

LED_pwm

20131010

This change brightness of LED by PWM.

Reference; Chapter7.Counter Modules and Circuit Application Lab in Propeller Education kit Labs

LED_pwm_f

We use NCO/PWM counter-mode.

1. Configure the ctra/ctrb register
2. Set the frqa/frqb register
3. Set the I/O pin to output

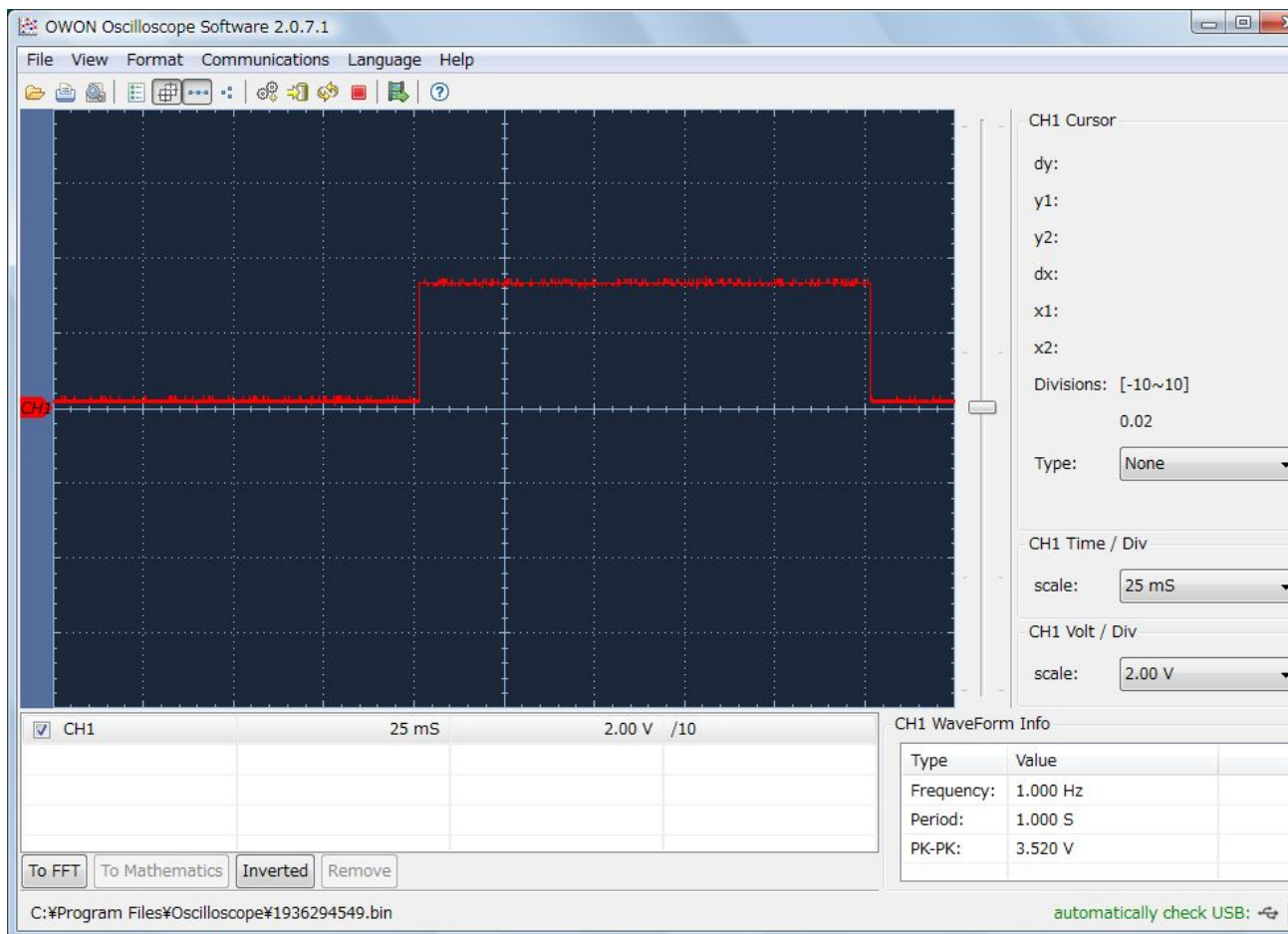
Single LED

demo1

Cycle time is 1second.

Value to phsa is given as negating dT(10000000ticks:125msec)

Wave below is on P0.

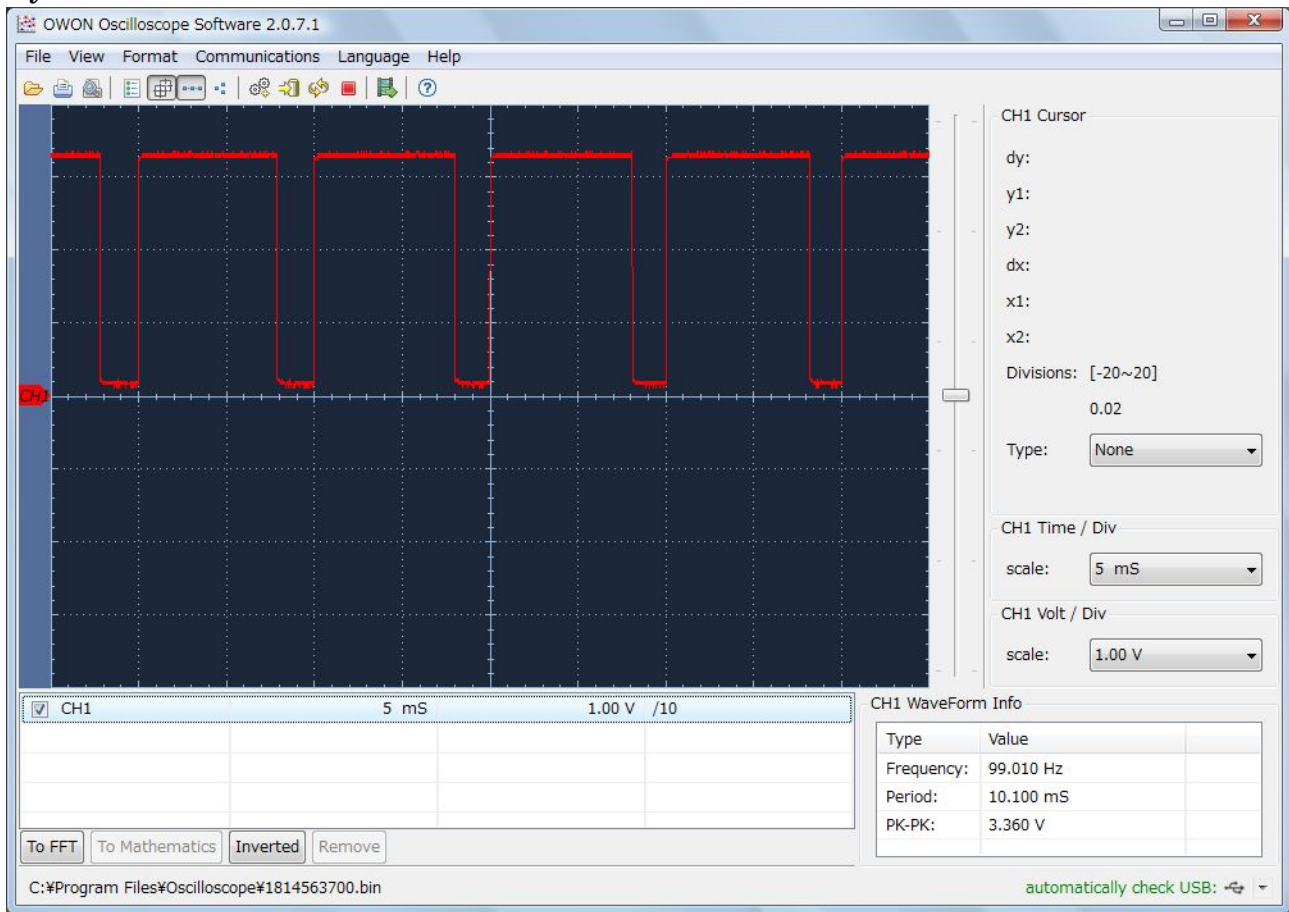


When phsa become to zero by adding frqa, signal(P0) become to Low(LED is on).

Hi-pulse is 125msec.

demo2

Cycle time is 10msec.



Loop(begin -- until) is repeated 100 times because base value for phsa is 100usec.
Off-time for LED change to increase 100usec by 100usec from 100usec to 10msec.

LED go on increasing brightness and turn off.
Again repeating.

demo3

This modified a little bit demo2.

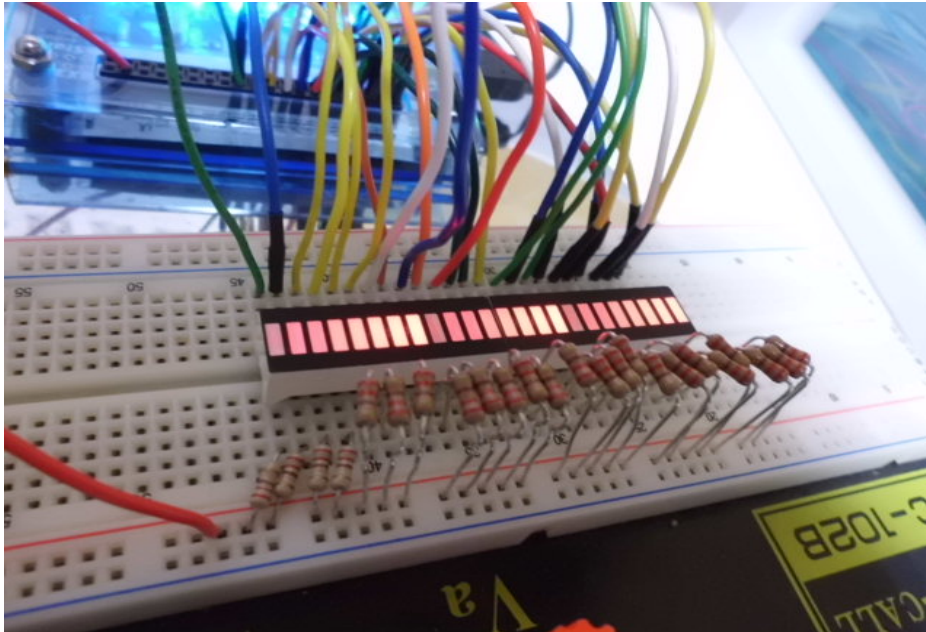
LED go on increasing brightness and decreasing brightness.
Again repeating.

LED_bar

LED_bar_test1

This use "drive_LED_bar"(forth-word).

Defined pattern is displayed on 2 LED_bars.



Time driving each LED is 1msecond.

Start for 1msec is input status.

Part at "dup i + pinout" is set to output and become to low.

It takes about 60usec at "begin - d24 0 do - dup i + pinout".

PWM start after finishing phsa-calculating and storing negative value to phsa.

So, PWM by forth-word is not correct.

Time driving each LED is 1msecond.

Start for 1msec is input status.

Part at "dup i + pinout" is set to output and become to low.

It takes about 40usec at "begin - d24 0 do - dup i + pinout".

PWM start on finishing phsa-calculating

Code below are time to take calculating phsa-value

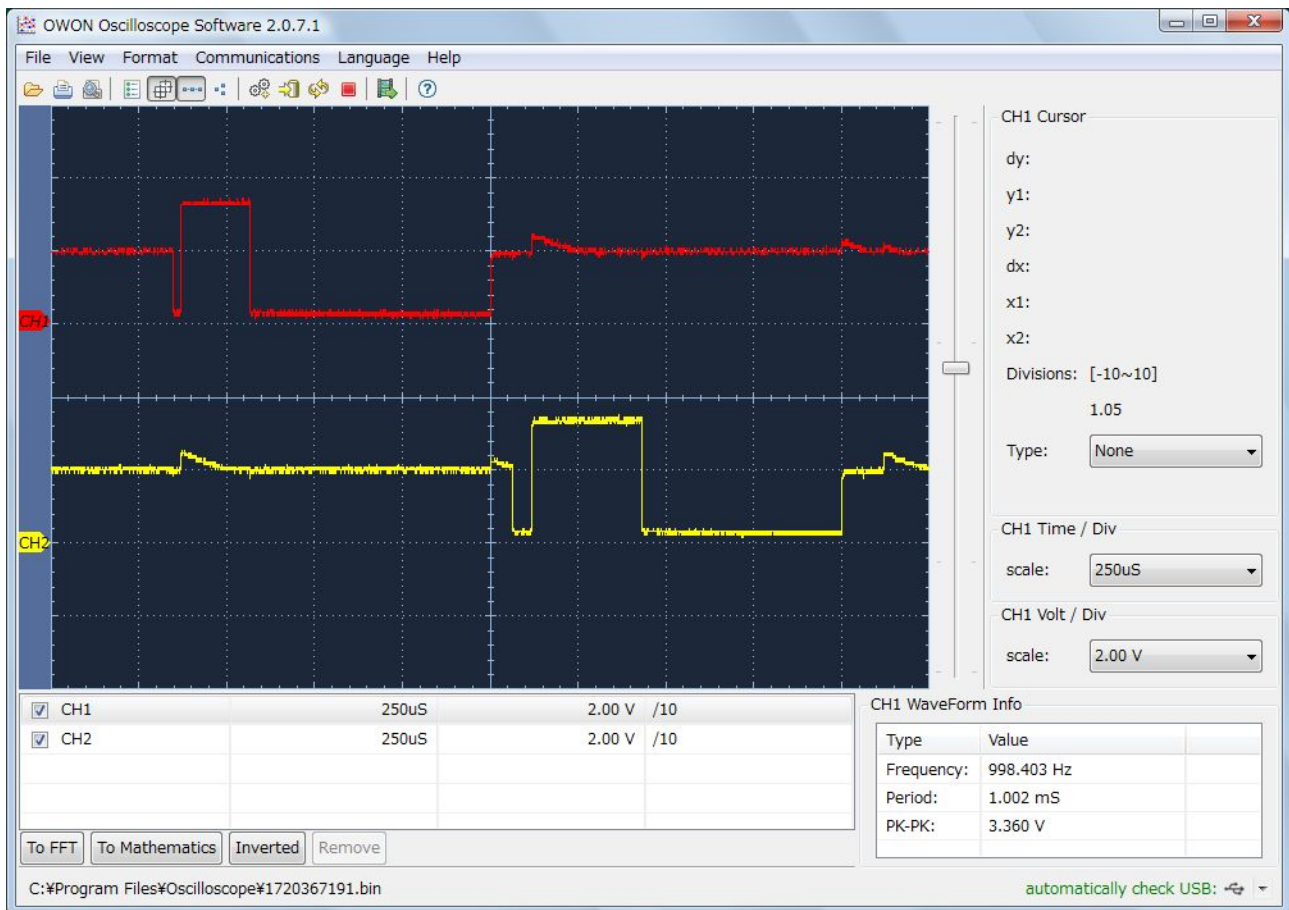
i.e. this is overhead-time for PWM

inside begin - until loop

```
dup i + h10000000 or ctra COG! \ PWM/NCO mode on LED pin
i LED_level + C@ \ Get level for target LED
dT L@ u*
negate phsa COG!
```

This take time about 5040ticks(63usec)

When pased 1msec, port is back to input.



Red:P0

Point from floating to low is P0=output.

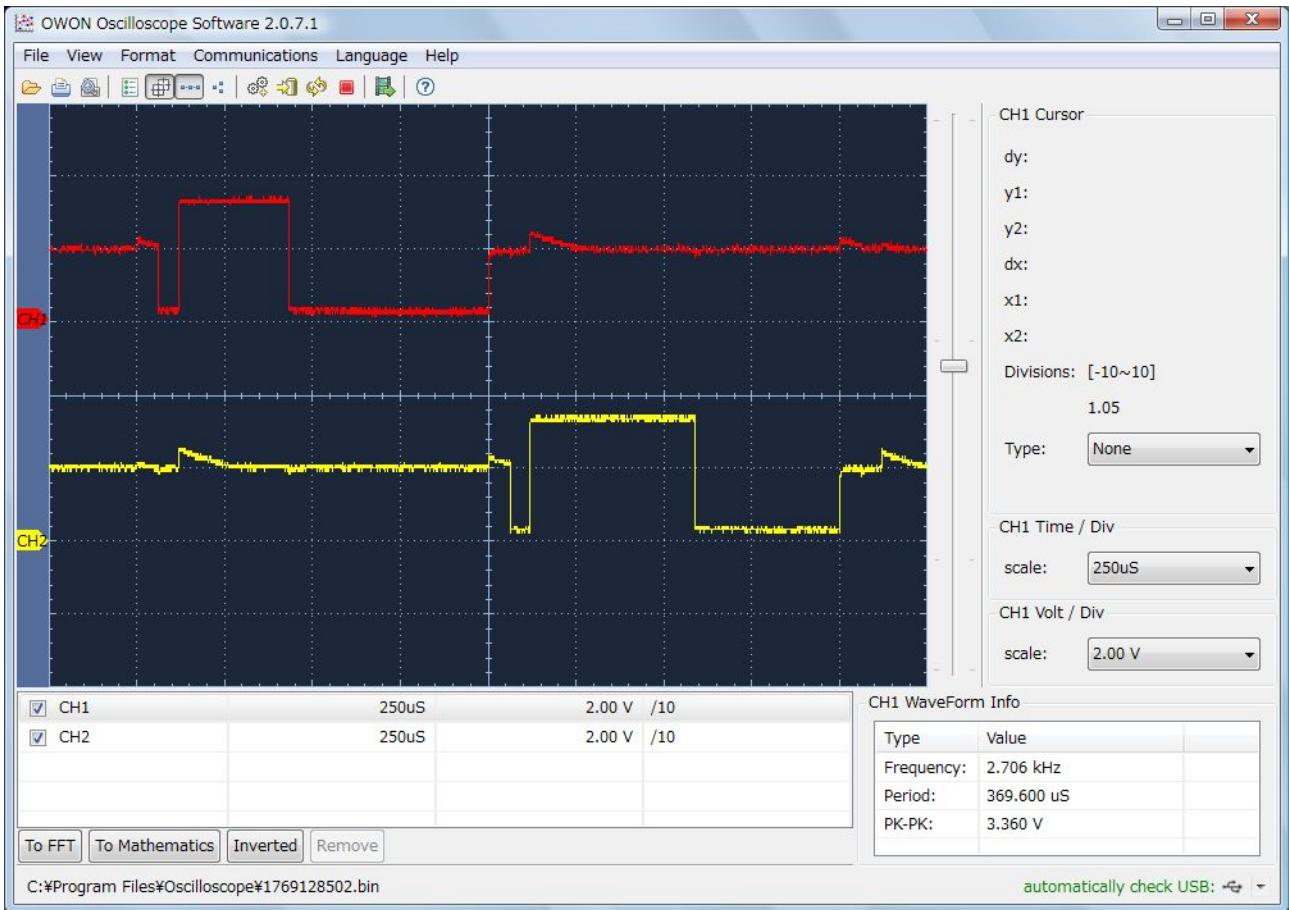
At next high, PWM start.

Yellow:P1

When P0 become to floating(P0 deselected), P1 is selected.

Low-pulse is shorter than P0. I have no idea about this.

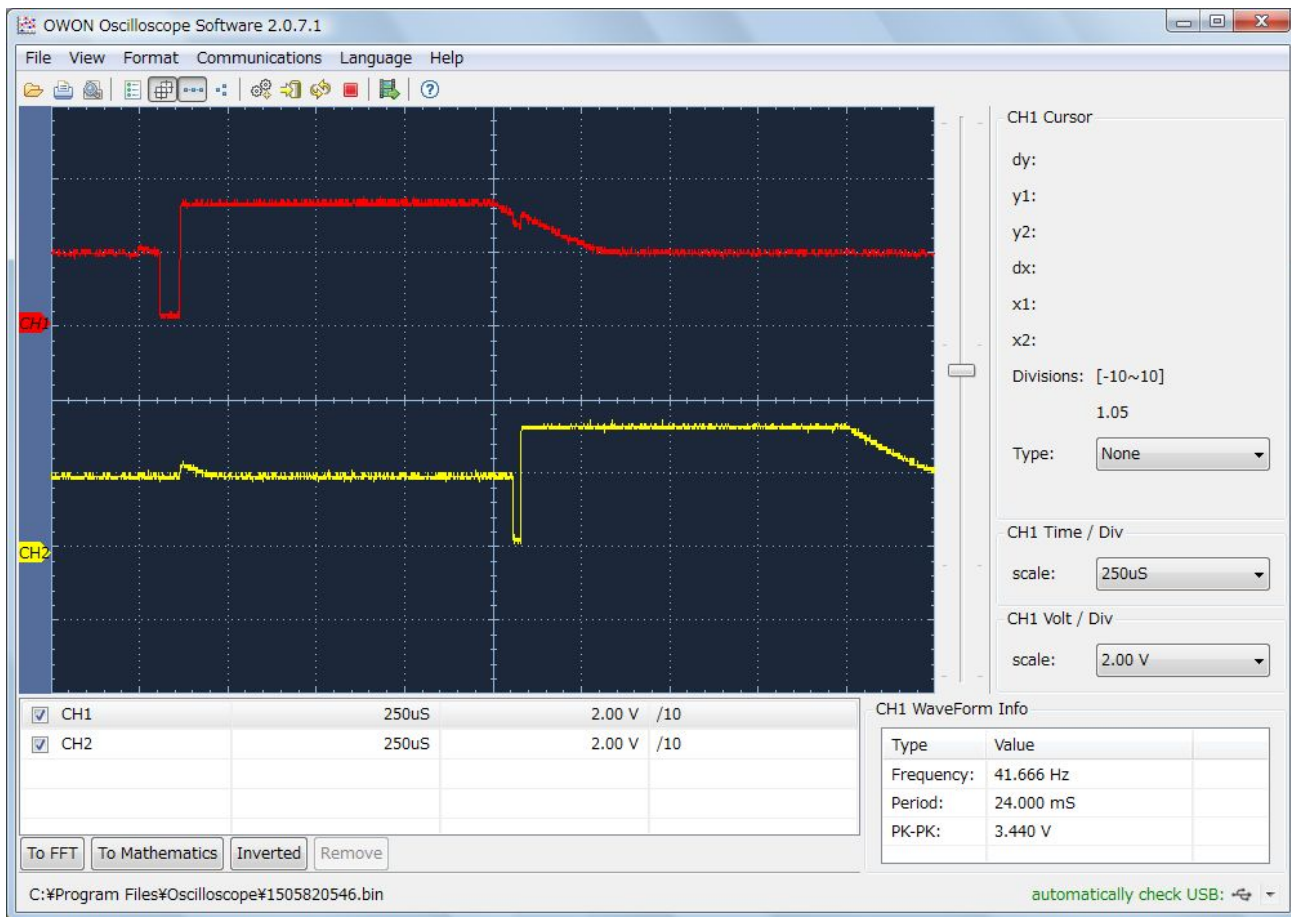
After P* selected, it takes about 100usec starting PWM.



Red:P1

Yellow:P2

Both signal almost same.



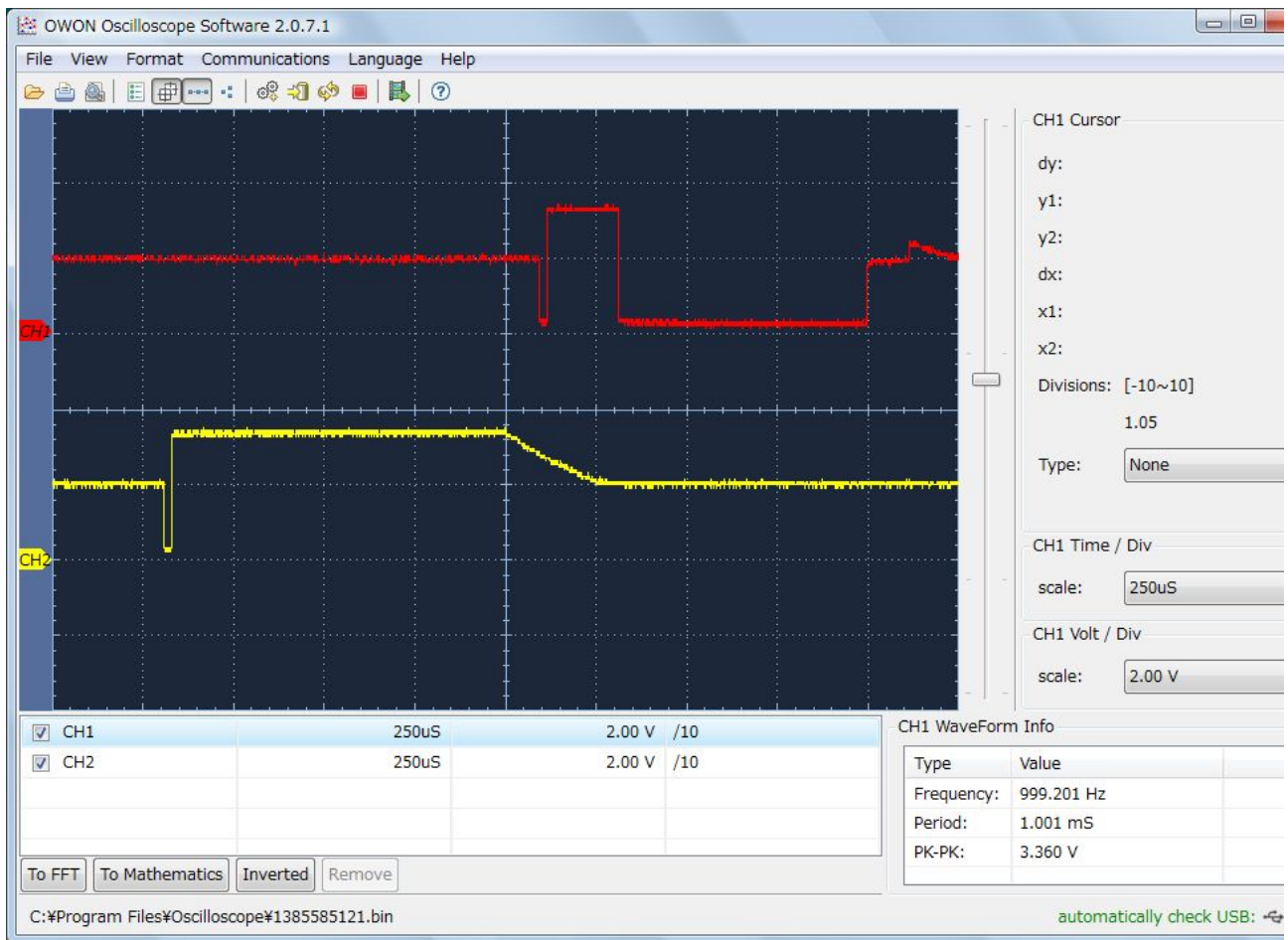
Red:P22

Yellow:P23

Both PWM-signal(High pulse) do not finish, because there is overhead-time for PWM. Although counter-output is high, P23 and P23 is swithed to input by waitcnt. So, High-level slowly become to floating.

P23's low pulse is short than other.

I also have no idea about this.



Red:P0

Yellow:P23

LED_bar_test2

This also use "drive_LED_bar"(forth-word).

Defined pattern(not LED_bar_test1) is displayed on 2 LED_bars.

demo4

This also use "drive_LED_bar"(forth-word).

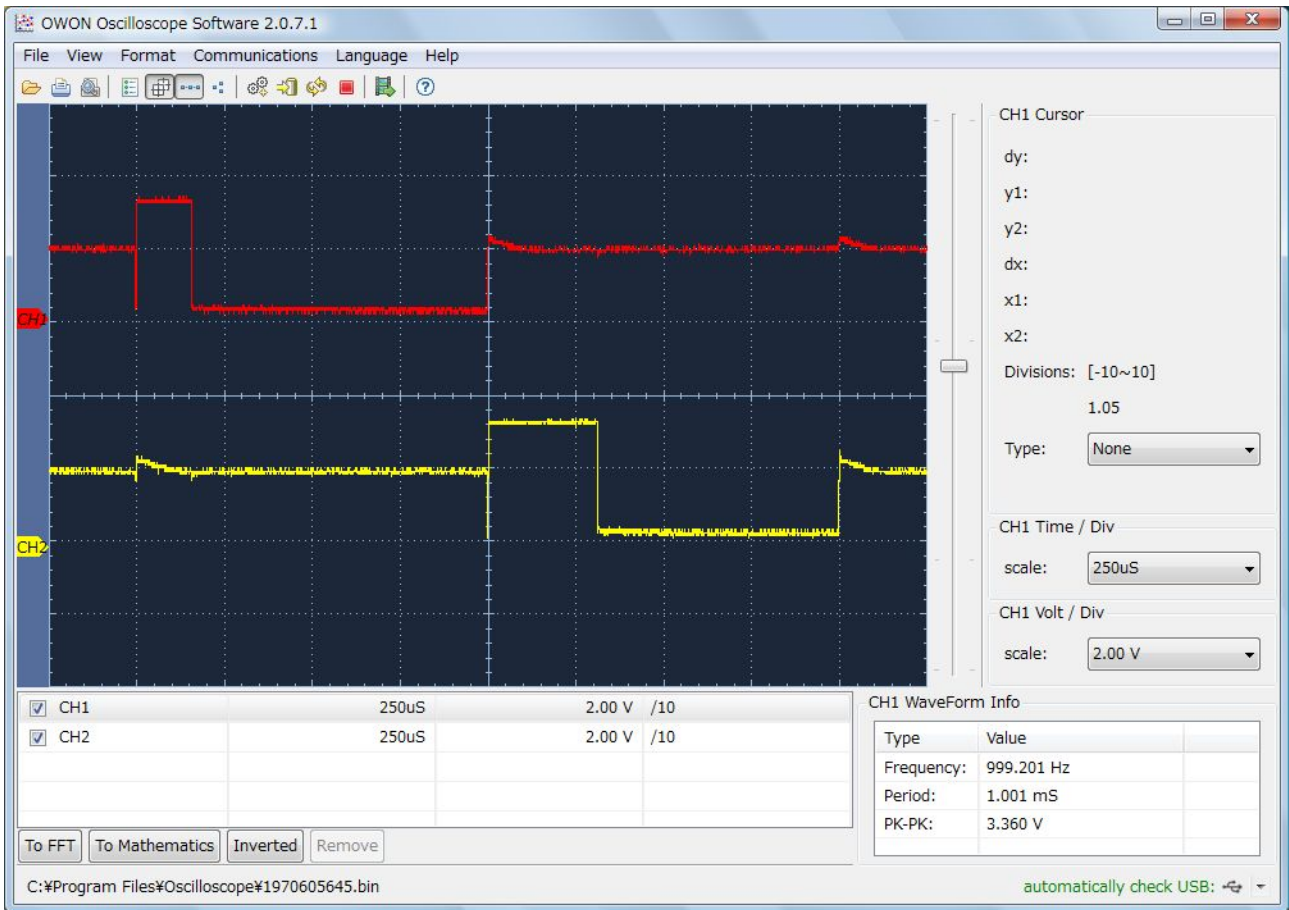
Defined pattern is moving.

Actually, human eyes recognize moving.

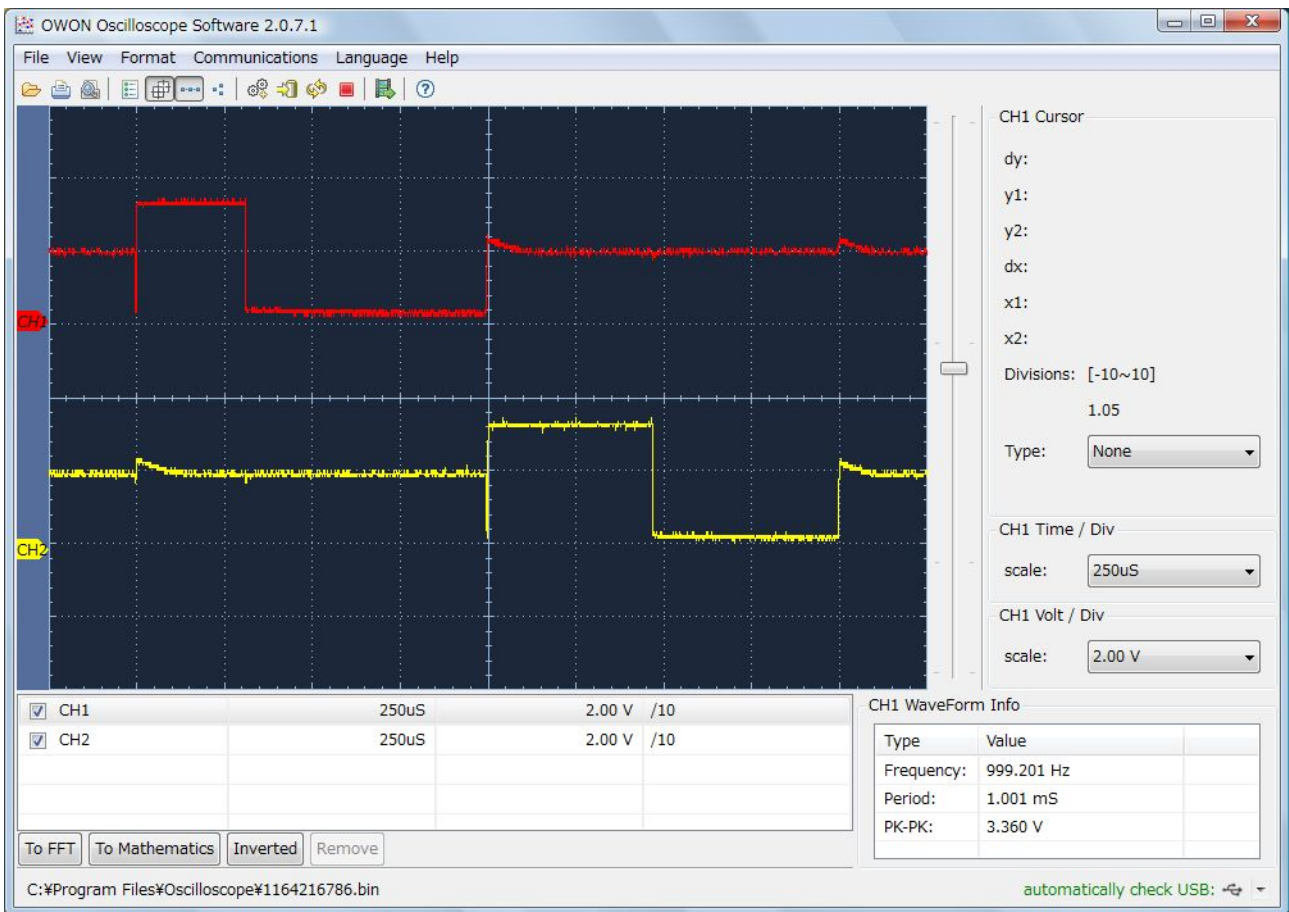
asm_LED_bar_test1

This use "a_drive_LED_bar"(fasmsembler-word).

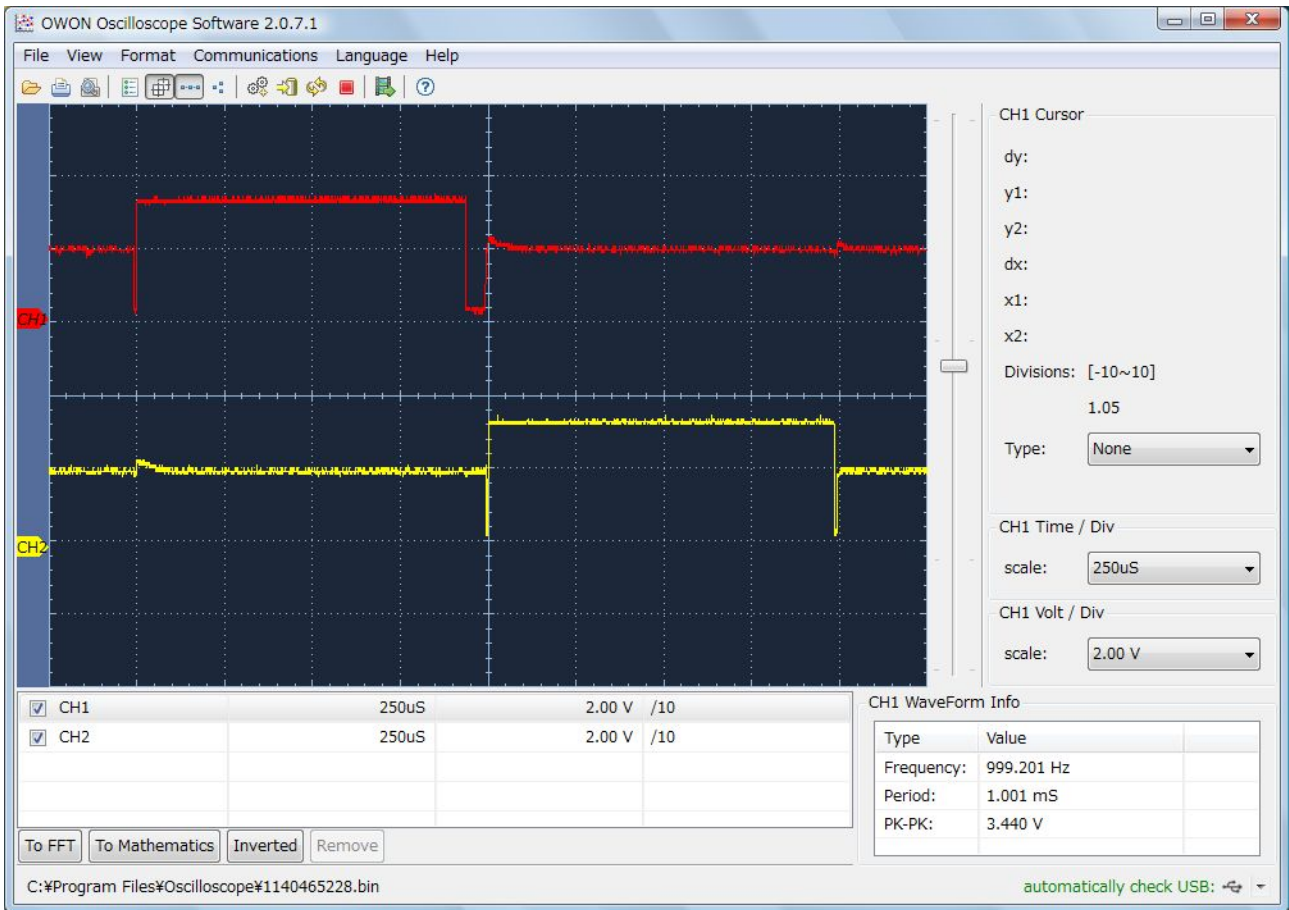
Pattern is same as "LED_bar_test1".



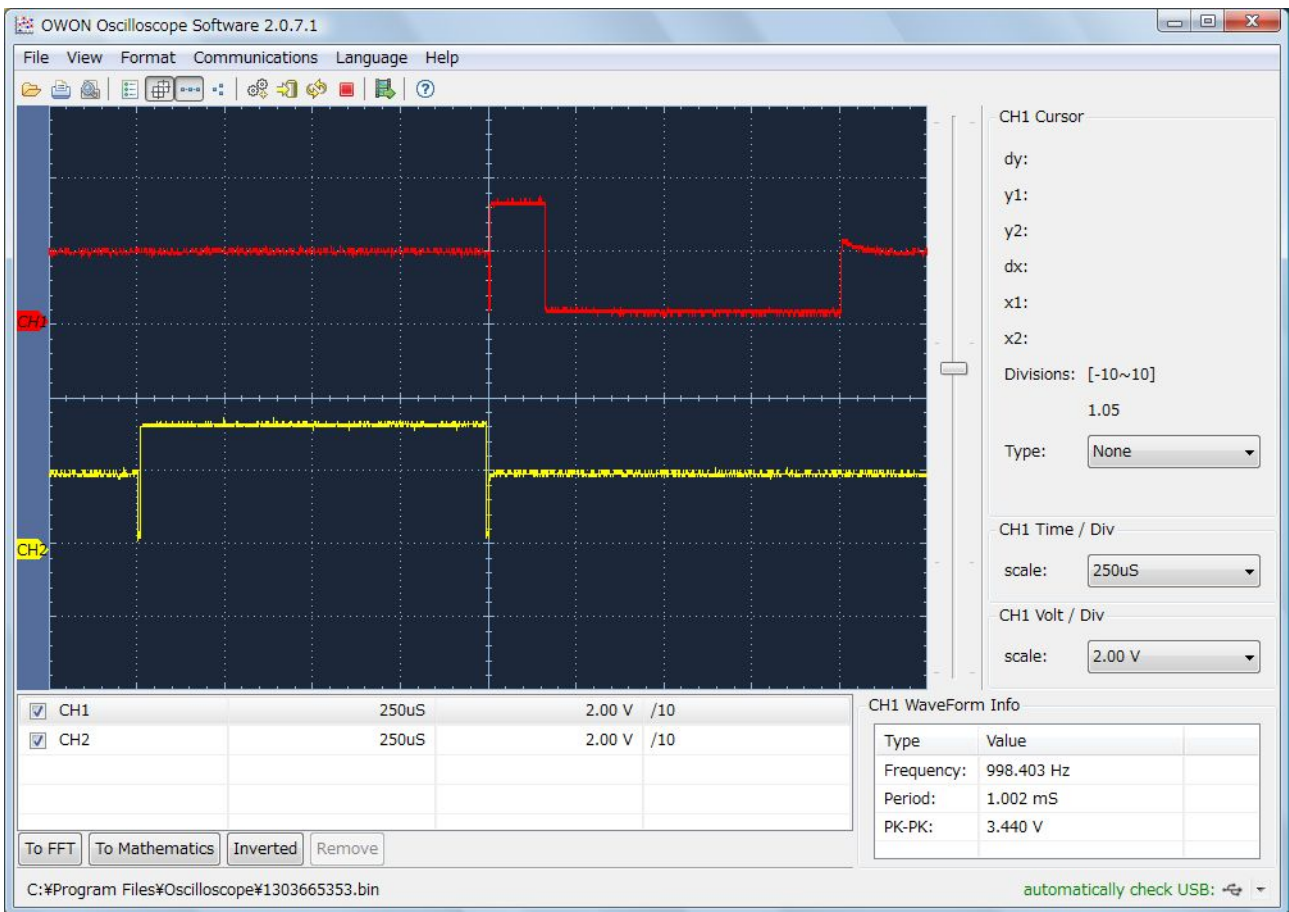
Red:P0 Yellow:P1



Red: P1 Yellow:P2



Red:P22 Yellow:P23



Red:P0 Yellow:P23

Time calculating phsa-value is shorter than forth-word because using assembler-word.