EGR 326 Embedded System Design and Build Project

Specifications Document

Project – Digital Monster Pet

Version 1.0

11/10/2024

Authors: Thomas Zoldowski and Emmett Topp

Instructor: Prof. Kandalaft

Approval and Signatures

K+K Circuit Critters

Signature *Date*

Project Developers

Signature *Date*

Signature *Date*

Revision History

Date	Version	Author	Change
10/23/2024	0.1	Thomas Zoldowski & Emmet Topp	Initial version created, including project overview, UML diagram, user interface description, requirements, design constraints, verification plan, Appendix A and Appendix B.
11/10/2024	1.0	Thomas Zoldowski & Emmet Topp	Fixed Requirements table to include one requirement per row. Added Verification to Requirements table. Updated Block diagram to be a high-level diagram.

1 Overview

The Digital Monster Pet is an interactive embedded system built on the STM32 Nucleo microcontroller, simulating a pet that responds to user actions and tracks its emotional and health states. The project integrates hardware components such as an RTC module, stepper motor, rotary encoder, Hall effect sensor, and graphic LCD to deliver a standalone, autonomous experience. Software will be developed in C using Keil uVision5, emphasizing real-time control, user interaction, and persistent data storage.

System Development, Operations, and Maintenance

- Development: Integrate hardware, develop software, and validate the system in the lab through extensive testing.
- Operations: Enable user interaction through feeding, petting, and clock setting while displaying emotional states and tracking health.
- Maintenance: Ensure basic functionality through documentation, with no long-term production support planned.

Stakeholders and Sites

- Sponsor: K+K Circuit Critters
- Developer: Thomas Zoldowski, Emmett Topp
- Support: Course instructors and lab assistants
- Sites: GVSU Lab (development); standalone operation (post-prototype)

Technical Overview

The system will display emotional states on the LCD, wag its tail with a stepper motor, and respond with distinct sounds for happiness and hunger. It will retain time and health data across power cycles using the RTC and allow clock adjustments through a rotary encoder. The watchdog timer will ensure stability, resetting the system if necessary. A menu interface on the LCD will guide users for clock setup and automatically revert to emotional states if idle for over a minute.

This project focuses on hardware-software integration, meeting safety and usability standards. It will stay within a $50 budget for additional components, with optional enhancements like temperature sensing or a second motor available if needed.

1.1 Purpose and Goals

The purpose of this project is to design and develop an autonomous Digital Monster Pet using the STM32F446 microcontroller, demonstrating practical applications of embedded system concepts. The primary goal is to create an interactive, user-friendly pet that simulates real-time responses through sensory inputs, emotional states, and health tracking. The system will focus on integrating hardware components, such as motors, sensors, and an RTC, with software to provide seamless functionality and persistent data storage. This project aims to deliver a fully functional prototype that meets user requirements while adhering to safety, usability, and budget constraints.

2 High Level Block Diagram

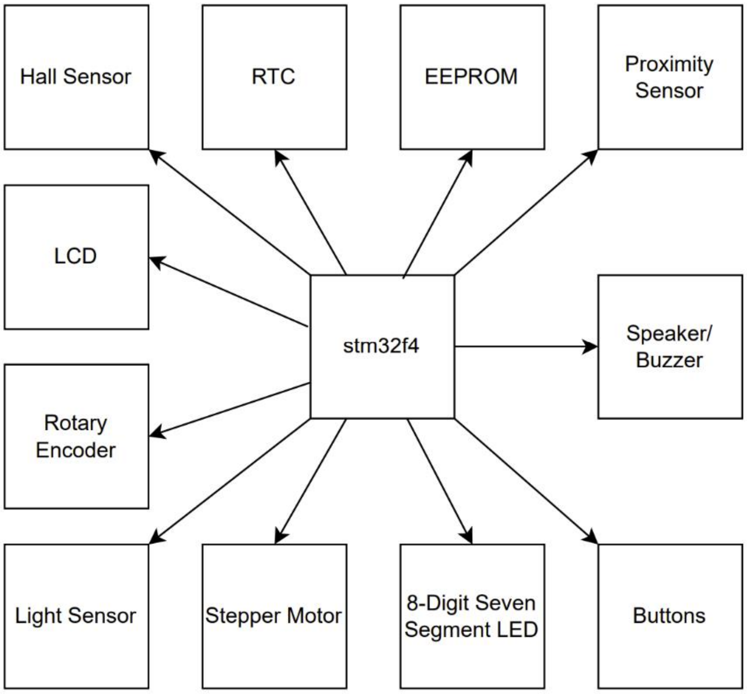


Figure 1. High-Level Block Diagram.

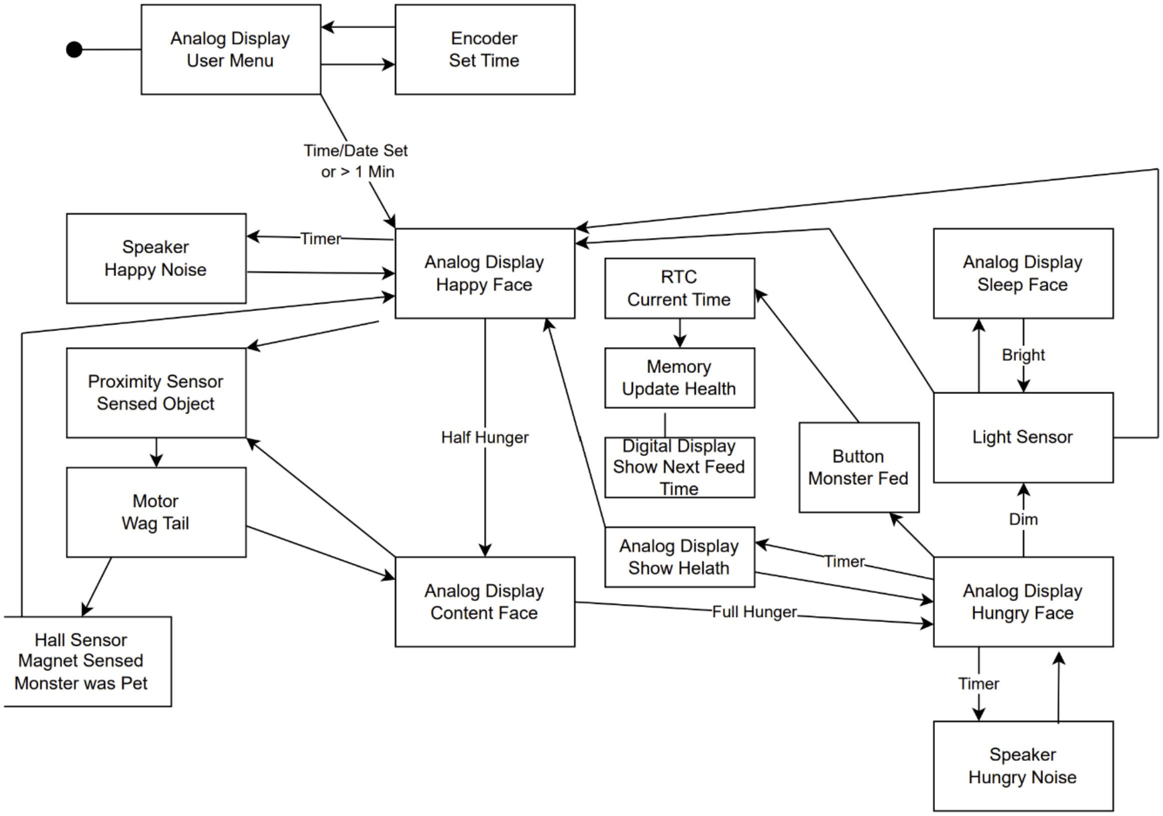


Figure 2. UML Block Diagram.

2.1 User Interface

Analog Display User Menu:

This block is where the user will enter the current date and time to be entered into the RTC using the encoder, moving to the analog display happy face block after the time and date have been set or if the user menu is open for more than 1 minute.

Encoder Set Time:

The encoder block is used by the user menu block to get the time and date information from the user for later use before returning to the user menu block.

Analog Display Happy Face:

The default mood of the monster pet is happy, which is where the program moves to after the user menu block. From the happy face block, there are many ways for the program to exit and reenter the block. When someone approaches the pet, the program moves from the happy face block to the proximity sensor block, sensing someone has come close to the pet. When the pet is at half hunger status, the program moves from the happy face to the analog display content face. A timer can also get the program to exit the happy face block in two ways. The first way is when the health status of the pet needs to be updated, moving to the show health block. The second way is when the timer reaches a certain value, the program moves to the happy noise block.

Happy Noise Block:

The happy noise block is where a speaker makes the monster pet's happy noise. The only way to enter this block is when the program is in the happy face block and a timer says enough time has passed to make more happy noises. The block makes this noise until the timer tells the program to move back to the happy face block.

Proximity Sensor Block:

In this block, the proximity sensor senses when someone or something has approached the monster pet, immediately moving to the tail wag block.

Motor Tail Wag Block:

This block is used when the proximity sensor tells the program someone has approached the pet, which causes the monster's tail to wag with a motor. The monster pet will be slightly offput since someone came near it and didn't show it attention by petting it, so after the tail wag block, the program moves to the display content face block. Or if a magnet is detected the program instead moves to the hall effect sensor block.

Hall Effect Sensor Block:

The hall effect sensor block indicates that the monster had been pet, so the next block to move to is the monster pet block.

Monster Pet Block:

This block is entered when the hall effect sensor detects a magnet, meaning the monster had been pet. Naturally since the monster had been pet, it will be happy, which is why the next block entered is the happy face block.

Analog Display Content Face:

This block is where the program goes when the monster is not happy, but also not sleepy or fully hungry. From this block, if the monster does get fully hungry, the program moves to the analog display hungry face block. Timers will also get the program out of this block if it is time to update the monster's hunger. If someone approaches the pet again while in the content face block, the proximity sensor block will be entered.

Analog Display Hungry Face:

The hungry face block will be entered when the monster pet reaches full hunger (no more food). A timer will tell the program to move to the show health block, lowering its health periodically. This block loops back into itself if the pet isn't fed. If the monster's feed button is pressed, the program moves to the monster fed block. If the light sensor senses darkness, the program will move to the light sensor block.

Light Sensor Block:

This block is entered anytime the light sensor senses a change in darkness or brightness. If the sensor senses darkness the monster will move to the analog display sleep face block. If instead brightness is detected, the monster will wake up happy by moving to the happy face block.

Analog Display Sleep Face Block:

This block can only be entered when the light sensor determines it is dark, meaning the monster will fall asleep. It is exited when brightness is detected by the light sensor, meaning the monster pet will wake up.

Monster Fed Block:

This block is only entered when the monster has been digitally fed with its feed button and immediately moves on to the RTC current time block.

RTC Current Time Block:

This block constantly keeps track of the current date and time. Since this block is entered when the monster has been fed, the block refills the monster pet's health and takes the time of last feeding in order for the program to determine when the monster pet's next feeding should occur. From this block the program moves to the memory update health block.

Memory Update Health Block:

This block updates the monster's current health status and stores it on an EEPROM so the health status of the monster is saved even after the system is shut off. From this block, the program moves to the next feed time block.

Next Feed Time Block:

This block uses the real time of next feeding for the monster found in the RTC current time block and displays it on a digital display, effectively showing the real time the monster pet needs to be fed next. After the time of next feeding has been updated on the digital display, the show health block is entered, showing the monster has been returned to full health.

3 Requirements & Verification Plan

#	Requirement	Referenced From	Verification Method	Verification Process
1	The system **MUST** display a "Happy" emotional state.	(Req. 1)	Test	Set the display to show the "Happy" emotion and confirm correct graphic display.
2	The system **MUST** display a "Hungry" emotional state.	(Req. 1)	Test	Set the display to show the "Hungry" emotion and confirm correct graphic display.
3	The system **MUST** display a "Sleeping" emotional state.	(Req. 1)	Test	Set the display to show the "Sleeping" emotion and confirm correct graphic display.
4	The system **MUST** display a "Content" emotional state.	(Req. 1)	Test	Set the display to show the "Content" emotion and confirm correct graphic display.
5	The system **MUST** provide a menu-driven interface for setting time and date.	(Req. 12)	Test	Use the menu interface to set the time and date, confirming usability and correct function.
6	The system **MUST** automatically revert to the emotional state display if the menu interface is idle for more than 1 minute.	(Req. 12)	Test	Leave the menu idle for over 1 minute and confirm the system reverts to emotional state display.
7	The system **MUST** wag its tail using a motor when proximity is detected.	(Req. 2)	Test	Test proximity detection by approaching the pet and confirming motor activation for tail wagging.
8	The system **MUST** emit a distinctive sound when the pet is "happy".	(Req. 3)	Test	Trigger the "Happy" state and confirm the correct sound is produced.
9	The system **MUST** emit a distinctive sound when the pet is "hungry".	(Req. 4)	Test	Trigger the "Hungry" state and confirm the correct sound is produced.
10	The system **MUST** enter a "sleep" state automatically when ambient light decreases.	(Req. 7)	Test	Simulate a decrease in ambient light and confirm the system enters "sleep" mode.
11	The system **MUST** allow the user to feed the pet using a button.	(Req. 5)	Test	Press the button to initiate feeding and verify that feeding state changes as expected.

12	The system **MUST** decrease health by one unit every minute when the pet is hungry.	(Req. 10)	Test	Simulate hunger for several minutes and confirm the health decreases at the correct rate.
13	The system **MUST** increase health when the pet is fed.	(Req. 10)	Test	Feed the pet and confirm that the health value increases as expected.
14	The system **MUST** display the current health status.	(Req. 9)	Test	Display health status and confirm the correct value is shown.
15	The system **MUST** maintain accurate time using a timekeeping module.	(Req. 13)	Test	Verify that the timekeeping module maintains correct time during system operation.
16	The system **MUST** retain time across power cycles using a timekeeping module.	(Req. 13)	Test	Power the system off and on, then verify that the timekeeping module retains the correct time.
17	The system **MUST** health status across power cycles using a storage module.	(Req. 13)	Test	Power the system off and on, then verify that the stoarge module retains the correct health status.
18	The system **MUST** operate autonomously without a computer connection.	(Req. 1, Design Constraint 1)	Inspection	Inspect system operation to confirm it runs independently.
19	The system **MUST** employ a watchdog timer to monitor performance.	(Req. 5, Design Constraint 5)	Test	Induce a software fault and confirm that the watchdog timer resets the system.
20	The system **MUST** securely house all components.	(Req. 17)	Inspection	Inspect the fixture to confirm stability, accessibility, and aesthetic appeal of components.
21	The project **MUST** adhere to a $50 budget for any additional components or enhancements.	(Design Constraint 5)	Analysis	Review component costs to confirm adherence to the budget.
22	The system **MUST** include options in the menu to set the hour.	(Req. 12)	Test	Use the menu interface to set time element and confirm correct update.
23	The system **MUST** include options in the menu to set the minute.	(Req. 12)	Test	Use the menu interface to set time element and confirm correct update.
24	The system **MUST** include options in the menu to set the day.	(Req. 12)	Test	Use the menu interface to set date element and confirm correct update.

25	The system **MUST** include options in the menu to set the month.	(Req. 12)	Test	Use the menu interface to set date element and confirm correct update.
26	The system **MUST** include options in the menu to set the year.	(Req. 12)	Test	Use the menu interface to set date element and confirm correct update.
27	The system **MUST** display the time remaining until the next feeding.	(Req. 8)	Test	Set feeding intervals and confirm the display updates with the countdown time.
28	The system **MUST** prevent the "tail wagging" behavior when in "sleep" mode.	(Req. 2, 7)	Test	Set the system to "sleep" mode and confirm the motor does not activate.
29	The system **MUST** record the last feeding time.	(Req. 8, 13)	Test	Feed the pet and confirm the last feeding time is recorded and retained across power cycles.
30	The system **MUST** provide a health display that reflects the pet’s current health.	(Req. 9)	Test	Vary the health level and confirm accurate health display.
31	The system **MUST** allow the user to set feeding intervals using an input device.	(Req. 8)	Test	Use the input device to set feeding intervals and confirm they are accurately applied.
32	The system **MUST** use a visual or audible indicator to alert the user if the pet is hungry.	(Req. 4)	Test	Let the feeding interval pass without feeding and confirm an alert is given.
33	The system **MUST** only play "hungry" sounds when the pet is hungry.	(Req. 4)	Test	Ensure that "hungry" sounds are only produced when the pet status is set to "hungry".
34	The system **MUST** log pet interaction history for up to the last 24 hours.	(Req. 13)	Test	Record interactions, power cycle the system, and verify that the last 24 hours of interactions are retained.
35	The system **MUST** show the current time on the display.	(Req. 8)	Test	Confirm that the current time is shown on the display and updates in real-time.
36	The system **MUST** allow feeding to be accomplished with a single button press.	(Req. 14)	Test	Press the button to feed the pet and confirm feeding is triggered on a single press.
37	The system **MUST** recognize petting with a single sensor activation.	(Req. 15)	Test	Activate the petting sensor once and confirm the system registers the petting.

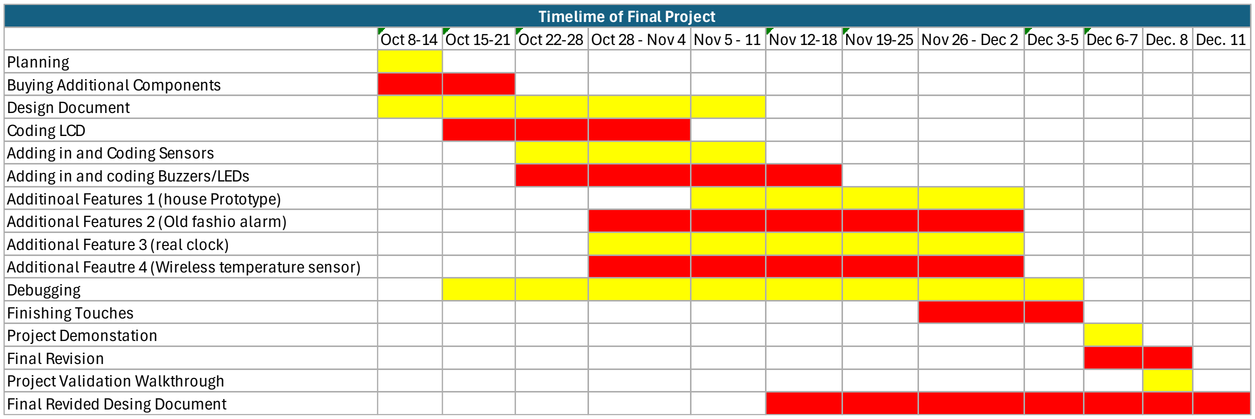
38	The system **MUST** display an error message if the user attempts to feed the pet while health is full.	(Req. 10)	Test	Attempt to feed the pet when health is full and confirm an error message appears.
39	The system **MUST** allow the user to adjust the volume of sounds.	(Extra Feature)	Test	Adjust the volume and confirm changes to sound levels.
40	The system **MUST** allow the user to mute sounds completely.	(Extra Feature)	Test	Mute the system and confirm that no sounds are produced.
41	The system **MUST** allow brightness adjustment for the display.	(Extra Feature)	Test	Adjust the brightness and confirm changes on the display.

3.1 Extra Requirements

#	Extra Requirements	Referenced From
E1	**MAY** include a temperature.	(Extra. Req. 4)
E2	**MAY** incorporate a second motor.	(Extra. Req. 2)
E3	**MAY** use a second 8-digit.	(Extra. Req. 3)
E4	**MAY** integrate all I2C modules on a single bus.	(Extra. Req. 5)
E5	**MAY** use a **custom PCB** to interface with the MCU, RTC, and sensors.	(Req. 18)

*All referenced costumer requirements and design constraints can be seen in Appendix A, B and C.

4 Project Timeline



Timelime of Final Project

	Oct 8-14	Oct 15-21	Oct 22-28	Oct 28 - Nov 4	Nov 5 - 11	Nov 12-18	Nov 19-25	Nov 26 - Dec 2	Dec 3-5	Dec 6-7	Dec. 8	Dec. 11
Planning	🟨											
Buying Additional Components	🟥	🟥										
Design Document	🟨	🟨	🟨	🟨	🟨							
Coding LCD		🟥	🟥	🟥								
Adding in and Coding Sensors			🟨	🟨	🟨							
Adding in and coding Buzzers/LEDs			🟥	🟥	🟥	🟥						
Additinoal Features 1 (house Prototype)					🟨	🟨	🟨	🟨				
Additional Features 2 (Old fashio alarm)				🟥	🟥	🟥	🟥	🟥				
Additional Feature 3 (real clock)				🟨	🟨	🟨	🟨	🟨				
Additional Feautre 4 (Wireless temperature sensor)				🟥	🟥	🟥	🟥	🟥				
Debugging		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨			
Finishing Touches								🟥	🟥			
Project Demonstation										🟨		
Final Revision										🟥	🟥	
Project Validation Walkthrough											🟨	
Final Revided Desing Document						🟥	🟥	🟥	🟥	🟥	🟥	🟥

Appendix A. Costumer Requirements Table

#	Requirement	Description
1	Emotional States on LCD	A bitmapped graphic LCD should display one of four expressions: Happy, Hungry, Sleeping, Content. These expressions must be drawn by the designer—not just uploaded images.
2	Tail Wagging	A tail should wag when someone approaches the digital monster pet.
3	Happy Sound	The digital monster pet should make a distinctive noise when it is happy.
4	Hungry Sound	The digital monster should make a different distinctive noise when it is hungry.
5	Feeding	If the digital monster pet is hungry, the user must be able to feed it.
6	Petting	Petting must be triggered electronically and make the pet happy.
7	Sleep Mode	The pet must fall asleep if the room gets dark.
8	Feeding Clock	A 4-digit clock must display the actual time for the next feeding (once every 1 minute).
9	Health Dial	A health dial must indicate the pet's current health status.
10	Health Management	The pet must lose one health unit every minute while hungry and gain health when fed.
11	Rotary Encoder Input	A rotary encoder must allow input to set the clock.
12	Menu and Idle Timeout	A menu must allow time and date entry. If idle for 1 minute, the system cancels the action and returns to showing expressions.
13	RTC Persistence	The system must retain clock time, date, and health state across power cycles using the RTC.
14	Feeding Button	Feeding must be triggered by a pushbutton.
15	Hall Sensor for Petting	Petting must be triggered by waving a magnet near a hidden Hall effect sensor.
16	AC-DC Power Supply	The system may be powered by an AC-DC switching power supply from a USB wall plug charging unit. A voltage regulator circuit on the microcontroller board provides power to the MCU, the RTC, and sensors. The USB 5V is provided to the motor circuit and other sensors that require 5V.
17	Component Fixture	A fixture must hold all components including the microcontroller board, connector board, display, pushbutton, and cable attachments attractively and securely.

Appendix B. Costumer Extra Requirements Table

#	Extra Feature	Description
E1	Attractive Body Design	Major: Design and create an attractive body for the prototype.
E2	Second Stepper Motor	Major: Incorporate a second stepper motor in a creative way.
E3	SPI 7-Segment Display	Major: Incorporate the second 4 digits of the SPI 7-segment display in a creative but useful way.
E4	Remote Temperature Sensor	Minor: Add a remote temperature sensor to affect noises if the sensor changes temperature.
E5	I2C Port Optimization	Minor: All I2C modules (if more than one) must be incorporated on the same port.

Appendix C. Costumer Design Constraints Table

#	Design Constraint	Description
1	Physical Fixture	All components must be securely housed in an attractive fixture, including the microcontroller, display, buttons, and cables.
2	Autonomous Operation	The system must operate autonomously without a connection to development tools like Code Composer.
3	Safety and Usability	The user interface must be clear, easy to use, and all human-interfacing components must meet safety standards.
4	MCU Requirement	The system must use the STM32F446 microcontroller on your Nucleo development board as the embedded controller.
5	Watchdog Timer	The built-in watchdog timer on the STM32F446 must be employed to confirm that the program is operating as designed and must be verified without Keil.
6	Budget Constraint	All additional components must stay within a $50 budget, aside from those provided in the lab kit or by the instructor.