LCD Animation with a 1602 display and Pic16f887

Aidan Pooler, Gabe Stotler

ap6522@ship.edu
gs1402@ship.edu

Abstract: This project produces an animation, with two unique frames, outputted to a backlit 1602 LCD. The LCD interacts with a PIC16f887 that has been programmed with a PICKit3. The pic16 operates with an external clock system that runs at a frequency of 8 MHz. The entire circuit is powered with 4.75V supplied with the PICKit3, which gets its power from any USB connection capable of supplying the necessary voltage. All of the programming was done in C using the Xc8 compiler.

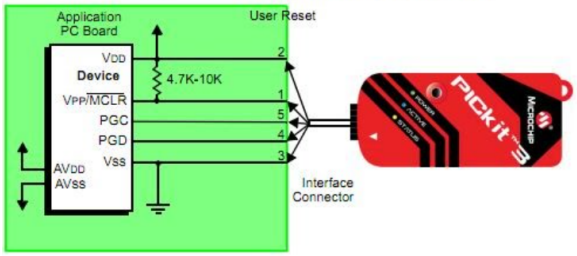
I. Introduction

The 1602 LCD display operates under parallel communication through 8 pins (D0 - D7), along with a R/W pin, an RS pin, an A and K pin,an enable pin and ground and power supply pins. Using this, we will communicate using the PICKit3 and PIC16f887 microcontrollers. In addition, an external crystal clock will be used and created and utilized. The MPLAB IDE will be used to compile and communicate code to the PIC16 microcontroller. This project will use each of these devices to create an animation that will be displayed on the LCD screen with a live-time contrast adjustment using a potentiometer.

II. Implementation

A. Programming the PIC16

In order to program the PIC16 which was purchased as an individual component and came with no way to easily program, the Pickit3 was utilized for this. The PICKit3 is the second latest release in the PICKit series. The PICKit4(the most recent model) relies on an 8 pin connection to interact with its pic. The PICKit3 however only has 6 connections and would provide everything necessary for the project. The connections corresponded to the similarly named pins of the PIC16.



B. Clock Source

In order for the PIC16 to operate in any instance where timing or pacing is required, a clock source would be necessary. There is no internal clock built into the PIC16 or the LCD, however an internal clock can be configured within the PICKit3. With the possibility of having the circuit operate on a separate voltage source and eliminating the PICKit3 power source. It made sense to have a separate clock system in the case of having a circuit redesign. The clock is based on the standard clock circuit, containing a 8 MHz resonator and two 27uF capacitors, connected to the OSC1 and OSC2 pins.

C. LCD configuration

1) Data Transmission: The LCD operates using parallel communication. There are 8 pins dedicated to communication on both

the LCD and the PIC16. There are several pins that are used to maintain the communication between the pins. These are the read/write(rw), enable(e), and the register select(rs). The rw pin is used to tell the LCD if it is reading or writing information to the PIC16. The LCD is only used to output information supplied by the PIC16. That means it is only going to read, so it was wired directly to ground. RS is used to switch between the LCDis receiving an instruction or character data to be displayed. When rs is 0, an instruction is being sent, and if it's 1 then the LCD will receive character data. The enable pin is used to tell the LCD that an instruction is being sent. The bus is filled with the values that are to be sent. The enable pin is set to one, the LCD receives the information, and then enable is set back to 0, signaling that data is no longer being sent

2) Power: There are three parts of the LCD that require power: main power, backlight, and contrast. The main power is the only part of the LCD that requires power in order for the LCD to work. The 5x8 matrix of cells are powered by the main power which is what visually displays the characters that the LCD receives. The backlight is an LED array that allows the cells to be more easily viewed. The LCD cells, without any light shining on the cells, makes them impossible to see. External light from the sun or room lights are able to create distinctions between lit and unlit cells however the constant even light of the backlight makes visualizing the cells much easier. The final usage of the power, the contrast, is used to adjust the contrast between lit and unlit cells. The reason this is useful is because different levels of ambient light make the LCD easier to see at different contrast levels. The way this is implemented is by using a potentiometer to output a voltage to compare with the input voltage. A potentiometer is wired with the same voltage being supplied to it as the main power. This voltage runs to the contrast pin of the LCD. The less voltage that's supplied, the higher the contrast. When all of the voltage is allowed through from the potentiometer, the contrast difference between lit and unlit cells is nonexistent. The potentiometer has a knob that allows for convenient, real time, contrast adjustment.

III. Code

A. Pin Configuration

The pins on the PIC16 require configuration in order to be used with the LCD. The pins that aren't being used for voltages, or for connection with the PICKit3, are being used with the LCD. The communication between the LCD and the PIC16 is only from the pic16 to the LCD. That means all of these pins should be configured as output. All of the data pins are on PortD which allows data to be transferred over a single port. The TRIS for PortD and the bits for the individual other pins are set to 0 in order to set them to be used as output

B. Data Transfer

There are two functions that are used to generate and send data. One to send instructions and one to send character data. Each function consists of 3 parts. The first part is to set the data to PortD. The data (least significant - most significant) is assigned to PortD where it can be sent to the corresponding D0-D7 bits. The next part is to set the rs bit to the correct value depending on the function(0 for instruction and 1 for character data). Then the enable portion is used. The enable pin is set to 1 to send the data, a delay occurs to make sure the LCD acquires the data, then the enable bit is set to 0 so the data transfer is recognized as over. This is the setup of the functions that can be used endlessly to send instructions or character data.

C. Initialization

The LCD must be initialized in order to be used. The LCD does not come set up to be ready to accept dta, and the entire LCD is not even on initially. There are two instruction

commands that make up the initialization function. To choose what instruction the PIC16 will send to the LCD, the first set bit is used. There is a set list of 11 instructions that can be used depending on what the leading bit is.

Instruction Table:

Instruction	Instruction Code										Description	Description Time (270KHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM, and set DDRAM address to "00H" from AC	1.52 ms
Return Home	0	0	0	0	0	0	0	0	1	x	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.52 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 us
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	D=1:entire display on C=1:cursor on B=1:cursor position on	37 us
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	x	x	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	37 us
Function Set	0	0	0	0	1	DL	N	F	x	x	DL:interface data is 8/4 bits N:number of line is 2/1 F:font size is 5x11/5x8	37 us
Set CGRAM address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter	37 us
Set DDRAM address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter	37 us
Read Busy flag and address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 us
Write data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM)	37 us
Read data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM)	37 us

The first instruction is sent as 38 in hex, the leading bit is bit 5 so the instruction is function set. The following bits are used to configure whatever the selected instruction is. The bits for this instruction are used to configure this specific LCD. The second instruction is 0c, this is used to set the CGRAM address. This is needed so that the starting position to write characters is set as well as where custom characters can be written to. This function is used before using any of the other commands for the given program.

D. Post Initialization Setup

The initialization portion of the code takes care of any setup that is necessary to utilize the LCD. However many projects that require unique uses of the LCD, require setup beyond that. The main portion of this is the generation of custom characters. The LCD comes with characters that are already known in the(read only memory). This may not be enough for many situations. In this scenario, custom characters must be assigned to the LCD.. In order to make this process simple, a function was designed to quickly generate custom characters

E. Custom Character Function

The function to generate custom characters and write them to the LCD, utilizes two parameters, a character array and the location to be stored. The array of characters consists of 8 characters that are written in hexadecimal form. The LCD character matrix consists of 8 by 5 individual cells. In order to assign a value for each of these cells, There are 8 characters but the characters themselves only ever use 5 of the 8 bits. The bits in the array assign the lit and unlit cells of the array for each row. All of this data must be passed to the LCD. This information is stored in the CGRAM. In order to put this information into its own spot a location parameters are used. The location parameters apply an offset of 8 multiplied by the location. So each location has 8 address locations allocated to them, at their given position past the start of the CGRAM. Once this is done, the characters can be used the same as any of the other built in characters. The location parameter is passed to the data function and the character is printed to the LCD.

IV Animation

A. Design

The previous coe has been designed to make this step of the process much simpler. The post setup of the code consists of a loop, the character printing, and the frame selection. An infinite while loop is used to to constantly iterate through the process of the animation. An iterator is used and checked to see whether it is even or odd. Depending on whether it's even or odd decides if it will play frame one. Frame one is a set of 4 characters that are connected and the second frame is another set of similar looking characters, when these are swapped back and forth quickly, the illusion of movement is given. There is a delay used between the swapping of the frames. These are used to balance the

swapping speed and make the animation look as natural as possible. After every frame is played, an iterator for the first two vertical characters and one for the second two vertical characters is incremented. This moves the animation forward one position. Once they reach the end of the LCD, their position is set back to the beginning to restart the animation cycle. The LCD is cleared between every frame so that the next frame replaces the previous one and they don't just multiply.

V Conclusion

The project resulted in an animation that was able to "move" along the 1602 LCD display. Trial and error had been a major portion in creating this animation. Initial circuit creation exploited communication issues with the LCD, causing certain LCD commands to react inappropriately with the PIC16 microcontroller. In addition, coding interfaces would unexpectedly fail (communication issues between the PICKit3 and the MPLAB IDE). However, the issues were able to be resolved by checking voltages, rewiring the circuit, and generating different code than previous versions. In general, results concluded to be accurate for our design and maintained functionality through the demo process. The concluded design allowed for a small animated "crab" to illusively "move" along the pixel segments of the LCD. Coding interfaces maintained functionality as well when improving circuit design and configurations. Overall, the project was able to successfully display an animation on the 1602 LCD using an external clock, the PICKit3, and a PIC16f887 microcontroller.

References

[1] Clark, C. (2022) LCD interface with PIC microcontroller: Beginner's Guide, Circuits Gallery. Available at: https://www.circuitsgallery.com/lcd-interface-with-pic-microcontroller/ (Accessed: December 3, 2022).

[2] Jayant (2015) LCD interfacing with 8051 Microcontroller, (89S52): Tutorial with Circuit Diagram and Code. Available at: https://circuitdigest.com/microcontroller-projects/lcd-interfacing-with-8051-microcontroller-89s52 (Accessed: December 2, 2022).

[3] Microchip (2009) Microchip PIC16F887-I/P Datasheet. Available at: https://octopart.com/datasheet/pic16f887-i%2Fp-microchip-484911 (Accessed: December 1, 2022).

[4] SLR (2022) PIC16F877A - LCD interfacing tutorial, EmbeTronicX. Available at: https://embetronicx.com/tutorials/microcontrollers/pic16f877a/lcd-interfacing-with-pic16f877a/ (Accessed: December 1, 2022).

[5] Unkown (2017) Scrolling text on LCD using PIC microcontroller: Mikro C, Microcontrollers Lab. Available at: https://microcontrollerslab.com/scrolling-text-lcd-pic-microcontroller/#:~:text=1%20Lcd_Cmd%20%28_LCD_CLEAR%29%3A%20To%20Clear%20the%20LCD%202,a%20shifted%20display%20to%20its%20to%20home%20position. (Accessed: November 25,, 2022).

[6] WaveShare LCD1602 系列 (no date) WaveShare. Available at: https://www.waveshare.com/datasheet/LCD_en_PDF/LCD1602.pdf (Accessed: November 28, 2022).