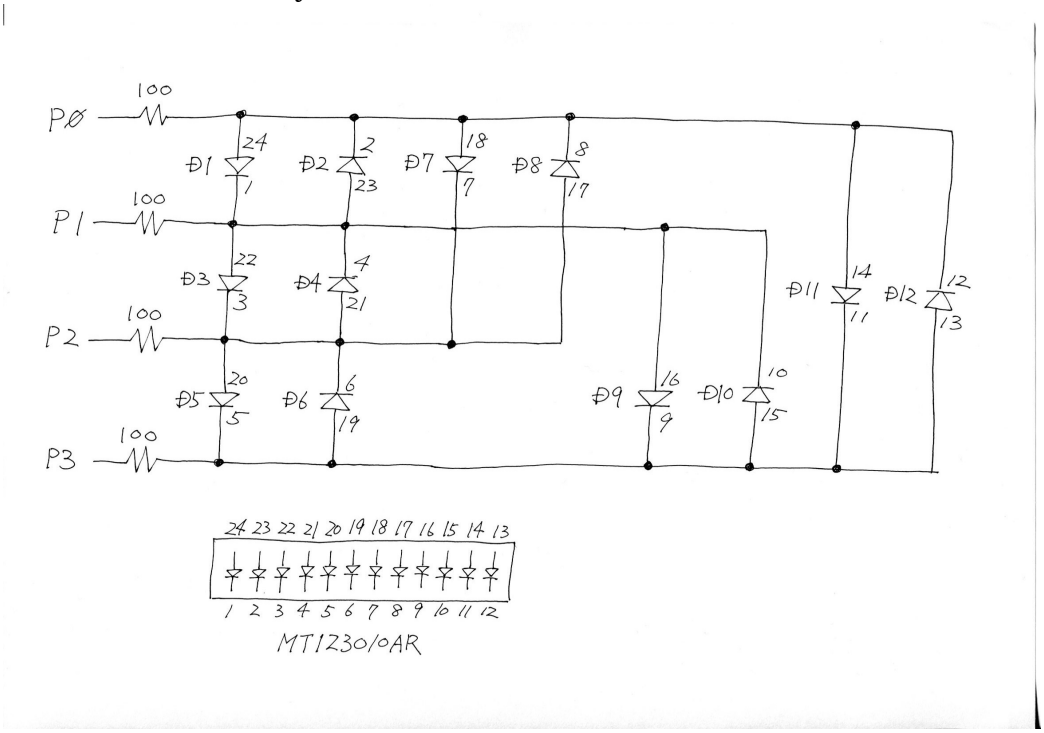
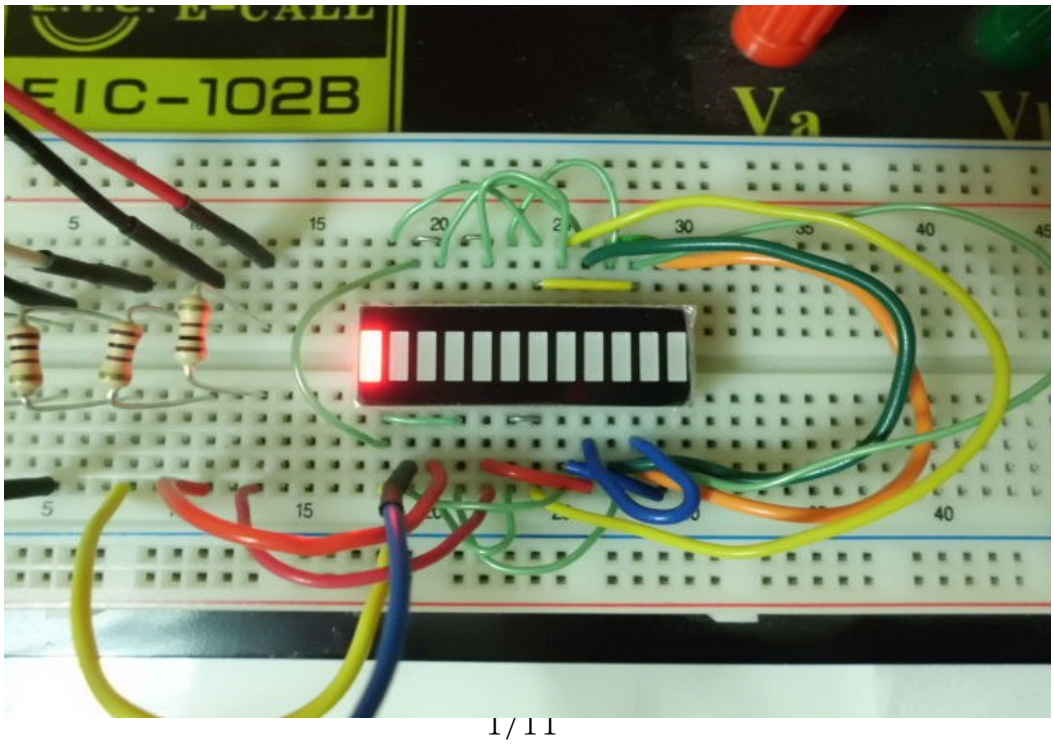
Charlieplexing

~~20141110~~
20151124

LED

Reference; Charlir_LEDs_0.1.f
N*(N-1) LEDs can controlled by N-wires.

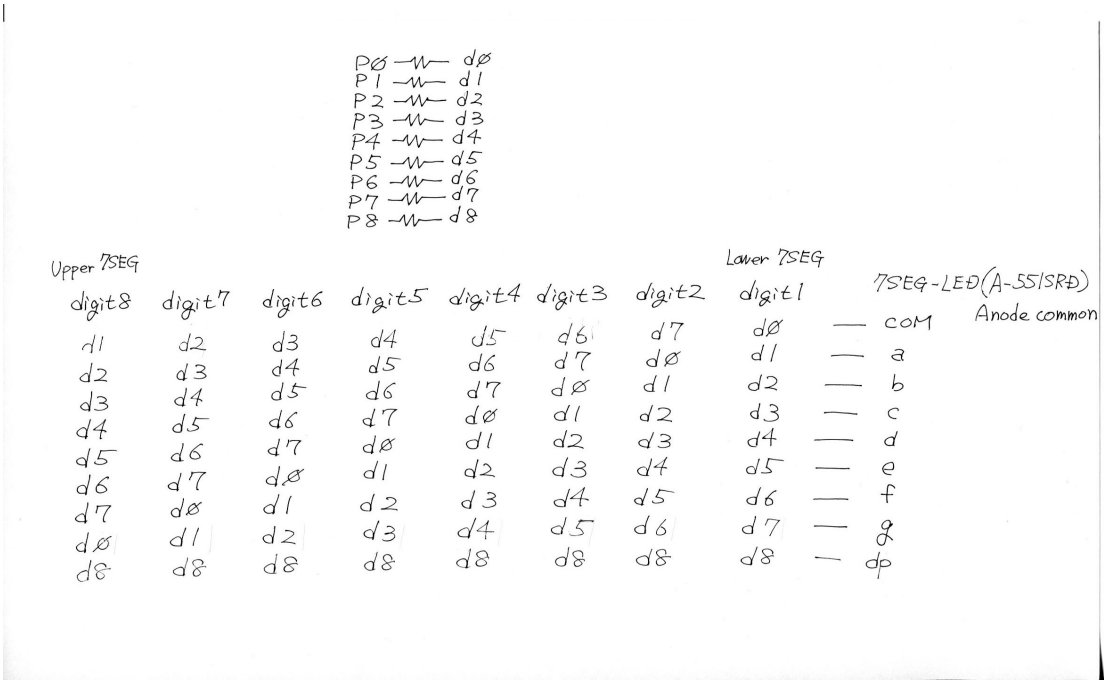


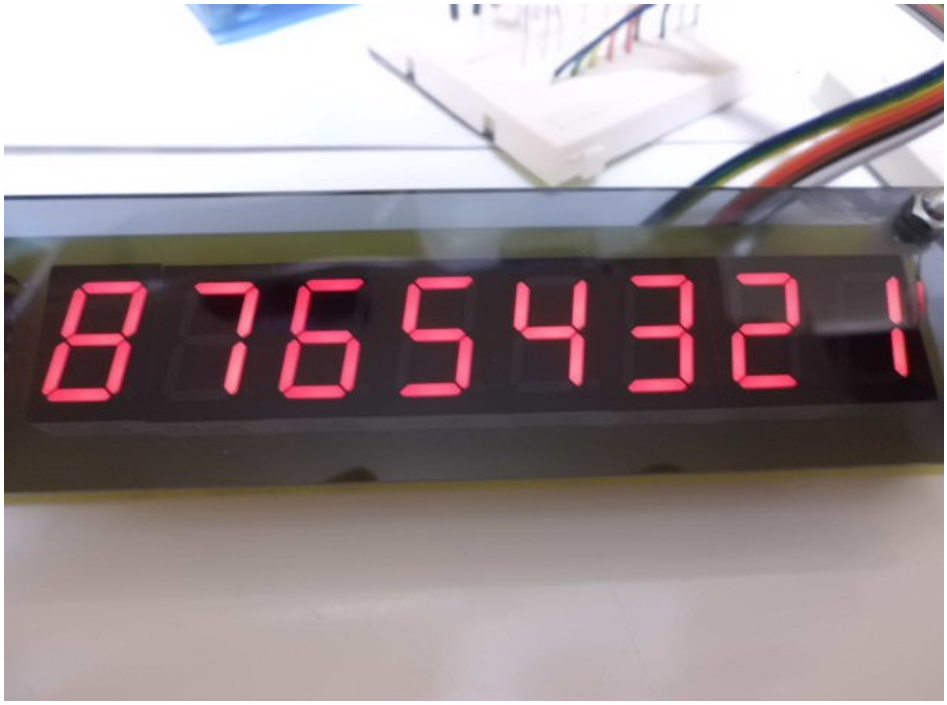


LED	IN/OUT Terminal pin				register	
	P3	P2	P1	P0	dira	outa
D1	Hi-Z	Hi-Z	0	1	3	1
D2	Hi-Z	Hi-Z	1	0	3	2
D3	Hi-Z	0	1	Hi-Z	6	2
D4	Hi-Z	1	0	Hi-Z	6	4
D5	0	1	Hi-Z	Hi-Z	hC	4
D6	1	0	Hi-Z	Hi-Z	hC	8
D7	Hi-Z	0	Hi-Z	1	5	1
D8	Hi-Z	1	Hi-Z	0	5	4
D9	0	Hi-Z	1	Hi-Z	hA	2
D10	1	Hi-Z	0	Hi-Z	hA	8
D11	0	Hi-Z	Hi-Z	1	9	1
D12	1	Hi-Z	Hi-Z	0	9	8

7Segment-LED

Reference; Charlieplexing_7LEDs_0.1_1.f
Wire connctions below;





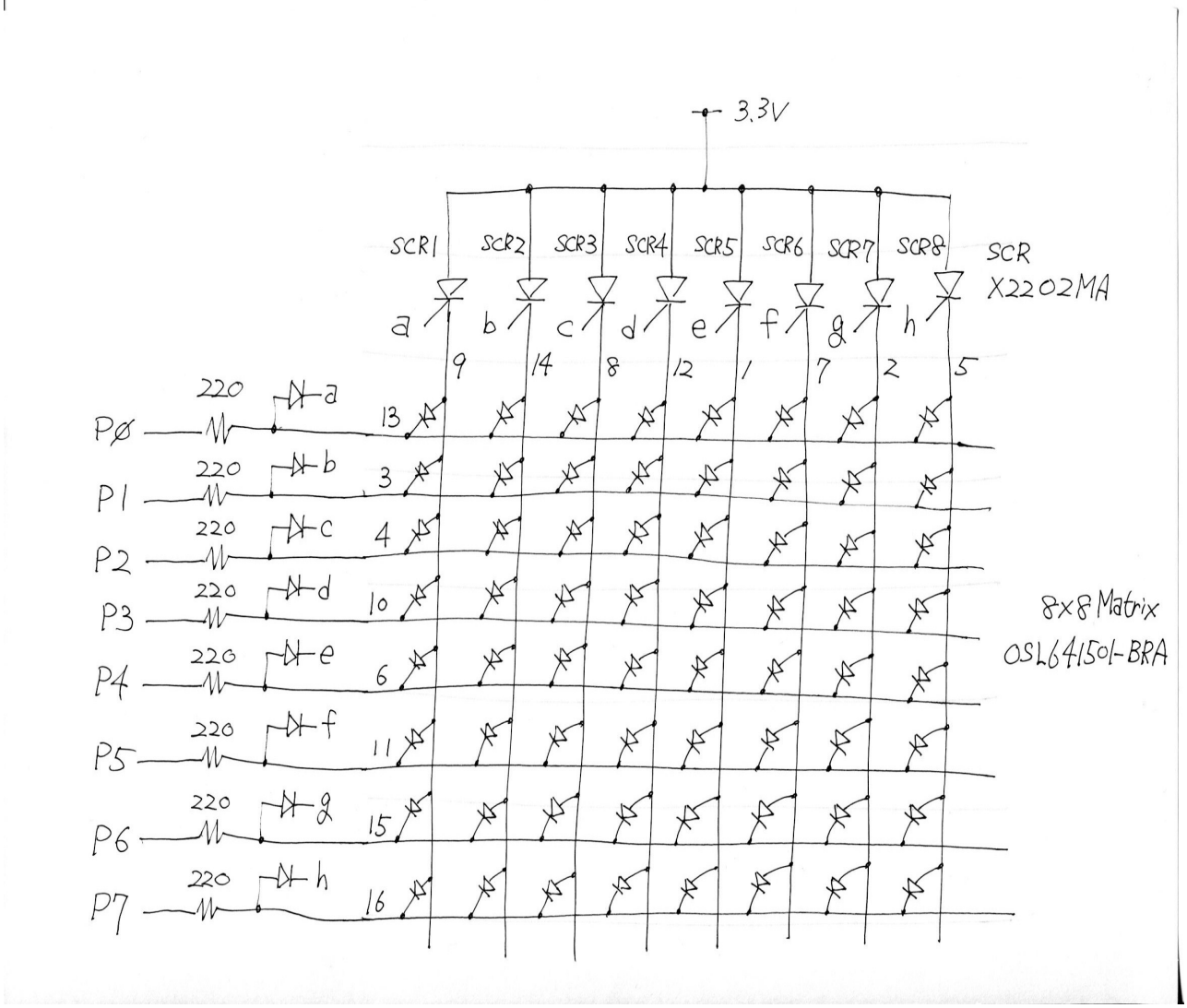
Current through 7Segment-LED's common is constant.
So, its brightness become low when going on many elements for 7Seg-LED, .
7Seg“1” is brighter than 7Seg“8” .
To prevent this, using Tr(emitter follower) on each pin[P0-P8] is recommended.

8x8Matrix-LED

Charlieplexing_8x8Matrix_0.3_2.f

N*N Matrix-LED can controlled by N-wires.

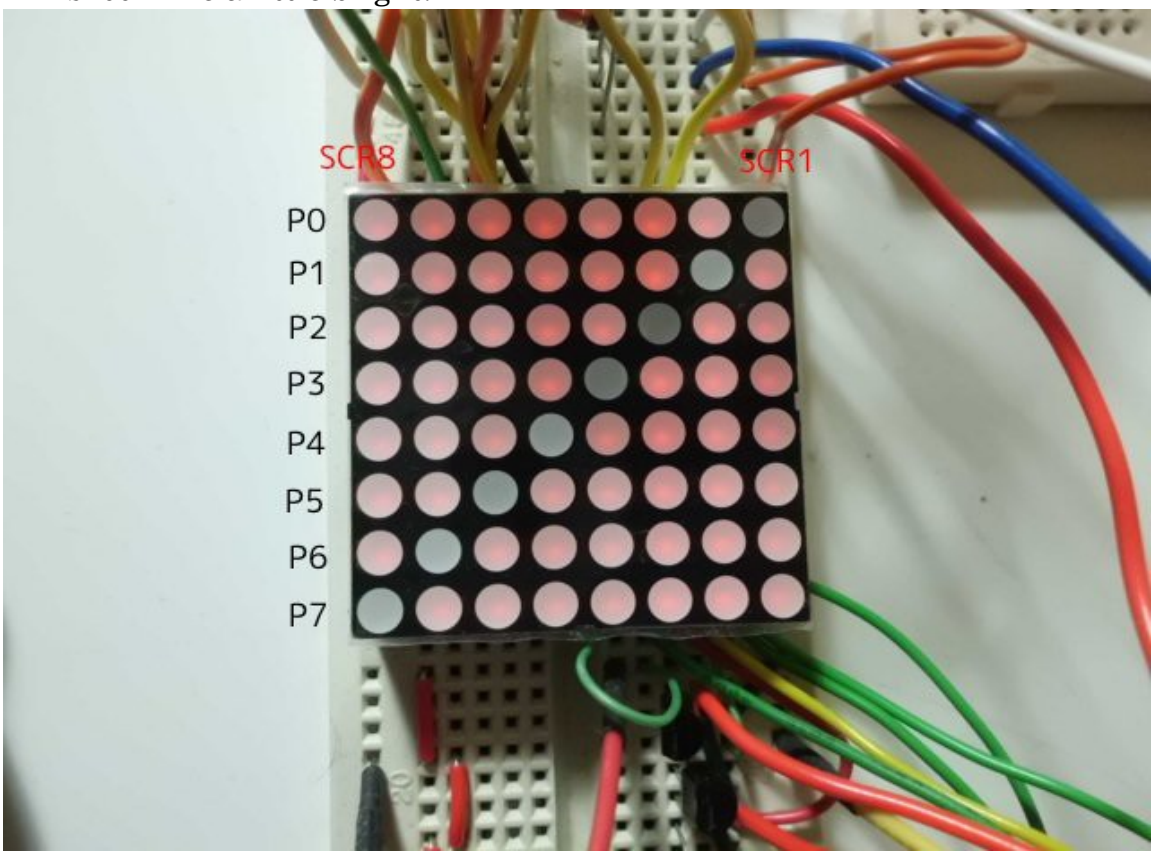
This use SCR.
SCR is like diode with gate terminal.
SCR flow current from anode to cathode when Hi-pulse is added to gate.
And current continue to flow when Hi-pulse lost.
When current from anode to cathode stop, activating SCR need Hi-pulse to gate again.



About SCR1-line for Forth-word[c" matrix_Charlie_fth" 0 cogx]

```
: matrix_Charlie_fth
begin
    1                           \ Set initial data for SCR-on
    8 0 do
        0 dira COG!                 \ Set P0 to P7 to Hi-Z (LED off)
        0 outa COG!                 \ Clear P0-P7
        dup
        \ Activate each SCR
        i lshift outa COG!
        hFF dira COG!
        \ Set data (L-active for outa register)
        matrix i + C@ dup
        invert outa COG!            \ Set inverted data to outa register
        dira COG!                   \ Set data to dira register
        1 delms                     \ Display data
    loop
    drop
0 until
;
```

Matrix is off because buffer"matrix" are fulled to 0.
But LEDs look like a little bright.



1. Set all pin to Hi-Z.

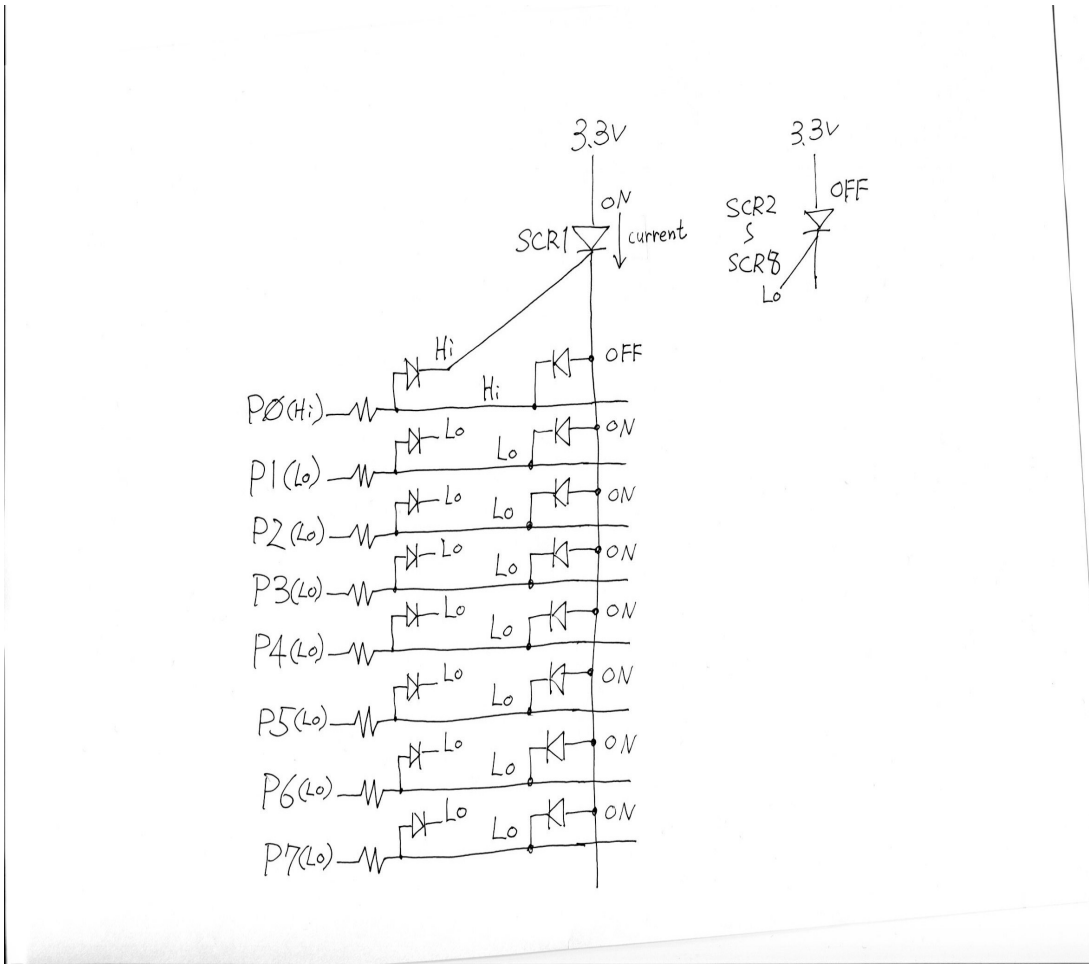
```
0 dira COG!
0 outa COG!
```

All SCRs are deactivated.

2. Activating SCR.

```
i lshift outa COG!
hFF dira COG!
```

Firstly,SCR1 is activated because P0 is Hi.
LED connected to P0 is off.
Other LEDs is on because port[P1-P7] is Lo.
They are dark because SCR-on-time is short.



3. Copy data inside matrix-array to outa register

```
matrix i + C@ dup
invert outa COG!
```

Data inside matrix-array is saved to outa register.
This is inverted because of L-active.
But when there are Hi-bit, iother SCRs is activated by it.
This cause that other LEDs on SCRs are ON.

Actually these are very dark. And time is very short.
We cannot watch other LEDS status.

4. Set data inside matrix-array to dira register

```
dira COG!
```

To prevent activating other SCR, data is saved to dira register.
Other SCRs are NOT activated, because connected line to Hi-bit is Hi-Z.

5. Keep this during display

```
1 delms
```

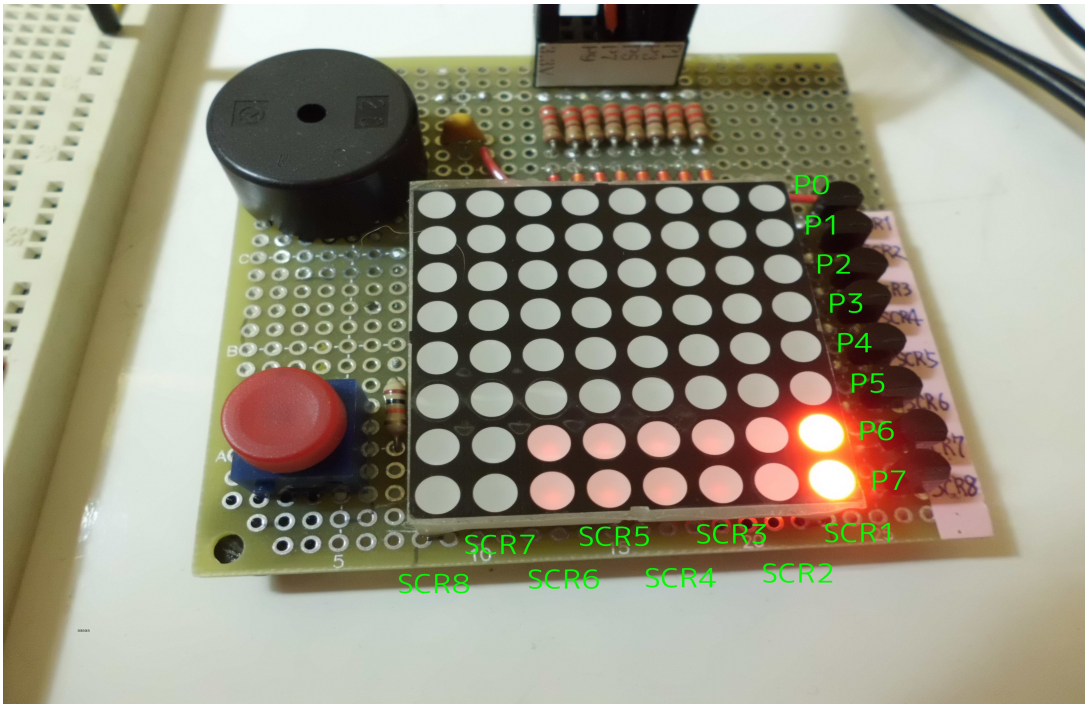
6. Repeat [step1-step5] 8 times

7. Repeat step6

matrix_Charlie_fth in Charlieplexing_8x8Matrix_0.3_1.f

```
: matrix_Charlie_fth
begin
    1                           \ Set initial data for SCR-on
    8 0 do
        0 dira COG!             \ Set P0 to P7 to Hi-Z (LED off)
        0 outa COG!             \ Clear P0-P7
        dup
        \ Activate each SCR
        i lshift outa COG!
        hFF dira COG!
        \ Set data (L-active)
        matrix i + C@ invert outa COG!
        1 delms                 \ Display data
    loop
    drop
0 until
;
```

Other LEDs are on by Old word"matrix_Charlie_fth", when executing 'matrix i + C@ invert outa COG!'.
But we cannot watch because delay-time is 1msecond.
If replaced '1 delms' to d1000 delms', below;



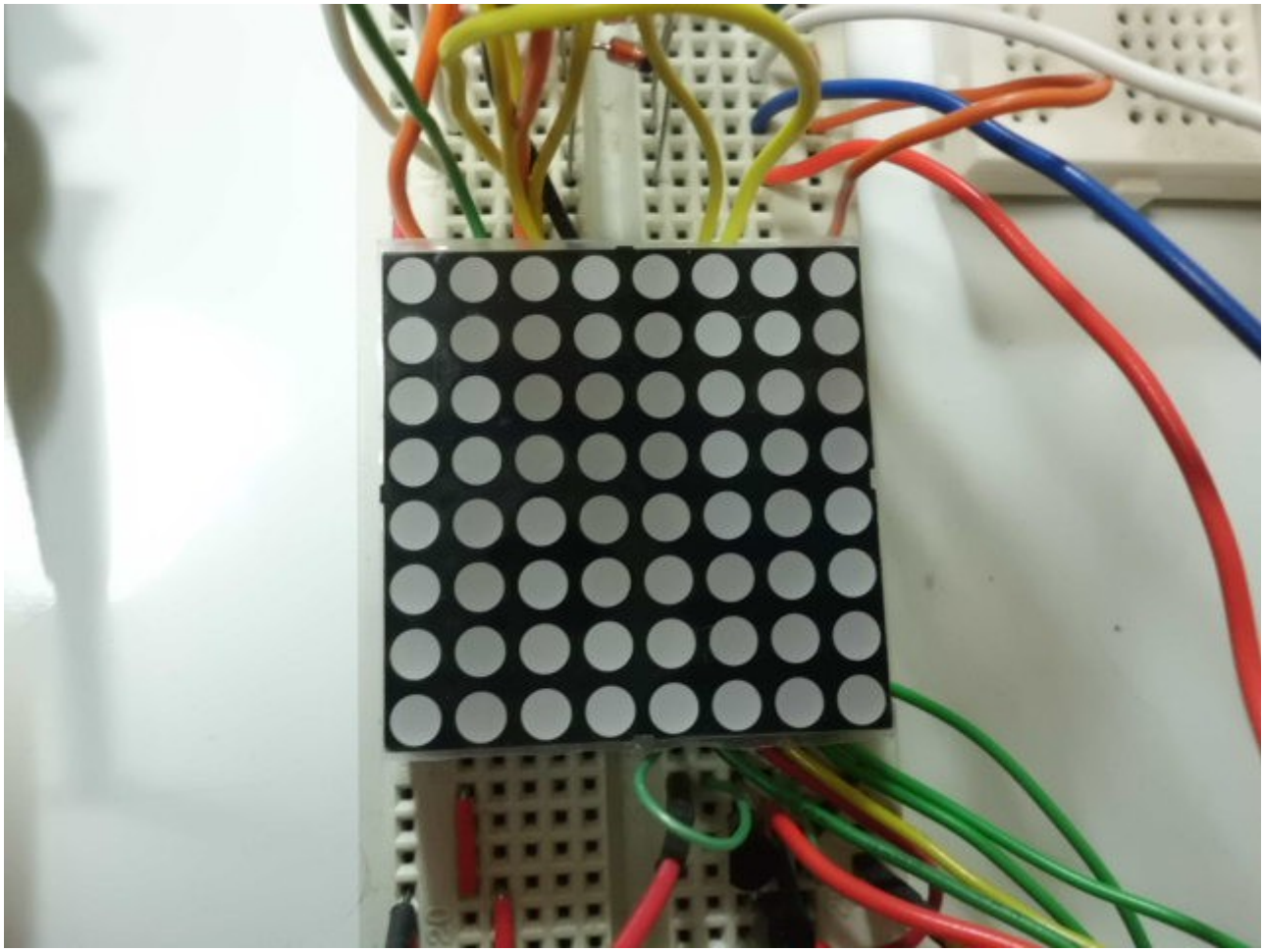
There is hC0 in top-address of matrix array. Other are h00.
After SCR1 is activating and data[hC0] is set, LEDs[P6:P7] for SCR1 are brightly.
But LEDs[P6:P7] for SCR2 - SCR6 line are darkly.

LEDs are dark because difference between 3.3V(anode side for SCR) and VOH(output volt for port) is very small.

About SCR1-line for assembler-word[c" matrix matrix_Charlie_asm" o cogx]

```
build_BootOpt :rasm
__1
    mov       $C_treg1 , # 8
    mov       $C_treg2 , __scr
    mov       $C_treg3 , $C_stTOS

__2
    \ Set from P0 to P7 to Hi-Z (All LEDs off)
    mov       dira , # 0
    \ Set data to 0
    mov       outa , # 0
    \ Wait to drive next SCR
    jmpret  __delayret , # __delay

    \ each SCR on
    mov       outa , $C_treg2
    mov       dira , # hFF
    mov       $C_treg4 , __2.5usec
            add       $C_treg4 , cnt
    waitcnt   $C_treg4 , # 0

    \ Set inverted data to outa register
    rdbyte    $C_treg4 , $C_treg3
    neg       $C_treg5 , $C_treg4
    sub       $C_treg5 , # 1
    mov       outa , $C_treg5
    \ Deactivate other SCRs
    mov       dira , $C_treg4

    \ Delay 1msec
    jmpret  __delayret , # __delay

    shl       $C_treg2 , # 1
    add       $C_treg3 , # 1

    djnz      $C_treg1 , # __2
    jmp       # __1


__delay
        mov       $C_treg4 , __1msec
                    add       $C_treg4 , cnt
        waitcnt   $C_treg4 , # 0
__delayret
                    ret

\ This value must be adjust when changing SCR
__2.5usec
    d200
__1msec
    d80000

__scr
    1
;asm _matrix_Charlie_asm
```

Matrix is off because buffer"matrix" are fulled to 0.
All LEDs look like off.



1. Set all pin to Hi-Z.

```
mov     dira , # 0
mov     outa , # 0
```

jmpret __delayret , # __delay
It needs delay-time to activate next SCR.

2. Activating SCR.

```
mov     outa , $C_treg2
mov     dira , # hFF
mov     $C_treg4 , __2.5usec
add     $C_treg4 , cnt
waitcnt   $C_treg4 , # 0
```

It takes time to activate SCR.
This time might be depended on using SCR-characteristic.

3. Set data inside matrix-array to outa register.

```
rdbyte    $C_treg4 , $C_treg3
neg       $C_treg5 , $C_treg4
sub       $C_treg5 , # 1
mov       outa , $C_treg5
```

Data are inverted and saved to outa-register.
All LEDs are off if data is hFF.
In this case, SCR1-line-LEDs is deactivated.
If data is “0”, SCR1-line-LEDs keep to activate.

4.Deactivate other SCRs

```
mov       dira , $C_treg4
```

To deactivate other SCRs, data inside matrix-array is saved to dira register.
Although Hi-pulse is added to other SCRs during very short time, activating SCR is too short time maybe.

5. Keep this during display.

```
jmpret  __delayret , # __delay
```

6. Repeat [step1-step5] for 8 times.

```
djnz      $C_treg1 , # __2
```

7. Repeat

```
jmp       # __1
```