CSE 331/EEE 332 (Microprocessor Interfacing & Embedded System Lab)

Lab 06 : Microprocessor – 8086 interfacing with PPI8255A using Proteus
7 Segment Display and Rotate Stepper Motor
Lab Instructor : Rokeya Siddiqua

Topics to be covered in class today:

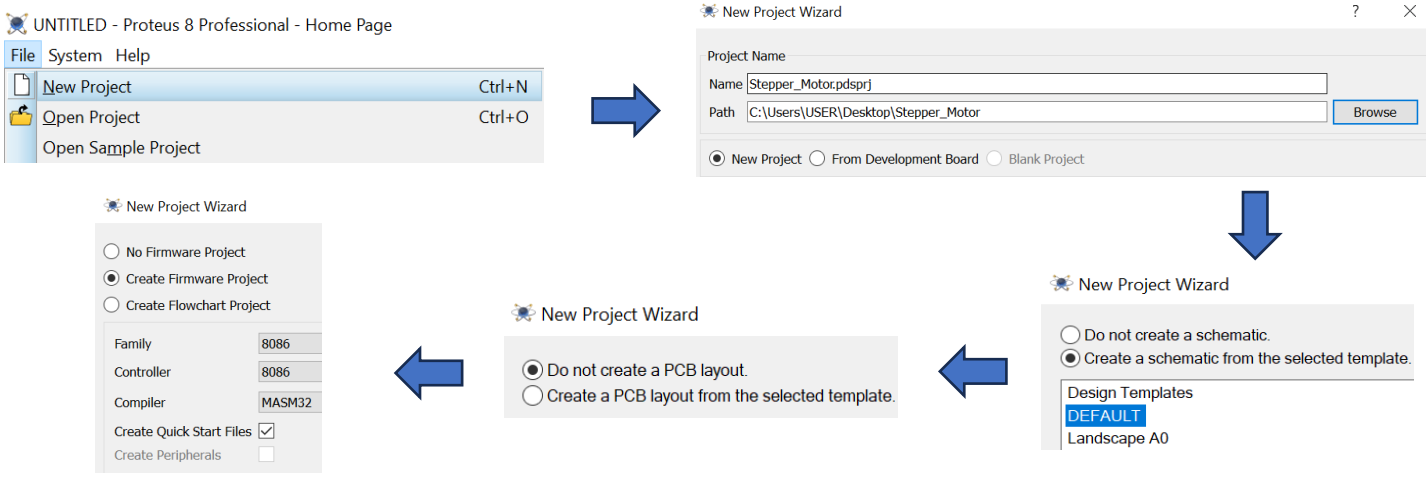
- Microprocessor - 8086 interfacing with PPI8255A
- 7 Segment Display and Rotate Stepper Motor

Components:

Component Details	Specification
Microprocessor	MPU8086
Programmable Peripheral Interface	PPI8255A
7 Segment Display	7SEG-MPX1-CA
Stepper Motor	MOTOR-STEPPER (Unipolar)
Motor Driver	L293D
Latch	74HC373
Logicstate	

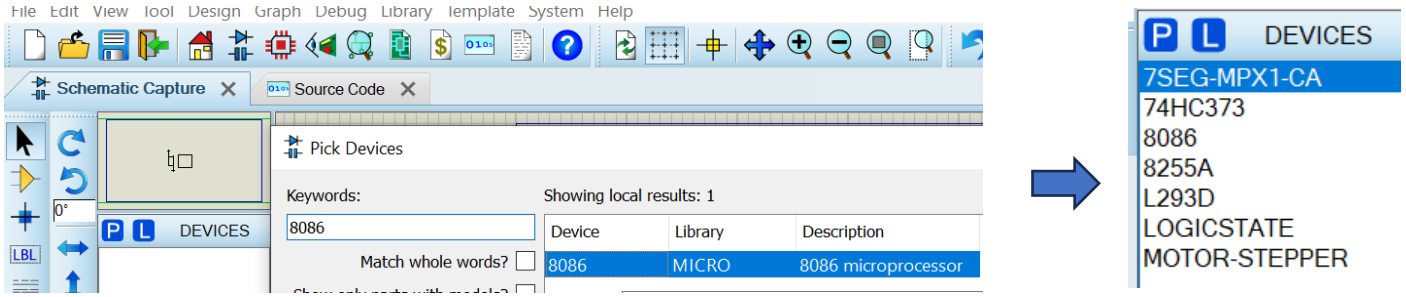
Step 1: Create New Project

Open Proteus software. *File* >> *New Project* >> *Project Name* and *Path*



Step 2: Components

Component Mode >> *Pick Devices* => *8086* (type). Similarly pick other devices listed in the components table.



Step 3: Connections (MPU8086)

GROUND => RESET

POWER => READY + MN/MX'

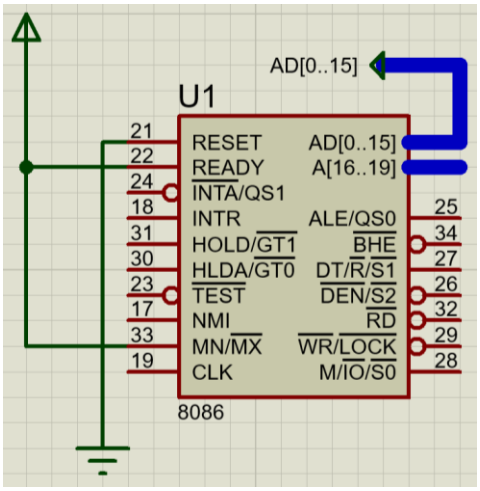
BUS => connect and Label wires

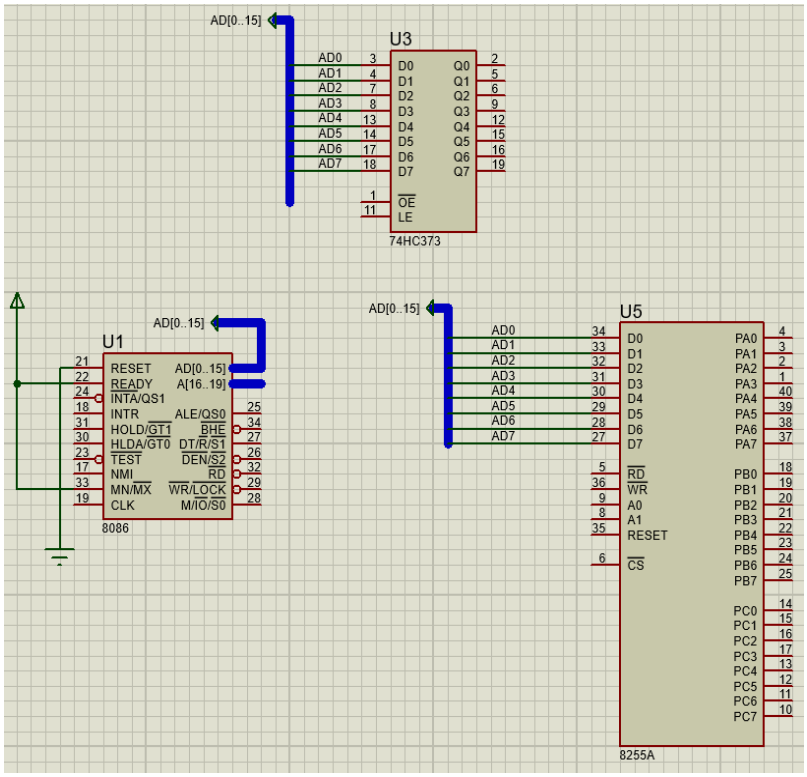
Connect => ALE (MPU8086) + LE (74HC373)

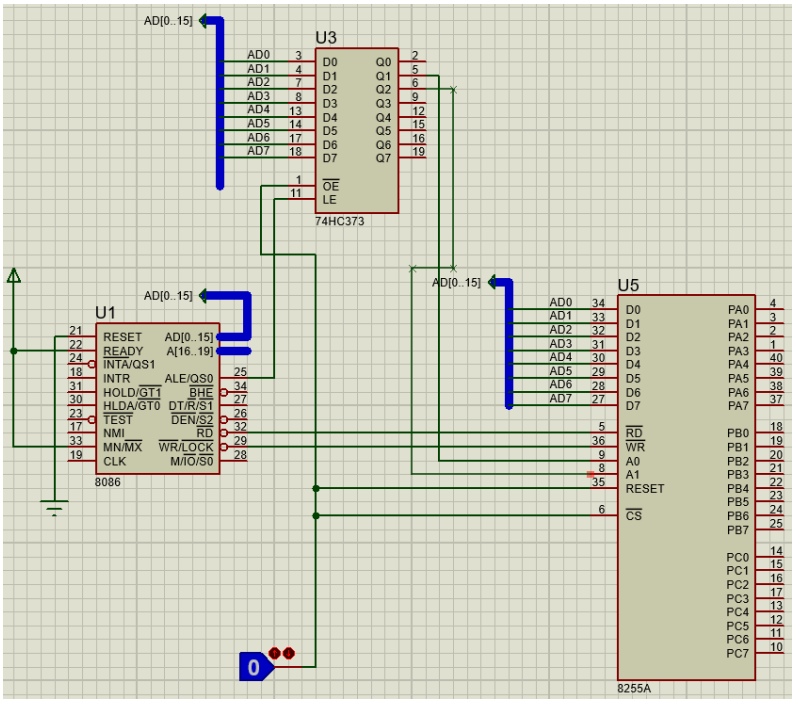
Logicstate(0)=>OE'(74HC373)+ [CS'+RESET(8255A)]

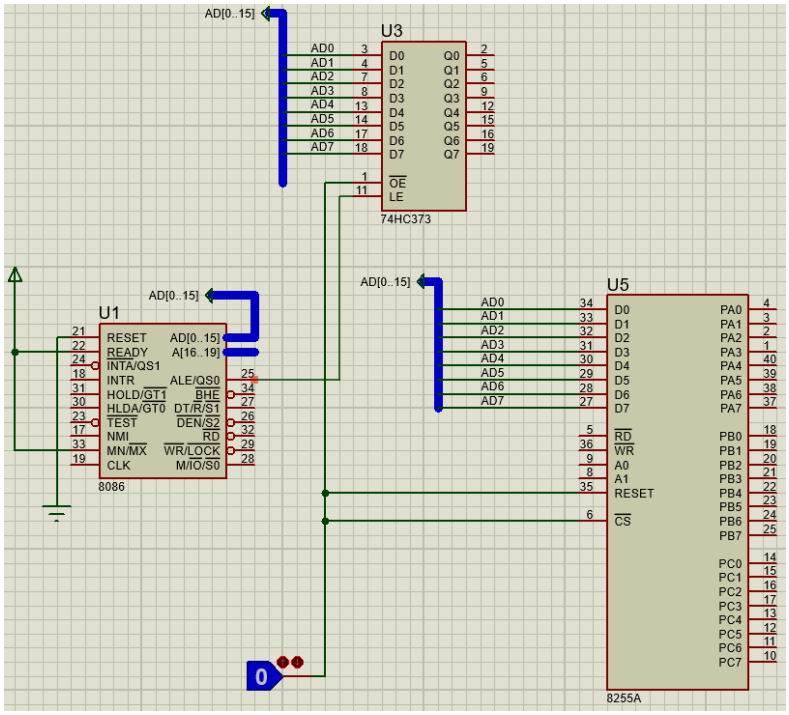
Connect=>[RD' & WR' (8086)]+[RD' & WR' (8255A)]

Connect=>[Q1 & Q2 (74HC373)]+[A0 & A1 (8255A)]



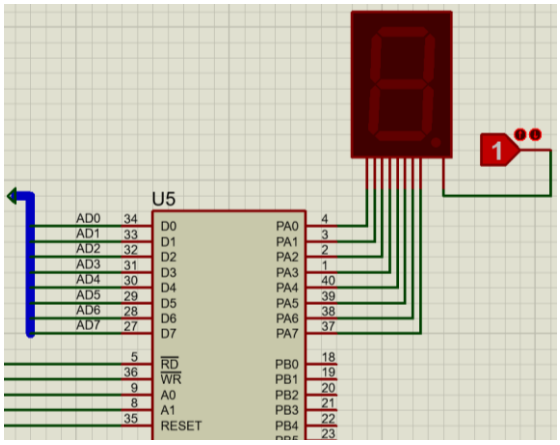


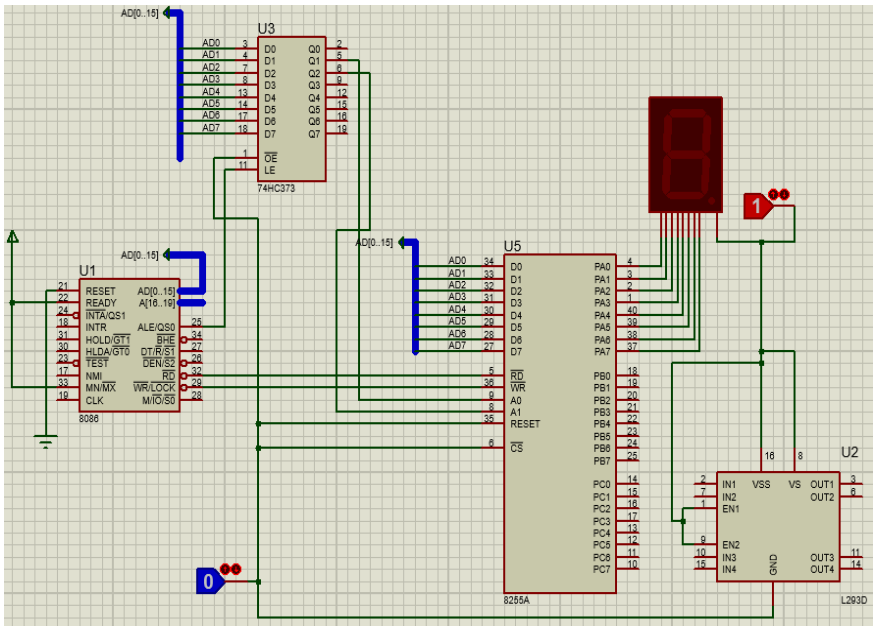


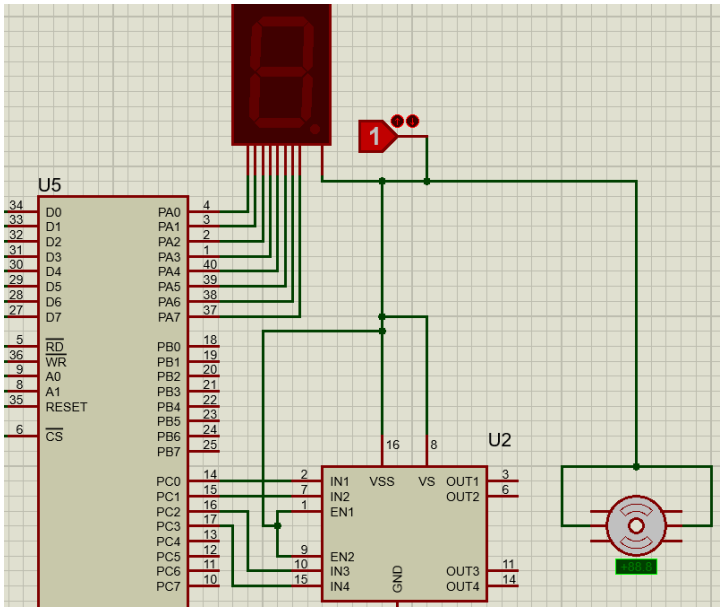


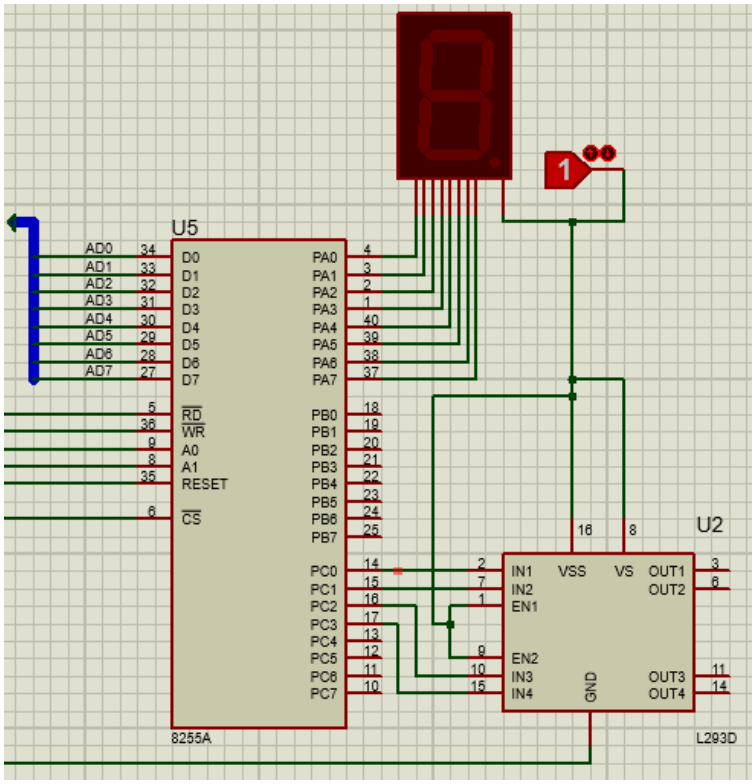
Step 4: Connections (PPI8255A + 7 Segment Display + Stepper Motor)

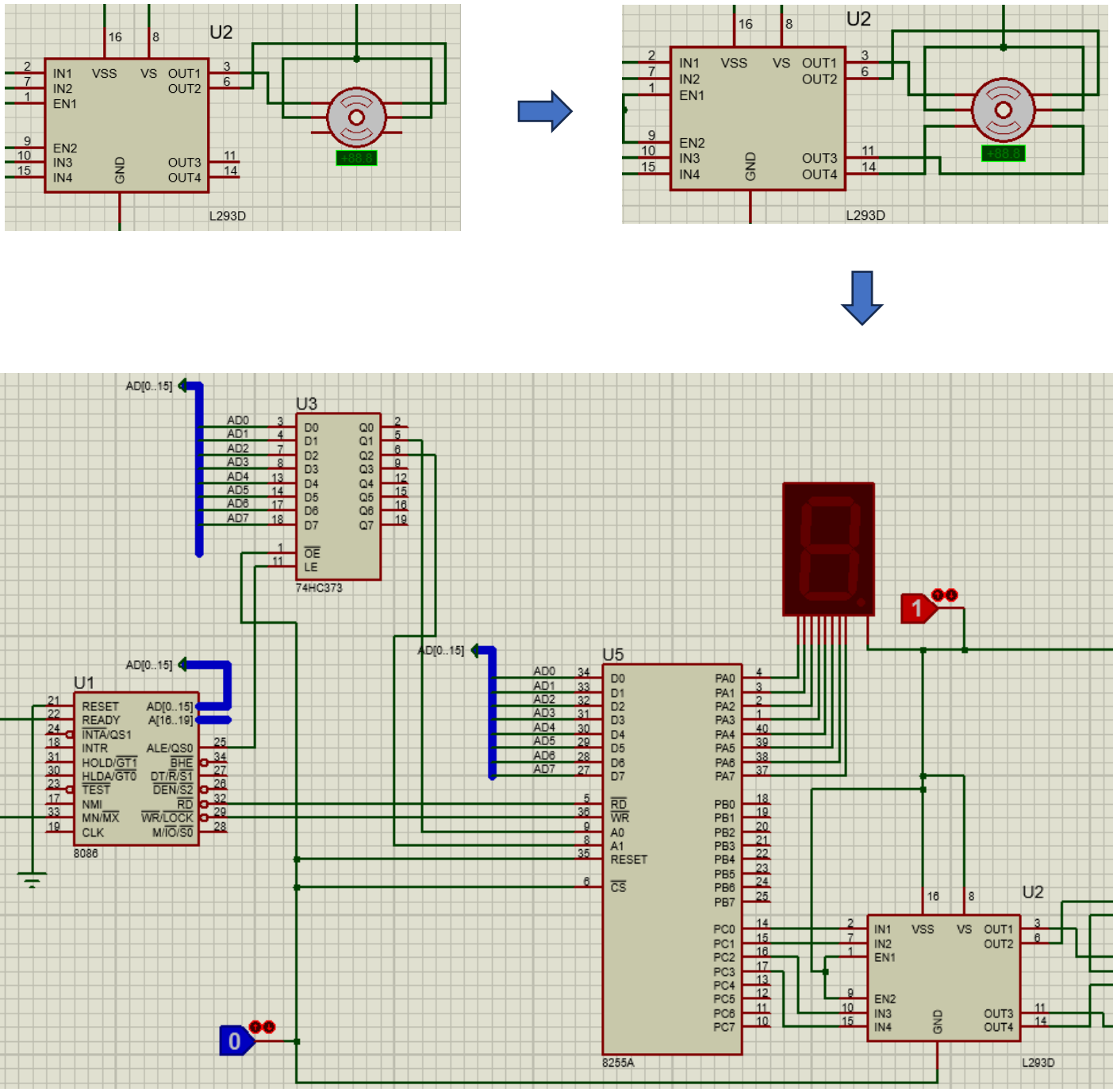
Port A => 7 segment Display (Active High pin)
Logicstate (0) => Ground (L293D)
Connect => VSS + VS + EN1 + EN2 (Active High)
Connect => [PC0, PC1, PC2, PC3 (8255A)] +
[IN1, IN2, IN3, IN4 (L293D)]
Logicstate (1) => Active Motor (Power)










U2
IN1
IN2
EN1
EN2
IN3
IN4
VSS
VS
GND
OUT1
OUT2
OUT3
OUT4
L293D
U3
74HC373
U1
8086
U5
8255A
AD[0..15]
A[16..19]
RESET
READY
INTA/QS1
INTR
HOLD/GT1
HLDA/GT0
TEST
NMI
MN/MX
CLK
ALE/QS0
BHE
DT/R/S1
DEN/S2
RD
WR/LOCK
M/IO/S0
WR
A0
A1
CS
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7

Step 5: Code

StepperMotor - Proteus 8 Professional - Source Code

File Project Build Edit Debug System Help

Schematic Capture | Source Code | Build Project | Debug

Projects

8086(U1)

Source Files

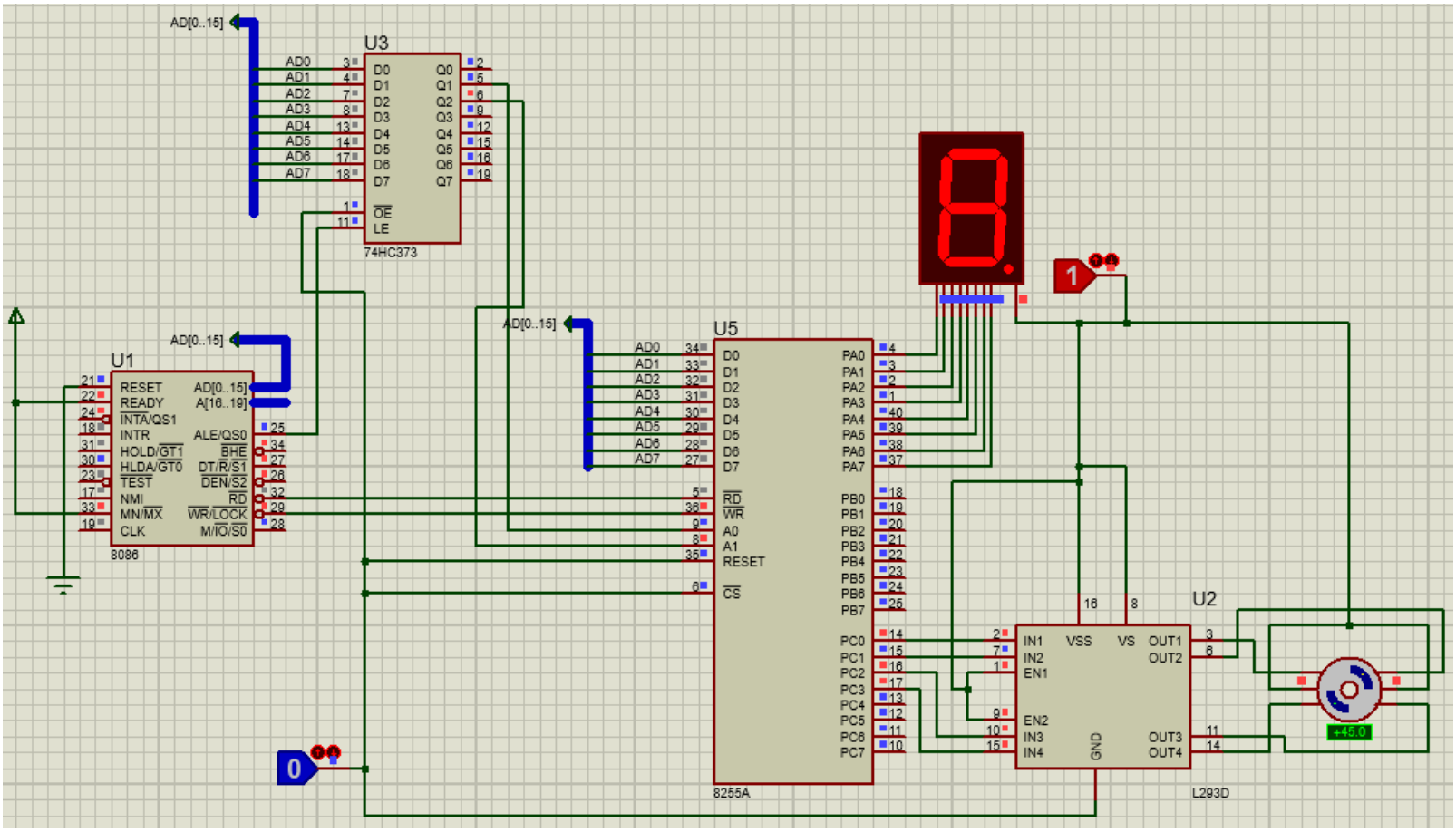
main.asm

main.asm

```
1  DATA SEGMENT
2      PORTA EQU 00H
3      PORTB EQU 02H
4      PORTC EQU 04H
5      PORT_CON EQU 06H
6  DATA ENDS
7
8
9  CODE SEGMENT
10     MOV AX, DATA
11     MOV DS, AX
12
13     ORG 0000H
14
15 START:
16     MOV DX, PORT_CON
17     MOV AL, 10000000B
18     OUT DX, AL
19
20     MOV SI, 0
21     MOV DI, 0
22
23 L0: MOV CX, 1FFFH
24 L1: MOV AL, S1[SI]
25     MOV DX, PORTA
```

VSM Studio Output

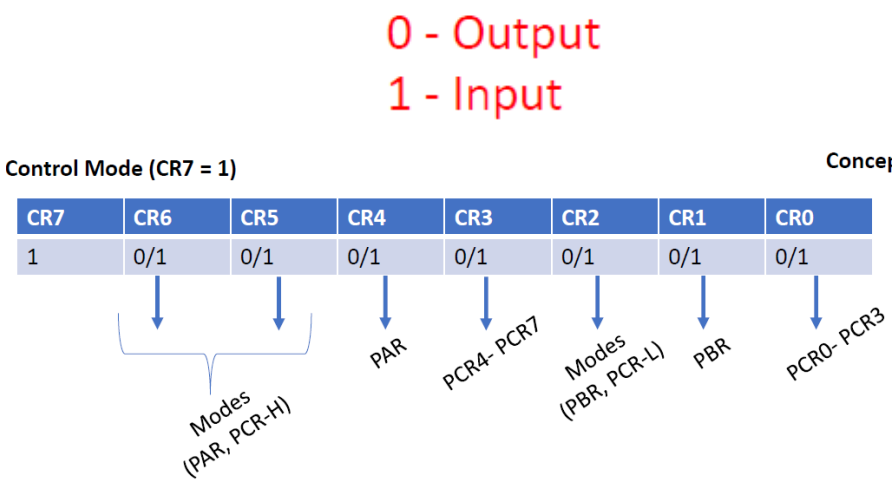
```
LINK : warning L4021: no stack segment
LINK : warning L4038: program has no starting address
Compiled successfully.
```



Code:

DATA SEGMENT ; segment for data

PORTA EQU 00H
PORTB EQU 02H
PORTC EQU 04H
PORT_CON EQU 06H
DATA ENDS ; end of the data segment



CODE SEGMENT ; segment for the code
MOV AX, DATA
MOV DS, AX

ORG 0000H ; origin point for the code in memory

START:
MOV DX, PORT_CON ; Moves the address of the control port (PORT_CON) into the DX register
MOV AL, 10000000B ; AL = 80H => port C (output), port A (output) in mode 0
OUT DX, AL ; sends the value to the control port

MOV SI, 0 ; Initializes the SI (Source Index) register to 0
MOV DI, 0 ; Initializes the DI (Destination Index) register to 0.

L0: MOV CX, 1FFFH ; outter loop label => cx = 8191d
L1: MOV AL, S1[SI] ; inner loop label => Moves a byte of data from the array S1 at the index specified by SI into the AL register.
MOV DX, PORTA ;
OUT DX, AL ; Outputs the content of the AL register to Port A
LOOP L1 ; Decrements CX and repeats the loop (L1) until CX becomes zero.

INC SI ; Increment SI
CMP SI, 16 ; Compares the value in SI with 16
JL L0 ; Jumps back to L0 if SI is less than 16, creating a nested loop.

MOV DX, PORT_CON ; Resets the control port as before.
MOV AL, 10000000B
OUT DX, AL

LL0:MOV CX, 2FFFH ; outer loop => cx = 12287d
LL1:MOV AL, S2[DI] ; Moves a byte of data from the array S2 at the index specified by DI into the AL register.

```
    MOV DX, PORTC ; moves address of Port C into the DX register
    OUT DX, AL ; Outputs the content of the AL register to Port C
    LOOP LL1 ; Decrements CX and repeats the loop (LL1) until CX becomes zero

    INC DI  ; Increment DI
    CMP DI, 4 ; Compares the value in DI with 4
    JL LL0  ; Jumps back to LL0 if DI is less than 4, creating another nested loop

JMP START  ; Jumps back to the START label, effectively creating an infinite loop to continue the program.

ORG 1000H ; beginning of another section of code, starting at address 1000H
S1 DB 11000000B
   DB 11111001B
   DB 10100100B
   DB 10110000B
   DB 10011001B
   DB 10010010B
   DB 10000010B
   DB 11011000B
   DB 10000000B
   DB 10010000B
   DB 10001000B
   DB 10000011B
   DB 11000110B
   DB 10100001B
   DB 10000110B
   DB 10001110B

S2 DB 1101B
   DB 1011B
   DB 0111B
   DB 1110B

CODE ENDS  ; Marks the end of the code segment
END  ; Indicates the end of the program
```

Task: Modify the code to display numbers from 0 to 8 on the 7-segment display for a longer duration, specifically 3 seconds (each number). Subsequently, instruct the motor to perform four rotations.