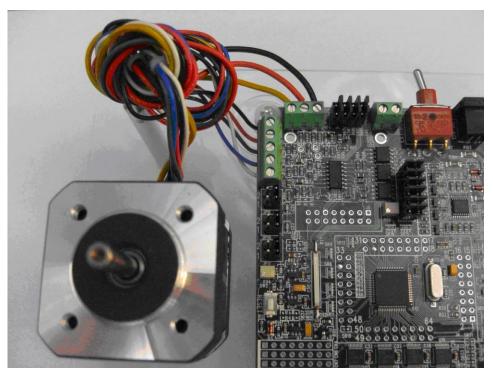


Figure 3-2 Motor Connections Image

The software to run the motor is pre-programmed into the microcontroller's flash memory and it is ready to run as soon as the power supply is connected and the power switch is turned on.

A quick start-up procedure is outlined in the chapter 4 below.

Chapter 4 Quick Start Procedure for Motor Operation and Control

The operation and control of the 78K0R-SPINIT starter kit and motor can be done in two ways:

- Standalone mode with onboard pushbuttons, potentiometer and 7-Seg LED
- PC mode with a PC based GUI and the onboard USB connector

4.1 Standalone Mode

The motor can be operated in standalone mode right out of the box after the motor terminals and hall sensor wires are connected and power is supplied. To operate in PC GUI mode a special NEC program "necgui.exe" program has to be installed. See the installation instructions and the operation in PC GUI mode in chapter 6 of this user manual.

To run the motor in standalone mode, connect the 15VDC power supply to CN2 connector and flip SW7 power selector switch towards CN2. At this point the green POWER LED should turn on and the 7-Segment LED should display "SELF". This is an indication that the motor is ready to be operated from the on-board pushbuttons and potentiometer.

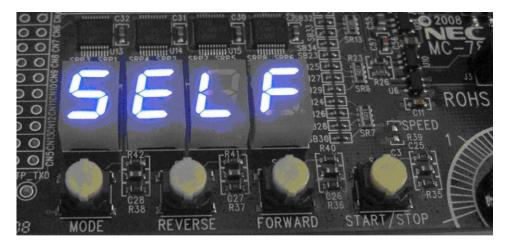


Figure 4-1 Power On Message

After three seconds the LED will display the reference RPM and can be set to the desired speed between 300 to 5000 RPM by the onboard potentiometer. The reference RPM is identified by the decimal point lit on the "units" digit. If the MODE button is pressed the display will toggle between reference RPM and measured or feedback RPM. The measured RPM is displayed without the decimal point.



Indicates in Reference Speed Mode -

Figure 4-2 Reference RPM Display

To start the motor, press the START/STOP pushbutton.

To stop the motor, press START/STOP again.

During the motor operation, the RPM can be set to the desired speed using the SPEED potentiometer.

Pressing MODE during operation will toggle the display between the reference RPM and the actual measured RPM.

Changing the direction can be done while the motor is running with the REVERSE and FORWARD push buttons. The motor will ramp down and stop for a brief time and will ramp up in the opposite direction to the same set RPM.



Figure 4-3 Measured RPM Display

4.2 PC Mode

In PC Mode the motor can be operated using a special GUI (Graphical User Interface) which can be downloaded from NEC Electronics website. See chapter 6 for the description of this mode of operation

Chapter 5 Drive and Motor Protection

The starter kit and the motor are protected against unexpected events such as overload, motor stall and malfunction of the Hall sensors. If such faults are detected, the motor stops rotating and the fault conditions are displayed on the seven-segment LED. For details on the protection functions implemented in hardware, consult the user's manual schematics. The sample code software also has built-in fault detection algorithms as an extra measure of protection.

In standalone (SELF) mode, the LED displays the following fault conditions:

- Motor over-current:
- Motor stall fault:
- Hall sensor fault:
- Software over-current:



In the Graphical Interface mode (PC), the LED displays "P.C." all the time and the GUI will display all fault conditions. Refer to section 6 for more information.



On power on the 78K0R-SPINIT kit will display "SELF":



Chapter 6 GUI Operation

The following information covers the installation and use of the remote control GUI application software.

The interface for the PC GUI is already built into the example software programmed into the board. so it Is not necessary to reprogram the device in order to use the GUI operation. If for any reason it is necessary to reprogram the device the complete IAR Embedded Workbench project can be downloaded from the Motor Control starter kit web site.

To reprogram the microcontroller please refer to section 7.

To operate the PC GUI you will need the USB cable included in the 78K0R-SpinIt starter kit and to install both the NEC Electronics Starter Kit Virtual UART driver and the NEC GUI application software.

6.1 Starter Kit Virtual UART for USB Installation

In order to use the remote control GUI you must install the NEC Electronics Starter Kit Virtual UART. First connect the USB cable to the 78K0R-SpinIt as shown in *Figure 6-1* below and then to the PC.

Prepare the 78K0R-SPINIT for GUI operation as shown below and ensure that DS1 is set to MINICUBE2 + GUI Operation settings (see *Table 3-1*).

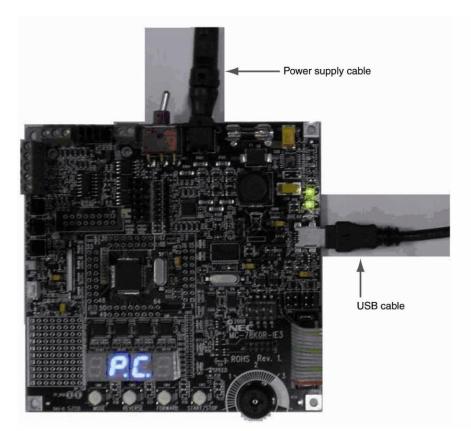


Figure 6-1 Connect USB cable to 78K0R-SPINIT



Windows will start the Found New Hardware Wizard. Select the "No, not this time" option.

Figure 6-2 Found New Hardware Screen

Once Windows has identified the hardware select the "Install from a list or specific location (Advanced) option".

Found New Hardware Wizard	
	This wizard helps you install software for: NEC Electronics Starter Kit Virtual UART
	If your hardware came with an installation CD or floppy disk, insert it now.
	What do you want the wizard to do? C Install the software automatically (Recommended)
	 Install from a list or specific location (Advanced) Click Next to continue.
	<back next=""> Cancel</back>

Figure 6-3 Install From List or Specific Location Screen

Select the "Search for the best driver in these locations" option and select the "Include this location in the search" box and in the browse box point to the location of the downloaded driver folder "MQB2" and click on the "Next >" button.

	rdware Wizard oose your search	and installa	tion options		5.0
	0.140.400000000000000000000000000000000		00590000 . 50090000080		No.
🖲 Sea	arch for the best drive	er in these local	tions.		
	the check boxes be is and removable me				cludes local
Г	Search removable	e media (floppy,	.CD-ROM)		
F	Include this locati	on in the searcl	h:		
	C:\MQB2			- Bro	owse
C Dor	n't search. I will choo	se the driver to	install.		
	ose this option to sel driver you choose wi				ot guarantee
				é i	

Figure 6-4 Choose Your Search and Installation Options Screen

Click on the "Continue Anyway" button which begins the driver installation.



Figure 6-5 Continue to Install the Virtual UART

Click on the "Finish" button which will complete the driver installation.



Figure 6-6 Finish Installation and Close the Wizard Screen

Please make a note of the COM Port number that windows has assigned to the virtual driver as you will need to select this port from the GUI. Using your mouse Right click on **My Computer** and select **manage**. From the computer management screen select **Device Manager** and then **Ports (COM & LPT)**. You should now see the **NEC Electronics Starter Kit Virtual UART**.

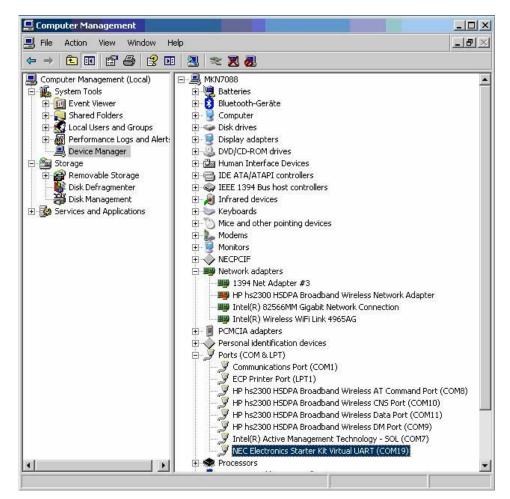


Figure 6-7 Display Virtual UART Port Number in Windows Device Manager Screen

The 78K0R-SpinIt starter kit uses the following UART settings:

Baud rate	57600
Data Bits	8
Stop Bits	1
Parity	None
Handshake	None

6.2 GUI Software Installation

The zip file "**NECGUI.zip**" contains all the files needed to install the NEC GUI application software. Simply extract the files to a folder named "**NECGUI**" and then click on the "**setup.exe**" application as shown below

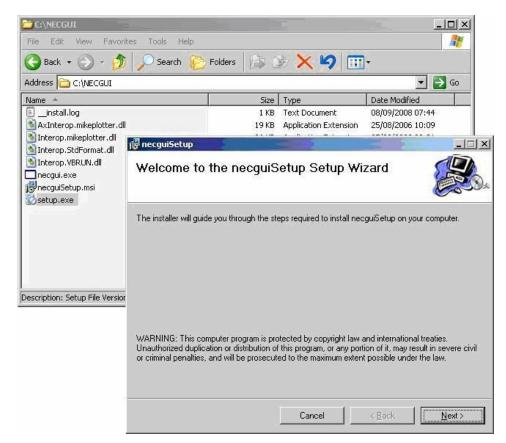


Figure 6-8 Launch the necguiSetup Wizard

Follow the instructions until the installation is complete.

6.3 To use GUI Software

Start the NEC GUI application software by selecting it from the programs list:



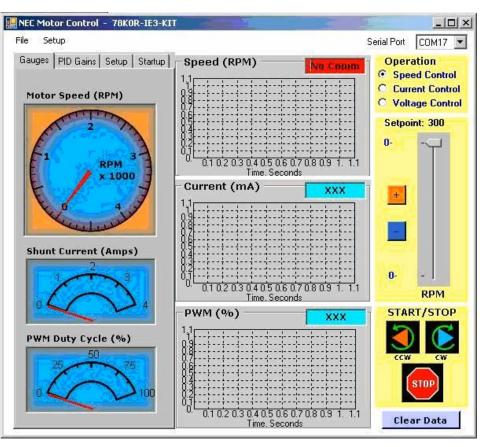
Figure 6-9 To Start NEC GUI

After selecting the program you may see a "Comm error" dialogue box appear

Error	X
	Comm error, try again
	ОК

Figure 6-10 NEC GUI Comm error dialogue box

If this occurs then simply click on "**OK**" as many times as the box appears. You will be able to set the serial port number in the application software.



The application will launch and you should see the main user interface as shown below.

Figure 6-11 NEC GUI Main Display View

6.4 Set the Serial Port Number

The first step should be to set the serial port to the correct port number.

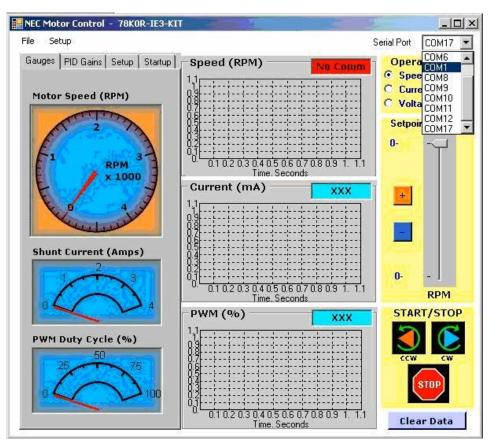


Figure 6-12 Setting the Comm Port Number

The Speed (RPM) text box will display "No Comm" when communications is interrupted

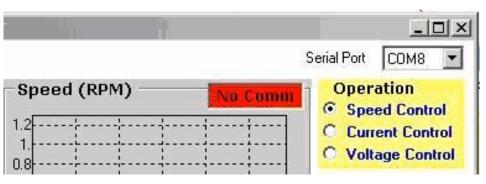


Figure 6-13 No Comm Displayed

Once communications is established then the Speed (RPM) text box will display the speed and the GUI can be used as normal.

6.5 Run the Motor Using Speed Control Mode

To operate the motor in Speed Control Mode, select Speed Control from the Operation menu (see below) and use the controls in the GUI window. The user has the same controls as described for the "Stand Alone" mode (Start / Stop, Clockwise / Anti-Clockwise, Speed increase / decrease)

🚂 NEC Motor Control - 78K0R-IE3-KI			_ 🗆 ×
File Setup	S	erial Port	СОМ17 💌
Gauges PID Gains Setup Startup	Speed (RPM) No Comm	SpeCurr	ation ed Control rent Control age Control

6.6 Change the PID Settings

It is also possible to change the PID parameters from the GUI interface. (Please note that changes can only be made when the motor is stopped). To change the PID parameters click on the "**PID Gains**" tab. The PID Gains editor will be displayed as shown below. Please use the "**RPM to Current Gains**" as this is for speed control.

Кр	0.15
Ki	0.005
Kd	0.001
Curr	ent to PW/M Gains —
Кр	1
Ki	0
Kd	0
S	end Gains to uC
G	et Gains from uC

Figure 6-15 GUI PID Gains Settings

To read back the current PID settings from the 78K0R-SPINIT click on the "**Get Gains from uC**" button and the GUI will be updated.

Changes can be made in this view and sent down to the 78K0R-SPINIT. Simply make the appropriate changes and click on the "**Send Gains to uC**".

To make the changes permanent, the user will have to modify the initialised values in the original IAR project files (main_mcio.c) and then rebuild the project. This requires a licensed version of either IAR Embedded Workbench (Full Version) or IAR Embedded Workbench (Kick Start Version)

Find the function calls in the file "**main_mcio.c**" and change the values that are passed to the functions.

Set these parameters to the values shown in the Tuning Window of the GUI:

Motor_SetSpeedKp 0.150 Motor_SetSpeedKi 0.005 Motor_SetSpeedKd 0.001

Please note that only Speed Control Mode is described in this manual. The full version of the NEC GUI manual (Motor Control Graphical User Interface Users

Manual) will further discuss all other operational modes and settings. While the other control modes shown in the GUI will operate the motor this operation and subsequent performance cannot be guaranteed.

The term "Motor Tuning" is defined as the adjustment of motor start values, motor stop values, and the PID values in order to improve the motor speed tracking control during normal running.

6.7 Change the Setup Settings

It is also possible to change the Speed Limits, Current Limits, and Current A/D parameters by selecting the following tab as shown below:

Current	A/D Parame	010	/
Gain	1		
Offset	0		
mA = G A/D = 0	ain*(A/D - Off I-1023	set)	
Current	Limits (mA)	đ	
Max	1023		
Min	10		
Max Ra (mA/sec)	te 900		
mA = G A/D = 0	ain*(A/D - Off I-1023	set)	
Speed	Limits (RPM)		
Max	5000		
Min	300		
Max Ra (RPM/se	te 4000 c)		
		uC	

Figure 6-16 GUI Setup Settings

To read back the current Setup settings from the 78K0R-SPINIT click on the "**Get Parameters from uC**" button and the GUI will be updated.

Changes can be made in this view and sent down to the 78K0R-SPINIT. Simply make the appropriate changes and click on the "**Send Parameters to uC**".

Current A/D Parameters

- Gain for current shunt value amplification
- Offset to correct any known constants

Current Limits (mA) – Used for current control mode not covered in this document

Speed Limits (RPM)

- Max the maximum RPM speed setting for the motor
- Min the minimum RPM speed setting for the motor
- Max Rate the acceleration/deceleration rate in RPM/sec

6.8 Change the Startup Settings

It is also possible to change the Open Loop Time, RPM for Open Loop, and starting PWM settings by selecting the following tab as shown below:

Gauges	s PID Ga	ains Setup	Startup	
Time (sec) RPM Curren (mA)	Initial 0 60 t 140	Middle 75 100 160	Final 1.5 200 160	
PWM (%)		10 ent Control age Control	2.5	Ensure "Voltage Control" is selected
Ĩ		arameters to u	۲.	
	Get Para	meters from (uC	

Figure 6-17 GUI Startup Settings

To read back the current Startup settings from the 78K0R-SPINIT click on the "**Get Parameters from uC**" button and the GUI will be updated.

Changes can be made in this view and sent down to the 78K0R-SPINIT. Simply make the appropriate changes and click on the **"Send Parameters to uC"**.

Time (sec) – start in open loop until final time is reached then switch to closed loop control

- Initial the length of time in seconds to run open loop up to the set RPM and PWM%
- Middle the length of time in seconds to run open loop up to the set RPM and PWM%
- Final the length of time in seconds to run open loop up to the set RPM and PWM%

RPM

• The startup rpm speed for each phase initial, middle, and final

Current (mA) – Used for current control mode not covered in this document

PWM%

• The startup max PWM% for each phase initial, middle, and final

Chapter 7 Flash Programming the 78K0R-SPINIT Kit (USB)

If the user wishes to modify the program in the 78K0R-SPINIT then this can be done either with the onboard USB interface or using a separate Flash Programmer. This chapter will describe how to flash the microcontroller using the onboard USB interface.

Download the WriteEZ4 USB Flash Programming Graphical Interface (from the *NEC Electronics Tool download web site*) and install.

7.1 78K0R-SPINIT Switch Settings

To prepare the 78K0R-SPINIT for flash programming follow the steps below:

- Power OFF the 78K0R-SPINIT kit
- Set the DIP switch DS1 to the following:
 - 1-8 **ON**
 - 2-7 **OFF**
 - 3-6 **OFF**
 - 4-5 **ON**
- Remove JP22 (OPEN = USB Flash/Debugging mode)
- Connect the USB cable from the PC to the 78K0R-SPINIT kit
- Power **ON** the 78K0R-SPINIT kit

7.2 Establish the Comm Port

The WriteEZ application requires a virtual comm Port so please note the appropriate comm Port number. Use "Device Manager" to view the ports.

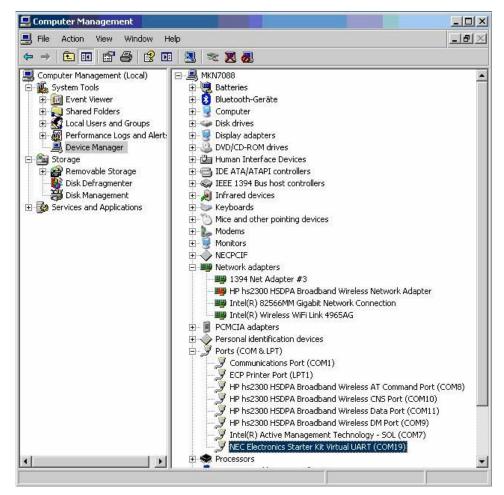


Figure 7-1 Display virtual UART port number in Windows Device Manager screen

7.3 Start the WriteEZ4 Application

Start -> All Programs -> NEC Electronics Tools -> WriteEZ4 -> v1.02 -> WriteEZ4

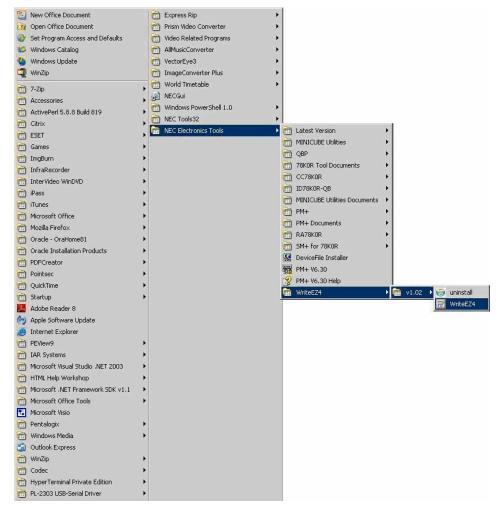


Figure 7-2 Starting WriteEZ4

The WriteEZ4 application will launch.

/P 🖦 🗔 🖏 🖋 🐉 🗞	3
>> FlashOpenning Flash Open OK >> ParameterFile Reading Success Read ParameterFile. >> LoadFile Reading Success read HEX file.	Device Name : Firm Version : ExtCode : Vendor : Name : 78F1235 Format : 0419 Version : E1.00a Processor Ver. 0200 Load file Name BLDC_HALL120_78K0F Date : 2009/03/05 09:39:58 Chksum :D241h Area : 000000h-00263Ah
	Connection to device Port : COM20 Speed 115200bps Range Chip Freq : Internal-OSC Multiply : 1.00

Figure 7-3 WriteEZ4 Screen

7.4 WriteEZ4 Device Setup

From the WriteEZ4 main menu select Device -> Setup ... and the Device Setup screen will appear. From the Host Connection box set the Port to the correct number to use the virtual UART port (displayed under device manager ... see above).

Parameter file 7	8F1235.prm		PRM File I	Read
Host connection-		- Supply oscilla	ator	
Port CO	M20 💌	Frequency	Internal-OSC	MH2
Speed 115	5200bps 👻	Multiply rate	1.00	
Operation Mode		<u>1.:</u>		
Chip	Start 000	7		
C Block	End 063	*		
C Area	L SH	iow Addres		

Figure 7-4 WriteEZ4 Device Setup Standard

Next ensure that the PRM file is correctly displayed. In this case it should be 78F1235.prm. If this is not the displayed PRM file then select the "PRM File Read" and select the 78F1235.prm file which is included in the project file.

	4-78F1235_E100A	- + E C	
378F1201.prm	🔤 78F1223.prm		
🗐 78F1203.prm	🚾 78F1224.prm		
🗟 78F1211.prm	78F1225.prm		
🗐 78F1213.prm	🛅 78F1233.prm		
🗟 78F1214.prm	🛅 78F1234.prm		
🗟 78F1215.prm	🚾 78F1235.prm		
-			
ïle name: 🏼 🕅 *.p	m	J	Open
iles of type: PR	M Files(*.PRM)		Cancel

Figure 7-5 WriteEZ4 Select Parameter File

Keep the advance settings as detailed below.

Command options	
Blank check before Erase	🗖 Wide Voltage mode
Read verify after Program	
🦳 Security flag after Program	
🔲 Checksum after Program	
Security flag settings	
Disable Chip Erase	Reset vector 00000000 h
🔲 Disable Block Erase	Boot block end 003 💌
🔲 Disable Program	FS Block start 000 💌
📕 Disable Read	FS Block end 063 💌
Disable Boot block cluster reprogra	ammir 🗖 Show Address
Target Reset Message	

Figure 7-6 WriteEZ4 Device Setup Advance

7.5 WriteEZ4 Hex File Selection

Next press the select the "HEX" file to be programmed from the Menu File -> Load

Or press the "Load File" symbol in the ICON taskbar.

The following window should open:

Open		and the second s	<u>? ×</u>
Look in: 🔀) Exe	- + E	💣 🎹 •
BLDC_HAL	L120_78KORIE3.hex		
File name:	BLDC_HALL120_78K0RIE3.h	iex	Open

Figure 7-7 WriteEZ4 Hex File Selection

Locate the file as shown in the window from the downloaded IAR project. The file will be located as follows:

\$saved directory\$\BLDC_HALL120_78K0RIE3\Debug\Exe\

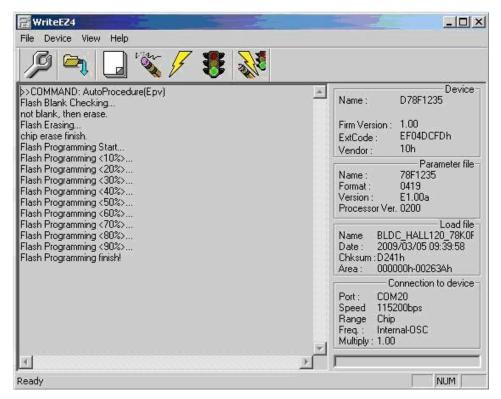
Select the file (BLDC_HALL120_78K0RIE3.hex) and press the "OPEN" button

This will close the "LOAD" file window. The following should be displayed on the main screen

> Open Load File Success read Load file.

The Flash programming setup is now complete

Now press the "AUTOPROCEDURE" button to start the programming sequence. The following sequence should be seen:



- Figure 7-8 WriteEZ4 Autoprocedure
 - **Note** The "Verify Chip" operation is enabled by setting the "Read Verify after Program" option in the "Advanced" Tab of the "Device Setup" Window.

The 78K0RIE3 device has now been reprogrammed with the example program which is suitable for both standalone operation, and remote operation using the GUI.

Please make a backup copy of the original hex file before programming or running the compiler.

Chapter 8 Flash Programming with a Flash Programmer

This section describes the steps necessary to modify the program in the 78K0R-SPINIT microcontroller using a Flash Programmer (not included in the Starter Kit).

Two programmers are available:

- PG-FP5 Full Programmer
- MINICUBE2 On-Chip debugger/programmer



Figure 8-1 PG-FP5 Programmer



Figure 8-2 MINICUBE2 On-Chip debugger/programmer

The Graphical Interface for either of these programmers can be downloaded from the *NEC Electronics Development tools web site*.

In this guide we have shown only the MINICUBE2. The interface for the microcontroller board is the same for both programmers.

8.1 Download the Following Files

Download the appropriate *IAR project folder* which contains all the required source code for the application.

Download the *MINICUBE2* (QB programmer) Flash Programming Graphical Interface.

Download the Appropriate IAR Work bench from the *IAR web site* or use the KickStart CD supplied with the MINICUBE2. The full IAR development tool can be purchased from NEC Electronics or through an NEC Electronics franchised distributor.

Uncompress and install the QB programmer GUI software (run the "SETUP" application).

8.2 Save and Uncompress the IAR Workbench Project

Uncompress the folder containing the 78K0R-SPINIT project to a local folder. This folder contains all the source code and IAR environment information required to build and compile both the debug files and the hex flash files.

8.3 Switch Settings and connecting the MINICUBE2

To prepare the 78K0R-SPINIT for flash programming follow the steps below:

- Power OFF the 78K0R-SPINIT kit
- Set the DIP switch **DS1** to the following:
 - 1-8 **ON**
 - 2-7 **OFF**
 - 3-6 **ON**
 - 4-5 **OFF**
- Install **JP22** (ON = MINICUBE2 flash programming mode)
- Check that the switches on the MINICUBE2 are set as shown below
 - Switch M1 / M2 is set to "M1"
 - Switch 3 T 5 is set to "T"
- Locate the MINICUBE2 16-pin connector (see Figure 2) and attach the MINICUBE2 programmer using the 16-pin cable
- Power ON the 78K0R-SPINIT kit

See the MINICUBE2 connected to the 78K0R-SPINIT kit as shown below:

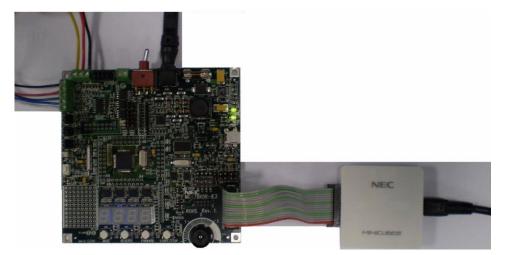


Figure 8-3 Connecting the MINICUBE2 to the 78K0R-SPINIT Kit

Copy the uPD78F1235 Flash programming parameter file (78F1235.prm) into the Installation directory for the QB programmer program or into a specific area which can be reached by the browse facility:

\$installation path\$\....\QBP\PRM
(This file is included with the IAR project download)

Once the Microcontroller board is configured and the programmer connected to the PC, open the Flash programming Graphical interface "QBP v2.22".

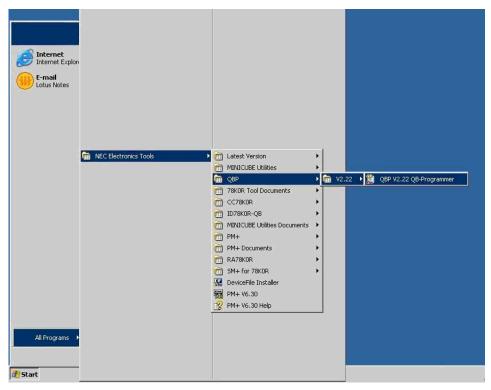


Figure 8-4 Starting the QBP V2.22 QB Programmer Graphical Interface

The following Screen should appear. (Note the text may differ after the 1st two lines.)

Chapter 8

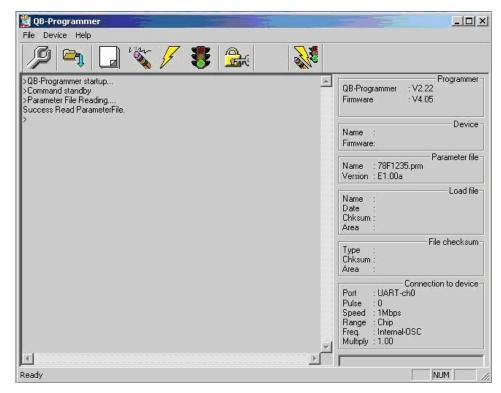


Figure 8-5 MINICUBE2 Programmer GUI (QB Programmer)

Next the device needs to be set up from the menu follow the following sequence: Device -> Setup...

Or press the "Spanner" symbol in the ICON taskbar

The following screen should appear:

ß 🍳 [evice Setup	×
>QB-Programmer startup >Command standby >Parameter File Reading	Standard Advanced	: V2.22 : V4.05
Success Read Paramet Device Setup Cancel Device Setup.	Parameter File 78F1235.prm PRM File Rea	d Device
Device Setup	Target Device Connection Supply Oscillator	Parameter file 5.prm
	Port UART-ch0 Frequency Internal-OSC MF Speed 1Mbps Multiply rate 1.00	fz Load file
	Operation Mode	
	Image: Chip Start 000 Image: Chip Start 000 Image: Chip Start Image: Chip Image: Chip Start 000 Image: Chip Start Image: Chip Image: Chip Start 000 Image: Chip Start Image: Chip Image: Chip Image: Chip	File checksum
	Show Address	Connection to device
	OKCan	
- d		: 1.00

Figure 8-6 QB Programmer Device Setup

Set the details for the COM port, Speed etc as shown above.

Note Enter your COM port number as this will vary.

Next press the "PRM File Read" button and the following screen should appear. Select the 78F1235.prm file and press the "Open" button.

QB-Programmer File Device Help			_		_0×
SQB-Programmer startu	Device Set		c 	X	Programmer -
>Command standby >Parameter File Reading Success Read Param	~	Advanced		1 ? ×	V2.22 V4.05
> >Device Setup Cancel Device Setup.	-	PRM-78F1235_E100A	• • •	ð 💣 🎟 •	Device -
> >Device Setup	78F1201.	orm 🛅 78F1224.prm			Parameter file - prm
	78F1213. 78F1214. 78F1215.	orm 🛅 78F1234.prm			Load file
	File name:	78F1235.prm		Open	File checksum
-	Files of type:	PRM Files(*,PRM)		Cancel	onnection to device -
			OK	Cancel	-osc
व			ج الح	Multiply : 1.00	1000
Ready					

Figure 8-7 Parameter File Read

Then press the "OK" button to return to the main menu system.

The text in the main screen should read

> Device Setup Parameter File Read Pass >

Next press the select the "HEX" file to be programmed from the Menu.

File -> Load

Or press the "Load File" symbol in the ICON taskbar.

The following screen should open:

QB-Programmer File Device Help						_ _ ×
19 🗖 🗖	a ver	F		200		
>QB-Programmer startup >Command standby >Parameter File Reading Success Read Param Op				-	QB-Programmer Firmware	V2.22 V4.05
> >Device Setup Parameter File Read F >	_ook in: 障	Exe L120_78KORIE3	eval.hex	• 🗢 🖻	<u>c</u> *	Device
>Open Load File Cancel Load File Rea > >Open Load File Cancel Load File Rea						Parameter file - prm Load file -
>Open Load File	ïle name:	BLDC_HALL12	20_78K0RIE3_eval.	hex	Open	File checksum
F	iles of type:	S-rec / Hex file	es (*.rec;*.s*;*.hex)	_	Cancel	onnection to device
4				÷.	Pulse : 0 Speed : 1Mbps Range : Chip Freq. : Internal- Multiply : 1.00	OSC
Ready						

Figure 8-8 Hex File Selection

Locate the file as shown in the window from the downloaded IAR project. The file will be located as follows:

\$saved directory\$\BLDC_HALL120_78K0RIE3\Debug\Exe\

Select the file (BLDC_HALL120_78K0RIE3.hex) and press the "OPEN" button.

This will close the "LOAD" file window. The following should be displayed on the main screen:

> Open Load File Success read Load file.

The Flash programming setup is now complete.

Now press the "AUTOPROCEDURE" button to start the programming sequence. The following sequence should be seen:

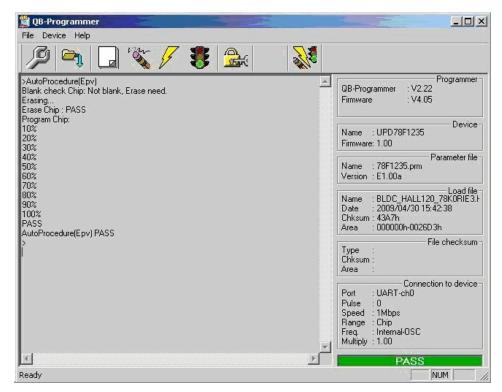


Figure 8-9 QB Programmer AutoProcedure

Note The "Verify Chip" operation is enabled by setting the "Read Verify after Program" option in the "Advanced" Tab of the "Device Setup" Window.

The 78K0RIE3 device has now been reprogrammed with the example program which is suitable for both standalone operation, and remote operation using the GUI.

Please make a backup copy of the original hex file before programming or running the compiler.

Chapter 9 Using the IAR Embedded Workbench

The example software for use with the 78K0R-SPINIT kit is for a HALL Sensored BLDC with 120 degree trapezoidal control for standalone/GUI operation. The kit is supplied with the 78K0R-SPINIT pre-programmed.

The complete example project program for the IAR 78K Embedded Workbench development tool environment can be downloaded from the *motor control web site* as detailed in starter kit package (i.e. where this manual was downloaded).

The software is supplied in source format and can be modified as required.

The following sections describe IAR 78K embedded Workbench development tool environment, how to install it on your computer, and how to rebuild and download executable code to the microcontroller's flash memory.

Before proceeding with the tools installation, however, refer to all of the documentation for the starter kit, On Chip Debugger tool and the IAR Embedded Workbench.

(Please note that a 16 Kbyte code limited version is included with the MINICUBE2 On-Chip Debugger/Programmer unit and can be used to run the example software.)

Please note that a Flash Programmer, On-Chip debugging/programming tool or IAR Embedded workbench are not included in this package. These items are available from your local NEC Electronics Distributor or contact your local NEC Electronics sales office.

9.1 Software Installation

- 1. If a version of the IAR tool is not already installed, then install the IAR embedded Workbench tool as per the instructions provided by IAR.
- 2. Ensure that if not already that the example software has been downloaded from the NEC starter kit web site and "unzipped" into an suitable location.

The example software can operate on any revision of either the IAR Kick Start or IAR Full versions. However it may be necessary to define your own project and workspace. This is described later in this chapter.

9.2 78K0R-SPINIT Setup for On-chip Debugging Using USB

To prepare the 78K0R-SPINIT kit for debugging using only the onboard USB interface, follow the steps below.

- Power OFF the 78K0R-SPINIT kit
- Set the DIP switch **DS1** to the following:
 - 1-8 **ON**
 - 2-7 OFF
 - 3-6 **OFF**
 - 4-5 **ON**
- Remove **JP22** (OPEN = USB Flash/Debugging mode)
- Connect the USB cable from the PC to the 78K0R-SPINIT kit
- Power **ON** the 78K0R-SPINIT kit

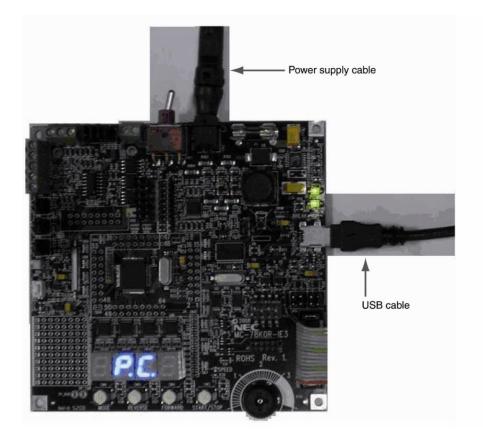


Figure 9-1 78K0R-SPINIT ready for USB debugging

9.3 78K0R-SPINIT Setup for On-chip Debugging Using MINICUBE2

To prepare the 78K0R-SPINIT kit for debugging using the MINICUBE2, follow the steps below.

- Power OFF the 78K0R-SPINIT kit
- Set the DIP switch **DS1** to the following:
 - 1-8 **ON**
 - 2-7 OFF
 - 3-6 **ON**
 - 4-5 **OFF**
- Install **JP22** (ON = MINICUBE2 flash programming mode)
 - Check that the switches on the MINICUBE2 are set as shown below
 - Switch M1 / M2 is set to "M1"
 - Switch 3 T 5 is set to "T"
- Locate the MINICUBE2 16-pin connector (see Figure 3-1) and attach the MINICUBE2 programmer using the 16-pin cable
- Power ON the 78K0R-SPINIT kit

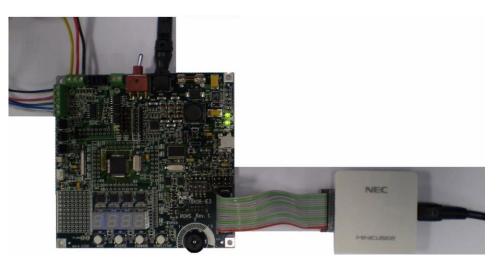


Figure 9-2 78K0R-SPINIT Kit ready for debug with MINICUBE2

9.4 IAR Embedded Workbench Start Up

Open the IAR workbench. The following Screen should be opened.

Note The exact display may vary depending on if this is a new installation.

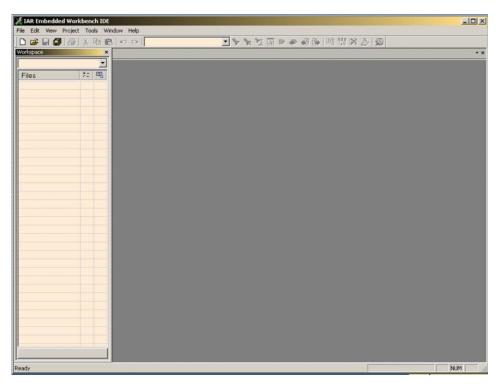


Figure 9-3 IAR Workbench Opening Screen

Next open the IAR Workspace by following the sequence and locating and then selecting the appropriate workspace file as shown below.

File -> Open -> Workspace -> BLDC_HALL120_78K0RIE3

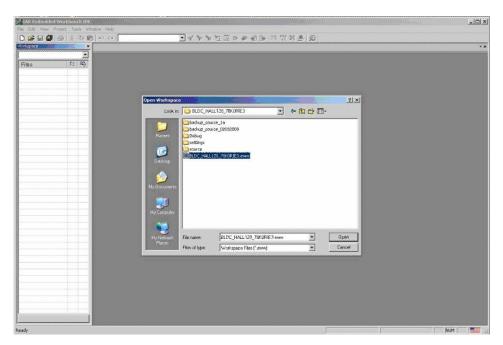


Figure 9-4 IAR Workbench Project Selection

Once the Workspace open the display should look something close to that as shown in the figure below.

This shows the workspace where the project is located and has opened the BLDC project. The display shows the following project files:

Left Hand side window – Project File (Source. Header, Map etc)

- Bottom Build Debug messages when the project is re built or the debugger is active.
- The main centre display shows any open files in a tabbed form. The file can be viewed by selecting the relevant Tab in the wind.

Any of the files shown can be opened by double clicking on the file in the "Project" (left hand side) window. Debugging windows are described later.

Image: Solution of the soluti	IAR Embedded Workbanch ID le Edit View Project Tools N		
Debug man ynick (Instar / Instar, finds,			
Georgenergy */ Files */ Georgenergy */	Vorkspiede		•
Implicit **/**********************************	Constant Constant	<pre>sm puice { noture / junture h sub_maine { juntu</pre>	
BLDC_HALLT0_78KORIE3	B motor. B sub B sub B motor. B sub_mci B sub_mci B sub_mci BLDC_HALL120_78KORIE3	*/ wanigmed char Get_Uart(void): /* UAIT receive */ uartigmed char Beed_UartBuf(void): /* UAIT data buffer roed */ uartigmed char Beed_UartBuf(void): /* UAIT with for new data in buffer */ * void Send Dart(voidnee char): /* "UAIT with for new data in buffer */	

Figure 9-5 Workspace & Project open screen

The build options for the project can then be set or changed using the following menus. The build options are entered as shown in *Figure 9-6* below, and then ensure all the options are set according to the remaining figures below.

💥 IAR Embedded Workbenc	h IDE	Notes the second se		
File Edit View Project Tool	s Window Help			
Add Files Workspace Import File		otor.h sub_mcio.c sub_mcio.h GUI_support	> +> 49 (b) (c) (c) (c) (c) (c) (c) (c) (c) (c) (c	9
Debug Edit Config		h"		
Files Remove		sics.h"		
Create New Create New Add Existin	w Project Ig Project	.235_64.h" .235_64_ext.h"		
Our Options	ALT+F7			
io7 Source Coo	de Control	•		
and a complete compl	F7 CTRL+F7 F8	\$ 		
io7 Stop Build	CTRL+Break	UartBuf[RX_BUFF_SIZE];	/* UART RX buffer	*/
Debug Debug Make & Re Debug Make & Re	CTRL+D start Debugger	UartRead_p; UartWrite_p; UartErrStat;	/* read pointer */ /* write pointer*/ /* UART error code	*/
intrins	/*			

Figure 9-6 Project Build Options

9.5 General Options

Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Device 78K0R - uPD78F1235_64 No DIVUW / MULU Near constant location Mirror ROM 1 Code banking Reg.address: 0xFFF3 Bank address:	nfiguration Library Options Stack/Heap Code model: Near Data model: Near Start address: Size (Kbytes): DxF1000 56.75 No.of banks: 4 Bank size: (Kbytes): 8	• • •
---	---	--	-------------

Figure 9-7 General Options – Setting the Target Device

Category: C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Target Output Library Configuration Library Options Stack/Heap
--	--

Figure 9-8 General Options – Setting the Output Locations

Category: C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Target Output Library Configuration Library Options Stack/Heap Library: Description: CLIB Use the legacy C runtime library. Library file: \$TOOLKIT_DIR\$\LIB\cLIB\clib\clib\clib\clib\clib\clib\clib\clib

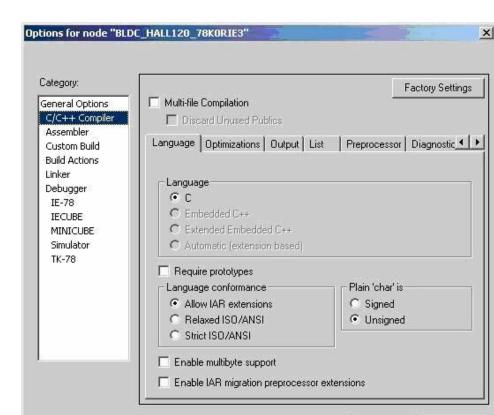
Figure 9-9 General Options – Selecting the C-Library

Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Target Output Library Configuration Library Options Stack/Heap Stack size: 0x80 Heap size Far: 4096 Near: 256 1
---	---

Figure 9-10 General Options – Setting the Stack and Heap

OK

Cancel



9.6 Compiler Options

Figure 9-11 Compiler Options – Language Settings

Category: Seneral Options C/C++ Compiler	Multi-file Compilation	Factory Setting:
Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Language Optimizations Output List I Level None Low Medium High Balanced Enabled transformations: Common subexpression elimination Common subexpression elimination Code motion Ucode motion Uppe-based alias analysis	Preprocessor Diagnostic
	Enable 'callt' runtime library calls	

Figure 9-12 Compiler Options – Optimisation

Category: General Options C/C++ Compiler	Multi-file Compilation Discard Unused Publics
Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Language Optimizations Output List Preprocessor Diagnostic Module type Override default Image: Compare Module Image: Compare Module Image: Compare Module Object module name: Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Compare Module Image: Co

Figure 9-13 Compiler Options – Output Set for Debug

General Options C/C++ Compiler Assembler	Multi-file Compilation	Factory Setting
Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Language Optimizations Output List C Output list file Assembler mnemonics Diagnostics Output assembler file Include source Include call frame information	Preprocessor Diagnostic

Figure 9-14 Compiler Options – Compiler Listings

Category: General Options	Factory Settings
C/C++ Compiler Assembler Custom Build	Discard Unused Publics Language Optimizations Output List Preprocessor Diagnostic
Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Ignore standard include directories TOOLKIT_DIR\$\INC\ TOOLKIT_DIR\$\INC\CLIB\ Additional include directories: (one per line) Preinclude file:
	Defined symbols: (one per line)

Figure 9-15 Compiler Options – Pre Processor Settings

Note All other Compiler Options settings can be remain as the default settings.

9.7 Assembler Options

Language Output List Preprocessor Diagr User symbols are case sensitive Allow mnemonics in first column Allow directives in first column Enable multibyte support Macro quote characters: <>	iostics Extra Options
	 User symbols are case sensitive Allow mnemonics in first column Allow directives in first column Enable multibyte support Macro quote characters:

Figure 9-16 Assembler Options – Language Settings

		<u> </u>	Factory Setting:
2	1.02	Diagnostics	Extra Options
		Language Output List Preprocessor	Language Output List Preprocessor Diagnostics

Figure 9-17 Assembler Options – Output set for Debug

Category:		Factory Setting
Seneral Options		
C/C++ Compiler Assembler		
Custom Build	Language Output List Preproce	ssor Diagnostics Extra Options
Build Actions Linker	Output list file	
Debugger	🔽 Include header	
IE-78	Include listing	Include cross reference
IECUBE MINICUBE	Macro definitions	T #defines
Simulator	Macro expansions	Internal symbols
TK-78	Macro execution info	🗖 Dual line spacing
	Assembled lines only	🗖 Lines/page:
	Multiline code	80
	Diagnostics	
	Structured assembly lines	Tab spacing:
	F #included text	8

Figure 9-18 Assembler Options – Listings

9.8 Linker Options

Category: General Options C/C++ Compiler Assembler Custom Build Build Actions	Output Extra Output #define Diagn	Factory Settings
Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Output file Override default BLDC_HALL120_78KORIE3.d26 Format Obebug information for C-SPY With runtime control modules With 1/0 emulation modules With 1/0 emulation modules Allow C-SPY-specific extra Other Output format: intel-ext Format variant: None Module-local symbols: Include all	output file ended

Figure 9-19 Linker Options – Primary File Output

Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Factory Settings Output Extra Output #define Diagnostics List Config Proce Image: Generate extra output file Output file Image:
---	---

Figure 9-20 Linker Options – Secondary File Output

Category: General Options		Factory Setting
C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78	Output Extra Output #define Dia Always generate output Segment overlap warnings No global type checking Warnings/Errors Suppress all warnings Suppress these diagnostics:	ignostics List Config Proce
	Treat these as warnings: Treat these as errors:	OK Cance

Figure 9-21 Linker Options – Diagnostic settings

General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger IE-78 IECUBE MINICUBE Simulator TK-78 Image: Module suppressed entries Image: Static overlay map	File format Text HTML Lines/page: 80
--	---

Figure 9-22 Linker Options – Generate MAP File Output

Category:	Factory Settin
General Options C/C++ Compiler Assembler Custom Build Build Actions	Output Extra Output #define Diagnostics List Config Proce_
Linker	✓ Override default
Debugger IE-78 IECUBE MINICUBE	\$TOOLKIT_DIR\$\CONFIG\lnk78f1235_64.xcl
Simulator TK-78	Entry labelprogram_start Defined by application Search paths: (one per line)
	\$TOOLKIT_DIR\$\LIB\
	Raw binary image File: Symbol: Segment: Align

Figure 9-23 Linker Options – Linker Command File Selection

Note The Linker Control File must match the device selected. The remainder of the Linker setup can be left as the default setting.

9.9 Integrated Debugger Selection

Caution

The MINICUBE debugger must be selected when using the MINICUBE2 and when using the onboard USB interface. DO NOT SELECT THE TK-78 debugger option.

ategory: eneral Options	Factory Setti
C/C++ Compiler Assembler Custom Build	Setup Extra Options Plugins
Build Actions	Driver
Linker Debugger	MINICUBE QB-78K0MINI, QB-78K0SxxxMINI, QB-MINI2 💌
IE-78 IECUBE MINICUBE Simulator	Run to
TK-78	Setup macros Use macro file
	Device descriptions
	\$TOOLKIT_DIR\$\CONFIG\DDF\io78f1235_64.ddf

Figure 9-24 Integrated Debugger Selection

Category:		Factory Settin
General Options		
C/C++ Compiler Assembler		
Custom Build	Setup Extra Op	otions Plugins
Build Actions Linker	Select plugins t	o load:
Debugger	SEGGER er	
IE-78 IECUBE	Code Cover	
MINICUBE	Profiling	
Simulator TK-78	Stack	
18-70		
	Description:	RTOS awareness for SEGGER embOS
	Location:	C:\Program Files\IAR Systems\Embedded Workbench Kid
	Originator:	Segger Microcontroller Systeme GmbH
	Version:	2.0.5.0
	5	

Figure 9-25 Integrated Debugger - Plug-in Selection

Note The Extra Option section can be ignored.

9.10 Workspace and Project Setup if Not Compatible with the Installed IAR Workbench

1. Set a new workspace File -> New -> Workspace

Create a new Project Project -> Create New Project -> Select "Empty Project" -> "OK"

Enter a project name and set the location for the project (This can be the same location as the downloaded example software or a new location)

3. Add the Source files to the project

C Source Files Project -> Add Files Locate and select all the C source files

- Main_mcio.c
- Sub_mcio.c
- Initialise_hardware.c
- Interrupt_handlers.c
- Motor.c
- GUI_support.c

Press "OPEN"

All these file should now appear in the Project Window (Left Hand Side of the IDE), as shown previously.

9.11 Build/Rebuild the Project

To build the project press the "make" icon in the task bar as shown below:

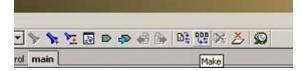


Figure 9-26 Make Button

The build results and any errors or warnings will be displayed in the Messages window at the bottom of the IAR Workbench window. These should be corrected before moving on to the Debugging section.

9.12 Debugging

Once the project has been built without errors the user can now start the debugging session.

This is done by pressing the "Debug" icon in the task bar.

~~~	ø	
	De	bug

Figure 9-27 Start the Debug Session

The debugger will connect to the OCD unit and download the code to the Flash memory on the microcontroller board.

Once downloaded the debugging window will open as shown below in *Figure 9-29*.

**Note** The IAR embedded Workbench provides an integrated debugger, so the debugging window opens as part of the IDE.

If the debugger is run for the first time in a new project the following set up window will open. This is to set the basic function of the debugging hardware (i.e. Mini Cube or IECube etc.).

D code	Time unit	ОК
FFFFFFFFFFFFFFFFFFF	nsec 🔻	
🔲 Erase flash before next ID		Cancel
tain clock	Sub clock	
🔿 Clock board	C Clock board	Default
External	External	Derault
C System	C System	Fail-safe break
None 💌 MHz	None 💌 kHz	☐ View setup
lash programming	Target power off	Target connect
Permit	C Permit	TOOLO
Not Permit	Not Permit	
'in mask	Peripheral break	Target
WAIT TARGET RESET		C Connect
NMI 🗖 INTERNAL RES	A REAL PROPERTY AND A REAL	C Not Connect
femory map		
Start address: Length:	Туре:	
0x0 960	Internal ROM	Add
0x00000 - 0x0FFFF Internal ROM	64 Kbytes	
0xFF300 - 0xFFEFF Internal RAM	3072 bytes	
		Remove

Figure 9-28 Debugging – Initial Hardware Setup

Ensure that the settings are as defined above.

- Notes 1. If the Main Clock shows the "Clock board" detected, then ensure that this is selected.
  - 2. If debugging with a motor connected, check the A (timer) box in the Peripheral break settings (as shown above) to avoid damaging the motor driver devices, fuse, or motor. Checking this box will force all timer output pins to a high impedance state during break conditions.

) 🖡 🕹 L L L L L L L L L L L L L L L L L L							
elsopace ebuo	×	restor.c   autor.h   aut_moio.c   aut_moio.h   GLE_support.c		Disassentbly Gio to		Memory	-
	12 B2		1	017		Hemory HOVU	N:T
Nes @BLDC_HALL129_78K9RIE3 - Debug -데 Ongenetic -데 Ontribatise_herdware.c -데 Ontribatise_herdware.c		//		1 017 017 017 EEG	C0 C4 C1 C2 C2 D7 PHASE U * duty	POP POP RET	DE BC
-e e mein_moio.c		solarce code start		Change_ 017	C3 RE64	MACH	TDR
-B motor.c		***************************************		- 017	CS BE68	HOVU	TDR
-El 🛅 sub_maio.c -El 🎦 Output				REG 017	PHASE W = duty	NOAR WORK	TDR
a a copor				1 017	C9 D7	RET	
		void main(void)		4 017	un(void)	PUSH	BC
		/* Hardware Set-up */		IEPUC 017	CB 711828	CIR1	PHB
		DEBUG_PORT_HODE = OUT;		017	CE 711308	CIRI	S:P
		DEBUG_FORT = 0;		Hotor 017	D1 FD4C13	CALL	N : H
		Notor_SetupPunc();		1011 017	LEOPORT(): D4 FDB425	CALL	N:i
		init_LEDPORT(): /* LED port set-up */ init_TIMER(): /* interval timer */		init_017	TIMER():	CALL	N:1
		anau _{en} annen(), /- anoueyua camea /		Hotor 017	SetNode(NOTOR	NODE SPEED	1.
		/*		017	DB FD4814	CALL	N H
				017	DE F6	CLRW	XA
		set up motor defaults for standalone mode		017	E2 FD8A14	HOWW CALL	BC. N.H
				Hotor 017	E5 F6	CLRW	AX
				017		CALL	BC. N:H
		*/		Notor 017	SetSpeedRef(10	WYWW (00)	AX.
		fif defined (SPINIT)		017		TIAD	N:B
		Motor_SetHods(HOTOP_MODE_SPIED);		017	F2 309600	HOVW	AX. N.H
		Notox SetCarrier(20);		Hotor	SetSpeedKp(0,1	50): //	vas 0.
		Notor_SetDeadtime(3);	*1	017		MOON MACH	AX. BC,
LDC_HALL120_78K0RIE3			ᆀ	•			3
Messages			File			Line	
Building configuration: BLDC_HALL120_7	8K0RIE3 - D	abug					
Updating build tree							
Configuration is up-to-date.							

Figure 9-29 Integrated Debugger - Main Window

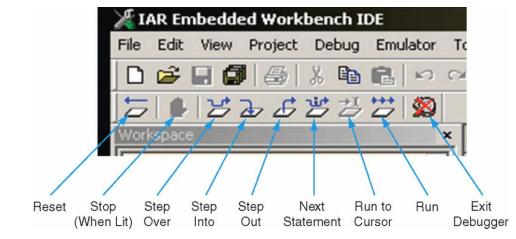


Figure 9-30 Debugger Task Bar Icons

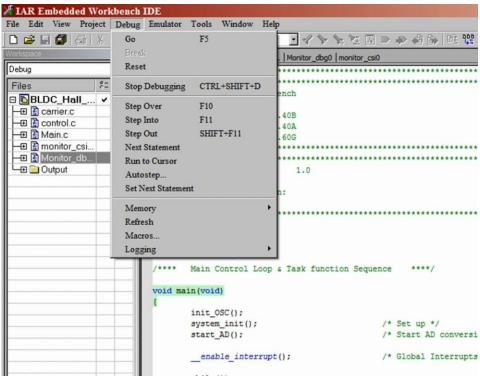


Figure 9-31 Debug Menus

ile	Edit	View	Project	Debug	Emulator	Tools	Windo	w H	telp
<b>V</b> orl	space	Wo So	ssages orkspace urce Brow eakpoints	vser	Build Find in Tool Ou Debug Find In	itput Log		×	main_mcio.c motor.c  r
Det	-		assembly			nace		27.	1*
	es 3 BL + (1) + (1)	Syi Re Uva Sta Au Liv Qu Ca Tei Co Pro	mory mbolic Mei atch cals atics to e Watch lick Watch ll Stack rminal I/O de Cover ofiling ack		ORIE3 -	De	Cm		source code sta */ void main(void) { /* Hardware Set
			olbars atus Bar	•					DEBUG_PORT_MODE DEBUG_PORT = 0;
									<pre>Motor_SetupFunc init_LEDPORT(); init_TIMER();</pre>
									/*
BL	DC_HA	LL121	)_78K0RI	E3			1.5 V		

Figure 9-32 Debug Views Windows

	Emulator Tools Window Help Hardware Setup	1.4.
D <b>╔╏┇</b>	Breakpoint Toggle During Run	
Workspace Debug	Mask Option Pseudo Emulation	motor.c   r
Files Files BLDC_HALL120_78K GUI_support.c FILE Initialise_hardware.c FILE Initialise_handlers.c	Live Watch Setup Trace Setup Timer Setup Edit Events Edit Sequencer	code sta
Hatin_mcio.c Hatility motor.c Hatility sub_mcio.c Hatility output	Trace Function Trace Live Memory	
	Breakpoint Usage	
	Enable Flash Self Programming Flash Programming Emulation PG-FPx Security Flags Setting Emulation Flash Shield Setting	<b>void</b> ) mare Set
	DEBUG_ DEBUG_ Motor_ init_L	PORT_MODE PORT = 0; SetupFunc EDPORT(); IMER();

#### Figure 9-33 Emulator Debug Options

Note The "Live Watch" does not operate in real time on the on-chip debug unit.

# **Chapter 10 Appendix**

#### Table 10-1 Port Pin Definitions

Pin	Port Pin	Definition of Use				
22	P33	Mode PB				
21	P32	Reverse PB				
23	P77	Forward PB				
24	P76	Start/Stop PB				
50	P152/ANI10	Speed POT				
17	P60 (78F0730)	RUN LED				
28	P12 (78F0730)	BREAK LED				
30	P70	HALL1				
29	P71	HALL2				
28	P72	HALL3				
46	TMOFF0	Hardware Motor OC				
49	P153/ANI11	AD_Shunt				
27	TXD0	TXD0 (USB)				
26	RXD0	RXD0 (USB)				
19	TXD1	TXD1 (ZigBee)				
20	RXD1	RXD1 (ZigBee)				
42	TO02	HI-U				
41	TO03	LO-U				
40	TO04	HI-V				
39	TO05	LO-V				
38	ТО06	HI-W				
37	TO07	LO-W				
34	P50	LED_segA				
33	P51	LED_segB				
32	P52	LED_segC				
31	P53	LED_segD				
60	P20	LED_segE				
59	P21	LED_segF				
58	P22	LED_segG				
57	P23	LED_segDP				
3	P42	LED_0				
2	P43	LED_1				
25	P141	LED_2				
53	P75	LED_3				

Schematics

Schematics descriptions are attached to this document. Use the *Attachments* tab for access (lower left side of the screen).