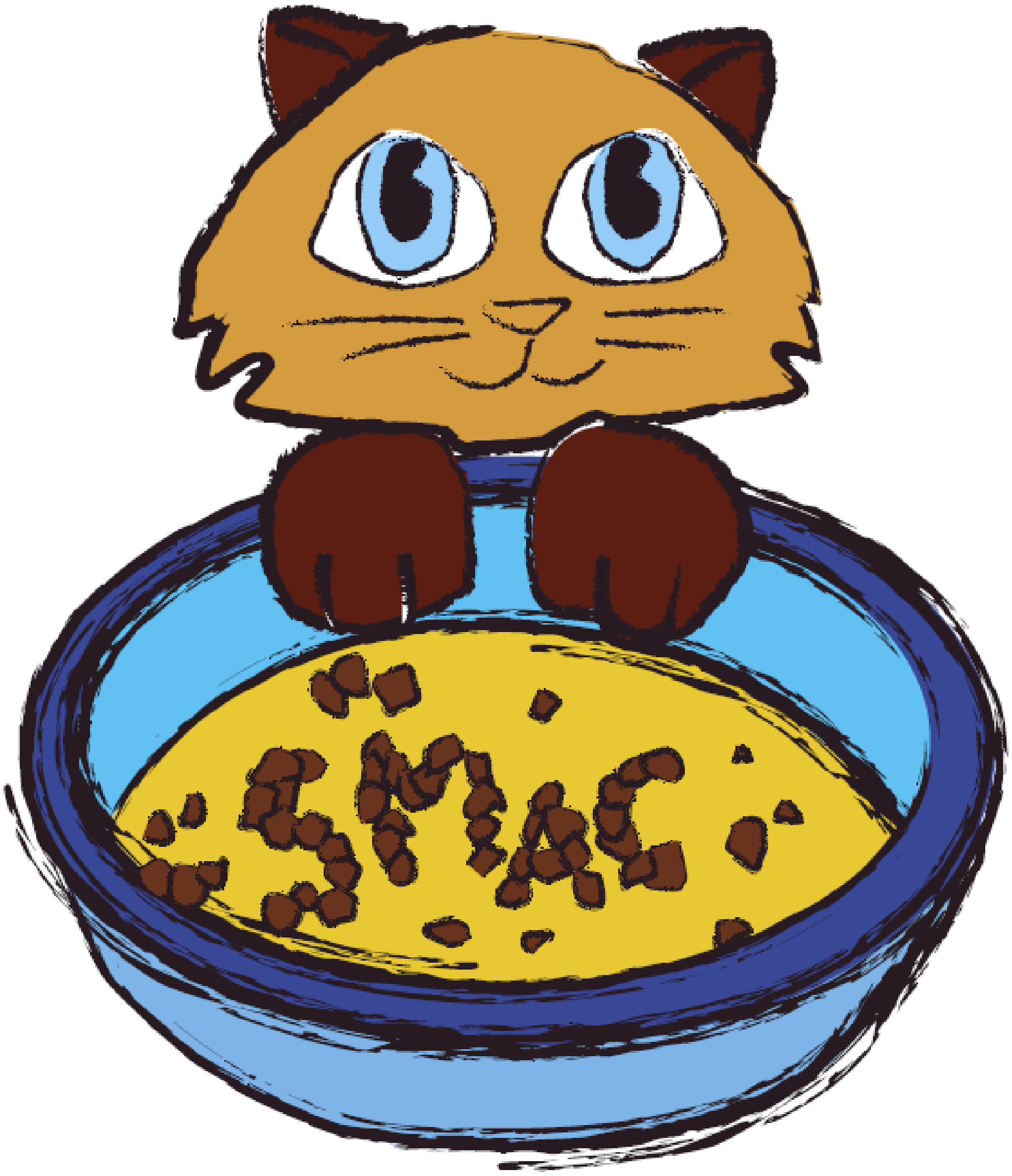

Kristin Soriano
Tim Forkenbrock
Austin Scruggs

Senior Design
Spring 2012
April 23, 2012

Contents

List of Figures

List of Tables

Executive Summary

The Smart Cat Feeder (SMAC) is the next generation of automatic cat feeders. The device will be programmed to each individual cat through a touch interface at the top of the feeder. The user will enter the type of food into the system as well as add each individual cat. The animals will be able to access the food by means of a RFID tag on their collar. When a cat walks up to the machine it will step on a scale and instantly be weighed and recorded. Once the RFID tag is read, the correct amount of food will be dispersed to be eaten. There will also be the option for the owner to be notified by the machine if the food is low, as well as many other options that are listed in the specifications section.

SMAC is a new improved way to correctly feed a cat, or multiple cats with different dietary needs. Previous ”automatic cat feeding machines” have failed because of the lack of complexity during the design stages. It is in this way that SMAC will work better, by combining many different technologies. These include but not limited to: RFID tags, scales, motors, a printed circuit board, and a touch screen display.

The main function of this feeder that separates it from previous attempts is that it will keep track of the daily calorie consumption and never overfeed the cat. Almost all other ”automatic cat feeding machines” only disperse a set amount of food at a time, but do not have a limit. This means that theoretically a cat can continuously get food from the machine. By keeping track of the food intake of the cat, this will never happen. To accurately keep track of the amount of food eaten there must be a scale in the bottom of the food dish to measure what is left after the cat leaves the dish.

Do to the nature of measuring the food left in the dish, a similar technique must be used to disperse the correct amount to the dish in the beginning. The average cat is around eight to nine pounds, making the correct daily calorie amount about one-third to one-half cup. This correlates to about three-hundred calories. Not all cat foods have the same nutritional values and not all cats are the same size. Due to that and the fact that not all cats eat the same food these exact measurements would not be correct. This can be avoided with a scale system that is in the food dish. By knowing the calorie count in the cat food, it is then possible to calculate the correct oz of food needed based on the suggested calorie count per cat. This count is around twenty to thirty calories per pound. The food can then be weighed out for a nine pound cat and a twelve pound cat using the same machine, with no inconvenience to the user.

1 Descriptions

1.1 Motivation

The common household cat can come in a variety of ”flavors”. There are currently over 80 different breeds of household cats, and within that every cat has a different personality. Cat owners have been faced with the problem of how to properly feed them when there are multiple cats in the same house. One cat may eat only twice a day, three times a day, or even small bits throughout the day. There are even cats that will eat everything that is given to them, which can lead to cat obesity. There is also the issue of going out of town. Many companies have food dishes that hold up to three days of food, but if the cat were to eat it all in one day then the cat will starve for the next two days.

What would be better than a machine that would take care of all the issue listed above? That is where the SMAC Feeder came about. There is a need for a machine that will not only feed cats when the owner is out of town, but also deal with the individual dietary needs of each cat. Many of the cat feeders currently on the market only appeal to one set of problems, either timed food release, or only when the cat comes up to the machine. They do not keep track of how much food is consumed by the animal, let alone multiple cats. There is no option to have different amounts of food per individual cat. The SMAC Feeder will monitor the cats weight and food intake, have the option to set the dietary needs of each cat, and notify the owner when something is wrong.

The main motivation for the project stems from the following scenario. The owner of two Siamese cats, Lea is a female age nine, Cappy is a male and is less than a year old. Lea is the type of cat that picks at her food all day, so food must be left out for her or she gets angry, or won’t eat. Cappy will eat all day, thus eating any food left out for Lea. To allow this owner (and many others) to make each cat happy, a solution would be a machine that will allow an owner with multiple cats to feed each pet differently, based on the personality, weight, and breed.

1.2 Goals and Objectives

This project consists of many different components and thus many goals and objectives. Most of these goals and objectives are specific to the many different components of the system.

Cost Efficiency - SMAC does not have an overall cost limit, although because it is simply ”an automatic cat feeding machine” the cost has to be reasonable to the normal cat owner. This being said, it is considered a high-tech feeder and will be a little on the high end. When choosing parts for the machine, price will play a very important role in selection, but the specifications will ultimately come first.

Power Efficiency - To keep the cost down when using the product the power efficiency should be minimal. Unfortunately to use RFID tags in the system, the transceiver/transponder must always be on to detect when the cat arrives. This will require the use of an outlet versus batteries. The other parts of the system, such as the touch screen and motor, can be chosen and used in such a way that limits there power consumption to offset that of the RFID power consumption.

Wireless Communication - In order for SMAC to be able to notify the user of any changes in the system, and access the database of information, there must be some sort of internet connection. To keep from limiting the machine placement in ones house, a Wi-Fi device will be used. The Wi-Fi device will require the use of a circuit board that can integrate this as well as be cost and power effective as listed above.

Database System - The use of a database system with SMAC is essential to the goal of cost efficiency and power consumption. The database will be used to store the majority of the data used to determine how the cat is fed, as well as the data collected from the cat on a daily bases. Using a database will reduce the on board memory needed thus reducing the cost of supplies and allowing the printed circuit board to use less power. Due to the different types of data being stored and retrieved from the database one goal is to make sure that the information is stored and retrieved in the correct format, and with no logical errors. Along with making sure the data is correct the links between the data must be set up in a way to minimize this error and maximize the speed. It is also important to make the database as secure as possible, so that no information can be obtained illegally.

Notifications - A very important aspect of SMAC is to allow the user to keep track of the cats eating habits. This goes in hand with the database, such that the data must be accurately stored so that when retrieved and analysed the owner can be notified properly. One goal of notifications is to allow the option to be notified through an Android application, or a text messaging service for the non-android users. This goal allows the range of users to be broader than limiting it to only Android users. The goal is to allow the notification of the amount of food left in the machine, the cats average weight for the week, and any change in food consumption.

Cat Scale - The main goal of the cat scale is to weigh the cat properly. To do this the scale needs to be smart enough to know when the cat is fully on the scale, due to the fact that cats don't know to step all the way onto the platform. This is important because the weight could be wrong if it is taken before checking if the cat is completely on the scale. Unfortunately this goal directly affects the cost efficiency wanted. The cheapest simplest scale will not achieve the desired result, and therefore a few more expensive options must be explored and chosen based on accuracy, price, and power consumption.

RFID system - This is a very key aspect of the machine. If this system does not work properly then it is possible that the cat cannot eat, or the cat gets the wrong amount of food. Thus the main goal of the RFID system (this includes the tags, transponder, and transceiver) is to accurately scan for and read the RFID tag on the cats collar. There is also the goal to correctly relay this information to the rest of the system.

Food Tray - The food tray is also a very important part of the project. The tray must accurately weigh the food placed into the tray from the dispenser as well as the left over food after the cat has eaten. It must also send this information correctly back to the main circuit board. The data will be be sent to the database and/or used to send a signal to stop the food dispenser. If the weight is not accurate enough then the whole premise of the machine is useless, per the fact that the cats will not be getting the correct amount of calories recommended per day. Another very important goal is the accuracy of the motor used to turn the food tray to the appropriate opening when the cat comes up to eat. It will have to turn the tray to the correct position to allow the food dispenser to disperse the food, weight the food, and then allow the cat to eat the food.

Food Dispenser - The food dispenser is just as important as many of the other pieces of the machine. It must be able to disperse food without jamming up. The food dispenser must also be able to disperse the food accurately enough to stay within the allowed percentage of error for the weight of the food. This will be achieved with the correct design as well as adhering to the goal of receiving the stop signal from the main printed circuit board and stopping the dispersion of the food.

Touch Screen Display - This display must be user friendly so that the user will want to use the product. A goal is also to make sure the user can easily learn how to use the system and maintain this knowledge over a period of none use. It should have a very similar display to what the android application will look like.

Main Printed Circuit Board - There will be more than one printed circuit board (PCB) in this project. The goal of the main printed circuit board will be to control the board that controls the motors and scales accurately. It will also have the goal of managing the power, touch screen and Wi-Fi capabilities.

1.3 Specifications

1.3.1 Overall Specifications

The system shall conform to the following specifications in a whole. These specifications, shown in Table 1, are then broken down into a few major parts of the system: the RFID system, the touch interface, the main printed circuit board, the food tray, the food dispenser, the software, the database, and the notifications.

The physical system shall take up no more room than a two foot by five foot space.
The system shall be composed of a scale to weigh the cat, a scale to weigh the food, a tray to hold the food, a printed circuit board with the Android Operating System, a Wi-Fi system, the ability to control one or more motors and an LCD touch screen.
The LCD touch screen shall be large enough to be user friendly.
The LCD touch screen must be placed in a position on the machine that allows ease of use.
The motor that spins the food tray shall be wired as to not interfere with the spinning of the tray.
The system shall run on an everyday household outlet
The system shall consume only twelve volts of power.
The system shall have Wi-Fi capabilities to be able to talk to the server.
The mobile application and LCD touch screen shall use the same interface for ease of use.
The PCB and mobile application shall use the Android Operating System to allow ease of coding for the touch interface.
All components of the system shall come together in such a way as to minimize space and appeal to the eye.

Table 1: Overall Specifications

1.3.2 RFID Tags

The RFID system is used to know when and which cat has walked up to the machine to eat. This information is sent to the main printed circuit board to look up the correct information about the cat and then uses this information to measure out the correct amount of food. Table 2 shows the RFID tag specifications.

The system shall be able to read a RFID tag set at 13.56 MHz.
The system shall be able to look up the correct cat in the database.
The system shall allow the addition of the cat if it is not already in the database.
The system shall allow the deletion of a RFID tag from the system.
The system shall allow the change of a RFID tag in the system.
The system shall be able to add the new information to the correct cat in the database.
The system shall scan for a RFID tag every thirty seconds.

Table 2: RFID Tag Specifications

1.3.3 Cat Scale

The cat scale is used to determine the weight of the cat during food consumption. This part of the system allows the monitoring of the cats weight. The specifications are listed in Table 3.

The system shall recognize when all four paws are on the scale.
The system shall know when to take the weight of the cat.
The system shall accurately weigh the cat within one percent error.
The system shall send the correct information to the main printed circuit board.
The system shall use the same power that is used to power the entire system.

Table 3: Cat Scale Specifications

1.3.4 Food Tray

The food tray is responsible for weighing and keeping track of the food for the cat and rotating to allow the cat to eat the food. It consists of a motor to turn the tray, a scale to weigh the food, a circular tray to hold the food dishes, and removable plastic food dishes. These specifications are listed in Table 4.

The scale in the system shall weigh the food to within one percent error.
The motor in the system shall turn the tray to within one percent error.
The plastic food dishes shall be easily removed and dishwasher safe for ease of use.
The system shall integrate with the cat scale and food dispenser without any errors.
The system shall send the collected information correctly to the main printed circuit board.
The system shall receive and execute the correct signals from the main printed circuit board.

Table 4: Food Tray Specifications

1.3.5 Touch Interface

The LCD touch screen is where the touch interface is located. It is on top of the feeder with easy access to the user. It allows the user to add and delete cats; add and delete the type of food in the machine; monitor the cats weight and eating habits; reduce or increase the recommended calorie consumption per cat; connect to a Wi-Fi hotspot; and set notification options. Table 5 shows these specifications.

The system shall use a touch screen for ease of interaction from the user.
The system shall have a login screen so that each users' information stays separate.
From the login screen the user shall have an option to create a new account.
The system shall always display the different pages available for viewing.
The system shall always show what the current page is.
The system shall allow the addition of a new cat.
When on the addition page, the system shall allow the following options: The scanning of a new RFID tag Select from an existing RFID tag in the system Selecting type of food from the database Adding a new food to the database Setting the cats name/weight/age
The system shall allow the deletion of a cat.

Table 5: Touch Interface Specifications

1.3.6 Food Dispenser

The food dispenser will consist of a mechanism to properly disperse food into the food tray to be properly weighed. The food dispenser specifications are listed in Table 6.

The system shall not allow food to block the hole where the food is dispersed.
The system shall instantly stop dispersing food when given the signal from the main circuit board.
The system shall use the same power that is used to power the entire system.
The system shall allow the placement of the main circuit board and LCD touch screen at the top.
The system shall be no more than three feet tall.
The system shall hold at least seven pounds of dry cat food.

Table 6: Food Dispenser Specifications

1.3.7 Wireless System

The wireless system is mainly used to communicate to the database. It is also allows the option for a text message to be sent to the user if they do not own an android, and therefore cannot use the android application. This system will connect to the users wireless network in the home and upload and receive information from the database. The specifications are shown in Table 7.

The system shall connect to a Wi-Fi spot.
The system shall send and receive information correctly through the database.
The system shall allow the sending of a text message to the users phone.
The system shall integrate to the main printed circuit board.
The system shall use the same power that is used to power the entire system.

Table 7: Wireless System Specifications

1.3.8 Database

The database is what will hold all the information about the different types of cats, their calorie count, average weight, and recommended daily calorie intake. It will also hold the information of many different types of cat foods. The last function will be to hold all the information collected daily from the machine. The specifications for the database are listed in Table 8.

The system shall organize the information in a neatly fashion.
The system shall allow the users to create and delete an account.
The system shall make the user login before accessing any information.
The system shall allow the saving, sending, and retrieval of information.
The system shall be secure to the average user.

Table 8: Database Specifications

1.3.9 Notification System

The notifications system is responsible for notifying the user of any changes. These could include: the machine is low on food, a potential problem with the RFID system, a potential problem with the food dispersion, and the average calorie consumption for each cat. The user can elect to be notified on which information they think is most important as well as when and how they want to get notified. Table 9 shows the specifications that will allow this notifications system to work properly.

2 Research

2.1 Similar Products

Upon researching similar products to SMAC three different types of cat feeders were found. There is a type of feeder that represents a food dish with a lid that opens when the cat approaches. Another basic type holds significantly more food and disperses a pre-set amount the food into an open dish based on a timer. The

The system shall allow the option to be notified of a certain event.
For each event the system shall allow the option of the frequency of notification and how they want to be notified.
The system shall update the Android application as well as the touch interface with the same information.
The system shall monitor the food levels to allow for plenty of time for replenishment.
The system shall recognize when the RFID system is not working and notify the user.
The system shall recognize when the food is blocking the food dispersion system and notify the user.

Table 9: Notification System Specifications

last type of feeder that was found is most like the SMAC because of the complexity of monitoring the cats.

Even though there are different types of automatic cat feeders they all share a few common features. Almost every feeder is fully cleanable, this meaning the parts that hold the food are able to be washed separately from the circuitry. This is a very important feature, without it there is more of a chance for the owners cat to get sick. The design of SMAC makes the food dishes removable so that they can be washes properly. The prices range from the simplest machines in the $20 range to the ones with the most options in the $200 range. This means that SMAC will be on the more expensive end when it comes to similar products.

2.1.1 Opens Lid

One type of similar products previously made holds a set amount of food and opens a lid when the cat walks up. The Autopetbowl by Autopetfeeder accomplished this task with an infrared proximity sensor. This feeder is an example of a food dish with a lid on it. The only purpose of this device would be to keep dirt, bugs, other animals, or any other foreign object out of the cats' food. It is currently in the market for a reasonably cheap price of under $35. If the owner does not want to purchase the pet feeder, there are instructions online to make the same type of machine.

The "RFID pet feeder" as its name suggests uses an RFID system instead of an infrared proximity sensor. Although the feeding concept is the same as the previous example given it is more like SMAC as it uses the RFID tag to identify the cat. Since this particular cat feeder is not on the market for consumers the design is not as nice as the previous one mentioned. The body is made of cardboard while the lid is an old cd tray. Both of the systems mentioned in this section were made to simple cover one bowls worth of food, equivalently one days worth of food.

The antenna for the RFID reader is made in such a way that the cat must stick its head through a loop. Some cats will be afraid of this contraption, thus SMAC will not make the cat stick its head in anything to get food. Rather the antenna will be placed in such a way that the cat will simple need to walk up to the machine normally. SMAC will be an improvement over the Autopetbowl and RFID pet feeder because it will allow multiple days of food and the feeding of more than one cat.

2.1.2 Timer

The second type of product that was looked at released a set amount of food by a timer. The Lentek Automatic 6 Day Pet dish is similar to the previous machines, were it already had the food in the bowl and a lid was just opened. The difference being that this time the lid opens based on a timer versus an RFID tag or infrared system. This feeder only holds about two pounds of food, which would last, for one cat, a maximum of six days. SMAC will not only work for two cats, but since it can hold seven pounds of food, it could potentially feed an average household cat for up to three weeks.

Another thing the SMAC will make sure is that unlike any type of cat food will not get caught in the machine preventing the cat from eating. One feeder that does not do that is the Petmate Le Bistro Portion-Control Automatic Pet Feeder by Petmate. There is a clear warning on the instructions about the which best food diameter will work best. This could cause an issue had the user not known this, possibly starving the cat. The best cat feeder found to properly feed a cat was the Automatic Pet Feeder by AutoPetfeeder. It not only holds five pounds of food, but it was also made to minimize the chance of the food getting caught upon dispersion.

2.1.3 Wi-Fi

The type of cat feeder that is most like SMAC involves the use of Wi-Fi to feed and monitor the cat. The only system that was found on the market that currently does this is the Remote Pet Feeding & Viewing Camera Kit by: Smarthome, Panasonic and Ergosys.net. Unfortunately this system was discontinued, suggesting that it did not do well on the market.

One thing that makes this system different than SMAC and most likely its downfall is that the owner needs to connect the machine to their computer to program the feeding times and amounts. Not only would the user have to be computer savvy to install and use the program, this is a huge inconvenience. To use SMAC the user will only have to know how to use a touch screen that is on top of the machine. This widens the customer audience that would purchase the product since much of the population already uses a touch screen on a daily bases. Overall the feeder mentioned is not very user friendly to the common household cat owner. It is in this way that SMAC will make an huge improvement.

While the previous feeder mentioned was the only one of its kind found on the market, there have been at home attempts at something similar. The Internet-Enabled Cat Feeder is a perfect example of this, and probably even a bit better. Like the at home feeder that was mentioned in section 2.1.1, it is not as appealing as the one that was on the market. While SMAC will have a built in wireless card this feeder simple has an entire router mounted on the back side. This makes the feeder too large for everyday use, which defeats the purpose of having one. This feeder does have a mobile application to monitor the cats, it is written for a Windows Mobile phone. SMAC will have a similar feature, but for an Android system.

2.2 RFID Transceiver/Transponder

The RF reader that was compatible with the RFID tag was the 13.56MHz RFID Module IOS/IEC 14443 type A. The MFRC500 R/W chip provides outstanding modulation and demodulation for passive contactless communication. The RFID reader will be connected to the PCB and will be mounted on the plastic external housing. It must be in a location that is easily accessible for the RFID tag to be scanned in.

It is very important that the RFID reader is easily accessible for the tag because the cat needs to be able to access its food. If the owner is not home and the cat needs to be fed, there is a huge responsibility that the SMAC takes on. There can be no room for error and there must be 100% accuracy for the scan ability of the reader.

Figure 2.1: 13.56 MHz RFID Reader

The tag will be hanging from the cats collar so the optimal read range would be towards the opening slot where the cat can access the food. Once the reader receives a signal from a tag, it is stored in a temporary location on the RF reader. The RF reader then sends that signal to the stepper motor to be translated into number of rotations. The number of rotations applies to the specified cats food tray

Item	Specification
Operating	Frequency 13.56 Mhz
Standard	ISO/IEC 14443 type a
R/W chip	MFRC500
Baud rate	9600-115200 bs
Power Supply	5V DC
Current	less than 70 mA
Operating Range	30-100 mm
Board Size	39 mm * 19 mm * 9 mm
Antenna size	54.2 mm *72.7 mm; Wire 80 mm

Table 10: Figure 2.1 Specifications

to give that cat the correct amount of food.

The RF reader that is compatible with the 125KHz RFID tag is the RDM630 125KHz RFID reader module. The big advantage to using this reader is that it supports an external antenna. If this is the reader chosen for the SMAC, then an external antenna will most certainly be made wide enough to ensure that there will always be a 100% readability of the RFID tag. Without an external antenna the maximum effective distance is 150 mm, so an external antenna would need to be made to increase the maximum effective distance to well beyond 150 mm. Another advantage to using this reader is that the decoding time for this reader is less than 100 ms and it has a built-in external bi-color LED and buzzer driver.

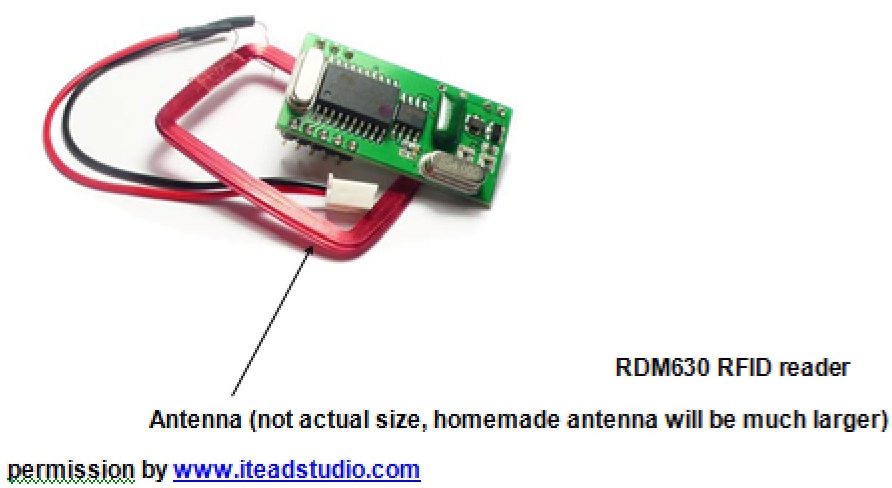

Figure 2.2: RDM630RFID Reader

The antenna for the reader will need to have a good read distance in case the

Item	Specification
Operating	Frequency 125 Khz
Baud Rate	9600
Interface	Weigang26 or TTL Electricity Level RS232
Power Supply	5V DC
Current	less than 50 mA
Operating Range	greater than 50 mm
Size	38.5 mm * 19 mm * 9 mm

Table 11: Figure 2.2 Specifications

cat does not step up on the scale close enough to the SMAC and risk the chance of not scanning resulting in the cat not receiving its food. The larger the antenna the better, for optimal read range. The position of the antenna is also very important. The antenna for the 13.56MHz RFID Module IOS/IEC 14443 type A will be tilted between 70-80 degrees to be angled up towards where the tag will be hanging from the cats collar. Since the operating range is max of 100 mm, the angle will need to perfect so the RFID tag will be within the threshold. Sometimes the RFID tag on the cats collar will not be perfectly facing the antenna.

This creates the option to build an RF antenna. To build an antenna, a 125 KHz RF reader would have to be used that allows an external antenna(such as the RDM630). The external antenna could then be any length and shape to maximize read potential of the RFID tag because even if the cat is in the same place every time, if the RF reader has a small read range, it may not always scan the RFID tag.

One of the ways to make an antenna is to take 24 gauge magnet wire and wrap it in a coil with the two ends connected to the reader. The coil will be placed around the food opening in a circular shape for maximum reading potential of the RFID tag when it is hanging from the cats collar.

2.3 Proximity Sensor

A proximity sensor will be mounted on the plastic external housing where the opening is for the cat to access the food. The proximity sensor works by emitting an electromagnetic field and then looking for any changes in the field. If the return signal detects a change in the field, then it is determined that an object is in close proximity.

The proximity sensor will either be inductive, capacitive, or infrared. An inductive proximity sensor requires a metal target for detection. To use an inductive sensor the cat would have to have metal tags hanging from the collar which would pick up on the inductive sensor when the cats head is down in the bowl eating.

Figure 2.3: Sharp GP2Y0A02YK0F Analog Distance Sensor

However since the inductive sensor can only pick up on metal, there is a risk of non-detection even when the cat is within range which would result in the cat not having full access to its food. A capacitive proximity sensor can detect a non-metal object like for instance a plastic tag. To use a capacitive sensor would be more ideal because it can pick up on many other objects which increase the accuracy of detection when the cat is within range. The disadvantage to inductive or capacitive proximity sensors is that in order to get the desired sensing distance greater than 25mm (1 in) it becomes very costly.

With a sensing distance of approximately 1 inch, the cats tag would have to be very close to the sensor in order to be read. An infrared proximity sensor emits an infrared light by applying a voltage to a pair of IR LEDs. When the sensor receives a strong reflected light, the sensing unit then becomes active. Upon this activation, a corresponding signal is sent to the output terminal which will be sent to the PCB to hold the motor from rotating the trays. The infrared proximity sensor is the best choice for the SMAC because it has the biggest sensing distance for the best price.

The proximity sensor chosen for the SMAC will be the Sharp GP2Y0A02YK0F Infrared Proximity Sensor Long Range. This sensor is a distance measuring sensor unit composed of an integrated combination of PSD (Position Sensitive detector), IRED (InfraRed Emitting Diode), and a signal processing unit.

The sensor will be used primarily for two main functions, 1) preventing the food from being rotated away while the cat is still eating and 2) preventing other cats from eating the leftover food when another cat walks away from the SMAC when it

Item	Specification
Operating	Voltage 4.5 to 5.5 V
Average	Current consumption 33 mA
Distance Measuring Range	20 cm-150 cm (8in-60in)
Output type	Analog voltage
Output Voltage differential over distance	2 V
Response time	38 plus/minus 10 ms
Weight	0.17 Oz (4.8 g)

Table 12: Figure 3.36 Specifications

Figure 2.4: Sharp GP2Y0A02YK0F Analog Distance Sensor

is done eating. The sensor will be mounted on the external housing in a way that when the cats head is down into the bowl to begin eating, it will be at a distance that makes the output reliably cross the threshold of readability (the cats is not to close or far away from the sensor). Any threshold too high or too low can provide false positives and cause the SMAC to not work properly. The first function of the proximity sensor is to prevent the food from being rotated away while the cat is eating. When the sensor detects the cat, the system holds the trays from rotating allowing the cat full unlimited access to the food. Once the cat completes the meal and walks away, the second function of the proximity sensor is activated. Once the cat is out of range from the sensor, the trays will rotate back to the blank tray. This function prevents other cats from eating the leftover food when another cat walks away from the SMAC when it is done eating.

2.4 RFID Tags

RFID Tags will be used to be attached to the cats collar for convenience and easy access scanning while approaching the food bowl. The size of the tag influences the read range so the bigger the tag the better. There a low frequency RFID tags(125KHz) and high frequency RFID tags(13.56MHz). Both frequencies are considered passive because there is no onboard power source. Typical applications of HF RFID tags are ones that require a read distance of less than three feet and work better on objects made of metal. LF RFID tags can better penetrate non-metallic substances but typically have a read range of less than a foot which provides the option for adding an external antenna.

Researching several different RFID options, one option is the Milfare-One RFID Tag. It is widely used in customer object identification i.e. SMAC identifying which cat it is to give it the correct amount of food. An advantage to using this tag is that it can be read by almost any 13.56 MHz RFID/NFC reader that can handle Milfare cards.

Figure 2.5: Milfare-One RFID tag

Item	Specification
Operating Frequency	13.56 MHz
Physical size	3.5 cm x 2.8 cm
Baud Rate	106 kbs
EEPROM storage	1K Byte
Variable identification distance	0-5 cm
Service life	10 years
Service time	100,000 rewrites

Table 13: Figure 4.1 Specifications

Another option for the RFID tag is to use a SeeedStudio RFR103B2B 125khz RFID Tag which is compatible with the RDM630 125KHz RFID reader module. This tag has very similar characteristics in size and type than the 13.56MHz RFID tag described previously. The only differences in these two tags are the operating frequencies, as one is high and the other low. The advantage of using this low frequency tag is that it is compatible with the RF reader that allows an external antenna. From thorough research, it seems that an external antenna will certainly be needed in order to make the SMAC more accurate and reliable on scanning.

Figure 2.6: RFR103B2B 125 KHz RFID Tag

Item	Specification
Operating Frequency	125KHz
Type	Contactless Read and Write
Material	ABS
Compliance	EM4100/EM42100
Physical size	3.5 cm x 2.8 cm

Table 14: Figure 2.6 Specifications

Choosing an RF reader/transponder that is compatible with the tag and has a big read range is the optimal solution for the SMAC. The RFID tag is mounted inside a plastic shell in the shape of a key fob that has a ring looped through it so it can easily be attached to the cats collar. The hanging tag makes it very convenient because the cat will not notice it. It is also convenient because the tag can easily be read by the RF reader that will be mounted to the top of the food container for maximum read range.

2.5 Motor Shield

The Adafruit Motor/Stepper/Servo shield for Arduino kit will be necessary for controlling the stepper motors. Original design considerations were for using a micro stepping drive however they are costly and two would have to be purchased. Having two separate cards only make things more complicated. To make things run more smoothly, the motor shield is ideal because it can control two stepper motors at once. This function eliminates parts and reduces the total cost of the machine making the SMAC more appealing to the users.

The shield will be connected to the PCB as an input device and is able to send separate control commands to each motor controlling them each individually. By having both motors being controlled by one unit and run on the same interface, consistency and accuracy is maintained throughout the entire usage of the SMAC. The motor shield can also power up to 4 DC motors, so if the DC motor is more suitable for the SMAC then this same board will be able to control it. The motor shield has the capability on controlling one stepper motor and one DC motor at the same time as well.

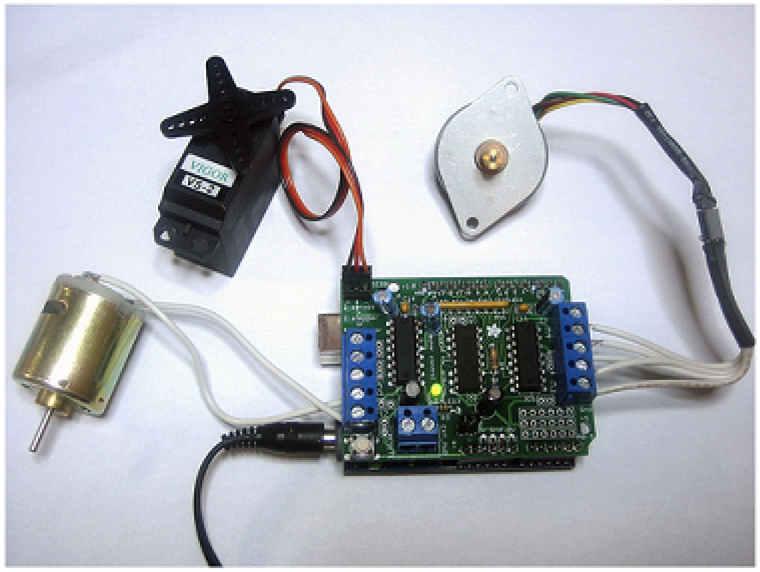

Figure 2.7: Adafruit Motor Shield

Can power up to 2 stepper motors, 2 servo motors, 4 bi-directional DC motors
4 H-bridges which provide thermal shutdown protection, internal kickback protection diodes, and can run motors between 4.5 VDC to 25VDC
Pull down resistors keep motors disabled during power up
Big thermal block connectors to easily hook up 18-26 AWG wires and power
2-pin terminal block and jumper to connect external power
Tested compatible with Arduino Mega 2560

Table 15: Figure 2.7 Specifications

2.6 Food Dispenser

The Food dispenser will consist of a dispensing lever and a funnel. The funnel will be mounted at an angle in order to funnel the food to one side. The funnel acts as a support mechanism for the sliding lever by taking the weight of the cat food directly off of the sliding lever. This also gives the slider an advantage by only having a small sliding distance for opening and closing. The lever part will consist of a sliding mechanism which will block/allow the food. It will be connected to the food scale mounted beneath it. When it receives a signal from the RF reader the bar slowly opens allowing food to be dispensed into the tray.

When it receives a signal from the scale that the amount of food desired is weighed out, the bar closes blocking the food. It will take several trial runs to determine the amount of speed at which the bar will slide open and closed to disperse the correct amount of food. The bar will not begin to slide closed until it receives the signal from the scale. We will account for this delay when determining speed. We do not want any slightly less food getting through if the speed of the sliding bar is too fast. Also, we do not want more food slipping through if the speed is too slow. The bar will have to be powered so it will not get food wedged in the middle when closing. The advantage of this option is that an exact amount of food is able to be measured as it is slowly released from the container. The disadvantage however is that by having a simply slider that operates slower, this increases the chances of food getting wedged while trying to close.

Another option for the food dispenser is to have a turn style type device. This would resemble the candy dispensing machines that dispense a certain amount of food upon rotating the metal lever. It would be in the shape of a square and have four equal containers that turn and each turn would be a 1/3 cup amount of food. This would require an electronic motor for rotating the trays however and food could only be measured in 1/3 cup. Each turn would be a ninety degree turn. This allows the top container to be filled up then rotated ninety degrees to a storage location while at the same time the new container on top gets filled. Upon the third rotation, the first container filled is now dispersing the food into the tray, the second container is now in the storage location while the third try is on top being filled. The advantage of this technique would be accurate amount of food in 1/3 cup measurements and quick precise turning with no chance of getting food wedged. Thee disadvantage however is that it is more complicated and requires more power and parts.

2.6.1 Motor

The motor for the food dispenser will be small scale because it will only be needed for dispensing small amounts of cat food at a time. If the food dispenser consists of a turning mechanism, then a stepper motor will be the best choice. The stepper motor chosen to operate the food dispenser for the SMAC will be the Adafruit small stepper motor PF35T-48. The reason this small stepper motor was chosen is because the only function it needs to operate is turning the paddle wheel food dispenser. This motor is perfect for small application like the food dispenser and it has a solid mounting plate which will hold nice and firm on the back of the plastic food container. The golden rotator shown below will be the only piece inside the food container housing and the paddlewheel will be mounted to this piece. This stepper motor also works very well with the Motor shield for Arduino which allows for micro stepping resulting in smoother motion.

The micro-stepping driver will be one of the options used to program the Stepper motor for micro steps which would send frequency pulses to only rotate the

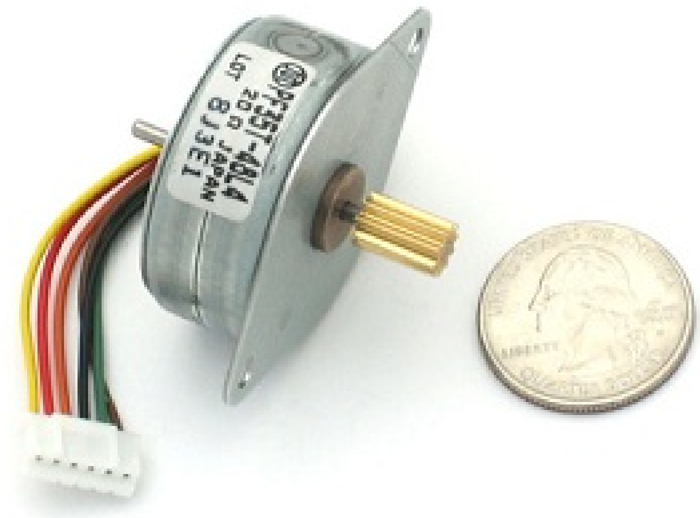

Figure 2.8: Adafruit Small Stepper Motor PF35T-48

turning mechanism at a small precise angle to position each tray in the proper location. The food dispenser will need to be properly positioned so it can receive the food, rotate, then dump the food in the tray all accurately. This is why the micro-step is perfect for this option of the food dispenser. The stepper motors are small in size but have a high power density which will be needed to pick up and transport the cat food. With the relatively small size, the stepper motor can be mounted inside the plastic housing and wired up to the dispensing mechanism mounted on top of the housing. The stepper motor is very efficient and does not consume a lot of power, as a 5V power supply is all that is needed. There are several advantages to using the stepper motor, but the main one is the use of micro stepping.

A DC motor was also considered in the initial design, however a DC motor may perform the function desired as well as the stepper motor. The DC motor connected to the turn style type mechanism would provide a constant slow rotation. This option however would have to be tested several times to see how much food is dispensed in a certain period of time. The motor would have to consistently provide the same amount of food each trial run to be considered. The trial runs would also be for determining the cut-off point for the motor. Once the motor is disconnected from the power, the turning mechanism will not instantly stop; there will be a slight delay in which could affect food being dispensed or food being filled. This delay is definitely a disadvantage to using the DC motor because it will have to be accounted for each time the food needs to be dispensed to the cat and if miscalculated the cat may receive more/less food than is needed. The SMAC needs to be extremely efficient and accurate, so the DC motor does not seem like the optimal choice. The DC motor that would be chosen for the food dispenser would be the Adafruit DC hobby motor 130 size. The reason this DC motor was chosen

Item	Specification
Drive mode	Unipolar — Bipolar
Step Angle	7.5°
Steps per revolution	48
Voltage	12 V
Winding Resistance	90 W/phase — 100 W/phase
Winding Inductance	48 mH/phase — 124 mH/phase
Holding Torque	20 mNm — 25mNm
Starting Pulse rate	500 pps
Weight	80 g

Table 16: Figure 2.8 Specifications

is because it comes with a wider operating range than most motors similar to its type which makes for easy controlling with an Adafruit Motor shield.

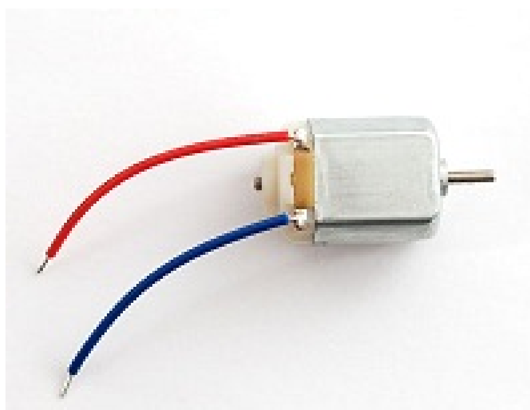

Figure 2.9: Adafruit DC Hobby Motor 130 size

If the food dispenser consists of the sliding mechanism, then the motor used will be either a stepper motor or DC motor. The stepper motor would be programmed for half-steps which would pertain to opening the slider and closing the slider. The stepper motor will be connected in parallel as supposed to the series connection for the stepper motor used to power the rotating tray tables. A parallel connection will lower the inductance which results in increased torque at faster speeds. The stepper motor will need to quickly slide open and closed to allow the precise amount of food every time. The increased torque will prevent any food getting stuck in the dispenser because it will power through until it is properly closed.

2.6.2 Plastic Housing

The Food container will be a plastic housing capable of holding up to 12 meals worth of dry food for one cat or 6 meals worth of food for 2 cats. This amount of

Item	Specification
Rated Voltage	6 VDC
Rated Load	10 g x cm
No-Load current	70 mA max
No-Load speed	9100 plus/minus 1800 rpm
Loaded current	250 mA max
Loaded speed	4500 plus/minus 1500 rpm
Starting Torque	20 g x cm
Starting Voltage	2 V
Stall current	500 mA max
Body Size	27.5 mm X 20 mm X 15mm
Shaft size	8mm X 2mm diameter
Weight	17.5 g

Table 17: Figure 2.9 Specifications

food will feed one cat for 6 days or 2 cats for 3 days. This will be ideal if the owner goes on vacation because they will not have to worry about their cat(s) not being fed the proper amount of food. It can easily be refilled as the top will pop open and the food can be poured in. It will be mounted to the center of the food housing and as a unit be connected to the plastic external housing. The bottom of the food container which is attached to the food dispenser can also be detached so the whole container can be removed for cleaning. The 2.4 inch LCD screen will be mounted on the pop off top for easy access for the user. The touch screen will not simply be placed on the pop off top, it will be incorporated into the top for a more professional look. A plastic decorated cover for the lid will hide the PCB and wires as well as having the user friendly touch screen displayed on top.

There is an option to attach a laser or some type of sensor device to the bottom quarter section of the food container. When the food reaches that point, the sensor will detect that the SMAC is running low on food and send an alert to the user. The alert could be as simple as a flashing red LED light or as complex as send a text message to the owner letting them know the SMAC is running low on food and needs to be refilled. The laser would work by having two sensors on either side of the food container that are lined up with one another. When there is food in the container there will not be a direct beam connecting from one sensor to the other because the food will be in the way. When the SMAC food container gets down to its last meals, the food will eventually go below this invisible line where the laser sensors are. When the sensors read a constant connection from one side to other, this means that the food is running low. This result will then activate a flashing red LED for the user to see. However, if the user is gone most of the day then this poses the risk of the user not seeing the red LED at all therefore resulting in not refilling the food. This creates the option to send the user a notification via email or text message to let them know the food is running low and needs to be refilled.

2.7 Scale to Weigh Cat

The cat scale will be connected to the SMAC as a unit. When the cat steps on the scale, the RFID tag hanging from the cats collar will then be scanned in by the RF reader. Once the tag is scanned in and the weight balances out, the two signals are then uploaded to the online file storage database. This way the owner will have a track of each cat when they come to eat as well as the weight of the cat at the time it scanned in. This way you are able to keep a track of the cats weight to determine if the cat needs more or less food as well as maintain a healthy diet.

One option considered was to build a scale using 4 mini digital scales placed in a rectangular shape with a small rectangular piece of glass on top. The piece of glass would measure 1 foot by 2 feet and would be wide enough to fit small and large cats. The 4 scales represent signal points, one on each corner, for the cats paws. All four scales will be wired to a single microprocessor. The microprocessor that is suitable for the SMAC is the Texas Instruments AM3715 (ACTIVE) Sitara ARM Microprocessor (MPU). This multipurpose microprocessor is a programmable device that accepts digital data as input, processes the data according to instructions that will be stored in its memory, and outputs a result. Since microprocessors operate only in the binary system, the four digital numerical inputs will be converted into to binary code for processing. This will ensure the most accurate measuring of the cats weight. This option would be cost efficient and highly effective. The architecture is designed to provide the best in class ARM and graphics performance while delivering low power consumption.

Figure 2.10: Texas Instruments AM3715 Sitara ARM Microprocessor

Another option is to get a digital scale which would have to be around the same size to accommodate for any size cat. The scale will be powered by batteries and will have a time-out to enter a low power state after a time period of no use as this will maximize the battery life. Once the weight balances out, the digital number

Item	Specification
Application	Consumer Electronics
Operating Systems	Neutrino, Integrity, Windows embedded CE, Linux, VX works, Android
ARM CPU	1 ARM Cortex-A8
ARM MHz	800,1000
ARM MIPS	1600,2000
Graphics Acceleration	1 3D
On-chip L1 Cache	64 KB (ARM Cortex-8)
On-chip L2 Cache	256 KB (ARM Cortex-8)
Other On-chip memory	64 KB
General Purpose Memory	1 16 bit GPMC, 1 32 bit (SDRC, Async SRAM, NAND NOR and OneNAND)
Input/ Output supply	1.8 Volts
Pin/Package	423FCBGA, 515POP-FCBGA

Table 18: Figure 2.10 Specifications

is recorded and uploaded to an online database via WIFI. The advantage to this option is that it will be more accurate as it is weighing the cat as a whole unit, not 4 separate inputs. With only 1 input, relaying the information online becomes a much simpler process. Another advantage is that the scale will enter a low power stand-by state to conserve the battery and not have to be turned on and off each time. The scale will enter full power state and become active when the cat first steps on the scale. However, the weight reading may not be 100% accurate if the scale does not enter the full power state in time. As the cat approaches and steps on the scale, the scale will immediately need to be ready to ensure that the whole cat is properly weighed.

Another option would be to use an LCD touch screen as the scale. An application on the screen would be able to electronically weigh the cat and store the information on a memory stick. The information could be uploaded to the online database or can be accessed at a later time by the user for checking the record of the cats previous weights. An advantage would be that every time the cats gets weighed out, the information could be time-stamped for easy readability for the user. The owner would be able to tell what time the cat came to eat and how much the cat weighed at that particular time. A disadvantage however would be that the touch screen would be very expensive.

2.8 Food Dish

The Food dish will consist of the internal and external food tray housing, the scale for weighing the food, and the stepper motor for rotating the trays to the desired locations. The scale to weigh the cat will also be connected to the food dish in the front of where the opening slot is for where the cat can access the food. The opening slot where the food is displayed must be the same width as the internal food tray so the cat can access all of the food directly. It is also important for the stepper motor to be correctly programmed to precisely rotate the trays to the opening slot.

2.8.1 Motor

The Microstepping motor drive accepts inputs of step pulses and direction signals from a controller. Each step pulse causes the motor to rotate to a precise angle. The speed of rotation is determined by the frequency of the pulses. The direction signal determines whether the direction of rotation will be clockwise or counter clockwise. The enable signal acts as the power button by turning the motor on and off. Microstepping is often called a sine cosine microstep because the winding current approximates a sinusoidal AC waveform. In orxder to make the microstep operation run more smoothly and reduce resonance in any parts, the microstep will need to be very small.

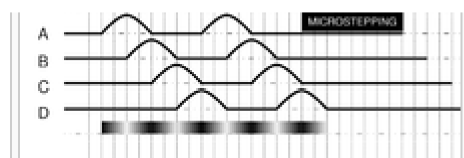

Figure 2.11: Micro Step Drive model for Stepper Motor

This drive is perfect for the stepper motor because by applying a step pulse, it can rotate a precise amount. This precise movement would be for rotating the food trays to the desired location such as the spot for food dispenser and the open slot to make the food available to the cat.

The Stepper Motor will be used for powering the food dispenser as well as the rotating mechanism that the food trays will be mounted too. We chose to use a stepper motor because of its many advantages. Some advantages to using a stepper motor include (but not limited to):

The stepper motor will be connected in series to provide a high inductance resulting in a greater torque at lower speeds (as supposed to a parallel connection which would provide low inductance). This increased torque is needed for rotating the trays , which will be filled with food, at a low speed. A cat eats roughly a cup of food per day so with weight of the food and weight of the tray, it is possible to calculate the frequency of the pulse which will provide the correct speed for rotation of the trays. The Stepper motor will receive an input signal when a cats RFID tag is scanned in by the reader and there will be six small equivalent rotations corre-

Rotational angle of the motor and the input pulse are proportional to each other
When the windings are energized the motor has full torque at standstill
Positioning and repeatability of movement are precise
Excellent response time to starting, stopping, and reversing the motor
The motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control
A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses

Table 19: Advantages to using a stepper motor

sponding with the six trays of food. Each tray will be assigned a number and the stepper motor will be programmed to rotate the trays to allow that specific cats food to be available to the cat.

The stepper motor that will be used for rotating the food dishes will be the Adafruit Bipolar stepper motor. The reason this stepper motor was chosen was because it was cost efficient while at the same time very powerful and reliable.

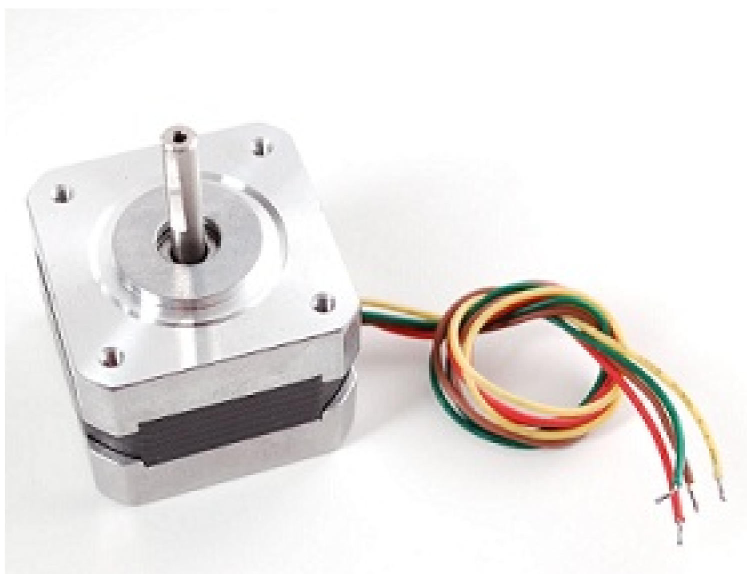

Figure 2.12: Adafruit Bipolar Stepper Motor

2.8.2 Scale

The food scale will be mounted underneath the tray housing and each tray can be rotated to that location for weighing the correct amount of food for the specified cat. The scale will be weighing the food in ounces. The scale will always be set to zero when turning on or powering back up from low power state, this allows for

Item	Specification
Drive Method	Bipolar
Number of Phases	2
Step angle	1.8 deg/step
Steps per revolution	200
Voltage	12 V
Current	0.35 A
Winding resistance	34 W/ phase
Inductance	4.3 mH/phase
Square body	1.65 in
Square mounting holes	1.22 in
Drive shaft	5mm diameter
Holding torque per phase	28 oz x in, 20 Nxcm, 2 Kgxcm
Weight	0.57 lb
Insulation Resistance	500 VDC 100MHomes min
Dielectric Strength	500VAC 50HZ min

Table 20: Figure 2.12 Specifications

maximum accuracy when weighing the cat food. When the tray is rotated to the scale a few seconds is allotted for the scale to balance out. Once the scale balances out, it will send a signal to the dispensing lever to slowly begin opening. The food will slowly be dispensed down into the tray and when the scale reaches an amount around 3 ounces (average amount of food per serving but may be higher or lower depending on the cats needs) which is close to the limit, the dispensing lever slowly begins to close. Once the scale reaches its specified limit, it sends a signal to the dispensing lever to quickly close shut. The advantage to using this option would be that food is slowly being dispensed and the scale would slowly be climbing to its limit, giving it plenty of time to begin closing when reaching the limit. The disadvantage however would be when the scale reaches its limit, there will be a slight delay in sending the signal to the sliding mechanism to promptly close. The delay will result in more food being dropped in than desired which ultimately results in over feeding the cat.

The turn style type dispensing mechanism would be more ideal for the scale because it would dispense a certain amount of food at a time and give the scale time to accurately balance out before weighing the next handful. Knowing the amount of food being dispensed each time it turns, the cat would receive a number of turns corresponding to the amount of food needed, and the scale would more be used for checking the weight

Figure 2.13: SMAC Main Page

2.8.3 Plastic Housing

The external skeleton of the SMAC will be a plastic covering that has the food container mounted to it. It will be a round shape and properly decorated to be inviting to the cat so it will come up to eat and not be frightened of it. The food container will be mounted on top in the middle and will be connected to the dispensing mechanism. The plastic housing will encompass the food container, food dispenser, and the food trays as well as cover all the wiring, motors, scales and circuit board. There will be a slot opening in the plastic housing where the cat will be able to walk up, get scanned in, and will then have access to the food. The stepper motor will be programmed to perfectly rotate the food tray to allow the cat to access all of the food in the tray.

The internal food dish for the SMAC will have six one-cup trays that fit into a circular housing compartment. These trays will be removable and can be put in the dishwasher as they will get dirty from the food. Each tray will assigned to a cat and every time the cat walks up, it will only be allowed to eat from that specified tray. One location for the trays will be where the scale is and also where the food will be dispensed. All trays should be able to be rotated to that location allowing food to be dispensed and the scale to be able to properly weigh all the food.

2.9 Control Unit (PCB)

The PCB control unit is the foundation for the SMACs functionality. The correct PCB must be chosen based on the needs of the electronics being installed in the cat feeder. Complicating the decision on what PCB should be chosen for the project is the fact that custom PCBs can be made specifically for ones requirements. For the sake of time however building a custom PCB for the SMAC was eliminated and for monetary limitations having someone else build a custom PCB was also eliminated. As far as prefabricated PCBs go Arduino seems to be the most widespread choice with the most coding and software available. This is why the first two choices for the SMAC are the Arduino Duemilanove and Arduino Mega 2560. The third option considered is the IteadMaple v1.0 and while it is not Arduino brand it is still Arduino-based. The fourth option is to build a custom PCB board to accommodate the SMACs specific needs.

Arduino Duemilanove - The Arduino Duemilanove is one of the most basic Arduino boards on the market. This board is very suitable for inexperienced operators of PCB boards. It is powered by either the USB connection or an external power supply. If an external power supply is used then it must be either a battery or an AC to DC adapter. The external power must be supplying between 6-20V for the board to be on but if it is supplied with less than 7 then the 5V pin could supply less than 5V and cause the board to become unstable. Any power supply over 12V could lead to overheating and eventual damage to the board so the recommended range is 7-12V. The board is programmed easily with Arduino software found readily from several databases. The simplicity of the board along with it being prefabricated, coded with easily obtained software and being the least expensive board considered makes this selection very desirable. The down side is that being the most simplistic board means less pins, less memory and a few less features than some of the other PCBs. All the specifications of the Arduino Duemilanove are listed in Table 21.

Item	Specification
Family	ATMega 328
SRAM	2k
Flash	32k
EEPROM	1k
Clock	16MHz
UART	1
PWM Outputs	6
Digital I/O Pins	14
Analog Input Pins	6
VCC	5V
Vin Range	7-12 V
5V	Yes
3V3	Yes
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Dimensions	2.7(L) x 2.1(W) in

Table 21: Arduino Duemilanove Specifications

Arduino Mega 2560 - The Arduino Mega 2560 has essentially the same features as the Duemilanove but has more pins, memory, and flash. It is powered by either the USB connection or an external power supply. If an external power supply is used then it must be either a battery or an AC to DC adapter. The board has an input voltage range of 7-12V for the same reason as the Duemilanove board. The board is programmed easily with Arduino software. The advantage in using this board is for the extra memory Flash and I/O pins. This board is a bit more expensive than the Duemilanove but if more I/O pins are needed then this board is a must. The specifications for the Arduino Mega 2560 are shown in Table 22.

Item	Specification
Family ATMega	2560
SRAM	8k
Flash	256k
EEPROM	4k
Clock	16 MHz
UART	4
PWM Outputs	14
Digital I/O Pins	54
Analog Input Pins	16
VCC	5V
Vin Range	7-12V
5V	Yes
3v3	Yes
DC Current Per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Dimensions	4(L) x 2.1(W) in

Table 22: Arduino Mega 2560 Specifications

IteadMaple v1.0 - The IteadMaple v1.0 is a microcontroller board based on Leaf Maple. Like the other boards it is powered by either a USB or an external power source. It has a much larger range for its input voltage, ranging from 7-23V. The IteadMaple board provides substantially more I/O pins than the Duemilanove while providing much more flash and even more SRAM than the Arduino Mega Board. This board is a happy medium between the two. It is compatible with most Arduino shields, more importantly with the touch screens and the WIFI shield also considered for the SMAC. Iteadmaple is programmable by Arduino style, sketch based programming environment and is open sourced so the code can be downloaded. The board is equipped with 7-channels of direct memory access. The specifications for the IteadMaple v1.0 are listed in Table 23.

Custom-built PCB - A custom-built PCB can be built to meet each specific need for the SMAC while using a common development board as a guideline to ensure compatibility with all components required in the design. Before designing a

Item	Specification
Family	ARM
SRAM	20k
Flash	128k
Clock	8-72 MHz
UART	3
Digital I/O Pins	39
Analog Input Pins	16
VCC	5V
Vin Range	7-23V
DC Current per I/O Pin	15 mA
Dimensions	2.05(L) x 2.1(W) in
Processor	72 MHz ARM Cortex M3

Table 23: IteadMaple v1.0 Specifications

PCB the circuits required for all the components must first be understood. There are several software programs that can be used to design a PCB including Eagle, PCB Artist, Texas Instrumentsâ WeBench, and many others. The WeBench from Texas Instruments allows the designer to get an idea for how the necessary circuits should be configured for specified input voltage and load attachments, the components needed to build the circuit, and has an optimization tool based on Footprint, cost of materials, and efficiency to compare possible solutions for the desired circuit. The block diagram shown in Figure 2.14 using WeBench with the input voltage and load attachments of the SMAC.

From the block diagram WeBench will assemble the required circuitry to achieve the appropriate voltages for the corresponding loads. The upper circuit of the block diagram, used for providing power to the 3.3V loads, is shown in Figure 2.15. The lower circuit, used for providing power to the 5V loads is shown in Figure 2.16.

WeBench also calculated the total bill of materials to be just under $9 total. It also gives you the opportunity to purchase all the components to build the circuits, which saves the hassle of choosing which specific circuit element is needed from the endless amounts of possibilites. The only downside is that WeBench only provides products that Texas Instruments sells themselves, which means it isnât really calculating all possible solutions. Now that a general idea of the circuit design is understood PCB design software can be used to develop the PCB. PCB Artist was experimented with for possible development of the board, but Eagle was eventually chosen for the design of the PCB because it is more popular. Since it is so popular there is an abundance of help with design and other information needed to have success with the program. Eagle also allows for users to upload there own custom symbols, schematics, and libraries for other users to download. Another benefit to

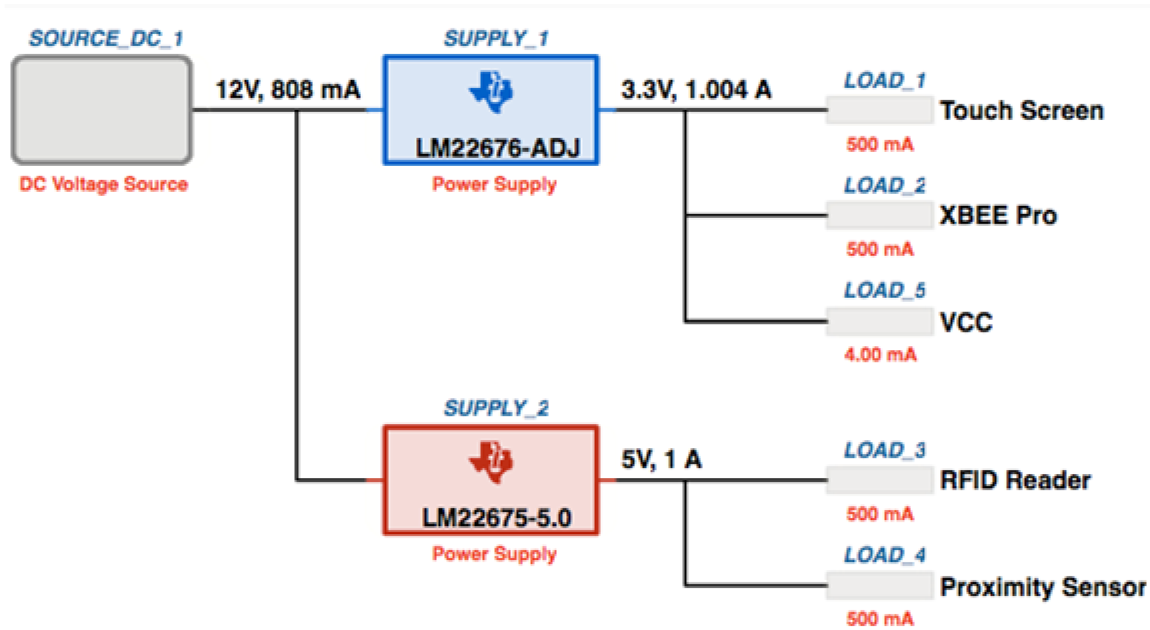

Figure 2.14: WeBench Block Diagram

Eagle is that almost every development board that may be used as a basis for the custom designed PCB can be found in an Eagle schematic, allowing the user to see what other boards look like in Eagleâs format.

2.10 WIFI Card

The SMAC will not only be feeding, as it will also be keeping an active log of the cats eating habits and weight. Each time the cat walks up onto the scale, the reader will scan the RFID tag on the cats collar and send the date/time as well as the weight and how much food the cat consumed via the option chosen to the online database. The user will then have a constant record of the cats weight and amount of food it has eaten and the day/time of the meal to insure everything is normal and that the cat is not overeating and gaining weight or not eating enough and losing weight. The first option for allowing information relay is to provide the SMAC with an Internet connection, either through an Ethernet cable or attaching a shield that would enable WIFI connectivity. The second option considered for enabling the SMAC with the power to communicate with the online database was the

An Internet connection is the first option in order to relay information from RF reader/transceiver and the scale to the online file storage database. The advantage to Internet is the ability to send an alert in the form of an email or text message to the pet owner notifying them when the food dispenser is close to being empty. Hardwiring a router to the PCB board enables the SMAC to have Internet

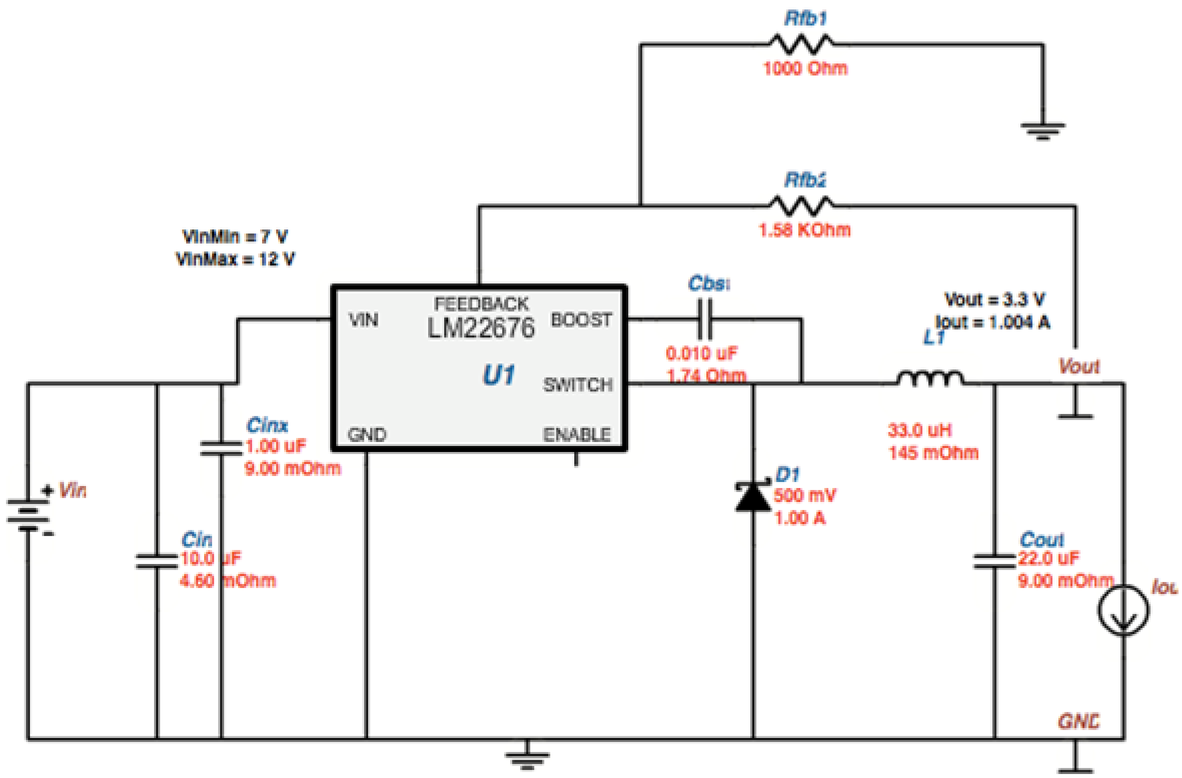

Figure 2.15: WeBench Block Diagram

connectivity and allows the relaying of information taken by the SMAC to the online database. The only requirement for the router is that it must be supported by OpenWrt. In order to connect the router to the PCB the back cover of the router must be removed, exposing its circuit board. On the circuit board there will be four solder filled vias aligned in a row. A pin will need to be soldered to each of the vias. A wire must then be connected from the GND pin on the router to the GND pin on the PCB and another wire must be connected from the RX pin on the router to the RX pin on the PCB. This is all that is required to physically connect the PCB to the router. OpenWrt firmware must now be created using a Linux machine. The firmware required can be found rather easily. Code must then be written in order to allow the router and the PCB to communicate information back and forth, code that again is found easily. The biggest downfall for this option is that the router must be connected with an Ethernet cable. This may hinder that ability to place the SMAC wherever is desired in order to be close to an Ethernet plug, or have a very long Ethernet cable coming from the SMACs router. In a sense the downfall for this option can also be seen as its best benefit. While it may be inconvenient to place the SMAC anywhere due to the required Ethernet plug being nearby, internet via Ethernet is more reliable than WIFI. WIFI can come and go sometimes while an Ethernet connection can be considered more concrete as long as it remains plugged in.

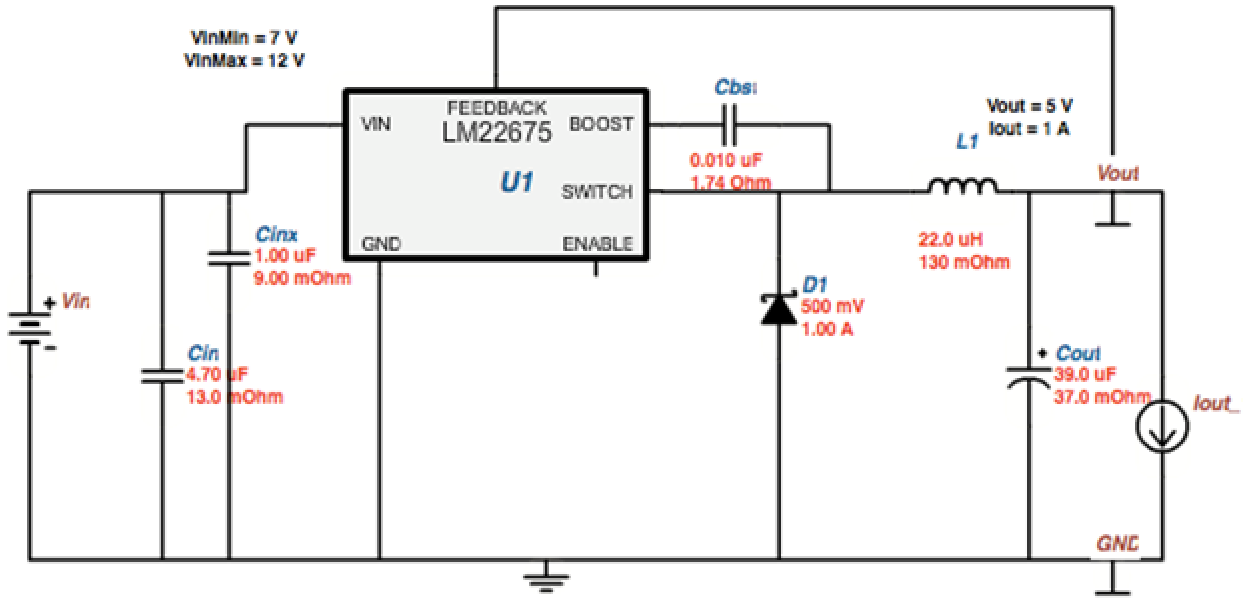

Figure 2.16: WeBench Block Diagram

Connecting to the Internet via WIFI seems to be the obvious solution to allowing the SMAC to be placed wherever one pleases. In order to connect to a wireless network a shield can be connected to the PCB. The two shields considered were the WiShield V2.0 for Arduino and the LinkSprite CuHead WIFI Shield for Arduino Mega. The WiShield V2.0, shown in photo below, will give the Arduino Duemilanove the ability to connect to a wireless network. The shield provides 802.11b connectivity with throughput speeds between 1-2 Mbps and simply plugs right on to the Arduino board. The WiShield also comes with built in 16MB serial flash for extra storage space. This shield has the capability of creating WEP, WPA, and WPA2 networks. WiShield uses SPI for host communication with a max speed of 25MHz. The WiShield is shown in Figure 2.17.

The LinkSprite CuHead WIFI Shield for Arduino Mega will allow the SMAC to connect to WIFI if the Arduino Mega board is used. The shield provides 802.11b connectivity with throughput speeds between 1-2 Mbps. The LinkSprite comes with built in 16MB serial flash for extra storage space. This shield has the capability of creating WEP, WPA, and WPA2 networks. It uses SPI for host communication with a max speed of 25MHz. The LinkSprite Shield also plugs directly to the Arduino board and is shown mounted in the photo below. Both shields come with required source code files. Purchasing a Shield add on for the PCB is slightly less cost efficient but is an extremely fast and easy way to bring WIFI to the SMAC.

Digi XBEE Pro- The XBEE Pro Wire Antenna from Digi is a reliable communi-

Figure 2.17: WiShield V2.0

cation module used to relay information to and from microcontrollers, computers and many other systems. It supports point to point and multi-point networks. The XBEE Pro module has a very easy to use serial command set. Using the Digi XBEE Pro eliminates the need for internet connectivity all together since it communicates directly between the PCB to the computer. After being relayed to the computer, information will be uploaded to the online database. The XBEE pro is also the most cost efficient way to get the information needed from the PCB to the database. Digi provides programming and testing software with the XBEE Pro and since it is easily connected to the PCB there is very minimal work required for installation. The Digi XBEE Pro with Antenna attachment is shown below in Figure 2.19.

Item	Specification
Output Power	60 mW
Max Data Rate	250 kbps
VIn	3.3V
IIn	55 mA
Indoor Range	300 ft
Digital IO pins	8
Operating Frequency	2.4 GHz
Dimensions	2.438 x 3.294 cm

Table 24: Figure 2.19 Specifications

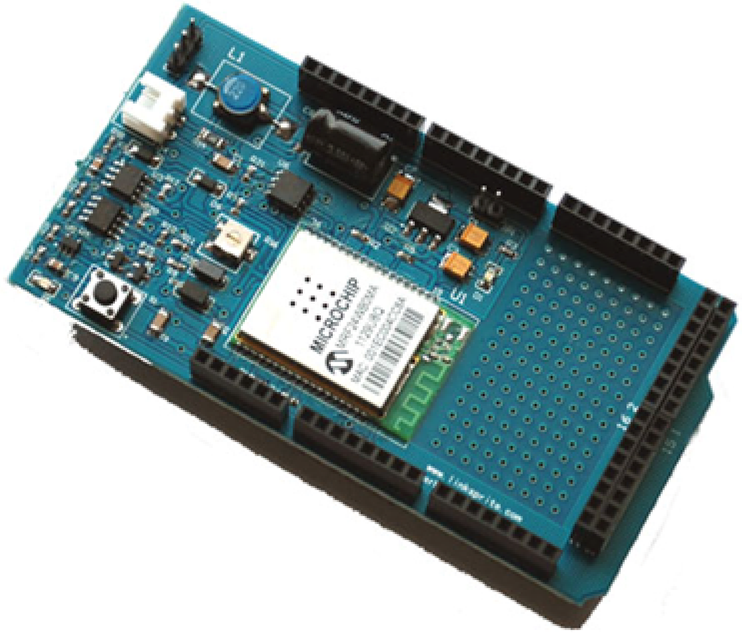

Figure 2.18: LinkSprite CuHead

2.11 Touch Screen

In order to enable the user to input valuable data into the SMAC an LCD touch screen will be mounted on the exterior. An LCD touch screen is necessary to enable the user to input valuable information into the SMAC. The user will be prompted to input specifics for each cat upon activating the device to insure the proper information is obtained. Information inputted will include type of cat, age of the cat, amount of food the cat should be given, and name of the cat to identify which cat corresponds to the appropriate RFID tag. Adding a touch screen to the device keeps up with technology of today since almost every device with a user interface is now touch screen. The touch screens that were considered were the 2.4 inch TFT LCD Screen Module and the Arduino 2.8 inch TFT Touch Shield.

2.4 Inch TFT LCD Screen Module: ITDB02-2.4D - This touch screen can be used with almost any Arduino board, more importantly both Arduino boards considered here, the Arduino Uno and the Arduino Mega board. The touch screen is small and compact at only 2.4 inches so it can easily fit on the SMAC and not preoccupy too much space. This touch screen comes equipped with a space for an SD card to ensure enough memory is available. The screen has a sleep mode to help save power. Demo code for enabling the touch screen through an Arduino board is provided from the merchant, which is a huge plus. The TFT LCD Screen

Figure 2.19: Digi XBEE Pro with Antenna

module is shown in Figure 2.20.

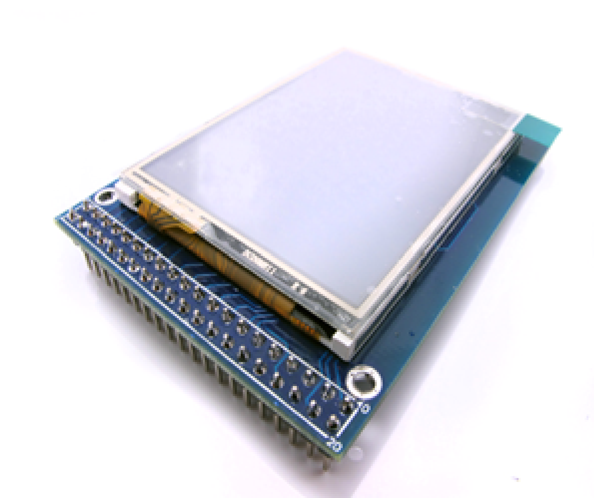

Figure 2.20: TFT LCD Screen Module

Arduino 2.8 Inch TFT Touch Shield - The Arduino 2.8 inch TFT Touch Shield is another touch screen display that can be used as the user interface for the SMAC. The most obvious advantage for this touch screen is that it can be mounted directly to the top of our Arduino board, as shown in the picture above on the left. However, if the Arduino Duemilanove board is chosen the Arduino Touch Shield will not quite fit due to the large USB socket, shown below. If the Arduino Mega board is chosen an extra, but inexpensive component can be purchased to mount the Touch Shield to the board. The Touch Shield is shown in Figure 2.22.

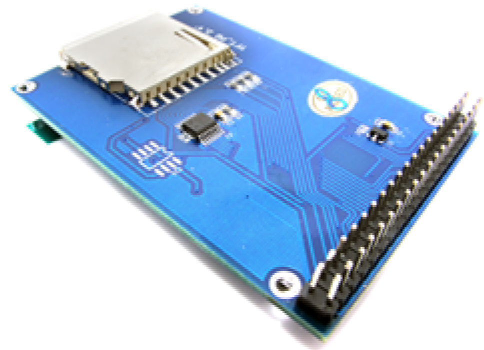

Figure 2.21: TFT LCD Screen Module

Item	Specification
Model	ITDB02-2.4D
Screen	2.4 inch LCD
Driver Element	a-Si TFT LCD Single Chip
Resolution	320 x 240 RBG
Display Model	TFT Transmissive, Normal White
Dot Pitch	0.153 x 0.153 mm
Active Area	36.72 (W) x 48.96 (H) mm
Module Size	42.72(W) x 60.62(H) x 3.7(T) mm
Power Voltage	3.3 V
Weight	40g
Operation Temp	neg 20°C neg 70°C

Table 25: Figure 2.20 Specifications

2.12 Software Components

2.12.1 Operating System

SMAC will require the use of an operating system to properly control all the hardware and allow for a sleek design that is user friendly. Four different operating systems were researched: Meego, Android, Ubuntu, and Windows Embedded System 7 (WES7). All but the WES7 operating system are free to install and use. The most ideal choice for SMAC looks to be the Android option due to all the features listed below.

Meego - Meego is classified as a mobile operating system. It is Linux based and can run on ARM and Intel x86 processors with Supplemental Streaming SIMD Extensions 3 (SSSE3) enabled. SSSE3 is a specific set of instructions created by Intel, thus limiting the number of processors that can be used with this operating

Figure 2.22: Arduino 2.8 inch TFT Touch Shield

Figure 2.23: Arduino 2.8 inch TFT Touch Shield

system. That being said, most of the processors used in common embedded systems are SSSE3 enabled.

Meego has three options to choose from for a graphical user interface that would be useful to SMAC. They are the Netbook UX (User Experiences), the Hanset UX, and the Tablet UX. The Netbook UX is very similar to many user interfaces seen on personal computers. The Handset UX would be best for SMAC because it is made for smaller window sizes. The Tablet UX would probably be too graphical for the purpose of SMAC and cause the use of a more expensive printed circuit board. The software development for Meego can be done with the Qt framework using the Qt creator. Qt is a cross-platform application and UI framework.

Android - Android is a Linux based operating system that was also initially intended for mobile devices. It is now used in products like laptops, netbooks, TVs, cameras, and various other embedded systems. Its main supported platform is

Item	Specification
Model	SHD040
Driver Element	A Si TFT LCD Single Chip Driver
Resolution	320 x 240 RGB
Module Size	72.5 x 54.7 x 18 mm
Active Area	43.2 x 57.3 mm
Power Voltage	3.3 to 5.0 V
Weight	24 g

Table 26: Figure 2.22 Specifications

ARM, but there are versions that support MIPS, and Intel x86 platforms. Comparing Android to Meego there are a lot more options when it comes to supported processors that can be used. While Android is open source like Meego is has a dedicated team, The Android Open Source Project (AOSP), that maintains it. This makes the operating system more reliable than a normal open source project.

Java is the main coding language used in Android. The Android SDK is the Java package that allows many different developers to write code independently of each other. Figure 2.24 shows the architectural design of Android. The Linux kernel is what is most important when choosing what operating system is compatible with what hardware. Although the main applications are done in Java and the lower level design (libraries, android runtime, and application framework) are done in C. This design is ideal for SMAC because it allows the design to be done in a more robust language, yet still have the option to add more libraries or drivers needed by the hardware.

Ubuntu - Ubuntu is also a Linux based operating system, but its main support is for the personal computers and servers. The supported platforms are not as large as the previous two options mentioned. Ubuntu supports the i386, AMD64, and ARM platforms. Ubuntu recently came out with a version called Ubuntu for Android phones. This would be ideal for SMAC except that it is meant to run along side of the Android operating system. The most reasonable option that would work for SMAC would be the Ubuntu for netbooks. It can run as the main operating system and has graphical abilities as well.

Windows Embedded System 7 - WES7 is the only non-Linux based operating system researched. As the name suggested it is windows based and can run on x86 and x64 architectures. One advantage of choosing this would be that it can run the same programs that run on Windows 7. This means that SMAC could be coded and tested in any language on a desktop and when it is completely done uploaded to the printed circuit board. This would save time from continuously re-uploading the software every time a change is made.

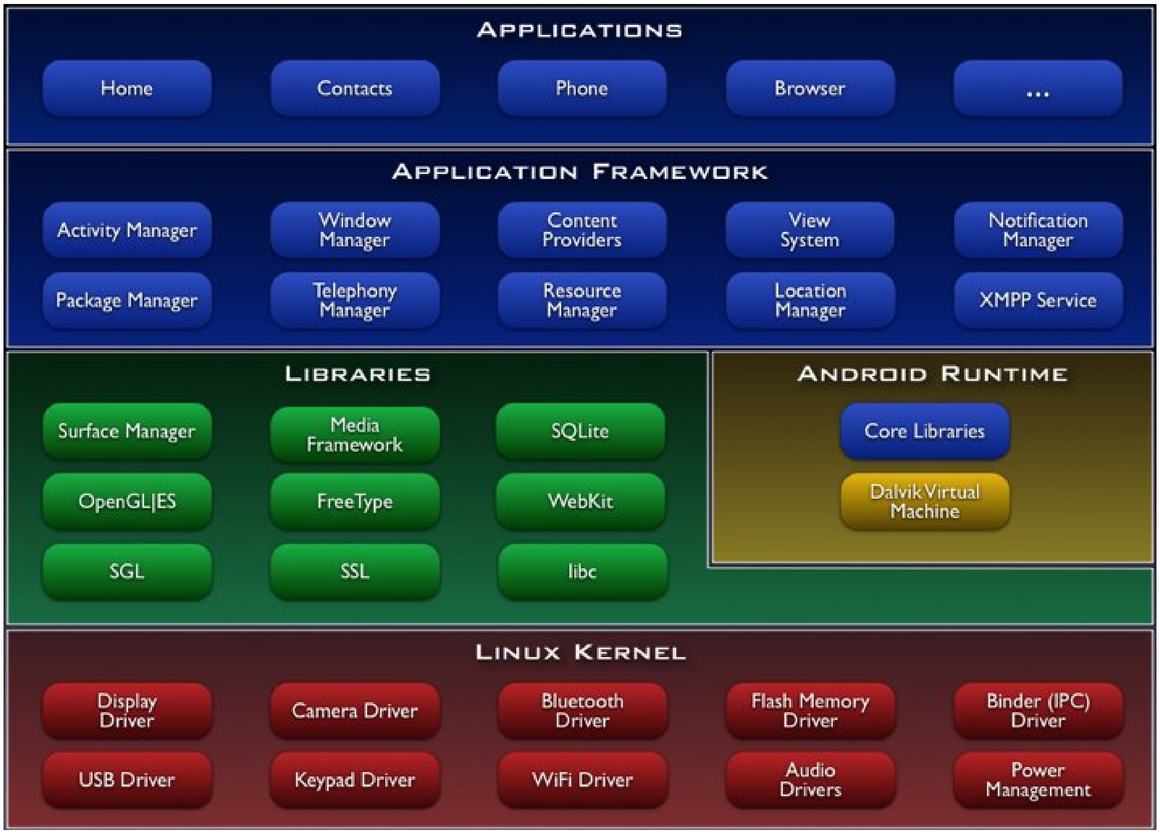

Figure 2.24: Android Architecture Diagram from Wikipedia

An interesting feature in WES7 is the option to remove fully customize the startup, logon screens, and dialog boxes. This would be ideal for a system like SMAC so that when booted up, it can immediately load into the program and prevent the user accessing the desktop preventing any errors. Unfortunatly this option is not free, one run time license is $95 and the WES7 Toolkit goes for about $1000. This is completely outside the budget for SMAC, thus making the decision to not use it.

2.12.2 Programming Language

To make sure SMAC runs properly the correct programming language must be used. It must be compatible with the operating system chosen and make the coding easier to minimize the time spent on it. Four main programming languages were researched: Java, C, C++, and C#. Java and C# are very similar while C and C++ are similar and compatible with each other. C# has a few variations when used with other languages, one is the Windows Presentation Foundation (WPF) which is part of the Dot Net 3.0 Framework (.NET 3.0). Java will be the language that SMAC is programmed in so that it runs on the Android operating system and allows the code to be reused for the Android application as mentioned in the next section.

Java - Java is basically a combination of C and C++ with a few extra features. It is considered an Object Oriented Programming Language. This type of language is ideal for programming a large program like SMAC. It allows the reuse of code and will thus cut down on the time required to code. SMAC will benefit from Java because it can be made very user friendly and will integrate well with the Android operating system.

Java is architecture-neutral, meaning that it can be written and compiled on an ARM architecture and ran on a MIPS or Intel architecture. This is ideal for coding on a desktop and then installing the compiled code onto the printed circuit board. This can be done because Java uses a Java Runtime Environment (JRE) so that it can be compiled into machine code relevant to the system it is currently on.

C - C was originally meant for implementing system software, as in the Android architecture shown in Figure 2.24. It has since expanded to the use in developing many high-level programs. C is the closest language to assembly language. Although unlike Java once it is compiled on one architecture it might not work on a different one. Many people avoid this dilemma by simply compiling the C program on every machine separately. With C running directly on the operating system it will be faster than the same program written in Java.

C++ - C++ was originally made from C to allow for objects, thus turning into an Object Oriented Programming Language like Java. It is not as robust as Java in many ways. One is it has the same compiling restriction as C, which will make it run faster than Java, but can be a pain to make cross-platform. It also does not have the same level of functionality as Java since it was made to be compatible with C. C++ is a mix of high-level and low-level programming while Java is strictly high-level.

C# - C# would be considered Microsoft's version of Java. It has most of the same features of Java and the syntax is almost the same. It still has the same issues with cross-platform as the previous two languages mentioned. C# can be used with XAML often called WPF which is part the .NET 3.0 framework, both described below. C# would most likely be a poor decision for SMAC since Android is a Linux based operating system and C# is meant more for Windows.

WPF - WPF combines XAML and C# together to allow for the user interface to be coded separately than the logical part of the program. It comes with three options for the user to interact with the server that SMAC will be using. The first one would allow the client to ignore the server, thus never getting any updates. The client can also have read-only access to the server, thus data will always be local. The third option, and best for SMAC, is that the user can read and write data to the server. This will allow the user to upload daily information on the cat, as well as download statistics from the server.

This would be ideal over Java because Windows Visual Studio allows a drag and drop kind of interface with XAML to make the GUI easier to code, this will also cut down on coding time. The fact that the GUI and logic are separated when coding allows for the coder to change one without effecting the other. This will come in handy if the design of SMAC were to change but the functionality was not. If this option was used SMAC would have to use the WES7 operating system instead of Android, raising the building cost.

.NET - The .Net framework like the previous languages can only run on a Windows operating system. This being said it has the same functionality as Java where it only has to be compiled once and it then cross platform. This is because the program is ran through a Common Language Infrastructure (CLI). This acts like the JRE, except it can run more than one language. While the JRE is only for Java, CLI is for C#, VB.NET, and J#.

2.12.3 Mobile

In order for the SMAC user to monitor their cat when not at home they will need access to a mobile application. The following three platforms are the most commonly used in the market. Based on the research done, Android seems like the best choice for the mobile SMAC application.

Android - The Android SDK would allow only Android users to keep track of their cats on the go. If the Android operating system is chosen for the PCB it would make sense for the SMAC mobile version to be Android as well. This would allow the reuse of most of the code, while changing the GUI to be more suited for a phone. The Android SDK comes with key libraries that would be save time while coding.

iOS - While much of the population uses iPhones it would not be cost or time effective to use the iOS SDK to make an application. This is because a different language would have to be learned, called Objective-C. Objective-C is an object-oriented language which allows the user to code in C as well. This would mean that the program coded to run on the printed circuit board would not transfer over to the iOS application. This would make using the iOS SDK counter productive.

Windows Mobile - There are a few advantages to using the Windows Mobile SDK for the mobile SMAC. One would be simply to expand the customer base, thus not limiting it to Android. Also if the WES7 was used and the main program coded in C++ then, like Java and Android, the code would be reusable.

jQuery Mobile - jQuery Mobile is a way to program an application to be used on multiple different platforms. The reason this option can be run on various platforms, is because it is ran through the internet. It is written in a language similar to HTML5. This would basically make the SMAC application a website that they user would pull up on their phone. The advantage to this would be that more users could use the application. Though they would have to open a web browser to access it. The disadvantage would be that it would require more time to code, since the main code could not be reused to code up the mobile application.

2.12.4 Database and Server

One of the specifications of SMAC is to allow the user to upload and download information over various platforms as well as be notified of any changes. To do this a server and database is required. The server will either be built from old computer parts or space will be bought over the internet. The database has more options to research and see how they would relate best to the specifications of SMAC. The four options researched where: SQLite, ProstgreSQL, Microsoft SQL Server, and MySQL.

SQLite - SQLite is a self-contained database/server. It is free and would alleviate the need to have a server, thus adding no cost to the project. It would run on a dedicated computer and would mimic a server, but instead would just read and write to the hard drive. The only problem is that it is only compatible with the C language, which is unlikely to be used when coding SMAC.

ProstgreSQL - ProstgreSQL is an open source database system that is free to use. Its best feature is that it is highly customizable. This means that it can run stored procedures in many different languages, including Java, Perl, Python, Tcl, C, C++, and its own PL/pgSQL. This would make it easier to use since no matter what language the main programs are written in they can be utilized in the database as well. Thus lowering coding time and room for error.

Microsoft SQL Server - Microsoft SQL server allows clients to store and retrieve data by using other applications other than the one running the server. This is ideal for SMAC because it will allow the access of information from any internet connection. The data is stored in a collection of tables with typed columns.

Like many of the other researched databases the following are some of the data types that can be used: Integer, Float, Decimal, Char, String, Varchar, binary, text and many others. This would allow the main program to simply send the information about the cat in a tabular form to the database and retrieve it the same way. Although this seems to be a good fit for what SMAC needs it is on the high side when it comes to cost.

MySQL - MySQL has a free open source project that can be used for SMAC. It would be unnecessary to purchase an edition with more features because of the simplicity of the system. The programming languages it is written in are C and C++. Websites like Wikipedia, Google, and Facebook use MySQL proving that is would be a reliable choice for SMAC. The MySQL Workbench is a free UI environment that allows the user to visually design the database. This would cut down on the time needed to code the database.

MySQL comes with a suite of command-line tools to get the user started. These range from querying the database to backing up data. Since it is open-source there are many other third-party tools that were created. The endless amount of tools available will make the coding more efficient and accurate if less methods have to be written from scratch. Overall MySQL would be a great choice for SMAC because it is open-source with many well known customers proving that it is more reliable than the previously mentioned options.

2.12.5 Notification System

SMAC will have a notification system that will send a text message or email to the user, per the users' specifications. There can be a daily, weekly, monthly statistics notification. Or the user might want to get notified if the food is low, or something else is malfunctioning on the machine. Since the main program will be running on Android, written in Java and have an internet connection this is not only possible, but reasonably simple.

The Java language has the ability to send emails by using the Javamail API. This API gives the coder the classes needed to connect to the mail server with a user name and password. SMAC will need its own email address in order for this method to work. The user then enters the message, which will be selected based on the notification being sent, and then the email address to send to. Since any email address can be used with this method, then text messages can be sent as well.

Company Name	[10-digit phone number]@
Alltel	message.alltel.com
AT&T	txt.att.net
AT&T - multi	mms.att.net
Boost Mobile	myboostmobile.com
Cricket Wireless	sms.mycricket.com
Cricket Wireless - multi	mms.mycricket.com
Nextel	messaging.nextel.com
Sprint	messaging.sprintpcs.com
T-Mobile	tmomail.net
Verizon	vtext.com

Table 27: Email Addresses for Phone Carriers

To send a text message from an email the phone company must be known. This will be collected from the user during the time in which they make a SMAC account. Once the phone company is known the email address can be constructed using a simple case statement. Below is a table of the available email addresses for some of the main phone carriers. The main format is the ten digit number followed by the symbol and a phrase indicating the company. Two of the companies in Table 27, AT&T and Cricket Wireless, have the option of sending a multimedia text message. This could be an option for the user to choose if they have that service.

3 Project Hardware and Software Designs

3.1 Software Design

A large part of how SMAC will operate is software based. Due to the scale of the project it will be broken down into a few different UML diagrams including use case diagrams, activity diagrams, and class diagrams. The use case diagrams will be key in understanding the most basic level of the SMAC operation with the users. From there the activity diagrams will give a more detailed description of how the user can interact with the system and the choices that can be made. The UML diagrams will help the reader understand the later class diagrams which will delve into the actual coding of the system.

3.1.1 Use Case Diagrams

To help describe how the SMAC software will work and interact with the users' some use case diagrams are presented. Figure 3.1 shows the interaction of the user with the system. The last Figure in this section, Figure 3.2, shows how the user will interact with the SMAC android system.

The use case diagram below shows the basic interaction that the owner (depicted as the user) has with the system. When the system is first turned on the user can either make a new account or login with an existing account. If a new account is made the server will then be updated. If the user already has an account, then once logged in, the user will have the following three options. The user can add a cat, delete a cat, and view the statistics on a cat. The first two options will update the server, while the last option will retrieve data from the server. When the cat walks up to the SMAC machine the RFID transceiver will send out a signal and it will receive the RFID tag number from the cats collar. This part of the system will then retrieve the correct information from the server to feed the cat the correct amount of food.

The interaction from the user is different in the Android SMAC system than the main SMAC system. In the Android system the user cannot add or delete cats from the system. This is because when the Android SMAC system is being used

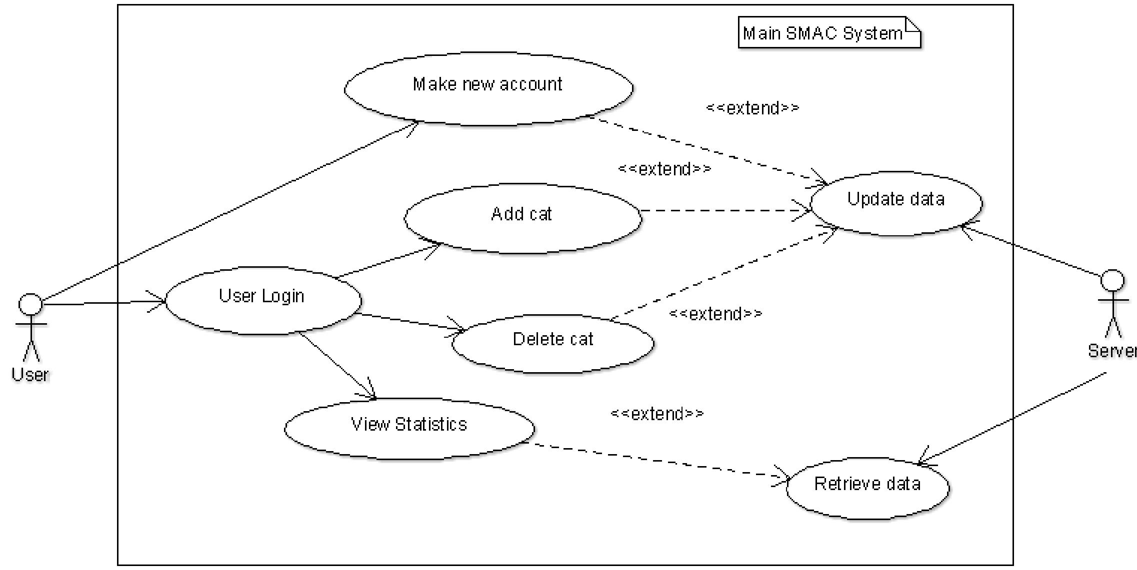

Figure 3.1: SMAC User Use Case Diagram

the user could accidentally delete a cat and thus preventing it from eating at home. To prevent this the user can only view the statistics of the system. For a user to login to the Android application they must first add in their information to the system. The information will be used to look up the user in the database and associate it with the phone. The user can then login and retrieve the statistics from the system.

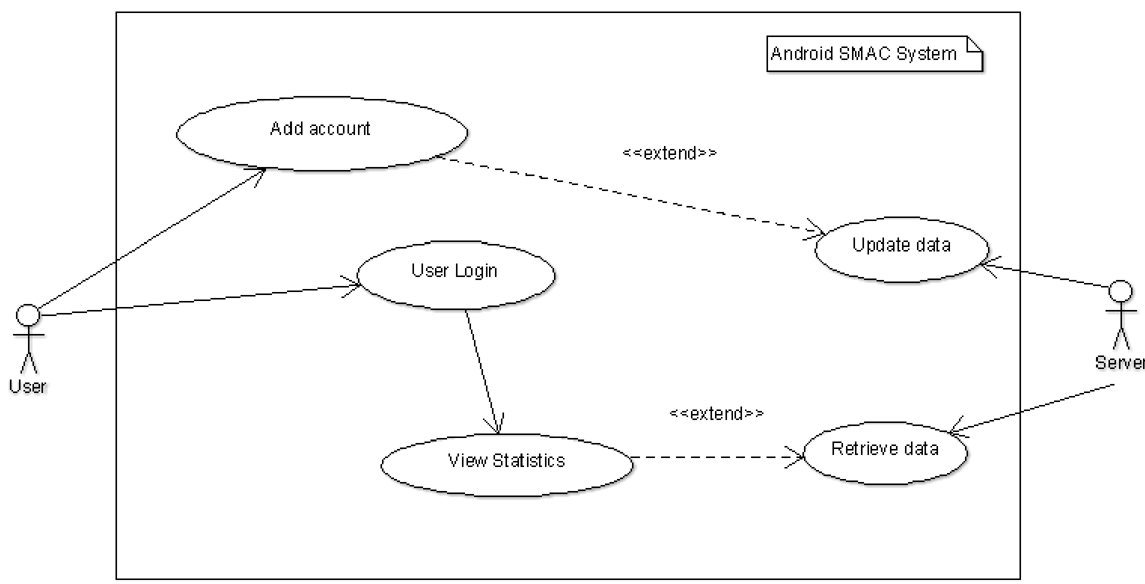

Figure 3.2: SMAC Android User Use Case Diagram

3.1.2 Activity Diagrams

Once the user has made an account and can login they can due one of the following three actions: add a cat, delete a cat, and view the statistics of the cats. The following three activity diagrams help describe the actions taking place during these three options. Figure 3.3 shows how a cat will be added to the system. Once the user logs in and chooses to add a cat to the system they will have to enter the name, age, and type of cat. The choice to either scan in a new RFID tag or use an existing one is presented. If an RFID tag is already in the system then the user can simply use it for the new cat. In the case that a new RFID wants to be used the user will be instructed to scan the RFID tag to enter it into the system. Whichever choice the user makes will lead to the final action of the system and server being updated with the new information. By updating both the system and the server allows for the system to know what cat walks up to the machine while allowing the server to associate the information received to the correct cat.

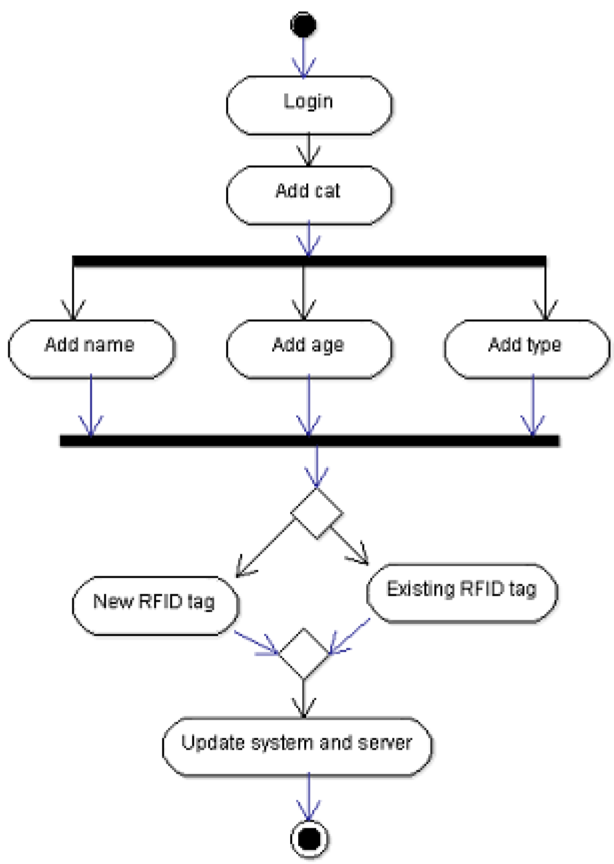

Figure 3.3: SMAC System Adding Cat Activity Diagram

The user also has the option of deleting a cat. This process is shown in Figure 3.4a below. The method is similar to adding a cat to the system where the only choice is about the RFID tag. Once the user decides to delete the cat from the system

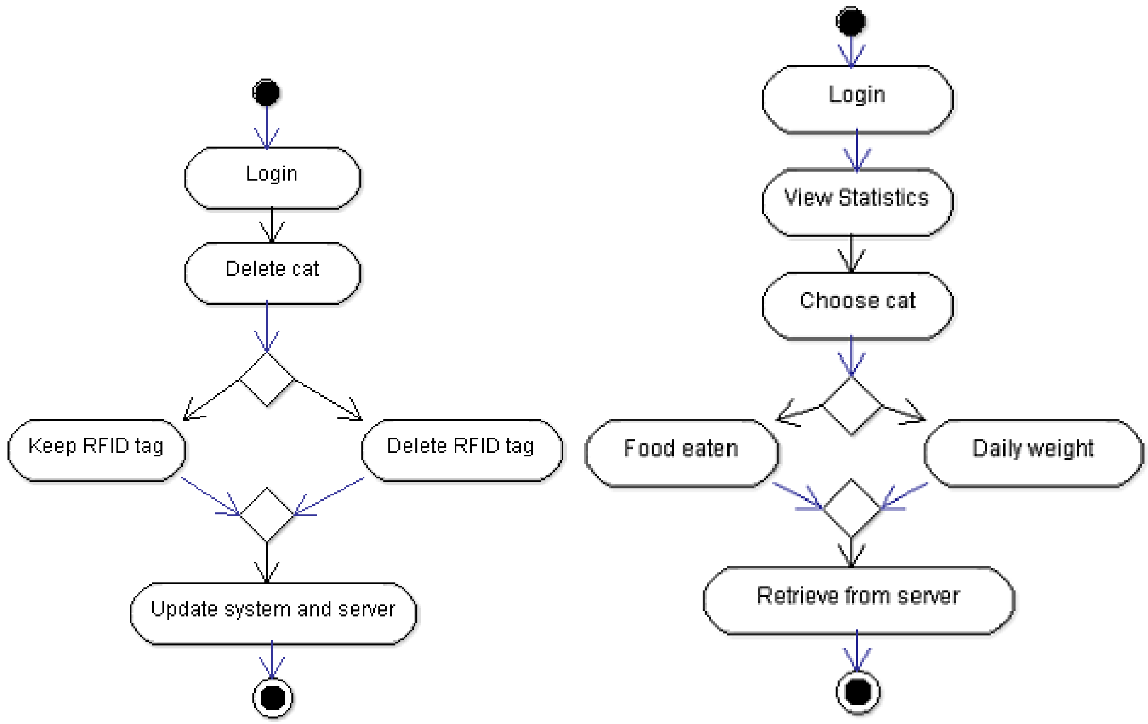

Figure 3.4: SMAC System Deleting Cat and Viewing Statistics Activity Diagrams

they have the choice to also delete the RFID tag information or to keep it on file. By having this option it allows the user the convenience of reusing the RFID tag for a different cat without having to re-scan it into the system. As with the previous activity diagram, after this decision is made, the system and server will be updated with the new information.

The last activity diagram, Figure 3.4b, shows the action of viewing the statistics of the cats. After the user logs in and chooses the statistics option they will then choose the cat to view, if there are multiple cats in the system. Since SMAC will keep track of the amount of food eaten and the daily weight of the cat, the user will have the choice which data they want to view. Once this decision is made the system will retrieve the information from the server and display a graph of the data.

3.1.3 State Diagram

The main purpose of the SMAC machine is to feed the cat the proper amount of food. Figure 3.5 is a state diagram that best describes the process in which this will happen. The diagram starts at the RFID System Scans state. During this state the system will continuously scan for an RFID tag. Once a tag is scanned into the system it will check to see if there is a cat associated with it. If no cat has the particular RFID tag number then it will revert back to the previously state. If the cat is in the system the next thing to check is if the daily food limit has been reached. If the cat has reached its daily allotted food limit then it will not get fed. If the cat has food still left to eat then the next state checks to see if any food is currently in the food dish.

If the dish is empty the correct amount of food will be added and if the food is added correctly the cat will then get fed. If there is food left in the dish from a previous meal time then the machine will check if the amount of food will reach the cats daily limit. If this is the case then the cat will get fed the remaining food, and if not the next state is the Add Food state, already described above. Once the cat has been fed the next state is the Cat Gone state. This state will continuously check to see if the cat is still eating. Once the cat is done eating the dish will close and the system will return back to the first state, the RFID System Scans state.

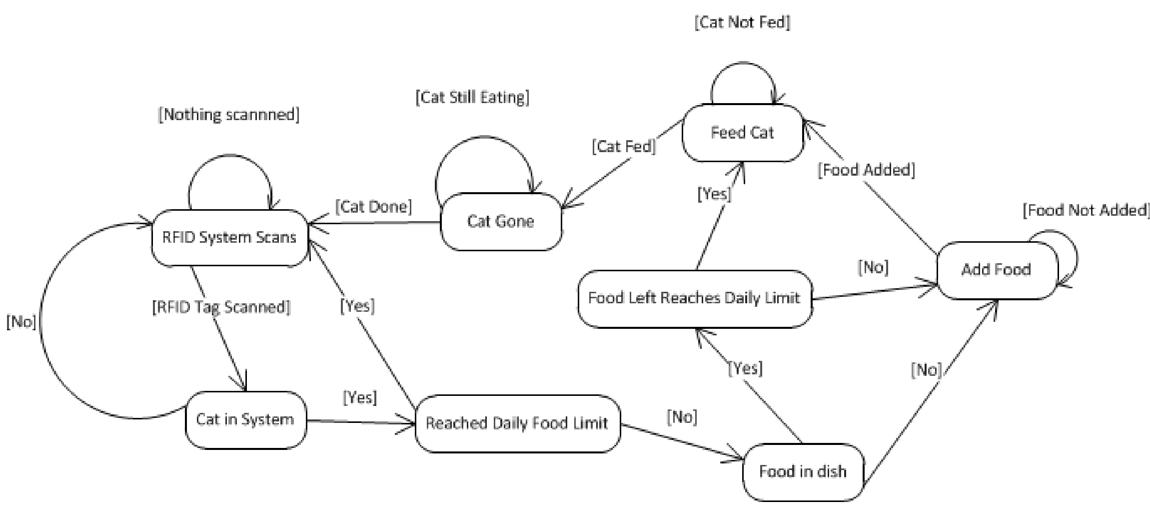

Figure 3.5: SMAC Cat Eating State Diagram

3.1.4 Class Diagrams

The following two class diagrams help describe the relationship between some of the key elements in the SMAC system. The first and most important is the storage of the cat information. The design of Figure 3.6 will allow for the easy expansion and/or modification of a particular piece of the code without affecting the rest of the code. This is very important when coding a large project since it helps cut down on errors. The second helps describe the relationship between the motors and the main system. It is very important to keep the majority of the code for moving the motors in one place. By doing this, again it will help cut down on errors, but also make the code smaller and more efficient.

Figure 3.6 below shows the class diagram that represents the cat in the SMAC system. The main class is the cat class which inherits the animal class. The cat class will hold only the breed of the cat and the required daily calories. It will contain a method for getting and setting these values. The animal class will be where the age, name, and current weight are contained and managed. This would allow for the future expansion of SMAC to other animals if it was wanted.

The animal class will make an instance of the food class and in tern an instance of the first scale class will be made. The food class holds the type of food currently in SMAC. It will also hold the total daily amount eaten by the cat. Each day the food class will get called to add the daily food amount to the array that holds the statistics for the food. This will simply be an array of integers that correspond to the oz of food eaten on each day the cat ate since being added to the machine. The first scale class will hold the major values for the food scale. There will be methods to get and set the current, previous and goal weight for the scale. The next step would be to feed the cat. To accomplish this each Animal class will also have an instance of the Feed class, described below.

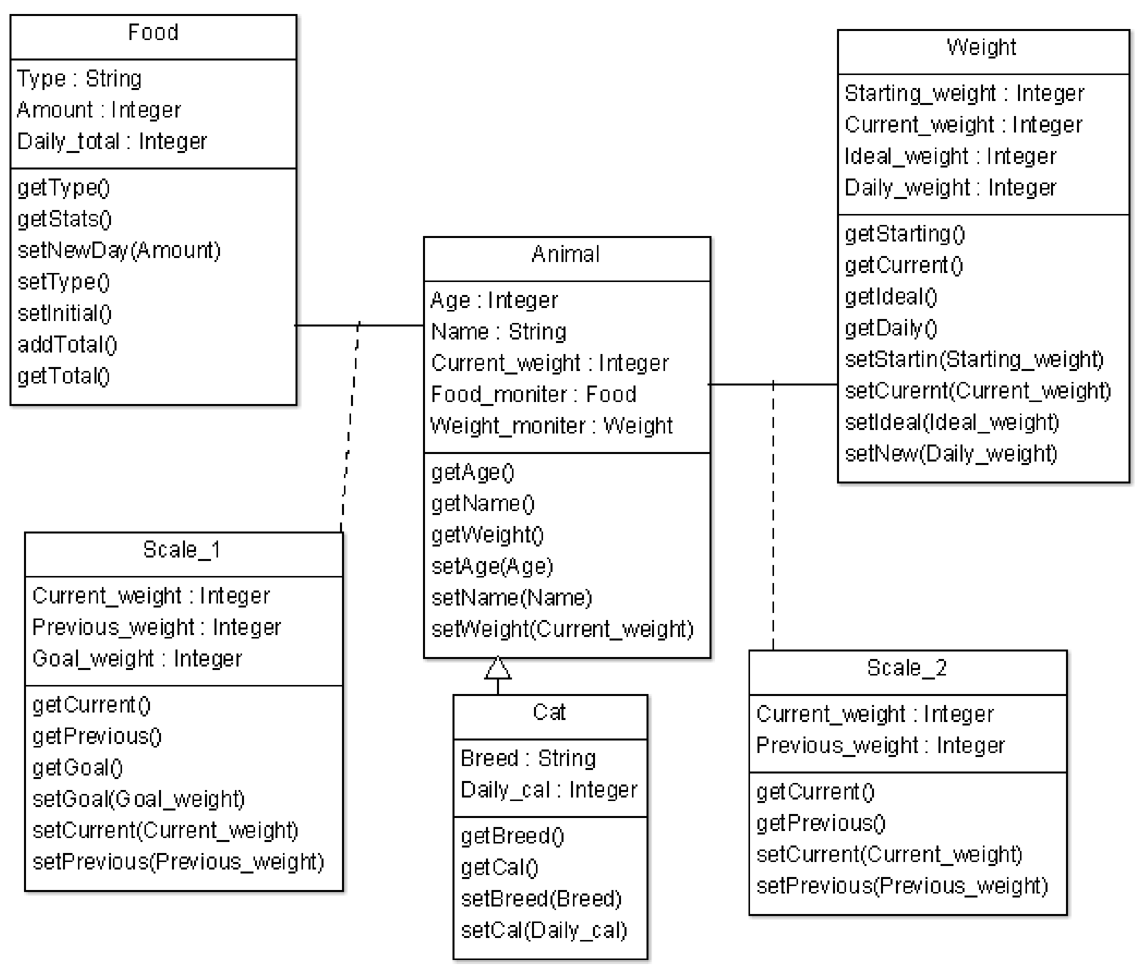

Figure 3.6: Cat Class Diagram

The animal class will also have an instance of the weight class. This class like the food class will require an instance of a scale class, but with a slightly different design. The scale class for the weight of the cat will only have the previous and current weights. There is no need for a goal weight since nothing is being measured out, just recorded. The weight class will hold all the information of the cats weight. This includes the starting, current, ideal, and daily weights of the cat. Like in the food class, the daily weight variable will be an array of integers keeping track of the change in weight over time. This will allow the owner to keep track of the cats health.

The code to move the two stepper motors that are going to be used in SMAC need to be efficient. The following figure, Figure 3.7, show the relationship between the motors and the rest of the SMAC system. The only parts in SMAC that will call the appropriate methods to move the motors will be the FoodMotor and ScalMotor classes. The FoodMotor class will move the motor that dispenses the food onto the scale in the food dish. It will consist of only two variables, the goal and current position. These variables can be set and retrieved with the simple functions seen below. The main function is to move the motor to the goal. This function will move the motor step-by-step until the correct weight is read on the food scale.

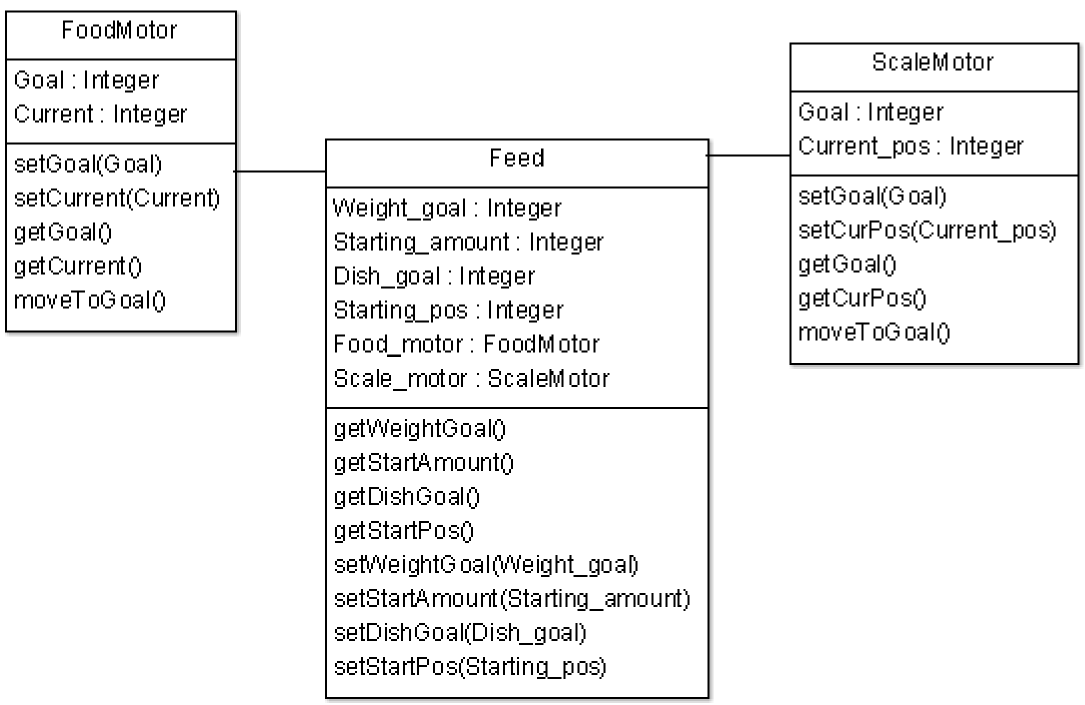

Figure 3.7: Motor Class Diagram

The ScaleMotor class is very similar to the FoodMotor class. Instead of turning until the correct weight is reached, the motor will turn until the correct food dish is accessible to the cat. It is very important to keep track of the current position of the food dish at all times. If the position of the dish is lost, then there is a possibility that the cat will not get fed. Because of this, the current position will always be kept, and incremented with each step of the motor, to minimize the chance for error. The Feed class will have an instance of each of the motor classes as well as variables to keep track of how much food to feed the cat. The main program will make an instance of Feed for each cat. By doing this the goal weight and dish goal can be set once in the beginning and would alleviate the need to reset them each time a different cat needs to be fed.

3.1.5 Database Design

The database for SMAC will hold all the information related to a user as well as a table of cat food and cat breed types. To best describe the relationship between the tables in the database two diagrams are needed. Figure 3.8 is the connection relation that relates the SMAC, the User and the Phone through their key attributes. The second figure in this section, Figure 3.9 shows the table Cat and how it relates to the other tables in the database.

If SMAC is to use a database properly then some constraints need to be put in place. SMAC shall update the database on a daily bases, or when important information is changed. Some of the immediate updates would include, any of the notification causes, a change in the amount of food, and the adding and deleting of a cat. By constantly updating this information the user can instantly access the data from either SMAC or the SMAC Android application.

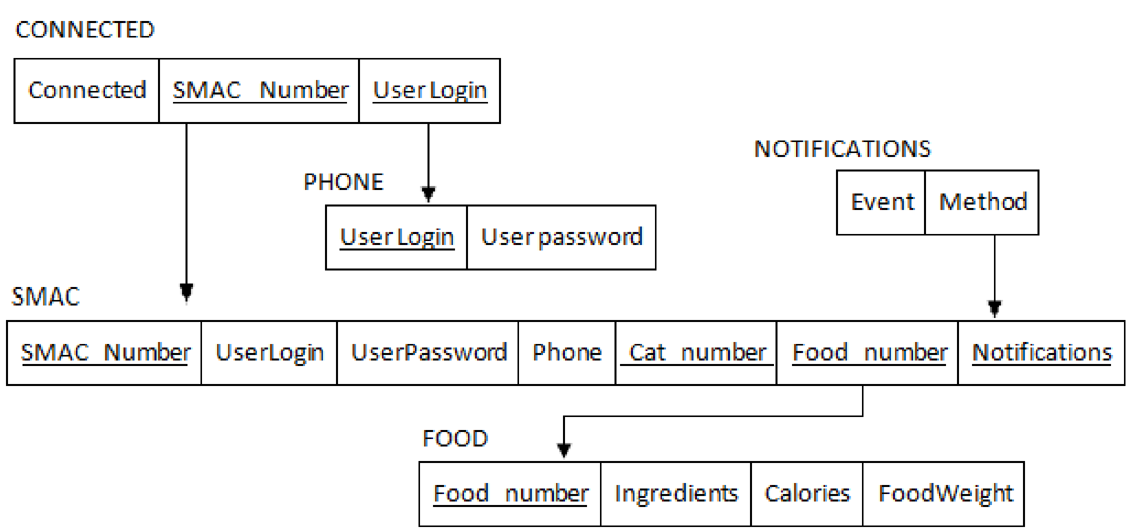

Figure 3.8: SMAC Connected Relations Tables

The key attributes in Figure 3.8 can access the rest of the tables and attributes in the database. The SMAC Anroid applications password attribute can be accessed through the User_Login key. The SMAC table is accessed by the SMAC_Number key. This table holds the users' login information, the phone number for notifications, the cat and food information, as well as the notifications information. Notifications is a multi-value attribute, meaning that there will be many notifications that can take place at one time. It is simple a table to hold all the options of notification the user chooses based on a particular event. The key attribute Food_Number that links to the Food table. This key looks up the specific type of food that is in the SMAC feeder. The Food table holds the ingredients, calories, and average weight for thirty ounces of food.

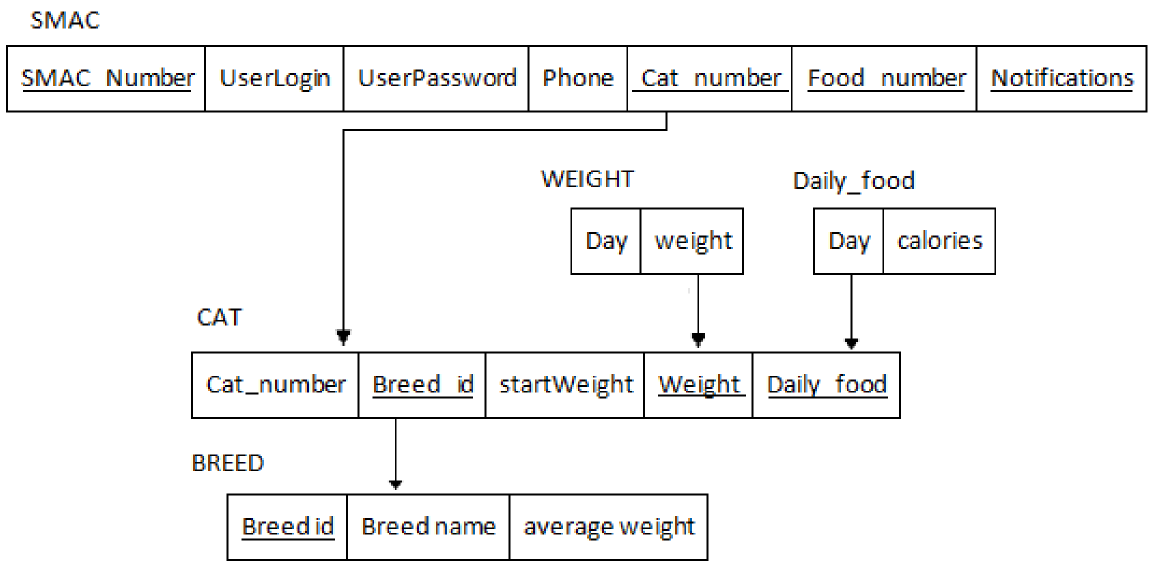

Figure 3.9: SMAC Key Attributes Tables

The most important table in the database holds the information about an individual cat. The table of cats is accessed by the multi-value attribute, Cat_Number, in the SMAC table. The Cat table holds information about the breed, starting weight, daily weight, and daily food eaten. The Breed_id key will be used to look up the particular breed in the Breed table. This table, as well as the Food table, will be made prior to SMAC being released from information gathered from companies. They will only be updated if a SMAC user needs to add a type of food or cat not in the system. To keep statistics on the cats weight and food consumption a multi-value attribute needs to be had in the database. The Weight and Daily_food are these such attributes. They simple add to the table each day with the weight, or calories consumed by the cat.

3.1.6 Server

The server will manage the database that connects the SMAC phone application and the SMAC main system. Both systems will connect to the server when requesting or adding information to the database. The server structure is very simple, since it only connects two different systems, and contains one database. The most important feature is to not allow any retrieval or changing of SMAC user account information other than the administrators. Making each user have a user name and password makes it so that they can only access their own information through the server.

Figure 3.10 shows how the server connections take place between the different systems. The main SMAC system will connect to the server any time the user needs to make a change or retrieve information from the database. Since the phone application only requires the retrieval of information, there is no need for the phone to ever connect to the server to try to change anything. If the phone tries to edit any data on the database through the server, the server shall block this action and lock the account.

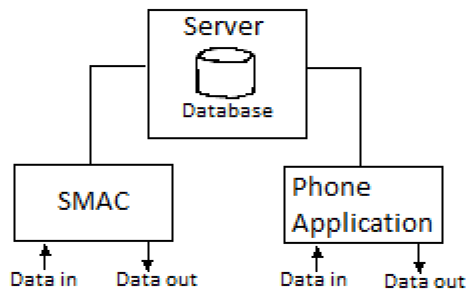

Figure 3.10: SMAC Server

3.1.7 Notifications

The notifications system will be set up in a way so that the user can select how they want to be notified, and of what. There are two options for notifications, the user can receive a message either through text or the Android application. When the main SMAC system gets the information that the user requested to me notified of it will connect to the server. Through the server it can retrieve information from the database. The method of notification will be looked up by event the via the users' notification table. Once this information is retrieved the program can begin the notification process.

If the method of notification is via text message then the Java program will send an email to the users phone number. When the user signs up for this option they would be required to provide their carrier information. This information, described in section 2.12.5, would determine which email address will be used to send the email as a text message. To send a message to the android application, the information will be sent through the server. Once the Android application receives the information it will alert the user they have a new notification waiting. When the user opens the application the notification will be the first thing to appear.

3.2 Food Dispenser Design

After thorough research, the best design for the SMAC will be the turning mechanism. The reason this option was chosen was because it will provide a more accurate way of dispensing the food. The turning mechanism also eliminates the option of the food dispenser getting jammed with food as was the case for the sliding mechanism.

Each side will be 2 inches will a total cross diameter of 4 inches. The plastic container where the dispenser will be placed will have a cross diameter of slightly more than 4 inches to allow the machine to rotate. The measurement must not be too big though because it will then allow pieces of cat food to slip through making the cat food inaccurate.

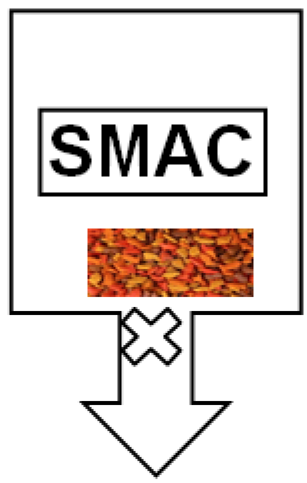

Figure 3.11: Food Container for SMAC showing dispenser

Turning Mechanism - The first trials with the paddle wheel turning mechanism will be measuring of food being dispensed at one time. Every trial turn should dispense 1/3 cup of food at a time to be accurate. A typical cat gets a cup of food per meal, two meals a day. So that equates to a typical cat receiving 4 turns of food for 1 cup of food in the tray. The angle of rotation will be the output from the stepper motor and is measured at ninety degrees. The micro stepping input that will correspond to the ninety degree rotation output will be tested for precision and accuracy.

Initial Position- The initial position of the dispenser allows the top food tray to get filled with 1/3 cup of food. The left tray will be the temporary storage location for the food after it is filled and rotated. The bottom tray is where gravity pushes the food down into the food bowl to be weighed out. The right food tray will remain empty as it just dispensed food and is now waiting to be rotated back up to the top to receive food again.

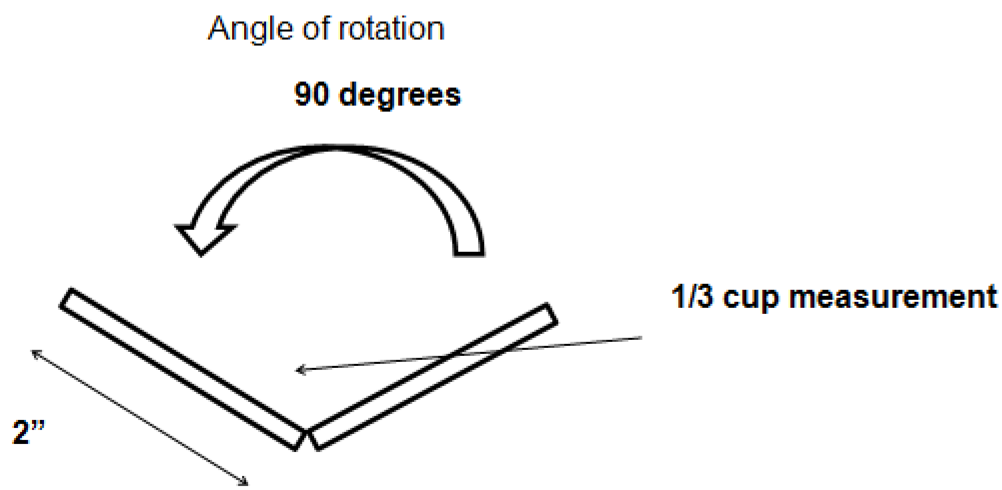

Figure 3.12: 1/3 cup measurement

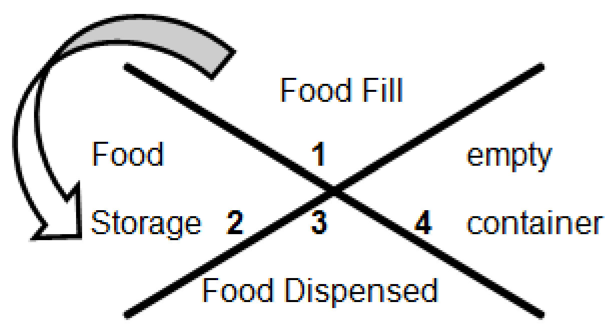

Figure 3.13: Initial Position

First Rotation - The first rotation of the food dispenser takes the food that was filled from the initial position and stores it in the temporary location. The second tray is now available to be filled with 1/3 cup of food. The food dispenser is now loaded with a total of 2/3 cup of food and will remain loaded with 2/3 cup of food at all times.

Second Rotation - The second rotation of the food dispenser takes the food from the first rotation and drops it down into the food bowl. It also takes the food filled from the second rotation into the temporary storage slot to allow the third cup to be filled with food.

Third Rotation - The third rotation of the food dispenser now drops the second tray of food down into the bowl totaling a total of 2/3cup in the bowl. The food from the third rotation is now moved to the temporary location to allow the fourth cup to be filled with food.

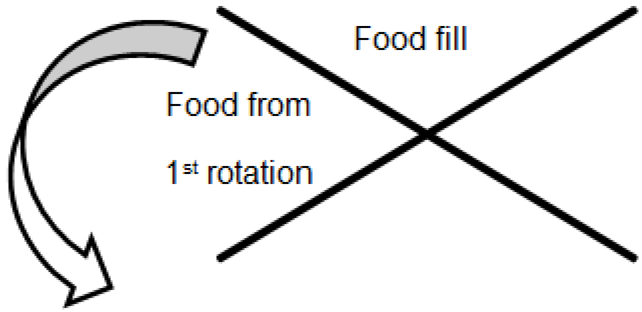

Figure 3.14: First Rotation

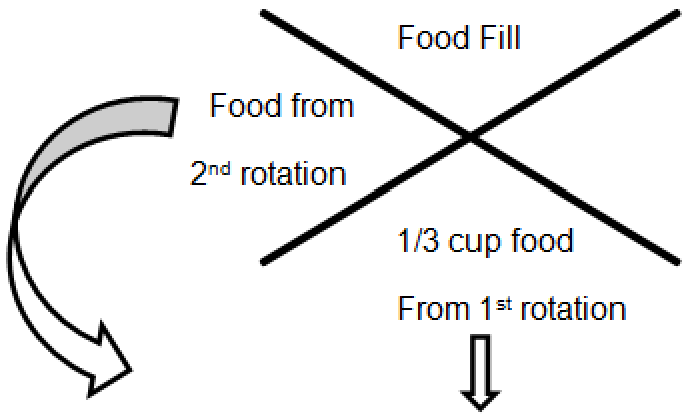

Figure 3.15: Second Rotation

Fourth Rotation - The fourth rotation of the food dispenser releases the last third of a cup of food down into the bowl. The total amount of cat food now in the bowl is 1 cup. A typical meal for a cat is around 1 cup so once the food tray is filled the second stepper motor, for the food trays, rotates the food tray to the open slot to give the cat access to the food.

The Adafruit small stepper motor for the food dispenser will be mounted to the back of the plastic food housing. The rotator for the stepper motor will be inside the food housing and the paddle wheel dispenser will be attached to it. The motor will be programmed for quarter steps, each step will produce a ninety degree rotation.

The speed of the stepper motor will be tested and adjusted to where it can be fully functional and notTo connect to the motor shield, the green and red wires are tied together and connected to ground. The brown and black wires are wired to one motor port with the orange and yellow wired to the other motor port.

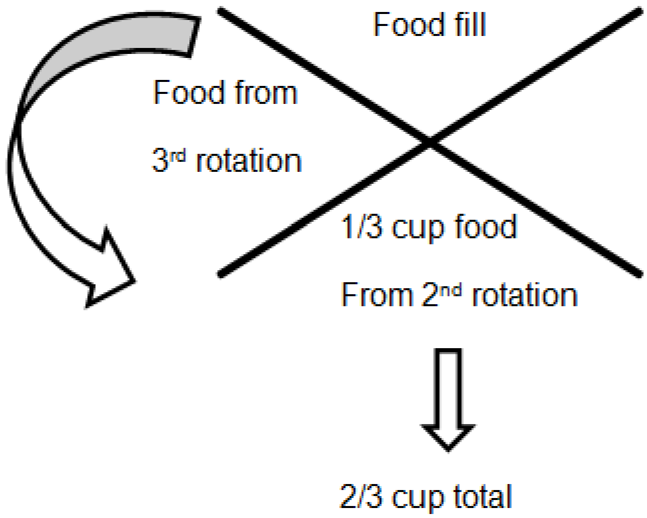

Figure 3.16: Third Rotation

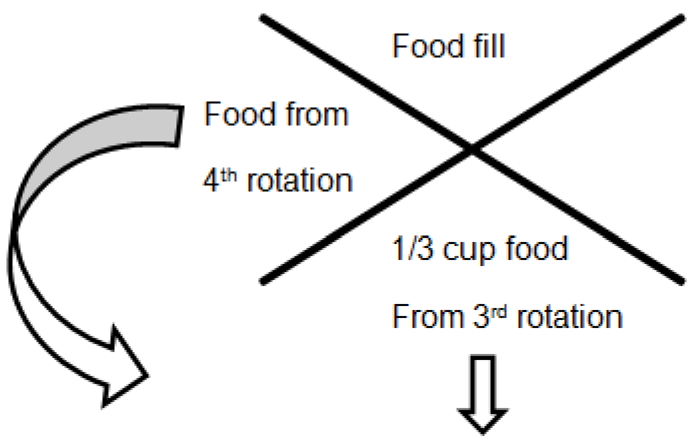

Figure 3.17: Fourth Rotation

The plastic housing around the paddle wheel dispenser will have to be slightly larger than the radius of the paddle wheel. It cannot be too large or else cat food will slip out. A typical piece of cat food in about 20 mm in length so the available gap range between the dispenser and plastic housing is between 1 and 19 mm. This gap range will be tested to determine the optimal gap distance for easy rotation and no food slippage.

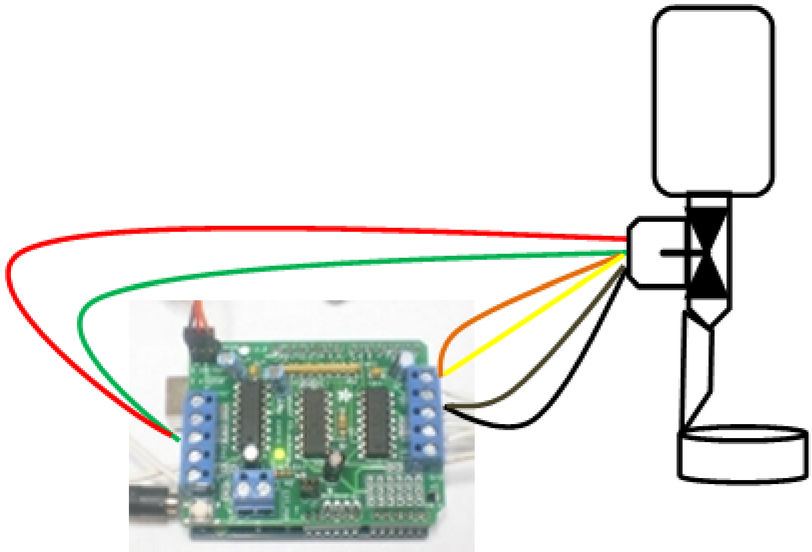

Figure 3.18: Stepper Motor Hookup

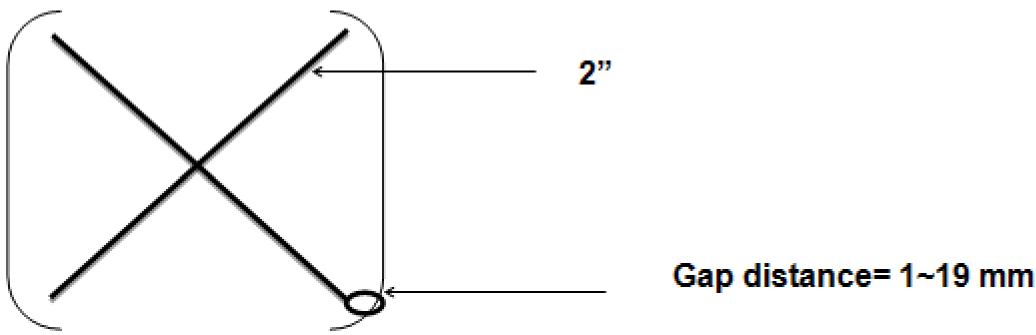

Figure 3.19: Paddle

3.3 Cat Scale Design

The cat scale will consist of 4 individual digital scales placed apart in a rectangular shape with a 1 foot X 2 foot sheet of glass placed on top. The four individual scales were chosen for the SMAC because it will provide the most accurate reading of the cats weight. This task is accomplished by providing a scale for each paw of the cat. When the cat steps on the scale, the four scales will balance out and send the weight data to a microprocessor for conversion. The four scales will be connected to a microprocessor as four data input lines. The four inputs will then be added together to equal the cats total weight.

Software code will take the digital inputs and convert them to binary numbers. Then an add function will be executed and the resulting output from the microprocessor transmits the data. This final number for the cats weight will be displayed on the LCD touch screen as well as stored in the online database. This data can be accessed at a later time and viewed on the LCD screen. This function allows the owner to keep track of the cats weight to determine if more or less food is needed for the cat to maintain a healthy diet.

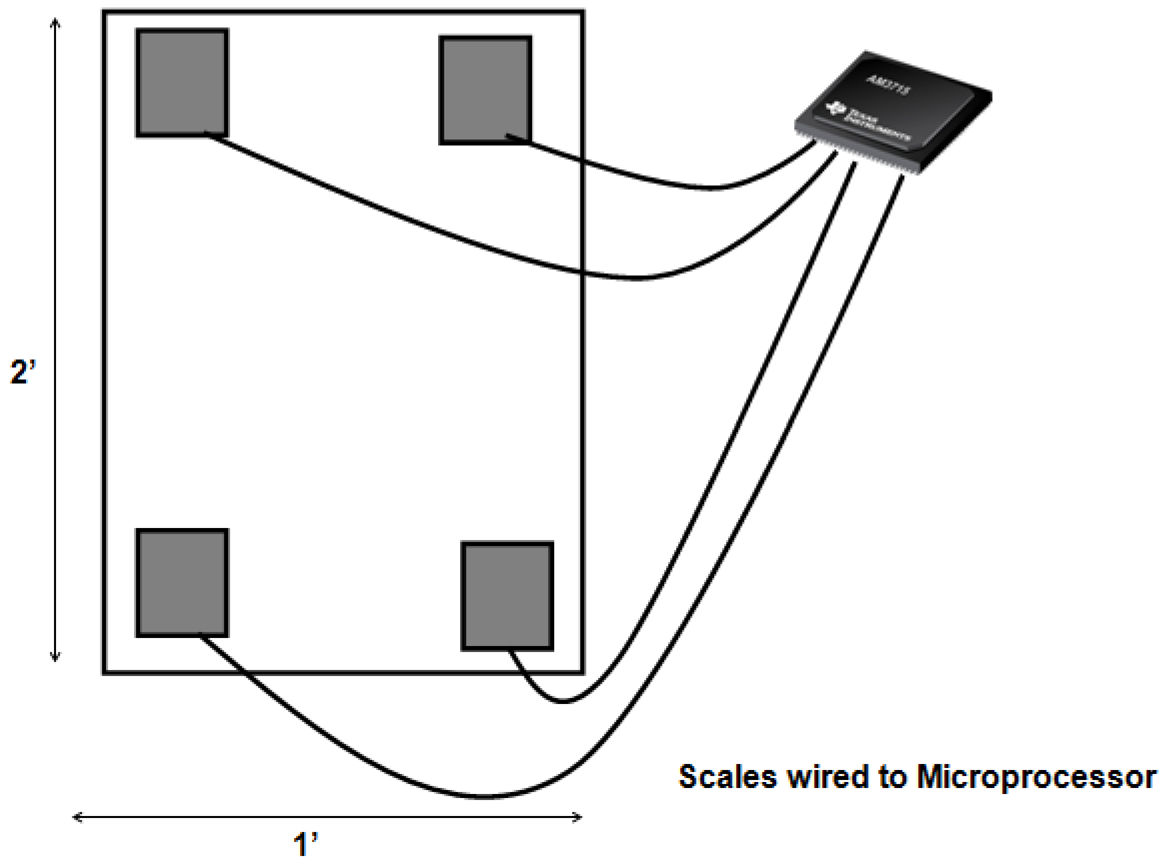

Figure 3.20: SMAC Main Page

The back of the scales must be removed to be able to gain access to the circuit board. The wire that connects to the numeric LCD module screen is the output wire containing weight information from the scale reading. This is the wire that needs to be connected to the microprocessor. Since the wire already used wont reach far, a few extra feet of 22 AWG wire will be used to run from all four scales to the center housing where the microprocessor is located. Each scale must be calibrated before using in the SMAC to maintain precision.

Once the glass top is mounted on top of the scales, the scales will balance out and then need to be tared to obtain net weight and provide a 00.00 reading with the glass on top of it. Now the scale can be tested by weighing the cat several times to see if an accurate reading is maintained. The microprocessor will be connected to the PCB so the weight data information can be stored.

3.4 Food Dish Design

The food dish design consists of several parts: internal housing, external housing, removable food tray, and the food container itself which stores the food. The internal housing will be able to hold six removable food trays. The diameter of the total housing will be 12 inches with a height of 5 inches. The housing will be designed with six 6 inch metal brackets to form the skeleton of the housing. The center bracket will have all six pieces attached to it and will also have the food container connected on top. Each slot in the housing for

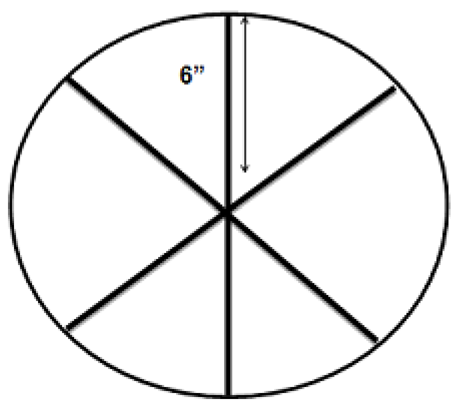

Figure 3.21: Internal Housing

There will six 1 cup removable trays that are 4.5in X 2in. Each tray will be assigned to a cat for consistency with the stepper motor. After four rotations from the food dispenser, the tray will be filled with a cup of cat food. The trays are easily removable so they can be run through the dishwasher as they will get dirty from the cat food.

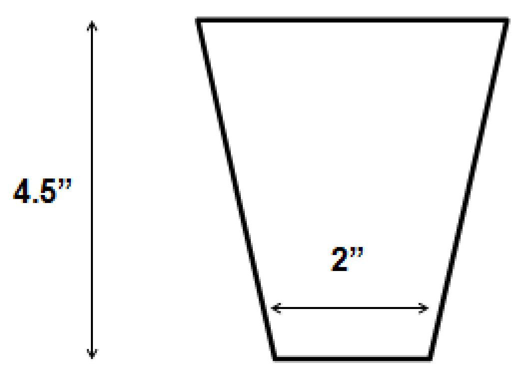

Figure 3.22: Removable Food Tray (1 cup)

The external housing for the SMAC will cover the food trays and only have one open slot where the cat will be able to access its own food tray. The dimensions for the external housing will be a diameter of 13in and a height of 6in. Allowing an extra inch for the diameter and height will make sure it is able to fit over the internal housing nicely. Unlike the internal housing, the external will be made of plastic.

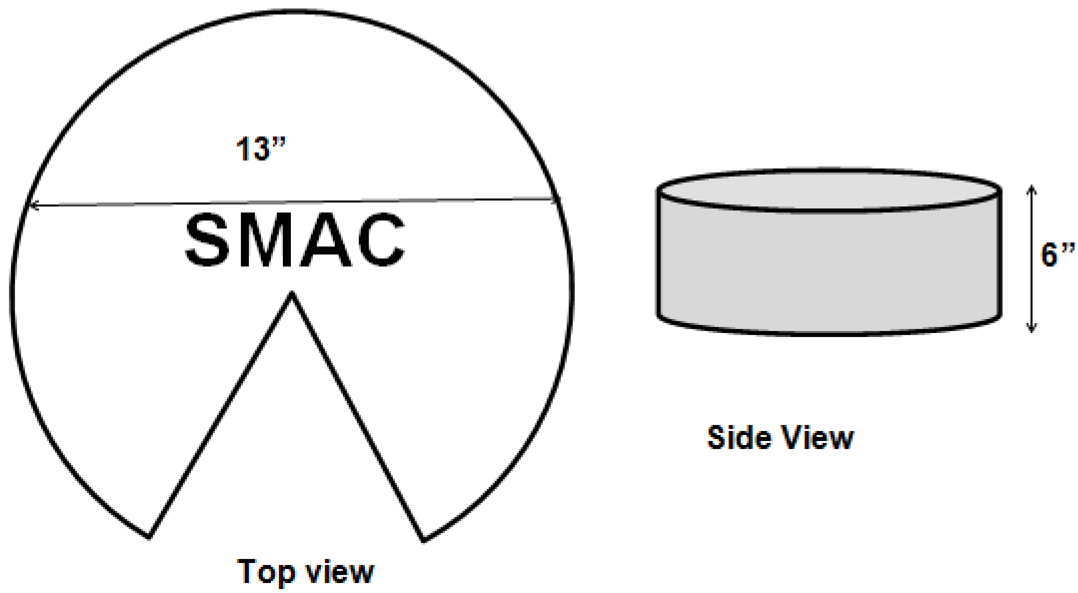

Figure 3.23: External Housing

The food container will be a plastic container holding up to 12 meals worth of dry cat food. The dimensions of the food container will be 12 in height X 10 in width. The stepper motor for the food dispenser as well as the paddle wheel dispenser itself will be mounted to the food container. The food container will be mounted to the top of the food tray mechanism. Mounted on top of the food container will be the twist off cap with the LCD touch screen attached to it. The wires from the LCD will run down the inside of the food container to be hidden from the user eye and give the SMAC a more professional look.

3.5 Control Unit

The PCB designed for the SMAC is modeled after the Arduino Mega 2560 development board. The Arduino Mega 2560 was chosen as the basis for the design due to the availability of Arduino compatible software as well as the compatibility of Arduino based boards with all hardware that will be connected to the SMAC. The touch screen, the RFID reader, XBEE Pro, proximity sensor, and motor driver were all found with code and attachment guides for Arduino based boards so it seemed to be the obvious choice for the model.

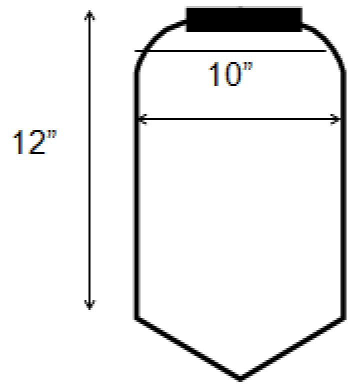

Figure 3.24: Food Container

The board will provide an ample amount of EEPROM, Flash Memory, SRAM, and more pins. The smaller boards do not leave much room for adding extra programming features that could improve the SMAC or very elaborate programming due to less memory capacity. Adding additional hardware with the limited I/O pins could be come a problem too. The mega board is not significantly pricier than the smaller versions either so it is worth the extra workspace. The hardware components required for the custom PCB are show in Table 28.

ATMega2560 Microcontroller
Standard USB Plug
DC Power Connector
Optional 3.3V or 5V Output Voltage
JTAG for debugging
FT232R

Table 28: Hardware Components

The schematic for the board and micro-controller are shown in Figure 3.25. This portion of the schematic, shown in Figure 3.26 is the input power circuit, including the 2.1mm DC power jack and an adjustable output low dropout voltage regulator. The voltage regulator provides the 5V output voltage that is required to power hardware attachments. The upper left portion of the microcontroller (Figure 3.26), which features the resonator, used for timing. It can be used as timer to make food bowl rotate back after cat has been away from the SMAC for a designated time. The upper right portion of microcontroller (Figure 3.27). This is the bottom left portion of the microcontroller, Figure 3.28. This portion consists mainly of digital output pins as well as a few PWM outputs that can be used to generate variable speeds in the motor. This is the upper right portion of microcontroller, Figure 3.29. This section features some of the PWM output pins and one of the status LEDs.

The bottom right portion of the microcontroller (Figure 3.30). Features the communication bus, more PWM output pins. This is the USB port Circuit. The FT232R enables the USB connectivity as it is a USB to UART serial converter. The right hand portion of the FT232R features two indication LEDâs and the power-on-reset button.

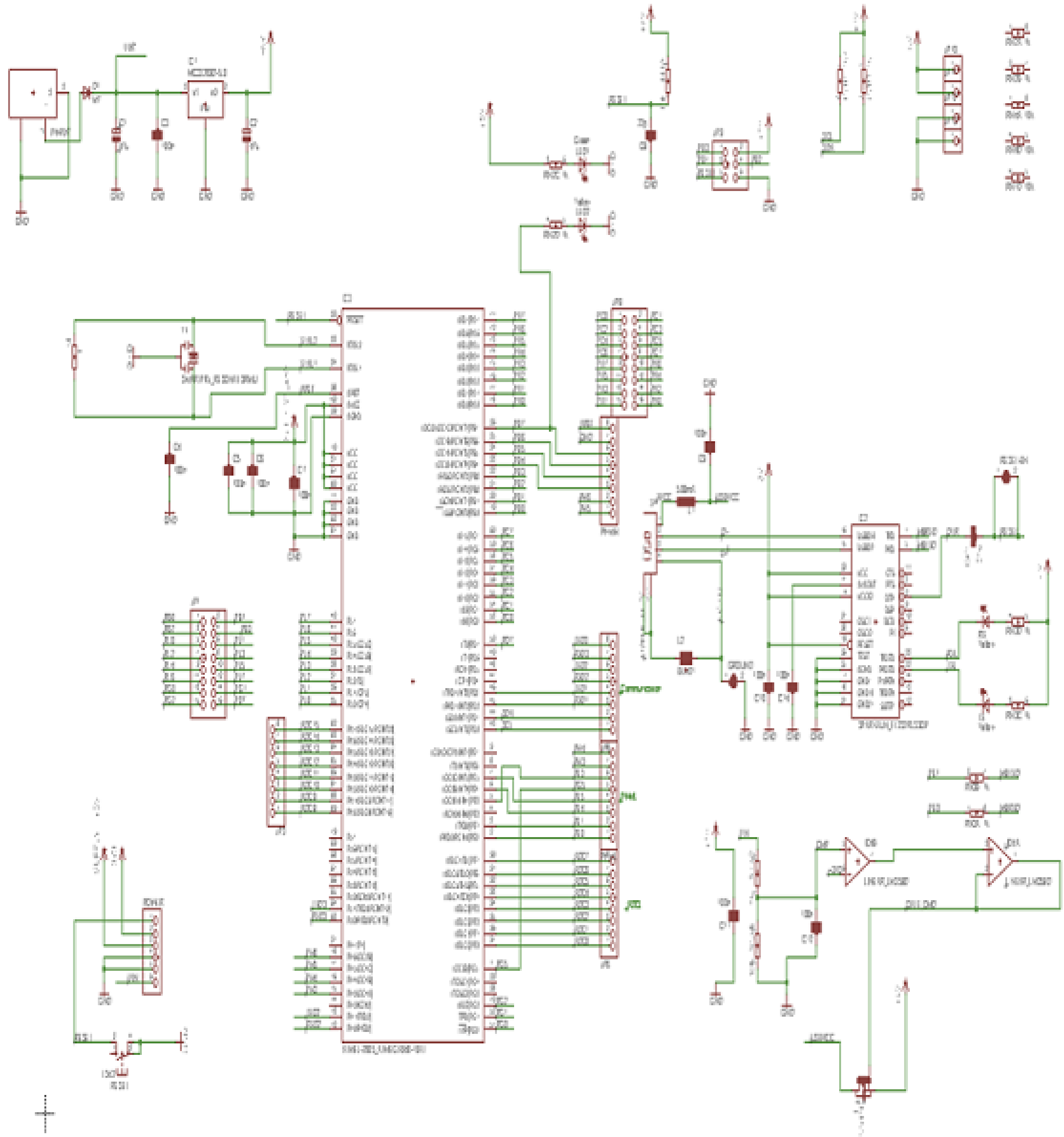

Figure 3.25: Entire Schematic

The hardware that will be attached to the PCB will include the XBEE Pro, the Touch Screen, the Proximity Sensor, the Motor Shield, and the RFID reader. The RFID Reader connection is shown below on an Arduino board. The RFID reader is very easily connected requiring only one input pin, a ground, and a 5V source.

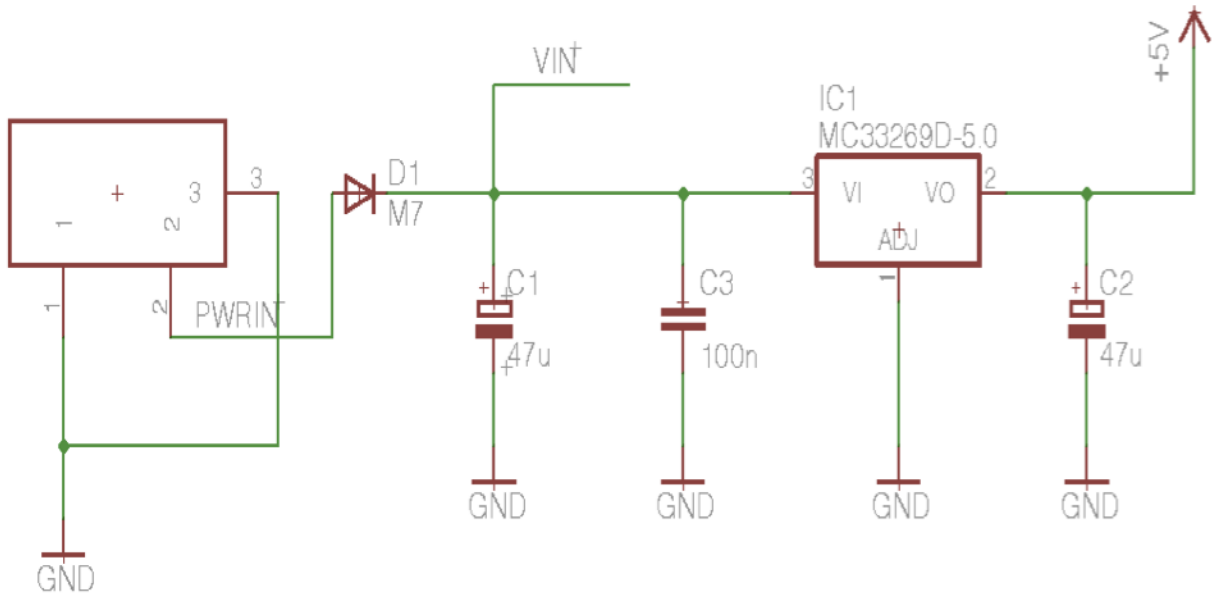

Figure 3.26: Input Power Circuit

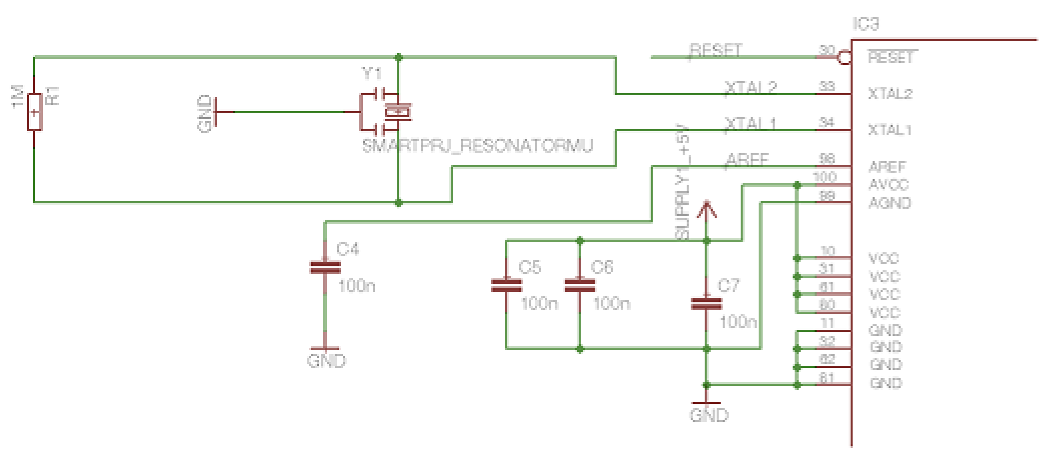

Figure 3.27: Upper Left Portion of the Micro-controller

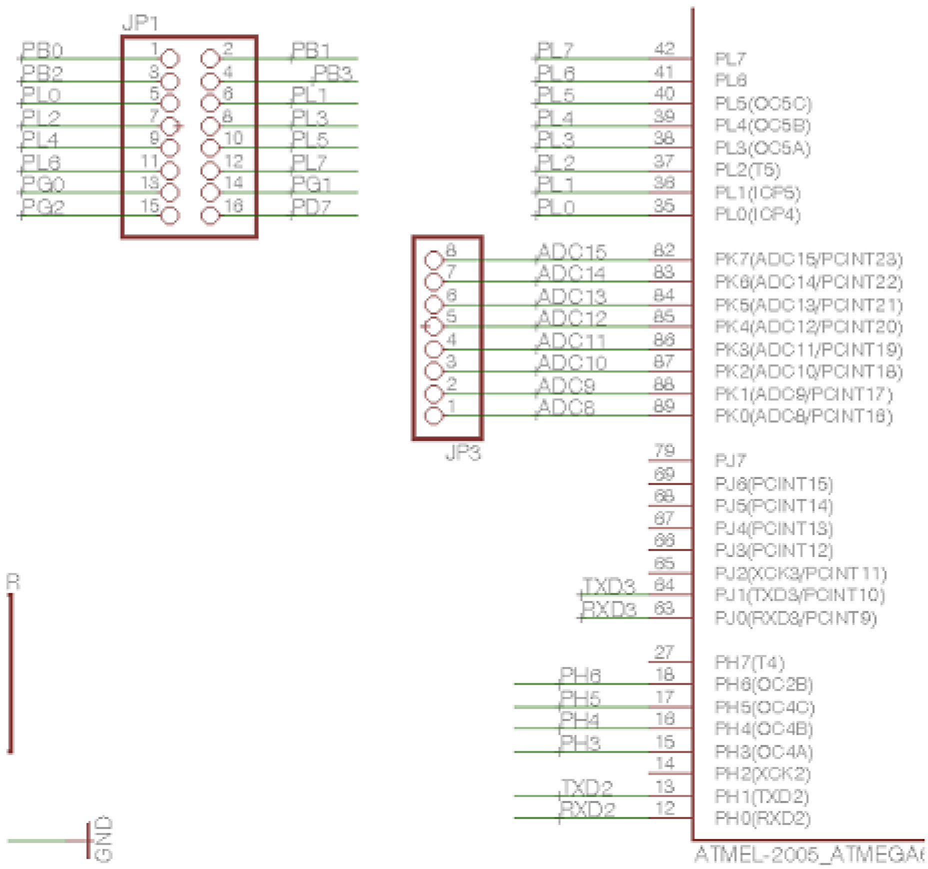

Figure 3.28: Bottom Left Portion of the Micro-controller

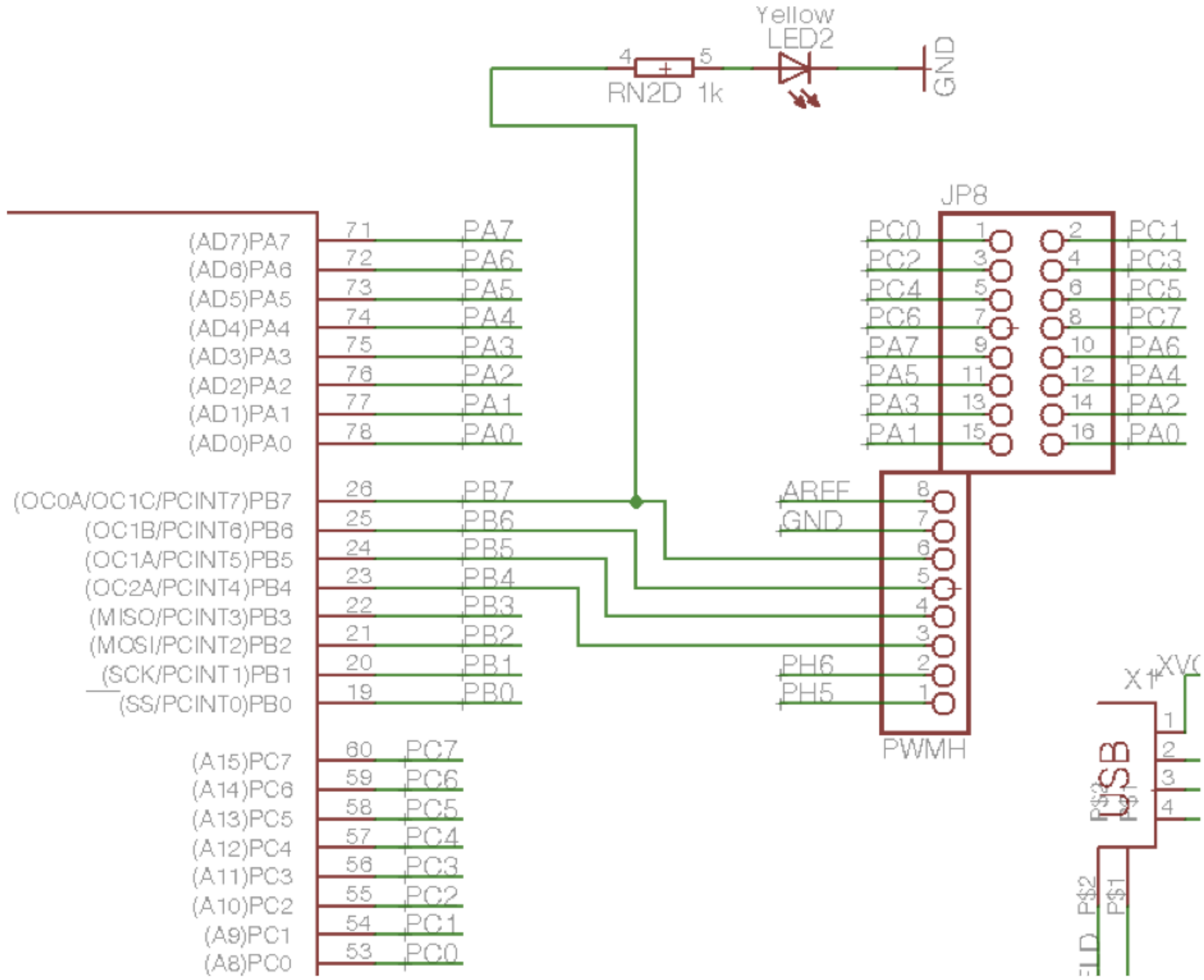

Figure 3.29: Upper Right Portion of the Micro-controller

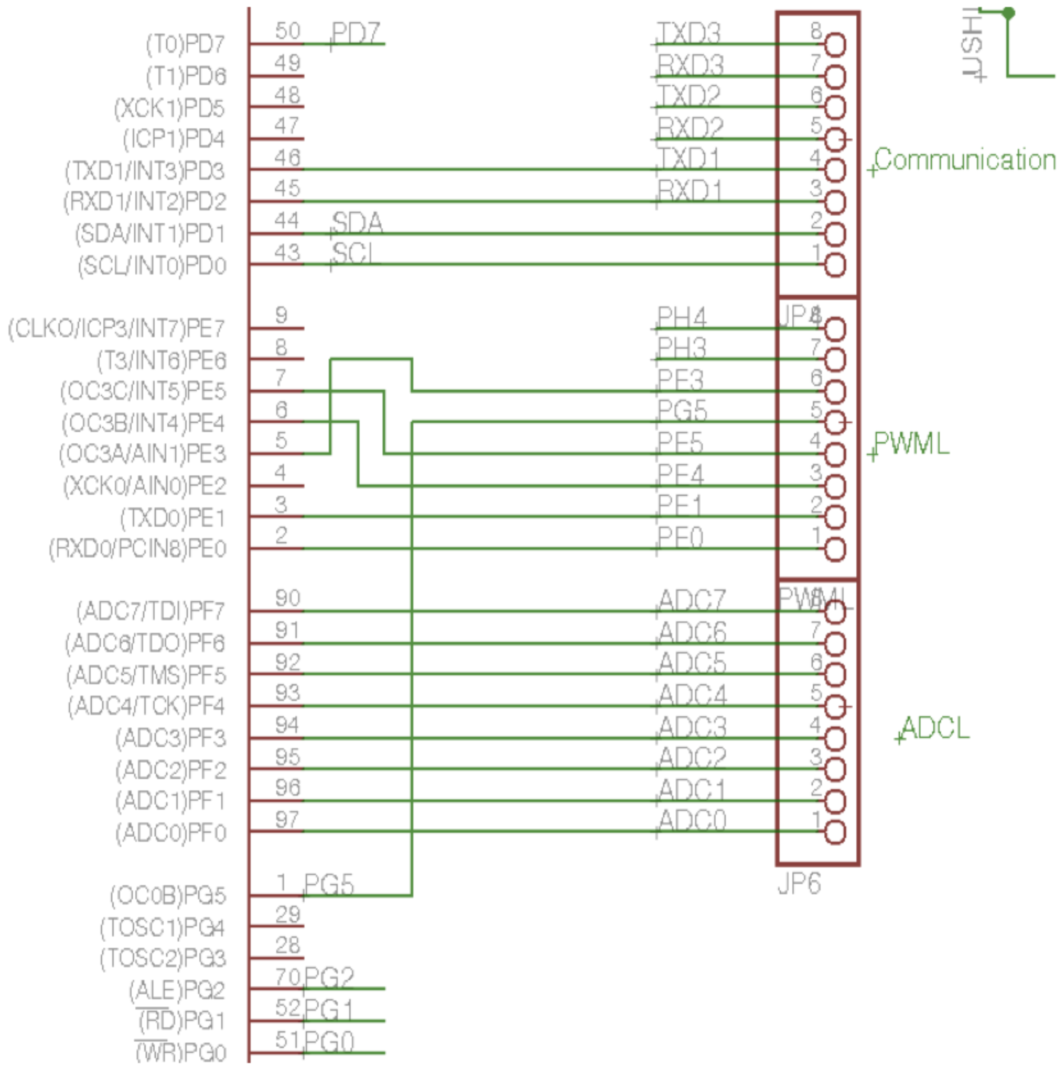

Figure 3.30: Bottom Right Portion of the Micro-controller

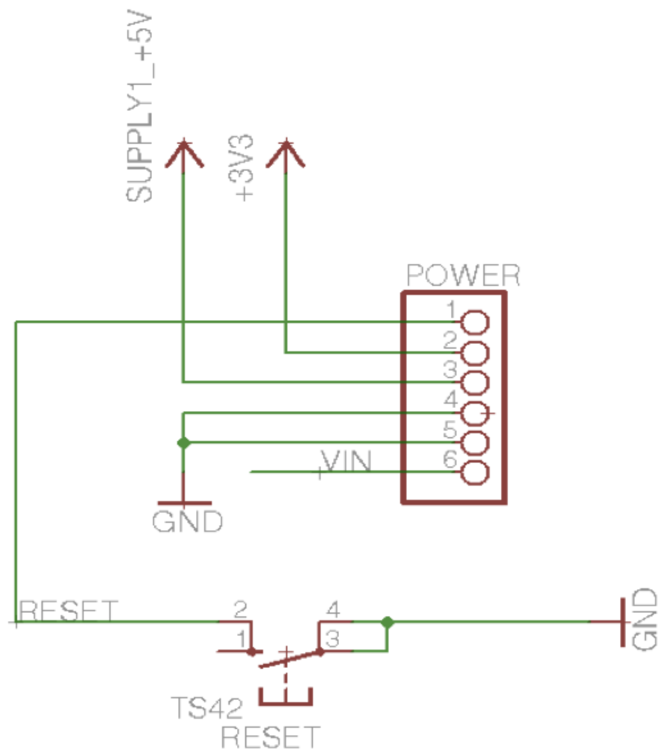

Figure 3.31: Power Bus Circuit

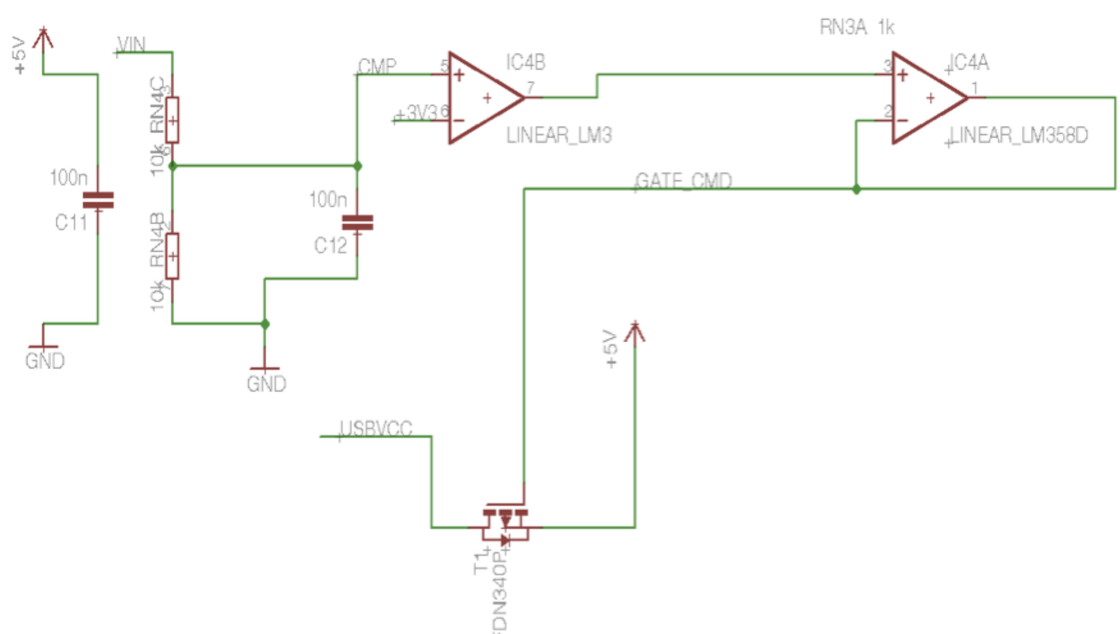

Figure 3.32: Linear Op-Amp Circuit

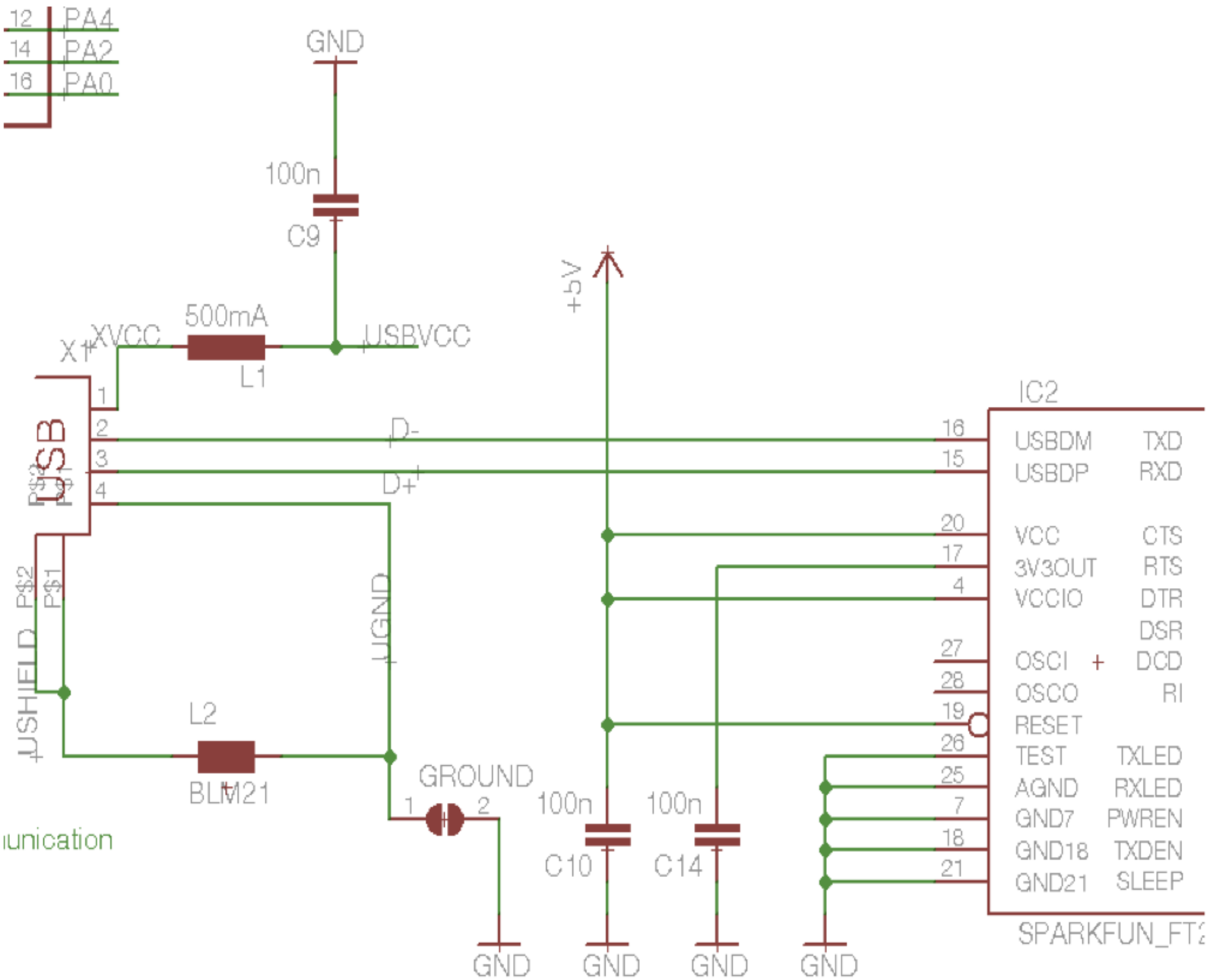

Figure 3.33: USB Port Circuit

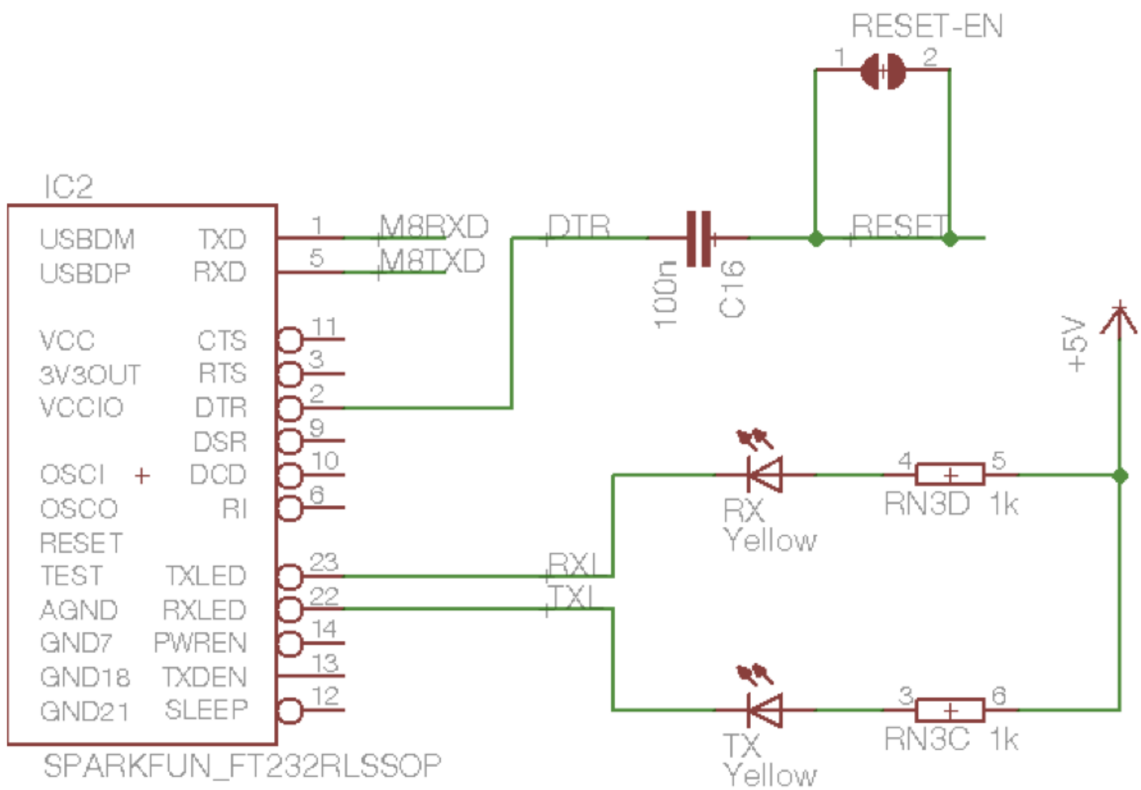

Figure 3.34: Right Hand pPrtion of the FT232R

The proximity sensor requires a 5V source, an analog input pin, and a ground. The resistor seen in the photo below is just to make sure that the proximity sensor is working, so it is optional. The Motor shield requires no hard wiring as it will sit on top of the PCB, shown in the picture below.

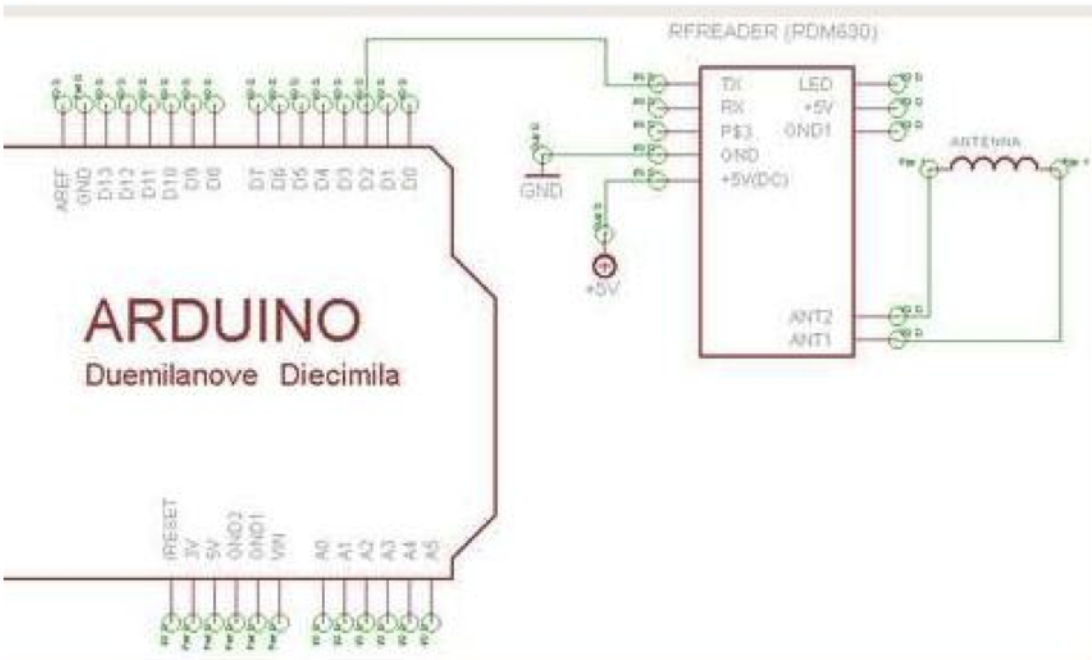

Figure 3.35: RFID Reader Connection

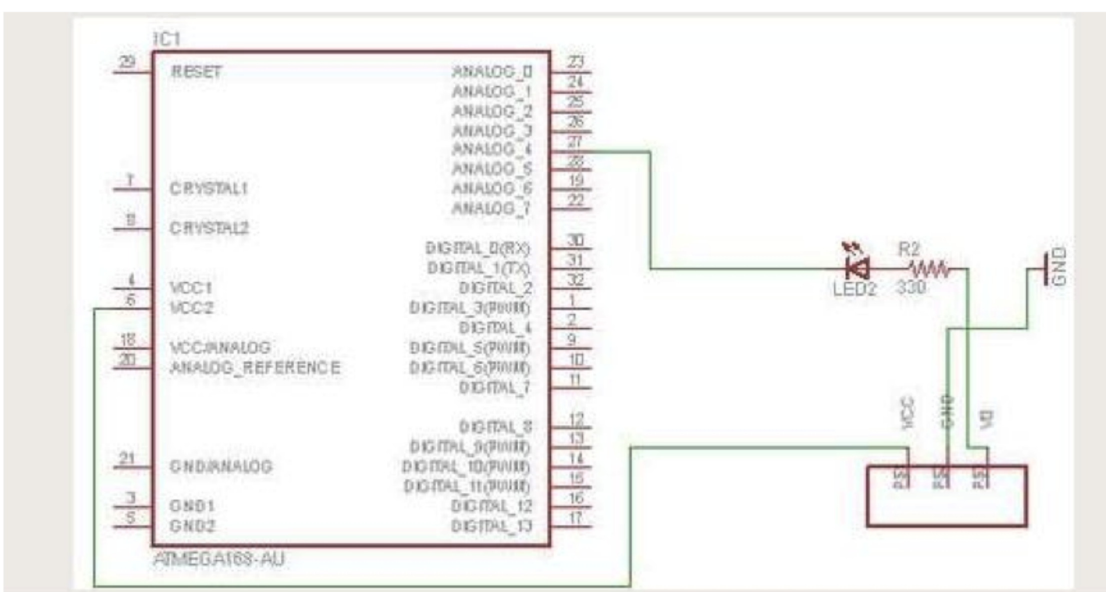

Figure 3.36: Proximity Sensor Connection

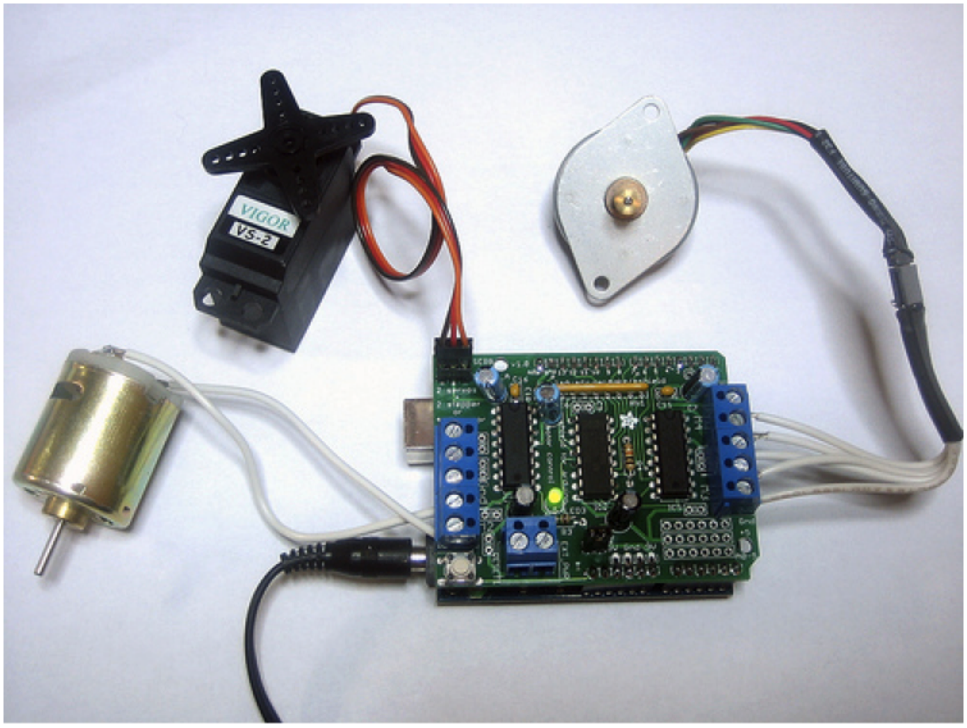

Figure 3.37: Motor Shield seen mounted on Arduino Mega board

To install the touch screen on the Arduino shield using 8-bit mode , which is the default mode, pins 31 and 37 of the touch screen must be shorted on the 40 pin connectors. Then D_IN is connected to SD_DIN, and D_DIN is connected to the Arduino. A bridge must be made between the terminals of the resistors leading to ports 8 and 9 of the circuit board. The 3.3V operated touch controller will now be able to communicate with the board. Now D_CLK must be connected to Analog Pin 1, D_CS to Analog Pin 0, D_P to Digital Pin 8, D_OUT to Digital Pin 9, and D_IN to Digital Pin 11. After a few lines of code the touch screen and the display are accessible directly from the PCB. The pin out for the touch screen is shown below. The XBEE Pro requires hard wiring but it is very simple. It requires a 3.3V source, a ground, and then 3 digital input pins. The picture below illustrates the connection, not to a circuit board, but it is the same concept.

4 Design Summary

This section summarizes all of the individual elements of the SMAC and explains how each part will contribute to the product as a whole. The only external part of the SMAC is the RFID tag that will be hooked onto the cats collar. The operation that kicks the SMAC into gear is when the cat approaches the SMAC and becomes within read range of the external antenna that is connected to the RF reader. When the RF reader recognizes the RID tag, the RF reader transmits a signal to the pro-

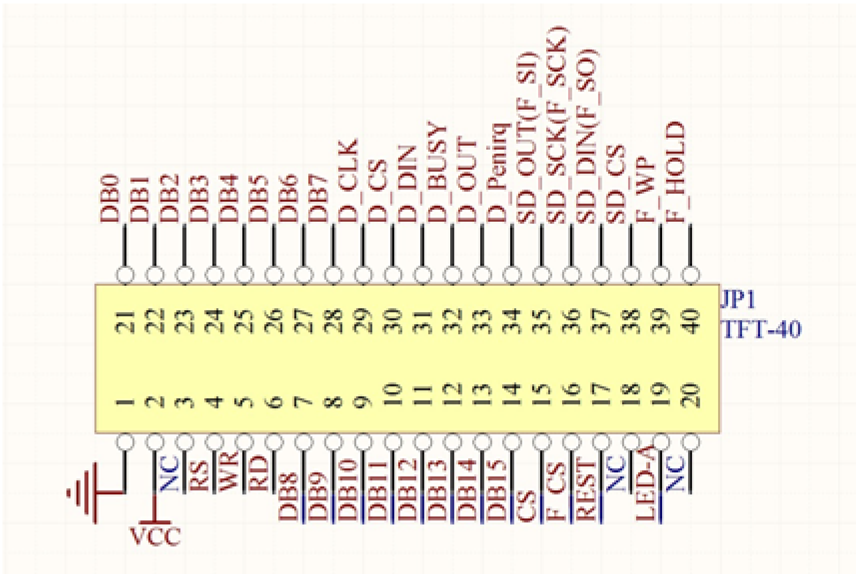

Figure 3.38: Touch Screen Pin Out

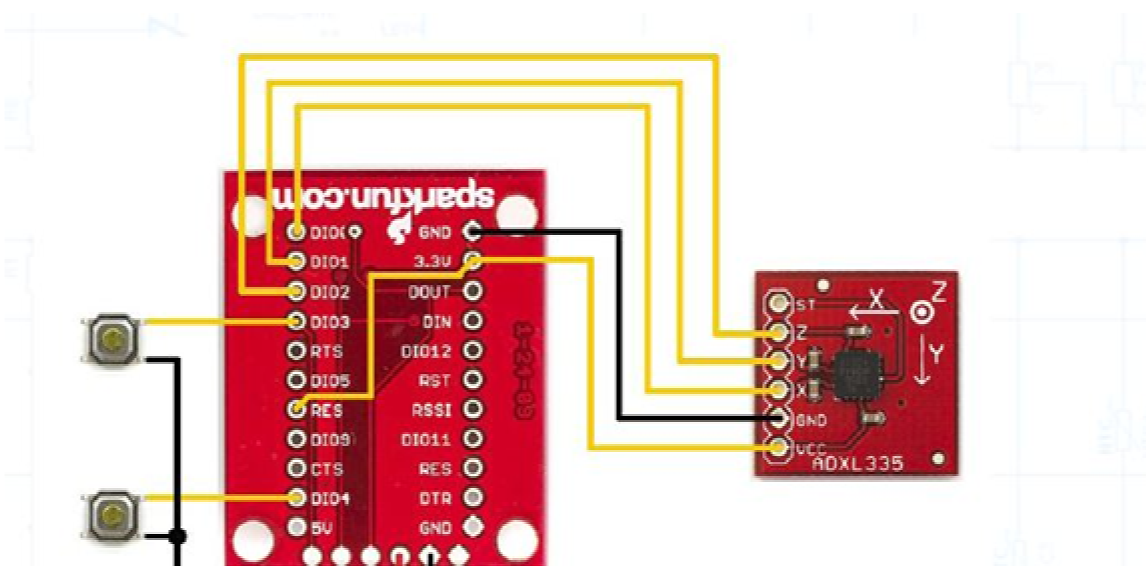

Figure 3.39: Connection for Digi XBEE Pro

cessor on the circuit board for further instruction.

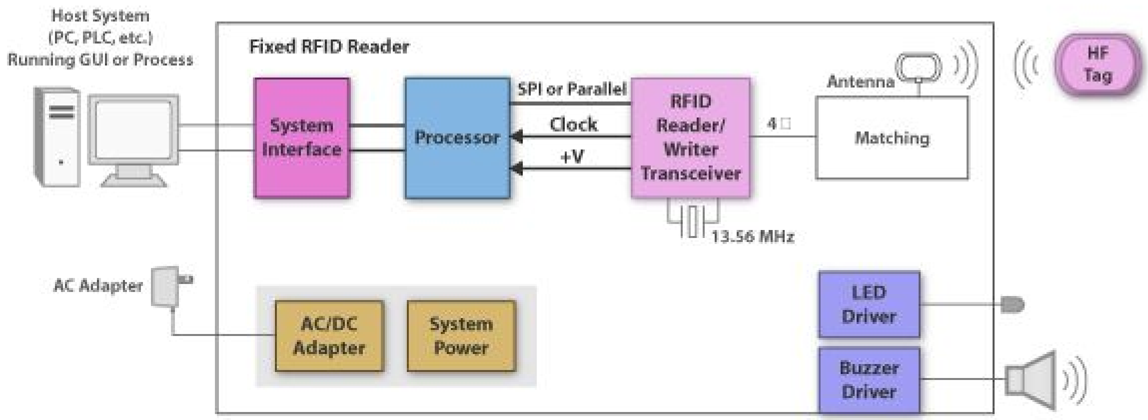

Figure 4.1: Fixed RFID Reader

The tag number data is then processed and the resulting output signal is sent to the stepper motor shield. The input data to the motor shield is processed and the resulting output is sent to the stepper motor on the food dispenser. This input data determines the number of steps (rotations) required to dispense the required amount of food for the specified cat based on the user inputs. Once this action is complete and the proper amount of food is in the food tray, the second stepper motors function comes into action. The motor shield then sends a signal to the stepper motor mounted underneath the food tray housing to rotate a predetermined number of steps to place the food tray from the location to where it receives the food to the open slot giving the cat access to the food.

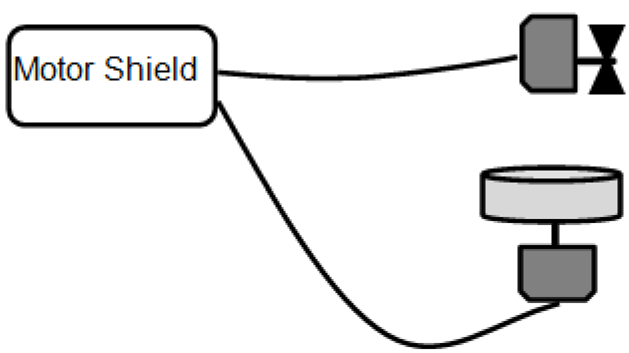

Figure 4.2: Motor Shield Hookup

When the cat steps onto the scale to be able to reach the food, the 4 digital scales begin to weigh the cat as a whole. Once the scales balance out, the information is sent to the microprocessor for binary conversion to be added together to equal the total cats weight. The output data from the microprocessor is then uploaded to the online file storage database. While the cat is eating, the proximity sensor is active allowing the cat unlimited access to the food as long as it is within the range of the sensor which is approximately 5 feet. When the proximity sensor is active, it holds the stepper motor from being active so it cannot rotate the food away from the cat. When the cat is finished eating and walks away from the SMAC, the proximity sensor is not activated anymore so the stepper motor becomes reactivated and rotates the food away from the open slot so other cats cannot eat any remaining food. It is a common feature that cats may not eat a whole meal at once as they may come and go several times throughout the day. The proximity sensors dual functionality allows the cat to come back and access that food whenever it desires.

The RDM630 RFID reader will be used for the SMAC because it allows for an external antenna. The reader transmits an electromagnetic field with the antenna and this EM field then induces a current in the antenna for the RFID tags to be picked up when within the read range. When the RFID is picked up it modulates a unique identification number which is transmitted back to the reader. In most cases the external antenna is made with thin copper wire arranged in a loop and the loop can generally be in any shape although a circular pattern is desired. To build a homemade antenna is rather simple and only takes three steps. For the SMAC external antenna, conductive copper tape will be cut into thin strips of approximately 2 mm thick. These strips will then be taped in a triangular loop shape pattern to fit around the food slot opening. Between 4 and 5 loops should be enough to increase the read distance of all around the SMAC which gives the cat more than enough opportunities to be scanned in to receive the food. The last step is to solder the connections together to ensure the antenna will work.

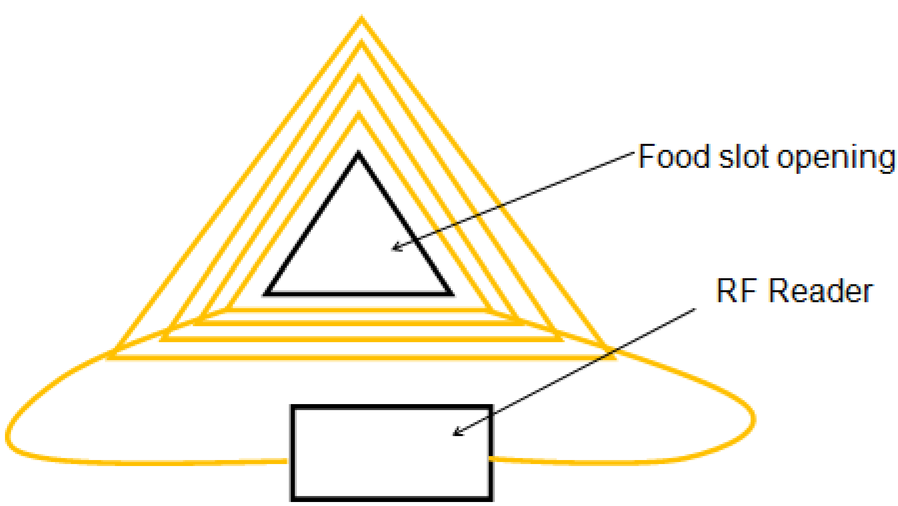

Figure 4.3: RFID Antenna

5 Prototype

5.1 Coding with Touch Screen

The user interface was made for the prototype to help show how the program will work, look, and feel. Figure 5.1 is what the user will see when they first start up the SMAC feeder. The user has two options, login or create a new account. The first time the user will create a new account by clicking on the ”New Account” button. They will be required to enter the main information, such as user name, password, and email. The user has the option to enter in their cell phone and cell carrier. This information will be used if any of the notifications is set to text messages.

Once the user has an account they can, on the login page, enter the user name and password. When they click the ”Login” button, the program will check the entered information against the tables in the database. If the user name is in the system, and the password is correct then the user will be taken to the ”Main” page of the system, Figure 5.2.

Figure 5.1: SMAC Login Page

The main page in the SMAC system will consist of four different options. The user can manage the cats, view the statistics, change the settings, or logout. Figure 5.3 shows what the user will see if the first option is chosen. The ”Manage Cats” page has only two options, to add or delete a cat. If the user wants to add a cat then they will be taken to the page depicted in Figure 5.4. Here they will enter the name and age of the cat, and choose the breed. The last option is where the user will decide whether to scan in a new RFID tag or to select from an existing one in the system. After the user adds the cat, they will be brought back to the manage cats page. Here they can use the buttons on the right side of the screen to navigate back to the other pages.

Figure 5.2: SMAC Main Page

Figure 5.3: SMAC Manage Cats Page

Figure 5.4: SMAC Add Cat Page

5.2 Plastic Housing with Trays and Scales

The first prototype of the SMAC shows the food container with the dispenser mounted inside it. This piece is connected to the rotating food tray external housing. The first prototype will be the food dispenser with the stepper motor. The paddle wheel attachment to the stepper motor will be mounted inside the plastic food housing. This prototype will be tested several times to check that it is dispensing the correct amount of food each time. The other stepper motor will also be tested at this time with the food trays. Dispensing the food and providing the food to the cat are two of the most important functions of the SMAC so these functions must be very precise and accurate. Once the stepper motors are running properly and smooth, the other plastic attachments will connect the top and bottom piece.

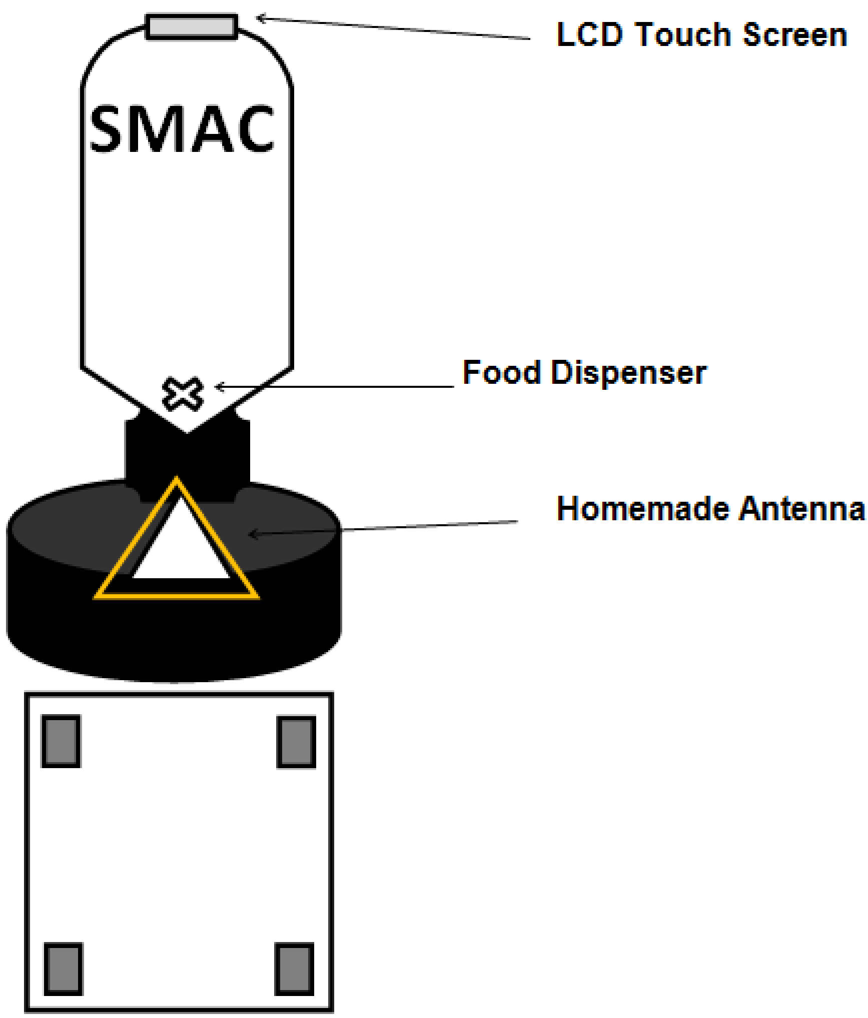

Figure 5.5: Prototype

This prototype will have a slide that guides the food into the tray from the dispensing location. The stepper motors will need to be properly controlled by the motor shield so they can be synced up. One stepper motor rotates the desired food tray based on the input and this will need to be synced with the other stepper motor so that it can then dispense the food into the correct tray. The first prototype will have the proximity sensor mounted to the front of the container for testing the range of distances that the sensor detects an object. The RFID tag will undergo severe testing since it is a very important feature of the SMAC because it risks the cat not getting its food it there is not a strong and wide read range. The toughest part for the first prototype will be successfully hooking everything up together as a unit. The prototype will help determine what the best way is to mount everything together so everything works properly and it looks presentable to the cat.

Food dispenser rotating to the proper angle with the stepper motor
4 digital scales wired to microprocessor and tested with software
Proximity Sensor accurately detecting whether cat is by SMAC
RFID Tag being recognized by homemade external RF antenna
Food trays being rotated at a good speed with the second stepper motor
Motor Shield precisely controlling both stepper motors

Table 29: Goals for First Prototype

6 Testing

6.1 Hardware Testing

In order for the SMAC to work efficiently, each individual piece must be tested on its own, then tested again while connected with other components to ensure that they work together. Proper testing precautions while still in the designing phase will prevent the likelihood of having to take components apart to try and figure out what isnât working once everything is connected.

Building the whole machine and trying to test it all at once is not practical. Taking the time to test each component, then each subsystem, and finally the SMAC as a whole will lead to a much smoother process.

6.2 Circuit Board Testing

Once the PCB is completely assembled it must be programmed. The PCB has to be programmed to operate every component of the SMAC. Then one by one each component must be connected to the PCB to make sure that all programs installed on it perform the desired task to the appropriate component.

The only problem with a custom made PCB is that it does not come preloaded with code, so it must be determined whether a problem that arises is occurring inside the PCB or somewhere within the attached component or its corresponding circuit. This is where the Arduino Mega 2560 board becomes useful. While it can not be used as the PCB for the SMAC, since it comes with code and so much more code can easily be found in databases, the Arduino board will be used for development and testing purposes. The Arduino Mega 2560 may have pre existing code to perform tasks with much more reliability than that of a custom PCB with a more custom code. The Arduino Mega will be used as the deciding factor for whether or not the problem is with the PCB and itâs programming or with the components

6.3 RFID Transceiver/Transponder Testing

Once the external antenna has been assembled and the antenna has been attached to the RFID reader testing can begin. Testing for the RFID reader is crucial to the success of the SMAC. If there are any blind spots to the reader then the whole design would be a failure as the cat would not be able to eat.

The reader should be tested to see if an object is between the RFID tag and the reader if it obstructs the signal. The reader should also be tested at different lengths, heights, and even angles to see if anything has a chance to interfere with the signal. Any interference means the cat goes hungry and no animals will be harmed or starved in the designing of this project.

6.4 Proximity Sensor

The proximity sensors only purpose is to ensure that when the cat walks away, the food bowl is rotated back to an empty spot holder. The proximity sensor will be hooked up to the PCB and a program to rotate the food tray to the empty space when the cat leaves the proximity sensors range will need to be installed.

Testing for the proximity sensor will need to include how to prevent objects from causing the proximity sensor to keep the food tray open after a cat walks away. A program may need to be written to also incorporate the chance of another cat walking up to the food bowl before the first cat is out of the proximity sensors range. The range of the proximity sensor is 5ft so it is a rather large range to have to keep clear.

6.5 Motor and Motor Shield Testing

The two stepper motors that will be powering the rotating food bowl and food dispensing mechanism will be powered by batteries and not actual loads on the PCB, but they will be driven by a motor shield that is operated on by the PCB. The first step in testing this system should be to program the PCB to manipulate the motor shield and then attach the motor shield to the PCB to test to make sure the program is working. Then power should be applied to the motors via the batteries.

After the batteries have powered the motors up and the PCB programming has been verified the motors are ready to be hooked up to the motor shield. The PCB can now control the motors through the motor shield and the motors can now be connected to the rotating food tray at the bottom of the mechanism and to the food dispensing mechanism. The whole motor system can now be tested to see if it has full functionality.

The food dispenser should be observed closely to be sure that food does not get hung up in the rotating, gumball-style, food dispenser. The motors also need to be tested in collaboration with the RFID tags to guarantee that when a cat walks up the motor will dispense itâs food and rotate the food tray too him. The motors also need to be tested with the proximity sensor to make sure that when the cat gets far enough away from the bowl the motor will rotate any unfinished food away to prevent other cats from stealing it.

6.6 Touch Screen Testing

The touch screenâs main purpose on the SMAC is to provide a user interface where the pet owner can enter can information to help the feeder identify which cat is what. The owner will enter the cats name verify what RFID tag corresponds to that cat. Once the touch screen has been mounted to the food dispenser and has been wired to the PCB a program to upload the user interface will need to be written to the PCB.

The first aspect that should be tested on the LCD Touch Screen is that the touch module is indeed enabled. The screen can be used as just a display with no touch feature, so if it is not wired or programmed correctly it may not be a touch screen as desired. The user interface will need to be tested by simply playing with the options. The SMAC has to be capable of all functions that could possibly be needed. Deleting a cat, changing a name, and anything else that a person might need to do will need to be tested to make sure the interface is ready for anything.

6.7 XBee Pro Testing

To allow the cat owner to keep a running log of their cats eating habits and any weight loss or gain an online database will be set up. The SMAC will be equipped with the XBee Pro, which enables communication between the PCB and a computer that will upload the information to the online database. The XBee pro will need to be tested first for its range. Users do not want to be limited to having to have the pet feeder 5 feet away from their computer in order to have the information relayed to the online database. The second test needs to determine if too many walls or obstructions hinder the connection to the computer, causing the signal to go in and out, leading to data loss. The online database will also need to be tested to make sure it is keeping an accurate log and is functioning appropriately.

6.8 Scale Testing

The four digital scales will have to be individually tested using several object of different weight. The average weight of a cat is between nine and eleven pounds and top sheet of glass weighs between three to five pounds so each scale will have to be able to accurately measure between three and four pounds. The scales circuit board will need to be accessed to gain entry to the output line that connects to that scales LCD module. This output wire will be tested with connecting to the microprocessor.

The next step will be to test that the microprocessor is properly receiving the 4 digital inputs from the scales. Once the inputs are correct, the next step of testing will be what the microprocessor outputs which will be tested with software. The output should equate to the total weight of the cat. Once the output is correct the next step of testing will be the connections from the microprocessor to the circuit board.

Further software testing will need to be done on communication from the circuit board to the online file storage database.Once all the sub tests are complete, everything will come together as a whole and be tested as a complete system. A properly functioning scale system will be able to take 4 digital input weights of the cat, encode the inputs for execution in the microprocessor, and then output the cats correct weight and upload it to the online file storage database

6.9 Testing the System as a Whole

The last and final test will be to test the SMAC in its entire completion. Once the SMAC is fully built the functionality of the system as a whole can be evaluated. The only way to really know if it works is to allow cats to use it as their means of eating. The biggest problem the SMAC could face is getting the cat to approach the bowl in order to make the food come out. This would also mean the SMAC should not be too loud or move too fast and scare the cat away. A trial and error period will be needed to guarantee that the cats can use the SMAC and will learn to approach a machine that at first seems empty in order to be fed. Once the SMAC achieves this it will be considered a real success.

6.10 Software

To make sure that the SMAC system is operating properly very thorough tests must be ran. Testing the software should be done in a way to minimize the time it takes to find faults in the system. There are three different types of software systems that need to be tested. The mains system and mobile application are both written in an object-oriented programming language. This will allow the use of the same testing method to test both systems. There are a few different techniques that will be used in testing the database and server. This is because the database can be accessed through the server as well as without it.

6.10.1 Main System and Mobile Application

Since the programming language is object-oriented, each individual class must be tested. A good researched method of testing comes from a publication [10]. The method described is called Adaptive Random Testing for object-oriented programming (ARTOO). This basically will test the program with a random set of data, but chooses the order of the ”random” tests based on ”object distance”. The distance between objects is based on three main components: their value, dynamic types, and the primitive value of the attributes.

The final equation that is used is a combination of: a distance between the direct value of the object, a distance between the object types, and a distance between the objects individual fields. This value is then normalized with a normalized function that satisfies the constraints of increasing monotonically and that the function is zero at zero. The final ”object distance” is one-third of the normalized value found previously.

To test using this method the SMAC development team will make a set of random test cases. This will be done by creating instance of the class wanting to be tested. Which might require the creation of instances of any other classes used inside the class, as well as generating the random values for the types in the class.

The first case tested will always be randomly chosen, since there is nothing in the already tested set of data. The algorithm will then calculate the ”object distance” of this case to the ones left in the list of untested cases. The next test case chosen will be the one with the highest value, or farthest away. If a fault is not found ARTOO will then pick the next test case based on the distance from the tested set. The case with the greatest distance will be tested next. This process is repeated until a fault is found. This method will be ran on all the classes shown in Table 30.

6.10.2 Database and Server

The server is most important to test first since it will allow the different parts of the system to communicate not only with each other but with the database as well. The first thing to check is to make sure each system can connect to the server. Once each device can connect, their ability to send and retrieve data will be tested. The systems will also be tested on their ability to disconnect from the server.

Feed Class
FoodMotor Class
ScaleMotor Class
Animal Class
Cat Class
Food Class
Weight Class
Scale_1 Class
Scale_2 Class

Table 30: SMAC Major Classes

Once the server is thoroughly tested the database can be tested with the knowledge that no faults will be caused by the server. The database testing will include various test cases to test the following criteria. The data should be stored in the correct keys in the correct tables. Each user should have only one table, with only one user name and password. The data in the tables should fit perfectly, and not overflow the keys. The data should be able to be sent and retrieved with no errors.

7 Administrative Content

7.1 Budget and Financing

The items required for the SMAC will be split up evenly amongst the three group members: Kristin, Tim, and Austin. The hardware parts will be purchased first so assembly and testing can begin. There is a potential sponsor form the Institute for Simulation and Training. Sponsor participation will either be providing any parts for the project or covering a portion of the finances.

7.2 Milestones

Item	Price$
125 KHz RFID Tag	2.50
RDM630 RFID reader	15.00
Sharp Proximity Sensor	15.00
Adafruit Bipolar Stepper motor	14.00
Adafruit Small Stepper motor	6.00
Adafruit Motor Shield	19.50
Cat Food	20.00
Paddlewheel Food Dispenser	5.00
2.4â LCD Screen	20.00
Texas Instruments Microprocessor	20.00
External Plastic Housing	20.00
Internal Plastic Housing	20.00
Food Trays	20.00
Food tray/Food Dispenser Connector	10.00
Arduino Mega 2560 for development	60.00
Food Container	15.00
Digi Xbee Pro	37.95
22 AWG wire	5.00
Digital Scales	15 X 4 = 60
Copper Tape for external antenna	23.65
12v Power Adapter	10.00
Xbee Pro Shield	9.00
TOTAL	427.60

Table 31: Budget

Date	Goal
August 20th	Purchase All Components
August 24th	Begin Assembly of PCB
August 30th	Complete PCB Assembly
September 3rd	Begin PCB initial programming
September 7th	PCB Testing
September 12th	Fully Functional PCB
September 14th	Begin Plastic Housing Construction
September 19th	Plastic Housing Complete
September 21st	Begin Assembling Food Tray
September 25th	Food Tray Complete
September 27th	Mount Stepper Motor to Plastic Housing
September 29th	Construct Food Dispensing Mechanism
October 3rd	Attach Stepper Motor to Food Dispenser
October 4th	Provide Battery Power to Stepper Motors
October 5th	Connect Stepper Motors to Motor Shield
October 8th	Connect Motor Shield to PCB
October 11th	Program Motors on PCB and Test Motors
October 17th	Attach 4 Digital Scales to Feeder
October 19th	Program Scales
October 22nd	Test Scales
October 24th	Construct External Antenna to for RFID Reader
October 27th	Mount Antenna and RFID Reader
October 30th	Test RFID Tags
November 2nd	Program RFID Read to Rotate Motor
November 5th	Test RFID w/ Motors
November 8th	Mount Proximity Sensor Program Proximity Sensor
November 11th	Test Proximity Sensor w/ Motors
November 13th	Mount Touch Screen to Top of Feeder
November 14th	Program Touch Screen
November 16th	Test Touch Screen and User Interface
November 19th	Online Database Set Up
November 19th	Begin Testing Relay of Information to Online Database
November 21st	Online Database Fully Functional Data Transfer Successful
November 22nd-25th	Begin Testing with Cats
November 26th â 29th	Allow For Any Adjustments From Cat Trial
November 30th	Fully Operational SMAC
December 3rd	Submit Project

Table 32: Milestones

8 Appendices

8.1 Copyright Permissions

Licensing [edit]

Figure 8.1: GNU Liecense for Figure 2.24

> I'm an Electrical Engineering student at the University of Central Florida working on my senior design project and I would like to request permission to use some pictures from your website in my project. The items I would like to use photos for include the 2.4" TFT LCD Screen Module: ITDB02-2.4D and the Arduino 2.8 TFT Touch Shield. Thank you for your time.

Dear Austin Scruggs,

Thank you very much for your selection in our store! You are allowed to use some pictures from our website in your project.

I hope you will get a successful project and enjoy your design!

Best regards from Itead Studio!

Wendy Yuan

Itead Studio
www.iteadstudio.com
Phone: +86 (0)755 29610458

Figure 8.2: Permission from Itead Studio

Jodry, Whitney
To tforkenbrock@knights.ucf.edu
4/20/12
Reply

Yes. Please just credit TI with the picture.

Thank you,

Whitney

From: tforkenbrock@knights.ucf.edu [mailto:tforkenbrock@knights.ucf.edu]
Sent: Friday, April 20, 2012 4:38 PM
To: Jodry, Whitney
Subject: Picture Permission Request

Good Afternoon,
My name is Tim Forkenbrock and I am an electrical engineering student from the University of Central Florida. I am using the Texas Instruments AM3715 (ACTIVE) Sitara ARM Microprocessor (MPU) in my senior design project and would like to request permission to use the picture provided on the TI website in my paper. Thank you for your time

Tim Forkenbrock

Figure 8.3: Permission from TI

Adafruit Industries Add to contacts
To Tim Forkenbrock

We've added this sender to your safe list. That way you can always see what they've sent you.

```
totally ok!

On Fri, Apr 20, 2012 at 4:48 PM, Tim Forkenbrock
<tforkenbrock@knights.ucf.edu> wrote:
> From:   Tim Forkenbrock
> Email:  tforkenbrock@knights.ucf.edu
>
> [[Press / media inquires]]
> -----------------------------------------------------
>
> Hello,
>   I am an electrical engineering student from the University of Central
> Florida and I am using the bipolar stepper motor, small stepper motor, and
> motor shield (for Arduino) for my senior design project. I would like to
> request permission to use the pictures for these items provided on your
> website in my project. Thank you for your time.
>
```

Figure 8.4: Permission from Adafruit Industries

General info   Full viev

Wendy Add to contacts   4/11/12
To tforkenbrock   Reply

Attachments, pictures and links in this message have been blocked for your safety.
Show content | Always show content from wendy@iteadstudio.com

> Good Afternoon, I am an Electrical Engineering student at the University of Central Florida and i am using the 125Khz RFID module RDM6300 - UART for a senior design project. I would like to request permission to use the pictures on the website. Thank you for your time.

Dear Tim Forkenbrock,

Thank you very much for your selection in our store! You are allowed to use the pictures on the website, but please state the origin of those pictures when you use them.

I hope you will get a successful project and enjoy your design!

Best regards from Itead Studio!

Wendy Yuan

Figure 8.5: Permission from Itead Studio

答复: 13.56Mhz RFID module - IOS/IEC 14443 type a

Leslie Liao Add to contacts
To 'Tim Forkenbrock', jianye.huang@seeedstudio.com

Hi Tim,

Feel free to any pictures in our website.

Best Regards,

-Liao
Seeed Studio

-----邮件原件-----
发件人: Tim Forkenbrock [mailto:info@seeedstudio.com]
发送时间: 2012年4月12日 4:19
收件人: info@seeedstudio.com
主题: 13.56Mhz RFID module - IOS/IEC 14443 type a

From: Tim Forkenbrock
Email: tforkenbrock@knights.ucf.edu

Figure 8.6: Permission from Seed Studio

Re: Sharp GP2Y0A02YK0F analog distance Sensor

Aaron Fuller Add to contacts
To tforkenbrock@knights.ucf.edu

Hello, Tim.

Thank you for asking us for permission to use our picture. You can use the images of the sensor from our website as long as you credit Pololu for them in your paper.

Good luck with your paper, and if you have any additional questions, please contact me.

Sincerely,
Aaron Fuller
(702) 262-6648
www.pololu.com
....
Pololu Corporation
920 Pilot Rd.
Las Vegas, NV 89119
USA

Figure 8.7: Permission from Pololu Corporation

8.2 Bibliography

References

[1] April 2012.
http://www.wikipedia.com.

[2] March 2012.
http://www.Adafruit.com.

[3] March 2012.
http://www.iteadstudio.com.

[4] April 2012.
http://www.seeedstudio.com.

[5] April 2012.
http://www.omega.com/prodinfo/stepper_motors.html.com.

[6] April 2012.
http://www.pololu.com.

[7] April 2012.
http://www.ti.com.

[8] Amazon. Automatic cat feeder combo, March 2011.
http://www.amazon.com/Automatic-Feeder-Combo-CSF-3XL-Analog/dp/B000MTK086.

[9] Amazon. Petmate lebistro portion control automatic pet feeder,, March 2011.
http://www.amazon.com/gp/product/B0016BVY2U/ref=pd_lpo_k2_dp_sr_2?pf_rd_p=486539851&pf_rd_s=lpo-top-stripe-1&pf_rd_t=201&pf_rd_i=B000MTK086&pf_rd_m=ATVPDKIKX0DER&pf_rd_r=00FK0D233J9VAMHXHRG7.

[10] Ilinca Ciupa, Andreas Leitner, Manuel Oriol, and Bertrand Meyer. Artoo: adaptive random testing for object-oriented software. In *Proceedings of the 30th international conference on Software engineering*, ICSE '08, pages 71–80, New York, NY, USA, 2008. ACM.

[11] National Instruments Corporation. What is microsoft windows embedded standard 7?, March 2011.
http://zone.ni.com/devzone/cda/tut/p/id/13048.

[12] Avnet Inc. Windows embedded standard 7 (wes 7), March 2011.
http://www.avnetmssolutions.com/products/operating-systems/win-embedded/westd/wes7#features.

[13] RobotShop Distribution inc. Automatic pet feeder (small), March 2011.
http://www.robotshop.com/autopetfeeder-small-4.html.

[14] Landmanr. Rfid pet feeder, March 2011.
http://www.instructables.com/id/RFID-pet-feeder/.

[15] Moore Pet Supplies LLC. Autopetbowl - small, March 2011.
http://www.moorepet.com/Autopetbowl-Small-p/2000pbs.htm.

[16] Microsoft. Windows embedded standard 7 (formerly xpe), March 2011.
http://www.microsoft.com/windowsembedded/en-us/evaluate/windows-embedded-standard-7.aspx.

[17] Mathew J. Newton. Internet-enabled cat feeder - mark 2, March 2011.
http://www.newtonnet.co.uk/catfeeder/.

[18] SMARTHOME. Remote pet feeding kit, large feeder, March 2011.
http://www.smarthome.com/6184AK/Remote-Pet-Feeding-Kit-Large-Feeder/p.aspx.

[19] Walmart. 6-day automatic pet feeder, March 2011.
http://www.walmart.com/ip/6-Day-Automatic-Pet-Feeder-Cats-Dogs/5969455.

[20] Wikipedia. Android (operating system), March 2011.
http://en.wikipedia.org/wiki/Android_%28operating_system%29.

[21] Wikipedia. C++, March 2011.
http://en.wikipedia.org/wiki/C%2B%2B.

[22] Wikipedia. C (programming languages), March 2011.
http://en.wikipedia.org/wiki/C_%28programming_language%29.

[23] Wikipedia. C sharp (programming language), March 2011.
http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29.

[24] Wikipedia. Java (programming language), March 2011.
http://en.wikipedia.org/wiki/Java_%28programming_language%29.

[25] Wikipedia. Meego, March 2011.
http://en.wikipedia.org/wiki/MeeGo.

[26] Wikipedia. .net framework, March 2011.
http://en.wikipedia.org/wiki/.NET_Framework.

[27] Wikipedia. Ubuntu (operating system), March 2011.
http://en.wikipedia.org/wiki/Ubuntu_%28operating_system%29.

[28] Wikipedia. Windows presentation foundation, March 2011.
http://en.wikipedia.org/wiki/Windows_Presentation_Foundation.