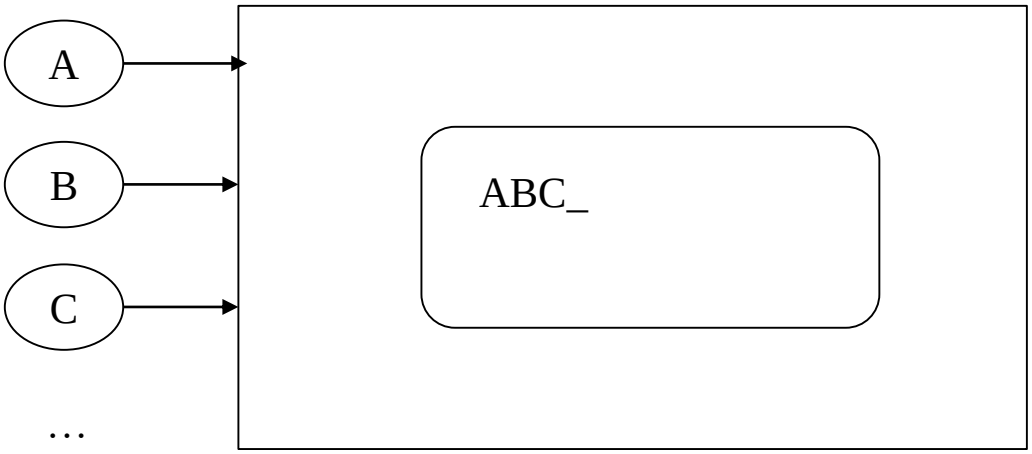
Lab 05

LCD Display

Your Task

- Display characters according to the button pressed

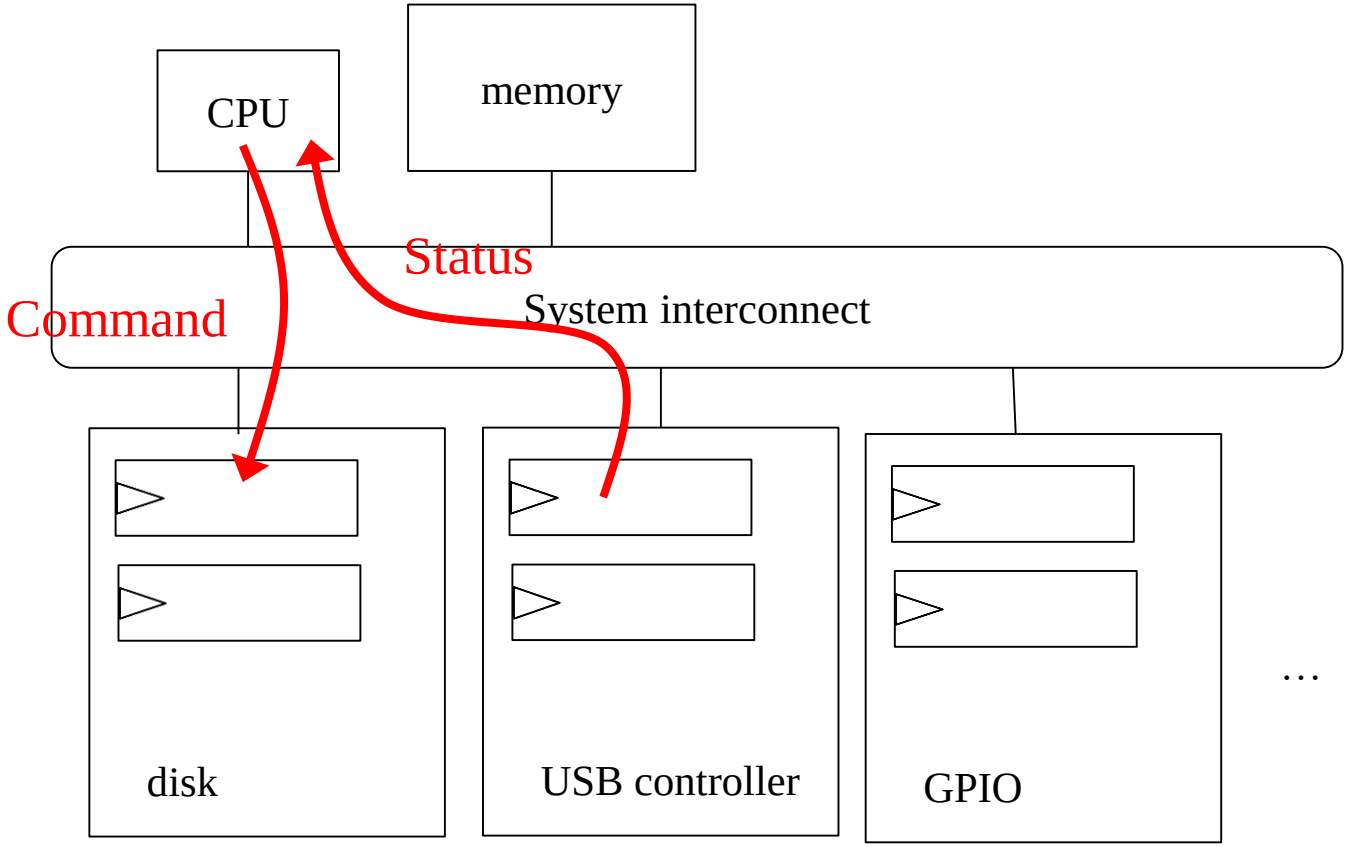


General Concepts: Controlling I/O Devices

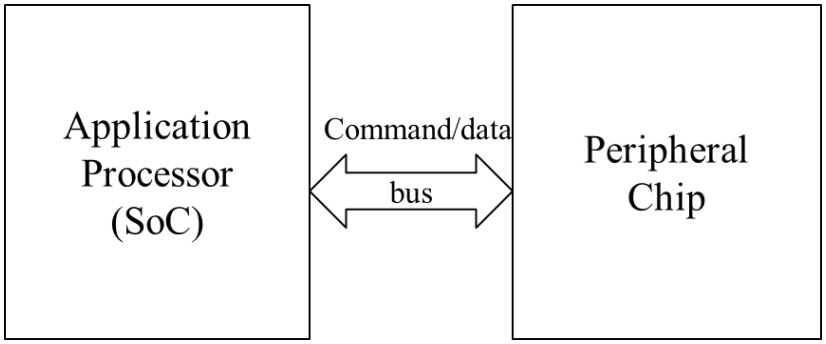
On-chip vs. off-chip

Control On-Chip Peripherals

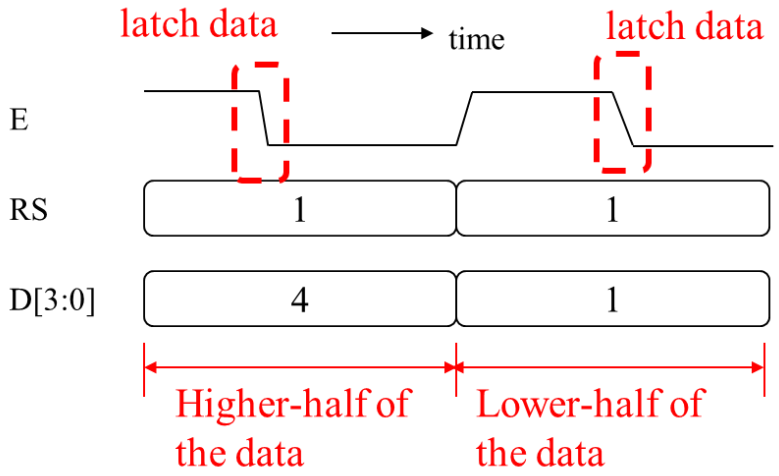
- access control registers with memory-mapped I/O

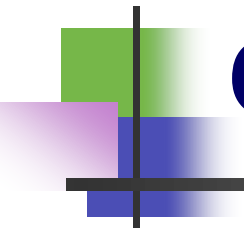


Control Off-Chip Peripherals



- Transfer command/data through off-chip buses
- Following some protocol (waveform)



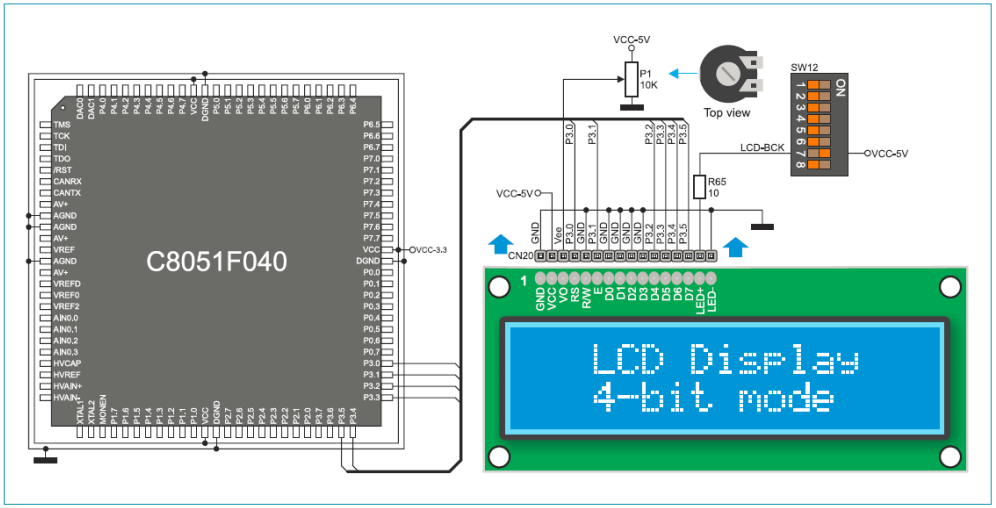


Control the LCD Display

From the hardware perspective

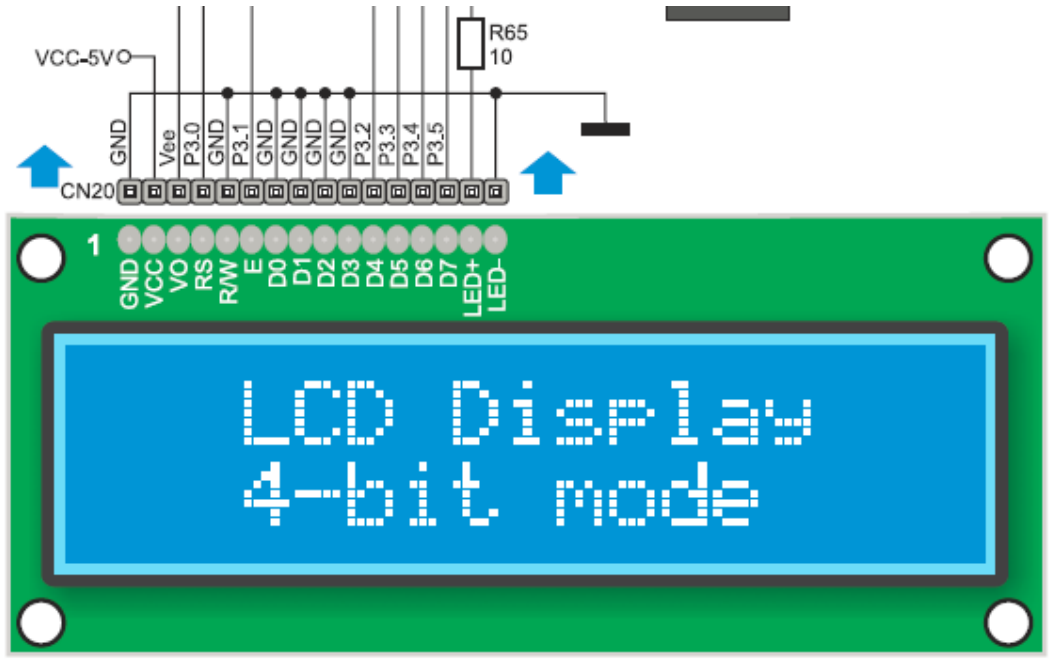
Signal Interface to the LCD

- The application processor sends commands and data through I/O port
- A command:
  - Moving cursor
  - Clear screen
  - …etc
- Data: the ASCII code of the character to display



Signal interface to the LCD

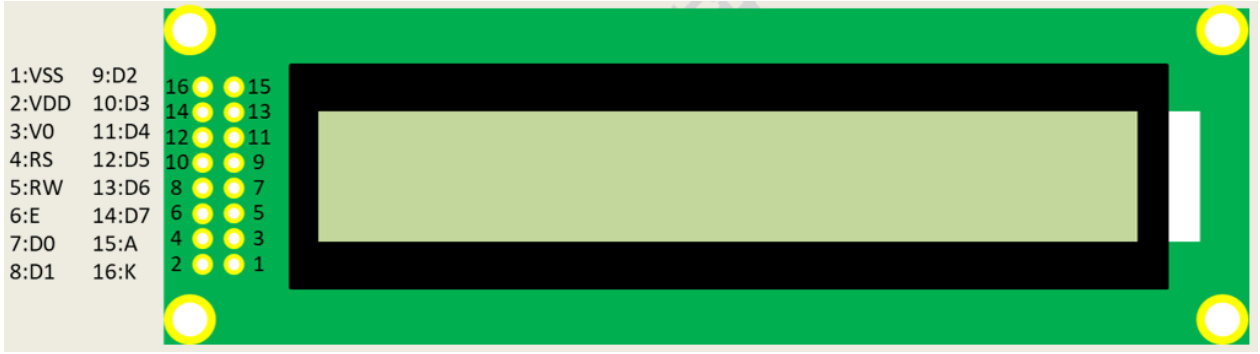
- RS: register select
  - 0: command
  - 1: data
- E: latch enable
  - the LCD latches the command/data at negative edge (1→0)
- D [7:0]
  - the 8-bit data/command
  - configured 4-bit mode
  - send higher portion first



Two Common LCD Devices

- LCD1602

- LCD2002

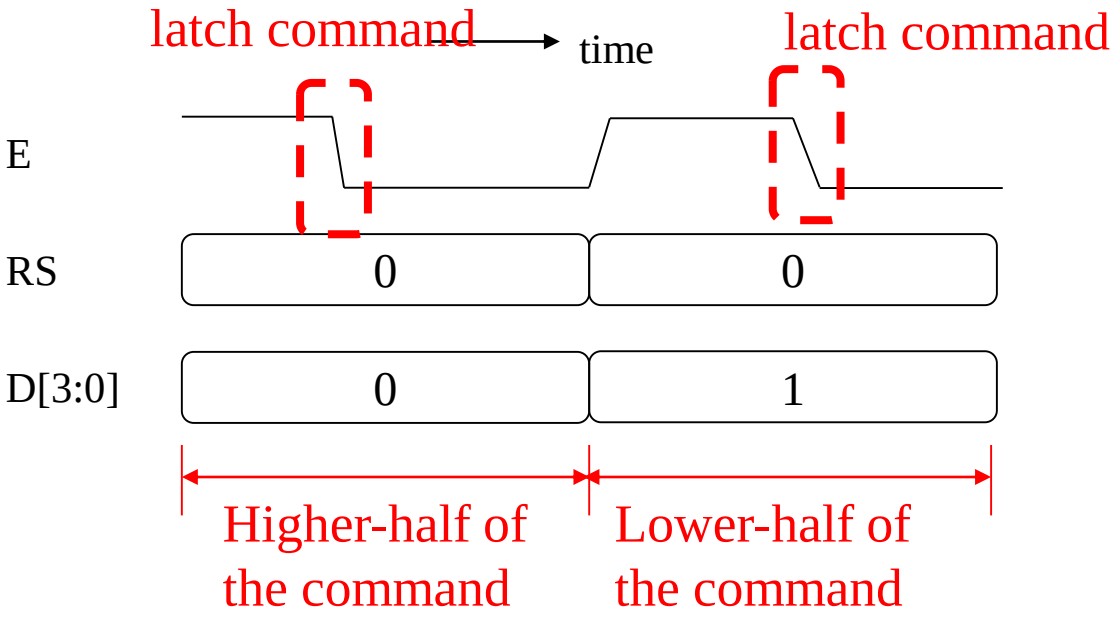


List of commands

Command	Code
Clear Display, Cursor to Home	0x0001
Cursor to Home	0x0002
Entry Mode:	
Cursor Decrement, Shift off	0x0004
Cursor Decrement, Shift on	0x0005
Cursor Increment, Shift off	0x0006
Cursor Increment, Shift on	0x0007
Display Control:	
Display, Cursor, and Cursor Blink off	0x0008
Display on, Cursor and Cursor Blink off	0x000C
Display and Cursor on, Cursor Blink off	0x000E
Display, Cursor, and Cursor Blink on	0x000F
Cursor / Display Shift: (nondestructive move)	
Cursor shift left	0x0010
Cursor shift right	0x0014
Display shift left	0x0018
Display shift right	0x001C
Display Function (2 rows for 4-bit data; big)	0x002C
Display Function (2 rows for 4-bit data; small))	0x0028
Display Function (1 row for 4-bit data; big)	0x0024
Display Function (1 row for 4-bit data; small)	0x0020
Display Function (2 rows for 8-bit data; big)	0x003C
Display Function (2 rows for 8-bit data; small)	0x0038
Display Function (1 row for 8-bit data; big)	0x0034
Display Function (1 row for 8-bit data; small)	0x0030
Move cursor to beginning of second row	0x00C0

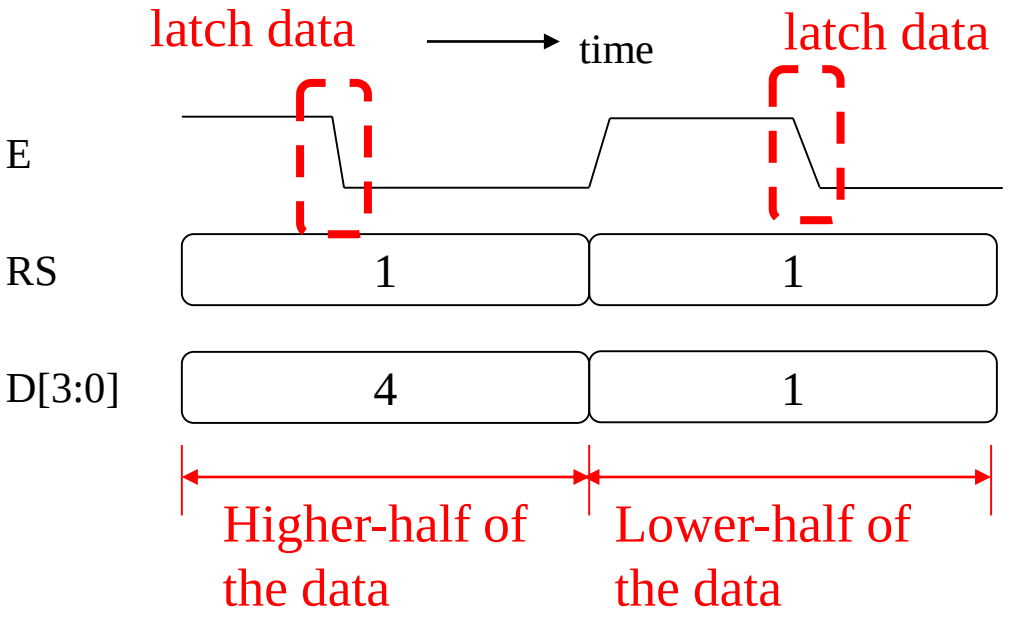
Timing diagram to send command/data

- Example: to send command 0x01
- clear screen, cursor home



Timing diagram to send command/data

- Example: to send data ‘A’=0x41
- The LCD prints ‘A’ at the cursor position



How to program the LCD

Demo: LCD_Hello

Initialize the LCD

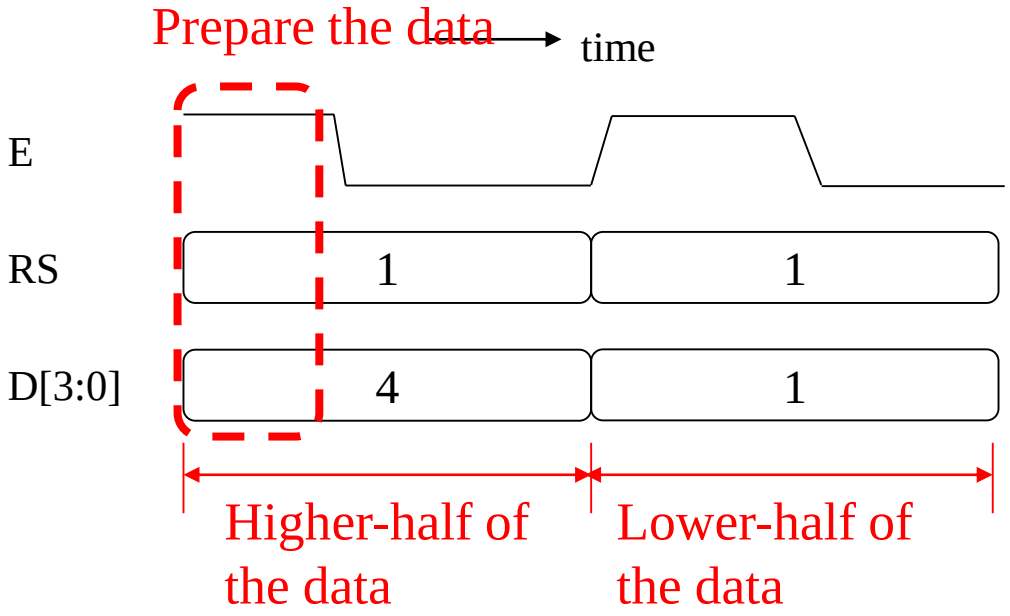
```
void
LCD_Init ()
{
    LCD_SendCommand (0x01);     //clear display, cursor to home
    LCD_SendCommand (0x28);     //Display function: 2 rows for 4-bit data, small
    LCD_SendCommand (0x0e);     //display and curson ON, curson blink off
    //LCD_SendCommand (0x10);       //cursor shift left
    //LCD_SendCommand (0x06);       //cursor increment, shift off
}
```

Program to send command/data

- Example: to send data ‘A’=0x41
- The LCD prints ‘A’ at the cursor position

```
void
LCD_SendData (char dat)
{
    LCD_Status_SetRS ();

    ///send the higher half
    LCD_Status_SetWord ((dat>>4) & 0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();

    ///send the lower half
    LCD_Status_SetWord (dat&0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();
}
```

Prepare the status word

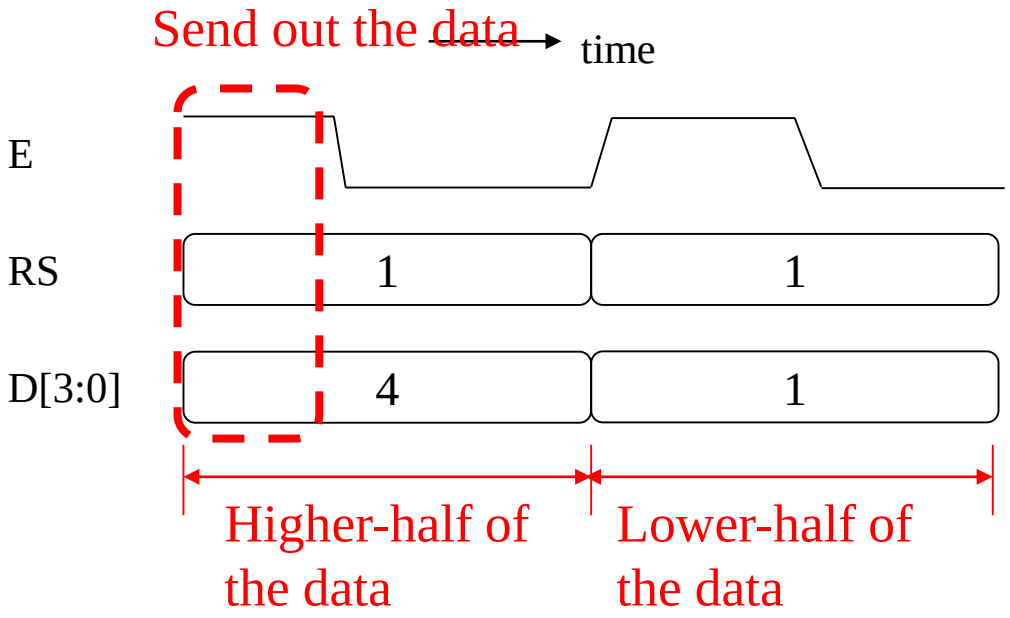


Program to send command/data

- Example: to send data ‘A’=0x41
- The LCD prints ‘A’ at the cursor position

```
void
LCD_SendData (char dat)
{
    LCD_Status_SetRS ();

    ///send the higher half
    LCD_Status_SetWord ((dat>>4) & 0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();

    ///send the lower half
    LCD_Status_SetWord (dat&0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();
}
```

Send out the status word

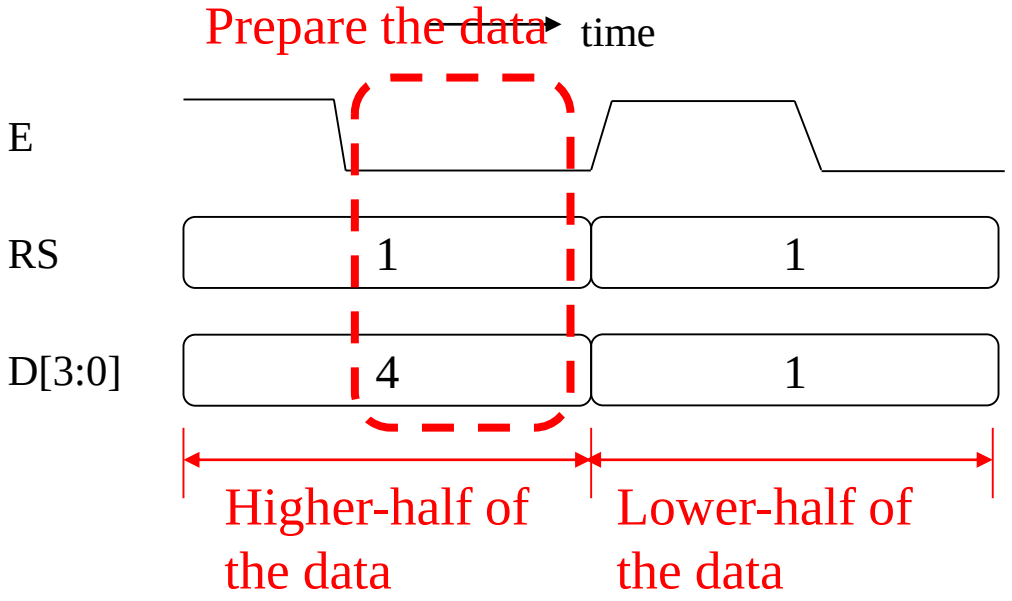


Program to send command/data

- Example: to send data ‘A’=0x41
- The LCD prints ‘A’ at the cursor position

```
void
LCD_SendData (char dat)
{
    LCD_Status_SetRS ();

    ///send the higher half
    LCD_Status_SetWord ((dat>>4) & 0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();

    ///send the lower half
    LCD_Status_SetWord (dat&0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();
}
```

Prepare the status word

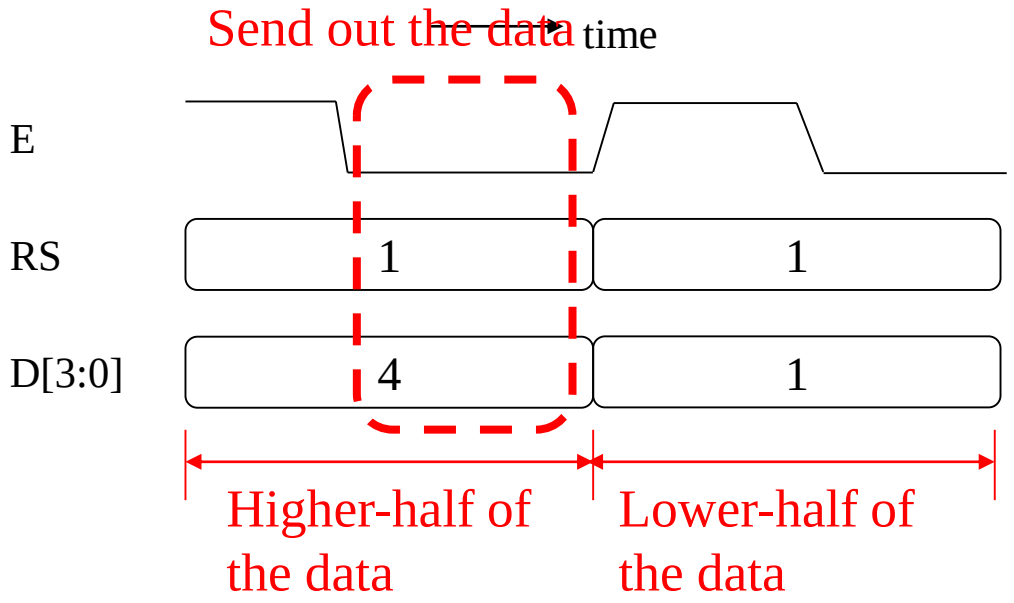


Program to send command/data

- Example: to send data ‘A’=0x41
- The LCD prints ‘A’ at the cursor position

```
void
LCD_SendData (char dat)
{
    LCD_Status_SetRS ();

    ///send the higher half
    LCD_Status_SetWord ((dat>>4) & 0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();

    ///send the lower half
    LCD_Status_SetWord (dat&0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();
}
```

Send out the status word

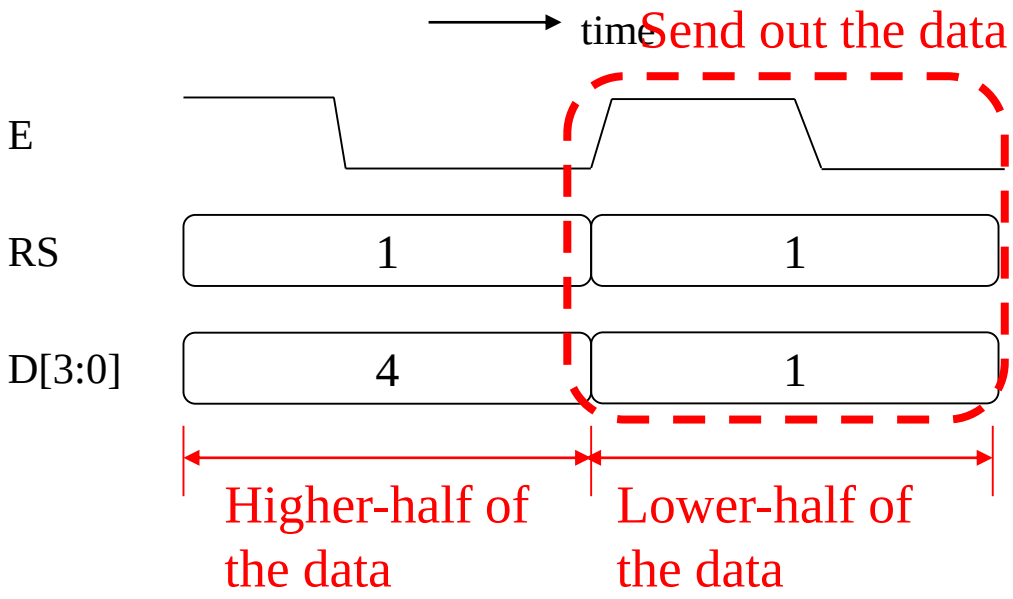


Program to send command/data

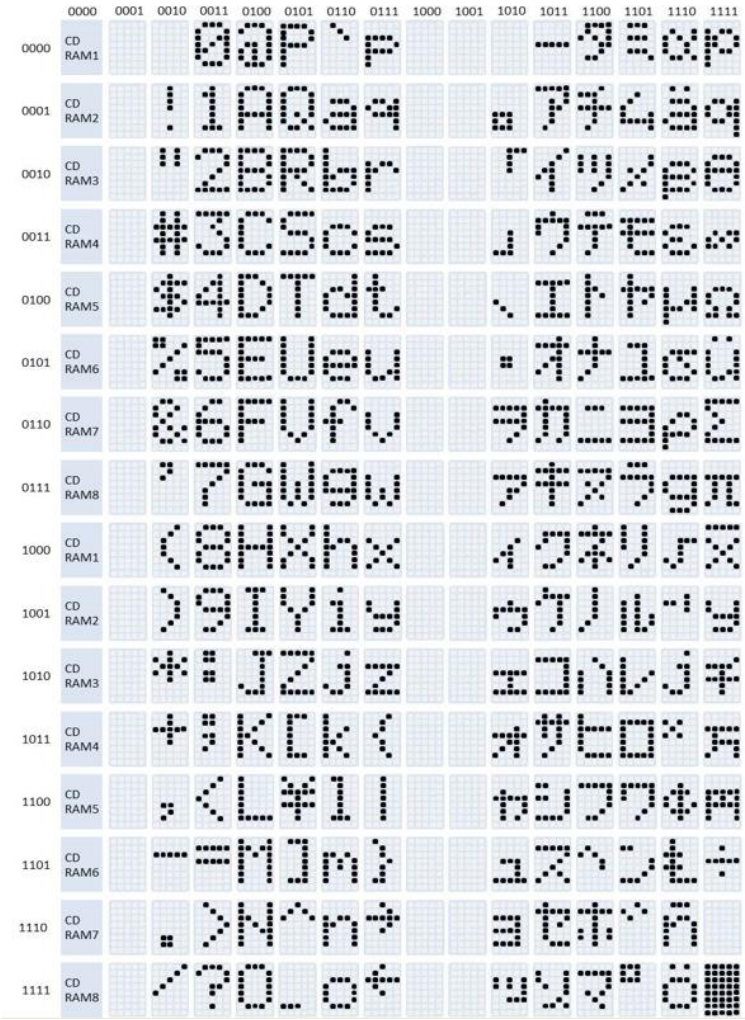
- Example: to send data ‘A’=0x41
- The LCD prints ‘A’ at the cursor position

```
void
LCD_SendData (char dat)
{
    LCD_Status_SetRS ();

    ///send the higher half
    LCD_Status_SetWord ((dat>>4) & 0x0f);
    LCD_Status_SetEnable ();
    P3 = LCD_status;
    LCD_Delay ();
    LCD_Status_ClearEnable ();
    P3 = LCD_status;
    LCD_Delay ();

   ///send the lower half
   LCD_Status_SetWord (dat&0x0f);
   LCD_Status_SetEnable ();
   P3 = LCD_status;
   LCD_Delay ();
   LCD_Status_ClearEnable ();
   P3 = LCD_status;
   LCD_Delay ();
}
```

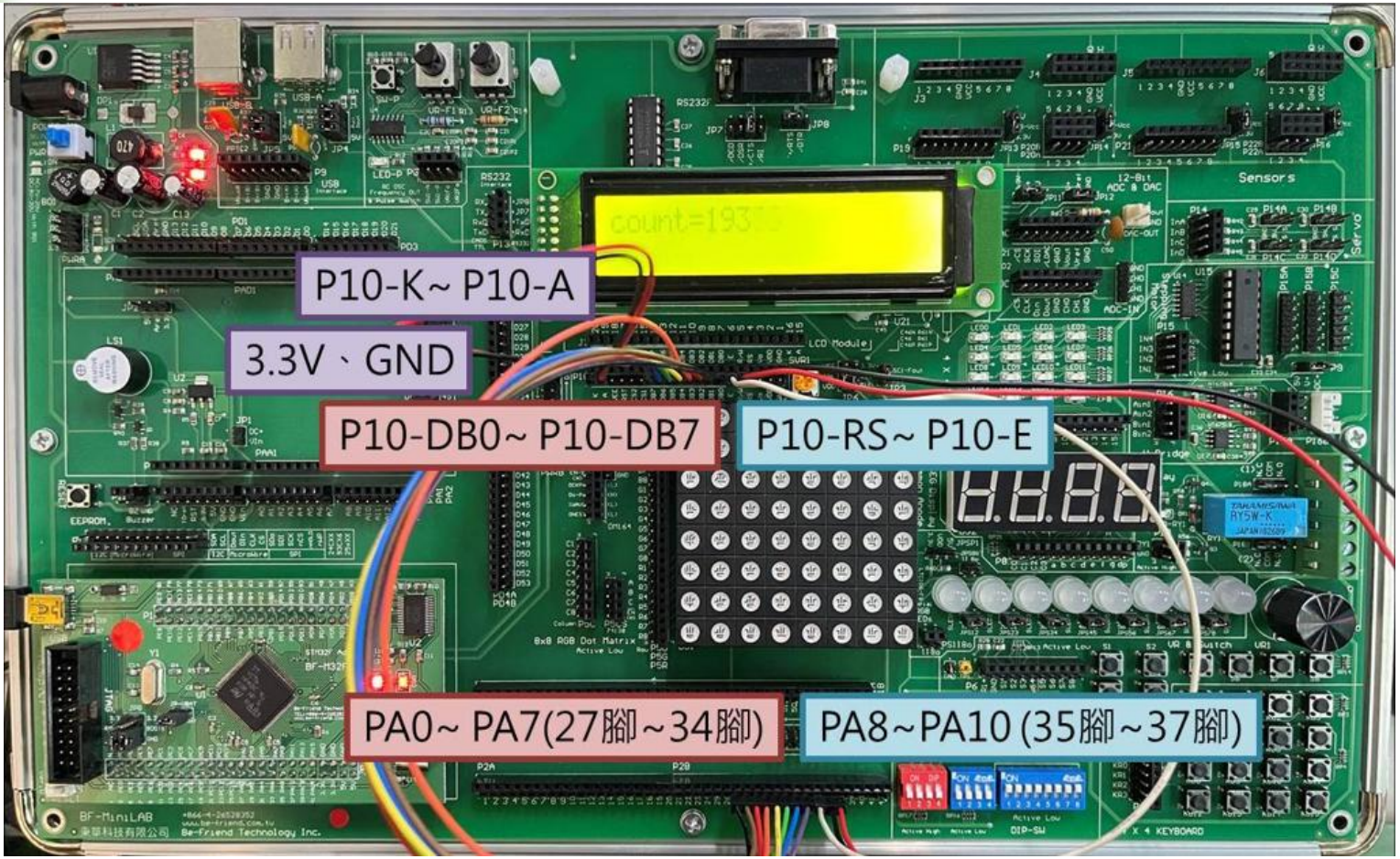
Send signal for lower-half

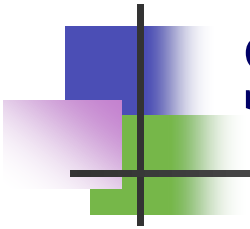


Mapping Table



Hardware Setting





Software Samples

- Please refer to the sample programs provided by the IDE tools

Demo Requirements

- Basic:
  - Display the character pressed at the cursor position
- Bonus 1: (5%)
  - Implement the `new-line' key
  - Change to the next line if new-line pressed at Line 1
  - Scroll the screen if new-line pressed at Line 2
- Bonus 2: (5%)
  - Implement the arrow keys (up, down, left, right)
  - Move the cursor by the arrow key
  - Insert character at the cursor position

Lab05 Study Report

- File name: Bxxxxxxx-MCE-Lab5-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: 2025/11/19 23:00 (不收遲交)
- Upload to e-learning system

Lab05 Lab Exercise Report

- File name: Bxxxxxxx-MCE-Lab5-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: 2025/11/26 23:00 (不收遲交)
- Upload to e-learning system