ZKit-51, 8051 Development Kit

User Manual

1.0, June 2009

Table of Contents

1. Introduction 1
    1. Board Features 1
2. Board Design 2
    1. Overview 2
    2. Locating Components 2
    3. Power Supply 4
    4. CPU 5
    5. USB Serial 6
    6. LCD Display 7
    7. SPI EEPROM 7
    8. I2C RTC 8
    9. Piezoelectric Buzzer 8
    10. Debug LEDs 9
    11. Keypad 9
    12. Interrupt Key 10
3. Software Tools 11
    1. SDCC 11
    2. Flash Magic 11
    3. Smash 11
    4. FT232R Drivers 11
4. Development Setup 12
    1. Hardware Setup 12
    2. Software Setup in GNU/Linux 12
    3. Software Setup in MS Windows 13
5. Hello Embedded World! 15
    1. Under MS Windows 15
    2. Under GNU/Linux 16
6. Jumpers and Switches 18
    1. EXTPWR/USBPWR Jumper 18
    2. INTR/BUZZ Dip Switch 18
    3. USBSIO Dip Switch 18
7. External Connectors 19
    1. SIO/I2C Header 19
    2. PORT-P1/P3 Header 19
    3. PORT-P0/P4 Header 19
    4. Phoenix Terminal - EXTIO 20
    5. Phoenix Terminal - PWR 20
8. Add-On Boards 21
A. Determining COM Port 22

Chapter 1. Introduction

1. Board Features

The ZKit-51 offers the following features

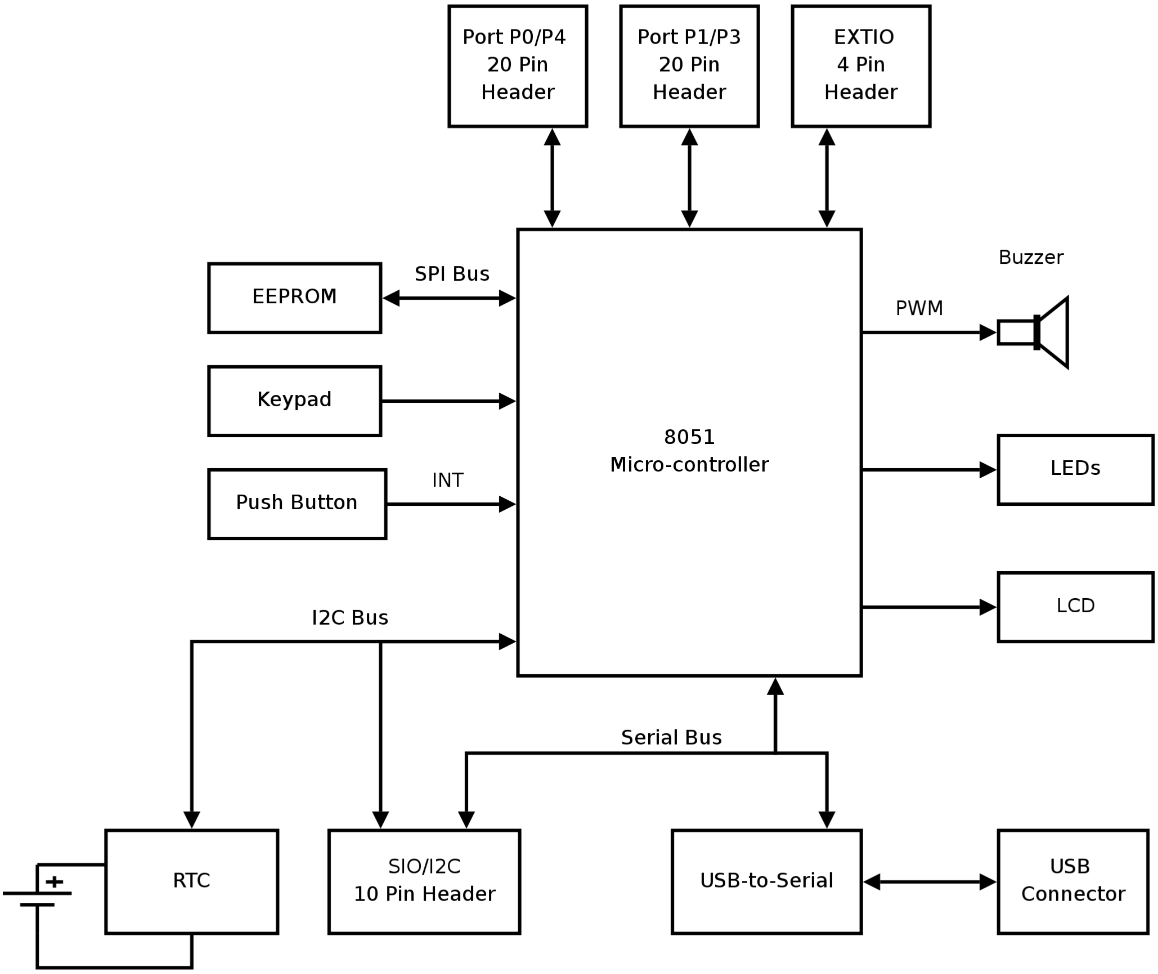
- NXP P89V664 micro-controller with 64KB Flash and 1KB RAM
- 18.432MHz crystal
- Power supply, jumper selectable between
  - USB
  - External 5V supply
- On-board Peripherals
  - 16x2 character LCD, with backlight
  - USB serial interface, for communication and program download
  - 2Kbit SPI EEPROM
  - I2C RTC with battery backup
  - Piezoelectric buzzer
  - Four button keypad
  - Push button with hardware de-bounce (interrupt input)
  - 2 debug LEDs
- Connectors
  - USB, type B connector
  - 2.1mm power supply connector
  - 20 pin header for P0/P4
  - 20 pin header for P1/P3
  - 10 pin header for serial communication / I2C
  - 2 pin header for powering external devices
  - 4 pin header for external I/O

Chapter 2. Board Design

1. Overview

A bird's eye view of the devices available on the board, is shown in the following block diagram. Each device connectivity is described in detail in the following sections.

Figure 2.1. Block Diagram



2. Locating Components

The location of the components on the board are indicated in the following diagrams.

Figure 2.2. Front View

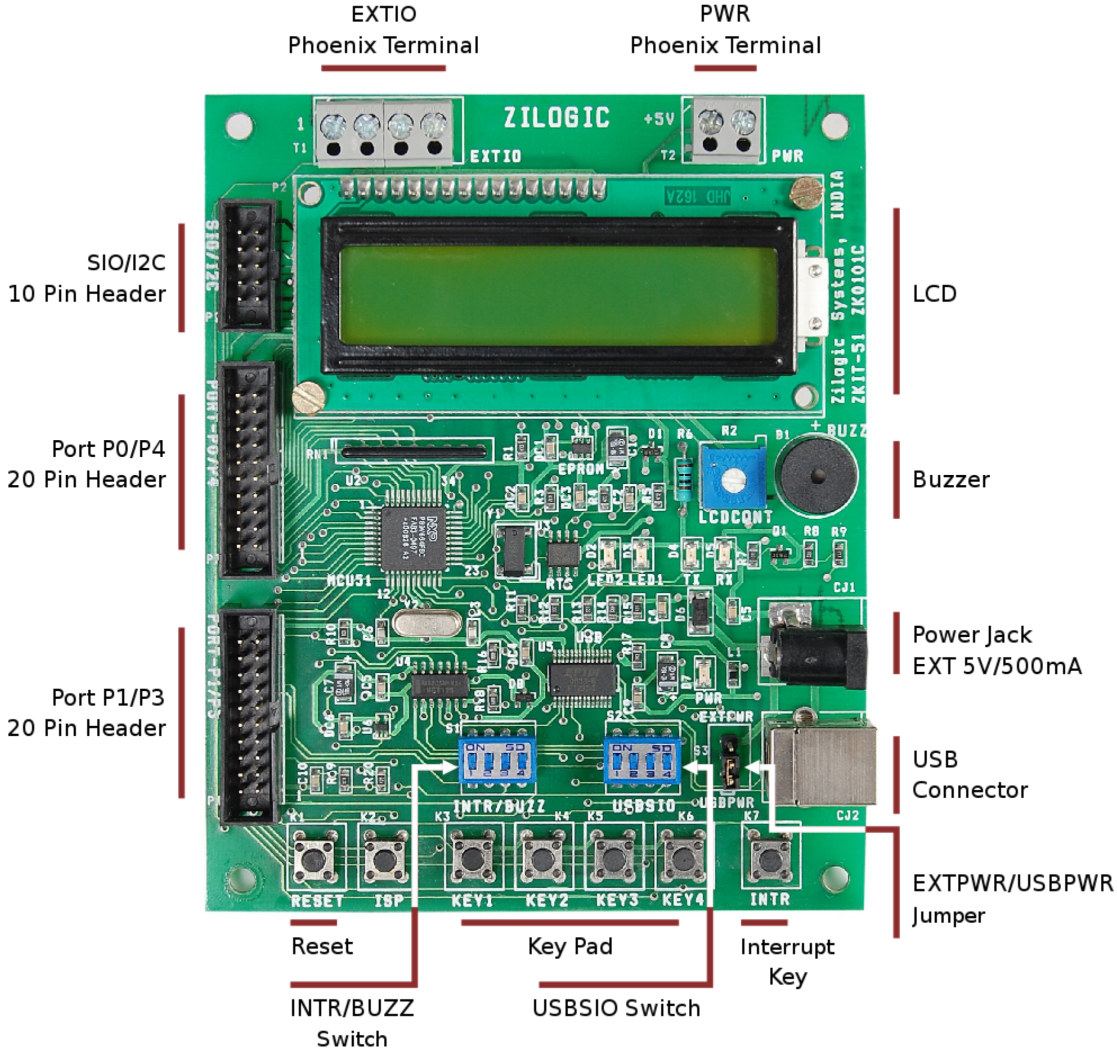
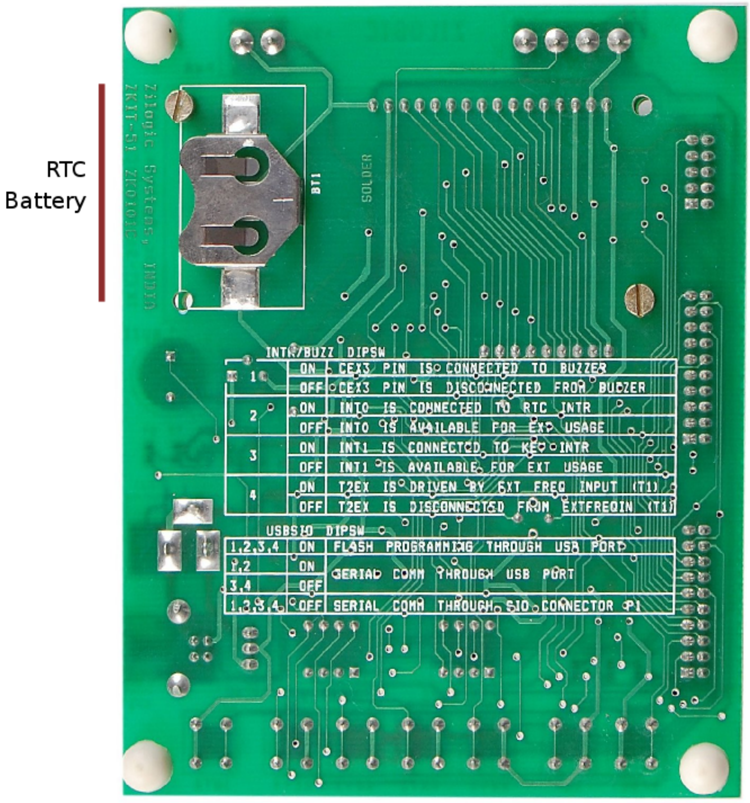


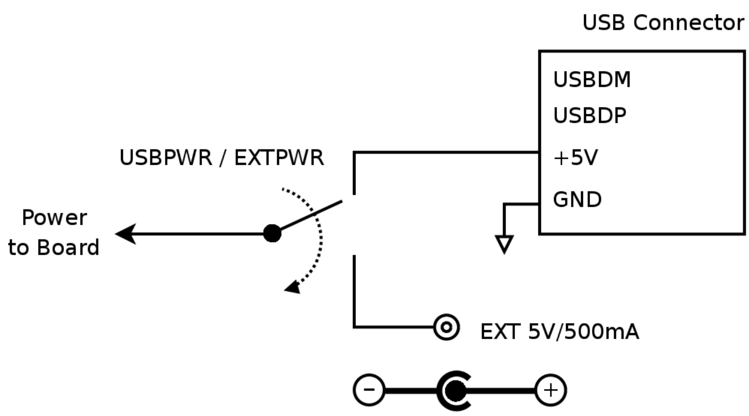
Figure 2.3. Back View



3. Power Supply

The ZKit-51 can be powered through USB or an external 5V regulated power supply. The power source can be selected through USBPWR / EXTPWR jumper setting.

Figure 2.4. Power Supply Connection Diagram



> **Caution**
>
> The external power supply, if used, should be a 5V/500mA regulated power supply, with the polarity shown in the power supply connection diagram.

4. CPU

The heart of the ZKit-51 is Philips P89v664 micro-controller. The P89v664 is an 8-bit 80C51 5V low power micro-controller with 64 kB Flash, 2KB of data RAM and supports In-System Programming (ISP).

The main features of the micro-controller are listed below.

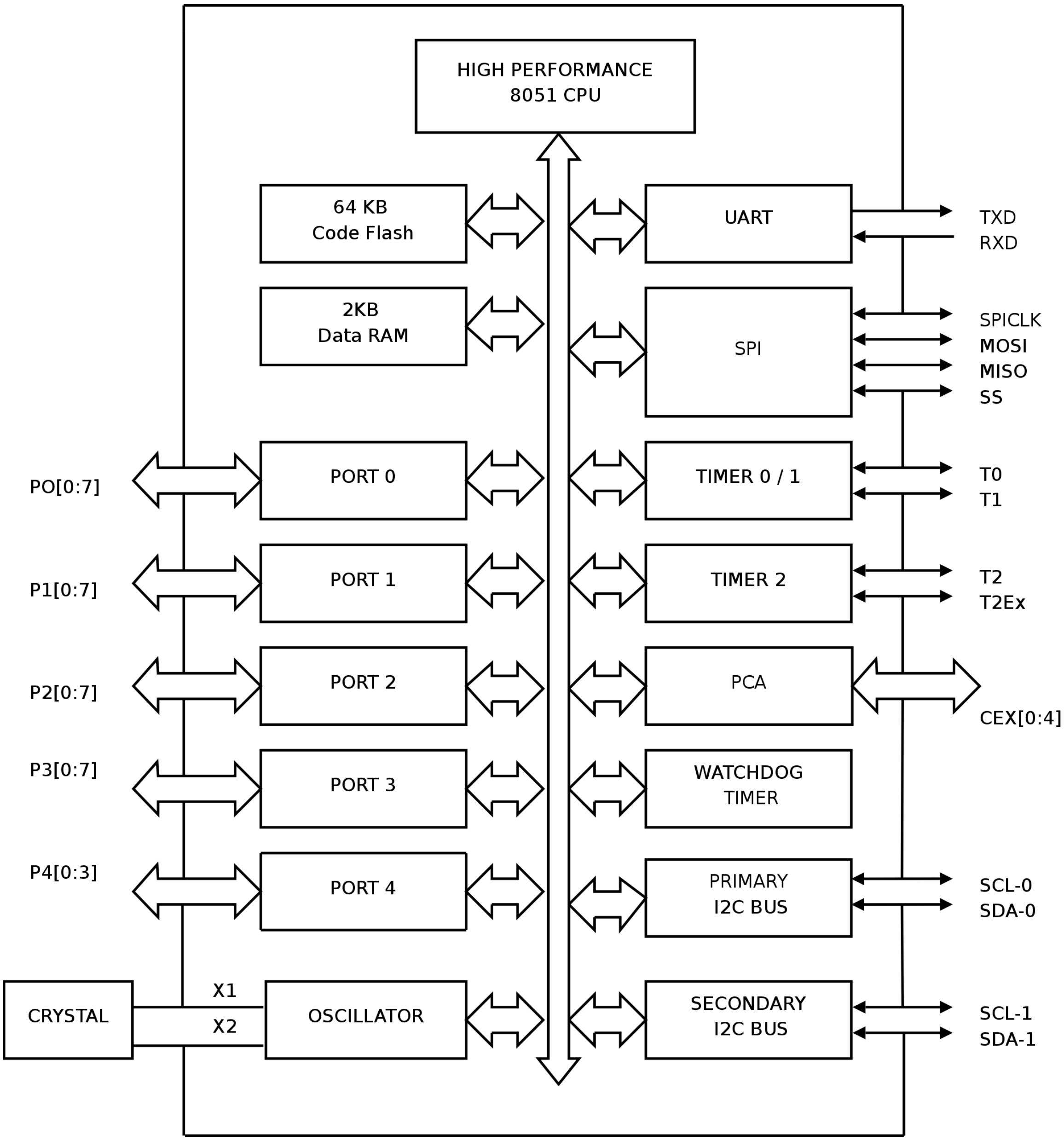
- Dual 100 kHz byte-wide I2C-bus interfaces
- 0 MHz to 40 MHz operating frequency in 12x mode, 20 MHz in 6x mode
- 64 kB of on-chip flash user code memory with ISP and IAP
- 2 kB RAM
- SPI (Serial Peripheral Interface) and enhanced UART
- PCA (Programmable Counter Array) with PWM and Capture/Compare functions
- Three 16-bit timers/counters
- Four 8-bit I/O ports, one 4-bit I/O port
- WatchDog Timer (WDT)
- Support for 12-clock (default) or 6-clock mode selection via ISP
- Low EMI mode (ALE inhibit)
- Power-down mode with external interrupt wake-up

The micro-controller crystal frequency is 18.432 MHz. 8051-based processors generate their serial port timing using a combination of external crystal and internal programmable divider chains. This crystal frequency has been selected in order to ensure the following

1. the timing requirements of the controller's serial interface are met.
2. the CPU runs at high speed in 6-clock mode.

Power to the board is sourced either from the +5V external regulated power supply or the via USB power with the help of jumper selection.

Figure 2.5. Micro-controller Block Diagram



5. USB Serial

The ZKit-51 has a FT232R USB to serial UART converter. The FT232R has the following advanced features:

- The FT232R is fully compliant with the USB 2.0 specification.
- Single chip USB to asynchronous serial data transfer interface.
- Entire USB protocol handled on the chip. No USB specific firmware programming required.
- Supports transmit and receive LED drive signals.

The ZKit-51 uses USB Serial UART for serial communication between PC and the P89V664 MCU. This is also used to download firmware by activatiing the bootloader of P89V664 MCU. This is called In-System Programming (ISP).

5.1. In-System Programming

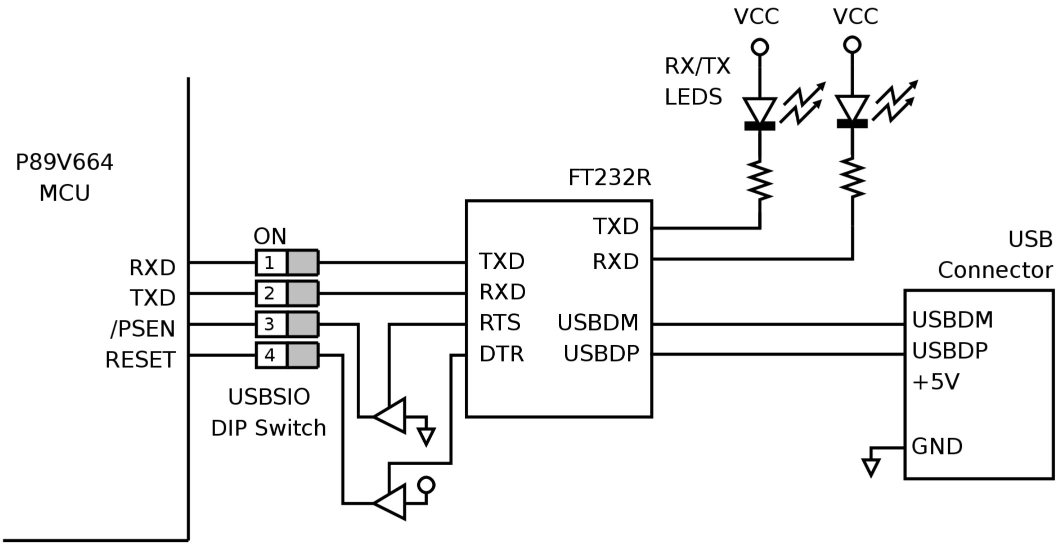
To switch the P89V664 MCU to ISP mode, the following sequence has to be followed.

1. Apply hardware reset.
2. Pull PSEN to ground.
3. Release hardware reset.
4. Release PSEN.

The ZKit-51 comes with RESET and PSEN push buttons that can be used to put the MCU into ISP. In the ZKit-51, the RESET and PSEN pins are also connected to the handshaking signals DTR and RTS of the USB to serial UART. Flash programming applications like Smash and Flash Magic can utilize this feature to switch the device into ISP mode automatically, without user intervention.

The following diagram shows the FT232R connection details.

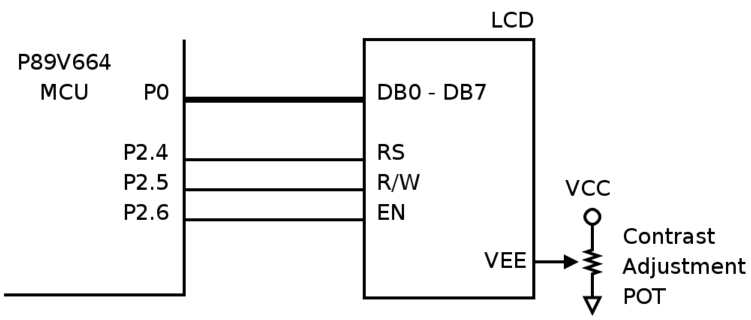
Figure 2.6. FT232R Connection Diagram



6. LCD Display

The ZKit-51 has a HD44780 Hitachi chipset compatible, 16x2 character, LCD. The LCD data lines are connected to Port 0 and the control lines (RS, R/W, EN) are connected to P2.4, P2.5, P2.6 respectively. The following diagram shows the LCD pin connection details.

Figure 2.7. LCD Connection Diagram

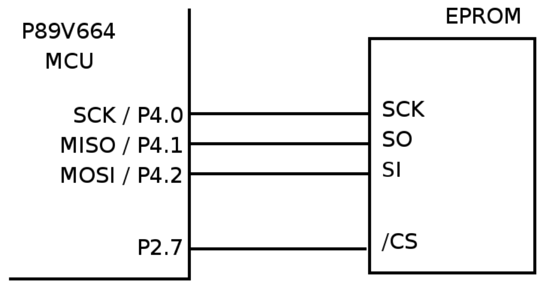


7. SPI EEPROM

The ZKit-51 has a Microchip 255AA020A EEPROM for data storage. The Microchip 25AA020A is a 2 Kbit Serial EEPROM. The memory is accessed via a simple Serial Peripheral Interface (SPI) compatible serial bus. The bus signals required are a clock input (SCK) plus separate data in (SI) and data out (SO) lines. Access to the device is controlled through a Chip Select (CS) input.

The SPI EEPROM is connected to the on-chip SPI controller of the P89V664 MCU. The following diagram shows the EEPROM pin connection details.

Figure 2.8. SPI EEPROM Connection Diagram

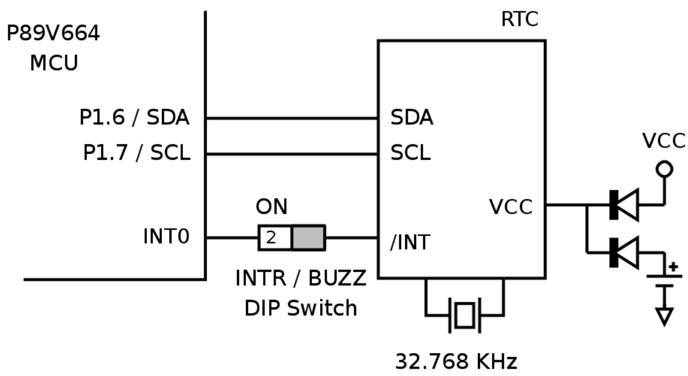


8. I2C RTC

The ZKit-51 has an battery backed NXP PCF8563 RTC to keep track of current date and time. The PCF8563 is a CMOS real-time clock/calendar optimized for low power consumption. A programmable interrupt output and voltage-low detector are also provided. All addresses and data are transferred serially via a two-line bidirectional I2C-bus. The maximum bus speed is 400 kbit/s.

The I2C RTC is connected to the primary on-chip I2C controller of the P89V664 MCU. The RTC interrupt is connected to `INT0` through an On/Off dip switch. When the switch is in Off position, `INT0` is available for external usage, through the `PORT P1/P3` header. The following diagram shows the RTC pin connection details.

Figure 2.9. RTC Connection Diagram

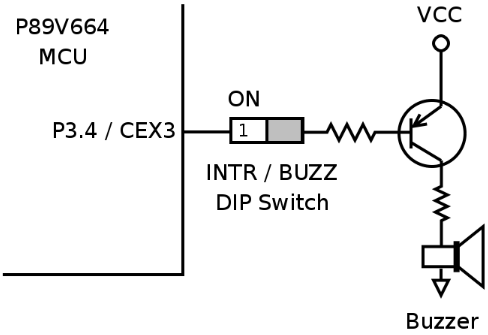


9. Piezoelectric Buzzer

The ZKit-51 has a piezoeletric buzzer that can be used to provide audible indications. The buzzer is connected to `P3.4` pin. By rapidly switching the pin, a tone can be generated on the buzzer. Alternatively, the on-chip PCA (Programmable Counter Array) can be used to generate a pulse train to the buzzer. Using PWM techniques, both volume and tone can be controlled.

The buzzer is connected to `P3.4` pin, through a On/Off dip switch. When the switch is in Off position, `P3.4` is available for external usage, through `PORT P1/P3` header. The following diagram shows the buzzer connection details.

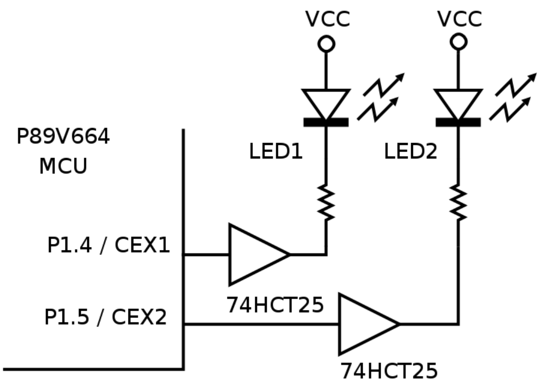
Figure 2.10. Buzzer Connection Diagram



10. Debug LEDs

The ZKit-51 has two debug LEDs connected to `P1.4` and `P1.5`, through a non-inverting buffer. By driving `P1.4` and `P1.5` low, the LEDs can be switched On. Alternatively, the on-chip PCA (Programmable Counter Array) can be used to generate a PWM signal to control the LED brightness.

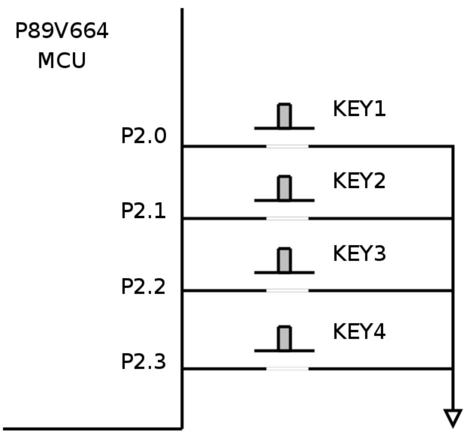
Figure 2.11. LEDs Connection Diagram



11. Keypad

The ZKit-51 has 4 tactile push button switches connected to `P2.0` to `P2.3`. The keypad connection details are shown in the following diagram.

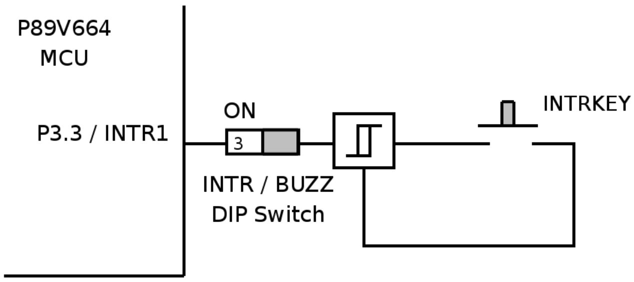
Figure 2.12. Keypad Connection Diagram



12. Interrupt Key

The ZKit-51 has 1 tactile push button switch for testing interrupts. The push button is hardware debounced and connected to `INTR1`, through a On/Off dip switch. When the switch is Off, `INTR1` is available for external usage, through `PORT P1/P3` header. The following diagram shows the interrupt key connection details.

Figure 2.13. Interrupt Key Connection Diagram



Chapter 3. Software Tools

1. SDCC

SDCC (Small Device C Compiler) is a Free and Open Source, re-targetable, optimising ANSI-C compiler by Sandeep Dutta designed for 8 bit Microprocessors. The current version supports 8051 and its derivatives. For the complete list of supported micro-controllers, visit the SDCC website — http://sdcc.sourceforge.net.

2. Flash Magic

From the Flash Magic website, Flash Magic is a PC tool for programming flash based microcontrollers from NXP using a serial or Ethernet protocol while in the target hardware. For more details visit the Flash Magic website — http://www.flashmagictool.com.

3. Smash

Smash is a 8051 micro-controller flashing tool. It has been developed by Zilogic Systems, and released under the GNU GPL, to enable programming the ZKit-51 from GNU/Linux. Smash is written in Python and uses GTK+ for the GUI interface.

4. FT232R Drivers

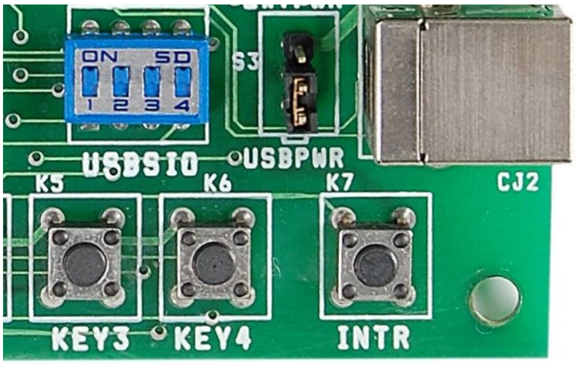
The host system talks to the board throug the USB-Serial interface provided by FT232R. Hence, the host system should have the FT232R driver. The drivers for Linux is available in the mainline kernel since version 2.6.9. The drivers for Windows are available from the vendor and is included in the CD-ROM.

Chapter 4. Development Setup

1. Hardware Setup

1. Connect the ZKit-51 development board to a PC using the USB cable.
2. Select USB power supply by setting `USBPWR / EXTPWR` jumper, in `USBPWR` position.
3. Ensure all the dip switches in `USBSIO` are in On position.

Figure 4.1. Jumper and Switch Settings for Development



2. Software Setup in GNU/Linux

2.1. SDCC Setup

SDCC is available as a package in Fedora, Debian GNU/Linux and Ubuntu distributions. The package name is `sdcc`. Make sure your package manager is configured correctly and install the package through your package manager.

To verify the install, execute the following command.

```
$ sdcc --version
```

This should print `sdcc`'s version number.

2.2. Smash Setup

Step 1. Ensure the following packages are available before installing `smash`.

- `python`, version 2.4 or greater
- `python-serial`, version 2.2 or greater (called `pyserial` in Fedora)
- `pygtk`, version 2.8 or greater
- `python-dbus`, version 0.71 or greater

> **Tip**
>
> It is possible to run `smash` without the GUI interface, in which case `pygtk` and `python-dbus` are not required.

Step 2. Insert the ZKit-51 CD-ROM. Copy the compressed archive `software/smash-1.7.0.tar.gz` to your home directory and extract it. This can be done using the following sequence of commands.

```
$ cp /media/cdrom/software/smash-1.7.0.tar.gz ~/
$ cd ~/
$ tar -xvzf smash-1.7.0.tar.gz
```

Step 3. Execute the `setup.py` Python script present in the extracted archive, to install the package.

```
# python setup.py install
```

Tip

Commands that are shown with the # prompt generally require root privileges to be executed. Ubuntu users will have to prefix the command with `sudo`. Other distribution users can get access to a root shell using the `su` command.

Step 4. To verify the install, execute the following command.

```
$ smash --version
```

This should print `smash`' version number.

2.3. FT232R Driver Setup

The driver is included in Linux kernels greater than version 2.6.9, and hence no setup is required.

3. Software Setup in MS Windows

3.1. SDCC Setup

Step 1. Insert the ZKit-51 CD-ROM.

Step 2. Execute the file `software/sdcc-2.9.0-setup.exe` present in the CD-ROM to install SDCC.

Step 3. To verify the install, open a DOS box and enter the following command.

```
> sdcc -v
```

This should print `sdcc`'s version number.

3.2. Flash Magic Setup

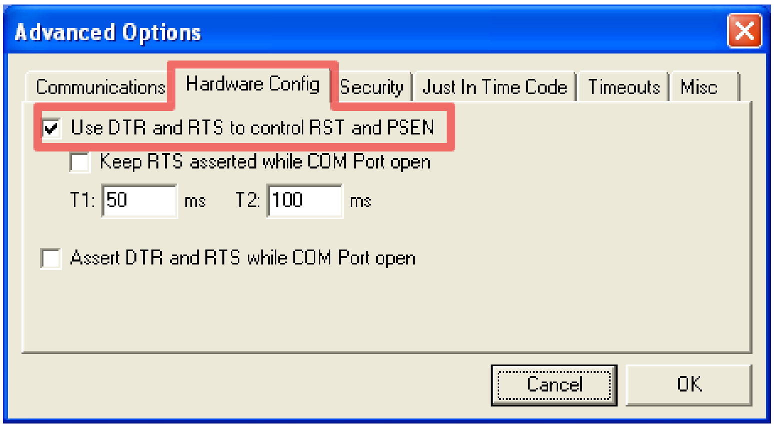
Step 1. Visit FlashMagic website http://www.flashmagictool.com/ and download the file `FlashMagic.exe`.

Step 2. Execute the downloaded file `FlashMagic.exe`, and follow the instructions.

Step 3. Start Flash Magic by selecting it from the Start Menu. In the Flash Magic windows select `Options > Advanced Options` … menu item. In the window that appears enable the check box that says `Use DTR and RTS to control RST and P0.14`, and click on Ok.

When this option is enabled, during code downlaod, the flashing tool will automatically switch the device into ISP mode. For more information on this, see Section 5, “USB Serial”.

Figure 4.2. Flash Magic Advanced Options



3.3. FT232R Driver Setup

Step 1. Insert the ZKit-51 CD-ROM.

Step 2. Execute the file `software/CDM 2.04.16.exe`, and following the instructions.

Chapter 5. Hello Embedded World!

This chapter shows how to build a C program using SDCC and how to download it onto the board using a flashing tool.

1. Under MS Windows

Step 1. Type in the following program in a text editor like notepad and save it as `blink-led.c`.

```
#include <8051.h>

#define LED P1_4

void delay(char count)
{
        int i;
        while (count--)
                for (i = 0; i < 30000; i++);
}

void main()
{
        LED = 0x0;
        while (1) {
                delay(5);
                LED = !LED;
        }
}
```

Step 2. Open a DOS box, switch to the folder containing the C file and compile it using the following command.

```
> sdcc blink-led.c
```

Step 3. Fix the errors, if any and check the folder for executable file with the extension `ihx`.

Step 4. Connect the ZKit-51 to the PC using the USB cable. Open Flash Magic, select the following parameters in the Flash Magic window, under Communications section.

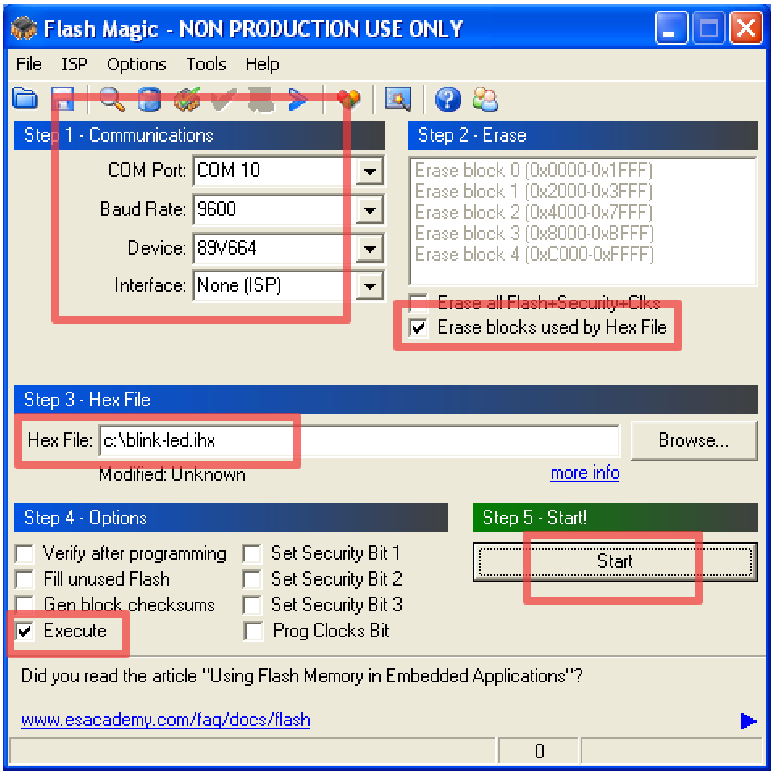
Device	P89V664
COM Port	To determine the COM port see Appendix A, *Determining COM Port*.
Baud Rate	9600
Interface	ISP
Oscillator	18.432

Step 5. Select the `Erase block used by Hex File` check box, under the Erase section.

Step 6. Select the hex file generated in Step 3, under the `Hex File` section.

Step 7. Click on the start button to download the hex file, on to the micro-controller.

Figure 5.1. Download using Flash Magic



2. Under GNU/Linux

Step 1. Type in the following program in text editor like gedit or kedit and save it as `blink-led.c`

```
#include <8051.h>

#define LED P1_4

void delay(char count)
{
        int i;
        while (count--)
                for (i = 0; i < 30000; i++);
}

void main()
{
        LED = 0x0;
        while (1) {
                delay(5);
                LED = !LED;
        }
}
```

Step 2. Open a terminal, switch to the folder containing the C file and compile it using the following command.

```
$ sdcc blink-led.c
```

Step 3. Fix the errors, if any and check the folder for executable file with the extension `ihx`.

Step 4. Connect the ZKit-51 to the PC using the USB cable. Start `smash` as show below.

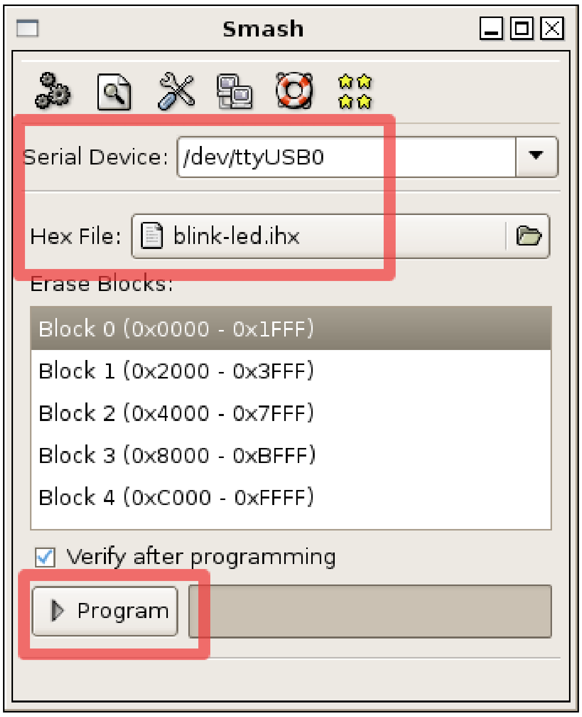
```
$ smash -g
```

Step 5. In the `smash` window, select the serial device from the drop down box. It is usually `/dev/ttyUSB0` or `/dev/ttyUSB1`.

Step 6. Select the hex file generated in Step 3, under the Hex File section.

Step 7. Click on the Program button to download the hex file, on to the micro-controller.

Figure 5.2. Download using Smash



Chapter 6. Jumpers and Switches

1. EXTPWR/USBPWR Jumper

The board can be powered from USB or an external power source. To select the required power source, the jumper has to set, as specified below.

USB power	To power the board through USB, the jumper should be on the USBPWR pin and the center pin.
External power	For external 5V power source, the jumper should be on the EXTPWR pin and the center pin.

Table 6.1. EXTPWR/USBPWR Jumper Pins

Pin	Description
USBPWR	USB Power +5V
Center	To board Vcc
EXTPWR	External Power +5V

2. INTR/BUZZ Dip Switch

The INTR/BUZZ dip switch is used to select the routing between external connector and on board devices, to the micro-controller. The routing details are given in the following table.

Switch	State	Description
1	ON	CEX3/P3.4 pin is connected to buzzer
	OFF	CEX3/P3.4 pin is disconnected from buzzer
2	ON	INT0 is connected to RTC interrupt
	OFF	INT0 is available for external usage
3	ON	INT1 is connected to INTR key
	OFF	INT1 is available for external usage
4	ON	T2EX is driven by EXTIO header
	OFF	T2EX is disconnected from EXTIO header

3. USBSIO Dip Switch

The serial interface of the micro-controller can be used for

1. Flash programming through USB
2. Serial communication to PC through USB
3. Serial communication to devices through SIO

To select the required functionality the USBSIO dip switch has to configured, as specified below.

Flash Programming	For flash programming through USB port, switches 1, 2, 3 and 4 should be in ON state.
USB Serial Communication	For serial communication through USB port, switches 1, 2 should be in ON state and 3, 4 should be in OFF state.
SIO Serial Communication	For serial communication through the SIO connector, switches 1, 2, 3 and 4 should be in OFF state.

Chapter 7. External Connectors

1. SIO/I2C Header

The SIO/I2C header is terminated with serial communication signals, I2C signals and power supply. Add-on boards, with different functionalities, can be connected through this header, to the ZKit-51.

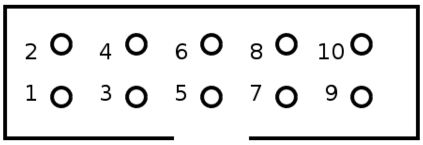


Table 7.1. SIO/I2C Header

Pin #	Signal	Pin #	Signal
1	VCC	2	P3.0/RXD
3	P3.1/TXD	4	P1.6/SCL
5	P1.7/SDA	6	P1.0/T2
7	P1.1/T2EX	8	P1.3/CEX0
9	P3.2/INTR0	10	GND

2. PORT-P1/P3 Header

The PORT-P1/P3 header is terminated with port P1 and P3 signals, along with power supply. Add-on boards, with different functionalities, can be connected through this header, to the ZKit-51.

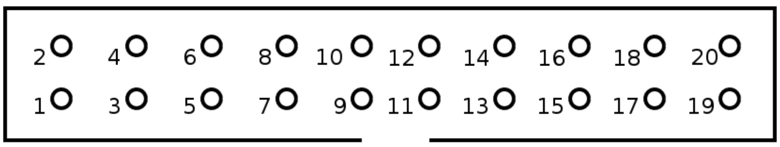


Table 7.2. PORT-P1/P3 Header

Pin #	Signal	Pin #	Signal
1	VCC	2	P1.0/T2
3	P1.1/T2EX	4	ECI/P1.2
5	CEX0/P1.3	6	CEX1/P1.4
7	CEX2/P1.5	8	SCL/P1.6
9	SDA/P1.7	10	RXD/P3.0
11	TXD/P3.1	12	INT0/P3.2
13	INT1/P3.3	14	CEX3/T0/P3.4
15	CEX4/T1/P3.5	16	WR/P3.6
17	RD/P3.7	18	PSEN
19	RESET	20	GND

3. PORT-P0/P4 Header

The PORT-P0/P4 header is terminated with port P0 signals, port P4 signals, I2C signals, external memory interface signals and power supply. Add-on boards, with different functionalities, can be connected through this header, to the ZKit-51.

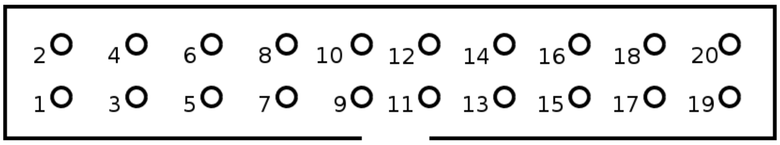


Table 7.3. PORT-P0/P4 Header

Pin #	Signal	Pin #	Signal
1	VCC	2	P0.0/AD0
3	P0.1/AD1	4	P0.2/AD2
5	P0.3/AD3	6	P0.4/AD4
7	P0.5/AD5	8	P0.6/AD6
9	P0.7/AD7	10	P4.0/SCK/SCL
11	P4.1/MISO/SDA	12	P4.2/MOSI
13	P4.3/SS	14	SCL
15	SDA	16	WR/P3.6
17	RD/P3.7	18	ALE
19	INT1/P3.3	20	GND

4. Phoenix Terminal - EXTIO

The Phoenix Terminal EXTIO is terminated with signals useful for external event couting and frequency measurement.

Table 7.4. Phoenix Terminal - EXTIO

Pin #	Signal
1	VCC
2	EXINTR (Active High)
3	EX-FREQ-IN
4	GND

5. Phoenix Terminal - PWR

The Phoenix Terminal PWR is a power for logic probes, used for debugging.

Table 7.5. Phoenix Terminal - PWR

Pin #	Signal
1	VCC
2	GND

Chapter 8. Add-On Boards

The following add-ons to the ZKit-51 are available from Zilogic Systems.

Analog Board	The Analog Board provides D-to-A and A-to-D capabilities to the main board. The board is based on NXP PCF8591 8-bit A/D and D/A converter, with four analog inputs, one analog output, and serial I2C interface.
RTC Display Board	The RTC Display Board provides a 6-digit, seven segment, LED display. The board can be used for real time clock and timer applications.
Keypad	The Keypad provides extended input capabilities to the ZKit-51 with a 16-key 4x4 row matrix.
RS232 Adaptor	The RS232 Adaptor provides RS232 level conversion, and allows ZKit-51 to connect directly to RS232 port of other devices.

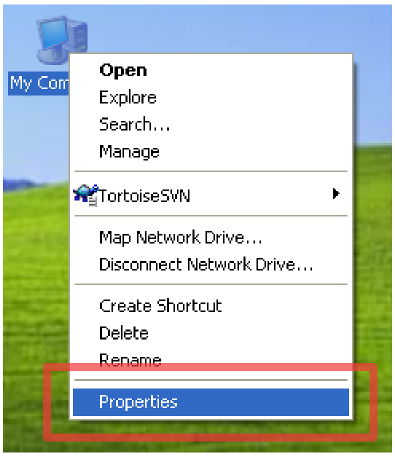
Upcoming add-ons to the ZKit-51 are listed below.

CAN Board	The CAN Board provides CAN bus interface to the ZKit-51. The board will be based on NXP's CAN controller and PHY.
Zigbee Board	The Zigbee Board provides Zigbee wireless interface to the ZKit-51. The board will be based on TI's Zigbee network processor.
Stepper Motor Board	The Stepper Motor Board provides stepper motor control interface to the ZKit-51.

Appendix A. Determining COM Port

The board uses a USB serial interface to communicate with the PC. When the USB port of the board is connected to the PC, MS Windows assigns a COM port to the board. To determine the assigned COM port, follow the instructions given below.

Step 1. Right click on the `My Computer` icon, to get the drop down menu. Select the `Properties` from the menu.



Step 2. In the `System Properties` window that appears, select the `Hardware` tab. In the tab, click on the `Device Manager` button.

Step 3. In the `Device Manager` window, a tree of devices present in the system is shown. Expand the `Ports` node in the device tree. Look for an entry called `USB Serial Port`. The assigned COM port is specified in parenthesis, as shown in the figure below. If this does not appear make sure you have installed the driver correctly as mentioned in Section 3.3, "FT232R Driver Setup"