THE UNIVERSITY OF TEXAS AT ARLINGTON

COMPUTER SCIENCE AND ENGINEERING

EMBEDDED PROJECT REPORT

**EMBEDDED**

Submitted toward the partial completion of the requirements for CSE 3442-001

**Submitted by,**

Servando Olvera

1001909287

Date 12/05/2023

**Introduction**

The goal of this project is to create an Automated Pet Feeder consisting of a combination of hardware and software components fitted to perform specific functions, just like an embedded system would. In this case, the device needs to have the ability to sense the current water level in the pet dish and automatically dispense water to sustain a desired water level. Along with this, the device should also have the ability to dispense water for short periods of time upon detecting that the pet is nearby, to ensure the freshness of the water. Furthermore, notifying the pet owner through both visual and audible alerts when the water is running low is also an important feature of this device. Moreover, the device is designed to offer customizable feeding events, allowing a user to dispense predetermined amounts of food at specified times throughout the day.

**Theory Of Operation**

The successful realization of this project stems from the efficient combination of hardware and software to a degree in which both of these aspects are able to communicate with one another effectively, yielding an adequate finished product.

The physical components of this project include two small motors, one dedicated to pump water and the other dedicated to driving an auger; a small buzzer which functions as an alert signaling an empty water container; a motion sensor to detect the pet's presence; a customized pet dish equipped with copper strips encircling one of its bowls, functioning as a capacitor to measure water volume; a microcontroller chip housing the controlling code for the above mentioned components; and lastly, a crucial PCB (Printed Circuit Board) which aided in the microcontroller's communication and management with the aforementioned components through its complex circuitry.

The software aspect of this project operates predominantly through user input. The water pump functions as a simple digital output, regulated to dispense water based on three key parameters: the fill mode (auto or motion), user-desired water level settings, and the current water level in the bowl. Conversely, the auger is configured as a PWM (Pulse Width Modulation) output, giving the user the ability of precise control over the speed at which to drive the auger. The buzzer, which again serves as an alert, can be activated or deactivated by the user, allowing for ON/OFF functionality. Also operating as a digital output, it triggers after six minutes of the desired water level not being reached, indicating that the water container has ran out of water. Activated during motion mode, the motion sensor detects the pet's presence and subsequently dispenses water for a short period of time, as a refreshener. Configured as digital input, the sensor is periodically read by the code every two seconds, dispensing some water upon reading a ‘high’ value.

Moreover, on the software aspect, the code enables users to plan feeding events, allowing for a maximum of 10 customizable events. Each scheduled feeding event consists of the following specifications: its identification number (0-9), the duration for which the auger must remain active, the PWM speed setting for dispensing food, and a specific time of the day, in hours and minutes, at which the event will execute. The execution of the event is triggered accordingly thanks to the microcontroller's integrated real-time clock functionality. This convenient clock feature allows users to set, manage, and execute feeding events as desired, ensuring that their pet will be fed.

**Observations**

1. The bowl's capacitance is quite unstable, notably affected by the slightest of movements or any contact with the wires linking it to the PC Board. Factors such as this disrupted the count on Timer 1, resulting in an mL reading being blown out of proportion. Furthermore, the Lab’s Wi-Fi signal also proved to be another source that would disrupt the bowl’s capacitance, causing calibration issues for fellow students, and later impacting my project.
2. Upon booting-up, or upon re-flashing the microcontroller, the initial tick reading is always read as 43, regardless of the actual water level in the bowl. As a result, this triggers the water pump by default due to the logic in my code, since 43 ticks from Timer equates to 0 mL on the bowl. Still unsure why that initial reading is so low.
3. Initially, I overestimated the water pump’s power. I anticipated the pump to dispense water at a fast rate so I configured it to run for 5 seconds, thinking that after five or six runs it would be capable of filling up the bowl. However, the pump fell short of expectations and its dispense rate was rather poor, so revisions and adjustments had to be made in regards to this.

**Conclusion**

All in all, the development of this project represents a successful integration of hardware and software components to create a functional Automated Pet Feeder. Through a clever combination of components which included motors, sensors, a buzzer, a microcontroller, intricate circuitry, and good enough programming, this project aimed to address basic pet needs in an automated manner.

Of course, along the development of this project significant observations were made. For starters, the sensitivity of the bowl's capacitance to external factors like movement or Wi-Fi signals presented calibration challenges which led to inaccurate water level measurements, which then entail making revisions on both hardware and software. On top of this, the initial tick reading discrepancy upon boot-up or re-flashing the microcontroller highlighted a needed work around this issue, since no solution could be determined. Lastly, while the water pump’s dispense rate did not present a significant setback, adjusting the code for an effective functionality required more work than initially needed.

Despite these challenges, the project is still able to successfully achieve its objectives, as an automated pet care system. The effective interactions between hardware components and software functionalities show the importance of a successful integration of the two.