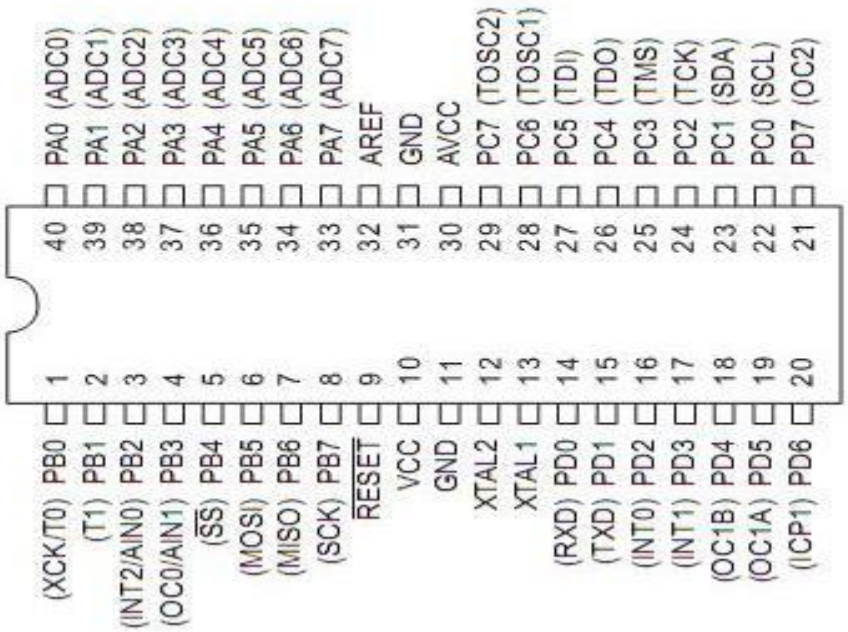
EMBEDDED HARDWARE DESIGN

Experiment 1A **Interfacing PC with Microcontroller** **2009-10/I**

This will be the first experiment that deals with microcontrollers. In this experiment you are supposed to learn how to write and compile C code for ATMega16 in Programmers Notepad and generate Hex file. In addition you will dump this Hex file into the microcontroller, run the program and observe its output.



1. **Connections:**

   (a)Take a 8 wire female connector and put it in PORTD connector, with Black wire connecting to PD0.

   (b) Connect Black wire to anode(Longer Leg) of LED on a breadboard. Connect cathode of LED to a 200 ohm resistance.

   (c) Connect the 3 pin female wire into LCD-JUMP connector. Connect together pin3(Red wire) and other end of 200 ohm resistance to make ground common.

   Repeat the same connections for Brown wire and Red wire of the 8 wire connector.

2. **Writing Test programs:**

   (a)Open Programmer Notepad and write a small test code as given below, save it in a new directory.

   (b) Generate or copy Makefile and save in the same directory. Compile the code in Programmers Notepad and generate Hex file.

   (c)Follow the procedure of dumping it as describe in Programming AVR Microcontroller Document.

(d)Observe and notedown the output pattern of LED's.
(e)Change the value of argument of _delay_ms( ) from 800/500 to several other values.
(f) Change the conditions of for loop and observe the resulting change.
After this observation what conclusion can you draw about the functionality of the program.

```
#include<avr/io.h>
#include<avr/delay.h>
#define PORT PORTD
#define DDR  DDRD
int main(void)
{
        DDR = 0xFF;
        PORT = 0x00;
        while(1)
        {
                for( int i =0 ; i<8 ; i++)
                        PORT = (1<<i),_delay_ms(800);
                for( int i = 7 ; i>=0 ; i--)
                        PORT = (1<<i), _delay_ms(500);
        }
}
```

(g) Try writing and dumping several other codes as told in class and observe their outputs.

EMBEDDED HARDWARE DESIGN

Experiment 2A I/O Port Interfacing in Microcontroller 2009-10/I

In this experiment we will do interfacing of Input and Output pins of microcontroller. Your task is to configure a PORT pin as input (both using internal and external pull up resistance) and control output of another PORT pin depending on the value of input.

1. **Connections:**
   (a) Connect a three pin female connector to LCD-JUMP connector. The Red wire is GROUND and Black wire is VCC. Take these wires on to breadboard.
   (b) Connect a LED to a PORT pin (PD0) with 200 ohm resistance in series.
   (c) Take a breadboard and connect a switch on it, identify the 2 terminals of the switch which are shorted internally. The other 2 terminals should get shorted when the switch is pressed.
   (d) In case of **Internal Pull-up** resistance, connect one pair of shorted terminals to ground and other pair of shorted terminals to a PORT pin (PD1).
   In case of **no Internal Pull-up** resistance, connect one pair of shorted terminals to ground and other pair of shorted terminals to Vcc through a 10Kohm resistance. Connect the terminals connected to pull-up resistance to PORT pin(PD1).
2. **Procedure:**
   (a) Configure PD0 as output and initialize its output value as low.
   (b) Configure PD1 as input without internal pull resistance i.e. Pull Up Disable. Write the main loop to continuously poll the input value at pin PD1. If the input voltage is HIGH , turn on the LED on PD0. If the input voltage is LOW, turn off the LED on PD0.
   (c) Repeat step 2(b) with Internal Pull up resistance enabled.

   Congratulations you have just completed the "**Hello World**" Program with microcontrollers !!

Experiment 2b.

In experiment 2a you had learned how to use the I/O ports of the AVR controller. In this experiment we try to improvise a bit asking if it is possible to take in inputs from sources other than pushbuttons as well as inputs on multiple pins.

1. Instead of a pushbutton send a square shaped waveform from a function generator at 1Hz frequency. Switch ON or OFF an LED in correspondence with the high and low of the input waveform