

CharacterLCD

20140503

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.

But I didn't want to buy i2c-LCD and serial-LCD.

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

3-wire LCD is a little bit much.

1-wire LCD need complex curcuit.

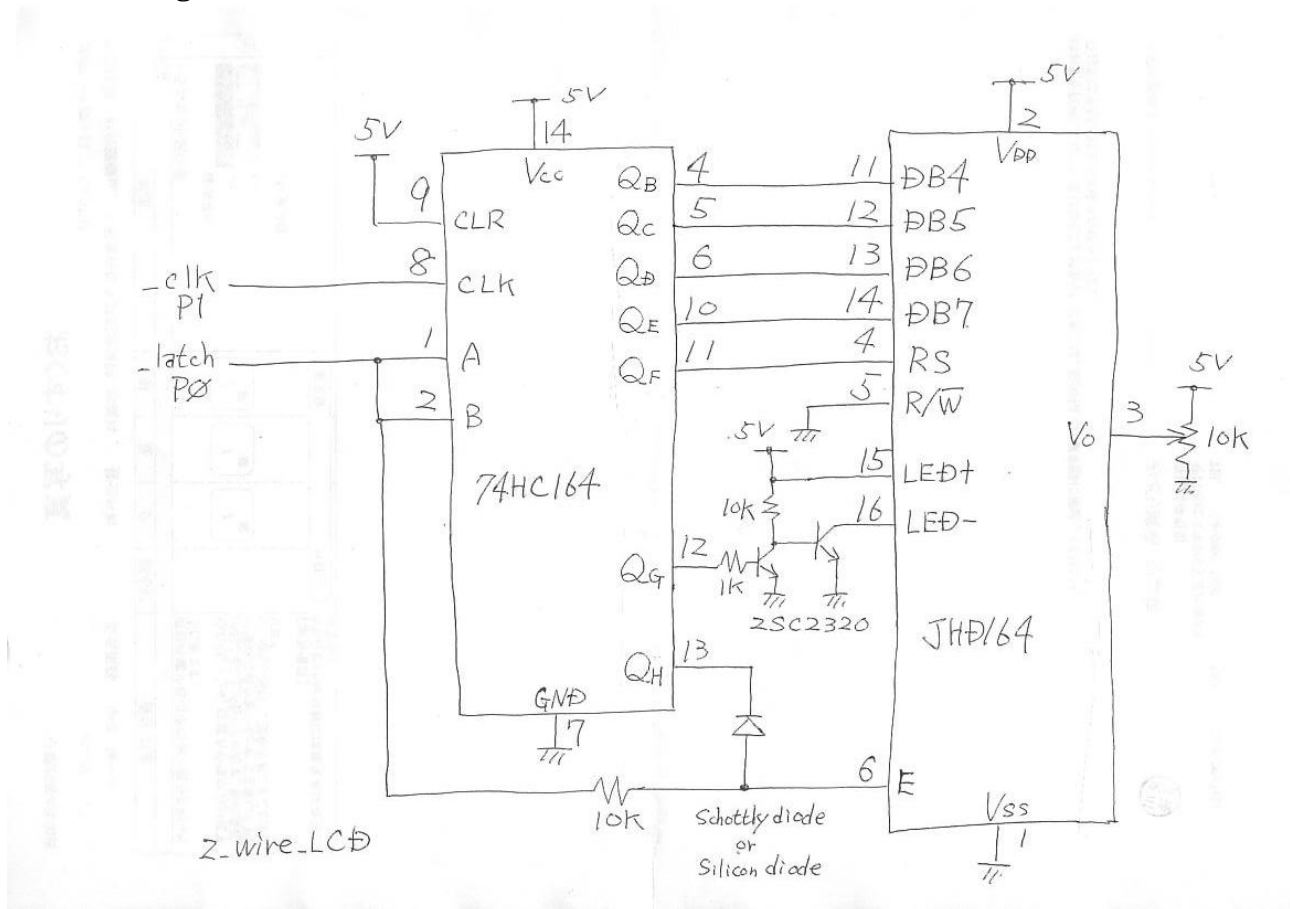
2-wire is best.

2_wire_LCD

Reference: 2_wire_LCD_0.2.f

This convert data from serial to parallel by using shift-regisre.

Curcuit diagram



Operation:

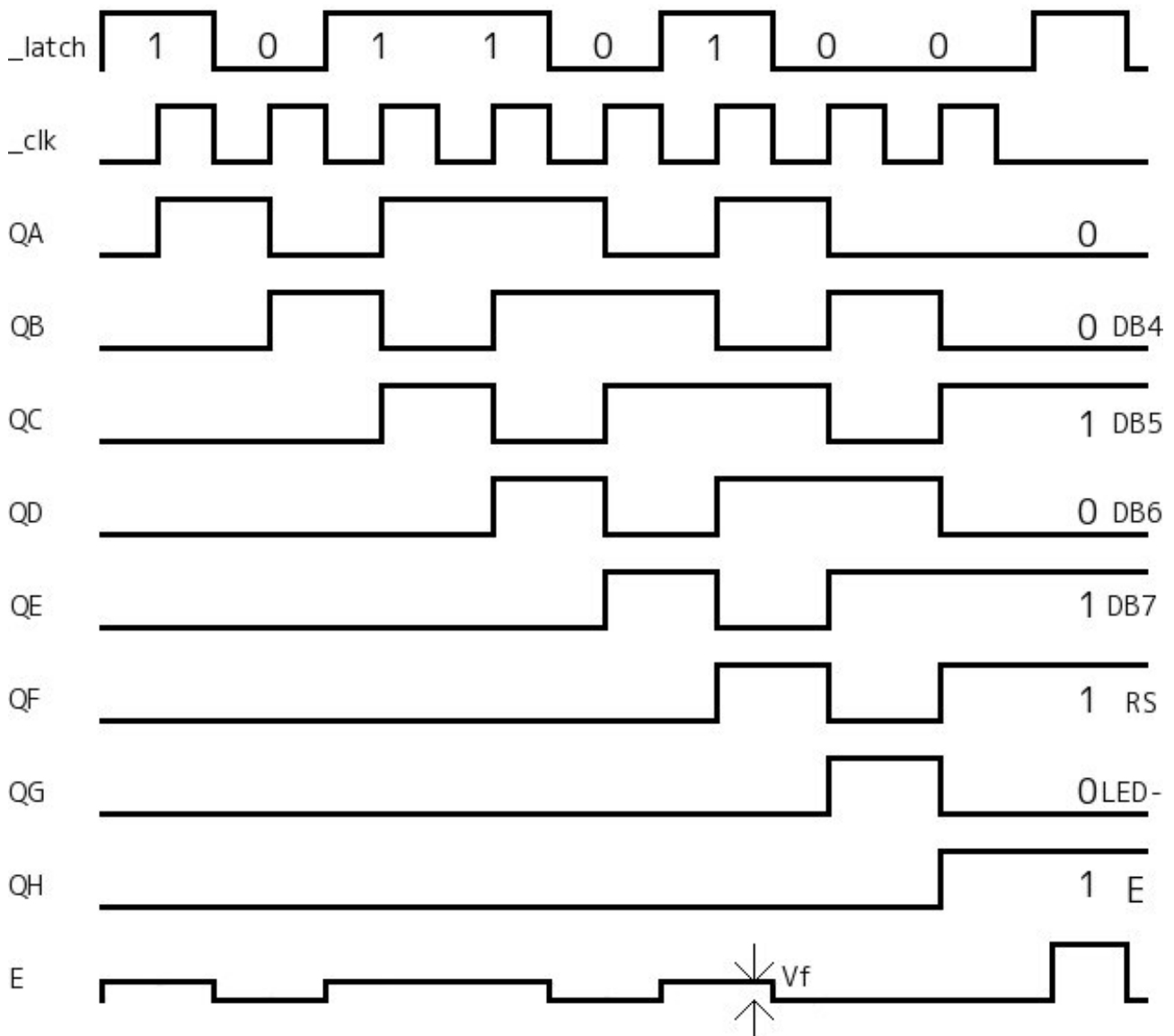
Firstly it make 7bit data by forth word.

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_4bit'.

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

Signal diagram is below;



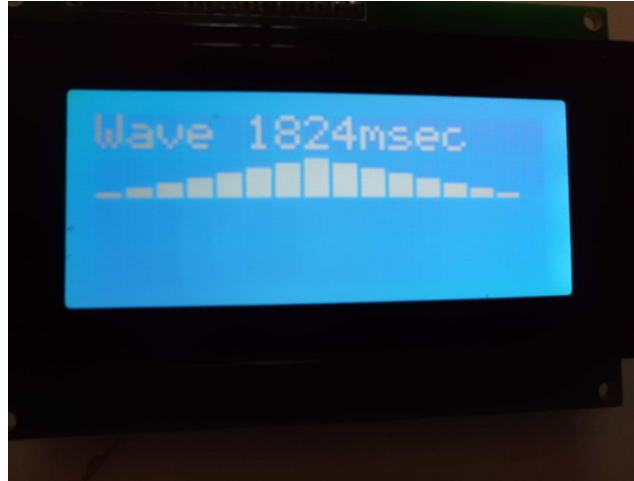
Enable signal on characterLCD is crippled by diode while QH for shift-register is low. After QH for shift-register reach to hi, data is written to characterLCD by Hi-pulse on _latch.

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_4bit'.

This is 4bit mode.
So lower4bit and upper4bit is executed.

Word'demo2' look like moving wave.
It takes 1824msec.



Code size is 1306bytes(not included demo-code).
This is simplest curcuit.

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(actually 3.3V).

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

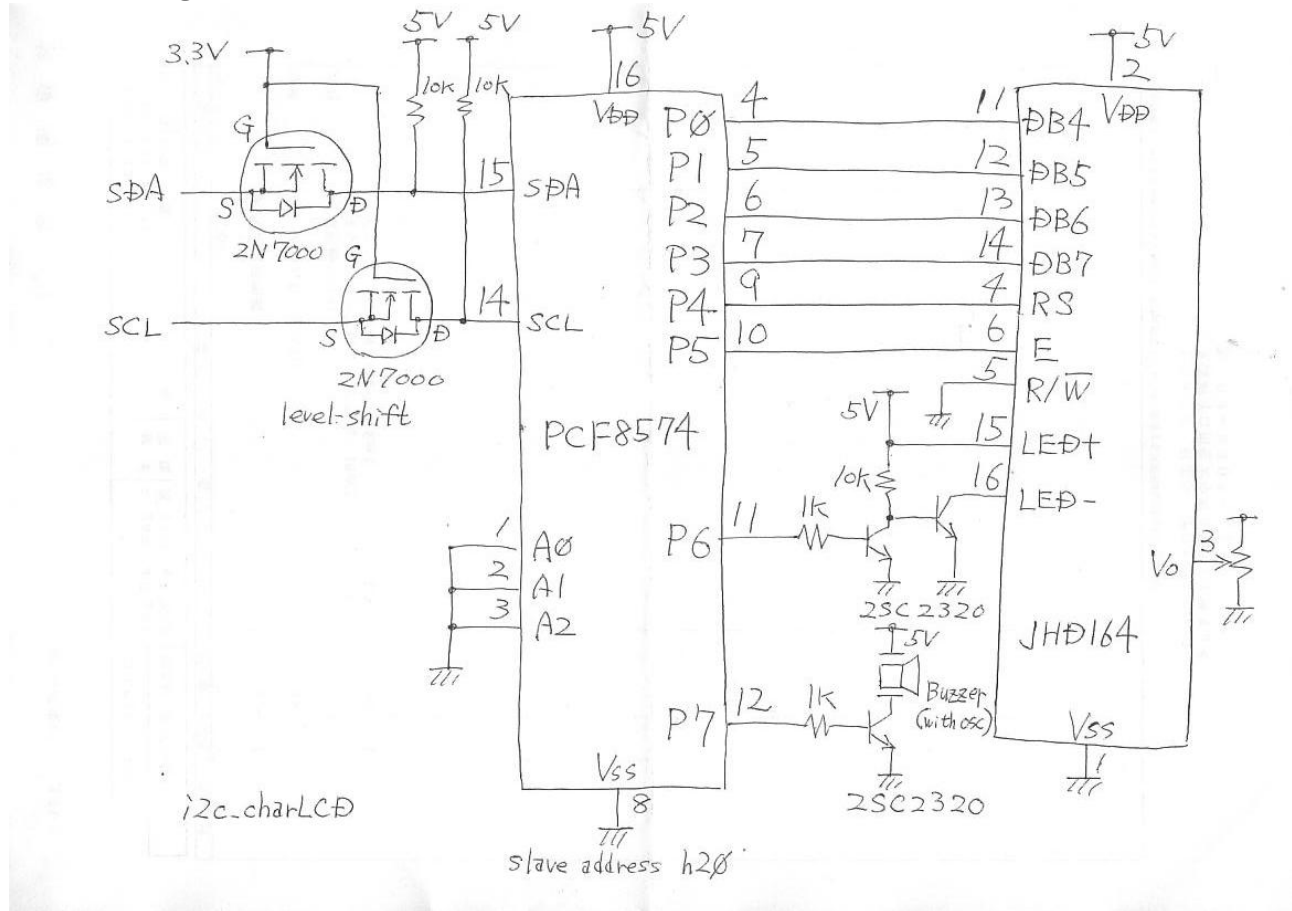
But level-shift might be needed on safety operation.

i2c_charLCD

Reference: i2c_charLCD_0.2.f

This use 8bit I/O expander(i2c-device).

Curcuit diagram



Operation:

MOSFET is level-shift between 3.3V and 5V on SDA/SCL-signal.

If using characterLCD for 3.3V, it doesn't need level-shift.

This is also 4bit mode.

By i2c, 8bit-data is written 2-times on characterLCD.

Speed of i2c is 400kHz.

Buzzer(with osc) is added.

i2c-word need to load because not using i2c-word of PropForth5.5.

Word'demo2' look like moving wave.
It takes 16205msec.



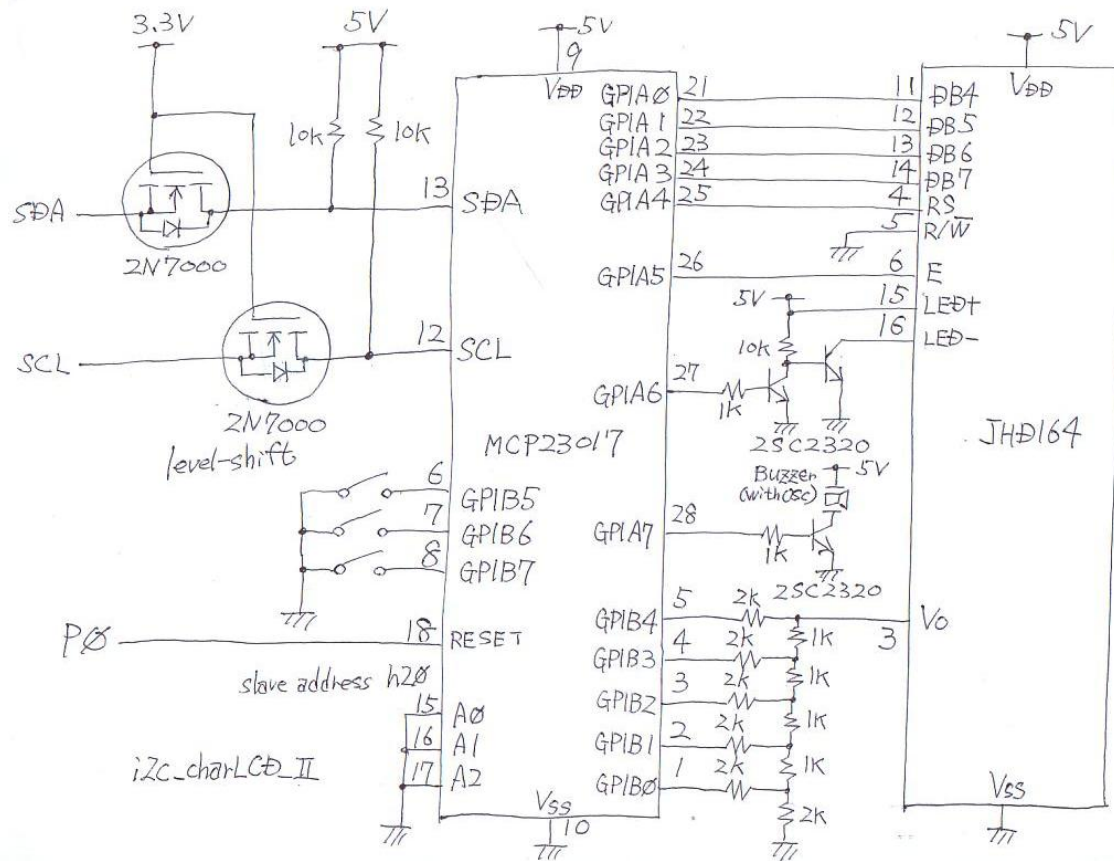
Display for string is no problem, although demo2 is slower than 2_wire_LCD.
Code size is 1146bytes(not included i2c-word and demo-code).
Code size for i2c-words is 484bytes.

i2c_charLCD_II

Reference: i2c_wire_LCD_II_0.2.f

This use 16bit I/O expander(i2c-device).

Curcuit diagram



Operation:

MOSFET is level-shift between 3.3V and 5V on SDA/SCL-signal.

If using characterLCD for 3.3V, it doesn't need level-shift.

Firstly, MCP23017 must reset to add hi-pulse to reset-pin.

So this use 3-wires(SDA/SCL,P0)

This is also 4bit mode.

By i2c, 16bit-data is written 2-times on characterLCD.

Speed of i2c is 400kHz.

Buzzer(with osc) is added.

R-2R-laddaer for contrast is added.

i2c-word need to load because not using i2c-word of PropForth5.5.

Word'demo2' look like moving wave.
It takes 20954msec.



Display for string is no problem, although demo2 is slower than 2_wire_LCD.
Code size is 1462bytes(not included i2c-word and demo-code).
Code size for i2c-words is 620bytes.

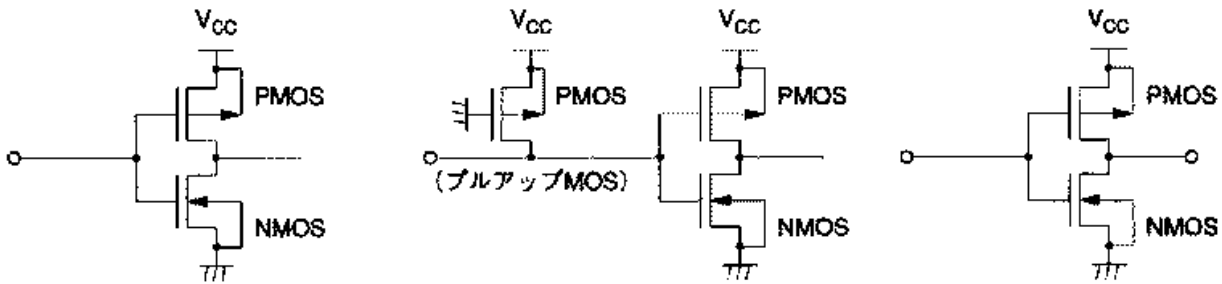
Port circuit diagram of HD44780 is below;

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, \overline{RW} (プルアップMOS付き)

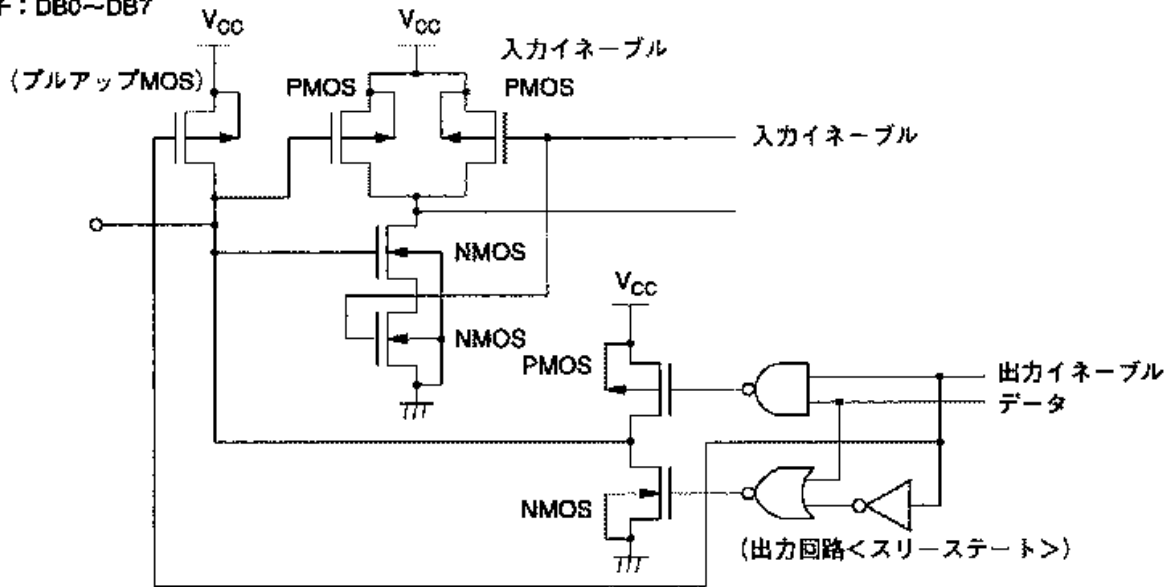
出力端子の形状

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_LCD

```
: 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
```

```
;
```

i2c_charLCD

```
: lcd_com
```

```
dup h100 and if RS else 0 then
```

```
\ RS bit
```

```
swap 2dup
```

```
\ ( h10/0 n h10/0 n )
```

```
\ upper 4bit
```

```
4 rshift hF and or
```

```
\ ( h10/0 n RS+data[upper4bit] ) Add RS-bit to data
```

```
sleep_on W@
```

```
if LED- or then
```

```
GPIO
```

```
1 delms
```

```
\ lower 4bit
```

```
\ ( 10/0 n )
```

```
hF and or
```

```
\ ( RS+data[upper4bit] ) Add RS-bit to data
```

```
GPIO
```

```
1 delms
```

```
\ Set RS and DB4-DB7 to Hi, and Set LED- to hi if sleep is on
```

```
h1F
```

```
sleep_on W@
```

```
if LED- or then
```

```
out_PCF8574
```

```
;
```

i2c_charLCD_II

```
: lcd_com
```

```
dup h100 and if RS else 0 then
```

```
\ RS bit
```

```
swap 2dup
```

```
\ ( h10/0 n h10/0 n )
```

```
\ upper 4bit
```

```
4 rshift hF and or
```

```
\ ( h10/0 n RS+data[upper4bit] ) Add RS-bit to data
```

```
shift_cont or
```

```
\ Add contrast
```

```
2GPIO
```

```
1 delms
```

```
\ lower 4bit
```

```
\ ( 10/0 n )
```

```
hF and or
```

```
\ ( RS+data[upper4bit] ) Add RS-bit to data
```

```
shift_cont or
```

```
\ Add contrast
```

```
2GPIO
```

```
1 delms
```

```
\ Set RS and DB4-DB7 to Hi, and Set LED- to hi if sleep is on
```

```
shift_cont h1F or
```

```
sleep_on W@
```

```
if LED- or then
```

```
GPIO
```

```
;
```

2_wire_LCD is fast to display.

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

But 74HC164 is cheaper than I/O Expander chips.

And code-size of 2_wire_LCD is smaller than i2c_charLCD and i2c_charLCD_II.

2_wire_LCD circuit is also simple.

I recommend 2_wire_LCD in case of using PropForth5.5.