
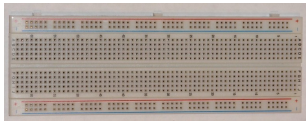





PROJECT 12

7 SEGMENT DISPLAY WITH CD4511 ENCODER

Besides on directly connected 7 segment with direct I/O port from microcontroller, we can also use the CD4511 (7 segment encoder) to help on decreasing the microcontroller I/O port and only required to send the binary data.

COMPONENT NEEDED

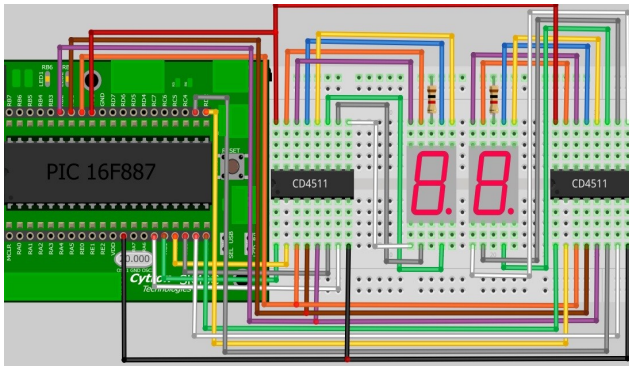
	7 SEGMENT DISPLAY x2 (Common Cathode)
	BREADBOARD
	1K RESISTOR x2
	CD4511 7 SEGMENT ENCODER
	JUMPER WIRES

CONNECTION

Referring to the figure below, **RED** and **BLACK** wire are VDD and GND and are share to both CD4511 encoder. Besides that, the LT, BI, LE pin from CD4511 are share to the same port of PIC MCU which is PORTB.0 , PORTB.1 and PORTB.2. Lastly, the LEFT CD4511 binary input (A,B,C,D) are connected to PORTC.0 until PORTC.3. While RIGHT CD4511 binary input (A,B,C,D) are connected to PORTD.0 until PORTD.3.

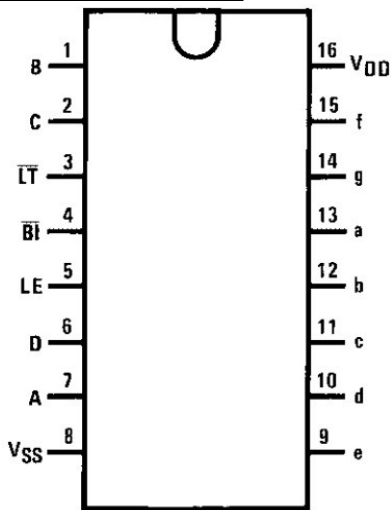
FROM CD4511 TO 7 SEGMENT & PIC16F887

CD4511		7 SEGMENT	PIC16F887	
PINS			PORTC	PORTD
PIN1	B	-	PORTC.1	PORTD.1
PIN2	C	-	PORTC.2	PORTD.2
PIN3	LT	-	PORTB.0	PORTB.0
PIN4	BI	-	PORTB.1	PORTB.1
PIN5	LE	-	PORTB.2	PORTB.2
PIN6	D	-	PORTC.3	PORTD.3
PIN7	A	-	PORTC.0	PORTD.0
PIN8	VSS	-	GND	
PIN9	e	e	-	-
PIN10	d	d	-	-
PIN11	c	c	-	-
PIN12	b	b	-	-
PIN13	a	a	-	-
PIN14	g	g	-	-
PIN15	f	f	-	-
PIN16	VDD	-	VDD(+5V)	



ADDITIONAL INFORMATION

CD4511 7-SEGMENT ENCODER



Top View

Figure above is the CD4511 7 segment encoder which use to encode the binary input given and convert it into 7 segment number. This Integrated circuit input are required the BCD code and it will display the number from 0 – 9 in 7 segment.

TRUTH TABLE

Inputs							Outputs							
LE	\overline{BI}	\overline{LT}	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	B
X	0	1	X	X	X	X	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	0	1	0	0	0	0	0	0	0	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X				*				*

X = Don't Care

*Depends upon the BCD code applied during the 0 to 1 transition of LE.

Above are the truth table for CD4511 encoder to convert the input BCD to output of 7-segment number.

CODE OVERVIEW

system.h

```
#define dis_1      0b00000001
#define dis_2      0b00000010
#define dis_3      0b00000011
#define dis_4      0b00000100
#define dis_5      0b00000101
#define dis_6      0b00000110
#define dis_7      0b00000111
#define dis_8      0b00001000
#define dis_9      0b00001001
#define dis_0      0b00000000

#define LE         RB2
#define BI         RB1
#define LT         RB0

#define SEG1_dis   PORTC
#define SEG2_dis   PORTD
```

```
#define dis_1 0b00000001
```

Declare the BCD number that will send to the CD4511 encoder. E.g: 0b00000001 are number "1" in 7 segment.

```
#define LE      RB2
```

```
#define BI      RB1
```

```
#define LT      RB0
```

In here, we have initially declare the I/O port that

use to control the CD4511 such as lamp test(LT), blanking (BI), latch enable(LE) and store a BCD code.

MAIN PROGRAM

```
LT = 0;
delay_ms(1000);

unsigned char num = 0;
unsigned int dis1[] = {dis_0,dis_1,dis_2,dis_3,dis_4,dis_5,dis_6,dis_7,dis_8,dis_9};
unsigned int dis2[] = {dis_9,dis_8,dis_7,dis_6,dis_5,dis_4,dis_3,dis_2,dis_1,dis_0};

while (1)
{
    LE = 0;
    BI = 1;
    LT = 1;
    for (num=0 ; num<10 ; num++)
    {
        SEG1_dis = dis1[num];
        SEG2_dis = dis2[num];
        delay_ms(1000);
    }
}
```

LT = 0;

Referring to the truth table, HIGH(1) the LT will stop the lamp test while LOW(0) the LT will on the lamp test.

LE = 0;

HIGH(1) to ON the latch enable and may store a BCD code. LOW(0) will OFF latch enable and will not store a BCD code.

BI = 1;

HIGH(1) will OFF the blanking and LOW(0) will ON the blanking.