**Electronic Etch A Sketch**

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

The Embedded Etch A Sketch was implemented using many hardware and software elements. Various user input options provide ease of access to the user to create any image they can imagine. Various menus and visual feedback tools were also implemented so that the user never feels lost when navigating the system. Through many design iterations and test programs we were able to implement a working solution that provides full sketching and image display functionality while remaining completely customizable by the user.

### Hardware Description

There was a good deal of hardware that went into the implementation of our Embedded Etch A Sketch project. 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 and RPGs 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  RGB LED Matrix Panel  Bidirectional logic level converter  1N5189 diode  IRF4905 PMOS  2N222A NPN transistor | 1  1  4  2  1  1  2  1  1 | Programmable µC  Programmable µC  Push button for user inputs  Etch A Sketch drawing user input  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 specific way in which our connections are made is outlined below in the project schematic. Many wire connections had to be made for this project, especially between the ESP32 and the LED matrix. The number of these connections quickly cluttered the back of the matrix housing. This is one area we can improve on our project in the future.

A diagram of a computer

AI-generated content may be incorrect.

**Figure 2:** Project Schematic

The user inputs integrated into our system consisted of two RPGs and four active low push buttons. The use of our RPGs is for drawing on the matrix, the RPG located on the left side of the matrix is used for moving the cursor and drawing left and right, while the RPG located on the right side of the matrix is used for moving up and down. This implementation is very similar to how the traditional Etch A Sketch works. The push buttons located on the side of the matrix are used for scrolling through menus and selecting menu options. The power button is used for toggling the entire system on and off. Images of these user inputs can be seen below in the figure.

A black speaker on a wood surface

AI-generated content may be incorrect. A close up of a device

AI-generated content may be incorrect.

**Figure 3:** User inputs

All connections were made through soldering and through-hole PCB. This was done to make a more modular solution capable of fitting inside the back of the Matrix assembly. A sizable learning curve was involved with soldering these wires as many different techniques had to be employed to make sound connections. A 5V and ground rail was placed in the middle of the enclosure so that all peripherals could have easy access to a voltage source while maintaining common ground. This common ground is vital to ensuring signal integrity and correct voltage references. An image of the physical wiring and soldered PCBs can be seen below.

INSERT IMAGE OF WIRING ON THE BACK WITH CIRCLES LABELING WHAT IS WHAT

To power the system a latch circuit was implanted using a PMOS and NPN transistor. The NPN transistor locks the PMOS in an ON or OFF state when a button press is toggled. There are two GPIO pins utilized with this system. One is used for sensing button inputs while the other is used as an output to the NPN transistor to turn it on and off hence making the latch. Originally, we tried to power the system on a 9V battery. At first, this worked but as we added implementation for colors other than red that draw more current the colors began to display incorrectly or flash. Since the LED matrix has 4096 individual LEDs this is a large current drawing, especially when many LEDs are lit up. Colors such as white or blue require a greater current draw than a color such as red. The matrix can draw up to 4 or more amps. This is not suitable for a 9V battery, it was this fact that led us to switch our power supply to the wall wart plug that was utilized in lab 4 for the PWM fan.

The debounce approach for the RPGs used in drawing was like the method employed in previous labs. For the home, up arrow, and down arrow buttons we used a software debounce approach as we were running low on hardware and the space within the back of the matrix enclosure was wearing thin. The software debounce approach proved to work just as well as the hardware debounce.

To communicate between the ESP32 and Atmega328P microcontrollers a logic level converter is a must. This is because the ESP32 operates at a 3.3V logic level while the Atmega operates at a 5V logic level. The bidirectional logic level converter ensures that signals that are meant to represent a 1 or high voltage stay recognized as that, and so low logic levels stay in that range. This ensures proper communication. Serial communication via the UART was opted for as it proves as a reliable and easy-to-implement solution. The serial communication experience gained in the previous lab proved to be helpful as the functions and initialization of the UART were already known.

Talk about connections from ESP32 to the MATRIX

### Software Description

## Experimental Methods

## Results

## Discussion of Results

## Conclusion

## Acknowledgements

## References

## Appendix – Source Code