

Design and Realization of Interconnection of Multifunctional Weighing Device with Sigfox Data Network

Martin Magdin, Miloš Valovič, Štefan Koprda, Zoltán Balogh

Department of Computer Science, Faculty of Natural Science, Constantine the Philosopher University in Nitra, Slovak Republic

Abstract

With the increasing expansion of IoT networks, people are trying to connect more and more devices to data networks. Nowadays, such devices are made possible by platforms such as Arduino and Espressif. In recent years, one of the areas of interest for IoT has been the development and maintenance of beekeeping. Although there are various commercial solutions for multifunctional hive scales (BeeSpy or Alya), they are quite expensive. In this paper, we present a hardware and software solution to this problem. From a hardware point of view, our goal was to design such a device that would be competitive, capture various data using ambient sensors, and communicate over the new SigFox data network that was introduced in Slovakia in 2018. From a software point of view, we aimed to design and implement an application for smartphones. With this application, it is possible to fully visualize all acquired sensor data. The communication between the multifunction weighing device and the user's smartphone is ensured via SMS messages.

Keywords

Arduino microcontroller, Sigfox network, LPWAN, bee multifunction weighing device.

Magdin, M., Valovič, M., Koprda, Š. and Balogh, Z. (2020) "Design and Realization of Interconnection of Multifunctional Weighing Device with Sigfox Data Network", *AGRIS on-line Papers in Economics and Informatics*, Vol. 12, No. 2, pp. 99-110. ISSN 1804-1930. DOI 10.7160/aol.2020.120209.

Introduction

The Internet of Things or IoT is a system of interconnected computational devices, machinery, objects, people or animals using a data network. The "thing" in IoT has a very general meaning. It can mean an everyday object like a car, which has a wide variety of sensors built-in; a washing machine, which reports its state on a regular schedule, etc.

IoT thus consists of intelligent devices connected to a data network. These are devices with integrated sensors gathering data, which the devices must act upon. They share their gathered data through the connection to an IoT gateway or the other nodes in the network. Shared data is sent to the cloud for analysis or are being analyzed locally. Such devices can operate without human interaction (M2M). When necessary, a human can interject and control the devices directly. This kind of interaction is labelled as M2H interaction. Sometimes the devices can act upon the data gathered from the other nodes of the network. The communication between the devices

and networking is dependent on the IoT platform in use.

In this paper, we describe the design and implementation of a multifunction weighing device connected to the Sigfox data network. This data network represents a communication framework for IoT in more than 60 countries. The multifunctional weighing device has been designed and constructed for beekeepers who maintain their hives outside built-up areas (forest areas) and transmitting data from sensors via GSM signal or Wifi is not viable. Paper is divided into several sections. The related work section describes works with similar research and implementation character. This section also describes current trends in IoT networks, in development platforms, and commercial hive weighing solutions. The next section explains our procedure in designing and implementing a multifunctional weighing machine. The discussion section represents an evaluation of the results achieved by our team.

Materials and methods

The decline in young people's interest in beekeeping, greater agglomeration and the overall approach to nature conservation (over-usage of pesticides, absence of flowering areas, forest logging) have caused a rapid decline in the world bee population in recent years. Their importance is unquestionable as they not only produce honey, propolis and other products but also serve a vital pollination function. Scientists, therefore, strive to draw attention to the need to monitor beehives so that we not have only an idea of the amount of honey produced in the combs, but also receive an early warning on diseases and pests. The following works focus on the design and implementation of bee weights that send the data to the user.

Although the first researches on beehive weight scales were carried out in 1977, these studies could not make use of the possibilities of currently available sensors and neither of wireless communication due to the lack of available technology (McLellan, 1977).

Fitzgerald (Fitzgerald et al., 2015) have addressed this issue and designed intelligent bee scales taking advantage of wireless communications technology. Using a simple strain gauge sensor to weigh the actual amount of honey in the hive, the data is sent to the base station via Zigbee. The obtained data showed that the initial platform scale has a linear output characteristic. Based on the data from the analogue-to-digital converter, they determined the accuracy of the scale to several tens of grams. In addition to weight, sensors were used to monitor temperature (MCP9700A) and humidity (808H5V5). The base station node was developed using an off-the-shelf ATmega 1281. This platform uses low power WSN and is powered by a solar panel. The firmware wakes up four times in a 24-hour cycle to send data to read from the weighing scales.

Zacepins (Zacepins et al, 2016) focused on how to help beekeepers remotely monitor bee colonies to identify the status of bee swarming. To identify this condition, they monitored 10 bee families in their hives for 4 months. They used temperature measurements to identify the condition, registering a temperature increase of 1.5-3.4 °C during the warm-up phase (last 10-20 minutes before take-off).

Edwards-Murphy (Edwards-Murphy et al., 2016) designed a monitoring system based on WSN. Data transmission from sensors (temperature, relative humidity, acceleration and measurement of gases in the atmosphere) is provided by the ZigBee 3G11

base station. The disadvantage of their proposed solution is the short range of the broadcasting station - only a few tens of meters.

Jiang (Jiang et al., 2016) also used WSN technology to detect the number of bees that flew from the alley in the morning and returned to the hive in the evening. In this way, they were able to predict the mortality rates or, conversely, the growth of the bee population in the hive. Data was transmitted via GPRS gateway to the backend platform.

Gil-Lebrero (Gil-Lebrero et al., 2017) used the IEEE 802.15.4 communication standard to retrieve data being sent from the sensors, so communication can be established via 3G/ GPRS or WiFi.

Debauche (Debauche et al., 2018) designed a technical solution based on available accurate sensors and cloud architecture. Their solution is designed not only for obtaining data from strain-gauge sensors (measuring the increasing quantity of honey) but also for monitoring the behaviour of bees. Their solution allows researchers to provide a platform to better understand and measure the various impact factors that can affect the mass extinction of bees.

Commercial solutions of selected hive scales in Slovakia

There are several different types of hive scales available on the Slovak market that communicate using different data networks. Besides weight sensors, they also use other sensors for data collection.

BeeSpy – The system for continuous monitoring of the bee condition in the hive. It is a modular system designed for all-year use. The main aim of this system is to keep the bee colony under constant surveillance. Thanks to wireless communication technologies it is possible to monitor the current state of the colony from the comfort of beekeeper's home. The weight scale is focused on measuring the weight of the beehive, from which it is possible to ascertain the stock levels in winter, and honey production in the summer. It is also possible to measure both the ambient temperature and also the temperature, relative humidity and the activity of the bees inside the hive. The scale has a metal construction and consists of a strain gauge bridge, control electronics and a connector for additional sensors. The electronic circuit is designed to allow a wireless module to be plugged in at any time. The user can choose between a Wi-Fi module and a GSM module. The scale is fitted

with a 3-digit segment display that shows the current weight. It can measure either the total weight of the hive or its relative weight. BeeSpy devices can be linked together which makes it possible to power scales from one power source and send data gathered from multiple hives to the cloud. The collected data are displayed in well-arranged graphs. The scale can weigh up to 200kg at a resolution of 50g. The price of this hive weight scale (Figure 1) is around €308 (BeeSpy, 2019).



Source: BeeSpy (2019).

Figure 1: BeeSpy hive weight scale.

VILKO - A multifunctional device allowing hive monitoring without the beekeeper needing to visit the hive location (Figure 2). The main task of the device is to process, store and transfer data via GSM connection. Since it uses the GSM network, it is necessary to provide a SIM card to the device to enable communication. Data from the device is collected via SMS messages or a website. Similarly, the commands for the device are also sent via SMS messages. The scale has one input and one output interface. The input can be connected, for example, to a door sensor, triggering an SMS message to be sent immediately, in case of the hive door getting opened. The output can be connected to various devices such as light bulbs, heaters, sirens or other parts of the device. The device is battery-powered and uses solar panels for charging. The scale can measure the hive weight, temperature, battery condition, movement of the device, or even unexpected and unwanted visitors. Additionally, it can measure humidity and ambient temperature. Users of the device can set an interval for measuring and storing the transmitted data (2 SMS messages per day maximum), set the time of automated transmission, and set up an alarm system. It is also possible to activate and deactivate alarms used to prevent unwanted handling of hives or scales. Users can enter 5 telephone numbers for sending the data and receiving commands from. The scale can measure with an accuracy of 0.1kg, allowing you to set up the hive's base weight to calculate the weight gains up to the maximum weight

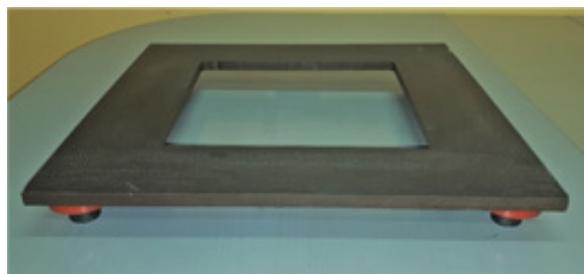
of 200kg. Cloud data storage can be switched on and off. When it is switched on the data is processed and stored on a web page where it can be viewed in graphical form or tables. The current price of the complete kit is €480 (Alya, 2019a).



Source: Alya (2019a)

Figure 2: VILKO hive weight scale.

Mellisa BL - an electronic scale intended for weighing hives and colonies (Figure 3). Its main task is to weigh (up to 150kg at 0.1kg deviation), display and record the measured values.



Source: Alya (2019b)

Figure 3: Mellisa hive weight scale.

The measured values are saved in a database. The weight and temperature are displayed when the scale is activated. The user can read the current status via the iBeehive application. Measured values are automatically transmitted to the app on your phone or tablet. The scale allows the user to set up four different times for measuring and saving values. This will automatically happen without the need for a beekeeper to switch on the scale. An external temperature sensor can be connected to the balance. To transfer data to the app, the scale connects to the phone or tablet wirelessly over Bluetooth. A common smartphone charger is used for recharging the battery of the device. A power bank is used to power the electronics. Establishing a connection, displaying the values, communicating

and transferring data to the application on the phone is only possible with the scale switched on. After a specified time, the electronics automatically switches to sleep mode and communication with the application is interrupted. Basic features include displaying the current weight, temperature, and battery charge status. The material used to make the scale is highly weather-resistant. Four sensors provide accurate weight measurement even in case of an uneven load. Android 4.3 and above or iOS 5 and above are required for the app. The scale has an opening in the centre that allows hive ventilation. Its price is €184 (Alya, 2019b).

Low power network technologies suitable for IoT

LPWAN or Low-power WAN is a wireless wide area network technology that connects low-bandwidth devices. Most LPWAN-enabled devices are battery-powered for long-range transmission. LPWAN is capable of sending packets ranging from 10 bits to 1 kilobit at a transmission rate of up to 200 kilobits per second. The range of LPWAN transmission starts at 2 kilometres to an incredible thousand kilometres, depending on devices used (Saelens et al., 2019).

In 2010 in Toulouse, France, a startup named SigFox developed a network technology by the same name. SigFox is a wireless data network that uses LPWAN principles. It is typically applied in Europe for water, gas and electricity metering devices, parking sensors in garages announcing free parking spaces, safety devices and many other sensory devices (Mekki et al., 2019). Sigfox operates in unlicensed ISM bands with frequencies lower than one gigahertz. The ISM band uses 915 MHz in the US and Canada, 868 MHz in Europe and 433 MHz in Asia. The network is built on a star topology, like all LPWAN-related technologies. Terminals connected to the base station use Binary Phase Shift Modulation (BPSK) in the ultra narrow band (100Hz). Thanks to this modulation, the interference is very low, which also decreases the power consumption, increases the sensitivity of the receiver and reduces the manufacturing cost of the antenna. This modulation also allows communication over very long distances to run smoothly compared to competing technologies. SigFox promises a signal range of up to ten kilometres in urban areas and up to forty kilometres in rural areas (Mekki et al., 2018).

Communication in the SigFox network could initially only transmit its data, as the device

was only allowed to send data to the base station - uplink. Later, however, the devices were also allowed to receive data from the base station - downlink. Downlink communication can take place only after receiving the sending message, which means the network does not support full-duplex communication, only half-duplex. SigFox also allows you to send an acknowledgement for downlink communication. Communication is limited to 140 messages with a maximum message size of 12 bytes in the uplink direction and 4 messages with 8 bytes size limit in downlink direction per day. Of course, these messages can be smaller, or even empty. However, the communication speed is limited to 100 bits per second. Since it is only possible to receive four downlink messages per day, it cannot be reliably confirmed whether each uplink message was successfully received at the base station. To mitigate this issue, SigFox uses frequency and data diversity and devices send duplicate messages, usually three at a time, to the base station. Because of this, the band is divided into 400 different channels, each 100Hz wide. There are about 40 of these channels that are not in use. The base station can receive messages from all 400 channels at once. The leaf node chooses the channel completely randomly (Razza et al., 2017).

A vital part of the network is a so-called SigFox Backend, which represents a kind of inter-point for processing data sent from the terminal. Here you can track message statistics, set up events, give privileges to other users, and set up callbacks. With callbacks, it is possible to set where the received data will be forwarded to. However, this is only possible after registering the device with ID and PAC numbers (Mekki et al., 2018).

SigFox data network currently covers half of Europe. It is being built up in the US, South America, Australia, Japan and parts of the world, representing the total of 60 countries and 5 million square kilometres. In Slovakia, this LPWAN network is provided by SigFox operator, Simplecell Slovakia. Simple cell covered 87% of the territory and 92% of the population of Slovakia (www.simplecell.sk). In Slovakia, SigFox also has competition in LoRaWan (coverage of only 16 built-up urban areas) and NB-IoT (coverage of the capital only).

LoRaWan was designed to improve battery life, network capacity and communication costs. The range of the signal depends very much on the environment, but usually, the LoRaWAN

data network covers the entire city in which the base station is located (Sinha et al., 2017), (Centenaro et al., 2016), (Hossain et al., 2018).

It is not always necessary for devices to communicate over long distances. Sometimes they only need to be able to communicate within a PAN or BAN network. ZigBee and Bluetooth Low Energy data networks are most common for this use-case.

ZigBee operates in unlicensed 915MHz bands in the US, 868MHz in Europe and 2.4GHz worldwide. It is based on the IEEE 802.15.4 standard. for PAN networks. The baud rate is 250 kbps. The range is from 30m in confined spaces to 100m in areas with direct visibility (Payne et al., 2019).

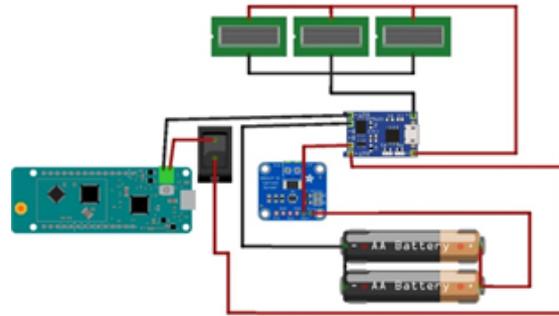
Bluetooth Low Energy (BLE) is an energy-efficient version of Bluetooth that uses other devices as base stations. These devices tend to be connected to a larger network. BLE uses FHSS modulations in the unlicensed ISM band at 2.4GHz. The maximum speed is 1Mbps, which is approximately one-third of the speed of classic Bluetooth. During transmission at this rate, it consumes about half of the Bluetooth 4.0 electrical power, which brings it to the range from 10mW to 500mW, making the BLE an excellent candidate for integration into various smart devices (Bulić et al., 2019).

Hardware design of our solution

At the heart of the system is the Arduino MKR FOX 1200 microcontroller. The following sensors are used to collect data on its environment:

1. Real-time clock,
2. vibration sensor,
3. electric current measurement sensor,
4. 2 temperature sensors,
5. strain gauge sensors for measuring weight,
6. rain gauge.

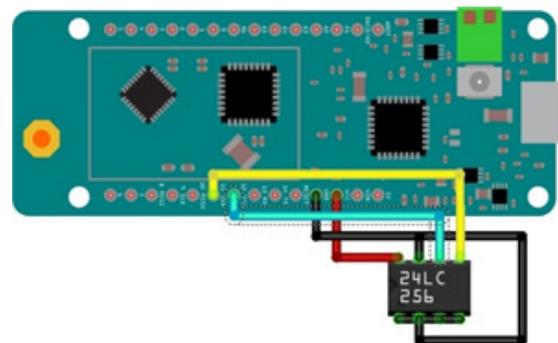
Both Arduino and the sensors are connected to the mainboard. All sensors, except the vibration sensor and rain gauge, have their ground outputs connected to the collector of the NPN transistor (to reduce energy consumption). The transistor is designed to disconnect the sensors from the power whenever they are not in use. The system is powered by a combination of two lithium-ion batteries that are connected to the charging module. The charging module (Figure 4) consists of three single crystal photovoltaic solar panels of 1.25W (to achieve energy independence during the days of sunshine).



Source: own creation

Figure 4: Connection of the power system.

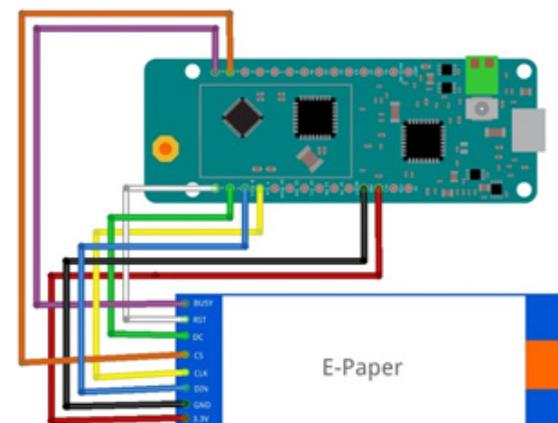
The external EEPROM memory 24LC256 is used to save the settings (Arduino MKR FOX 1200 does not contain EEPROM memory). The EEPROM (Figure 5) stores the settings such as the selected mode and weight calibration. The external memory was selected because the data still needs to be backed up even after the device loses access to electricity. The settings in the EEPROM can be changed using two buttons.



Source: own creation

Figure 5: EEPROM Connection Schema.

Waking up the device from standby and triggering the screen update procedure is done by a pressing down a dedicated button. An e-ink screen was used as a display part of the device to reduce energy consumption (Figure 6).



Source: own creation

Figure 6: E-Ink (E-Paper) Screen Connection Schema.

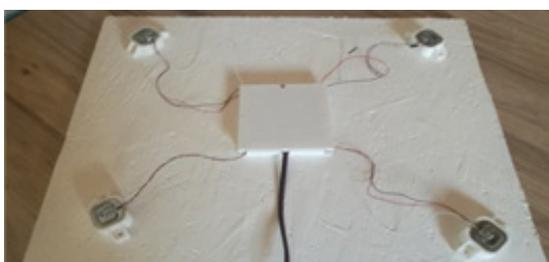
It is a monochrome electronic paper display with a 2.9-inch screen, displaying in black and white. The active area has a resolution of 296x128 pixels. When using the display, we decided to use both the full and partial image updates depending on the situation.

The casing and some other components were printed using a 3D printer for storage and protection from external (weather) influences. All parts were sealed to meet IP64 requirements. For data visualization we used an IoT platform ThingSpeak, where the obtained data are displayed in the form of graphs and data can be exported and sent through a Twilio SMS gate if necessary. The original 2dB gain antenna was changed to a 5dB gain antenna.

Selection and implementation of sensors

The proposed system is equipped with various sensors that any beekeeper will certainly make use of. The most important is a strain gauge sensor for weight measurement. Temperature and humidity information both inside and outside the hive is also of vital importance for maintaining the beehive. Another important data for the beekeeper is the amount of rainfall, for which we've added a rain gauge. If someone attempts to steal a hive or hive weight scale, we have equipped the system with a vibration sensor to give the owner a movement report of either hive or a weight scale. Last but not least the real-time clock and current measurement sensor also provide the device with important data.

Weight scale sensor - the weight scale sensor was created by assembling four separate strain gauges connected to the Wheatstone Bridge. The strain gauges are connected to a 24-bit HX711 analogue-to-digital converter with amplifier. One strain gauge can measure weight up to 50kg. Thus, in a four-sensor configuration, we can achieve weight measurements up to 200kg (Figure 7).



Source: own creation

Figure 7: Weight scale sensor.

When the pressure (force) is applied to the strain gauge, its electrical resistance changes. The HX711 can record and amplify these changes so that Arduino can measure the value. The HX711 divides

this weight into 224 (16777216) voltage levels.

Temperature and humidity sensor - DHT22 is a sensor that allows temperature measurement with accuracy to one decimal place and much more accurate relative humidity measurements than the standard DHT11. It can measure temperatures between -40°C and 80°C . It measures relative humidity from 0% to 100%. The sensors are connected to unused UART lines. An external sensor located on the side of the 3d printed case box is connected to pin 14 (TX). The sensor on the inside of the hive was connected by a 1.5m cable to the RX line, which is located on pin number 13 on the board and was led out of the box.

Rain gauge sensor - we have chosen the tipping-bucket rain gauge to be a part of our proposed multifunctional device (Figure 8).



Source: own creation

Figure 8: Rain Gauge sensor.

The rain gauge was made using 3D printing. The tipping-bucket was designed to tip over after filling up with 6 ml of water (usually tipping-bucket rain gauges tip over after filling up with 4 ml of water), thus reducing the number of wake ups of the microcontroller and slightly improving energy consumption. We placed a neodymium magnet onto the tipping-bucket that closes a switch whenever the flip occurs. The switch is connected to a pin that supports external interrupts, thus waking up the microcontroller. The microcontroller will count the tip over after each interruption. We used a formula to calculate the volume of a cylinder to know how much rain had fallen since the last message was sent. In the formula, we express the height as "v". The volume of fluid per flip is known to be 6ml, which we can substitute for the volume "V" in the formula. Our 3D model design has a radius of 5.2 cm which we use as the radius in the formula. After the calculation, we know that one flip of the tipping-buckets

represents about 0.71mm of rainfall, which means that to record a dropped 1mm, the rain gauge must flip 1.4 times. We later incorporated this knowledge into the program and we send the recalculated value in millimetres in a SigFox message.

Electric current sensor - for electric current measurement we have chosen INA219 sensor, which allows for communication with microcontroller via the I2C interface. Using I2C interface we can better ration the pins on an Arduino board. We have mounted the sensor onto the mainboard and connected a cathode of the lithium-ion battery to the Vin+ pin. To the Vin- we have connected the positive pin of the charging module so that we create a serial connection required for measuring the current. Based on which directions the electrons flow, we can determine whether the batteries are being charged or discharged. When the batteries are being charged, the electric current measurement shows negative values. Otherwise, when the batteries are being discharged, the measurement shows positive values. We show these values on the display.

Real-time clock - the board was also fitted with a DS3231 module. Although it connects to the same bus as the EEPROM memory and current measurement sensor, since it has a different address from memory and the current sensor, there is no mixing of data. The use of the module allows us to implement different kinds of modes that change time intervals for transmissions. It also plays an important role in external interrupts. With this module, we can measure time before and after the interrupt so that Arduino can switch to standby for the remaining time, during which must remain suspended.

Vibration sensor - similar devices usually contain a kind of accelerometer that communicates via the SPI or I2C bus. In our case, however, given a large number of connected sensors, such a solution would not be ideal, as it would not be able to wake Arduino up from standby state. We could only find out the state of the situation while Arduino is active, which is a very short time. That's why we chose a vibration sensor. Not only because of its simple construction but also because of its low price. We placed the sensor on the board and connected it to the microcontroller terminal that supports interrupting the processor. The ground terminal has not been connected to the transistor collector, so the sensor can operate continuously.

Creating a User Interface

The main element of the interface is the screen. On the screen, we decided to display information

on the date, power supply, microcontroller status (active mode/power saving mode), messaging mode, current hive temperature, ambient temperature and weight of the hive (Figure 9). To activate the screen, the user has to press the "WAKE" button. Upon waking up, the microcontroller changes its state from "Sleep" to "Awake". Once awoken, the user can control the interface using two buttons to change settings. The "MODE" button, which is used to switch the device to a SigFox message transmitting mode, has been connected to the analogue terminal via a 1:1 voltage divider. We used two resistors with a resistance value of 1k Ω for the voltage divider. The "TEAR" button is directly connected from the VCC to the analogue terminal.

The system contains 8 messaging modes:

15min - the message is sent every 15 minutes,

30min - the message is sent every 30 minutes,

60min - the message is sent every 60 minutes,

ECO - During winter, in the time from 6 pm to 6 am, the report will be sent every 3 hours. Outside of this range, it will be sent every hour. In other seasons, the reports will be sent every 2 hours in the range from 10 pm to 4 am and every hour outside this time range.

ECO+ - almost identical to ECO, but during winter, in the time range from 6 am to 6 pm, messages are sent every 2 hours,

DAY15 - messages are sent every 15 minutes until 10 pm; after 10 pm the system goes into standby for 8 hours,

DAY30 - messages are sent every 30 minutes until 10 pm; after 10 pm the system goes into standby for 8 hours,

DAY60 - messages are sent every 60 minutes until 10 pm; after 10 pm the system goes into standby for 8 hours.



Source: own creation

Figure 9: Front of the multifunction weighing device.

The program running on Arduino MKR FOX 1200 was written in Arduino IDE using C++ language. The program consists of a part (void setup) that happens once at boot time (the so-called

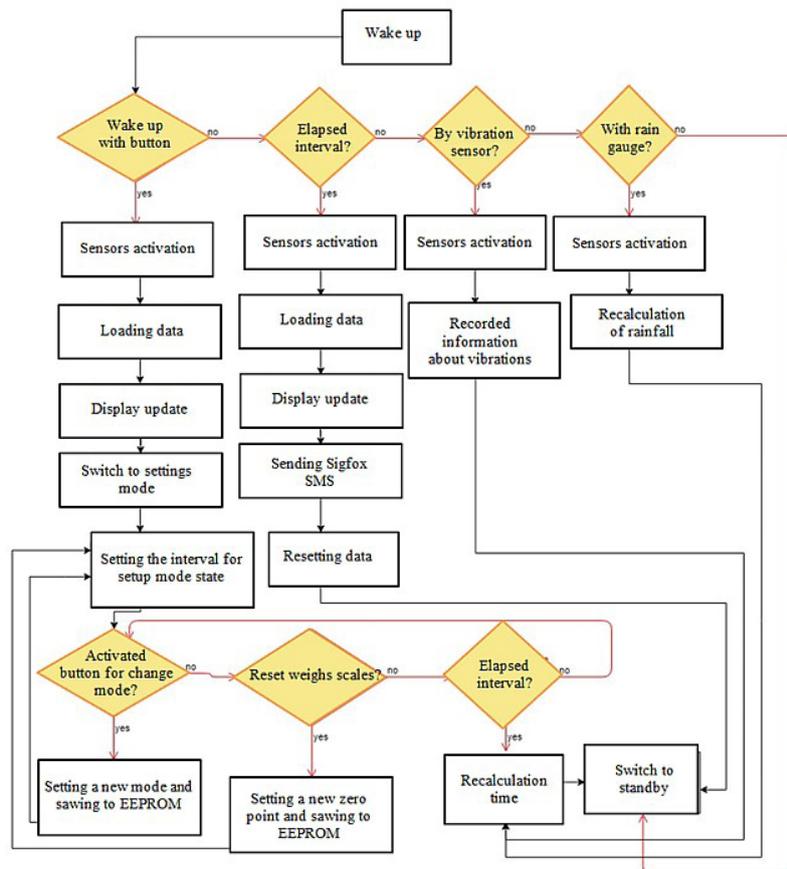
initialization part) and a part that will happen repeatedly in a loop (void loop).

The loop function (Figure 10) contains interrupt commands that are executed based on external influences (sensor data), commands to display information on the display, and commands for transmitting data to the SigFox backend. The message sent through SigFox must not exceed 12 bits. We used a float data type variable for the temperatures obtained, but we limited the sending of the temperature to sending only once in 1 hour. We have proceeded similarly for the obtained data on indoor and outdoor humidity. Both temperature and humidity are important data for a beekeeper, so they must be as accurate as possible. Even with a small change in these values, the beekeeper can predict what may be happening to the hive. For example, the elevated temperature often means the colony is infested with a disease (*Varroa destructor*), in which the bees are trying to rid themselves of the parasite by increasing their internal temperature. We have used the uint_8 data type for vibration and rainfall data as we do not need to measure negative

values and we only need values from 0 to 255. We used the float data type for the weight measured by the weight sensor because we want the beekeeper to know as accurately as possible hive's changes in weight.

Backend setup is done using the website <http://backend.sigfox.com>, entering the ID and PAC of the Arduino MKR FOX 1200 microcontroller and creating a new callback (data is sent in the uplink direction). Twilio SMS gateway (providing free SMS messaging until the user overspends the prepaid credit) is used to send the current status SMS without adding any additional hardware. To avoid having to create a dedicated application, we have used the widely used ThingSpeak IoT platform (Figure 11, left). ThingSpeak lets you visualise the following data:

1. Vibrations (first item, left panel),
2. Outdoor temperature (second item, left panel),
3. Indoor (hive) temperature (third item, left panel),



Source: own creation

Figure: 10 Initialization function flowchart .

4. The amount of rainfall (fourth item, a panel on the left).
5. Weight (first item, right panel),
6. External humidity (second item, right panel),
7. Internal (hive) humidity (second item, a panel on the right),
8. Vibration history (second item, right panel).



Source: own creation

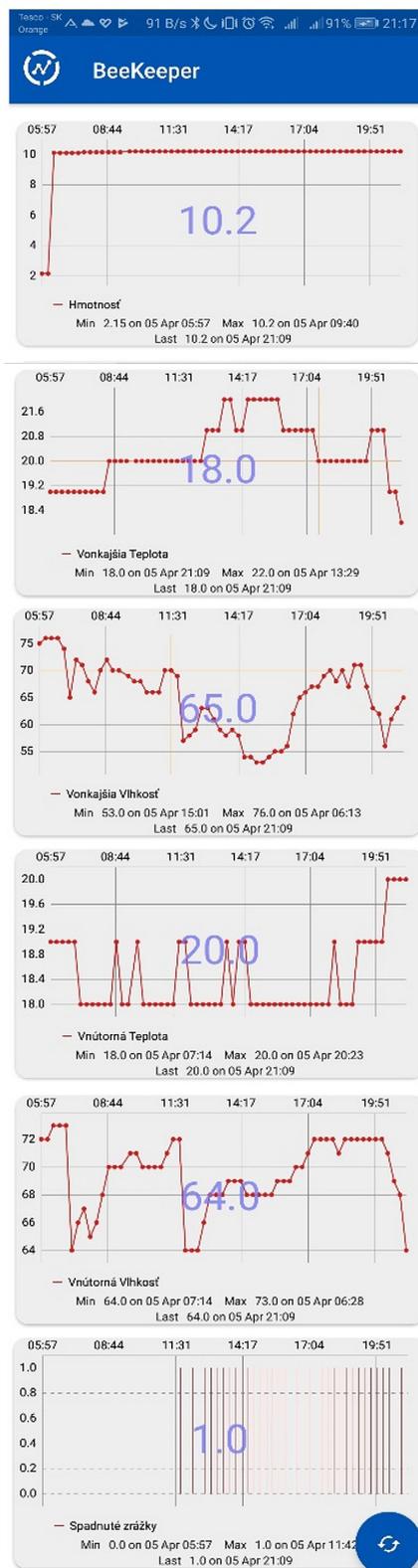


Figure 11: Data displayed by ThingSpeak (left) and ThingView (right).

We have chosen the ThingView app for Android smartphones (Figure 11, right). The beekeeper using the multifunctional weighing device we have designed can monitor the hive status not only through the web application but also through a smartphone. All parameters that would have to be set directly through the MFP can also be set through ThingSpeak or ThingView. Using both applications, it can get a detailed overview of the hive status based on the available statistics (Figure 11, right):

1. Weight,
2. Outdoor temperature,
3. External humidity,
4. Indoor temperature,
5. Indoor humidity,
6. The amount of rainfall.

In case of adding a new sensor to the device, the platform allows the creation of dependencies and processing of statistics by comparing this parameter with other available sensory parameters or time interval.

Discussion – Comparing another DIY solution with our solution

The device we designed was created according to the requirements specification made by a beekeeper and the Department of Ecology and Environmental Sciences of the Faculty of Natural Sciences, Constantine the Philosopher

University in Nitra (Figure 12).



Source: own creation

Figure 12: Our designed and implemented device.

The device is currently being actively used. Any device that has been designed, created and registered for a Sigfox data network backend must go through the registration process. Currently, there are various commercial solutions (described in the Commercial Solutions section of selected hive scales), but their cost is often prohibitively high for beekeepers. Also, the possibilities of commercial solutions are often limited by the capabilities of the used data network. For this and other reasons, the market has seen an increase in DIY (Do-It-Yourself) solutions. Thus, the Discussion section will be comparing our solution (Table 1) with the DIY solution by Nathan Seidle (hereinafter referred to as DIY-NS).

Component	DIY-NS	Price (€)	Our solution	Price (€)
Microcontroller	Arduino UNO	20.00	Arduino MKR FOX 1200	36.41
Wi-Fi module	Spark Core	35.33	X	X
Weight sensor	HX711 + personal weight	79.56	HX711 + strain gauge sensor + OSB board	10.00
Rain sensor	X	X	Printed on a 3D printer	1.16
Outdoor thermometer and humidity meter	X	X	DHT22	2.54
Internal thermometer and humidity meter	SparkFun Si7021	7.00	DHT22	2.54
Sensor of real time	X	X	DS3231	2.10
electric current sensor	X	X	INA219	1.21
Vibration sensor	X	X	SW18010P	0.36
EEPROM	X	X	24LC256	0.26
Display	X	X	E-Paper	16.18
Accumulator	6Ah Li-Po	26.50	2x3,4Ah 18650 Li-Ion	4.60
Solar panel	2.5W	34.00	3x1,25W	4.53
Case	X	X	Printed on 3D printer*	3.22
Antenna	2.4GHz	4.38	868MHz 5db	2.46
Buttons	X	X	3x waterproof buttons	0.87
Together		206.77		88.44

Note: * The print was realized with the ABS material type, filament diameter 1.75mm, consumed filament length 80m. The price of new filament 21.70€/kg. The full price of material cost 4.38€ (rain sensor + case).

Source: own creation

Table 1: Price comparison of the two solutions

DIY-NS uses Wi-Fi to transmit data from the hive. Data is gathered by the Arduino Uno microcontroller using only two sensors: weight sensor and integrated temperature and humidity sensor. The DIY-NS uses an off-the-shelf personal scale connected to the HX711 as a weight sensor. Strain gauge cells are, like in our solution, connected to the Wheat Bridge. The HTU21D sensor, which communicates with the microcontroller via I2C, is used to collect hive temperature and humidity data. The DIY-NS is powered by a 6Ah lithium polymer battery that is recharged using a 2.5W solar panel. The analog.io platform is used for data visualization, and data.sparkfun.com is used for data processing (Seidle, 2019).

DIY-NS solution is ultimately several times more expensive than our proposed solution, also contains fewer sensors and uses Wi-Fi to collect data. This means, that the device needs to be within the range of a Wi-Fi access point (Wifi has a much lower range compared to Sigfox and isn't usable in mountain areas unlike SigFox transmitters).

Conclusion

In this paper, we have presented a design and implementation of a multifunctional weighing device intended for beekeepers' needs. The designed device can communicate with the user via the IoT SigFox data network. The device periodically sends information

about the condition of the hive and its surroundings. It can measure the indoor (hive) temperature and relative humidity of the hive as well as the outdoor temperature and humidity. The hive weighing is accurate to 50g (as opposed to 100g in current commercial solutions compared in this paper) at a maximum hive weight of up to 200kg. Measurement of rainfall is also possible, thanks to a tipping bucket rain gauge. The system is secured by a vibration sensor. The device is powered by two lithium-ion batteries that can maintain the system (when the system is in power saving mode of 4-7mA) for about 2 months, with a frequency of sending messages every 15 minutes. 7 additional modes are available to extend battery life, and 3 solar panels are mounted on the device for charging the batteries. Data received from the device is displayed using ThingSpeak IoT website or the ThingView Android app. Regular SMS status messages are also a feature of the system. Overall, our designed device is several times cheaper than commercial devices available on the market. Currently, several beekeepers in Slovakia are actively using the unit.

Acknowledgements

This work was supported with the financial support of the project KEGA 036UKF-4/2019, Adaptation of the learning process using sensor networks and the Internet of Things.

Corresponding authors

PaedDr. Martin Magdin, Ph.D.

Department of Informatics, Faculty of Natural Science

Constantine the Philosopher University in Nitra, , Tr. A. Hlinku 1, 949 74 Nitra, Slovakia

E-mail: mmagdin@ukf.sk

References

- [1] Alya (2019a) "*VILKO*", Oct. 2019. [Online]. Available: Available: <http://www.alya.sk/vcelarska-vaha-vilko> [Accessed: 25 Oct. 2019].
- [2] Alya (2019b) "*MELLISA*", Oct. 2019. [Online]. Available: Available: <http://www.alya.sk/vcelarska-vaha-mellisa-2> [Accessed: 25 Oct. 2019].
- [3] BeeSpy (2019) "*BeeSpy*", Oct. 2019. [Online]. Available: <http://www.beespy.cz> [Accessed: 25 Oct. 2019].
- [4] Bulić, P., Kojek, G. and Biasizzo, A. (2019) "Data Transmission Efficiency in Bluetooth Low Energy Versions", *Sensors* (Switzerland), Vol. 19, No 17. E-ISSN 1424-8220. DOI 10.3390/s19173746.
- [5] Centenaro, M., Vangelista, L., Zanella, A. and Zorzi, M. (2016) "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios", *IEEE Wireless Communications*, Vol. 23, No. 5, pp. 60-67. E-ISSN 1558-0687, ISSN 1536-1284. DOI 10.1109/MWC.2016.7721743.
- [6] Debauche, O., Moulat, M. E., Mahmoudi, S., Boukraa, S., Manneback, P. and Lebeau, F. (2018) "Web Monitoring of Bee Health for Researchers and Beekeepers Based on the Internet of Things", *Procedia Computer Science*, pp. 991. ISSN 1877-0509. DOI 10.1016/j.procs.2018.04.103.

- [7] Edwards-Murphy, F., Magno, M., Whelan, P. M., O'Halloran, J. and Popovici, E. M. (2016) "B+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring", *Computers and Electronics in Agriculture*, Vol. 124, pp. 211-219. ISSN 0168-1699. DOI 10.1016/j.compag.2016.04.008.
- [8] Fitzgerald, D.vW., Edwards-Murphy, F., Wright, W. M. D., Whelan, P. M. and Popovici, E. M. (2015) "Design and development of a smart weighing scale for beehive monitoring", *26th Irish Signals and Systems Conference*, ISSC 2015. ISBN 978-1-4673-6974-9. DOI 10.1109/ISSC.2015.7163763.
- [9] Gil-Lebrero, S., Quiles-Latorre, F. J., Ortiz-López, M., Sánchez-Ruiz, V., Gámiz-López, V. and Luna-Rodríguez, J. J. (2017) "Honey bee colonies remote monitoring system", *Sensors* (Switzerland), Vol. 17, No. 1. E-ISSN 1424-8220. DOI 10.3390/s17010055.
- [10] Hossain, T., Doi, Y., Tazin, T., Ahad, M. A. R. M. and Inoue, S. (2018) "Study of LoRaWAN technology for activity recognition", *UbiComp/ISWC 2018 - Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pp. 1449-1453. ISBN 978-1-4503-5966-5/18/10. DOI 10.1145/3267305.3267510.
- [11] Jiang, J.-A., Wang, C.-H., Chen, C.-H., Liao, M.-S., Su, Y.-L., Chen, W.-S., Huang, C.-P., Yang, E.-C. and Chuang, C.-L. (2016) "A WSN-based automatic monitoring system for the foraging behavior of honey bees and environmental factors of beehives", *Computers and Electronics in Agriculture*, Vol. 123, pp. 304-318. ISSN 0168-1699. DOI 10.1016/j.compag.2016.03.003.
- [12] McLellan, A. (1977) "Honeybee colony weight as an index of honey production and nectar flow," *Journal of Applied Ecology*, Vol. 14, pp. 401-408. ISSN 1365-2664. DOI 10.2307/2402553
- [13] Mekki, K., Bajic, E., Chaxel, F. and Meyer, F. (2019) "A comparative study of LPWAN technologies for large-scale IoT deployment", *ICT Express*, Vol. 5, No. 1, pp. 1-7. ISSN 2405-9595. DOI 10.1016/j.ict.2017.12.005.
- [14] Mekki, K., Bajic, E., Chaxel, F. and Meyer, F. (2018) "Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT", *2018 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2018*, pp. 197. E-ISSN 978-1-5386-3227-7, ISBN 978-1-5386-3228-4. DOI 10.1109/PERCOMW.2018.8480255.
- [15] Payne, E. K., Lu, S., Wang, Q. and Wu, L. (2019) "Concept of designing thermal condition monitoring system with ZigBee/GSM communication link for distributed energy resources network in rural and remote applications", *Processes*, Vol. 7, No. 6. E-ISSN 2227-9717. DOI 10.3390/PR7060383.
- [16] Raza, U., Kulkarni, P. and Sooriyabandara, M. (2017) "Low Power Wide Area Networks: An Overview", *IEEE Communications Surveys and Tutorials*, Vol. 19, No. 2, pp. 855-873. E-ISSN 1553-877X. ISSN 2373-745X. DOI 10.1109/COMST.2017.2652320.
- [17] Saelens, M., Hoebeke, J., Shahid, A. and Poorter, E. D. (2019) "Impact of EU duty cycle and transmission power limitations for sub-GHz LPWAN SRDs: an overview and future challenges", *Eurasip Journal on Wireless Communications and Networking*, Vol. 2019, No. 1. E-ISSN 1687-1499. DOI 10.1186/s13638-019-1502-5.
- [18] Seidle, N. (2019) "The Internet of Bees: Adding Sensors to Monitor Hive Health", Oct. 2019. [Online]. Available: <https://makezine.com/projects/bees-sensors-monitor-hive-health> [Accessed: 25 Oct. 2019].
- [19] Sinha, R. S., Wei, Y. and Hwang, S. (2017) "A survey on LPWA technology: LoRa and NB-IoT", *ICT Express*, Vol. 3, No. 1, pp. 14-21. ISSN 2405-9595. DOI 10.1016/j.ict.2017.03.004.
- [20] Zacepins, A., Kviesis, A., Stalidzans, E., Liepniece, M. and Meitalovs, J. (2016) "Remote detection of the swarming of honey bee colonies by single-point temperature monitoring", *Biosystems Engineering*, Vol. 148, pp. 76-80. ISSN 1537-5110. DOI 10.1016/j.biosystemseng.2016.05.012.