

CharacterLCD

20160130

CharacterLCD generally use HD44780 compatible products.

This use 6 wires[RS,E,DB4-DB7] for signals on 4bit mode.

Adding power line, needing wires are 8!!.

I have used breadboard at test.

I often have been wrong connections.

So, I wanted to reduce connected wires.

I wanted to make reduced wire characterLCD by using normal characterLCD.

2_wire_LCD

Reference: 2_wire_chaLCD_0.1.f

This convert data from serial to parallel by using shift-regisre(74HC164).

Operation:

Firstly it make 7bit data by forth word.

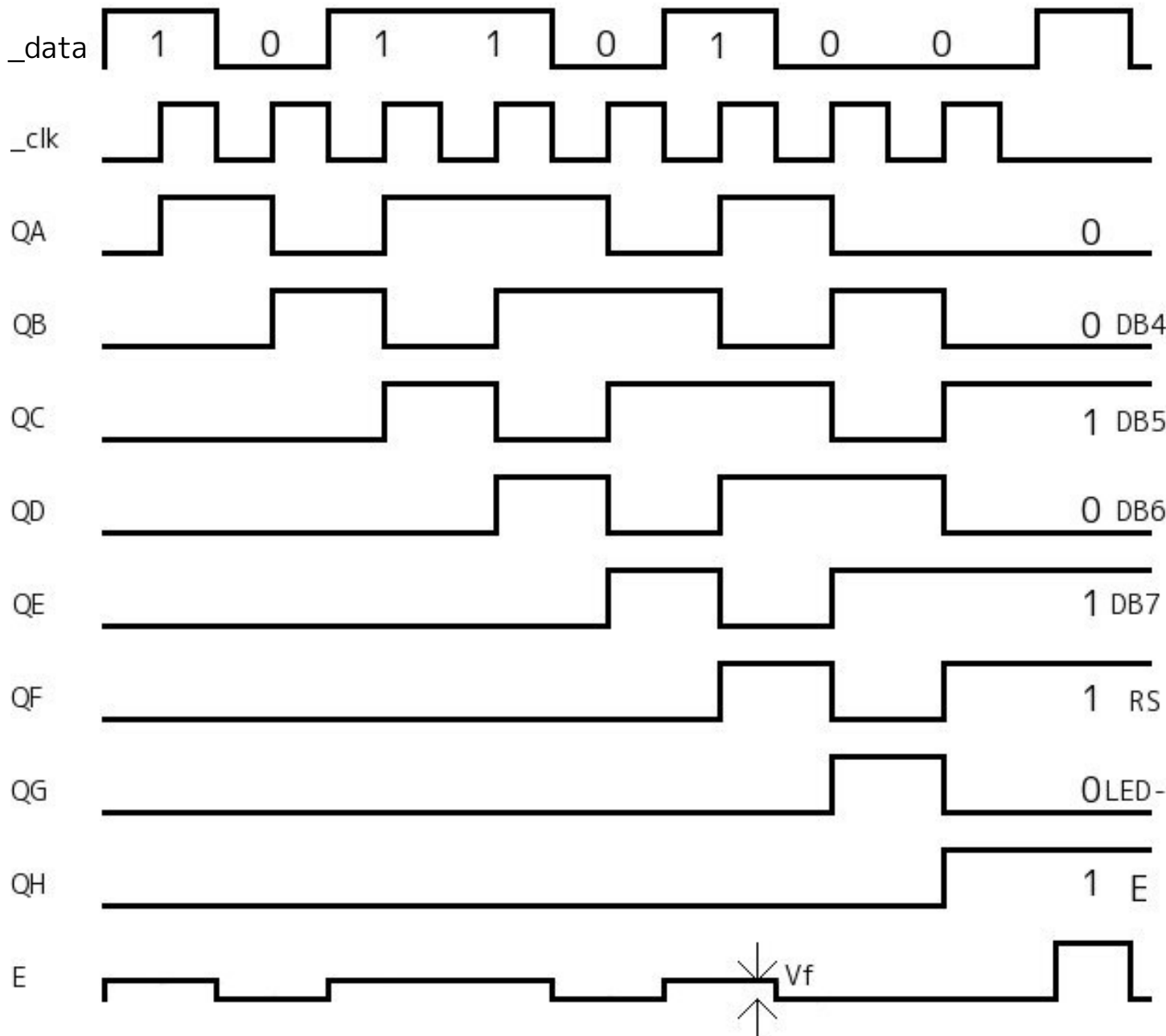
bit6:Enable bit5:LED-(always'0') bit4:RS bit3:DB7 bit2:DB6 bit1:DB5 bit0:DB4

Secondly 8bit data is sent to shift-register by assembler-word'a_shift_data'.

bit7:Enable bit6:LED- bit5:RS bit4:DB7 bit3:DB6 bit2:DB5 bit1:DB4 bit0:'0'

This use 4bit control mode.

Signal diagram is below;



Enable signal on characterLCD is tripped by diode while QH for shift-register is low. After QH for shift-register reach to hi, data is written to characterLCD by Hi-pulse on _data.

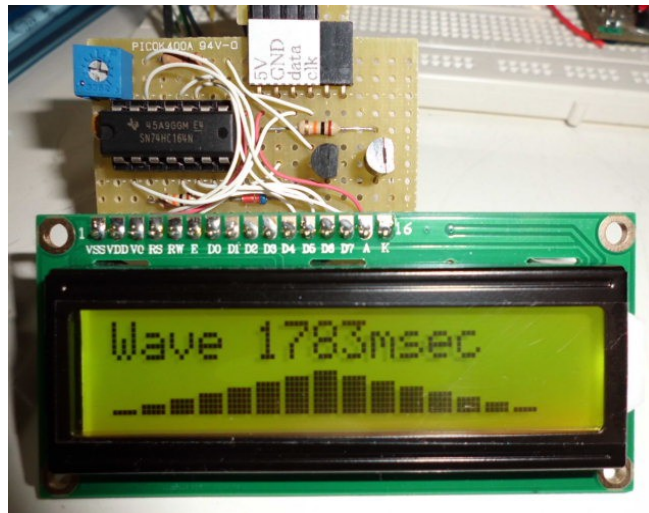
Hi-pulse is entered to E on characterLCD because diode is off.

And then signal on shift-register are shifted out by assembler-word 'a_clk_out'.

'demo1' display characters[h20 – h7F].



'demo2' look like moving wave.
It takes 1783msec.



Caution;

From 74HC164 datasheet, V_{IH} is min3.49V at $V_{CC}=5V$ (min3.15V at $V_{CC}=4.5V$).

It doesn't satisfy V_{IH} for 74HC164 because Propeller's V_{OH} is min2.85V(actualy 3.3V).

There is no problem although direct connecting between Propeller's output and 74HC164.

But level-shift might be needed on safety operation.

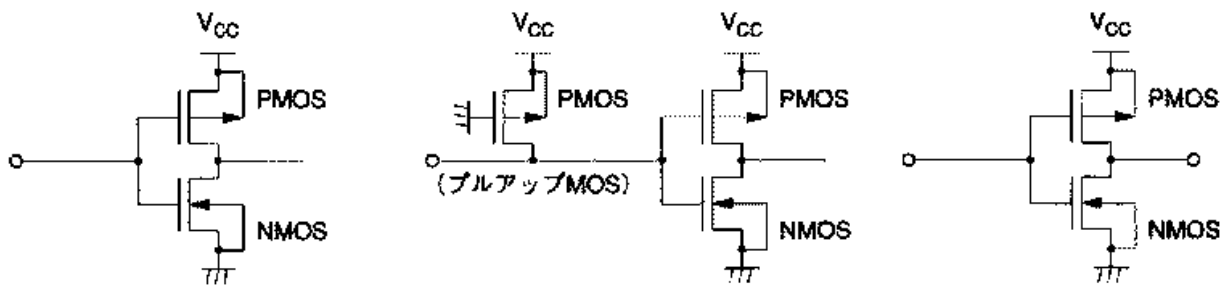
Port circuit diagram of HD44780 is below;

入力端子の形状

適用端子：E (プルアップMOSなし) 適用端子：RS, R/W (プルアップMOS付き)

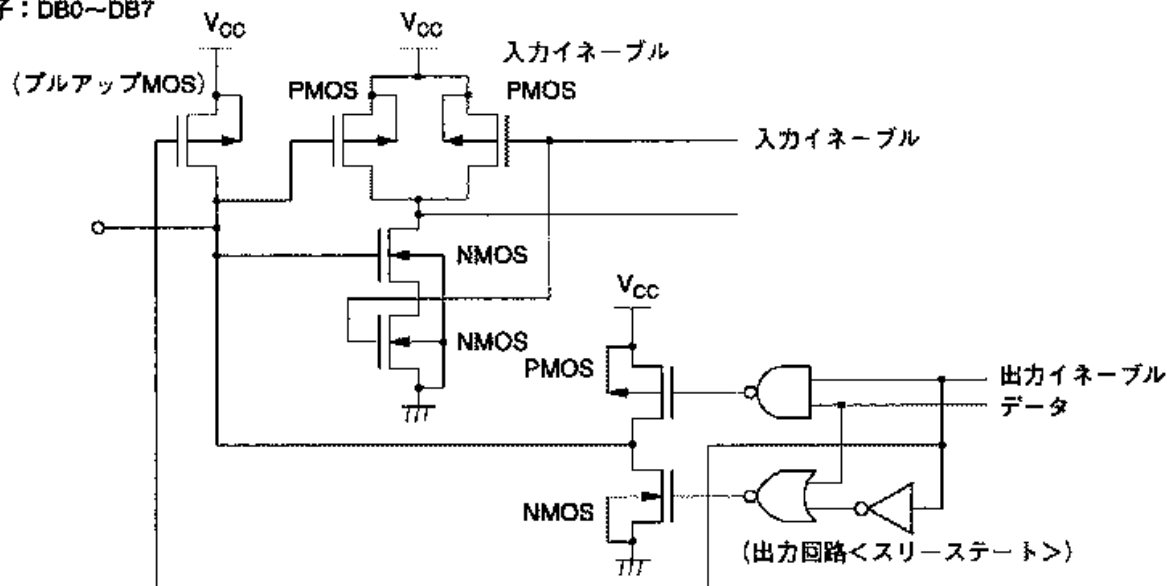
出力端子の形状

適用端子：CL1, CL2, M, D



入力端子の形状

適用端子：DB0~DB7



Reference:Page54 in HD44780.pdf

Ports for E,RS and DB0-DB7 are pulled up by FET(pull-up resister).

Sending data to characterLCD need small time.

CharacterLCD merely display much time.

If ports to E,RS and DB0-DB7 is low, current continue to flow.

So after sending data to characterLCD, ports are set to Hi.

It's h1F. [Bit4:RS bit3:DB7 bit2:DB6 bit1:DB1 bit0:DB4]

2_wire_charLCD_0.1.f

```
: lcd_com
reset_sr
dup h100 and if RS else 0 then          \ RS bit
swap 2dup                               \ ( 10/0 n 10/0 n )
\ upper 4bit
4 rshift hF and or E or                 \ Add RS-bit and Enable-bit
shift_out
lcd_enable
reset_sr

\ lower 4bit                             \ ( 10/0 n )
hF and or E or                           \ Add RS-bit and Enable-bit
shift_out
lcd_enable
reset_sr

\ Set RS and DB4-DB7 to Hi, and Set LED- to hi if sleep is on
h1F
sleep_on W@
if LED- or then
shift_out
;
```

2_wire_charLCD is fast to display.

But it is meaningless although characterLCD don't need to change rapidly.

2_wire_charLCD curcuit is also simple.