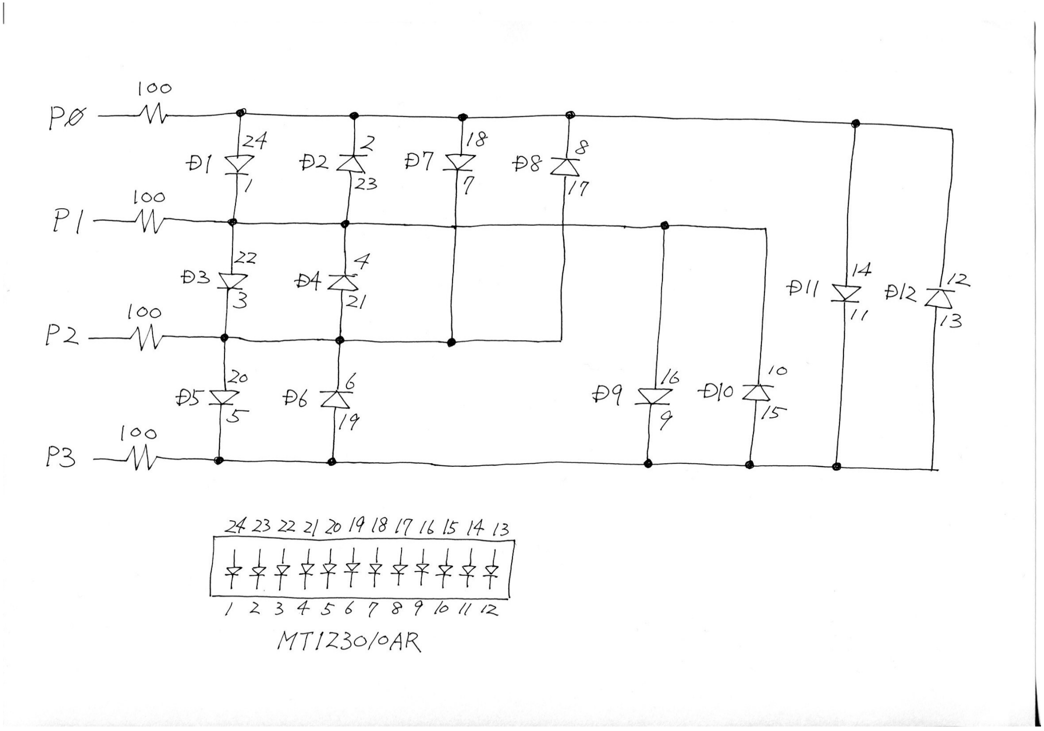
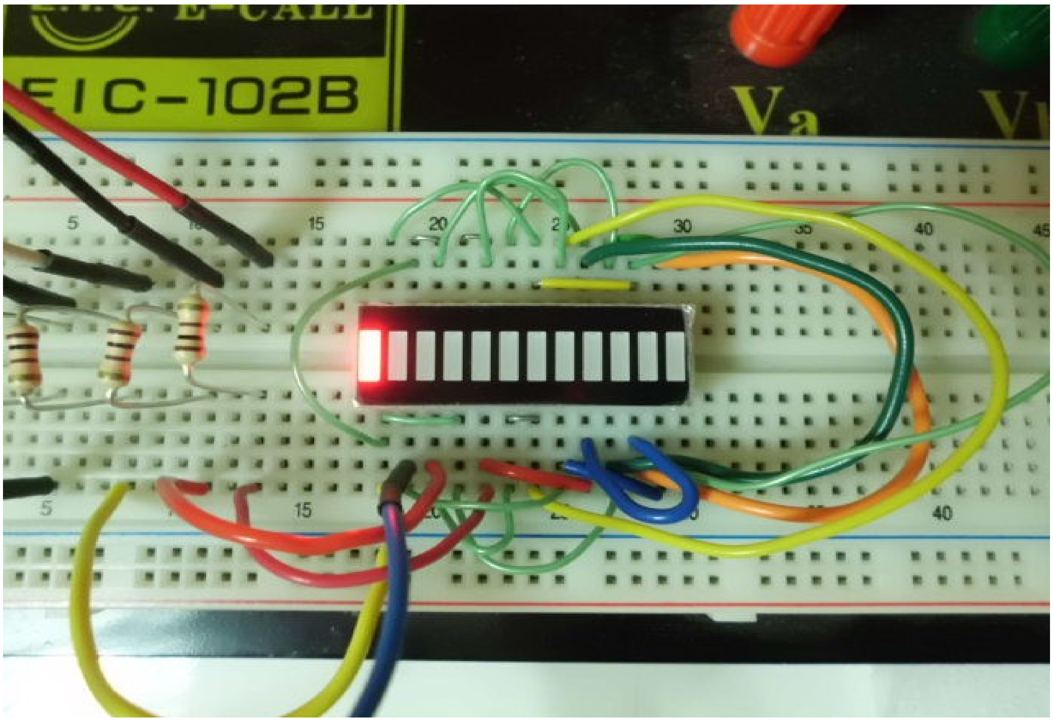
Charlieplexing

20141110

LED

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

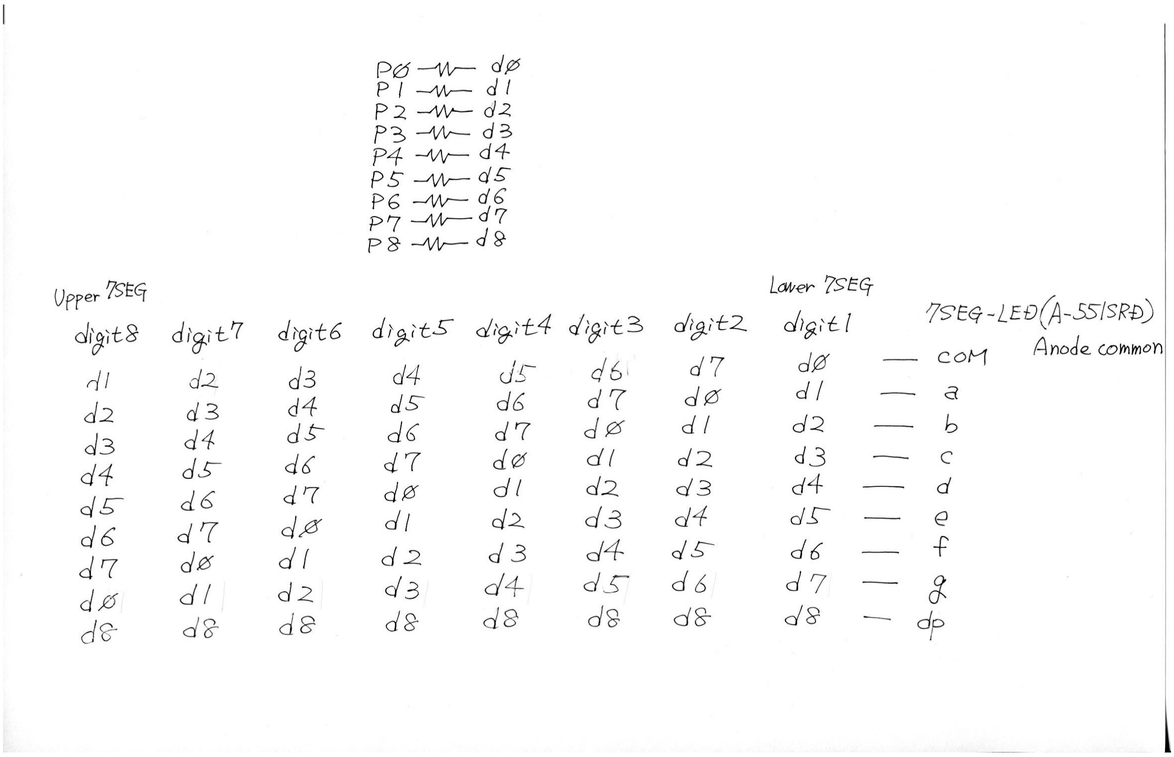




LED	IN/OUT Terminal pin				register	
	P3	P2	P1	P0	dira	outa
D1	Hi-Z	Hi-Z	0	1	3	1
D2	Hi-Z	Hi-Z	1	0	3	2
D3	Hi-Z	0	1	Hi-Z	6	2
D4	Hi-Z	1	0	Hi-Z	6	4
D5	0	1	Hi-Z	Hi-Z	hC	4
D6	1	0	Hi-Z	Hi-Z	hC	8
D7	Hi-Z	0	Hi-Z	1	5	1
D8	Hi-Z	1	Hi-Z	0	5	4
D9	0	Hi-Z	1	Hi-Z	hA	2
D10	1	Hi-Z	0	Hi-Z	hA	8
D11	0	Hi-Z	Hi-Z	1	9	1
D12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

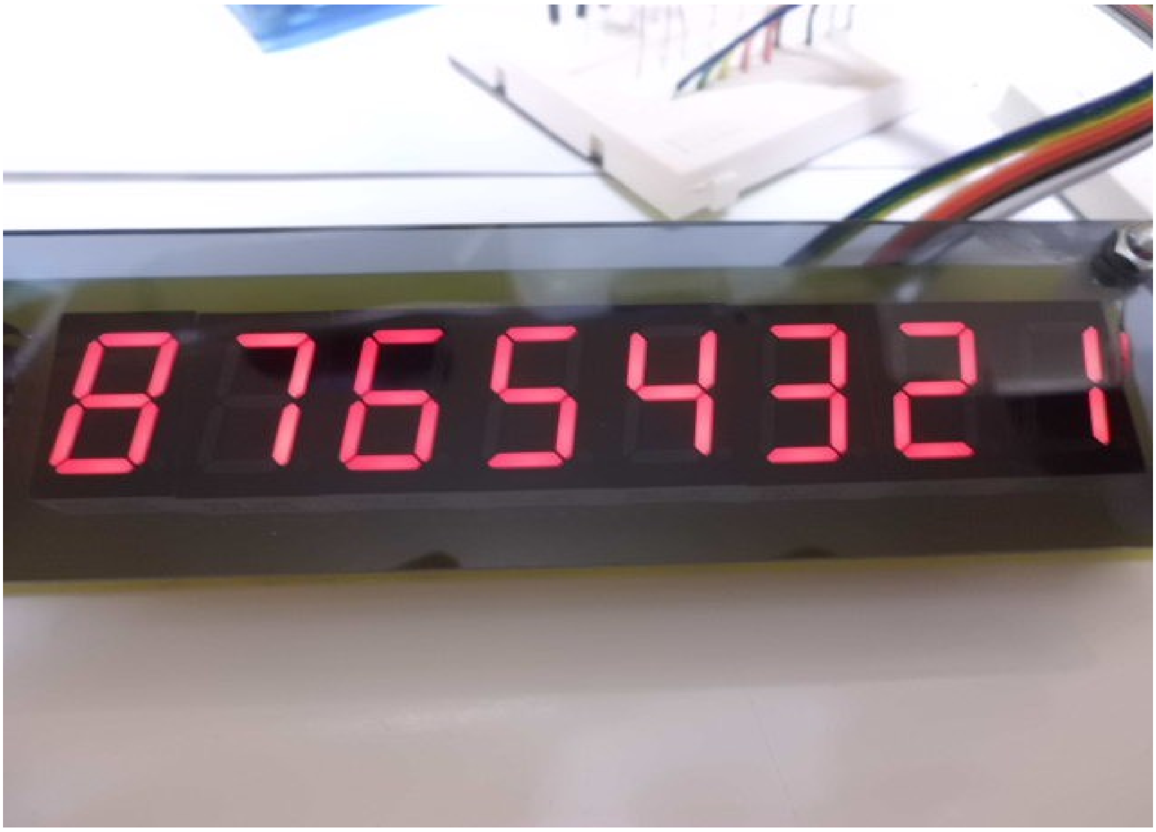
7Segment-LED

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



P0 —W— d0
P1 —W— d1
P2 —W— d2
P3 —W— d3
P4 —W— d4
P5 —W— d5
P6 —W— d6
P7 —W— d7
P8 —W— d8

Upper 7SEG							Lower 7SEG		
digit8	digit7	digit6	digit5	digit4	digit3	digit2	digit1		7SEG-LED(A-551SRD) Anode common
d1	d2	d3	d4	d5	d6	d7	d0	—	COM
d2	d3	d4	d5	d6	d7	d0	d1	—	a
d3	d4	d5	d6	d7	d0	d1	d2	—	b
d4	d5	d6	d7	d0	d1	d2	d3	—	c
d5	d6	d7	d0	d1	d2	d3	d4	—	d
d6	d7	d0	d1	d2	d3	d4	d5	—	e
d7	d0	d1	d2	d3	d4	d5	d6	—	f
d0	d1	d2	d3	d4	d5	d6	d7	—	g
d8	d8	d8	d8	d8	d8	d8	d8	—	dp



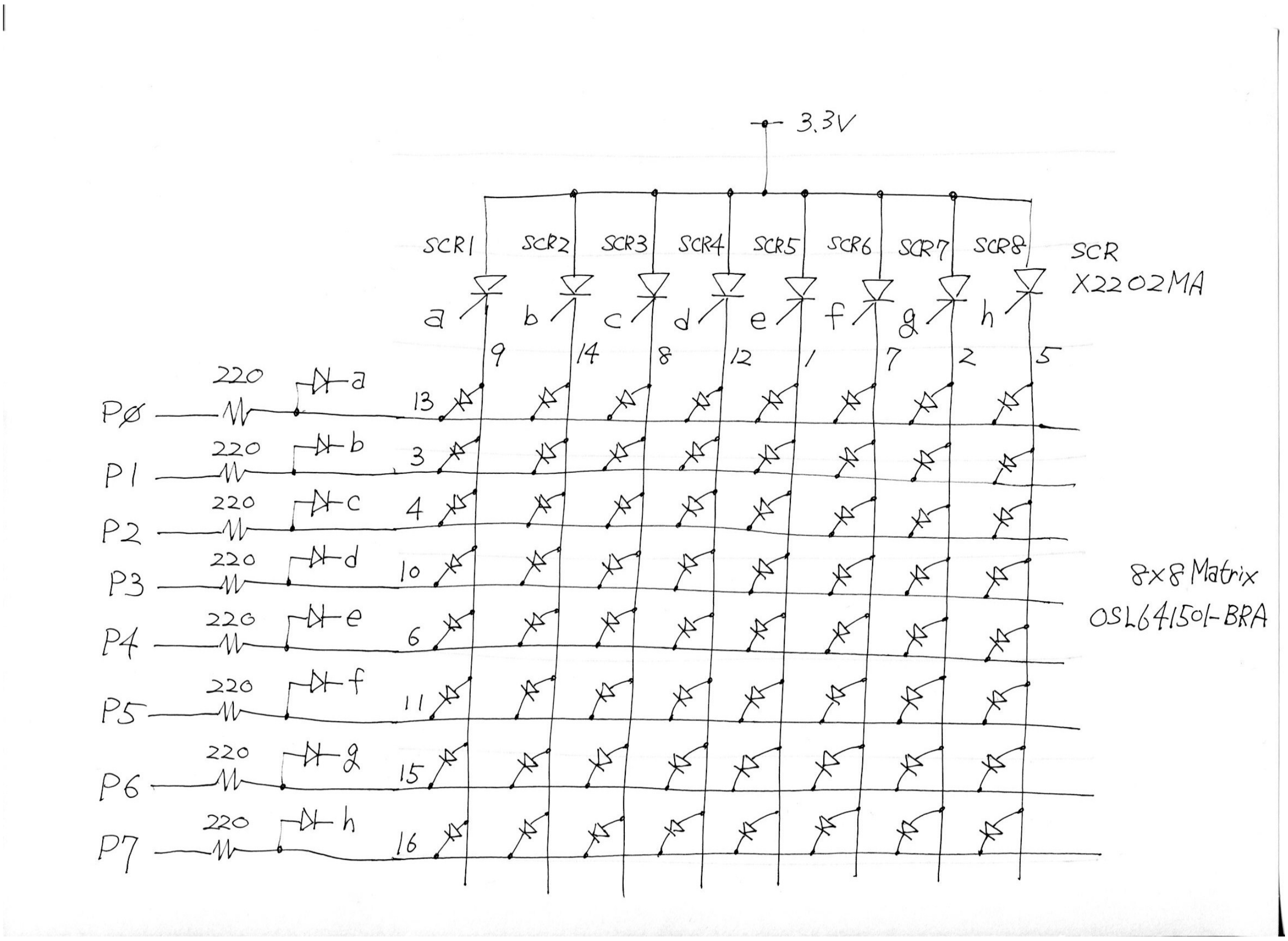
Current through 7Segment-LED's common is constant.
So, its brightness become low when going on many elements for 7Seg-LED, .
7Seg“1” is brighter than 7Seg“8” .
To prevent this, using Tr(emitter follower) on each pin[P0-P8] is recommended.

8x8Matrix-LED

Charlieplexing_8x8Matrix_0.3_1.f

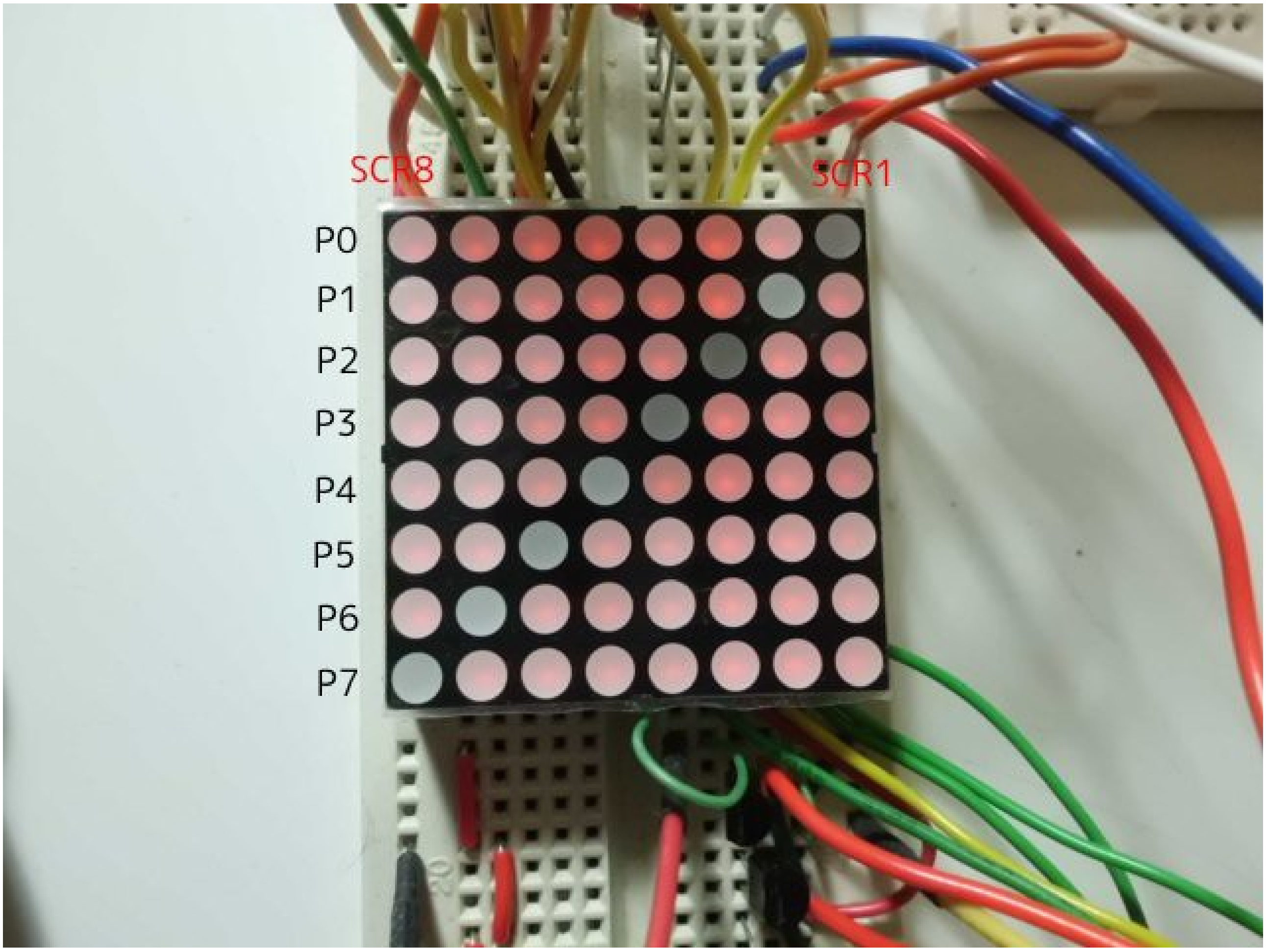
N*N Matrix-LED can controlled by N-wires.

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



1. Set all pin to Hi-Z.

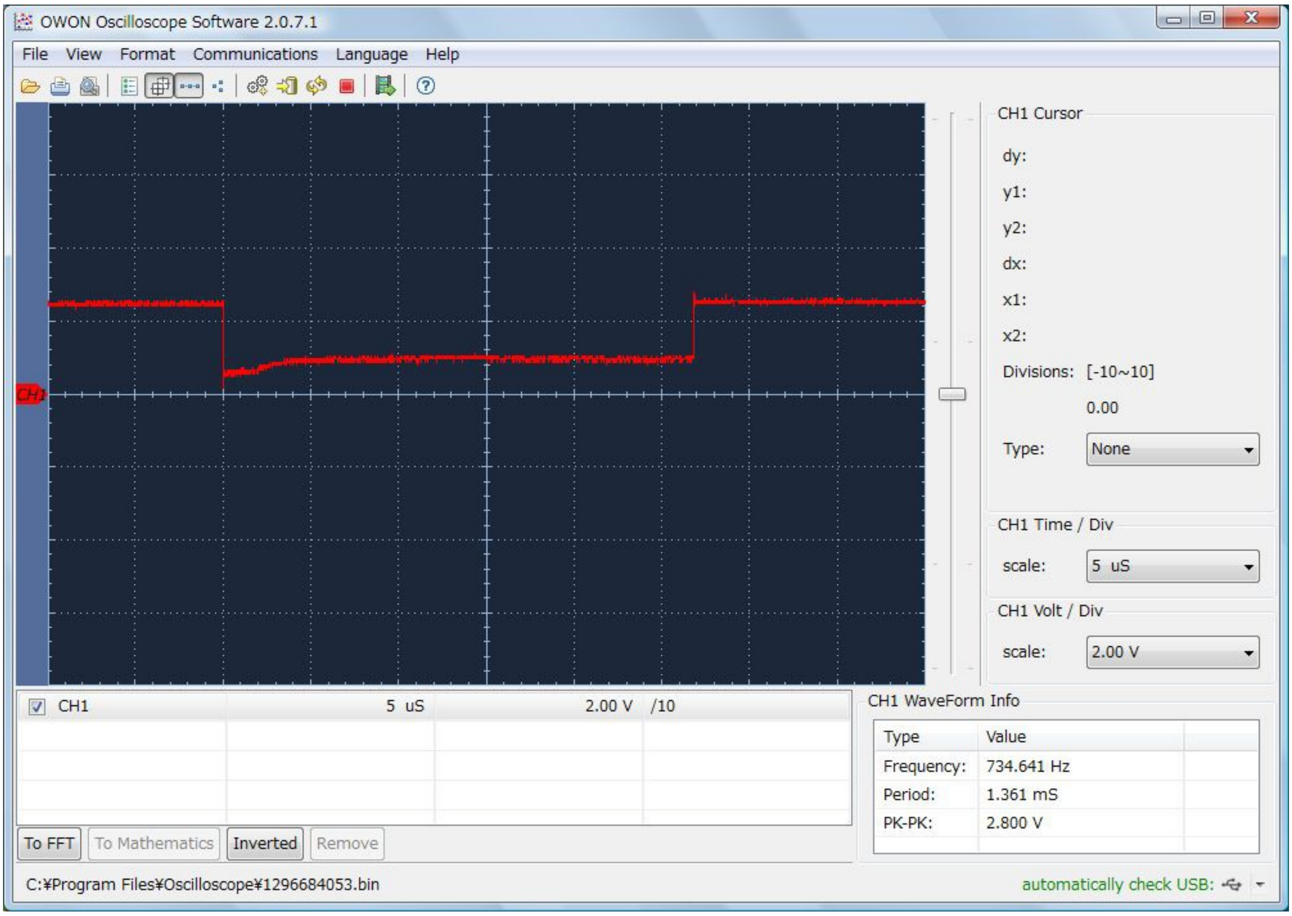
```
0 dira COG!
0 outa COG!
```

All SCRs are deactivated.

2. Activating SCR.

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

SCR1 is activated because P0 is Hi.
LED connected to P0 is off.
Other LEDs is on because port[P1-P7] is Lo.
SCR-on-time is short.   About 27usec.
Its time is from saving hFF to dira to saving hFF to outa.
[buffer”matrix” are fulled to 0]
Osciloscope below indicate SCR1's cathode.



CH1 Cursor

dy:

y1:

y2:

dx:

x1:

x2:

Divisions: [-10~10]

0.00

Type: None

CH1 Time / Div

scale: 5 uS

CH1 Volt / Div

scale: 2.00 V

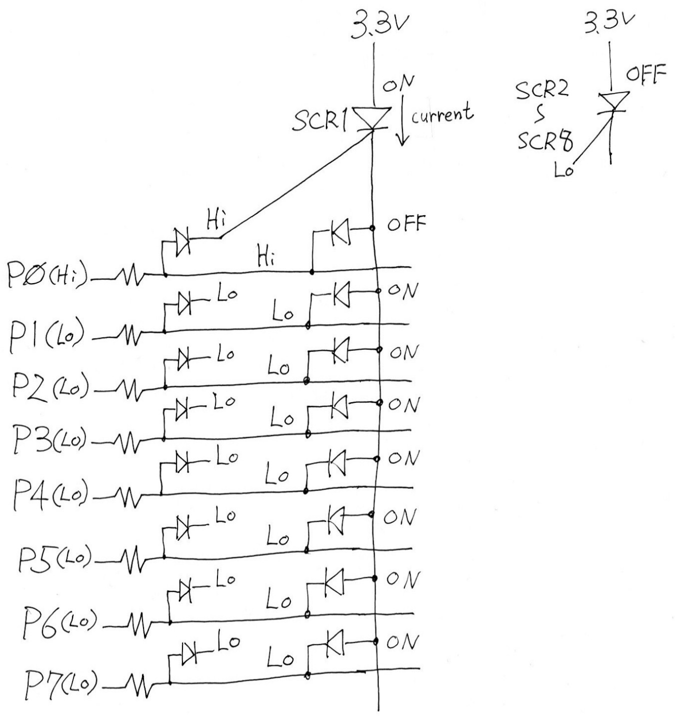
☑ CH1 5 uS 2.00 V /10

CH1 WaveForm Info

Type	Value
Frequency:	734.641 Hz
Period:	1.361 mS
PK-PK:	2.800 V

C:¥Program Files¥Oscilloscope¥1296684053.bin

automatically check USB:



3. Set data inside buffer[matrix] to outa.

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

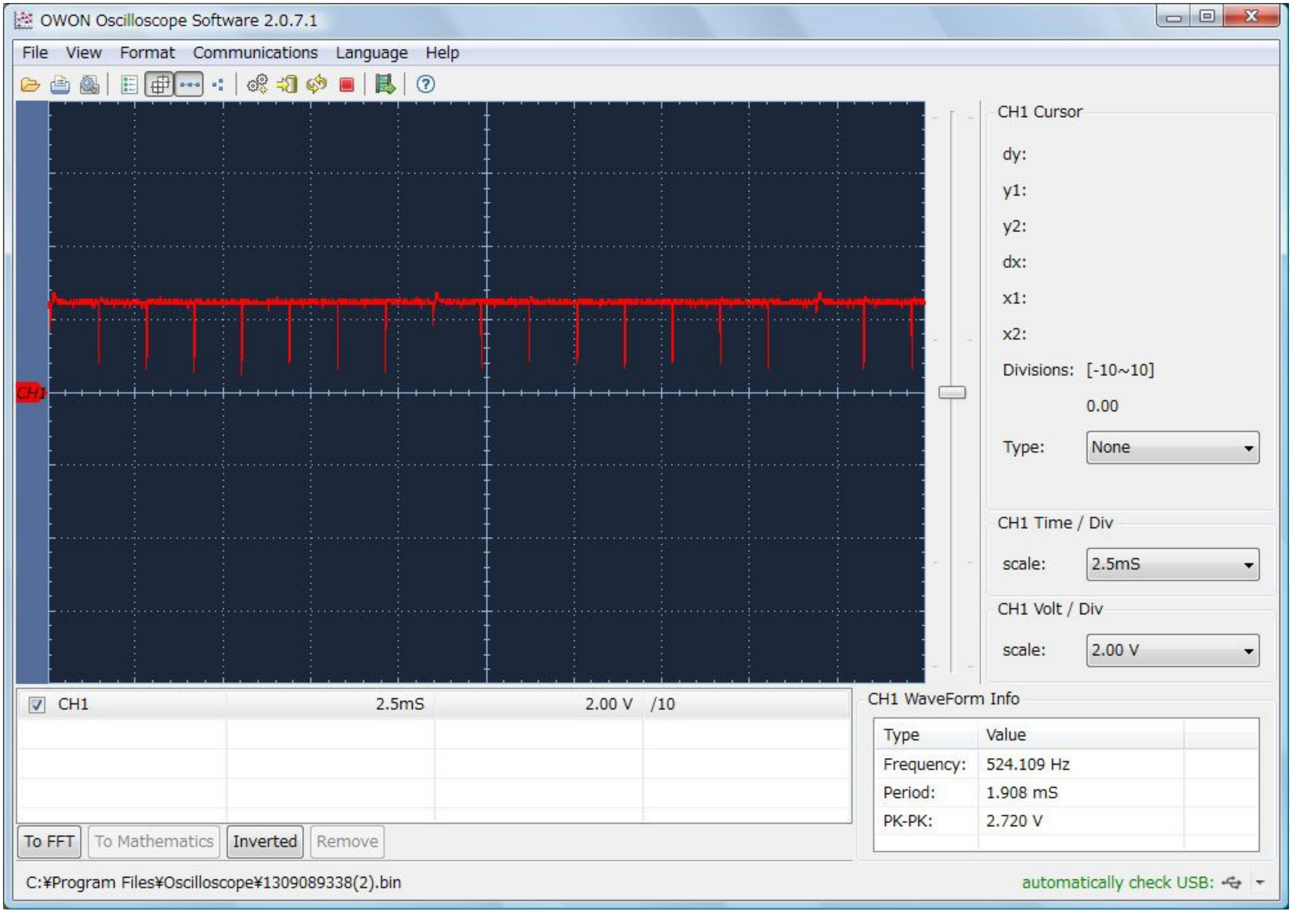
Data are inverted and saved to outa-register.
All LEDs are off because of hFF.
In this case, SCR1 is deactivated.
If there is data“0”, SCR1 keep to activate.

4. Keep this during display.

```
1 delms
```

5. Repeat [step1-step4] 8 times.

6. Repeat step5.
1-cycle is about 11msec.
First SCR1's wave is left.
Osciloscope below indicate SCR1's cathode.



CH1 Cursor

dy:
y1:
y2:
dx:
x1:
x2:
Divisions: [-10~10] 0.00
Type: None

CH1 Time / Div
scale: 2.5mS

CH1 Volt / Div
scale: 2.00 V

☑ CH1 2.5mS 2.00 V /10

CH1 WaveForm Info

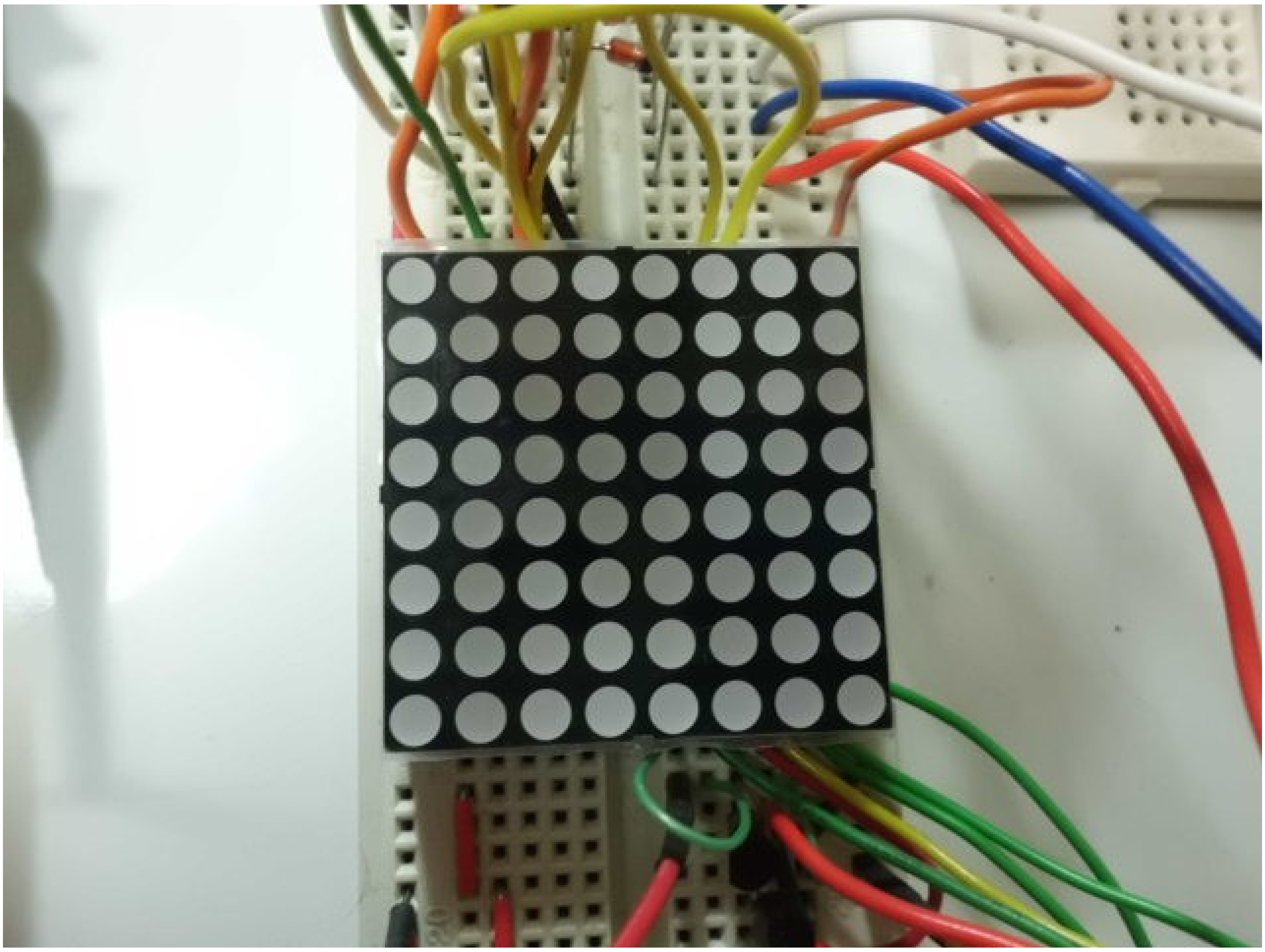
Type	Value
Frequency:	524.109 Hz
Period:	1.908 mS
PK-PK:	2.720 V

C:¥Program Files¥Oscilloscope¥1309089338(2).bin

automatically check USB:

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

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-character.

3. Set data inside buffer[matrix] to outa.4. Keep this during display.

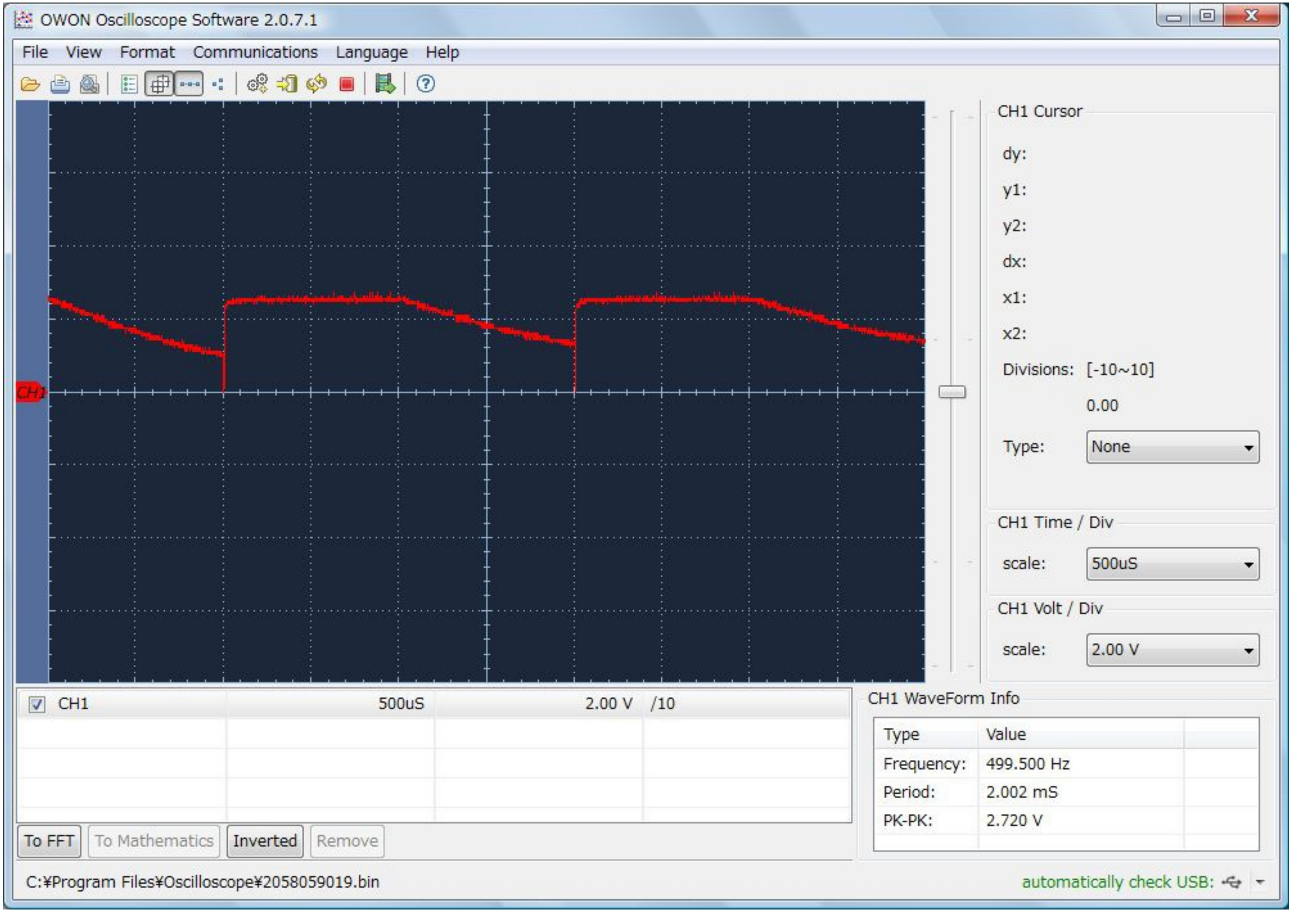
```
rdbyte   $C_treg4 , $C_treg3
neg      $C_treg5 , $C_treg4
sub      $C_treg5 , # 1
mov       outa , $C_treg5
```

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

4. Keep this during display.

```
jmpret  __delayret , # __delay
```

Each SCR-line takes about 2msec.
Part of wave(low pulse) is step 2.
Next wave(3.3V) is step3 and 4.
Next discharged wave is step1.
Osciloscope below indicate SCR1's cathode.



CH1 Cursor

dy:
y1:
y2:
dx:
x1:
x2:
Divisions: [-10~10]
0.00
Type: None

CH1 Time / Div
scale: 500uS

CH1 Volt / Div
scale: 2.00 V

☑ CH1 500uS 2.00 V /10

CH1 WaveForm Info

Type	Value
Frequency:	499.500 Hz
Period:	2.002 mS
PK-PK:	2.720 V

C:¥Program Files¥Oscilloscope¥2058059019.bin

automatically check USB:

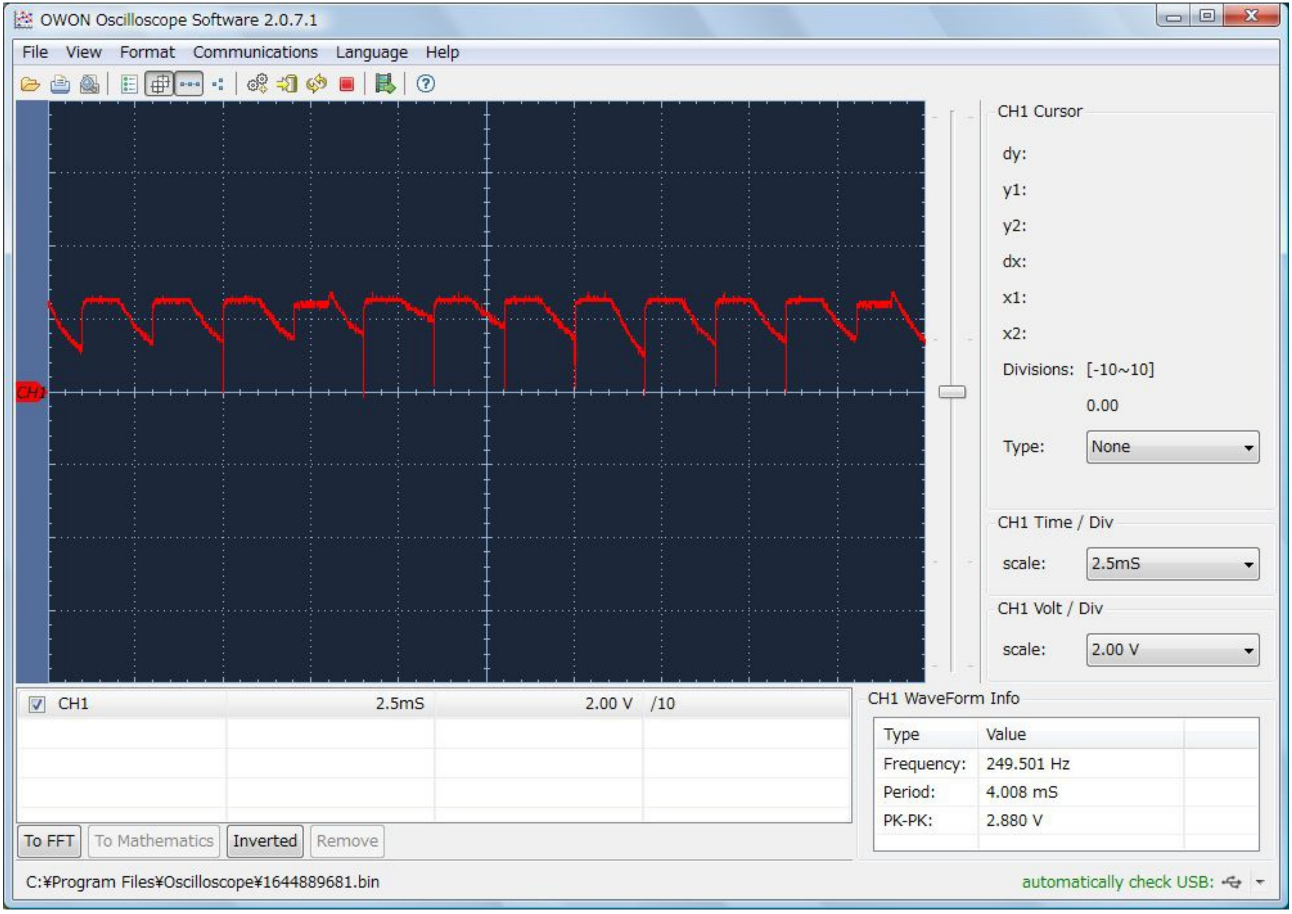
5. Repeat [step1-step4] 8 times.

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

6. Repeat step5.

```
jmp      # __1
```

1-cycle is about 16msec.
Osciloscope below indicate SCR1's cathode.



OWON Oscilloscope Software 2.0.7.1

CH1 Cursor

dy:
y1:
y2:
dx:
x1:
x2:
Divisions: [-10~10] 0.00
Type: None

CH1 Time / Div
scale: 2.5mS

CH1 Volt / Div
scale: 2.00 V

☑ CH1 2.5mS 2.00 V /10

CH1 WaveForm Info

Type	Value
Frequency:	249.501 Hz
Period:	4.008 mS
PK-PK:	2.880 V

C:¥Program Files¥Oscilloscope¥1644889681.bin

automatically check USB: