User's Manual

MC-CPU-78K0RIE3 CPU Daughter Card

For use with the Low Voltage Motor Control Starter Kit

Document No. U19824EE1V0UM00

Date published June 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 anddistributors. 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]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554,
U.S.A.
Tel: 408 5886000
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 MC-CPU-78K0RIE3 CPU Daughter Card for motor control.

Purpose This manual presents the hardware manual of the MC-CPU-78K0RIE3 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: $\overline{xxx}$ (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; explanation of (Note) in the text

Caution Item deserving extra attention

Remark Supplementary explanation to the text

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

Chapter 1 Introduction

The MC-CPU-78K0RIE3 CPU Daughter Card is designed to form part of the low-voltage starter kit for motor control (MC-LVKIT-78K0RIE3) and is a complete 3-phase motor control evaluation system for NEC Electronics' microcontroller application-specific standard products (ASSP's) for motor control.

The MC-CPU-78K0RIE3 kit contains all necessary hardware and software to quickly set up and run a low-voltage brushless DC motor (BLDCM).

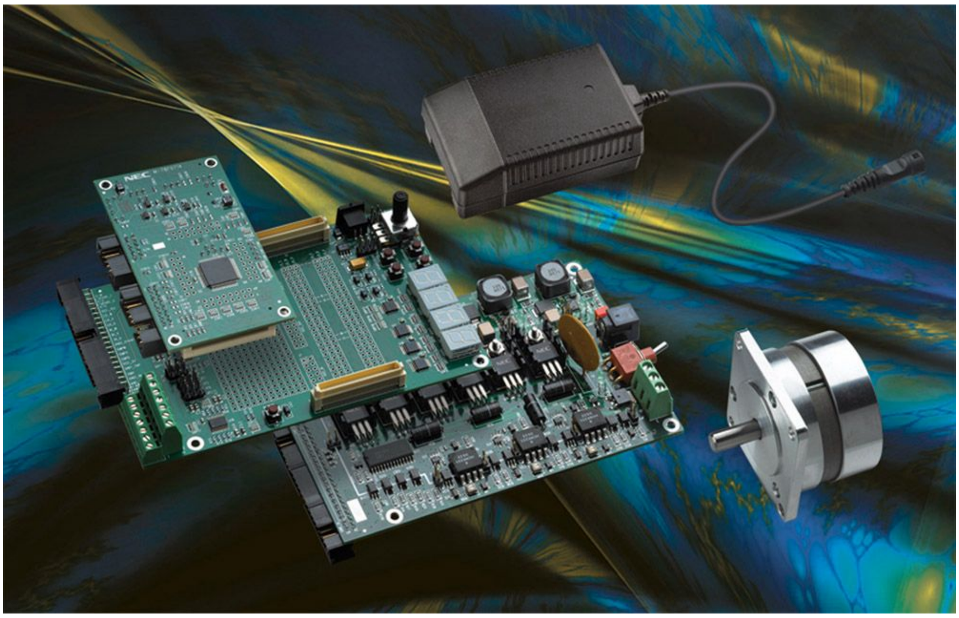


Figure 1-1 MC-LVKIT 3 phase Motor Control Evaluation System

The MC-CPU-78K0RIE3 CPU Daughter Card is used to demonstrate and evaluate CPU and on-chip peripheral functions of NEC Electronics16-bit microcontrollers, μPD78K0R-IE3 and μPD78K0R-IC3.

The MC-CPU-78K0RIE3 CPU Daughter Card will interface with the Motor Control I/O Board (MC-I/O Board), in motor control applications. In addition, the MC-CPU-78K0RIE3 supports Flash programming and debugging when the CPU Daughter Card is connected to a host-PC through the MINICUBE2 on-chip debugger/programmer.

In order to provide sufficient details for the users to interface with MC-I/O Board, some details for both MC-I/O board and the MC-CPU-78K0RIE3 CPU Daughter Card will be provided in this manual.

Please see the MC-CPU-78K0RIE3 CPU Daughter Card below:

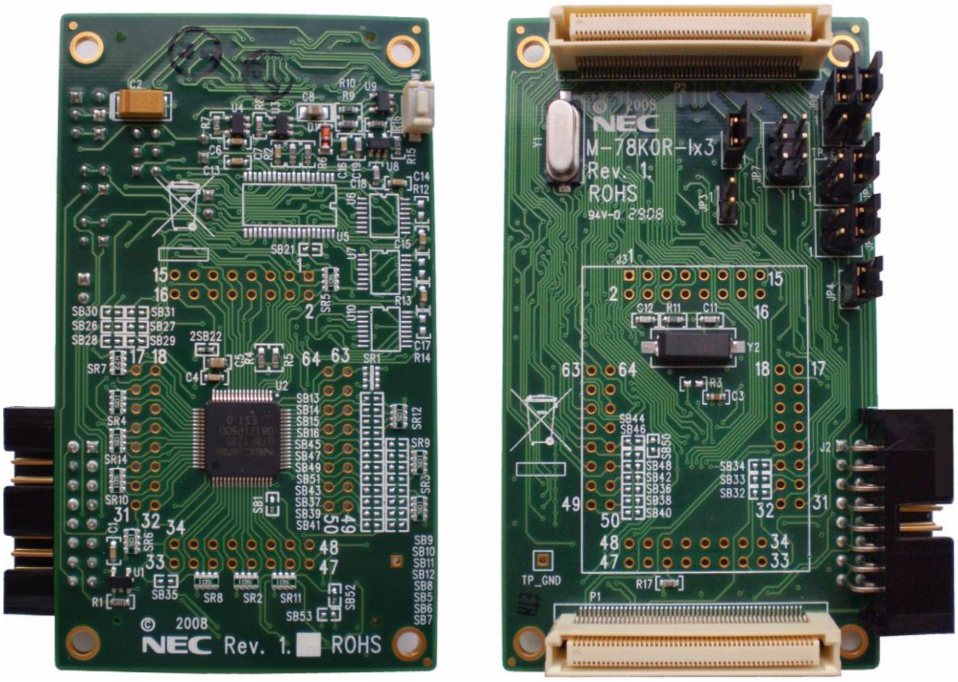


Figure 1-2 CPU Daughter Card

Notes

1. To flash program or debug the MC-CPU-78K0RIE3 CPU Daughter Card it is recommended to use the MINICUBE2 on-chip debugger/programmer unit.
2. The source code and project files are available free of charge and can be downloaded from the *NEC Electronics motor control website*.
3. 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 Electronics or through an NEC Electronics franchised distributor.
4. This User's Manual covers details which are specific to the MC-CPU-78K0RIE3 CPU Daughter Card and not the complete evaluation kit. Please see the *User's Manual for the MC-LVKIT-714* which covers the standalone operation of the evaluation kit.
5. For information on the GUI please see the relevant section.

Chapter 2 MC-CPU-78K0RIE3 CPU Daughter Card Specifications

The MC-CPU-78K0RIE3 CPU Daughter Card supports uPD78K0R-IE3 and uPD78K0R-IC3 microcontrollers. The specifications for these microcontrollers are:

µPD78K0R-Ix3 Features

Flash Self-Programming (with Boot Swap Function/Flash Shield Window Function)

Built In On-chip Functions

- On-chip Debugging Function
- On-chip Power-on Clear
- On-chip Watchdog Timer
- On-chip Multiplier/Divider
- On-chip BCD Adjustment

I/O Ports

Timer (TAU)

- 12 channel x 16-bit @ 40 MHz TAU (Timer Array Unit)
- Watchdog Timer
- Real-time Counter
- On-chip Motor Control Option Unit

On-Chip Comparator/Operational Amplifier

Serial Interface (SAU)

- UART (LIN)
- CSI
- Simplified I^2C
- I^2C (Multi-Master)

10-Bit Resolution A/D Converters @ 2.5uS conversion time

Power Supply Voltage

- V_{DD} = 2.7 V_{DC} - 5.5 V_{DC}

Chapter 3 An Overview of MC-IO Interface

3.1 General Descriptions of MC-I/O Interface with Motor Control Micro-Boards

The MC-CPU-78K0RIE3 CPU Daughter Card provides controller functions for motor control operation. It processes and determines microcontroller actions from various sensor inputs. The Motor Control Evaluation System consists of:

- Power Module (MC-PWR-LV)
- Motor Control I/O Module (MC-I/O)
- MC-CPU-78K0RIE3 CPU Daughter Card (this product)

The power module (MC-PWR-LV) drives the motor and provides sense signals through the 40-pin ribbon cable (J5). The MC-I/O board sends and receives signals to the power module via the 40-pin ribbon cable at (J4) and the 14-pin terminal block at (J5) and provides user interface and operation control functions. The removable CPU Daughter Card is connected to the MC-I/O board, as shown below.

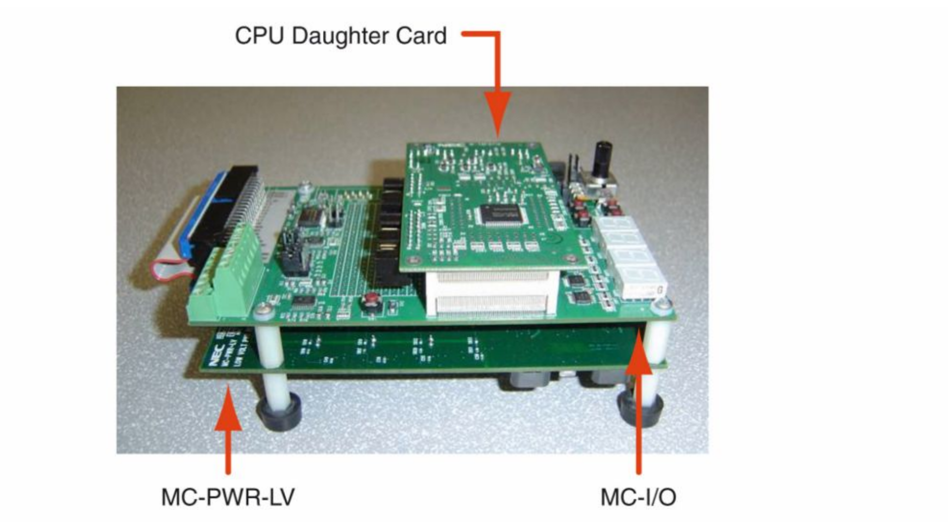


Figure 3-1 MC-CPU-78K0RIE3 installed on MC-IO Board

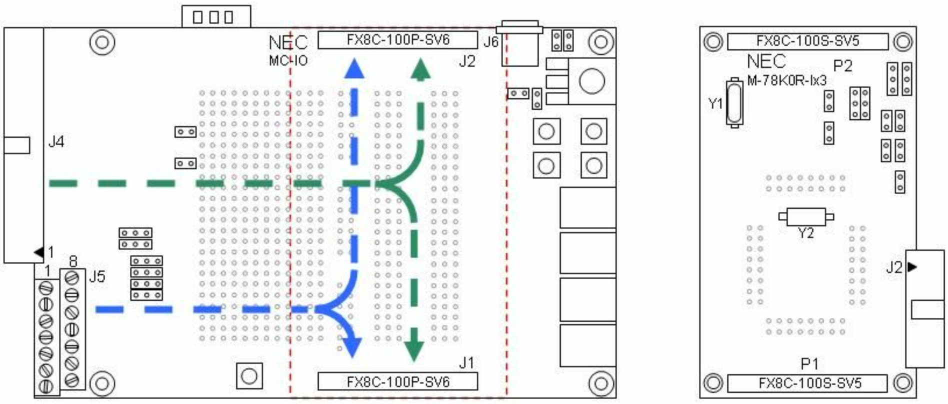


Figure 3-2 MC-IO and MC-CPU-78K0RIE3 CPU Daughter Card

The MC-CPU-78K0RIE3 CPU Daughter Card control signals use two 100-pin high density connectors. All control signals are placed on the two interface connectors at pre-selected positions. This method enables multiple Motor Control CPU Daughter Cards to interface with the MC-I/O interchangeably. Therefore, it is important to know the motor control signals used.

This section provides general overview of module interfaces so that users can become familiar with the system. The next section provides specific information how the MC-CPU-78K0RIE3 signals are interfacing with the system.

3.2 Motor Control Interface Signals

Table 3-1 Signal on MC-I/O J5 14-Pin Terminal Block

Categories	Signal Names	Description of Signals
Motor Position Encoder	ENC_A, ENC_B, ENC_Z	Motor position encoder signals connected to interrupt inputs of CPU
HALL Effect Sensor Signal	HALL_1, HALL_2, HALL_3	HALL effect sensor input connected to interrupt inputs of CPU
Motor Temperature Input	ANI3_TEMP	Motor temperature input connected to A/D converter input of CPU

The signals on 14-pin terminal block are the sensor signals from the motor unit. These signals are input to microcontroller on the CPU Daughter Card through MC-I/O module.

Table 3-2 Signals on MC-I/O J4 40-Pin Ribbon Cable

Categories	Signal Names	Description of Signals
System Power	VCC_15V	Power input to MC-I/O board
	VCC_5V	Regulated 5V Power
PWM Signals	HI_U, HI_V, HI_W - High-side FET Drive	PWM Signals from CPU
	LO_U, LO_V, LO_W - Low-side FET Drive	
Back-EMF Comparator	CMPU, CMPV, CMPW	Back-EMF comparator signals from power module connected to interrupt inputs of CPU
Current Sense Signals	ANI0_IU, ANI1_IV, ANI2_IW	Motor phase current - Low-side current detect connected to A/D
	ISHUNT	Motor shunt current - Low-side current detect connected to A/D
Safety Control Signals	PX_ITRIP	Over-current detect signal from power module connected to TMOFF0 or TMOFF1 of 78K0RIx3
	TRIP	CPU generated signal to turn off power to Power-MOSFET

Categories	Signal Names	Description of Signals
Phase Voltage Detect	V-U, V-V, V-W	Motor phase voltage detect signal connected to A/D inputs of CPU
Power-Module Temperature	ANI7_TMP	Power module temperature sense signal connected to A/D input of CPU

The above signals are coming from power module (MC-PWR-LV) through 40-pin cable interface. These signals are connected to two 100-pin connectors on the MC-I/O Board. The microcontroller inputs and outputs are connected to these signals when the CPU Daughter Card is connected to the MC-I/O board, through two 100-pin connectors.

3.3 Signal Positions on the 100-pin Connectors

Table 3-3 Signals on MC-I/O J1-Connector (P1-Connector on MC-CPU-78K0RIE3)

	J1 Pin	J1 Signal	J4 Pin	J4 Signal	Notes
1	**RS232 Host Interface Signals**				
	J1.003	RS232_RXD			Connects to RS232 transceiver interface U23
	J1.004	RS232_CTS			
	J1.005	RS232_TXD			
	J1.006	RS232-RTS			
2	**PX_ITRIP Signal from Power Module**				
	J1.011	PX_ITRIP	J4.02	PX_ITRIP	Signal to tri-state Power MOSFETs connect to TMOFF0 or TMOFF1 on 78K0RIx3
3	**Speed Adjust Potentiometer and other A/D**				
	J1.023	ANI4 (Spd Pot)			ANI4 adjust speed pot use A/D
	J1.024	ANI5_ISHUNT	J4.07	ANI5_ISHUNT	From power module shunt current signal
	J1.025	ANI6_SPARE	J4.04	ANI6_SPARE	Spare A/D input
4	**PX_TRIPB Signal to turn off power to power MOSFETs**				
	J1.031	PX_TRIPB	J4.19	TRIP	Signal from CPU to turn off power to power MOSFETs. Use GPI/O on 78K0R/Ix3.

Table 3-4 Signals on MC-I/O J2-Connector (P2-Connector on MC-CPU-78K0RIE3)

	J2 Pin	J2 Signal	J4 Pin	J4 Signal	J5 Pin	J5 Signal	
5	**LD_LED - data latch signals for the 7 segment LED displays**						
	J2.015	LD_LED0					Use GPI/O on 78K0R/Ix3
	J2.016	LD_LED1					
	J2.017	LD_LED2					
	J2.018	LD_LED3					
6	**LED_x - common data lines for 7 segment LED displays**						
	J2.019	LED_A					Use GPI/O on 78K0R/Ix3
	J2.020	LED_B					
	J2.021	LED_C					

	J2 Pin	J2 Signal	J4 Pin	J4 Signal	J5 Pin	J5 Signal	
	J2.022	LED_D					
	J2.023	LED_E					
	J2.024	LED_F					
	J2.025	LED_G					
	J2.026	LED_DP					
7	**PWM signals - Motor phase control signals IU, IV, IW**						
	J2.029	PWM_0	J4.21	HI_U			CPU signal PWM_0
	J2.030	PWM_1	J4.27	LO_U			CPU signal PWM_1
	J2.033	PWM_2	J4.23	HI_V			CPU signal PWM_2
	J2.034	PWM_3	J4.29	LO_V			CPU signal PWM_3
	J2.037	PWM_4	J4.25	HI_W			CPU signal PWM_4
	J2.038	PWM_5	J4.31	LO_W			CPU signal PWM_5
8	**SPD_MSR - select one from INTP1_PX / INTP2_PY / INTP3_PZ**						
	J2.041	SPD_MSR					Timer/counter input for speed measurement
9	**Temperature sense signal from power module**						
	J2.044	ANI7_TMP	J4.09	ANI7_TMP			
10	**INTP1_PX / INTP2_PY / INTP3_PZ to interrupt inputs**						
	J2.047	INTP1_PX	J4.13	CMP_U			INTP1_PX = CMP_U or V-U or HALL1
	J2.048	INTP2_PY	J4.15	CMP_V			INTP2_PY = CMP_V or V-V or HALL2
	J2.051	INTP3_PZ	J4.17	CMP_W			INTP3_PZ = CMP_W or V-W or HALL3
11	**Phase current sense signals from power module**						
	J2.052	ANI0_IU	J4.01	ANI0_IU			Current Sense Phase U
	J2.055	ANI1_IV	J4.03	ANI1_IV			Current Sense Phase V
	J2.056	ANI2_IW	J4.05	ANI2_IW			Current Sense Phase W
12	**ANI3_TEMP Motor temperature sense signal**						
	J2.059	ANI3_TEMP			J5.08	ANI3_TEMP	Motor temperature connect to A/D
13	**Encoder signals - PX_ENCA, PX_ENCB, PX_ENCZ**						
	J2.060	PX_ENCA			J5.11	ENC_A	
	J2.063	PX_ENCB			J5.10	ENC_B	
	J2.064	PX_ENCZ			J5.03	ENC_Z	
14	**Operation control push buttons**						
	J2.067	START					GPI/O pins on the 78K0R/Ix3
	J2.068	FORWARD					
	J2.071	REVERSE					
	J2.072	MODE					

Chapter 4 MC-CPU-78K0RIE3 Hardware Design Descriptions

4.1 An Overview of Signal Connections for the MC-CPU-78K0RIE3 Micro-Board

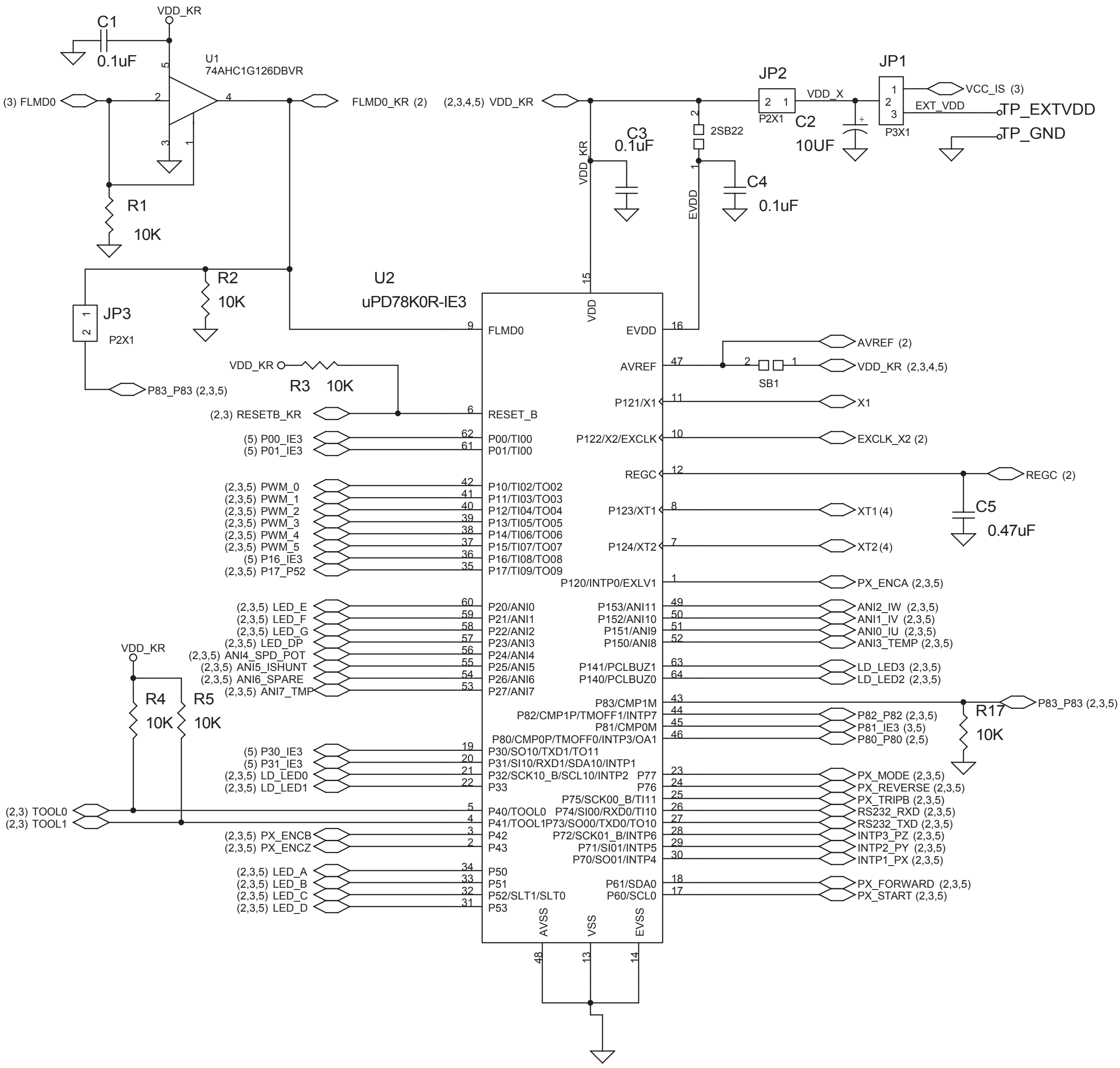


Figure 4-1 Signals on uPD78K0RIE3

Special treatment for the REGC signal. REGC is connected to GND through 0.47uF capacitor. Used EIA_A size, tantalum capacitor.

4.2 Power Source Selection

When the MC-CPU-78K0RIE3 CPU Daughter Card is connected to MC-I/O board, the micro board receives power from the MC-I/O board, **VCC_IS**

External Power Option

- Use external power connection terminal post TP_EXTVDD
- External power may be input to TP_EXTVDD TP_EXTVDD = 2.7 VDC to 5.5 VDC

Table 4-1 Power Source Jumper Settings

	Power Source Select Jumper	
Jumper No.	**Connection Descriptions**	**Description of Functions**
JP1	JP1.2 -to- JP1.1	VDD_X = VCC_IS: Select Power form MC-I/O Board
	JP1.2 -to- JP1.3	VDD_X = EXT_VDD: Select Externally Supplied Power
JP2	Normally connected	Default Setting is JP2 connected
		CPU Current Measurement: Connect Ampere Meter on JP2
SBx	Solder Jumpers	EVDD: Connected to VDD_KR through 2SB22, Normally Connected
		AVREF: Connected to VDD_KR through SB1, Normally Connected

4.3 Reset Generation

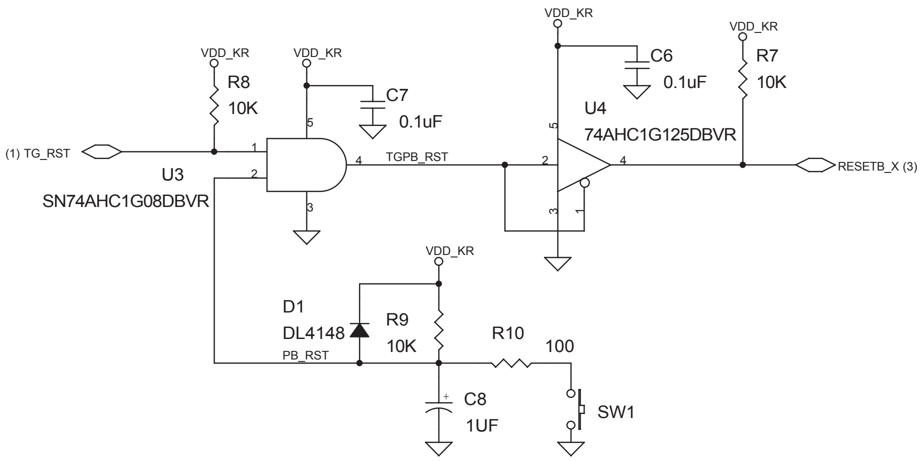


Figure 4-2 Micro-Board Reset Generation

Sources of Reset

- Push-button Switch: a Push-Button Reset Switch is provided SW1
- Reset from User's Target System TG_RST

RESETB_X

- RESETB_X is connected to J2, 16P_Flash/Debug_Header
- When MINICUBE2 Programmer/Debugger is connected to the it generates RESETB_KR
- When Mini-Cube2 is not connected, RESETB_X is connected to RESETB_KR by jumper JP6

4.4 FLMD0 - Flash Programming Mode Setting Signal

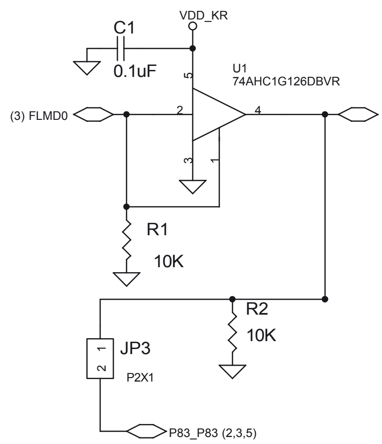


Figure 4-3 FLMD0 – Flash Programming Mode

» In normal operation	FLMD0 Input is pull-down to GND by a 10K-Resistor

» When MINICUBE2 is connected FLMD0 is driven by MINICUBE2

» For Flash Self-Programming Mode FLMD0 is Driven by P83 of IE3 (P83_P83)

» To use P83_P83 for Mode setting, JP3 should be connected

Table 4-2 FLDM0 Jumper Settings

Jumper	Condition	Description of Functions
JP3	Open	FLMD0 is driven by MINICUBE2 if connected
		FLMD0 is pulled down by 10K resistor to GND for normal operation
	Connected	During Flash self-programming, FLMD0 can be driven by P83

4.5 16P_FLASHDEBUG_HEADER

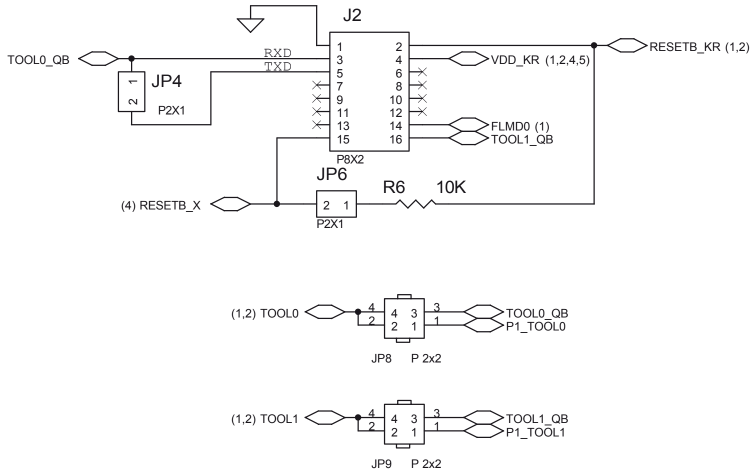


Figure 4-4 Flash/Debug Header Schematic

Table 4-3 Flash/Debug Jumper Settings

JP4	OPEN	OPEN when Mini-Cube2 has internal connection for RXD and TXD
		Other cases (Refer to User's Manual) insert JP4
JP6	OPEN	OPEN when Mini-Cube2 is Connected for Debugging or Flash Program
	Connected	Insert Jumper for Normal Operation
JP8	JP8.4 -to- JP8.3	Connect TOOL0 Output to Mini-Cube2 for Debug/Flash Program
	JP8.2 -to- JP8.1	Connect TOOL0 to P1 Connector for Users to use TOOL0 as Port-pin
JP9	JP9.4 -to- JP9.3	Connect TOOL1 Output to Mini-Cube2 for Debug/Flash Program
	JP9.2 -to- JP9.1	Connect TOOL1 to P1 Connector for Users to use TOOL1 as Port-pin

4.6 Main Clock and Sub-Clock

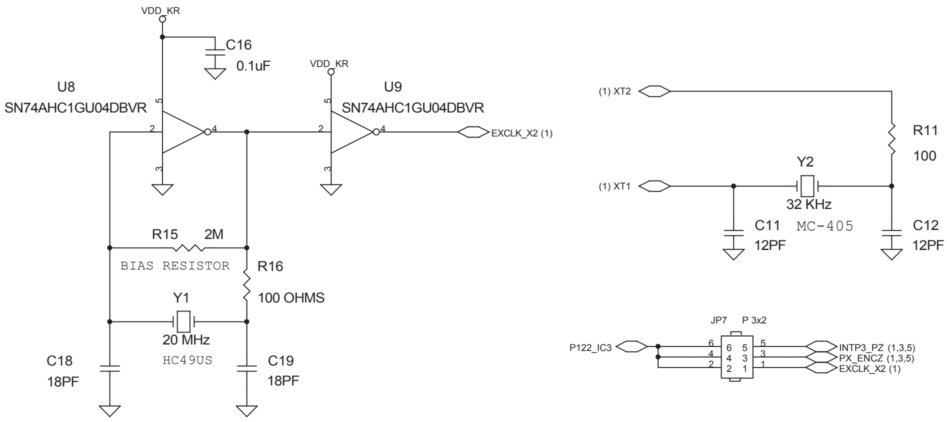


Figure 4-5 Main Clock and Sub Clock Schematic

The MC-CPU-78K0RIE3 CPU Daughter Card uses external clock (EXCLK) as main clock for the µPD78K0RIE3. A sub clock oscillator 32KHz crystal is connected to µPD78K0RIE3 only.

4.7 Target System Connector

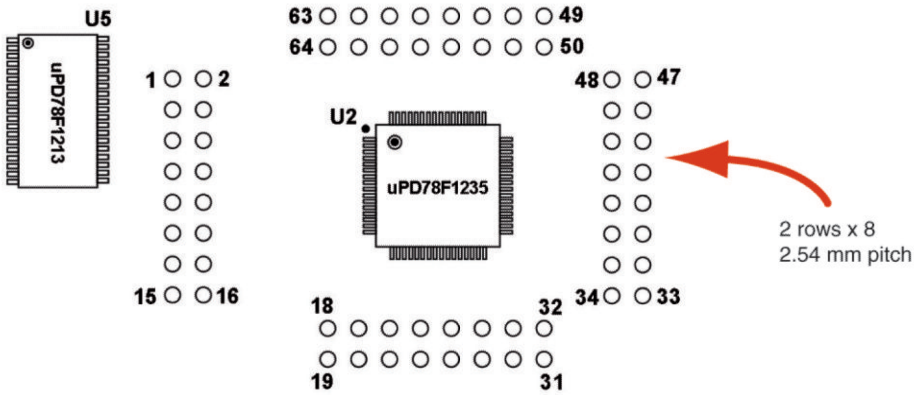


Figure 4-6 Target System Connector Diagram

Target system connector is provided all around µPD78K0R-IE3 to be connected to the users' target system. It has same pin outs with µPD78K0R-IE3.

4.8 Optional RS232-UART Connections

The MC-I/O board provides optional RS232-UART connections to a host-PC. This section describes RS232 circuit and describes the use of RxD and TxD inputs of the microcontroller.

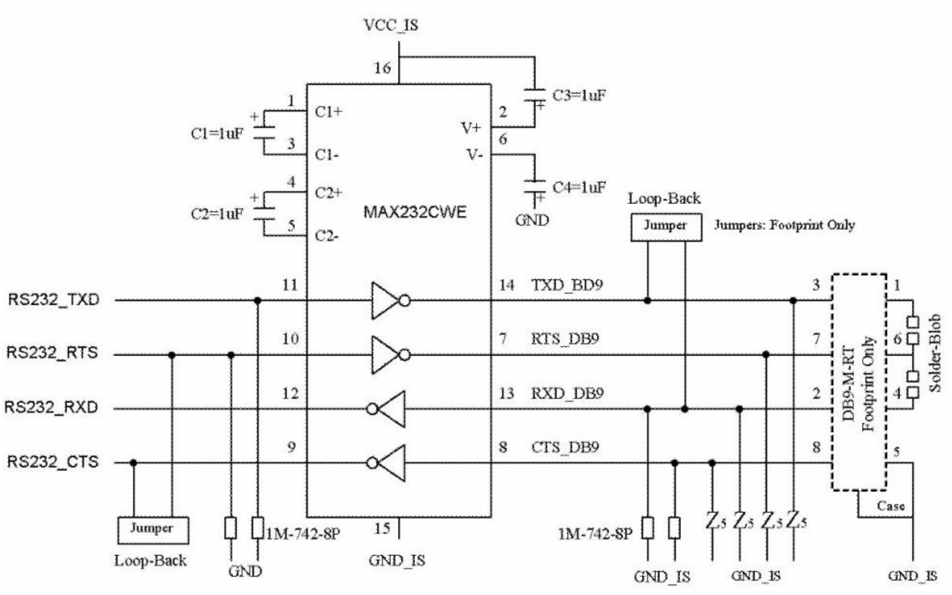


Figure 4-7 RS232 Transceiver Circuit on the MC-I/O

On the MC-CPU-78K0RIE3 CPU Daughter Card

- RS232_TXD is the outgoing signal form microcontroller to host-PC
- RS232_RXD is the incoming signal from host-PC to microcontroller

Looking from host-PC, transmit output is connected to RXD_DB9. Likewise, receiving input of the host-PC is connected to TXD_DB9 input. The MC-CPU-78K0RIE3 UART inputs and outputs are connected to RS232_RXD and RS232_TXD according to the above signal directions.

4.9 LED Data Loading

The LED data latch is implemented on the MC-I/O board. The MC-CPU-78K0RIE3 assigns port signals for LED data and control signals for latch enable.

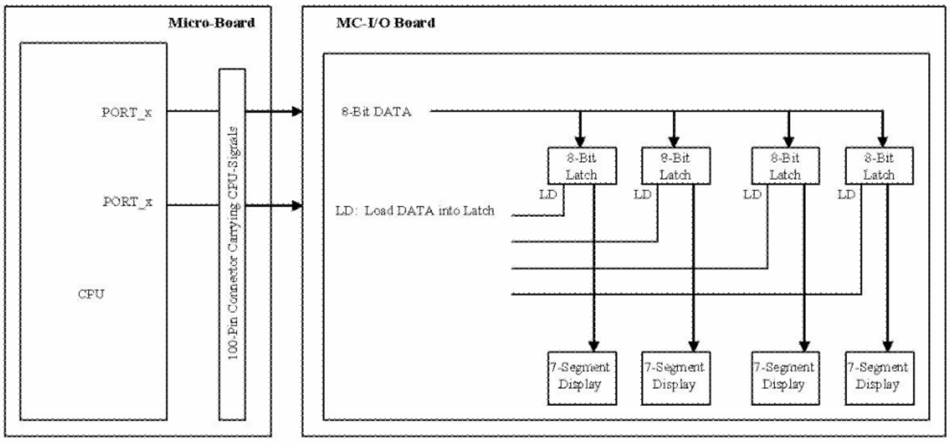


Figure 4-8 LED Data and Load Enable Connections

Table 4-4 Port Assignments for for LED-Segment Data

PORT5[3:0]	= LED Segments	LED_A through LED_D
PORT2[3:0]	= LED Segments	LED_E through LED_DP
PORT3[3:2]	= Latch Enable	LED-Digit [1:0]
PORT14[1:0]	= Latch Enable	LED-Digit [3:2]

4.10 Motor Speed Measurement

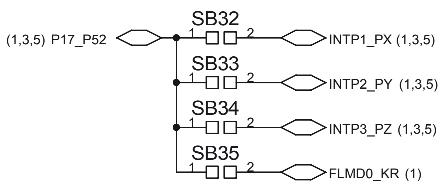


Figure 4-9 Speed Measurement Selection

P17 (MC-CPU-78K0RIE3) and **P52** (MC-CPU-78K0RIC3) are timer capture inputs for the internal timers. A selected INTP1_PX, INTP2_PY or INTP3_PZ signal is input to timer capture input to measure motor speed. The P17_P25 can also be used for setting Flash self-programming mode.

4.11 Use of Signals: Signal Assignments and Signal Multiplexing

The µPD78K0R-IC3 is a 38-pin device, whereas the µPD78K0R-IE3 is a 64-pin device. In order to provide necessary real-time operation signals and operation control signals for the MC-I/O board, it is necessary to multiplex µPD78K0R-IC3 signals. The µPD78K0R-IE3 has 64-pins and it is not necessary to multiplex the microcontroller port pins.

Table 4-5 Signal Assignments for µPD78K0R-IE3

K0R_IE3	Pin Name	Signal Name	Termination	Motor Control Functions
K0R_IE3_01	P120/INTP0/EXLVI	PX_ENCA	SRP-1M742-GND	PX_ENCA
K0R_IE3_02	P43	PX_ENCZ	SRP-1M742-GND	PX_ENCZ
K0R_IE3_03	P42	PX_ENCB	SRP-1M742-GND	PX_ENCB
K0R_IE3_04	P41/TOOL1	TOOL1	10K to VDD_KR	QB16P_Pin-16 (Tool1)
K0R_IE3_05	P40/TOOL0	TOOL0	10K to VDD_KR	QB16P_Pin-03 (Tool0)
K0R_IE3_06	RESET_B	RESETB_KR	10K to VDD_KR	QB16_Pin02 (Reset)
K0R_IE3_07	P124/XT2	XT2		
K0R_IE3_08	P123/XT1	XT1		
K0R_IE3_09	FLMD0	FLMD0_KR	10K to GND	QB16P_Pinxx (FLMD0)
K0R_IE3_10	P122/X2/EXCLK	EXCLK_X2		
K0R_IE3_11	P121/X1	X1		
K0R_IE3_12	REGC	REGC	0.47ufd to GND	
K0R_IE3_13	VSS	GND		
K0R_IE3_14	EVSS	GND		
K0R_IE3_15	VDD	VDD_KR	0.1ufd to GND	
K0R_IE3_16	EVDD	EVDD	2SB22 to VDD_KR	
K0R_IE3_17	P60/SCL0	PX_START	SRP-5K742-VDD	PX_START
K0R_IE3_18	P61/SDA0	PX_FORWARD	SRP-5K742-VDD	PX_FORWARD
K0R_IE3_19	P30/SO10/TXD1/TO11	P30_IE3	SRP-1M742-GND	
K0R_IE3_20	P31/SI10/RXD1/SDA10/ INTP1	P31_IE3	SRP-1M742-GND	
K0R_IE3_21	P32/SCK10_B/SCL10/ INTP2	LD_LED0	SRP-1M742-GND	LD_LED0
K0R_IE3_22	P33	LD_LED1	SRP-1M742-GND	LD_LED1
K0R_IE3_23	P77	PX_MODE	SRP-1M742-GND	PX_MODE
K0R_IE3_24	P76	PX_REVERSE	SRP-1M742-GND	PX_REVERSE
K0R_IE3_25	P75/SCK00_B/TI11	PX_TRIPB	SRP-1M742-GND	PX_TRIPB
K0R_IE3_26	P74/SI00/RXD0/TI10	RS232_RXD	SRP-1M742-GND	RS232_RXD
K0R_IE3_27	P73/SO00/TXD0/TO10	RS232_TXD	SRP-1M742-GND	RS232_TXD
K0R_IE3_28	P72/SCK01_B/INTP6	INTP3_PZ	SRP-1M742-GND	INTP3_PZ
K0R_IE3_29	P71/SI01/INTP5	INTP2_PY	SRP-1M742-GND	INTP2_PY
K0R_IE3_30	P70/SO01/INTP4	INTP1_PX	SRP-1M742-GND	INTP1_PX
K0R_IE3_31	P53	LED_D	SRP-1M742-GND	LED_D
K0R_IE3_32	P52/SLTI/SLTO	LED_C	SRP-1M742-GND	LED_C
K0R_IE3_33	P51	LED_B	SRP-1M742-GND	LED_B
K0R_IE3_34	P50	LED_A	SRP-1M742-GND	LED_A
K0R_IE3_35	P17/TI09/TO09	P17_P52	SRP-1M742-GND	SPD_MSR = INTP1_PX/PY/PZ
K0R_IE3_36	P16/TI08/TO08	P16_IE3	SRP-1M742-GND	

K0R_IE3	Pin Name	Signal Name	Termination	Motor Control Functions
K0R_IE3_37	P15/TI07/TO07	PWM_5	SRP-1M742-GND	PWM_5 = LO_W
K0R_IE3_38	P14/TI06/TO06	PWM_4	SRP-1M742-GND	PWM_4 = HI_W
K0R_IE3_39	P13/TI05/TO05	PWM_3	SRP-1M742-GND	PWM_3 = LO_V
K0R_IE3_40	P12/TI04/TO04	PWM_2	SRP-1M742-GND	PWM_2 = HI_V
K0R_IE3_41	P11/TI03/TO03	PWM_1	SRP-1M742-GND	PWM_1 = LO_U
K0R_IE3_42	P10/TI02/TO02	PWM_0	SRP-1M742-GND	PWM_0 = HI_U
K0R_IE3_43	P83/CMP1M	P83_P83	SRP-1M742-GND	FLMD0 (Self Program Flash)
K0R_IE3_44	P82/CMP1P/TMOFF1/ INTP7	P82_P82	SRP-1M742-GND	SBx to PX_ITRIP
K0R_IE3_45	P81/CMP0M	P81_IE3	SRP-1M742-GND	
K0R_IE3_46	P80/CMP0P/TMOFF0/ INTP3/OA1	P80_P80	SRP-1M742-GND	PX_ITRIP
K0R_IE3_47	AVREF	AVREF	SB1 to VDD_KR	
K0R_IE3_48	AVSS	GND		
K0R_IE3_49	P153/ANI11	ANI2_IW	SB7 to GND	ANI2_IW
K0R_IE3_50	P152/ANI10	ANI1_IV	SB6 to GND	ANI1_IV
K0R_IE3_51	P151/ANI9	ANI0_IU	SB5 to GND	ANI0_IU
K0R_IE3_52	P150/ANI8	ANI3_TEMP	SB8 to GND	ANI3_TEMP (motor temp)
K0R_IE3_53	P27/ANI7	ANI7_TMP	SB12 to GND	ANI7_TMP (power module temp)
K0R_IE3_54	P26/ANI6	ANI6_SPARE	SB11 to GND	ANI6_SPARE
K0R_IE3_55	P25/ANI5	ANI5_ISHUNT	SB10 to GND	ANI5_ISHUNT
K0R_IE3_56	P24/ANI4	ANI4_SPD_POT	SB9 to GND	ANI4_SPD_POT (speed adjust)
K0R_IE3_57	P23/ANI3	LED_DP	SB16 to GND	LED_DP
K0R_IE3_58	P22/ANI2	LED_G	SB15 to GND	LED_G
K0R_IE3_59	P21/ANI1	LED_F	SB14 to GND	LED_F
K0R_IE3_60	P20/ANI0	LED_E	SB13 to GND	LED_E
K0R_IE3_61	P01/TO00	P01_IE3	SRP-1M742-GND	
K0R_IE3_62	P00/TI00	P00_IE3	SRP-1M742-GND	
K0R_IE3_63	P141/PCLBUZ1	LD_LED3	SRP-1M742-GND	LD_LED3
K0R_IE3_64	P140/PCLBUZ0	LD_LED2	SRP-1M742-GND	LD_LED2

4.12 Optional Selection for P80_P80/CMP0P/OA1 and P82_P82/Comparator Input

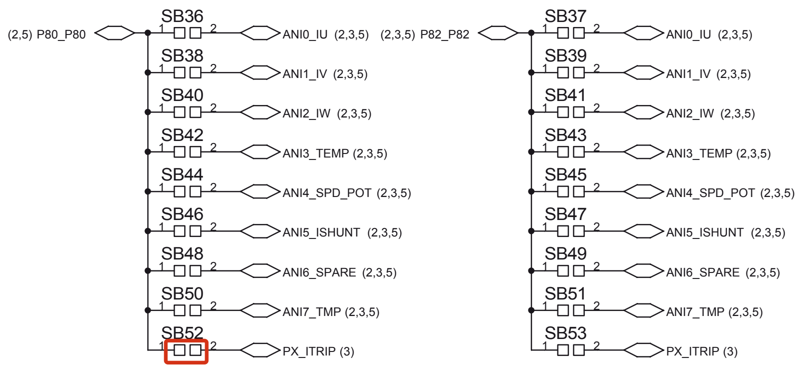


Figure 4-10 Comparator/Op-Amp Selection Input

The P80_IE3 and P80_IC3 may be used as comparator functions or op-amp functions. To demonstrate these convenient features any one of the listed analog signals may be selected by connecting the solder jumper for the selected signal.

Note SB52 is fitted as default setting for over current detection and HI-Z control.

Chapter 5 Flash Programming with a Flash Programmer

This section describes the steps necessary to modify the program in the MC-CPU-78K0RIE3 microcontroller using a Flash Programmer (not included in the MC-CPU-78K0RIE3 CPU Daughter Card).

Two programmers are available:

- PG-FP5 Full programmer
- MINICUBE2 On-chip debugger/programmer

Figure 5-1 PG-FP5 Programmer

Figure 5-2 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.

5.1 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 Workbench 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).

5.2 Save and uncompress the IAR Workbench project

Uncompress the folder containing the MC-CPU-78K0RIE3 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.

5.3 Switch settings and connecting the MINICUBE2

To prepare the MC-CPU-78K0RIE3 for flash programming follow the steps below:

- Power **OFF** the MC-IO interface module and plug in the MC-CPU-78K0RIE3 CPU Daughter Card.
- 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 J2 and attach the MINICUBE2 programmer using the 16-pin cable
- Power **ON** the MC-IO interface module

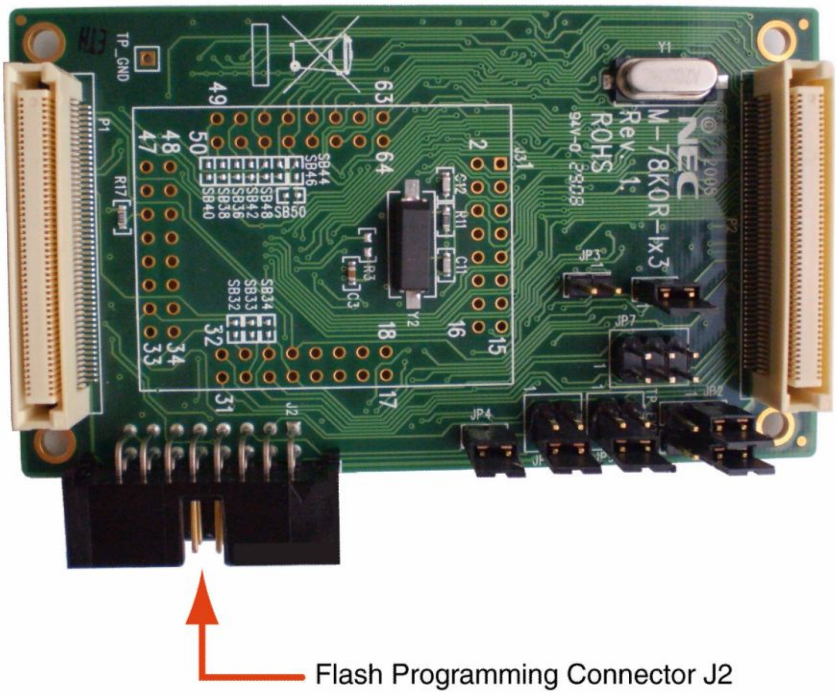


Figure 5-3 Connecting the MINICUBE2 to the CPU Daughter Card

Copy the µPD78F1235 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".

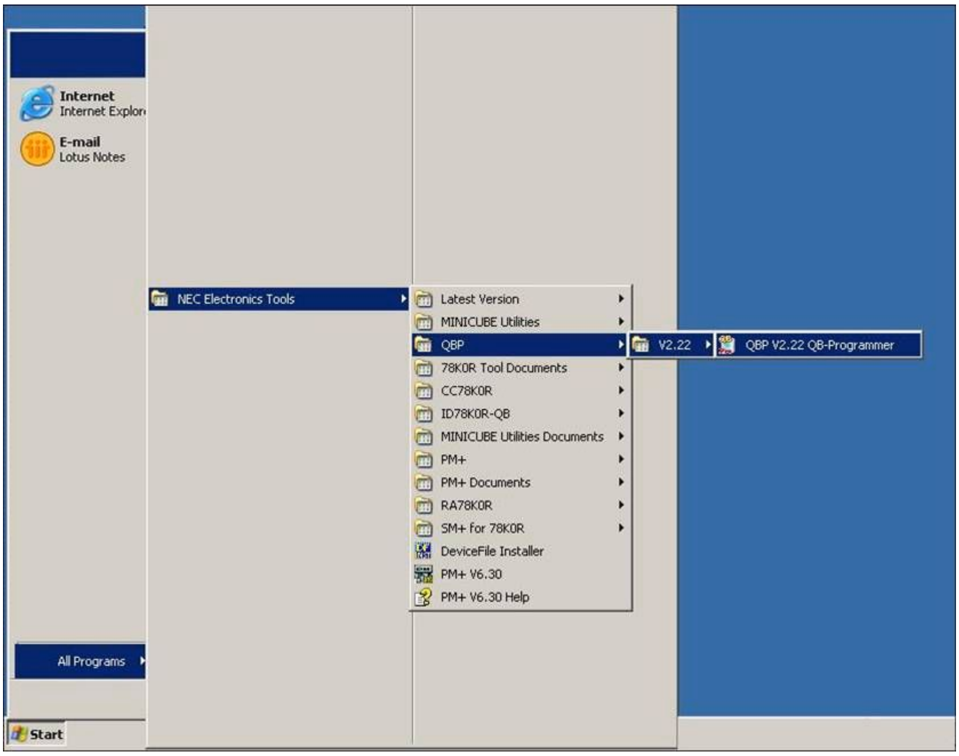


Figure 5-4 Starting the QBP V2.22 QB Programmer Graphical Interface

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

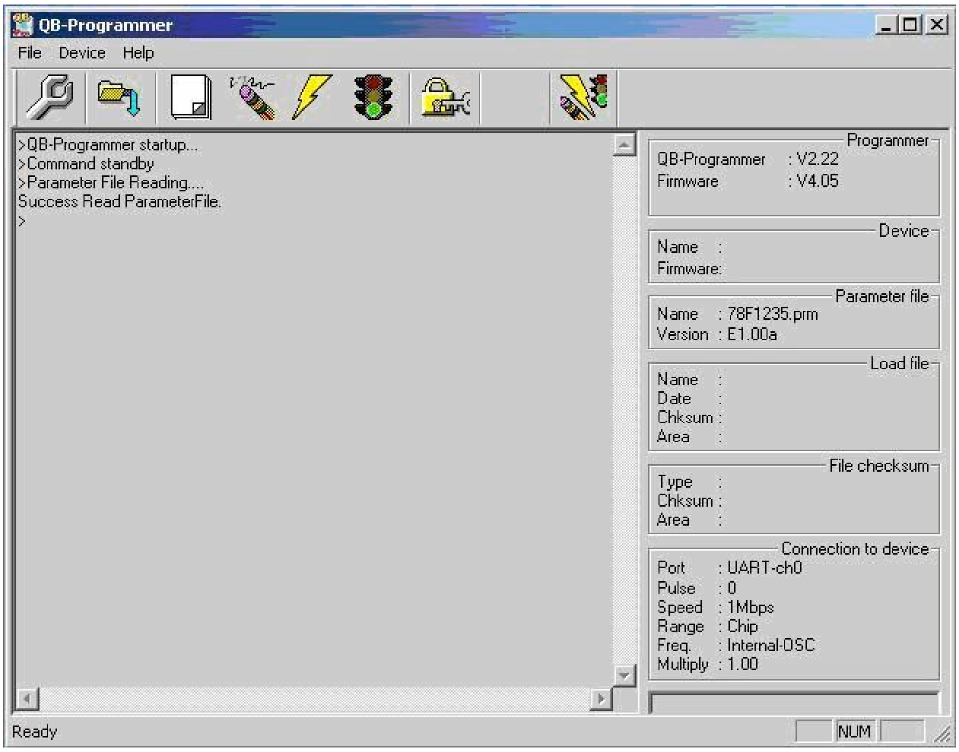


Figure 5-5 MiniCube2 Programmer GUI (QB Programmer)

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:

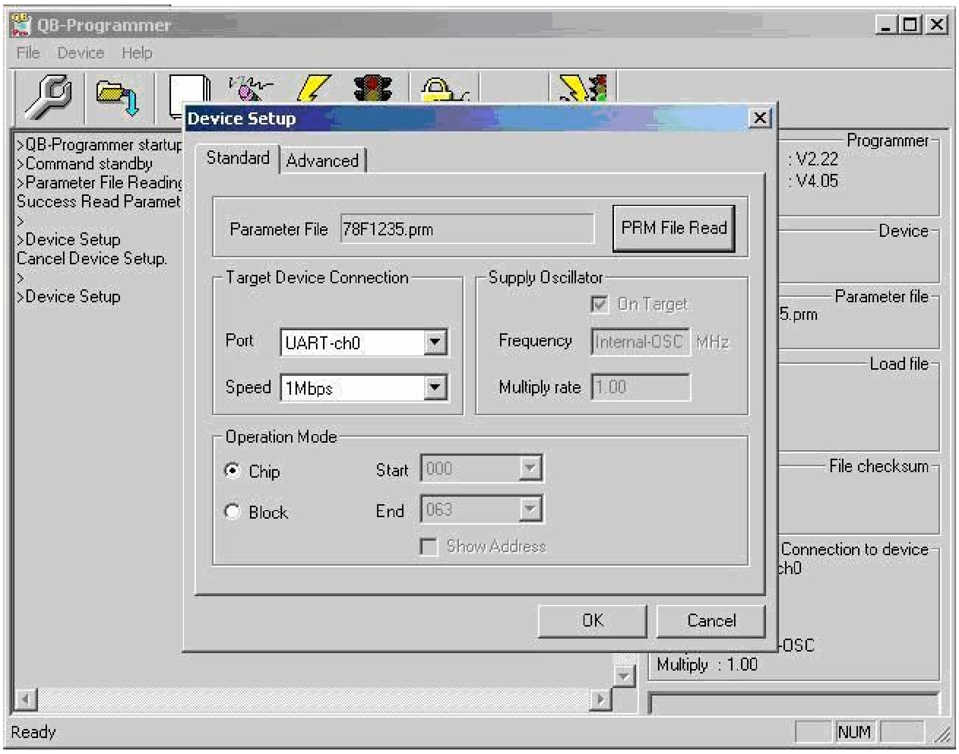


Figure 5-6 QB Programmer Device Setup

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.

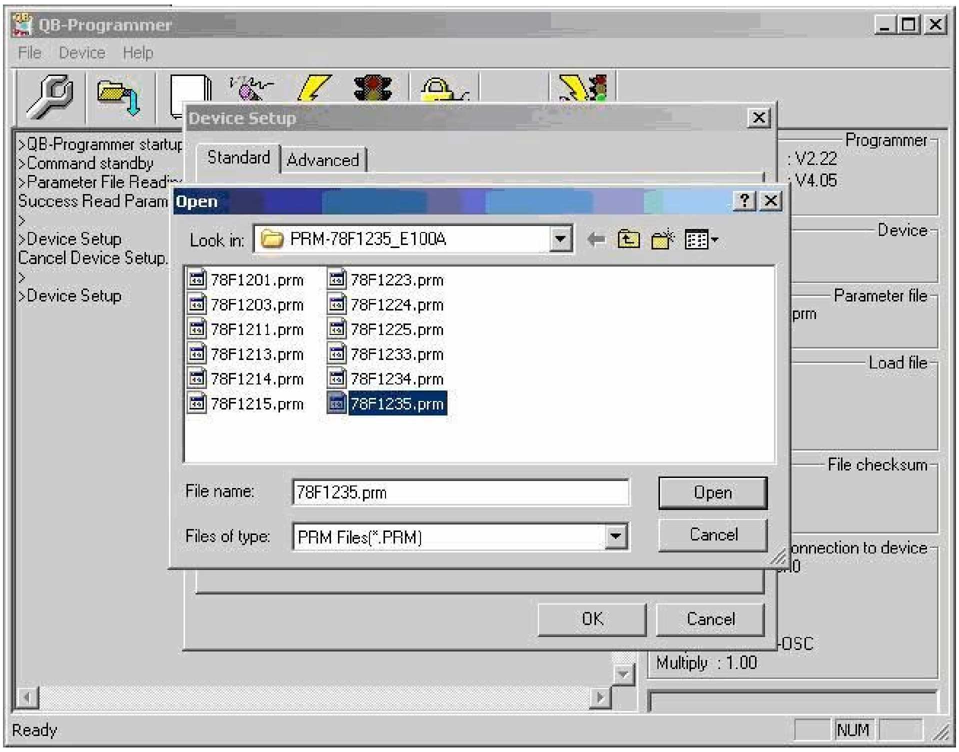


Figure 5-7 Parameter File Read

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 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:

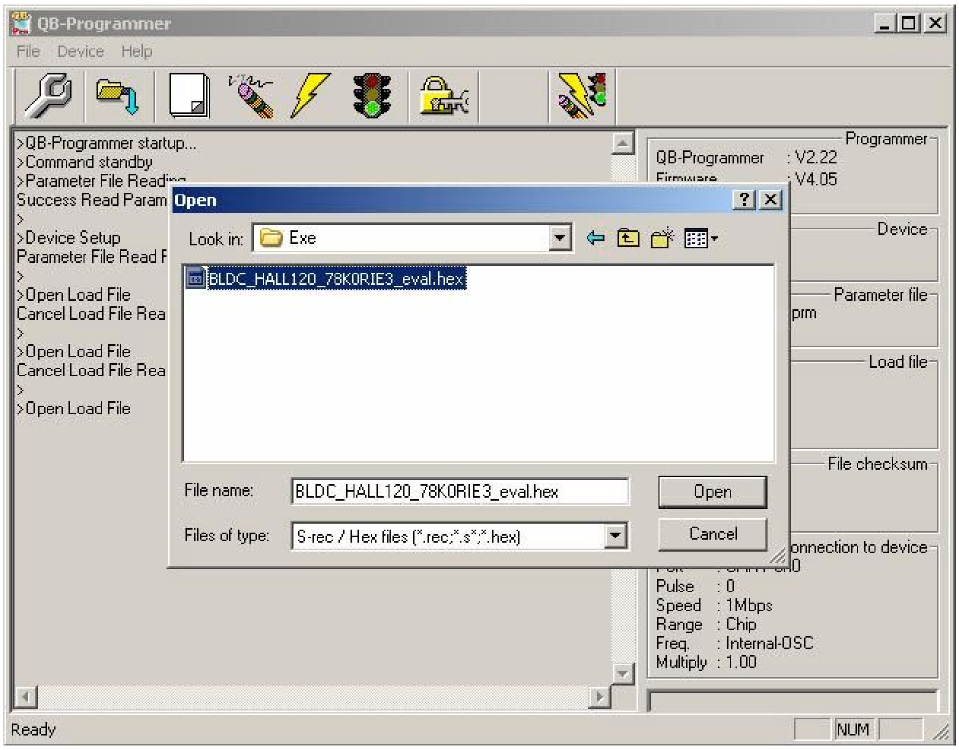


Figure 5-8 Hex File Selection

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:

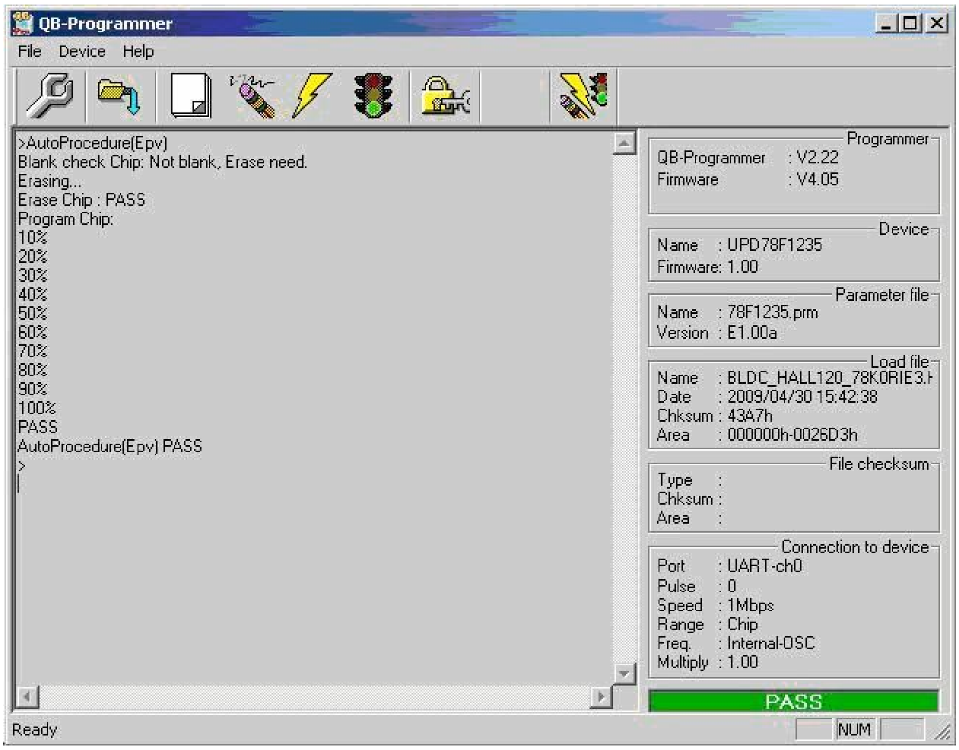


Figure 5-9 QB Programmer AutoProcedure

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.

5.4 To program the MC-CPU-78K0RIE3 while not connected to the MC-IO module

To prepare the MC-CPU-78K0RIE3 for flash programming follow the steps below:

- Unplug the MC-CPU-78K0RIE3 CPU Daughter Card from the MC-IO module
- 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 "5" (supplies power to the CPU daughter card)
- Locate the MINICUBE2 16-pin connector J2 and attach the MINICUBE2 programmer using the 16-pin cable
- The MC-CPU-78K0RIE3 is now ready to flash program

Chapter 6 Operation of the MC-CPU-78K0RIE3

When the MC-CPU-78K0RIE3 CPU Daughter Card is attached to the motor control evaluation platform it will be referenced as MC-LVKIT-78K0RIE3. The user should note the following references and documents for correct operation.

Note There is only one IAR source code project for the HALL Sensored BLDC with 120 degree trapezoidal control for standalone/GUI operation application and it is designed to run the MC-CPU-78K0RIE3 CPU Daughter Card or the MC-CPU-78K0RIC3 CPU Daughter Card or the 78K0RIX3-SPINIT kit hardware. The user is only required to change a single definition in one of the project header files in order to select between the three.

In the IAR project folder BLDC_HALL120_78K0RIE3 find the file "common.h", at the beginning of the file there are 2 definitions, simply comment out the incorrect target hardware.

```
// define which target hardware to use
//#define _78K0RIE3SPINIT
#define _MCCPU78K0RIE3
//#define _MCCPU78K0RIC3
```

The code example above would compile for target hardware MC-CPU-78K0RIE3.

6.1 MC-CPU-78K0RIE3 in Standalone Operation

For standalone use (i.e. without the GUI), please check that the following jumpers and links are as defined below.

6.1.1 CPU Daughter Card Jumper Settings

Please ensure that jumper settings are as detailed below:

JP1	Pins 1 - 2 shorted
JP2	Pins 1 - 2 shorted
JP4	Pins 1 - 2 shorted
JP6	Pins 1 - 2 shorted
JP8	Pins 3 - 4 shorted
JP9	Pins 3 - 4 shorted
2SB22	shorted
2SB1	shorted

6.1.2 Operation

Example software to run the motor is pre-programmed into the microcontroller’s flash memory. After the motor is connected, the program is ready to run the motor as soon as the 15 VDC power supply is plugged into J6 of the MC-IO board and power switch SW1 on the MC-PWR-LV power module is turned ON.

When the kit is powered up or reset, the LED displays "SELF", indicating that the kit is in standalone mode and you can use the pushbuttons and potentiometer on the MC-IO board to control the motor.

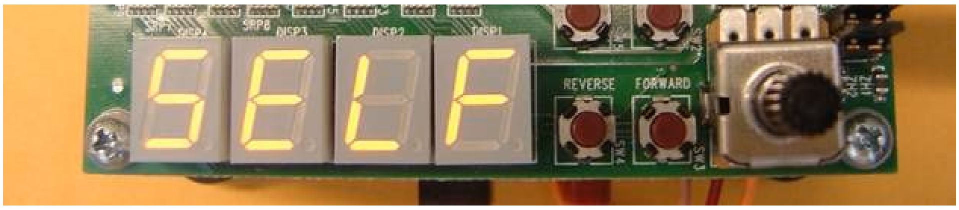
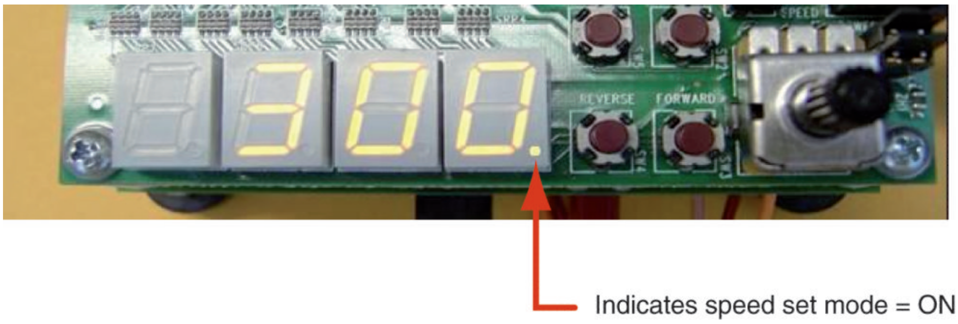


Figure 6-1 **Standalone Operation**

Three seconds after power up, the LED displays the current (start up) set speed. The speed setting mode is indicated by the decimal point on the last display digit.

On = speed set mode.

Off = displays actual speed.



Indicates speed set mode = ON

Figure 6-2 **Initial Speed Display**

Pressing the mode key toggles between speed set mode and actual speed display mode.

Note Pittman motor starting speed is 300 RPM.

After power up in standalone mode, the motor can be operated as follows:

- Press the START/STOP button to run the motor.
- Turn the potentiometer clockwise to increase the speed of the motor or counter clockwise to decrease the speed of the motor.
- The LED will display the actual motor speed calculated from the Hall sensor interrupts. The display can be toggled between actual and demanded speed by use of the MODE button.
- Press the FORWARD or REVERSE button to change the rotation direction.
- Press START/STOP to stop the motor.

Caution If the potentiometer position is set to higher speeds, and the motor rotation is suddenly reversed, an error condition may occur due to a timeout or over-current detection and the motor will stop operating. To reboot, press the RESET switch on the CPU Daughter Card, adjust the potentiometer to a lower speed and restart.

To control the motor from the Graphical Interface (GUI) from your PC, please refer to the chapter on GUI operation.

6.1.3 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 for MC-PWR-LV low-voltage power module. The sample code software also has built-in fault detection algorithms as an extra measure of protection. Consult the software manual for details.

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

Motor overcurrent:	**"O.C."**
Motor stall fault:	**"FAIL"**
Hall sensor fault:	**"HALL"**
Software overcurrent:	**"S OC"**

In the graphical interface mode (PC), the LED displays "P.C." all the time and the GUI will display all fault conditions. Please refer to the chapter on GUI operation.

6.2 MC-CPU-78K0RIE3 in 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 MC-CPU-78K0RIE3 CPU Daughter Card, 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 MC-CPU-78K0RIE3 CPU Daughter Card please refer to the chapter on flash programming.

To operate the PC GUI with the MC-CPU-78K0RIE3 CPU Daughter Card you will need a PC RS232 serial comm port and a RS232 cable configured as a "DB9 Female – Female" crossover, which is not included in any of the motor control starter kits and must be supplied by the user.

Connect the PC serial port to the starter kit J9 DB-9 RS232 connector with theRS232 serial cable.

The pin connection of the RS232 cable needs to be as follows (crossover):

Table 6-1 RS232 Cable Connections

Signal Name	PC Connection	Starter Kit connection
DCD (not used)	Pin 1	Pin 1
Rx Data	Pin 2	Pin 3
Tx Data	Pin 3	Pin 2
DTE Ready	Pin 4	Pin 4
Signal Ground	Pin 5	Pin 5
DCE Ready	Pin 6	Pin 6
RTS	Pin 7	Pin 8
CTS	Pin 8	Pin 7
Ring Indicator (not used)	Pin 9	Pin 9

The MC-CPU-78K0RIE3 CPU Daughter Card requires the following comm. port settings:

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

6.2.1 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

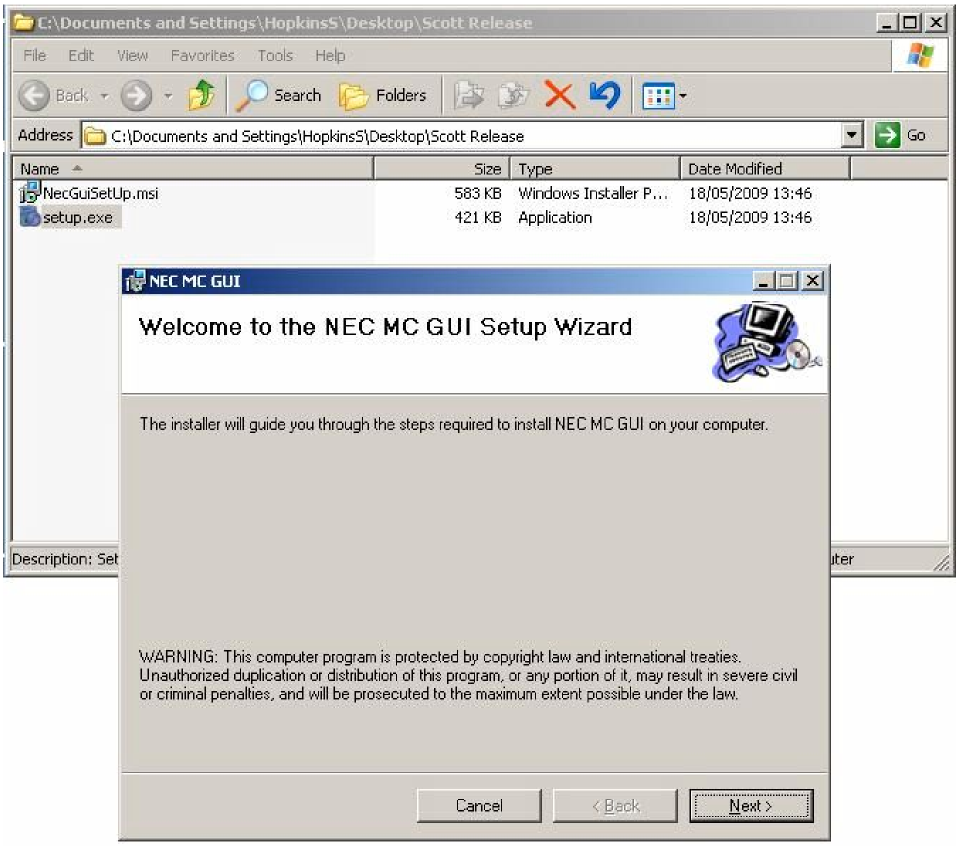


Figure 6-3 Launch the NEC MC GUI Wizard

Follow the instructions until the installation is complete.

6.2.2 To Use GUI Software

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

Figure 6-4 To Start NEC GUI

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

Figure 6-5 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.

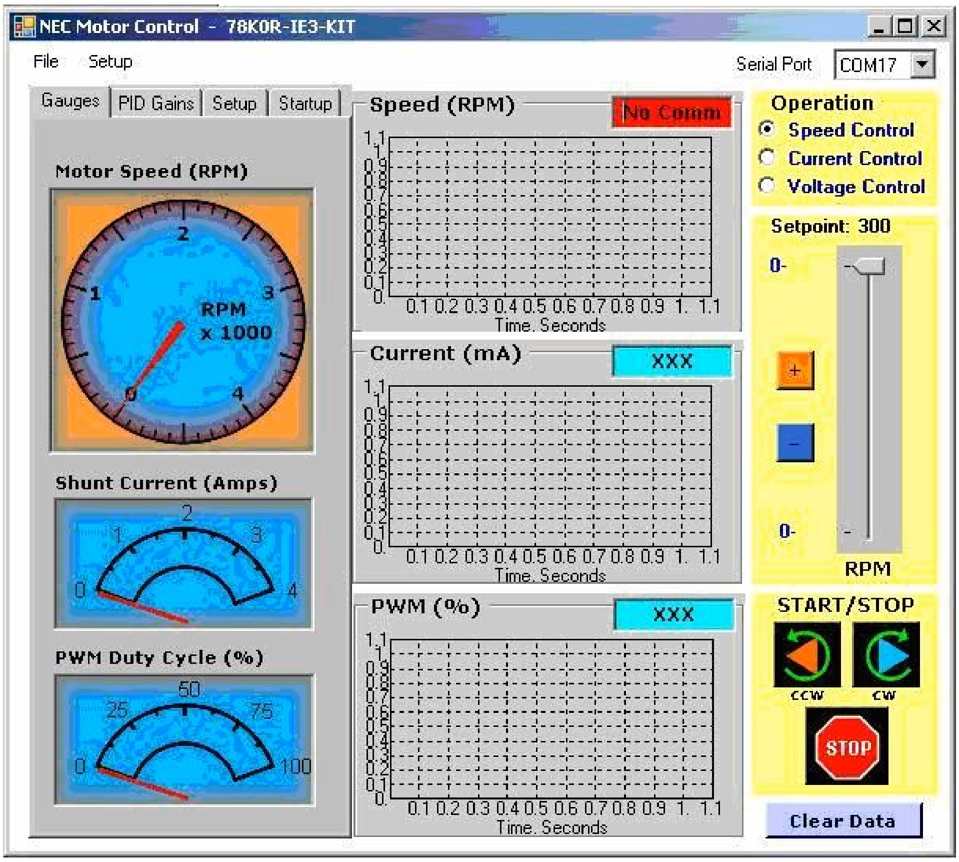


Figure 6-6 NEC GUI Main Display View

6.2.3 Set the Serial Port Number

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

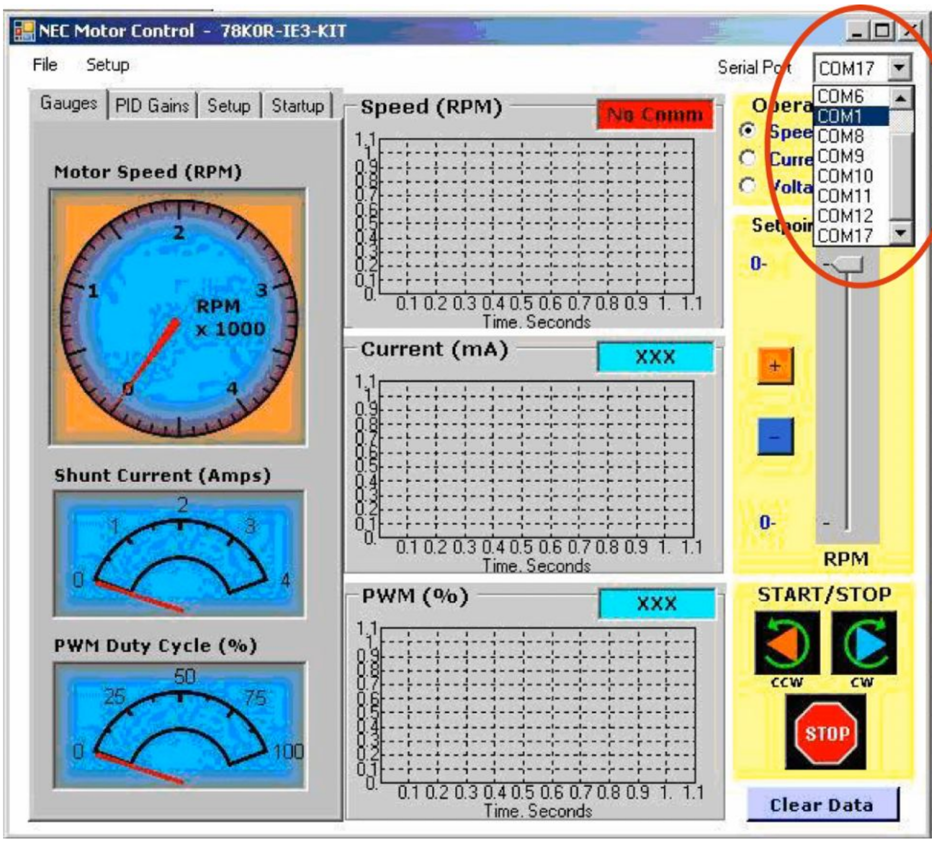


Figure 6-7 Setting the Comm Port Number

The speed (RPM) text box will display "**No Comm**" when communication is interrupted.

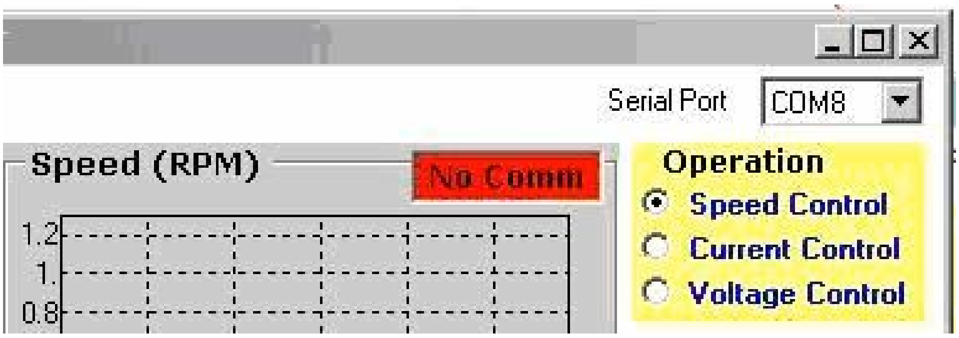
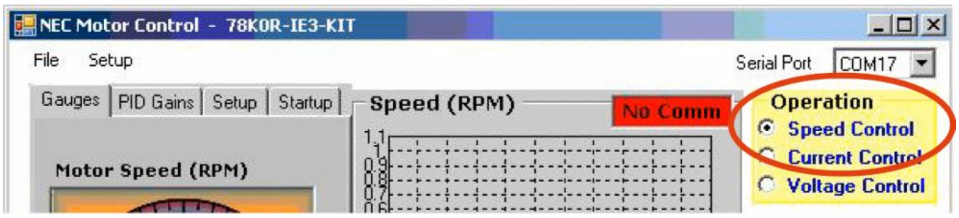


Figure 6-8 No Comm Displayed

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

6.2.4 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)



6.2.5 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.

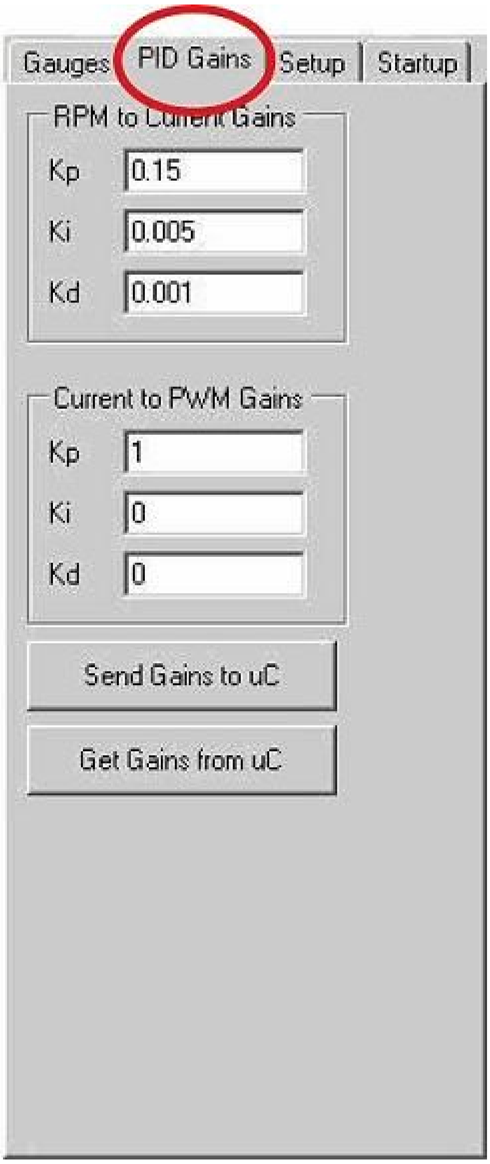


Figure 6-10 GUI PID Gains Settings

To read back the current PID settings from the MC-CPU-78K0RIE3 CPU Daughter Card 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 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 (kickstart 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.

6.2.6 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:

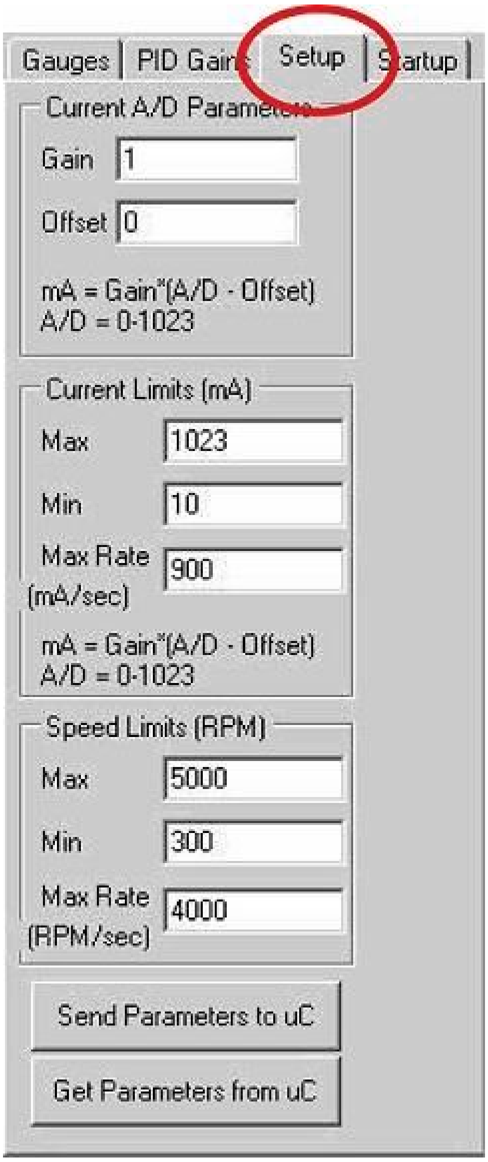


Figure 6-11 GUI Setup Settings

To read back the current setup settings from the MC-CPU-78K0RIE3 CPU Daughter Card 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 MC-CPU-78K0RIE3 CPU Daughter Card. 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

6.2.7 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:

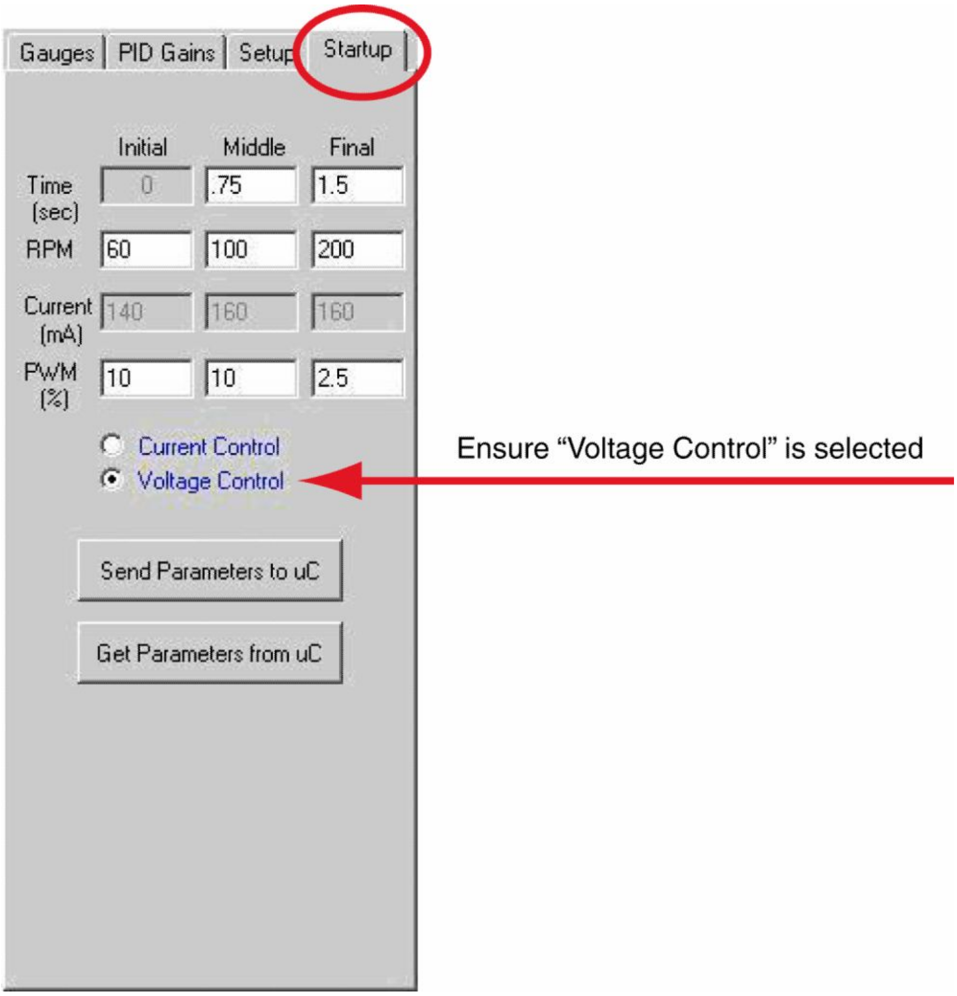


Figure 6-12 GUI Startup Settings

To read back the current startup settings from the MC-CPU-78K0RIE3 CPU Daughter Card 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 MC-CPU-78K0RIE3 CPU Daughter Card. 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 MC-CPU-78K0RIE3 with IAR Embedded Workbench

The example software for use with the MC-CPU-78K0RIE3 CPU Daughter Card is for a HALL sensored BLDC motor with 120 degree trapezoidal control for standalone/GUI operation. The MC-CPU-78K0RIE3 CPU Daughter Card is supplied 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.

7.1 Software Installation

1. If a version of the IAR tool is not already installed, then install the IAR Eembedded 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 kickstart or IAR full versions. However it may be necessary to define your own project and workspace. This is described later in this chapter.

7.2 Switch Settings and Connecting the MINICUBE2

To prepare the MC-CPU-78K0RIE3 CPU Daughter Card for debugging with the IAR Workbench follow the steps below:

- Power **OFF** the MC-IO interface module and plug in the MC-CPU-78K0RIE3 CPU Daughter Card.
- 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 J2, and attach the MINICUBE2 programmer using the 16-pin cable.
- Power **ON** the MC-IO interface module.

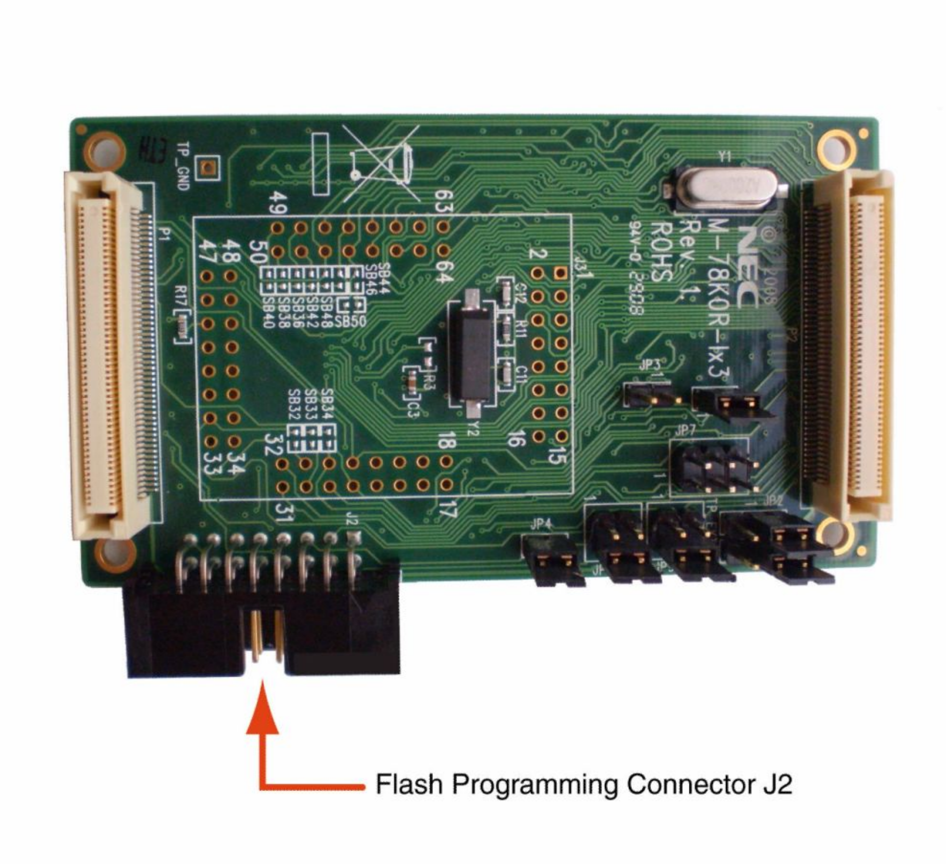


Figure 7-1 Connecting the MINICUBE2 to the CPU Daughter Card

7.3 IAR Embedded Workbench Startup

Open the IAR Workbench. The following screen should be opened:

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

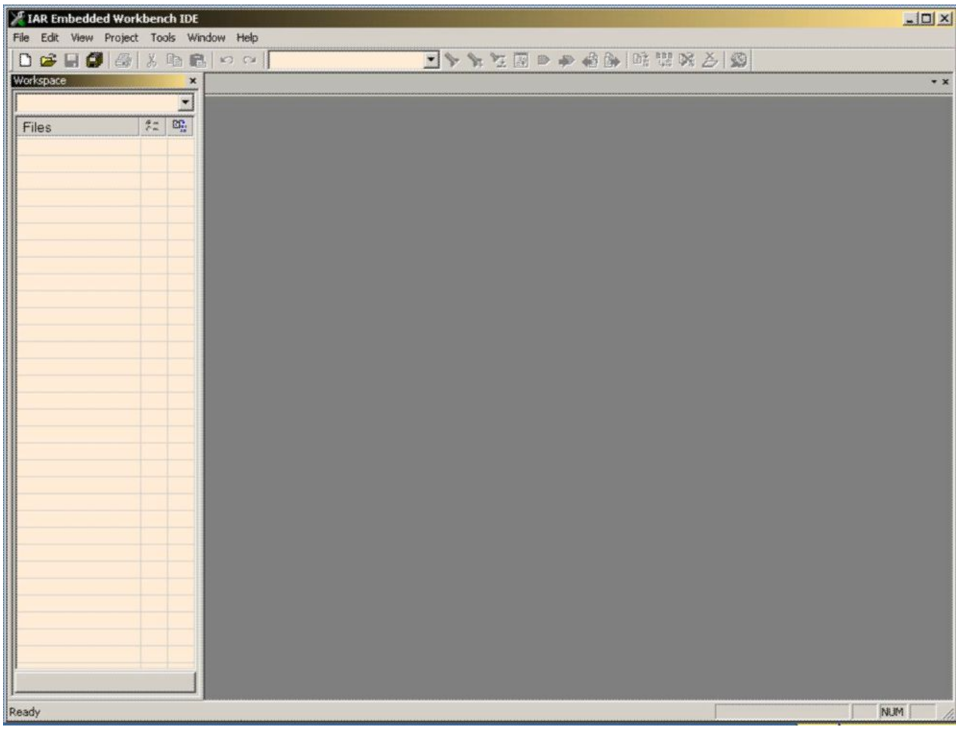


Figure 7-2 IAR Workbench Opening Screen

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

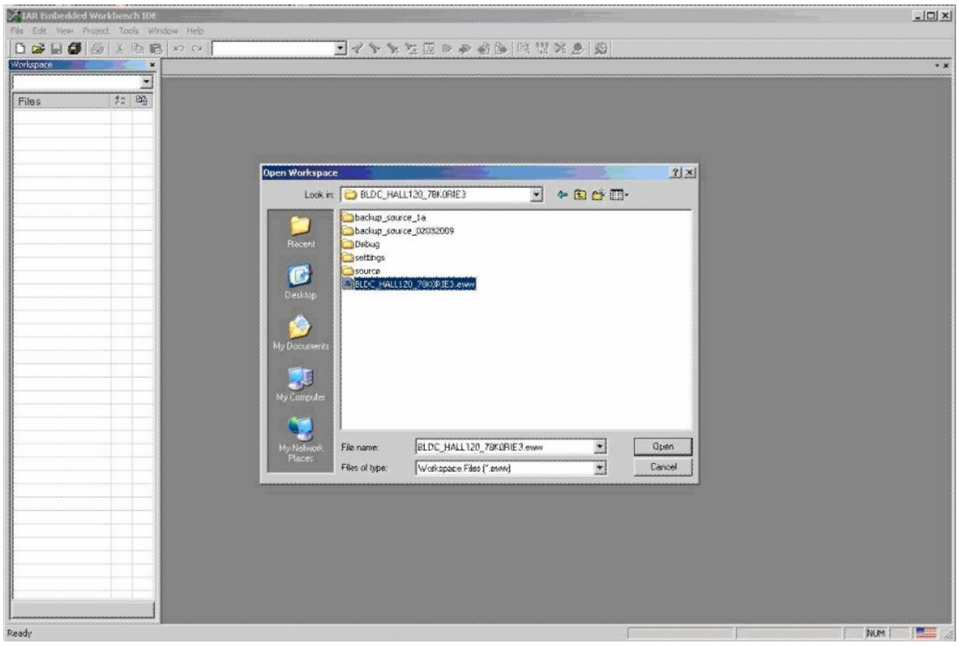


Figure 7-3 IAR Workbench Project Selection

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.

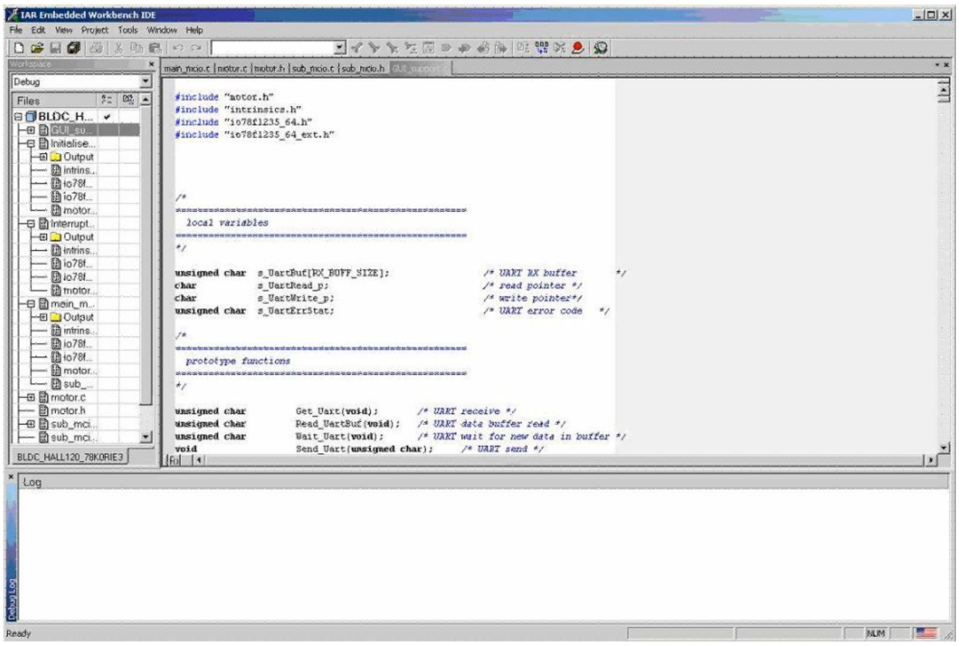


Figure 7-4 Workspace & Project Open Screen

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 7-5* below, and then ensure all the options are set according to the remaining figures below.

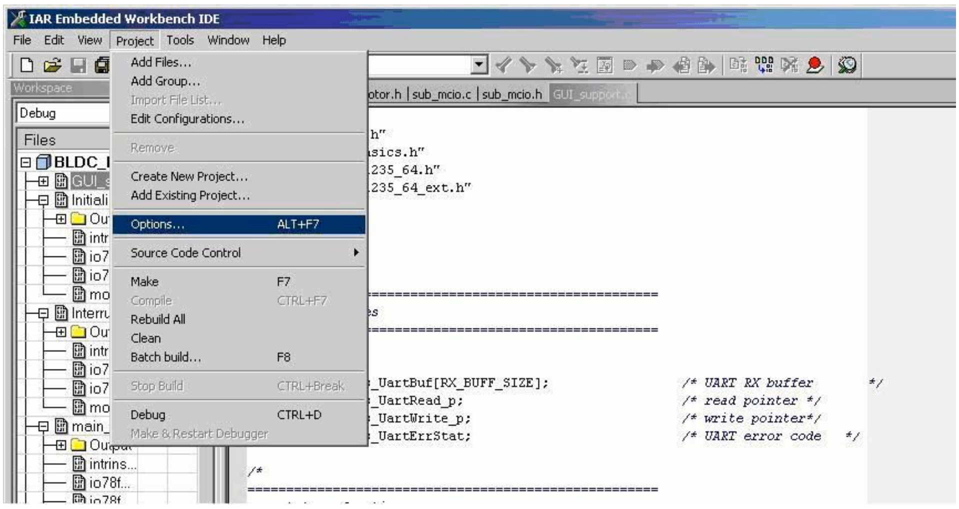


Figure 7-5 Project Build Options

7.4 General Options

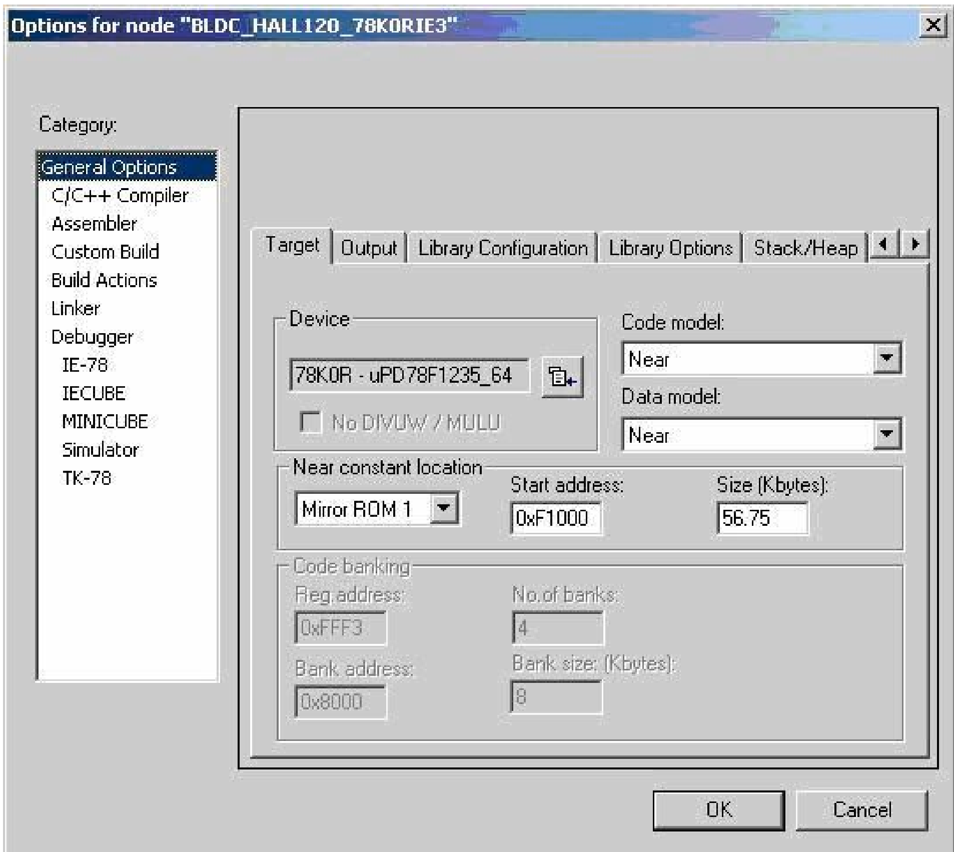


Figure 7-6 General Options – Setting the Target Device

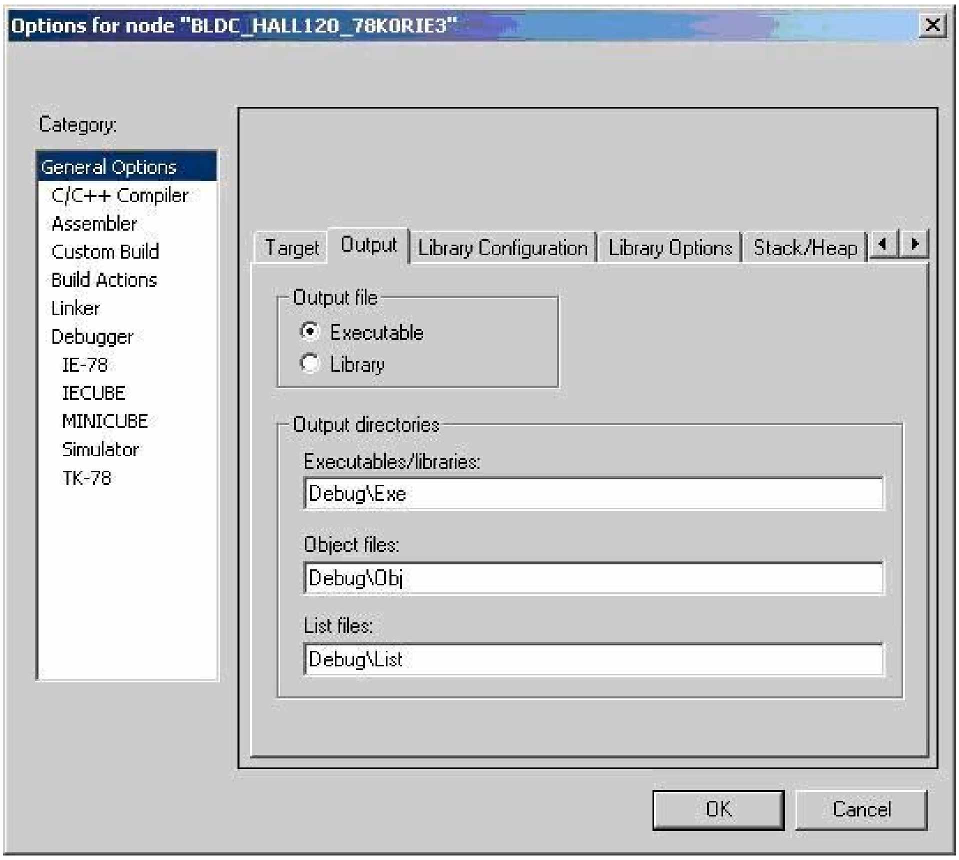


Figure 7-7 General Options – Setting the Output Locations

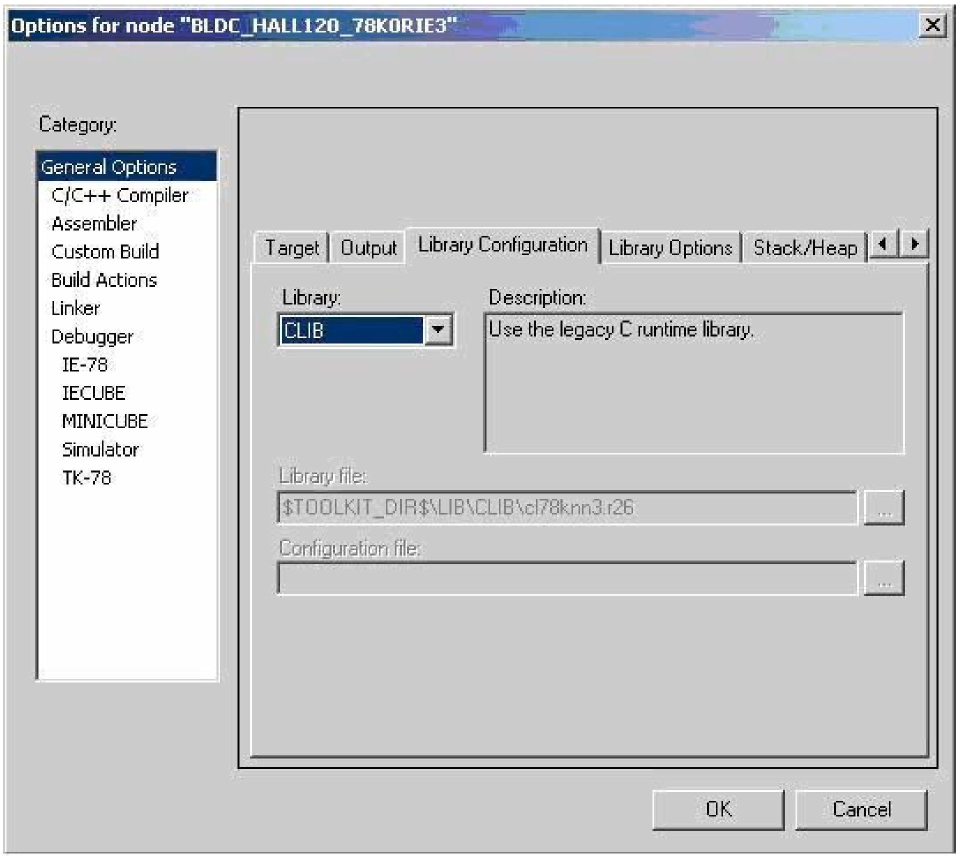


Figure 7-8 General Options – Selecting the C-Library

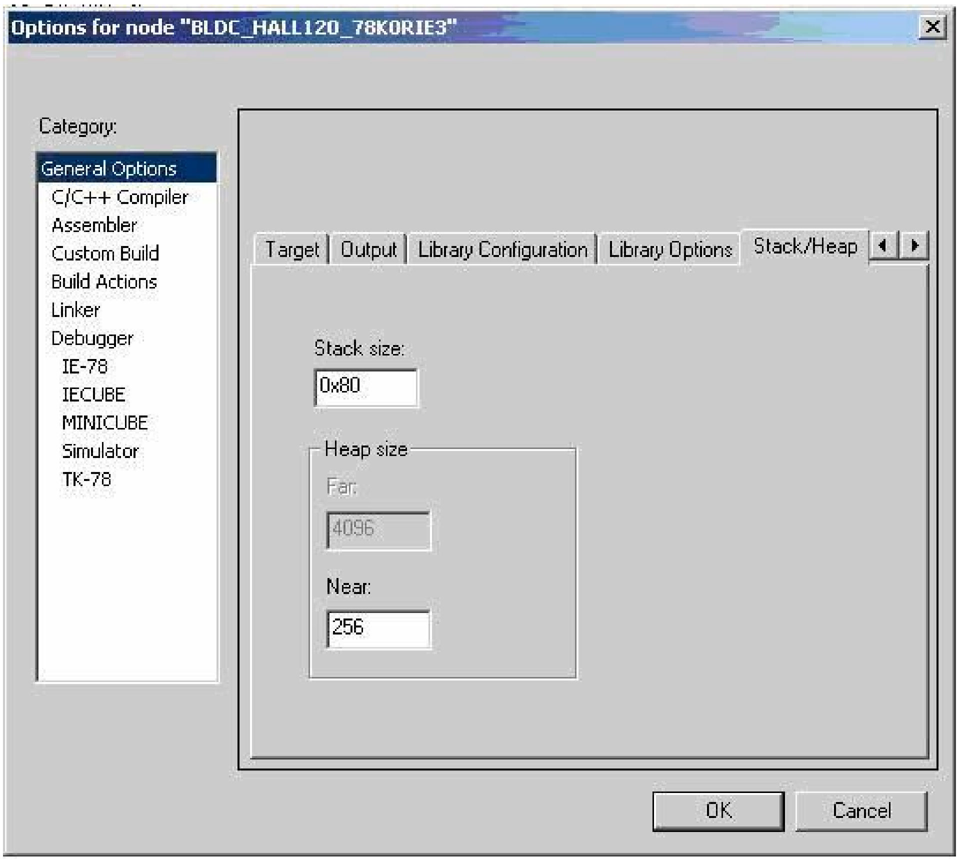


Figure 7-9 General Options – Setting the Stack and Heap

7.5 Compiler Options

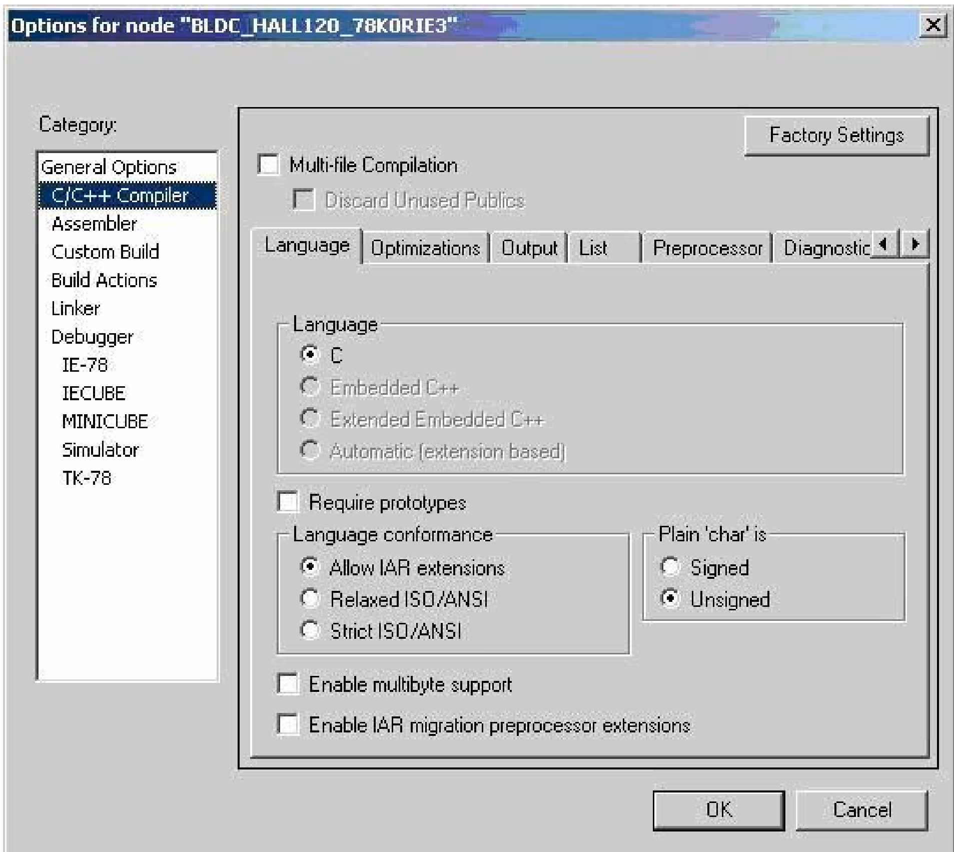


Figure 7-10 Compiler Options – Language Settings

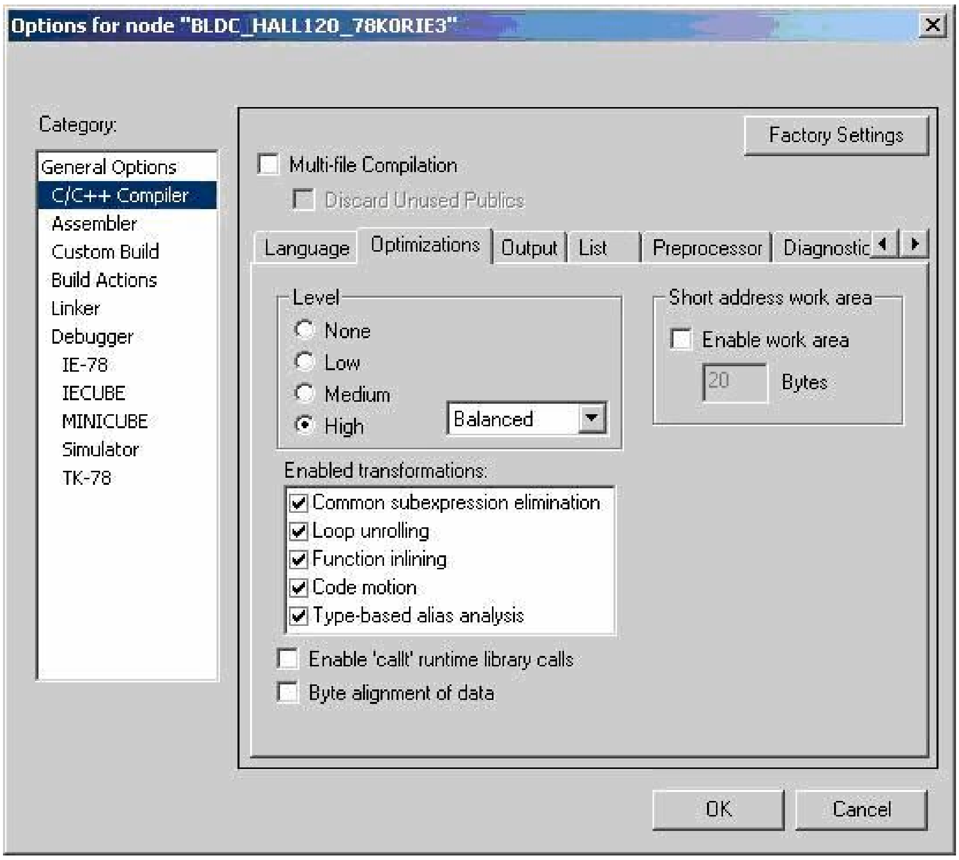


Figure 7-11 Compiler Options – Optimisation

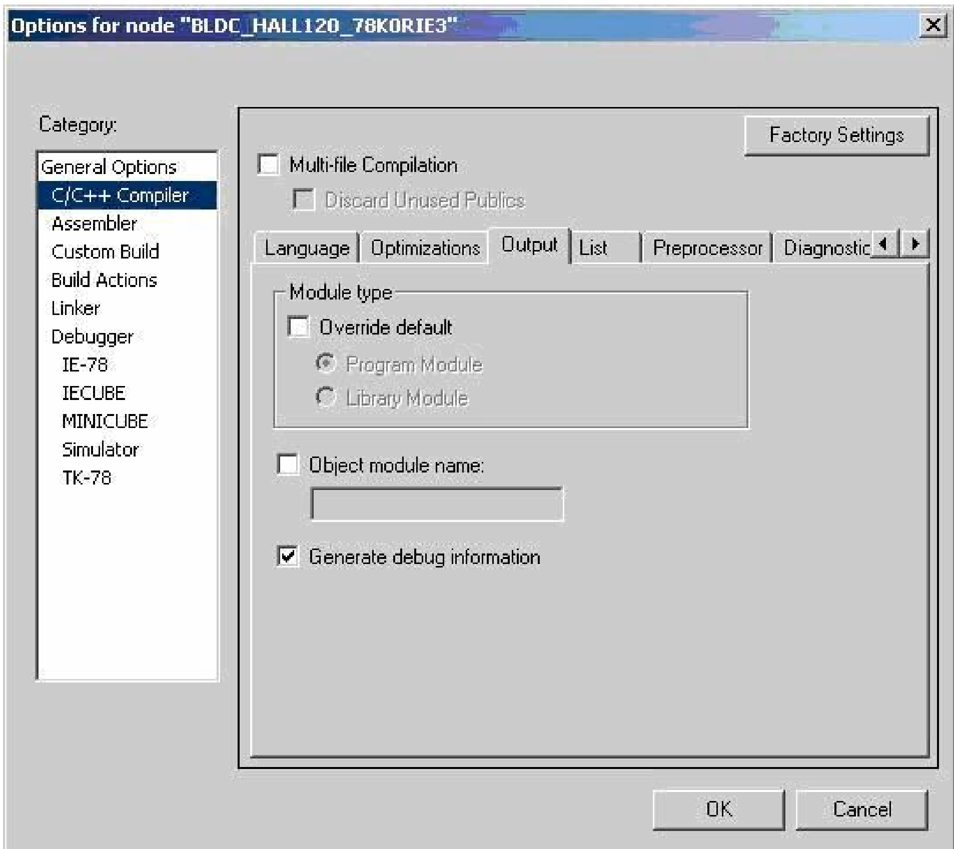


Figure 7-12 Compiler Options – Output Set for Debug

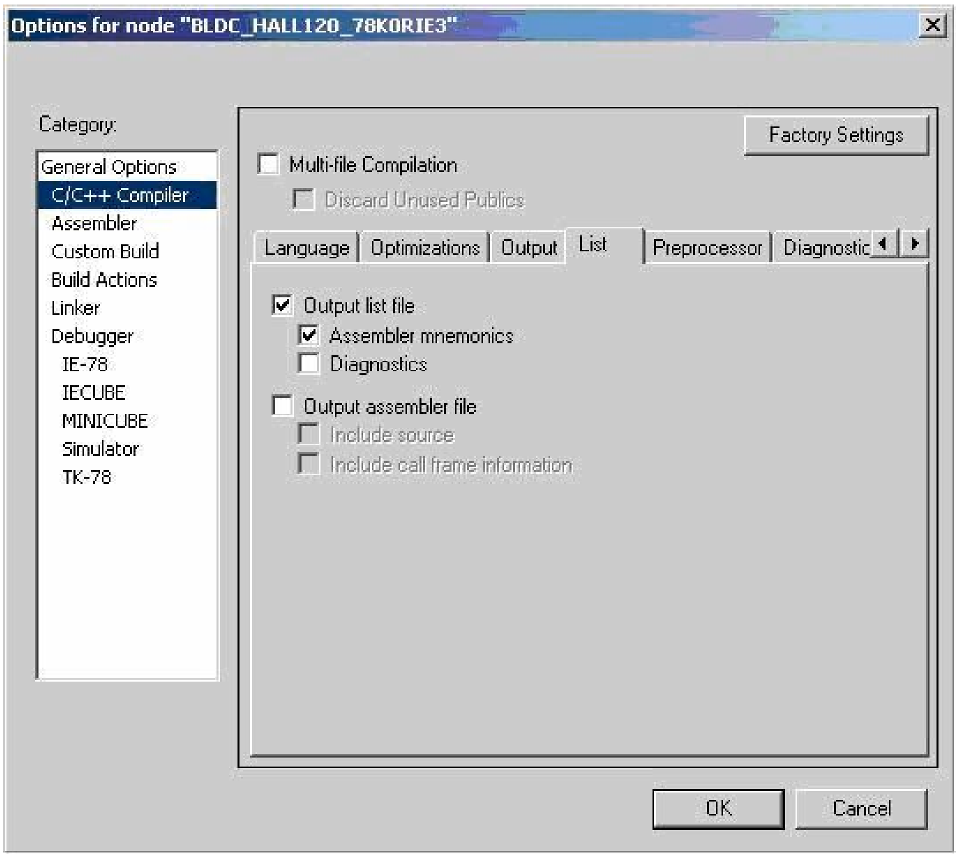


Figure 7-13 Compiler Options – Compiler Listings

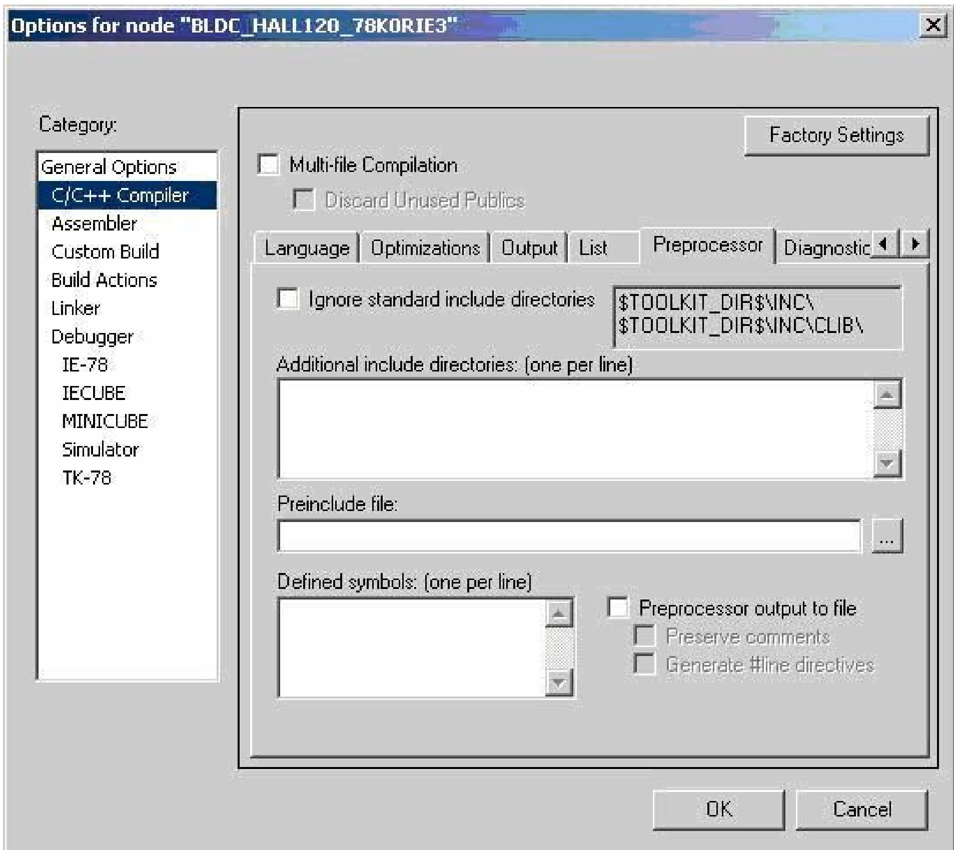


Figure 7-14 Compiler Options – Pre Processor Settings

Note All other compiler options settings can remain as the default settings.

7.6 Assembler Options

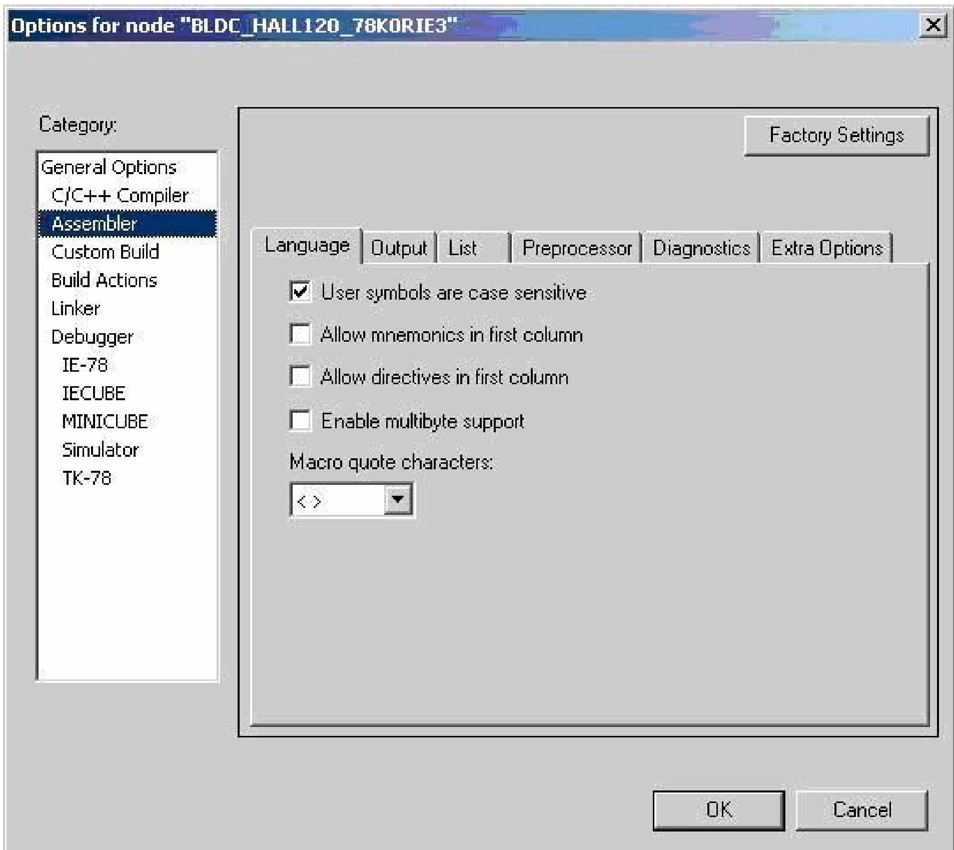


Figure 7-15 Assembler Options – Language Settings

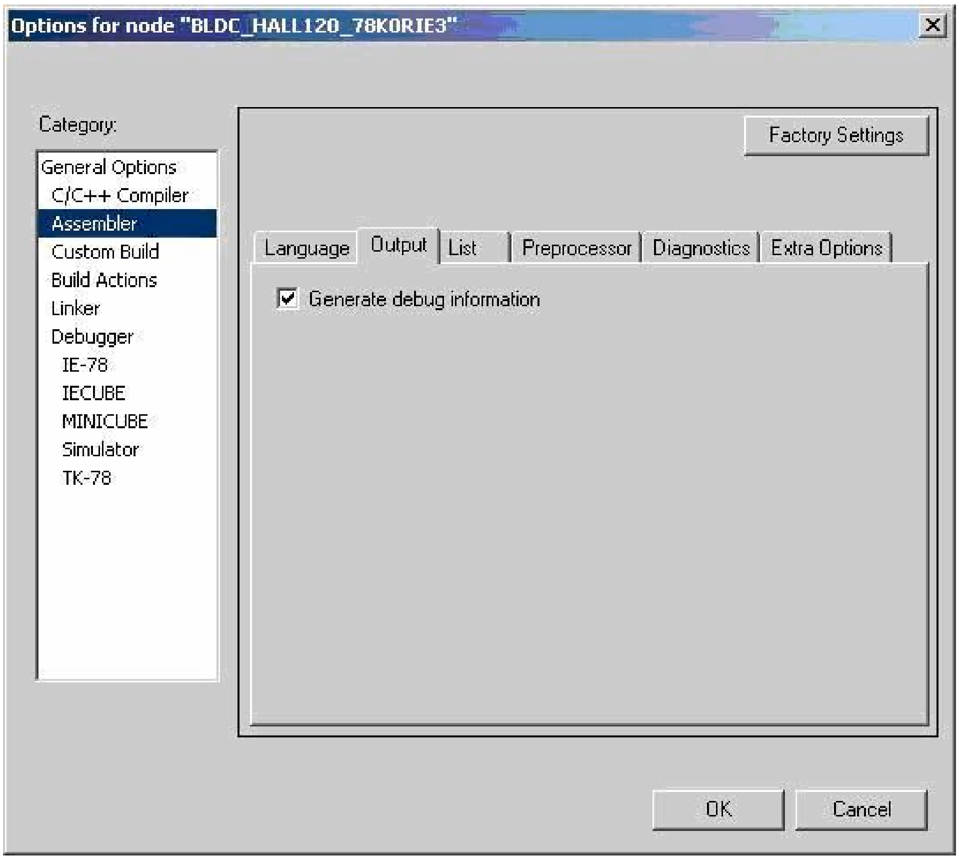


Figure 7-16 Assembler Options – Output Set for Debug

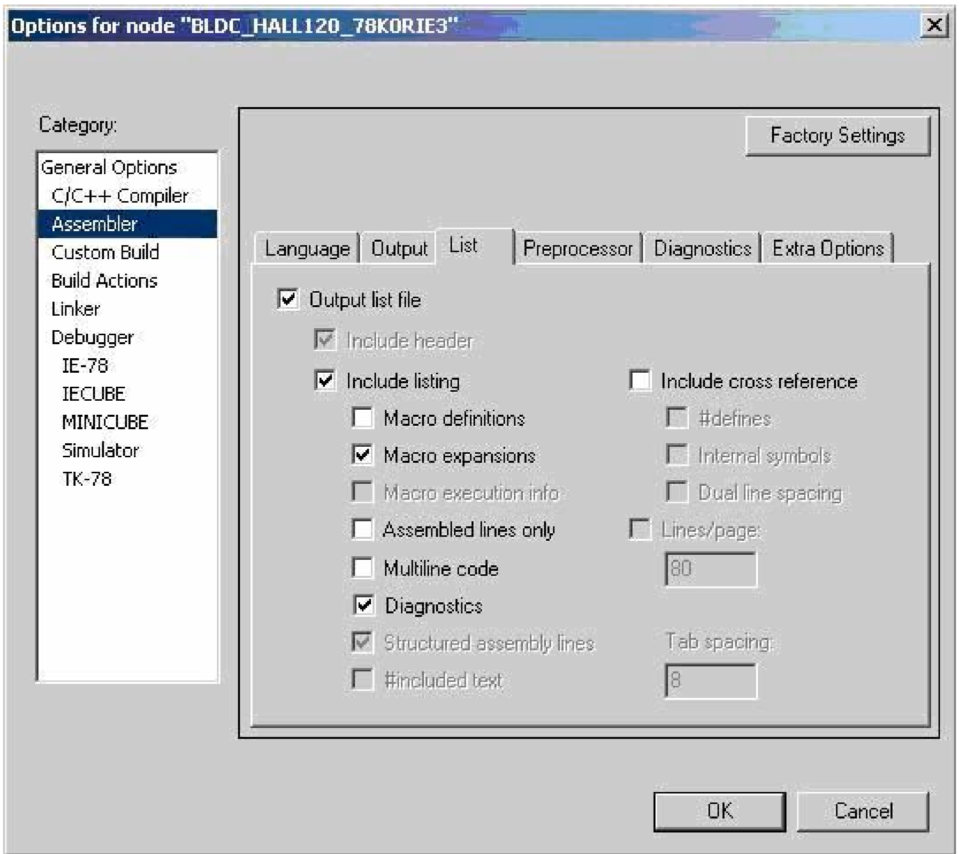


Figure 7-17 Assembler Options – Listings

7.7 Linker Options

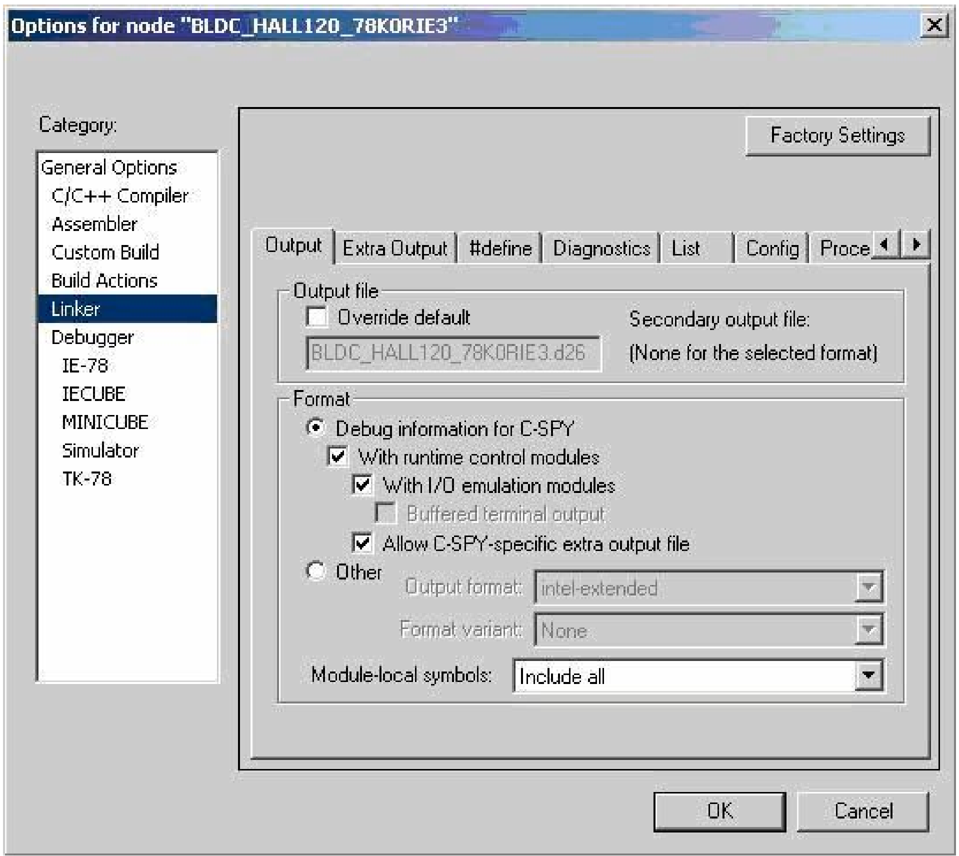


Figure 7-18 Linker Options – Primary File Output

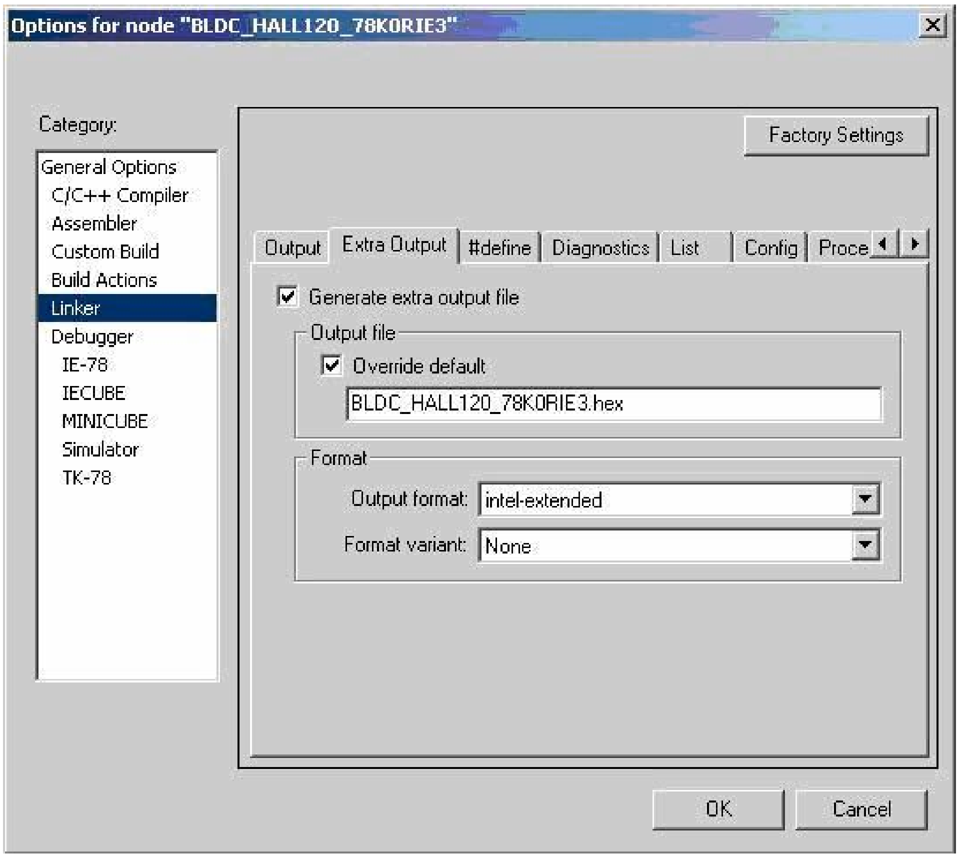


Figure 7-19 Linker Options – Secondary File Output

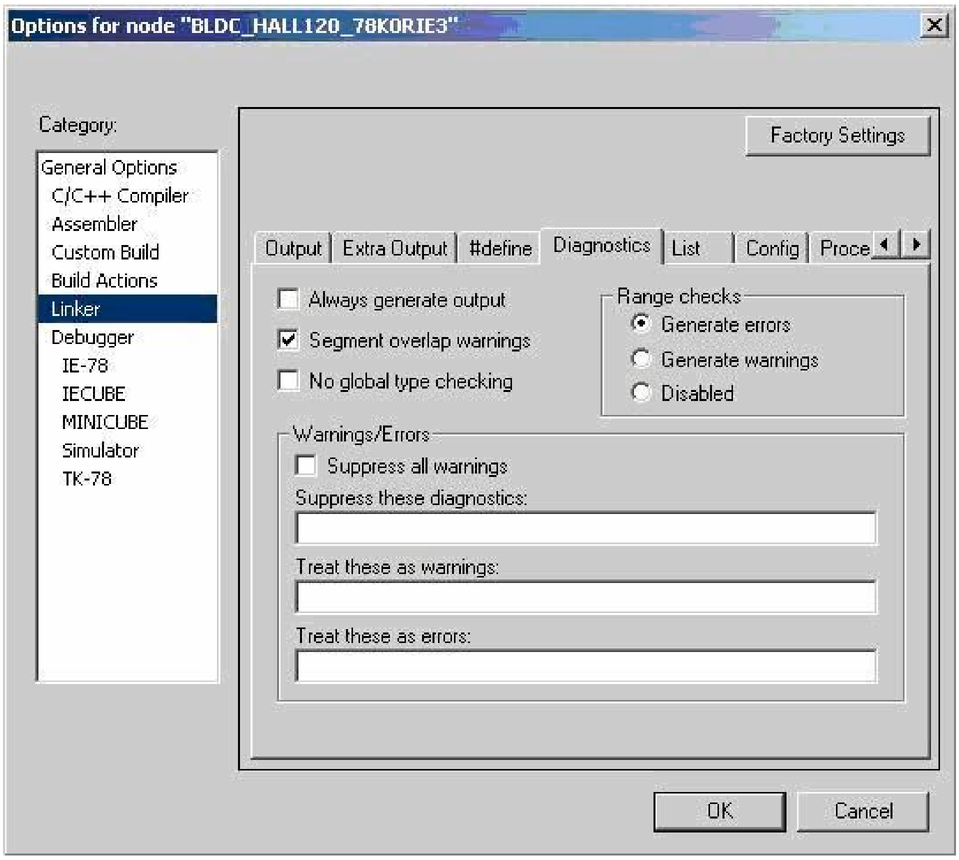


Figure 7-20 Linker Options – Diagnostic Settings

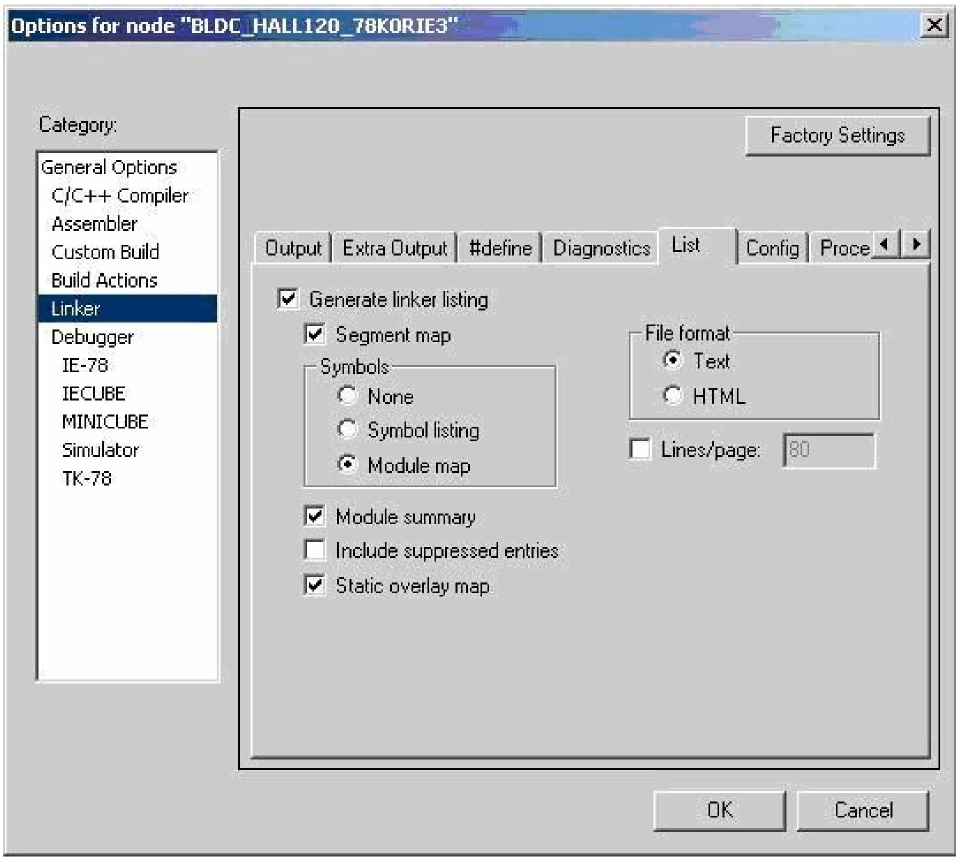


Figure 7-21 Linker Options – Generate MAP File Output

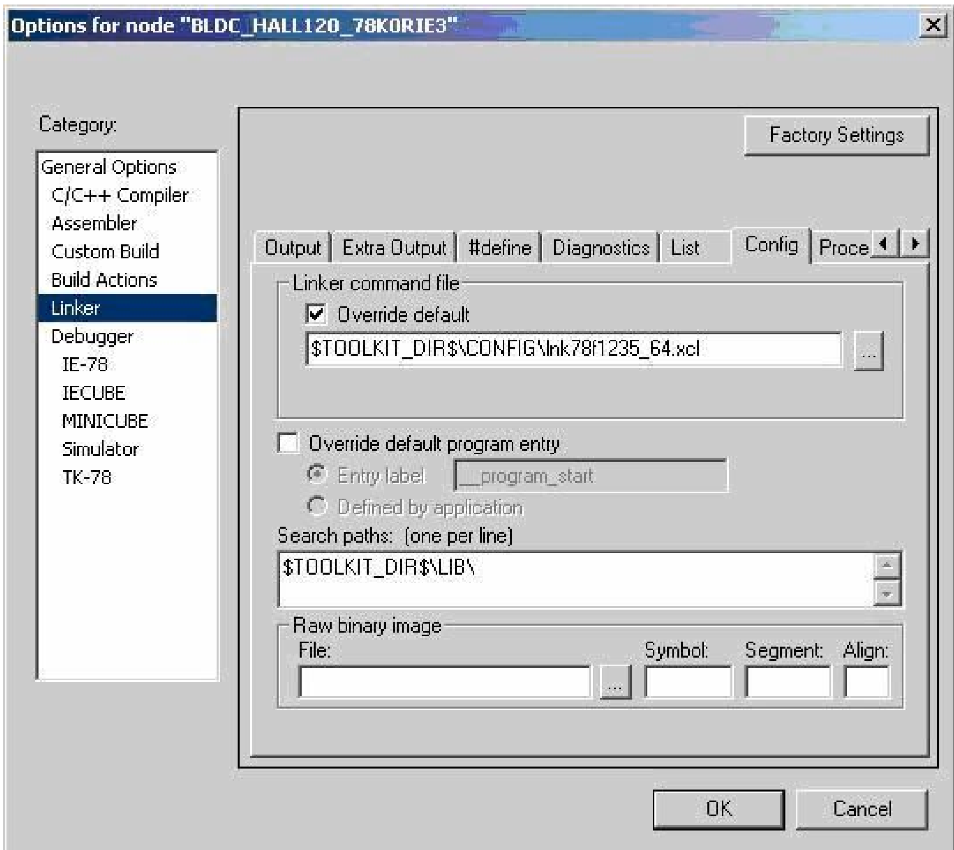


Figure 7-22 Linker Options – Linker Command File Selection

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

7.8 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.**

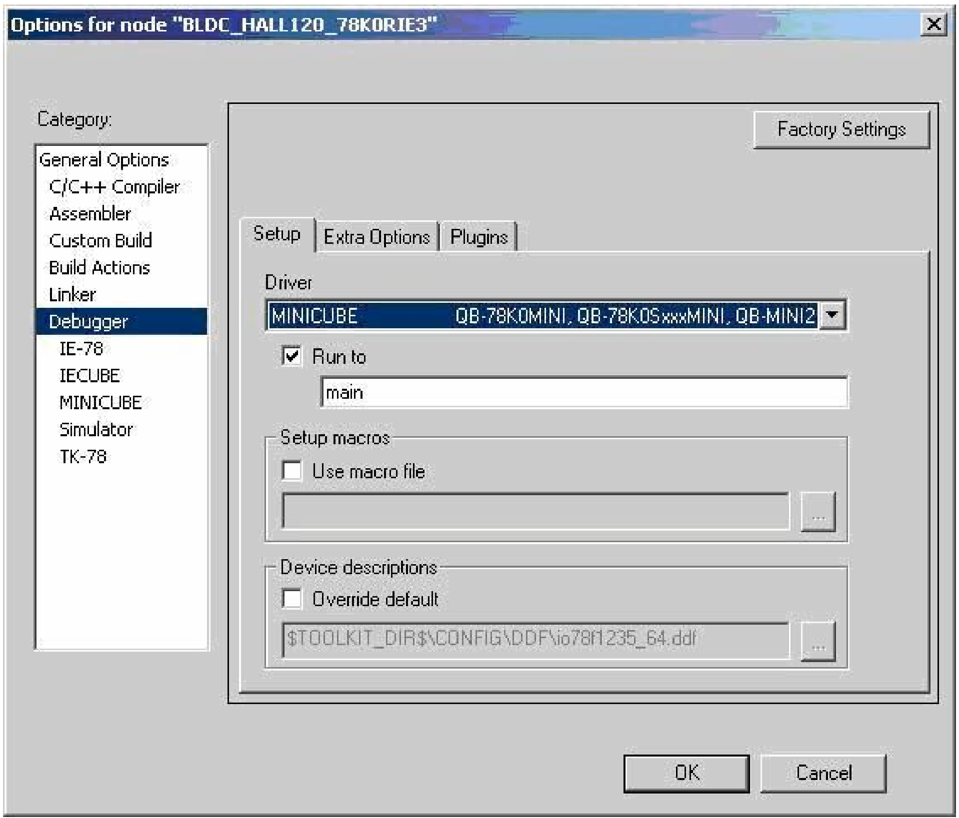


Figure 7-23 Integrated Debugger Selection

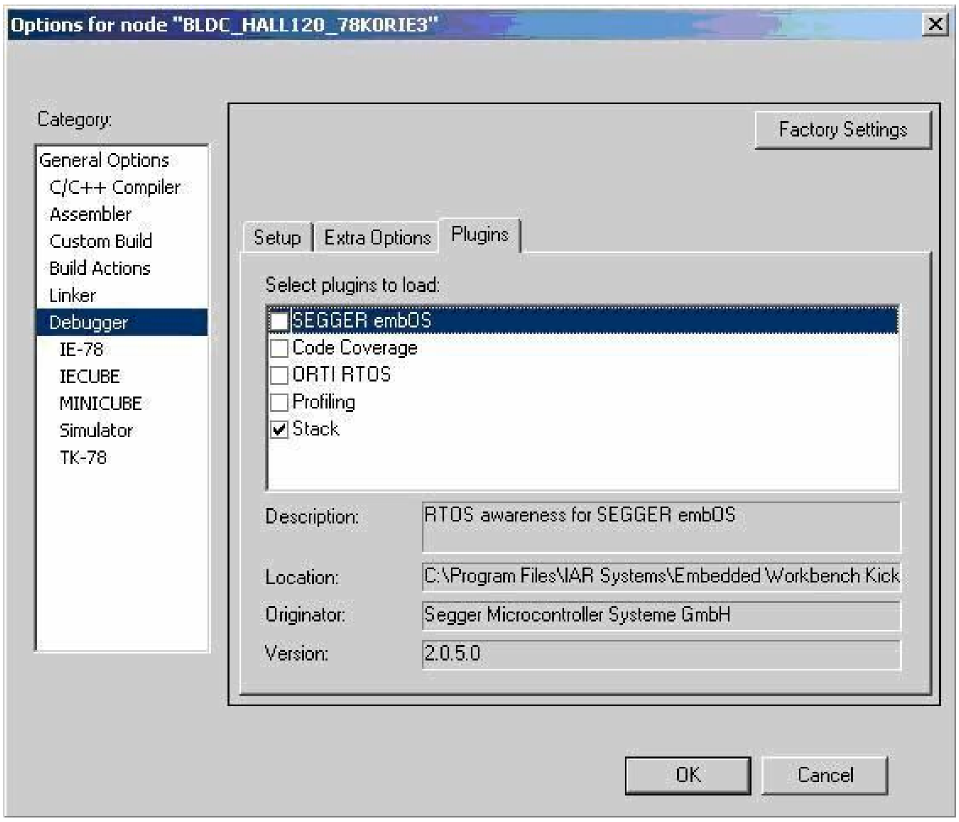


Figure 7-24 Integrated Debugger - Plug-in Selection

Note The "Extra Options" section can be ignored.

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

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

2. **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 files should now appear in the project window (left hand side of the IDE), as shown previously.

7.10 Build/Rebuild the Project

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

Figure 7-25 **Make Button**

The build results and any errors or warnings will be displayed in the messages window at the bottom of the IDE. These should be corrected before moving on to the debugging section.

7.11 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.

Figure 7-26 **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* .

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. MINICUBE2).

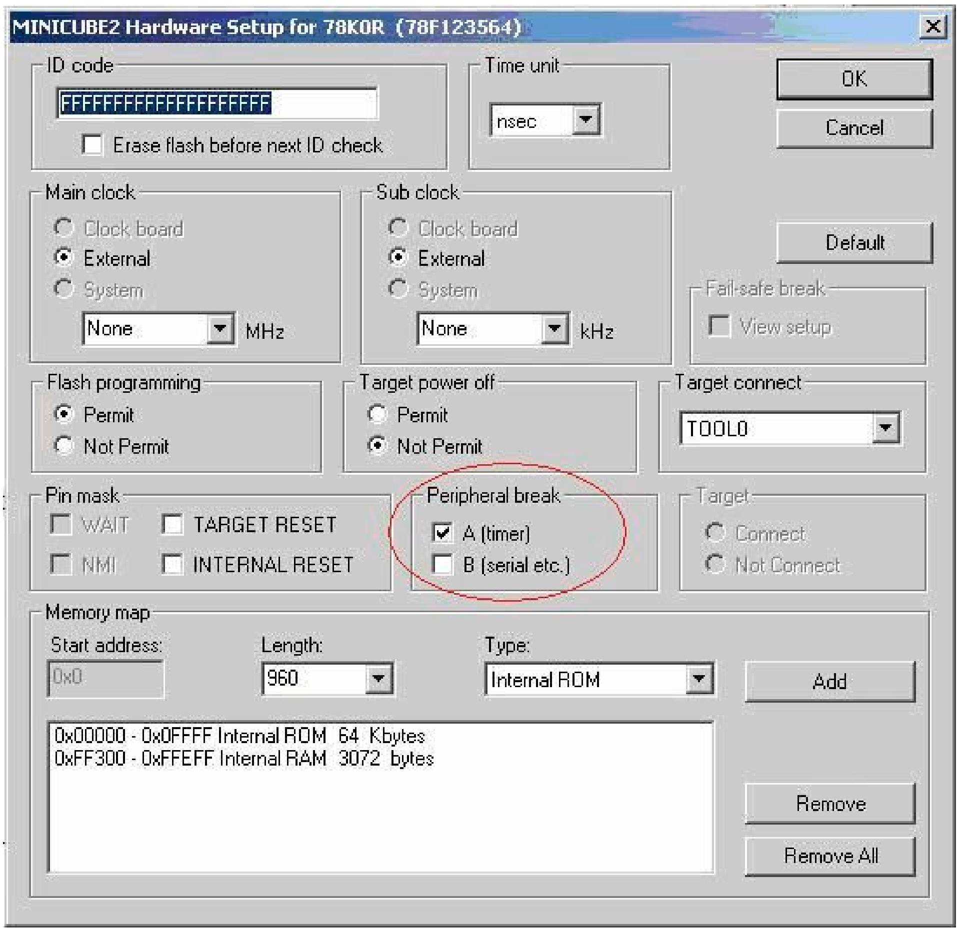


Figure 7-27 Debugging – Initial Hardware Setup

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.

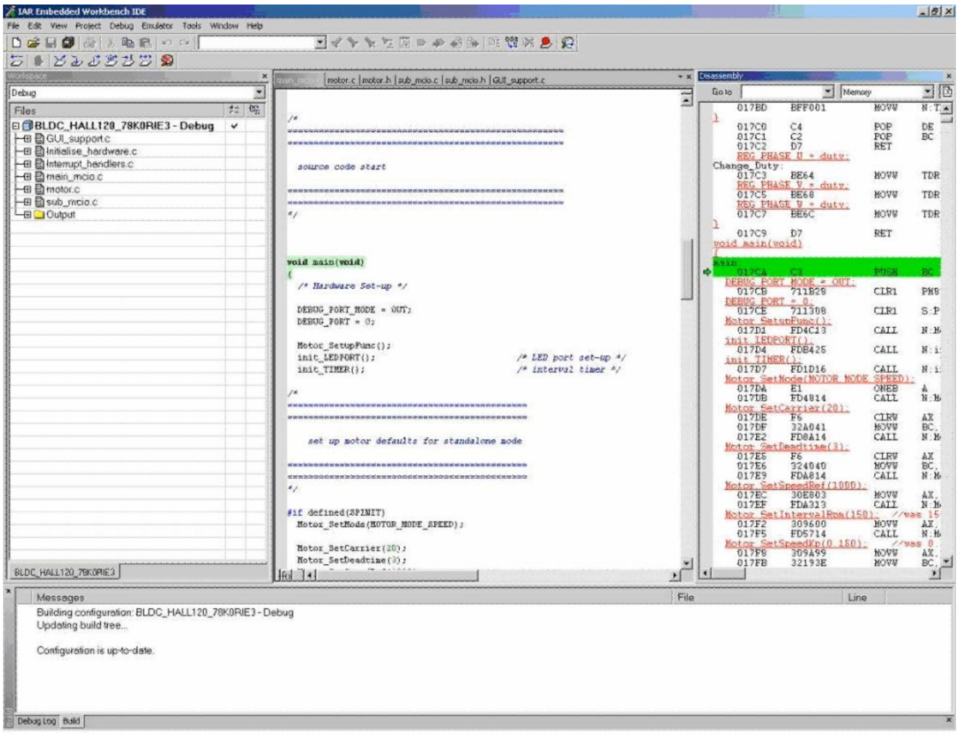


Figure 7-28 Integrated Debugger - Main Window

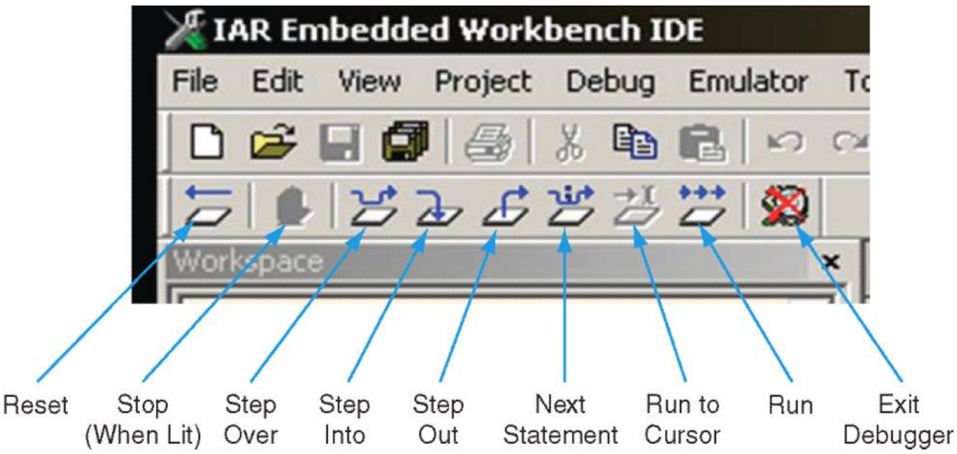


Figure 7-29 Debugger Task Bar Icons

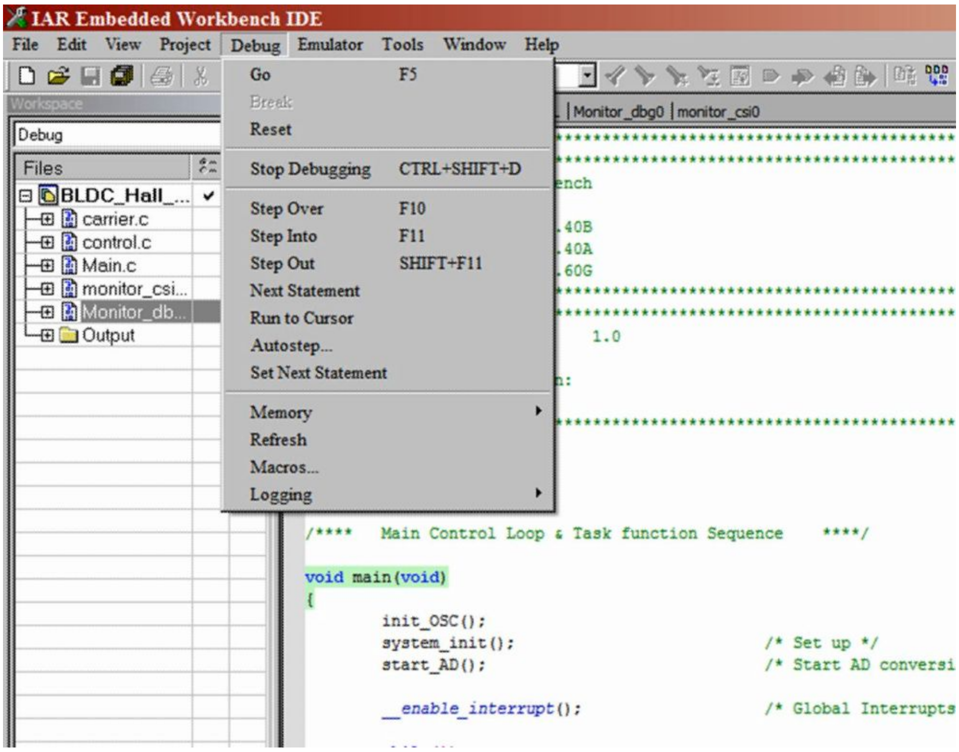


Figure 7-30 Debug Menus

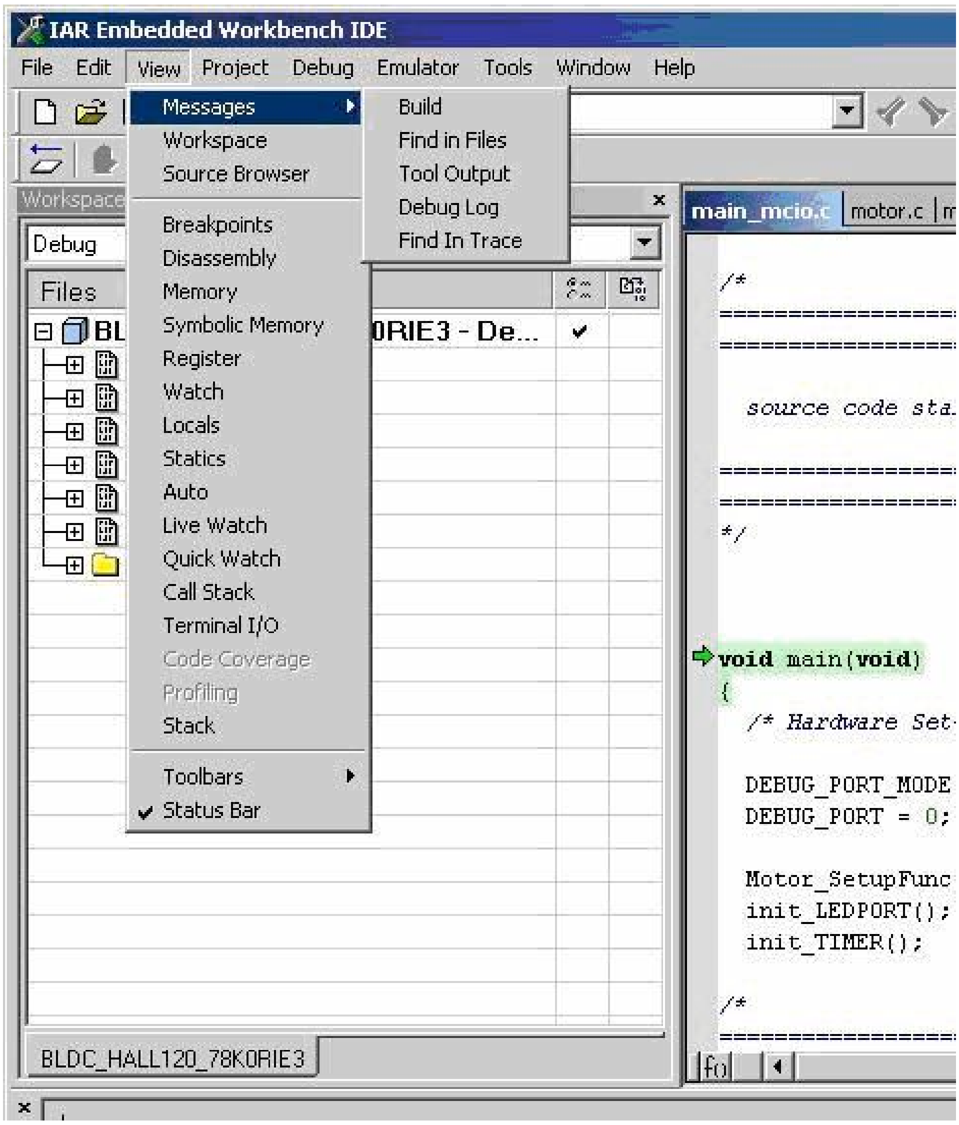


Figure 7-31 Debug Views Windows

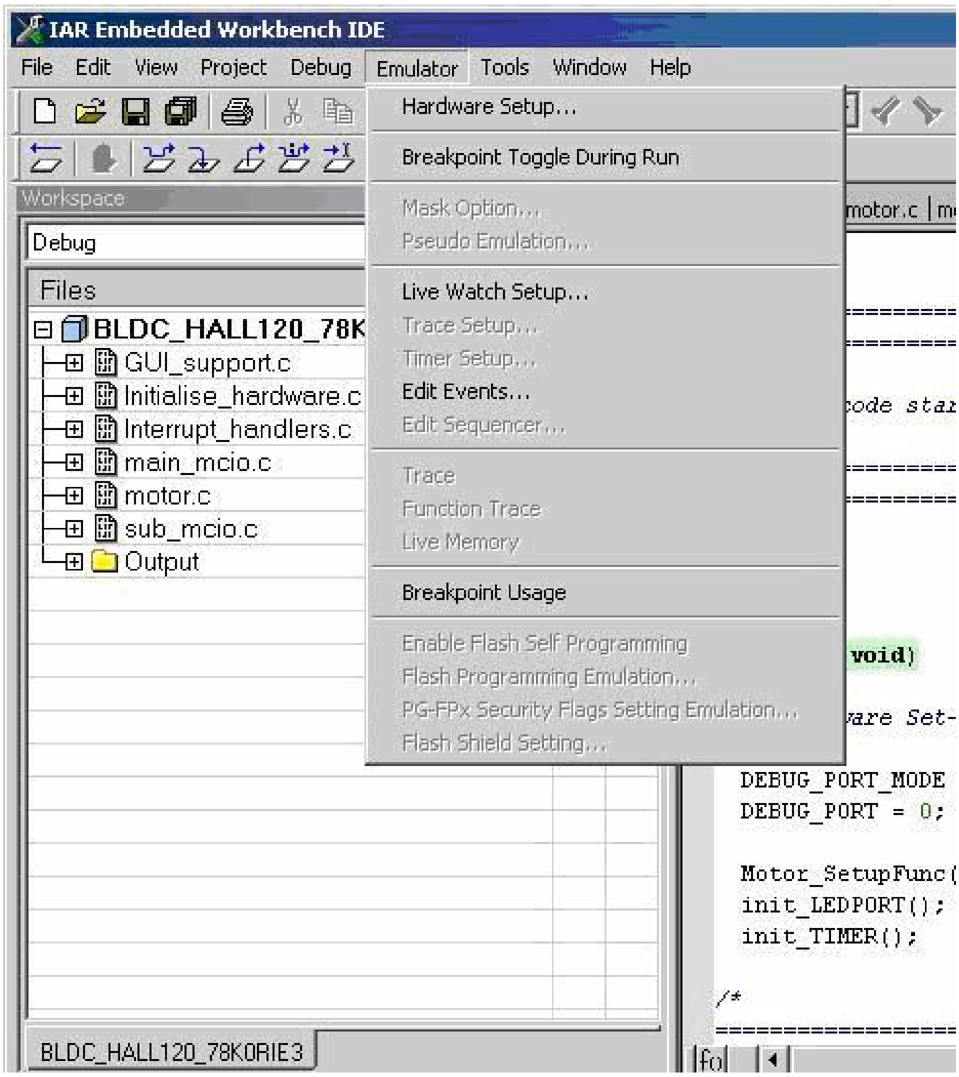


Figure 7-32 Emulator Debug Options

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

Chapter 8 Appendix

Schematics descriptions for the MC-CPU-78K0RIE3 CPU Daughter Card are attached to this document. Use the *Attachments* tab for access (lower left side of the screen).