

Innehåll

Project Deer hunter	2
Project overview:.....	2
Problem:	2
Vision:	2
Project overview:.....	2
Time:.....	2
Objective	2
Why this project?	2
Purpose of the project:.....	3
Project insights:	3
Material	3
Computer setup.....	4
Chosen IDE.....	4
Some thoughts	5
Installation and preparation.....	5
Putting everything together	6
C-SR501 PIR Sensor	6
Loudspeaker 8 ohm 0,25 W direct connection	6
Loudspeaker 8 ohm 0,25 W via amplifier class-D 2,5 W mono connection.....	6
Components to put all together	6
The current solution/functions	7
Next steps.....	7
The code	7
Current solution code.....	7
Main.py.....	7
mqtt.py	9
circljud.py	9
mixer.py.....	9
Transmitting the data / connectivity.....	9
Presenting the data	10
Download of Adafruit motion data	10
Finalizing the design	10

Project Deer hunter

Kari Talltjärn - kt222qq

Project overview:

Problem:

Deer's are eating my garden plants.

Vision:

Get a solution that detects the animals, trigger some sound and get them running away.

Project overview:

1. Gather knowledge about technical platform, communication, sensors, programming and environment for presentation of statistics.
2. Establish a test environment with motion detector, play sound and get statistics when animals visit the garden. The latter in order to be able to take further action.
3. Establish a technical solution to be able to use the solution outdoors and in Nordic climate.

Time:

Estimated total time for delivery of the above strategies: 80+40+80 hours

Objective

Why this project?

The main reason is that I want to preserve my seedlings in the garden. Every year I sow seeds or buy plants / bulbs that I want to build my garden with.

I do not want the deer's to eat them.

Another reason is my interest in IOT. During my last years in worklife, I learned how to make applications for mobile phones using google as "teacher".

Developed some mobile apps where you can use the phone in (Volvo Eskilstuna) factory. E.g. Maintenance reporting and Transmission inspection.

At the same time, there were many test activities to buy different sensors that were connected to suppliers' cloud services. The solutions were suppliers' closed systems. Many experiments but no strategy!

Would like to learn how to establish more open solutions and use them in my home. Have meanwhile invested in building solutions such as Ikea lighting, control of floor heating via mobile application, solar cells, geothermal heating, google home, etc.

There is more to learn and to do. This is now a hobby for me.

Purpose of the project:

Via a motion detector receive signals to an application, send a "frightening" sound to a speaker and hopefully scare the animals away.

Optionally add flashing lights, etc. as an addition.

Collect statistics and present a graph of the periods during when the animals visit the garden. The latter is to take other actions such as using e.g. "Gold Water". For more information on this, I refer to google!

Project insights:

In addition to hopefully getting a solution for the garden, I hope to learn about microcomputer environment.

I already have a red-white Raspberry pie that I haven't really done anything with. It is just collecting dust!

With insight into this environment, in addition to the possibilities of the microcomputer, I get to learn sensors and their opportunities, Python (for microcomputers), communication solutions, cloud services for presentation.

As a private person, I appreciate that there are solutions that can be established at low costs.

Material

Included material and components in this project:

Component	Usage	Price SEK	Supplier	Comment
Raspberry pie pico W	CPU and 40 connections for sensors, etc.	310.2 ¹	Electro:kit	Low cost cpu with enough amount of connections for IN/OUT signals and host applications with logic
Cables	Connecting Raspberry Pie with sensors and output devices	310.2 ¹	Electro:kit	
PIR motion detector HC-SR501	Trigger motions in front of sensor and causing event in program	39.2	Electro:kit	
Speaker 8 ohm 0,25 W	Testing output sound when motion triggered	7.6	Electro:kit	Required CircuitPython (my knowledge?). Sound became just noise and not just as an .mp3 file
Amplifier klass-D 2,5W mono PAM8302A	To improve the sound. Connected between CPU and speaker	47.2	Electro:kit	Required CircuitPython (my knowledge?). Could not get this to work.
Python 3	Programming of application logic	Free	https://www.python.org/	
Circuit Python	Programming of application logic. For testing sound.	Free	https://circuitpython.org/	Tried to install Adafruit-Blinka 8.19.0 and Adafruit-PlatformDetect-3.46.0 to run both python 3 and Circuit Python. Not enough space to install both.

¹ Included in startkit

Adafruit	Presentation GUI as cloud service. Showing no of motions per hour	Free ²	Adafruit industries	
Thonny	Development and testing environment	Free	https://thonny.org/	Did not get VS Code with PyMakr to work for Rasberry Pie. Took too much time and used Thonny instead.
Private wifi	Sending motion messages via internet to presentation in Adafruit IO	Free		

Computer setup

Chosen IDE

I tried a number of times to install VS Code, Node js and PyMakr but failed to get program running on my unit Rasberry Pie Pico W.

I normally use Visual Studio for creating applications in C# and Xamarin, Windows Forms, WFP so I would prefer to work in similar environment.

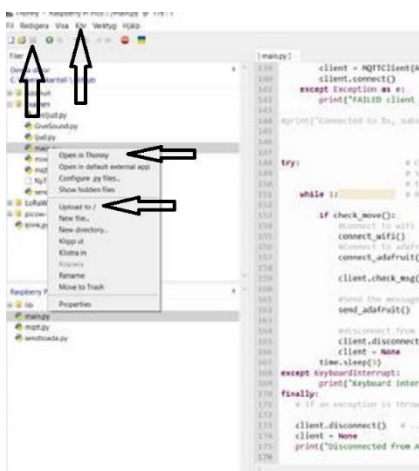
I instead installed and use Thonny as IDE. Can be downloaded for free and you will work with two file systems. Windows dictionary and Rasberry Pie dictionary. Normally you edit your program version in Windows dictionary, copy it to Rasberry Pie dictionary and then execute the code in Rasberry Pie.

The interpreter (making source code to machine code) informs you with row number and error text if there are any issues with the code. The code used is Python 3.

Hint: Be aware in what version you are editing. In Windows filesystem or in Rasberry pi. I always close files when starting new activity and always edit in windows filesystem version.

Most used actions are below. Just right click on mouse and select.

- Save
- Open in thonny
- Upload to Rasberry pie
- Open in thonny (Rasberry pie version/file system) and run



² Some maximum limitations but enough for this project

Some thoughts

This technology area is very much new for me and during these weeks I have become aware of other technologies within same solution area.

It seems that there is a variety of brands, platforms, services in this fast growing area. IOT was “hot” in my last years in work life and I understand it still is?

So the interesting question is what will be the “global standards” in future or segments big enough to provide solutions and “earn money”?

Some areas I have touched:

- Raspberry pie Pico W – Arduino Uno
 - E.g. Kjell and company are providing equipment for the Arduino platform
- Python 3 – Circuit Python
 - E.g. I found only sound solutions made with Circuit Python on google
- Communication solutions/protocols
- Cloud solutions

The future will be interesting.

Installation and preparation

As mentioned above I decided to use Thonny as IDE.

Steps to prepare environment:

1. Download and install Thonny - <https://github.com/thonny/thonny/wiki/Windows>
2. Install python 3 - <https://www.python.org/downloads/>
 - a. Download the package
 - b. Connect an USB cable to your Raspberry Pico. Do not connect to your Windows computer
 - c. Hold down the white button on your Raspberry pico and connect the USB cable to your Windows computer.
 - d. Release button and you will see a new logical disk is showing in file explorer
 - e. Copy the python download file to the new disk. Python is initiated/flashed and the disk disappear in file explorer. Done!
3. Create a folder in windows file explorer where you store different project folders/applications
4. Adafruit configuration
 - a. See the tutorial - <https://learn.adafruit.com/welcome-to-adafruit-io/getting-started-with-adafruit-io>
5. Sound test – It seems that supporting sound from Raspberry Pico I need to use circuit python. I did some installations and testing but did not get this to work. Here is a summary
 - a. Installed just circular python to test connecting a loudspeaker directly to Raspberry pie. Got some noise but not as my sound file (.mp3)
 - b. Added an amplifier to the a solution but did not get any sound at all
 - c. Since the found solution for sound required Circuit Python I tried to install Adafruit blinka and Adafruit platform detect to use both python 3 and circuit python code. This failed because lack of space.

Putting everything together

Circuit diagram for used platforms are shown as below. Images in Github.

C-SR501 PIR Sensor

The HC-SR501 has a 3-pin connector. The markings are hidden by the Fresnel lens, so refer to the following image for pinout.

See image in Github.

Loudspeaker 8 ohm 0,25 W direct connection

Simple loudspeaker.

See imaged in Github.

Loudspeaker 8 ohm 0,25 W via amplifier class-D 2,5 W mono connection

Simple loudspeaker vid amplifier between Rasberry Pie Pico W and Loudspeaker.

See image in Github.

Components to put all together

Until now I have not found any solution for the loudspeaker. Do I have to use circular python? Learning Python 3 is new for me and Circular Python will be the same. In my current timeplan I am just focusing that current used components should work together.

I have made inquiries about motion detectors for outdoor conditions. Request made to:

- Kjell och Company – They have solutions but all are “closed” using the supplier’s presentation GUI. Have not yet found a solution where I could e.g. export the motion detector file
- Electro:kit – No response

The current solution is limited to detecting motions and sending motion info to Adafruit for presentation. The presentatiuon is of secondary interest for taking other actions avoiding deer’s in gardens. Taking the info presented (Date/Time of day/night) to take actions.

Since this is not prioritized I just selected a cloud service that was simple to setup. I am not going to upgrade to a paid service.

Another alternative is to investigate how to just create a file and load it into a local Excel-sheet or similar. The sensor data is simple – number of motions per hour, date and time.

The next step is to find sensor and loudspeaker for outdoor conditions. I also need to find out how to set up a solution for sound in Python (or do I have to convert to Circuit Python?).

The current solution/functions

- One time activity - Setup Dashboard and data feed in Adafruit cloud service
- Program summary
 - Login to WiFi
 - Create client for connecting/sending data to Adafruit data feed
 - Execute a loop checking sensor for motions
 - If motions
 - add one to summary of motions
 - set Led lamp on Raspberry Pico on
 - If new hour
 - Enable Adafruit client
 - Send summary motions to Adafruit
 - Disable Adafruit client
 - If no motions
 - Disable Led lamp

Next steps

- Solve sound solution
 - This is the core of this solution. I want to frighten the deer's with some scary sound.
- Find outdoor equipment's

The code

Current solution code

Below is an overview of included functions in the program used for the current solution:

- Main.py – Main program defining variables, functions, execution. Done in Python 3.
- MQTTClient.py – Program for describe in detail all functions for using the MQTT client/transport protocol communicating with Adafruit dashboard. Done in Python 3.
- Circljud.py – For testing sound with direct connected loudspeaker. Done in Circuit Python.
- Mixer.py – For testing sound with amplifier and loudspeaker. Done in Circuit Python.

Main.py

Initialization

Use code in other modules needed for the program

```
import time          # Allows use of time.sleep() for delays

from mqtt import MQTTClient # For use of MQTT protocol to talk to Adafruit IO – MQTTClient
program contains all functions needed to define actions for the MQTT-client

import machine       # Interfaces with hardware components

import micropython   # Needed to run any MicroPython code

from machine import Pin # Define pin
```

Define variables like no of motions and current/prev hour. Initiate the variables in start

Define Pin for setting LED lamp

Set user and password for accessing WiFi

Set variables for access and where to send motion data to Adafruit dashboard

FUNCTIONS

Check if sensor indication any motions

Check current hour against prev hour

Set true if new hour

If motion

Add 1 to number of motions per hour

Set led lamp on

Set false

If no motion

Set led lamp off

Set false

Function to connect Pico to the WiFi

Check if already connected

If not connected

Connect to Wifi

Confirm with IP address if connected

Function to publish number of detection to Adafruit IO MQTT server at fixed interval (hour)

Send the total numbers of motions to Adafruit using the predefined client

Try WiFi Connection

Check of connection to Wifi

Do above connect function

Check if failed

If failed print error text

Use the MQTT protocol to connect to Adafruit IO

Connect to Adafruit with prior defined client

Check if failed

If failed print error text

Program loop with main functions

Repeat this loop forever until user interrupting

Do check motions as above

If new hour

Do Connect to wifi

Do `connect_adafruit()`

Do Send the message to adafruit

disconnect from adafruit

KeyboardInterrupt:

If user stop the program

[mqtt.py](#)

The program contains of a number of detailed functions for communicate via the MQTT-client/protocol to selected server (Adafruit in this case). The module is generic and the main application (main.py) are requesting each function with required parameters. The functions used in `mqtt.py` are connect, disconnect and publish.

[circljud.py](#)

Installed CircuitPython just to check if I get any sound. Did get some “noice” but no sound as used .mp3 file. Code as attachment. This is not implemented since I did not manage to use both Python 3 and CircuitPython.

[mixer.py](#)

Installed CircuitPython just to check if any sound. Got no sound at all. Code as attachment. I did not investigate what went wrong since did not manage to use both Python 3 and CircuitPython.

[Transmitting the data / connectivity](#)

The data sent to presentation dashboard is managed by below steps:

1. login to WiFi with user-id and password
2. Define a MQTT client with user keys as defined in Adafruit.
 - a. MQTT client is an all-round client that runs an MQTT library. In this case sending messages and acts as publisher
3. Loop until user interruption
 - a. When new hour send a message to Adafruit Feed which is part of a Dashboard. The data is only number of motions per last hour
 - i. Logon to WiFi and get internet access (if not connected)
 - ii. Connect MQTT-client to Adafruit
 - iii. Send message – number of motions - via MQTT-client to Adafruit
 - iv. Disconnect MQTT-Client

The package contains just feed-name (defined in Adafruit as to be presented in dashboard) and number of motions. The package is sent once per hour.

Reusing my private WiFi to get to internet and MQTT protocol which seems to be “standard” for IOT solutions.

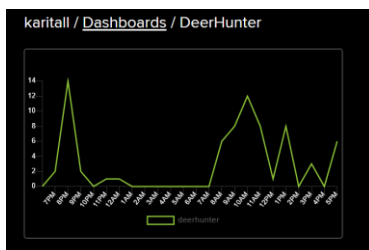
I have not yet looked into a battery solution. The testing of motion sensor loop is what I guess the primary energy user.

Presenting the data

The data presentation is just a simple line diagram showing number of motions per hour. In future solution I will import the file and just make a diagram in e.g. Excel to see time of year when deers/other animals are visiting the garden. Will be used for adding some other physical actions to keep animal away from the garden.

Download of Adafruit motion data

It is very easy to download a file with data from Adafruit. It can be of format .json or .csv and when in “production” mode it could be a routine e.g. once a week. The volume of data is small and the solution could also be changed that hours with 0 motions data not will be sent.



Finalizing the design

The vision and strategy for this project are fulfilled when it comes to objective 1 and 2.

I have learned about Microcomputer Raspberry Pie Pico W, Sensors, Thonny, Python 3, Adafruit as cloud solution, communications and also provided and running a first test environment.

I have put in some print statements in the program code just to see the flow.

```
motion detected
Antal rörelser 3
Timme nu 15 Timme då 14 tid (2023, 7, 2, 15, 0, 0, 6, 183) Antal 3
Antal rörelser sätter true 3

Connected on 192.168.1.163
Publishing on Adafruit: 3 Nya ack 0
DONE ADAFRUIT 3
```

Presentation of GUI is showed above.

Had some challenges in python programming when it came to use of variables and calling functions. Python is totally new for me.

Had good guidance using the components from the roadmap document in this course. It was pretty easy to make a program using the python code provided. Had some Python challenges as above.

I was not successful in installing the IDE Visual Studio code. Have to look into this further on.

Main source of support has been as always – Google. There seems to be solutions/answers for “everything” in the “competence space”.

Next step will be investigating objective 3 to find sensors and setup for outdoor solution.

IOT is interesting and hope to come up with more solutions combining Mobile Apps, Windows forms/WFP, Google home, automated home solutions and solutions based on microcomputers, sensors and python.