Mazda CX-0.005

“Will drive you and your kids crazy”

Group 2B or not 2B

Revision	Description
1.0	There may be more revisions than this and they can span to the next page
1.1	Natasha Abramowicz – updated section 1
1.2	Emma Hassler – Added block diagrams to section 3
1.3	Dayne Downie- Filled out section 6
1.4	Matthew Denoncourt – Initial update to section 4
1.5	Ethan Wilson - Made revisions to section 6, formatting
1.6	Calvin Ford- Updated Table of Contents, Table of Figures, and added sections 4.3 and 4.4
1.7	Natasha Abramowicz – updated section 1

Team Members

Natasha Abramowicz
Matthew Denoncourt
Justin Dhillion
Dayne Downie
Emma Hassler
Calvin Ford
Ethan Wilson

Originator:AuthorT

Checked: | Released:

Filename: Mazda CX-0.005 Write-up.docx

Title: **Mazda CX-0.005 “Will drive you and your kids crazy”**

Date:	Document Number:	Rev:	Sheet:
2/6/2023	**3-1415-926-5258-97**	**A**	**1 of 8**

Table of Contents

Table Of Figures

1. Scope

This document highlights an electromechanical car with the ability to drive in a circle, figure eight, and triangle twice. The car's components are powered by 4 "AA" batteries, each measuring at 1.5 V.

2. Abbreviations

PCB – Printed Circuit Board

LCD – Liquid-Crystal display

ROM - Read-Only Memory

RAM – Random-Access Memory

FRAM – Ferroelectric Random-Access Memory

MOSFET – Metal–Oxide–Semiconductor Field-Effect Transistor

NMOS/NFET – N-type MOSFET

PMOS/PFET – P-type MOSFET

PC – Program Counter

AD2 – Analog Discovery 2

MSP430 – The MSP430FR2355 Microcontroller

GND – Ground

DAC - Digital to Analog Converter

IAR - References the program "IAR Embedded Workbench" used to program the FRAM board

3. Overview

The current iteration of the car contains 5 components, which, respectively, are an MSP processor to control each of the other system components, an LCD display to project messages to the user, switches for the user to interact with the LCD display, a power display circuit board with makes sure each of the system components are receiving power, and a battery pack that supplies power to each of the powered components.

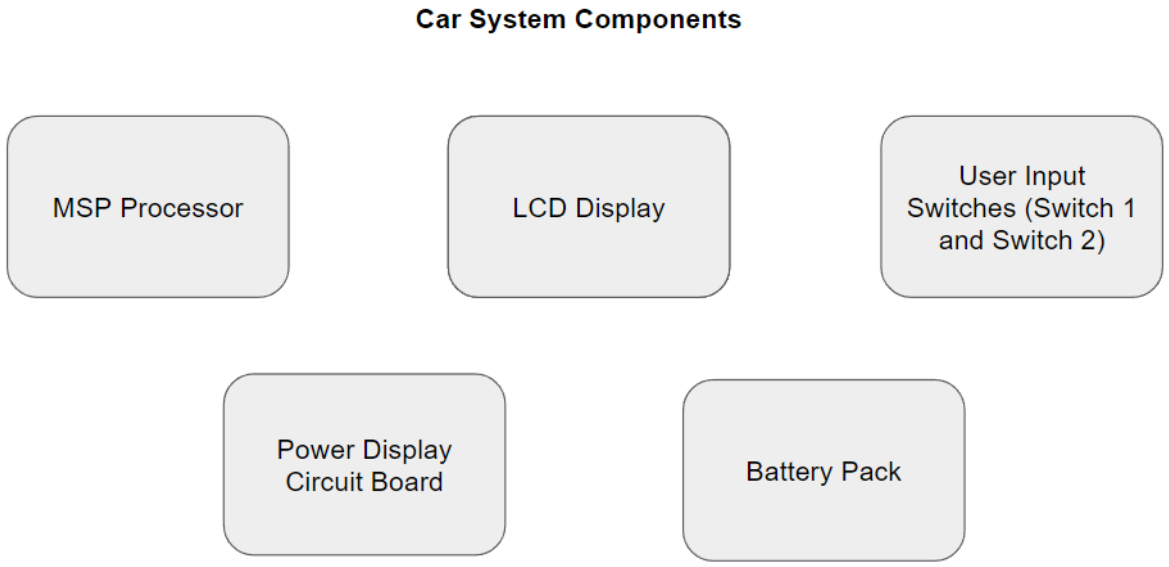
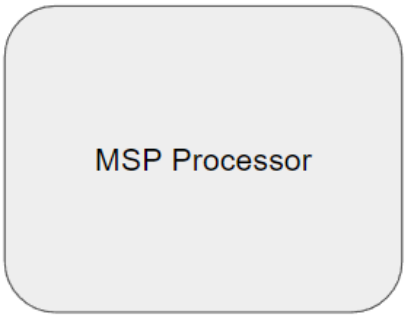


Figure 1: System Diagram of the Car

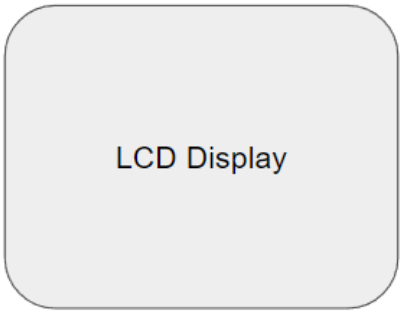
3.1. MSP430 Processor



- The MSP processor is the “brain” of the entire project. It tells each of the other subsystems what to do.
- The processor is a small computer chip that is mounted on a circuit board, and it is connected to all other components in the high-level system.
- The computer chip holds computer programs that contain instructions for the other system components, and the processor sends these instructions to the system components.
- The MSP processor takes messages from its programs and displays them on the LCD display.

Figure 2: MSP Processor Block

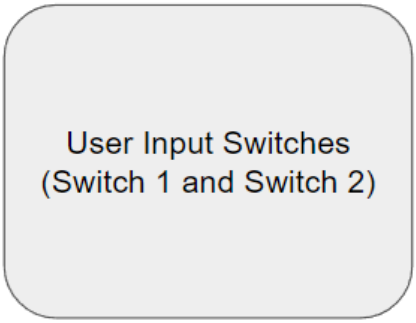
3.2. LCD Display



- The LCD display prints messages provided by the processor for users to read.
- Control of what is being displayed on the LCD display comes from the user input switches, allowing users to switch between different messages.

Figure 3: LCD Display Block

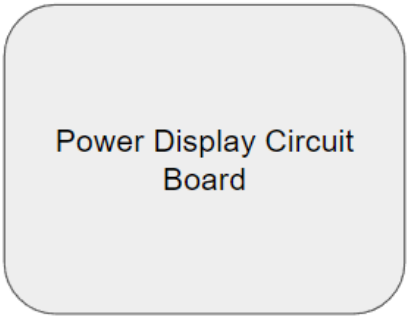
3.3. User Input Switches



- Two switches are available for users to change what message is being displayed on the LCD display, allowing users to toggle back and forth between two different messages.

Figure 4: User Input Switches Block

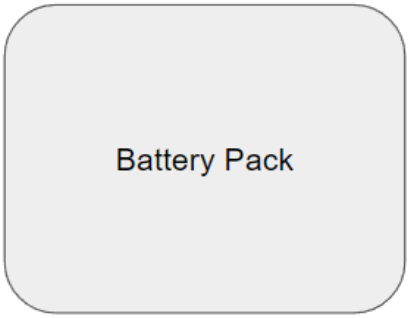
3.4. Power Display Circuit Board



- The power display circuit board is responsible for safely transmitting power from the battery pack to the other components in the car's system.
- This board contains circuitry which directs powers to the MSP processor and the LCD display, allowing each of them to perform their respective functions.

Figure 5: Power Display Circuit Board Block

3.5. Battery Pack



- The battery pack contains the batteries that provide power to other car system components. Namely, it provides power to the MSP processor and the LCD display, allowing each of those systems to perform their respective functions.
- The battery pack holds 4 "AA" batteries, and has a connector tab that plugs into the power display circuit board.

Figure 6: Battery Pack Block

4. Hardware

4.1. MSP430 Processor

The MSP430 Embedded system was developed by TI. It operates as the central processing unit of our car. It has several pinouts that allow it to interface with our LCD display board as well as any future parts we may add. We have and will continue to program the inputs and outputs of the MSP430 with C.

4.2. LCD Display

The LCD Display can display up to 4 lines by writing to the controllers DDRAM. It also has a backlight that can be enabled via pin 6.0 on the MSP. The reset for the LCD is hooked up to pin 2.0. In it's current implementation pressing SW1 and SW2 will display 2 unique messages to the LCD Display.

4.3. User Input Switches

Located on the sides of the FRAM board, two switches are used as a means of interfacing with the PCB once it is disconnected from any computers. The switches are equipped with the ability to digitally assign pull-up or pull-down resistors alongside their function.

4.4. Power Display Circuit Board

The power display circuit board is the board responsible for providing power to the rest of the assembly via an attached battery pack. The board takes the voltage being provided by the attached battery pack and converts it into a 3.3 volt signal. The battery pack is connected to the board using a 2 pin, right-angle connector. Attached to the board is a switch that disables or enables the power to the rest of the assembly. Alongside the switch, there is a thumbwheel potentiometer.

4.5. Battery Pack

The battery pack for this device consists of 4 AA batteries. The batteries are in series so the voltage supplied by each one will be added together while the current will stay the same. The battery pack connects directly to a header on the display/power board.

5. Power Analysis

Provide a description of the power consumption of each part. How long will it last running off of a battery(s).

6. Test Process

6.1. Power System

A) The power board was first tested using a voltmeter to make sure it was functioning; this was done by putting the positive probe of the voltmeter on terminal J0 and the negative probe on the ground terminal. The voltmeter then produced a reading of approximately 3.3V indicating that the circuit was working correctly.

B) The battery connector was installed and had to be tested. Once the battery connection was installed at J10 and the switch was installed at SW1, A voltmeter was used to see if the circuit was able to draw power from the batteries. This was done by connecting the battery pack to the power system and then measuring the voltage using a voltmeter with the positive probe at J0 and the negative probe at J5, the switch was then turned on and a voltage readout of 6.27V was collected indicating that the power system was functioning properly. Voltage was also measured between the terminals J12/13 and J0 and a voltage of approximately 3.3V was obtained indicating that the circuit was working properly.

6.2. Display

A) The display and backlight were installed onto the power board and needed to be tested. This was done by connecting the battery pack to the power board and turning on the switch to see if the board, which had been preloaded with custom code, would display preloaded text.

6.3. FRAM

A) The FRAM board which was purchased from Texas Instruments needed to be tested to make sure that it was working as intended, this was done by loading custom code onto the board using the IAR embedded workbench and then seeing if the board's LEDs would light up in accordance with the preloaded code.

The software is configured using a modular approach. Describe the code structure. Remember to identify the various functions and what operates when. This is a description of how your software is configured. You should be able to give this to one of your class mates and they would understand what you tried to do.

7. Flow Chart

The following is the flow chat for your code.

7.1. Main Blocks

Each function should have its own section and be on a separate page. Insert page breaks when necessary. This describes the software. What calls the function? Etc. Do not just put C code statements in blocks. Also do not write novels. Be reasonable.

7.2. Initialization Blocks

7.3. Interrupt Blocks

8. Software Listing

This is just a printout of the actual code, with each file in its own section.

8.1. Main.c

8.2. System_Init.c

8.3. Interrupt.c

9. Conclusion

What did you learn? If something did not work what went wrong.