Exercise 6 Prep

LCD Displays

Equipment/Parts Needed

- Microstick II kit
- Breadboard
- Potentiometer
- LCD Display
- Wires
  - red (Vdd)
  - black/blue (GND)
  - other (data/control)

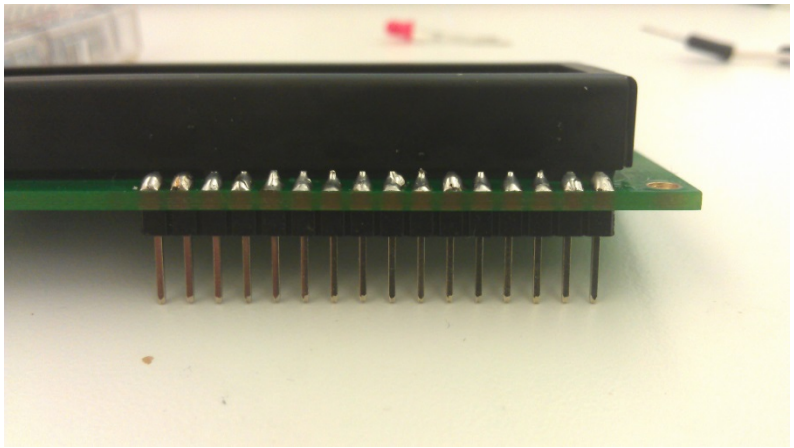
Exercise Overview

- Solder headers (for some)
- Follow exercise to

1. Connect LCD to PIC24
2. Program PIC24 to print welcome message.
3. Modify program to make welcome message scroll.

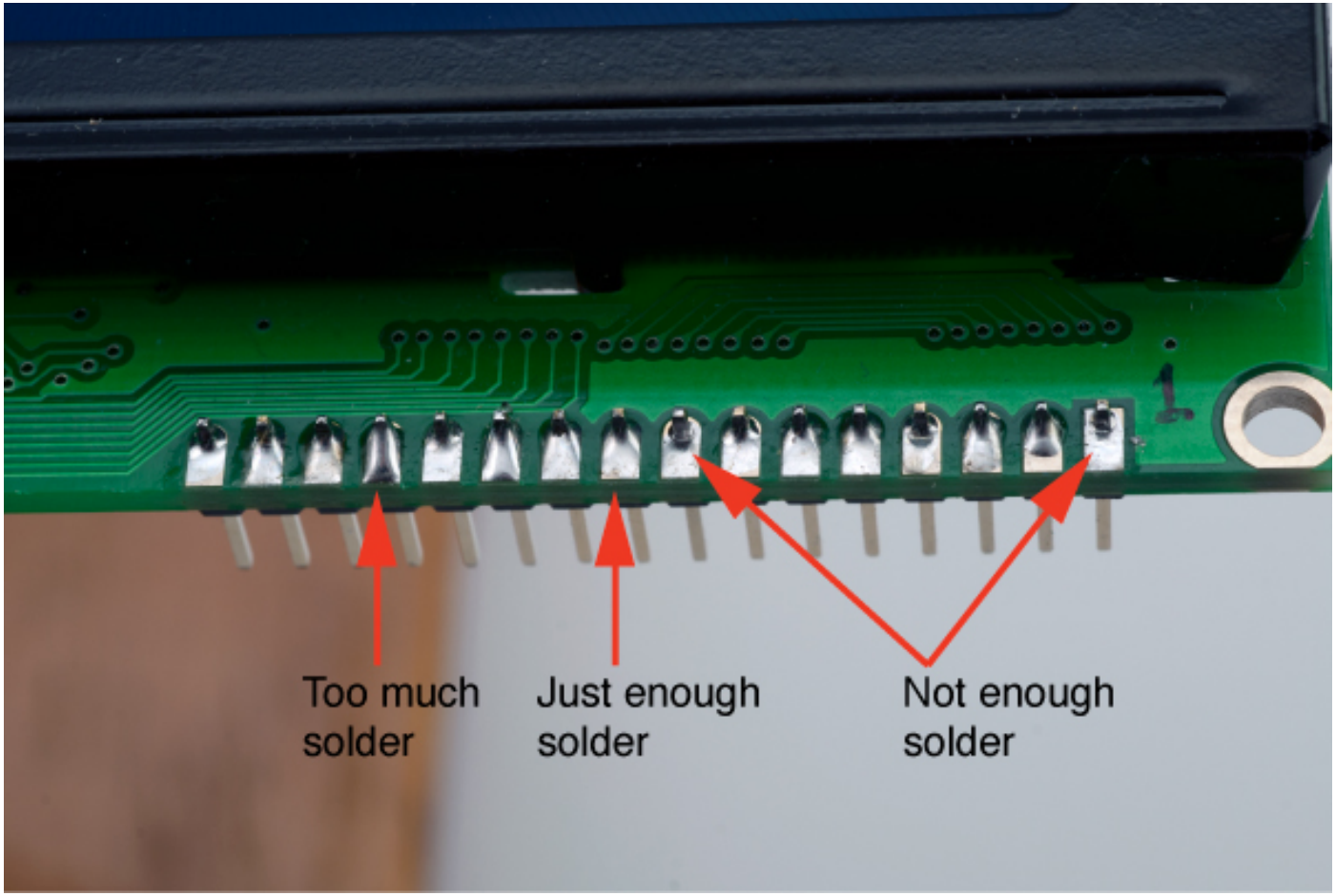
Soldering Tasks (for some)

Solder header on LCD Display



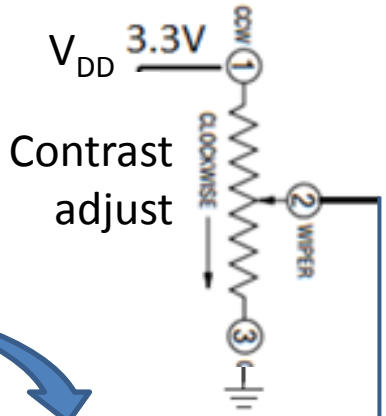
- Place header (short side) through holes from bottom.
- Solder in place on top.

Soldering LCD Display Header



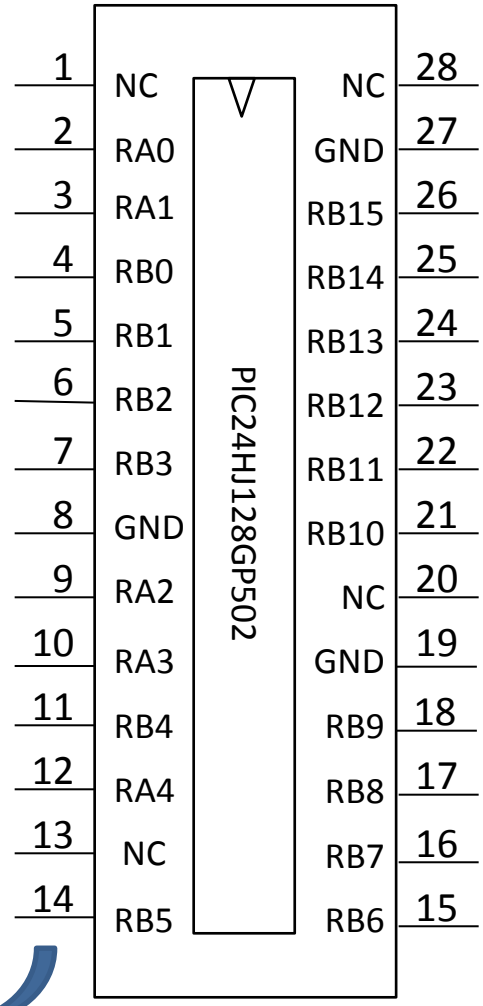
LCD Interface to PIC24

Sparkfun AMD1602K-FSY-YBS/3.3V



PIN	SIGNAL	
1	VSS	GND
2	VDD	V_{DD}
3	V0	
4	RS	RB9
5	R/W	RB13
6	E	RB14
7	DB0	NC
8	DB1	NC
9	DB2	NC
10	DB3	NC
11	DB4	RB5
12	DB5	RB6
13	DB6	RB7
14	DB7	RB8
15	LED+	V_{DD}
16	LED-	GND

LCD Display Pins

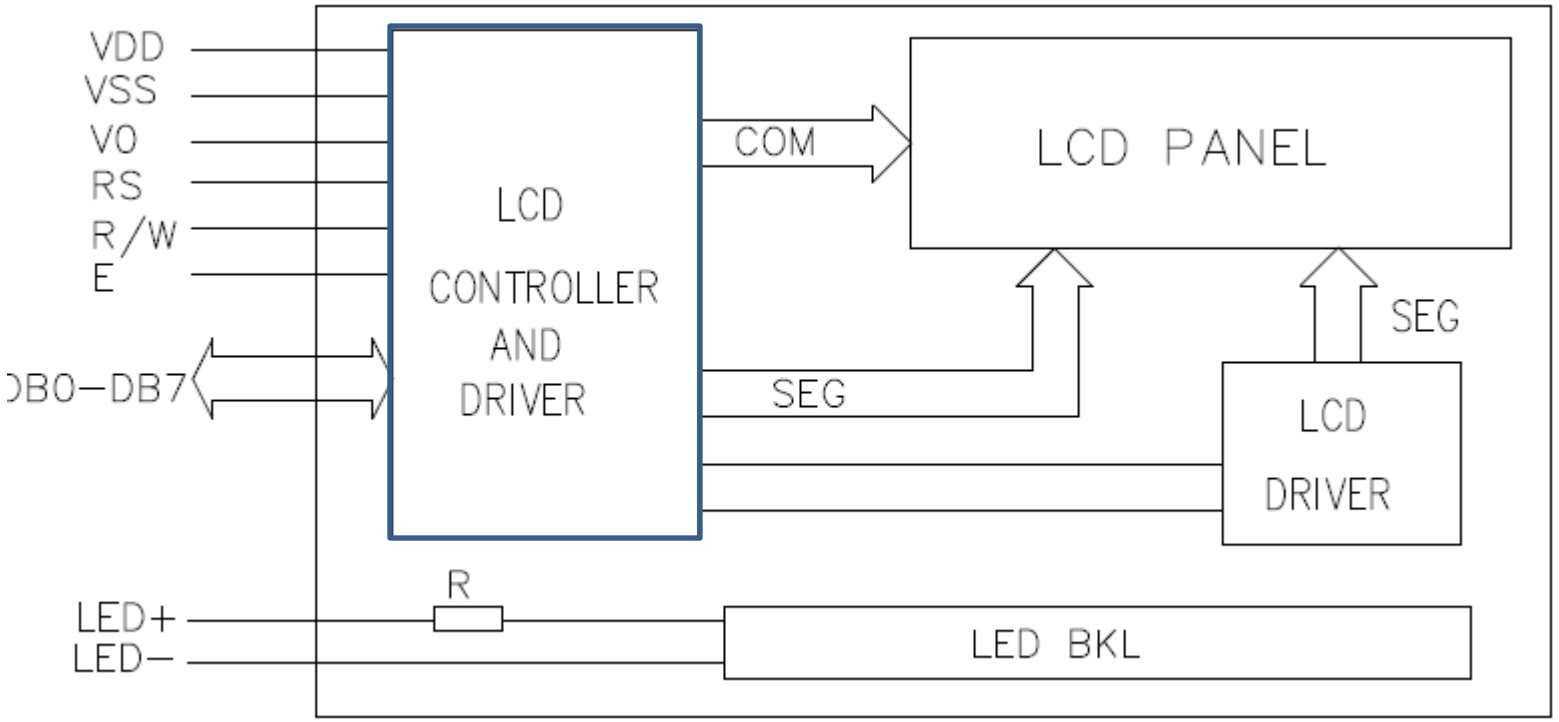


PIC24 Pins on Microstick II

LCD Displays

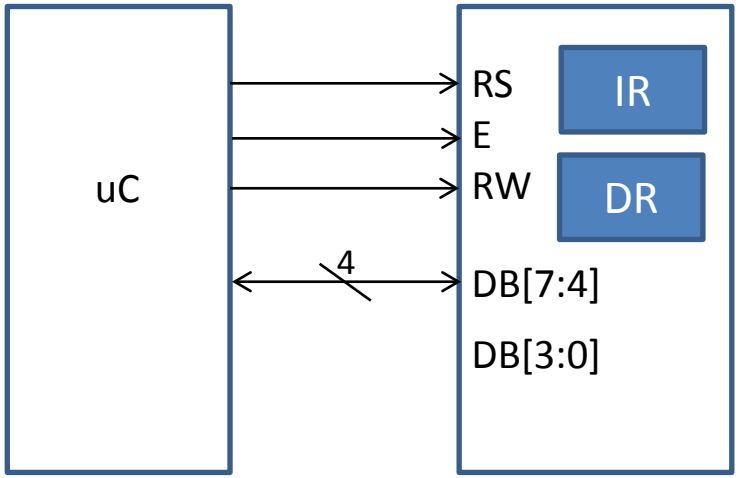
- Use linear polarizers to pass or block light.
- Arranged in matrices, controlled by ROM to create patterns that contrast with background.
- Backlit or ambient light

Inside the LCD Display



LCD Display Control Interface

- 4-bit bus or 8-bit bus
- Instruction Register (IR)– uC writes commands to this register
- Data register (DR) used to store data read or written by uC
- RS – Instruction/Data
- E – Enable
- RW - Read/Write

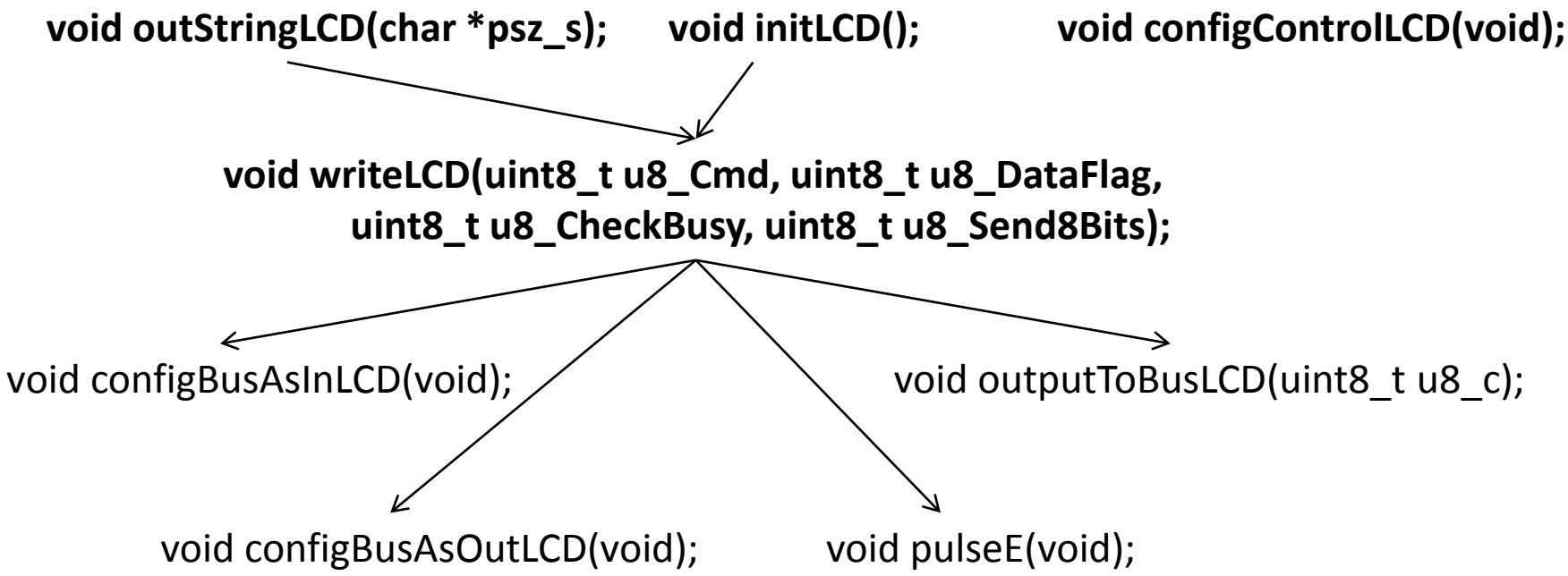


Interfacing LCD Display - Software

LCD Library Functions defined in lcd4bit_lib.c

```
void configBusAsOutLCD(void);
void configBusAsInLCD(void);
void outputToBusLCD(uint8_t u8_c);
void pulseE(void);
void configControlLCD(void);
void writeLCD(uint8_t u8_Cmd, uint8_t u8_DataFlag,
              uint8_t u8_CheckBusy, uint8_t u8_Send8Bits);
void initLCD();
void outStringLCD(char *psz_s);
```

Library Function Dependencies



InitLCD(void)

```
// Initialize the LCD, modify to suit your application and LCD
void initLCD() {
  DELAY_MS(50);         //wait for device to settle
  writeLCD(0x20,0,0,0); // 4 bit interface
  writeLCD(0x28,0,0,1); // 2 line display, 5x7 font
  writeLCD(0x28,0,0,1); // repeat
  writeLCD(0x06,0,0,1); // enable display
  writeLCD(0x0C,0,0,1); // turn display on; cursor, blink is off
  writeLCD(0x01,0,0,1); // clear display, move cursor to home
  DELAY_MS(3);
}
```

writeLCD(uint8_t **u8_Cmd,** uint8_t **u8_DataFlag,**
uint8_t **u8_CheckBusy,** uint8_t **u8_Send8Bits);**

```
void writeLCD(uint8_t u8_Cmd, uint8_t u8_DataFlag,  uint8_t u8_CheckBusy, uint8_t u8_Send8Bits) {
  uint8_t u8_BusyFlag;
  uint8_t u8_wdtState;
  if (u8_CheckBusy) {
       RS_LOW();          //RS = 0 to check busy
       // check busy
       configBusAsInLCD();  //set data pins all inputs
       u8_wdtState = _SWDTEN;  //save WDT enable state
       CLRWDT();         //clear the WDT timer
       _SWDTEN = 1;          //enable WDT to escape infinite wait
       do {
          E_HIGH();
         DELAY_US(1);  // read upper 4 bits
          u8_BusyFlag = GET_BUSY_FLAG();
          E_LOW();
          DELAY_US(1);
          pulseE();           //pulse again for lower 4-bits
       } while (u8_BusyFlag);
      _SWDTEN = u8_wdtState;   //restore WDT enable state
  } else {
       DELAY_MS(10); // don't use busy, just delay
  }
  configBusAsOutLCD();
  if (u8_DataFlag) RS_HIGH();   // RS=1, data byte
  else    RS_LOW();          // RS=0, command byte
  outputToBusLCD(u8_Cmd >> 4);  // send upper 4 bits
  pulseE();
  if (u8_Send8Bits) {
    outputToBusLCD(u8_Cmd);     // send lower 4 bits
    pulseE();
  }
}
```

outStringLCD(char *psz_s)

```
void outStringLCD(char *psz_s) {
  while (*psz_s) {    //until null character
      writeLCD(*psz_s, 1, 1,1);
      psz_s++;
  }
}
```

References for LCD Display

- Datasheet on Nexus
- On-line datasheet for ST7066 LCD Controller