2020 年臺灣國際科學展覽會
優勝作品專輯

作品編號　200018

參展科別　環境工程

作品名稱　Moly.net

得獎獎項　大會獎：三等獎

國　　家　Hungary

就讀學校　Krúdy Gyula High School

作者姓名　Miklos Zsigo

關鍵詞　moth, AI, agriculture

作者照片

Motivation

Pests can ruin every healthy tree's crop within 10-15 days. This huge loss can only be detected at harvest season. If the pests have already got inside the fruits or the fruits had fallen down prior to the time of harvest. If pest-control is unconscious, 50-75% loss may occur.

In Szabolcs-Szatmár-Bereg county the most common fruit production is apple. Codling moth can cause significant damage to the crops. Codling moths damage the apple directly and they are expected during the whole year. If there are signs of any nuisance we are not able to protect the crops anymore, because in this case they protect themselves from chemicals.

The most important part of protection against pests is prevention. Protection can be successful thanks to insects killers and expert's prognosis. Pheromone traps developed against wrong-doers are very practical. They are easy to use, but the problem is that the manufacturer has to check these traps repeatedly to find out about the infection, and when one has to spray the trees with the right chemicals.

Precision Agriculture

Prevention is the most important component of pest control. The control is effective when it comes to prevent the damage. The automated insect traps can give a professional prevention. If we defend our trees with the information which is based on the prevention, we could be more succesful.

These pheromone traps are very practical in the harvest season. Easy to use them, but if the manufacturers want a good prediction from these traps they have to check them every day. In order to be able to protect properly against pests, we need to collect relevant data in the field. The parameters that are important for each plant are mostly different. In my case the codling moth was the main target, so I have read several professional documentations to create my own trap with my own control. Therefore I had to make from this I had to make a summary what specifications I have to work with.

Modelling of biological processes

The success of moth control is highly dependent on forecasting. Using the pheromone trap results, and the calculated heat sum it is possible to obtain very accurate results. There is a strong connection between the moth evolution and the daily average temperature. These data can be used in defence Based on the professional documentations, I could find every heat sum what I need. I came to the conclusion that, depending on the temperature, the birth of the larvae may begin before, after or at the same time as the harvest. Theese data can give a good prevention about the harvest time.

The codling moth granulovirus can only be used against the codling moth larvae. To prevent fruit damage, one has to always schedule the first treatment for the first generation of larvae. The timing of the first treatment can be determined by pheromone trapping and heat sum calculation. The heat sum calculation should start from the first day when the apple tree is drawn (pheromone traps will take the males) and the temperature measured at 9 o'clock will exceed 16-17 ° C. From this day, the daily average temperatures are subtracted from 10 °C and the resulting values are summed daily. When the calculated active heat sum reaches 85 ° C, the first Madex treatment should be performed, the prescribed dose is 100 ml / ha, applied to a volume of 500-1500 l / ha. Treatments should be repeated at 7-8 in case of intense sunshine and every 10-12 days in other cases. Excellent resistance to breakage.

CODLING MOTH AMOUNT OF HEAT

BIOFIX POINT (FIRST APPEARANCE)

TEMPERATURE LIMITS: 10-31.1 ° C

EGG LAYING: 55-60 ° C

EGG TIP: 85-90 ° C

LARVAE: 150 ° C

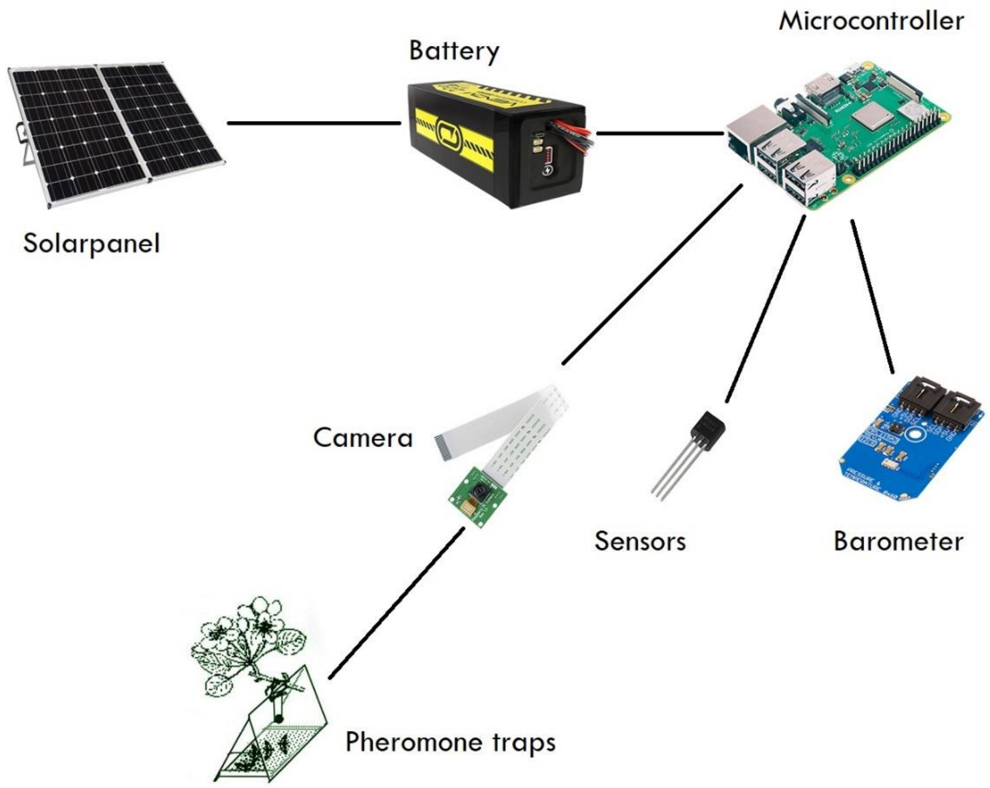
Gábor Moravszky, Managing Director of Demo-Trade Kft. provided these data (the largest agricultural consultant in the region).

Solution

How is it possible to keep track about all of the pests inside the fruit? – I developed a system which can monitor the insect traps. In this trap there is a high quialty camera and a monitoring microcontroller (in this case a Raspberry Pi 3). This system takes pictures inside from the traps either automatically or remotely controlled. This pictures are uploaded to an online database, which provides an interface for the manufacturers to check these pictures. On this interface they can check the data and the pictures. They can check it back to months ago, so they can see the difference between the pictures.

Inside the traps there is the pheromone, and the glue. We can change the pheromone inside the traps if we have another type of pests. We have to put the camera over the glue. Inside the new trap, we have to put in a modem to make a connection with a database, a battery, and a solar panel to charge the battery.

The traps are placed in the required number and in the well-chosen location on the basis of the manufacturer's proposal. The high-quality picture taken by the camera is evaluated by the image recognition software. The software can count how many insects are shown in the picture. The data are stored in the online database and displayed in the desired format (eg graph), which contains data of the previous days or weeks, and it even could compare these images or data for years. Every manufacturer would have his/her own page, and database, where the data could be stored.



How many stations do the manufacturer need for a particular area?

How many stations are to be placed in an area, will always depend on the size of the site and how cautious a farmer is. Based on the experiences, this is decided by the manufacturer, or by the experts. If other pest species are found, the traps can be modified according to the appropriate software (image recognition intelligence), but with the same specifications.

Process of the development

The whole process was done alone, but sometimes I asked for help from professionals both in the hardware-software part of the device and in the use of biology models. The software was written in Python 3 and Python 2.7. The test of the device started in the spring season.

Structure of the device

The brain of the device is a Raspberry Pi 3, which is the most practical microcontroller what I found on the market.

The camera is a LogiTech 270 web camera, which is not too expensive, and It can record high quality photos. (720p)

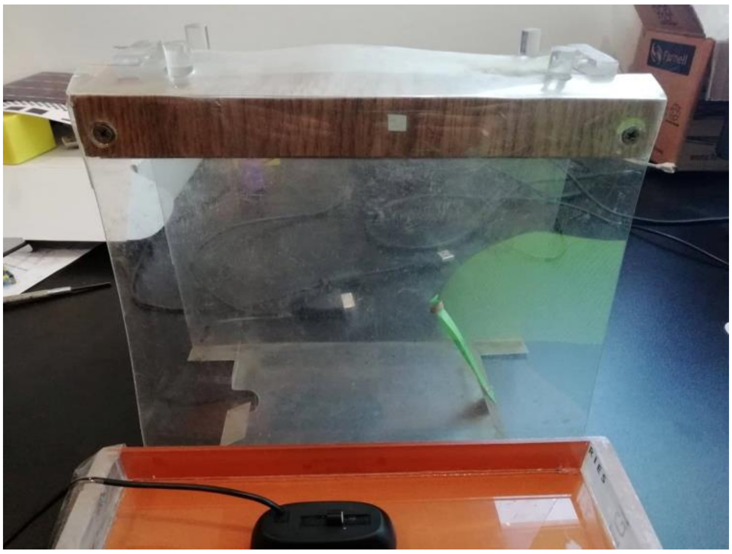
Currently a homemade power bank for the power supply of the device (13000 mAh), which perfectly satisfies the power supply of the device, but it sometimes needs to charing.

The device has a DS18B20 digital temperature sensor which is responsible for temperature measurement.

A web camera built into the protective cover before it is ready for installation.

Integrated in the weather station (Raspberry Pi, GPS Module, Powerbank, Camera USB Connector).

The dismantled camera in front of the trap.



The working principle

Raspberry Pi sometimes resets in energy saving mode to run only programs that need to collect data. The tool runs more than one program at the same time, and it knows what tasks you need to perform at a time.

With the help of a specialist and by studying the literature, I have found the dates that are suitable for data collection. The photo is made once a day at 9 am, as the light conditions are already good and moths are on their way, and the device runs the image recognition program, which counts the amount of moths and transmits it to the database. Temperature data is measured at 1 am, 7 am, 1 pm and 7 pm. According to this data, the program calculates the daily average temperature that is transferred to the database. The database is based on mySQL, so it can easily linked to any Dashboard software to display data.

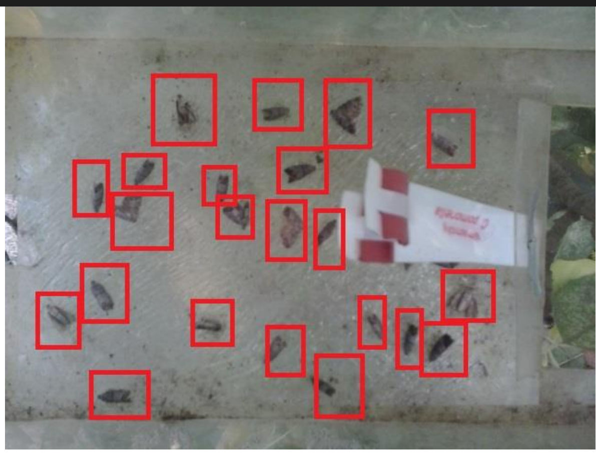
The software

The code responsible for photography is a short, compact software written in Python 3 that instantly sends the image to the evaluation program as soon as the image is finished.

```python
import cv2, time, os

from time import sleep

from os import system
from counter import proc_img

os.chdir('out/')

SleepTimeL = 0 # was 15
FrameCount = 0
FrameStop = 240

capture = None

def read_frame(port=0):
    global capture
    if capture is not None and capture.isOpened():
        capture.release()
    capture = cv2.VideoCapture(port)
    _, img = capture.read()
    return img

def save_to_csv(timestamp, result):
    file = open('results.csv', 'a')
    file.write('\"{}\",\"{}\"\n'.format(timestamp, result))
    file.close()

WAIT = int(FrameStop) * int(SleepTimeL) / 60

while(FrameCount < FrameStop):
    img = read_frame()
    timestamp = int(__import__("time").time())
    path = "{}.png".format(timestamp)
    cv2.imwrite(path, img)
    result = proc_img(path, display_result=True)
    save_to_csv(timestamp, result)
    time.sleep(SleepTimeL)
    FrameCount = FrameCount + 1

```

I used Python 2.7, OpenCV 3.4.1 for the evaluation program. The program is based on the principles of machine learning, thus generating knowledge from the experience gained. I would teach the tool how a moth looks and what it should be known for. I used several tools for this. I asked for positive data from apple farmers and photographed one by one and taught me what the machine would recognize. As a negative data, I used tiny beans and coffee beans that are different from the mole because they usually get black or brown dirt into the trap. After learning the program, it was possible to connect with the photo program.

```python
import cv2 as cv
import numpy as np

def proc_img(path, display_result=False):
    kernel = np.ones((3, 3), np.uint8)
    original = cv.imread(path, cv.IMREAD_COLOR)
    cv.resize(original, dsize=(320, 240))

    img = cv.blur(original, (11, 11))
    img = cv.erode(img, kernel, iterations=2)
    img = cv.normalize(img, None, 0, 255, cv.NORM_MINMAX)

    detector = cv.SimpleBlobDetector_create()
    kpts = detector.detect(img)

    if not display_result:
        return len(kpts)

    font = cv.FONT_HERSHEY_SIMPLEX

    for i, kp in enumerate(kpts):
        size_factor = 0.6
        a = (int(kp.pt[0] - kp.size * size_factor), int(kp.pt[1] - kp.size * size_factor))
        b = (int(kp.pt[0] + kp.size * size_factor), int(kp.pt[1] + kp.size * size_factor))
        cv.rectangle(original, a, b, (33, 33, 255), 2)
        cv.putText(original, 'Target ' + str(i), (a[0], a[1] - 10), font, 0.37, (33, 33, 255), 1, cv.LINE_AA)

    cv.imshow('Process result', original)
    cv.waitKey(0)
    cv.destroyAllWindows()

    return len(kpts)
```

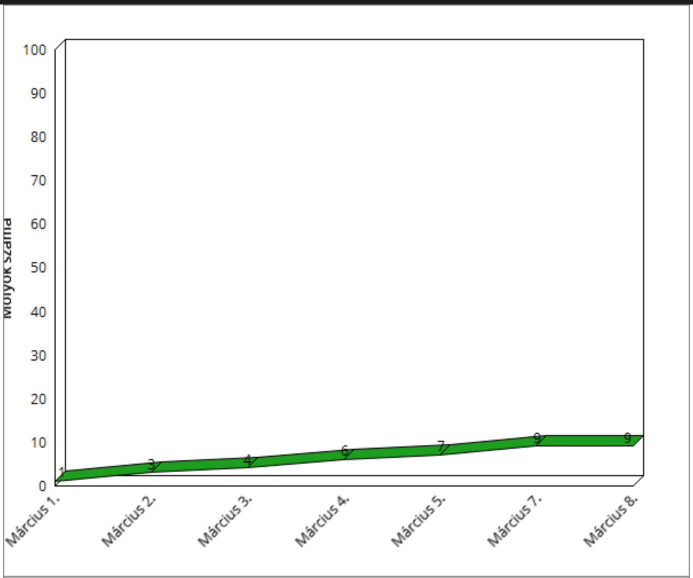


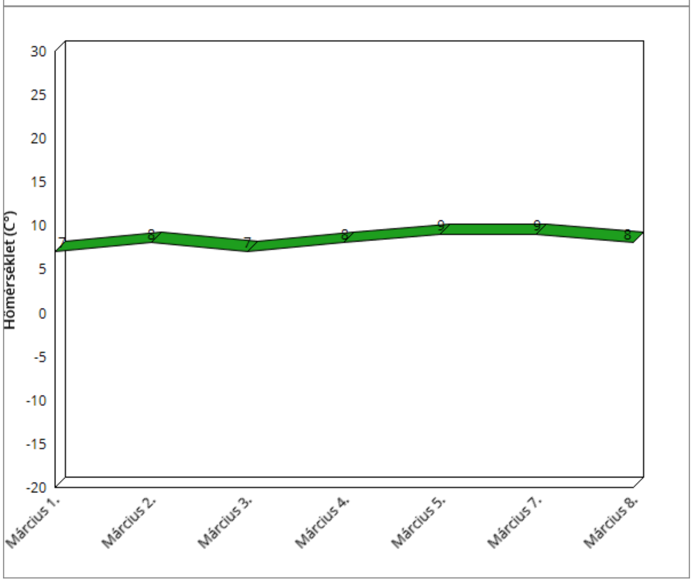
After taking the picture, the photographer saves the image between the images he wants to test and runs the software that counts the moths and sends them to a CSV file that they simply upload to the online database.

Number of moths:	22

Temperature data is also stored in a CSV file, and these data are added to the database and displayed on a Dashboard

```python
import os, time

os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

temp_sensor = '/sys/bus/w1/devices/28-01143ce432aa/w1_slave'

def temp_raw():
    f = open(temp_sensor, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = temp_raw()
    temp_output = lines[1].find('t=')
    if temp_output != -1:
        temp_string = lines[1].strip()[temp_output+2:]
        return float(temp_string) / 1000.0
    return -1

print(read_temp())
```





The connection - Narrowband IoT (NB-IoT)[1]

Narrowband IoT (NB-IoT) is a cellular technology specializing in communication between "things" that require small amounts of data, over long periods, in hard to reach places.

Narrowband IoT (NB-IoT) is a Low Power Wide Area (LPWA) technology that works virtually anywhere. Many potential "connected things" are located in remote or hard-to-reach areas, at long distances from the next cellular base station, or in shielded areas, such as deep within buildings or underground structures. When there is coverage under the aforementioned circumstance, it is often poor and requires the device transmitter to operate at high power, draining the battery. Additionally, cellular networks are not optimized for applications that only transmit small amounts of infrequent data. Furthermore, the existing cellular standards don't support the level of power saving capabilities that NB-IoT does, which makes these standards unsuitable for inexpensive devices that require battery lives of several years. NB-IoT has the ability to connect many more devices to the Internet of Things and make many new applications a reality. It is optimized for applications that need to communicate small amounts of data over long periods of time. Since it operates in licensed spectrum, it is secure and reliable, providing guaranteed quality of service. Using other cellular air interfaces for a typical NB-IoT application would be too expensive, as NB-IoT applications don't need all their capabilities. NB-IoT applications focus on low speed, robust data transfer, and an appropriate level of reliability. The low device price and long lifecycle cuts operating expenses and reduces the risk of theft.

[1] https://www.u-blox.com/en/solution/technology/narrowband-iot-nb-iot


```python
import serial, time

if __name__ == 'sender':
    print('Starting')
    s = serial.Serial()
    s.port = 'COM28'
    s.baudrate = 115200
    s.bytesize = serial.EIGHTBITS
    s.parity = serial.PARITY_NONE
    s.stopbits = serial.STOPBITS_ONE
    s.timeout = 1
    s.xonxoff = False
    s.rtscts = False
    s.dsrdtr = False
    s.writeTimeout = 2

    try:
        s.open()
    except Exception as e:
        print('error opening port:', str(e))
        exit()

    if s.isOpen():
        print('serial is open')

        s.flushInput()
        s.flushOutput()

        s.write(str.encode('AT+QIOPEN=1,0,"UDP","188.166.124.211",41239,0,1\r\n'))
        time.sleep(0.5)
        print(s.readall())


        # temperature: ; count:

def flush():
    s.flushInput()
    s.flushOutput()

def sendTemp(temp):
    flush()
    temp = 't:' + str(temp)
    msg = 'AT+QISEND=0,' + str(len(temp)) + ',' + temp + '\r\n'
    s.write(str.encode(msg))
    time.sleep(0.3)
    print(s.readall())
    flush()


def sendCount(count):
    flush()
    count = 'c:' + str(count)
    msg = 'AT+QISEND=0,' + str(len(count)) + ',' + count + '\r\n'
    s.write(str.encode(msg))
    time.sleep(0.3)
    print(s.readall())
    flush()
```

Market

I have opponents. They would like to make an automated system, and a whole international supported market, but they stopped refreshing the page since March 2019. They do not use the image recognition software, just taking the pictures from the trap and send the pictures.

I have made a market discover. I found that in Europe there are a lot of potential countries (Poland, Germany, Italy) where my system could be used. In these countries there are big marketplaces for every fruit, what I can detect.

Summary

The system I developed would be effective for farmers for economic, environmental and environmental reasons. Helps the owner to buy the right amount of pheromone, glue and spray, so you can plan ahead financially. For environmental reasons, this tool would be important because of conscious substance use, as you would know, based on the evaluated graphics, the amount of infection, so you would not use more chemicals than you need, and would buy and use the right amount. Thus, the burden on the environment would only occur to the necessary extent. It is also important for chemical manufacturers to put the right product in place at the right time, so that users know that it can be used effectively.

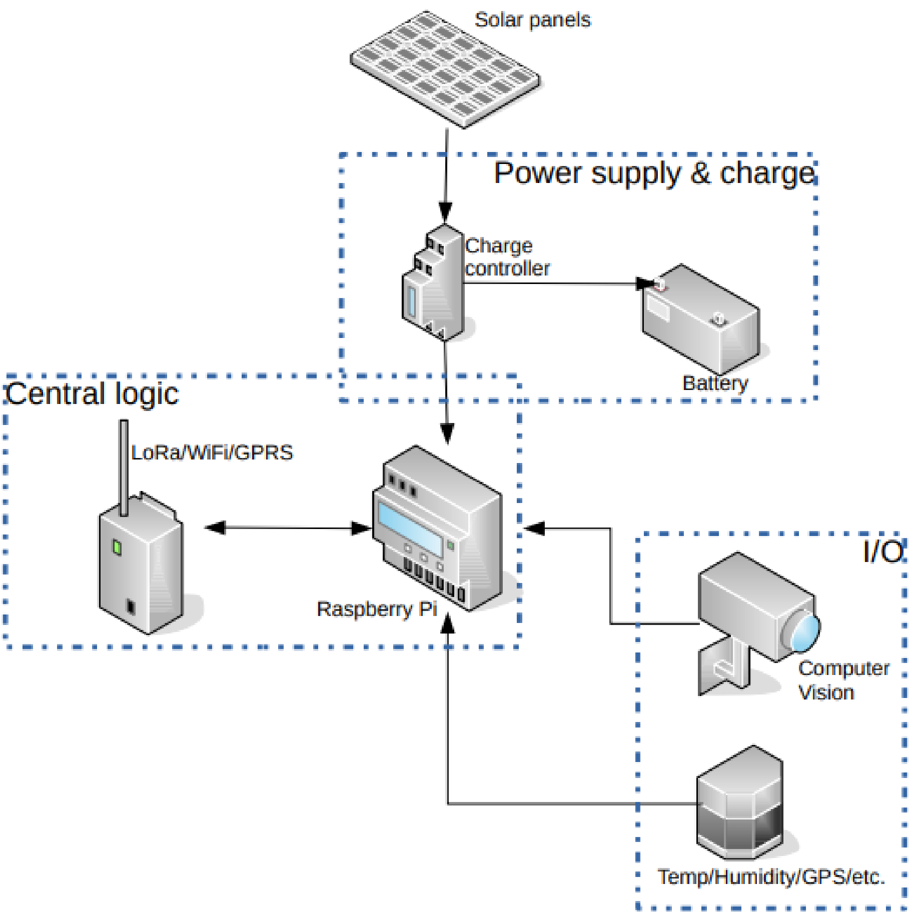
If we analyze the data of the traps in a larger orchard, we can get a picture of the spatial and temporal flow of the moth. Such an analysis can also help refine the biological model. The biological model can help every manufacturer to keep safe the orchard.

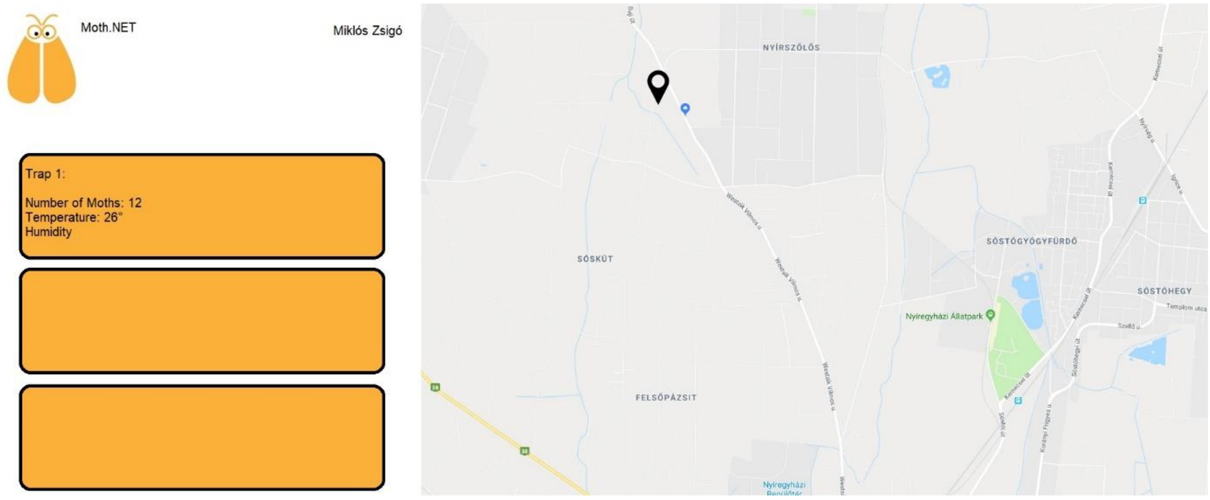
Under construction

There is a complete weather station under development that would be available to farmers as a more complex system. The station would be controlled by a Raspberry Pi 3, but several types of sensors could be placed beside the insect trap, and we would be able to analyze further data (humidity, soil moisture, etc.).

This weather station has a webpage.

The external solar system is just under testing and development, so the device will charge itself in the future.





Bibliography

- https://dea.lib.unideb.hu/dea/handle/2437/240886
- http://biocont.hu/termek/madex-100-ml-ncpgv01
- http://www.trapview.com/v2/en/
- http://phd.lib.uni-corvinus.hu/804/1/Hari_Katalin.pdf
- https://pdfs.semanticscholar.org/137c/049e7bb645071e186e71236928e22e0b5bc0.pdf
- https://www.youtube.com/watch?v=OVziqXv7g2k
- http://entnemdept.ufl.edu/creatures/fruit/tropical/bactrocera_dorsalis15.jpg
- https://www.amazon.co.uk/d/Moth-Protection/Agralan-Pheromone-Codling-Moth-Trap/B0039O632S
- https://www.researchgate.net/figure/Mean-of-codling-moth-weekly-catches-in-traps-baited-with-high-dosage-sex-pheromone-lure_fig1_264899256
- https://www.ahrt.hu/hu/intelligens-iot-megoldasok
- https://www.magyar-elektronika.hu/10005-tartalom/2114-mero-adatgyujto-rendszerek-adatkiolvasasa-lorawan-halozaton

【評語】200018

This study uses the NB-IOT technology which can take photos and measure the ambieat temperature of a farm. The goal is to monitor the moths and reduce the harm induced by the pests. The application of the IOT technology on agriculture industry is a fairy new concept. The pests monitoring by taking photos and temperature measurbment is a low cost approach which could have high potential application.