Automated Pet Feeder

A smart solution to automate and regulate
the feeding of common household pets

CMPE 495 Independent Risk Investigation

Microcontroller Options

Name: Chris Ranc
Team Members: Matt Smith
Shabab Siddiq
Submitted: November 7, 2017

Contents

I Overview

Like any embedded system a microcontroller that is cost effective and provides adequate hardware for the requirements of the system is needed. These requirements being to operate the mechanical communication aspects of the feeder. It must have available documentation and easy to use toolchains, with useful SDKs, for efficient development.

II Risk specification

Marketing requirements relevant for microcontrollers:

1. Supplementary and Master Feeders should operate unattended
2. Supplementary and Master Feeders should be able to connect to a wireless network
3. Master Feeder should be able to brodcast localized network for managing Suplementary Feeders
4. Master Feeder should be able to store feeding schedules and the amount of food eaten by owners pets

The relevant specifications can be found in Table 1.

Table 1: Engineering Specifications

Marketing Requirement(s)	Engineering Requirement	Justification
1,2,3,4	A. Suplementary Feeders need to be able to transmit and recieve data to and from master feeder on a private localized network.	Allow for automation of supplementary feeders, without interfering with owners network if any, and to reduce cost for owners of multiple pets.
2,3	B. Suplementary and Master Feeders need wireless pheripherals or to be able to integrate with a wireless module.	Needed to permit network communication between Master and Suplementary feeders.
2,3,4	C. User needs to be able to communicate with Master Feeder via its localized network or the User's personal network.	Needed to allow for user to configure eating times and food portions in pet profiles
1	D. Supplementary and Master Feeders should be able to measure food dispensed and detect pets	Needed to adequately feed pet/s when it is their feeding time.

III Risk investigation

The Master Feeder needs to be advanced enough to manage multiple suplementary feeders with its own localized network, store feeding configuration data, and control the pheripherals of itself. The brands of embedded computer that can suffice for this task are Raspberry Pi, Bananna Pi, and Beaglebone. These brands all have variants that can run an operating system to manage this multitude of tasks. They can run an embedded linux operating system thus allowing for the use of a database for storing information. Each variant also has enough pheripherals for controlling the sensors and motors of the feeder and hava built in wireless for connecting to and broadcasting a network. There were enough differences however to utilize a decision matrix between them. This can be seen in table Table 2 on the following page.

Table 2: Decision Matrix: Master Feeder

Criteria	Weight	Raspberry Pi 3 B	Bananna Pi Zero	Raspberry Pi Zero W	Beaglebone Black
Price	3	0	0	1	-1
Adequate Peripherals	4	1	1	1	1
Wireless Built-in	3	1	1	1	1
Development Resources	2	1	0	1	1
	total	9	7	12	6

Of the devices it came to be that the Raspberry Pi Zero W variant was the best choice. The Raspberry Pi 3 B is much faster and has more cores than the Zero W, but it being 3.5 times the cost of the zero at $35 works against it being cost affective. The Beaglebone not as popular as the Raspberry pi and being almost twice the price of the Pi 3 B at $68.75 made it unideal to choose. The competitor to the Raspberry Pi brand ,Bananna Pi, has its own Zero variant with wireless and operating with a faster quad-core processor. The board however though doesn't appear available on any online shops to purchase or see its price. Which makes it unable to be fully considered.

The Suplementary Feeder simply requires some kind of microcontroller to operate concurrently with controlling the detection of pets, dispensing of food when needed, and communicating with the Master feeder to transmit and recieve data. These tasks can be performed by most microcontrollers provided they have proper connections to appropriate modules or sensors to perform those tasks. The task of communicating with the master feeder via its localized network requires integrating a wireless module. The ESP8266 wireless module by espressif is normally configured to run firmware called nodemcu allowing versatile and cost effective controllers like the KL64 and Arduino Teensy to send command strings via UART to connect to wireless networks. However the ESP8266 can also operate using custom firmware allowing access to its other pheripherals such as its UART and PWM related pins. Allowing it to control the sensors and motors operating the feeder while at the same time utilizing its internal wireless communication hardware to communicate with the Master Feeders network. The network communication is made easier due to the SDK provided for the ESP8266 making the communication aspect more direct. Thus making the ESP8266 the more ideal option for the the Suplementary Feeder to cut out and middle man hardware and save cost.

Table 3: Decision Matrix: Suplementary Feeder

Criteria	Weight	KL64	Teensy	ESP8266
Price	3	0	1	1
Adequate Peripherals	4	1	1	1
Wireless Built-in	3	0	0	1
Development Resources	2	1	1	1
	total	6	9	12

IV Risk mitigation design

Using the Raspberry Pi Zero W and ESP8266 for managing the Master and Suplementary feeders respectively autonumous function of the feeders will be easily met. When fully developed the Suplementary Feeders managed by the ESP8266 will be able request the information pertaining to the Pet it should look out for when it is time to dispense food. As well as return information regarding whether or not the pet came and got its food. The Master Feeder also communicates with the user upon request when changing configurations for their pets feeding times. It will do this through a app while connected to the the units localized network or over the users local network, provided the feeder is connected. A high level diagram of the controllers interacting with one another and their pheripherals can be seen in Figure 1 on the following page.

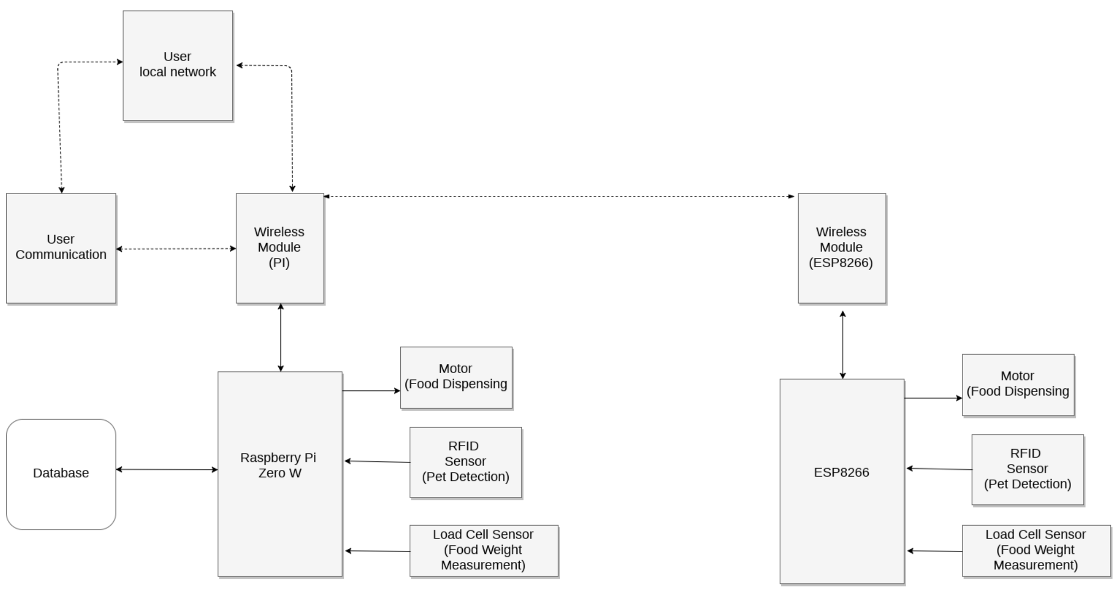

Fig. 1: Controller Subsystems

This design for the feeders operation mitigates their risk due to their builtin wireless capabilities and their various available pheripherals as seen in Figures 2 and 3. Allowing for them operate the sensors and motors inside of the feeder.

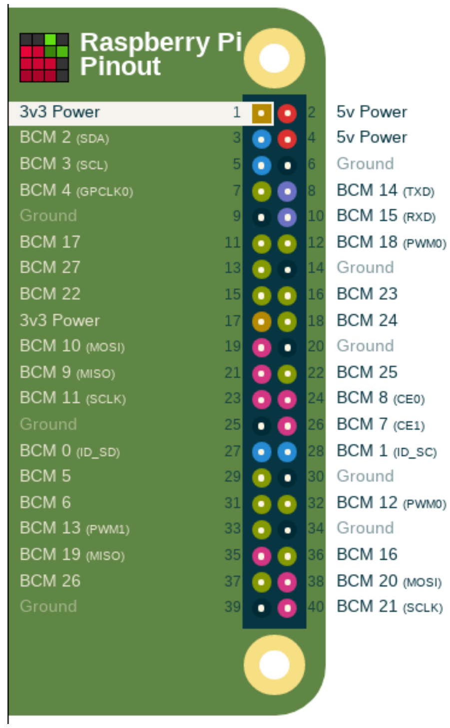

Fig. 2: Pi Zero W GPIO

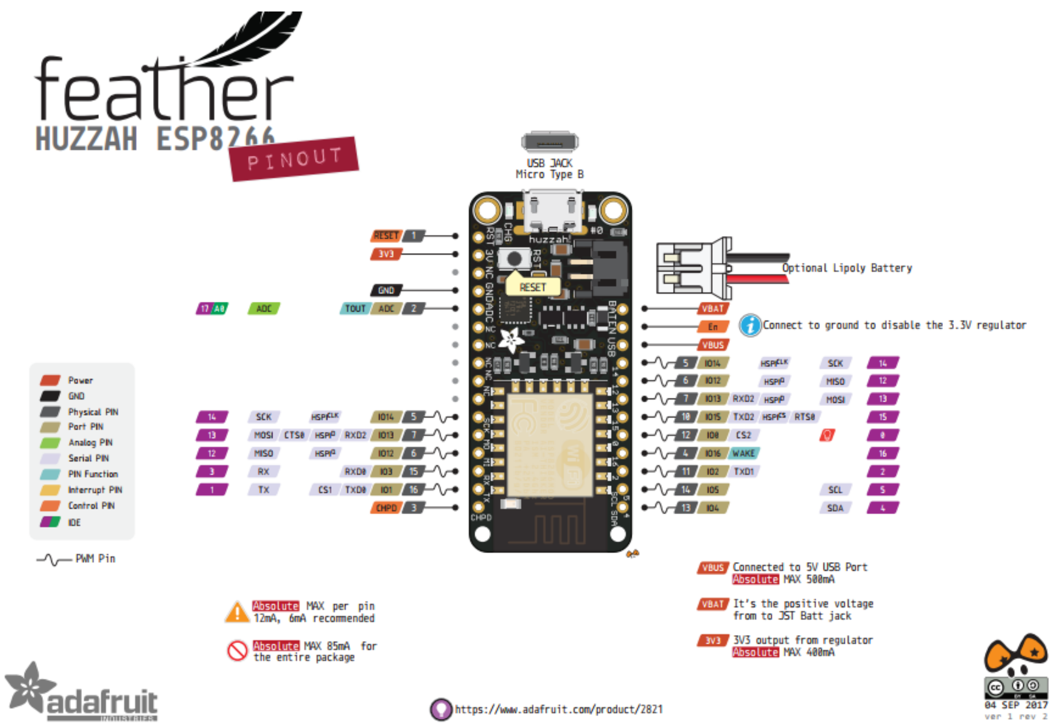

Fig. 3: ESP8266 GPIO

V Parts List

Table 4: Parts List

Part	Description	Cost	Actually Paid	Availability
Raspberry Pi Zero W	A tiny Raspberry Pi with built in Wireless and Bluetooth	\$10.00	\$0 (owned)	Cana Kit Link 3-5 business days
Micro SD Card	SD card for OS and data storage of Raspberry Pi	\$7.80	\$0 (owned)	Amazon Link 2 day shipping (Prime)
ESP8266(Feather)	Adafruit dev board variant of the ESP8266	\$17.12	\$0 (owned)	Amazon Link 2 day shipping (Prime)

VI Testing strategy

VII Uncertainties

Appendices