KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRICAL AND ELECTRONICS ENGINEERING

Shathya Arumugam Kodeeswaran

RESEARCH AND DEVELOPMENT OF WIRELESS NETWORK NODE FW OPTIMISING POWER CONSUMPTION

Master's Degree Final Project

Supervisor

Prof. dr. Vytautas Deksnys

KAUNAS, 2016

KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRICAL AND ELECTRONICS ENGINEERING

RESEARCH AND DEVELOPMENT OF WIRELESS NETWORK NODE FW OPTIMISING POWER CONSUMPTION

Master's Degree Final Project

Electronics Engineering (621H61002)

Supervisor

(signature) Prof. dr. Vytautas Deksnys

(date)

Reviewer

(signature) Prof. dr. Vaidotas Marozas

(date)

Project made by

(signature) Shathya Arumugam Kodeeswaran

(date)

KAUNAS, 2016

KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRICAL AND ELECTRONICS ENGINEERING

(Faculty)

Shathya Arumugam Kodeeswaran

(Student's name, surname)

Electronics Engineering (621H61002)

(Title and code of study programme)

"RESEARCH AND DEVELOPMENT OF WIRELESS NETWORK NODE FW OPTIMISING POWER CONSUMPTION"

DECLARATION OF ACADEMIC INTEGRITY

10 June 2016

Kaunas

I confirm that the final project of mine, **Shathya Arumugam Kodeeswaran**, on the subject "RESEARCH AND DEVELOPMENT OF WIRELESS NETWORK NODE FW OPTIMISING POWER CONSUMPTION" is written completely by myself; all the provided data and research results are correct and have been obtained honestly. None of the parts of this thesis have been plagiarized from any printed, Internet-based or otherwise recorded sources. All direct and indirect quotations from external resources are indicated in the list of references. No monetary funds (unless required by law) have been paid to anyone for any contribution to this thesis.

I fully and completely understand that any discovery of any manifestations/case/facts of dishonesty inevitably results in me incurring a penalty according to the procedure(s) effective at Kaunas University of Technology.

(name and surname filled in by hand) *(signature)*

Arumugam Kodeeswaran, Shathya *RESEARCH AND DEVELOPMENT OF WIRELESS NETWORK NODE FW OPTIMISING POWER CONSUMPTION:* Final Project of Electronics Engineering Master Degree / supervisor Prof. dr. Vytautas Deksnys; Kaunas University of Technology, Electrical and Electronics Engineering Faculty, Electronics Engineering Department. Kaunas, 2016. 58 pages.

SUMMARY

This master thesis deals with weather monitoring system using LoRa technology which fetches weather conditions using various sensors like temperature, humidity, soil moisture, solar radiation, accumulated precipitation, etc., The measured data from the weather station are transmitted to a farm management information system for further processing and planning field works (seeding, irrigation, crops protection, harvesting, etc.) using featured precision agriculture technologies.

The main objective of this work is to develop long-range wireless data transmission solution optimising power consumption using license-free frequency ranges (169 MHz and 868 MHz). The advantage of this solution is high transmission distance using low RF power (till 100mW), high sensitivity, and data transfer security optimizing power consumption creating the possibilities to use renewable energy sources (solar energy and/or wind energy).

The research methodology part in this thesis deals with the details of weather monitoring system, hardware and software architecture development optimising for electric power reduction considering time synchronization between network nodes, using of power control modes for MCU and RF transceiver and data encoding for integrity and security.

Experimental part shows the workbench chosen for testing and analysing the overall energy consumption.

Research area and field: Wireless network node and optimising power consumption

Keywords: *Weather Monitoring System, RTC, Time Synchronization, Power Consumption, Time drift*

Arumugam Kodeeswaran, Shathya. Belaidžio ryšio mazgo su optimizuotomis maitinimo galios sąnaudomis projektavimas ir tyrimas. Elektronikos inžinerijos magistro baigiamasis projektas / vadovas prof. Vytautas Deksnys; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Kaunas, 2016, 58 pages.

Santrauka

Šiame darbe yra nagrinėjami aplinkos meteorologinių parametrų perdavimo klausimai minimizuojant elektrinės energijos sąnaudas tolimos zonos 169MHz ir 868 MHz radijo kanalais užtikrinant duomenų šifravimą. Naudojamos maitinimo galios sumažinimui panaudoti specialūs duomenų tėkmės valdymo kontrolerio ir siųstuvo/imtuvo darbo rėžimai ir papildoma laiko sinchronizacija laiko vienovei tarp ryšio sistemos mazgų užtikrinti.

Vykdant darbą buvo sukurti metodai, aparatinės bei programinės priemonės, kurie buvo eksperimentiniu būdu patikrinti vertinant maitinimo galios poreikį priėmimo, siuntimo ir laukimo režimuose.

Tyrimo sritis ir objektas: belaidžio ryšio mazgas su maitinimo energijos optimizavimu.

Raktiniai žodžiai: oro stotelė, laiko sinchronizavimas, galios sąnaudos, realaus laiko laikrodis, laiko poslinkis.

Table of Contents

List of Figures

List of Tables

1. Introduction

Weather monitoring is of great significance in numerous spaces, for example, precision agriculture, horticulture, air transport, amusement and so forth [1]. An automated weather station (AWS) is a system that measures and records meteorological parameters utilizing sensors without the interception of people. The measured parameters were transmitted to a remote area with the help of a communication link [2] [3].

LoRa is a wireless technology that has been developed for enabling long distance communication with low data rate by sensors and actuators for M2M and Internet of Things, IoT applications [4]. LoRa technology has some key features: secure spread spectrum modulation, extended transmission range (till 15-20 km on 500 mW RF power) and possibilities to use in many types of network topologies like broadcast, mesh, star, point-to-point and scanning.

The LoRa network consists of several elements: end points, LoRa gateway, server and remote computer. The LoRa network architecture is shown in Figure 1. As far as the actual structure of the LoRa system, the nodes are typically in a star-of-stars topology with gateways shaping a transparent bridge [4].

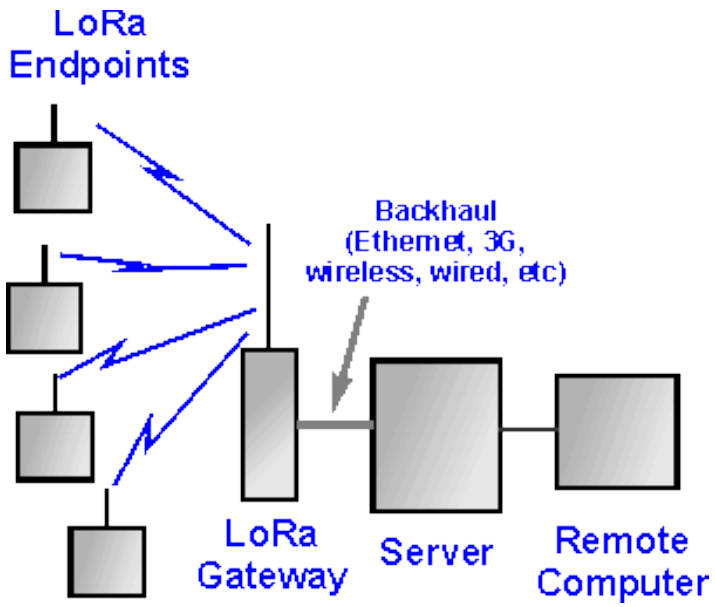


Figure 1 LoRa network architecture [4]

Communication to endpoint nodes are mostly bi-directional; however it additionally supports multicast operation, and this is useful for software upgrades and so forth or other mass distribution messages [4].

The SX1276 transceiver is a single-chip integrated circuit preferably suited for today's high-performance ISM band RF applications [5]. It was expected for use as long-range, high-performance, half-duplex bi-directional RF links, which requires stable and constant RF performances over the full operating range of the device down to 1.8V [5].

STM32L1 ultra-low-power microcontrollers are designed for reducing the overall system power consumption to maximize battery life cycle [6]. Power consumption calculation and the investigation are mainly dependent on how the developer utilizes the chose MCU's low-power features while dealing with the various factors that can influence the overall power consumption [6].

The proposed system collecting weather information based on Semtech LoRa Wireless RF technology using low power microcontroller and SX1276 transceiver.

1.1 Objective and Tasks

The main aim of this work is to analyze the parameters for weather conditions monitoring and to investigate the working mode of the wireless network node for optimizing power. The primary objective is to develop the wireless network node for long-range data transmission optimizing energy consumption in real time.

The following tasks are necessary to achieve the goal:

1. To create the experimental prototype of LoRa data flow controller and the transceiver for investigation of data transfer.
2. To develop firmware supporting data transfer integrity, security and retransmission possibilities for data exchange.
3. To develop firmware supporting time synchronization, time drift correction and optimizing power consumption.
4. To analyze the transmission and power consumption characteristics in three working modes: receiving, transmitting and sleep modes.
5. To calculate and analyze the energy consumption of the device.

1.2 Criteria of task

For data flow control, STM32L162 microcontroller from Cortex-M3 core running at 32 MHz with programmed sleep modes for reducing power consumption was chosen and the specialized hardware accelerator for AES128 ciphering purposes. The STM32L162 utilizes optimized

architecture and proprietary ultra-low-leakage process which combines high processing performance and ultra-low power consumption.

The LoRa proprietary protocol was used for data communication between only one master node and several slave nodes to form a star architecture network. The addressing modes used are unicast and broadcast. Each device can work as a receiver, transmitter or repeater. The master device receives data packets from the weather station and sends to slave device through intermediate slave (hop) device. Hop devices help to increase the transmission distance. In addition to that master sends data packets consisting of few parts ensuring addressing of destination and an intermediate device.

The two major factors affecting the microcontroller dynamic power consumption are system clock frequency and operating supply voltage. Decreasing the latter together makes a significant reduction of current consumption in this mode [6]. In our case, to reduce it, all the devices should be in deep sleep mode and wakes up at preamble interrupt to receive the packets and after successful communication, it goes back to deep sleep mode. The main advantage of this solution is to increase the time for sleeping optimizing power consumption.

For time synchronization between the nodes, Real Time Clock (RTC) of microcontroller were programmed. The current consumption for three working modes: Receiving mode (Rx mode), transmitting mode (Tx mode) and sleep mode need to be measured. The energy consumption of the entire device per day has to be measured. The time drifts also taken into consideration and for correcting it, the master broadcasts time synchronization packets twice a day to all the devices connected to the network.

2. Literature Review

2.1 Existing weather monitoring system and solutions

The existing system for weather monitoring, data acquisition and power consumption are discussed in this section. There are many types of weather stations in the market, which differ in their parameters and applications. For analysis, I have chosen few types of the systems used for agriculture applications.

Vaisala HydroMet Automatic Weather Station MAWS201

The MAWS201 is easy to set up using Vaisala set-up software wizard which is provided to configuring the sensor measurements, calculations, data transmissions and logging schedules [7]. It is widely employed in many applications meteorological research, fire weather, waste management and environmental impact studies. This weather station has primary sensors for measuring parameters such as wind speed and direction, atmospheric pressure, air temperature, precipitation and relative humidity. Configurable and generic 16 bit A/D conversion is incorporated for client's sensor. Optical sensors can be added to measure soil/water temperature, soil moisture, solar radiation and water level. Benefits of this system are portability, low power consumption, reliability, accuracy, extensive calculation and data logging capacity. MAWS201 automatic weather station is shown in Figure 2.

Figure 2 Vaisala HydroMet Automatic Weather Station MAWS201 [7]

Ratnik AutoSno Wireless Weather Monitoring System

This system is cost-effective that will help snowmaking operations run more efficiently and conveniently [8]. This system is battery powered, portable and solar-powered backup option and wind monitoring is available which measures temperature and relative humidity. The system is modular and can grow with snow resort needs. In a single network, up to 15 remote stations can operate which features temperature and relative humidity sensors associated with wireless radio transmitter with a typical range of 1 - 10 miles depending on antenna selection and viewable pathway [8]. AutoSno wireless weather monitoring system is shown in Figure 3.

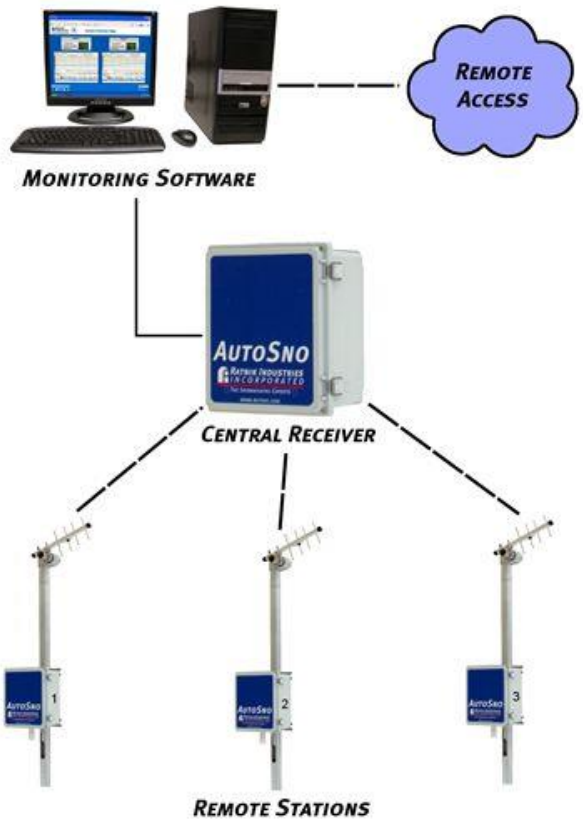


Figure 3 AutoSno Wireless weather monitoring system [8]

Global Water WE900 Weather Station

This system monitors wind direction, wind speed, humidity, temperature and solar radiation. The global water weather station is a low-cost system and rugged for the variety of weather monitoring including agriculture, landfills, water conservation, budget analysis, environmental studies and education [9]. Figure 4 shows the WE900 global water weather station.

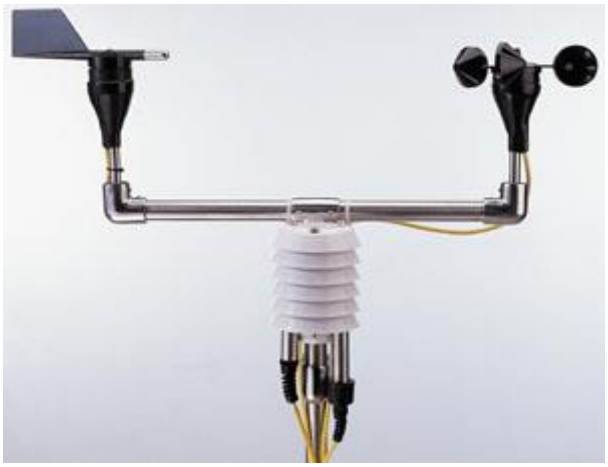
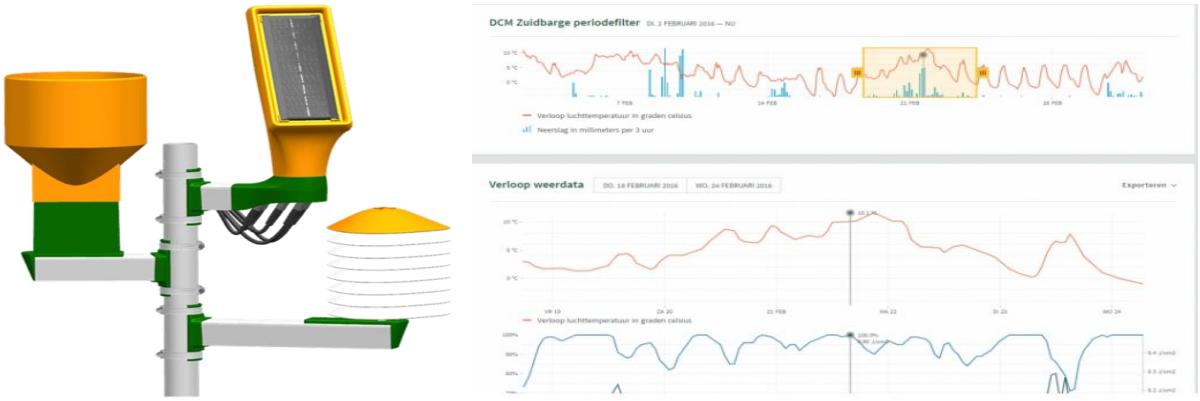


Figure 4 WE900 weather station [9]

Dacom weather station BASIC

It consists of a combi-sensor to measure temperature and relative humidity, a rain gauge and an SSG 3.0 datalogger with solar panel and communication functionality [10]. It offers temperature (◦C), precipitation (mm) and relative humidity (%). It works entirely on solar energy. Figure 5 a) and b) shows the Dacom weather station BASIC and simple weather data respectively.



a) b)

Figure 5 a) Dacom weather station BASIC b) weather data [10]

PESSL instrument iMetos – Model AG/CP/DD 280 – Weather station

The iMetos AG/CP/DD is a reliable weather station and robust suited for any requirements on weather and soil monitoring including evapotranspiration calculation using Penman-Monteith, perfect for irrigation management [11]. It is powered by solar panel and it also has an internal battery and transmits weather data in real time over GSM/ GPRS. It can send SMS Alarms to alert for events like rain, temperature, frost, etc. It comes with all the sensors for evapotranspiration

calculation: relative humidity, air temperature, rain gauge, wind speed and global radiation. Figure 6 shows the iMetos – Model AG/CP/DD 280 weather station.

Figure 6 iMetos – Model AG/CP/DD 280 – Weather station [11]

Wisconsin AWS system

University of Wisconsin-Madison AWS Specifications, the meteorological variables measured by sensor systems are air pressure, air temperature, wind direction, wind speed, temperature string and humidity [12]. This system stands out for its capacity to work in the harsh environment of Antarctic. An increasing number of AWS frameworks measure the accumulation of snow at sites utilizing Acoustic Depth Gauges (ADG) [13]. Figure 7 shows the final tower of Wisconsin AWS system and Figure 8 shows the data acquisition system of the weather station.

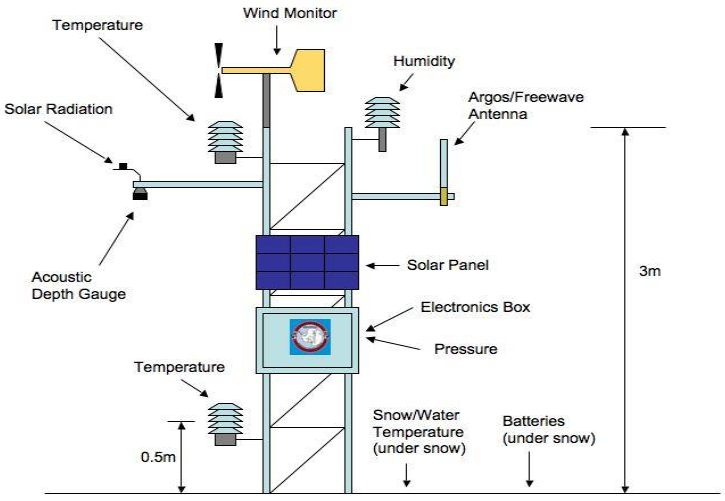


Figure 7 Final tower Wisconsin AWS system [13]

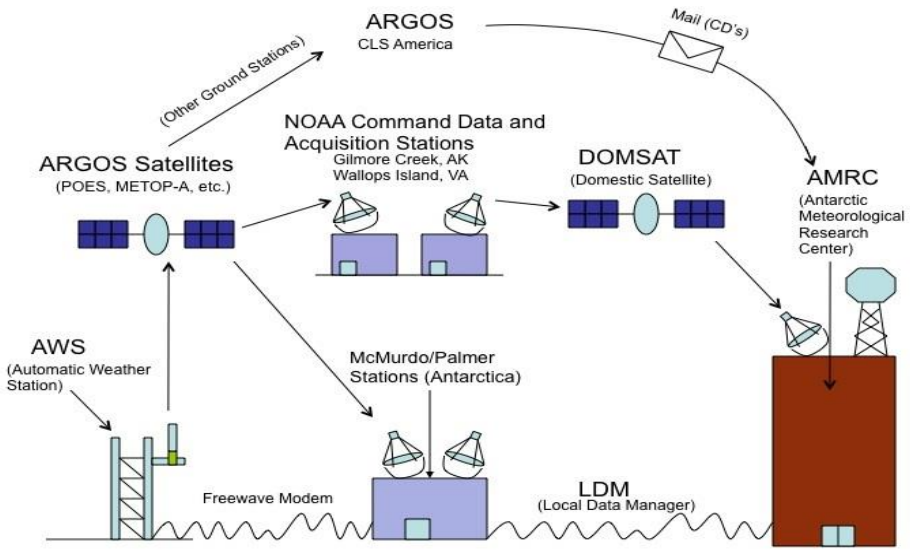


Figure 8 Data acquisition system [12]

From the above analysis, PESSL instruments and Dacom weather station BASIC are working using solar energy which is more suitable for our application measuring soil moisture and precipitation depth for irrigation purpose. These methods communicate using GSM/GPRS and SSM 3.0 data logger respectively. iMetos – Model AG/CP/DD 280 by PESSL also send SMS alert for the event like rain, frost, etc., Dacom BASIC weather station offers relative humidity and temperature. This type of weather station is robust, reliable and sustainable as it uses renewable energy sources to charge their internal battery.

2.2 Existing ciphering algorithm and low-power wireless technology

In this section, various ciphering processes and low-power wireless technologies for short-range communication are discussed.

AES algorithm

Advanced Encryption Standard (AES) gives symmetric block cipher that can process data blocks of 128 bits utilizing cipher keys with lengths of 128, 192 and 256 bits. Encryption plays a crucial role in information security system. The AES algorithm might be utilized with the three different key lengths as mentioned above and referred to as AES-128, AES-192, and AES-256. Cipher is series of transformations that converts plaintext to ciphertext using the cipher key which is secret, cryptographic key i.e. used by the key expansion routine to generate a set of round keys as a rectangular array of bytes having four rows and N columns. The number of cycles of repetition is

10, 12, and 14 for 128-bit keys, 192-bit keys, and 256-bit keys respectively. AES-192 and AES-256 have the higher level of decryption which is used at the top secret level. Our proposed system requires a secret level of communication and it should contain simple decryption process, so AES-128 was chosen.

AES128 Ciphering Process

This algorithm consists ten rounds of encryption as shown in Figure 9. Firstly, the 128-bit key is expanded into eleven supposed round keys, the size of each is 128 bits. Each round incorporates a transformation utilizing the corresponding cipher key to ensuring the security of encryption [14]. After first round key is XORed to the plain text nine equally structured rounds follow each consists of substitute bytes, mix columns, shift rows and add round key.

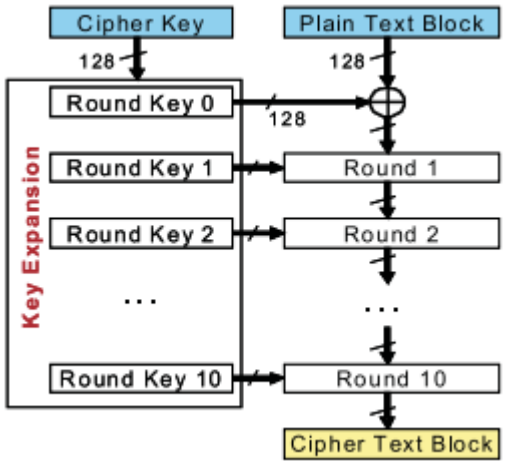


Figure 9 AES Algorithm Structure [14]

Our goal is to research and develop wireless network node firmware optimizing power consumption so it is necessary to analysis the existing low-power wireless technology for short range and long range. The upcoming analysis part deals with various existing low-power wireless technology, supported topology to form a network, power efficiency, energy per unit, range, throughput, latency, peak current consumption and battery life.

Existing Low- Power Wireless Technology for shorter range

In this section, the comparison of various low-power wireless technology is discussed which includes Bluetooth low energy, ANT, ANT+, ZigBee, RF4CE, Wi-Fi, Nike+, IrDA, and NFC. This part also deals with the detailing of features and analyzing the pros and cons of each protocol in various applications.

This technology tends to utilize an enormous amount of power and need specialized hardware to set up communication. Most target markets are portrayed by periodic transfer of small measures of sensor data between sensor nodes and a central device. The applications such as home automation, heating, ventilating and air conditioning (HVAC), human interface devices (HID), cell phones, health, fitness devices, etc. are all constrained by following essential critical requirements: ultra-low-power, physical size, and low cost.

Bluetooth® low energy (LE)

The aim of LE is to enable power sensitive devices to be permanently connected to the internet [15]. It primarily aimed at mobile telephones with star network topology, will often be created between the phone and other devices.

ANT

ANT is a low-power wireless technology which works in the 2.4 GHz spectrum [15]. The ANT transceiver device is treated as black box and it shouldn't require much effort to implement into a network. Its goal is to allow fitness and sports sensors to communicate with a display unit. ANT+ is a certification process in 2011 which is chargeable.

ZigBee

ZigBee introduced mesh networking to the low-power wireless and targeted towards home automation, remote control units and smart meters. Its complexity and power requirements do not make it suitable for unmaintained devices and it is not a frequency hopping technology, therefore, it requires careful planning deployment to ensure that there are no interfacing signals in the vicinity [15].

Radio Frequency for Consumer Electronics (RF4CE)

It is based on ZigBee standardized by four consumer electronics companies: Sony, Panasonic, Philips and Samsung. Texas Instruments and Freescale Semiconductor, Inc. supports RF4CE. It overcomes the problems associated with infrared and intended to use as a device remote control system.

Wi-Fi

Wireless Fidelity (Wi-Fi) ready to decrease its power consumption including IEEE Standard 802.11v and other proprietary standards [15]. It is a very efficient wireless technology optimized for large data transfer using high-speed throughput.

NIKE+

This technology allows users to monitor their activity levels while exercising. These devices are shipped as a single unit: processor, radio, and sensor. It consumes relatively high power and the design of the two-chip solution, consisting of a processor and a Nordic nRF2402 radio transceiver integrated circuits (IC) [15].

IrDA

Infrared Data Association (IrDA) works over a distance of less than 10 cm and has ultra-high-speed connectivity. The biggest problem is a line-of-sight requirement. It is not power efficient when compared to radio technologies. It needs a processor and a transceiver for its very nature.

Near Field Communication (NFC)

It works up to a range of approximately 5 cm and consumes relatively more power [15]. It is a flawless fit for its proposed use cases and has few competing technologies.

Network topologies

In low-power radio networks, there are five main network topologies:

Broadcast: A message is sent from a device that has been received by a receiver within the range. The broadcaster doesn't receive signals [15].

Mesh: A message can be relayed from one point to any other in a network by hopping through multiple nodes [15].

Star: A central device can communicate with many connected devices.

Scanning: This device is constantly in receive mode, waiting to pick up a signal from transmitting device within the range.

Point-to-point: In this mode, a one-to-one connection exists, where only two devices are connected.

Topology Vs Technology

Table 1 shows which technology support which network topologies

Table 1 Network topology supported by wireless technologies *[15]*

	BLE	**AN**	**A+**	**ZiB**	**RF**	**WF**	**Ni**	**Ir**	**NFC**
Broadcast	√	√[1]	√[1]	x	x	x	x	x	x
Mesh	√[2]	√	√	√	√	x	x	x	x
Star	√	√	√	√	√	√	x	x	x
Scanning	√	√[3]	√	√	√	x	√	x	x
Point-to-Point	√	√	√	√	√	√	√	√	√

Keywords: BLE (Bluetooth low energy), AN (ANT), A+ (ANT+), ZiB (ZigBee), RF (RF4CE), WF (Wi-Fi), Ni (Nike+), Ir (IrDA)

Notes:

1 Not simply broadcasting, it also needs to listen.

2 An application can be put on LE to empower meshing.

3 All associations stop and power consumption is high

Power Efficiency

It is often queried by customers who are prolonging the battery life of their devices while still achieving good user experience. The quality derived is the power per bit measurement [15]. Energy per unit for comparing the power efficiency of various technology is listed below:

Energy per bit

- ANT – 0.71 uJ/bit
- Bluetooth low energy – 0.153 uJ/bit
- IrDA – 48.2 uJ/bit
- Nike+ – 2.48 uJ/bit
- Wi-Fi – 0.0525 uJ/bit
- Zigbee – 185.9 uJ/bit

From the above results, it is clear that Wi-Fi is the most power efficient technology.

Latency

It can be characterized by a client activity sent to a receiving device [15]. It is basic in applications, for example, HID (mice and keyboards), security devices, sports and fitness (instantaneous body readings). Some of the typical latencies are listed below:

- ANT ~ zero
- Wi-Fi ~ 1.5 ms
- LE ~ 2.5 ms
- NFC ~ polled typically every second
- Nike+ ~ 1 second
- ZigBee ~ 20 ms
- IrDA ~ 25 ms

Due to the possibility of low latencies, ANT and Wi-Fi utilize considerable power to listen continuously to the receiving device.

Range

The range of wireless technology is proportional to the power of a transmitter and the radio frequency (RF) sensitivity of a receiver. The following list shows the ranges that can be expected for ultra-low powered technologies in an open environment [15]

- NFC – 5 cm
- IrDA – 10 cm
- Nike+ – 10 m
- ANT+ – 30 m
- Zigbee – 100 m
- RF4CE based on ZigBee – 100m
- Wi-Fi – 150 m
- LE – 280 m

Throughput

It can be measured in two ways:

- On air signaling rate
- Measuring how rapidly helpful payload information can be transferred

The following data shows how different technologies payload throughputs compare:

- RF4CE (same as ZigBee)
- ANT+ ~ 20 kbps
- Nike+ ~ 272 bps
- Wi-Fi (lowest power 802.11b mode) ~ 6 Mbps
- ZigBee ~ 100 Vkbps

- NFC ~ 424 kbps
- LE ~ 305 kbps
- IrDA ~ 1 Gbps

Peak current consumption

It is a critical figure when designing long-life, low power sensor devices. Table 2 shows peak current consumption for wireless technologies which can operate from CR2032

Table 2 Peak current consumption for wireless technologies *[15]*

• IrDA peak current draw ~ 10.2 mA • ANT peak current draw ~ 17 mA • LE peak current draw ~ 12.5 mA • Nike+ peak current draw ~ 12.3 mA	CR2032 OK
• NFC ~ 50 mA • RF4CE peak current draw ~ 40 mA • Wi-Fi peak current draw ~ 116 mA (@1.8 V)	Excess current demand

Our system needs long range communication regarding kilometers, compared to FSK, LoRa technology has the much higher sensitivity which is why SemTech LoRa Wireless technology was chosen which has secure spread spectrum modulation, requires very less RF power of 100 mW and communication ranges between 15- 20 km.

2.3 Scientific paper analysis

This section analysis various scientific article, journals and papers related to weather monitoring using various technology such as ZigBee, wireless sensor network (WSN) in the tropical region and polar regions.

In 2014, P. Susmitha and G. Sowmyabala designed weather monitoring system with a pair of sensors like humidity, temperature, and gas which measures weather parameter using an LPC1768 (ARM9) microcontroller and sends data to LABVIEW using serial communication, through GSM module, SMS will be sent to the mobile. Their future scope was to add more sensors to monitor

other environmental parameters such as CO_2 and oxygen sensor, soil PH sensor including integration of Wi-Fi camera to monitor the growth of agricultural product and also to upload the collected data to web server continuously [16].

Rong-Hua Ma, Chia-Yen Lee & Yu-Hsiang Wang in 2011, proposes wireless remote weather monitoring system based on Micro-Electro-Mechanical Systems (MEMS) and Wireless Sensor Network (WSN) technologies advances involving sensors for measuring pressure, humidity, temperature, wind direction, and wind speed, integrated on a single chip [17]. Their proposed sensor array in the system comprised through different sensors and integrated through bulk micromachining technology. The signals were transmitted and received between Octobus II-A sensor nodes using WSN Technology.

Pros: The advantage of this paper was obtaining the electrical characteristics of all the sensors and analyzing the result.

Cons: It would be better if it also analysis the power consumption and weather results at the specific time.

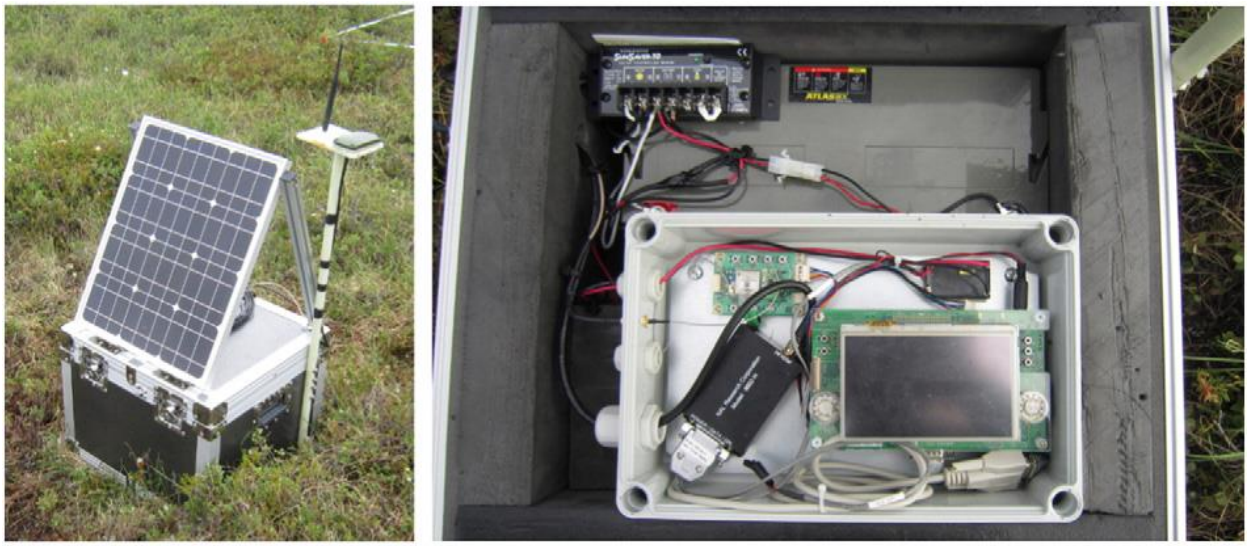
Advances in digital electronics, wireless communication and MEMS technology have enabled the improvement of low-cost, low-power, multifunctioning sensor nodes for short distance communication which are small in size [18]. Sensor networks consist of seismic, low sampling rate magnetic, thermal, visual, acoustic, radar and infrared. These can monitor temperature, vehicular movement, pressure humidity, noise levels, current characteristics such as direction, speed and size of an object, etc. for various applications such as environmental, military, health, home and other commercial applications.

Pros: Fault tolerance, production cost, operating temperature, production cost, transmission media and power consumption are the factors influenced in this sensor network design.

Cons: This survey suggests for new wireless ad hoc networking techniques for the above constraints.

Spatial variation of soil CO_2 flux was estimated in a permafrost region using ubiquitous sensor network based remote monitoring system. The temporal and spatial variations in air temperature and relative humidity near the surface and soil water content and soil temperature were measured to monitor the permafrost active layer in Alaska from September 2012 to January 2013 and from July to September 2013 [19]. The temporal variations in soil near the surface depended on the

thawing and freezing of the snow in the regions. The spatial pattern of soil temperature in three periods various because of distribution of snow coverage and depth of winter. Figure 10 shows the installation of the gateway for (a) complete assembly and (b) internal components



(a) (b)

Figure 10 Installation of the gateway for (a) complete assembly and (b) internal components [19]

Pros: The advantage of this system is long battery life, light weight, high spatial resolution, easy deployment and the need for minimal human interventions. This system helps to measure soil temperature and soil water content, relative humidity and air temperature in the different location. Therefore, this method is suitable for observation of described parameters in vast areas without electricity and communication infrastructure, which also monitor seasonal, annual variations in continuous and real-time data.

Cons: Unsuitable for long-term monitoring of annual change and it would be better if it supports for irrigation in the agricultural field.

In [20], the authors were designed and implement inexpensive wireless portable weather monitoring station using PIC16F887 microcontroller in order to overcome the requirement of skilled labor for maintenance and life cycle cost. This system equipped with sensors for measuring relative humidity, rainfall, solar radiation, wind direction, wind speed, surface temperature and atmospheric pressure. They used Modbus communication protocol for real-time measurements to the base station over wired and wireless interfaces. Figure 11 shows the advanced hardware for weather monitoring station in this system

Figure 11 Installed wireless weather monitoring station [20]

Pros: The advantage of this scheme is reducing the manual operation to reduce the life cycle cost of the station.

Cons: It measures all the parameters every 30 seconds from massive data logger to store the data and uploading the obtained results into the server without analyzing errors, power consumption and time drift per day are not taken into consideration.

Two field tests were conducted to verify its performance and functionalities, (1) environmental monitoring on tropical horticulture cultivation in Yogyakarta, Indonesia. (2) Implementation of the monitoring and control for programmed drip watering system control based on soil moisture content for tomato [21]. The supporting features of LSM and GSM framework are offline and online environmental monitoring, offline management, synchronization of system configuration and actuation. The first test gives the sustainability of unstable network connection over 80% availability of data with local offline measurement and second test result supports the real-time monitoring and control of soil moisture as well as flexibility adjustment of the system configuration remotely. Figure 12 shows the system architecture of the developed remote environment control and monitoring with global and local management subsystem implemented for tropical horticulture cultivation.

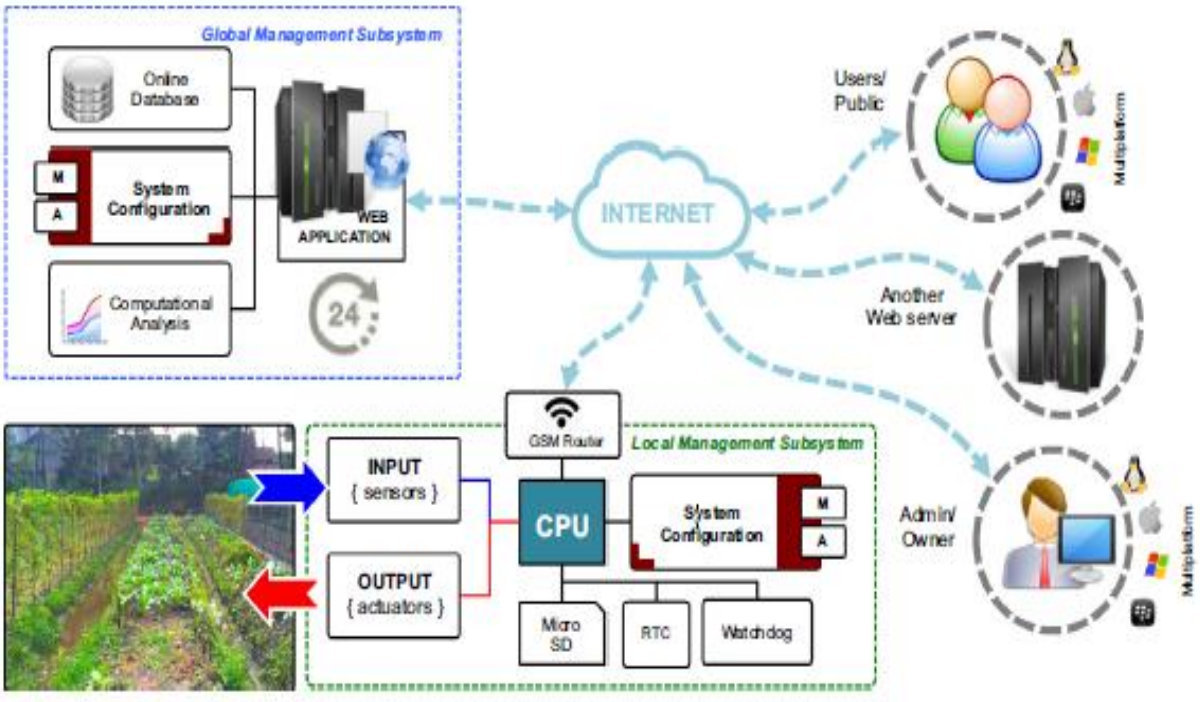


Figure 12 System architecture of local and global management subsystem for tropical horticulture [21]

Pros: The authors concluded that the above framework might become a useful tool for a simple remote environmental monitoring and to control under unstable network connection in the rural area [21]. This system aimed for long-term environmental monitoring and controlling local facilities for the possibility to be adopted in cloud-based tropical horticulture[21].

Cons: The future scope is to improve the functionality and sustainability on long-term additional power sources or energy scavenging like PV system, or portable UPS might help on improving the performance and reducing the missing data.

Real-time weather monitoring device is presented [22] to monitor the real-time temperature, relative humidity, atmospheric pressure and dew point temperature of the atmosphere through GSM network using digital and analog components. The analog outputs of the sensors are connected to an MCU through ADC. An LCD displays the measurement results. For analysis and archiving purposes, the information can be exchanged over GSM. Received data is further processed to generate graphical display using weather modeling algorithms. The pros of this system are smaller in size, low cost, portable, on-device display and robust. It transmits data to a wireless receiver board connected to RS-232 port. Figure 13 shows the assembly of weather monitoring system.

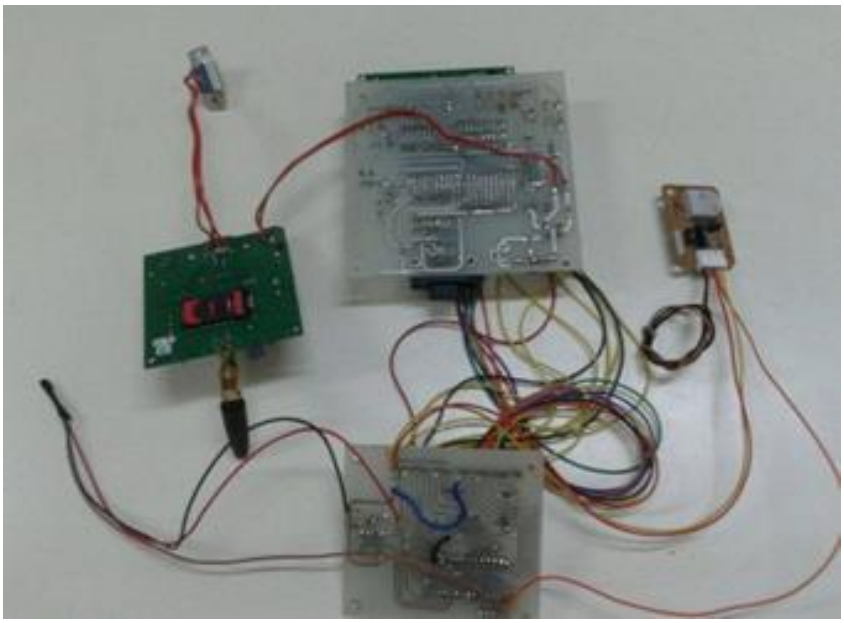


Figure 13 Assembly of weather monitoring device [22]

Cons: This system collects and sends the weather parameters, there are no details of storing, power consumption and time drift. It would be better for taking measurements of soil water content and wind temperature and speed.

World Agricultural Outlook Board (WAOB), employs an agricultural meteorologist whose mission is to monitor and survey the effects of weather on crops in the major growing areas throughout the world [23]. In precision agriculture, real nature of soil moisture and soil temperature is required for good yielding. The combination of two separate processes, water, is lost from the crop by transpiration and the soil surface by evaporation is referred to as evapotranspiration (ET) [24]. The weather parameters affecting evapotranspiration are solar radiation, air temperature, humidity and wind speed [24].

The wind direction is an angle measurement in a clockwise direction between the true north direction and direction of the wind movement [20] in degrees. The wind speed is defined as the speed at which air particles moves in the atmosphere measured in kmph [20]. The wind vane sensor comprises of pointer system, and its shaft is appended to the continuous free pivoting potentiometer. As the wind alters its direction, measuring the changes in output voltage of wind vane we can estimate the wind direction. The motion of air near the earth surface is affected by the roughness of ground, surface type, heat sources, trees, etc. The wind speed normally increases with height above the ground surface. It is affected by Rossby waves, pressure gradient, jet streams and nearby climate conditions [25].

Wireless communication refers to use of radio frequency signals for sending and receiving data between devices with wireless interfaces [26]. It also provides flexible use, cost reduction, network scaling, mobility and convenience for both users and service providers. The comprehensive guide for applications, functionality, protocols and algorithms for WSNs is provided, [27] which includes physical measurements and simulations for latency, energy consumption, bit rate, memory and lifetime.

Choosing an optimal path from a source to a destination can be done by optimizing one or more routing metrics (e.g. delay, packet loss rate, a number of hops, distance, and energy consumption) [28]. In different network paradigms and their routing unifying and distinguishing features were studied and they concluded that self-routing approach has a significant advantage in adapting to changes in the link status and connectivity for all the network paradigms.

Summary of scientific paper analysis

The scientific paper deals with design and implementation of weather monitoring system with few environmental parameters in the various field using different technology. From the above discussion, in precision agriculture, temperature, precipitation and soil moisture are the relevant parameters to understand the weather's potential impact on crop yields. Methods to measure the wind speed and wind direction were discussed. Use wind/ solar renewable energy sources for charging the internal batteries is energy efficiency and sustainable. Weather station system took three to four parameters for analyzing, no information about the time synchronization in real-time calculations, no details of power consumption and battery powered system. An automated weather monitoring system was designed to obtain the parameters such as air temperature, moisture (relative humidity), precipitation, wind direction, wind speed, air pressure, leaf wetness, soil temperature and soil moisture from different kind of sensors to overcome this disadvantage. Our system collects information and sends data every ten minutes to remote areas through low-power microcontroller and SemTech transceiver. This system considers time synchronization, time drift and power consumption.

3. Research Methodology

According to meteorological standards, Automatic Weather Station (AWS) downloads information of various weather parameters continuously in the interval of 10 minutes. In our methodology, weather monitoring system sends information to the master device and then to slave through intermediate device every 10 minutes. The intermediate device helps to increase the transmission distance. Creating the data packets for data transfer containing a command, an address of receipt, an address of hop, an address of the owner, an address of destination and payload. Each device can work as a receiver, transmitter or repeater. To get the weather data in real time, time synchronization is needed for all the devices in the network which allows each node to sleep for a given time then wake up to receive the packets. In our case, this helps the device to deep sleep for more than 580 seconds and wake up at preamble interrupt to receive packets containing the weather information and was carried out by programming Real-Time Clock (RTC). Due to variations in the LSE (32.768 kHz) oscillator connected to RTC for synchronization, time drift may occur. The time drift error can be corrected, if master broadcast time packets in 00:00:00 hr. and 12:00:00 hr. to all the remaining device. The slave and intermediate RTC will re-synchronize its time after receiving this time packets. The system uses STM32L162 microcontroller which is programmed using the IDE Keiluvision5. ULink2 JTAG debugger is used for loading program into the microcontroller. The operating frequency for the overall system is 169.4 MHz. The upcoming session deals with the parameters of the weather station, hardware and software development.

3.1 Weather Station

The data exchange part of the weather station is taken into consideration for our methodology and analyze the power consumption characteristics to optimize it. The weather station measures different parameters using different sensor type are listed in Table 3. The measured values from the sensor were combined into an array of string & decimal packets and these packets were sent to the microcontroller using RS232 through RF transceiver. The technical details of weather station and details to connect the device are described in the upcoming section. All data can be downloaded from the device by sending command ALOADDATA. List of commands and their description are described in Table 4.

3.1.1 Device and sensors

Technical details

- Device dimensions (height × width × thickness): 1700 × 900 × 300 mm
- Weight: 15 kg
- Supply voltage: 8…14 VDC
- Power consumption (quiescent state): <1 mA
- Power consumption (active state): 20…100 mA
- Data communication: RS232
- Operating temperature: -20…+50°C

Sensors

Table 3 Sensors and their specifications

Parameter	Sensor type	Range	Resolution	Accuracy
Air temperature	PT1000	-50…+50°C	0.1°C	±0.2°C
Moisture (relative humidity)	capacitive	0…100%	1%	2%
Precipitation	tipping bucket	0 mm ->	0.2 mm	
Wind direction	potentiometer	0…360°	0.1°	
Wind speed	pulse	0…30 m/s	0.1 m/s	
Air pressure*	absolute pressure sensor	150…1100 mbar	1 mbar	
Leaf wetness*	resistive	0/1 (dry/wet)	1	
Soil temperature*	semiconductor	-50…+80°C	0.1°C	±0.5°C
Soil moisture*	capacitive	0…100%	1%	

* The sensors in grey color are additional sensors which are not part of the basic setup.

3.1.2 Communications

Serial port settings:

- Baud rate: 57600 bits/s
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow control (handshaking): None

Connecting to device

Connect the weather station to a serial port using the cable which comes with the weather station. Turn off the device by pushing the power switch for 10 seconds. After reboot, following information should be seen in the terminal program:

AWS

Serial Number: 000009DBCB19

Software Version: Weather 1.04

System Time: 13:15:56

System Date: 19/05/04

After this screen, commands may be sent to the device. The commands start with a letter following a two-digit number. Commands that are used to set values are followed by a '=' sign and the new value. Queries should be followed by a '?' sign. The commands sent to the device during the first minute after the restart. The device will go to sleep after 1 minute of inactivity.

Example: changing station name

Type S05=myStation and press Enter -> device answers "OK"

Check the new name by sending S05? -> devices answers "S05=myStation" and "OK"

Data file download

All data can be downloaded from the device by sending command ALOADDATA. The device will answer first "OK", then "Xmodem file download…" The data is sent using Xmodem protocol.

Table 4 Serial commands

Command	?	=	Description	Range	Default value	Value format (when value is set)
S00	X		Get the firmware version			String (max. 31 chars)
S05	X	X	Get/set device name		AWS	String (max. 31 chars)
C00	X	X	Get/set seconds	0-59		
C01	X	X	Get/set minutes	0-59		
C02	X	X	Get/set hours	0-23		
C03	X	X	Get/set day	1-31		
C04	X	X	Get/set month	1-12		
C05	X	X	Get/set year	00-99		
F50	X	X	Get/set the threshold for the degree day calculation		5.0	Decimal
F51	X	X	Get/set the average temperature in the last 24h			Decimal

F52	X	X	Get/set the degree day in the last 24h			Decimal
F53	X	X	Get/set the accumulated degree day			Decimal
I00	X		Get relative humidity	0…100		
I01	X		Get air pressure	150…1150		
I02	X		Get wind direction	0…360		
I03	X		Get air temperature	-50…+50		
I11	X		Get leaf wetness	0/1		
I13	X		Get the accumulated precipitation	>0		
I18	X		Get wind speed	0-30		
I20	X		Get soil temperature (soil temperature sensor 1)	-20…+50		
I21	X		Get soil temperature (soil temperature sensor 2)	-20…+50		
I22	X		Get soil temperature (soil temperature sensor 3)	-20…+50		
I23	X		Get soil temperature (soil temperature sensor 4)	-20…+50		
I24	X		Get soil moisture (soil moisture sensor 1)	0…100		
I25	X		Get soil moisture (soil moisture sensor 2)	0…100		
I26	X		Get soil moisture (soil moisture sensor 3)	0…100		
I27	X		Get soil moisture (soil moisture sensor 4)	0…100		
ACCNT2			Sets the precipitation counter to zero			
ALOADDATA			Starts downloading the datafile (all data) using Xmodem protocol			
ACLRDATA			Deletes the datafile form the device			

*The red values differ from the sensor specifications (not sure which are correct).

3.2 Hardware Development

3.2.1 System prototype

The hardware is shown in Figure 14 and its system prototype are listed below:

1) The STM32L162 microcontroller is connected to the SemTech SX1276 transceiver using SPI interface.

2) A microcontroller communicates with the external device through the USB interface using RS232.

3) Battery charging circuit: a lithium battery nominal voltage of 7.2 V, Capacity 2.3 Ah.

4) Power supply unit: 12V is connected to the battery for charging it.

5) USART / USB communications protocol converter

6) The transceiver RF part is connected to the antenna through RF filter. It works in two frequency ranges 169 MHz and 868 MHz.

7) The omnidirectional antenna of λ/4 length is connected to 169 MHz for testing.

8) MCU firmware can be uploaded using JTAG or SWD interface.

9) The power supply to the entire board is 3.3 V.

10) Crystal oscillator of 20 ppm is connected to RTC_CLK for time synchronization.

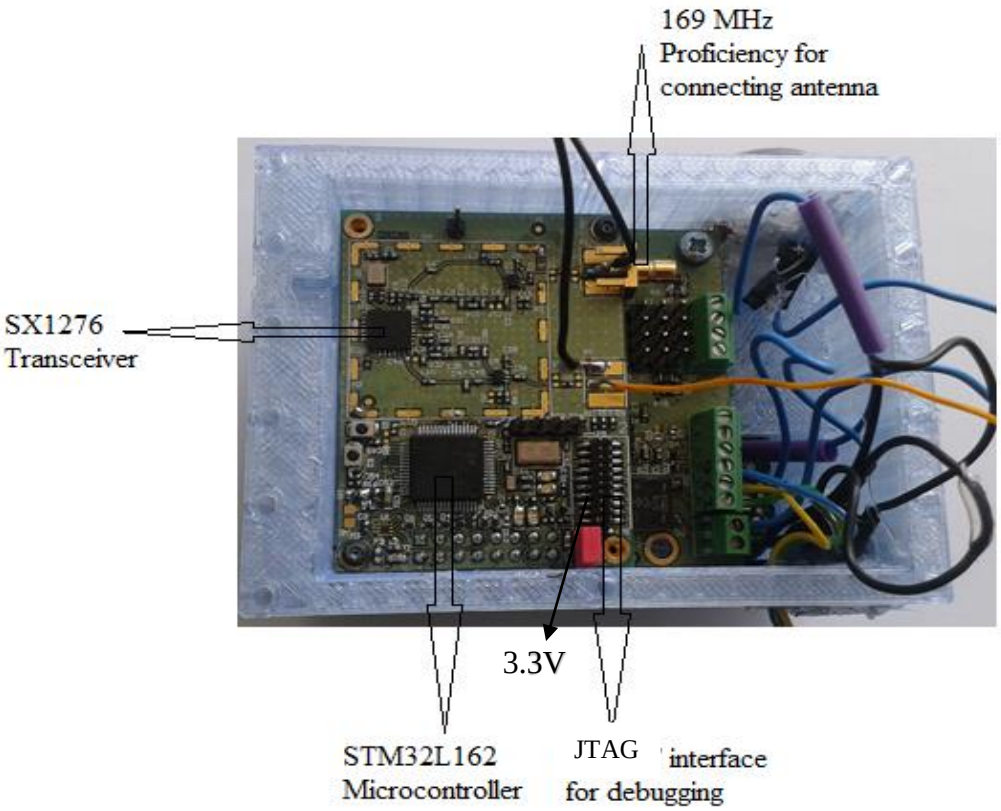


Figure 14 Hardware system prototype

3.2.2 Schematic and its description

The transceiver is connected with MCU using SPI communication interface working in full duplex mode sending and receiving data. It commands both ends in byte format. MCU firmware can be uploaded using JTAG or SWD interface. Using the same interface debugging has also been done. For communication with the external device, USART interface realizing packets communication protocol is used.

Transceiver's RF part is connected with antenna through RF filter which is intended for duplexer's building blocks to combine or separate multiple frequency bands. Thus, our devices will be filtering on the transmitted or received signals. The transceiver can work in two frequency ranges 868 MHz and 169 MHz depending on the chosen output stage and firmware options.

Two Rx/Tx multiplexers are used for splitting the receiver from the transmitter. Synchronization of the transceiver is done using external 30 MHz quartz oscillator with 20 ppm frequency stability. Additionally, MEMS accelerometer is connected to the MCU for tilt measurement possibilities. MCU is controlled by external 8 MHz quartz crystal and 32768 Hz watch crystal.

Two pushbuttons are connected to the MCU. First for reset purposes, second for loading factory settings by booting firmware (if this button is in the active state by starting MCU, factory default settings are loaded for future work). Figure 15 shows the schematic of hardware. The clear, detailed schematic is shown in Appendix VI.

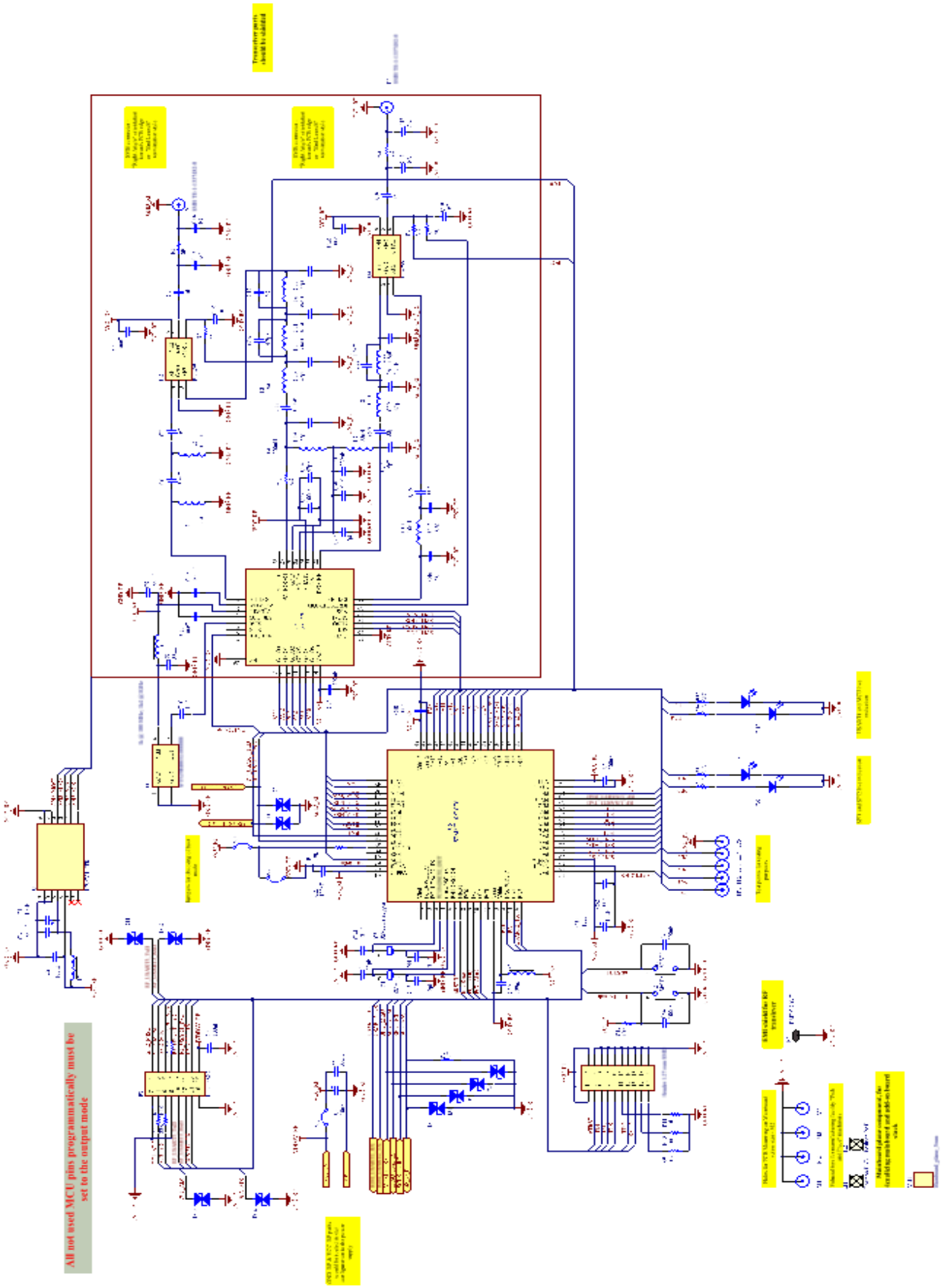


Figure 15 Schematic of hardware

3.2.3 Packets Description

The header of the packets is used for nodes synchronization between each other. Address consist of two parts – address of destination device and address of subnet in which this device works. The address consists of source and destination parts where the source address is the origination of the data and the destination address is to which the data is to be delivered. The intermediate address is used to find the route to the destination from the source. The incremental counter is used to count the number of packets received entirely at the destination side. The counter increments for every successful receiving in the source. CRC32 is used for data integrity control to find errors in whole packet content. By mistakes, the master should repeat the connection session.

The packets structures communicate between transceivers and firmware for PC are described detailed in the upcoming section.

3.2.3.1 From source to destination

The encrypted part AES128 covers net number, a unique number of the destination, subset number and i-number of hops to the destination. "i" pairs of address which gives the details about the intermediate device and subset number for i-hop. It also includes command code, data and the incremental counter which is used to increase the security level because each packet with some data content after AES128 encoding will be different. Data integrity is checked with CRC32. Randomization (data whitening) addresses the issue that radio transmission requires the data bits to alternate often, which is used for AC balancing and decreasing power consumption. Figure 16 shows the packet content sent forward from source to destination.

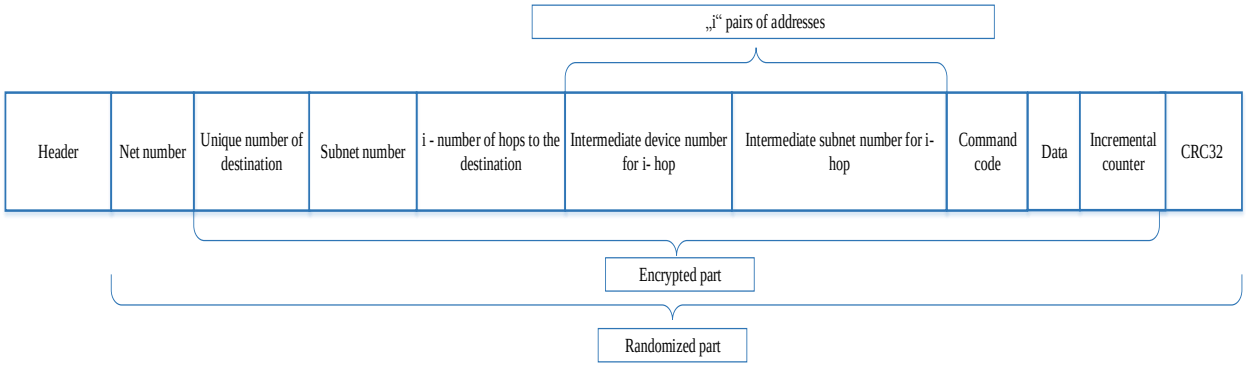


Figure 16 Packet structure sent forward from source to destination

3.2.3.2 From destination to source

The difference between the two packets is RSSI - Received Signal Strength Indicator to measure environmental conditions. For Ex.: Temperature. Figure 17 shows the packet structure received back from the destination.

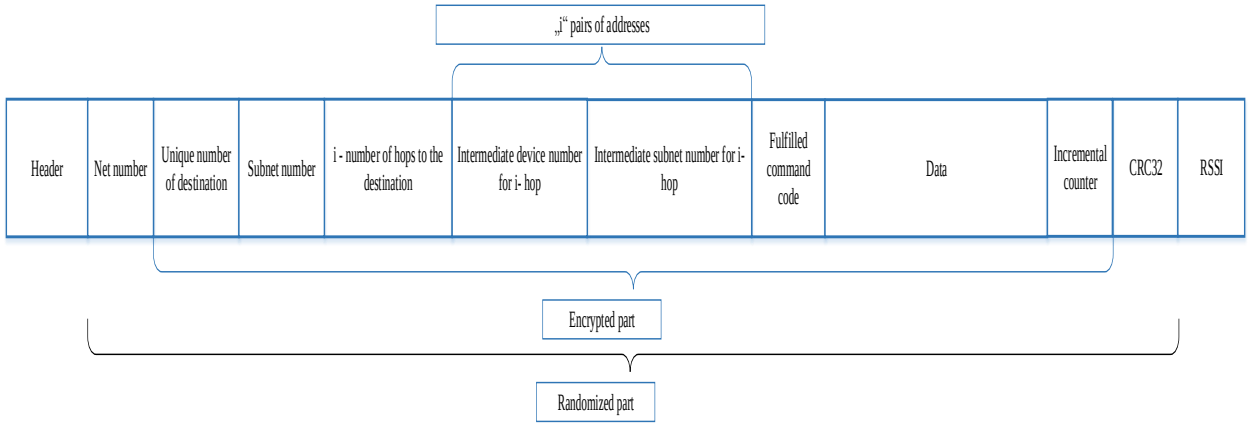


Figure 17 Packet structure received back from destination

3.2.4 Hardware preparation for testing

The transceiver is mounted in the plastic case containing chargeable batteries with an omnidirectional antenna of λ/4 length for testing purpose. Figure 18 shows the transceiver mounted in the plastic case for testing purpose with the antenna.

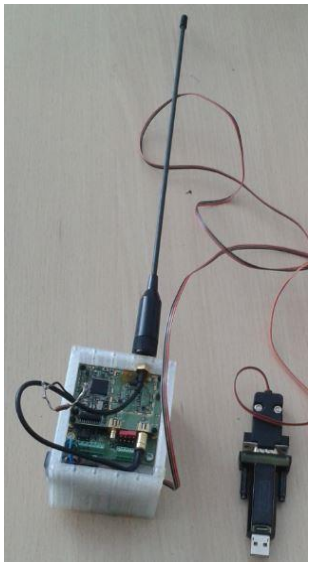


Figure 18 Transceiver mounted in plastic case for testing purpose

3.3 Software Development

3.3.1 Packet structure

Data packets have 8 bytes of data containing a command, receipt address, home address, hope address, destination address and payload as mentioned above. These packets were used for the communication between nodes. The packet content codes are mentioned in Appendix IV.

3.3.2 Time synchronization

The crystal frequency oscillator of 20 ppm with operating temperature range from -10◦C till + 60◦C. It is connected to the LSE of Real Time Clock (RTC) for synchronization. The RTC can be synchronized to a remote clock with a high degree of precision. After reading the sub-second field (RTC_SSR or RTC_TSSSR), a calculation can be made for the precise offset between the times being maintained by the remote clock and the RTC [29]. Then it can be adjusted to eliminate offset by "shifting" its clock by a fraction of a second using RTC_SHIFTR.

Firmware functions manage the initialization of the RTC peripheral to synchronize with the computer time is performed by following steps:

After reset, the RTC domain (RTC clock source configuration, RTC registers, and RTC Backup data registers) are protected against stray write accesses. To enable access to the RTC Domain and RTC registers, proceed as follows [29]:

- Enable the Power Controller (PWR) APB1 interface clock using the RCC_APB1PeriphClockCmd() function [29];
- Enable access to RTC domain using the PWR_RTCAccessCmd() function [29];
- Select the RTC clock source using the RCC_RTCCLKConfig() function [29];
- Enable RTC Clock using the RCC_RTCCLKCmd() function [29].

LSE = 32.768 kHz is selected as a clock source for RTC, it is enabled using the RCC_LSEConfig() function and the prescalers value should be assigned according to the Table 5. The RTC initialization codes are mentioned in Appendix II.

Table 5 RTC Clock source [29]

RTCCLK Clock source	Prescalers		ck_spre
	PREDIV_A[6:0]	PREDIV_S[12:0]	
HSE_RTC = 1MHz	124 (div125)	7999 (div8000)	1 Hz
LSE = 32.768 kHz	127 (div128)	255 (div256)	1 Hz
LSI = 32 kHz(1)	127 (div128)	249 (div250)	1 Hz
LSI = 37 kHz(2)	124 (div125)	295 (div296)	1 Hz

3.3.3 Time drift and its correction

Any crystal oscillator centered at the right frequency at ambient temperature will exhibit a temperature dependence, also called "drift" [30]. In crystal oscillator of 20 ppm, the time drift ranges for -20◦ C is -7.35 sec/day and for +50◦ C is -2.27 sec/day with crystal frequency error ranging -85.05 ppm & -26.25 ppm respectively which is calculated [31]. Figure 19 shows the temperature drift of the crystal oscillator.

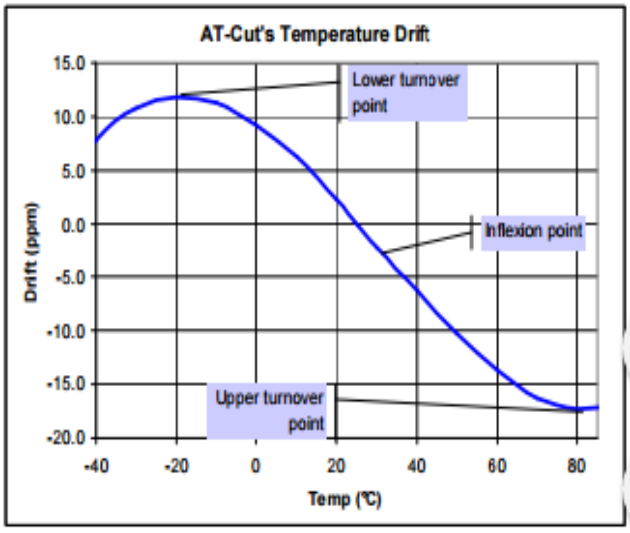


Figure 19 Temperature drift [30]

The time drift error for 20 ppm crystal oscillator with operating temperature 25◦ C, is calculated as 1.73 seconds/day [31]. To reduce the time drift error, master broadcast the time packets in two different timings 00:00:00 hr. and 12:00:00 hr. The RTC time packet codes are mentioned in Appendix V.

3.3.4 LoRa setting

LoRa calculator

This tool from st.com used to calculate the output parameters by entering the input parameters. Figure 20 shows the output values of time on air, power consumption during transmission and sleep mode current consumption. The LoRa setting codes are mentioned in Appendix III.

Input parameters

Spreading Factor = 10: 1024

Bandwidth = 0: 7.8 kHz

Coding Rate = 2: 4/6

Payload Length = 10 Bytes

RF Centre frequency = 169400000 Hz

Transmit power = 20 dBm (100 mW)

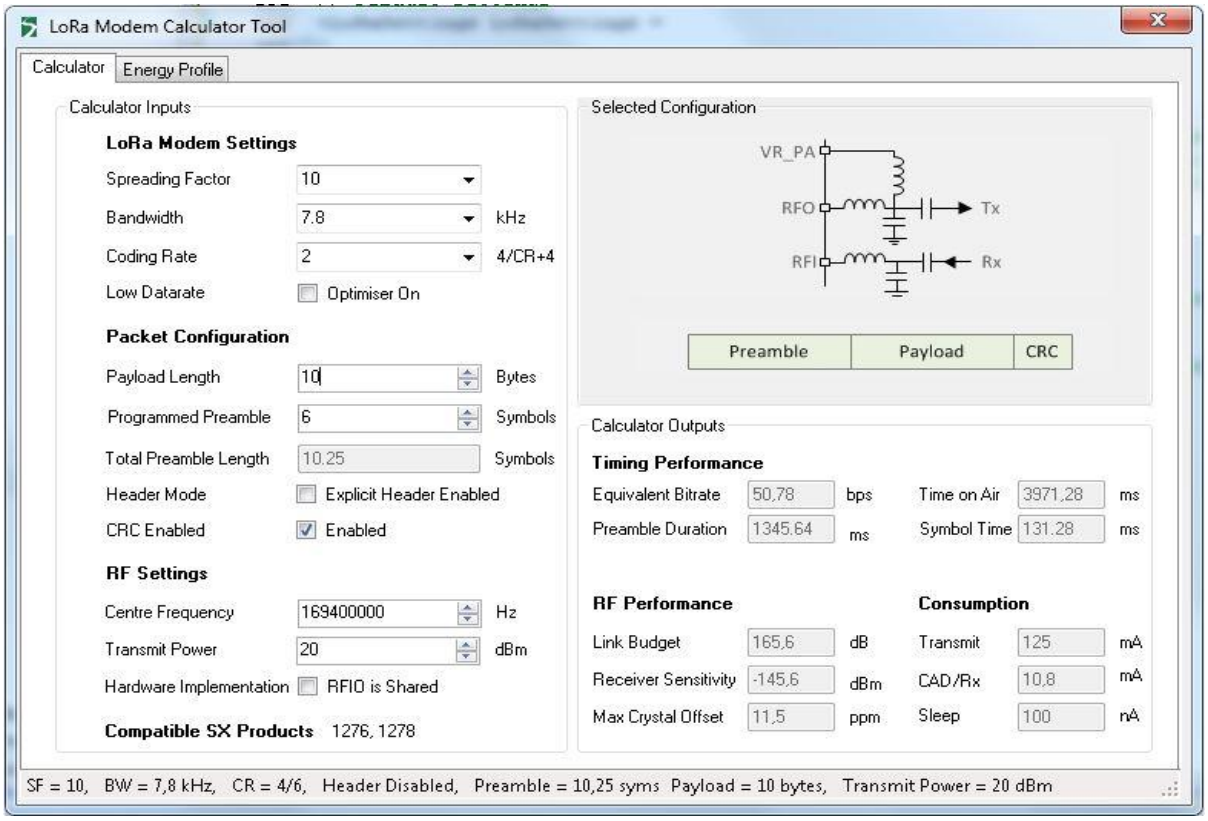


Figure 20 LoRa calculator input and output

3.3.5 Power consumption

Our primary goal is to optimize the power consumption. To achieve this, the device should be in deep sleep mode when there is no event. Total power consumption can be calculated by adding dynamic and static power consumption. The static power consumption can be calculated by knowing the current consumption of the device in sleep mode. From the STMicroelectronics site forum discussion, when the transceiver and MCU are in deep sleep mode the current consumption is 3.2uA. To know the amount of energy consumed for the entire day we have to measure the current in dynamic working mode. The current consumption during transmitting and receiving data packets has to measure.

4. Experimental Workbench

4.1 Functional block diagram

The weather station collects all the information when the command ALOADDATA sends to it. This information from weather station sends to the devices connected to the network. For optimizing the power, all the nodes should be in sleep mode for 590 seconds then it wakes up to receive the packets. Figure 21 shows the functional block diagram of proposed system.

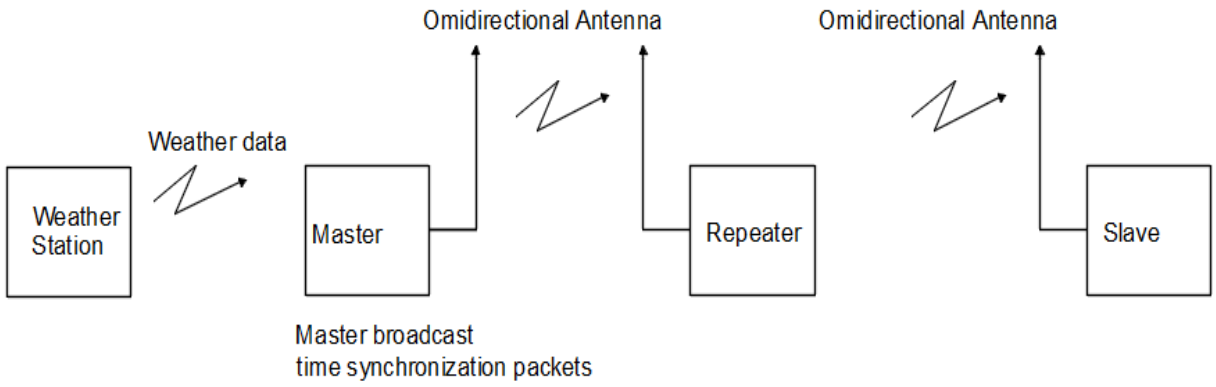


Figure 21 Functional Block Diagram

4.2 Data flow diagram

Figure 22 shows the data flow diagram of proposed system and its functionalities. The weather station collects all the parameters information from the sensors and sends to the device every 10 minutes for further needs. The packets from weather station are received by master and sends to other device connected to the network. Time synchronization is required to obtain the data packets in real time which is achieved by connecting LSE oscillator of 20 ppm to the RTC_CLK. Due to various reason as we discussed before, time drift may occur which is rectified by broadcasting two-time packets 00:00:00 hr. and 12:00:00 hr. from the master for resetting the RTC time of another device. To optimize the power, time for sleeping should increase according to our methodology.

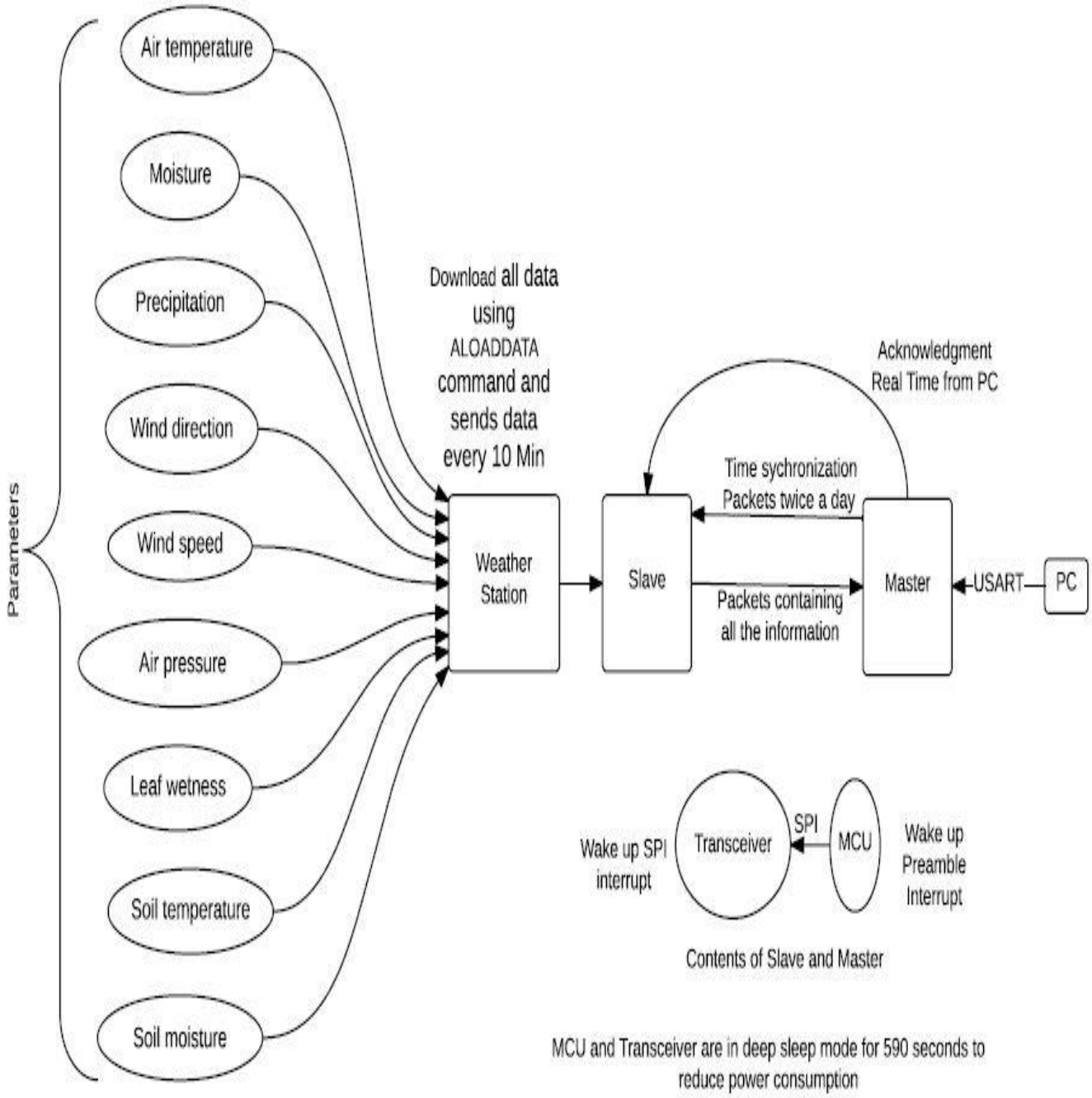


Figure 22 Data flow diagram in the system

4.3 Algorithm

All the nodes are in deep sleep mode when there is no event. This part discusses the course algorithm of the slave, intermediate (hop) and the master device connected to the network.

4.3.1 Master device algorithm

When the device starts, it waits for the signal from RTC which appears periodically every 590 secs. Additionally, if the clock of master RTC reaches either 00:00:00 hr. or 12:00:00 hr. it broadcast the time synchronization packet to all the other devices in the network. It retransmits the packet from the weather station to another device. The flowchart of this course algorithm is shown in Figure 23.

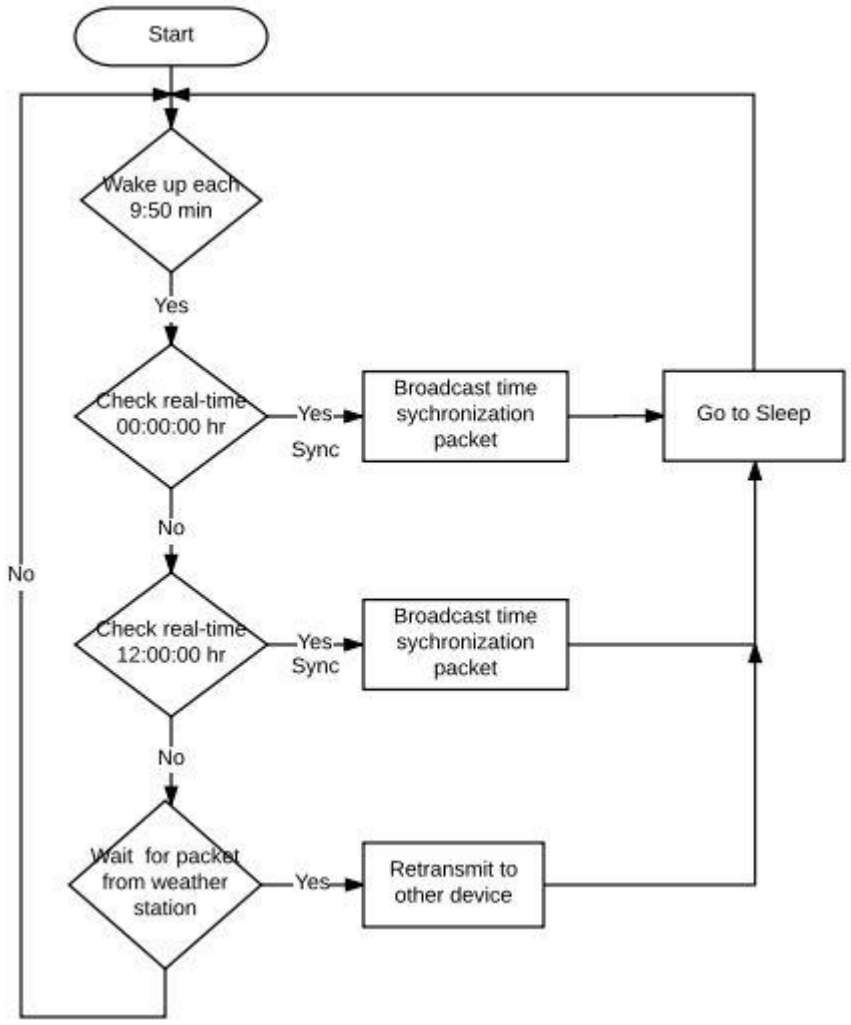


Figure 23 Master device Algorithm

4.3.2 Slave device algorithm

When the device starts, it waits for the signal from RTC which appears periodically every 590 secs. Additionally, if it receives the time synchronization packet from the master, it updates RTC clock according to the packets received. It sends the packets from weather station to the server. The flowchart of this course algorithm is shown in Figure 24.

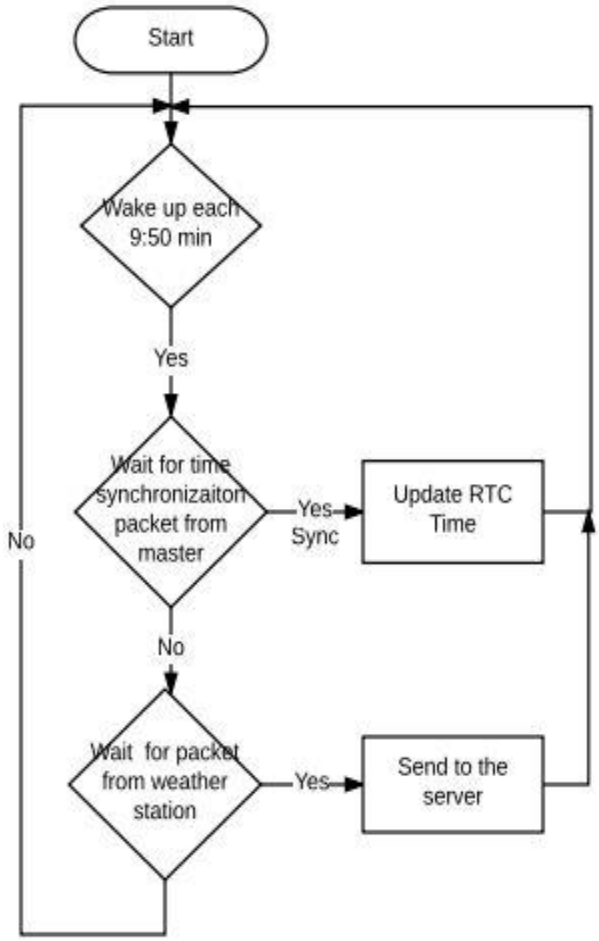


Figure 24 Slave device algorithm

4.3.3 Intermediate device algorithm

When the device starts, it waits for the signal from RTC which appears periodically every 590 secs. Additionally, if it receives the time synchronization packet from the master, it updates RTC clock according to the packets received. It resends the packets from weather station after changing address. The flowchart of this course algorithm is shown in Figure 25.

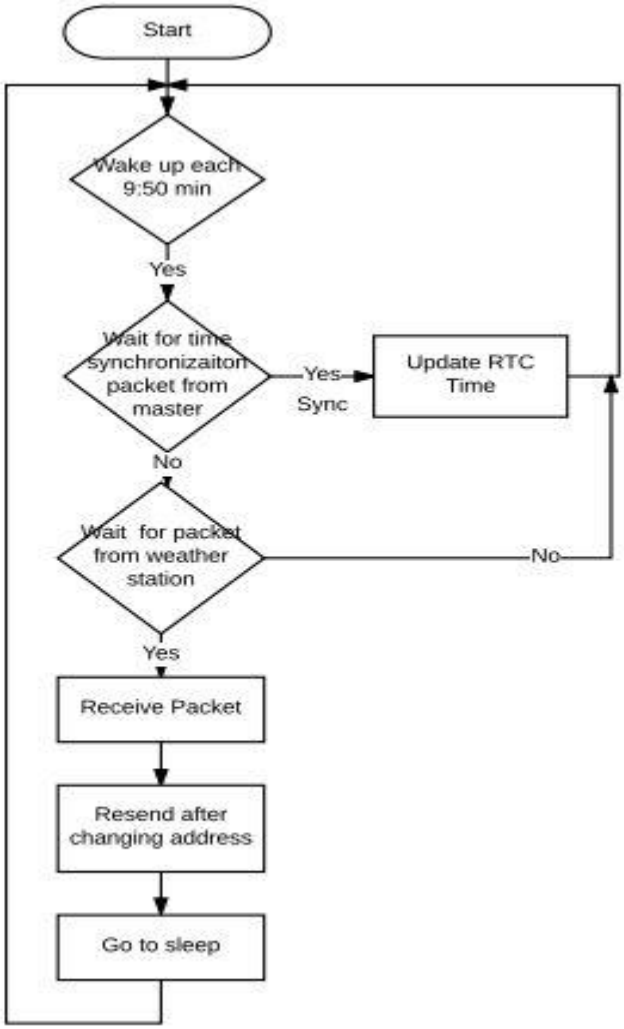


Figure 25 Intermediate device algorithm

4.4 Testing Module

The operating frequency for the overall system is 169.4 MHz. The RF transceiver power is 100 mW. The firmware contains the possibilities of changing spreading factor, bandwidth and transfer rate. The firmware developed should be downloaded to all the devices. Master connected to the PC using USART interface which sends and receives the packets. It also sends time packets for correcting the time drift in all the remaining devices. Figure 26 shows the working panel for testing purpose. The COM 1 OPEN is used to specify the connection of transceiver to the PC, which is identified by the light indication of LED. The destination address and the hop address is used to specify the address where to send the message. When we click the push button, the device started to communicate. The sended message specifies the number of times the message has been sending and the confirmed message specifies the number of times the message has been received successfully. The timeout specifies the number of seconds taken for the communication.

Usually, the message is sent successfully within ten secs without repeater while it takes 20 secs with the repeater. The timeout for transmitter without the repeater is 14 secs and with the repeater, it is about 28 secs. The Time string indicates the PC time. The LabView CVI programming code is shown in Appendix I.

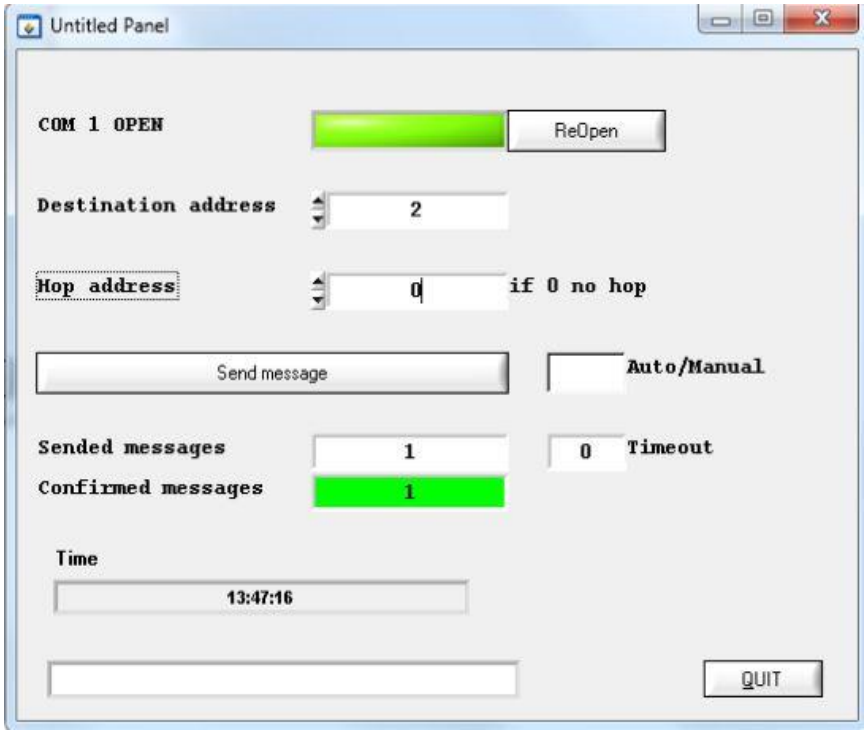


Figure 26 Working panel for testing

4.5 Experimental setup to measure current consumption

Figure 27 shows the experimental setup to measure the current consumption during receive (Rx) and transmit (Tx) mode of packets. The power supply is connected to the shunt resistor (10 Ω) serially and differential probe for voltage drop measurement on shunt connected to the oscilloscope, to measure the voltage difference. The current consumption is calculated by using the formula,

$$I = \frac{\Delta V}{R} \qquad \text{- equation (1)}$$

Where, I is current (mA),

ΔV is voltage difference (mV),

R is resistance (Ohm)

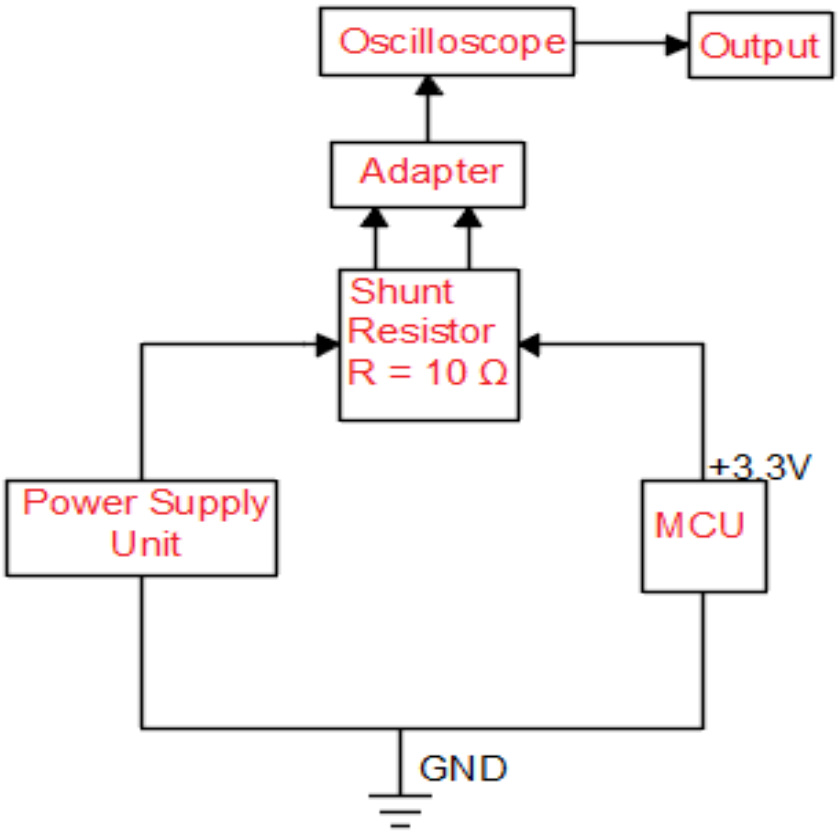


Figure 27 Setup to measure current

5. Experimental Result and Analysis

This part deals with test result for communication distance and measurement of current consumption in dynamic working modes. The first test was carried out for measuring the signal strength in the real environment and the second was performed by measuring the voltage difference in receiving and transmitting modes of data packets.

5.1 Communication distance test result

To analyze the transmission characteristics and to investigate the dependence of communication distance vs. bandwidth, spreading factor and transfer rate, the tools (firmware for controller and SW for PC with GPS data mapping, parameters change on both transceivers and chat possibilities) for RF169 investigation was developed. The data exchange fact from source to destination in the following places was checked using data pooling method and the distance was measured using GPS receiver and software which tracks measured coordinates to the Google map. The experiment was done with repeater and without repeater for checking the packets receiving confirmation and RF power of 100mW with the following parameters:

Spreading Factor = 10: 1024

Bandwidth = 0: 7.8 kHz

Coding Rate = 2: 4/6

Payload Length = 10 Bytes

RF Centre frequency = 169400000 Hz

Transmit power = 20 dBm (100 mW)

To achieve the maximum possible distance, lowest bandwidth is selected. The transfer rate for this parameter is about 50bps. Each test was carried by sending packets from master to slave with or without hop devices.

5.1.1 Experiment without repeater

1. The first experiment was carried out in a city environment near Studentu 50.g towards Taikos Pr. The position of master and slave is marked in Figure 28 Google map screenshot, the GPS coordinates are 54.903614, 23.958179 and 54.911169, 23.945823. The result states that the communication distance is 1.15km. The Google map screenshot is shown in Figure 28.

Figure 28 Map Studentu.g to Taikos. Pr

2. The experiment was carried out in slope region from Studentu 50.g towards nemunas river. The position of master and slave is marked in Figure 29 Google map screenshot, and its GPS coordinates are 54.903614, 23.958179 and 54.894442, 23.969782. The result states that the communication distance is 1.26km. Figure 29 shows the Google map screenshot of the testing location.

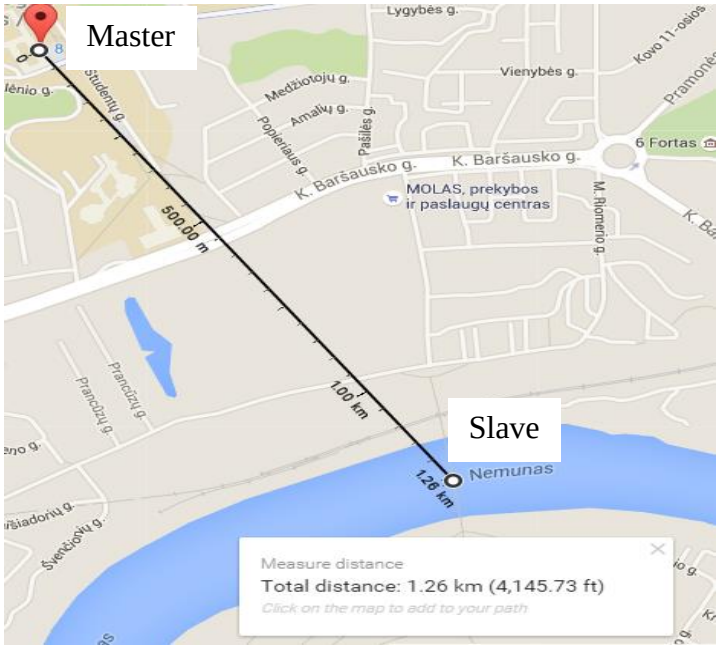


Figure 29 Map Studentu.g to Nemunas Bridge

3. The experiment was carried out near hill region from Vokiečių g. 162 towards Sąnašos g. 1where the elevation varies. The position of master and slave is marked in Figure 30 Google map screenshot, and its GPS coordinates are 54.862644, 23.942386 and 54.851628, 23.928567. The result states that the communication distance is 1.53km. The maximum height is 70m and minimum height is 17m. In this location, there are natural obstacles which are shown clearly in Figure 30 with the Google map screenshot and the elevation chart.

Figure 30 Map Vokiečių g. 162 to Sąnašos g. 1 & Elevation Chart

4. The experiment was carried out near forest region from Davalgonių g. towards Karmėlavos sen where the elevation varies. The position of master and slave is marked in Figure 31 Google map screenshot, and its GPS coordinates are 54.946870, 24.060322 and 54.955566, 24.051027. The result states that the communication distance is 1.14km. The maximum height is 74m and minimum height is 60 m. Figure 31 shows the Google map screenshot of this experimental test and its elevation chart.

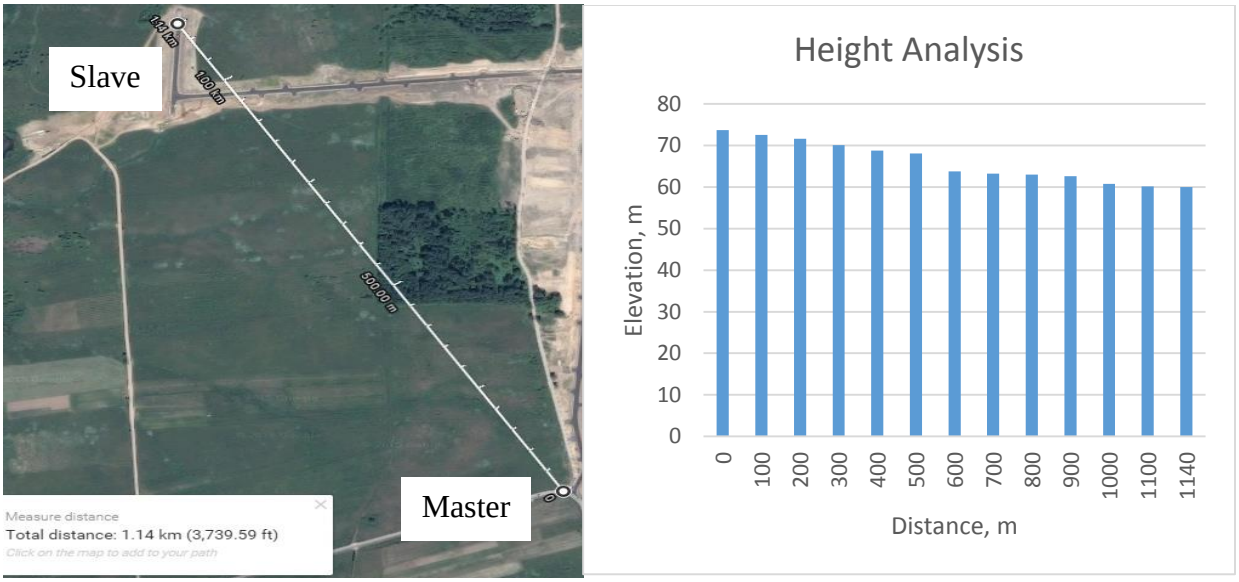


Figure 31 Map Davalgonių g. to Karmėlavos sen & Elevation chart

5. The experiment was carried out near Neveronys forest region where there is a slope. The position of master and slave is marked in Figure 32 Google map screenshot, and its GPS coordinates are 54.932273, 24.086206 and 54.940791, 24.102768. The result states that the communication distance is 1.43km. Figure 32 shows the Google map screenshot of this experimental test and its elevation chart.

Figure 32 Map Neveronys Forest Road & Elevation chart

The communication distance of the desired LoRa parameters was tested in different regions and locations in a city environment, slope region, forest region and hill region without repeater and its result was analyzed. The elevation chart was created by analyzing the height of the particular location using Google elevation calculator. To increase the communication distance in the city

environment, the experiments was carried out in four different directions surrounded by Student.g 50 with the repeater.

5.1.2 Experiment with repeater

In the city environment, the signals are disturbed by lots of obstacles like building, noise, natural obstacles like slope, etc. The research was carried out in four direction north, south, east and west. The experimental results were described in detail in the upcoming section.

Towards north

The experiment was carried out with the above-mentioned LoRa parameters, from Student.g 50 to Savanorių pr. The GPS coordinates and address of the master, repeater and slave position were listed below and marked in Figure 33. The communication distance is about 1.71 km.

Transmitter place: 54.903741, 23.958049 - Studentų g. 50, Kaunas 51368

Repeater place: 54.908468, 23.951545 - Studentų g. 19, Kaunas 50240

Receiver place: 54.914525, 23.940227 - Uosio g. 4, Kaunas 50132

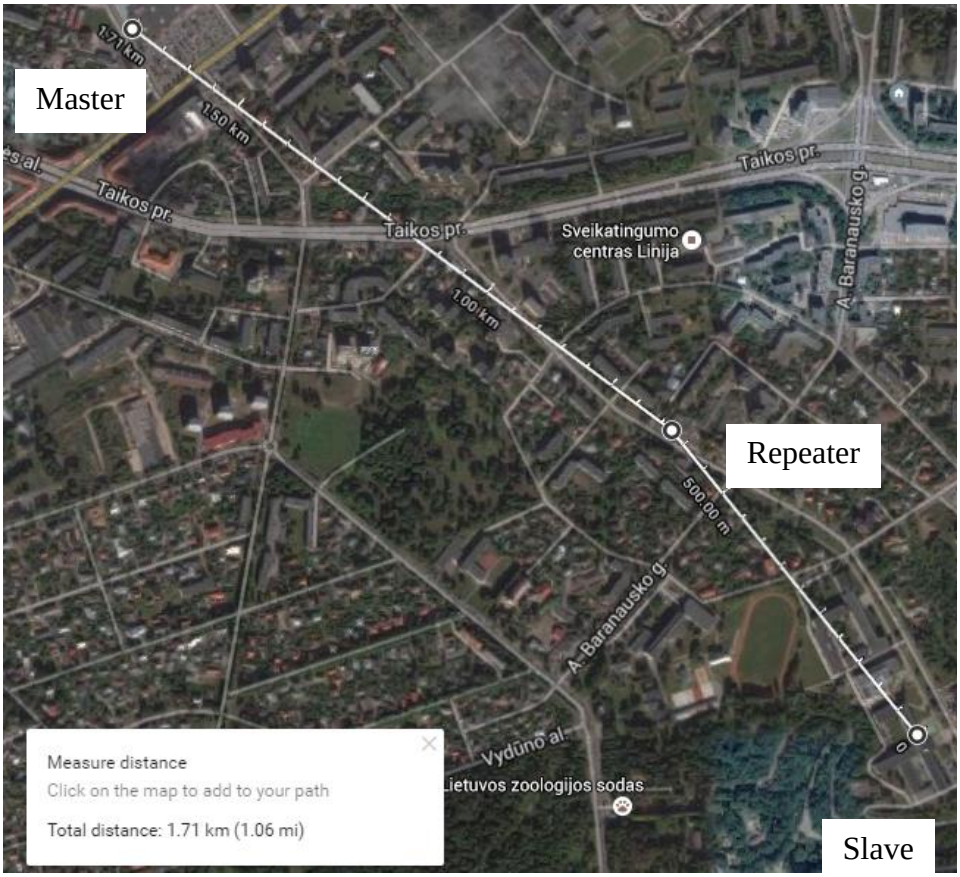


Figure 33 Map Studentu.g 50 to Savaroriu. pr

Towards south

The experiment was carried out with the above-mentioned LoRa parameters, from Student.g 50 to Molas. The GPS coordinates and address of the master, repeater and slave position were listed below and marked in Figure 34. The communication distance is about 1.22 km.

Transmitter place: 54.903741, 23.958049 - Studentų g. 50, Kaunas 51368

Repeater place: 54.900757, 23.967922 - K. Baršausko g., Kaunas 51436

Receiver place: 54.896781, 23.969107 - Biržiškio g. 1, Kaunas 51455

Figure 34 Map Studentu g. 50 to Nemunas Bridge

Towards east

The experiment was carried out with the above-mentioned LoRa parameters, from Student.g 50 towards Policija. The GPS coordinates and address of the master, repeater and slave position were listed below and marked in Figure 35. The communication distance is about 1.63 km.

Transmitter place: 54.903741, 23.958049 - Studentų g. 50, Kaunas 51368

Repeater place: 54.905250, 23.969403 - Salų g. 1, Kaunas 51353

Receiver place: 54.903598, 23.983283 - Chemijos g. 4E, Kaunas 51344

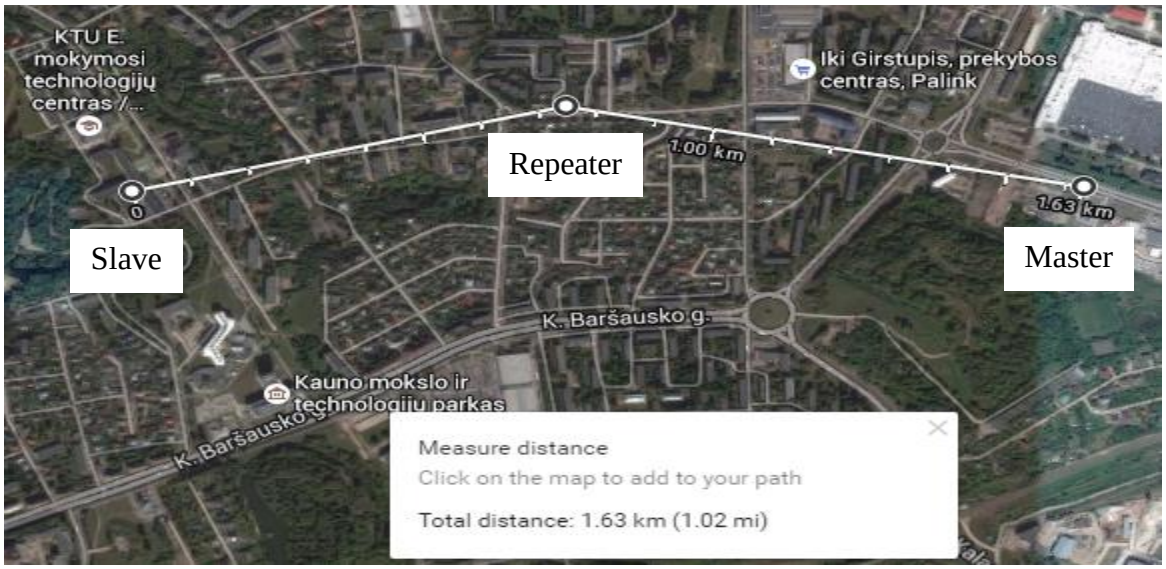


Figure 35 Map Studentu g.50 to Policija

Towards west

The experiment was carried out with the above-mentioned LoRa parameters, from Student.g 50 towards Sporto.g. The GPS coordinates and address of the master, repeater and slave position were listed below and marked in Figure 36. The communication distance is about 1.42 km.

Transmitter place: 54.903741, 23.958049 - Studentų g. 50, Kaunas 51368

Repeater place: 54.905756, 23.950596 - A. Baranausko g. 3, Kaunas 50256

Receiver place: 54.901901, 23.937679 - Gėlių Rato g. 3, Kaunas 50287

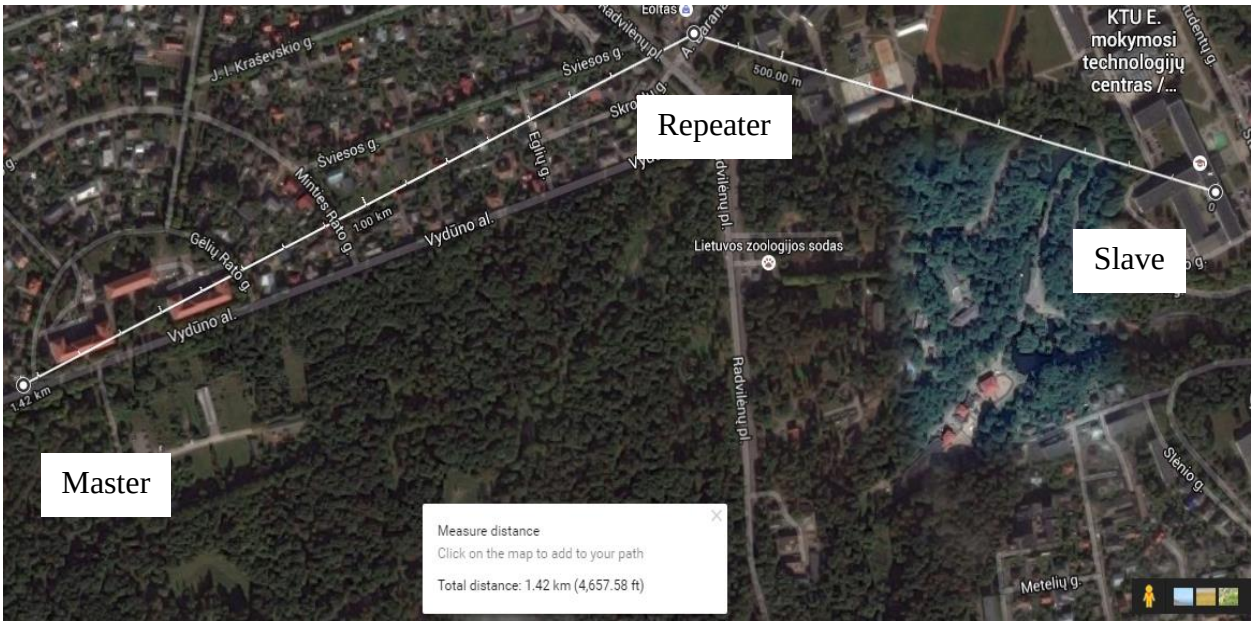


Figure 36 Map Studentu g. 50 to Geliu.g

The above experimental result shows the communication distance in a city environment near the Faculty of Electronics Engineering, Studentu 50.g. The experiment with repeater covered more distance nearly 500m compared to experiment without repeater.

5.2 Experimental result for current consumption

To measure the current consumption, resistor R= 10 Ω is connected in series with the microcontroller.

5.2.1 Receiving (Rx) mode

To measure the Rx mode current, the voltage difference during Rx mode needs to measure. Figure 37 shows the oscilloscope output screenshot of Rx mode voltage difference ΔV = 240mV on the shunt resistor measured with the oscilloscope.

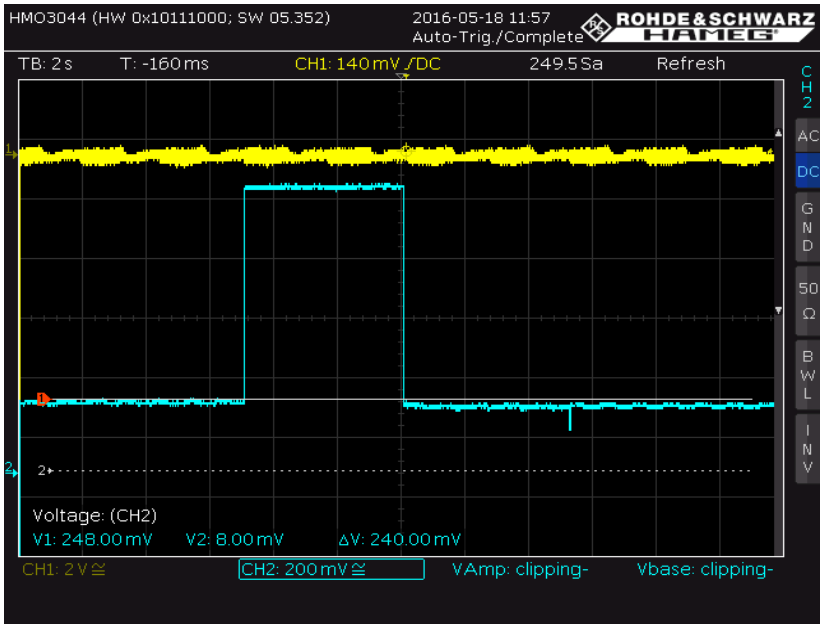


Figure 37 Rx mode ΔV = 240mV

5.2.2 Transmitting (Tx) mode

To measure the Tx mode current, the voltage difference during Tx mode needs to measure. Figure 38 shows the Tx mode voltage difference ΔV = 240mV on the shunt resistor measured with the oscilloscope.

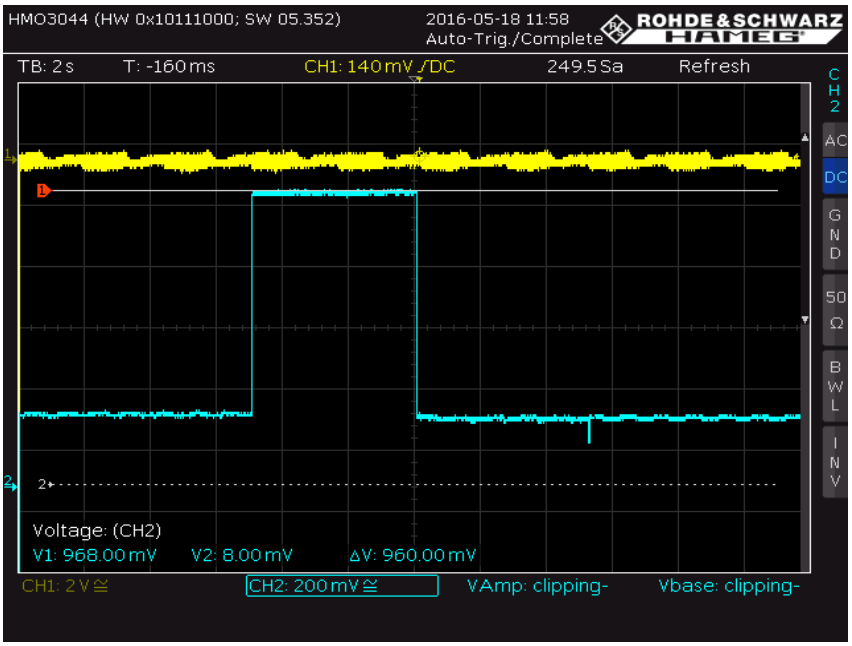


Figure 38 Tx mode ΔV = 960mV

5.3 Timing diagram

From the Figure 37 & 38, the voltage difference (ΔV) for Rx and Tx mode is 240 mV and 960 mV respectively, and resistor R= 10 Ω, substituting these values in equation (1), we get current consumption during Rx mode = 24 mA and Tx mode = 96 mA.

Figure 39 shows the timing diagram of 3 phases on which we can see current consumption in sleep, receiving and transmitting modes.

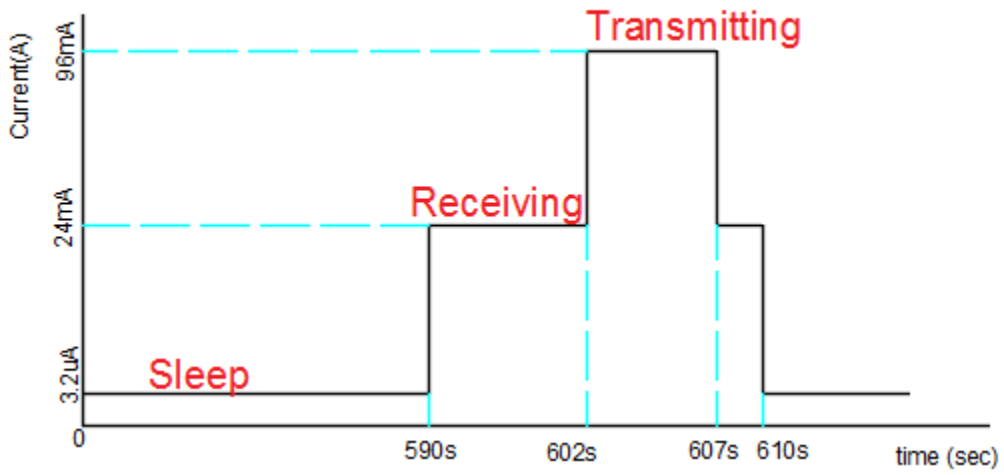


Figure 39 Timing diagram

5.4 Analysis

The power consumption is calculated by using the formula,

$$P = V * I \qquad \text{- equation (2)}$$

where, P is Power (W)

V is Voltage (V)

I is current (A)

From the timing diagram, we can find the integral of energy consumption per day using equation (2). Here V = 3.3V, the results are shown in Table 6

Table 6 Energy Consumption per day

Mode	Current (A)	mW/10min	mW/h	mW/day
Sleep	3.2u	0.01045	0.0173	0.00007
Receiving	24m	1.98	0.33	0.01375
Transmitting	96m	2.64	0.44	0.01833

Energy consumption for synchronization broadcasting packets are 2.64 mW/10min, 0.44 mW/h, 0.0366 mW/12h and 0.01830 mW/day

5.4.1 Calculation of energy consumption per day

Sleeping mode energy consumption per day = 0.00007 mW/day

Receiving mode energy consumption per day = 0.01375 mW/day

Transmitting mode energy consumption per day = 0.01833 mW/day

Synchronization packets energy consumption per day = 0.01830 mW/day

Total energy consumption per day = 0.05045 mW/day

The total energy consumption per day is approximately equal to 0.05 mW/day.

5.4.2 Battery life

Battery capacity that we used is 2300 mAh and load current is calculated by using the formula,

$$\text{Load current, } I = \frac{Power}{Voltage} \qquad \text{- equation (3)}$$

Energy consumed per day is approximately equal to 0.05 mW/day and voltage = 3.3 V. Substituting these values in equation (3) gives load current equals to 0.015 mA.

The Battery life can be calculated by using the formula,

$$\text{Battery Life} = \frac{\text{Battery Capacity in Milli amps per hour}}{\text{Load Current in Mill amps}} \times 0.70^{*} \qquad \text{- equation (4)}$$

*The factor of 0.7 makes allowances for external factors which can affect battery life [32].

Substituting the load current and battery capacity in equation (4) gives working hours of battery. So, the battery life is 105921 hours, approx. 12 years without charging.

6. Conclusion

In this work, weather monitoring system data exchange part based on LoRa technology using low power microcontroller and SemTech SX1276 transceiver was developed. For data exchange, firmware supporting data transfer integrity, security and retransmission possibilities were developed.

The parameters affecting the increased power consumption and solution for this problem were analyzed. The experimental prototype for investigation of data transfer and power consumption characteristics were created. The experimental test of communication distance vs data transfer rate showed the maximum distance of 1.7km in the city environment.

The current consumption in three working modes: Rx mode, Tx mode and sleep mode were calculated as 24mA, 96mA and 3.2uA respectively in case of RF power 100mW.

For time synchronization between nodes, Real Time Clock (RTC) were used and special time synchronization packets were programmed for re-synchronizing the RTC time caused by oscillator time drift.

The energy consumption of the entire node per day was calculated as 0.05 mW/day. It can be concluded that by using this proposed system of increasing the time for sleeping, we can expect the node to work continuously for 12 years without charging the batteries.

References

[1] **P. M and I. C,** "Embedded weather station with remote wireless control," *TELFOR*, no. 19, pp. 297-300, 2011.

[2] Campbell Scientific, [Online]. Available: https://www.campbellsci.com/weather-climate. [Accessed 13 May 2016].

[3] Prodata Weather Station, [Online]. Available: http://www.weatherstations.co.uk.. [Accessed 13 May 2016].

[4] Radio-Electronics, [Online]. Available: http://www.radio-electronics.com/info/wireless/lora/lorawan-network-architecture.php. [Accessed 29 May 2016].

[5] "SemTech Corporation," [Online]. Available: http://www.semtech.com/images/datasheet/SX1276DevKit_Userguide_STD.pdf. [Accessed 25 May 2016].

[6] "Accurate power consumption estimation for STM32L1 series of ultra-low-power microcontrollers," May 2013. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/technical_article/57/8f/9e/f3/7e/d5/42/2a/DM00024152.pdf/files/DM00024152.pdf/jcr:content/translations/en.DM00024152.pdf. [Accessed 29 May 2016].

[7] "Vaisala HydroMet Automatic Weather Station MAWS201," [Online]. Available: http://www.vaisala.com/Vaisala%20Documents/Brochures%20and%20Datasheets/MAWS201-Datasheet-B211006EN.pdf. [Accessed 10 May 2016].

[8] "Wirelss weather monitoring systems," Ratnik Industries, [Online]. Available: http://www.ratnik.com/weather_station.html. [Accessed 18 May 2016].

[9] "WE900 Weather Station," Global Water, [Online]. Available: http://www.globalw.com/products/we900.html#Options . [Accessed 23 May 2016].

[10] "Products that optimize crop yield," Dacom, [Online]. Available: http://en.dacom.nl/products/ . [Accessed 23 May 2016].

[11] "Pessl Instruments GmbH Werksweg 107," Environmental Expert, [Online]. Available: http://www.environmental-expert.com/products/imetos-model-ag-cp-dd-280-weather-station-419565. [Accessed 1 June 2016].

[12] "Space Science & Engineering Center," UW-Madison AMRC & AWS, [Online]. Available: http://amrc.ssec.wisc.edu/aws/index.html. [Accessed 23 May 2016].

[13] "Space Science & Engineering Center," UW-Madison AMRC & AWS, [Online]. Available: http://amrc.ssec.wisc.edu/news/2010-May-01.html. [Accessed 28 May 2016].

[14] **I. Texas,** "AES128-AC Implementation for Encrytion and Decryption," Texas Instuments, 2009.

[15] **S. Phil**, "Comparing low-power wireless technologies," Digi-Key Electronics, 2011.

[16] **G. Sowmyabala and P. Sushmita,** "Design and Implementation of weather monitoring and controlling system," *International Jonrnal of computer applications,* vol. 97, no. 3, pp. 19-22, July 2014.

[17] **M. Rong Hua, Y. Hsiang Wang and L. Chia Yen,** "Wireless remote weather monitoring system based on MEMS technologies," *Article Sensors ISSN 1424-8220,* vol. 11, 2011.

[18] **I. F. Akyildiz, Y. Sankarasubramaniam, E. Cayirci and W. Su,** "Wireless Sensor Networks: a Survey," *Elsevier's Computer Networks,* vol. 38, no. 4, pp. 393-422, 2002.

[19] **C. Namyi, Y. Heekwon, L. Bang Yong and L. Chankil,** "Measurement of environmental parameters in polar regions based on a ubiquitous sensor network," *Cold Regions science and technology,* pp. ISSN: 0165-232X, 2015.

[20] **J. T. Devaraju, K. R. Suhas, H. K. Mohana and A. P. Vijaykumar,** "Wireless Portable Microcontroller based weather monitoring station," *Measurement,* 2015-08-24.

[21] **A. P. Nugroho, T. Okayasu, H. Takehiko, I. Eiji, H. Yasumaru, M. Muneshi and S. Lilik,** "Development of a remote environmental monitoring and control framework for tropical horticulture and verification of its validity under unstable network connection in rural area," *Computers and Electronics in Agriculture,* 2016- 06- 03.

[22] **K. C. Gouda, V. R. Preetham and M. N. Shanmukha Swamy,** "MICROCONTROLLER BASED REAL TIME WEATHER MONITORING DEVICE WITH GSM," *International Journal of Science, Engineering and Technology Research (IJSETR),* vol. 3, no. 7, July 2014.

[23] **R. Johansson, E. Luebehusen, B. Morris, H. Shannon and S. Meyer,** "Monitoring the impacts of weather and climate extremes on global agricultural production," *Weather and Climate Extremes,* 2015-09-04.

[24] **G. A. Richard, S. P. Luis, R. Dirk and S. Martin,** "Crop evapotranspiration - Guidelines for computing crop water requirements - FAO Irrigation and drainage," *ISBN 92-5-104219-5,* 1998.

[25] **G. R. Ackermann,** "Means and standard deviations of horizontal wind components," *Journal of Climate & Applied Meteorology,* pp. 959-961, 1983.

[26] **S. Basagni, S. Giordano and I. Stojmenovic,** "Mobile Ad Hoc Networking.: IEEE PRESS," John Wiley & Sons, Inc.,, 2004.

[27] **K. Mauri, M. Kohvakka, J. Suhonen, P. Hamalainen, M. Hannikainen and D. H. Timo,** "Ultra-Low Energy Wireless Sensor Networks in Practice: Theory, Realization and Deployment," John Wiley & Sons, Ltd, Finland, 2007.

[28] **G. Takahara, S. Abdel Hamid and H. Hassanein,** "Routing for Wireless Muti Hop Network Unifying and Distinguishing Features," Kingston, Ontario, Canada, K7L 3N6 December 2011.

[29] STMicroelectronics, "STMicroelectronics; RM0038 Reference manual," July 2015. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/reference_manual/cc/f9/93/b2/f0/82/42/57/CD00240193.pdf/files/CD00240193.pdf/jcr:content/translations/en.CD00240193.pdf. [Accessed 30 May 2016].

[30] "ADVANCED COMMUNICATIONS & SENSING," [Online]. Available: http://www.semtech.com/images/datasheet/xo_precision_std.pdf. [Accessed 1 June 2016].

[31] Maximintegrated, [Online]. Available: https://www.maximintegrated.com/en/design/tools/calculators/product-design/rtc.cfm . [Accessed 15 June 2016].

[32] Digikey Electronics , [Online]. Available: http://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-battery-life. [Accessed 22 May 2016].

Appendices

Appendix I

LabWindows CVI Code for Testing

```
#define HOME_ADDRESS          1
#include <utility.h>
#include <ansi_c.h>
#include <rs232.h>
#include <cvirte.h>
#include <userint.h>
#include "169_echo.h"
/*-------------------------------------------------------*/
/*DEFINES        */
/*-------------------------------------------------------*/
#define COM_MESSAGE           1
#define COM_RESPONSE          2
#define COM_BUSSY             3
#define RTC_Zero              4
#define RTC_12                5
/*-------------------------------------------------------*/
/*TYPEDEFS     */
/*-------------------------------------------------------*/
typedef struct
{
char command;
char Address_recipient;
char Address_owner;
char Address_hop;
char Address_last;
char payLoad[3];
}T_169;
/*-------------------------------------------------------*/
/*VARIABLES   */
/*-------------------------------------------------------*/
unsigned int SendedPackages       = 0;
unsigned int ReceivedPackages     = 0;
int hours,minutes,seconds;
struct
{
char      COM1_OPEN;
char      TransmitionActive;
time_t    TransmitionStartTime;
}options;
/*-------------------------------------------------------*/
/*FUNCTION PROTOTYPES       */
/*-------------------------------------------------------*/
void CVICALLBACK Callback169 (int portNumber, int eventMask, void *callbackData);
int OpenCom169(void);
void StartTest(void);
/*-------------------------------------------------------*/
/*MAIN             */
/*-------------------------------------------------------*/
static int panel1;
int main (int argc, char *argv[])
{
        if (InitCVIRTE (0, argv, 0) == 0)
```

```
		return -1;		/* out of memory */
	if ((panel1 = LoadPanel (0, "169_echo.uir", PANEL)) < 0)
		return -1;
	DisplayPanel (panel1);
	if(OpenCom169())SetCtrlVal(panel1,PANEL_LED,1);
	else SetCtrlAttribute (panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,1);
	//printf("%d",sizeof(T_169));
	RunUserInterface ();
	CloseCom(5);
	DiscardPanel (panel1);
	return 0;
}
int CVICALLBACK QuitCallback (int panel, int control, int event,
		void *callbackData, int eventData1, int eventData2)
{
	switch (event)
	{
		case EVENT_COMMIT:
			QuitUserInterface (1);
			break;
	}
	return 0;
}
/*---------------------------------------------------------------------------
Send message over radio
---------------------------------------------------------------------------*/
int CVICALLBACK C_send (int panel, int control, int event,
		void *callbackData, int eventData1, int eventData2)
{
	switch (event)
	{
		case EVENT_COMMIT:
			StartTest();

			break;
	}
	return 0;
}
void StartTest(void)
{
T_169 message = {0};
	SetCtrlVal(panel1,PANEL_NUMERIC_3,++SendedPackages);
	SetCtrlAttribute (panel1,PANEL_NUMERIC_4,ATTR_TEXT_BGCOLOR ,VAL_YELLOW);
	message.command 				= COM_MESSAGE;
	message.Address_owner   = HOME_ADDRESS;
	GetCtrlVal(panel1,PANEL_NUMERIC,&message.Address_recipient);
	GetCtrlVal(panel1,PANEL_NUMERIC_2,&message.Address_hop);
	SetCtrlAttribute (panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,1);
	ComWrt(5,(char*)&message,sizeof(T_169));
	options.TransmitionStartTime = time(NULL);
	options.TransmitionActive = 1;
	SetCtrlVal(panel1,PANEL_STRING,"");
}
/*---------------------------------------------------------------------------
Open COM Controller
---------------------------------------------------------------------------*/
int OpenCom169(void)
{
```

```
	CloseCom(9);
	options.COM1_OPEN = 0;
	if(OpenComConfig (5,"COM5",9600,0,8,1,4096,4096)< 0)return 0;
	if(InstallComCallback (5,LWRS_RECEIVE,sizeof(T_169),'\n',Callback169,NULL) < 0) return 0;
	SetComTime (5,0.3);
	FlushInQ (5);
	options.COM1_OPEN = 1;
return 1;
}
/*---------------------------------------------------------------------
COM callback
---------------------------------------------------------------------*/
void CVICALLBACK Callback169 (int portNumber, int eventMask, void *callbackData)
{
T_169 message = {0};
	switch(eventMask)
	{
		case LWRS_RXCHAR:
			break;
		case LWRS_RXFLAG:
			break;
		case LWRS_TXEMPTY:
			break;
		case LWRS_RECEIVE:
			if(ComRd(portNumber,(char*)&message,sizeof(T_169)) == sizeof(T_169))
			{
			switch(message.command)
			{
			case COM_MESSAGE:
				SetCtrlAttribute (panel1,PANEL_NUMERIC_4,ATTR_TEXT_BGCOLOR
,VAL_GREEN);
				SetCtrlVal(panel1,PANEL_NUMERIC_4,++ReceivedPackages);
				SetCtrlAttribute (panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,0);
			break;
			case COM_RESPONSE:
				SetCtrlAttribute (panel1,PANEL_NUMERIC_4,ATTR_TEXT_BGCOLOR
,VAL_GREEN);
				SetCtrlVal(panel1,PANEL_NUMERIC_4,++ReceivedPackages);
				SetCtrlAttribute (panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,0);
				options.TransmitionActive = 0;
			break;
			case COM_BUSSY:
				SetCtrlVal(panel1,PANEL_STRING,"Radio module bussy");
			break;
			}
			}
			break;
		default: break;
	}
}
/*---------------------------------------------------------------------
reopen com
---------------------------------------------------------------------*/
int CVICALLBACK C_openCom (int panel, int control, int event,
		void *callbackData, int eventData1, int eventData2)
{
	switch (event)
	{
```

```
		case EVENT_COMMIT:
			if(OpenCom169())
			{
				SetCtrlVal(panel1,PANEL_LED,1);
				SetCtrlAttribute (panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,0);
			}
			else
			{
				SetCtrlVal(panel1,PANEL_LED,0);
				SetCtrlAttribute (panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,1);
			}
			options.TransmitionActive		= 0;
			options.TransmitionStartTime	= 0;

			break;
	}
	return 0;
}
/*---------------------------------------------------------------------
Timer callback function Interval 1s
---------------------------------------------------------------------*/
int CVICALLBACK C_timer (int panel, int control, int event,
		void *callbackData, int eventData1, int eventData2)
{
int	autoManual	= 0;
int	buttonDim	= 0;
char	Timeout		= 0;
int t;
unsigned char str[9];
T_169 message ={0};
	switch (event)
	{
		case EVENT_TIMER_TICK:
			GetSystemTime (&hours, &minutes, &seconds);
			if((hours==0) && (minutes==0) && (seconds==0))
				//if((seconds%10)==0)
			  {
				message.command				= COM_MESSAGE;
				message.Address_owner	= HOME_ADDRESS;
				message.payLoad[0]			= RTC_Zero;
				GetCtrlVal(panel1,PANEL_NUMERIC,&message.Address_recipient);
				GetCtrlVal(panel1,PANEL_NUMERIC_2,&message.Address_hop);
				SetCtrlAttribute
(panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,1);
				ComWrt(5,(char*)&message,sizeof(T_169));
			  }

			if((hours==12) && (minutes==0) && (seconds==0))
			{
				message.command				= COM_MESSAGE;
				message.Address_owner	= HOME_ADDRESS;
				message.payLoad[0]			= RTC_12;
				GetCtrlVal(panel1,PANEL_NUMERIC,&message.Address_recipient);
				GetCtrlVal(panel1,PANEL_NUMERIC_2,&message.Address_hop);
				SetCtrlAttribute
(panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,1);
				ComWrt(5,(char*)&message,sizeof(T_169));
			}
```

```
                        sprintf(str,"%2d:%2d:%2d",hours,minutes,seconds);
                        SetCtrlVal(panel1,PANEL_STRING_2,str);

                        GetCtrlVal(panel1,PANEL_NUMERIC_2,&Timeout);
                        if(Timeout)Timeout = 4*7;
                        else    Timeout = 2*7;

                        if((options.TransmitionActive)&&((time(NULL) - options.TransmitionStartTime) >
Timeout))
                        {
                                options.TransmitionActive = 0;
                                SetCtrlAttribute (panel1,PANEL_NUMERIC_4,ATTR_TEXT_BGCOLOR
,VAL_RED);
                                SetCtrlAttribute (panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED ,0);
                        }
                        GetCtrlVal(panel1,PANEL_CHECKBOX,&autoManual);
                        GetCtrlAttribute (panel1,PANEL_COMMANDBUTTON,ATTR_DIMMED
,&buttonDim);
                        if(autoManual && (!buttonDim))
                        {
                                StartTest();
                        }
                        if(!options.TransmitionActive)
                        {
                                SetCtrlVal(panel1,PANEL_NUMERIC_5,0);
                        }
                        else
                        {
                                SetCtrlVal(panel1,PANEL_NUMERIC_5,(time(NULL) -
options.TransmitionStartTime));
                        }
                        break;
        }
        return 0;
}
```

Appendix II

RTC Initialization

```
/*-----------------------------------------------------*/
/*DEFINES*/
/*-----------------------------------------------------*/
#define RTC_Zero                                4
#define RTC_12                                  5
/* Masks Definition */
#define RTC_TR_RESERVED_MASK    ((uint32_t)0x007F7F7F)
#define RTC_INIT_MASK           ((uint32_t)0xFFFFFFFF)
#define RTC_RSF_MASK            ((uint32_t)0xFFFFFF5F)
#define INITMODE_TIMEOUT        ((uint32_t) 0x00002000)
#define SYNCHRO_TIMEOUT         ((uint32_t) 0x00008000)
RTC_TimeTypeDef RTC_TimeStruct;
RTC_InitTypeDef RTC_InitStruct;
void RTC_Initt(void);
/*-------------------------------------------------------------------------------
  Initt RTC
 *-------------------------------------------------------------------------------*/
void RTC_Initt(void)
{
RTC_WriteProtectionCmd(ENABLE);
RCC_ClocksTypeDef RCC_Clocks;
RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR, ENABLE);
PWR_RTCAccessCmd(ENABLE);
RCC_LSEConfig(RCC_LSE_ON);
RCC_RTCCLKConfig(RCC_RTCCLKSource_LSE);
RCC_RTCCLKCmd(ENABLE);
RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR, ENABLE);
RTC_InitStruct.RTC_AsynchPrediv=127;
RTC_InitStruct.RTC_SynchPrediv=255;
RTC_InitStruct.RTC_HourFormat=RTC_HourFormat_24;
RTC_RefClockCmd(ENABLE);
RTC_Init(&RTC_InitStruct);
RTC_SetTime(RTC_Format_BCD, &RTC_TimeStruct);
RCC_GetClocksFreq(&RCC_Clocks);
}
```

Appendix III

LoRa Setting

```
// Default settings
tLoRaSettings LoRaSettings =
{
    169400000,        // RFFrequency
    20,               // Power
    0,                // SignalBw [0: 7.8kHz, 1: 10.4 kHz, 2: 15.6 kHz, 3: 20.8 kHz, 4: 31.2 kHz, [Default 6, Tested 0]
                      // 5: 41.6 kHz, 6: 62.5 kHz, 7: 125 kHz, 8: 250 kHz, 9: 500 kHz, other: Reserved]
    10,               // SpreadingFactor [6: 64, 7: 128, 8: 256, 9: 512, 10: 1024, 11: 2048, 12: 4096  chips]  [Default 7, Tested 10]
    2,                // ErrorCoding [1: 4/5, 2: 4/6, 3: 4/7, 4: 4/8] [Default 2, Tested 4]
    true,             // CrcOn [0: OFF, 1: ON]
    false,            // ImplicitHeaderOn [0: OFF, 1: ON]
    1,                // RxSingleOn [0: Continuous, 1 Single]
    0,                // FreqHopOn [0: OFF, 1: ON]
    4,                // HopPeriod Hops every frequency hopping period symbols
    7000,             // TxPacketTimeout
    7000,             // RxPacketTimeout
    128,              // PayloadLength (used for implicit header mode) Default: 128
}
```

Appendix IV

Packet content

```
/*-------------------------------------------------------*/
/*TYPEDEFS                                               */
/*-------------------------------------------------------*/
typedef struct
{
char command;
char Address_recipient;
char Address_owner;
char Address_hop;
char Address_last;
char payLoad[3];
}T_169;          //Packet Structure
```

Appendix V

RTC Time packets

```
if(((T_169*)&data)->payLoad[0] == RTC_Zero)
{
RTC_TimeStruct.RTC_Hours = 0;
RTC_TimeStruct.RTC_Minutes = 0;
RTC_TimeStruct.RTC_Seconds = 0;
}
if(((T_169*)&data)->payLoad[0] == RTC_12)
{
RTC_TimeStruct.RTC_Hours = 12;
RTC_TimeStruct.RTC_Minutes = 0;
RTC_TimeStruct.RTC_Seconds = 0;
}
```

Appendix VI

Schematic diagram of transceiver