Experiment-7: Keypad 4x4 Matrix

- Po Pin configuration in the kit for the keypad
- Scan code look up table for keypad
- Scanning mechanism
- Hands On
    - interfacing keypad with 7S display

- Keypad (4 Rows X 4 Columns):

- The switches SW4 to SW19 are organized as 4 rows X 4 columns matrix.

- One end of all the switches are connected to port lines P0.4 - P0.7, which is configured as columns.

- The other end of the matrix is connected to the port lines P0.0 – P0.3 which is configured as rows.

- The interface diagram for keypad is shown below.

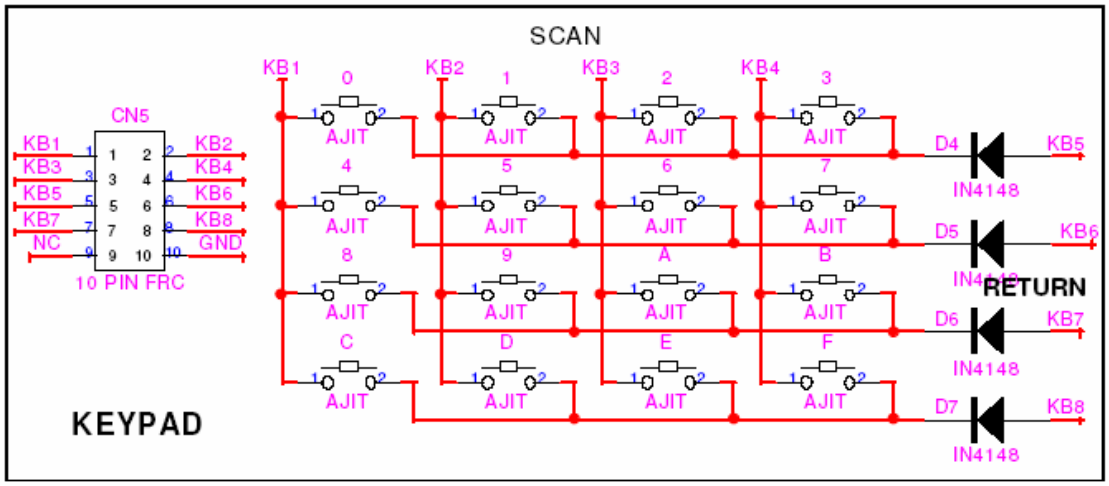
Operation:

- Initially take one column line to logic HIGH, then check for each row.

- If first row is at logic '0' and other rows are at logic '1' then save the read data and compare with the look-up table.

- Similarly repeat the procedure for all the columns.

Keypad Interface:



KEY MATRIX

Key pad Scan Code Table
(Scan column by keeping row low or vice-versa)

Digit	Scan code	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
0	0xEE	On	On	On	Off	On	On	On	Off
1	0xDE	On	On	Off	On	On	On	on	Off
2	0xBE	On	Off	On	On	On	on	On	Off
3	0x7E	Off	on	on	on	On	On	On	Off
4	0xED	On	On	On	off	On	On	Off	On
5	0xDD	On	On	Off	On	On	On	Off	On
6	0xBD	On	off	On	On	On	On	Off	On
7	0x7D	Off	On	On	On	On	On	Off	On
8	0xEB	On	On	On	Off	On	Off	On	On
9	0xDB	On	On	Off	On	On	Off	On	On
A	0xBB	On	Off	On	On	On	Off	On	On
B	0x7B	Off	on	On	on	On	Off	On	On
C	0xE7	on	on	on	Off	Off	On	on	on
D	0xD7	On	on	Off	on	Off	On	on	on
E	0xB7	on	Off	on	On	off	on	on	on
F	0x77	Off	on	On	on	off	On	on	on

```
//Sample Code skeleton for scanning the key press from keypad and displaying it on 7
//Segment displays
//Keypad scan code lookup table
unsigned char scan_code[16]={
                                0xEE,0xDE,0xBE,0x7E,
                                0xED,0xDD,0xBD,0x7D,
                                0xEB,0xDB,0xBB,0x7B,
                                0xE7,0xD7,0xB7,0x77
                                    };
//7 Segment code look up table from previous exercise
unsigned char LED_CODE[16]= {
                                0x3f,0x66,0x7f,0x39,
                                0x06,0x6d,0x6f,0x5e,
                                0x5b,0x7d,0x77,0x79,
                                0x4f,0x07,0x7c,0x71
                                };
//main routine
while(1)
{
        get_key();
        display();
        P3 = 0xFF; //No 7 Segment Digit selected
}
```

```
//Get_key()
//scan the keypad to get the scan code of the key pressed
//this function is in an eternal loop will return to main () only after getting a key
flag = 0x00;
while(flag == 0x00)
{
        // This for loop makes the one of the ROW low at one time .Then scan function is called
        for(row=0;row<4;row++)
        {
                if( row == 0)
                    temp3=0xFE;
                else if(row == 1)
                    temp3=0xFD;
                else if(row == 2)
                    temp3=0xFB;
                else if(row == 3)
                    temp3=0xF7;

                // each time temp3 value is put into this
                P0 = temp3;
                // on sensing a key scan() function will make flag = 0xff
                        scan();

                delay_ms(10);
                if(flag == 0xff)
                break;
        }// end of for

                if(flag == 0xff)
                break;
}  // end of while
```

```
//Enable U21
P3 = 0x00;

 // in this for for loop scan code received  which is in res1 variable is compared with
// the lookup table for array scan code[] and when a match is found will return the correspoding
//led code for the key pressed

for(i=0;i<16;i++)
{
	//res1 is the scan code received from keypad
	if(scan_code[i] == res1)
		P1 =  LED_CODE[i];
}


//scan
{
	// Both row lines and column lines  are connected to
	unsigned char t;			//Port 0 only.rows are connected to P0.0-P0.3
	temp4 = P0;				// P0.4-P0.7 are connected to cols
	temp4 = temp4 & 0xF0;	//read port0 ,mask with 0xF0h

	if(temp4 != 0xF0)		// Means a key is sensed
	{
		delay_ms(30);
		delay_ms(30);		// give some delay for debouncing

		temp4 = P0;		// read the port again
		temp4 = temp4 & 0xF0;

		if(temp4 != 0xF0)	// debounce
		{
			flag = 0xff;	// set the flag denoting a key is received
			res1 = temp4;
```

```
                t = temp3 & 0x0F;
                res1 = res1 | t;                // and OR it with column value
            }       // to get the scan code of the key pressed
            else
            {
                flag = 0x00;
            }

        }
}
```