****

**COMSATS UNIVERSITY ISLAMABAD**

**MICROPROCESSOR SYSTEMS AND INTERFACING**

**LAB REPORT 2**

**SUBMITTED TO:**

SIR KHIYAM IFTIKHAR

**SUBMITTED BY:**

JUNAID AHMAD

Ibrar ahmad

Jarrar malik

**REGISTRATION NO:**

CIIT/FA19-BEE-089/ISB

CIIT/fa19-bee-083/isb

CIIT/fa19-bee-087/isb

**DATE:**

01-10-2021

## Introduction to AVR Microcontroller Hardware Circuitry and Digital I/O Ports

**Objectives:**

* Understand the minimal circuit required to start using a microcontroller
* Learn to program (download code to program memory of) a microcontroller using Arduino board.
* To understand and use digital I/O ports of AVR microcontroller

**Required Tools:**

**Software Tools:**

* Microchip Studio (Version 7) or AVR Studio (Version 4)
* Proteus ISIS
* Arduino IDE
* AVRDUDESS

**PRE-LAB:**

**Task 1:**

Enlist 5 peripherals (Devices) that can be interfaced with Atmega328P Microcontroller.

1. LCD display
2. Keypad
3. Control motor
4. It can be used to control for IOT things
5. Analog input to digital output

**In Lab:**

**Task 1:**

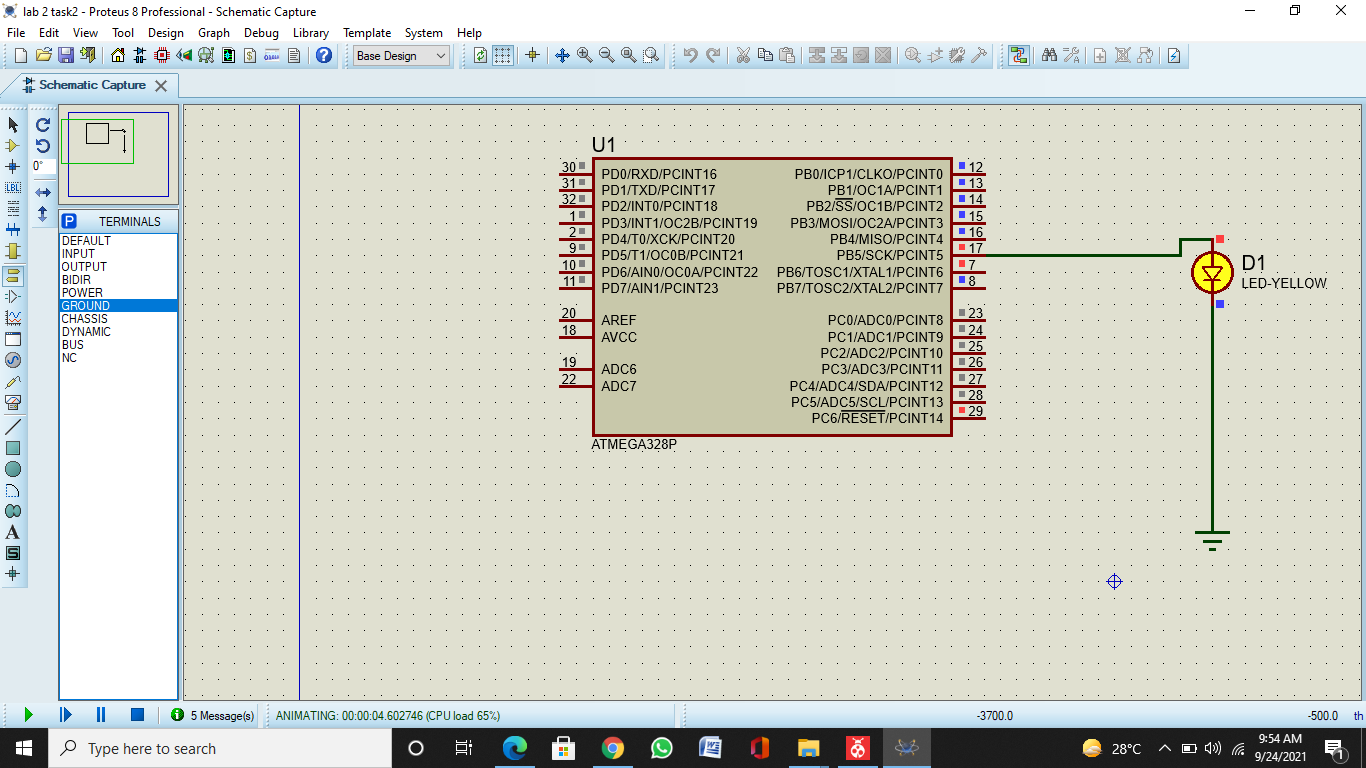
**Write and test following program in AVR Studio/Atmel Studio. Also, simulate it on Proteus.**

|  |
| --- |
| **/\*This code will configure Port B for output and then toggle**  **Port B pin 3 with a delay of 1000ms\*/**  **#include <avr/io.h> /\*This header file includes the apropriate**  **I/O definitions for the device \*/**  **#define F\_CPU 16000000UL //XTAL Frequency =16MHz**  **#include <util/delay.h> /\*This header file is required for generating delays\*/ int main()**  **{ // Write your code here**  **DDRB = 0b11111111; //Configure Port B as Output while(1) //This loop will run forever**  **{**  **PORTB = 0b00001000; /\*Send 1 on Port B Pin 3, rest of the pins are zero\*/ \_delay\_ms(1000); //Delay of 1000ms**  **PORTB = 0b00000000; //Send 0 on Port B Pin 3**  **\_delay\_ms(1000); //Delay of 1000ms**  **} }** |

**A screenshot of a computer

Description automatically generated**

**Hex file to ATMega328P:**

****

**Burning .Hex file to ATMega328P:**

Graphical user interface, text, application

Description automatically generated

A picture containing text, electronics, circuit

Description automatically generated

**Task 2:**

**Switches are connected to one port of ATmega328p for input and LEDs are connected to another port for output. Using these, perform a task assigned by your lab instructor. Students should write the code for given task, simulate it on proteus and then implement it on hardware.**

A picture containing graphical user interface

Description automatically generated

/\*

\* LAB2 TASK2.c

\*

\* Created: 12/12/2021 3:25:38 PM

\* Author : AEGON

\*/

#ifndef *F\_CPU*

#define *F\_CPU* 16000000UL // telling controller crystal frequency (16 MHz AVR ATMega328P)

#endif

#include <avr/io.h> //header to enable data flow control over pins. Defines pins, ports, etc.

#include <util/delay.h> //header to enable delay function in program

#define BUTTON1 1 // button switch connected to port B pin 1

#define LED1 0 // Led1 connected to port B pin 0

#define LED2 1 // Led2 connected to port C pin 1

#define LED3 2 // Led3 connected to port D pin 2

#define DEBOUNCE\_TIME 25 // time to wait while "de-bouncing" button

#define LOCK\_INPUT\_TIME 300 // time to wait after a button press

void init\_ports\_mcu()

{

DDRB=0xFF; // Set all pins of the PORTB as output.

DDRB &= ~(1<<BUTTON1);//Makes first pin of PORTB as Input

PORTB = 0xFF; // Set all pins of the PORTB as HIGH. Led is turn on, also the internal Pull Up resistor of first pin PORTB is enable.

DDRC=0xFF; // Set all pins of the PORTC as output.

PORTC=0x00; // Set all pins of PORTC low which turns it off.

DDRD=0xFF; // Set all pins of the PORTD as output.

PORTD=0x00; // Set all pins of PORTD low which turns it off.

}

unsigned char button\_state()

{

/\* the button is pressed when BUTTON1 bit is clear \*/

if (!(PINB & (1<<BUTTON1)))

{

*\_delay\_ms*(DEBOUNCE\_TIME);

if (!(PINB & (1<<BUTTON1))) return 1;

}

return 0;

}

int main (void)

{

unsigned char n\_led = 1; // LED number is on now

init\_ports\_mcu();

while (1)

{

if (button\_state()) // If the button is pressed, toggle the LED's state and delay for 300ms (#define LOCK\_INPUT\_TIME)

{

switch(n\_led){

case 1:

PORTB ^= (1<<LED1); // toggling the current state of the pin LED1. LED1 is turn off.

PORTC ^= (1<<LED2); // toggling the current state of the pin LED2. LED2 is turn on.

break;

case 2:

PORTC ^= (1<<LED2); // toggling the current state of the pin LED2. LED2 is turn off.

PORTD ^= (1<<LED3); // toggling the current state of the pin LED3. LED3 is turn on.

break;

case 3:

PORTD ^= (1<<LED3); // toggling the current state of the pin LED3. LED3 is turn off.

PORTB ^= (1<<LED1); // toggling the current state of the pin LED1. LED1 is turn on.

n\_led=0; // reset LED number

break;

}

n\_led++; // next LED is turn on

*\_delay\_ms*(LOCK\_INPUT\_TIME);

}

}

return (0);

}

**CRITICAL ANALYSIS:**

In this lab we got to know about AVR Microcontroller. we also understood ARDUINO UNO characteristics of Reset Switch Button, TX, Rx, USB Port, and its pinout configuration. Moreover, ATMega328P has 23 I/O pins to interact with outside world in the form of logic values. Each of the I/O port is associated with three I/O registers i.e. DDRx, PORTx, PINx where x is the port B,C or D. AVRDUDESS was used to upload the hex file generated by software programming in AVR microcontroller. Firstly, we built the program given in the lab manual on AVR Studio and simulated it on proteus to check the functioning of LED according to the given code. The LED started blinking with a certain delay of 1msec given in the code The settings (baud rate, COM port etc) were modified in the device manager window. After clicking the program button, the program got burned into ATMega328p memory and we could see the LED blinking with a certain time delay.