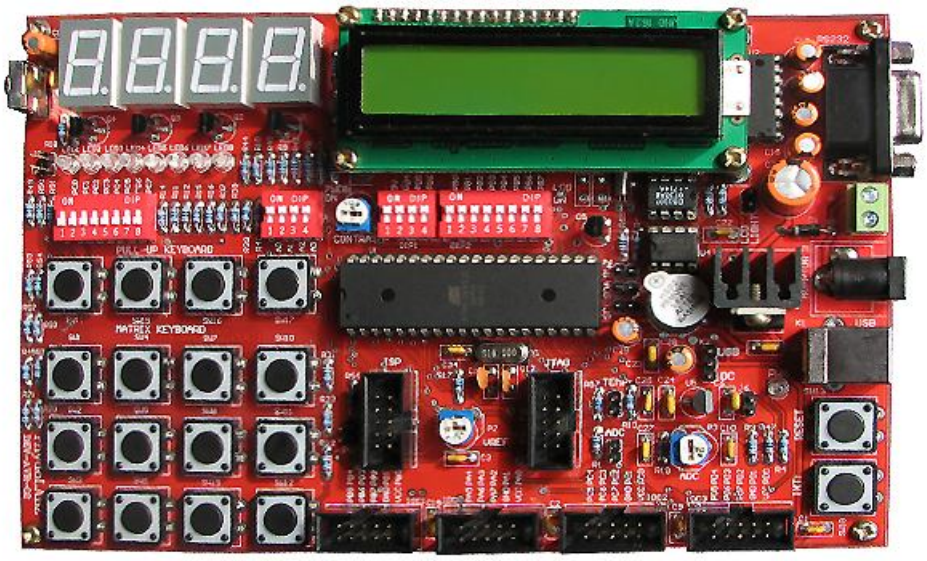
eCee AVR AT Mega32/32L

Development Board

User Manual



Rhydo Technologies (P) Ltd.
(An ISO 9001:2008 Certified R&D Company)
Golden Plaza, Chitoor Road,
Cochin – 682018, Kerala State, India
Phone : 0091- 484-2370444, 2371666
Cell : 0091- 99466 70444
Fax : 0091 - 484-2370579
E-mail : info@rhydolabz.com, sales@rhydolabz.com
WebSite : http://www.rhydolabz.com

rhydolabz.com

CONTENTS

The eCee AVR AT Mega32 Evaluation Board allows you to generate and test application programs for the AVR AT Mega32 microcontroller family. With this hands-on process, you can determine hardware and software components required for product development. The eCee AVR AT Mega32 Board contains all hardware components that are required in a single-chip AT Mega32 controller system plus 1 COM ports for serial RS232 interface.

FEATURES

- Compact and Ready to use design
- Professional EMI/RFI Complaint PCB Layout Design for Noise Reduction
- High Quality Two layer PTH PCB
- Includes AVR AT Mega 32 Microcontroller with 8 MHz
- Board Supports AVRMega32/16L/32/32L/8535 Microcontrollers
- No separate programmer required (Built in Boot loader)
- No Separate power adapter required (USB power source)
- Screw terminal for External power Supply (with Jumper Select Option)
- External Power Supply range of 7V to 20V
- Adaptor (any standard 9-12V power supply) option
- RS-232 Interface (For direct connection to PC's serial port)
- On board Two Line LCD Display (2x16)
- On board TWI EEPROM (4K-AT24C04)
- On board TWI RTC (DS 1307) with Crystal and Battery
- On board 32.768 KHz Crystal for RTC
- Four multiplexed 7-Segment LED Display
- Built in Matrix keyboard (12 keys)
- Built in Pull-Up (4 Keys) Keyboard
- Built in IR Sensor Interface – TSOP 1738
- Built in 8 LED Interface to test I/O
- On Board External Interrupt and Reset buttons
- Built in Potentiometer interface for ADC with variable reference voltage input

- On Board Temperature Sensor Interface
- On Board Buzzer Interface
- On Board PWM Output pin
- On Board JTAG Connector for Debugging/Programming
- On Board ICSP Connector
- On Board 8 MHz Crystal Oscillator
- On Board Power LED Indicator
- On Board DB9 Connector
- On Board USB Connector
- All Port Pins available at IDC (2x5) Connector
- Power Supply Reverse Polarity Protection
- On Board 1 Amp Voltage Regulator
- Can be used as main board for developing applications
- Demo HEX codes included for testing of board features
- Example codes included

eCee AT Mega 32 PACKAGE INCLUDES

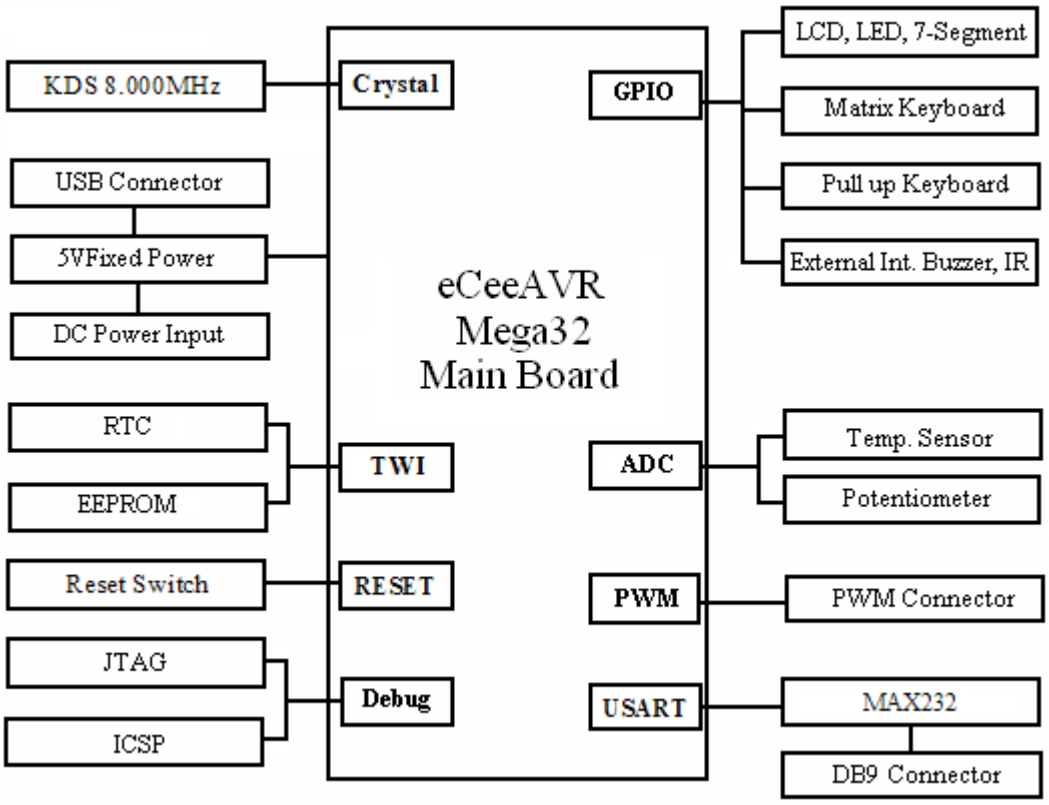
- Fully Assembled and Tested eCee AVR AT Mega 32 Development board
- Software CDROM with
  - Schematic
  - Programming Software
  - Sample Hex Code
  - Example Codes for
    - Led Blinking
    - LCD Display
    - Matrix Keyboard
    - Pull-Up Keyboard
    - 7-Segment Display
    - External Interrupt Handling
    - EEPROM Writing
    - EEPROM Reading
    - PWM Generation
    - RTC Interfacing
    - ADC Interfacing
    - UART Transmission of Temperature

AT Mega32 SPECIFICATION

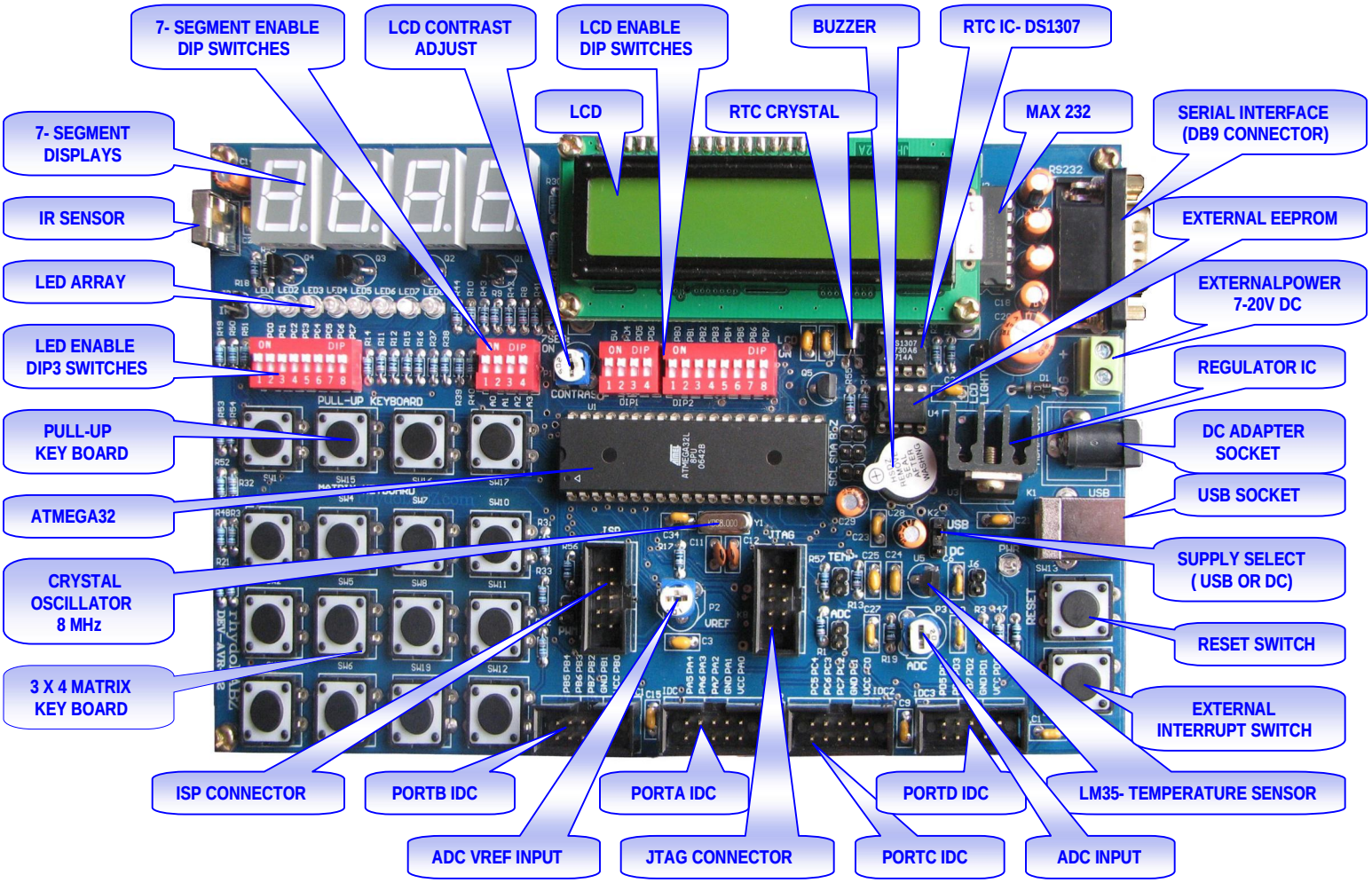
- Atmel AVR AT Mega 32 with 8 MHz Crystal Oscillator (With Boot loader Software)
- High Performance RISC CPU
- 131 Powerful Instructions – Most Single-clock Cycle Execution
- On-chip 2-cycle Multiplier
- 32 K bytes programmable flash
- 2 K bytes SRAM
- 1024 bytes EEPROM
- Upto16 MHz operation (Up to 16MIPS for ATMega32)
- Internal Calibrated RC Oscillator
- 32 I/O pins
- 8 channel 10 bit ADC
- On-chip Analog Comparator
- 4 PWM Channels
- 4 Compare modules associated with each Timer
- Two 8-Bit Timer/Counter
- One 16-Bit Timer/Counter
- Real Time Counter with Separate Oscillator
- One Serial USART
- 1 Two Wire Serial Interface TWI Module
- One Serial Peripheral Interface(SPI - Master/ Slave) Module
- Power Consumption ~ 15mA
- Power-On Reset (POR), Power-Up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Interrupt Capability (up to 20 sources)
- 32 x 8 General Purpose Working Registers
- ICSP Programming
- JTAG (IEEE std. 1149.1 Compliant) Interface
- Brown Out Reset
- Low Voltage Programming

- Power Saving Sleep Mode
- Programmable Code Protection
- Optional Boot Code Section with Independent Lock Bits
- Fully Static Design
- Wide Operating Voltage 2.7V to 5.5V
- Low Power Consumption

FUNCTION MODULE



INTERFACE OVERVIEW



PERIPHERALS	DESCRIPTION
USB(K1)	USB Socket
K2	Power Choose(USB AND DC)
K3	USART Interface via Female DB9 Connector
K4	Screw Terminal(DC Socket)
K5	PWM Terminal
K8	JTAG
U1	AT Mega 32 IC
U2	MAX232(Level converter)
U3	LM7805(Regulator IC(5V))
U4	EEPROM(24C04)
U5	Temperature Sensor (LM35)
U6	BUZZER
U7	RTC(DS1307)
S1	IR Sensor
LCD1	LCD Module, HD44780U
B1	Battery for RTC(3V)
Y1	Crystal (8MHz)

Y2	Crystal for RTC (32.768KHz)
SW1-SW12	Matrix Keypad switches
RESET(SW13)	Reset Key
SW14-SW17	Pull-Up Switches
LED1-LED8	Light Emitting Diodes
INT	External Interrupt key
DISP1-DISP4	7 Segment Display modules
CONTRAST(P1)	LCD Contrast Adjust through Pot
IDC1	PORTB pins are available on 5x2 IDC
IDC2	PORTC pins are available on 5x2 IDC
IDC3	PORTD pins are available on 5x2 IDC
IDC4	PORTA pins are available on 5x2 IDC
ISP	To program the IC
ADC(P3)	Potentiometer As ADC input
Vref(P2)	Potentiometer to provide variable reference voltage to ADC
J1	Adaptor socket

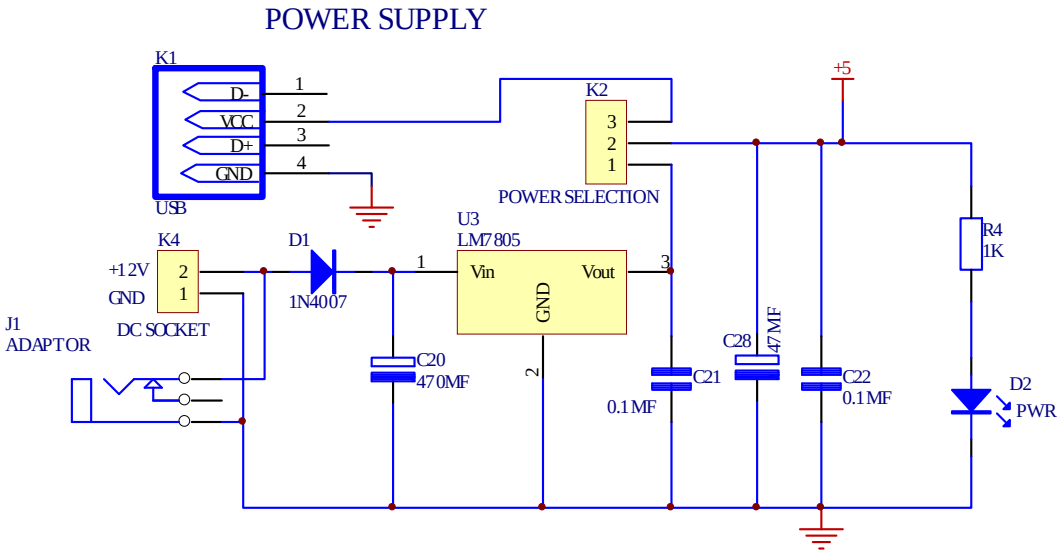
Jumper no	Description	Set option	Set description
k2	Supply Select	1-2 2-3	DC Power is selected, USB Power is selected
LCD Light	LCD Light	Short to access	LCD backlight ON
TEMP	TEMP	Short to access	LM35 Access via PA5(channel 5)
ADC	ADC	Short to access	Potentiometer Output connected to PA6 input(channel 6)
SCL	Clock pin of EEPROM and RTC	Short to access	EEPROM,RTC could be accessed
SDA	Data pin of EEPROM and RTC	Short to access	EEPROM,RTC could be accessed
BZ	Buzzer	Short to access	Buzzer could be accessed
J6	External Interrupt	Short to access	External interrupt 0 could be accessed
J3	IR	Short to access	IR sensor could be accessed

POWER SUPPLY

Atmel eCee AVR AT Mega 32 Board has three power supplies; you can choose one of the following ways to supply power

(1) Through an Adaptor (any standard 9-12V power supply)
(2) Through the USB port
(3) Through the DC socket (7-20V power supply)

The external Power Supply circuit is given below:

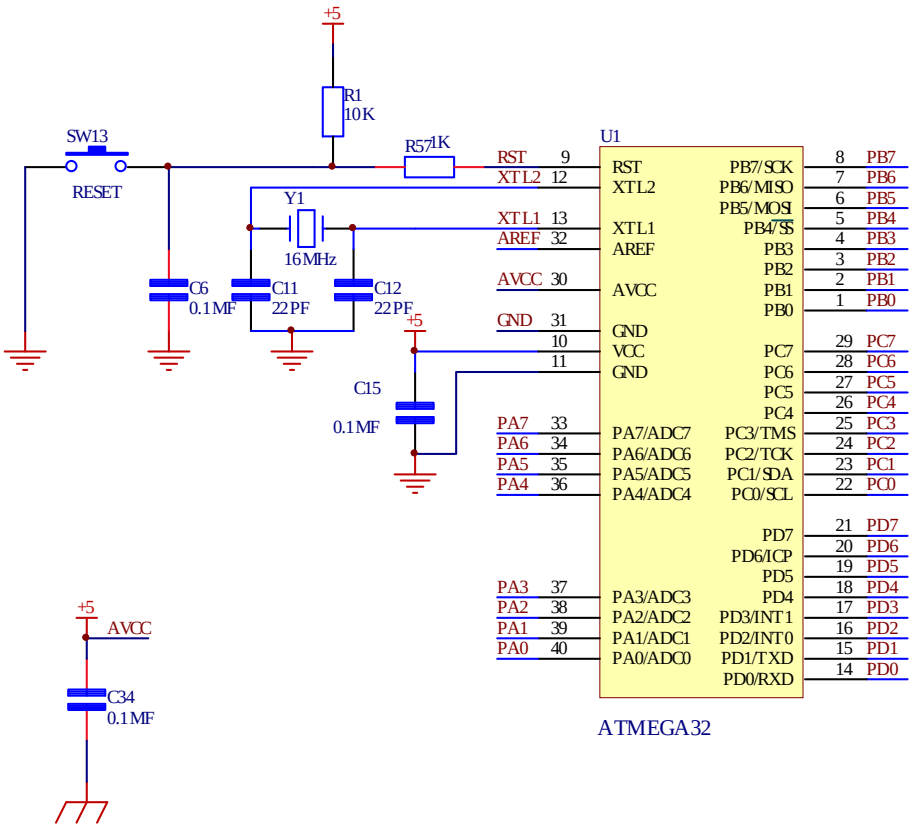


CLOCK SOURCE

eCee AVR AT Mega 32 board uses:

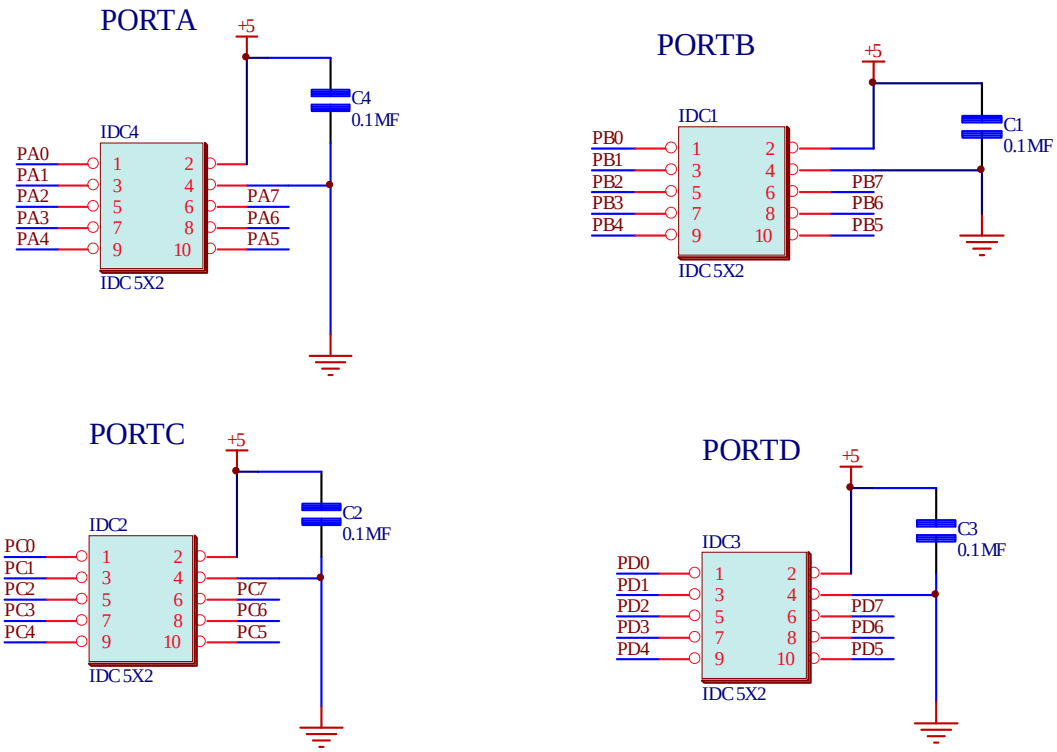
- 8 MHz Crystal as the MCU clock source
- 32.768 KHz Crystal as the RTC clock source

MICROCONTROLLER - PIN OUT

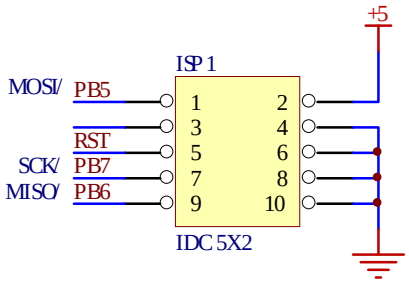


PORT PINS - 5x2 IDC CONNECTOR

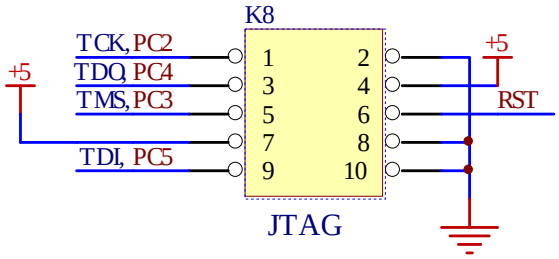
The eCee AVR AT Mega 32 board has all port pins available at 5X2 IDC. The connection is as given below.



ISP CONNECTION

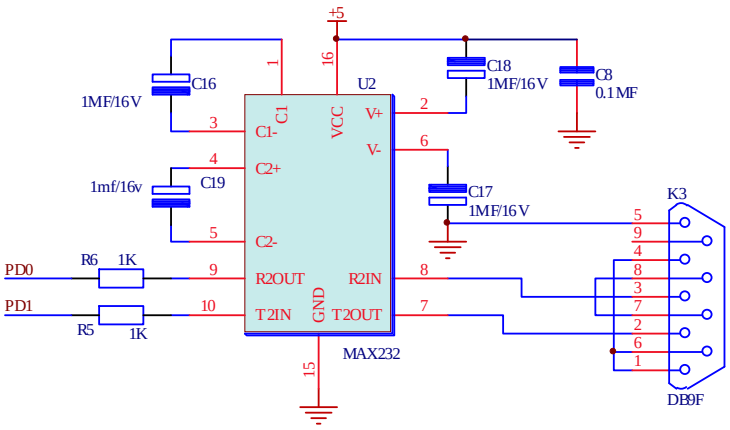


JTAG CONNECTION



UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

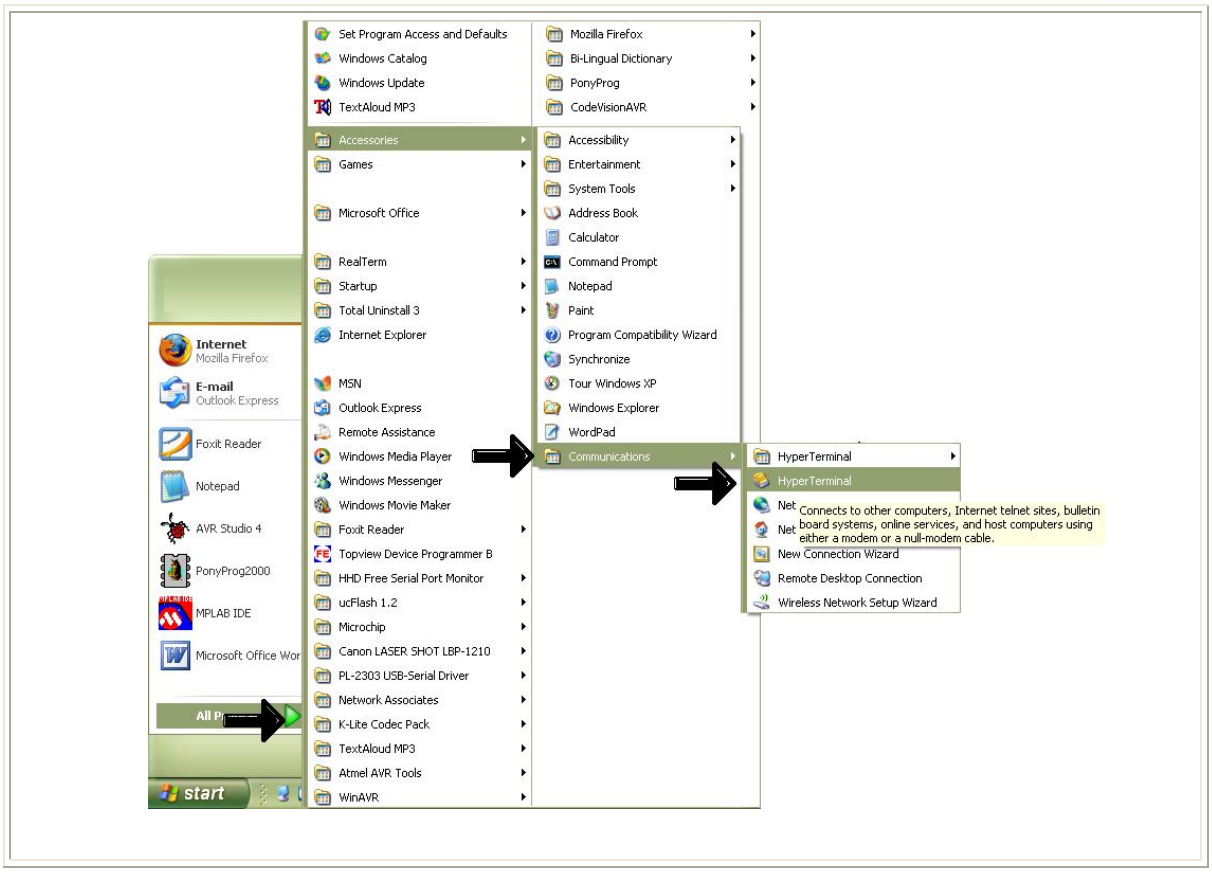
The AVR AT Mega32 USART module operates through **PD1 (TXD)** and **PD0 (RXD)** pins.

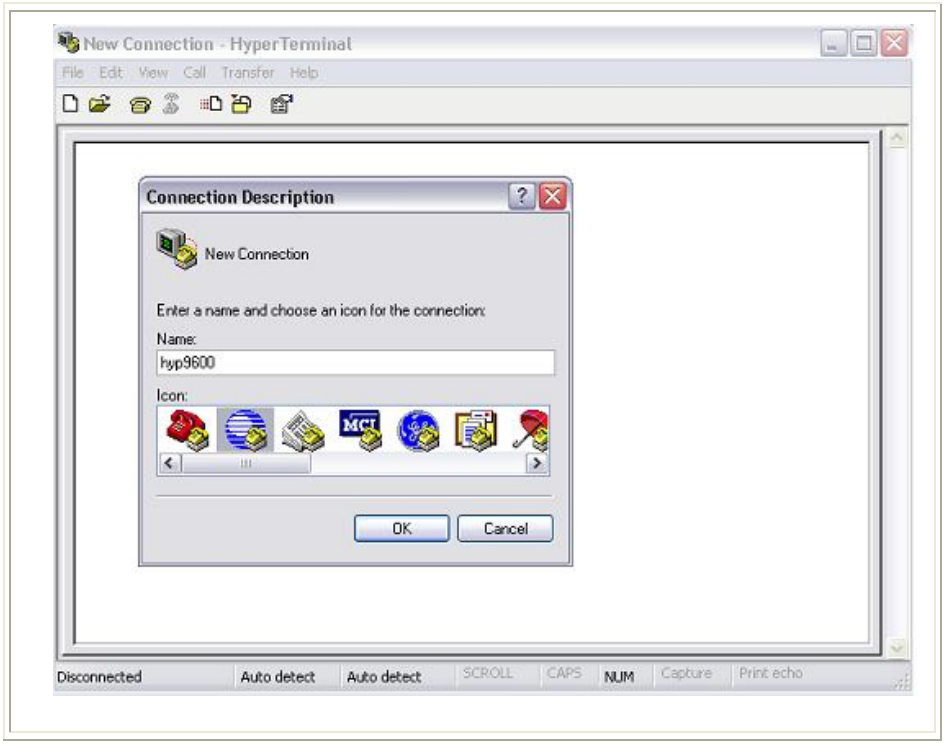


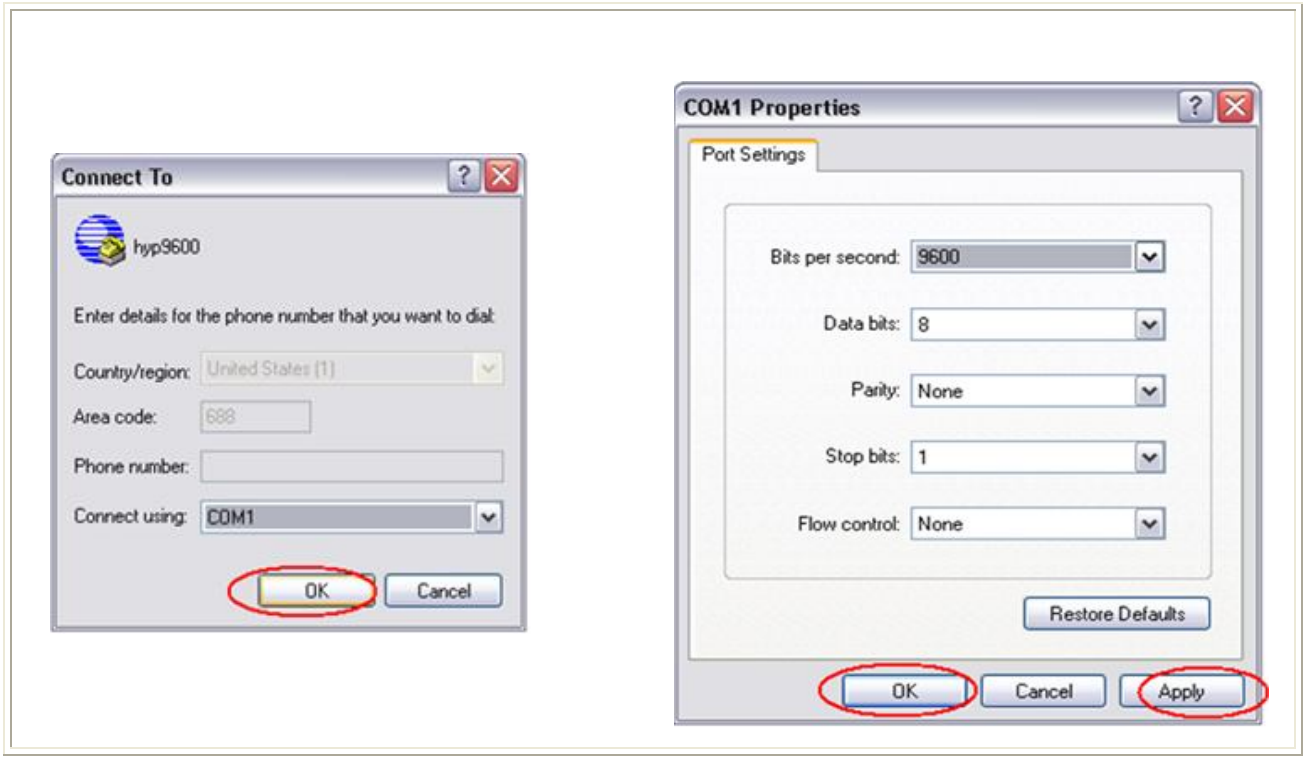
The AVR AT MEGA 32 USART output itself operates at CMOS voltages, and needs an external serial line driver to convert its output into a higher symmetrical line voltage. The MAX 232 serial driver serves this purpose.

CREATING HYPERTERMINAL IN PC

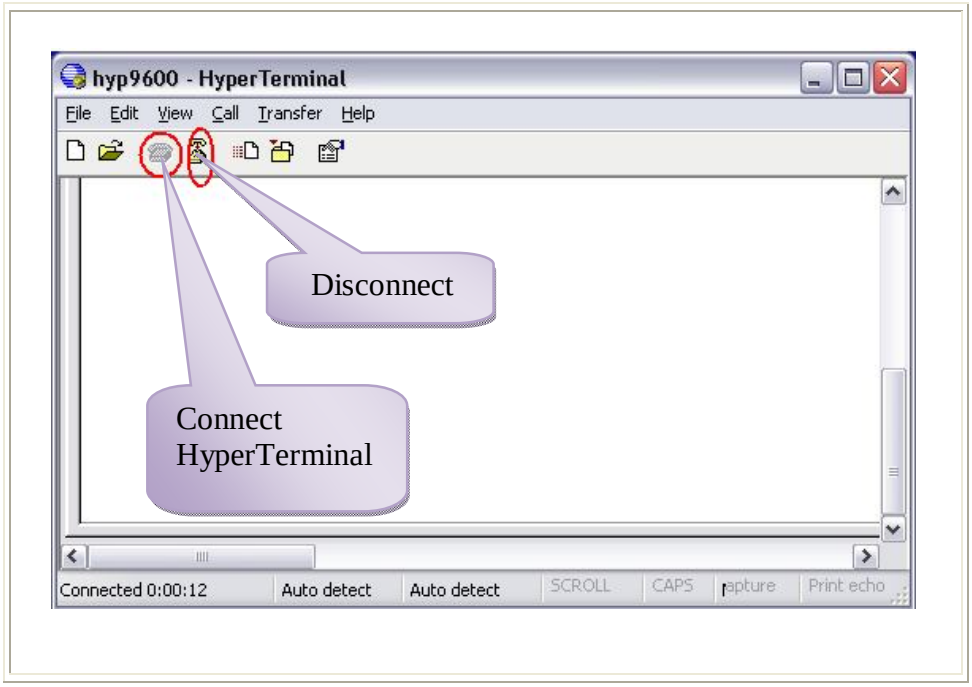
The serial data transmitted through USART can be viewed on a PC using a Windows tool for Serial Port Communication called HyperTerminal.

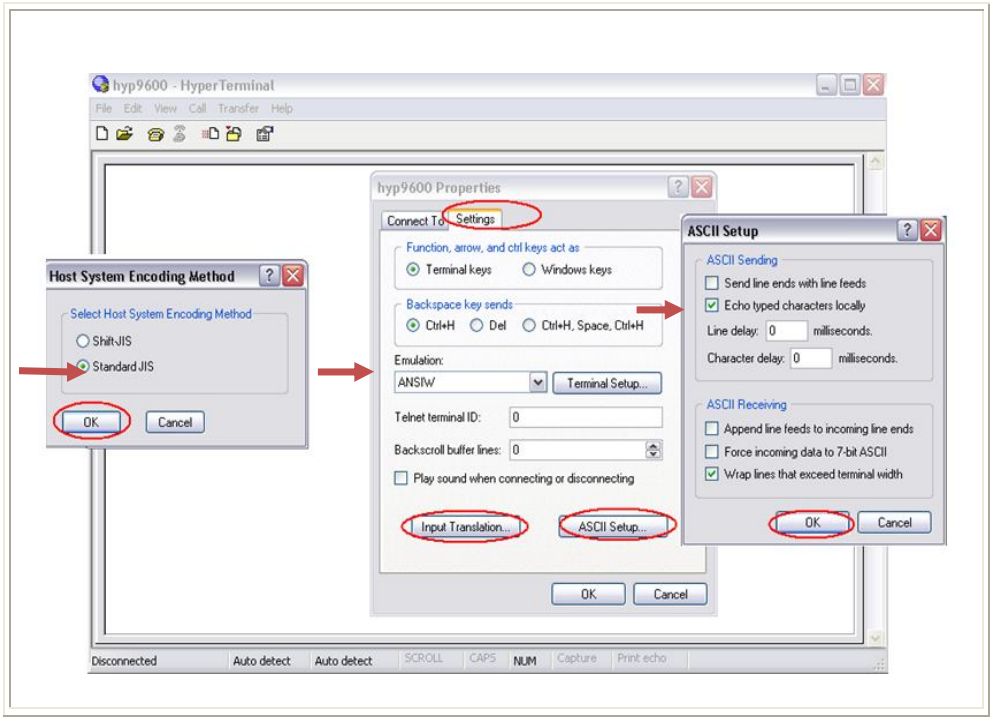



New Connection - HyperTerminal
File Edit View Call Transfer Help
Connection Description
New Connection
Enter a name and choose an icon for the connection:
Name:
hyp9600
Icon:
OK
Cancel
Disconnected
Auto detect
Auto detect
SCROLL
CAPS
NUM
Capture
Print echo

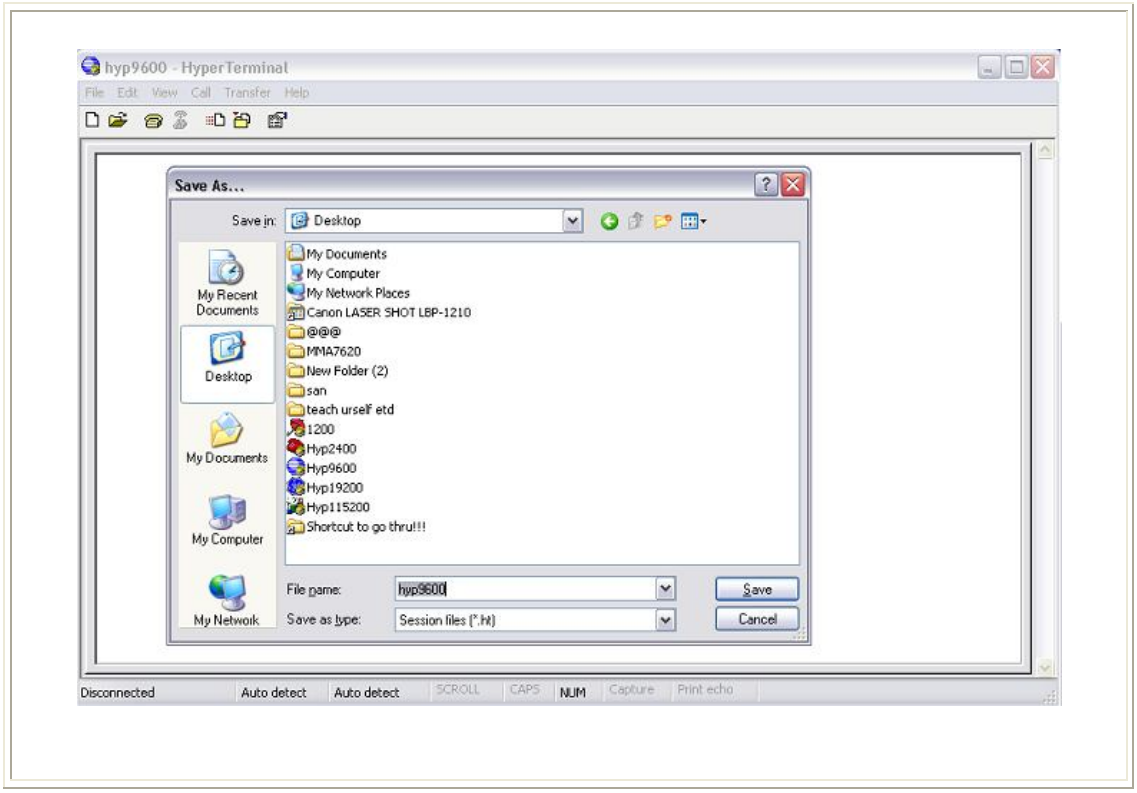

Connect To
hyp9600
Enter details for the phone number that you want to dial:
Country/region:
United States (1)
Area code:
688
Phone number:
Connect using:
COM1
OK
Cancel
COM1 Properties
Port Settings
Bits per second:
9600
Data bits:
8
Parity:
None
Stop bits:
1
Flow control:
None
Restore Defaults
OK
Cancel
Apply

Now Disconnect the Hyper Terminal. Then open the Property from 'File'.



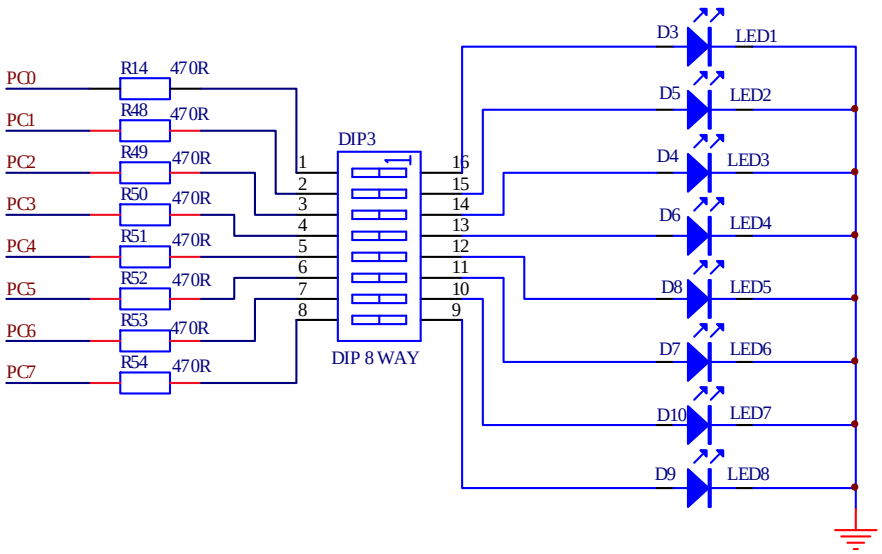


Select File → Save As → Desktop



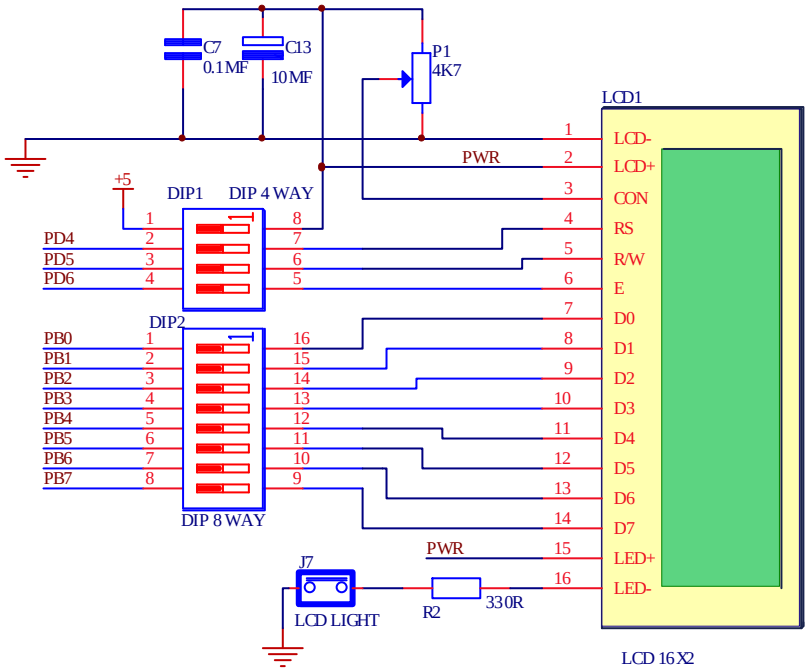
LED INTERFACING

LED's are the simplest devices to test port functioning. The board contains 8 LED's connected to PORTC of the AVR Mega32.



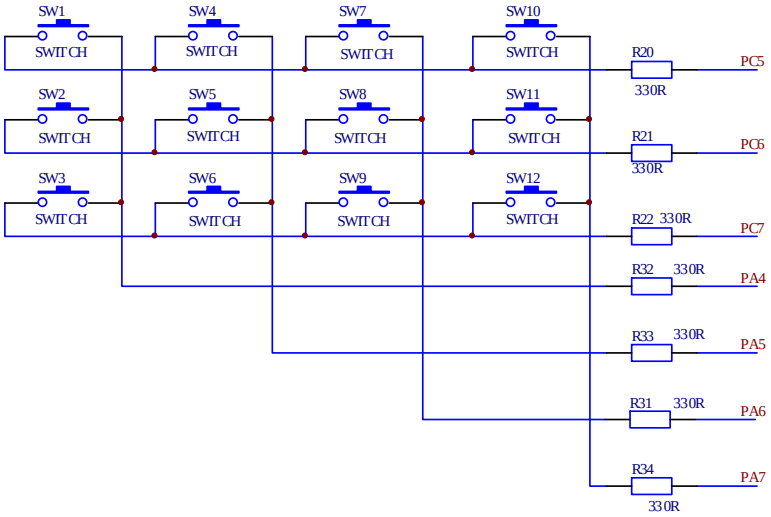
LCD - LIQUID CRYSTAL DISPLAY

The display is a standard 16x2 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). 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 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, RW, E pins.



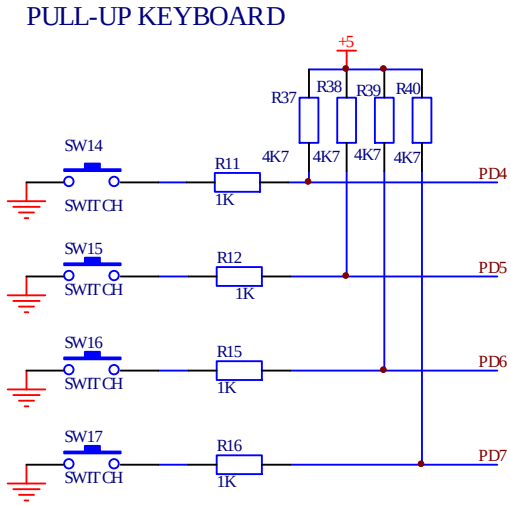
MATRIX KEYBOARD

A keypad is simply an array of push buttons connected in rows and columns, so that each can be tested for closure with the minimum number of connections. There are 12 keys on a phone type pad (0–9, A-B), arranged in a 4X3 matrix. Assume the columns are labeled 1, 2, 3, 4 and the rows A, B, C. If we assume that all the rows and columns are initially high, a keystroke can be detected by setting each row to low in turn and checking each column for a zero. In the KEYPAD circuit the 3 keypad row selection pins are connected to PORTC and Column lines are connected to PORTA. Pins PC5-PC7 are initialized as outputs and pins PA4-PA7 are used as inputs. These input pins are pulled high to logic 1. The output rows are also initially set to 1.



PULL UP KEYBOARD

The simplest input is a switch or push button. This can operate with just one additional support component, a pull-up resistor, but there are still some significant issues to consider, such as input loading and de-bouncing.

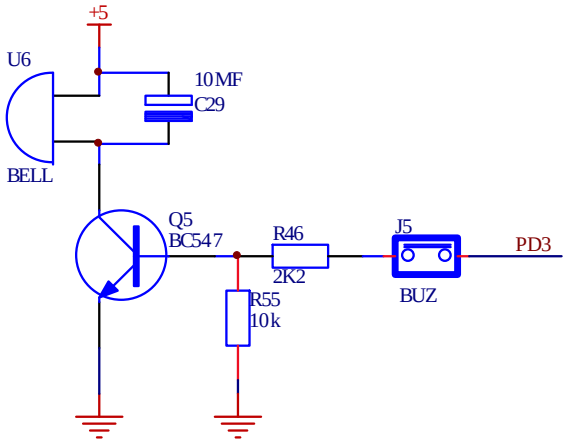


When the switch is open, the output voltage of the circuit is pulled up to 5V via the resistor. There are 4 pull up switches in the board connected to four PORTD bits; 4, 5, 6 and 7. User can access the switch status, by simply reading the PIND register.

BUZZER INTERFACING

Interfacing a buzzer is a simple I/O operation. Normally we use piezo electric devices as buzzer. Buzzer is driven using a simple NPN transistor with a biasing. The transistor's base is connected to the I/O port 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 triggered on and which in turn switch on the Buzzer.

BUZZER



In our development board Buzzer driving is made possible through the I/O pin PD3 (PORTD's third pin). In order to sound the buzzer the jumper, J5 should be kept in position. When a logical 1 is out through the PD3 pin, the transistor will turn on and which in turn sound the buzzer. If logical 0 is provided, the buzzer will be turned off.

ADC INTERFACING

eCee AVR AT Mega32 has a 10-bit resolution ADC module.

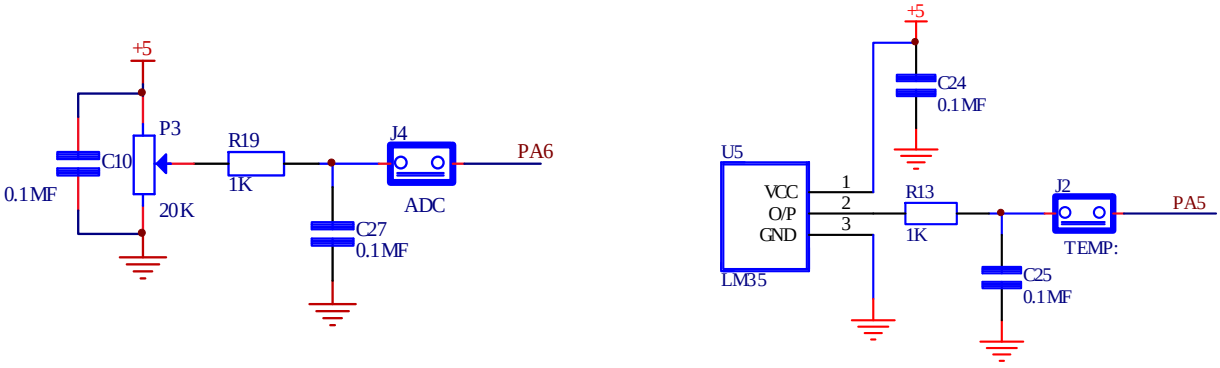
Steps to Configure ADC without interrupt and transmit the result through USART Software

1) Configure the SFIOR register, so that the free running Trigger source is selected.
2) Select the reference and the **Channel 6** using the register ADMUX to select Potentiometer as ADC input or Select **Channel 5** to interface with the Temperature Sensor.
3) Configure ADCSRA, so that ADC is enabled and no start condition is selected.
4) Initialize USART transmission at **9600** Baud Rate.
5) Enable Start Condition.
6) Wait until conversion completes
7) Read the LSB Result from ADCL first
8) Then Read the MSB Result from ADCH
9) Split the result into digits and transmit through USART

Hardware

1) Place the Jumper **J4 (ADC)** in position for Potentiometer input or Connect Jumper **(Temp)** if you are using Temperature sensor as input. And power the board, eCeeAVRMega32.
2) Download the program
3) Connect the Serial Port to the computer using serial cable
4) Open Hyper Terminal with baud rate **9600**
5) Digital data will be displayed in the hyper terminal

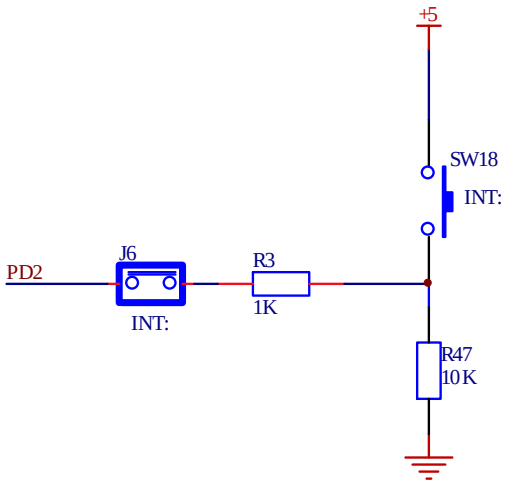
TEMPERATURE SENSOR



EXTERNAL INTERRUPT

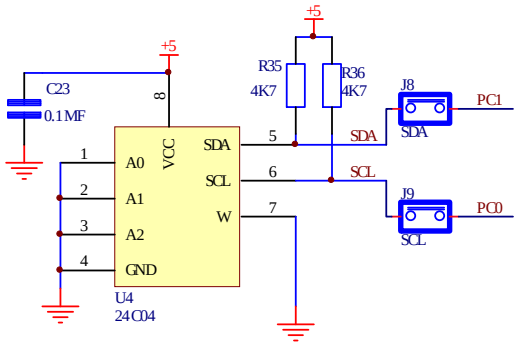
In eCee AVR AT Mega 32 development board, there is a provision for giving external trigger to the microcontroller through hardware. This key is connected to the 2nd Pin of the PORTD register (INT0).

INTERRUPT



EXTERNAL EEPROM INTERFACING

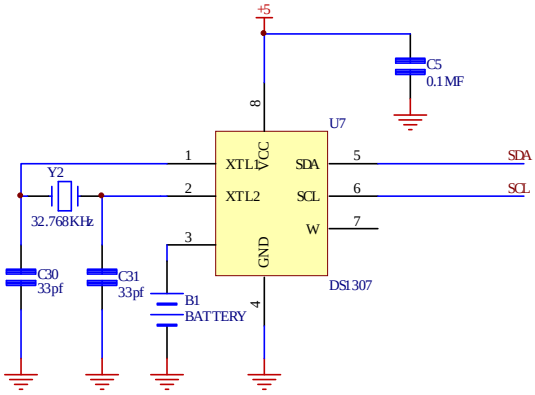
The Two-wire Serial Interface (TWI) is ideally suited for interfacing with External EEPROM. The memory used here is an Atmel 24C04, which stores 4 kb of data and uses a standard TWI interface. The 4K is internally organized with 256 pages of 2 bytes each. Random word addressing Chip Number requires a 9-bit data word address. TWI is a more versatile system level serial data transfer method. It only needs two bus connected signals; clock (SCL) and data (SDA) lines. The signal lines are pulled up to 5V so that the device connected to it can control the line by pulling it down; this allows slaves to acknowledge operations initiated by the master.



RTC INTERFACING

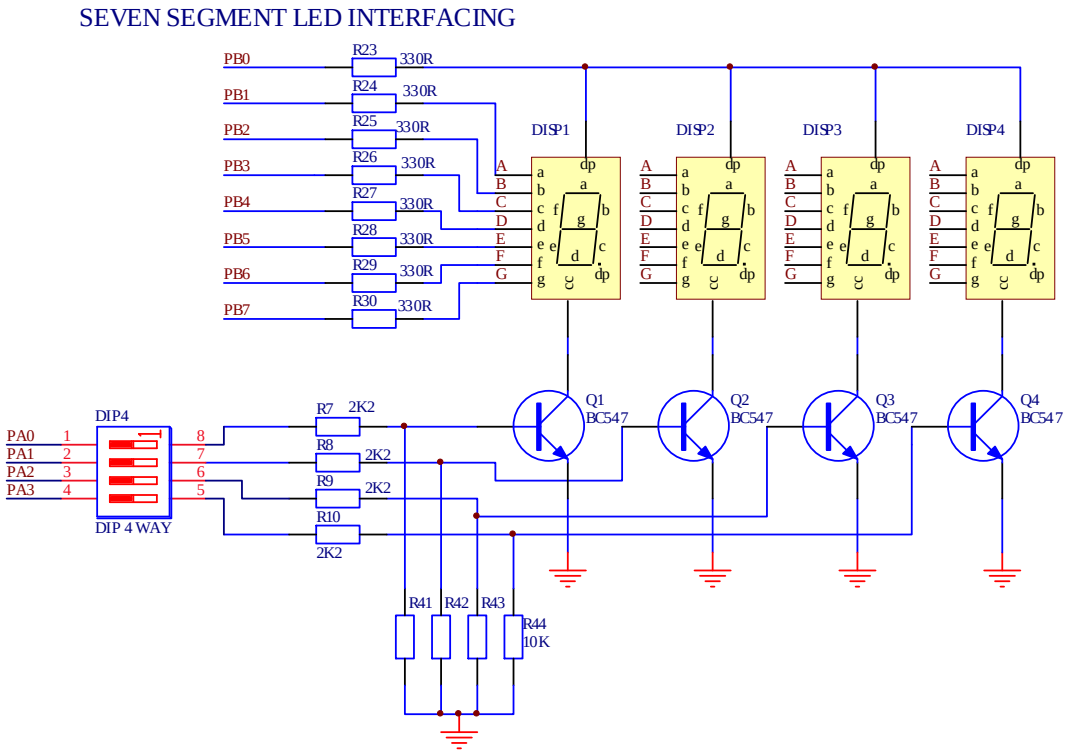
The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. The Two-wire Serial Interface (TWI) is ideally suited for interfacing with RTC.

RTC



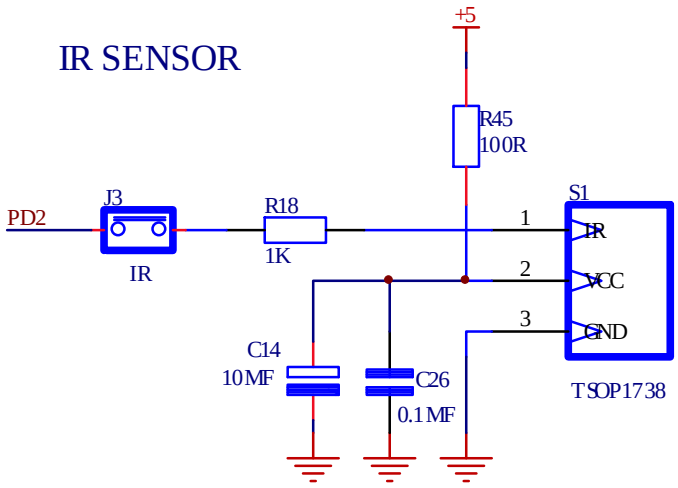
7 SEGMENT MODULE INTERFACING

Each segment is driven separately from PORTB via a current-limiting resistor. There are 4 seven segment modules in the board. PORTA pins 0,1,2,3 are used to select among the modules.



IR SENSOR INTERFACING

IR (TSOP) out is connected to the PORTD pin, PD2 through the jumper J3 (IR). In order to test the IR module, place the Jumper in position. And configure the PD2 as input port.



No pull-up is to be enabled to test the IR module. Normally the terminal value of the pin will be active high. If an IR signal with carrier 38 KHz is received the port terminal value will be pulled low. If you want to check an RC5 protocol, scan the PIN status in the prescribed interval.

USING AVR STUDIO IDE

CREATING A PROJECT

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

- Create Project File
- Select the Project type (Whether Assembly or using winAVR)
- Create New Source Files.
- Add Source Files to the Project.
- Set Configuration Options for Target Hardware.
- Create a HEX File.
- Build Project and Generate Application Program Code.

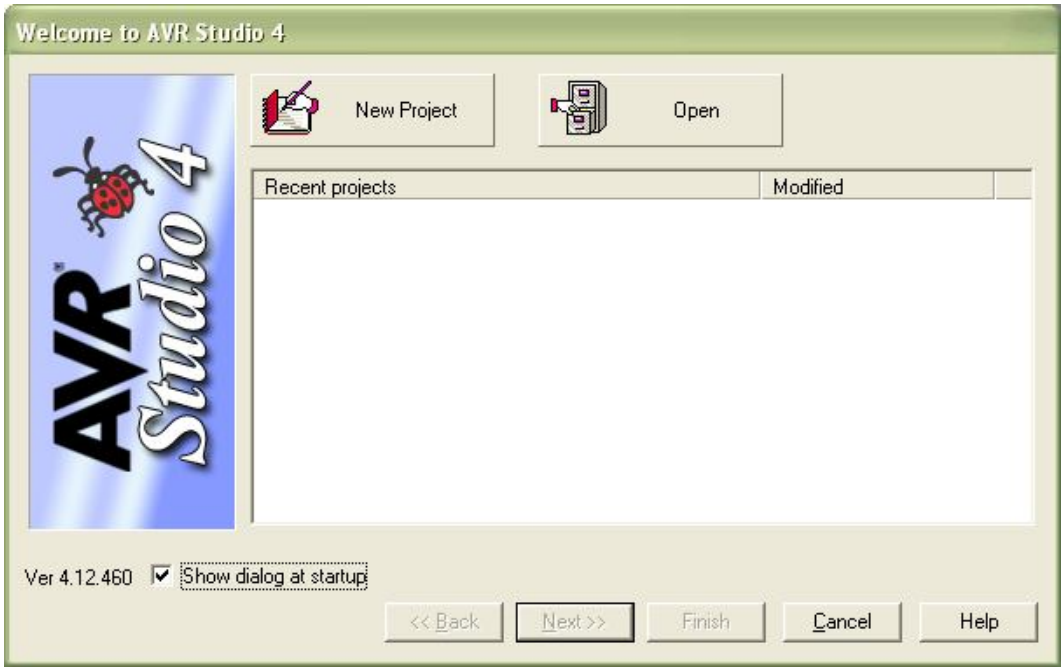
This section provides a step-by-step tutorial that shows you how to create a simple µVision3 project.

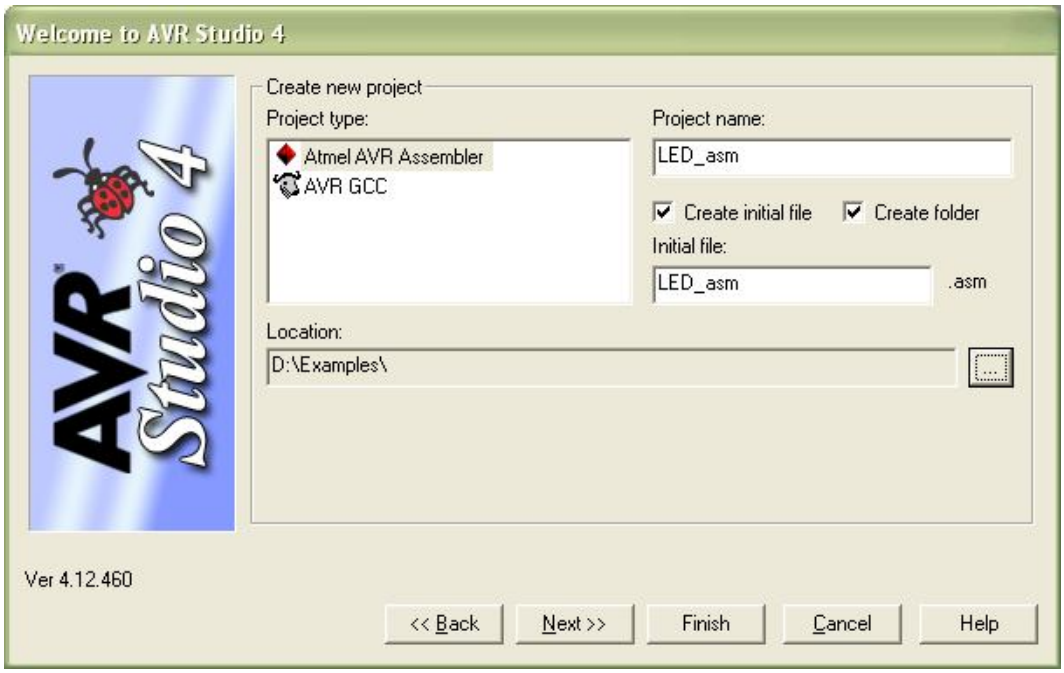
Create Project File and Select CPU

Create Project File Folder and Specify Project Name

To create a new project file select from the AVR Studio menu **Project – Project wizard** for the new AVR project. You should you use a separate folder for each project.

Now select the **New Project** option. If you want to re-open a pre done project select the **Open** option.





Select the Project type (Whether Assembly or using winAVR)

Select the **AVR Assembler** Tool. Now Provide a Project name for our proposed project. You can provide a specified path to save the project. You can simply use the icon **Create New Folder** in this dialog to get a new empty folder.

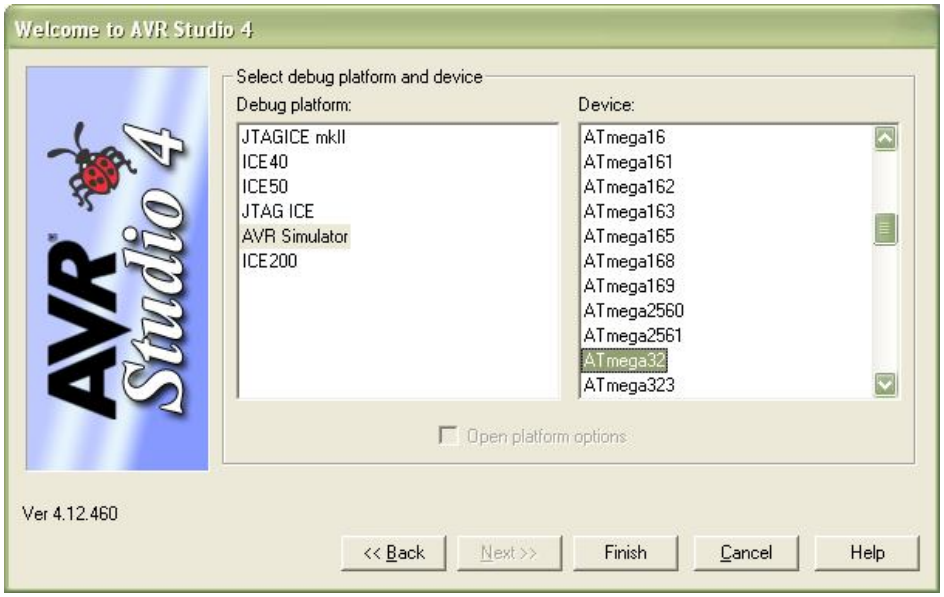
If the **Create Initial file** is selected, an **.asm** file will be created in the same working folder in the name of the project. And will be added to the project workspace. In that **.asm** file you can enter your source code (Recommended). And if the **Create Initial file** is not selected, simply select the **File – New File** option to write the program.

If the **Create Folder** option is selected, a new folder will be created inside the selected folder in the name of the project that we provided.

Select the project location, Name the project and Click **Next**

Select Microcontroller from Device Database

Now Select the Debug tool and the micro controller (From the Device list). Select the debugger platform as **AVR Simulator**.



While you creating a new project AVR Studio asks you to select a CPU for your project. The **Device** section shows the AVR Studio device database. Just select the microcontroller you use. For example; in this chapter we use the AVR AT Mega 32 controller. This selection sets necessary tool options for the AVR AT Mega 32 device and simplifies in this way the tool configuration.

Now Select the **Finish** to complete the project creation.

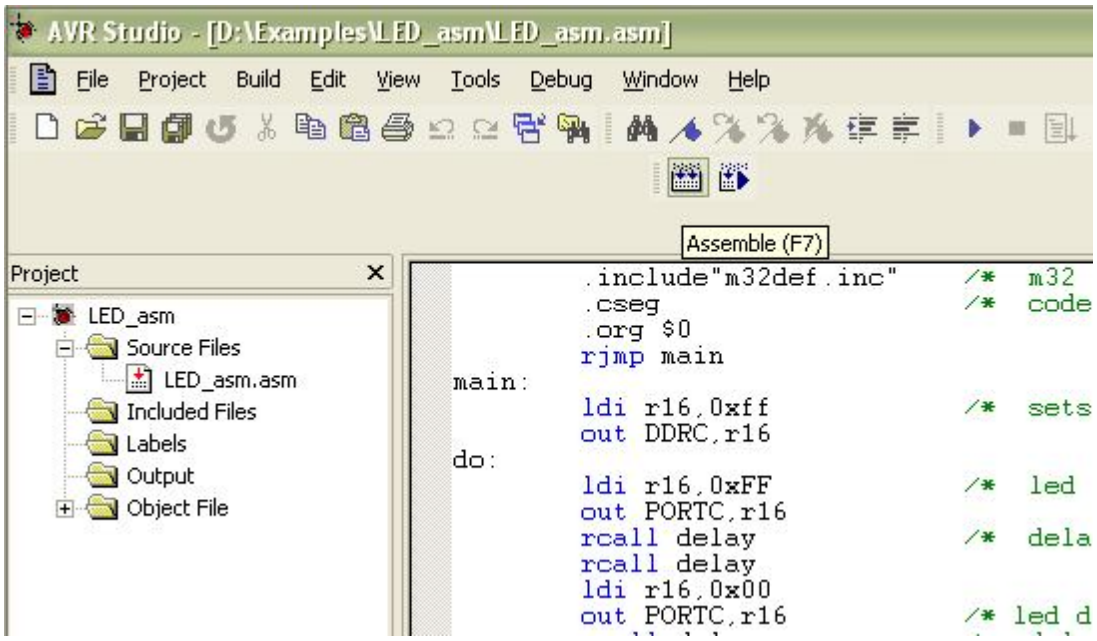
Create New Source Files

A new source file will be opened after completing the project creation since we select 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** unless 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 Working folder, where project file is being saved. AVR Studio enables the color syntax highlighting when you save your file with the dialog **File – Save As…** under a filename with the extension *.asm. We are saving our example file under the name **LED_asm.asm**.

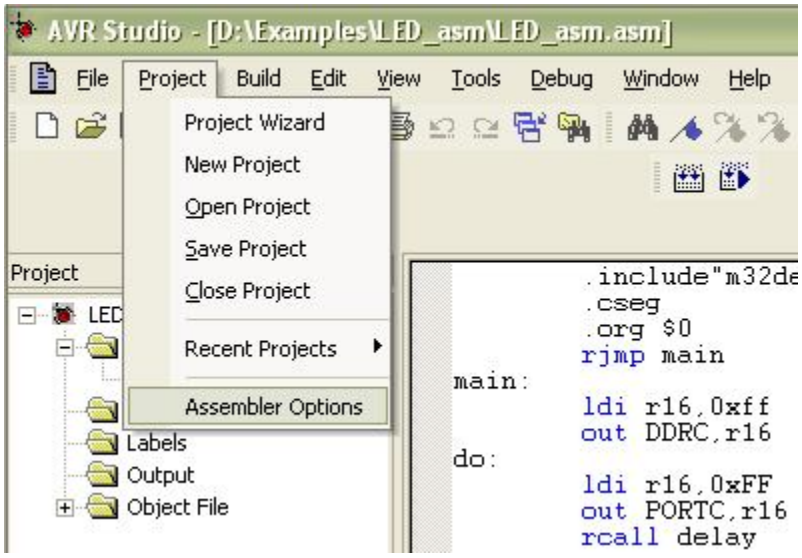
Add Source Files to Project

Once you have created your source file you can add this file to your project. AVR Studio 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 **LED_asm.asm** you have just created.

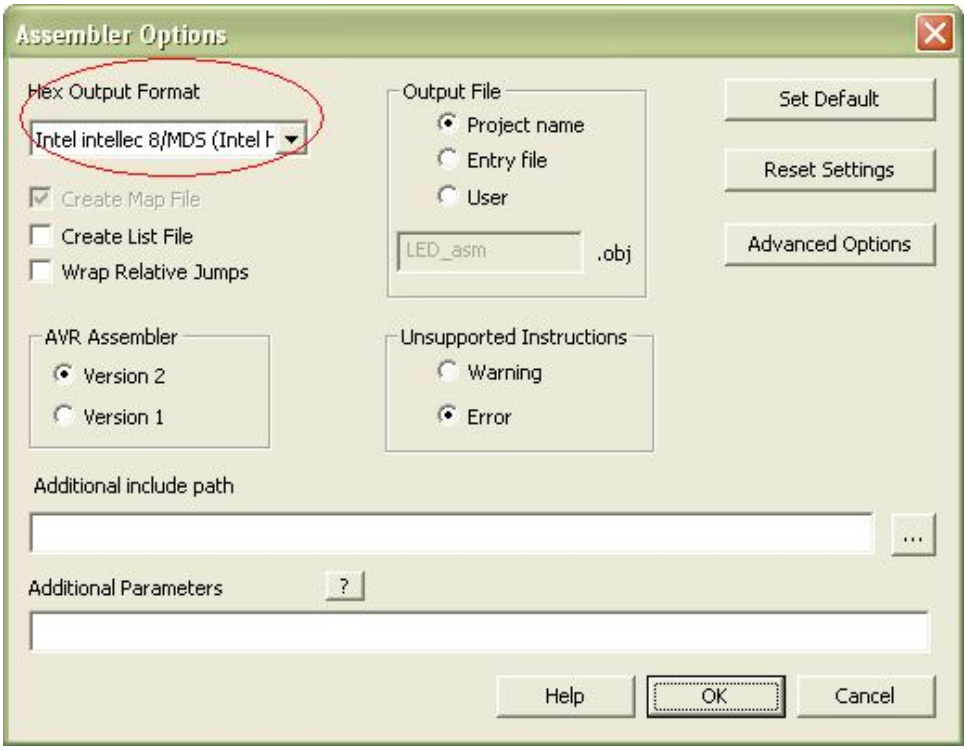
```
        .include"m32def.inc"        /*  m32 header file included        */
        .cseg                       /*  code segment selected           */
        .org $0
        rjmp main
main:
        ldi r16,0xff                /*  sets portc as o/p port          */
        out DDRC,r16
do:
        ldi r16,0xFF                /*  led display  on                 */
        out PORTC,r16
        rcall delay                 /*  delay for 50 ms                 */
        rcall delay
        ldi r16,0x00
        out PORTC,r16               /* led display  off                 */
        rcall delay                 /*  delay for 50 ms                 */
        rjmp do
```



Set Configuration Options for the Project



AVR studio lets you to set options for your target hardware. The dialog **Assembler Options** opens via the **Project** menu icon. In this option you can specify all relevant and parameters of your target hardware the on-chip components of the device you have selected. The following dialog shows the settings for our example.

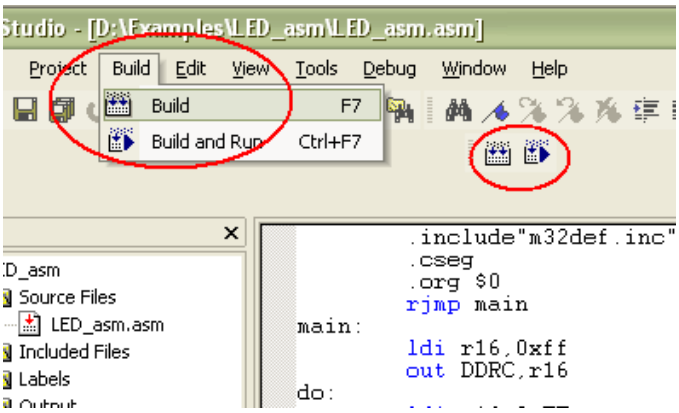


Create HEX File

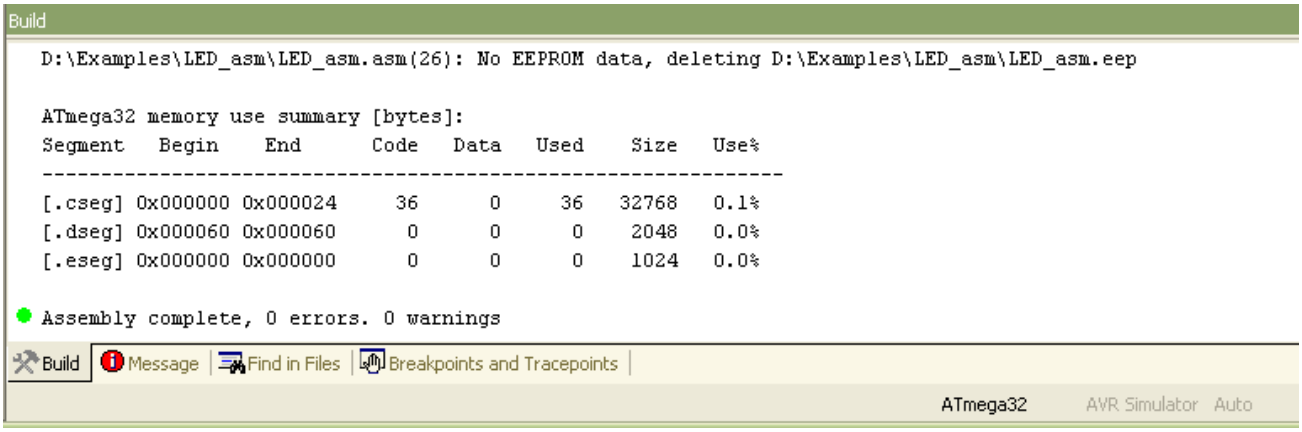
Once you have successfully generated your application you can start debugging. After you have tested your application, it is required to create an Intel HEX file to download the software into an EPROM programmer or simulator. AVR Studio creates HEX files with each build process when **Output HEX File** in the **Assembler Option** is selected as **Intel Hex**.

Build Project

After Configuring the Device, Build the program that attached to the project as shown in the Fig above (Press F7). Build option can be taken 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. In this example the **Hex** file is **LED_asm.HEX**. This file will be available in the folder **LED_asm.**

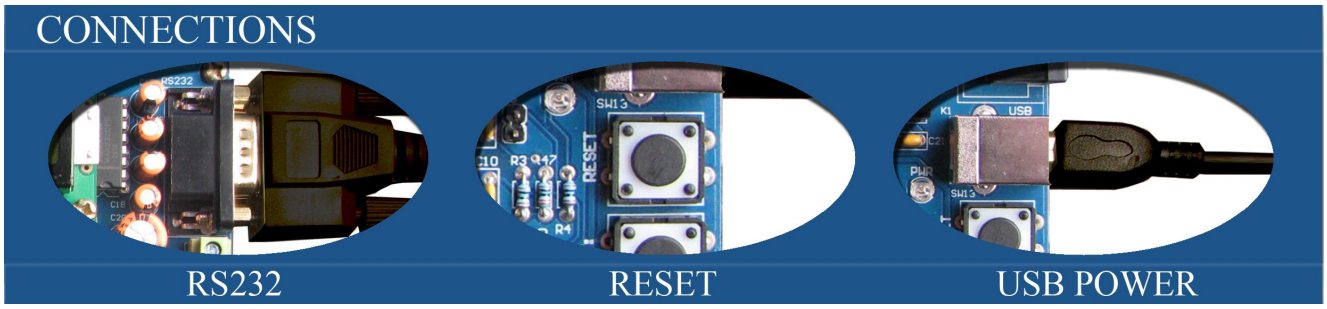


When you build an application with syntax errors, AVR Studio will display errors and warning messages in the **Output Window – Build** page. A double click on a message line opens the source file on the correct location in the AVR Studio editor window.



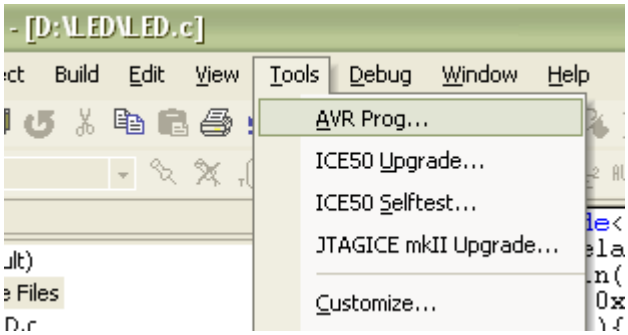
SETTING UP eCee AVR Mega 32

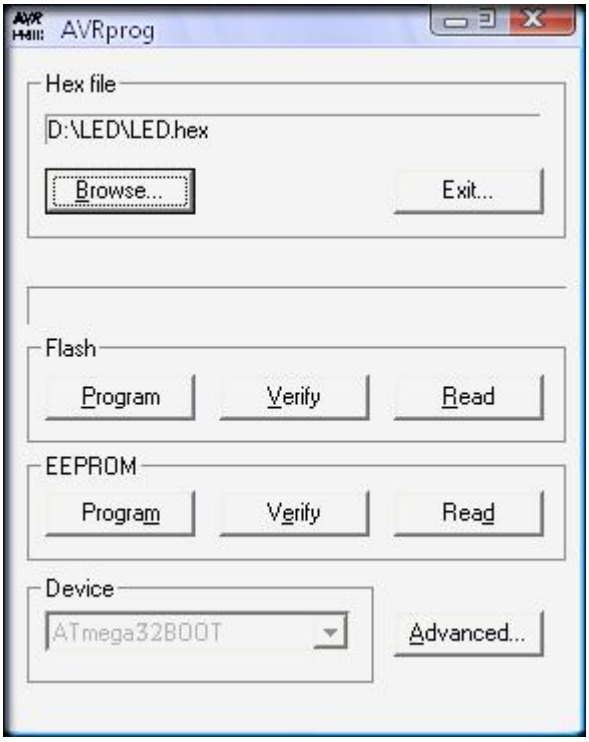
- Power the demo board by means of a USB connector or from screw terminal.
- Make sure that the Power-On LED is ON and Jumper in proper position.
- Connect the Rs-232 Cable to the COM port of your computer.
- Connect the other end to the RS232 connector of AVR AT Mega 32 board.



PROGRAMMING STEPS

1. Open the AVR Studio (No need to open any project)
2. Press the reset button in eCeeAVRMega32and within 3 Seconds open the ‘AVR Prog’ from Tools of AVR Studio or Open from (C:\ProgramFiles\Atmel\AVRTools\AvrProg\AvrProg.exe).



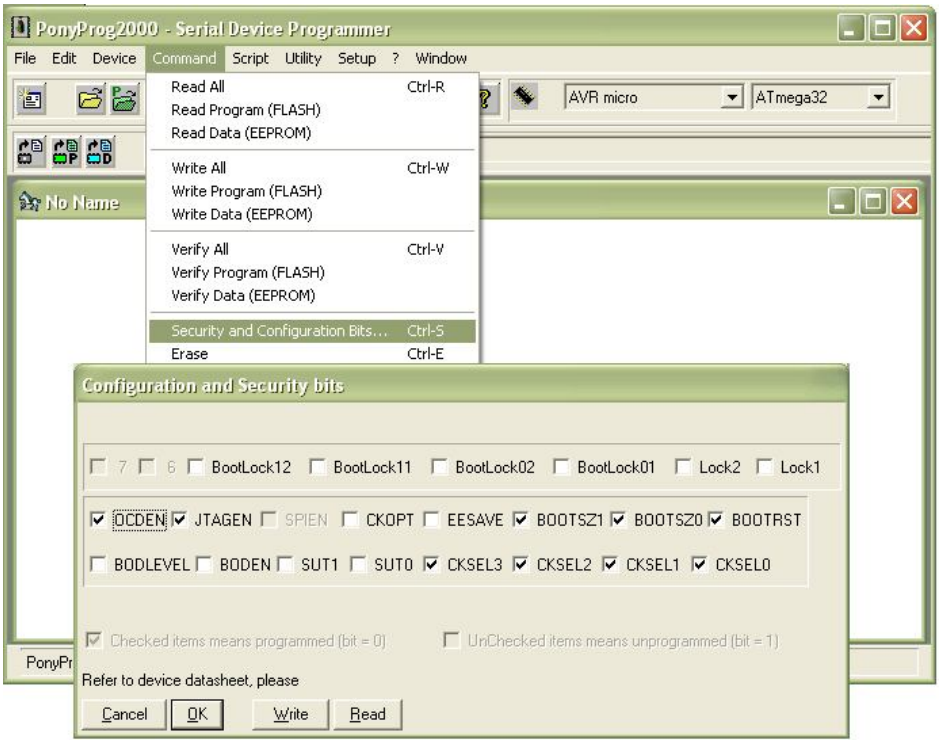


3. Select the hex file to be downloaded [Eg: D:\LED\LED.hex] using the 'Browse' Button and select the 'Program' option to the flash. The Program will be get downloaded to the controller.
4. Reset the AVR-programmer using the RESET switch or Press the 'Exit' option in the window. Now the controller will work according to the program after 3 seconds.

JTAG

JTAG Fuse has been disabled in the Microcontroller AVR AT Mega 32 included in the eCee AVR AT Mega 32 Development kit. In order to use JTAG you need to enable the fuse bit JTEN in the Fuse bits Register. By means of AVR ICSP programmer and PonyProg software, you can simply configure the Fuse bits.

If the JTAG is enabled, you cannot use the I/O pins PC2, PC3, PC4 and PC5 port pins as IO lines. (That is, if the JTAG is enabled the LEDs interfaced to the port pins PC2, PC3, PC4 and PC5 will not work properly).


PonyProg2000 - Serial Device Programmer
File Edit Device Command Script Utility Setup ? Window
Read All Ctrl-R
Read Program (FLASH)
Read Data (EEPROM)
Write All Ctrl-W
Write Program (FLASH)
Write Data (EEPROM)
Verify All Ctrl-V
Verify Program (FLASH)
Verify Data (EEPROM)
Security and Configuration Bits... Ctrl-S
Erase Ctrl-E
AVR micro
ATmega32
No Name
Configuration and Security bits
7 6 BootLock12 BootLock11 BootLock02 BootLock01 Lock2 Lock1
OCDEN JTAGEN SPIEN CKOPT EESAVE BOOTSZ1 BOOTSZ0 BOOTRST
BODLEVEL BODEN SUT1 SUT0 CKSEL3 CKSEL2 CKSEL1 CKSEL0
Checked items means programmed (bit = 0)
UnChecked items means unprogrammed (bit = 1)
Refer to device datasheet, please
Cancel OK Write Read

I/O DISTRIBUTION of eCee ATMega32 Development Board

ATMega32PIN No	NAME	TYPE	THE I/O ASSIGN OF ATMega32 Development Board
1	PB.0	I/O	LCD(D0)/ 7SEGMENT
2	PB.1	I/O	LCD(D1) / 7SEGMENT
3	PB.2	I/O	LCD(D2) / 7SEGMENT
4	PB.3	I/O	LCD(D3) / 7SEGMENT
5	PB.4	I/O	LCD(D4) / 7SEGMENT
6	PB.5	I/O	LCD(D5) / 7SEGMENT/ISP(MOSI)
7	PB.6	I/O	LCD(D6) / 7SEGMENT/ISP(MISO)
8	PB.7	I/O	LCD(D7) / 7SEGMENT/ISP(SCK)
9	RESET		RESET SWITCH
10	VCC		+5V
11	GND		GROUND
12	XTAL2		CRYSTAL
13	XTAL1		CRYSTAL
14	PD.0	I/O	MAX232(R20UT)
15	PD.1	I/O	MAX232(T2IN)
16	PD.2	I/O	IR SENSOR/INT
17	PD.3	I/O	BUZZER
18	PD.4	I/O	LCD(RS)/PULLUPKEY1
19	PD.5	I/O	LCD(R/W)/ /PULLUPKEY2
20	PD.6	I/O	LCD(E)/ /PULLUPKEY3
21	PD.7	I/O	PWM //PULLUPKEY4
22	PC.0	I/O	LED1/EEPROM(SCL)/RTC(SCL)
23	PC.1	I/O	LED2/ EPROM(SDA)/RTC(SDA)
24	PC.2	I/O	LED3/JTAG(TCK)
25	PC.3	I/O	LED4/JTAG (TMS)
26	PC.4	I/O	LED5/JTAG(TD0)
27	PC.5	I/O	LED6/JTAG(TDI)/MATRIX
28	PC.6	I/O	LED7/MATRIX
29	PC.7	I/O	LED8/MATRIX
30	AVCC		VCC

31	GND		GND
32	AREF		POTENTIOMETER(AREF)
33	PA.7	I/O	MATRIXKEYPAD
34	PA.6	I/O	MATRIXKEYPAD/ADC POTENTIOMETER I/P
35	PA.5	I/O	MATRIX KEYPAD/TEMP.SENSOR
36	PA.4	I/O	MATRIXKEYPAD
37	PA.3	I/O	7 SEGMENT(DISP4)
38	PA.2	I/O	7 SEGMENT(DISP3)
39	PA.1	I/O	7 SEGMENT(DISP2)
40	PA.0	I/O	7 SEGMENT(DISP1)

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*

LIMITATIONS AND WARRANTEES

This product is intended for personal or lab experimental purpose and in no case should be used where it harmfully effect human and nature. No liability will be accepted by the publisher for any consequence of its use. Use of the product software and or hardware is with the understanding that any outcome whatsoever is at the users own risk. All products are tested for their best performance before shipping, still rhydoLABZ is offering One year Free service warranty (Components cost + Shipping cost will be charged from Customer).

DISCLAIMER

Rhydo Technologies (P) Ltd.
(An ISO 9001:2008 Certified R&D Company)
Golden Plaza, Chitoor Road,
Cochin – 682018, Kerala State, India
Phone : 0091- 484-2370444, 2371666
Cell : 0091- 99466 70444
Fax : 0091 - 484-2370579
E-mail : info@rhydolabz.com, sales@rhydolabz.com
WebSite : http://www.rhydolabz.com

rhydolabz.com