**ECE 4220**

**Project Proposal**

**3/28/15**

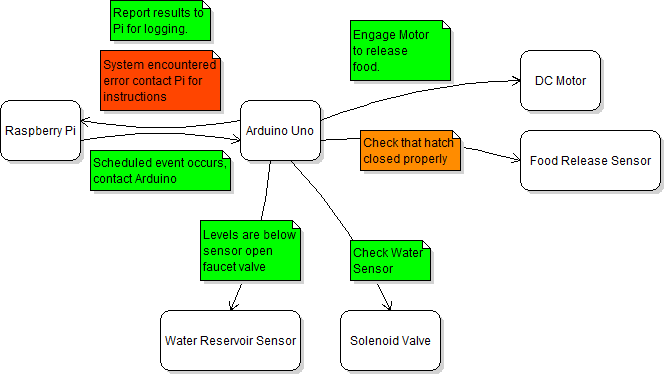
**Kyle O’Day**

Pets are extremely common in households all over the country. Many families or individuals own a dog, a cat, both, or multiple of either. With so many pets in the country it is obvious that caring for them is a very common chore. For specialized care owners will seek the help of professionals such as veterinarians and groomers but daily care is left to the owners. When many lead very active and busy lives being home on time to feed your animals can be difficult. Leaving food out for them is an option but if you are one of the many homes that owns multiple pets, how can you ensure that each animal gets their share without direct supervision. This is my inspiration for this project, which is an automated system that will dispense food and water for an owner’s dogs and cats. The system would be enough for owners of single animals but would require training to be used with multiple animals.

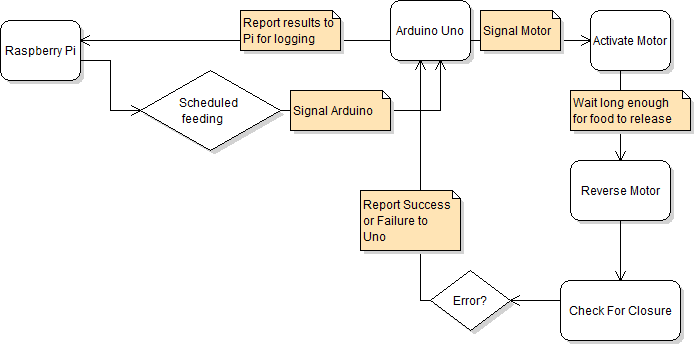
There are many products currently on the market that aim to fulfill these needs for the busy pet owner. In my research I have found multiple automated dog/cat feeders and most of these systems receive negative reviews with very similar complaints, lack of programmability, ungraceful system failure due to jams, and poor structure to prevent the animal from the reservoir. Automatic watering stations for animals have been around for a very long time. These systems consist of a large water reservoir that sits upended over a dish and simple fluid dynamics allows for the dish to be replenished without the reservoir overflowing it.

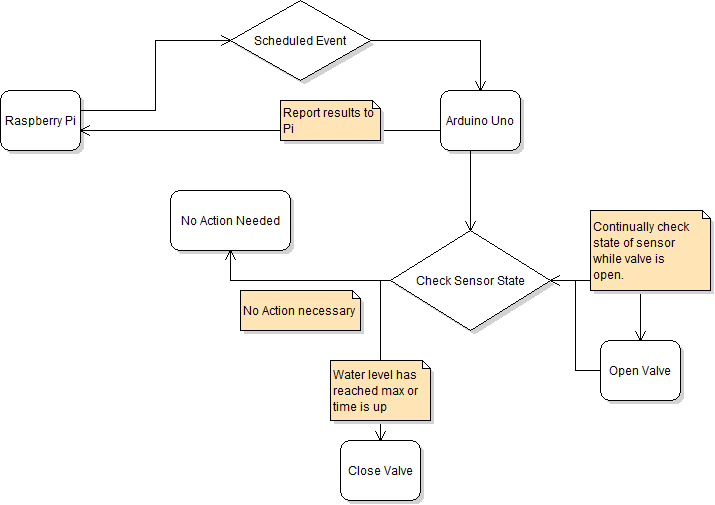
While these systems seem to fulfill the needs of some owners, the system I have envisioned seeks to improve on their shortcomings. While my project will not seek to improve on the mechanical problems of current feeding stations, it will seek to improve upon the programmability of the schedule and the gracefulness of what should happen in the event of a jam. With regards to an automatic watering station, my solution seeks to improve on its singular flaw, the need to be refilled. This improvement will automatically fill the reservoir ensuring that the user will never have to worry about remembering to do so themselves. These improvement will provide peace of mind to their uses and ease the stress of a busy schedule.

The constraints of this system are fairly self-evident. Five major physical components are required for its operation. A small computer system, sensors for the water and food reservoirs, a small motor, a small electric valve, and an optional microcontroller board. The system will consist of a Raspberry Pi, which will communicate with an Arduino Uno microcontroller.



The Arduino will act as an intermediary between the Pi and the systems hardware, sensors, motor, and solenoid valve circuits. When an event is scheduled the Pi will contact the Arduino, to trigger the necessary action. In the event food is to be released the Arduino will sent start the motor and release the food. The motor will then be reversed and the trap sensor will be checked to ensure the hatch has closed. In either success or failure the Arduino will respond to the Pi to store the time of the events and the results.



The second scenario will result in the water reservoir sensor being checked. If the sensor does not read the water the solenoid valve will be opened allowing water to flow. The sensor will then be checked at regular intervals until the water level is registered. When it is registered a signal will be sent from the Arduino to close the valve. In the event it does not register a failsafe will trigger the valve to close after an allotted period of time has elapsed. No matter what the case is all action taken will be reported back to the Pi for logging.

Between the Uno and the motor and solenoid valve there will need to be some circuitry. The specifics of these circuits are currently undecided but they will provide additional power needed for the components, voltage or current switches, and buffering to protect the Uno.

The tasks themselves will be programmed into the Arduino Uno, while the Raspberry Pi will act as a scheduler. The feeding task will be able to be programmed to occur as twice a day exactly twelve hours apart. The task controlling the water flow will be scheduled to occur every four hours since it is not typically a life or death scenario for the animal. The timing of checks to the sensor during filling should be relatively short and based on the size of the reservoir used for the dispenser. For an estimate I would say every one to two seconds. The failsafe time for the valve to open should again depend on reservoir size but should be between thirty seconds and one minute. Logging messages sent from the Uno to the Pi will be in the form of type string and will contain critical information (success, failure, cause) as well as a timestamp.

Overall I believe this may be a very string system due to its simplicity.