

MINI DEVELOPMENT BOARD

AVR ATMEGA32

USERS GUIDE

make your own
intelligent embedded world ...

rhydolabz™

TABLE OF CONTENTS

CHAPTER – 1 : OVERVIEW	3
1.1 : Features	4
1.2 : Package Contents.....	5
1.3 : Microcontroller Specification	5
 CHAPTER - 2 : HARDWARE DEVELOPMENT.....	 7
2.1 : ATmega32/32L Function Module.....	8
2.2 : Interface overview	9
2.3 : Peripherals Description.....	10
2.4 : Jumper List.....	11
2.5 : Power Supply	12
2.6 : Clock Source	12
2.7 : Microcontroller - Pin out	13
2.8 : Port Pins.....	14
2.9 : LED Interfacing	15
2.10 : LCD Interfacing	15
2.11 : Pull-up Keyboard Interfacing.....	16
2.12 : Buzzer Interfacing	17
2.13 : Universal Synchronous Asynchronous Receiver Transmitter.....	18
2.13.1 : RS 232 Interface.....	18
2.13.2 : USB Interface.....	19
2.13.3 : 5V UART.....	20
2.13.4 : 3V3 UART.....	20
2.13.5 : Using RealTerm in PC	21
2.14 : Servo motor Interface.....	26
2.15` : ADC Interfacing.....	26
2.16 : Temperature sensor interface	27
2.17 : Zigbee module Interface.....	27
2.18 : RFID Interface	28
2.19 : ISP Connection.....	28
2.20 : JTAG Connection	29



CHAPTER -3 : SOFTWARE DEVELOPMENT.....30

3.1 : Familiarization of AVR studio.....	31
3.1.1 : Launch AVR Studio 4	31
3.1.2 : Create new Project File and Select CPU.....	32
3.1.3 : Create New Source Files.....	33
3.1.4 : Add Source Files to the Project.....	34
3.1.5 : Build Project and Create a HEX File.....	35
3.2 : Setting up mini AT Mega32/32L	39
3.3 : Programming Steps.....	40
3.4 : JTAG Programming/Debugging.....	42

CHAPTER-4 : I/O DISTRIBUTION.....43

4.1 : The Pin Distribution of ATMega32 mini Development Board.....	44
--	----

OVERVIEW

AVR mini Development Board is a miniature and powerful hardware platform to evaluate the Atmel ATmega32/32L Flash memory microcontroller. The AVR ATmega32/32L mini Board contains all hardware components that are required in a single-chip ATmega32 controller system and 1 COM port for serial RS232 interface.

1.1 : FEATURES

1. Compact and ready to use design
2. Includes ATmega32 Microcontroller with 8MHz Crystal Oscillator
3. No separate power adapter required (USB power source)
4. External power supply range of 7V to 12V
5. Adapter (any standard 7-12 power supply) option
6. On-board 5V 1A and 3V3 voltage regulator ICs
7. RS-232 interface(for direct connection to PC's serial port)
8. Built-in user programmable LEDs (3 nos.) to test I/O
9. On-board Reset Switch
10. On-board Buzzer interface
11. On-board Power LED indicator
12. On-board 3 Pull-up keys
13. On-board Temperature sensor
14. LCD Interface (16x2)
15. Berg strips available to interface Servo motor
16. Provision for interfacing Zigbee module
17. Provision for interfacing RFID module
18. On-board Prototyping area
19. All controller pins available at male and female berg strips
20. RMC connectors available in 5V and 3V3 for UART
21. On-board DB9 and USB Connector
22. RMC Connector and DC Barrel Jack for external power supply
23. JTAG Connector for debugging/programming
24. On-Board ISP Connector
25. Built-in potentiometer interface for ADC with variable voltage input
26. Power supply reverse polarity protection
27. Can be used as header board for developing applications
28. High quality two layer PTH PCB



1.2 : MINI ATmega32 PACKAGE INCLUDES

- Fully Assembled and Tested AVR ATmega32/32L mini Development board
- Software CDROM with
 - Schematic
 - Programming Software
 - Sample Hex Code
 - Example Codes for
 - Led Blinking
 - LCD Display
 - UART Communication
 - Buzzer interface
 - Pull-Up Keyboard interface
 - Servo motor interface
 - ADC Interface
 - Temperature sensor Interface
 - RFID Interface



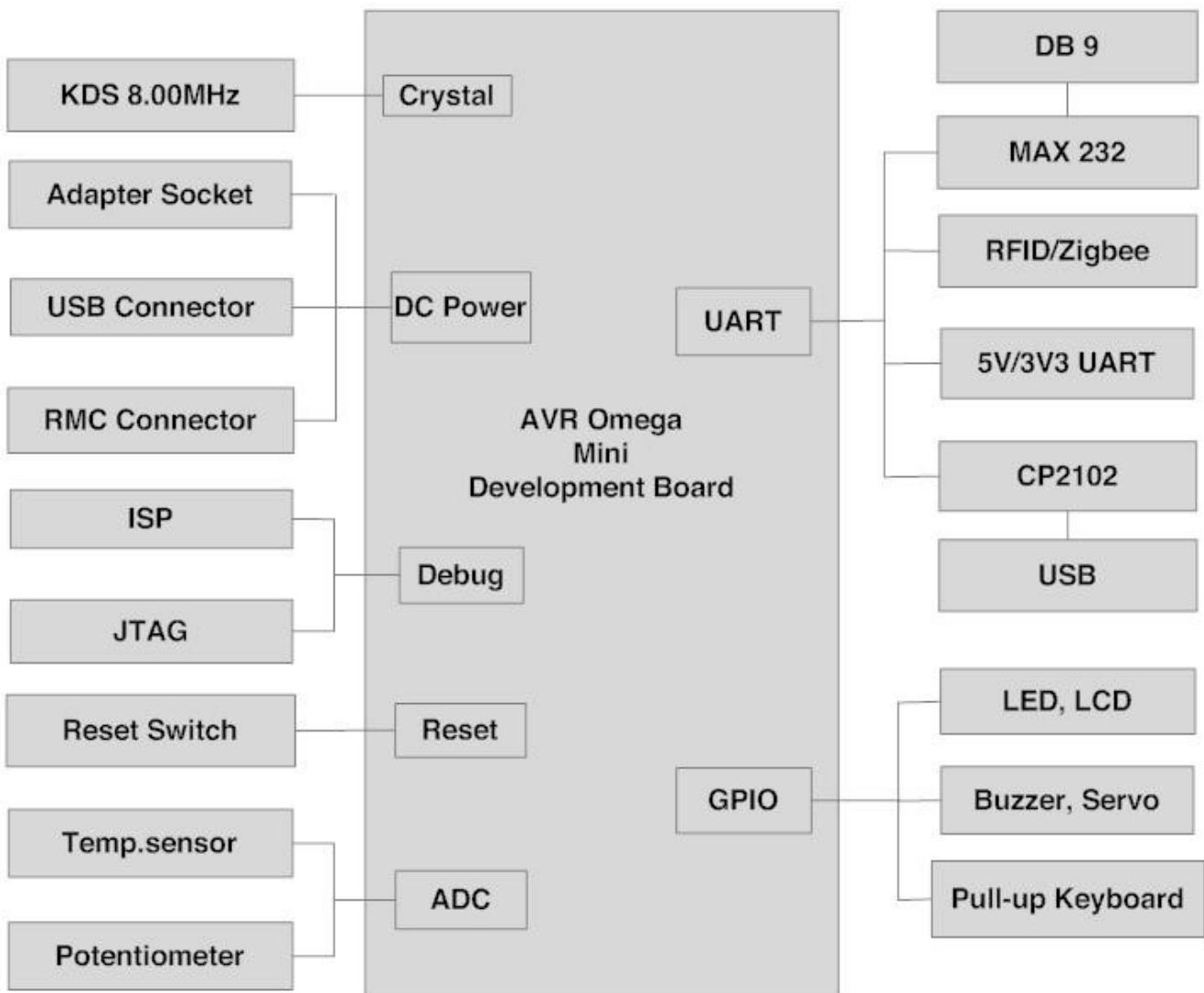
1.3 : ATmega32/32L SPECIFICATION

1. Atmel AVR ATmega32/32L microcontroller with 8 MHz Crystal Oscillator (With Boot loader software)
2. High Performance RISC CPU
3. 131 Powerful Instructions – Most Single-clock Cycle Execution
4. On-chip 2-cycle Multiplier
5. 32 K bytes programmable flash
6. 2 K bytes SRAM
7. 1024 bytes EEPROM
8. Up to 16 MHz operation (Up to 16MIPS for ATmega32)
9. Internal Calibrated RC Oscillator
10. 32 I/O pins
11. 8 channel 10 bit ADC
12. On-chip Analog Comparator
13. 4 PWM Channels
14. 4 Compare modules associated with each Timer
15. Two 8-Bit Timers/Counters
16. One 16-Bit Timer/Counter
17. Real Time Counter with Separate Oscillator
18. 1 Serial USART
19. 1 Two Wire Serial Interface TWI Module
20. 1 Serial Peripheral Interface (SPI - Master/ Slave) Module
21. Power Consumption ~ 15mA
22. Power-On Reset (POR), Power-Up Timer (PWRT) and Oscillator Start-up Timer (OST)
23. Interrupt Capability (up to 20 sources)
24. 32 x 8 General Purpose Working Registers
25. ICSP Programming
26. JTAG (IEEE std. 1149.1 Compliant) Interface
27. Brown Out Reset
28. Low Voltage Programming
29. Power Saving Sleep Mode
30. Programmable Code Protection
31. Optional Boot Code Section with Independent Lock Bits
32. Fully Static Design
33. Wide Operating Voltage 2.7V to 5.5V
34. Low Power Consumption

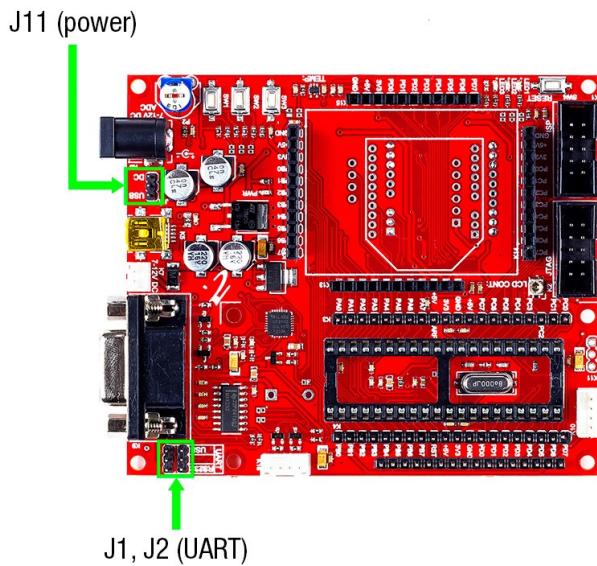


HARDWARE INTRODUCTION

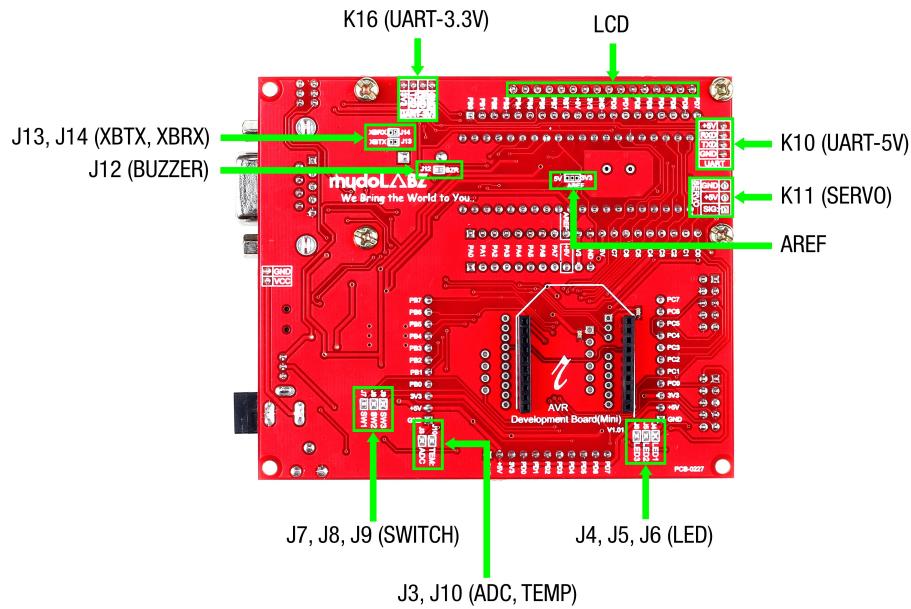
2.1 : FUNCTION MODULE



2.2 : INTERFACE OVERVIEW



Top view of the Development board -Mini



Bottom view of the Development board -Mini

2.3: PERIFERALS AND DESCRIPTION

PERIPHERALS	DESCRIPTION
ISP(K1)	ISP Connector,To program the IC
JTAG(K2)	JTAG Connector,To program the IC
K3,K4	PORT pins are available on male berg strip
K5	Mini USB Socket
K7	RMC connector for power
K8	Screw Terminal(DC Socket)
K9	USART Interface using Female DB9 Connector
K10	RMC connector for 5V UART
K11	Servo connector pin
K12-K15	PORT pins are available on female berg strip
K16	RMC connector for 3V3 UART
LED1-LED3	Light Emitting Diodes
PWR	Power indicator
SW1-SW3	Pull-Up Switches
RESET(SW4)	Reset Button
LCD CONT(P1)	LCD Contrast Adjust through Pot
ADC(P2)	Potentiometer as ADC input
U1	ATmega32
U2	LD1117 3V3
U3	LM7805(Regulator IC(5V))
U4	CP2102(USB interface)
U5	MAX232(Level converter)
U6	Temperature Sensor (MCP9700)
U7	Zigbee module
M1	connector for EM-18 RFID Module



2.4 : JUMPER LIST

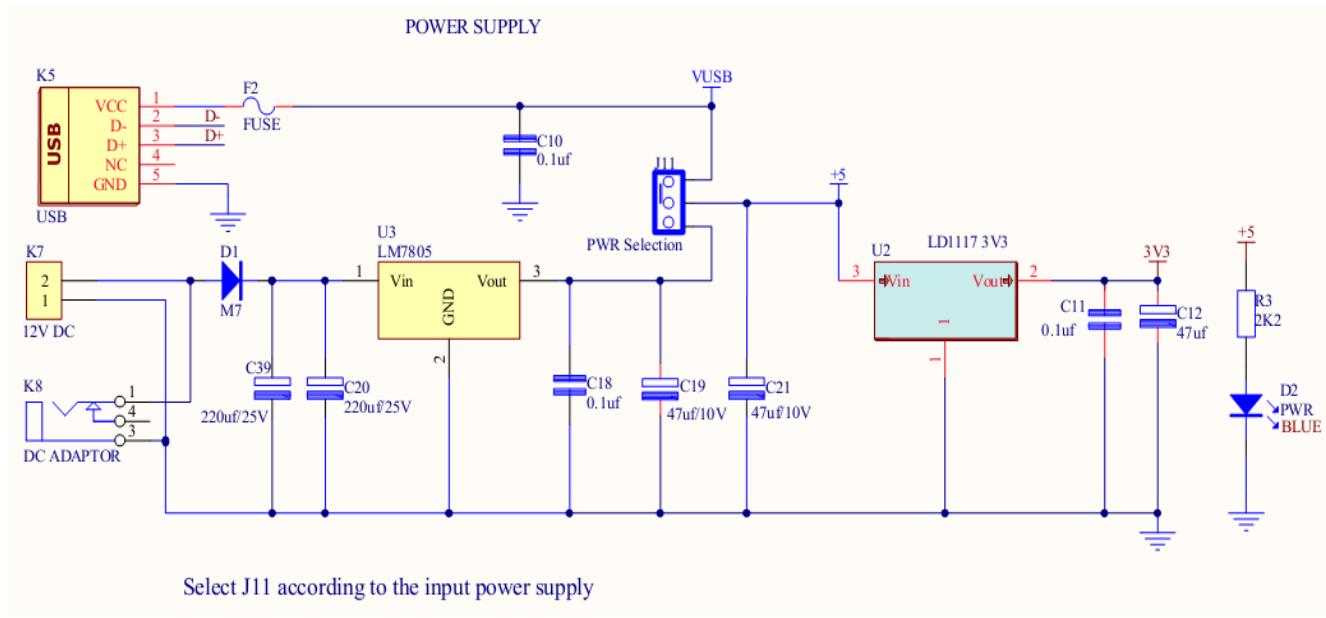
JUMPER NO	DESCRIPTION	SET OPTION	SET DESCRIPTION
J1-J2	UART SELECTION	1-2 2-3	RS232 USB
J3	ADC	Potentiometer	Potentiometer Output connected to PA0
J4-J6	LED	Short to access	LED access via PC0-PC2
J7-J9	Matrix keys	Short to access	Matrix keys access via PC0-PC2
J10	Temperature sensor	Short to access	MCP9700 Access via PA1(channel 1)
J11	Supply Select	1-2 2-3	USB Power is selected DC Power is selected
J12	Buzzer	Short to access	Buzzer can be access via pin PC7
J13-J14	Zigbee Tx & Rx	Short to access	Can be access via UART

2.5 : POWER SUPPLY

You can power the Atmel AVR ATmega32/32L mini Board by one of the following ways-

- Through the on board USB port (5V)
- Through 2 Pin RMC Connector (7V - 12V External DC Power Supply)
- Through Barrel Jack Connector (7V - 12V External DC Power Supply)

The external Power Supply circuit is given below:

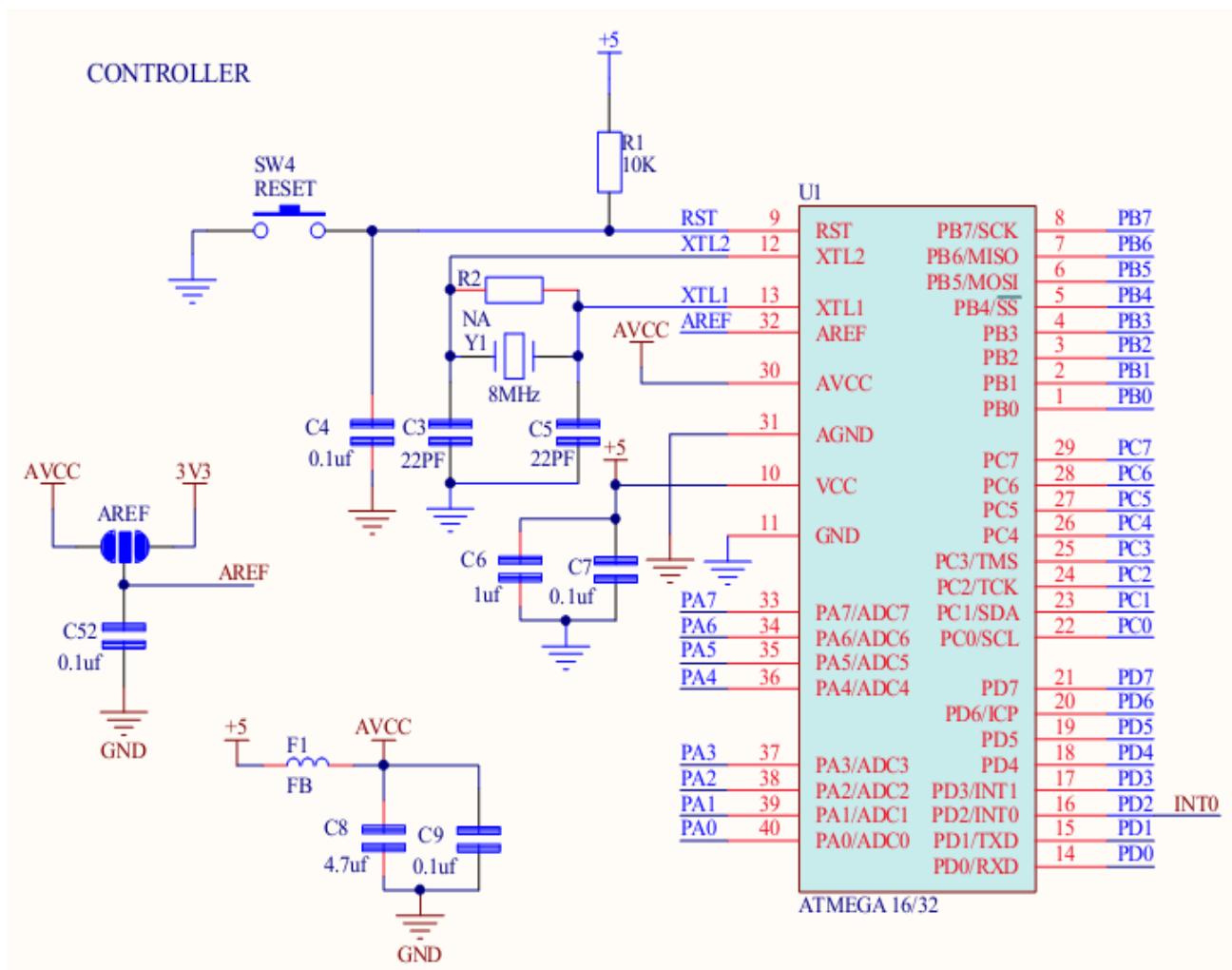


2.6 : CLOCK SOURCE

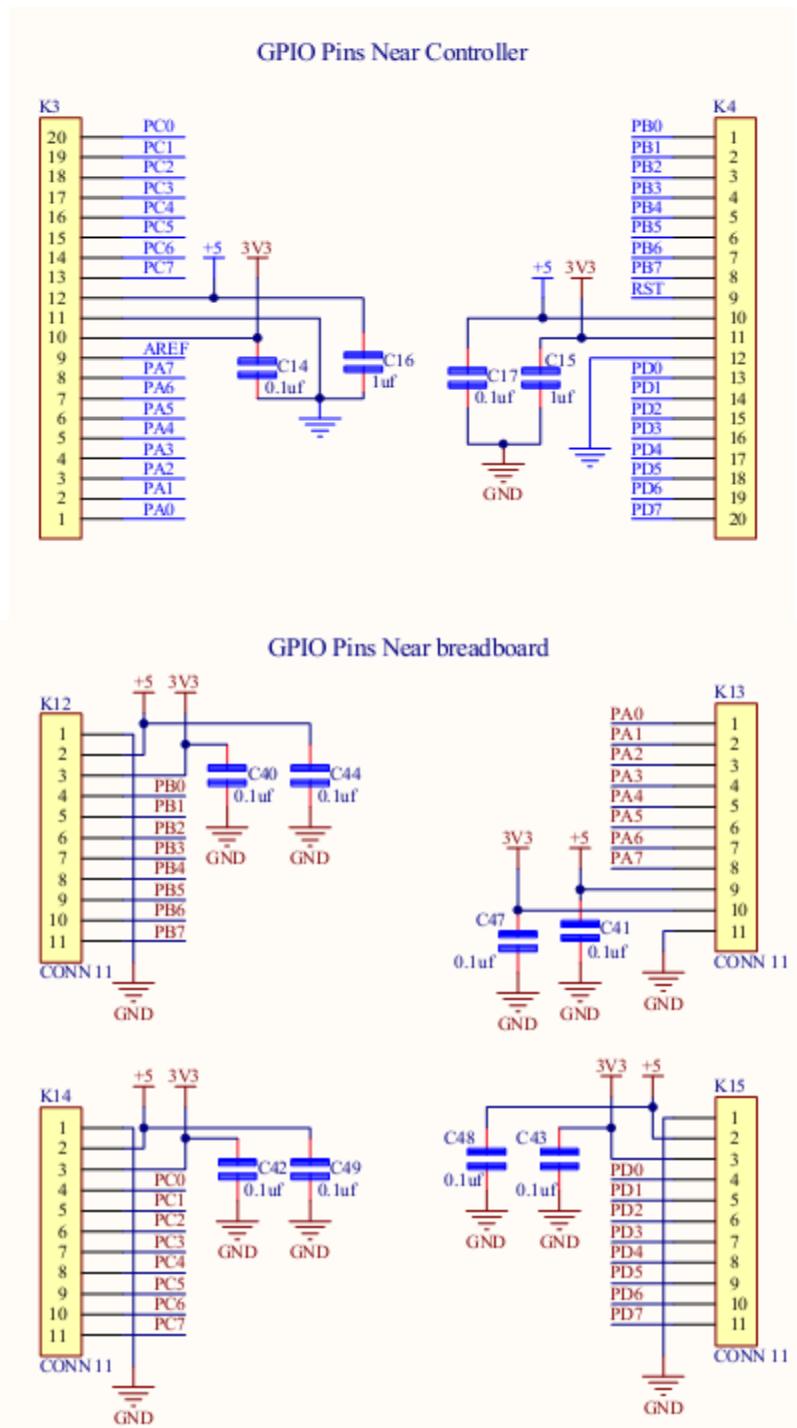
AVR ATmega32 Mini Board uses:

- 8 MHz Crystal as the MCU clock source

2.7 : MICROCONTROLLER - PIN OUT

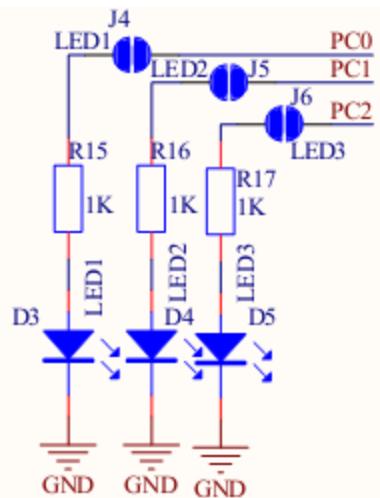


2.8 : PORT PINS



2.9 : LED INTERFACING

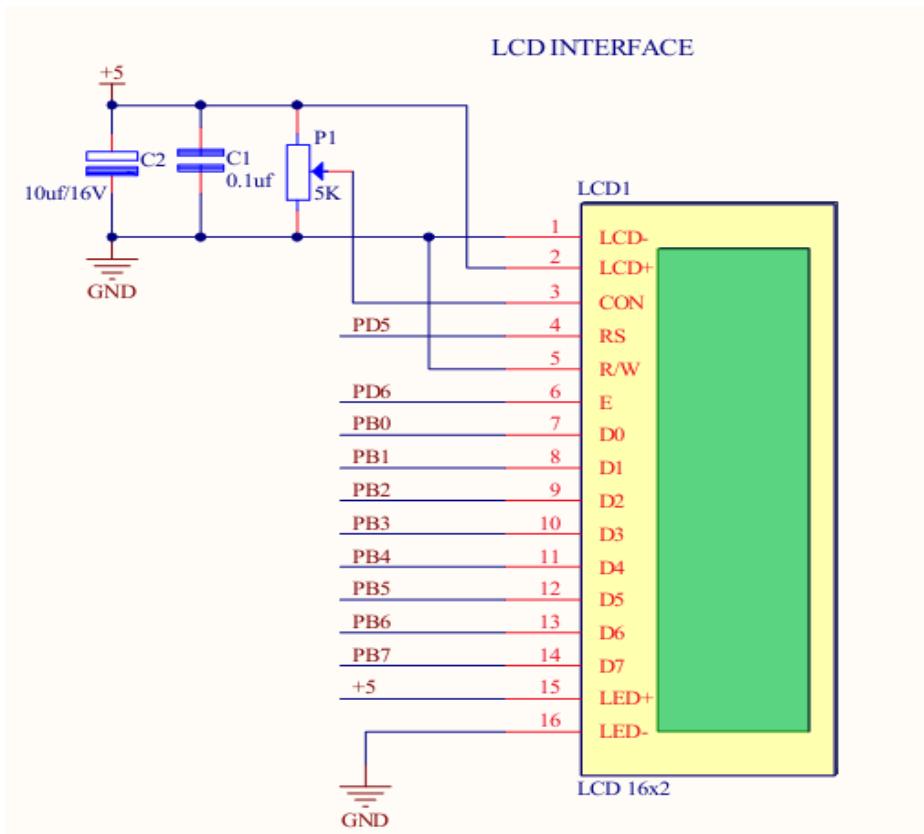
LEDs are the simplest devices to test port functioning. The board contains 3 LEDs which are connected to PC0, PC1 and PC2 pins of the AVR ATmega32 through jumpers J4, J5 and J6. Each LED is interfaced using a current limiting resistor.



Note: Remove J4, J5 & J6 when PC0, PC1 & PC2 are used for other purposes.

2.10 : LCD - LIQUID CRYSTAL DISPLAY

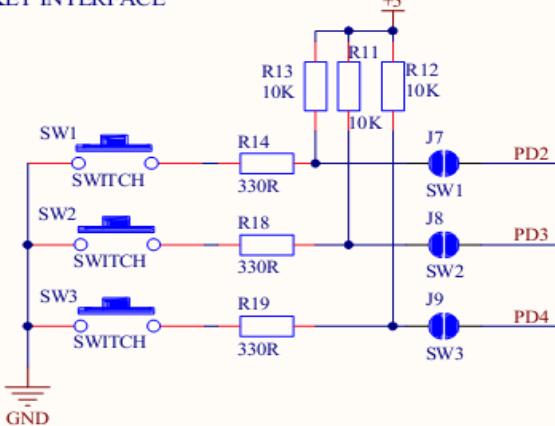
The display is a standard 16x2 LCD which displays 2 lines of 16 characters. Each character is 40 pixels, making the LCD display 1280 pixels overall. The display receives ASCII codes for each character at the data inputs (D0–D7). The data is presented to the display inputs by the MCU, and latched in by triggering the E (Enable) input. The RW (Read/Write) line can be tied low (write mode), as the LCD is receiving data only. The RS (Register Select) input allows commands to be sent to the display. RS selects command/data mode. The display itself contains a microcontroller; the standard chip in this type of display is the Hitachi HD44780. It must be initialized according to the data and display options required. The module can be used in 4-bit or 8-bit mode. The development board uses 8-bit interface. PORTB pins are used for data/command pins and PORTD for RS, E pins. The R/W pin is grounded. LCD contrast can be adjusted by means of the potentiometer to get a suitable visible contrast.



2.11 : PULL-UP KEYPAD

The simplest input is a switch or push button. 3 externally pulled-up pins are incorporated to form a pull-up key pad. When the switch is open, the output voltage of the circuit is pulled up to 5V using a resistor. The pins used for this are PD2, PD3 and PD4 of the AVR ATmega32/32L through jumpers J7, J8 and J9. User can access the switch status by reading the PIND register.

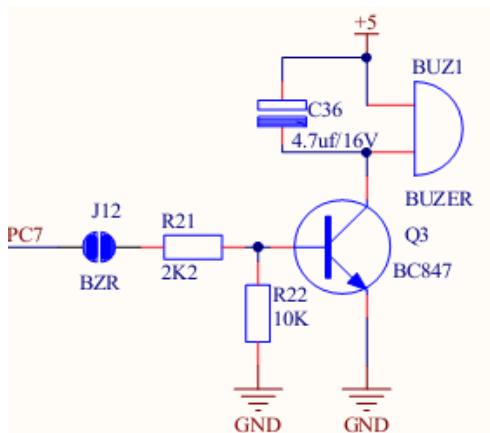
PULLUP KEY INTERFACE



Note: Remove J7, J8 and J9 when PD2, PD3 and PD4 are used for other purpose.

2.12 : BUZZER INTERFACING

Buzzers are another simple means of checking the port functioning. Normally we use piezo electric elements as buzzers. Buzzer is driven using a simple NPN transistor with a biasing. The transistor's base is connected to the I/O pin PC7 of the microcontroller. If the port pin is configured as an output port and a logical high is given to that port, the transistor will be triggered on, which in turn switches on the Buzzer. If logical 0 is provided, the buzzer will be turned off.



Note : In order to sound the buzzer, the jumper J12 should be kept in position. Remove J12 if PC7 is to be used for other purposes.

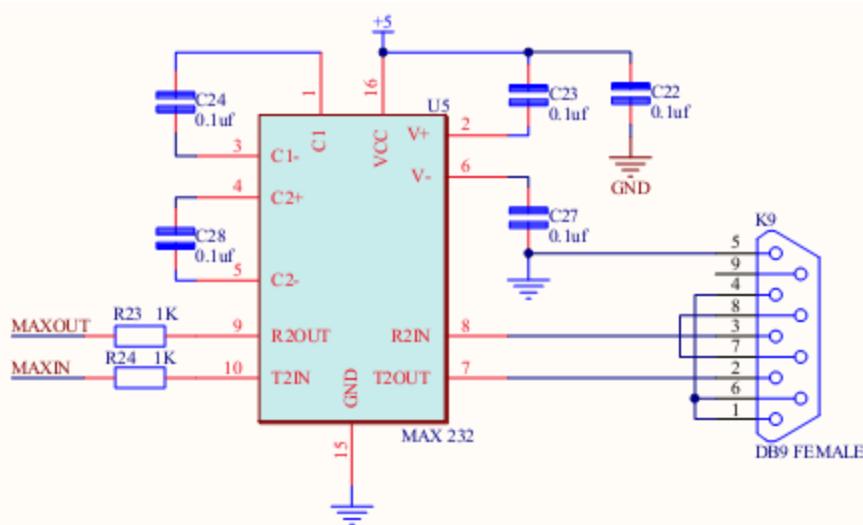
2.13 : UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The AVR ATmega32/32L UART module operates through PD1(TXD) and PD0(RXD) pins. The board has four serial communication interfaces; you can choose one of the following ways to communicate with external modules/PC.

- RS232 Interface via DB9 connector
- USB Interface using CP2102
- 5V UART
- 3V3 UART

2.13.1 : RS232 Interface

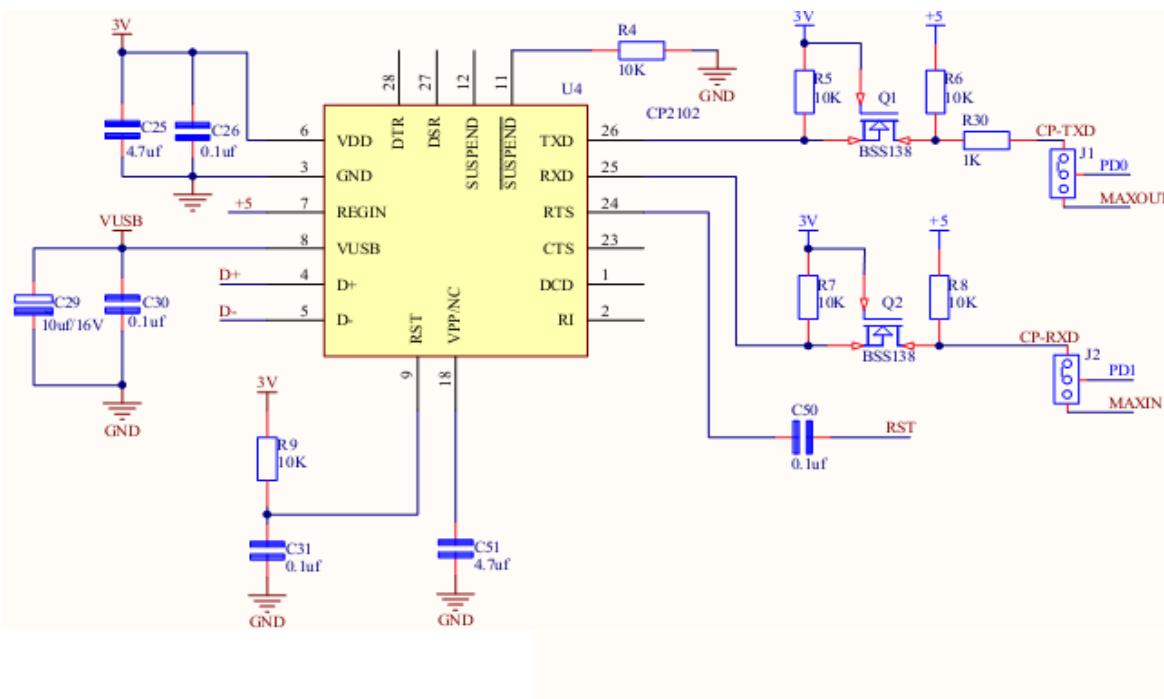
The RS232 interfacing is done using the serial driver MAX 232 and a DB9 connector. The MAX232 is an IC that converts signal from RS232 serial port to signal suitable for use in TTL compatible digital logic circuit. The MAX 232 is a dual driver/receiver and typically converts RX, TX, CTS and RTS



Note: Use proper jumper (J1&J2) selection when using RS232

2.13.2 : USB Interface

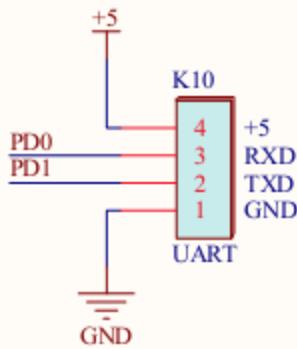
USB interfacing is done using a CP2102 IC. The CP2102 is a highly-integrated USB-to-UART Bridge Controller providing a simple solution for updating RS-232 designs to USB using a minimum of components and PCB space. The CP2102 includes a USB 2.0 full-speed function controller, USB transceiver, oscillator, 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.



Note: Use proper jumper (J1&J2) selection, while using USB. Remove J1&J2, if PD0 & PD1 are to be used for other purposes. Remove J1 & J2 when you connect Zigbee or RFID module.

2.13.3 : 5V UART

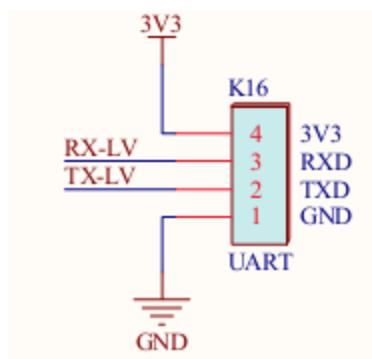
The 5V UART can be used for the communication between 5V TTL modules and the microcontroller .



Note: Remove J1 & J2 when K10 is used for UART communication.

2.13.4 : 3V3 UART

The 3V3 UART can be used to interface 3V3 TTL modules. The ATmega32 development board has a built-in 3V3 to 5V (and vice versa) converter for communicating between the 3V3 devices and the microcontroller.



Note: Short J13 & J14 when using 3V3 UART. Remove J1&J2 when connecting to 3V3 UART



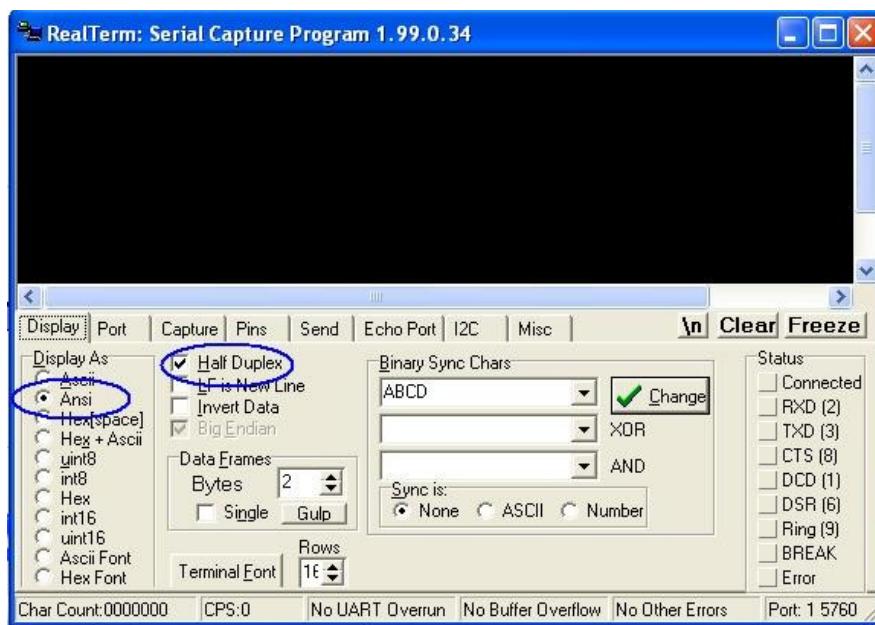
2.13.5 : USING REALTERM IN PC

The serial data transmitted through USART can be viewed on a PC using a Windows tool for Serial Port Communication called RealTerm

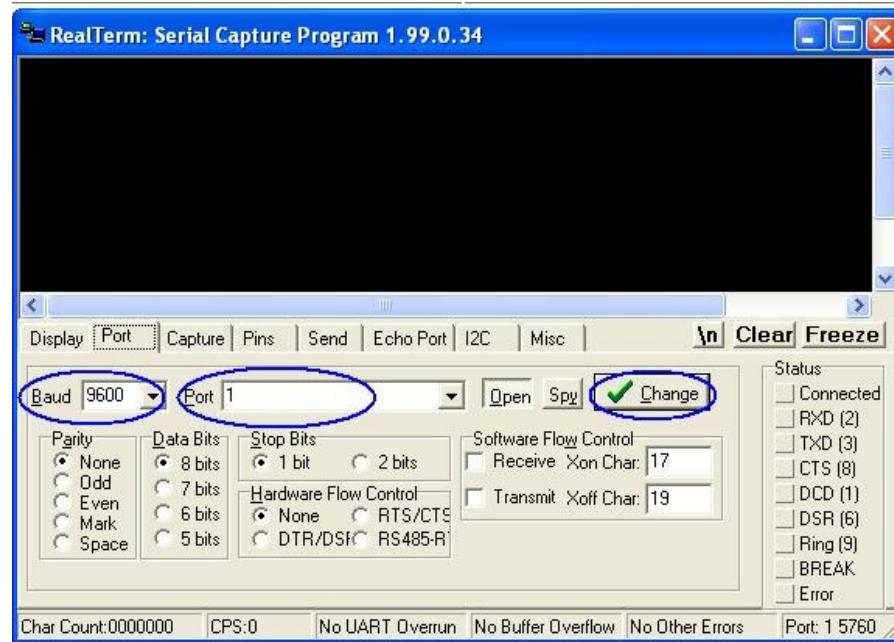
Step 1 : Open **realterm** from start menu or Desktop shortcut



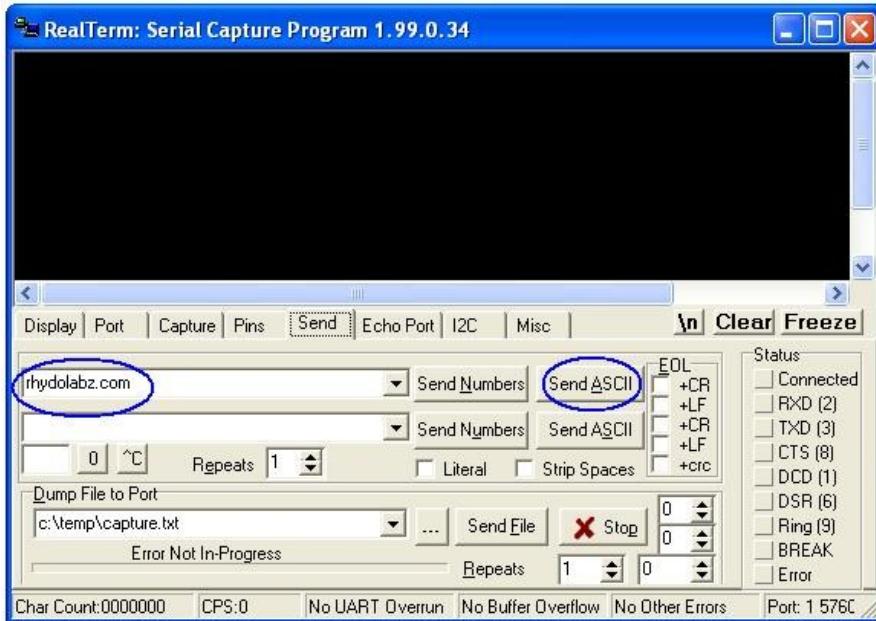
Step 2 : Display Tab- Here the output text format selected is ANSI and Half Duplex mode is enabled to view the data sent by the user.



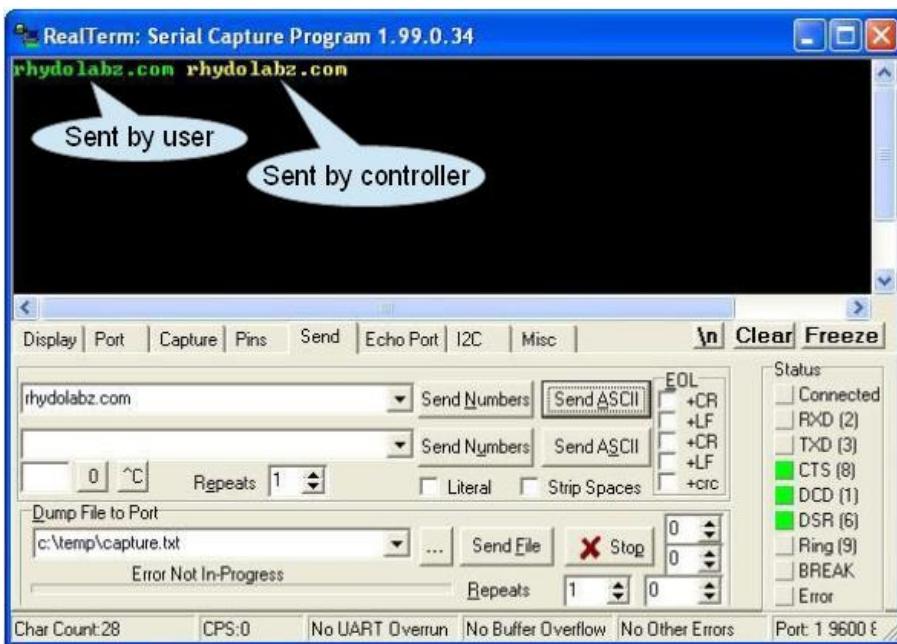
Step 3: Port Tab- To test the connection - make sure the **Open** button is pressed, Select required baud rate and the "Port" dropdown here, select the number of your COM port and then press the **Change** button.



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

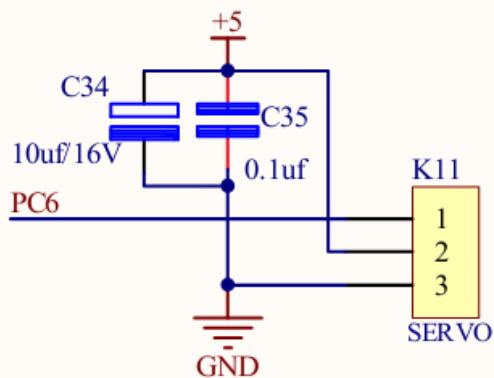


Step 5 : The output after data transmission to the controller is shown in the following diagram. The text sent by user and controller are highlighted by callouts in the figure.



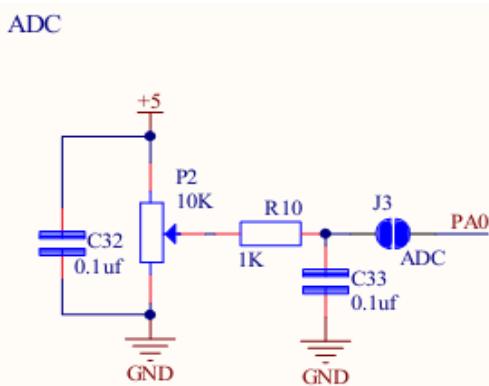
2.14 : SERVO MOTOR INTERFACING

Servos are small mechanical motorized devices which rotate a tiny shaft attached to a servo wheel in a specified position. Servos are controlled by sending a pulse width signal from an external electronic device that generates PWM signals. PWM signal sent to the servo are translated into position values by electronic components inside the servo. Here the servo is connected to the pin PC6.



2.15 : ADC INTERFACING

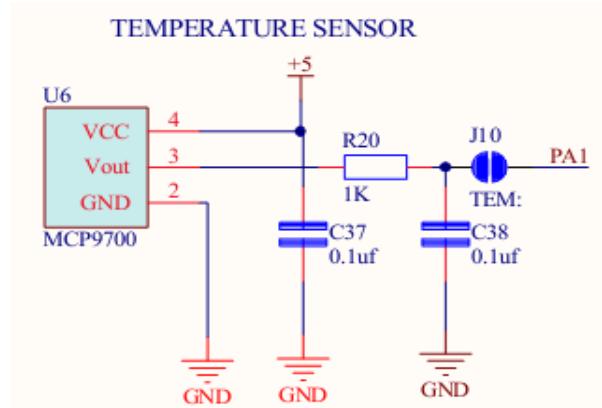
AVR ATmega32/32L mini development board has a potentiometer connected to its ADC pin PA0 (channel 0).



Note: Place jumper on A2DIN (J9) pin

2.16 : TEMPARATURE SENSOR INTERFACING

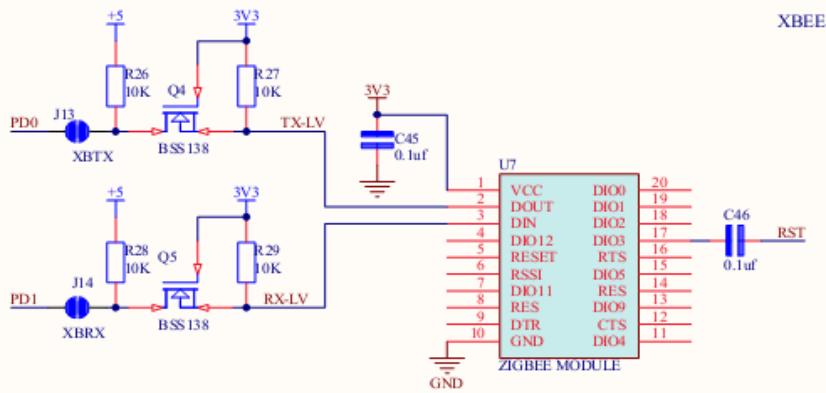
LM35 temperature sensor can be used to measure environment temperature, in the range of -55 Deg C to 150 Deg C. It is connected to channel 1 of PORTA (PA1).



Note: For the temperature sensor to work, the TEMP jumper (J10) must be in position.

2.17 : ZIGBEE MODULE INTERFACING

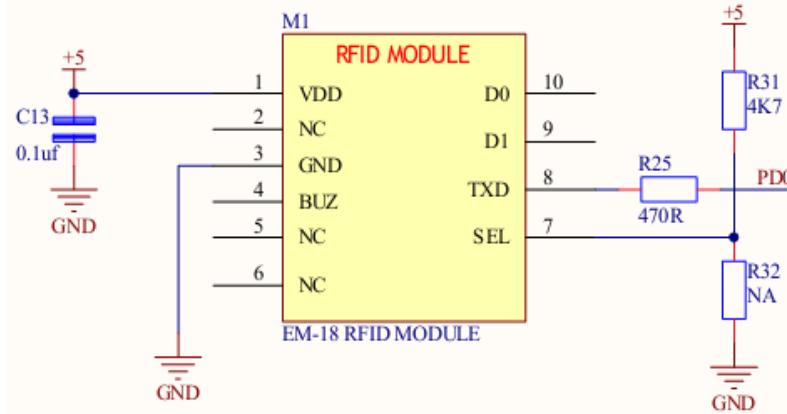
ATmega32/32L mini development board has pin out compatibility for zigbee reader interfacing. Communication between the RFID reader and controller uses a standard UART interface compatible with 3V3.



Note: Short J13 & J14 when using 3V3 UART. Remove J1 & J2 when connecting to Zigbee.

2.18 : RFID INTERFACING

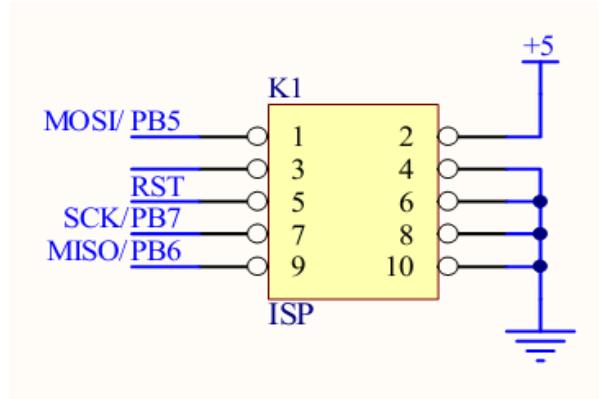
ATmega32/32L mini development board has pinout compatibility for micro RFID module interfacing. Communication between the RFID reader and controller uses a standard UART interface compatible with 5V.



Note: Remove J1 & J2 when connecting RFID module

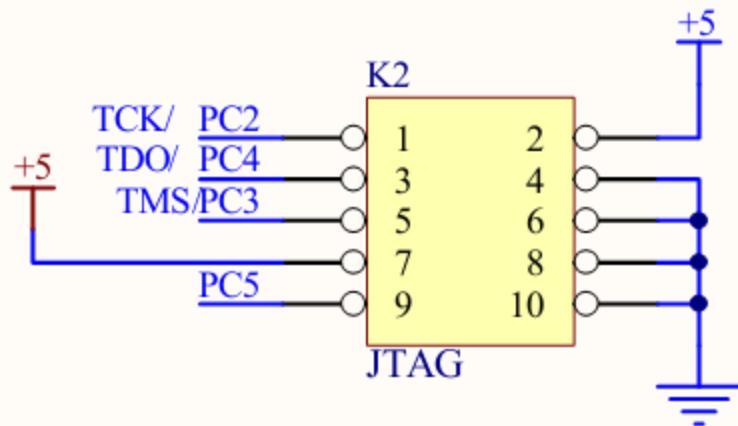
2.19 : ISP CONNECTION

ISP programming uses the Atmel AVR internal SPI(Serial Peripheral Interface) to upload the program into the FLASH and EEPROM memory of AVR microcontroller. A minimum of six wires required to connect the target board: VCC,GND,Reset and three signal lines. The pins used for the signal lines are PB5, PB6 and PB7 of the AVR ATmega32.



2.20 : JTAG CONNECTION

JTAG Connector provides a standard JTAG interface, allowing users to connect and program the controller. A minimum of six wires required to connect the JTAGICE to the target board: VCC,GND, and four signal lines. The pins used for the signal lines are PC2, PC3,PC4 and PC5 of the AVR ATmega32.



Note: Disable the JTAG for normal I/O operation of PORTC pins

SOFTWARE DEVELOPMENT

3.1 : FAMILIARIZATION OF AVR STUDIO IDE

- Open **AVR Studio 4** from start menu or Desktop shortcut
- Create new Project File and Select CPU.
- Create New Source Files.
- Add Source Files to the Project.
- Set Tool Options for Target Hardware.
- Create a HEX File.
- Build Project and Generate Application Program Code.



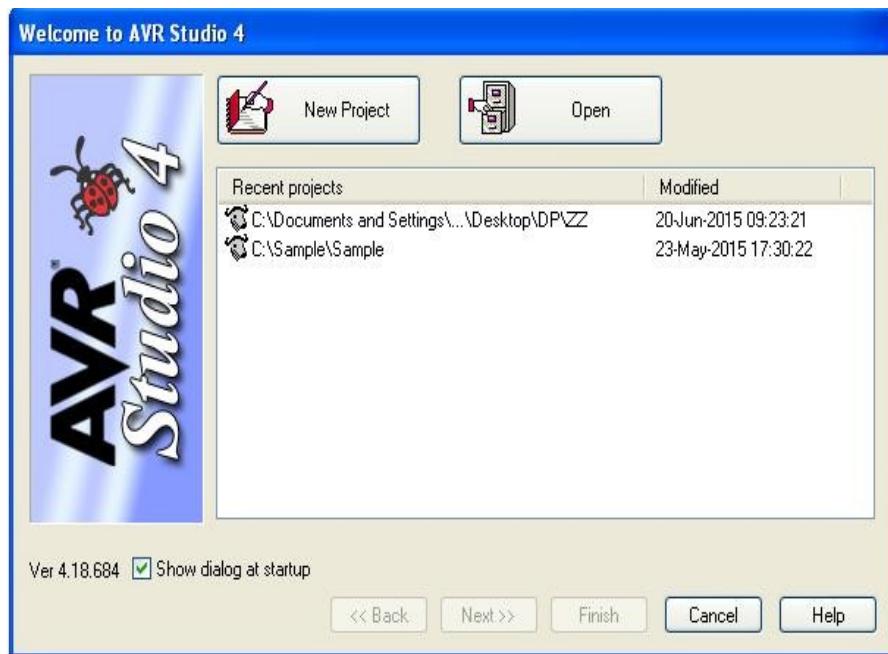
3.1.1 : Launch AVR Studio



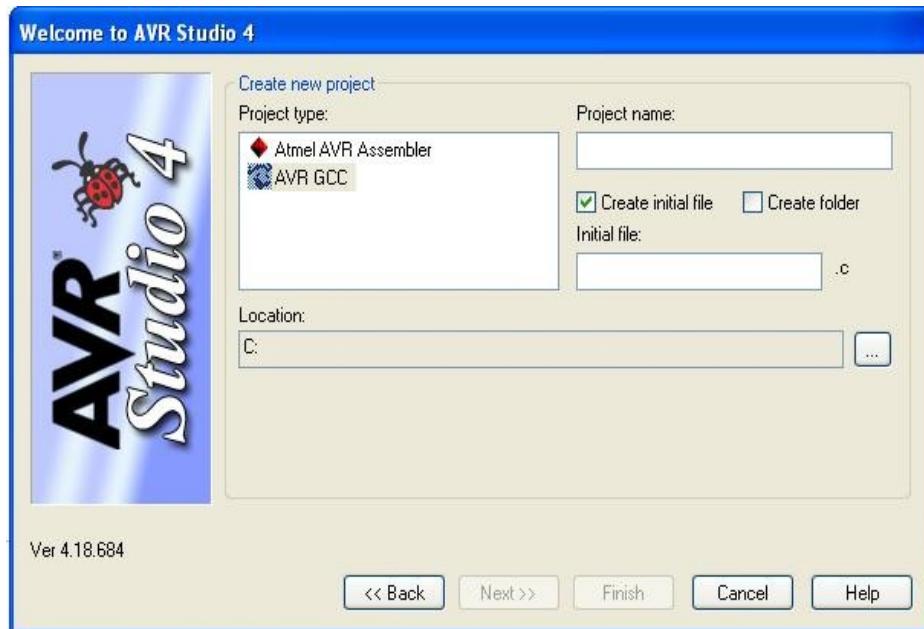
3.1.2 : Create new project file and select CPU.

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

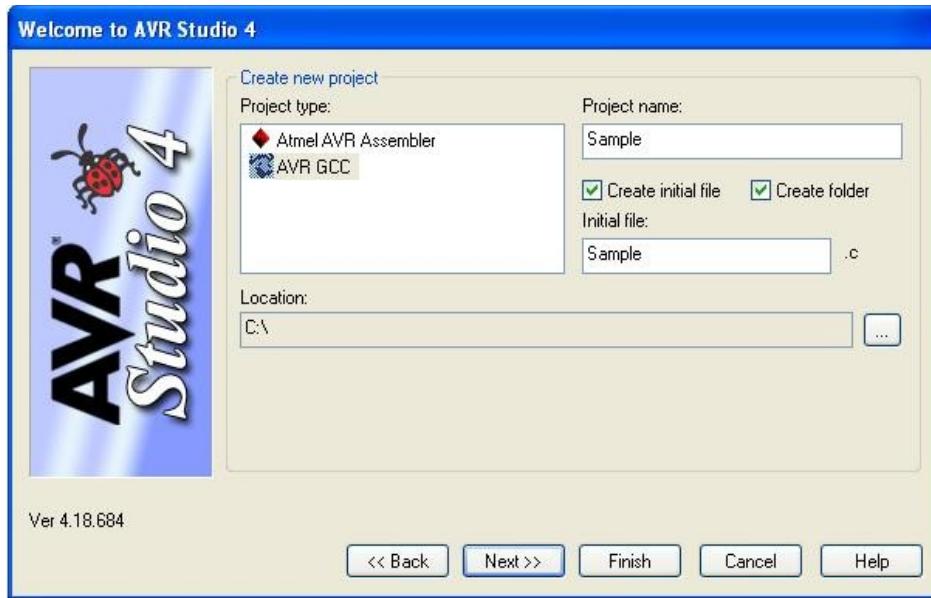
Step 1 : Click *New Project*.



Step 2 : Under *Project Type* select *AVR GCC Compiler*(for C program). Select *Create Initial file* option.

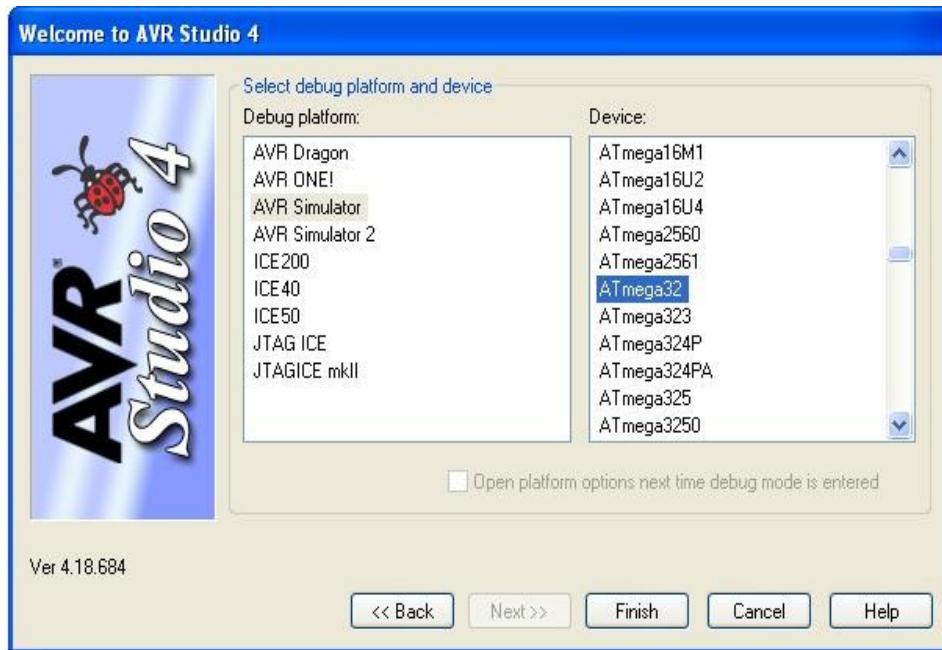


Step 3 : Give **Project name , Initial file name** and location to save your project and click **Next>>**.



Note: You can use the icon "Create New Folder" in this dialog to get a new empty folder inside the selected folder. If the "Create Initial file" is selected, a '.c' file will be created in the same working folder and will be added to the project work space. Enter the source code in the '.c' file. If the "Create Initial file" is not selected, select File-New File option to create a new file and write the program in this file.

Step 4 : Select the **Debug platform** and the **Device** (From the Device list). Click **Finish**.



Note: Select the debugger platform as AVR Simulator. While you create a new project, the AVR Studio asks you to select a CPU for your project. The Device section shows the AVR Studio device database. Select the microcontroller you use (For example; in this chapter we use the AVR ATmega 32 controller). Now click Finish to complete the project creation.

3.1.3 : Create New Source Files

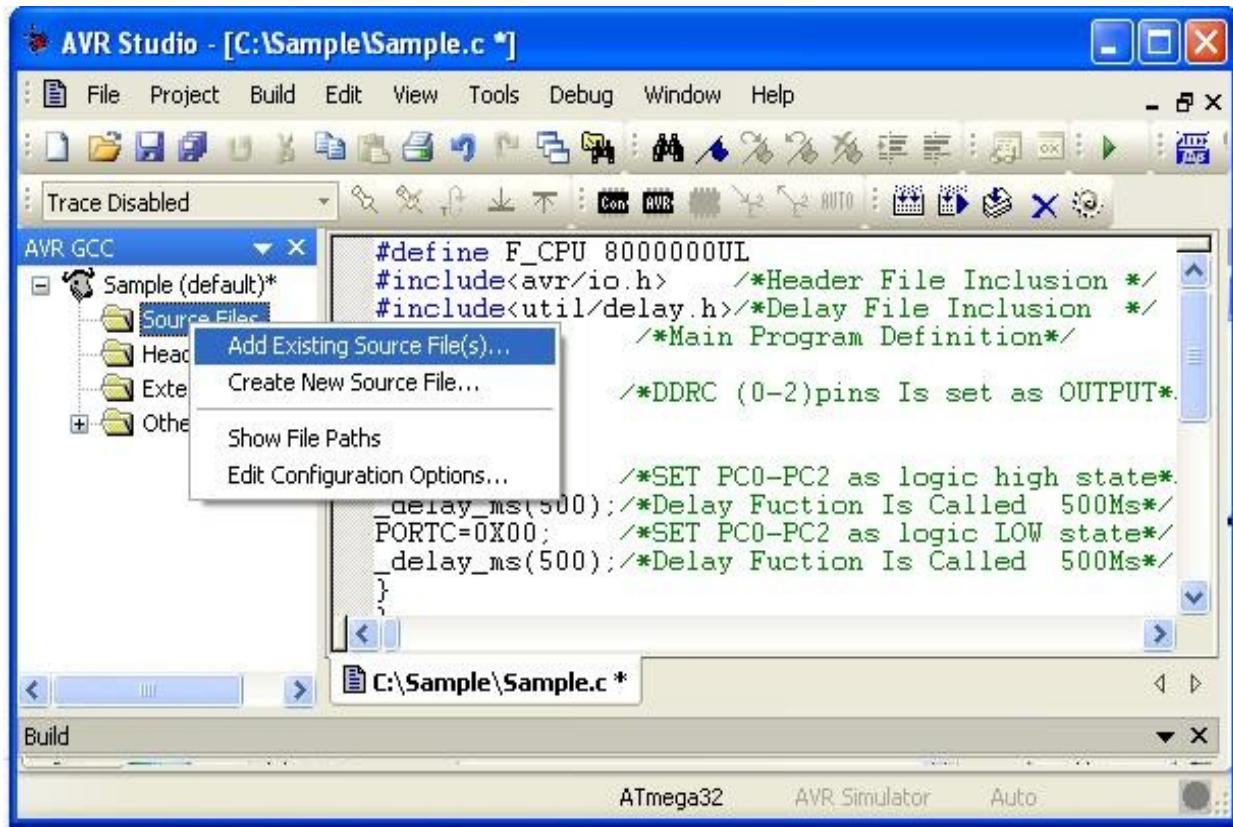
A new source file will be opened after completing the project creation since we selected the option *Create Initial File*, and will be automatically added to the project source files. You can create a new source file with the menu option *File – New* if you deselect the option to *Create Initial File* during the project creation. This opens an empty editor window where you can enter your source code. Save this file in the work folder, where the project file is saved. AVR Studio enables the colour syntax highlighting when you save your file using the *File – Save As* dialog with the extension '.c'.

Here we save our example file under the name Sample.c

```
#define F_CPU 8000000UL
#include<avr/io.h>      /*Header File Inclusion */
#include<util/delay.h>    /*Delay File Inclusion */
int main()                /*Main Program Definition*/
{
    DDRC=0X07;           /*DDRC (0-2)pins Is set as OUTPUT*/
    while(1)
    {
        PORTC=0X07;       /*SET PC0-PC2 as logic high state*/
        _delay_ms(500);   /*Delay Function Is Called 500Ms*/
        PORTC=0X00;       /*SET PC0-PC2 as logic LOW state*/
        _delay_ms(500);   /*Delay Function Is Called 500Ms*/
    }
}
```

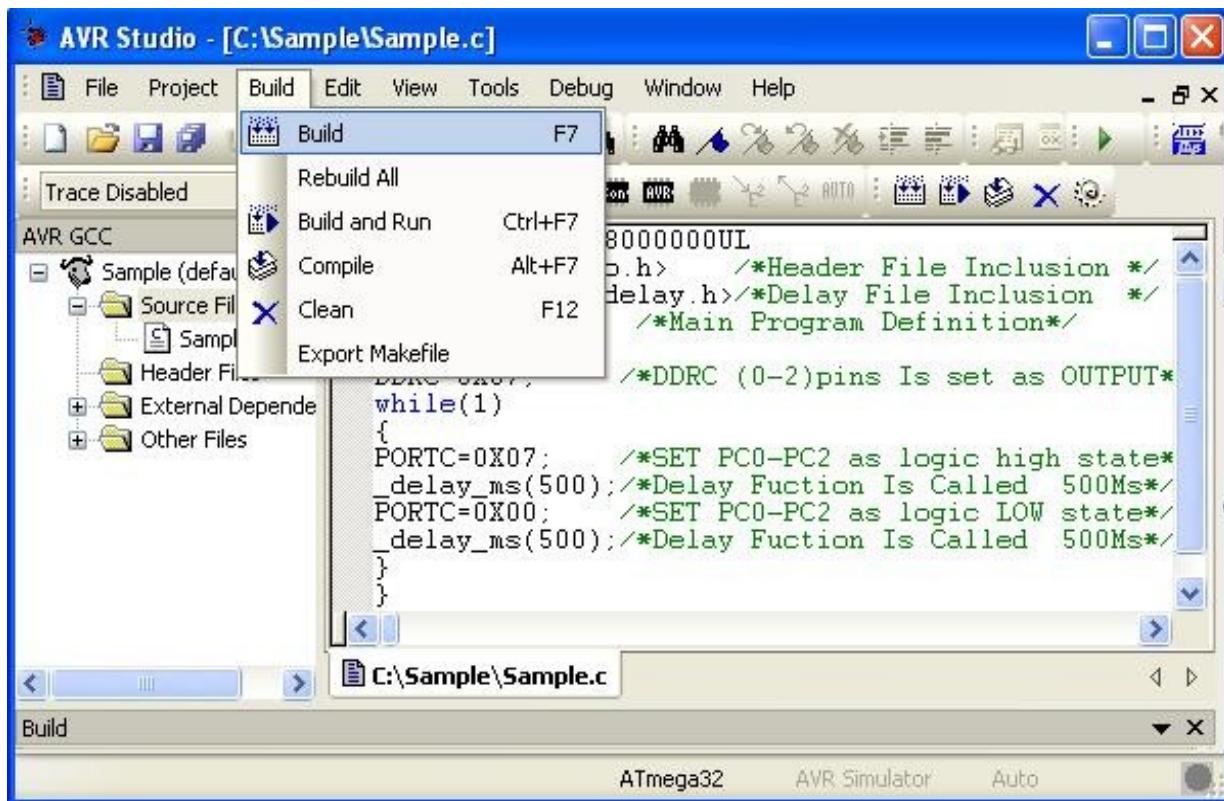
3.1.4 : Add Source Files to the Project

Once the source file is created, add this file to the project. AVR Studio offers several ways to add source files to a project. For example, you can right click **Source Files** in the **AVR GCC** and add files using the option **Add Files**.



3.1.5 : Build project and Create HEX File

Once you have successfully completed the program, you can start debugging. After debugging, an Intel HEX file must be created to upload the software into an EEPROM programmer or simulator. AVR Studio creates HEX files with each build process (Build option can be accessed from the Build Menu).



Now the corresponding hex file of the program will be generated inside the working folder in the name of the project that we created. When you build an application with syntax errors, AVR Studio 4 will display errors and warning messages in the **Build page**. A double click on a message line opens the source file in the correct location in the editor window. In this example the Hex file is **Sample.hex**. This file will be available in the folder Sample.

```
Build started 20.6.2015 at 12:39:17

AVR Memory Usage
-----
Device: atmega32

Program:      156 bytes (0.5% Full)
(.text + .data + .bootloader)

Data:         0 bytes (0.0% Full)
(.data + .bss + .noinit)

Build succeeded with 0 Warnings...
```

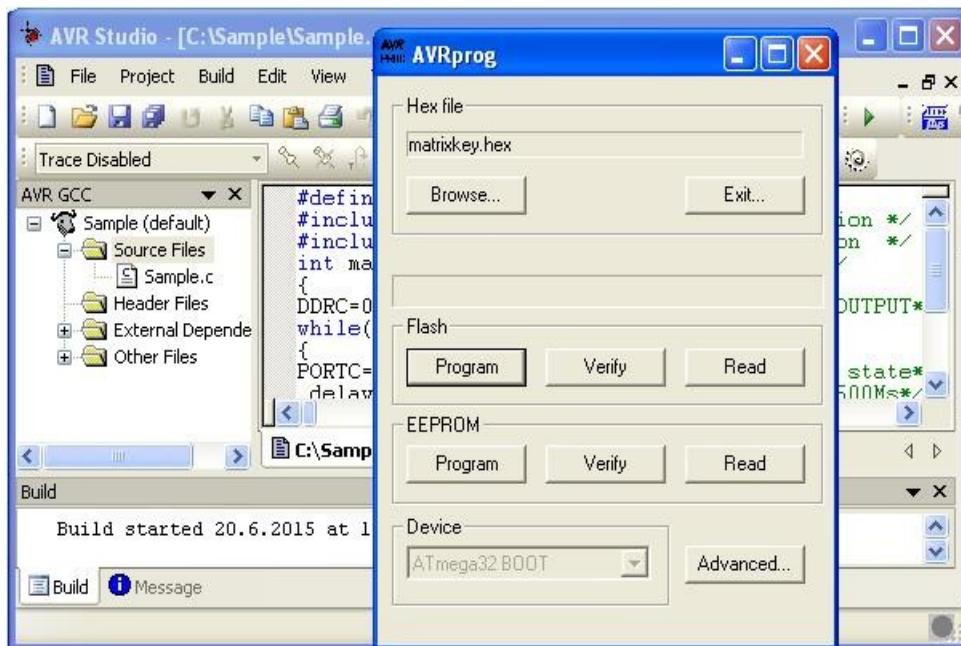
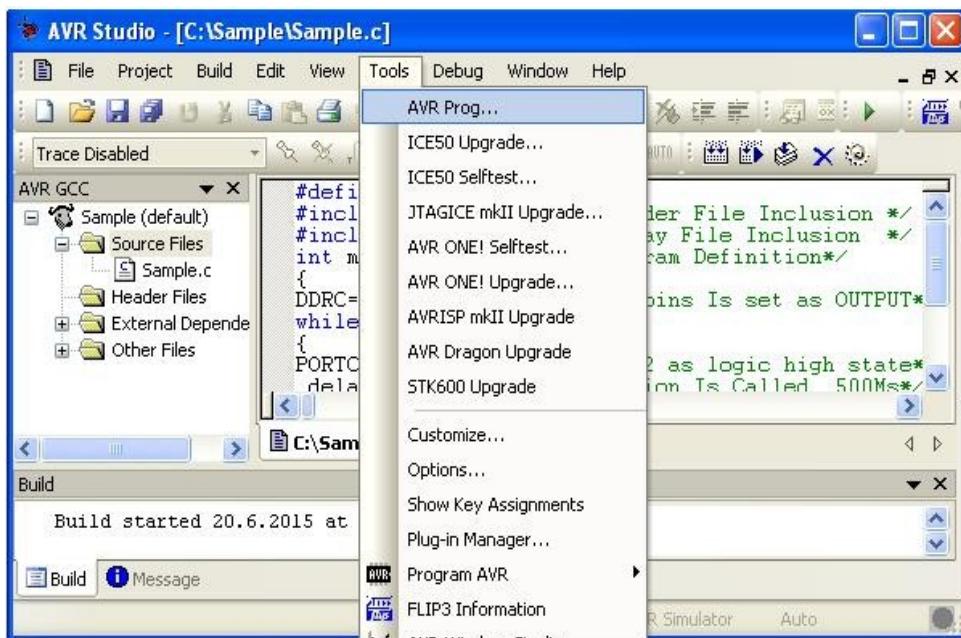
[Build] [Message] [Find in Files] [Breakpoints and Tracepoints]

3.2 : SETTING UP AVR ATmega32 mini

- Power the development board using a USB connector or from screw terminal.
- Make sure that the Power-On LED is ON and Jumper is in proper position.
- Connect the RS232 Cable to the COM port of your computer.
- Connect the other end to the RS232 connector of AVR ATmega32 board.

3.3 : PROGRAMMING STEPS

1. Open AVR Studio
2. Press the reset button on the board and open the 'AVR Prog' from tools of AVR Studio or from \\Atmel\\AVRTools\\AvrProg\\AvrProg.exe within 3 seconds



3. Select the hex file to be downloaded [Eg: C:\Sample\Sample.hex] using the 'Browse' button and select 'Program' option under Flash. The program will be uploaded to the controller.

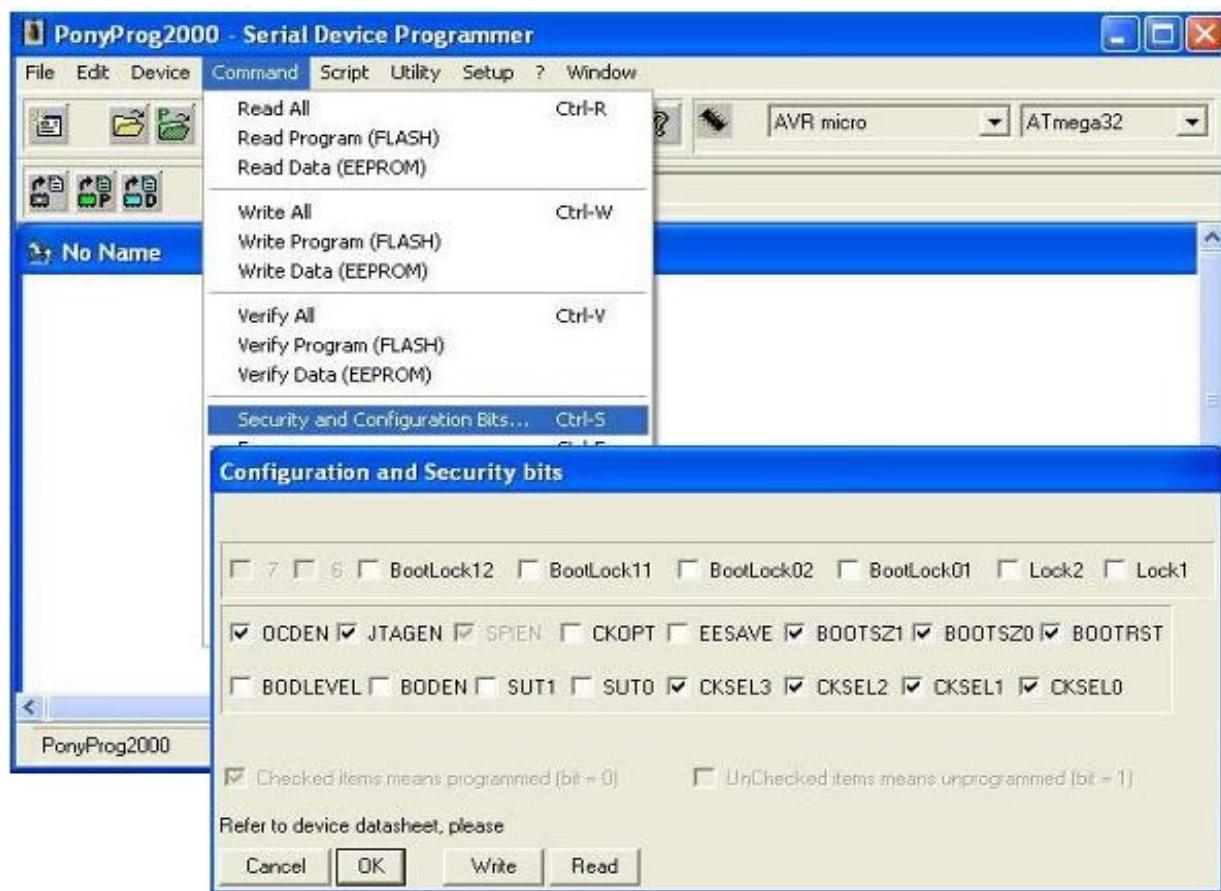
4. Reset the AVR programmer using the RESET switch or Press the 'Exit' option in the window.

The controller will now work according to the program after 3 seconds.

3.4 : JTAG PROGRAMMING

JTAG Fuse has been disabled in the microcontroller included in the development kit. In order to use JTAG you need to enable the fuse bit JTEN in the fuse bits register. The fuse bits can be configured using the AVR ICSP programmer and PonyProg software.

If the JTAG is enabled, the PC2, PC3, PC4 and PC5 port pins cannot be used as IO lines. (For example, if the JTAG is enabled, the LED connected to the port pin PC2 will not work properly).



I/O DISTRIBUTION

4.1 : The Pin Distribution of ATMega32 mini Development Board

PIN No	NAME	TYPE	THE I/O ASSIGN OF ATMega32 Development Board
1	PB.0	I/O	LCD(D0)
2	PB.1	I/O	LCD(D1)
3	PB.2	I/O	LCD(D2)
4	PB.3	I/O	LCD(D3)
5	PB.4	I/O	LCD(D4)
6	PB.5	I/O	LCD(D5) / ISP(MOSI)
7	PB.6	I/O	LCD(D6) / ISP(MISO)
8	PB.7	I/O	LCD(D7) / ISP(SCK)
9	RESET	-	RESET SWITCH
10	VCC	-	+5V
11	GND	-	GROUND
12	XTAL1	-	CRYSTAL
13	XTAL2	-	CRYSTAL
14	PD.0	I/O	MAX232(R20UT) / CP2102(TX)/ 5V,3V3 UART/ ZIGBEE/ RFID
15	PD.1	I/O	MAX232(T2IN)/ CP2102(RX) / 5V,3V3 UART/ ZIGBEE/ RFID
16	PD.2	I/O	PULL-UP KEY1
17	PD.3	I/O	PULL-UP KEY2
18	PD.4	I/O	PULL-UP KEY3
19	PD.5	I/O	LCD(RS)
20	PD.6	I/O	LCD(E)
22	PC.0	I/O	LED1
23	PC.1	I/O	LED2
24	PC.2	I/O	LED3/JTAG(TCK)
25	PC.3	I/O	JTAG (TMS)
26	PC.4	I/O	JTAG(TD0)
27	PC.5	I/O	JTAG(TDI)
28	PC.6	I/O	SERVO MOTOR



29	PC.7	I/O	BUZER
30	AVCC	-	VCC
31	GND	-	GROUND
32	AREF	-	5V/3V3
33	PA.7	I/O	UNUSED
34	PA.6	I/O	UNUSED
35	PA.5	I/O	UNUSED
36	PA.4	I/O	UNUSED
37	PA.3	I/O	UNUSED
38	PA.2	I/O	UNUSED
39	PA.1	I/O	TEMP.SENSOR
40	PA.0	I/O	POTENTIOMETER





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



rhydoLABZ™