

2018 - 2019 GÜZ DÖNEMİ
INTRODUCTION TO MICROCOMPUTER
PROJECT

152120131100 _ Gizem Taşcı

GRUP NO:10

Problem:

interface a keypad (use IRQ), 2 seven segment displays, 1 led to display system ON/OFF, 1 walse to fill pool and a pressure sensor . Functions: Control water level in a pool (5m-7m). Water level is a user set value and displayed on two seven segment displays. Water is discharged by another valve by user. Start/stop button : starts and stops operation Mode S : Start/Stop water filling pump Mode+ : Increment water level value Mode - : Decrement water level value Mode CS : Set fixed water level value to 6m.

Çözüm

Donanım olarak tuş takımı,seven segment kullanılacak.

Bunların dışında 8Kx8 Eprom (0000H) ve 2 adet 8155 register kullandık.1.registere tuş takımını 2.registere seven segment displayı bağladık.

Burada bir tuşa art arda basıldığı zaman ortaya çıkacak sıkıntıydı.Bunun için de bu problemi önlemek için süre koyduk.Bunun sebebi bir metin yazmak için, aynı tuşta bulunan farklı harflerin art arda gelebilmesi ihtimaliydi. Örneğin süresiz çalışan bir donanımda '2' tuşuna 3 kez basıldığında program "C" yazacaktır, ancak süreli çalışan bir donanımda 2 kez

basılıp bir müddet beklenildikten sonra tekrar basıldığında bu metin "BA" olacaktır. Bu süreyi 0.5 saniye olarak belirledik. Ardından gelen 'IRQ' lar ile aynı tuşun belirli süre içerisinde kaç kez basıldığını saymak için RAM'e 'COUNT' adlı bir değişken atadık. Kullanıcı hızlı bir şekilde üst üste bastığında ortaya çıkacak sorunu önlemek için de Şöyle bir yol düşündük. Tuşlardaki karakter sayısına göre matematiksel olarak mod alınması oldu. Hepsinin sayesinde basılma adedi ile oluşturduğumuz 'TABLE' lardan gerekli karakterin kodunu almayı düşündük. Butona basıldığı zaman programın bunun hangi tuş olduğunu anlamasını ise alt program yazarak kontrol ettik. Art arda yapılan iki işlemin aynı tuş ya da farklı tuş olup olmadığının farkında olabilmek için de RAM'e PREVBUT ve CURBUT olmak üzere 2 adet değişken atamanın yararlı olacağını düşündük. Bu değişkenler adreslerinde basılan tuşun sayısal değerini tutacak ve ikisi arasında karşılaştırma yapıldıktan sonra aynı tuşun basılıp basılmadığını öğrenebilecektik. Alt program mantığını örnek alarak, benzer şekilde 'Display' fonksiyonlarını da alt programlar ile halledebileceğimizi düşündük ve bunlarını hepsini yaparken bir yandan da program 'Stop Button' basılmış mı diye kontrol etsin diye düşündük.

Programı IRQ mantığı ile yazacağımız için, programın PREVBUT ve CURBUT değişkenlerinin sayısal olarak hafızasında tutması gereken (*programın ilerleyen safhalarında karşılaştırma yapılacağı için*) buton numaralarını nasıl adresine atanacağı üzerinde uzun süre düşündük. TEMPBUT adı altında sadece IRQ içerisinde kullanılan bir değişken daha yarattık. Program IRQ aldığında bu değişkeni kontrol etsin ve basılan tuşun değerini "WHICHBUTTONWASPRESSED" alt programı ile TEMPBUT'a atasin dedik. Ana programa döndüğünde de program TEMPBUT kontrolü için hazırlanan bir döngüye girerek IRQ adına bir hamle yapılmış olsun seçimini yaptık.

Akla takılan ikinci büyük sorun ise programın hexadecimal olarak tuttuğu bir sayıyı seven-segment'lere nasıl decimal olarak yollayacağımızdı. Bu konuda tecrübesiz olduğumuzdan ötürü BCD (Binary Coded Decimal) sayı sistemini kullanmanın en mantıklı yol olacağına karar kıldık. Programda ise; her bir karakter ekrana basıldığında bir artması gereken CHARCOUNTBCD değişkeninin birler basamağını '9' mu diye kontrol etmeliyiz, eğer cevap evet ise 1 arttırmak yerine 6 arttırmalıyız dedik. Örneğin sayı '09' iken hexadecimal sayı 1 arttırıldığında '0A' olacaktır, ancak 6 arttırıldığında '10' olacaktır. Yani sistem hexadecimal işlevler uygularken, kodlar sayesinde görünürde decimale dönüşmüş olur ve istediklerimizi elde edebiliriz dedik.

Programın flowchart aşamasında sona yaklaşmışken büyük bir sorunla karşılaştık. Atadığımız değişkenlerden PREVBUT ve CURBUT tan yazdırma aşamasında CURBUT'u tercih edecektik, fakat kod işlemlerin bitmesini beklemeden harekete geçebilirdi. Bunu önlemek adına PREVBUT değişkenindeki değeri yazdırmaya karar verdik. Yani program ya 0.5 saniyelik bekleme süresini atlatacak ya da şu anki işlemin bitmesini bekleyecek, yeni işlemin ilk sinyalleri geldiği an yazdırma safhasına atlayacaktı.

DONANIM ve YAZILIM SÜRECİ

***Data Sheets:** Programın çalışması için gerekli sistem aletlerinin bilgilerini içeren 'Data Sheet'ler rapor kapsamında yer almaktadır.

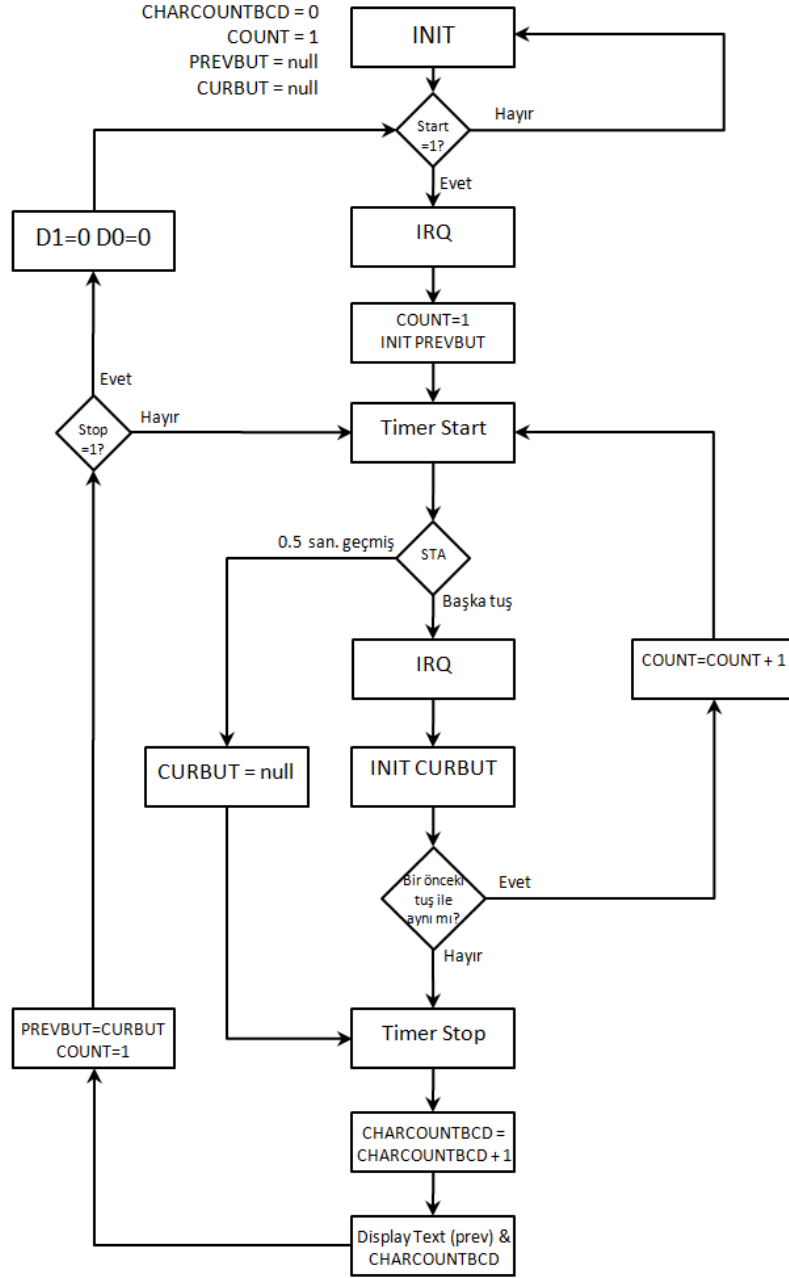
***Yazılım için gerekli Table'lar :** Centronics Printer için yazılması gereken karakterlerin Seven-segment kodlarını, tuşlara yönelik oluşturduğumuz Table'lara atadık. Bu Table'lar da şu şekilde düzenlendi:

TABLE1 DB 06H, 06H, 06H, 06H	; 1
TABLE2 DB 77H, 7CH, 39H, 5BH	; a, b, c, 2
TABLE3 DB 5EH, 79H, 71H, 4FH	; d, e, f, 3
TABLE4 DB 3DH, 76H, 06H, 66H	; g, h, i, 4
TABLE5 DB 1EH, 76H, 38H, 6DH	; j, k, l, 5
TABLE6 DB FFH, 31H, 5CH, 7DH	; n, o, 6
TABLE7 DB 73H, 50H, 6DH, 07H	; p, r, s, 7
TABLE8 DB 78H, 3EH, 3EH, 7FH	; t, u, v, 8
TABLE9 DB FFH, 6EH, 5BH, 6FH	; y, z, 9
TABLE0 DB 00H, 48H, 40H, 3FH	; _ , = , - , 0

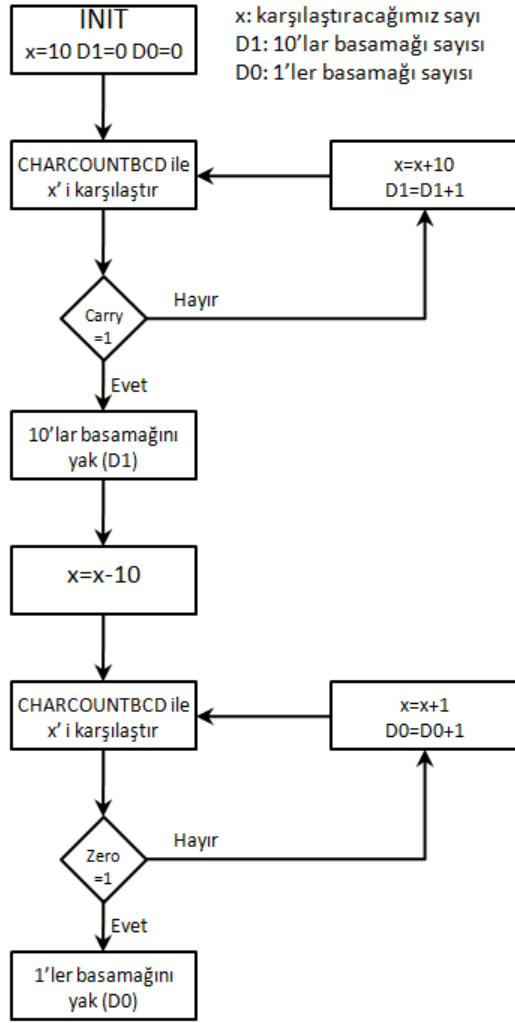
Seven-segment display'lerde de yazılması gereken sayıların Seven-segment kodlarını ayrı bir table da birleştirdik:

NUMTABLE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH ; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

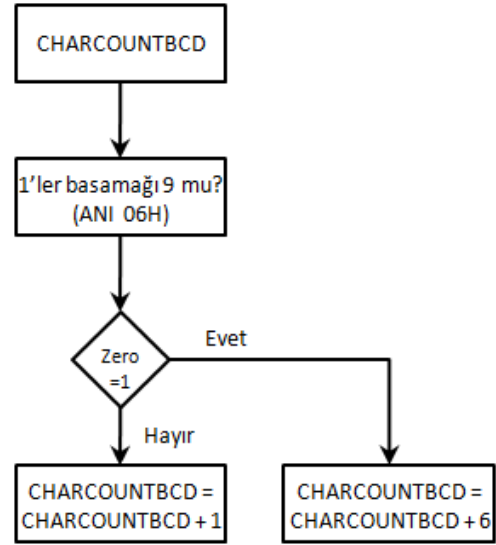
***Flowchart Çizimi :** Ana program için flowchart çizimi aşağıda gösterilmiştir:



Alt programlar için oluşan flowchartlar ise şu şekildedir:



'Display' Altprogramı



'Increment' Altprogramı

***Yazılım :**

CMD1 EQU 80
STA1 EQU 80

```

PA1    EQU 81
PB1    EQU 82
PC1    EQU 83
LSB1   EQU 84
MSB1   EQU 85
CMD2   EQU C0
STA2   EQU C0
PA2    EQU C1
PB2    EQU C2
PC2    EQU C3
COUNT EQU 8000H
PREVBUT EQU 8001H
CURBUT EQU 8002H
TEMP   EQU 8002H
CHARCOUNTBCD EQU C000H
D1     EQU C001H
D0     EQU C002H

```

```

TABLE1 DB 06H, 06H, 06H, 06H      ; 1
TABLE2 DB 77H, 7CH, 39H, 5BH      ; a, b, c, 2
TABLE3 DB 5EH, 79H, 71H, 4FH      ; d, e, f, 3
TABLE4 DB 3DH, 76H, 06H, 66H      ; g, h, i, 4
TABLE5 DB 1EH, 76H, 38H, 6DH      ; j, k, l, 5
TABLE6 DB FFH, 31H, 5CH, 7DH      ; n, o, 6
TABLE7 DB 73H, 50H, 6DH, 07H      ; p, r, s, 7
TABLE8 DB 78H, 3EH, 3EH, 7FH      ; t, u, v, 8
TABLE9 DB FFH, 6EH, 5BH, 6FH      ; y, z, 9
TABLE0 DB 00H, 48H, 40H, 3FH      ; _ , =, -, 0

```

```

NUMTABLE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH      ; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

```

```

0000H: JMP MAIN
002CH: JMP IRQSER5.5
IRQSER5.5:

```

```

    IN PA1
    CALL WHICHBUTTONWASPRESSED
    EI
    RET

```

```

MAIN: MVI A, 06H      ; set PA1-input, PB1-output
      OUT CMD1
      MVI A, 03H      ; set PA2-output, PB2-output, PC1-input
      OUT CMD2

```

```

MVI A, 08H           ; make timer wait 0.5 seconds
OUT LSB1
MVI A, 00H           ; make clock singles single square
OUT MSB1
MVI A, 00H           ; initialize CHARCOUNTBCD=0, TEMP=0
STA CHARCOUNTBCD
STA TEMP
MVI A, 01H           ; initialize COUNT=1
STA COUNT
MVI A, AAH           ; initialize PREVBUT=null, CURBUT=null
STA PREVBUT
STA CURBUT
START: IN PC2         ; check start button
ANI 01H
JZ START
LDA COUNT
MVI A, 01H
STA COUNT
MVI A, 0EH           ; unmask RST 5.5
SIM
EI
TMP1: LDA TEMP        ; check TEMP to know if there is IRQ or not
ANI 55H
JZ TMP1
LDA TEMP
STA PREVBUT          ; set PREVBUT
MVI A, AAH           ; set TEMP=null
STA TEMP
TMR1: MVI A, B6H      ; timer start
OUT CMD1
STA:  IN STA1         ; check the waiting time or any other IRQ
ANI 40H
JNZ NOT
LDA TEMP
ANI 55H
JZ STA
MVI A, 0EH
SIM
EI

TMP2: LDA TEMP
ANI 55H
JZ TMP2
LDA TEMP

```

```

    STA CURBUT          ; set CURBUT
    MVI A,AAH           ; set TEMP=null
    STA TEMP
    LDA CURBUT
    MOV M, A
    LDA PREVBUT
    CPI M               ; check if the same button was pressed or not
    JNZ TMR0
    LDA COUNT
    INR A               ; same button so increment COUNT
    STA COUNT
    JMP TMR1
NOT:  LDA CURBUT
    MVI A, AAH          ; waiting time is over so CURBUT=null
    STA CURBUT
TMR0: MVI A, 46H        ; timer stop
    OUT CMD1
    CALL INCCHARCOUNTBCD ; increment CHARCOUNTBCD
    CALL DISPTXT        ; show character on printer
    CALL DISPLAYCHARCOUNTBCD ; show how many characters displayed on 2x7-segments
    LDA CURBUT
    STA PREVBUT         ; CURBUT is now PREVBUT
    LDA COUNT
    MVI A, 01H
    STA COUNT
STOP: IN PC2            ; check stop button
    ANI 02H
    JNZ TMR1
    MVI A, 00H
    STA D1
    STA D0
    JMP START
    HLT

```

```

WHICHBUTTONWASPRESSED:
    ANI EEH             ;1110 1110
    JZ 1WAS
                                ANI DEH             ;1101 1110
                                JZ 2WAS
                                ANI BEH             ;1011 1110
                                JZ 3WAS

```


ANI EDH	;1110 1101	JMP RET3
JZ 4WAS		3WAS: MVI A, 03H
ANI DDH	;1101 1101	JMP RET3
JZ 5WAS		4WAS: MVI A, 04H
ANI BDH	;1011 1101	JMP RET3
JZ 6WAS		5WAS: MVI A, 05H
ANI EBH	;1110 1011	JMP RET3
JZ 7WAS		6WAS: MVI A, 06H
ANI DBH	;1101 1011	JMP RET3
JZ 8WAS		7WAS: MVI A, 07H
ANI BBH	;1011 1011	JMP RET3
JZ 9WAS		8WAS: MVI A, 08H
ANI D7H	;1101 0111	JMP RET3
JZ 0WAS		9WAS: MVI A, 09H
		JMP RET3
		0WAS: MVI A, 00H
		RET3: STA TEMP
		RET
1WAS: MVI A, 01H		
JMP RET3		
2WAS: MVI A, 02H		
INCCHARCOUNTBCD:		
LDA CHARCOUNTBCD		
ANI 06H		
JNZ INCR1		
LDA CHARCOUNTBCD		
ADI 06H		
STA CHARCOUNTBCD		
JMP RET2		
INCR1: LDA CHARCOUNTBCD		
INR A		
STA CHARCOUNTBCD		
RET2: RET		
DISPLAYCHARCOUNTBCD:		
MVI A,00H		
STA D1		
STA D0		
LXI H, 10H		
CMP1: LDA CHARCOUNTBCD		
SUB M		
JC DISP1		

```

LDA D1
INR A
STA D1
MOV A, M
ADD 10H
MOV M,A
MOV C,A
JMP CMP1
DISP1: CALL GETNUM1ANDDISP
MOV A, C
SUI 10H
MOV M, A
CMP2: LDA CHARCOUNTBCD
SUB M
JZ DISP0
LDA D0
INR A
STA D0
MOV A, M
ADD 01H
MOV M,A
JMP CMP2
DISP0: CALL GETNUM0ANDDISP
RET
GETNUM1ANDDISP:
LXI H, NUMTABLE
AGN1: LDA D1
JZ OUTD1

DCR A
STA D1
INX H
JMP AGN1
OUTD1:MOV A,M
OUT PA2
RET

GETNUM0ANDDISP:
LXI H, NUMTABLE
AGN2: LDA D0
JZ OUTD0
DCR A
STA D0
INX H
JMP AGN2
OUTD0:MOV A,M
OUT PB2
RET

DISPTXT:
LDA PREVBUT
CPI 0
JNZ WITH1
LXI H, TABLE0
CALL GETCHAR
JMP RET4
WITH1: CPI 1

```

```

        JNZ WITH2
        LXI H, TABLE1
        CALL GETCHAR
        JMP RET4
WITH2: CPI 2
        JNZ WITH3
        LXI H, TABLE2
        CALL GETCHAR
        JMP RET4
WITH3: CPI 3
        JNZ WITH4
        LXI H, TABLE3
        CALL GETCHAR
        JMP RET4
WITH4: CPI 4
        JNZ WITH5
        LXI H, TABLE4
        CALL GETCHAR
        JMP RET4
WITH5: CPI 5
        JNZ WITH6
        LXI H, TABLE5
        CALL GETCHAR
        JMP RET4
WITH6: CPI 6
        JNZ WITH7
        LXI H, TABLE6
        CALL GETCHAR
        JMP RET4
WITH7: CPI 7
        JNZ WITH8
        LXI H, TABLE7
        CALL GETCHAR
        JMP RET4
WITH8: CPI 8
        JNZ WITH9
        LXI H, TABLE8
        CALL GETCHAR

        JMP RET4
WITH9: CPI 9
        JNZ RET4
        LXI H, TABLE1
        CALL GETCHAR
RET4:   RET

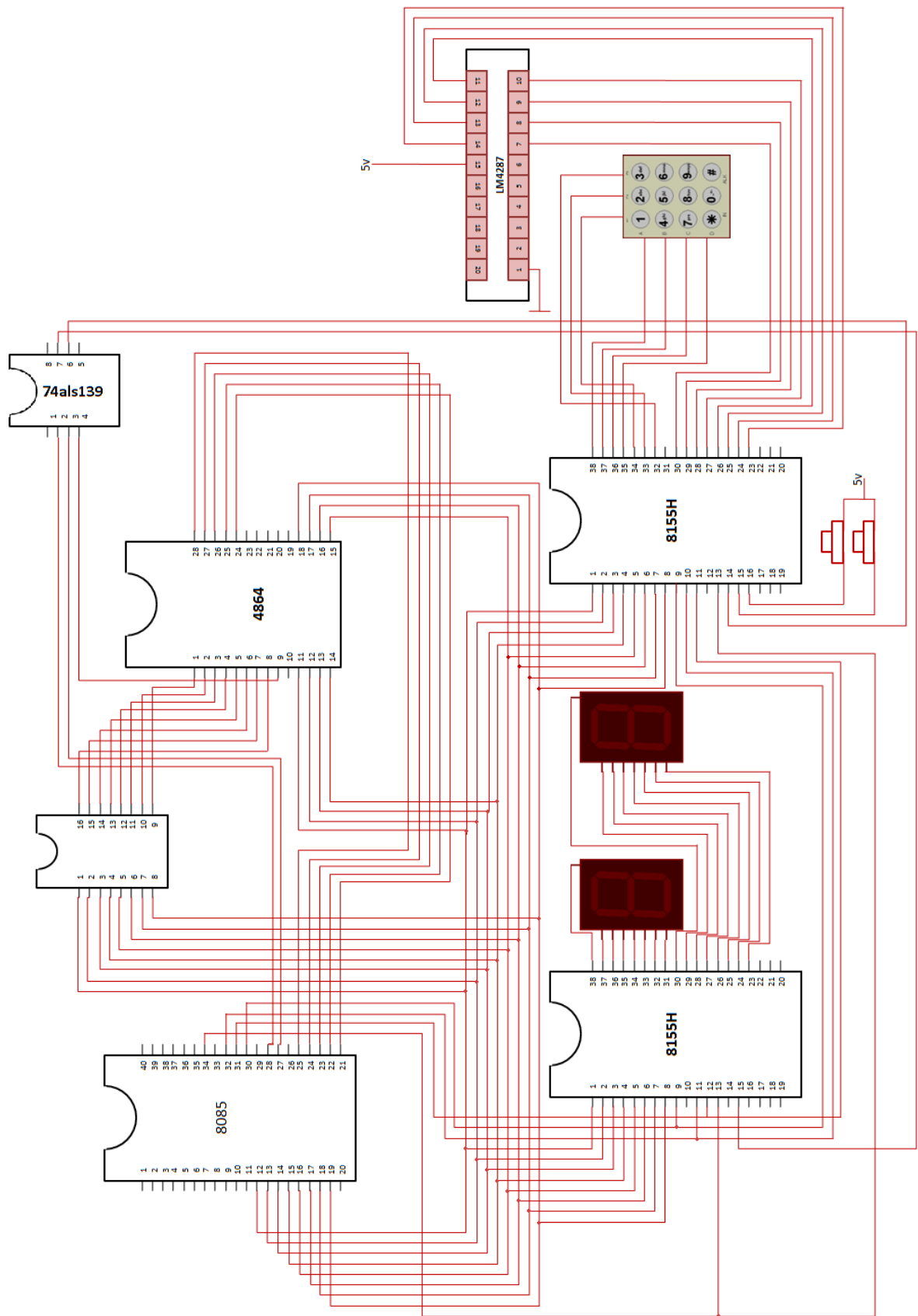
GETCHAR:
CTRL1: LDA COUNT
        CPI 04H
        JC MODE
        DCR A
        STA COUNT
        JZ DONE
        INX H
        JMP XY
MODE:  LDA COUNT
        SUB 04H
        STA COUNT
        JMP CTRL1
DONE:  MOV A,M
        OUT PB2
        RET

```

Design daha detaylı olarak cd içerisindeki dosyalardan incelenilebilir.

-12-

***PC Board Design :**



***Hex Codes :**

	<u>HEX CODES</u>		
0000H:	JMP MAIN	C3 0010	
002CH:	JMP IRQSER5.5	C3 8709	
IRQSER5.5:			
	IN PA1	DB 81	
	CALL WHICHBUTTONWASPRESSED	CD 4C11	
	EI	FB	
	RET	C9	
MAIN:	MVI A, 06H	3E 06	
	OUT CMD1	D3 80	
	MVI A, 03H	3E 03	
	OUT CMD2	D3 C0	
	MVI A, 08H	3E 08	
	OUT LSB1	D3 84	
	MVI A, 00H	3E 00	
	OUT MSB1	D3 85	
	MVI A, 00H	3E 00	
	STA CHARCOUNTBCD	32 00CC	
	STA TEMP	32 0280	
	MVI A, 01H	3E 01	
	STA COUNT	32 0080	
	MVI A, AAH	3E AA	
	STA PREVBUT	32 0180	
	STA CURBUT	32 0280	
START:	IN PC2	DB C3	
	ANI 01H	E6 01	
	JZ START	CA 2910	
	LDA COUNT	3A 0080	
	MVI A, 01H	3E 01	
	STA COUNT	32 0080	
	MVI A, 0EH	3E 0E	
	SIM	30	
	EI	FB	
TMP1:	LDA TEMP	3A 0280	
	ANI 55H	E6 55	
	JZ TMP1	CA 4310	
	LDA TEMP	3A 0280	
	STA PREVBUT	32 0180	
	MVI A,AAH	3E AA	
	STA TEMP	32 0280	
TMR1:	MVI A, B6H	3E B6	
	OUT CMD1	D3 80	
STA:	IN STA1	DB 80	
	ANI 40H	E6 40	
	JNZ NOT	C2 0111	
	LDA TEMP	3A 0280	
	ANI 55H	E6 55	
	JZ STA	CA 5910	
	MVI A, 0EH	3E 0E	
	SIM	30	
	EI	FB	
TMP2:	LDA TEMP	3A 0280	
	ANI 55H	E6 55	
	JZ TMP2	CA 7C10	
	LDA TEMP	3A 0280	
	STA CURBUT	32 0280	
	MVI A,AAH	3E AA	
	STA TEMP	32 0280	
	LDA CURBUT	3A 0280	
	MOV M, A	77	
	LDA PREVBUT	3A 0180	
	CPI M	FE 00	
	JNZ TMR0	C2 1411	
	LDA COUNT	3A 0080	
	INR A	3C	
	STA COUNT	32 0080	
	JMP TMR1	C3 5410	
NOT:	LDA CURBUT	3A 0280	
	MVI A, AAH	3E AA	
	STA CURBUT	32 0280	
TMR0:	MVI A, 46H	3E 46	
	OUT CMD1	D3 80	
	CALL INCCHARCOUNTBCD	CD 2212	
	CALL DISPTXT	CD BE12	
	CALL DISPLAYCHARCOUNTBCD	CD 3312	
	LDA CURBUT	3A 0280	
	STA PREVBUT	32 0180	
	LDA COUNT	3A 0080	
	MVI A, 01H	3E 01	
	STA COUNT	32 0080	
STOP:	IN PC2	DB C3	
	ANI 02H	E6 02	
	JNZ TMR1	C2 5410	
	MVI A, 00H	3E 00	
	STA D1	32 01C0	
	STA D0	32 02C0	
	JMP START	C3 2910	

WHICHBUTTONWASPRESSED:

	ANI EEH	E6 EE
	JZ 1WAS	CA 0012
	ANI DEH	E6 DE
	JZ 2WAS	CA 0312
	ANI BEH	E6 BE
	JZ 3WAS	CA 0612
	ANI EDH	E6 ED
	JZ 4WAS	CA 0912
	ANI DDH	E6 DD
	JZ 5WAS	CA 0C12
	ANI BDH	E6 BD
	JZ 6WAS	CA 1012
	ANI EBH	E6 EB
	JZ 7WAS	CA 1312
	ANI DBH	E6 DB
	JZ 8WAS	CA 2012
	ANI BBH	E6 BB
	JZ 9WAS	CA 2312
	ANI D7H	E6 D7
	JZ 0WAS	CA 2612
1WAS:	MVI A, 01H	3E 01
	JMP RET3	C3 1E12
2WAS:	MVI A, 02H	3E 02
	JMP RET3	C3 1E12
3WAS:	MVI A, 03H	3E 03
	JMP RET3	C3 1E12
4WAS:	MVI A, 04H	3E 04
	JMP RET3	C3 1E12
5WAS:	MVI A, 05H	3E 05
	JMP RET3	C3 1E12
6WAS:	MVI A, 06H	3E 06
	JMP RET3	C3 1E12
7WAS:	MVI A, 07H	3E 07
	JMP RET3	C3 1E12
8WAS:	MVI A, 08H	3E 08
	JMP RET3	C3 1E12
9WAS:	MVI A, 09H	3E 09
	JMP RET3	C3 1E12
0WAS:	MVI A, 00H	3E 00
RET3:	STA TEMP	32 0280
	RET	C9

INCCHARCOUNTBCD:

	LDA CHARCOUNTBCD	3A 00C0
	ANI 06H	E6 06
	JNZ INCR1	C22D12
	LDA CHARCOUNTBCD	3A 00C0
	ADI 06H	C6 06
	STA CHARCOUNTBCD	32 00C0
	JMP RET2	C3 3212
INCR1:	LDA CHARCOUNTBCD	3A 00C0
	INR A	3C
	STA CHARCOUNTBCD	32 00C0
RET2:	RET	C9

DISPLAYCHARCOUNTBCD:

	MVI A,00H	3E 00
	STA D1	32 01C0
	STA D0	32 02C0
	LXI H, 10H	21 10
CMP1:	LDA CHARCOUNTBCD	3A 00C0
	SUB M	96
	JC DISP1	DA 6112
	LDA D1	3A 01C0
	INR A	3C
	STA D1	32 01C0
	MOV A, M	7E
	ADD 10H	87 10
	MOV M,A	77
	MOV C,A	4F
	JMP CMP1	C3 4212
DISP1:	CALL GETNUM1ANDDISP	CD 8012
	MOV A, C	79
	SUI 10H	D6 10
	MOV M, A	77
CMP2:	LDA CHARCOUNTBCD	3A 00C0
	SUB M	96
	JZ DISPO	CA 7A12
	LDA D0	3A 02C0
	INR A	3C
	STA D0	32 02C0
	MOV A, M	7E
	ADD 01H	87 01
	MOV M,A	77
	JMP CMP2	C3 6E12
DISPO:	CALL GETNUM0ANDDISP	CD A412
	RET	C9

GETNUM1ANDDISP:

	LXI H, NUMTABLE	21 0001
AGN1:	LDA D1	3A 01C0
	JZ OUTD1	CA 9812
	DCR A	3D
	STA D1	32 01C0
	INX H	23
	JMP AGN1	C3 1288
OUTD1:	MOV A,M	7E
	OUT PA2	D3 C1
	RET	C9

GETNUM0ANDDISP:

	LXI H, NUMTABLE	21 4001
AGN2:	LDA D0	3A 02C0
	JZ OUTD0	CA 8812
	DCR A	3D
	STA D0	32 02C0
	INX H	23
	JMP AGN2	C3 AB12
OUTD0:	MOV A,M	7E
	OUT PB2	D3 C2
	RET	C9

DISPTXT:

LDA PREVBUT	3A 0180
CPI 0	FE 0
JNZ WITH1	C2 C612
LXI H, TABLE0	21 0001
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH1: CPI 1	FE 1
JNZ WITH2	CD12
LXI H, TABLE1	21 0401
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH2: CPI 2	FE 2
JNZ WITH3	C2 D312
LXI H, TABLE2	FE 0801
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH3: CPI 3	FE 3
JNZ WITH4	C2 DA12
LXI H, TABLE3	21 1201
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH4: CPI 4	FE 4
JNZ WITH5	C2 E012
LXI H, TABLE4	21 1601
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH5: CPI 5	FE5
JNZ WITH6	C2 12E8
LXI H, TABLE5	21 2001
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH6: CPI 6	FE 6
JNZ WITH7	C2 EE12
LXI H, TABLE6	21 2401
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH7: CPI 7	FE 7
JNZ WITH8	C2 0613
LXI H, TABLE7	21 2801
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH8: CPI 8	FE 8
JNZ WITH9	C2 0D13
LXI H, TABLE8	21 3201
CALL GETCHAR	CD 1313
JMP RET4	C3 1213
WITH9: CPI 9	FE9
JNZ RET4	C2 1213
LXI H, TABLE1	21 0401
CALL GETCHAR	CD 1313
RET4: RET	C9

GETCHAR:

CTRL1: LDA COUNT	3A 0080
CPI 04H	FE 04
JC MODE	DA 2113
DCR A	3D
STA COUNT	32 0080
JZ DONE	CA 3213
INX H	23
JMP XY	C3
MODE: LDA COUNT	3A 0080
SUB 04H	97 04
STA COUNT	32 0080
JMP CTRL1	C3 1413
DONE: MOV A,M	7E
OUT PB2	D3 C2
RET	C9

***Rapor Hazırlanması:** Rapor hazırlanması için projenin sonu beklenmedi. Çalışmaların başlamasıyla birlikte rapor da bir yandan güncellenmekte idi. Karşılaşılan problemler, dikkat çeken ayrıntılar, programın aşamaları raporda anlatıldı.

SONUÇ

Derste öğrendiklerimizi en iyi şekilde kullandık, ufak tefek hatalar olsa da onları en aza düşürmek için gerçekten çok çalıştık ve projeyi sistematik bir şekilde, başarılı sonuçlanan bir takım çalışmasıyla tamamladık.

NOT: Raporda yer alan tüm belgeler ve daha ayrıntılı dökümanlar hazırlanan cd'nin içerisinde yer almaktadır.