

Spelling Game

Custom Project Final Report

Jacques Fracchia

UCR CS120b

Spring 2019

https://youtu.be/CC_dZanOxe0

Spelling Game Description and User Guide

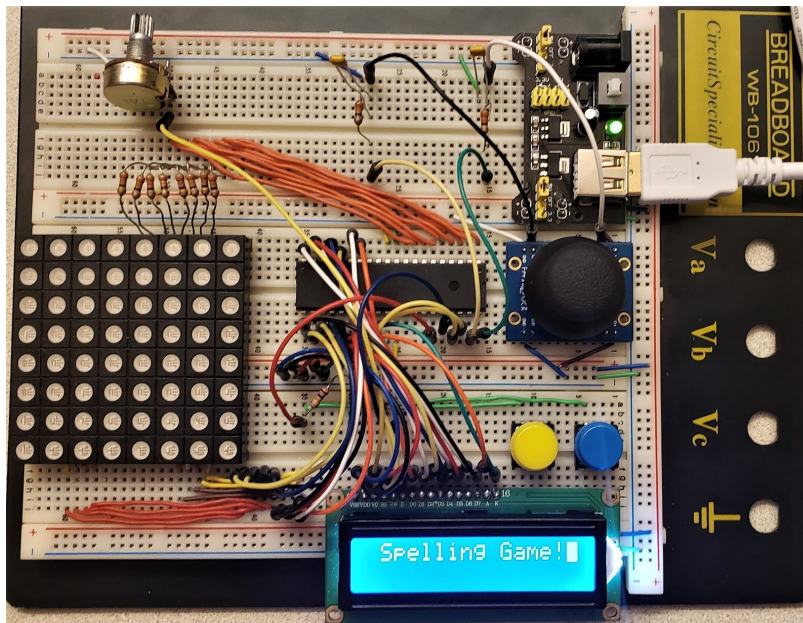
Spelling game is a prototype that is made to help the user memorize how to spell words. It was created using an 8x8 LED matrix, LCD screen, two push buttons and a 2-axis joystick. After a the LCD screen displays “Spelling Game!”, the player is prompted to a press start screen. Once any button has been hit the LCD displays a few directions on how the game is played, beginning level one.

At the beginning of each level the LED matrix scrolls through word printing each letter at a time. After the word is finished displaying, the LCD screen prints out a letter. The player must then use the joystick to scroll up or down through the alphabet, selecting the blue button to accept the letter moving on to the next letter in the word.

The player repeats this process until they have the whole word printed out on the screen. Once they feel that they have accurately spelled out the correct word, pressing the yellow button will check to see if the inputted word was correct or not. If the word was spelled correctly they will be prompted to a “LEVEL COMPLETE” display on the LCD, or if not, the LCD screen will display “GAME OVER”.

Depending on whether the inputted word was a success or not determines what happens next in the system. If the word was spelled correctly the game moves on to the next level, and displays the next level along with the high score. Each next level displays a different word on the LED matrix quicker than the previous level. If the word inputted was incorrect, after the game over screen the player is prompted to a “Last Level: “ display and a “High Score: “ display. The game then resets and restarts the “Spelling Game!” prompt.

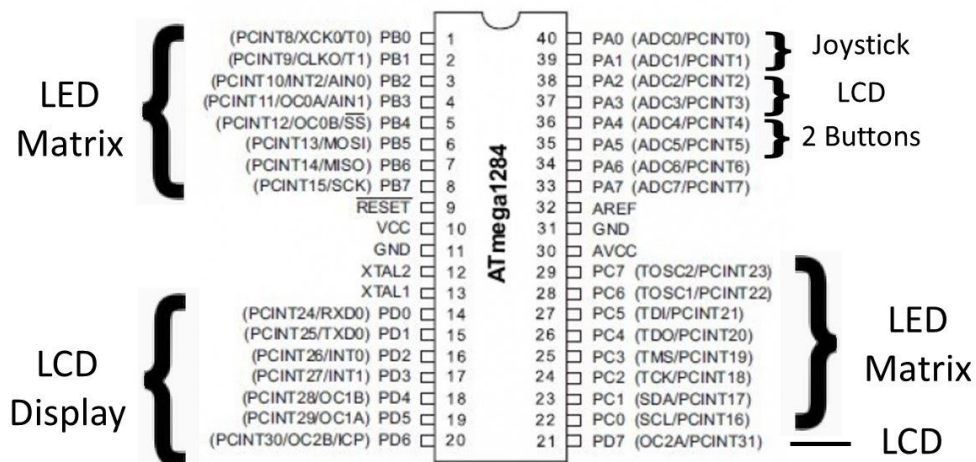
Parts List



Spelling Game Prototype

ATmega1284p microcontroller

- LCD Screen
- 8x8 LED Matrix
- 2-Axis Joystick
- 2 Push Buttons



Components Overview:

- 8x8 LED Matrix
- 2-Axis Joystick
- EEPROM
- LCD Special Characters
- Spelling Game Logic (Non-Trivial State Machine)

Components Description:

- 8x8 LED Matrix:
 - This component was used to display the words the player will read and memorize.
- 2-Axis Joystick:
 - The joystick was used to take input from the user to either scroll up the alphabet or down the alphabet. This was needed to choose each correct letter to spell out the word on the current level.
- EEPROM:
 - The ATmega1284 came with 128kb of memory on the EEPROM. This memory was used to save the current high score value. This was needed so that once the system was restarted it wouldn't lose the high score value.
- LCD Special characters:
 - Special characters were created by programming each pixel on the LCD screen's individual cursor position. Once the system starts the LCD screen prints out a scrolling zig zag across the entire screen to build up to the game's name.
- Spelling Game Logic:
 - Multiple non-trivial state machines, functions and include files were needed to implement the game logic.

What I learned

This project required constant attention. I came up with a project I believed I could build and accomplish within a short two week sprint. Buying new components that didn't come with instructions on how to wire it correctly or how to develop code for forced me to look for resources online. After scavenging for spec sheets on how to wire it up promptly I finally was able to connect each component in the system and have them integrate with each other.

Once I had it wired up, I had about a week left to design and build the game logic and code the implementation of each component. Working with C, I found how easily it is to have data accidentally overflow and affect other components in the system. With each component being so sensitive I had to pay close attention to every bit that was moving throughout the system.

This introduction to embedded systems project gave me a crash course on how basic embedded systems works. Working on this final project I learned that when building an embedded system work with small components first. It is worth to start small and test things thoroughly versus trying to build everything at once, have it all break, and not know where any of the bugs are. Having patience and dedication was necessary to build the spelling game, the end result made it all worth it.

Source Code and Youtube Demo:

Final Project Folder:

- https://drive.google.com/open?id=1CXnRBr1QLVY7_FJS-1v7Q3YemCVYA3bn

“Io.c” used for the basic functionality of the LCD Screen:

- https://drive.google.com/open?id=1WoA7Pt2n2diCRAJoFoz7iEy2hsXzk_Fk

‘levels’ is a folder that includes two .h files. “levelOne.h” contains tick functions for the implementation of each level and “levelFunctions.h” contains all functions that main.h and levelOne.h require:

- https://drive.google.com/open?id=1_PACp3ZhA98TgBwGmvH23DEZWYJW4OjA

Youtube Demonstration:

- https://youtu.be/CC_dZanOxe0

