Following Schedule (Updated)

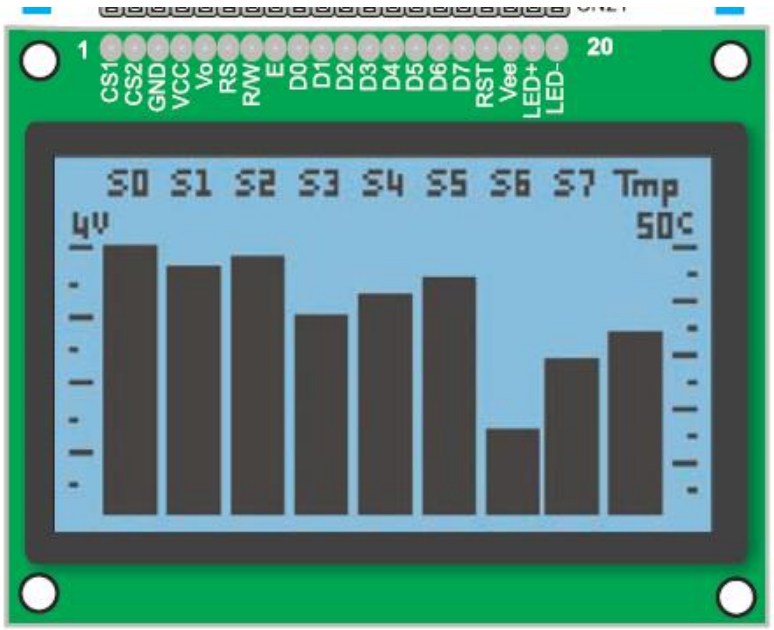
- 12/23:
  - Exercise for Lab 5
  - Announcement of Lab 6
- 12/30
  - 畢業專題展
- 1/6
  - Exercise for Lab 6
  - Announcement of Final Project
- 1/13
  - Exercise for Lab 6
  - Demo of Final Project
- 1/20
  - Demo of Final Project

Lab 06

Graphics-LCD Display

Your Task

- Draw a still image on the graphics LCD
- Bonus: show some animation

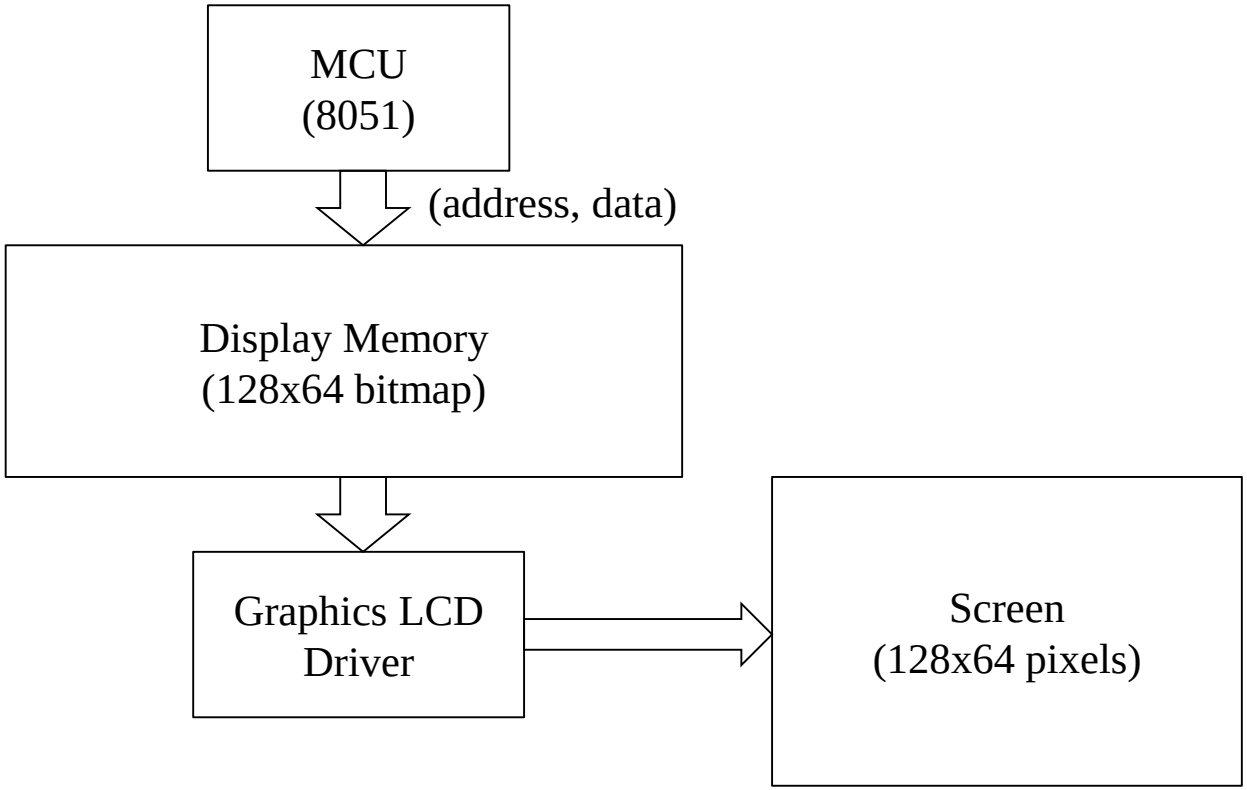


General Scheme to Control Graphics LCD

Fill the image bitmap into the display memory

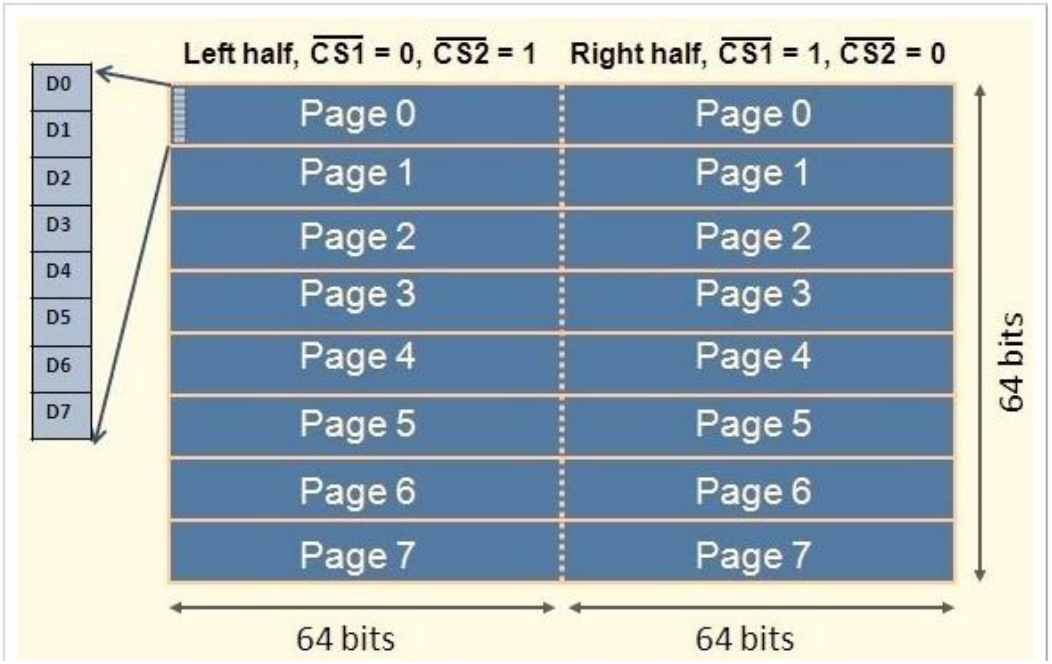
The control scheme

- Fill-in the bitmap into the display memory



Detailed look of the control scheme

- Fill the 0-1 bitmap into two-dimensional array
- But each time we fill 8-bit for 8 pixels on the x-axis



Signal Interface to Fill the Image Memory

And the timing

Signal Interface on Experiment Board

- Through P2 and P4

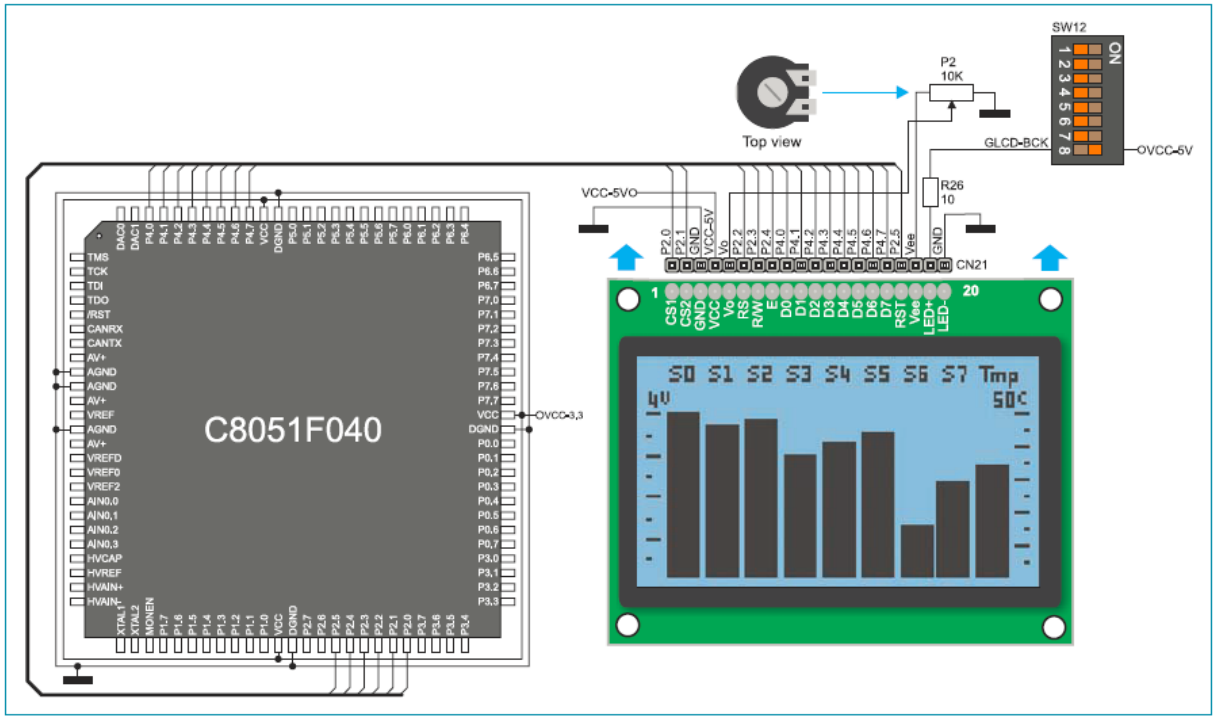


Figure 19-3: GLCD display connection schematic

Signal interface to the LCD

- RS: register select
  - 0: command
  - 1: data
- RW: read/write select
  - 0: write
  - 1: read
- E: latch enable
- DB [7:0]
  - the 8-bit data/command
- CS1, CS2: select left/right half
- #include "C8051F040.h"
- #include "glcd.h"

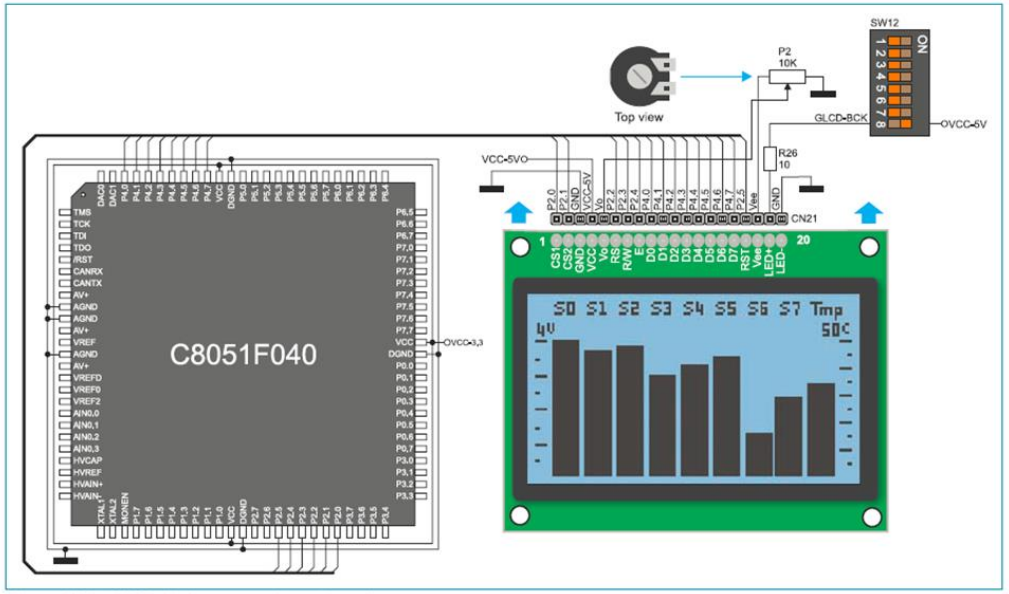


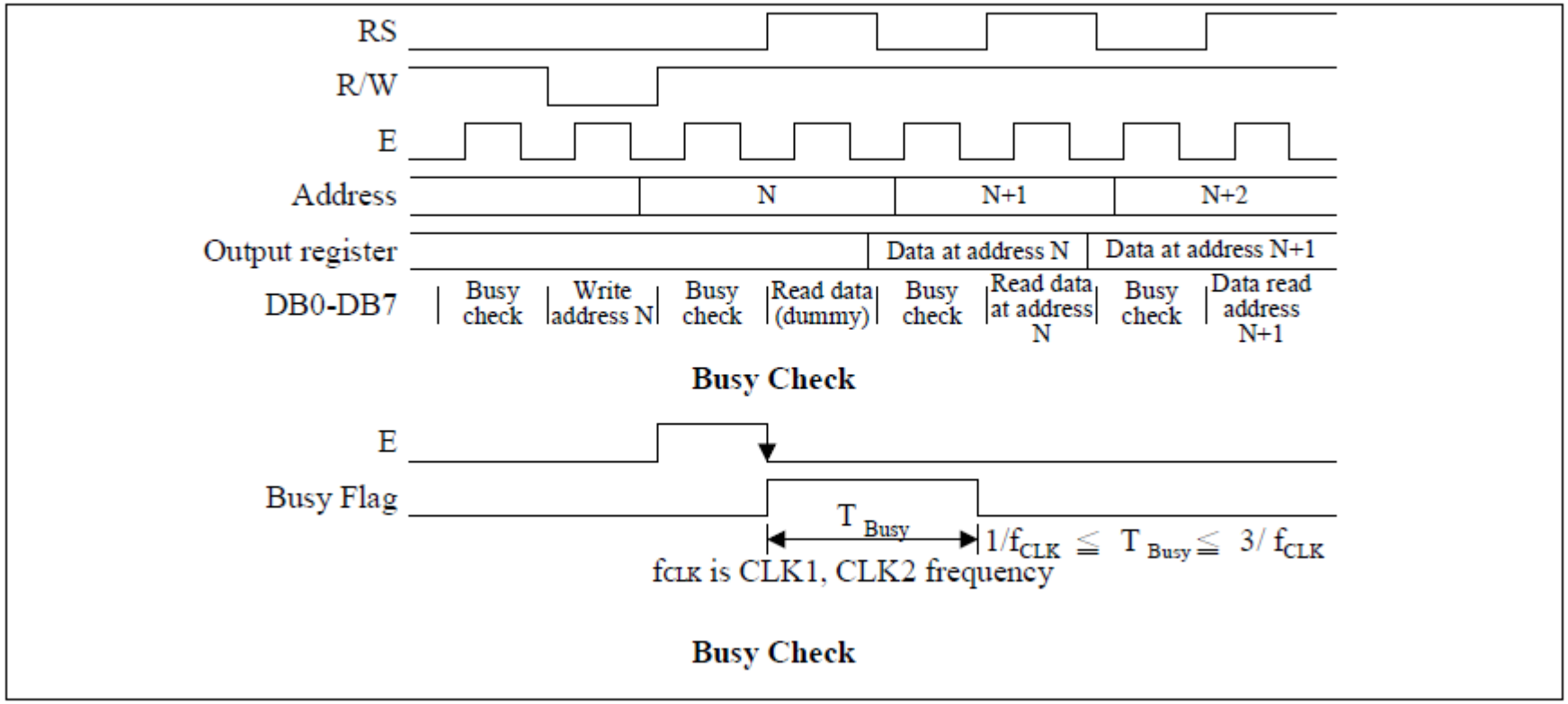
Figure 19-3: GLCD display connection schematic

List of commands

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Function
Display on/off	L	L	L	L	H	H	H	H	H	L/H	Controls the display on or off. Internal status and display RAM data is not affected. L:OFF, H:ON
Set address (Y address)	L	L	L	H	Y address (0-63)						Sets the Y address in the Y address counter.
Set page (X address)	L	L	H	L	H	H	H	Page (0-7)			Sets the X address at the X address register.
Display Start line (Z address)	L	L	H	H	Display start line (0-63)						Indicates the display data RAM displayed at the top of the screen.
Status read	L	H	Busy	L	On/ Off	Reset	L	L	L	L	Read status. BUSY L: Ready H: In operation ON/OFF L: Display ON H: Display OFF RESET L: Normal H: Reset
Write display data	H	L	Write data								Writes data (DB0: 7) into display data RAM. After writing instruction, Y address is increased by 1 automatically.
Read display data	H	H	Read data								Reads data (DB0: 7) from display data RAM to the data bus.

Timing diagram to send command/data

- Notice the latch trigger on E
- Has to check for busy



Example to Set Start Line

```
void
Set_DisplayStartLine (char z)
{
        char P2_cword, P4_cword;

        ///prepare control words
        P2_cword = P2_CWORD_TEMPLATE;
        P2_cword = P2_cword & (~P2_RS);          //clear RS bit
        P2_cword = P2_cword & (~P2_RW);          //clear RW bit
        P4_cword = P4_Set_Zaddr_TMPL;
        P4_cword = P4_cword | (z & 0x3f);

        ///flush out control signals
        while (GLCD_IsBusy());
        GLCD_Write (P2_cword, P4_cword);
}//end of function Set_DisplayStartLine
```

Example to Check Busy

```
int
GLCD_IsBusy ()
{
        char status;

        status = GLCD_ReadStatus ();
        if (status&P4_Busy)
                return 1;
        else
                return 0;
}//end of function GLCD_IsBusy
```

Example to Read LCD Status

```
char
GLCD_ReadStatus ()
{
        char P2_cword;
        char status;

        P2_cword = P2_CWORD_TEMPLATE;
        P2_cword = P2_cword & (~P2_RS);
        P2_cword = P2_cword | (P2_RW);
        status = GLCD_Read (P2_cword);

        return status;
}//end of function GLCD_ReadStatus
```

Example to Do GLCD Read (1/2)

```
char
GLCD_Read (char P2_cword)
{
        char status;
        char P2_cword_rep;

        P2_cword_rep = P2_cword;
        set_GLCD_ReadMode ();
        GLCD_delay ();

        P2_cword_rep = P2_cword_rep & (~P2_E);          //clear E bit
        P2 = P2_cword_rep;
        GLCD_delay ();
```

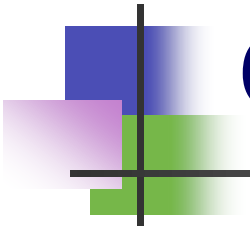
Example to Do GLCD Read (2/2)

```
        P2_cword_rep = P2_cword_rep | P2_E;                  //set E bit
        P2 = P2_cword_rep;
        GLCD_delay ();

        status = P4;

        P2_cword_rep = P2_cword_rep & (~P2_E);               //clear E bit
        P2 = P2_cword_rep;
        GLCD_delay ();

        return status;
}//end of function GLCD_Read
```

Pseudo Code of the General Control Sequence

- Convert each part to a sequence of accesses to ports P2 and P4

```
do {
    read status;
} while (status==busy);

Set address x;
Set address y;
Write bitmap A;
```



Grading

- Basic: (80%)
  - “Draw” some still image
- Optional: (animation)
  - Move the pattern horizontally (+10%)
  - Move the pattern vertically (+20%)

- But this time you have to start from scratch

Pre-Lab Questions

- Q1: how to enable port P4 for input and output?
- Q2: write a C function to set 8-bit bitmap on address (x, y)
  - Draw (char x, char y, char bitmap)
- Q3: find a tool to generate the bitmap of your image
  - Please google!

Lab06 Study Report

- File name: Bxxxxxxx-MCE-Lab6-Study
- File type: PDF only
- The requirements of report
  - Summarize the content of this slide set
  - Provide your plan for this lab exercise
  - No more than one A4 page
  - Grading: 80 ± 15
- Deadline: 2021/12/22 23:00 (不收遲交)
- Upload to e-learning system

Lab06 Lab Exercise Report

- File name: Bxxxxxxx-MCE-Lab6-Result
- File type: PDF only
- The requirements of report
  - Summarize the problems and results you have in this exercise
  - Some screen shots or some code explanation can be provided
  - No more than two A4 pages
  - Grading: 80 ± 15
- Deadline: 2021/12/29 23:00 (不收遲交)
- Upload to e-learning system