Lab 5 – 8051 I/O ports

1. Write an embedded C program to send values 00 – FF to port P0 of 8051.

***#include <reg51.h> // Header file for 8051 microcontroller***

***void main(void) {***

***unsigned char value; // Variable to store the current value***

***while(1) { // Infinite loop***

***for(value = 0x00; value <= 0xFF; value++) { // Loop from 0x00 to 0xFF***

***P0 = value; // Output the current value to Port 0***

***}***

***}***

***}***

1. Write an embedded C program to toggle all bits of P2 continuously.

***#include <reg51.h>***

***void delay(unsigned int count);***

***void main(void)***

***{?***

***while(1){***

***P2 = ~P2;***

***delay(1000);***

***}***

***}***

***void delay (unsigned int count)***

***{***

***int i,z;***

***for(i =0; i<count;i++)***

***{***

***for(z=0;z<500;z++);***

***}***

***}***

Verify it in hardware.

1. Write an embedded C program to toggle bit D0 of port P2 (P2.0) 50,000 times. Verify it in hardware.

***#include <reg51.h> // Header file for 8051 microcontroller***

***void main(void) {***

***unsigned int i; // Variable to keep track of the number of toggles***

***for(i = 0; i < 50000; i++) { // Loop 50,000 times***

***P2^0 = ~P2^0; // Toggle bit P2.0 (D0 of Port P2)***

***}***

***while(1); // Infinite loop to stop the program after 50,000 toggles***

***}***

1. Write an embedded C program to toggle bits of P0 port continuously with a 500 ms delay. Verify it in hardware.

***#include <reg51.h> // Header file for 8051 microcontroller***

***void delay\_500ms(void) {***

***unsigned int i, j;***

***for(i = 0; i < 50; i++) {***

***for(j = 0; j < 1000; j++) {***

***// Each iteration of this loop takes approximately 10 microseconds***

***// when using an 11.0592 MHz crystal frequency.***

***}***

***}***

***}***

***void main(void) {***

***while(1) { // Infinite loop***

***P0 = ~P0; // Toggle all bits of Port 0***

***delay\_500ms(); // Wait for 500 milliseconds***

***}***

***}***

1. Write an embedded C program to configure port 0 as an input port. This port is read continuously and assigned to P1 if the value is less than 100 else the value is output on P2.

***#include <reg51.h> // Header file for 8051 microcontroller***

***void main(void) {***

***unsigned char value; // Variable to store the value read from P0***

***while(1) { // Infinite loop***

***P0 = 0xFF; // Configure Port 0 as input by writing 0xFF***

***value = P0; // Read the value from Port 0***

***if(value < 100) { // Check if the value is less than 100***

***P1 = value; // Assign the value to Port 1***

***} else {***

***P2 = value; // Assign the value to Port 2***

***}***

***}***

***}***

1. Write embedded C program to put 0x55 on P0 if the P1.7 status is 1, else to put 0xAA.

***#include <reg51.h> // Header file for 8051 microcontroller***

***void main(void) {***

***while(1) { // Infinite loop***

***if(P1^7 == 1) { // Check if P1.7 is 1***

***P0 = 0x55; // Output 0x55 to Port 0***

***} else {***

***P0 = 0xAA; // Output 0xAA to Port 0***

***}***

***}***

***}***

1. Implement binary counter in P1 using embedded C program

***#include <reg51.h> // Header file for 8051 microcontroller***

***void delay(void) {***

***int i, j;***

***for(i = 0; i < 1000; i++)***

***for(j = 0; j < 100; j++);***

***}***

***void main() {***

***unsigned char counter = 0; // Initialize counter to 0***

***while(1) { // Infinite loop***

***P1 = counter; // Output the counter value to Port 1***

***counter++; // Increment the counter***

***delay(); // Wait for a short period***

***}***

***}***