# OmNomCat - An automated pet feeder

**Francesca Madeddu**
Computer Science MSc
Swansea University
SA2 8PP, UK
803623@swansea.ac.uk

**Hua Cai**
Computer Science MSc
Swansea University
SA2 8PP, UK
@swansea.ac.uk

**Michael Waterworth**
Computer Science MSc
Swansea University
SA2 8PP, UK
805956@swansea.ac.uk

## Abstract
Our task is to develop a novel cat-feeder which could be operated automatically by system. We have used arduino system as our main development technology which include: arduino board, LCD, Servos, and RFID sensors and several wires and controllers such as potentiometers and push buttons. There are two main functions we have implemented. The first is to display counter time about food dispense on LCD and allow users feed cat as they wish instead of fixing. The second is to keep track of the cat by reading their tags.

## Introduction
There are many people living in this world who are fans of cat. Due to this creature is so cute that we do not want to make them feel starving or thirsting because of our absence. This is the reason why we attempt to design this system for those cat owners. Our task is to feed cat automatically and remotely with recording the track of cat in order to make owners feel comfort.

## Related works
Erik Rydell and Yangchen Zhang provide a framework about how to use Arduino board to build distant cat feeder [2]. Their work have used two servos and sensors which is controlled by mobile within wireless environment. Their work give us a overview about how a system should work, es-

pecially their circuit. Rachel Heil [1] et al in 2008 provide their design about how build an automatic pet feeder system. Their work provide a detailed hardware design and control circuit design which provide us with many inspiration about how to construct side panels. But their system is a little complicated than the first work. The official Arduino book [3] - Arduino projects book - written by Arduino company give us the basic idea about how to use Arduino board and how to build time-counter system and servos by Arduino starter kit.

**Figure 1:** The final prototype.

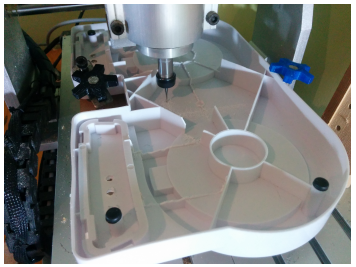

**Figure 2:** Gaining the space for the LCD with a CNC milling machine.

## The final application

*Requirements and Specification*

Our team propose to create an automated pet feeder machine. This machine, placed in a user's house, will allow pets to access food in an automated way, removing the need for the owner to be around during feeding time. In this way it has the potential to improve the well being of the animals. The specific implementation we plan to build will center around the feeding of a pair of cats.

These cats are left during the day, a frequent problem for pet owners, and may be hungry during the day - relying on external sources of food. The feeding machine will automatically dispense food at pre-set times to ensure the pets get an adequate amount, without over-feeding.

*Development Technologies*

A number of different software technologies were employed in developing the prototype. These included:

- Arduino 1.6.3 IDE [1] was used to write the code, compile it and upload it to the board. It was also used to read console data back from the board during development for debugging purpose.
- Two external library were imported in to the project for the Timer[2] and RFID[3] management.
- Fritzing [4] was used to document the Arduino-based prototype by exporting models as images.
- JSCut[5], a full featured web-based CAM package, was used to create cut files from scalable vector graphics which were then used to mill the plastic housing.
- GIMP[6] was used to draw the SVG path used by JSCut.
- Git / GitHub[7] was used on development machines to synchronise code and to version control it effectively. GitHub was used to synchronise these repositories using pull requests.
- Dropbox[8] was used to synchronise files that weren't code. It was also used to share binary files like PDF and word documents to allow simple project management.

The delivered system is comprised of the following hardware components:

- 1x Zevro double dry food dispenser[9] was used as base for the system;
- 2x Arduino Uno Rev3[10] (figure 3a) board, a microcontroller board based on the ATmega328, were used separately to construct two independent systems. One board was used to set and display the time between

[1] arduino.cc/en/Main/Software.

[2] github.com/JChristensen/Timer.
[3] github.com/miguelbalboa/rfid
[4] fritzing.org.
[5] jscut.org.
[6] gimp.org.
[7] github.com.
[8] dropbox.com.
[9] zevro.com/classic-dry-food-dispenser-double-canister.
[10] arduino.cc/en/Main/ArduinoBoardUno.

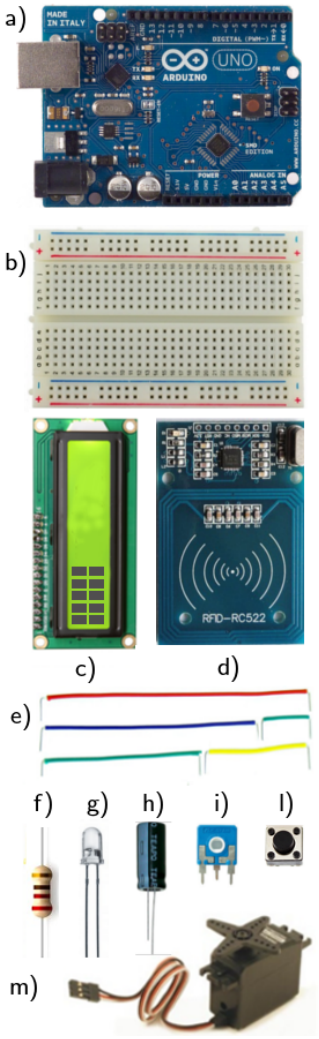


**Figure 3:** The main components used for the implementation of the system.

each food dispense and to control the servo motor which controlled the food dispense, while a second board was used with the RFID module to display, through 12 LEDs, the amount of time passed since each cat used the feeder;

- 1x Alphanumeric LCD Module 1602A-1[11] (figure 3c) was used to display the time remaining until the next food dispense and to provide feedback while changing the time value;
- 1x 10$k\Omega$ Potentiometer[12] (figure 3i) was used to change the time between food servings;
- 1x Pushbutton[13] (figure 3l) was used to dispense extra food outside the defined schedule;
- 1x Small servo motor (figure 3m) was used to turn the mechanism which allowed the food to be dispensed;
- 1x Mifare MFRC522 RFID module[14] (figure 3d) was used to read the 2 NFC tags associated with two different cats, in order to show the last time they were seen;
- 12x LED[15] (figure 3g), divided in 2 groups of 6, were used to display the time passed since the last time each cat was seen;
- 2x 9v Battery-snap[16] and 1x 5v power supply were used to give power to the system.

Finally, other minor components were used to make the system work, including:

- 1x Capacitor 100uF[17] (figure 3h), 8x 220$\Omega$ Resistor[18] (figure 3f), 2x Breadboard (figure 3b), n jump-wires (figure 3e), Sugru[19].

All the cited components were assembled together to construct two independent systems as shown in figure 2.

The first system displays on the LCD the time between servings, which can be changed through the potentiometer manipulation; moreover through the button it is possible to dispense food outside scheduled time intervals.

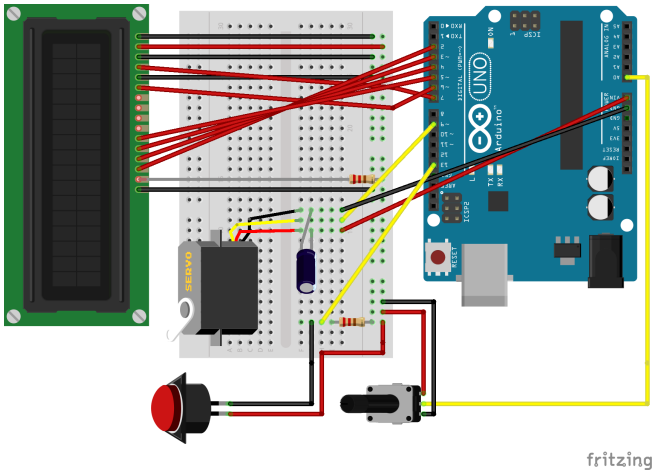


**Figure 4**

The second system keeps track of the last time cats were seen by reading their NFC tags. 12 LEDs are used to inform the user about the passed time.

[11] openhacks.com/uploadsproductos/eone-1602a1.pdf.
[12] arduino.cc/documents/datasheets/ACP_potentiometers.pdf.
[13] arduino.cc/documents/datasheets/Button.pdf.
[14] nxp.com/documents/data_sheet/MFRC522.pdf.
[15] arduino.cc/documents/datasheets/LEDWC503B-WAN-CBADA151.pdf.
[16] arduino.cc/documents/datasheets/9VCABLE150.pdf.
[17] arduino.cc/documents/datasheets/cap100uF.pdf.
[18] arduino.cc/documents/datasheets/Resistors.pdf.
[19] sugru.com.

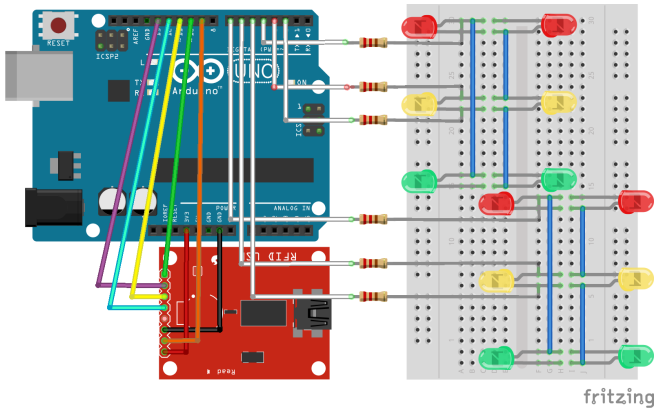


**Figure 5:** Because of the high amount of LEDs required and the limited number of pins, the Charlieplexing technique was used to control 12 LEDs with only 6 pins.

The separation in two independent subsystems aided rapid prototyping by having two separate project boards to work on. It was also done for space reasons and because a large number of pins were in use on the first board.

*Charlieplexing*
Charlieplexing was used in the designed system to control the LEDs. This was done as we had to drive a large number of LEDs (12) with only a limited number of pins available on the Arduino board. Charlieplexing uses the ability of microcontrollers to change their pins between three states: Input, Low and High. This works well with LEDs, which are diodes - meaning they only allow current to flow in one direction. By controlling the state of each pin connected to the board it's possible to turn on individual LEDs at a time. A further complication was added in the fact that multiple LEDs need to be lit at the same time. This was accomplished in by separating the system in to two charlieplexed arrays of LEDs but could also have been achieved in software by rapidly switching between multiple states using binary code modulation. In this way the 12 pins were controlled with a small amount of pins

*LCD*
The LCD is used to display the time remaining before the next dispense and to give feedback to a user whilst they are adjusting this interval with the control knob on the back of the device. This has been mounted vertically in the Y axis to fit in best with the shape of the dispenser. A bar graph was used to show these and a custom character was designed that adds a visual style to the system.

*Motor*
A number of motors were investigated for the development of the prototype device. A servo motor was chosen over a stepper motor or DC motor because of its simple controls and small package size. The stepper motor used runs directly from the 5v line provided to the Arduino board and allows precise control of the dispensing wheel - allowing exact control of portion size.

## Delivered Prototype

All of the code and circuit designs for the prototype are available on GitHub: `github.com/moodymood/CatFeeder`.

*Running the prototype*
The delivered prototype is easy to set up and use. Once plugged in to a power supply the LCD should light up and the dispenser should default to its starting position.

*Appearance and Implemented functionalities*
The developed system can be divided in the following independent subsystems:

*The cat feeder*
The cat feeder can further be broken down in to multiple tasks: set interval duration, show changes on display, show time on display, dispense food automatic, and dispense food with button.

*The cat checker*
The cat checker allows to monitor the presence of cats near the feeder machine: with two groups of six LEDs, where each group is associated with a specific NFC tag, the user is informed about the time passed since the last time a cat was seen. Whenever a tag is recognised by the system, the timer is reset and the led series updated.

*Limitations of the delivered prototype*
The delivered prototype is affected by some limitations that should be solved for any future development. Firstly, the the small servo motor, even when supplied with an external power supply, wasn't strong enough to turn the food dispenser mechanism, therefore it is recommended to substitute the servo motor with a more powerful stepper motor. This choice would increase the complexity of the circuit, but it is essential to the correct operation of a final system. Secondly, the maximum distance allowed for the RFID module to be able to read the NFC tag is around 4cm, which might be too short to catch the cat collar. Solving this problem is more critical: even if a more powerful RFID readers is available on the market, they are very expensive so that the coast would not be justified by the purpose of the project. An alternative solution would be using a proximity sensor, which however would not allow to keep track of each cat individually.
Finally, the last problem, regards the timers in the device. The base Arduino platform doesn't have a real-time clock. This affects the implementation as it means the intervals are based on CPU 'ticks' instead of a hard and fast milliseconds. This extends over in to the use of hardware interrupts and pauses, affecting the program execution and delaying the 'ticks'.

## Roles

As can be expected the work carried out was split in to a number of tasks that could be accomplished separately. This worked out well because of the limited number of practicals available. From here development of the application could continue at a rapid pace, with the report documentation tapering in near the end. A detailed breakdown of tasks and roles in presented in the following table:

| **Task** | **H** | **F** | **M** |
|---|---|---|---|
| Develop concept in to project plan | | √ | |
| Order components | | √ | |
| Modify food dispenser | | √ | √ |
| Prototyping motor attachment | | √ | √ |
| Laser cut food dispenser | | √ | √ |
| Code RFID Reader | | √ | |
| Code Charlieplexing Timer | | √ | |
| Code Motor Arduino | √ | | √ |
| Code Motor Timer | | | √ |
| Code LCD Timer | | | √ |
| Code Potentiometer Timer | | | √ |
| Build Charlieplexing LED board | | √ | √ |
| Assemble final prototype | | √ | √ |
| Write up sections of the report | √ | √ | √ |

**Table 1:** Task achieved by person where H = Hua Cai, F = Francesca Madeddu and M = Michael Waterworth

## Conclusion

In this paper we have described a prototype product that can automate the feeding of pets. This is just a prototype and leaves a lot of scope for future improvements. These include the option of adding a more powerful motor; A stepper motor, for example, would be great for this and could significantly increase the torque provided to the dispenser. Another possible extension of the prototype could be to use NFC in conjunction with the dispensing function. This could be designed to only dispense food when a cat that hasn't recently been fed tries to use the feeder.
Additionally a web interface could be added to the project allowing remote monitoring of the feeder and even the possibility of remote feeding - allowing greater interaction with the animals. Web interactivity could perhaps be better accomplished with a different platform like Raspberry Pi which has access to higher level programming languages and a full networking stack.

## References


[1] Heil, R. The Smart Pet Feeder. Tech. rep., 2008.

[2] Rydell, E., and Zhang, Y. Assisting minor everyday pet chores.

[3] Scott, F., and Michael, S. *Arduino projects book*, arduino cc ed. 212.