# Name: Neelangshu Roy

# Group: C1

Reg. No.: 20214060

# Embedded System Assignment 8

Edsim51

1. Code:

ORG 00H

MOV R0, #16 ; Initialize counter with 16

MOV A, #your\_16\_bit\_data ; Load your 16-bit data into Accumulator CLR C ; Clear carry flag

LOOP: RRC A ; Rotate Accumulator right through carry JNC SKIP ; If carry = 0, skip the increment

INC R1 ; If carry = 1, increment R1

SKIP: DJNZ R0, LOOP ; Decrement R0. If R0 != 0, go to label LOOP

; Now, R1 contains the number of 1's in the binary data

; Code to display R1 on a seven-segment display MOV P2, #0FFH ; Clear port P2

MOV A, R1 ; Move the count to Accumulator

MOV DPTR, #LUT ; Load the address of the lookup table

MOVC A, @A+DPTR ; Get the seven-segment code from the lookup table MOV P2, A ; Display the count on the seven-segment display

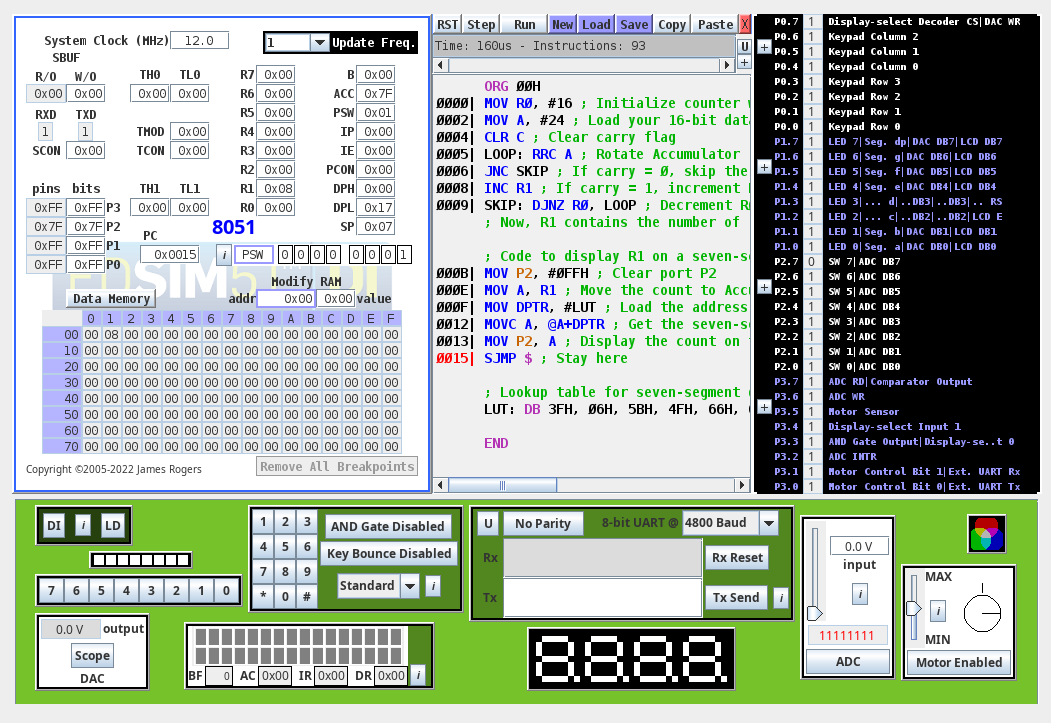
SJMP $ ; Stay here

; Lookup table for seven-segment display (common cathode)

LUT: DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H ; 0-9, A-F

END

Output:



2. Code:

ORG 00H

MOV TMOD, #10H ; Timer 1 in mode 1

MOV TH1, #0FCH ; Load TH1 with the initial count SETB TR1 ; Start Timer 1

HERE: JB P1.5, $ ; Wait for P1.5 to go low ACALL DELAY ; Call delay

CPL P1.5 ; Toggle P1.5

SJMP HERE ; Stay here

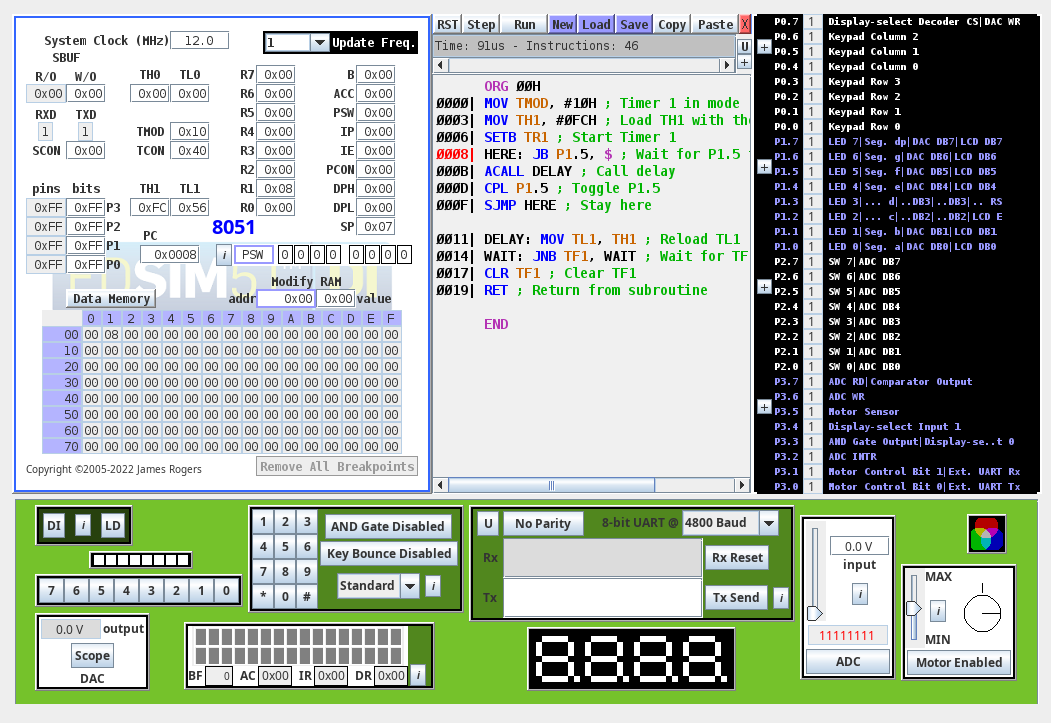
DELAY: MOV TL1, TH1 ; Reload TL1

WAIT: JNB TF1, WAIT ; Wait for TF1 to go high CLR TF1 ; Clear TF1

RET ; Return from subroutine

END

Output:



3. Code:

ORG 00H

MOV TMOD, #10H ; Timer 1 in mode 1

MOV TH1, #0FCH ; Load TH1 with the initial count SETB TR1 ; Start Timer 1

HERE: MOV A, P0 ; Get 8-bit data from P0 MOV P1, A ; Send it to P1

JB P2.1, $ ; Wait for P2.1 to go low ACALL DELAY ; Call delay

CPL P2.1 ; Toggle P2.1

SJMP HERE ; Stay here

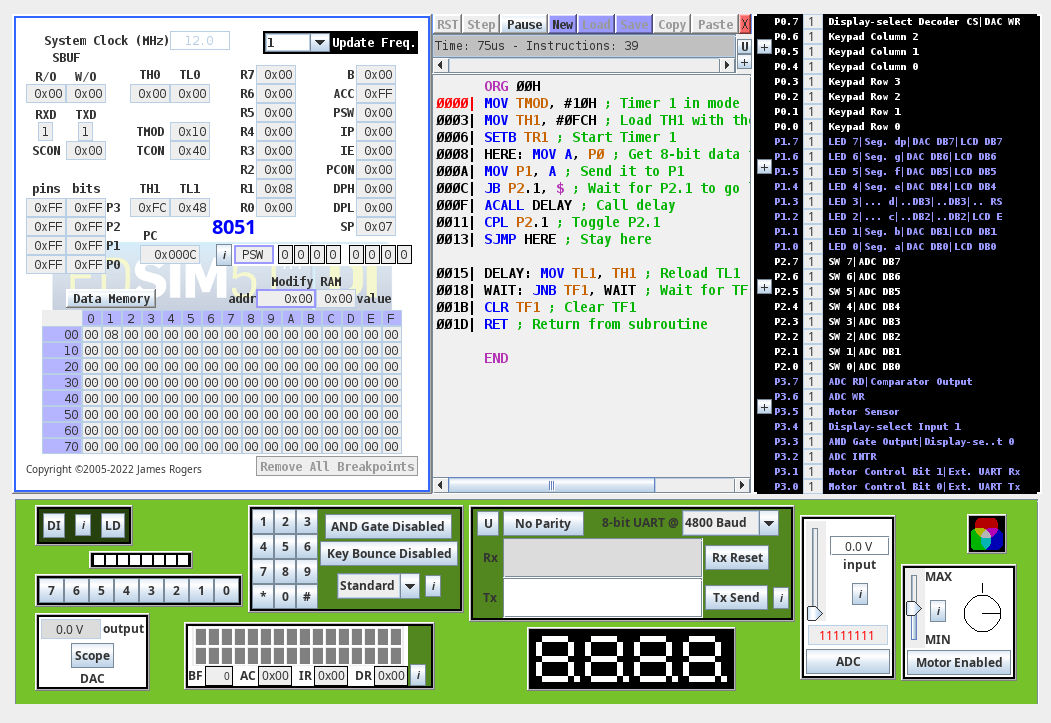
DELAY: MOV TL1, TH1 ; Reload TL1

WAIT: JNB TF1, WAIT ; Wait for TF1 to go high CLR TF1 ; Clear TF1

RET ; Return from subroutine

END

Output:



Keil:

1. Code:

#include <reg51.h> // Include the 8051 register definitions

// Use the sfr keyword to declare the port addresses sfr P0 = 0x80;

sfr P1 = 0x90; sfr P2 = 0xA0;

void delay\_ms(unsigned int ms) {

// Function to generate a delay of ms milliseconds unsigned int i, j;

for (i = 0; i < 10\*ms; i++)

for (j = 0; j < 1275; j++); // Approximately 1 ms delay

}

void main() { while (1) {

P0 = ~P0;

//Toggle all the bits of P0

P1 = ~P1;

// Toggle all the bits of P1

P2 = ~P2;

// Toggle all the bits of P2 delay\_ms(250);

// Wait for 250 ms

}

}

2. Code:

#include <reg51.h> // Include the 8051 register definitions

// Use the sfr keyword to declare the port addresses sfr P0 = 0x80;

sfr P2 = 0xA0;

void delay\_ms(unsigned int ms) {

// Function to generate a delay of ms milliseconds unsigned int i, j;

for (i = 0; i < ms; i++)

for (j = 0; j < 1275; j++); // Approximately 1 ms delay

}

void main() {

while (1) {

P0 = ~P0; // Toggle all the bits of P0 using the inverting operator P2 = P2 ^ 0xFF; // Toggle all the bits of P2 using the Ex-OR operator delay\_ms(250); // Wait for 250 ms

}

}