PROJECT PROPOSAL

PiFeed – Feed Your Pets with a Raspberry Pi

Due November 9, 2014 at 11:55 PM.

Created by:

Danny Duangphachanh: bboydd@vt.edu

Daniel Friedman: adfriedman@vt.edu

Igor Janjic: ijanjic@vt.edu

November 9, 2014

[ECE 4564] Network Applications Design

Virginia Polytechnic Institute and State University

1 Concept of Operations

The purpose of PiFeed is to be able to remotely monitor and control a fish tank and cat feeder from the internet. The system will be composed of two Raspberry Pi's and one computer. Rasp1 will collect information about the fish tank and control the fish feeder. Rasp2 will collect information about the cat and control the cat feeder. A user will interface with the two Raspberry Pi's with a client application and be able to monitor both the aquarium and the cat feeder using Pi cameras. Additionally the client application will allow the user to customize the feeders and control them manually. We will consider the system successful if our pets can be fed remotely. The ability to maintain a healthy eating schedule for pets is a concern for members of our group, as well as many others who travel or are away from home for an extended amount of time. By successfully implementing a pet feeding and monitoring system using Raspberry Pis, the stresses of animal care while away will be a thing of the past!

2 System Overview

PiFeed is separated into the following modules: PiFeedControl, PiFeedFish, and PiFeedCat. The programming language that will be used is Python. Rabbitmq will be used for publishing the sensor information and images from both the fish tank and cat bowls to a message broker that the PiFeedControl client application can subscribe to. The Rabbitmq server will be located on a machine on the same network that Rasp1 and Rasp2 are connected on. Python sockets will be used for controlling the feeders remotely and the LCD connected to Rasp2.

2.1 PiFeedControl

The purpose of this module is to be able to allow a user to remotely access and control the feeders as well as write messages to the LCD display connected to Rasp2. The module is a python client application and can be run using either the local network or the internet. It will talk to PiFeedCat and PiFeedFish by subscribing to a message broker that routes the messages from the two modules. It will use sockets to manually start the feeders and configure the feeders for automatic feeding as well as for writing to the LCD display connected to Rasp2.

2.2 PiFeedCat

The purpose of this module is to publish messages to a message broker containing information generated by the sensors on the cat bowls as well as images from the Pi camera mounted on the bowls. It also controls the hardware for the automatic cat feeder and the LCD.

2.3 PiFeedFish

The purpose of this module is to publish messages to a message broker containing information generated by the sensors on the fish tank as well as images from the Pi camera mounted on the tank. It also controls the hardware for the automatic fish tank feeder.

2.4 Testable Requirements

A completely successful project should demonstrate that the following requirements are satisfied.

1. PiFeedControl is able to remotely feed the fish both manually and at specific times of the day.

2. PiFeedControl is able to remotely feed the cat both manually and at specific times of the day.

3. PiFeedControl is able to remotely monitor the fish using the Pi camera.

4. PiFeedControl is able to remotely monitor the cat using the Pi camera.

5. PiFeedControl is able to remotely view the fish tank sensors.

6. PiFeedControl is able to remotely view the cat tank sensors.

7. PiFeedFish is able to feed the fish.

8. PiFeedFish is able to interface with the sensors on the fish tank and get correct readings.

9. PiFeedCat is able to feed the cat.

10. PiFeedCat is able to interface with the sensors on the cat bowls and get correct readings.
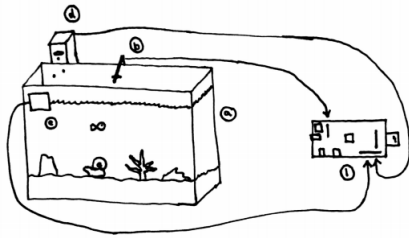
2.5 Use Case Diagram

Figure 1: Diagram of Rasp1. (a) Aquarium. (b) Pi camera used for monitoring fish. (c) Sensors used for monitoring aquarium environment. (d) Automatic fish food feeder.
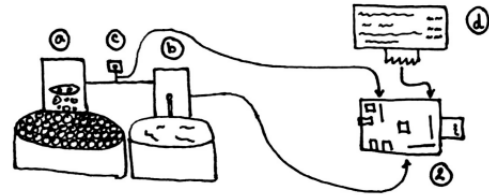
Figure 2: Diagram of Rasp2. (a) Automatic cat food feeder and sensors. (b) Automatic cat water feeder (c) Pi camera used for monitoring cat. (d) LCD display.

3 Hardware List

1. 2 Raspberry Pi boards

2. 2 SD cards (16GB)

3. 2 WiFi + Bluetooth 3.0 USB adaptors

4. 2 MicroUSB cables

5. 2 USB chargers

6. 2 5MP camera board modules

7. Cheap Pi LCD display

8. Jumper wires

9. DS18B20 waterproof temperature sensor

10. AM2302 (wired DHT22) temperature/humidity sensor

4　GitHub Details

Team Member	::	PID	::	GitHub
Daniel Friedman	::	adfriedm	::	adfriedm
Danny Duangphachanh	::	bboydd	::	daduang
Igor Janjic	::	ijanjic	::	zergler

5　Project Schedule

5.1　Development Tasks

1. Get hardware components (Everyone)
2. Set up fish tank and get a cat food and water bowl (Igor)
3. Design fish tank automated feeder (Daniel, Igor)
4. Design cat automated feeder (Daniel, Igor)
5. Build fish tank automated feeder design (Everyone)
6. Build cat automated feeder (Everyone)
7. Connect hardware components (Danny)
8. Write PiFeedControl python module (Daniel, Igor)
9. Write PiFeedCat python module (Danny, Igor)
10. Write PiFeedFish python module (Danny, Igor)
11. Test all hardware component interfaces (Everyone)
12. Test PiFeedControl python module interface (Everyone)
13. Test PiFeedFish (Everyone)
14. Test PiFeedCat (Everyone)
15. Test communication between PiFeedCat and PiFeedFish with PiFeedControler (Everyone)
16. Beta build demonstration (Everyone)
17. Final demonstration (Everyone)

5.2　Milestones

1. Beta build demonstration (12-01-14)
2. Final demonstration (12-16-14)

H

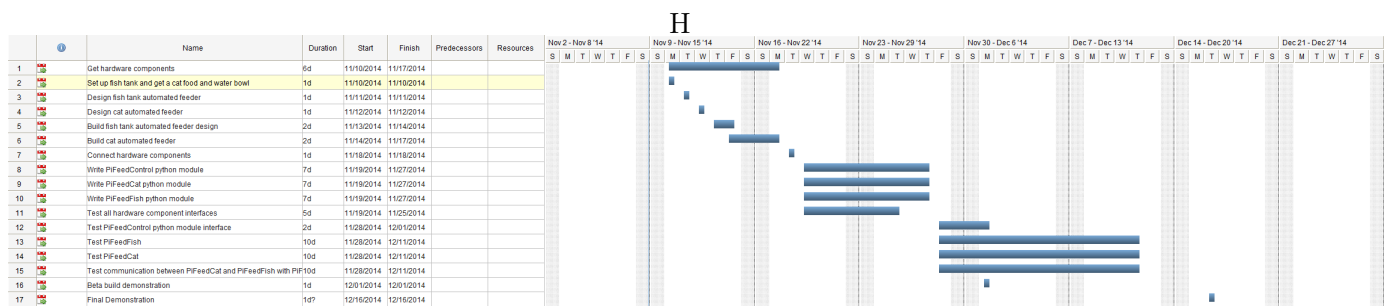|	ⓘ	Name	Duration	Start	Finish	Predecessors	Resources
1		Get hardware components	6d	11/10/2014	11/17/2014		
2		Set up fish tank and get a cat food and water bowl	1d	11/10/2014	11/10/2014		
3		Design fish tank automated feeder	1d	11/11/2014	11/11/2014		
4		Design cat automated feeder	1d	11/12/2014	11/12/2014		
5		Build fish tank automated feeder design	2d	11/13/2014	11/14/2014		
6		Build cat automated feeder	2d	11/14/2014	11/17/2014		
7		Connect hardware components	1d	11/18/2014	11/18/2014		
8		Write PiFeedControl python module	7d	11/19/2014	11/27/2014		
9		Write PiFeedCat python module	7d	11/19/2014	11/27/2014		
10		Write PiFeedFish python module	7d	11/19/2014	11/27/2014		
11		Test all hardware component interfaces	5d	11/19/2014	11/25/2014		
12		Test PiFeedControl python module interface	2d	11/28/2014	12/01/2014		
13		Test PiFeedFish	10d	11/28/2014	12/11/2014		
14		Test PiFeedCat	10d	11/28/2014	12/11/2014		
15		Test communication between PiFeedCat and PiFeedFish with PiF	10d	11/28/2014	12/11/2014		
16		Beta build demonstration	1d	12/01/2014	12/01/2014		
17		Final Demonstration	1d?	12/16/2014	12/16/2014		

Figure 3: Gant chart of the proposed schedule.