



# CATHUB – INTERACTIVE CAT FEEDING DEVICE

Petar Petrov – ISE III



# Motivation

- Create something helpful for our household
- Feed the cat while away from home
- Gather cat's behavior data
- Track cat's behavior and check if it remembers and learns



# Idea and Main Goals

- Create a device, which shall be connected to the Internet and able to dispense food
- Use a Telegram bot to communicate with the device
- Have two different melodies – One to dispense food and another without food
- Track cat's movement and store the data
- Send feedback to the user regarding the cat's 'response time'
- Check if the cat learns to respond only to the melody designated for food – using a Motion Sensor
- Turn on a LED diode, if it is dark in the room



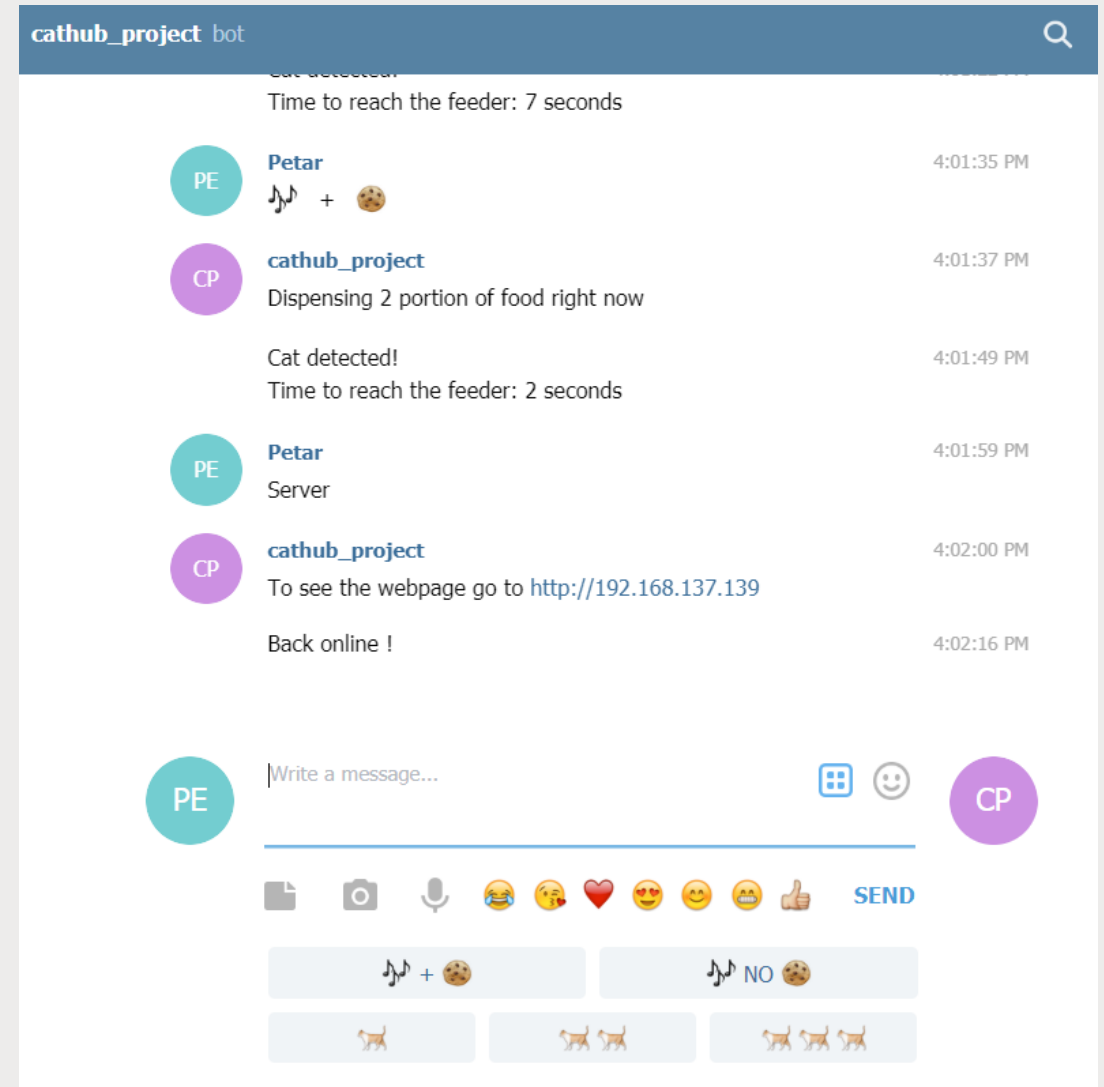
# Component List

- Arduino MKR1000 – Selected because of the built-in access to Wi-Fi
- Servo motor – Open the hole to dispense food
- PIR Motion Sensor – Checks for the presence of a cat
- Buzzer – Plays the Melodies
- Adafruit TSL2561 Luminosity Sensor – Checks for Light Condition in the room
- LED Diode – Switches On if it is dark in the room
- 3.7v LiPo Battery 1000mAh– Main power source
- 4x1.5v AA Batteries – To supply 5v to the Servo and Motion sensors
- 1.5mm thick Cardboard – To create the box
- Breadboard – Small
- Wires



# Telegram Bot Interface

- User-friendly Interface
- Buttons instead of Text commands
- 3 different Portion Sizes
- Two-Way communication – including messages from the device



# Server Mode

- Type “Server” in Telegram to start a local Web Server, which runs on the MKR1000
- Receive a response from the Arduino in the form: To see the webpage go to [http://local\\_ip\\_address](http://local_ip_address)
- While Server Runs – No Access to the Dispense options
- Press Disconnect and food dispense service continues
- Server Webpage:

Food dispenser report		
Food/no Food	Cat detected	Time to reach the food
Food	Yes	12 s
Food	No	/ s
No Food	Yes	0 s

Disconnect

# Device Description and Way of Operation

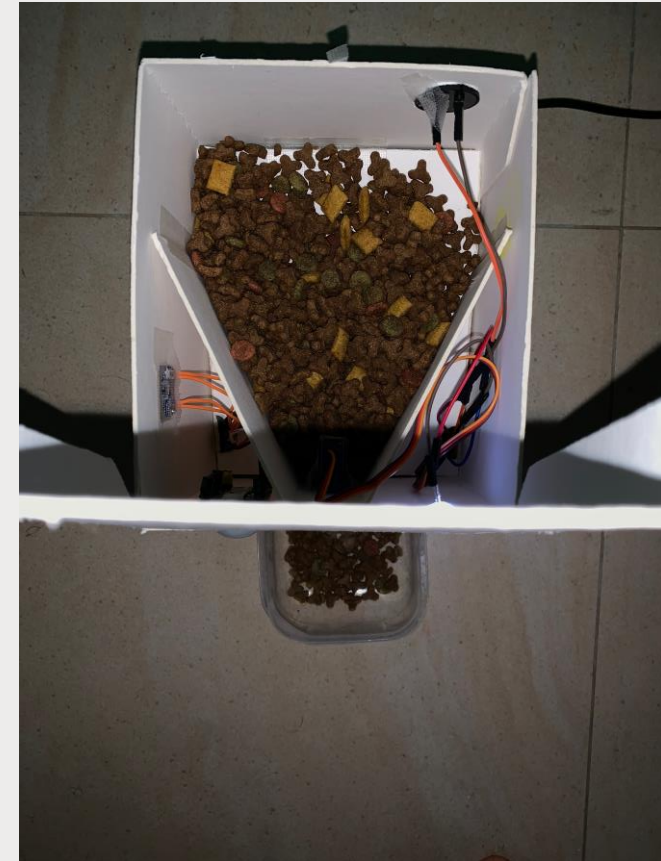
- There are 2 layers – On the bottom is the Arduino and the Batteries, on the upper one – the food and the different sensors
- The food is separated in a different compartment
- The Buzzer is at the back of the Device
- On the front, the device looks like a human – with eyes (Motion sensor and a LED diode) and a nose (The Moving part, which is used to open and close the hole to the food).
- On the right side is the Luminosity Sensor – After playing of either of the melodies, it checks if it is dark in the room (depending on a pre-defined threshold) and lights the LED diode. It then stays ON for 2 minutes or until it gets lighter in the room.
- The Servo is at the back of the mentioned moving part and is moving it. It opens at 90 degrees and stays open longer, depending on the portion size.

# Device Photos

Outside – Sensors and Movement

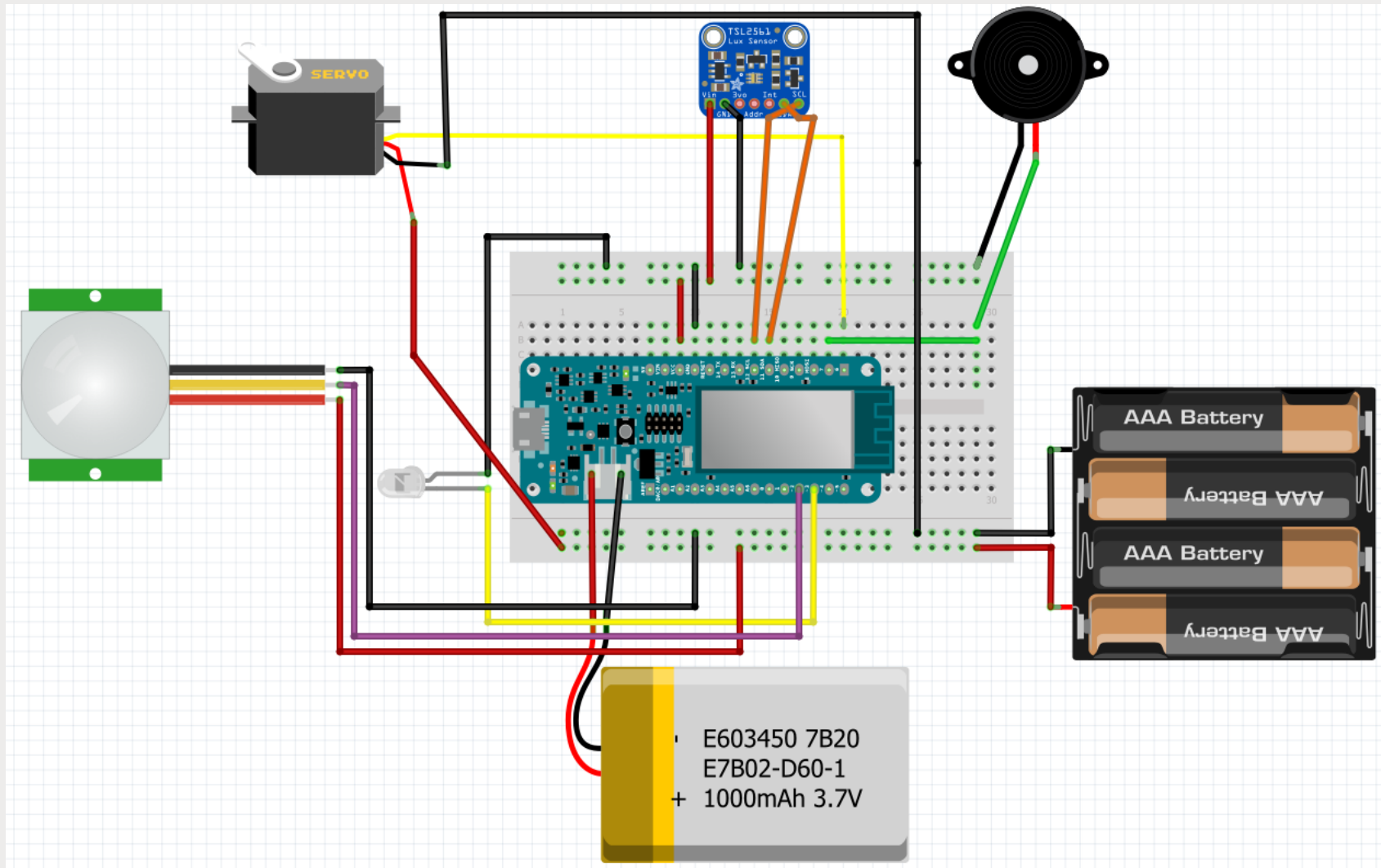


Under the Hood – Wires and Food

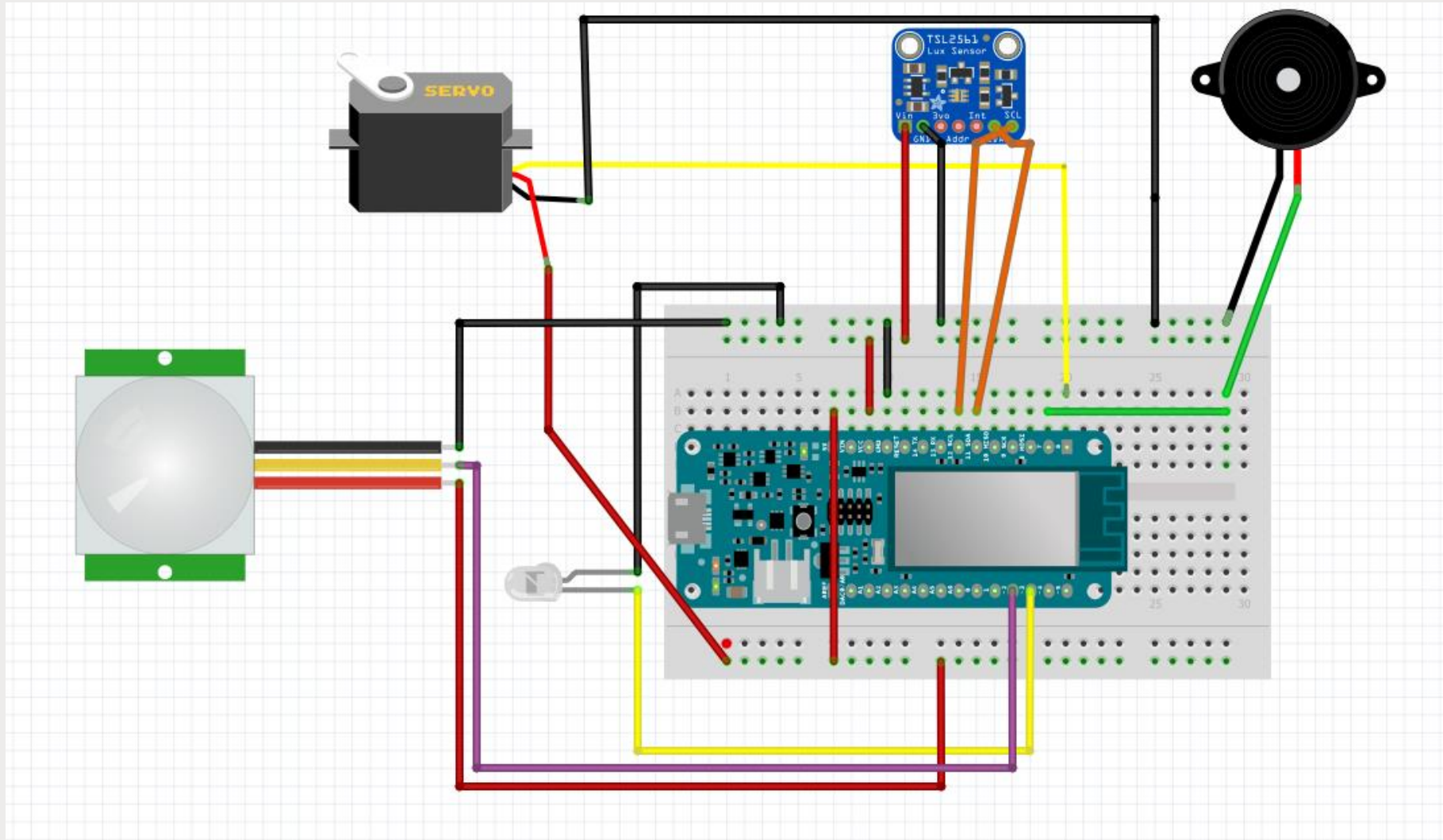




# Component Schema v1 – with Batteries



# Component Schema v2 – without Batteries



# Limitations and Problems Encountered

- Initial idea was to have a Laser Pointer at the top – Didn't work out well with the cat – It was destroyed. Nevertheless, I had 2 more servos (one for vertical movement and one for horizontal), which were randomly moving in pre-defined boundaries.
- The MKR1000 is working at 3.3v input in contrast to the other Arduino devices and this has caused problems, while trying to integrate different sensors.
- The layout of the pins of the MKR1000 is different compared to that of the regular Arduino and there is still not that much support and documentation on how to connect some of the components.
- The device works properly only when there is plenty of food inside, because it uses the gravity to dispense it and only opens and closes the hole.



# Contents of the files in the final submission:

- case.pdf – the Case of the device – needs to be printed, then cut and folded from the 1.5mm cardboard.
- cathub\_full.ino – the Scetch, which needs to be uploaded to the Arduino. In order to make it work, update the values, that are used to connect to the network and bot:

```
const char* ssid = "arduino_test"; // your network SSID (name)
const char* password = "88888888"; // your network password
const char BotToken[] = "820970992:AAGMisPtTsAAdDw6WakeEEYce1M31T7QuqR0";
```

- case.pdf – the Case of the device – needs to be printed, then cut and folded from the 1.5mm cardboard.
- html.h – HTML file, used to run the server. Can be changed according to needs and taste.
- pitches.h – File containing the note definitions, used to play the different melodies. One is able to compose his own melody using the notes defined in this file.