Liam Spinner

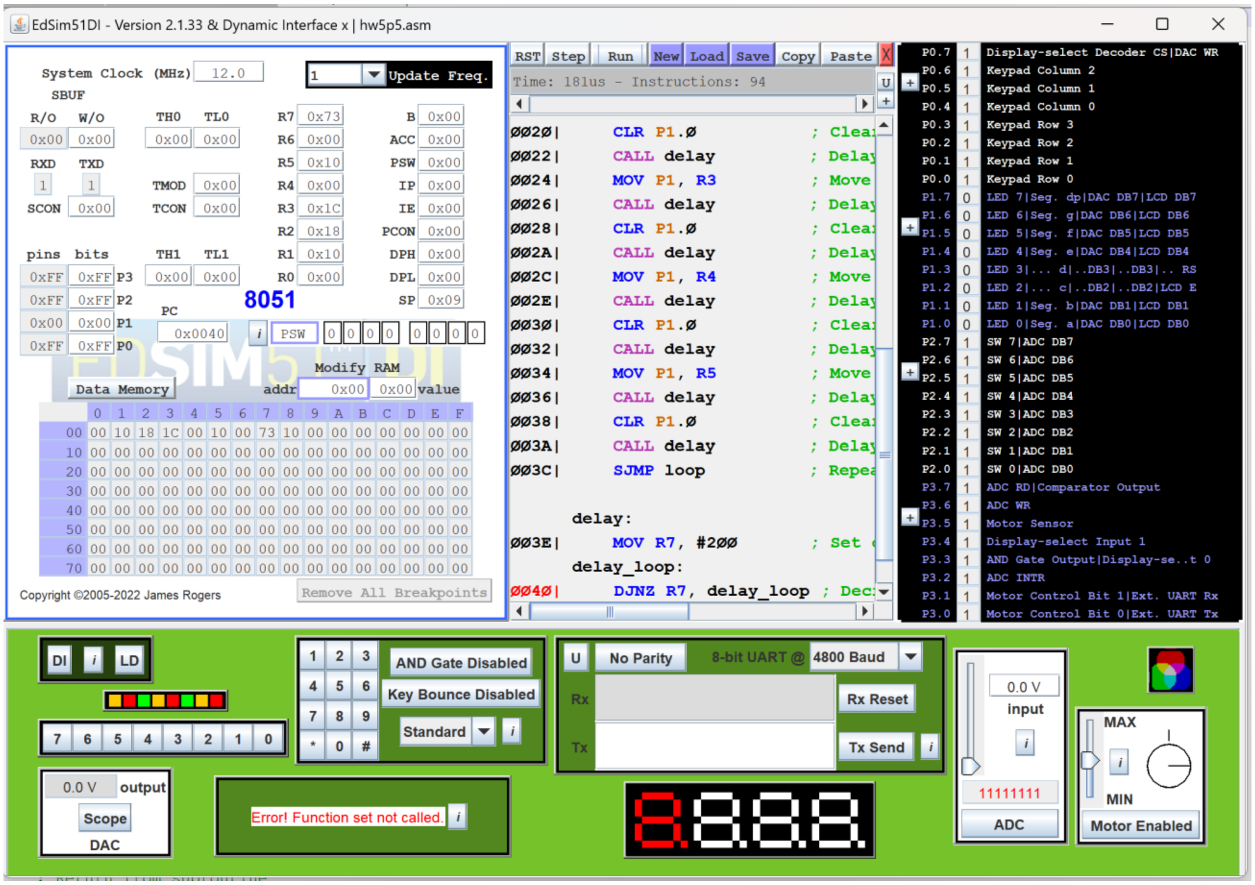
CSE2120

Dr. Caraway

24 Apr 2024

Homework 5.5

Build a basic assembly code program to drive the Ford Mustang motor now being used in ECE Senior Design. Six transistors are used to drive the three phase motor. The task at hand is to turn the six transistors on and off following a distinct pattern. The switching pattern is illustrated in the paper "Traction Inverter Design" on pages six and seven, table 1 and figure 6. Be sure and include a Flow Chart.



EdSim51DI - Version 2.1.33 & Dynamic Interface x | hw5p5.asm

System Clock (MHz) 12.0 — Update Freq. 1

Time: 181us - Instructions: 94

```
ØØ2Ø|     CLR P1.Ø              ; Clea
ØØ22|     CALL delay            ; Delay
ØØ24|     MOV P1, R3            ; Move
ØØ26|     CALL delay            ; Delay
ØØ28|     CLR P1.Ø              ; Clea
ØØ2A|     CALL delay            ; Delay
ØØ2C|     MOV P1, R4            ; Move
ØØ2E|     CALL delay            ; Delay
ØØ3Ø|     CLR P1.Ø              ; Clea
ØØ32|     CALL delay            ; Delay
ØØ34|     MOV P1, R5            ; Move
ØØ36|     CALL delay            ; Delay
ØØ38|     CLR P1.Ø              ; Clea
ØØ3A|     CALL delay            ; Delay
ØØ3C|     SJMP loop             ; Repea

      delay:
ØØ3E|     MOV R7, #2ØØ          ; Set
      delay_loop:
ØØ4Ø|     DJNZ R7, delay_loop   ; Dec
```

```
SBUF  R/O 0x00  W/O 0x00
RXD 1   TXD 1   SCON 0x00
TH0 0x00  TL0 0x00  TMOD 0x00  TCON 0x00
TH1 0x00  TL1 0x00
pins  bits
0xFF  0xFF  P3
0xFF  0xFF  P2
0x00  0x00  P1
0xFF  0xFF  P0
PC 0x0040
R7 0x73   B    0x00
R6 0x00   ACC  0x00
R5 0x10   PSW  0x00
R4 0x00   IP   0x00
R3 0x1C   IE   0x00
R2 0x18   PCON 0x00
R1 0x10   DPH  0x00
R0 0x00   DPL  0x00
          SP   0x09
PSW 0 0 0 0  0 0 0 0
Modify RAM  addr 0x00  0x00 value
```

```
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 00 10 18 1C 00 10 00 73 10 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

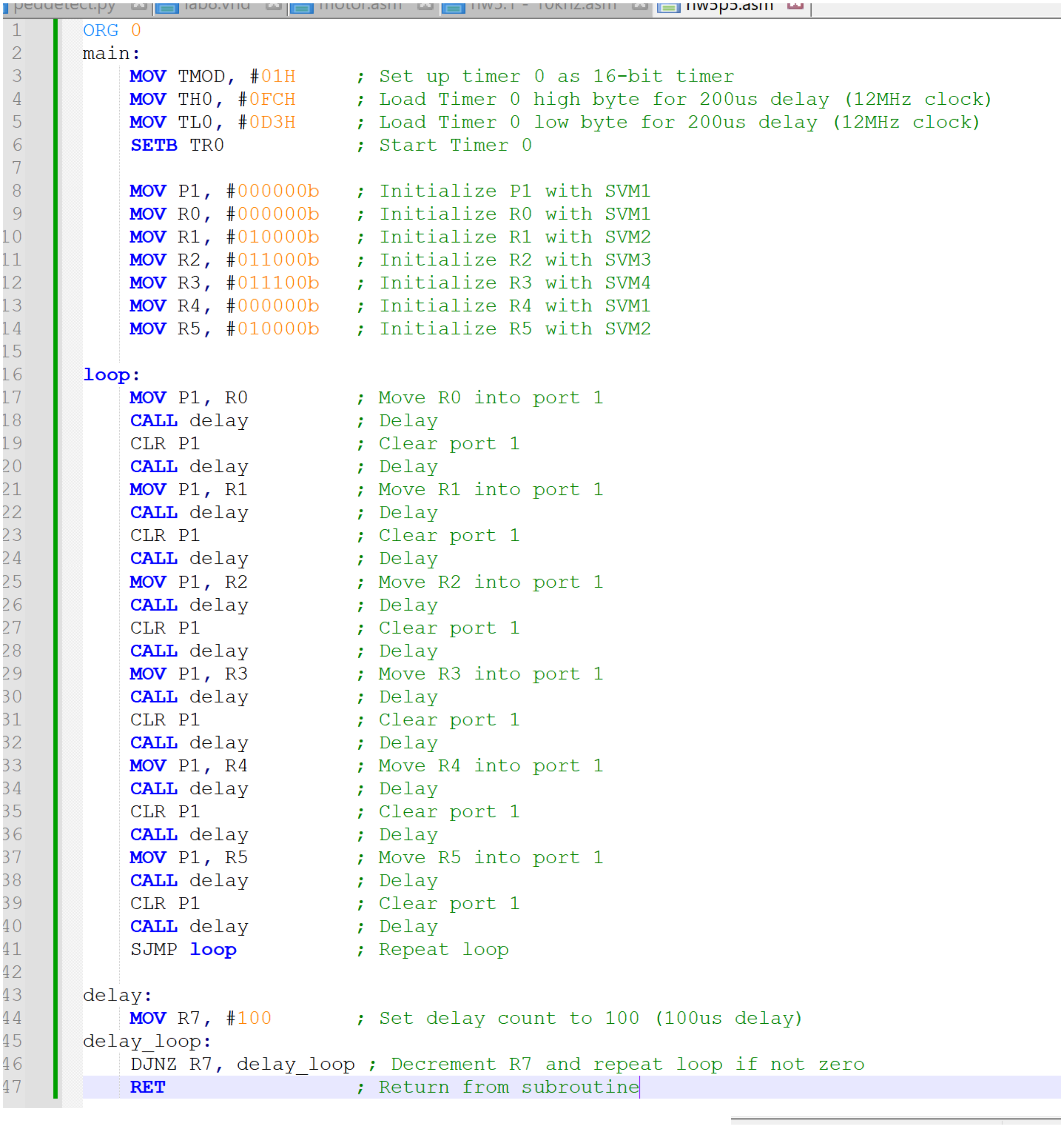
```
P0.7 1 Display-select Decoder CS|DAC WR
P0.6 1 Keypad Column 2
P0.5 1 Keypad Column 1
P0.4 1 Keypad Column 0
P0.3 1 Keypad Row 3
P0.2 1 Keypad Row 2
P0.1 1 Keypad Row 1
P0.0 1 Keypad Row 0
P1.7 0 LED 7|Seg. dp|DAC DB7|LCD DB7
P1.6 0 LED 6|Seg. g|DAC DB6|LCD DB6
P1.5 0 LED 5|Seg. f|DAC DB5|LCD DB5
P1.4 0 LED 4|Seg. e|DAC DB4|LCD DB4
P1.3 0 LED 3|... d|..DB3|..DB3|.. RS
P1.2 0 LED 2|... c|..DB2|..DB2|LCD E
P1.1 0 LED 1|Seg. b|DAC DB1|LCD DB1
P1.0 0 LED 0|Seg. a|DAC DB0|LCD DB0
P2.7 1 SW 7|ADC DB7
P2.6 1 SW 6|ADC DB6
P2.5 1 SW 5|ADC DB5
P2.4 1 SW 4|ADC DB4
P2.3 1 SW 3|ADC DB3
P2.2 1 SW 2|ADC DB2
P2.1 1 SW 1|ADC DB1
P2.0 1 SW 0|ADC DB0
P3.7 1 ADC RD|Comparator Output
P3.6 1 ADC WR
P3.5 1 Motor Sensor
P3.4 1 Display-select Input 1
P3.3 1 AND Gate Output|Display-se..t 0
P3.2 1 ADC INTR
P3.1 1 Motor Control Bit 1|Ext. UART Rx
P3.0 1 Motor Control Bit 0|Ext. UART Tx
```

8-bit UART @ 4800 Baud

DAC output 0.0 V

ADC input 0.0 V — 11111111

Error! Function set not called.

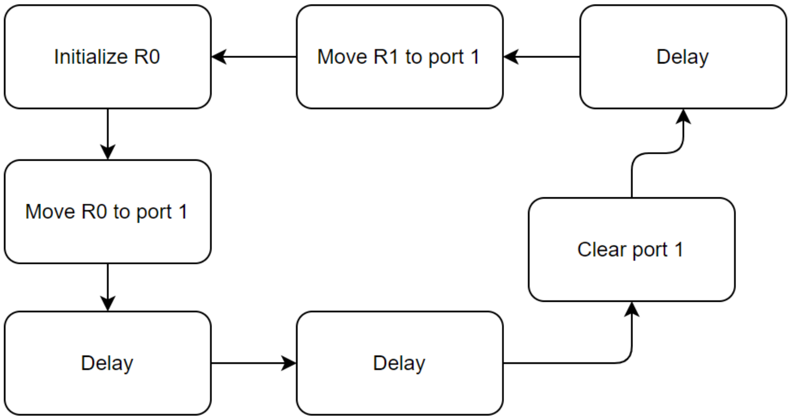


```
ORG 0
main:
    MOV TMOD, #01H      ; Set up timer 0 as 16-bit timer
    MOV TH0, #0FCH      ; Load Timer 0 high byte for 200us delay (12MHz clock)
    MOV TL0, #0D3H      ; Load Timer 0 low byte for 200us delay (12MHz clock)
    SETB TR0            ; Start Timer 0

    MOV P1, #000000b    ; Initialize P1 with SVM1
    MOV R0, #000000b    ; Initialize R0 with SVM1
    MOV R1, #010000b    ; Initialize R1 with SVM2
    MOV R2, #011000b    ; Initialize R2 with SVM3
    MOV R3, #011100b    ; Initialize R3 with SVM4
    MOV R4, #000000b    ; Initialize R4 with SVM1
    MOV R5, #010000b    ; Initialize R5 with SVM2

loop:
    MOV P1, R0          ; Move R0 into port 1
    CALL delay          ; Delay
    CLR P1              ; Clear port 1
    CALL delay          ; Delay
    MOV P1, R1          ; Move R1 into port 1
    CALL delay          ; Delay
    CLR P1              ; Clear port 1
    CALL delay          ; Delay
    MOV P1, R2          ; Move R2 into port 1
    CALL delay          ; Delay
    CLR P1              ; Clear port 1
    CALL delay          ; Delay
    MOV P1, R3          ; Move R3 into port 1
    CALL delay          ; Delay
    CLR P1              ; Clear port 1
    CALL delay          ; Delay
    MOV P1, R4          ; Move R4 into port 1
    CALL delay          ; Delay
    CLR P1              ; Clear port 1
    CALL delay          ; Delay
    MOV P1, R5          ; Move R5 into port 1
    CALL delay          ; Delay
    CLR P1              ; Clear port 1
    CALL delay          ; Delay
    SJMP loop           ; Repeat loop

delay:
    MOV R7, #100        ; Set delay count to 100 (100us delay)
delay_loop:
    DJNZ R7, delay_loop ; Decrement R7 and repeat loop if not zero
    RET                 ; Return from subroutine
```


Initialize R0
Move R1 to port 1
Delay
Move R0 to port 1
Clear port 1
Delay
Delay