Project Report

Marina Polupanova, NetID = marinap2. (no group provisioned)
The access to the video demo: https://youtu.be/WKOKULkf7LI

My project contains development and testing of a hamster surveillance system.

Motivation.

I have a dwarf hamster at home, and I wanted to create an automation system, which would warn me if the hamster ran out of water, or if it would appear that the cage was not cleaned in time (as such situations already happened in the past). Also, I wanted to know more about my hamster and detect if any of his life conditions could be enhanced.

Main features of the system should be monitoring of the level of the water in his water bottle, how clean the cage is, and how content he is with his toys (which ones he prefers more often). In addition, out of curiosity, I will try to develop a way to measure the speed of a running hamster in the wheel.

Dwarf hamsters are very energetic, curious and fast creatures. They do not hibernate in winter, live in the semi-desert and steppe areas, and make tunnels one meter down the ground to make a house and hide from predators ([2]).

In all the directions for the hamsters care, for example [1], [4], it is emphasized that the hamsters need to have fresh water and proper food. I'm using the pellets and seeds mix, as recommended there, and the hamster does not finish it till I clean the cage for the next time, so I will not monitor this parameter. But as for water, it runs out quickly, so I will include the water monitoring component.

In [1], [4] it is told that the hamsters need many toys, the maze and the wheel for exercising and sharpening their teeth. So I wanted to explore whether the toys which I have now are suitable for my hamster: some toys to bite, ladders, swing, and a wheel. Reading [2], I inferred that the hamsters should love big space, and many movements, so I really thought that my big cage is a good place for my hamster, but I wanted to make sure. When we have got this hamster, we have received a special bucklet, telling us that the hamsters need to be entertained a lot, and in case if the hamster becomes bored, it can cause some kind of depression, and even life shortening.

Even if in many booklets for the hamster owners there is explicit mentioning of the wheel, I have found a paper [3], which does not find the convincing experience

arguments, that a wheel will add significant impact to level of activity or boldness of dwarf hamsters when they already have a maze.

Technical approach.

Initial plan of work was following:

1. To measure the level of water in the bottle I wanted to explore infrared sensor with receiver, and also the variant with sending the light from the bright diode to the bottle (as the bottle is not transparent with the normal light level) and receiving it on the other end with different intensity depending whether there is or there is no water in this level of sensor position.
2. For getting the info on how dirty the cage is, I was going to use the camera and make the dataset of the clean cage and the dirty cage photos.
3. For calculating the speed of the hamster in the wheel, I was going to use the color sensor and a tape of some bright color on the side of the wheel. The other variant, if the first will not succeed - to put a foil tape on the side of the wheel, and detect moving of the wheel by some send-receive light sensor.
4. For understanding which toys hamster likes more, I was going to use the sensors of presence.

I can tell from the beginning, that the sensors' performance and the hamster behaviour corrected all those plans a lot, so I will describe further the reason for some approach failure and which approach finally worked.

The level of the water in the bottle.

Probably this one item is the only one which worked out of the initial plan. I have used the infrared source of signal from one side of the bottle, and infrared sensor on the other side of the bottle, and the result was 100% reproducible, that the signal with the absence of water between infrared source and infrared receiver was detected, and with presence of water - not detected. So, I have finally put the source and the receiver of infrared signal on both sides of the bottle approximately on the lower ¼ side of the bottle, to have a trigger that the bottle need to be refilled.

How clean the cage is.

Initially, it was visible how dirty the cage is, just looking at it, when I was planning my project. But it happened, that the hamster by the time of the project implementation changed its behaviour, and there were no excrements in the sight of the camera (where I had the chance to put it). Or, alternatively, he constantly changed the place, where they appeared the most, so it was really hard to judge visually whether it is time to change the padding or not. But what did not change - is the smell from the cage in the days, when the cage became too dirty. So, I have assumed that I will need to take some

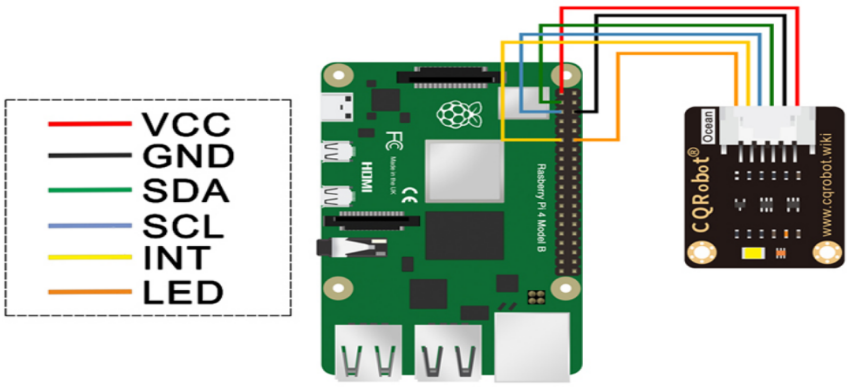
smell sensor and try to detect this odor. I had a sensor of methanol (sensor model MQ-2), and some other dangerous gas, but it did not work. After some search I was able to purchase a sensor of ammonia (model MQ135), which gave the change of the readings when you put him near the cage with odor and far from this cage. The change was not very high, but for me it was enough to detect the difference and tell that the padding need to be changed. By the way, one of the reason why the padding should be changed regularly is because ammonia and some other gases from the dirty padding irritate the organs of breathing hamsters and can cause diseases.

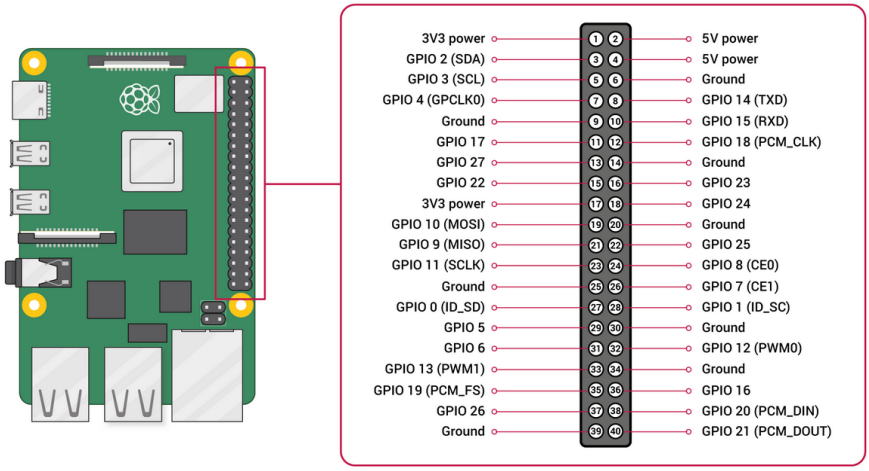
The speed of the hamster.

I was planning to use the sensor TCS34725 (Color sensor, www.cqrobot.wiki) to detect the passing of the color line on the side of the wheel and count the number of such passes. This count, multiplied on the circle length and divided on time would give the speed of the hamster when he runs in the wheel.

The specification of the sensor and the SW to run it were found on the following reference:
http://www.cqrobot.wiki/index.php/TCS34725_Color_Sensor

I have connected it using the diagram below. As the real sensor had the wires of different colors than in the schema on the right, I used the chart on the left to identify how to connect.





The picture above was taken from:
https://www.raspberrypi.org/documentation/usage/gpio/

The sensor output looks like following:

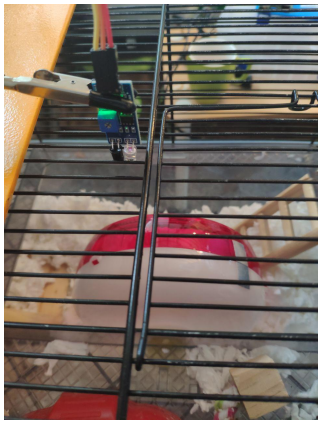
```
LUX: 23
CT: 3765K
INT: 0
R: 225
G: 181
B: 169
C: 0xeee
RGB565: 0xe5b5
RGB888: 0xe1b5a9
```

It outputs the value for each of the color channels, and final color code.

The testing of the sensor for the purpose of detecting how the different color line on the wheel coming across the wheel near sensor during movement failed on 2 reasons:

1. The speed of recognition is pretty slow for the current wheel movement
2. The sensor needs to be not more than 5 cm from the wheel to measure the other color passing by correctly. I have a wheel in the middle of the cage, so it is really challenging to reach it by such a sensor as it does not react to changing the color at this distance.

As this attempt failed, I have tried the other sensor which I had: infrared transmitter-receiver



I have fixed a foil tape on the side of the wheel (seen on the photo), and put the sensor and transmitter slightly inside the cage, as completely outside the cage it was reacting not on the reflections from foil, but reflections from the cage wires. And it worked: the sensor was able to detect the foil line passing and I have inserted the count inside my code to be able to calculate the speed of the hamster each time, when the hamster would run in the wheel each attempt. My code saves the output in 2 files:

1. log_wheel_rotations.txt - saves the timestamps (in absolute value in seconds) when the foil tape passed the sensor.
2. log_speeds.txt - saves the values of the speed each time, when the hamster runs a continuous time with the brakes not less than 1 second. As it can run each attempt with different speed and during different time, I decided to store each attempt speeds, and not the average speed over all the measurements. So that I would be able to detect the maximum one.

The only one drawback of the solution which I have chosen was that sometimes, due to close proximity to the wires, the sensor sometimes start to give the wrong readings constantly, but those can be easily filtered out, as the speed in that case would be number of times higher than the average reading. Also, due to the fact that the glass housing of the sensor is inside the cage, hamster has risk of eating it up (I have put it in the most inconvenient space for him to chew, but often he walk by hands on the wires, and in general he had some chance to grab it), so I did not make measurements longer than 2 nights in a row, not to increase this risk.

As infrared sensor appeared to be the most reliable of all which I have tried, that would be good as a continuation of this work in future, to try to find a way how to isolate it from the hamster, and at the same time constantly monitor the wheel for several weeks to detect certain behavioral patterns.

Position of hamster in the cage.

The first idea was to use the sensors like HC-SR501 Pir Motion IR Sensor to detect the position of the hamster in the cage (at least, which half of the cage he is, left

or right), but it did not work, as each sensor appeared to cover several meters of space, so all of them had threshold readings if the hamster was present.

As I have not yet used my camera, and I had several ones, I decided to explore the possibilities of OpenCV (as I did not have enough time for that in the lab 1) and try to capture the hamster with the camera night vision.

There were 3 cameras which I have explored during my testing:

1. Camera for Raspberry PI, used in Lab1
2. Camera for Raspberry PI with day/night vision https://www.amazon.com/Raspberry-Camera-Vision-IR-Cut-Longruner/dp/B07R4JH2ZV/ref=sxts_sxwds-bia-wc-rsf-lq2a1_0?crid=VTLQ4ELZLVQJ&cv_ct_cx=camera+raspberry+pi&dchild=1&keywords=camera+raspberry+pi&pd_rd_i=B07R4JH2ZV&pd_rd_r=f530e048-9ed3-45d7-bfa9-c94899df3e74&pd_rd_w=8PsAr&pd_rd_wg=pP2kZ&pf_rd_p=26b0e770-de1c-4342-bf97-c57fd874dbaf&pf_rd_r=W6N7AEYHJ5G0P07R7EC9&psc=1&qid=1620511588&sprefix=camera+raspbe%2Caps%2C367&sr=1-1-49946e88-733b-44df-869b-c05699555c56
3. USB web camera with night vision: https://www.amazon.com/dp/B07RQ1MVZ3?psc=1&ref=ppx_yo2_dt_b_product_details

The code, which I had made with OpenCV library, captured the pictures with reduced size (not to overload the Raspberry PI flash memory during night) each ¼ seconds, and compared two consequent frames for similarity. If the mean absolute value of similarity for 2 frames was more than 2, it stored the frame as a picture on the disk, naming it “image{}”, where {} is the count of the frame taken.

I have tried different means absolute difference values thresholds, and figured out, that :

- Night cameras have bigger threshold values (around 3), where the hamster movement is seen, but the background noise is ignored
- Both day and night cameras have different threshold values in day and night. Day camera (1) had the average day value around 1.5 - 2.1 when nothing was changing in the day, and around 2.5 - 3 in the darker room, same ratio was valid for the night camera except for the figures in the day were starting from higher values like 2.5.

Please see for your reference two photos, taken by camera (1) and (3) in the light environment, and in the dark environment.

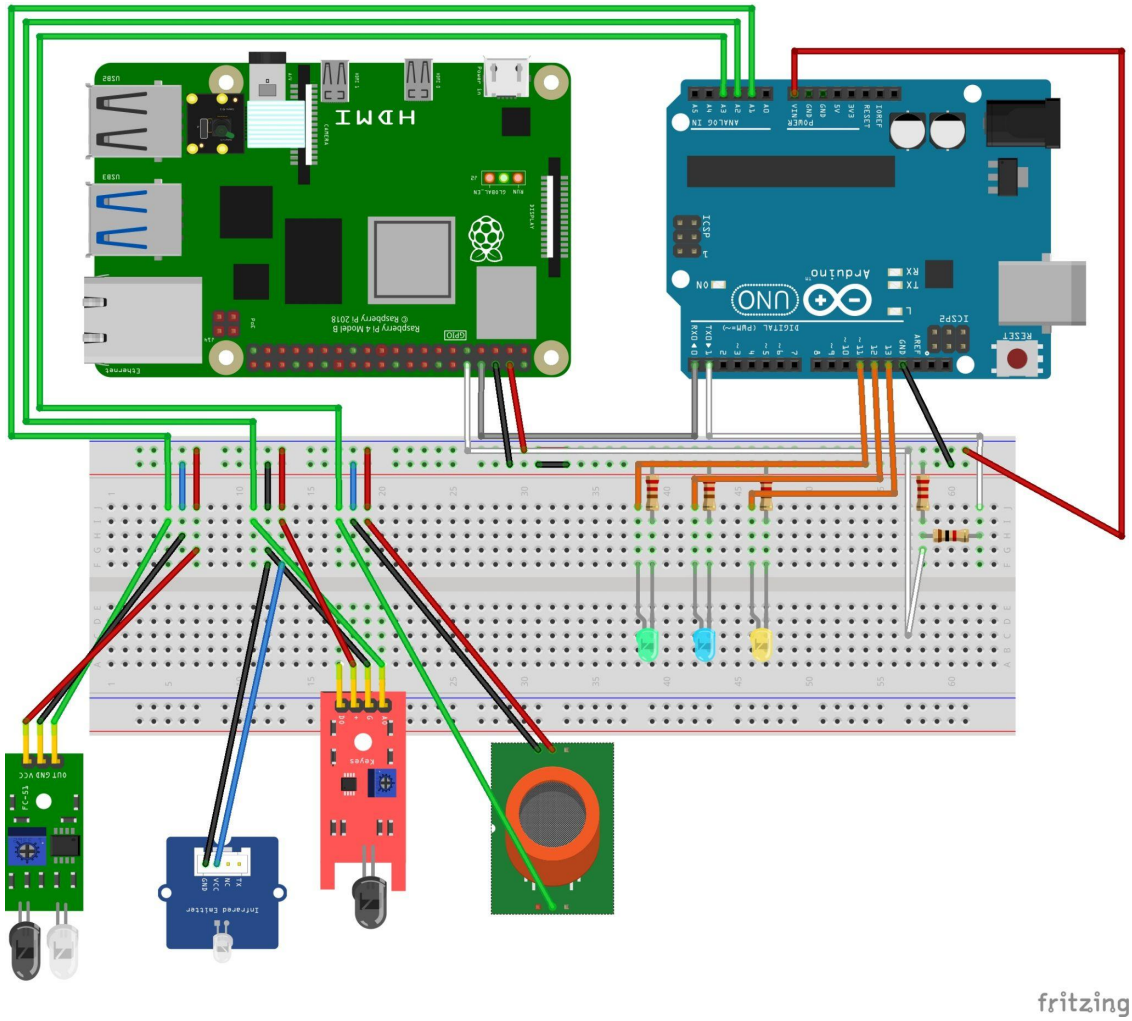
Camera (3) - in dark photo was a very small white lamp around:


Camera (1) - in dark photo was a red lamp and a small white lamp:


In case, if I switched all lamps, no photos could be taken from all 3 cameras, even the night ones. The pictures came like that:

I will give the final results and conclusions on this task at the last chapter of the report, but telling shortly based on the results and the conclusions I decided not to go with this task further than that, so no model was constructed based on the pictures taken.

Overall architecture of the monitoring system.



Implementation details.

Camera and OpenCV

In order to make the code working, I first double-check the installation of Open CV which I made in lab 1 using the link, and found out that I have lost several steps, so that's why it was not working correctly before. Also, I ensured that numpy library is installed in the virtual environment as well.

I used the following resource to get some ideas how to capture the images from the camera to get the hamster moving statistics:

https://docs.opencv.org/4.1.0/dd/d43/tutorial_py_video_display.html

https://people.revoledu.com/kardi/tutorial/Python/Video+Analysis+using+OpenCV-Python.html

Based on the code snippets, learned from those links, I have done the following code.

First, I have imported the library and started the video capture.

```
import numpy as np
import cv2 as cv
import time


index = 0
cap = cv.VideoCapture(0)
scaleFactor=0.5


if not cap.isOpened():
    print("Cannot open camera")
    exit()
```

Then, the camera starting capturing the frames one by one with the interval of 1/25 seconds.

```
while True:
    # Capture frame-by-frame

    ret, frame1 = cap.read()  # first image
    time.sleep(1/25)          # slight delay
    ret, frame2 = cap.read()  # second image
```

Then, we calculating the difference between the first and the second frame, and resize one of them to occupy less space on the disk further:

```
  difference = np.mean(cv.absdiff(frame1,frame2) ) # image difference
  print("difference", difference)
  frameSmall=cv.resize(frame2,(0,0),fx=scaleFactor,fy=scaleFactor)
  print("printing", index)
```

Initially, I was going to make the pictures in gray color, to save even more space (as I planned to have big dataset), but it appeared that even I was not able to reliably notice the hamster due to his fur colour in the cage when the image was gray so it was hard for me to label the images. So I decided to come back to the color image.

In case if this difference between frames was bigger than 2, the program wrote the image on the disk with the index of frame capturing iteration.

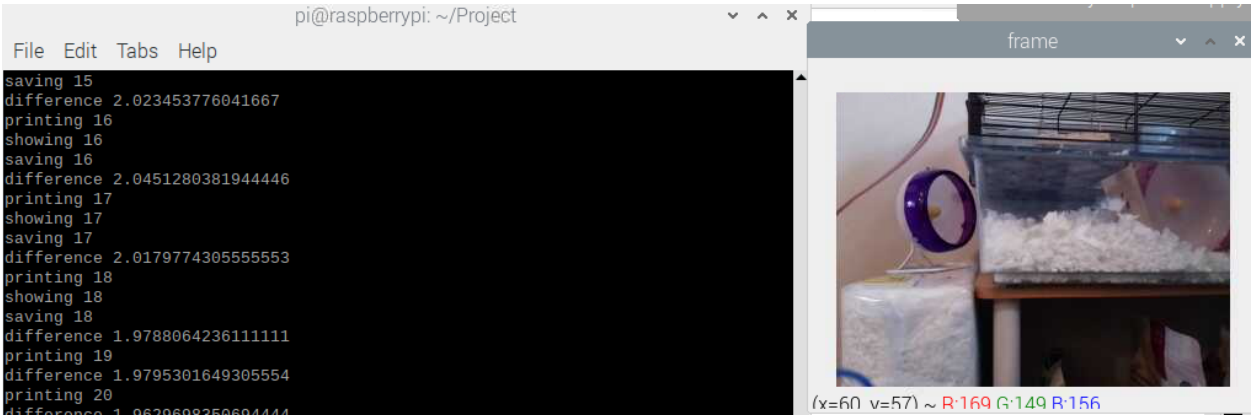
```
  if difference > 2:

      cv.imshow('frame', frameSmall)
      print("showing", index)
      cv.imwrite('image{}.png'.format(index),frameSmall)
      print("saving", index)
  index =index + 1
  time.sleep(1)
```

Then, in case if the button “q” was pressed, the program would exit, and the camera capture would stop:

```
    if cv.waitKey(1) == ord('q'):
        break
# When everything done, release the capture
cap.release()
cv.destroyAllWindows()
```

The output of the capturing program was printing the difference between frames, the index of what frame is being shown now or saved, and a frame window, which showed the frame itself.



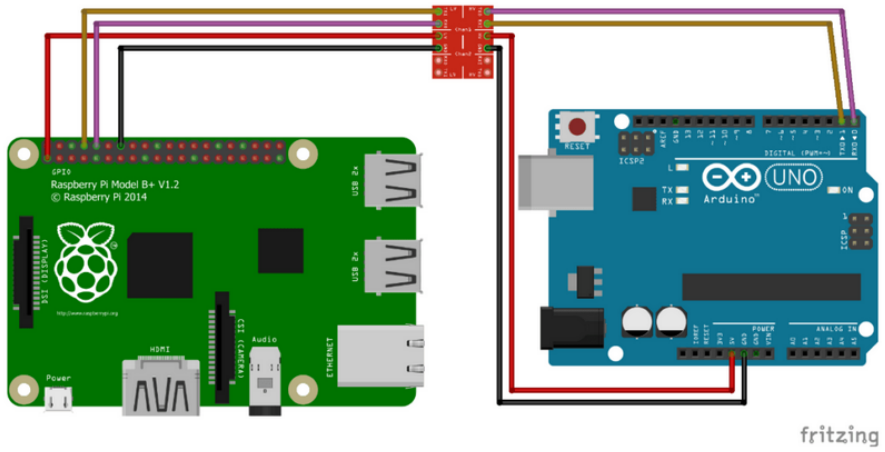
To connect the camera, I have used the standard RaspberryPI slot/shield for 2 first chosen cameras, and USB port for the last one. It appeared, that in case if you are rebooting Raspberry PI after the camera change, it lans on the same video capture port 0, so I did not change the code when I moved from standard shield connection to USB connection.

Arduino sensors.

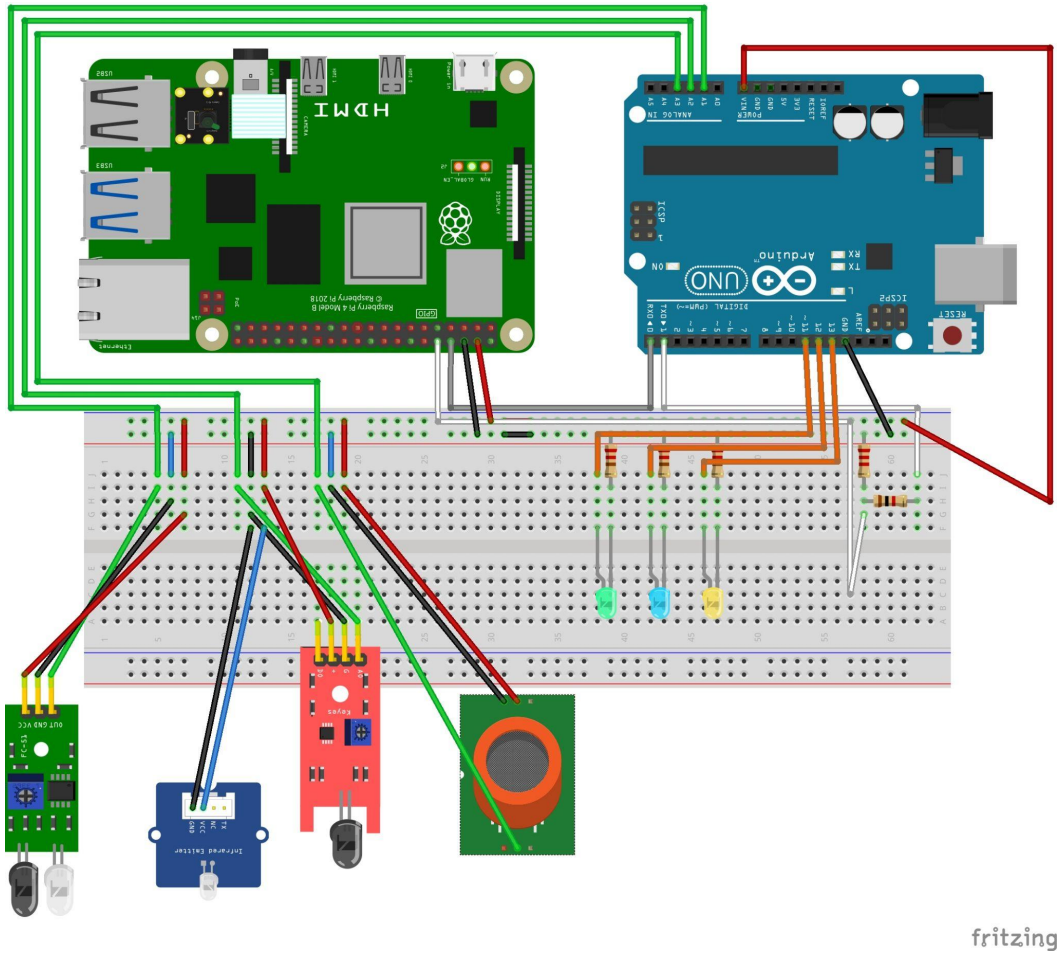
It happened, that the most sensors, available on the market are using the input/output of 5V, which is more suitable for Arduino, than for Raspberry PI (at the link of GPIO description it was told that the input should not be higher than 3.3 V in any case for Rasbperry PI), but at the same time as I had camera, which was already connected to Raspberry PI, I decided to connect the sensors with Arduino, and then with Raspberry PI. Alternative to that would be to create a power dividing scheme for sensors connections, but I decided to go with the serial Arduino-Rasbperry PI connection, for which I have found a good article with the connection and code examples:

https://roboticsbackend.com/raspberry-pi-arduino-serial-communication/

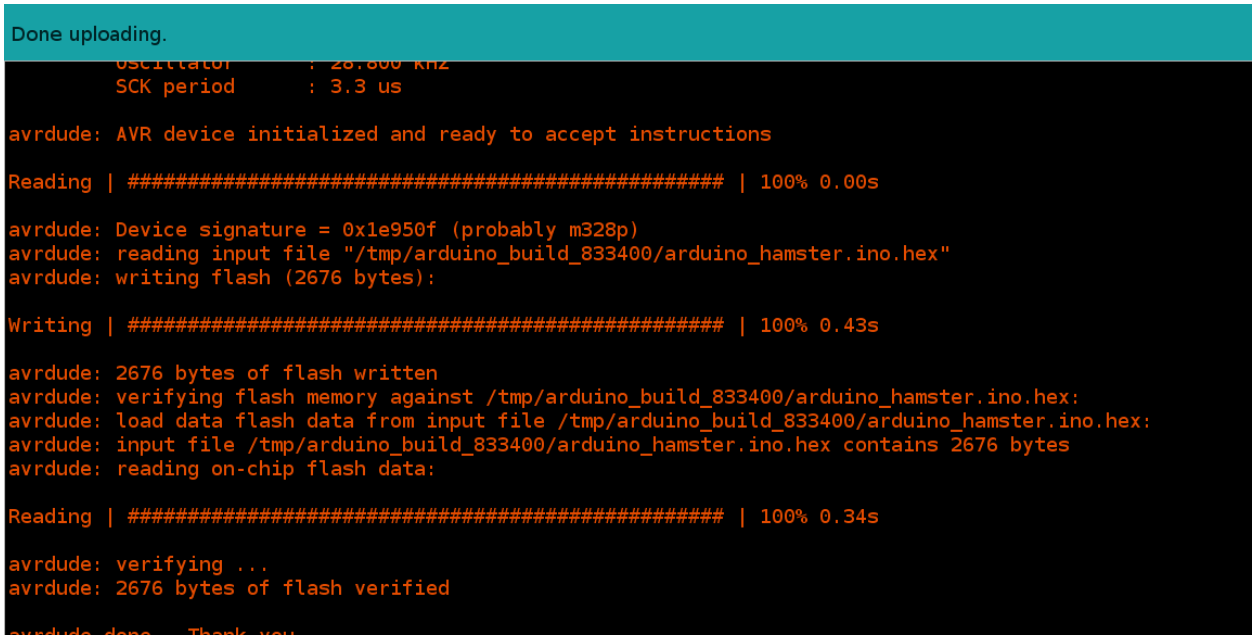
There was a way to connect two boards with serial cable, which worked, but as I needed anyway to make the feeding and grounding sensors, I took the second proposed connection scheme from that manual:



Then, I have connected it to breadboard, and connected the sensors, as I have shown above in the overall connection diagram:



I have installed Arduino Arduino IDE 1.8.13 from the link, and have constructed the code, which would receive the inputs from 3 analog pins, and send the values to the serial port. Arduino program load output can be seen on the screen below:



For making the program for Arduino, I have taken the ideas from the following link:
https://create.arduino.cc/projecthub/Aritro/smoke-detection-using-mq-2-gas-sensor-79c54a

The program contains the pin numbers announcing for all 3 types of sensors and 3 leds which will light if they will get a signal.

```
//Numbers of leds for different sensors inputs
int sensor_1_LED = 11;
int sensor_2_LED = 12;
int sensor_3_LED = 13;

//Number of pins for sensors inputs
int sensor_1_DATA = A1;
int sensor_2_DATA = A2;
int sensor_3_DATA = A3;

//Thresholds for sensors
int sensor_1_TRESH = 600;   // Wheel
int sensor_2_TRESH = 600;   // Water
int sensor_3_TRESH = 230;   // Ammonia gas

//Status of sensor (above threshold or not)
bool sensor_1_MSG = false;
bool sensor_2_MSG = false;
bool sensor_3_MSG = false;

int ammoniaTimer = 0;
```

Then, we specify which variable is captured on which port and which led will light in case if the LED variable will get a value.

```
void setup() {
  pinMode(sensor_1_LED, OUTPUT);
  pinMode(sensor_2_LED, OUTPUT);
  pinMode(sensor_3_LED, OUTPUT);
  pinMode(sensor_1_DATA, INPUT);
  pinMode(sensor_2_DATA, INPUT);
  pinMode(sensor_3_DATA, INPUT);
  Serial.begin(115200);
}
```

Next, we read in the loop the values of 3 analog pins:

```
void loop() {
  int sensor_1_VALUE = analogRead(sensor_1_DATA);
  int sensor_2_VALUE = analogRead(sensor_2_DATA);
  int sensor_3_VALUE = analogRead(sensor_3_DATA);
```

Then, we check, if the value of the sensor is bigger than threshold, Arduino checks if the value was already sent (if not it sends it and record in “MSG” variable that it was sent as passed the threshold, led is lighting), and if the value is smaller than threshold, the led do not light, and the “MSG” variable put back to FALSE in order that next time when threshold will be passed it will send the message to serial port again. This logic is done for each of 3 sensors.

```
  if (sensor_1_VALUE > sensor_1_TRESH) {
    if (!sensor_1_MSG) {
      Serial.print("Sensor_1: ");
      Serial.println(sensor_1_VALUE);
      sensor_1_MSG = true;
    }
    digitalWrite(sensor_1_LED, HIGH);
  } else {
    digitalWrite(sensor_1_LED, LOW);
    sensor_1_MSG = false;
  }
```

In addition, for the gas sensor, it is sending the value itself each 1 minute in order to be able to discover some statistics with it:

```
// Send ammonia sensor value to RPi
if (ammoniaTimer >= 6000) {
  ammoniaTimer = 0;
  Serial.print("Sensor_3i: ");
  Serial.println(sensor_3_VALUE);
} else {
  ammoniaTimer++;
}
```

Then, on Raspberry PI I launch the program, which reads the data from serial port, and prints the data to the screen and to files with corresponding data.

First, we define all the variables for list of speeds (speeds), list of timestamps in seconds format when the hamster was running (wheel_register), and files where we will put all the data.

```
import serial
import time
if __name__ == '__main__':
    ser = serial.Serial('/dev/serial0', 115200, timeout=1)
    ser.flush()
    wheel_register = [0]
    speeds = [0]
    counts = 0
    time_t = 0
    f_wheel  = open("log_wheel_rotations.txt", "a")
    f_speeds = open("log_speeds.txt", "a")
    f_ammonia = open("ammonia.txt", "a")
    f_water = open("water.txt", "a")

```

Then, we are recording current time stamp and read the data from serial port in the variable “line”:

```
while True:

    if ser.in_waiting > 0:
        cur_time = time.time()
        print("have data")
        line = ser.readline().decode('latin-1').rstrip()
        print(line)
```

In case, if we are receiving the data from the speed sensor (Sensor_1), we recording the timestamp to the file “wheel_register.txt”, and figure out if between the neighbour timestamps it is less than 1 second (means, the iteration of run continues). In case if yes, we add the time between timestamps to the “time_t” accumulator. In case if no - means hamster stopped running. In this case, we calculate the speed of current run iteration, and writing it to the file “speeds.txt” with the final timestamp. And we put back to zero the counter, the time accumulator, and the speed variable.

```
        if line.startswith("Sensor_1"):
            wheel_register.append(cur_time)
            f_wheel.write(str(cur_time) + "\n")
            print("1", wheel_register[-1])
            print("2", wheel_register[-2])
            if wheel_register[-1] - wheel_register[-2] < 1:
                counts = counts + 1
                time_t = time_t + wheel_register[-1] - wheel_register[-2]
                print("no")
            else:
                if time_t>0:
                    print("speed")
                    speed = counts * 53.38 / time_t
                    speeds.append(speed)
                    f_speeds.write(str(cur_time) + " " + str(speed) + "\n")
                    counts = 0
                    time_t = 0
```

In case if the data sent from sensor of gas, or sensor of water, I just display it and record in the file with the corresponding time stamp.

```
    elif (line.startswith("Sensor_2")):
        print(line)
        values = line.split()
        f_water.write(str(cur_time) + " " + str(values[-1]) + "\n")
    elif (line.startswith("Sensor_3")):
        print(line)
        values = line.split()
        f_ammonia.write(str(cur_time) + " " + str(values[-1]) + "\n")
```

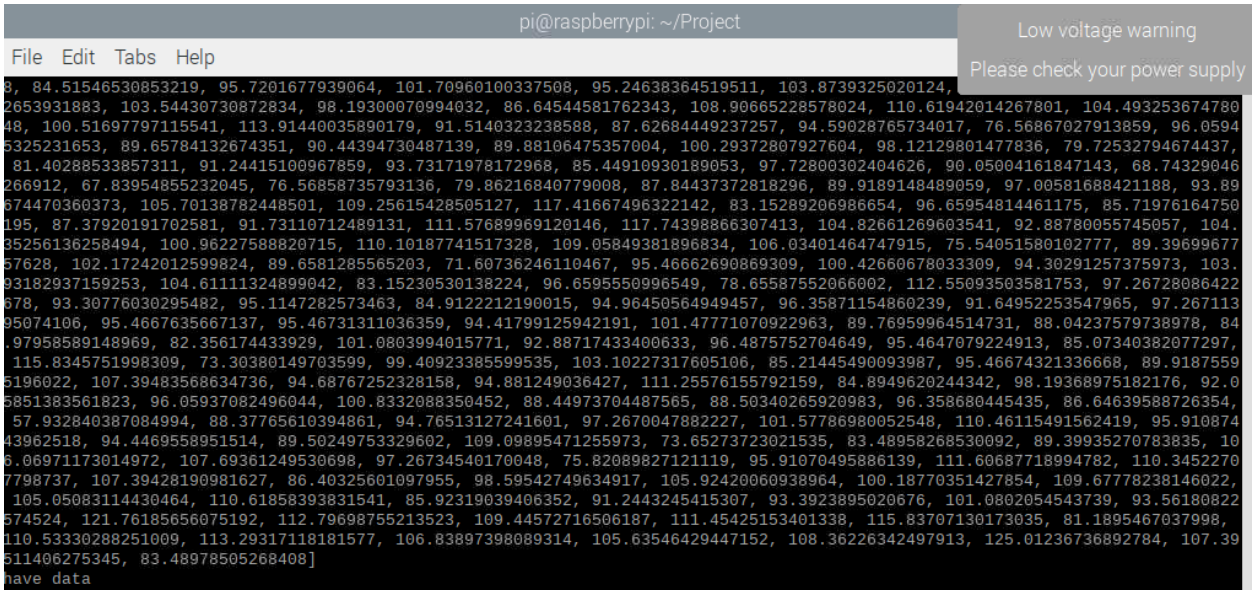
Results.

Results from Arduino sensors.

Here are some examples of the terminal output from the “arduino_capture.py”:

- The day time readings, when I have just launched the program, and the “speeds” and “wheel_register” arrays are zero, and the data from water sensor and gas sensor are coming. The value 1023 from water sensor means, that there is no water now, so as possible further project evaluation is possible to make some notification to the cage owner that the water bottle needs refill. The value from the gas sensor means the current reading of the gas value, so it would change over the week as the cage will become more dirty.

- Here is the example of the screen after night, when the hamster was running all night, and the arrays of “speeds” and “wheel_register” became so huge, that occupy more than one screen.



Here are some examples of the files content, when there are readings of gas, water, speed and also the timestamp when the hamster was running:

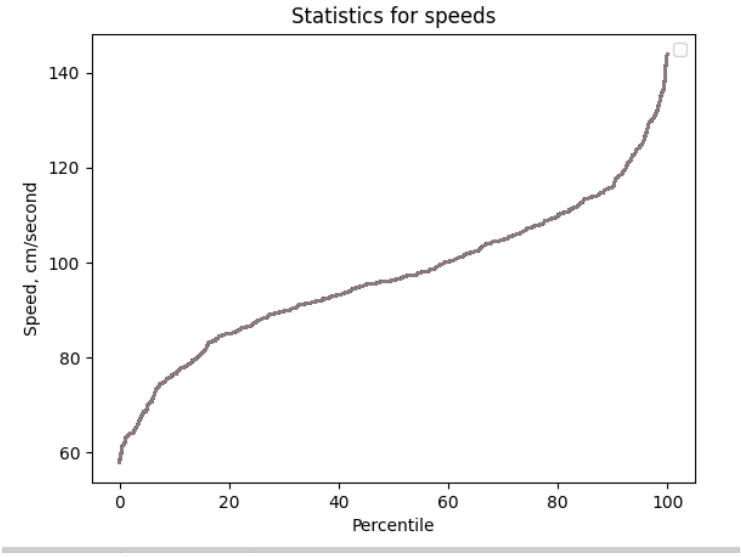
Fragment of “ammonia.txt”:

```
1620593880.960458 275
1620593943.0700674 272
1620594005.1798155 267
1620594067.2896273 269
1620594129.3995175 273
1620594191.5094879 270
1620594253.6195424 263
1620594315.7296593 270
```

Fragment of “log_speeds.txt”:

```
1620454887.4749591 111.57972447239183
1620454889.1211987 642.4390823608405
1620454895.395293 123.99180409777074
1620454900.810113 131.59849877095866
1620454904.1959643 119.32324049448819
1620454916.6606922 130.33876398695477
1620454921.4760182 138.46627283777562
```

I have captured the readings of the speed for 2 consequent nights (did not do more, as I did not want to stress out the hamster with the lights too much (as the diodes were blinking, and most probably made him very nervous, so he was almost not appearing in the light time. I was even worried that he would have issues without water and food sitting in the shelter for so many consequent hours. Based on those results, the maximum hamster speed is 143 cm/second. I have a few bigger readings in the file (bigger 600), but it was due to the error with the sensor as at a certain moment of time the lamp went upper and started interfering with cage wires. Here is the distribution of the speed statistics over 2 days:



I liked the stability of the IR sensor results so much, that in case if I would set up the targets for the future survey project, I would put the task to discover the average time/speed of running over the night during long observation period (several months), and correlate it with some events in the hamster's live and also temperature of the room and from how clean is the cage. The only one thing which would be needed for this type of task is to make the sensor as invisible for the hamster as possible not to stress him with blinking lamps (one idea would be to put the wheel on the cage wall, and put the sensor behind the wheel, and not on top as I did now.

For the gas sensor the readings were in the range from 231 when the cage was just cleaned to 410 when it started smelling, and it took it about 4 days to come to that level. I assume that approximately at this level of sensor reading, or at least not more than 450 the cage should be cleaned.

The quality of work for gas sensor was not completely satisfying: sensor had a heating curve, when the readings were not reliable at the sensor start for about half an hour, so in case if for some reason the program was stopped or crashed (especially when I was tuning the other sensors), the curve again appeared and messed up the results. So, currently for my reference, I took as correct and reliable only the values, which were popping up at least 10 times in a row.

For the water sensor, I have got a reliable indication of absence of water in the bottle, so that in case if the water would be finishing, it sends the "1023" value immediately. For this kind of sensor - it can be used constantly, but same as for the speed sensor, some work needs to be done to make it less visible for the hamster and less stressful than now when I have some diodes lighting and blinking at night.

Summarizing the experience with sensors, I can tell that they are efficient enough to capture what they were supposed to capture, but the output needs to be filtered sometimes to present the results without side effects, and it should be created some more hamster-friendly appearance of the whole construction.

Camera results.

For the camera, as I have described in the first part of my report, I had multiple issues:

1. At night all the cameras gave very vague picture, that in case if the hamster would not go straight to the camera - it was not possible to recognize his presence. The level of background noise, especially at night, was preventing to see him either by eyes, or by the Open CV tools
2. The level of background noise was preventing to recognize hamster movement even in the light time. Strange, but if I set the threshold 3.5 in the morning time, the camera did not detect the hamster movement, but saved photos (as the frames difference without hamster movement was more than 3.5). Later I detected that it was rays of light and movement of people around, reflecting in the glass of the cage. I would suspect that it possibly can be fixed by putting the cage and the camera to some room without reflecting surfaces (now a window and a white wall was near, so some reflections possibly could be caught as a background noise).
3. More serious obstacle to the observation: when the hamster detected that he is being observed, he stopped to come out from the shelter, or, if he came out to drink or to eat, he quickly did it, and run as fast as he can to the shelter. He went out for a walk only in the time, when it was complete darkness and no lamps were around (one day I have left the lamp on low light and other night - a slight red lamp, and he did not came out until the lamps run out of battery and only at those times it started the readings of the wheel. One night I succeeded to start recording from early morning, when the light started to appear in the window, shot 8000 photos for at least 4 hours till 9 am, but hamster appeared only on 20 photos and it was the case when he ran to drink. I made experiments with the camera mostly on weekend, and on working day I put it on my table. In that days the hamster left the shelter much more often.
4. Second even more serious obstacle to continue the task, put for camera observation (get the more often position of hamster in the cage) - was recognizing that the initial task was put incorrectly. When I started my work, I assumed that I would detect which toy the hamster loves more, and so get the conclusion if he is content with his presence in this cage. While I was trying to adjust the camera and other sensors, I was looking at

him much more than I did before at the times he went out of the shelter, so even without the camera I'm now able to say that the task was put incorrectly. What I have detected:

a. The hamster was chewing the wooden toys almost equally. Sometimes he used the swing
b. He run on the wheel only at night (very rarely in the evening)
c. Most of the time that he was not eating/drinking he was walking on his arms/paws on the ceiling and side walls of the cage and was chewing it. He spent for that much more time during the light time than chewing/drinking or anything else (and remind, never run a wheel). In case if the camera capturing would succeed, I would never be able to detect if he is walking on his arms on the cage or doing anything else near this place using the neural net, so I decided to quit working with pictures. For me, the speed/duration of running a wheel seem to be more reliable indicator of his mood.

Adding to my experience of using the camera for hamster capturing, I found perfect paper (unfortunately, already in 2-nd part of my project journey, so did not have time to implement some of my findings) [3], where it was described the process of making camera observations of the hamsters. First, I have picked up from there the idea to use the red light for night camera shots. My issue was that the power of light which I had was too small even to help my night camera. For any next attempts or steps in the project, more powerful lamp will be needed. A reason why I started to search for the papers was that the hamster was constantly chewing the cage, and did not use the wheel, so I tried to understand why. In some internet articles, for example [5] it was considered that he is bored or anxious, or requiring people's attention. So second, I was surprised to see that the wheel in the [3] was considered to be just an addition to the maze, and the maze is exactly what we do not have.

So, summarizing, as a result of observation, the hamster is not content of his cage, and the way of detecting his being not content was not right. In future, I will need to add a maze, and probably some more entertainment in his life, and in case if I will try to make any automatic camera observations, I will need to have a more powerful red lamp, and make the camera more invisible for the hamster not to stress him even more. The task would be not to detect which place (half) of the cage the hamster is, but certain positions of his body, which will reflect more what he is doing rather than where he is standing. For this the dataset would need to be much bigger than I was able to obtain so far.

References:

1. https://www.hssv.org/wp-content/uploads/2019/04/Hamster-Care.pdf

2. https://animalia.bio/djungarian-hamster
3. L.Leann Kanda, Amir Abdulhay, Caitlin Erickson. Adult wheel access interaction with activity and boldness personality in Siberian dwarf hamsters /BehaviouralProcesses138(2017)82–9
4. https://petponder.com/how-to-care-for-your-pet-djungarian-hamster
5. https://firsthamster.com/reasons-hamsters-chew-on-cage-bars-and-how-to-stop-them/