

# Pavlov's Cat

## Introduction

My project is a food device which dispenses a specified quantity of candies every time a certain melody is playing and when the sensor detects the presence of a hand.

The idea of the project started when my friend received a cat as a present and she was struggling to teach her cat when it is time to eat and when it is not. Of course, I adapted it to dispense candies, not cat food, given the fact that I don't have one.

The purpose of the project is to teach your cat when it is time for her to eat and when it is not. This will be a relieve for the owners, not having to worry that if they leave the cat at home, she will eat all the food at once.

I believe that this project is useful for others because you can control the amount of food and also the time when the device will dispense the food. The fact that you can control it from your phone is a big plus especially that you can't guess when the cat is in front of the device.

## General description

-----

Pavlov's Cat

-----

- Pavlov's Cat: the whole device that dispenses a specified quantity of candies every time a certain melody is playing and when the sensor detects the presence of a hand.
- Arduino MKR1000: the wi-fi board
- Buzzer: it is a signaling device that will play a melody
- Servo motor: this small motor will move the "door" allowing the candies to be dispensed
- Phototransistor: this will detect the presence of the hand

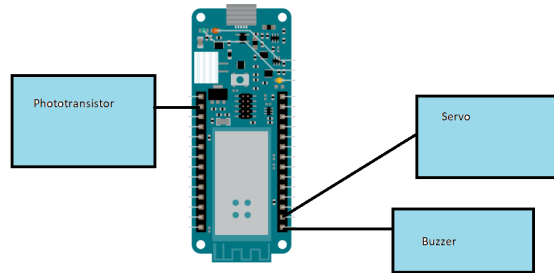
## Hardware Design



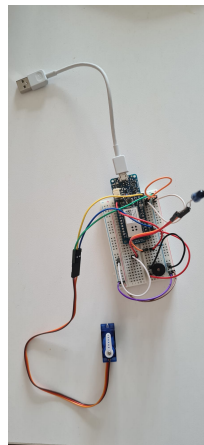
- Servo motor: A servo motor is a BLDC motor with a sensor for positional feedback. This allows the output shaft to be moved to a particular angle, position, and velocity that a regular motor cannot do.

- Buzzer: A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, train and confirmation of user input such as a mouse click or keystroke.
- Phototransistor: A semiconductor device like a phototransistor is used to detect the light levels and changes the flow of current among emitter & collector terminals based on the light level it gets.
- Arduino MKR1000: The Arduino MKR 1000 WiFi is the easiest point of entry to basic IoT and pico-network application design.

This is the block schema of the project



This is the scheme with all the components connected.



## Software Design

### Code description

First of all, we include all the libraries needed for Arduino, but also for the wi-fi and telegram part. We set the portion to the default value of 1, we declare the servo object, we declare the buzzerPin (i used pin 6 on the Arduino board), we set the initial value of startDetecting as being false, the timer will be 0, also the oldchatid will be null. In the method setup() we are starting the connection at wi-fi, we declare the token of the telegram bot, we declare the emoji used at the telegram keyboard, initialize the rows of the keyboard, and also we make the buzzerPin an output. In the loop we check if there are any updates in the chat and if there is an update, we will print the message (the codes used for the keyboard emoji). If the text is indicating melody and food, then we play a certain melody, we display a message, the detection begins, we display a message that the servo is moving and we also calculate

the time that is required until the hand reaches the range of the phototransistor. If the message is melody and no food, another melody will be played, we send a message that indicates that nothing will be done. Here are 2 cases. First, the hand is detected and it will display that the hand is detected, but nothing will happen and the second case, no hand is detected in the past minute and there will be a message saying that no hand was detected in the past minute. We also can set the amount of candies that will be dispensed. To be able to detect the hand, we need to know the threshold of the phototransistor and we adapt it in order to be able to differentiate when a hand is in range and when it is just ambient light. We also have a function for moving the servo. In the for loop we open the "door" by moving the servo from 0 to 90 degrees, we delay the closing by multiplying the portionAmount with 300 and after that we close the "door" by moving the servo from 90 to 0 degrees. The last function in the loop is playMelody which will play the melody chosen before.

```
Library used:  
* ArduinoJson.h  
* WiFi101.h  
* SPI.h  
* TelegramBot.h  
* Servo.h  
* pitches.h
```

## Final results

After finishing the project these are the final results <https://youtu.be/Hcyqg9mMQoI>

## Conclusions

The project was a bit harder than I thought when i chose it. It required more knowledge on the software part, but in the end i managed to get it done. I used some of the knowledge from the labs, but i still had some problems with the bot part, because it was not recognized. I forced myself to make it on my own, without any help so i can learn more things about microprocessors and the connection that we can make between them and an app. The project is working as expected, i managed to solve the problem with the servo motor which was moving right after i was uploading the code on the board. From my point of view, this is a success project given the fact that it is working as i was expecting.

## Download

Here is the archive of the project. It contains:

- block schema

- the code file
- the pitches file
- the video

[dumitru\\_andreea-raluca\\_fils\\_1221a\\_materials.zip](#)

## Journal

- 22.04.2023 - Choosing the project theme
- 07.05.2023 - Published the description of the project
- 20.05.2023 - Added the hardware design
- 28.05.2023 - Added the software design part + the results part + the bibliography part

## Resources

1. <https://docs.arduino.cc/hardware/mkr-1000-wifi>
2. [https://projecthub.arduino.cc/Arduino\\_Genuino/pavlovs-cat-296cc2](https://projecthub.arduino.cc/Arduino_Genuino/pavlovs-cat-296cc2)

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/dumitru>



Last update: **2023/05/28 21:35**