IoT Solutions for Crop Protection against Wild Animal Attacks

Stefano Giordano, Ilias Seitanidis and Mike Ojo
Department of Information Engineering,
University of Pisa
Via G. Caruso 16, 56122- Pisa, Italy
s.giordano@iet.unipi.it, i.seitanidis@studenti.unipi.it
mike.ojo@ing.unipi.it

Davide Adami
CNIT Research Unit,
Galleria Gerace 18,
56124, Pisa, Italy
davide.adami@cnit.it

Fabio Vignoli
Natech Srl,
Via Algero Rosi 46,
56100 Siena, Italy
fabiovignoli@natechescape.com

***Abstract*—Technology plays a central role in our everyday life. There has been a surge in the demand of Internet of Things (IoT) in many sectors, which has drawn significant research attention from both the academia and the industry. In the agriculture sector alone, the deployment of IoT has led to smart farming, precision agriculture, just to mention a few. This paper presents the development of Internet of Things application for crop protection to prevent animal intrusions in the crop field. A repelling and a monitoring system is provided to prevent potential damages in Agriculture, both from wild animal attacks and weather conditions.**

***Index Terms*—Ungulates repulsion, IoT in agriculture, Sensor Cloud, RIOT**

I. Introduction

Crop damage caused by animal attacks is one of the major threats in reducing the crop yield. Due to the expansion of cultivated land into previous wildlife habitat, crop raiding is becoming one of the most conflicts antagonizing human-wildlife relationships [1]. With a population of more than 200,000 wild boars *(Sus-scrofa)* and 300,000 deer in the region of Tuscany alone in Italy, being estimated to be four times more than any other region in Italy, there is an increasing damage of vineyards and farm lands that has resulted in a huge drop in wine production. According to [2], [3] annual production loss in the wine industry is estimated at 130,000 bottles of wine, which amounts to the range of 8,962,250 – 13,036,000 euros, with an annual cost to the government estimated around 2.5million euro per year. In addition to crop damages, up to 1000 road accidents are also caused by these wild animals annually.

The current methods used to counter this problem include the use of electrified welded mesh fences (usually 30cm in the ground), chemicals or organic substances and gas cannons. Other traditional methods applied by farmers include the use of Hellikites, Ballons, Shot/Gas guns, String & stone, etc. These solutions are often cruel and ineffective. They also require a vast amount of installation and maintenance cost and some of the methods have environmental pollution effect on both humans and animals [4]. On the other hand, the chemical products used to prevent these animal attacks have an application cost per hectare and their effectiveness is dependent on weather condition, as rain may cause a dilution effect. Technology assistance at various stages of agricultural processes can significantly enhance the crop yield. Sensor networks express a substantial improvement over traditional invasive methods of monitoring [5]. Our proposed method is based on an animal friendly ultrasounds generator, which does not produce physical or biological harm to the animals nor sounds audible to humans.

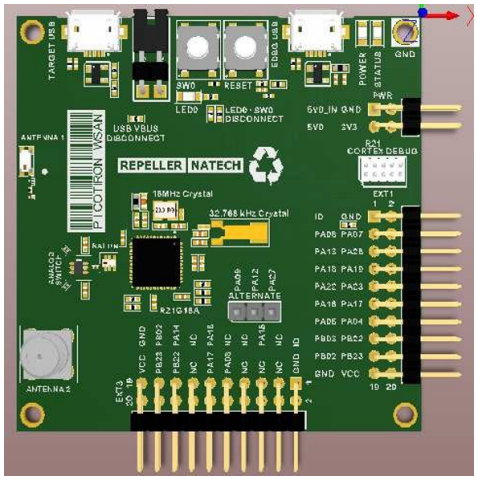
Moreover, in agriculture and especially in wine production, very small changes in the micro climate can impact the quality of the product. It is vital for agronomists to have a clear view of the meteorological conditions in a very small area of the entire territory usually characterized by very different soil characteristics. Diseases such as Bunch-rot "Botrytis cinerea" and Peronospora in vineyards [6] can be prevented by an hourly monitoring of the plant and by providing timely based required treatment to the plant. Based on the above problems, a complete system that will both protect and monitor a vineyard is of significant importance.

In this work, we present the coordination among heterogeneous sensors and actuators interacting with the cloud to provide an enabling platform for new services in this domain. In particular in the peripheral part, we adopted wireless technologies such as 6LoWPAN, WiFi, Zigbee etc., cooperating with the data center by an advanced IoT gateway. Another important feature that we had to consider is the lifespan of the devices before deployment. As a result, we selected low energy consuming motes equipped with batteries and solar panels for energy harvesting in order to achieve this goal.

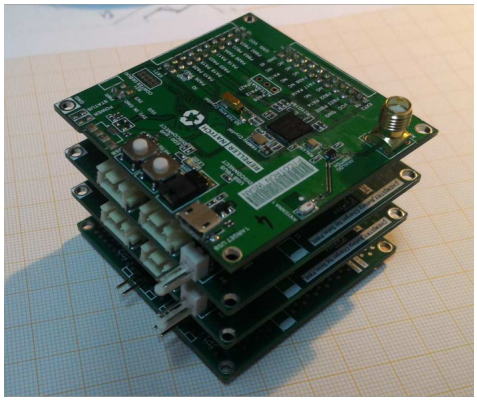
In the rest of the paper, we describe in Section II the system architecture for the ultrasound repeller device, the weather monitoring system and the back-end implementation. Section III describes the network deployment and in Section IV, we carried out a test to demonstrate the effectiveness of the architecture. In Section V, the roadmap of the project is described and finally in Section VI, we conclude our paper.

II. System Architecture

In this section, we present the system architecture for the ultrasound repeller device, the weather monitoring system and



(a) Microcontroller board



(b) Possible configurations

Fig. 1: Boards deployed

the back-end system implementation.

A. The Repeller Device

Our repeller system consists of a low power state-of-the-art Cortex ARM M0+ microprocessor which takes care of the frequency production and the networking operations as shown in figure 1. One of the key features used is the IEEE 802.15.4 standard in the unlicensed *2.4GHz* band, which allows for the transmission of small size frames (∼11 bytes in a distance of 50m with a very low power consumption). The device is using a solar panel along with LiPo batteries, which makes it an autonomous energy device able to work even in periods of partial or total darkness.

Figure 1 shows the development board that was used for this project. All the pins of the micro-controller are exposed, allowing a developer to interface sensors and actuators over digital and analog interfaces. Figure 2 shows the repeller devices enclosure with the lid open.

To improve the energy efficiency of the device, we made use of a Passive Infrared Sensor (PIR) sensor, which activates the driver responsible for the ultrasound generation and as well as the networking communication only when an animal

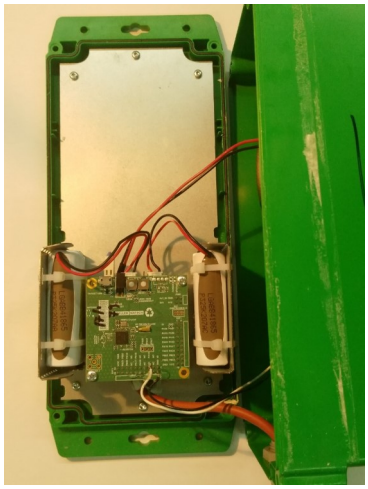


Fig. 2: The repeller device encased in an enclosure

is detected. The sound produced by the device is $120dB$ in ∼1m distance and in a wide band of $20kHz - 40kHz$ that allows us to tune the device according to the animal that is desired to be repelled. To transmit, process and store the information retrieved by the device, we are using a Proxy software that collects the activity of the animals from the device and forwards it to the back-end system. Each repeller device is placed in a water-tight box that is fixed on a 4m high pole. See figure 3.

As of the Operating System, we use an open source called RIOT [7], which offers a lot of features like, multi-threading, efficient network stack and memory allocation, etc. Our decision for adopting RIOT as the operating system is based on the viable characteristics it offers to the Internet of Things community. The RIOT OS was released on 2013 and it is based on the microkernel architecture which makes it perfect for Real-Time use cases. In the Networking part, the RIOT

Fig. 3: The repeller device powered by a solar panel

is using a network stack that is based on IP with support of IEEE 802.15.4, 6LoWPAN, IPv6, RPL, UDP and CoAP. As for the Programmability, it uses a C/C++ syntax with support of multiple threads and a memory passing IPC among the threads.

In our implementation, we exploited the features of RIOT OS mentioned above by adding one thread for the detection using a PIR and transmitting a multicast message to the gateway and the nodes. Another thread is used for receiving multicast messages from other nodes that detect movement from the animals to produce a barrier. All the messages exchanged are UDP messages with a numeric code that will distinguish the animal detected to produce the appropriate sound. The gateway is also transmitting the node's IPv6 addresses to the back-end system in order to keep track of the animal detected. At the lower layer, IPv6 for end-to-end communication alongside with RPL for routing traffic messages is utilized. At the DataLink layer, we use the default scheduling provided by the GNRC network stack of RIOT OS with the low energy protocol IEEE802.15.4 that is widely used for low-cost, low-rate wireless personal area networks.

B. Back-End System

We call the "back-end" a system where all the CPU intensive task processes take place. An important criterion in the IoT is the amount of data produced by the device. A traditional Relational Database System (RDBMS) is not suitable for our work because it would require multiple high-end computational machines to efficiently store and read the data, write policies on the tables' rows in order to avoid deadlocks. On the contrary, the usage of a NoSQL database would provide a more expandable solution at a much lower cost. In addition, the lack of a predefined format allows us to modify the structure of the data that will be stored in the database on demand and only at the level of the application.

With more than 255 different NoSQL Databases, the need to select the appropriate database is of paramount importance. Since our devices are producing data over time, a Time-Series Database is selected as a short period storage and a column-family database as a long period storage. The short period database used is the OpenTSDB, a Time-Series database that provides a REST API for interaction and a Graphical representation utility tool. OpenTSDB can store data for long term in many different databases. As a result of attaining high performance computing analysis to the data collected, we chose the HBase column-family database.

The HBase offers a wide variety of features such as high data compression, block-cache and bloom filters for real-time queries. Furthermore, HBase operates perfectly with the High Performance Distributed platform i.e. Hadoop. Hadoop is a distributed filesystem, with a wide variety of tools, like a map-reduce API. Apart from the storage and representation tools that our back-end system provides, there is also an email and push notification system in order to inform our users.

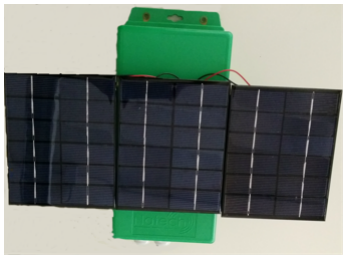


Fig. 4: Weather Monitoring System

C. Weather Monitoring System

The monitoring system provides a real-time and historic data regarding the weather conditions on their territory to the farmers. Our weather monitoring system consists of two parts, the device and the back-end. The device is a solar powered ESP-8266 Ardunio based board connected to various sensors, like temperature and humidity sensors, wind speed and direction sensors, rain sensors, air pollution sensor etc., as shown in figure 4. The device communicates over Wi-Fi to the back-end system. The back-end consists of the database system described in Section II-B, an API for the user interaction and a computational system that is responsible for making weather forecasts and also estimate the probability of attack of pests. The device transmits weather data in a periodic or on-demand way, helping us create a "database" of historic data regarding the micro-climate of the territory. It provides a real-time knowledge of the weather conditions to the farmers helping them to determine the most preferable treatments to the crops. In addition, if the weather data collected show any kind of hazard for the crops, an email notification is sent to alert the farmer.

III. Network Deployment

The deployment consists of low-power solar repeller devices equipped with a driver for amplifying the sound and a speaker. These devices are running on RIOT-OS and are interconnected in a full-mesh 6LowPAN network. The usage of the 6LowPAN along with the Cortex ARM M0+ MCU allows us to have long periods before maintenance is required. The activity is being transmitted from the repeller devices to the 6LowPAN gateway whenever movement is being detected from the onboard PIR sensor. The gateway consists of a Raspberry $Pi2$ $B+$ communicating over a serial interface with a low power microcontroller, embedded with IEEE 802.15.4 capabilities, and it is used to receive data from the repeller devices. On the gateway, a server software is responsible for receiving the data from the repeller devices, format them in a predefined format (e.g. JSON, XML, etc) and transmit them to both the database system and the presentation Graphical User Interface system. Both the database and the presentation systems are Time Series based tools, which means that the storage and the presentation of the data is based on time.

The backhauling of the gateway can be obtained reaching a wired network i.e., using Ethernet or it can be reached with the use of Wifi. The gateway can also be equipped by 4G/3G

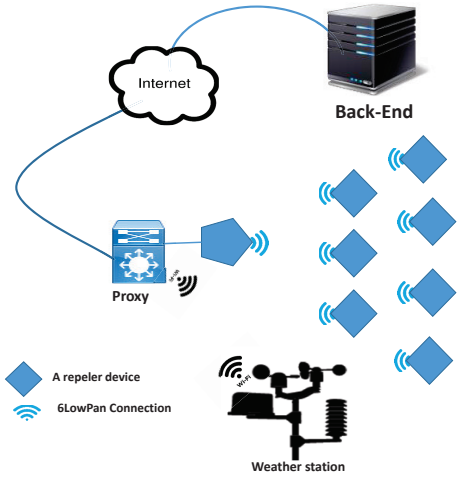


Fig. 5: Network Deployment Architecture

Fig. 6: Range testing

modules where the coverage of these networks are present. The weather monitoring system is deployed alongside with the repeller devices. These devices are equipped with a powerful but low-power microcontroller that transmits weather data to the WiFi gateway. A process at the gateway is responsible for collecting the weather data both on a defined period of 10-20 minutes and on demand. Except from collecting and transmitting the weather data to the presentation and storing system, the gateway is also responsible for alerting the user. Figure 5 shows the network deployment architecture of the whole system.

TABLE I: The IP addresses of the Repeller Devices

Device ID on Map	IPv6 Address
A	2001:fde::4b00:0615:a5cf
B	2001:fde::4b00:0615:a5e9
C	2001:fde::4b00:0615:a635
D	2001:fde::4b00:0615:a60c
E	2001:fde::4b00:0615:a5df
F	2001:fde::4b00:0615:a55c
G	2001:fde::4b00:0615:a616
GW	2001:fde::4b00:0615:a5c0

IV. TESTING

As depicted in figure 6, the devices in this use case are located close to a building and the activity is transmitted over a leased HDSL line to the cloud. The sensors are uniformly randomly distributed in a distance of at least 50 meters away from the gateway device. This is important to have an understanding about the communication range of our deployed board. The gateway device is located at the exterior of the building at a height of 4 meters for maximum performance and it is connected over WiFi to the gateway HDSL capable router that is located inside the building.

During our testing period, we carried out some network performance tests. We deployed 7 repeller devices running the RIOT Operating System. Table I shows the IP addresses of the Repeller Devices and the gateway. During this test we wanted to test and analyze the maximum distance that a IEEE 802.15.4 can operate and the packet loss at several distances. As shown in figure 7, at a distance of up to 60 meters from the gateway, we have 100% packet delivery. At $\geq 60meters$, the packet delivery ratio decreases. The maximum distance reached before the connection was lost is 104.58 meters at Line-of-Sight (LOS). From figure 7, the yellow links from the gateway to the repeller devices indicate a very good packet delivery ratio compare to the red links. Furthermore, we were able to estimate the maximum number of devices that the 6LoWPAN network could host per gateway in order to have the maximum effectiveness.

As this implementation is critical for agriculture, we had to make sure that the repeller devices will operate for long time before maintenance will be required. Our system uses two $3.7V$ rechargeable batteries with a capacity of 4200mAh as the power source and supports charging by both power adapter and a $25W$ solar panel. The batteries were tested every four days in order to guarantee the effectiveness of the solar power charging board. Apart from this, our devices were transmitting a simple UDP message with part of the MAC address of each mote at a sustainable interval as a remote point of presence for management reasons. The length of this message was 2 bytes, and it was selected as the most efficient packet length with less possible impact to both energy consumption and network traffic.

The effectiveness of our weather station system was also tested in this trial. The weather station was deployed at 1m from the ground, in order to protect it from animal attacks and flooding, while the anemometer and the rain bucket were deployed at 2 meters from the ground. We noticed from our test that if the weather station is deployed too close to an obstacle or a wall, the data retrieved from the wind measurements (the wind speed and directions) are inaccurate and misleading. The weather station was connected over WiFi to our gateway, which was responsible to retrieve in preselected intervals measurements, format them in a predefined type, e.g. JSON and transmit them to our back-end system for storage and further analysis. For powering up our weather station system, we used a solar power board that is using 3 small

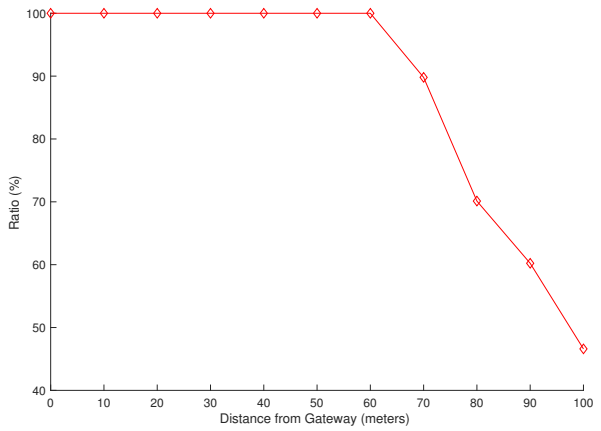


Fig. 7: Packet Delivery Ratio

solar panels capable to produce 330mA and two 3.7 Volt 4.2A batteries connected in parallel. The deployment period of the weather station was for a period of 3 months.

In addition to our hardware support, we provided two mobile applications, currently only available for the Android Operating System. There is a separate application for the weather monitoring system and one for the Repeller System. In the repeller system application, the user is able to login and view the location of the devices in a map, and the statistics of the activity of the devices. In the weather monitoring system application, the user is able to view the weather stations in his/her vicinity based on GPS functionality or search in a specific area and see the data retrieved from this weather station in real-time. In both applications, push notifications are used for either informing the user for animal activity or for changes on the weather in a selected location as an addition to the emailing system.

V. Road Map

The next steps that will be carried out in both hardware and software will be critical in order to further improve the effectiveness of our monitoring and repelling system. On the hardware part, we will enhance the monitoring system with sensors for monitoring not only the weather conditions but also the condition of the plant, like soil and moisture sensors. In addition, an image detection system based on small cameras will be used instead of the PIR. In this way, the ultrasound generation will be properly tuned on each species. On the software side, it is crucial to add and further improve a data analytic system that will process the collected data and make predictions for both the weather conditions the following days and the possible animal attacks in a specific area of the territory.

VI. Conclusions

In this paper, we presented an integrative approach in the field of Internet of Things for smart Agriculture based on low power devices and open source systems. The goal of this work is to provide a repelling and monitoring system for crop protection against animal attacks and weather conditions. In our future work, we will extend the current functionalities of our system and investigate the chance of incorporating the features of our system to other sectors.

References

[1] A. Veeramani, P. Easa, and E. Jayson, "An evaluation of crop protection methods in kerala," *J. Bombay Nat. Hist. Soc*, vol. 101, pp. 255–260, 2004.

[2] "www.telegraph.co.uk/news/worldnews/europe/italy/12105887/tuscan-wine-makers-back-cull-of-250000-wild-boar-and-deer.html."

[3] "www.reuters.com/article/us-italy-boar/italy-hunts-for-solution-to-wild-boar-emergency-iduskcn0su1jn20151105."

[4] B. Hamrick, T. Campbell, B. Higginbotham, and S. Lapidge, "Managing an invasion: effective measures to control wild pigs," 2011.

[5] A. R. Tiedemann, T. Quigley, L. White, W. Lauritzen, J. Thomas, and M. McInnis, "Electronic (fenceless) control of livestock," *US Department of Agriculture Forest Service Pacific Northwest Research Station PNW-RP-510*, 1999.

[6] C. Thomas, J. Marois, and J. English, "The effects of wind speed, temperature, and relative humidity on development of aerial mycelium and conidia of botrytis cinerea on grape." *Phytopathology*, vol. 78, no. 3, pp. 260–265, 1988.

[7] M. Lenders, P. Kietzmann, O. Hahm, H. Petersen, C. Gündoğan, E. Baccelli, K. Schleiser, T. C. Schmidt, and M. Wählisch, "Connecting the world of embedded mobiles: The riot approach to ubiquitous networking for the internet of things," *arXiv preprint arXiv:1801.02833*, 2018.