

MINI
DEVELOPMENT
BOARD

ATMEL
AT89S52

USERS GUIDE

make your own
intelligent **embedded** world ...

rhudoLABZ™

TABLE OF CONTENTS

1.	Overview	2
2.	Hardware Introduction	4
2.1.	The ATMEL AT89S51/52 Function Module.....	4
2.2.	Interface Overview	5
2.3.	Peripheral List.....	6
2.4.	Jumper List	7
2.5.	Microcontroller - Pin Out	8
2.6.	Port Pins	9
2.7.	Power	10
2.8.	Clock source	10
2.9.	Led interfacing	11
2.10.	Liquid crystal display	11
2.11.	Universal asynchronous receiver transmitter (UART).....	12
2.12.	Pull up key board	19
2.13.	External EEPROM Interfacing	20
2.14.	Buzzer Interfacing	21
2.15.	Servo Motor Interfacing	22
2.16.	RFID Interfacing	23
2.17.	Zigbee Interfacing	24
3.	Software Development	25
3.1.	Familiarization Of Keil Professional Development Suite	25
3.1.1.	Creating a Project	27
3.1.2.	Creating Project File	27
3.1.3.	Select Microcontroller from Device Database	28
3.1.4.	Copy and Add the CPU Startup Code	29
3.1.5.	Create New Source Files	29
3.1.6.	Add Source Files to Project	31
3.1.7.	Set Tool Options for Target	32
3.1.8.	Creating HEX File	33
3.1.9.	Build Project	33
3.2.	Connecting the ATMEL AT89S51/52 Development Board	35
3.3.	Programming Steps	36
4.	I/O Distribution	37
4.1.	The Pin Distribution Of ATMEL AT89S51/52 Development Board	37



OVERVIEW

The AT89S52 Development Board Mini from rhydoLABZ is a general purpose, low cost development board for Intel 8051-compatible family of 8 bit microcontrollers (**AT89xxx** and **P89Vxx** manufactured by ATMEL and PHILIPS respectively). It is the new addition to the family of AT89S52 Development kits, featuring a spec level that ranges between the Main Development Boards and the Quick Start Boards from rhydolabz. This mini development board is perfect to start out with 8051 programming and also for designing and testing, simple to highly advanced, 8051 projects.

1.1. FEATURES

1. Compact and ready to use design
2. Two layer High Quality PTH PCB
3. Compatible with AT89xxx / P89Vxx controllers
4. Includes AT89S52 microcontroller
5. ICSP connector ports (5X2 IDC)
6. Resistant to voltage variation (7V to 12V DC)
7. Power supply reverse polarity protection
8. 3 power supply* options (USB/RMC connector/Wall Wart)
9. 5V (1A) and 3.3V (800mA) regulated power output on board
10. All port pins available at male berg strip connector pins
11. Provision to attach mini breadboard on the board
12. Port pins connected* with switches (3x), LEDs (3x) , buzzer
13. On board pull-up for PORT0 pins
14. On board 11.0592 MHz crystal oscillator
15. On board reset switch and power indicator LED
16. UART Interface (RS232/USB/3V3/5V)*
17. I2C interfaced external EEPROM (24C04)
18. On board peripheral support (LCD/ZigBee/RFID/Servo)

Note: *jumper selectable



1.2. ATMEL AT89S5x PACKAGE INCLUDES

- Fully Assembled and Tested AT89S52 Development board
 - Software CDROM with
 - Schematic
 - Programming Software
 - Sample Hex Code
 - Example Codes for

• Led Blinking	LCD Display
• Pull-Up Keyboard	I2C Protocol
• Buzzer Interfacing	External Interrupt Handling
• Led Control with Timer	UART Transmission and Reception

1.3. At8952 SPECIFICATION

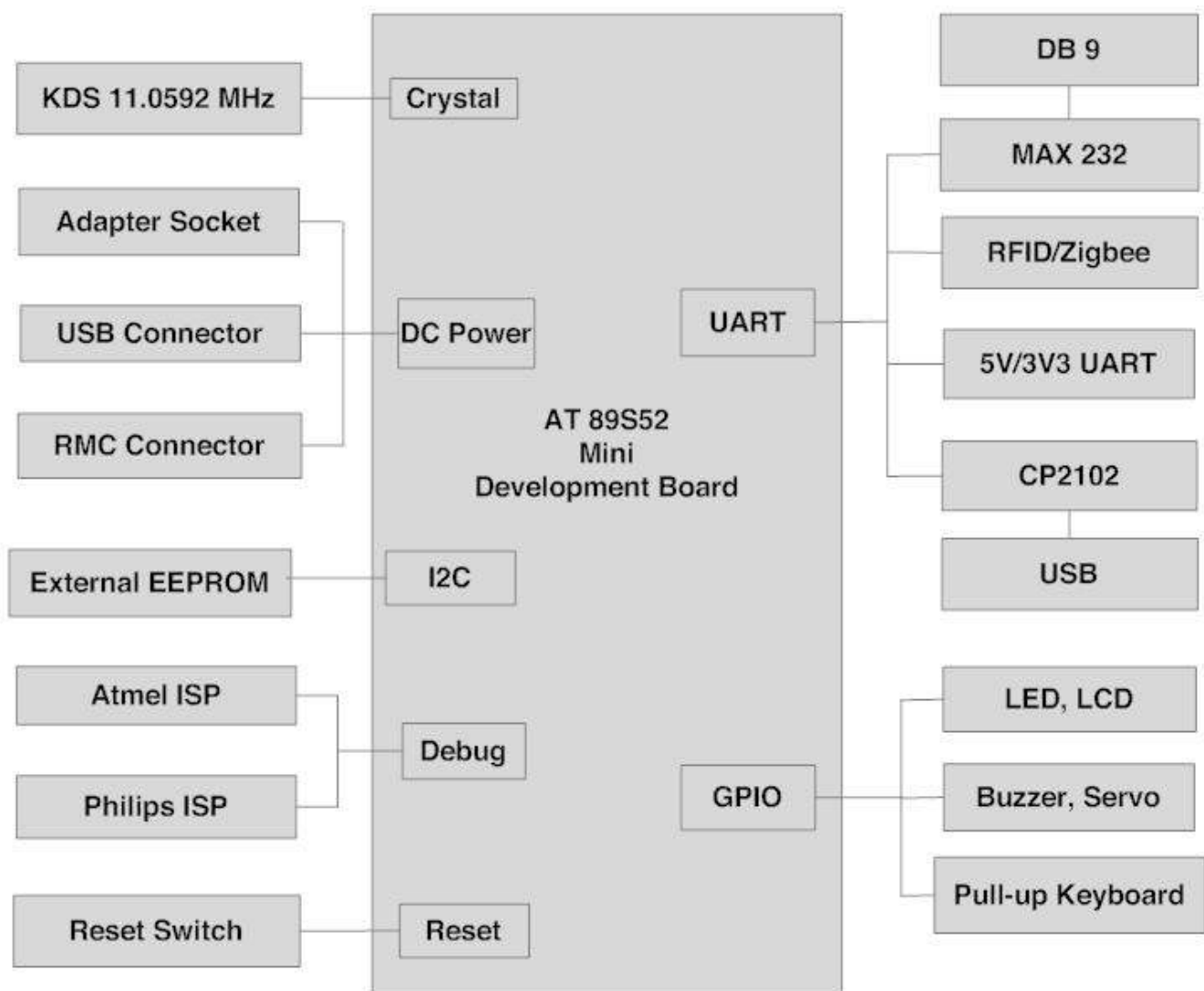
1. 8K Bytes of In-System Reprogrammable Flash Memory
2. Fully Static Operation: 0 Hz to 33 MHz
3. Three-level Program Memory Lock
4. 256 x 8-bit Internal RAM
5. 32 Programmable I/O Lines
6. Three 16-bit Timer/Counters
7. Eight Interrupt Sources
8. Programmable Serial Channel
9. Low-power Idle and Power-down Modes
10. 4.0V to 5.5V Operating Range
11. Full Duplex UART Serial Channel
12. Interrupt Recovery from Power-down Mode
13. Watchdog Timer
14. Dual Data Pointer
15. Power-off Flag
16. Fast Programming Time
17. Flexible ISP Programming (Byte and Page Mode)



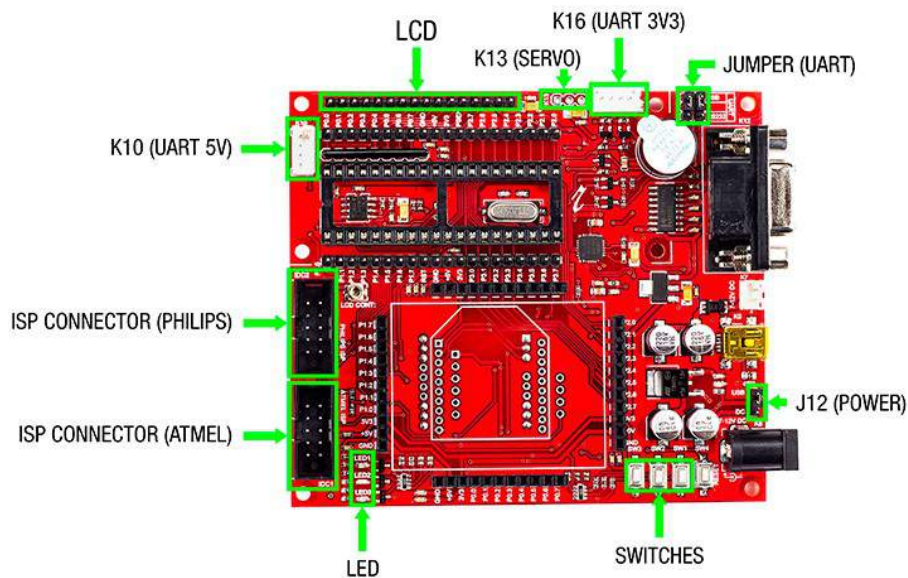
HARDWARE INTRODUCTION

The Atmel mini development board is a perfect tool for students and engineers to easily test and explore the abilities of AT89xxx microcontrollers. The board supports many peripheral, to enable them the appropriate jumpers have to be properly set. The AT89 series remains very popular as general purpose microcontrollers, due to their industry standard instruction set, and low unit cost. AT89S52 Development Board (Mini) V1.00 package as its name suggests includes the ATMEL make AT89S52 microcontroller (DIP40).

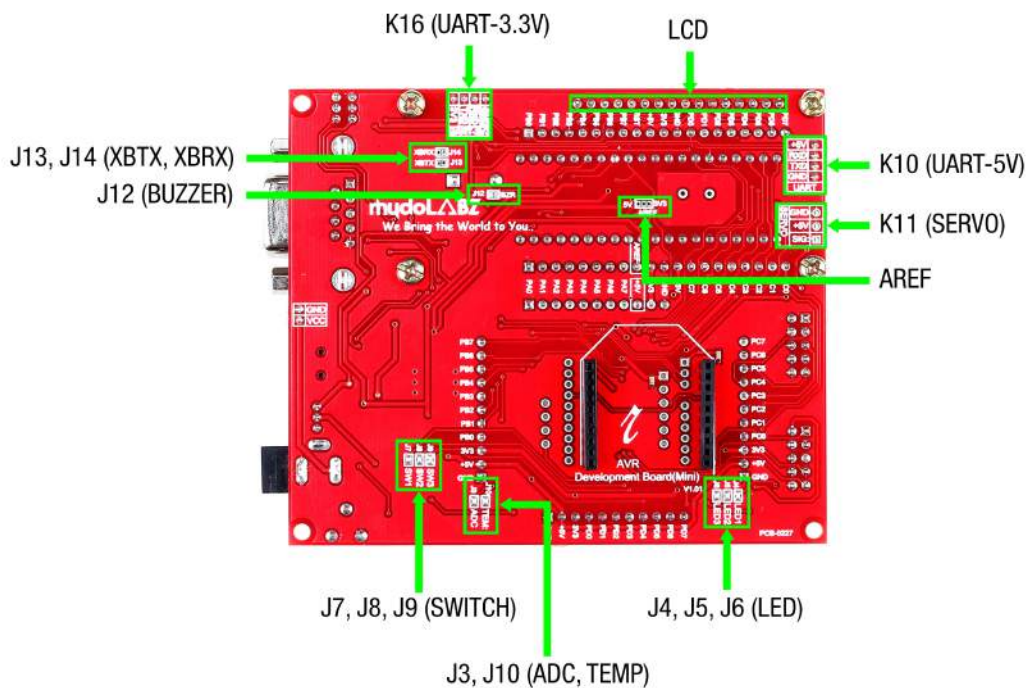
2.1. FUNCTION MODULE



2.2. INTERFACE OVERVIEW



Top view of the Development board -Mini



Bottom view of the Development board -Mini



2.3. PERIPHERALS DESCRIPTION

Peripheral	Description
K1,K2,K3 & K4	GPIO Pins Near Breadboard
RA1	10K Resistor Pack
K5	USB Socket
K7	7-12VDC Connector
K8	Barrel Jack Connector
K9 & K11	GPIO Pins Near Breadboard
K10	5V UART Connect Pins
K12	DB9 Connector
K13	Servo Connect Pins
K16	3V3 UART Connect Pins
U1	AT89XXX / P89VXX IC
U2	LD1117 (Regulator IC (3V3))
U3	LM7805(Regulator IC(5V))
U4	CP2102 IC
U5	MAX232 IC
U6	EEPROM(24C04)
U7	Zigbee Module Connector
Y1	Crystal (11.059MHz)
SW1-SW3	Pull-Up switches
SW4	Reset Switch
LCD1	LCD Module, HD44780U
ATMEL ISP	To Program ATMEL IC
PHILIPS ISP	To Program PHILIPS IC
LED1-LED3	Light Emitting Diodes
PWR	Power LED
BUZ1	Buzzer
M1	RFID MODULE INTERFACE



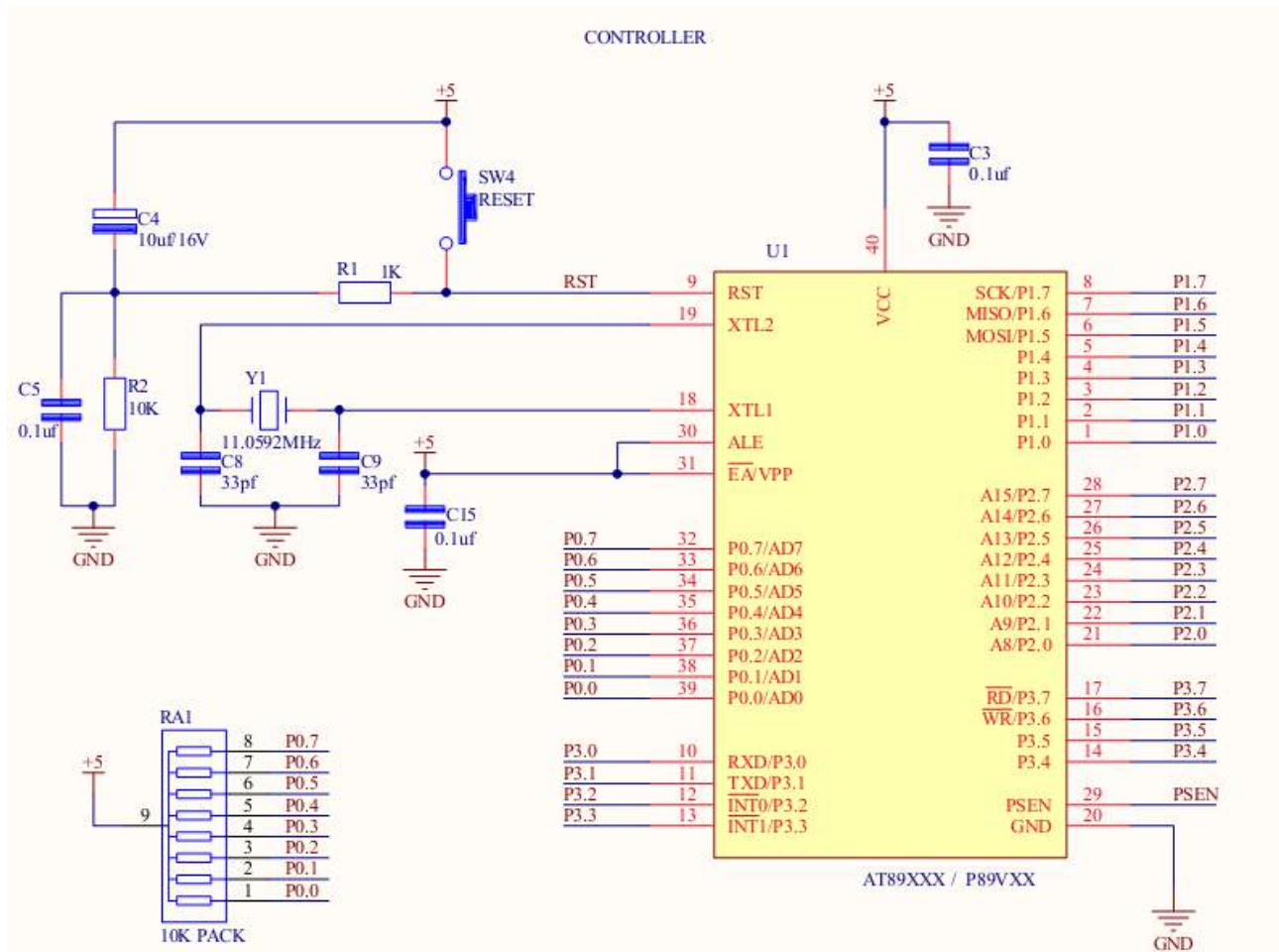
2.4. JUMPER DESCRIPTION

Jumper no	Description	Set option	Set description
J1 & J2	UART Selection	1-2 2-3	CP2102 Selected MAX232 Selected
J3, J4 & J6	LED Connection	Short to Access	Connecting LED1, LED2 & LED3 with P1.5, P1.4 & p1.3 respectively
J4	Buzzer Connection	Short to Access	Connecting buzzer to P1.6
J7, J8 & J9	Pull-up Key Connection	Short to Access	Connecting switches SW1, SW2 & SW3 with P2.0, P2.1 & P2.3 respective
J10 & J11	I2C Interfacing	Short to Access	Connecting SDA & SCL of External EEPROM to P3.4 & P3.5
J12	Power Selection	1-2 2-3	DC Power is selected USB Power is selected
J13 & J14	Zigbee Connection	Short to Access	Connects Zigbee RX and TX with UART



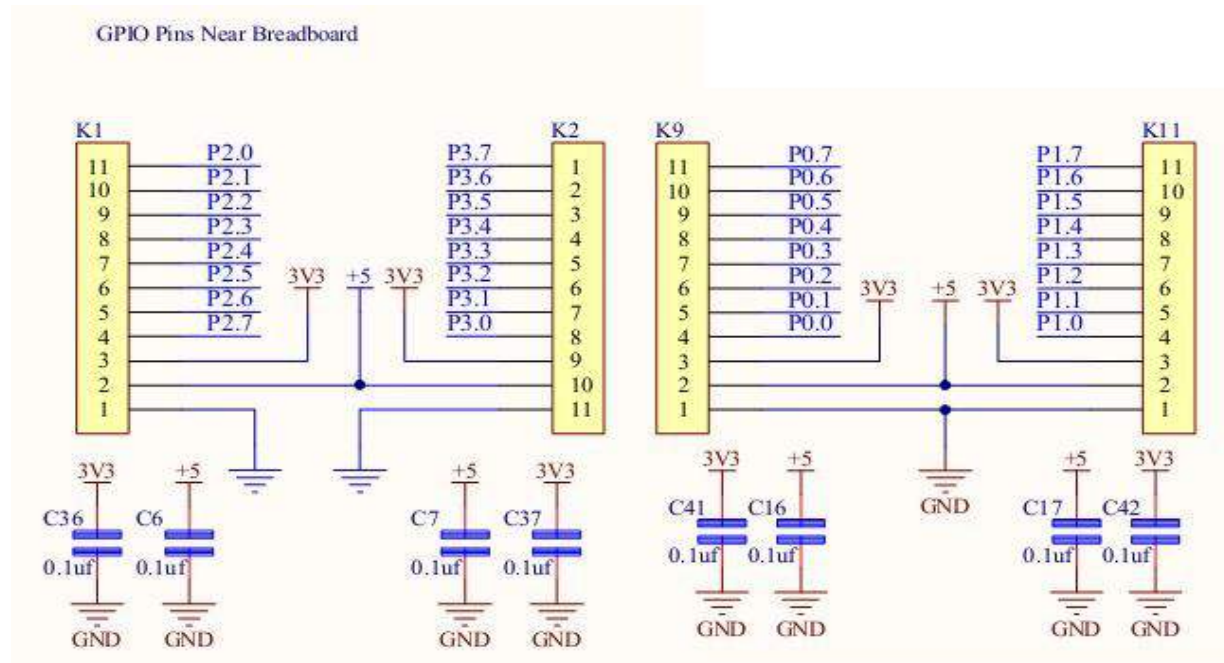
2.5. MICROCONTROLLER - PIN OUT

The AT89S52 Development Board (Mini) V1.00 is compatible with the AT89 series of microcontrollers from ATMEL and the P89V series of microcontrollers from PHILIPS. The microcontroller is held on a 40 pin IC base provided on the board. Hence DIP40 package of any microcontroller that belongs in the foretold series (AT89xxx/P89xx) will be apart for use with this development board.

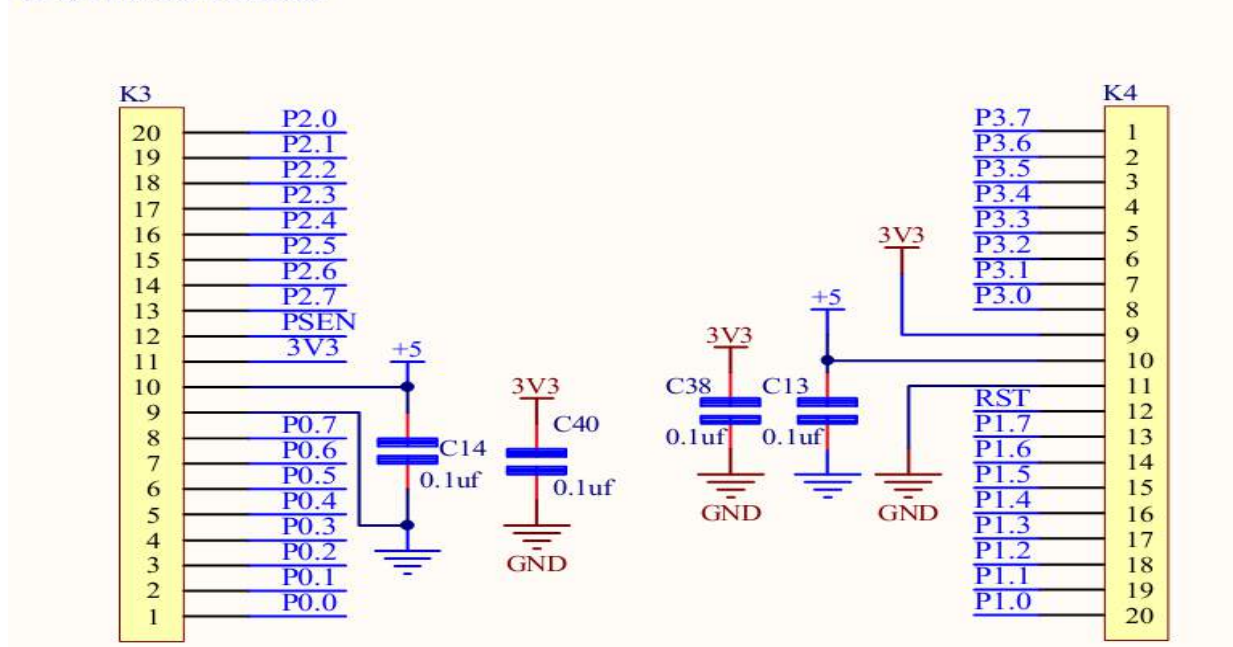


2.6. PORT PINS

The Atmel mini development board has all port pins available at direct port access connectors. The connection is as given below.



GPIO Pins Near Controller



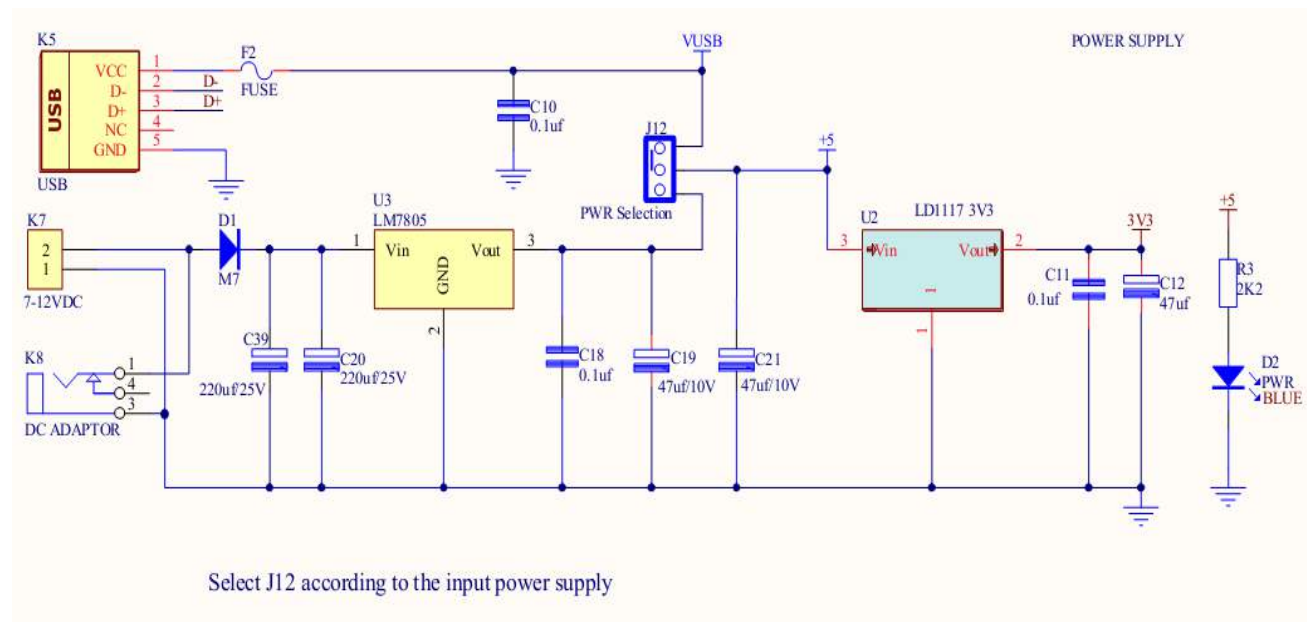
2.7. POWER SUPPLY

The AT89S52 Development Board (Mini) V1.00 comes with three options of input power supply. It can be powered via either of the following (jumper selectable)

1. USB Programming cable (+5V) connected to on board USB connector
2. DC adapter (7V - 12V) connected to the on board DC barrel jack
3. External DC power supply connected to on board 2 Pin RMC connector

Of the above three input options, the suitable one is selected using on board jumper. The USB/DC powering is selected using the power selection jumper J12. When the board is USB powered it survives slight supply variations (if any) with a help of a fuse provided on the board. If it is powered using an external DC source either via DC barrel jack or the RMC connector, then the board is capable to withstand a voltage variation of 7 to 12 volts. Upon powering up, the blue power indication LED provided on the board will glow. Output pins, are provided on the board with 5V and 3.3V out, which will be very helpful while testing circuits on bread board as well as to power up other external modules.

The external Power Supply circuit is given below:

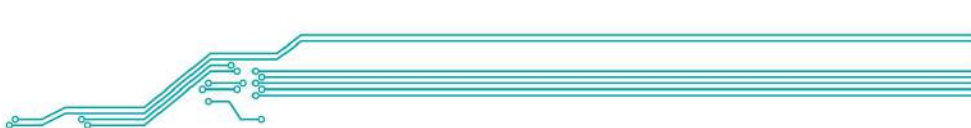


2.8. CLOCK SOURCE

In general case a quartz crystal is used to make the clock circuit. Clock frequency limits (maximum and minimum) may change from device to device. Standard practice is to use 12 MHz frequency. If serial communications are involved then it is best use 11.0592 frequency.

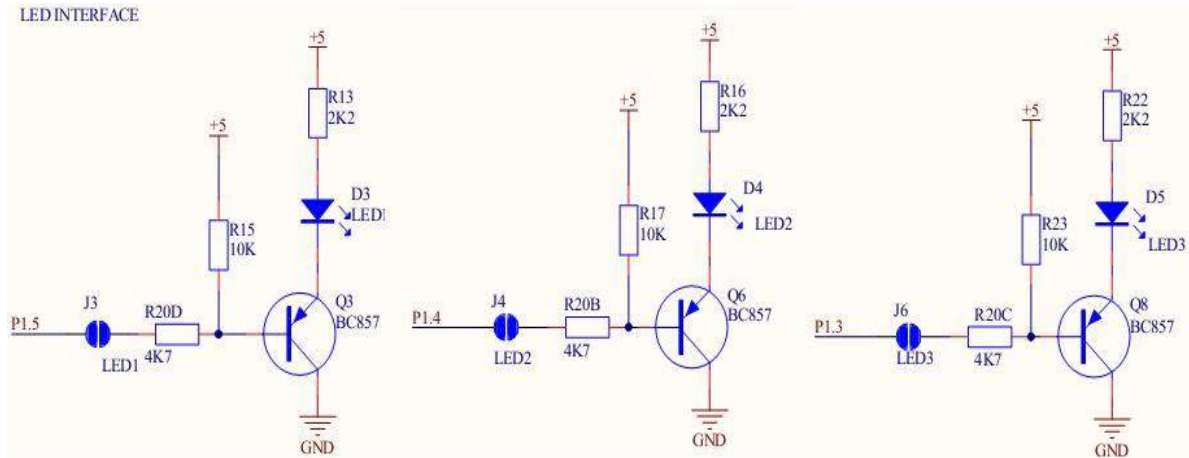
The ATMEL AT89S51/52 evaluation board has a clock source

- 11.0592 MHz Crystal as the MCU clock source



2.9. LED INTERFACING

LED's are the simplest devices to test port functioning. The board contains 3 LEDs connected to PORT1 pins 3 to 5 (P1.3to P1.5).



Note: For the LED to work, the jumpers J3, J4 & J6 must be shorted .

2.10. LCD - LIQUID CRYSTAL DISPLAY

The display is a standard 2x16 LCD which displays 2 lines of 16 characters. Each character is 40 pixels, making it 1280 pixels overall. The display receives ASCII codes for each character at the data inputs (D0–D7).

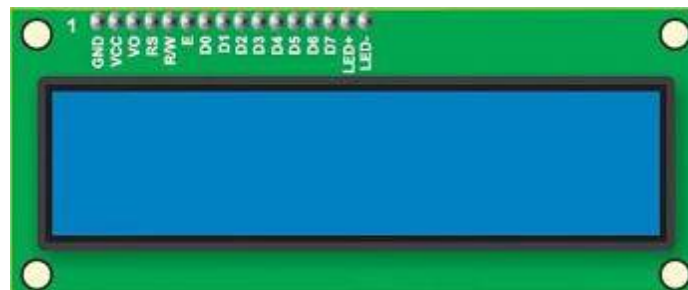
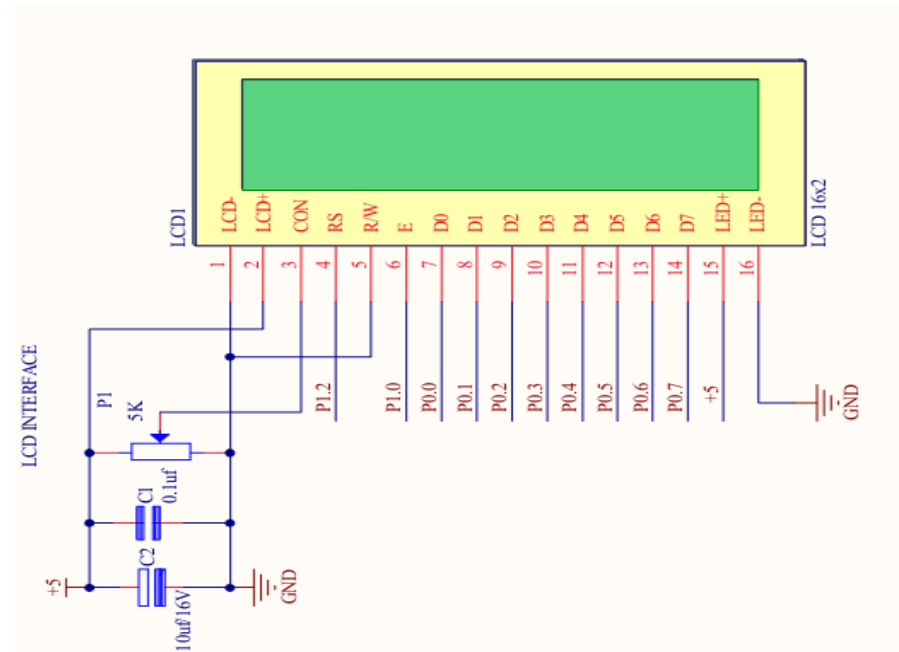


Figure: Connection Diagram.



The LCD module can be used in 4-bit or 8-bit mode. The module uses HD44780U (from Hitachi) as the controller IC. The ATMEL AT89S51/52 development board uses 8-bit interface. **PORT0** pins are used for sending **data/commands** and **PORT1** for **RS (P1. 2)** and **E (P1. 0)** pins. **R/W** pin is directly **grounded** for LCD write select.

An On-Board potentiometer enables to adjust the LCD contrast to a better view in every angle.



2.11. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

The ATMEL AT89S51/52 microcontroller comes with a single UART module. It has only asynchronous (no clock connection) mode of operation. The UART operates through P3.1 (TXD) and P3.0 (RXD) pins. The ATMEL AT89S51/52 UART output itself operates at CMOS voltages, and needs an external serial line driver to convert its output into a higher symmetrical line voltage. The ATMEL AT89S51/52 development board has two serial drivers for this purpose.

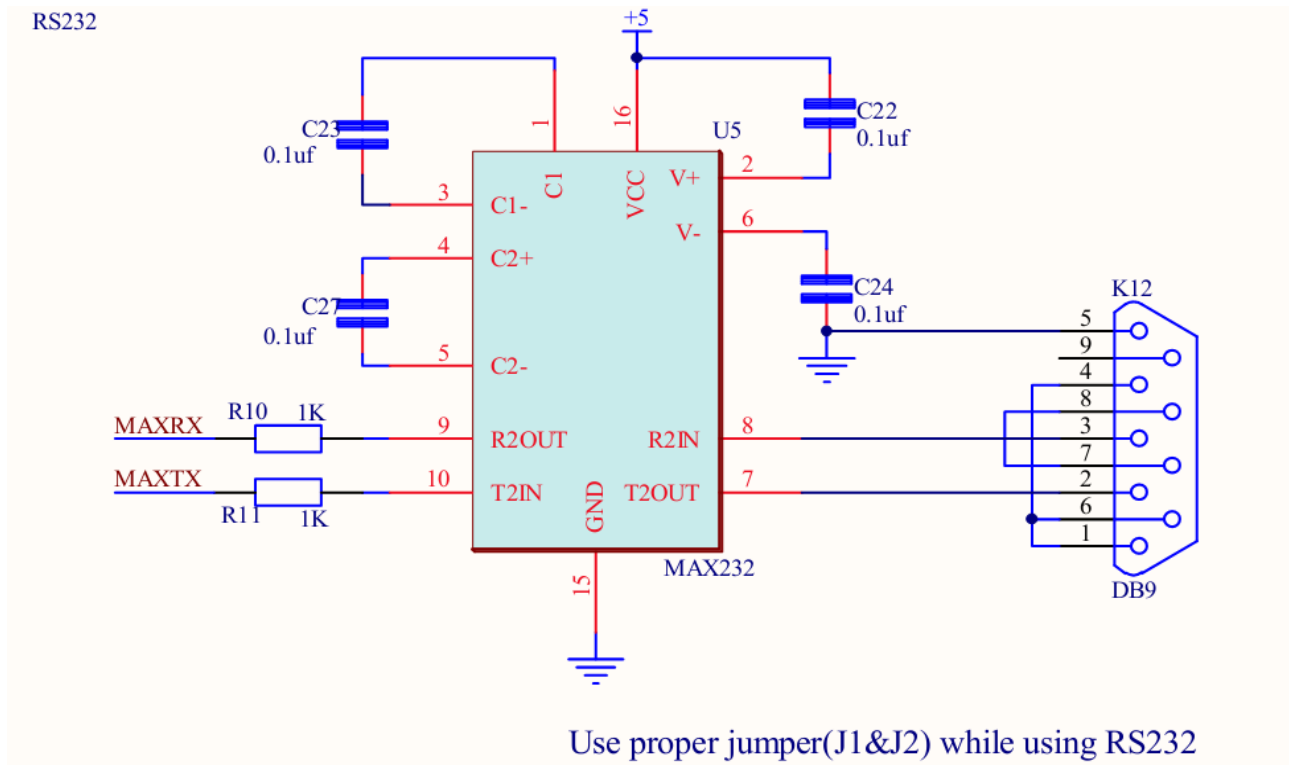
- The MAX232 serial driver for RS232 Interfacing and
- The CP2102 serial driver for USB Interfacing.

Note: Make use of jumpers J1 and J2 for choosing the type of Interfacing.



RS232 Interfacing

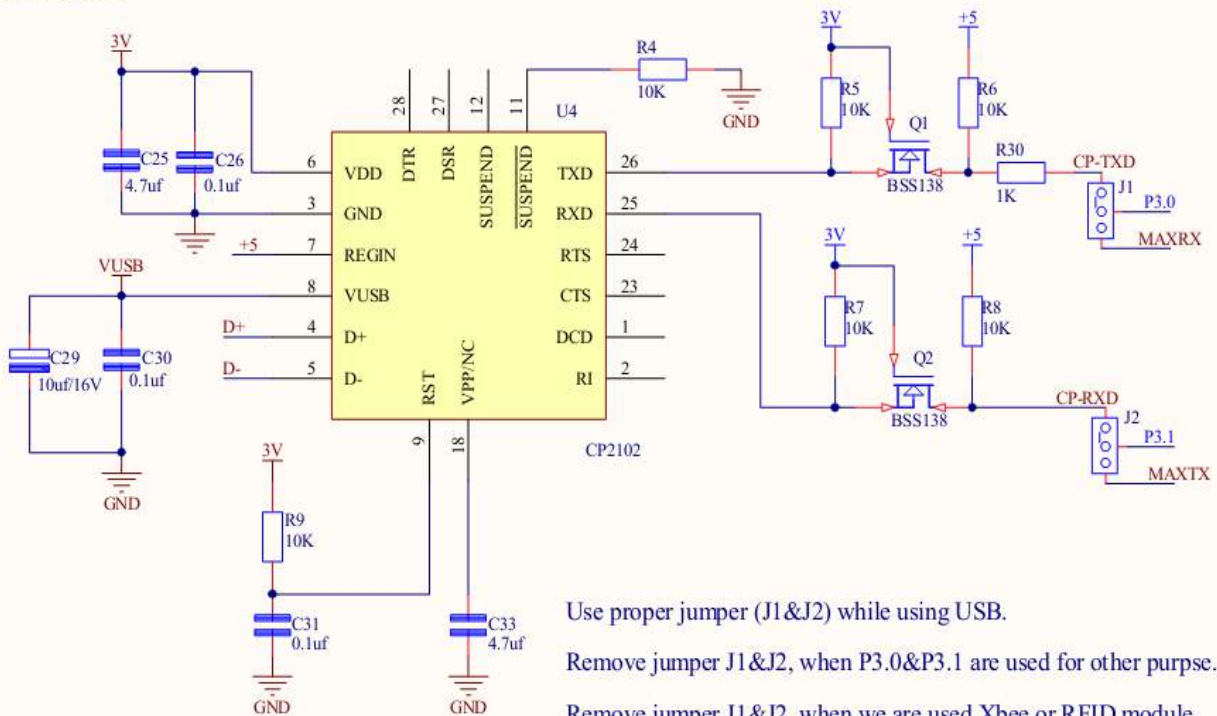
RS-232 is a standard communication protocol for linking computer and its peripheral devices to allow serial data exchange. Since RS-232 communication voltage levels are different than microcontroller logic levels, it is necessary to use a RS-232 Transceiver circuit. Here we use serial driver IC MAX232 for interfacing RS-232 with microcontroller UART module. A DB9 female connector is provided to make direct connection with the serial port of a device or PC.



CP2102 USB Interfacing

The CP2102 is a highly-integrated USB-to-UART Bridge Controller providing a simple solution for updating RS-232 designs to USB using a minimum of components and PCB space. The CP2102 includes a USB 2.0 full-speed function controller, USB transceiver, oscillator, EEPROM, and asynchronous serial data bus (UART) with full modem control signals in a compact 5 x 5 mm MLP-28 package. No other external USB components are required.

USB INTERFACING



In addition to the serial drivers the ATMEL AT89S51/52 development board has two TTL UART, one for interfacing the 3V3 and other for interfacing 5V TTL modules.

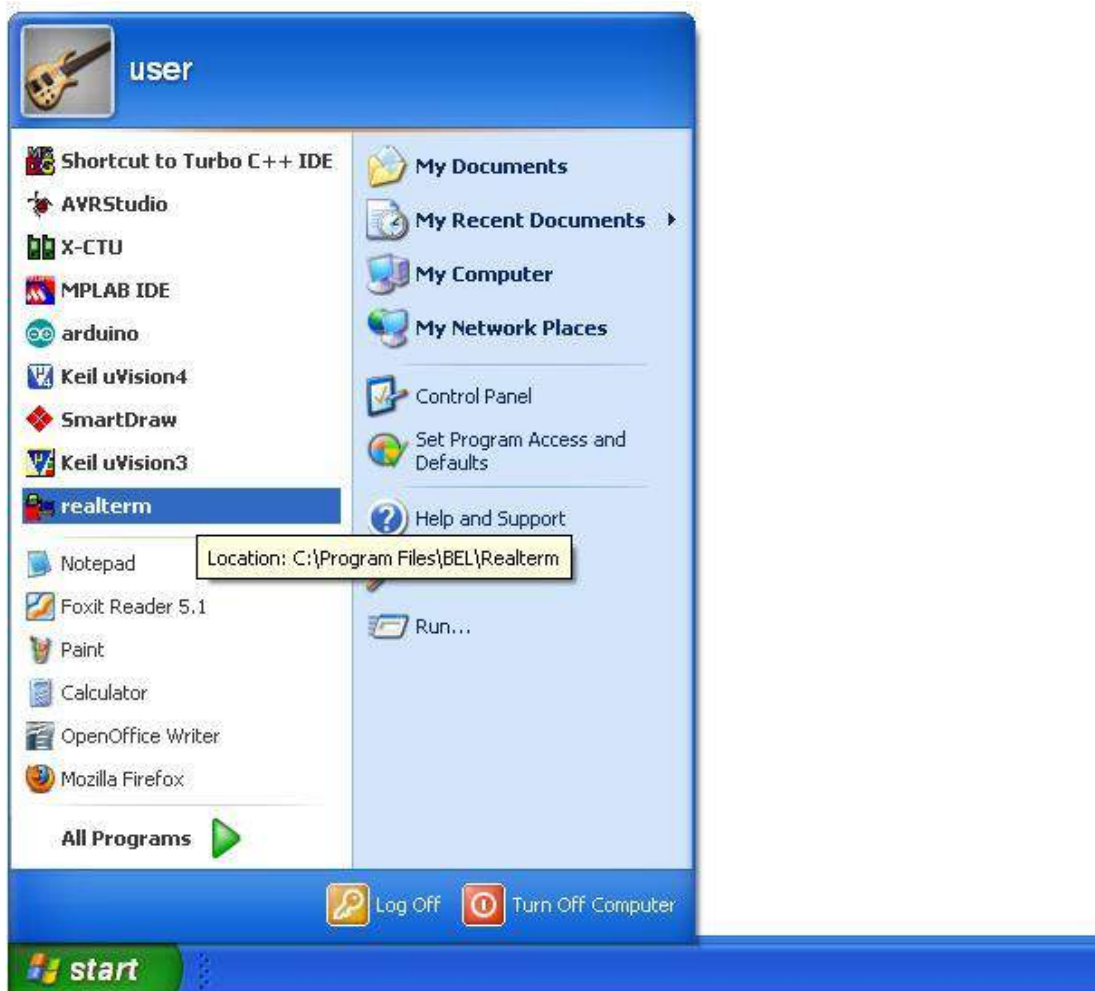


The RealTerm Software

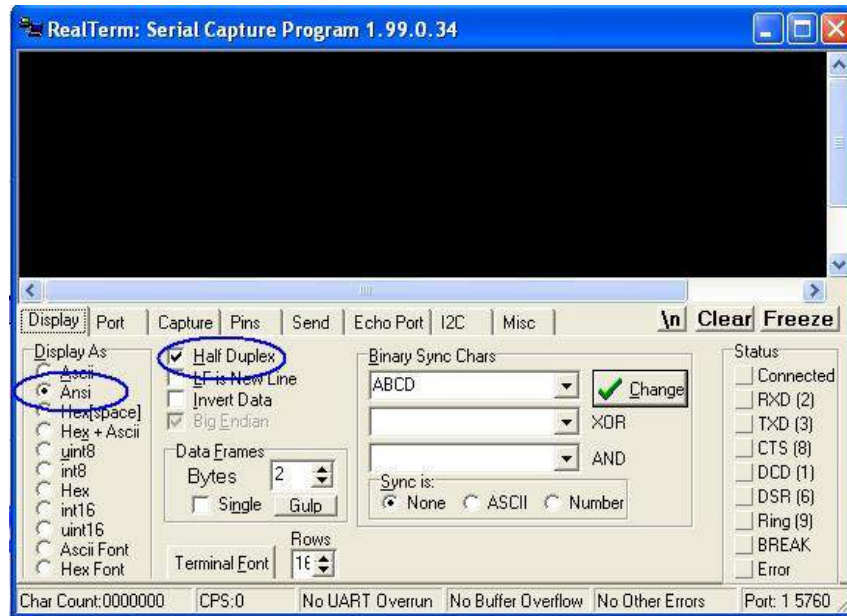
The serial data transmitted through UART can be viewed on a PC using the realterm software.

The following section provides step by step tutorial of serial communication using realterm software. The realterm software can be downloaded from <http://realterm.sourceforge.net/>

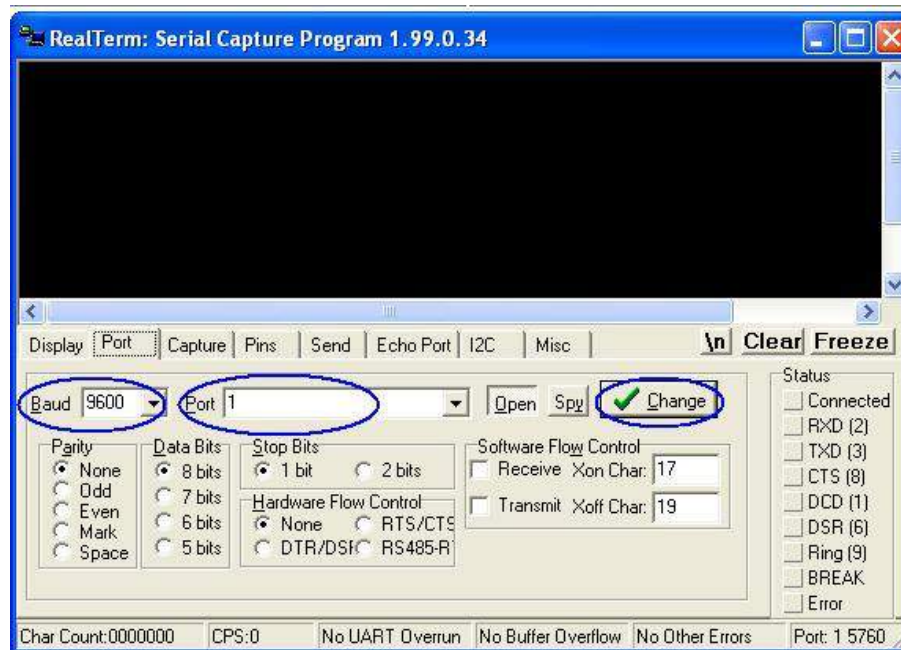
Step 1 : Open realterm software from Windows Start menu \Rightarrow All Programs \Rightarrow Real Term



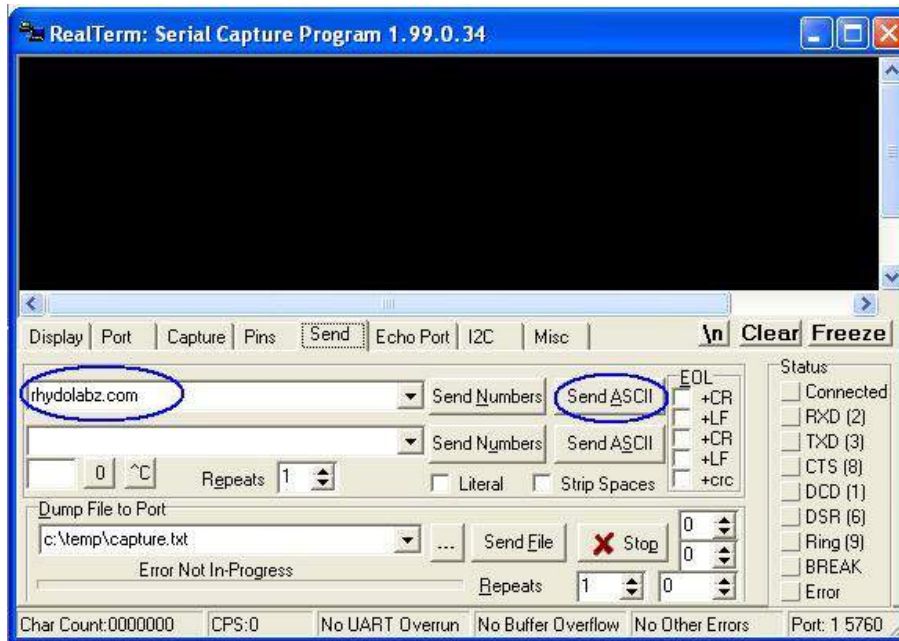
Step 2: Display Tab- Here the output text format selected is ANSI and Half Duplex mode is enabled to view the data sent by the user.



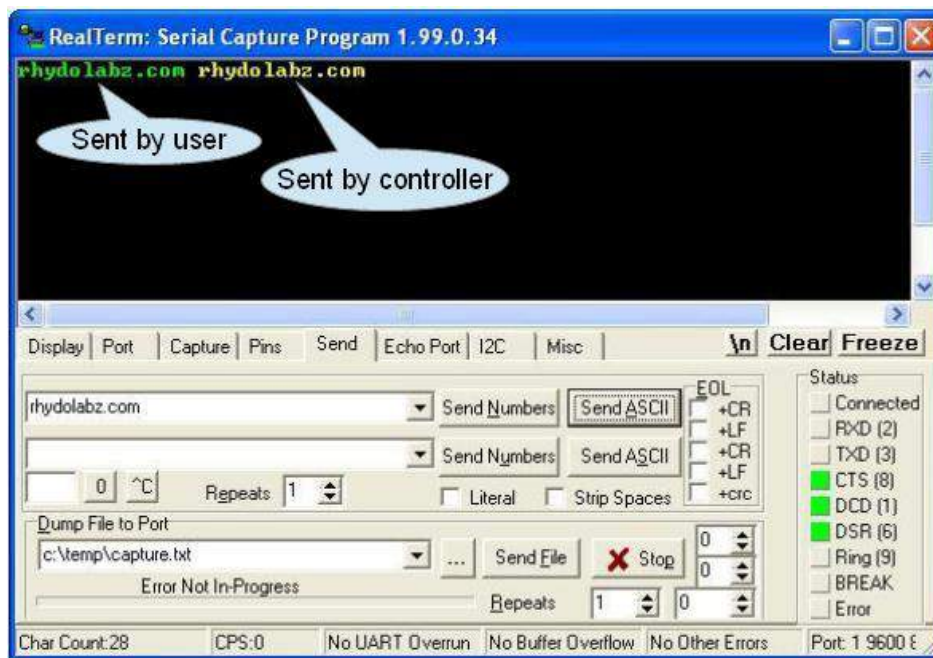
Step 3: Port Tab-To test the connection - make sure the **Open** button is pressed, Select required baud rate and the "Port" dropdown here, select the number of your COM port and then press the **Change** button.



Step 4: Send Tab- Insert the desired data to be transmitted and press "Send ASCII" button.

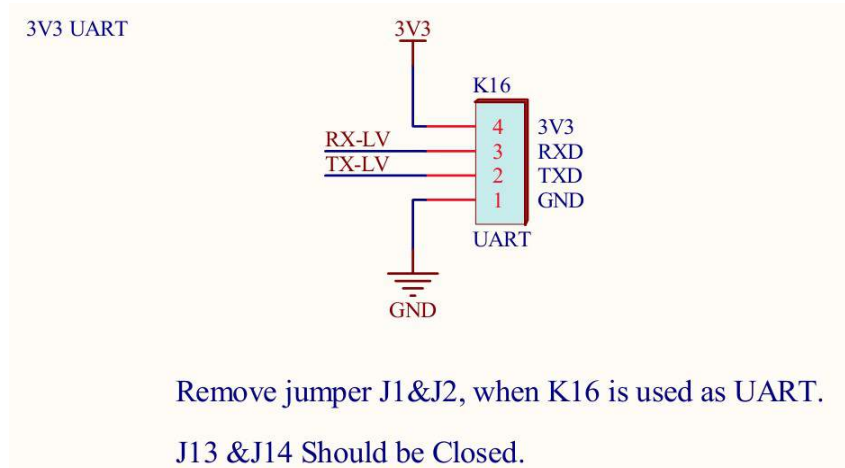


Step 5 : The output after data transmission to the controller is shown in the following diagram. The text sent by user and controller is highlighted by callouts in the figure.



3V3 UART

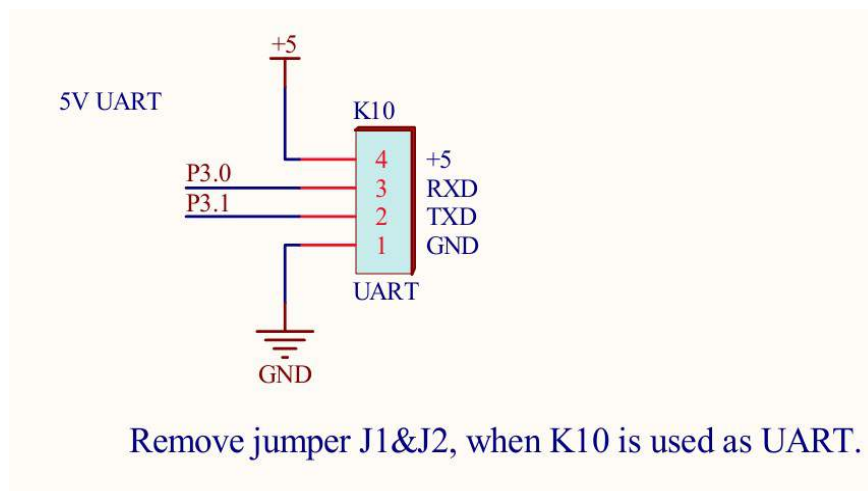
The 3V3 UART can be used to interface 3V3 TTL modules. The ATMEL AT89S51/52 development board has built in 3V3 to 5V converter and vice versa for communicating between the 3V3 devices and the microcontroller.



Note: Remove jumper J1 and J2 and close the jumpers at J13 and J14 while using 3V3 UART.

5V UART

The 5V UART can be used for the communication between 5V TTL modules and the microcontroller .

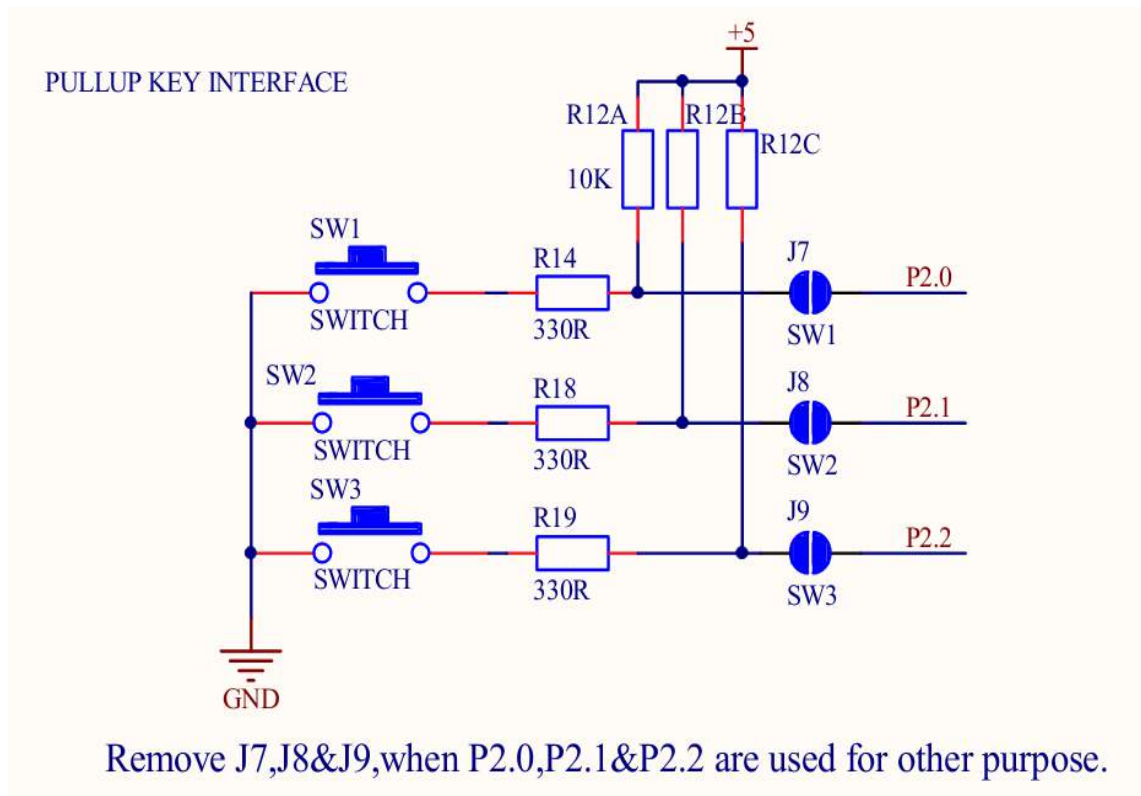


Note: Remove jumpers J1 and J2 while using 5V UART

2.12. PULL UP KEYBOARD

The simplest input to a microcontroller is a switch or push button. This can operate with just one additional support component, a pull-up resistor. When the switch is open, the output voltage of the circuit is pulled up to 5V via the resistor. Another way to look at it is that there is no current in the resistor (assuming there is no load on the output), so there is no volt drop, and the output voltage must be the same as the supply (5 V). When the switch is closed the output is connected direct to 0 V; the resistor prevents the supply being shorted to ground.

There are 3 pull up switches in the board connected to three PORT2 pins P2.0, P2.1 and P2.2

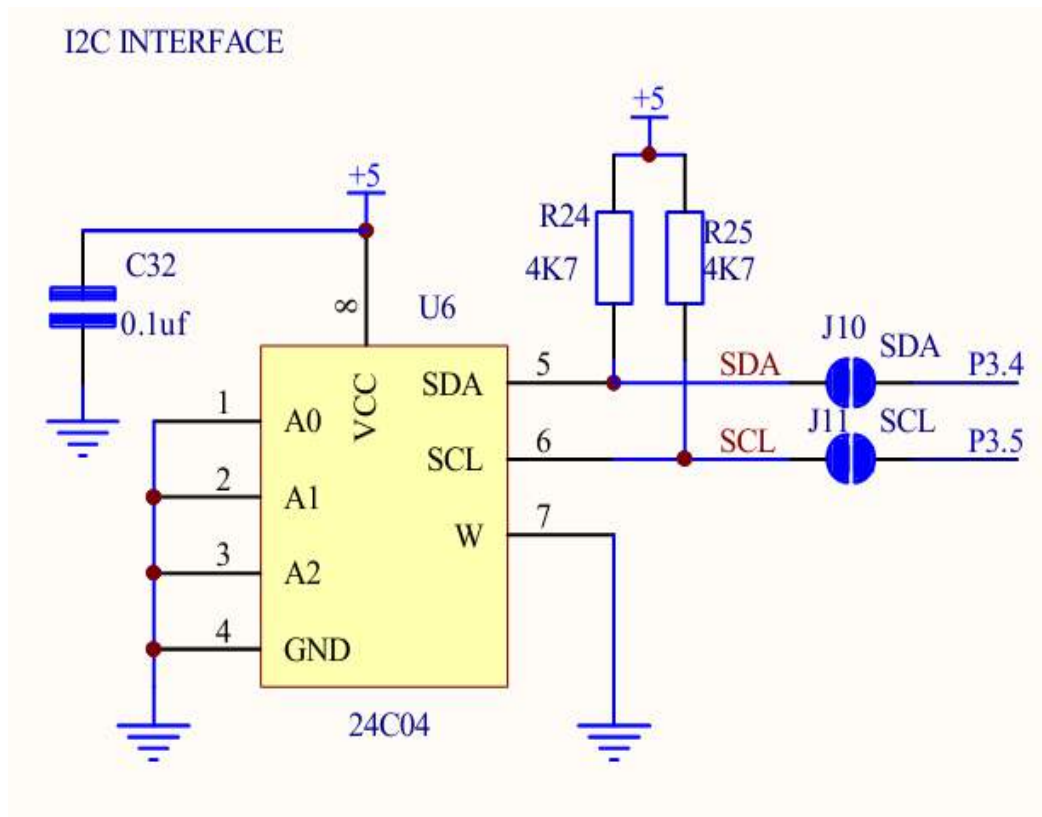


Note: short the jumpers J7, J8 and J9 to access the pullup switches.

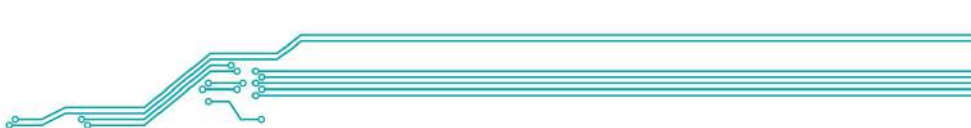


2.13. EXTERNAL EEPROM INTERFACING

The development board has 4K SERIAL EEPROM interfacing. Internally organized with 256 pages of 2-byte each, the 4K requires an 9-bit data word address for random word addressing. Also it is to be noted that an external EEPROM (AT24C04) is the slave device to be communicated with the microcontroller, via I2C protocol. PORT3 pins P3.4 and P3.5 are the SDA and SCL pins respectively.

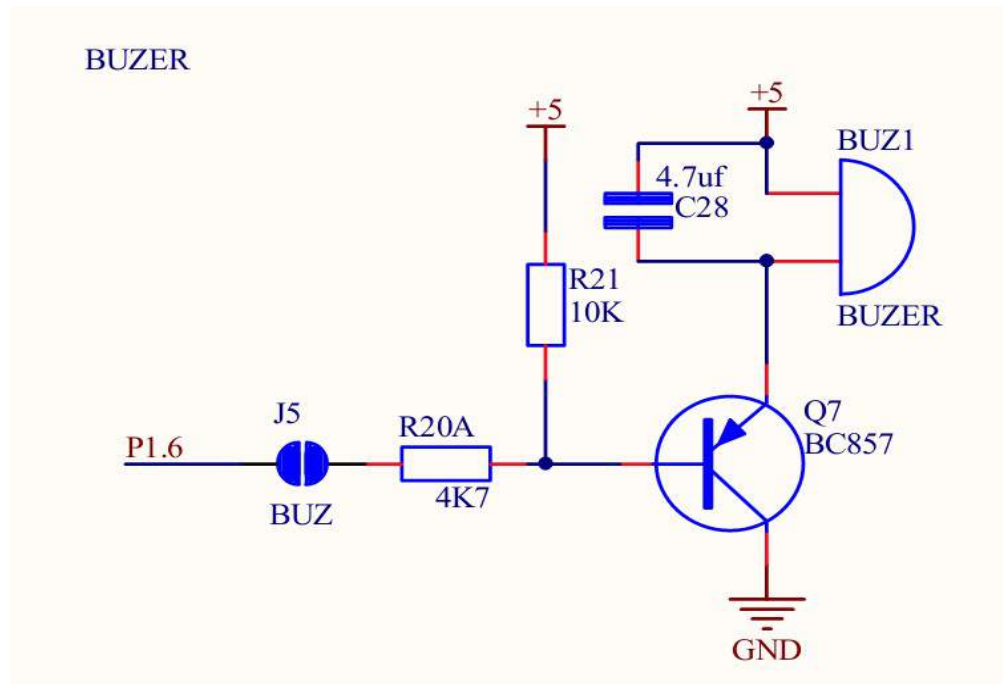


Note: Close the jumpers J10 and J11 while using I2C protocol.

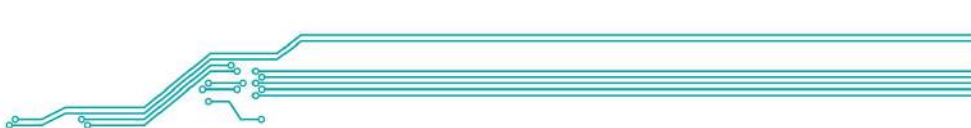


2.14. BUZZER INTERFACING

Buzzer is a piezoelectric material. Basically, the sound source of a piezoelectric sound component is a piezoelectric diaphragm. Buzzer is another method to test port functioning. The buzzer is connected to PORT1 (Pin P1.6) and is driven using a transistor (BC 857).



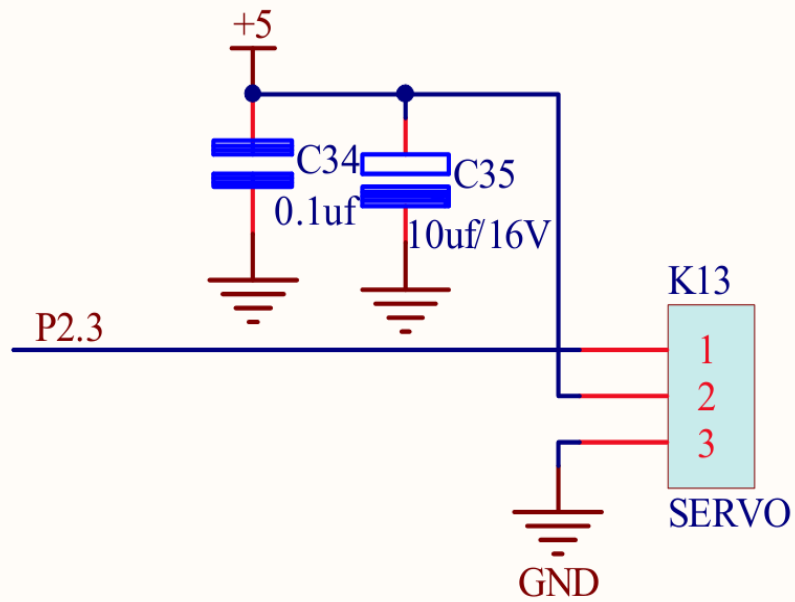
Note: Short jumper J5 to access connection to buzzer.



2.15. SERVO MOTOR INTERFACING

A servo is a mechanical motorized device that can be instructed to move the output shaft attached to a servo wheel or arm to a specified position. Servos are controlled by sending pulse width modulation to the signal pin. The ATMEL AT89S51/52 development boards are provided with the pinout for servo control. The signal pin of servo is connected with PORT2 (pin P2.3).

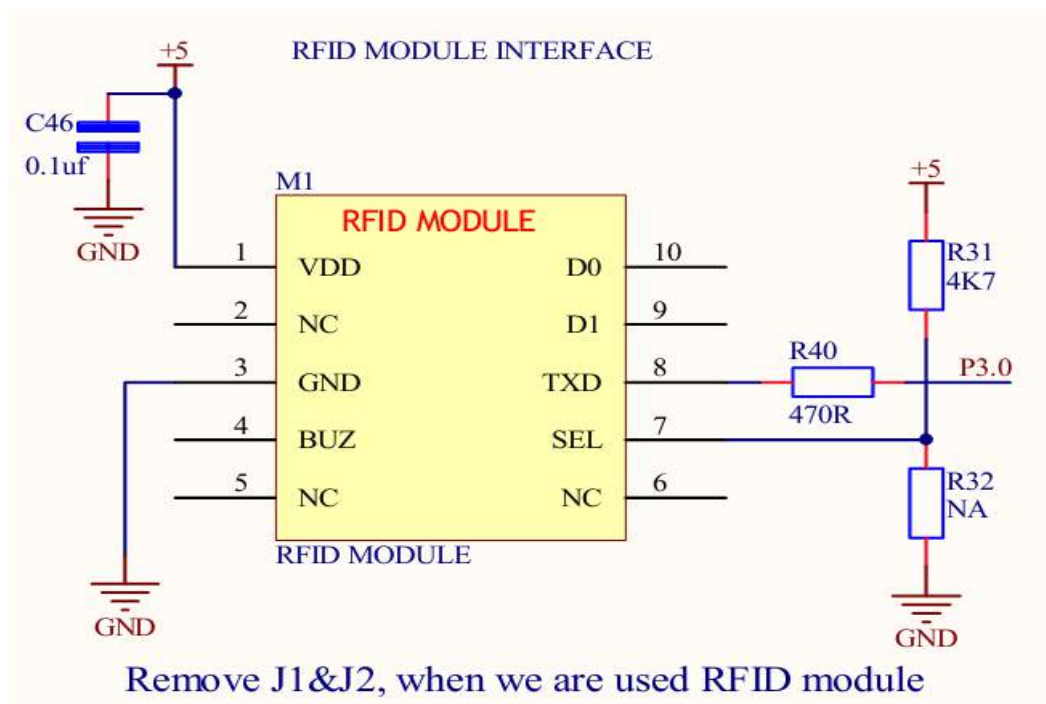
SERVO MOTOR



2.16. RFID INTERFACING

Radio Frequency Identification is the wireless use of electromagnetic fields to transfer data. RFID technology is composed of RFID Tags and RFID readers. Tag is transponder which receives a radio signal and in response sends out a radio signal. An RFID reader is a device that is used to interrogate an RFID tag.

Rhydolabz ATMEL AT89S51/52 mini development board have pinout compatibility for micro RFID reader interfacing. Communication with the RFID reader and controller uses a standard UART interface compatible with 5V.

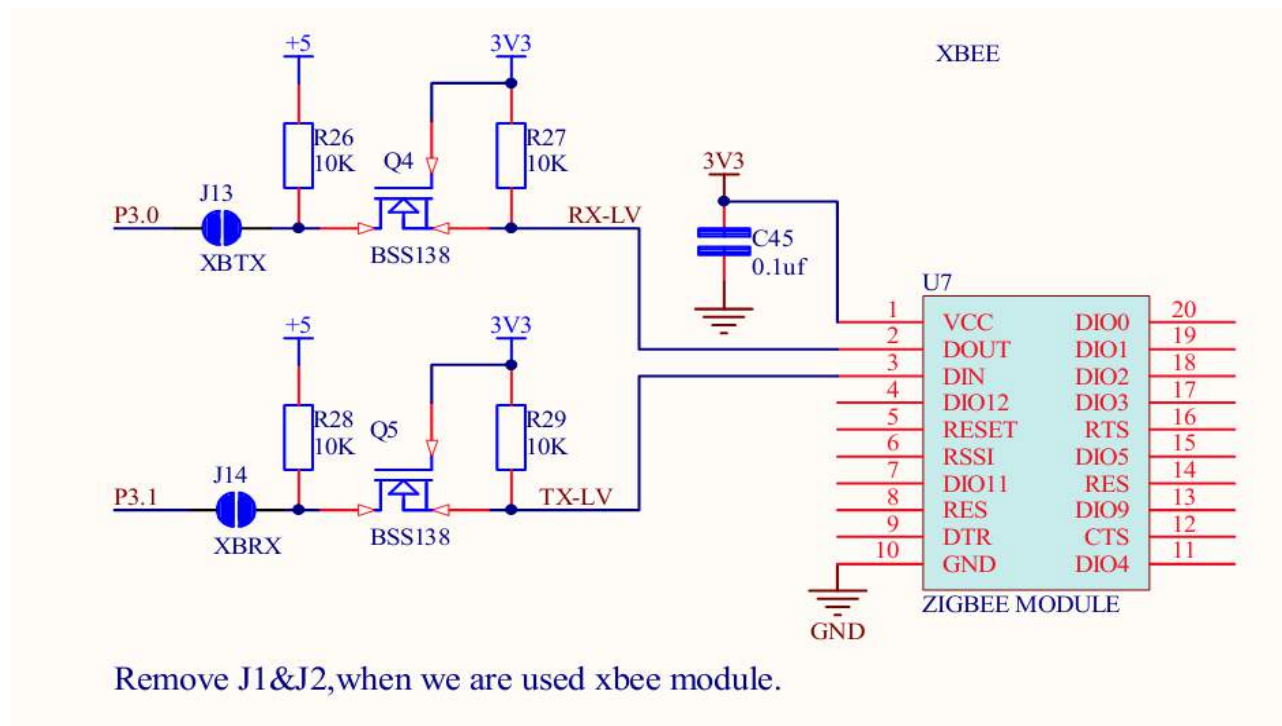


Note: Remove jumpers J1 and J2 while using RFID module.



2.17. ZIGBEE INTERFACING

ZigBee is a wireless networking standard that is aimed at remote control and sensor applications which is suitable for operation in harsh radio environments and in isolated locations. ZigBee technology builds on IEEE standard 802.15.4. Rhydolabz ATMEL AT89S51/52 mini development board have pinout compatibility for zigbee interfacing. Communication with the ZigBee module uses a standard UART interface compatible with 3V3.



Note: Remove jumpers J1 and J2 and close jumpers J14 and J13, when Zigbee module is using.



3.1. FAMILIARATION OF KEIL COMPILER

CREATING A PROJECT

µVision3 is a standard Windows application and started by clicking on the program icon. µVision3 includes a project manager which makes it easy to design applications for an AT89s52 microcontroller. You need to perform the following steps to create a new project:

- Select the Toolset (only required for AT89s52 Projects).
- Create Project File and Select CPU.
- Create New Source Files.
- Add Source Files to the Project.
- Set Tool Options for Target Hardware.
- Configure the CPU Startup Code.
- Create a HEX File.
- Build Project and Generate Application Program Code.

The section provides a step-by-step tutorial that shows you how to create a simple µVision3 project.



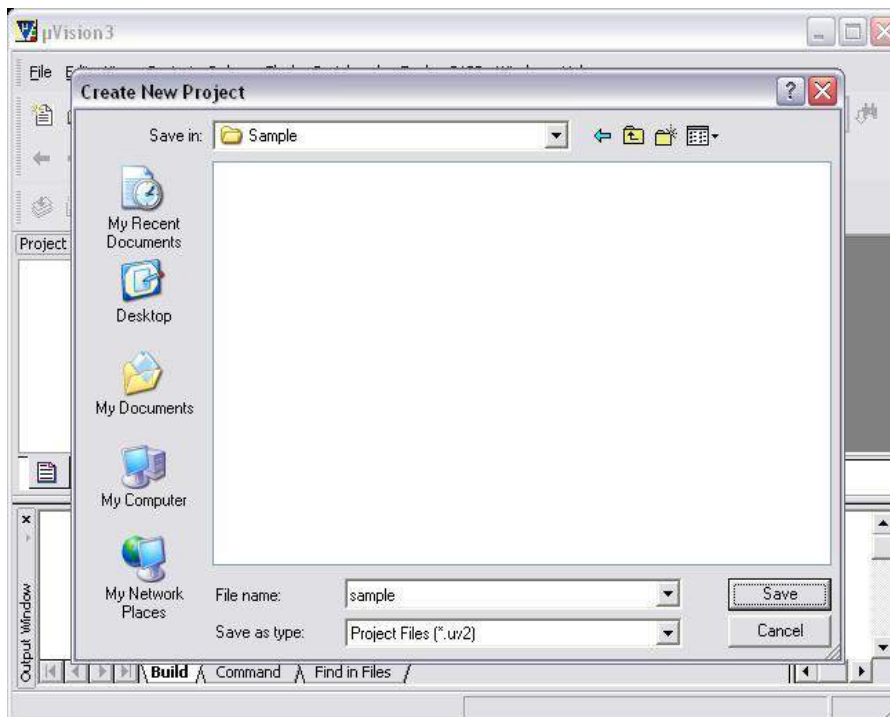
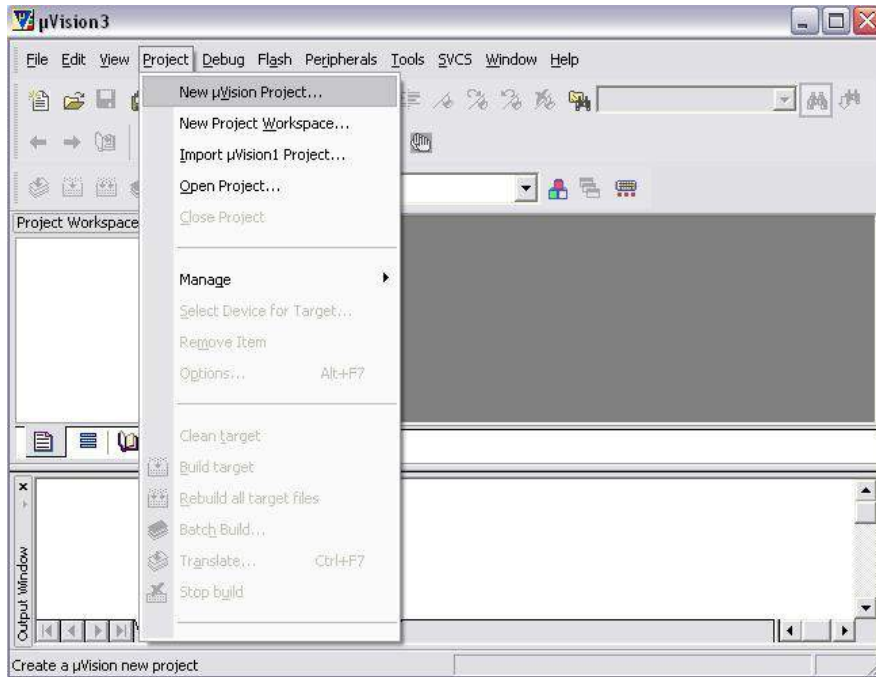
Select the Keil μ Vision3 tool as shown below.



Create Project File Folder and Specify Project Name

To create a new project file select from the μ Vision3 menu File –Project...NEW PROJECT. This opens a standard Windows dialog that asks you for the new project file name. You should use a separate folder for each project. You can simply use the icon Create New Folder in this dialog to get a new empty folder. Select this folder and enter the file name for the new project.



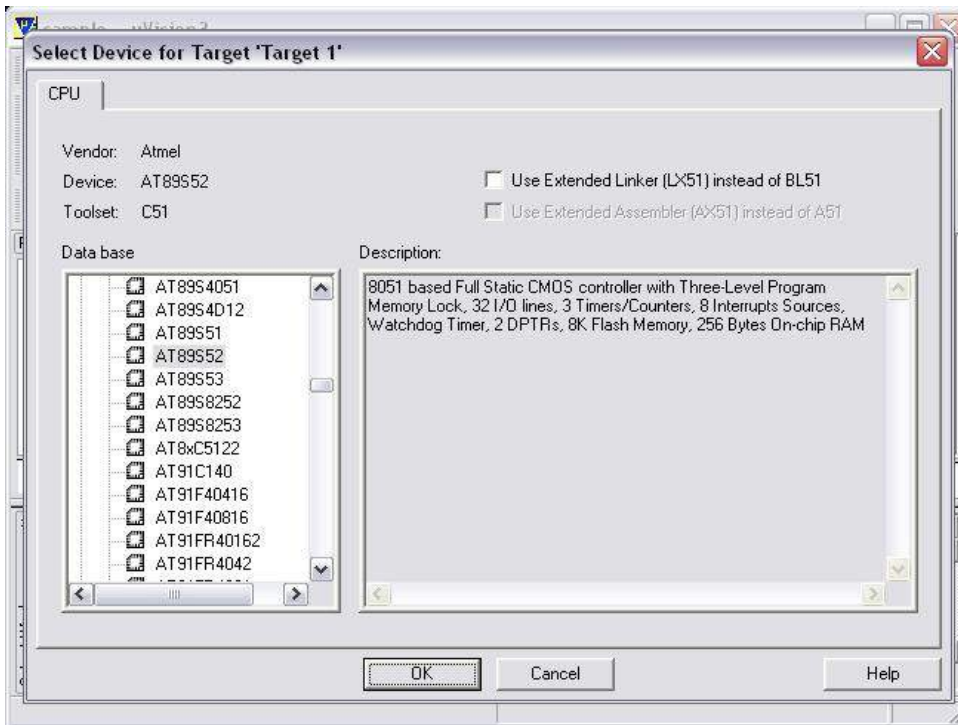


Select Microcontroller from Device Database

When you create a new project µVision3 asks you to select a CPU for your project. The Select Device dialog box shows the µVision3 device database. Just select the microcontroller you use. For the example in this chapter we are using the Atmel AT89s52 controller. This selection sets necessary tool options for the AT89s52



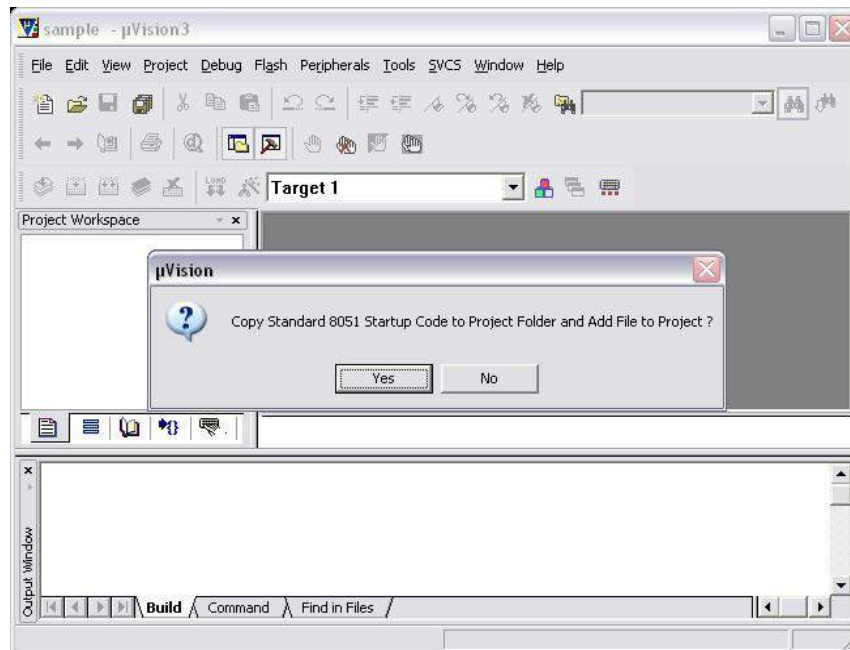
device and simplifies in this way the tool configuration



Copy and Add the CPU Startup Code

An embedded program requires CPU initialization code that needs to match the configuration of your hardware design. This Startup Code depends also on the tool chain that you are using. Since you might need to modify that file to match your target hardware, the file should be copied to your project folder. For most devices, μ Vision3 asks you to copy the CPU specific Startup Code to your project. This is required on almost all projects (exceptions are library projects and add-on projects). The Startup Code performs configuration of the microcontroller device and initialization of the compiler run-time system. Answer with YES to this question.





Note : The CPU Startup Code typically requires some configuration; however the default configuration gives you a good starting point for single chip applications.

Create New Source Files

You may create a new source file with the menu option File – New. This opens an empty editor window where you can enter your source code. µVision3 enables the C color syntax highlighting when you save your file with the dialog File – Save As... under a filename with the extension *.c. We are saving our example file under the name sample.c



```

/*****
 *
 *           LED Blinking Program
 *****/
#include <REGX52.H>      /* Header File Inclusion */
void delay(void);
/*
=====
                Main Program
=====
*/
void main()
{
    P1=0xC7; /* P1.5,P1.4 & P1.3 Configured as output*/
    while(1)
    {
        P1=0xFF; /* PORT1 Set-Turn ON LED */
        delay(); /* Delay Provided */
        P1=0x00; /* PORT1 Clear-Turn ON LED */
        delay(); /* Delay Provided */
    }
}
/*
=====
                Delay Function
=====
*/
void delay(void)
{
    int i,k;
    for(i=0;i<255;i++)
    for(k=0;k<255;k++);
}

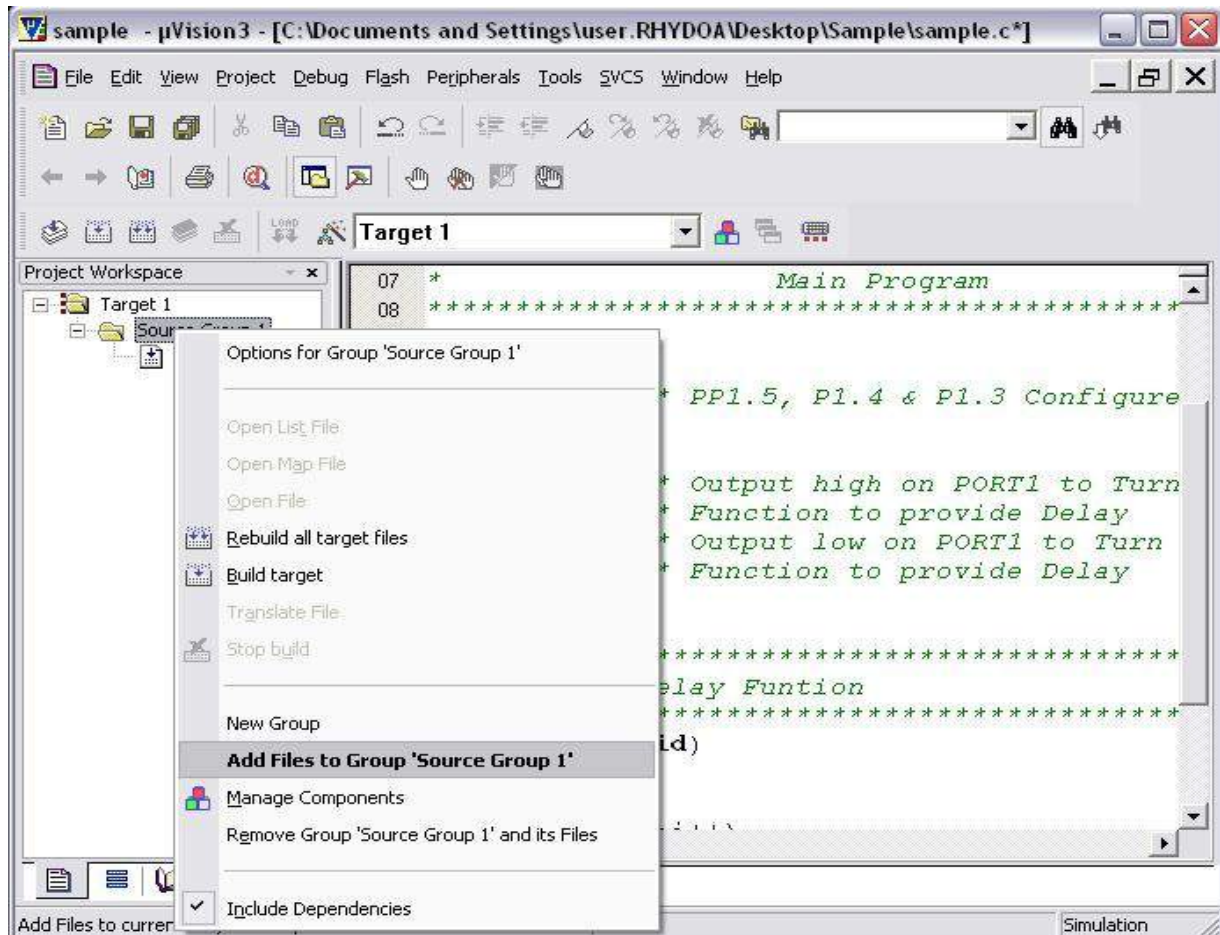
/*****End Of Program*****/

```



Add Source Files to Project

Once you have created your source file you can add this file to your project. μ Vision3 offers several ways to add source files to a project. For example, you can select the file group in the Project Workspace – Files page and click with the right mouse key to open a local menu. The option Add Files opens the standard files dialog. Select the file sample.c you have just created.



Set Tool Options for Target

µVision3 lets you set options for your target hardware. The dialog Options for Target opens via the toolbar icon or via the Project - Options for Target menu item. In the Target tab you specify all relevant parameters of your target hardware and the on-chip components of the device you have selected. The following dialog shows the settings for our example.

Options for Target 'Target 1'

Device Target Output Listing User C51 A51 BL51 Locate BL51 Misc Debug Utilities

Atmel AT89S52

Xtal (MHz): 11.0592 ☐ Use On-chip ROM (0x0-0x1FFF)

Memory Model: Small: variables in DATA

Code Rom Size: Large: 64K program

Operating system: None

Off-chip Code memory

	Start:	Size:
Eprom		
Eprom		
Eprom		

Off-chip Xdata memory

	Start:	Size:
Ram		
Ram		
Ram		

☐ Code Banking

Banks: 2 Bank Area: 0x0000 0x0000

☐ 'far' memory type support

☐ Save address extension SFR in interrupts

OK Cancel Defaults Help

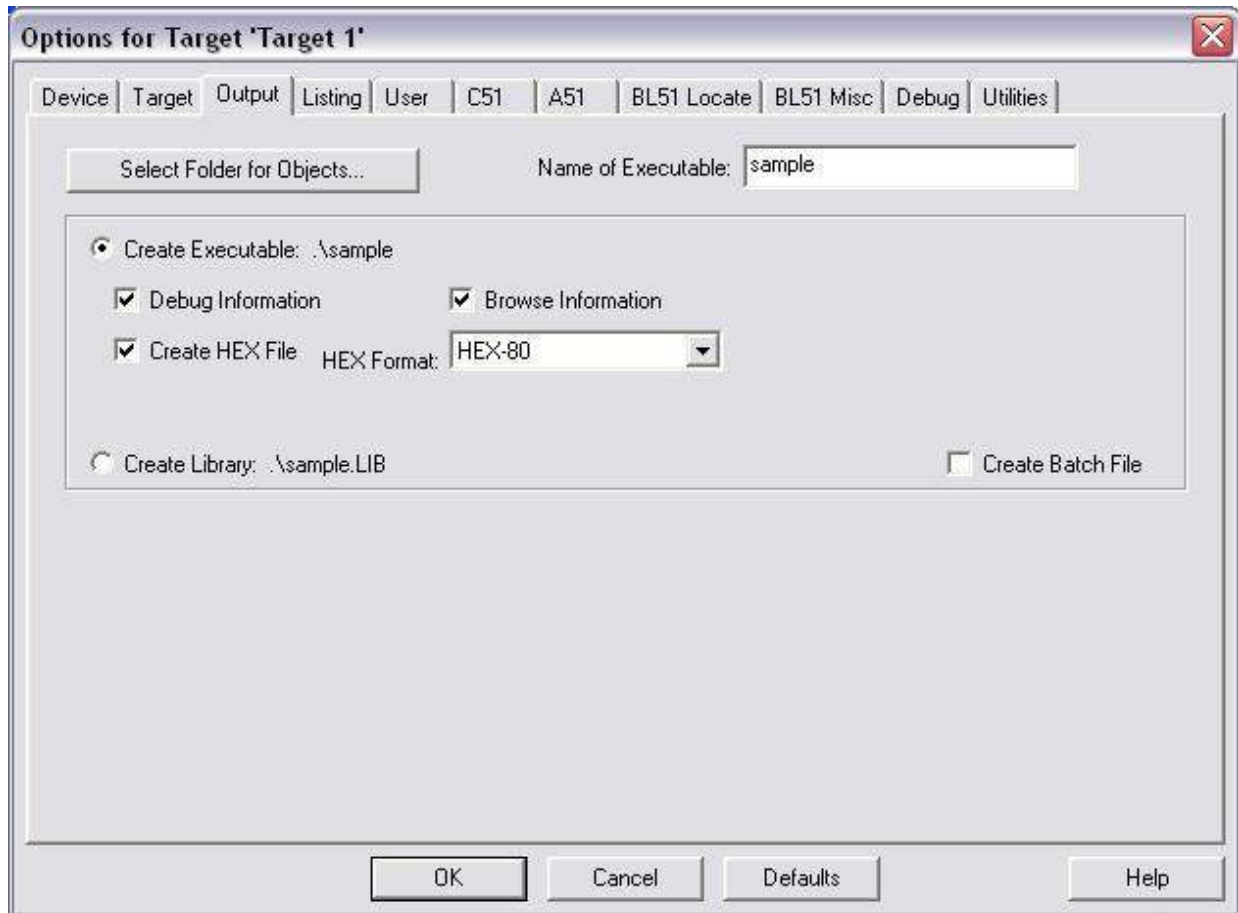
The following table describes the options of the Target dialog:

Dialog Item	Description
Xtal	Specifies the external clock frequency of your device. Several microcontrollers use an on-chip PLL to generate the CPU clock. In these cases the value is not identical with the XTAL frequency. Check your hardware design carefully to determine the correct value.
Operating System	Allows you to select a Real-Time Operating System for your project.
Use On-chip ROM / RAM	Defines the address spaces for the on-chip memory components for the linker/locator. Note that on some devices you need to reflect this configuration in the Startup Code.



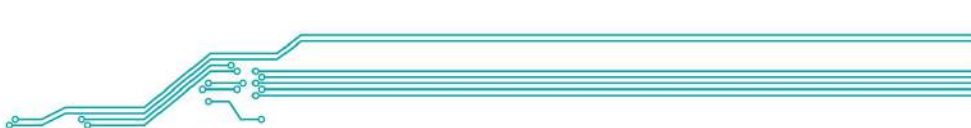
Create HEX File

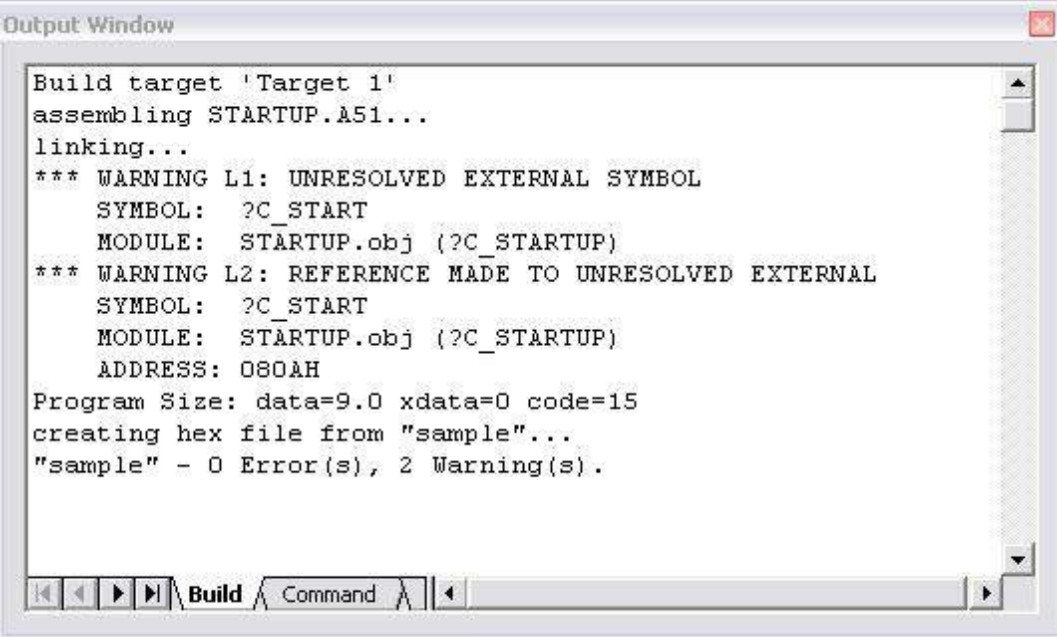
Once you have successfully generated your application you can start debugging. After you have tested your application, it is required to create an Intel HEX file to download the software into an EPROM programmer or simulator. µVision3 creates HEX files with each build process when Create HEX file under Options for Target – Output is enabled.



Build Project

Typical, the tool settings under Options – Target are all you need to start a new application. You may translate all source files and link the application with a click on the Build Target tool bar icon. When you build an application with syntax errors, µVision3 will display errors and warning messages in the Output Window – Build page. A double click on message line opens the source file on the correct location in a µVision3 editor window.





The screenshot shows a window titled "Output Window" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a text area with the following output text:

```
Build target 'Target 1'
assembling STARTUP.A51...
linking...
*** WARNING L1: UNRESOLVED EXTERNAL SYMBOL
    SYMBOL:  ?C_START
    MODULE:  STARTUP.obj (?C_STARTUP)
*** WARNING L2: REFERENCE MADE TO UNRESOLVED EXTERNAL
    SYMBOL:  ?C_START
    MODULE:  STARTUP.obj (?C_STARTUP)
    ADDRESS: 080AH
Program Size: data=9.0 xdata=0 code=15
creating hex file from "sample"...
"sample" - 0 Error(s), 2 Warning(s).
```

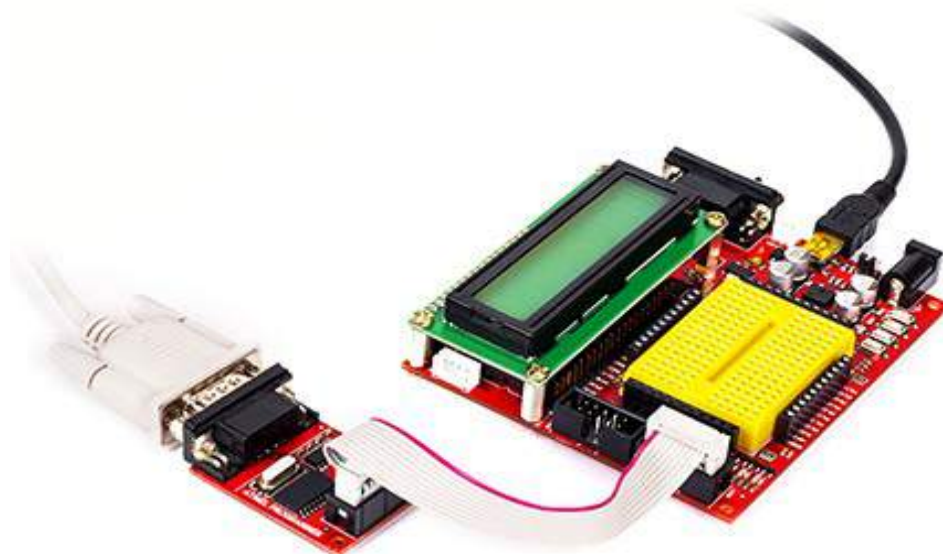
At the bottom of the window, there is a toolbar with several icons (back, forward, search, etc.) and a tab labeled "Build" with a "Command" input field.



3.2. CONNECTING THE DEVELOPMENT BOARD

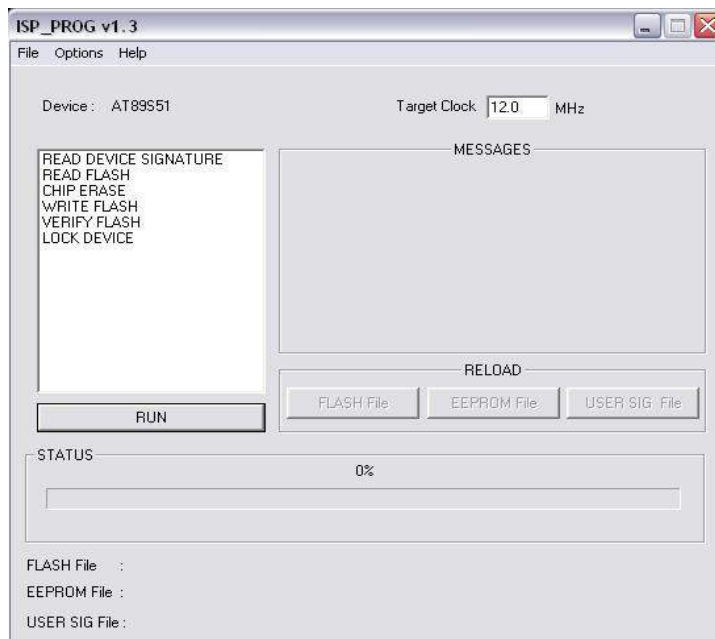
The AT89s5x requires a 9-12V adapter or USB connection or DC Supply for power and a serial connection for In-system programming. It also requires a separate programmer (available at www.rhydolabz.com/) for downloading the hex code. Use ISP_AT89S software at the PC side.

Connect AT89s5x development board to your PC using USB cable (for powering it) and serial cable (for In-serial programming) to the Programmer board as shown in the figure.



3.3. PROGRAMMING STEPS

1. Connect the Serial cable from the system to the Programmer board .
2. Connect the Programmer board to the eCee AT89S52 at the Programming Terminal labeled 'ATMEL' using the 10 pin BUS provided.
3. Power the AT89S52 board.
4. Open the ISP_AT89S.exe



5. Open the Hex file to be programmed. Select the Target clock frequency to 12.0 MH



6. Select the 'CHIP ERASE' option first. Then Click the 'RUN' button.
7. Now Select the 'WRITE FLASH' option and click the 'RUN' button.
8. After programming is completed, the development board can be disconnected from the Programmer and is ready to use.



I/O DISTRIBUTION

4.1. THE PIN DISTRIBUTION OF ATMEL AT89S51/52 DEVELOPMENT BOARD

AT89S52 Pin N0	Name	Type	The I/O Assign of AT89SS52 Development Board
1	P1.0	I/O	LCD (E)
2	P1.1	I/O	Unused
3	P1.2	I/O	LCD(RS)
4	P1.3	I/O	LED3
5	P1.4	I/O	LED2
6	P1.5	I/O	LED1 / ISP ATMEL (MOSI)
7	P1.6	I/O	ISP ATMEL(MISO) / Buzzer
8	P1.7	I/O	ISP ATMEL(SCK)
9	RST	-	Reset Switch
10	P3.0	I/O	ISP PHILIPS / RXD
11	P3.1	I/O	ISP PHILIPS / TXD
12	P3.2	I/O	Unused
13	P3.3	I/O	Unused
14	P3.4	I/O	EEPROM(SDA)
15	P3.5	I/O	EEPROM(SCL)
16	P3.6	I/O	PWM
17	P3.7	I/O	Unused
18	XTAL2	-	Crystal
19	XTAL1	-	Crystal
20	GND	-	Ground
21	P2.0	I/O	Pullup Key (SW1)
22	P2.1	I/O	Pullup Key (SW2)
23	P2.2	I/O	Pullup Key (SW3)
24	P2.3	I/O	Servo
25	P2.4	I/O	Unused
26	P2.5	I/O	Unused
27	P2.6	I/O	Unused
28	P2.7	I/O	Unused
29	PSEN	-	ISP



30	ALE/PROG	-	VCC
31	EA/VPP	-	VCC
32	P0.7	I/O	LCD(D7)
33	P0.6	I/O	LCD(D6)
34	P0.5	I/O	LCD(D5)
35	P0.4	I/O	LCD(D4)
36	P0.3	I/O	LCD(D3)
37	P0.2	I/O	LCD(D2)
38	P0.1	I/O	LCD(D1)
39	P0.0	I/O	LCD(D0)
40	VCC	-	+5V



TECHNICAL SUPPORT

If you are experiencing a problem that is not described in this manual, please contact us. Our phone lines are open from 9:00 AM – 5:00 PM (Indian Standard Time) Monday through Saturday excluding holidays. Email can be sent to support@rhydolabz.com

DISCLAIMER

Copyright © Rhydo Technologies (P) Ltd.

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice.

Contact Us

Rhydo Technologies (P) Ltd.

(An ISO 9001:2008 Certified R&D Company)

Golden Plaza, Chittoor Road, Cochin – 682018, Kerala, India.

Tel: 0091 484 2370 444, 2371 666, Cell: 0091- 99466 70444

Fax: 0091 484 237 0579, Email: info@rhydolabz.com, sales@rhydolabz.com

For more information please visit: www.rhydolabz.com

