Cloud Connected Smart Birdhouse for Environmental Parameter Monitoring

Jordan Raychev, Georgi Hristov, Diyana Kinaneva, Plamen Zahariev and Emil Kyostebekov
[*] University of Ruse "Angel Kanchev", Ruse, Bulgaria
{jraychev, ghristov, dkyuchukova, pzahariev}@uni-ruse.bg and emo.kyostebekov@gmail.com

Abstract – **The Internet of Things (IoT) is an emerging technology that is changing our world by making it a smarter and a better place to live in. The idea to build a smart and connected world cannot be imagined without the IoT. Many modern houses and buildings are equipped with different sensors, which are constantly monitoring specific parameters and are reporting them to the owners or to the residents. However, we are not the only inhabitants of this planet and as dominant species, we have the obligation to take care and look after the wildlife and the natural resources of Earth. According to a recent study, which was carried out by the National Center for Scientific Research of France (CNRS), the population of farmland birds has rapidly decreased by more than 30% in the last 17 years. This trend is presenting a significant problem for the bird populations, the ecosystems and the biodiversity as whole. Finding nutrition and water is quite daunting task for some bird species, especially during cold weather or in the winter. This is one of the many situations, where the IoT technology can be utilized. With the help of the modern information and communication technologies, we are planning to build smart cloud connected birdhouses, which could help protect the endangered bird species.**

Keywords - Cloud birdhouse, IoT, wildlife preservation

I. INTRODUCTION

The population of the world has increased tremendously during the past few decades. According to several studies, the human population is growing at a rate of around 1.1% annually [1]. This expanding has a negative effect on the nature and puts a lot of pressure on the wildlife. Because of this, the list of extinct species is constantly growing and many new species are being enlisted as endangered. As endangered species, we consider all species of organisms that face the risk of partial or full extinction within the geographical ranges of their habitats. There are several widely recognized organizations, which are focused on the wildlife protection. Some of them are the World Wildlife Fund (WWF), the Wildlife Conservation Society (WCS), the International Union for Conservation of Nature (IUCN) and Greenpeace [2, 3, 4 and 5]. In Bulgaria, all endangered species are enlisted in the Red Data Book [6], which is published in three volumes – Plants and Fungi, Animals and Natural Habitats. A well-known nature park in the northeast part of Bulgaria, called Rusenski Lom [7], is focused on the preservation of many of the species, which are found in the Red Data Book. The nature park is a real paradise for many birds, which caused the region Lomovete, located around the park, to be approved as ornithological site of significant value. There are around 110 species of birds, which are nesting in the park area, but the total number of bird species exceeds 190. One of these bird species is the Stock Dove (Fig. 1). With less than 400 breeding pairs left [8], the Stock Dove is considered as one of the endangered species on the territory of Bulgaria. In the past, the population of this dove was spread over the whole territory of the country. In the last decade, the deforestation processes have caused the destruction of natural biotopes of the Stock Dove and its population was significantly reduced.

In this paper, we are going to present the impact, which the modern information and communications technologies could have on the wildlife. Our focus is specifically on bird populations and we will show some of the steps for the creation of cloud-connected smart birdhouses.

Birdhouses are widely used to help the birds find suitable places to live during their breeding session. A major drawback of the regular birdhouses is the fact that they cannot provide any information if the house is being used or not. Additional details, including the number of birds, which are visiting or living in the birdhouse, their type, species, etc., are also impossible to get, unless an onsite visits for inspection and cleaning of the birdhouse is performed. In order to resolve this issue, we are proposing to build a cloud connected smart birdhouse. This house will have all of the necessities of the regular birdhouses, but will also have additional smart features. In this way, we will have the opportunity to collect information about the birds, which are visiting the house, as well as to perform measurements on the environmental parameters of the surrounding area, including temperature, atmospheric pressure, humidity, gases, etc. All of the collected environmental parameters will be used to analyze the behavior of the birds and will guarantee that the deployed birdhouses are still habitable.

Figure 1. Drawing of the Stock Dove (Columba oenas) [9]

Section II of the paper is devoted to the design of the birdhouse, including the design of the electrical schemes. The design phase is an important one, as the components, which are providing the smart capabilities to the birdhouse, do not have to obstruct the natural life or endanger the birds. Most birdhouses are placed in areas where the internet connectivity or any wireless connectivity is not available, except for the (3G or 4G) cellular networks. In Section III, we are going to make an analysis on several wireless technologies and evaluate their advantages and disadvantages. We will then select the most suitable technology for our case. Section IV will cover the application layer of the smart birdhouse and more specifically everything related to the collection, storage and visualization of information. Section V will conclude the paper and we will make some final remarks.

II. DESIGN OF THE BIRDHOUSE AND THE ELECTRICAL SCHEMATICS

As already stated in the previous section, this part of the paper will be devoted entirely to the design of the birdhouse and will show the electrical schemes, which were used to describe the links between the components that are providing the smart capabilities to the birdhouse.

There is abundance of commercially available designs of wooden birdhouses, but most of them do not meet our requirements. The design of the house, as already stated in the previous section, does not have to obstruct the natural life of the birds. Due to this reason, we have decided to build our own prototype of the birdhouse.

The first stage of the birdhouse design process is to hide and protect all of the electronic components, so that the birds cannot reach them. For that reason, we have decided to create a hidden compartment underneath the house, which will house all electronic components. Another important aspect of the birdhouse design process is to provide the possibility to detect the presence or absence of birds. This will be achieved using an infrared (IR) sensor, which is going to be placed inside the house.

The second part of the design process is planning and choosing electronic components, which are going to provide the smart capabilities to the house. The most important step in this phase is the selection of suitable microcontroller. The most renowned and most widely used in similar applications microcontrollers are probably the Arduino family of microcontrollers. The Arduino boards come in variety of shapes, models and capabilities. Our main requirement for the microcontroller is to be relatively small, but at the same time powerful enough, so that it can power all other electronic components, which will be used. Another important requirement for the microcontroller is its power consumption. As we are planning to deploy the birdhouses in a remote location, where electricity is not available, we have to power all electronic components with batteries. In order to maximize the lifetime of the whole system, we are also planning to install a solar panel on the rooftop of the house. The microcontroller that meets the aforementioned requirements is the Arduino Nano microcontroller board and more specifically its 3.3V version.

The measurements of the environmental parameters will be made using a BME680 sensor board, which can measure temperature, humidity and barometric pressure. In addition to these parameters, the BME680 contains also a small metal-oxide (MOX) sensor. The heated metal-oxide changes resistance based on the volatile organic compounds (VOC) in the air, so it can be used to detect gasses & alcohols such as ethanol, alcohol and carbon monoxide. The VOC detection capabilities of the sensor will provide us with the ability to collect information about the forming of gasses, which is a sign that the birdhouse is contaminated and requires on-site visit for cleaning. The BME680 sensor is has both I^2C and SPI interfaces, which makes it suitable not only for Arduino boards, but also for other microcontrollers.

Additional sensors and electronic components will also be included in the birdhouse. An infrared sensor will allow us to sense the presence or absence of birds in the house. We will also use a solar panel, which will maximize the lifetime of the system. Li-Po batteries will be used to power the whole system and last, but not least, a LoRa transceiver will provide wireless capabilities to the smart birdhouse and connect it to the cloud. The electrical diagram for the birdhouse components is shown in Fig. 2.

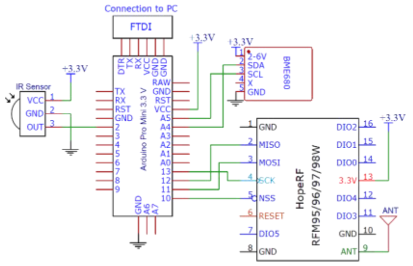

Figure 2. Logical representation and links between the electrical components of the smart birdhouse

The electrical schematic, shown in Fig. 2, is a logical representation of the electrical components and the connections between them. Each of the components is connected to the microcontroller through a specific communication interface. The BME680 sensor utilizes the I^2C interface, while the LoRa transceiver (RFM 95/96/97/98W) uses the SPI communication protocol. The IR sensor on the other hand is a simple device with a single output pin that could be easily connected to any of the digital input pins on the microcontroller. Another part of the electrical design process is the development of the battery management system BMS. The BMS includes a solar panel, constant current (CC) and constant voltage (CV) linear charger and a Li-Po battery (Fig. 3).

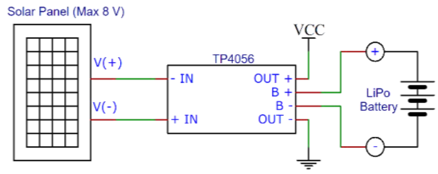

Figure 3. Electrical diagram of the smart birdhouse battery management system

The output of the solar panel is fed to the input of the constant current/voltage linear charger. The typical input voltage in the 4-8 V range. The output of the charger is used to power the electronics of the smart birdhouse. A Li-Po battery pack is used as redundant power source during nighttime or in bad weather. The typical output voltage of the charger is around 4.2 V.

III. COMPARATIVE ANALYSIS OF THE AVAILABLE WIRELESS TECHNOLOGIES

There are plethora of wireless standards and technologies that exists nowadays. Each of these technologies is characterized with a trade-off between power consumption, processing capabilities, coverage and bandwidth [10]. Many of the available technologies use the TCP/IP protocol suit and provide reliable data delivery. Other technologies are simpler and are characterized with low power consumption and longer range. The Internet of Things, as technology, has different aspects and depending on the application, the appropriate wireless technology needs to be selected.

The wireless communication protocols from the IEEE 802.15 family, including Bluetooth, UWB (Ultra-wide band) and ZigBee, are renowned for their low power, but short communications range. Bluetooth is a wireless standard, typically used for communication between devices that are in close range. To meet the needs of IoT, Bluetooth has evolved to a new version called BLE (Bluetooth Low Energy) [11]. BLE has approximately the same communications range as its predecessors, but is a reduced power consumption standard. BLE is widely used for IoT and can be found mainly in portable medical, healthcare and lifestyle devices.

ZigBee is another low power consumption standard that, in contrast to Bluetooth, was specially designed for communication purposes at longer distances. However, this is achieved as a trade-off for the lower data rates [12, 13]. ZigBee uses the carrier sense multiple access with collision avoidance (CSMA-CA) technique, which specifies how multiple devices can access the same channel using different time slots and without any interference. The devices in the ZigBee networks are connected either in a point-to-point or in a star topology. The combination of these two topologies forms a ZigBee mesh network. The mesh topology is commonly used to extend the coverage of the short-range networks. The ability for the signal to travel using multiple paths provides flexibility in the case of a failing node, but the multi-hop communication in the mesh networks can cause huge delays and leads to additional power consumption. The lower power consumption in the star networks makes them suitable for nodes with built-in batteries, which, in some cases, are not intended to be replaceable. The Wi-Fi family of standards can also be used for the Internet of Things. Compared to the other standards, Wi-Fi provides significant data rates and robustness, but usually the communication range of the standards in the family is limited to no more than 100 meters. To fix this issue, the IEEE 802.11 family of standards was extended with another addition. The improved communications range of the IEEE 802.11ah (also known as HaLow) is achieved by the lowering of the transmission frequency to 900 MHz [14]. The use of this lower frequency is not only beneficial in terms of the improved range, but also guarantees lower power consumption.

Another promising communications solution for the Internet of things is the LoRaWAN standard [15]. This multilayer standard includes the LoRa protocol at the physical layer and the LoRaWAN protocol at the data-link layer where it is used for the media access control (MAC). The main advantage of LoRaWAN, over the aforementioned standards, is the lower power consumption, especially during the transmission and the sleep cycles. Our initial experiments with the birdhouse prototype have shown power consumptions of 100 to 150 mA and around 6 μA, correspondingly in the transmission and in the sleep mode. Besides the lower power consumption, the LoRa technology has a higher communications range, compared to the other wireless technologies. That is another key advantage, because our plan is to deploy the birdhouses in remote locations in the forests of the nature park. The higher range and the lower power consumption comes at a price – the standard has lower data rates of around 50 kbps and the maximum Time-On-Air (TOA) per device per day is 30 seconds [16], which is more than insufficient for applications with demands for higher data rates.

A thorough analysis of the available wireless networking technologies showed that the LoRaWAN technology could cover our requirements and would be a perfect fit for the project. The lower power consumption of the technology will provide us with the possibility to power the birdhouse with a relatively small, but enough powerful Li-Po battery, which will last for several months in conjunction with the solar panel. In addition to that, the information for the environmental parameters, which we will measure, is just several kilobytes and we will probably never reach the maximum threshold of the LoRaWAN technology.

IV. APPLICATION LAYER AND FUNCTIONALITY OF THE CLOUD CONNECTED SMART BIRDHOUSE

This section is entirely devoted to the application layer of the cloud connected smart birdhouse. The smart capabilities of the birdhouse provide us with the ability to collect, store and analyze information about the presence or absence of birds in the living compartment of the birdhouse, as well as information about several environmental parameters, such as temperature, humidity and barometric pressure. Additionally, we will also collect information about any indoor gases, which could be a potential sign of contamination of the living compartment. Since the sampling rate for the environmental parameters does not have to be very high, we can use the LoRaWAN technology for our communication purposes. The LoRaWAN technology is a preferred choice when it comes down to applications that require low data rates, high range and low power consumption.

The functionality of the application layer of the system is divided into several specific stages. The first stage is related to the programming of the microcontroller and guarantees that the connection to the nearest

LoRaWAN gateway is established. The second stage encompasses everything related to the transport of the data to the TCP/IP network. Since the nodes broadcast LoRaWAN messages over the LoRa radio protocol, these messages are not compatible with the TCP/IP networks. This issue is resolved at the network gateway (controller). The main job of this gateway is to encapsulate the LoRaWAN messages in TCP/IP packets and to route these packets to the corresponding network server. When the data has reached the network server, we can move on to the third stage, which includes the storage and visualization of the data. This stage depends entirely on the application requirements and is strictly individual for each case.

A. *Functionality within the nodes*

There are two functional sub-stages for each node – data gathering from the sensors and encapsulation of the sensed data in LoRaWAN message. The BME680 sensor comes with a pre-build library for the Arduino boards that could be used to pull the data from the sensors. The second sub-stage is related to the encapsulation of the data and its transmission via the network gateway to the cloud. Fig. 4. Presents a code snippet that shows this functionality.

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <lmic.h>
#include <hal/hal.h>
#include "Adafruit_BME680.h"

static const ul_t DEVKEY[16] = { 0x01, ... ,0xC4 };
static const ul_t ARTKEY[16] = { 0x0F, ... ,0x01 };
static const u4_t DEVADDR = 0x12345678;

bme.setTemperatureOversampling(BME680_OS_8X);
bme.setHumidityOversampling(BME680_OS_2X);
bme.setPressureOversampling(BME680_OS_4X);
bme.setIIRFilterSize(BME680_FILTER_SIZE_3);
bme.setGasHeater(320, 150); // 320*C for 150 ms

float Temp = bme.temperature
float Pressure = bme.pressure / 100.0
float Humidity = bme.humidity
float Gas = bme.gas_resistance / 1000.0
uint16_t mydata[] = {Temp, Pressure, Humidity, Gas};

void do_send(osjob_t* j){
  LMIC_setTxData2(1, mydata, sizeof(mydata), 0);
  os_setTimedCallback(j, os_getTime()+sec2osticks(25), do_send);
}

void setup() {
  LMIC_reset();
  LMIC_setSession (0x1, DEVADDR, (uint8_t*)DEVKEY, (uint8_t*)ARTKEY);
  LMIC_setAdrMode(0);
  LMIC_setLinkCheckMode(0);
  LMIC_disableTracking ();
  LMIC_stopPingable();
  LMIC_setDrTxpow(DR_SF(7-12),14);
}

void loop() {
  do_send(&sendjob);
}
```

Figure 4. Code snippet of the cloud connected smart bird application

B. *The network gateway*

The network gateway, a.k.a. LoRaWAN concentrator, plays an important role – it takes care for the routing of the LoRaWAN messages to the TCP/IP network. Most network gateways are constructed using single-board microcomputers from the Raspberry Pi family. Fig. 5 presents the architecture of a typical LoRaWAN concentrator.

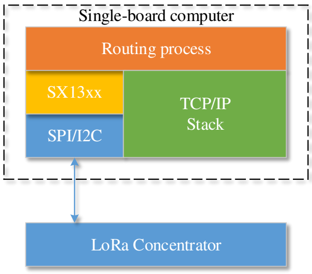

Figure 5. Architecture of a typical LoRaWAN network gateway

The LoRaWAN concentrator is a wireless network device, compatible with the LoRa standard, which listens for the LoRaWAN messages transmitted by the nodes. If any messages are received, they are encapsulated and sent over the TCP/IP network to a particular network server, which is pre-configured on the device.

There are plethora of commercial gateways [17] currently available on the market. Our objective for this research was to find a gateway that supports multiple channels (up to 8) and is yet cheap enough, so that we do not exceed our budget. Table I. summarizes some of the commercially available gateways for the European market (868 MHz).

TABLE I. COMMERCIALLY AVAILABLE LORAWAN GATEWAYS

#	Name	Devices[1]/ Channels[2]	Range, km LOS	Price, $
1	MultiConnect Conduit [18]	1K[1] / 8[2]	15	740
2	Laird Sentrius RG1 [19]	8[2]	16	260
3	IMST Lite Gateway [20]	10K[1] / 8[2]	15	220
4	LG01N / OLG01N [21]	50-300[1] / 1[2]	5-10	70
5	IC880A [22]+RP3 [23]	10K[1] / 8[2]	15	180
6	Pycom LoPy [24]	100[1] / 1[2]	22	40

There are few gateways, which cover our requirements, but we have chosen to work with the IC880A concentrator (number five in the table). At a relatively lower price, compared to some of the other gateways, this device supports up to 8 parallel channel and up to 10000 nodes, which makes the IC880A concentrator a promising solution. We have paired the concentrator with a Raspberry Pi 3 model B+ [23].

C. *Publishing of the data to the cloud*

Publishing the collected data to the cloud is the third and final stage of the process. "The cloud" is a very broad term that can encompass many things and cannot be reviewed or explained in a single paper. For the purpose of the current work, we will briefly explain only how we plan to store and visualize the data using the web and a smartphone application.

The basic flow of information is depicted in Fig. 6. As already mentioned in the previous sections, the collected data from the birdhouse is sent to a network gateway via the LoRaWAN wireless technology. This data is then routed to the public cloud servers of The Things Network (TTN) [25]. The TTN cloud and network servers only process and store the data for a short period.

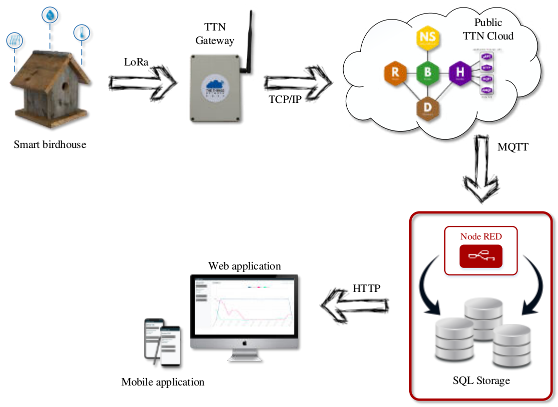

Figure 6. Information flow of the proposed system

Because our main objective is to locally store and further visualize the collected information, we have to use one of the SDKs (Software Development Kits) that is freely available. These SDKs (Go, Java, Node.js, Python, Node-RED) are comprised of two APIs (Application Programming Interfaces) – data API and application API. The Data API, in which we are mainly interested, uses the MQTT [26] (Machine-to-Machine) connectivity protocol to publish the collected information. The Application API, on the other hand, provides a way to manage all applications and devices, which are registered on the TTN platform. Since we are only interested in the information that the sensors are providing, we are going to use only the Data API. From the publically available SDKs, we have chosen to work with the Node-Red platform [27]. Node-Red is a programming tool for connecting and wiring hardware devices, APIs and online services in a quick and easy fashion, by using the provided browser-based editor. Since Node-Red is built on Node.js, we can take full advantage of its event-driven and non-blocking model, which makes it ideal to run on low-cost hardware, such as the Raspberry Pi as well as in the cloud. Fig. 7 shows how a typical node connected to the TTN network looks like and how we can quickly download and store the information on a local SQL (Structured Query Language) server.

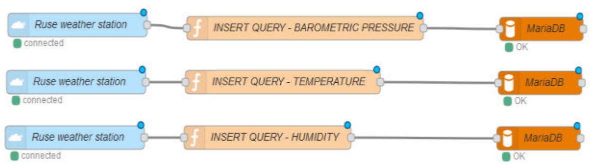

Figure 7. Typical Node-Red flows for collecting and storing information on a local SQL server

The flows depicted in Fig. 7 are composed of three parts. The first one (in blue) is representing the node that is connected to the backend of the TTN platform and guarantees that the connection to the application and its particular device is established. The node itself takes as input the information about the server, where the data has to be stored, and of course, an access key that is giving permissions to the application to register with the TTN platform (Fig. 8). The second part of the flow (colored in light orange) is representing the SQL queries, which are mapping the appropriate fields from the received messages and is encoding and sending them to a SQL server. In our case, the server is the MariaDB (dark orange node) server. From this point forward, the encoded information is stored on the local server for further processing and analysis.

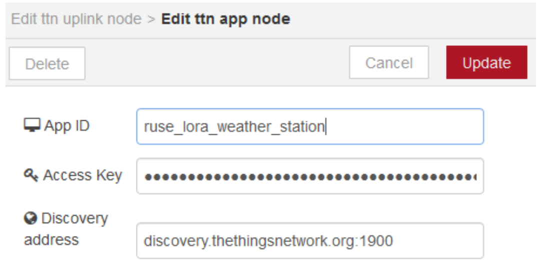

Figure 8. Configuration of an uplink TTN node in Node-Red

The visualization of the information is the last and final stage of the publishing of the available information that was collected from the smart birdhouse. Node-Red has a built-in functionality that could be utilized for this purpose. The issue with the built-in functionalities is that sometimes they do not meet the requirements of the application. Our application is intended to show relevant graphs about the measurements of the environmental parameters around the birdhouse, as well as its condition and location. Fig. 9 presents the environmental parameters measured by the smart birdhouse.

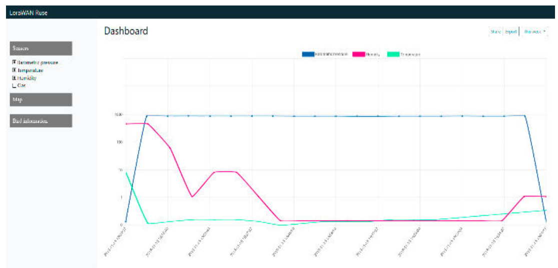

Figure 9. Graphical representation of the environmental parameters, which were measured by the smart birdhouse

The web application is hosted on our own web servers. It provides access to three sections – for the sensor data, for the map and for the birds. The first section is designed to plot the information received from the sensors for a specific interval of time. If no time interval is specified, the graphics display the available information in real-time.

The second section of the application provides basic information about the birdhouse and its location. Since the smart birdhouse will be in a known remote location, we can obtain its GPS coordinates, encode them into the web application and plot them on Google Maps (Fig. 10). Future revisions of the system might also include a GPS sensor, which will report the GPS coordinates of the smart birdhouse. The third section of the web application shows information about the number of birds visiting the house, as well as some additional functionalities that are going to be implemented in the near future. These functionalities include the measurements of the food and water supplies available in the birdhouse, as shown in Fig. 11.

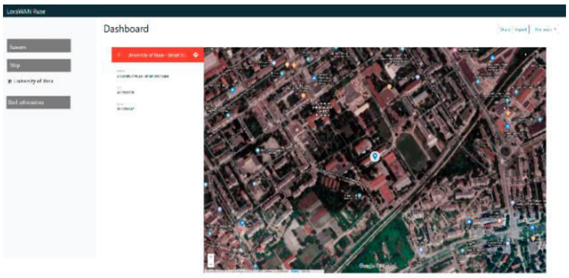

Figure 10. Map information about the smart birdhouse

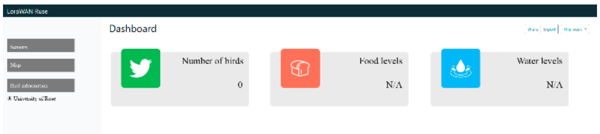

Figure 11. Measurements of number of birds visiting the house and other functionalities

V. CONLUSIONS

The population of the world has grown significantly in the last several decades. This has put a lot of pressure and has caused severe negative effects on the wildlife and their habitats. Because of that, many animal species went extinct, while others are being enlisted as endangered species. The Stock Dove (*Columba oenas*), which is found in the norther parts of Bulgaria, is one of these endangered species. The rapid deforestation and the contamination of the natural habitats of these doves has led to their inclusion in the Red Data Book for endangered species of Bulgaria.

In the paper, we have proposed a workflow for the creation of smart birdhouses. These birdhouses will be deployed in different locations and they will potentially help the endangered birds in their search for suitable living places, especially during the breading season. Several dozen regular birdhouses are already installed in different places, but they are not having any smart functions. In this way, they cannot record or transmit any information if the houses are actually being used by any birds. For this reason, we have proposed to introduce smart capabilities to the birdhouses. The smart functions of the birdhouses and the collected information will help ornithologists in their observation and analyses on the behavior of the birds and will potentially help in the preservation of the endangered species.

The ability of the smart birdhouses to track the presence or the absence of the birds and the possibility to monitor the environmental parameters will help for the better utilization of the birdhouses and will provide adequate information for the proper and timely maintenance of the living compartments within them.

ACKNOWLEDGMENTS

This paper is supported by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)", financed by the Ministry of Education and Science of Bulgaria.

REFERENCES

[1] http://www.worldometers.info/world-population/

[2] https://www.worldwildlife.org/ - The official webpage of the WWF

[3] https://www.wcs.org/ - The official webpage of the WCS

[4] https://www.iucn.org – The offical webpage of the IUCN

[5] https://www.greenpeace.org – The offical webpage of Greenpeace

[6] http://e-ecodb.bas.bg/rdb/en/

[7] http://lomea.org/ - The offical webpage of the Nature Park Rusenski Lom

[8] http://e-ecodb.bas.bg/rdb/en/vol2/Cooenas.html

[9] Image aquired form the official page of the electronic version of the Red Book of endangered spicies in Bulgaria, available at http://e-ecodb.bas.bg/rdb/bg/vol2/Cooenas.html

[10] Abbas, A., M. T. Jilani, and M. K. Khan. "Comparative Analysis of Wireless Technologies for Internet-of-Things Based Smart Farm." Sci. Int. 29 (2017): 373-378.

[11] J. Decuir, "Bluetooth 4.0: low energy," Cambridge, UK: Cambridge Silicon Radio SR plc, vol. 16, 2010.

[12] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards," Computer communications, vol. 30, pp. 1655-1695, 2007.

[13] P. Kinney, "Zigbee technology: Wireless control that simply works", Proceedings of the Communications design conference, 2003, pp 1-7

[14] S. Aust, R. V. Prasad, and I. G. Niemegeers, "IEEE 802.11 ah: Advantages in standards and further challenges for sub 1 GHz Wi-Fi," in 2012 IEEE International Conference on Communications (ICC), 2012, pp. 6885-6889.

[15] LoRa Alliance Technical Committee, LoraWAN 1.1 Specification version 1.1, October 11, 2017, available at https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf

[16] Petäjäjärvi, J., Mikhaylov, K., Pettissalo, M., Janhunen, J.,& Iinatti, J., "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage", International Journal of Distributed Sensor Networks, 2017

[17] Commercially available TTN gateways, list available at https://www.thethingsnetwork.org/docs/gateways/start/list.html

[18] MultiConnect Conduit LoRa gateway datasheet , available at https://www.multitech.com/documents/publications/data-sheets/86002198.pdf

[19] Laird Sentrius RG1 LoRa gateway datasheet, available at https://www.mouser.bg/ds/2/223/Product%20Brief%20-%20Sentrius%20RG1xx%20Series%20Gateway-1113271.pdf

[20] IMST Lite Gateway LoRa gateway datasheet, available at https://wireless-solutions.de/downloads/Radio-Modules/iC880A/iC880A_Datasheet_V1_0.pdf

[21] LG01N / OLG01N LoRa gateway datasheet, available at http://www.dragino.com/downloads/downloads/LoRa_Gateway/LG01N/Datasheet_LG01N_OLG01N.pdf

[22] IC880A LoRa Concentrator datasheet, available at https://wireless-solutions.de/downloads/Radio-Modules/iC880A/iC880A_Datasheet_V1_0.pdf

[23] Raspberry Pi 3 Model B+ datasheet, available at https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[24] https://pycom.io/product/lopy

[25] https://www.thethingsnetwork.org/

[26] http://mqtt.org/

[27] https://nodered.org/