HARDWARE AND SOFTWARE FOR TEACHING LORAWAN TECHNOLOGY IN ENGINEERING EDUCATION

D. Tokmakov

Plovdiv University Paisii Hilendarski (BULGARIA)

Abstract

The Internet of Things (IoT) has become one of the fastest growing fields and an increasing number of jobs require expertise in this field. To connect "things" to Internet we need a radio communication technology, which transmits the data from wireless sensor nodes to Internet like Bluetooth, Wi-Fi, Zigbee, 3G, 4G/LTE, 5G and LPWAN. Each solution has its strengths and weaknesses in various network criteria and is therefore best suited for different IoT use cases. LoRaWan is the new phenomenon in IoT. By providing long-range communication on small, inexpensive batteries that last for years, this communication technology is purpose-built to support large-scale IoT networks sprawling over vast industrial and commercial campuses. LoRaWan technology connects all types of IoT sensors – facilitating numerous applications from asset tracking, environmental monitoring and facility management to occupancy detection and consumables monitoring.

In this paper, we share our experience in teaching of LoRaWan technology in the discipline "Internet of Things" for the students from the bachelor's degree in engineering at Plovdiv University, Bulgaria. We present the design of the hardware and software of two sensor nodes with LoRaWan transceivers, which we use in the lab experiments for teaching LoRaWan connectivity. The first sensor node incorporates Atmega 328p microcontroller, RFM95 Long Range, Low Power RF Transceiver 860-1000 MHz with LoRa® Technology, TPL5110 nano timer and BMP280 - an absolute barometric pressure sensor, which is especially feasible for mobile applications. The nano timer TPL5110 is used to "wake-up" the microcontroller from "deep sleep" mode with external interrupt attached to PD3 of Atmega328p. This decreases significantly the current consumption of the sensor node in deep sleep mode to 290nA which is very low value ensuring a long battery life of several years.

The second sensor node consists of Atmega 328p microcontroller, RN2483 a fully certified 868 MHz module based on wireless LoRa® technology, I2C temperature sensor and MCP16251 Low Quiescent Current, PFM/PWM Synchronous Boost Regulator that converts a single AA battery of 1.5V to 3.3V power supply.We use Atmega328p microcontrollers that is easily programmed with the Arduino IDE with which the students are familiar. In our lab experiments, we teach all fundamental aspects of LoRaWan modulation like spreading factor (SF), Coding Rate (CR), Chirps Spread Spectrum and how all this parameters affects the communication range and power consumption of the sensor nodes. The students are given the opportunity to change all communication parameters such as SF, CR, the amount of payload sent to LoRaWan application servers, as well as the type of authentication between Over-the-Air Activation (OTAA) and Activation by Personalization (ABP). The developed sensor nodes allows students to fully perform their laboratory exercises in the discipline "Internet of Things" for LoRaWan communication technology and enhance their learning capabilities, increase their interests in the field of IoT and especially in LoRaWan for which technology there is a lack of educational resources and educational hardware modules.

Keywords: Internet of things, LoRaWan lab experiments, Arduino.

1 INTRODUCTION

The LoRaWAN® specification is a Low Power, Wide Area (LPWA) networking protocol designed to wirelessly connect battery operated 'things' to the internet in regional, national or global networks, and targets key Internet of Things (IoT) requirements such as bi-directional communication, end-to-end security, mobility and localization services. LoRaWAN® network architecture is deployed in a star-of-stars topology in which gateways relay messages between end-devices and a central network server. The gateways are connected to the network server via standard IP connections and act as a transparent bridge, simply converting RF packets to IP packets and vice versa. The wireless communication takes advantage of the **Lo**ng **Ra**nge characteristics of the LoRa physical layer, allowing a single-hop link between the end-device and one or many gateways. All modes are capable of bi-directional communication, and there is support for multicast addressing groups to make efficient use of spectrum during tasks such as Firmware Over-The-Air (FOTA) upgrades or other mass distribution messages.

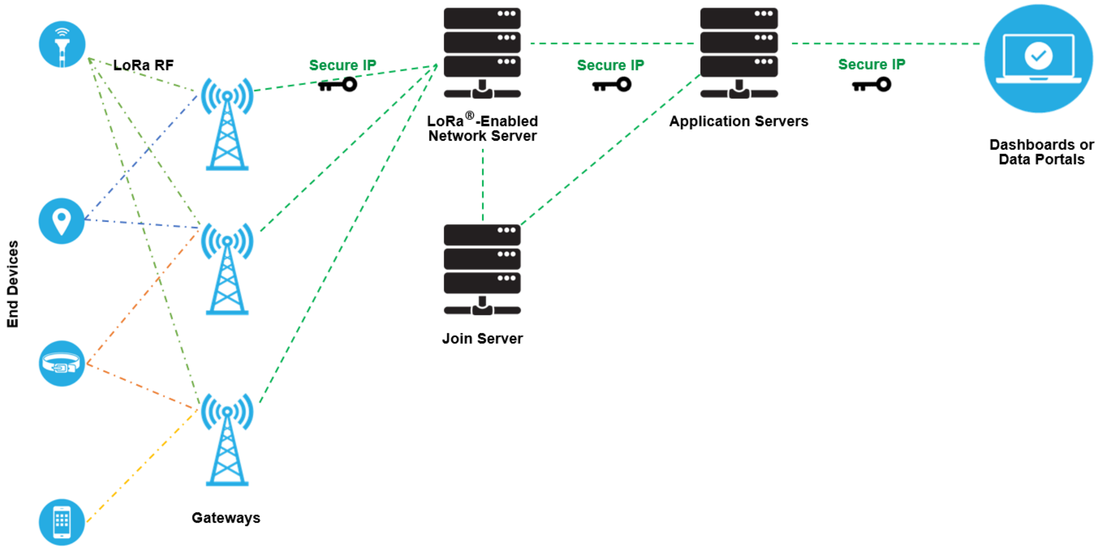

Fig.1 LoRaWan architecture [1]

A LoRaWAN-enabled **end device** is a sensor or an actuator, which is wirelessly connected to a LoRaWAN network through radio gateways using LoRa RF Modulation. In the majority of applications, an end device is an autonomous, often battery-operated sensor that digitizes physical conditions and environmental events. Typical use cases for an actuator include: street lighting, wireless locks, water valve shut off, leak prevention, among others.

A LoRaWAN **gateway** receives LoRa modulated RF messages from any end device in hearing distance and forwards these data messages to the LoRaWAN network server (LNS), which is connected through an IP backbone- fig.1. [1]

The IP traffic from a gateway to the network server can be backhauled via Wi-Fi, hardwired Ethernet or via a Cellular connection. LoRaWAN gateways operate entirely at the physical layer and, in essence, are nothing but LoRa radio message forwarders. They only check the data integrity of each incoming LoRa RF message. If the integrity is not intact, that is, if the CRC is incorrect, the message will be dropped. If correct the gateway will forward it to the LNS, together with some metadata that includes the receive RSSI level of the message as well as an optional timestamp.[1]

For LoRaWAN downlinks, a gateway executes transmission requests coming from the LNS without any interpretation of the payload. Since multiple gateways can receive the same LoRa RF message from a single end device, the LNS performs data de-duplication and deletes all copies. Based on the RSSI levels of the identical messages, the network server typically selects the gateway that received the message with the best RSSI when transmitting a downlink message because that gateway is the one closest to the end device in question.[1]

The LoRaWAN network server (LNS) manages the entire network, dynamically controls the network parameters to adapt the system to ever-changing conditions, and establishes secure 128-bit AES connections for the transport of both the end to end data (from LoRaWAN end device to the end users Application in the Cloud) as well as for the control of traffic that flows from the LoRaWAN end device to the LNS (and back). The network server ensures the authenticity of every sensor on the network and the integrity of every message. At the same time, the network server cannot see or access the application data.[1]

Application servers are responsible for securely handling, managing and interpreting sensor application data. They also generate all the application-layer downlink payloads to the connected end devices.[1]

The join server manages the over-the-air activation process for end devices to be added to the network. The join server contains the information required to process uplink *join-request* frames and generate the downlink *join-accept* frames. It signals to the network server which application server should be connected to the end-device, and performs the network and application session encryption key derivations. It communicates the Network Session Key of the device to the network server, and the Application Session Key to the corresponding application server[1]

The LoRaWAN specification defines three end device types: **Class A**, **Class B**, and **Class C**. All LoRaWAN devices **must** implement Class A, whereas Class B and Class C are extensions to the specification of Class A devices.

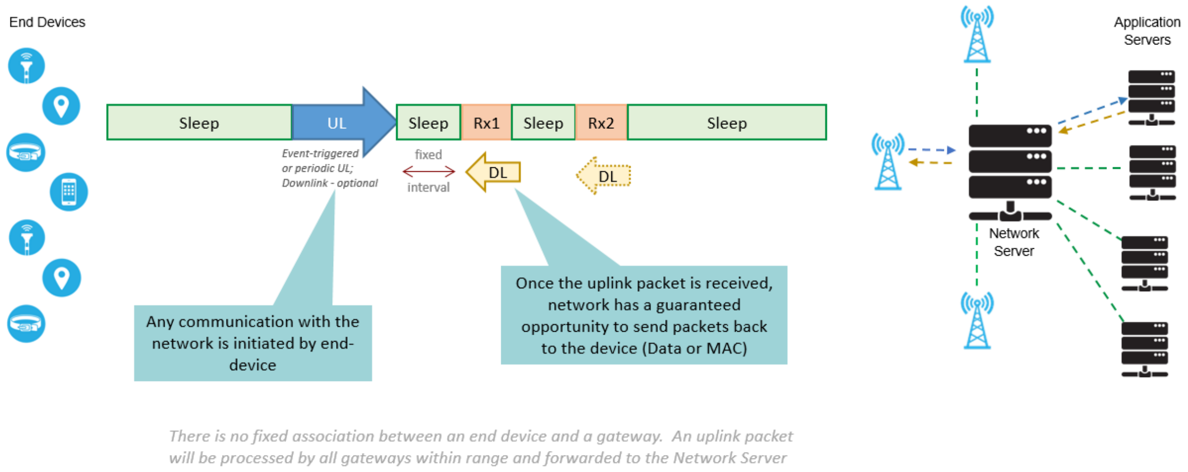

Fig.2. Class A operation [1]

In this case, (fig2) the end device spends most of its time in an idle state, (that is, in sleep mode). When there is a change in the environment related to whatever the device is programmed to monitor, it wakes up and initiates an uplink, transmitting the data about the changed state back to the network (Tx). The device then listens for a response from the network, typically for one second (although this duration is configurable). If it does not receive a downlink during this *receive window* (Rx1), it briefly goes back to sleep, waking a moment later, again listening for a response (Rx2). If no response is received during this second Rx window, the device goes back to sleep until the next time it has data to report. The delay between Rx1 and Rx2 is configured in terms of a delay from the end of the uplink transmission. [1]

An enhancement of Class A, LoRaWAN Class B mode offers regularly-scheduled, fixed-time opportunities for an end device to receive downlinks from the network, making Class B end devices suitable for both monitoring sensors as well as actuators. All LoRa-based end devices start in Class A mode; however, devices programmed with a Class B stack during manufacturing may be switched to Class B mode by the application layer.[1]

Class C are always “on”; that is, they do not depend on battery power. Class C devices include such things as street lights, electrical meters etc. These devices are always listening for downlink messages, unless they are transmitting an uplink. As a result, they offer the lowest latency for communication from the server to an end device.[1]

2 END DEVICES FOR TEACHING LORAWAN TECHNOLOGY

In this part we present the hardware and software of two end devices of class A, developed by us, which we use for laboratory experiments in the subject "Internet of Things" at the Faculty of Physics and Technology in Plovdiv Univerity Paisii Hilendarski.

The end devices (sensor nodes) are battery operated and are designed to have the lowest possible power consumption.

2.1 LoRaWan end device with ATMEGA 328 and SX1272 Long Range, Low Power RF Transceiver 860-1000 MHz with LoRa® Technology

Figure 3 shows the schematic diagram of the LoRaWan end device (sensor node) with ultra low power consumption. It consists of the ATMEGA328 microcontroller, LoraWan Tranciever SX1272, atmospheric pressure and temperature sensor BMP280 and nano timer TPL5110.

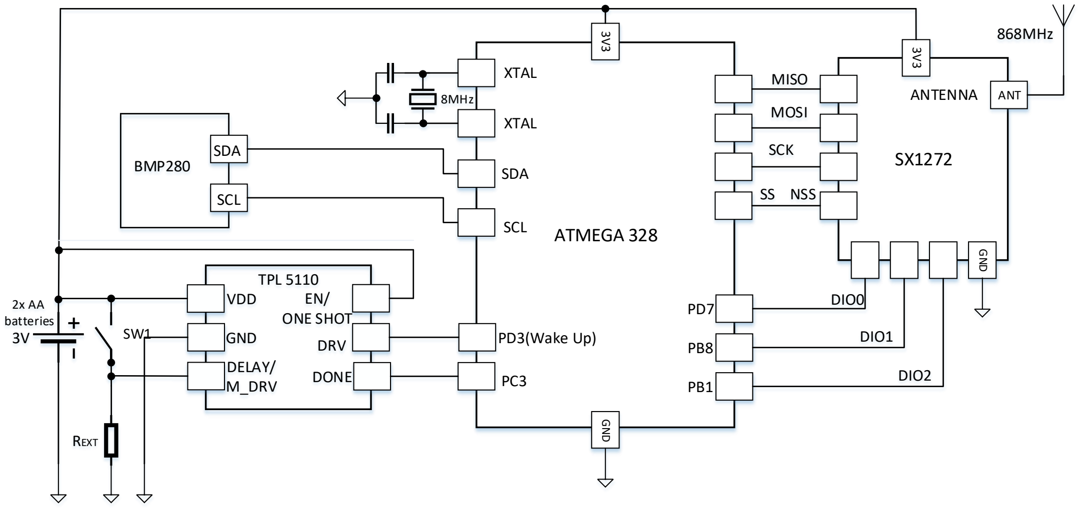

Fig.3 Ultra low-power LoRaWan end device

Two 1.5V AA batteries provide the 3V power supply to the sensor node. The choice of the ATMEGA328 microcontroller was made due to the ability to use the Arduino IDE development environment to develop software for low power sensor units which is which is very suitable for applications in engineering education.

The TPL5110 nano timer is used to wake the ATMEGA328 microcontroller from low power deep sleep mode via an external interrupt attached to pin PD3 of the microcontroller. The TPL5110 provides selectable time intervals from 100 ms to 7200s, adjustable by external resistance (REXT) between the DELAY / M_DRV output of the TPL5110 and GND. The SW1 switch provides the possibility for forced waking up and exit from the low-energy mode of the ATMEGA328 microcontroller. Pressing the switch SW1 generates a low signal at the DRV-PD3 pins, which is triggered as an external interrupt of the microcontroller.

BMP280 is a barometric pressure sensor specially designed for mobile applications. It measures pressure ranging from 300 hPa to 1100 hPa, altimetry up to 9000 m and temperature from -20 ° C to + 85 ° C. Data reading is performed via I2C digital communication interface. Its use in the architecture of the sensor unit is dictated by its ultra low power consumption. Also, to have real data to be measured at certain intervals and sent via the LoraWan interface to the LoraWan gateway and from there to the application server.

Figure 4 shows the timing diagram of the ultra low consumption sensor node. In state 1, the sensor node is placed in low power mode with the analog-to-digital converter, monitoring timer (WDT) and detection circuit (BOD) of the ATEMGA328 microcontroller switched off. This leads to a significant reduction in current consumption. The time interval of tDRV in low energy mode is about 5 minutes and is determined by the R_{EXT} connected to the TPL5110 nano timer.

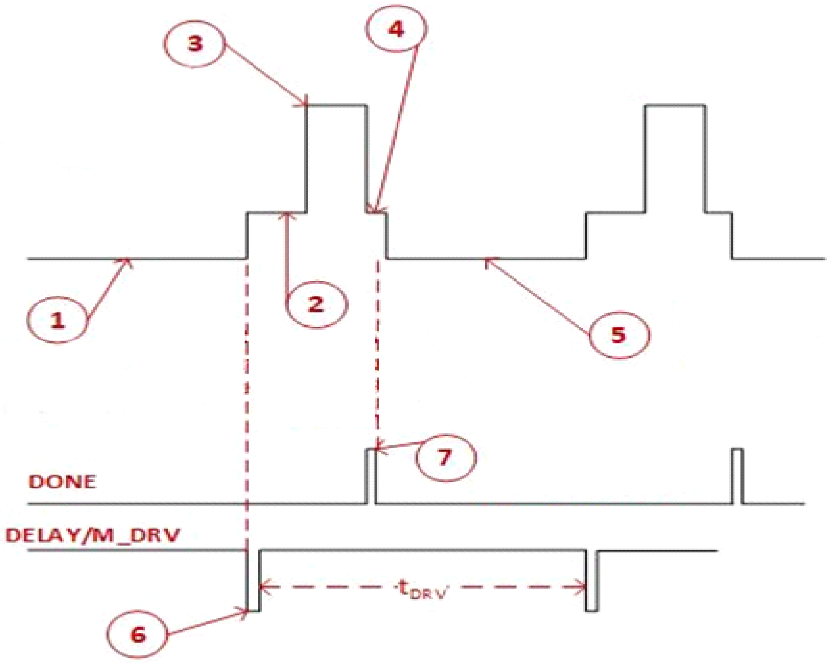

Fig.4 Timing diagram of ultra-low power sensor node

In the 6 state, tDRV time interval expires and the DRV goes to a low level, which triggers the ATMEGA328 microcontroller as an external interrupt to the PD3 port. The sensor node “wake-up” the low power mode and goes to state 2, where the data from the BMP280 sensor is read and the data is prepared and transmitted. The duration of state 2 is about 3.3 seconds. The current consumption in state 2 is Ic = 3.66mA. In state 3, the sensor node switches to data transmission mode, where the data from the BMP280 is sent to the application server via a LoRaWan transceiver SX1272. The duration of the state time 3 is 200ms, and the measured consumption current is Ic = 12mA with spreading factor of 9 (SF9) and a bandwidth of 125KHz (BW125) parameters of LoraWan modulation.

In state 4, the sensor node waits for the two downlink Rx receive windows from the application server. At the end of state 4, the ATMEGA328 microcontroller generates a sleep signal from the PC3 terminal, which is connected to the DONE input of the TPL5110 nano timer. This leads to the generation of a new tDRV interval, which again puts the microcontroller in a low-energy mode - "deep sleep", where the current consumption from the batteries is only 290 nA.

The described sensor node algorithm is implemented in the firmware of the ATMEGA328 microcontroller, which uses the IBM LMIC (LoraMAC-in-C) v1.5 library, modified to work in the Arduino environment, allowing the use of Semtech SX1272 / SX1276 transceivers with the Arduino IDE development environment. A fragment of the code is presented in Figure 5.

Figure 6 shows a practical realized hardware module of the described sensor node with ultra-low energy consumption, which is used to train students in engineering education in the subject of Internet of Things.

A specialized measurement device current ranger is used to measure the current in deep sleep mode. As can be seen from Fig. 6, the current consumption in the deep sleep mode is only 290 nA.

Figure 7 shows the data transmitted from the described sensor node myDevices application server.

```
void do_send(osjob_t* j) {
  int batt = (int)(readVcc() / 100);  // readVCC returns  mVolt need just 100mVolt steps
byte batvalue = (byte)batt; // no problem putting it into a int.
  byte buffer[3];


  sensors.requestTemperatures(); // Send the command to get temperatures
  float degC = sensors.getTempCByIndex(0);
  int16_t celciusInt=degC*100;

      buffer[0] = batvalue;
      buffer[1] = celciusInt >> 8;
      buffer[2] = celciusInt;
   // Check if there is not a current TX/RX job running
  if (LMIC.opmode & OP_TXRXPEND) {
    Serial.println(F("OP_TXRXPEND, not sending"));
  } else {
    // Prepare upstream data transmission at the next possible time.
    //LMIC_setTxData2(1, (uint8_t*) buffer, 2 , 0);
    LMIC_setTxData2(1, buffer, sizeof(buffer), 0);
    Serial.println(F("Sending: "));
  }
}

// initial job
static void initfunc (osjob_t* j) {
    // reset MAC state
    LMIC_reset();
        // start joining
    LMIC_startJoining();
    // init done - onEvent() callback will be invoked...
}

void wakeUp()
{
```

Fig.5 Fragment of the software of ultra-low power sensor node

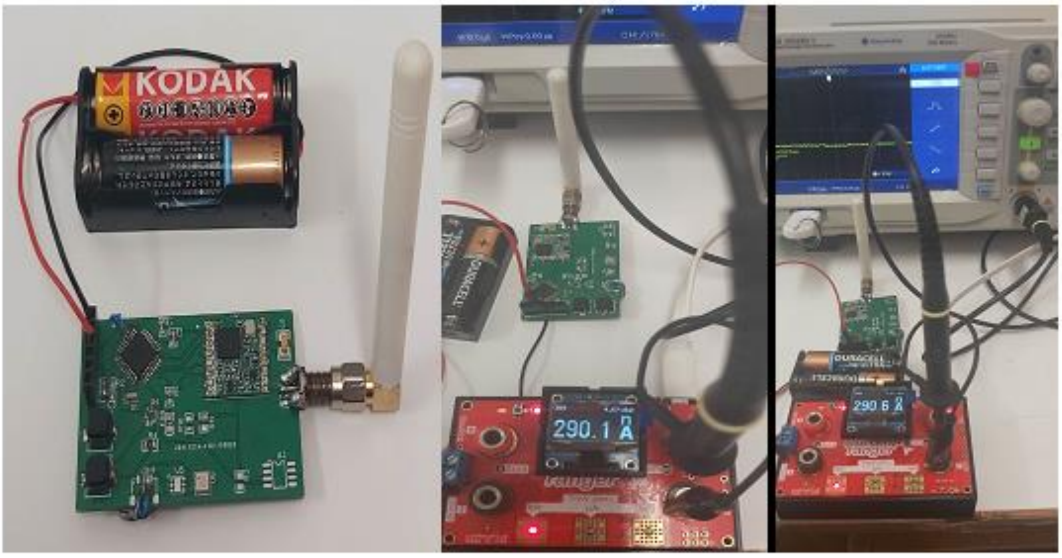

Fig.6 Hardware module for lab experiments with LoRaWan sensor node in IoT engineering course

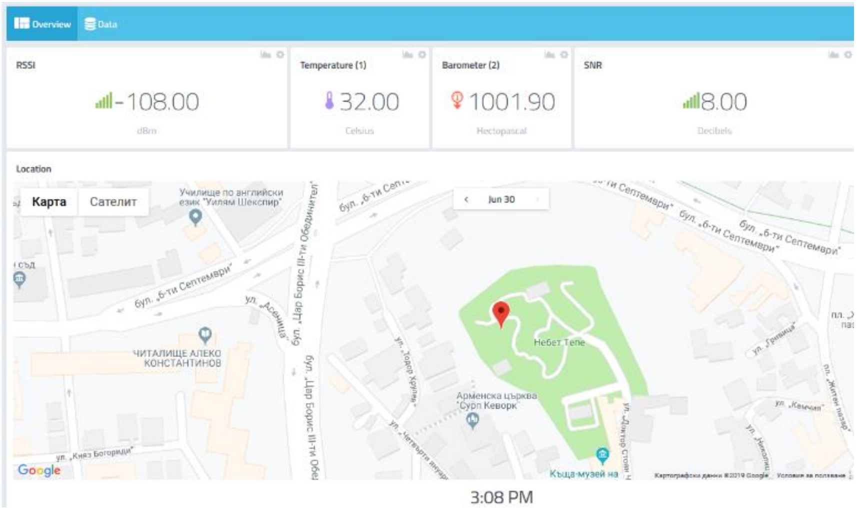

Fig.7 Data send from ultra-low power sensor node to myDevices application server

2.2 LoRaWan sensor node with ATMEGA328 microcontroller and RN2483 transceiver

The circuit diagram of another sensor node is shown on fig.8. and consists of the ATMEGA328 microcontroller, the BME280 barometric sensor and the LoRaWan tranciever RN2483.

A 1.5V AA battery and MCP1251 step-up DC-DC converter are used to power the sensor node. The BME 280 is an improved version of the BMP280 sensor, with the addition of relative humidity measurement.

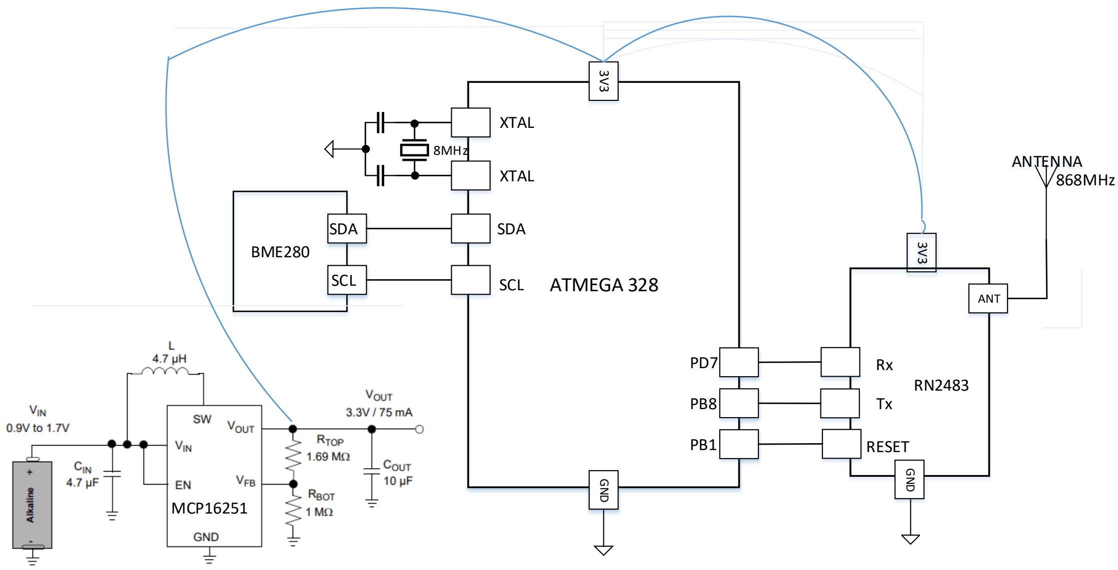

Fig.8 LoraWan sensor node with ATMEGA328 and RN2483 transceiver

The communication between the microcontroller ATMEGA328 and a fully certified 868 MHz LoRaWan module RN2483 is carried out through a serial interface (Rx, Tx) on two of the pins of the microcontroller, where software serial is implemented on PD7 and PD8. The RESET signal of RN2483 is controlled by PB1 of the ATMEGA328 microcontroller.

The use of a step-up DC-DC converter and 1 1.5V AA battery to power the sensor unit is for educational purposes. In this way, students get acquainted with the use of specialized integrated circuits for powering battery equipment.

Fig.9 shows the fragment of sensor node code. Arduino IDE is used for software development of the sensor node firmware and different software libraries including rn2xx3.h, SoftwareSerial.h, Adafruit_BME280.h and LowPower.h.

bme280_2483_batt_int_sleep_by_radio_bme_power_from_pin_step_up

```
#include <rn2xx3.h>
#include <SoftwareSerial.h>
#include <CayenneLPP.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include "LowPower.h";
//int sleepcycles = 30;  // every sleepcycle will last 8 secs, total sleeptime will be sleepcycles * 8 sec
//int sleepcounter=0;
SoftwareSerial mySerial(7, 8); // RX, TX
#define RST  2
#define GND_BME280  5
Adafruit_BME280 bme;

// Copy the following lines from TTN Console -> Devices -> Overview tab -> "EXAMPLE CODE"
const char *devAddr = "26011253";
const char *nwkSKey =  "7D60FB971443DA78AA04909161DE8857";
const char *appSKey =  "24857D228C3AC09BCEE70493337C47B8";

rn2xx3 myLora(mySerial);
CayenneLPP lpp(51);

// Setup routine runs once when you press reset

void setup() {
  pinMode(13, OUTPUT);
  led_on();

  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  mySerial.begin(9600);
   pinMode(GND_BME280, OUTPUT);
   digitalWrite(GND_BME280, LOW);
   delay(500);
  if (!bme.begin(0x76)) {
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
    while (1);
  }
```

Fig.9 Fragment of the software of sensor node with RN2483

Fig. 9 shows the interface of the myDevices application server, showing the measurement data sent by the sensor node. In addition to temperature, atmospheric pressure, and relative humidity, Vbaatt battery voltage is also measured using the ATMEGA328's built-in ADC.

The Cayenne Low Power Protocol (LPP) is used for data packaging, the software implementation of which is performed by the CayenneLPP.h library.

From Fig. 9 it is clear that the developed sensor node with ATMEGA328 and RN2483 powered by a step-up converter MCP16251 is operational even at low battery voltages of 0.7V.

Low-voltage technology allows the regulator to start-up without high inrush current or output voltage overshoot from a low-voltage input.

Firmware of ATMEGA328 is designed so that after transmitting the data to the LoRaWan application server goes into low power consumption mode (class A end device) for a period of 10 minutes, thus transmitting 6 measurements in 1 hour.

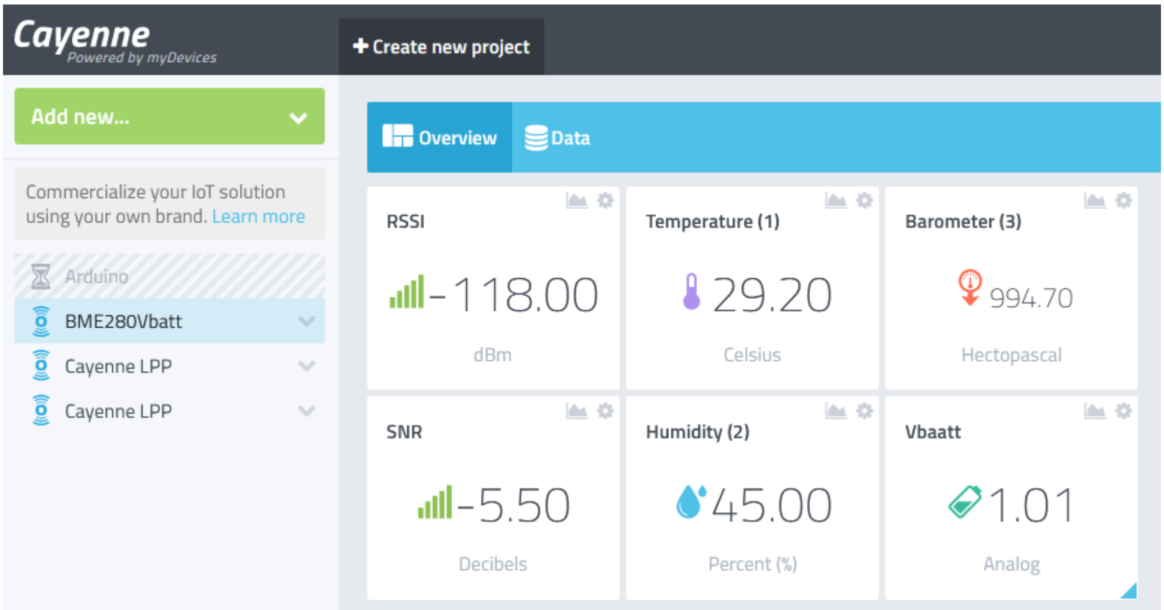

Fig.10 Data send from sensor with RN2483 node to myDevices application server

3 RESULTS

The LoRaWan sensor nodes described in 2.1 and 2.2 are used in the training of engineering students in the subject Internet of Things. By using the ATMEGA 328 microcontroller in both implementations of sensor nodes, it is possible for software development to use the Arduino IDE environment with which students are familiar with the course on microprocessor systems. In the laboratory course exercises students are given the opportunity to change the parameters of LoRaWan communication such as SF, CR, the type of authentication ABP (Activation By Personalization) and OTAA (Over-The-Air Activation). Students program the cryptographic keys in the firmware of the sensor nodes as well as work with the cloud applications in The Things Network and myDevices.

With project-based learning and performance-centered approach, students acquire not only theoretical knowledge of LoRaWan technology but also practical skills for programming and developing wireless sensor nodes with low power consumption.

4 CONCLUSIONS

In our lab experiments, in Internet of Things course we teach all fundamental aspects of LoRaWan modulation like spreading factor (SF), Coding Rate (CR), Chirps Spread Spectrum and how all this parameters affects the communication range and power consumption of the sensor nodes. The students are given the opportunity to change all communication parameters such as SF, CR, the amount of payload sent to LoRaWan application servers, as well as the type of authentication between Over-the-Air Activation (OTAA) and Activation by Personalization (ABP).The developed sensor nodes allows students to fully perform their laboratory exercises in the discipline “Internet of Things” for LoRaWan communication technology and enhance their learning capabilities, increase their interests in the field of IoT and especially in LoRaWan for which technology there is a lack of educational resources and educational hardware modules

ACKNOWLEDGEMENTS

This work was funded by the following projects: МУ21-ФтФ-012 /2021/ of the Department for Scientific Research of Plovdiv University, Bulgaria, National program "Young scientists and postdoctoral students" at the Ministry of Education and Science, Bulgaria, and EU project №620101-EPP-1-2020-1-BG-EPPKA3-VET-COVE - European Centre of Vocational Excellence in Microelectronics.

REFERENCES

[1] "What are LoRa® and LoRaWAN®?",https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan, accessed 18/09/2021

[2] M. Mihálik, M. Hruboš and E. Bubeníková, "Wireless Proximity Sensor in LoRaWAN Network," *2020 International Conference on Applied Electronics (AE)*, 2020, pp. 1-4, doi: 10.23919/AE49394.2020.9232725.

[3] S. Tzanova, "Implementation of an Internationalised Master Degree Courses in Nanoelectronics and Nanotechnologies in Asian Universities," 2021 12th National Conference with International Participation (ELECTRONICA), 2021, pp. 1-4, doi: 10.1109/ELECTRONICA52725.2021.9513661.

[4] V. Mengov, "Remote access to computer aided lab experiment for students in electronics engineering courses," 2020 XXIX International Scientific Conference Electronics (ET), 2020, pp. 1-4, doi: 10.1109/ET50336.2020.9238209.

[5] A. H. Chekichev, D. A. Shehova, S. Y. Lyubomirov, S. M. Asenov and K. K. Asparuhova, "Researching and Modeling of Discrete Linear Time-Invariant Systems with Difference Equations in Matlab," *2020 XXIX International Scientific Conference Electronics (ET)*, 2020, pp. 1-4, doi: 10.1109/ET50336.2020.9238252.