ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY
January 2020, Vol 13(03), 286 – 298
DOI: 10.17485/ijst/2020/v13i03/142755,

Low-Power Early Forest Fire Detection and Warning System

Dogan Ibrahim*

Near East University, Department of Computer Information Systems, Cyprus

Article Type: Article

Article Citation: Dogan Ibrahim. Low-power early forest fire detection and warning system. *Indian Journal of Science and Technology*. 2020; 13(03), 286-298. DOI: 10.17485/ijst/2020/v013i03/142755

Received date: March 13, 2019

Accepted date: April 2, 2019

***Author for correspondence:**
Dogan Ibrahim dogan.ibrahim@neu.edu.tr Near East University, Department of Computer Information Systems, Cyprus

Abstract

In this study, the design of a microcontroller based early forest fire detection and warning system is described. The system can be installed in a forest and it consists of a network of microcontroller based fire detection sensors and long-range RF transmitter–receiver modules. As soon as a fire is detected, the system sends warning messages to the relevant emergency command and control centres so that appropriate action can be taken as soon as possible to put off the fire and minimize the damage. **Objectives**: The objective of this study is to develop and design a low power microcontroller based early forest fire detection system. **Methods/statistical analysis**: The system described in this study consists of a network of microcontroller based fire detection sensors and long-range raw RF transmitter/receiver modules, scattered around a forest. The system uses low power microcontrollers which are kept in deep sleep mode most of the time, thus conserving power which is one of the novelties of the designed system. **Findings:** One of the aims of this study has been to design a low power early forest fire detection and warning system. The designed system makes use of specially selected microcontrollers which can operate in deep sleep modes and as a result they can be configured to consume low currents. The results showed that the current consumption in deep sleep mode was only 10 mA, which is in general much lower than the current consumption of most ordinary microcontrollers. With such low power consumption, it should be possible to operate the overall system for longer hours before the batteries need charging via solar energy cells. Most forest fire detection systems rely on Bluetooth, ZigBee, WiFi, or similar technology for communication, which is not practical to implement in a forest environment. The designed system makes use of raw RF for communication which is normally available and very reliable. **Application/improvements:** The early forest fire detection and warning system described in this article can be improved by using arrays of multiple scattered sensors.

Keywords: Forest Fire, Early Fire Detection and Warning, Early Warning System, ESP32, DevkitC.

1. Introduction

Forest fires are uncontrolled fires occurring in wild areas and costing millions of dollars in damages and also claiming many human lives as well as damaging the natural resources and the wildlife. Such fires usually have serious consequences, causing significant damage to natural resources, burning houses, and resulting in unrecoverable damage to the environment. Although it is not possible to eliminate forest fires, it is possible to detect them early and take necessary actions to minimize the damage caused by such fires.

The causes of forest fires include lightning, extreme heat and aridity, human carelessness, and deliberate attempt to cause fires. In recent years, the frequency of forest fires has increased considerably due to climate change, human activities, and other factors [1]. According to prognoses, wildfires, including fire clearing in tropical rain forest, will halve the world forest stand by the year 2030 [2]. In Europe, up to 10,000 km^2 of vegetation is destroyed by fires every year, and up to 100,000 km^2 in North America and Russia. Wildfires are responsible for approximately 20% of CO_2 emission into the atmosphere [3].

The most efficient and effective way to detect and fight forest fires is their early identification. Several techniques are currently implemented for early fire detection. The methods used for forest fire detection can be divided into three groups [4]: ground-based, aerial, and space borne detection. The ground-based systems comprise a number of observation stations (OS) installed at carefully selected locations in or around forests and linked via radio or wire cables to a Command and Control Center (CCS). The OS uses different technologies such as thermal cameras, IR detectors, flame sensors, and smoked detectors to detect fires and send information to the CCS. Traditionally, some personnel in a lookout tower located in a high point performed the monitoring tasks. This method of monitoring is still used in some countries such as in the US, Canada, and Australia (Towers).

Airborne fire detection relies on the techniques of observing the area from above, e.g. by flying over the forest areas. These systems are not very efficient as they require constant observation which is not possible in practice. Many environmental factors such as strong winds and low clouds also may prohibit flying over the forest areas.

The use of geostationary space satellites to keep an eye on the forests and detect possible early fires is another technique. Satellite detection is relatively expensive and has the disadvantage that the presence of clouds is usually an obstacle to such detection techniques. Conventional fire detection and warning systems rely on fire detectors which detect the presence of particles generated by smoke and fire. They may also monitor the ambient temperature and humidity to detect fires.

Early detection of forest fires using sensor networks is not a new topic. In proposed describes a GPRS and ZigBee-based system where the information is transmitted using FTP server. The problem in such a system is that GPRS is not normally available in a remote forest environment.

In Ref. [5], Samanta proposes an early forest fire detection system based on mesh sensor networks of temperature and humidity, light, and CO, where the sensor nodes communicate with the moving nodes deployed on the belt of forest animals, using the ZigBee technology. When animals move in the forest and come in range of stationary ZigBee wireless nodes, data will be collected and sent to the cloud for analysis.

In Ref. [6], Jamdahade et al. describe a wireless sensor network for early detection of forest fires, powered using solar energy. The proposed system is based on Atmel AT89c/s52 microcontroller where a number of semiconductor temperature sensor chips are used, scattered around the forest. Raw RF technology is used to send any warning messages to a base.

In Ref. [7], Ganesh et al. propose a solar energy powered forest fire detection system based on the Atmega microcontroller and using the ZigBee technology for communication. Fire is detected using temperature sensor, pressure sensor, and rain sensor. Authors comment that the need for rain sensing is that when rain falls in some forest areas, the areas near the forest may be subjected to floods and landslides. With the help of the rain sensor people near the forest area cam be warned of the possibility of flooding and landslides due to rain.

This study describes the design and development of a microcontroller based early forest fire detection and warning system. The system is based on a network of low-power microcontrollers that can detect fire using various sensors. The novelty of the developed system is that it is centred around the low-power ESP32 type microcontroller, and it uses raw RF for communication.

2. The Aim

The aim of the project described in this study has been:

- To develop forest fire detection system that uses the lowest possible power consumption
- To use raw RF techniques for communication since Wi-Fi, Bluetooth, or SMS may not be available in a remote forest
- To use long-range communication techniques, assuming the Command and Control Centre could be over 20 km away from the forest
- To use reliable and low-cost components

3. System Architecture

The overall system has a distributed sensor network structure and its block diagram is shown in Figure 1. The overall system consists of a number of *Slave* processors and a *Master* processor. The slave processors are scattered around the forest and they have two main functions: to detect a possible fire as soon as possible, and to inform the master processor when a fire is detected. The slave processors are therefore equipped with various sensors which are configured to detect fire. The details of both the slave processors and the master processor are given in the next sections.

3.1. The Slave Processors

The slave processors are scattered around the forest in the form of a circle with the master processor located somewhere at the central position of this circle. Figure 2 shows the block

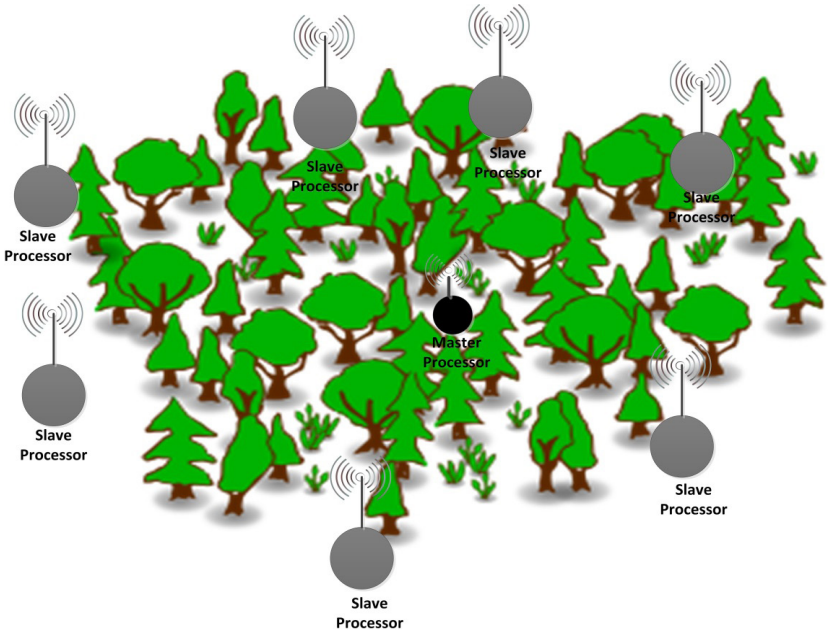

FIGURE 1. The system block diagram.

diagram of a slave processor. The slave processors are placed such that altogether they cover the complete forest area so that a possible fire starting at different parts of the forest can be detected early. The main aim here has been to make the design such that the power consumption is as low as possible. The design is based on the DevKitC microcontroller development board [8], which makes use of the ESP32 type [9] processor.

ESP32 is a low-cost and low-power processor which can be put into deep sleep mode with extremely small power consumption, and this is the main reason why this particular processor was chosen. The basic features of the ESP32 processor include the following. It is interesting to note that the deep sleep mode current consumption is only 5 μA:

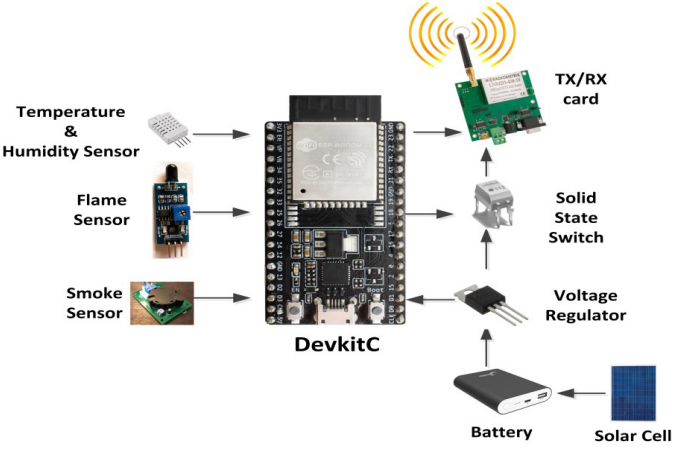

FIGURE 2. Block diagram of a slave processor.

- Xtensa dual-core 32-bit LX6 microprocessor with 160 MHz clock
- 520 KB SRAM
- Wi-Fi (802.11/b/g/n) and Bluetooth (v4.2)
- 12-bit 18 channel ADC
- 2 × 8-bit DAC
- 10 × touch sensors
- 4 × SPI, 2 × I^2S, 2 × I^2C, and 3 × UART interfaces
- CAN bus 2.0
- Infrared remote controller
- 5 μA deep sleep current

Fire is detected using several types of sensors connected to the slave processors. Again, it was aimed to use low-power components to minimize the overall power consumptions of the slave processors. The sensors detect the environmental variables such as the temperature, humidity, flame, and smoke in real-time and based on this collected data they decide whether or not fire has occurred. The details of the sensors used in the design are given in the next sections.

3.1.1. Temperature and Humidity Sensor

It is well known [10] that the two environmental variables temperature and humidity can provide further information for the early detection of fires. In a fire situation, the temperatures rises sharply. Humidity is also very important since it affects the fuel content in the air and therefore its flammability. Rising temperature results in decreasing humidity. In dry conditions, the moisture from fuels is transferred to the atmosphere and as a result they become more flammable.

There are many types of temperature and humidity sensors available. In this project, the DHT22 temperature and humidity sensor chip [11] is used (Figure 3) because it combines both the temperature and humidity sensors in a chip, and also its power consumption is low. This is low-cost and low-power sensor is made up of a capacitive humidity sensor and a thermistor. The humidity range of this sensor is 0–100%, while the temperature range is −40 °C to +125 °C. The sensor operates with 3–5 V where the maximum standby current consumption is only 40 μA. The sensor is connected to one of the digital inputs of the ESP32 processor.

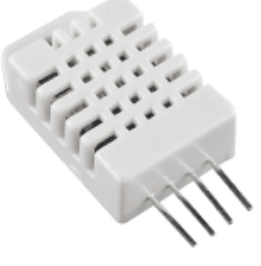

FIGURE 3. DHT22 temperature and humidity sensor.

3.1.2. Flame Sensor

A flame detector is a small sensor chip designed to detect the presence of flame or fire. When fire burns it emits infra-red light which can be detected by an IR photodiode. There are different types of flame sensors and flame detection techniques. Some examples are infrared cameras, ultraviolet detectors, infrared (IR) detectors, and so on. In this project, the YG1006 sensor type flame detector module [12] is used (Figure 4). This module basically consists of an NPN phototransistor, a resistor, capacitor, a potentiometer, and a comparator. The module provides digital logic high output in the presence of fire, and logic low output in the absence of flame or fire. The sensitivity of the sensor can be adjusted via the potentiometer. The sensor is connected to one of the digital inputs of the processor.

FIGURE 4. IR-based flame sensor module.

3.1.3. Smoke Sensor

Smoke sensors are essential parts of any fire detection system. Smoke in a fire can travel for great distances with the help of the wind. Therefore, detection of the smoke can help to detect and prevent early forest fires. There are many types of smoke sensors available that can be used with microprocessor systems. One such popular sensor is the MQ2 which can be used to detect gases such as H_2, LPG, CH_4, CO, Alcohol, Smoke, or Propane. Although this is a popular gas sensor module, it has the disadvantage that its current consumption is relatively high, i.e. around 130 mA. Since one of the aims of the system developed was to minimize the power consumption, it was decided to use a different type of gas sensor.

In this project, the DYP-ME0010 smoke detector module [13] (Figure 5) is used. This module is based on the Free scale Semiconductor MC145012DW smoke detector chip [14] and it consists of an on-board potentiometer and a relay.

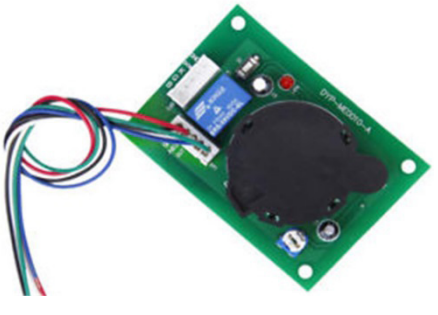

FIGURE 5. DYP-ME0010 smoke sensor module.

The potentiometer is used to adjust the sensitivity of the sensor. The reason for using this module is because its standby current is only 10 μA. The DYP-ME001 smoke detector module has the following basic specifications:

- 9 V operation
- 10 μA standby current, 30 mA working current
- Relay output
- Sensitivity adjustable

3.1.4. The Transmitter/Receiver Module

The slave processors detect the fire and communicate with the master processor to inform that fire has been detected. Because the processors are located in a forest remote from any towns or villages, there is no access to the Internet or to a Wi-Fi. In addition, the range of any Bluetooth communication is rather limited. The only other option here for communication was to use RF transmitter/receiver modems. One of the requirements has been to use professional quality reliable modems. The required range of the modem depends on the size of the forest and the size of the area to be monitored. In this project, the TX/RX modem cards from Radio metrix with the LNM2H-458-19 type [15] modem chips are used (Figure 6). These modem cards have the following specifications (different modem chips with different operating frequencies are available for different countries):

- 458 MHz (UK) TX/RX chip
- 19200 bps data rate
- ETSI EN 300 220-1 Category 1 high performance
- 5 V operation
- 500 mW transmit power
- USB, RS232. And RS485 interface
- Operation with baud rates from 300 to 38400
- Antenna connector with antenna

FIGURE 6. LNM2H-458-19 TX/RX modem card.

LNM2H-458-19 modem chips are professional grade, high performance, and programmable, and they have the very useful "Store and Forward Repeater mode" of operation where the modem can be used as a repeater and thus the range of the modem can be increased considerably using several such modems. This is very important for example, when the Command and Control Centre is a long distance away from the forest. A range test mode and received signal strength indicator are also provided which could be useful in positioning the modems.

3.1.5 Solid State Switch

The Transmiter/Receiver modem card consumes the largest power in the system. In order to minimize the power consumption in the system, this module is normally turned off using a MOSFET solid state switch under the control of the slave processor. If fire has been detected by the slave processor, then the solid state switch is activated which in turn supplies voltage to the transmitter/receiver modem card to send the early fire warning message to the master processor. A G3VM-61A1 [16] type MOSFETsolid solid state switch is used in the design, capable of switching loads with current up to 500 mA.

3.1.6. Voltage Regulator

The ESP32 processor, temperature and humidity sensor module, and the flame sensor module all operate with +3.3 V. The smoke sensor module, on the other hand, requires +9 V for its normal operation. A 9 V rechargeable battery is used to supply the required voltage to the smoke sensor module. The voltage required by other parts of the circuit is supplied using a L7805 type [17] voltage regulator chip to step down the voltage from +9 V to +5 V. The DevKitC board is supplied with this +5 V, which is then converted into +3.3 V by its internal regulator circuitry since the ESP32 processor chip operating voltage is +3.3 V. Power to the temperature and humidity sensor module and the flame sensor module are derived from the DevKitC +3.3 V output pin.

3.1.7. Slave Processor ID

Each slave processor is given a unique ID which is hardware programmed by configuring the ESP32 processor I/O pins. Currently, 8 unique IDs can be configured from 00 to 07 by configuring 3 I/O pins. The unique ID is transmitted to the master processor when fire is detected so that the exact location of the fire can easily be determined.

3.1.8. Slave Processor Antenna

In general, the locations of the slave processors and the master processor are well known. Because of this, it is possible to use directional antenna such as [18] to establish reliable long distance communications between the slave processors and the master processor in a forest environment.

3.2. The Master Processor

The master processor receives early fire detection message from the slave processors and relay this information to the Command and Control Center. The architecture of the master processor is therefore simpler and its block diagram is shown in Figure 7. The same ESP32 processor and the same TX/RX modem card are used in this design, although a more powerful TX/RX module could also be used for extended range. The LNM2H-458-19 has the capability to operate in repeater mode where several such modules can be used to extend the range if the CCS is a long way from the master processor.

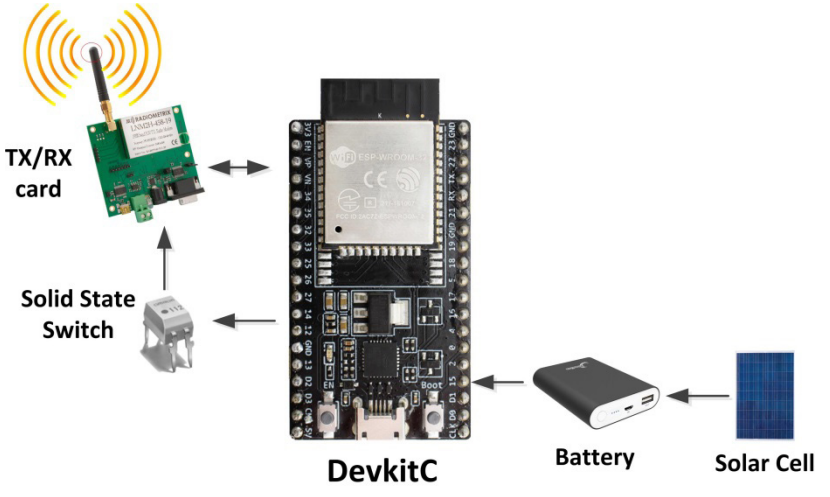

FIGURE 7. Block diagram of the master processor.

3.3. The Software

Two programs were developed for the system: the slave processor software, and the master processor software. Both programs were developed using the Arduino IDE integrated development environment [19] customized for the ESP32 processor. Serial monitor of the IDE was used during the debugging and development of the software. Details of both programs are described in the next sections in the form of Program Description Language (PDL) [20].

3.3.1. The Slave Processor Software

Power saving has been one of the main aims of the system developed in this article. As a result, the ESP32 processor is put into deep sleep mode most of the time and it wakes up at pre-defined intervals and checks the sensors to determine if fire has been detected. If fire has not been detected, then the slave processors go back into deep sleep mode. If, on the other hand, fire has been detected, then the slave processor that detected the fire sends an early fire detection warning message to the master processor. This message is repeated

several times just to make sure that it is received by the master processor. The actual repeat time and the processor deep sleep time can be selected as desired.

The relationship between the slave processor wake up time from deep sleep, message repeat count, and the master processor wake up time can be derived as follows:

Let the slave and the master processor wake up times from deep sleep to be every n minutes. In addition, let the slave processor to transmit messages for a message every minute for m minutes when fire is detected. In order for the master processor to detect a transmitted message its receiver must be powered on for at least (m – n) minutes. For example, if the slave and the master processor are configured to wake up every 15 minutes, and a message is to be transmitted every minute for the duration of 12 minutes, then the receiver of the master processor should be active for at least 3 minutes in order not to miss any messages sent by the slave processor.

In this demonstration project, the slave and master processors are in deep sleep mode for 15 minutes. After this time they wake up and check the sensors for possible fire detection. If fire has been detected, then a loop is setup and 12 early warning messages are transmitted with the ID of the slave processor that detected the fire. The transmitted messages are in the following format, where nn is the slave processor ID:

```
    nn:FIRE DETECTED
BEGIN/MAIN
    Configure temperature and humidity sensor port as digital input
    Configure flame sensor port as digital input
    Configure smoke detector sensor port as digital input
    Configure solid state switch port as output
    Configure the UART for TX/RX communication
    Read slave processor ID
    Deep sleep for 15 minutes, call PROCESS on wake up
END/MAIN

BEGIN/PROCESS
    Read temperature and humidity sensor output
    Read flame sensor output
    Read smoke sensor output
    Determine if fire occurred
    IF fire detected THEN
        DO 12 times
            Enable solid state switch
            Transmit early warning message with ID
            Wait one minute
        ENDDO
    ENDIF
    Set deep sleep mode
END/PROCESS
```

3.3.2. The Master Processor Software

The master processor wakes up from deep sleep mode and check if any messages are being transmitted by any slave processor. If a message is received, then the ID of the message is extracted and an early warning message is relayed to the Command and Control Center.

BEGIN/MAIN
Configure solid state switch as output
Configure the UART for TX/RX communication
Deep sleep for 15 minutes, call PROCESS on wake up
END/MAIN
BEGIN/PROCESS
Wait for 3 minutes and check for received message
IF a message is received **THEN**
Extract the slave processor ID
Relay an early fire message to the Command and Control Centre
ENDIF
Set deep sleep mode
END/PROCESS

3.3.3. The Construction

The master processor and the slave processors were built on breadboards and connections to various components were made using jumper wires. Figure 8 shows a slave processor built on a breadboard with the components marked.

4. Results

A test system was setup on three separate breadboards consisting of two slave processors and a master processor. The operation of the overall system was tested by introducing

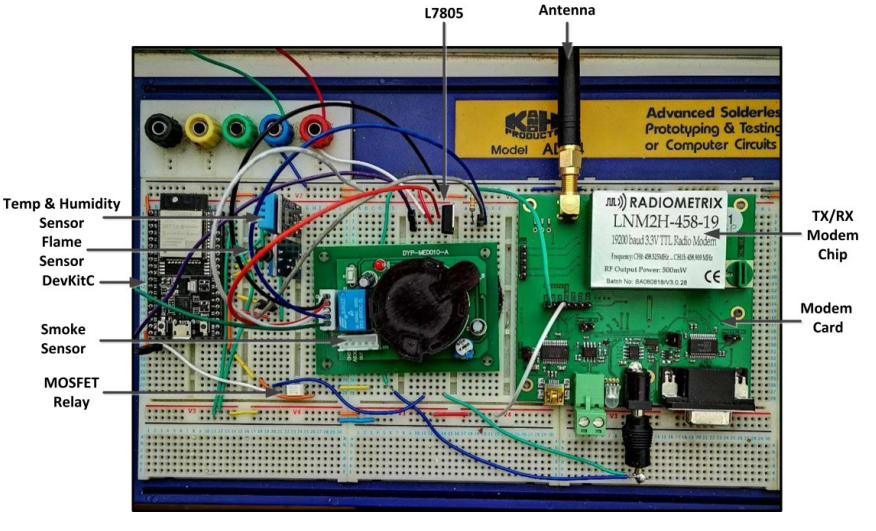

FIGURE 8. Slave processor built on a breadboard.

TABLE 1. Power consumption of the system (in mA)

	While in deep sleep	Woke up, checking for fire	Fire detected, transmit message	Checking for receive message
Slave processor	10	50	129	–
Master processor	10	–	–	112

smoke to the slave processors, by increasing the temperature, and by burning fire close to the slave processors. An independent standalone LNM2H-458-19 transmitter/receiver modem card was connected to a PC via its USB port in order to monitor the data communication in the system and all communication data was captured and displayed on the PC using the Putty [21] terminal emulation software. It was observed that the required early warning messages were sent to the master processor when fire was simulated near the slave processors.

The current consumption of the slave processors and the master processor were measured using a USB voltage monitoring device [22] and the results are shown in Table 1.

5. Conclusions

The design of a low-power early forest fire detection and warning system has been described. The novelty of the developed system is that its power consumption is rather low while in standby mode and when fire has not been detected. The system is based on using multiple slave processors scattered around the forest. Possible early fire is detected by the slave processors and these processors send warning messages to a master processor, which is located at the central point of the slave processors. The main novelty of the designed system is that its power consumption is very low. It is recommended to use solar energy to recharge the batteries since no matter how low the power consumption of the system is, the batteries will always become flat after sometime.

References

1. Roohi UKM. Comparison and application of forest fire detection system based on a ZigBee wireless sensor network. *International Journal of Engineering Research and General Science*. 2015; 3(1), 427–430.
2. Intelligent forest fire monitoring system – from idea to realization. https://bib.irb.hr/datoteka/612374.HATZ_Annual_2010_2011_Stipanicev.pdf. Date accessed: 2011.
3. An automatic early warning system for forest fires. https://www.researchgate.net/publication/224789267_An_automatic_early_warning_system_for_forest_fires. Date accessed: 01/2001.
4. Kolaric D, Skala K, Dubravic A. Integrated system for forest fire early detection and management. *Periodicum Biologorum*. 2008; 110(2), 205–211.
5. Samanta S. An architecture of future forest fire detection system. *International Journal on Recent and Innovation Trends in Computing and Communication*. 2016; 4(8), 139–140.

6. Jamdahade PC, Kawate AD, Lachake S, Shaikh AA, More SM, Kadam RS. Forest fire detection using optimized solar powered wireless sensor networks. *International Journal of Advance Research in Science and Engineering*. 2018; 7(2), 785–790.
7. Ganesh UA, Anand M, Arun S, Dinesh M, Gunaseelan P, Karthik R. Forest Fire detection using optimized solar-powered ZigBee wireless sensor networks. *International Journal of Scientific & Engineering Research*. 2013; 4(6), 586–596.
8. Smart emergency response system. https://smartamerica.org/teams/smart-emergency-response-system-sers/. Date accessed: 2017.
9. Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. https://ieeexplore.ieee.org/document/8101926. Date accessed: 12/09/2017.
10. Lozano C, Rodriguez O. Design of forest fire early detection system using wireless sensor networks. The Online Journal on Electronics and Engineering (OJEEE). 2015; 3(2), 402–405.
11. Mihai B. How to use the DHT22 sensor for measuring temperature and humidity with the Arduino board. ACTA Universitatis Cibiniensis. 2016; 68(1), 22–25.
12. Flame detector with Arduino. https://www.instructables.com/id/Flame-Detector-With-Arduino/. Date accessed: 2019.
13. DYP-ME0010-a smoke detector review, specifications and more. http://henrysbench.capnfatz.com/henrys-bench/arduino-projects-tips-and-more/dyp-me0010-smoke-detector/. Date accessed: 2019.
14. Photoelectric smoke detector IC with I/O and temporal pattern horn driver. https://www.nxp.com/docs/en/data-sheet/MC145012.pdf. Date accessed: 2019.
15. Data sheet. http://www.radiometrix.com/lnm2h-lnm3h. Date accessed: 2019.
16. Compact, general-purpose, analog switching MOS FET relays, with dielectric strength of 2.5 kVAC between I/O using optical isolation. https://omronfs.omron.com/en_US/ecb/products/pdf/en-g3vm_61a1_d1.pdf. Date accessed: 2019.
17. Positive voltage regulator ICs. https://www.st.com/resource/en/datasheet/l78.pdf. Date accessed: 2019.
18. Practical antenna handbook. https://www.amazon.com/Practical-Antenna-Handbook-Joseph-Carr/dp/0071639586. Date accessed:
19. The working principle of an Arduino. https://ieeexplore.ieee.org/document/6997578. Date accessed: 29/09/2014.
20. PIC microcontroller projects in C: basic to advanced. https://www.amazon.com/PIC-Microcontroller-Projects-Basic-Advanced/dp/0080999247. Date accessed: 07/05/2014.
21. Download putty. https://www.putty.org/. Date accessed: 2019.
22. USB voltage monitor. https://www.amazon.co.uk/Voltmeter-Bluetooth-Multimeter-Multi-function-Detector/dp/B07DCS11GM/ref=sr_1_5?keywords=usb+voltage+monitor&qid=1552408314&s=gateway&sr=8-5. Date accessed: 2019.