**Customizable LED Matrix**

Final Project

Sage Marks and Matthew Krueger

3360:0001 – Embedded Systems

Professor Beichel

University of Iowa, College of Engineering

5/9/25

Table of Contents

[1. Introduction 3](#_Toc197362986)

[a) Motivation and background 3](#_Toc197362987)

[b) Goals and Specifics 3](#_Toc197362988)

[2. Implementation 3](#_Toc197362989)

[a) Overview 3](#_Toc197362990)

[b) Hardware Description 3](#_Toc197362991)

[c) Software Description 3](#_Toc197362992)

[3. Experimental Methods 4](#_Toc197362993)

[4. Results 4](#_Toc197362994)

[5. Discussion of Results 4](#_Toc197362995)

[6. Conclusion 5](#_Toc197362996)

[Acknowledgements 6](#_Toc197362997)

[References 6](#_Toc197362998)

[Appendix – Source Code 6](#_Toc197362999)

## Introduction

### Motivation and Background

When brainstorming the final project for the embedded systems course at the University of Iowa, our focus was to develop a project that can be used in our everyday lives. One interesting project area we discussed was a customizable LED matrix. The LED matrix has endless project opportunities from games to drawing to just displaying images as décor. Having this product would not only be great for this class but it is something that can be further developed down the line. Any idea or something that we feel is a need in our lives can be implemented with a bit of software. The matrix posed the perfect opportunity to develop a new project that includes hardware and software that differs from previous labs in the course.

### Goals and Specifics

The goals of this project were to develop a customizable LED matrix, that can be interacted with by the user. The interaction with this matrix was to be done through buttons, RPGs, and joysticks. The free draw mode on the matrix functions similarly to an existing product the “Etch A Sketch.” This mode allows the user to draw with the RPGs and change colors while drawing. Other modes that were implemented were the games mode and the image mode. The games mode is for any future games or functionality that the programmer wishes to add. The image mode can have any pixelated image displayed on the screen. The implementation of these modes and the use of the matrix with the Arduino proved to have some challenges, especially relating to memory.

## Implementation

### Overview

### Hardware Description

There was a good deal of hardware that went into the implementation of our customizable LED matrix. The memory required to operate the LED matrix exceeded the amount provided by the Atmega328P. Knowing this fact, we used an ESP32 for all interactions with the LED matrix. The Atmega328P served as a microcontroller that operated with input devices only. This meant that the push buttons, RPGs, and joysticks all were connected to the Atmega328P. All these devices were interrupt-driven besides our power button. Periodically the flags these interrupts set were sent over USART to the ESP32 to be processed. Below is an exhaustive list of all the hardware that was included in our project.

|  |  |  |
| --- | --- | --- |
| Hardware | Quantity | Description |
| Atmega 328P µC  ESP-WROOM-32  Active Low Push Button  Rotary Pulse Generator  KY-023 Joystick module  RGB LED Matrix Panel  Bidirectional logic level converter  1N5189 diode  IRF4905 PMOS  2N222A NPN transistor | 1  1  8  2  2  1  1  2  1  1 | Programmable µC  Programmable µC  Push button for user inputs  Etch A Sketch drawing user input  Joystick for controllers (XY-axis)  LED display  Communication ESP32-Atmega 328P  Power latch circuit- reverse voltage spikes  Power latch switching  Power latch |
| 10KΩ Resistor | 12 | Debouncing and power latch circuit |
| 0.01µF Capacitor | 5 | RPG and push button debouncing |

**Figure 1**: Materials List

The connection of these parts is also outlined below in the project schematic.

### Software Description

## Experimental Methods

## Results

## Discussion of Results

## Conclusion

## Acknowledgements

## References

## Appendix – Source Code