Ru03
USER'S GUIDE

Version 1.0
September 2007

RIGEL CORPORATION
PO Box 90040, Gainesville, Florida 32607
(352) 384-3766,
www.rigelcorp.com,
techsupport@rigelcorp.com

HARDWARE WARRANTY

Limited Warranty. Rigel Corporation warrants, for a period of sixty (60) days from your receipt, that READS software, RROS, hardware assembled boards and hardware unassembled components shall be free of substantial errors or defects in material and workmanship which will materially interfere with the proper operation of the items purchased. If you believe such an error or defect exists, please call Rigel Corporation at (352) 384-3766 to see whether such error or defect may be corrected, prior to returning items to Rigel Corporation. Rigel Corporation will repair or replace, at its sole discretion, any defective items, at no cost to you, and the foregoing shall constitute your sole and exclusive remedy in the event of any defects in material or workmanship. Although Rigel Corporation warranty covers 60 days, Rigel shall not be responsible for malfunctions due to customer errors, this includes but is not limited to, errors in connecting the board to power or external circuitry. This warranty does not apply to products which have been subject to misuse (including static discharge), neglect, accident or modification, or which have been soldered or altered during assembly and are not capable of being tested.

DO NOT USE PRODUCTS SOLD BY RIGEL CORPORATION AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS!

Products sold by Rigel Corporation are not authorized for use as critical components in life support devices or systems. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

THE LIMITED WARRANTIES SET FORTH HEREIN ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

YOU ASSUME ALL RISKS AND LIABILITY FROM OPERATION OF ITEMS PURCHASED AND RIGEL CORPORATION SHALL IN NO EVENT BE LIABLE FOR DAMAGES CAUSED BY USE OR PERFORMANCE, FOR LOSS PROFITS, PERSONAL INJURY OR FOR ANY OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. RIGEL CORPORATION'S LIABILITY SHALL NOT EXCEED THE COST OF REPAIR OR REPLACEMENT OF DEFECTIVE ITEMS.

IF THE FOREGOING LIMITATIONS ON LIABILITY ARE UNACCEPTABLE TO YOU, YOU SHOULD RETURN ALL ITEMS PURCHASED TO RIGEL CORPORATION PRIOR TO USE.

Return Policy. This policy applies only when product purchased directly from Rigel Corporation. If you are not satisfied with the items purchased, prior to usage, you may return them to Rigel Corporation within thirty (30) days of your receipt of same and receive a full refund from Rigel Corporation. This does not apply to books. Books are non-returnable.

Please call (352) 384-3766 to receive an RMA (Returned Merchandise Authorization) number prior to returning product. You will be responsible for shipping costs. All returns must be made within 30 days of date of invoice and be accompanied by the original invoice number and a brief explanation of the reason for the return. Return merchandise in original packaging. All returned products are subject to a $15 restocking charge. "Custom Items" are not returnable.

Repair Policy. If you encounter problems with your board or software after the 60 day warranty period, please call Rigel Corporation at (352) 384-3766 or email techsupport@rigelcorp.com for advice and instruction.

Rigel Corporation will test and attempt to repair any board. You will be responsible for shipping costs and repair fees. If you send a detailed report of the problems you encountered while operating the board, Rigel Corporation will inspect and test your board to determine what the problem is free of charge. Rigel Corporation will then contact you with an estimated repair bill. You will have the choice of having the board fixed, returned to you as is, or purchasing a new board at a reduced price. Rigel Corporation charges repair fees based on an hourly rate of $50.00. Any parts that need to be replaced will be charged as separate items. Although Rigel Corporation will test and repair any board, it shall not be responsible for malfunctions due to customer errors, this includes but is not limited to, errors in connecting the board to power or external circuitry.

Board Kit. If you are purchasing a board kit, you are assumed to have the skill and knowledge necessary to properly assemble same. Please inspect all components and review accompanying instructions. If instructions are unclear, please return the kit unassembled for a full refund or, if you prefer, Rigel Corporation will send you an assembled and tested board and bill you the price difference. You shall be responsible for shipping costs. The foregoing shall apply only where the kit is unassembled. In the event the kit is partially assembled, a refund will not be available, however, Rigel Corporation can, upon request, complete assembly for a fee based on an hourly rate of $50.00. Although Rigel Corporation will replace any defective parts, it shall not be responsible for malfunctions due to errors in assembly. If you encounter problems with assembly, please call Rigel Corporation at (352) 384-3766 for advice and instruction. In the event a problem cannot be resolved by telephone, Rigel Corporation will perform repair work, upon request, at the foregoing rate of $50.00 per hour.

Governing Law. This agreement and all rights of the respective parties shall be governed by the laws of the State of Florida.

Rigel Corporation’s Software License Agreement

This Software License Agreement ("Agreement") covers all software products copyrighted to Rigel Corporation, including but not limited to: Reads51, rLib51, RbHost, RitaBrowser, rFLASH, rChipSim, rP51, Reads166, and rFLI. This Agreement is between an individual user or a single entity and Rigel Corporation. It applies to all Rigel Corporation software products. These Products ("Products") includes computer software and associated electronic media or documentation "online" or otherwise. Our software, help files, examples, and related text files may be used without fee by students, faculty and staff of academic institutions and by individuals for non-commercial use. For distribution rights and all other users, including corporate use, please contact:

Rigel Corporation, PO Box 90040, Gainesville, FL 32607
E-mail: techsupport@rigelcorp.com

TERMS AND CONDITIONS OF THE AGREEMENT

These Products are protected by copyright laws, intellectual property laws, and international treaties. Rigel Corporation owns the title, copyright, and all other intellectual property rights in these Products. We grant you a personal, non-transferable, and non-exclusive license to use the Products. These Products are not transferred to you, given away to you or sold to you.

Non-commercial use: These Products are licensed to you free of charge.

Commercial use: You must contact Rigel Corporation to find out if a licensing fee applies before using these Products.

You may install and use an unlimited number of copies of these Products.

You may store copies of these Products on a storage device or a network for your own use.

You may not reproduce and distribute these Products to other parties by electronic means or over computer or communication networks. You may not transfer these Products to a third party. You may not rent, lease, or lend these Products.

You may not modify, disassemble, reverse engineer, or translate these Products.
These Products are provided by Rigel Corporation "as is" with all faults.

In no event shall Rigel Corporation be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the Product, even if Rigel Corporation has been advised of the possibility of such damages. Because some states do not allow the exclusion or limitations of consequential or incidental damages, the above limitations may not apply to you.

Rigel Corporation makes no claims as to the applicability or suitability of these Products to your particular use, application, or implementation.

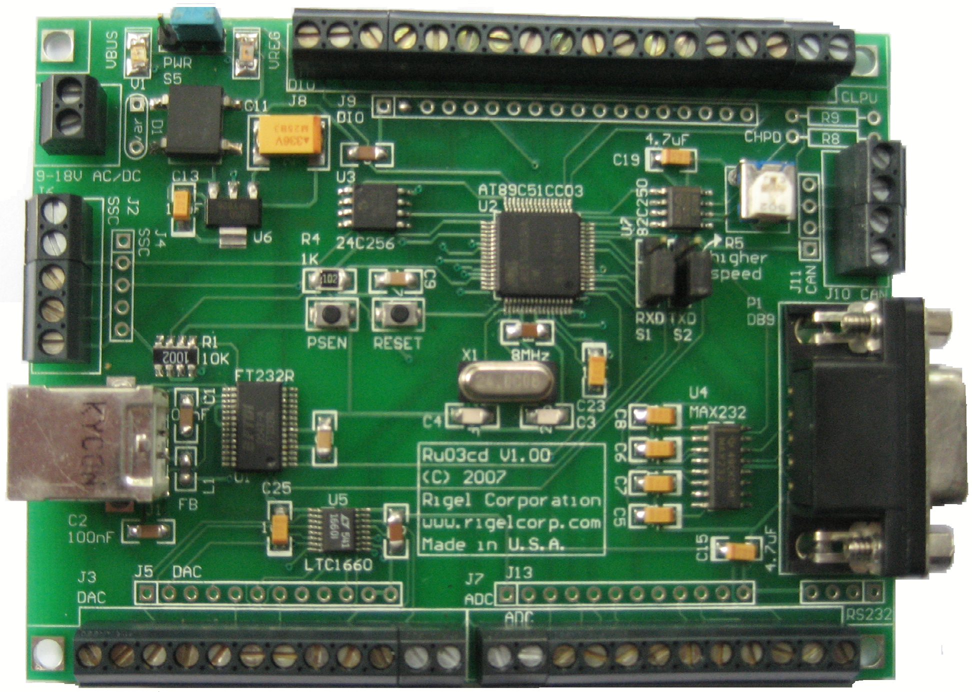
Rigel Corporation reserves all rights not expressly granted to you in this Agreement.

If you do not abide by or violate the terms and conditions of this Agreement, without prejudice to any other rights, Rigel Corporation may cancel this Agreement. If Rigel Corporation cancels this Agreement; you must remove and destroy all copies of these Products.

If you acquired this Product in the United States of America, this Agreement is governed by the laws of the Great State of Florida. If this Product was acquired outside the United States of America, all pertinent international treaties apply.

1 INTRODUCTION

The Ru03 uses the Atmel AT89C51CC03 microcontroller with 2K of internal RAM and 64K of internal FLASH memory. The code memory is programmed via the Atmel FLIP utility. The processor has a serial port which is used for both programming the internal FLASH code memory, and for communications by the application. The Ru03 board has a serial-to-USB converter (FT232R) that can be used to communicate with the microcontroller.



The serial-to-USB converter appears to the PC as a (Virtual Communications Port (VCP) with a COMM port number, e.g. COMM 9. Besides the USB port, the Ru03 has a Controller Area Network (CAN) port, and a Serial Port Interface (SPI) port. These make the Ru03 a highly network-able controller.

The Ru03 board has an external Digital-to-Analog Converter. The AT89C51CC03 has an on-chip Analog-to-Digital Converter (ADC). Having both ADC and DAC makes the Ru03 a powerful platform for signal processing.

The processor has 2 KB of on-chip EEPROM and 64KB of FLASH. In addition, the Ru03 has an external serial 32KB (or 64KB) EEPROM. The capability to store data in nonvolatile memory makes the Ru03 a good choice for data collection tasks.

The Ru03 has three types of ports. There are 14 digital input/output bits and 8 analog output bits. In addition, there are 8 analog input bits. Any subset of the analog input bits may be used as general-purpose digital input/output bits.

1.1 Board Features

- 32KB external serial EEPROM (optional 64KB external serial EEPROM)
- On-board DAC (LTC1660)
- 14 general purpose digital input/output bits on screw type terminal blocks
- 8 Analog input bits on screw type terminal blocks
- 8 Analog output bits driven by an the serial DAC
- CAN (Controller Area Network) physical layer (82C250 or 82C251) built into the board
- CAN port terminates on screw-type terminal blocks
- Serial Port Interface (SPI) port which terminates on screw-type terminal blocks
- Serial-to-USB converter which appears as a VCP (Virtual Communications Port) with a COMM port number
- Power on LED
- Board operates on +5 volts supplied by one of two methods:
  – Power supplied to the board by external power source through a 2 position terminal block
  – Though the USB post
- Operating temperature 0 to 70C
- See Section 7 for OEM options

1.2 Chip Features

- 80C51 Core Architecture
- 256 Bytes of On-chip RAM
- 2048 Bytes of On-chip ERAM
- 64K Bytes of On-chip Flash Memory
  – Data Retention: 10 Years at 85°C

- Read/Write Cycle: 100K
- 2K Bytes of On-chip Flash for Bootloader
- 2K Bytes of On-chip EEPROM
- 14-sources 4-level Interrupts
- Three 16-bit Timers/Counters
- Full Duplex UART Compatible 80C51
- High-speed Architecture
- Five Ports: 32 + 4 Digital I/O Lines
- Five-channel 16-bit PCA with
  - PWM (8-bit)
  - High-speed Output
  - Timer and Edge Capture
- Double Data Pointer
- 21-bit WatchDog Timer (7 Programmable Bits)
- A 10-bit Resolution Analog to Digital Converter (ADC) with 8 Multiplexed Inputs
- SPI Interface, (PLCC52, VPFP64 and CABGA 64 packages only)
- Full CAN Controller
  - Fully Compliant with CAN Rev 2.0A and 2.0B
  - 15 Independent Message Objects
  - Each Message Object Programmable on Transmission or Reception
  - Individual Tag and Mask Filters up to 29-bit Identifier/Channel
  - 8-byte Cyclic Data Register (FIFO)/Message Object

2 SOFTWARE

The Ru03 uses two different software packages, Reads51 for code development, and Atmel FLIP to download the code to the board.

2.1 Reads51

Reads51 is an integrated applications software development system, which runs on an IBM PC or compatible host. Reads51 allows writing, compiling, assembling, debugging, downloading, and running applications software in the MCS-51 language. Reads51 contains a C compiler, relative assembler, linker/locator, editor, chip simulator, assembly language debugger, and host-to-board communications in a user-friendly, menu-driven environment.

2.1.1 System Requirements

All of Rigel’s software is designed to work with an IBM PC or compatible, Pentium 120MHz or better, running Windows 95, 98, 2000, ME, NT, Windows XP and Vista.

2.1.2 Software Installation

Reads51 may be downloaded from Rigel's website www.rigelcorp.com. Go to the **Downloads | Download 8051 Software** and click on the .exe file you wish to install. The program will then download to your computer. Follow the standard install directions that come up on screen. The newest versions of our software are always available to download off our web site. We encourage you to check our web site often to keep up-to-date.

2.2 FLIP Software

Software is written using Reads51 and the hex file is downloaded to the board using the Atmel FLIP utility. FLIP communicates with the microcontroller and programs the internal FLASH on the processor.

FLIP is a free PC-based software tool from Atmel that supports the In-System Programming (ISP) of Atmel’s Flash C51 microcontrollers through various communication interfaces, including RS232, USB and CAN links. FLIP can be used through a Graphical User Interface or launched from a DOS command window. FLIP is a powerful toolset which enables the user to easily embed Atmel’s ISP function libraries (DLL) within their applications without having to know the details of Atmel's ISP protocols. FLIP adds flexibility, power and user-friendliness to Atmel Flash C51 devices In-System Programming.

For more information on the FLIP utility and to downloaded the software go to the Atmel website (www.atmel.com). The current version of FLIP is 3.2.

3.0 OPERATING NOTES

3.1 Power Options

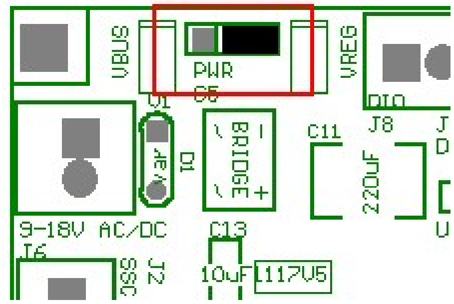
The board may be powered either through the USB port, or by an external power source. The Ru03 has a power regulator for use with the external power source.

Jumper S5 determines which power source is used.

To power the board using one of two the methods:

a. Using the USB port power, J1. Place the power jumper S5 in the left position. Use a standard USB cable and connect the board to a USB port on the PC.

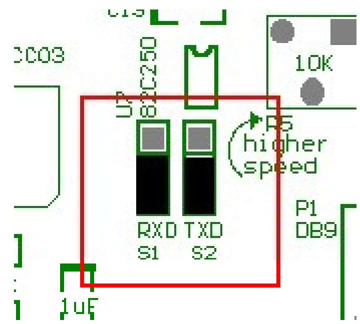
b. Using an external power supply. Connect a 6-18V AC or DC power supply to the terminal block, J6. Place the power jumper S5 in the right position. (Picture shows S5 in the right position for use with external power supply)



You may remove jumper S5 to disconnect the board from power.

3.2 PC Communications Options

The board uses the serial port of the microcontroller to communicate with the PC. The Ru03 can be connected to the PC by either the RS232 serial port or the USB port. If the RS232 port is used the jumpers S1 and S2 must be placed in the upper position. If the USB port is to be used, move the jumpers S1 and S2 to the lower position. When using USB port, the FT232R drivers must be installed so that the USB connection appears as a Virtual COM Port (VCP) to the PC. See Section 6.2.1 for details. (Picture shows jumpers S1 and S2 in the lower position, for use with the USB port)



3.3 Reset

The microcontroller has its own circuitry to reset upon a power up.

S3 is a push button on the board that is connected to the reset input of the microcontroller. Pressing the button resets the board. This is useful for in-system debugging.

S4 is an additional push button on the board marked "PSEN." This button grounds the PSEN pin of the microcontroller through a resistor. Upon reset, the microcontroller inspects the state of the PSEN pin. If grounded, the microcontroller enters the programming mode. In this mode, the microcontroller communicates with the Atmel FLIP utility. The PSEN button may be used to force the microcontroller into the programming mode.

The PSEN signal is normally an output, keeping the PSEN button pressed for a long time may damage the microcontroller. Please refer to Section 6.2.4 on programming for further information.

3.4 LEDs

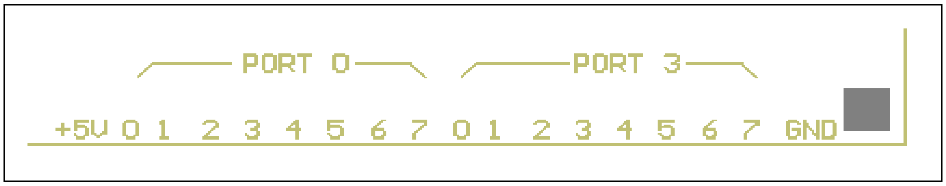
There are two LEDs, one for each power source. The LEDs are lit if the corresponding power source is active, irrespective of the state of jumper S5. Note that the USB LED (VBUS) will be lit when the USB port is connected to a host, even if the board is powered from the external source. Similarly, the external power source LED (VREG) is lit as long as external power is applied, even when the power is taken from the USB.

4 RU03 PORTS AND CONNECTIONS

The board has a variety of connector choices for ports, power, and the USB. Board connections include terminal blocks used for power, digital input/output ports, analog input ports, analog output ports, the CAN port, and the SPI port. The board can be populated with headers instead of the terminal blocks for all of these connections. In addition there is a USB connector and a DB9 for the serial port.

4.1 Digital Input/Output Connections

The processor ports P0.0 to P0.7 and P3.2 to P3.7 are available as digital input output bits. In addition, any pin of Port 1 (P1.0 to P1.7) that is not used as an analog input may also be used as a digital input/output bit.



Pin 1 is marked by a square pad, visible from the bottom side of the board. The port numbers are shown on the bottom silkscreen.

J8 is a 18-position screw-type terminal block for the digital input/outputs. J9 could be populated with a header if the terminal blocks are not needed. The two connectors (J8 and J9) have the same pin order.

J8 / J9 Pinout

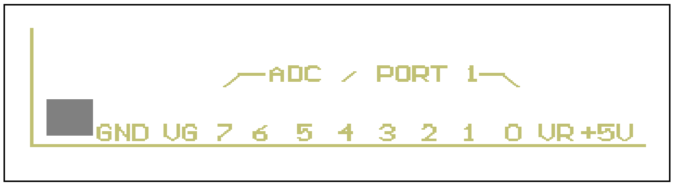
Position	Signal	Remark
1	VCC	+5V
2	P0.0	
3	P0.1	
4	P0.2	
5	P0.3	
6	P0.4	
7	P0.5	
8	P0.6	
9	P3.0	Used by RxD
10	P3.1	Used by TxD
11	P3.2	
12	P3.3	
13	P3.4	
14	P3.5	
15	P3.6	
16	P3.7	
17	P3.0	
18	GND	

4.2 Analog Input Ports

Port 1 pins of the microcontroller may be used as analog inputs. The ADC is a 10-bit Analog-to-Digital Converter with 8 multiplexed inputs with an input voltage range of 0 to 3V. Any pin of Port 1 (P1.0 to P1.7) that is not used as an analog input may be used as a digital input/output bit.

4.3 Analog Input Connections

J7 is a 12-position screw-type terminal block for the analog inputs. Pin 1 is marked by a square pad, visible from the bottom side of the board.



J13 could be populated with a header if the terminal blocks are not needed. The two connectors (J7 and J13) have the same pin order.

J7 / J13 Pinout

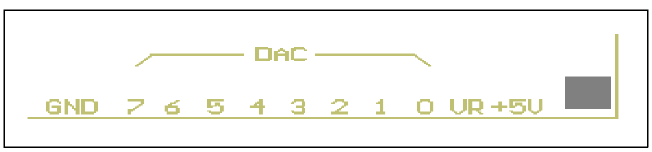
Position	Signal	Remark
1	VCC	+5V
2	VAREF	Analog reference (3.0V max)
3	P1.0	
4	P1.1	
5	P1.2	
6	P1.3	
7	P1.4	
8	P1.5	
9	P1.6	
10	P1.7	
11	VAGND	Analog ground
12	GND	

4.4 Analog Output Ports

The analog output ports are driven by an serial DAC. The DAC interface is accomplished by Port 2 pins. See Section 4.7.1 for more information.

4.5 Analog Output Connections

J3 is a 12-position screw-type terminal block for the analog outputs. Pin 1 is marked by a square pad, visible from the bottom side of the board.



J5 could be populated with a header if the terminal blocks are not needed. The two connectors (J3 and J5) have the same pin order.

J3 / J5 Pinout

Position	Signal	Remark
1	VCC	+5V
2	VDREF	DAC reference (5.0V max)
3	DAC.0	
4	DAC.1	
5	DAC.2	
6	DAC.3	
7	DAC.4	
8	DAC.5	
9	DAC.6	
10	DAC.7	
11	GND	
12	GND	

4.6 SPI Port Connections

J2 is a 5-position screw-type terminal block for the SPI communications port. Pin 1 is marked by a square pad, visible from the bottom side of the board. J4 could be populated with a header if the terminal blocks are not needed. The two connectors (J2 and J4) have the same pin order.

J2 / J4 Pinout

Position	Signal	Remark
1	VCC	+5V
2	MISO	Master In. Slave Out
3	MOSI	Master Out, Slave In
4	SCK	Clock
4	GND	Ground

4.7 Port 2 and Port 4

The processor Ports P2 and P4 are dedicated for specific uses.

4.7.1 Port 2

Port 2 is used for interfacing with the on-board serial devices, the serial EEPROM and the serial DAC.

Port	Signal	Device	Comment
2.0	E_WP	Serial EEPROM	Write Protect
2.1	E_CL		Clock
2.2	E_DA		Data
2.3	D_CR	Serial DAC	Clear
2.4	D_CL		Clock
2.5	D_IN		Data In
2.6	D_OT		Data Out
2.7	D_CS		Chip Select / Load

The Linear Technology LT1660 is used for the serial DAC. Refer to the data sheets and example projects for information on how to use the serial DAC.

The serial EEPROM is the industry-standard 24C256. The same spot may be populated with a 24C512, which has a 64KB capacity. Refer to the data sheets and example projects for information on how to use the serial EEPROM.

4.7.2 Port 4

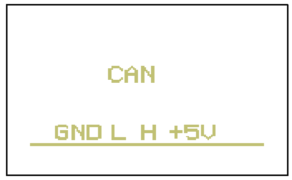
The AT89C51CC03 has only five bits of Port 4 available. These are used for the CAN and SPI communications.

Port	Signal	Communications	Comment
4.0	CTxD	CAN	Transmit
4.1	CRxD		Receive
4.2	MISO	SPI	Master In Slave Out
4.3	CLK		Clock
4.4	MOSI		Master Out Slave In

5 CAN

5.1 CAN Port Connections

J10 is a 4-position screw-type terminal block for the CAN communications port. Pin 1 is marked by a square pad, visible from the bottom side of the board.



J11 could be populated with a locking header if the terminal blocks are not needed. The two connectors (J10 and J11) have the same pin order.

J10 / J11 Pinout

Position	Signal	Remark
1	VCC	+5V
2	CANH	CAN high signal
3	CANL	CAN high signal
4	GND	Ground

5.2 CAN Programming

The Ru03 has a built in CAN controller with 15 message objects. Information about the CAN controller is available from the Atmel data sheet.

The project cc03_can_00 illustrates how to set up the various CAN registers. The CAN terminating resistors are needed for higher bit rates (for example 1 Megabit-per-second)

5.3 CAN Bit Rate Adjustment

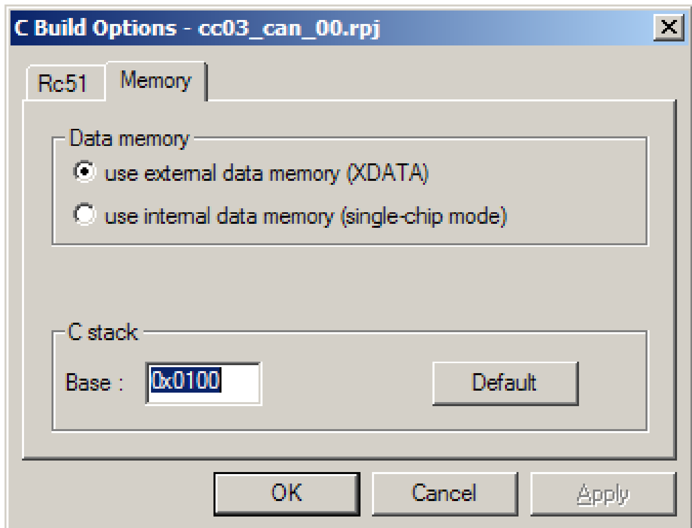
The AT89C51CC03 has an on-chip CAN unit that supports 15 message objects. Refer to the data sheet and the sample project for information about the operation of the CAN unit. The CAN interface chip features a potentiometer, R5. Rotate R5 clockwise to adjust the CAN bit rates. Turning the potentiometer counterclockwise introduces a low-pass filter that interprets fast-changing signals as noise.

6 PROGRAMMING NOTES

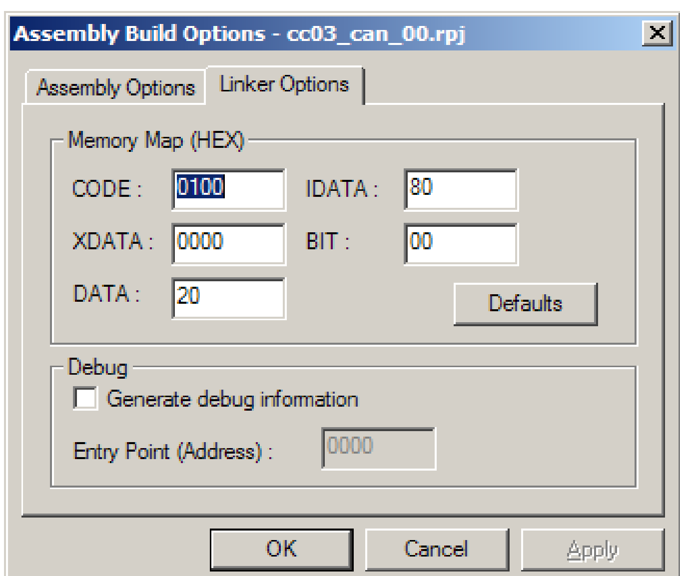
Reads51 may be used to write code to run on the RU30 board. There are several steps that must be followed for the code to run correctly on the board.

6.1 Reads51 Memory Map

The Reads51 memory map must be set correctly for the code to run on the board. Use the Reads51 Project menu to set the Project Build Options. The Compiler Options Memory determines the parameters of the C-language stack. Select XDATA to use the 2K internal RAM. The base of the C stack is set to 0x100 below. This leaves 256 bytes ([0..0xFF]) for global variables. The remainder of the internal RAM is assigned to the stack for function calls, automatic (local) variables, and the expression stack.



Under Assembler Options, select the Linker Options tab to set the segment addresses. In the example below, code starts at 0x100. This leaves the first 256 bytes of memory to the various interrupt vectors. The range from the beginning of XDATA to the beginning of C Stack is reserved for the global variables. Let XDATA start from 0. This allocates 256 bytes for global data.



The example project "Hello" uses these options. Compiler the project to obtain Hello.hex.

6.2 Downloading the Code to the Ru03 Board

The hex file is downloaded to the board using the Atmel FLIP utility. FLIP communicates with the microcontroller and programs the internal FLASH. The FLIP utility may be downloaded from the Atmel website (www.atmel.com). The current version is FLIP 3.2.

6.2.1 Step 1. Connection

The microcontroller board has a serial port with TxD and RxD signals. These are signals may be directed to a level converter and to the DB9 connector. Alternatively, these signals may be directed to the serial-to-USB converter on the board. There is a USB connector, J1, on the board to connect to a host PC. Note that the serial-to-USB converter driver installs the USB device as a Virtual Communications Port (VCP). The Windows Device Manager reports this as another COMM port. The selection of RS232 versus USB is determined by the two jumpers S1 and S2.

If you want to use the USB port, you must first install the serial-to-USB drivers. The converter is made by Future Technology Devices International, who maintains drivers for various platforms on their web site (www.ftdichip.com). Download the VCP drivers in compressed form (currently version 2.02.04). The FTDI web site has extensive installation instructions. Basically, uncompress the drivers in a folder. Then, when your PC detects the new USB hardware, specify the folder where you have the uncompressed drivers.

Connect the Ru03 board to the PC using one of two methods:

a) Using the RS232 port. Place jumpers S1 and S2 in the upper position.

b) Using the USB port. Place jumpers S1 and S2 in the lower position. If you will use the USB port to communicate with the board, you must first install the FTDI serial-to-USB drivers.

6.2.2 Step 2. Power the board

Power the board using one of two methods

a) Using the USB port power. Place the power jumper S5 in the left position.

a) Using an external power supply. Connect the 6-18V AC or DC power supply to the terminal block J6. Place the power jumper S5 in the right position.

Note that the USB LED (on the left) is lit as long as the USB port receives power, even if the power jumper S5 is in the right position.

6.2.3 Step 3. Run FLIP 3.2.

3.1. Select the device (menu Device -> Select) as AT89C51CC03.

3.2. Select connection (menu Settings -> Communication -> RS232). Note that even if you use the USB connection on the board, select RS232. The serial-to-USB converter on the board installs its driver to appear as a Virtual Comm Port (VCP).

3.3. Select the COM port. Note that the serial-to-USB converter appears as a Virtual COM Port (VCP).

3.4. Select the Baud rate. With a 8 MHz crystal, select 19200 or 9600 Baud. With a 24 MHz crystal, you may select a higer Baud rate such as 57600. <u>Do not click the "Connect" button yet.</u>

6.2.4 Step 4. Reset the board

Press and hold the RESET button on the board. While you hold RESET down, press the PSEN button. While the PSEN button is held down, release the RESET button. After a second release the PSEN button. The board is now in the program mode and ready to communicate with FLIP.

The PSEN signal is normally an output, keeping the PSEN button pressed for a long time may damage the microcontroller.

6.2.5 Step 5. Communicate with the board

Click the "Connect" button on the FLIP RS232 dialog. If FLIP establishes communications, its various controlls will be enabled.

6.2.6 Step 6. Download the HEX file

6.1. Load the HEX file (menu File -> Load HEX file...).

6.2. Click the X2 box to run the microcontroller with 6 oscillator cycles per instruction. (with a 8 MHz crystal, the effective clock speed is 16 MHz; with a 24 MHz crystal, the effective clock speed is 48 MHz).

6.3. Use the FLIP program steps on the left to erase, blank check, program, and verify.

6.4. Check the FLIP Reset box.

6.2.7 Step 7. Start the application

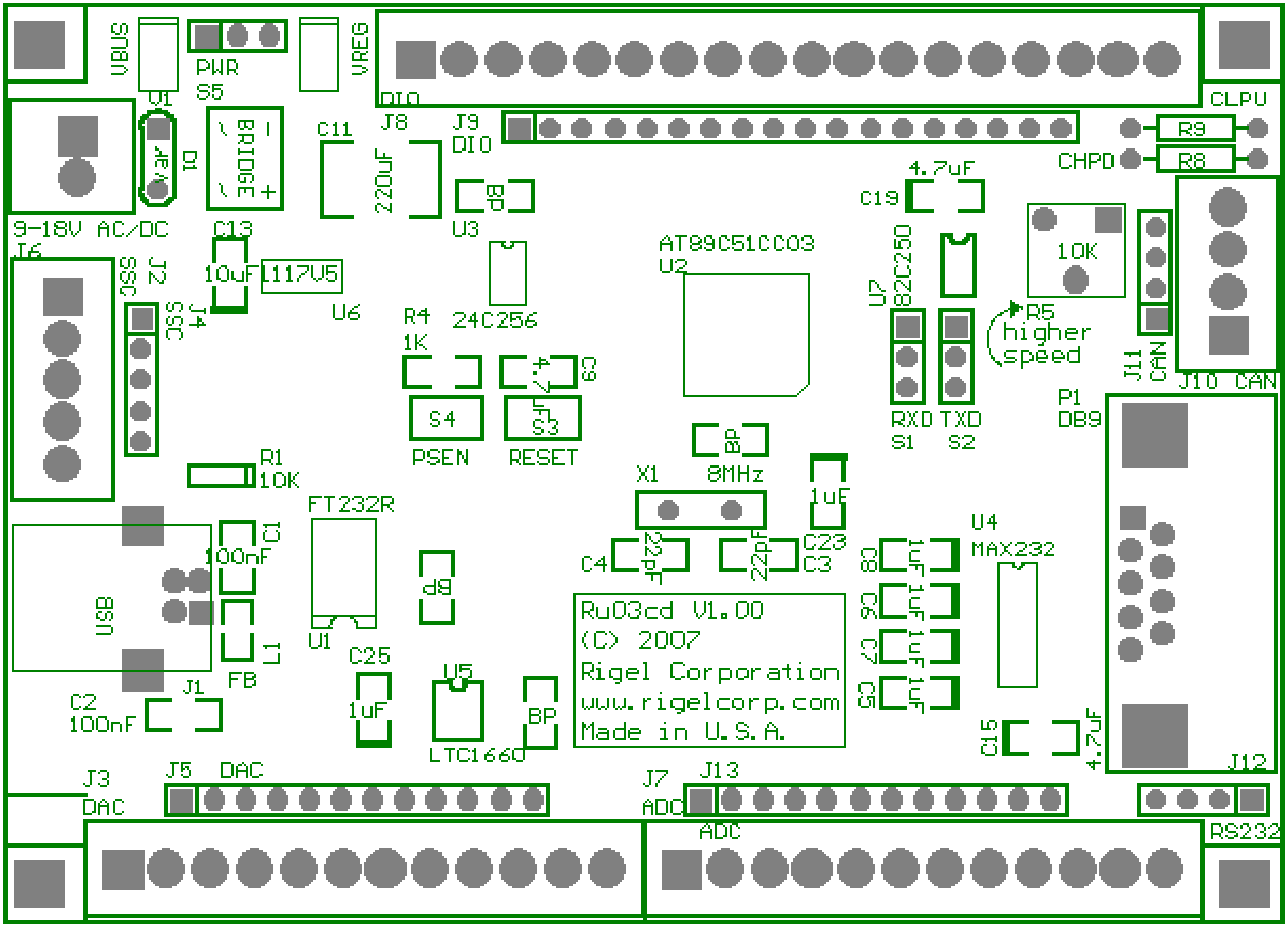
Click the Start Application button (red button) in the FLIP window. This terminates FLIP-to-board communications and runs the downloaded program.

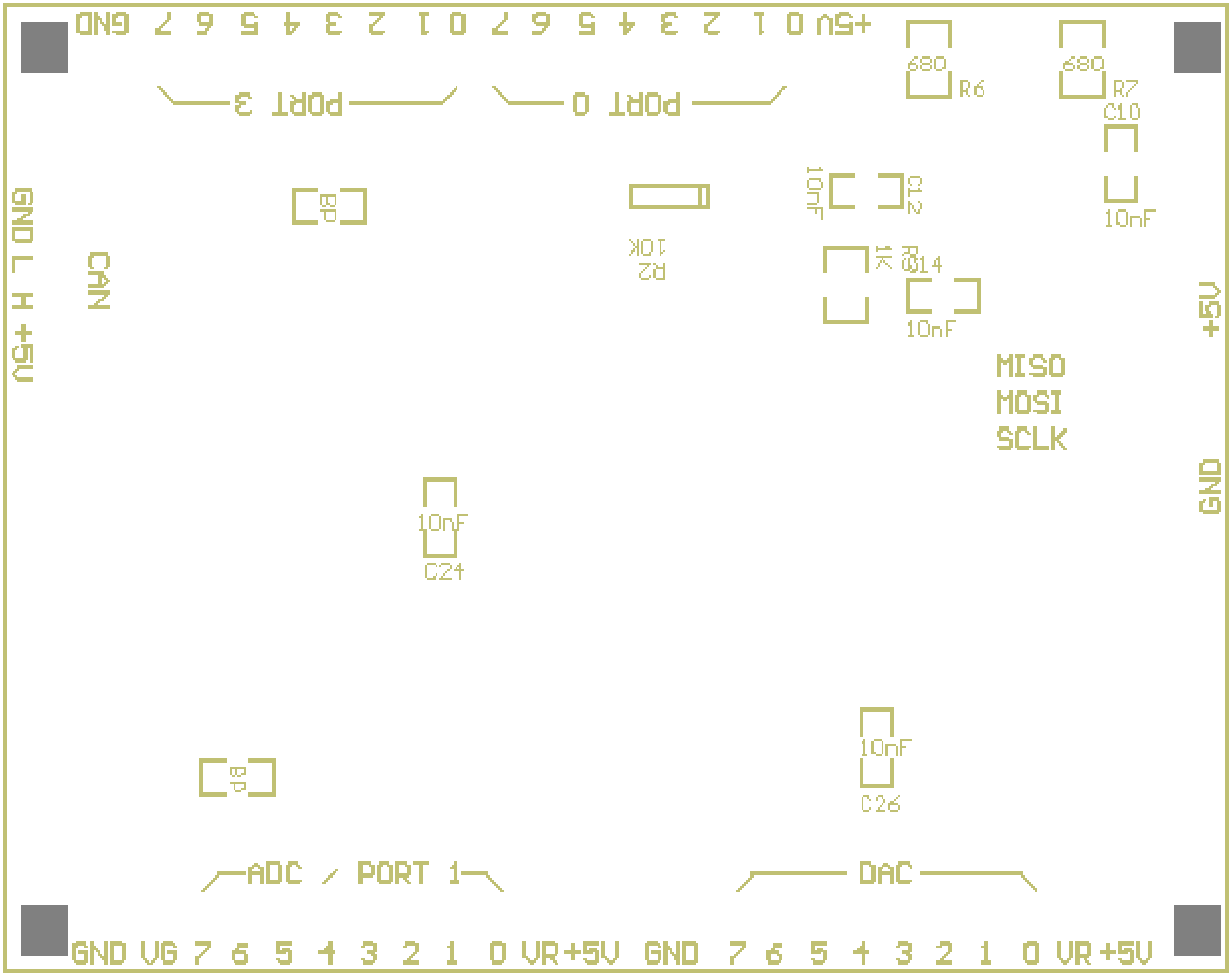
7 OEM OPTIONS

Below are the more popular OEM choices. Contact Rigel Corporation for other custom options.

1. The boards may be factory programmed.
2. The boards may be populated with the specific connectors needed by the application. We use 3.5mm terminal blocks which come in many styles: right-angle, upright, locking, or pluggable. We can populate the board with any of these styles. Instead of terminal blocks the board may be populated with 100mil headers. These come in many styles as well, upright, right angle, shrouded, and locking just to name a few.
3. The DB9 may be left off
4. The power LED's may be left off.
5. The USB port may be left off.
6. The CAN may be left off.
7. Industrial temperature chips may be used.

8 BOARD OVERLAY





GND 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 +5V
PORT 3
PORT 0
680
R6
680
R7
C10
10nF
C12
10nF
BP
10K
R2
1K
R3
C14
10nF
GND L H +5V
CAN
MISO
MOSI
SCLK
+5V
GND
10nF
C24
BP
10nF
C26
ADC / PORT 1
DAC
GND VG 7 6 5 4 3 2 1 0 VR +5V GND 7 6 5 4 3 2 1 0 VR +5V