Arduino Based Embedded Systems and Remote Access Technologies of Environmental Variables Monitoring

Nikolaos F. Voudoukis

***Abstract*—A description of two systems as examples for remote access technologies using Arduino is presented. They are Arduino based embedded systems remotely controlled for environmental variables monitoring. The first system concerning variables of plant growth and the second is a home automation system (for smart homes). An embedded system is an applied computer system that is built to control a range of functions. A brief discussion of embedded systems is presented. Remote access technology is the ability to control aspects of a manufacturing process from any location. Sensors are devices which can measure a physical quantity and convert it into a signal. One solution for remote access technology is the control of the different variables through a website. The website displays monitoring data to the user. Another solution is the remote access to be based on the GSM (Global System for Mobile Communication) network. Theory explains the use of Arduino microcontroller, the use of sensors and how they are used in embedded systems.**

Index Terms—Arduino, embedded systems, sensors, remote access technologies, environmental variables, home automation

I. Introduction

Today embedded systems are replacing various systems that used to be designed with a set of complex electronic circuits. Usually the heart of the embedded system is a microcontroller. One example of a microcontroller is Arduino. Arduino is an open source based prototyping platform used to sense and control physical devices.

It is described an Arduino-based embedded system for monitoring environmental variables. As key environmental parameters in the field of agriculture we define parameters which are determining growing and ripening conditions, such as temperature, precipitation, radiation, soil moisture as well as parameters describing the status of vegetation like biomass (yield). For example radiation and light intensity obey the inverse-square law of the distance from the source and is not difficult to be measured [1]. It is worthy to be mentioned that an aspect which can be taken into account is energy saving regarding embedded systems. Green practices should extend to embedded systems, remote access technologies and generally communication networks [2].

An embedded system is an applied computer system that is built to control a range of functions. Because of rapidly evolving technology the meaning of embedded systems is a vastly fluctuating definition. Advancing technology causes decrease in the cost of manufacturing and allows implementation of various hardware and software components to embedded systems. Embedded system is dedicated to a specific task.

N. Voudoukis is with National Technical University of Athens, Greece (e-mail: nvoudoukis@mail.ntua.gr).

Systems normally consist of inputs, outputs and a small processing unit. Most of the devices used in our everyday life are some kind of embedded systems. Devices like mobile phones, watches, and elevators are all embedded systems. Most of the embedded systems are reactive systems, which mean that the information received by the system is constantly processed and the system acts based on the information. The example of a reactive embedded system is the car-braking system. Car brakes need to react instantly to a user input. System needs to react in a split second to prevent collision [3].

Remote access technology is the ability to control, manage, report, and troubleshoot any or all aspects of your manufacturing process from any location.

Arduino is an open source tool for developing computers that can sense and control more of the physical world than desktop computer.

Important factors for the quality and productivity of plant growth are temperature, humidity, light and the level of the carbon dioxide. Monitoring of these environmental variables gives better understanding on how efficiently plants are growing and how to achieve maximal plant growth. The optimal greenhouse climate adjustment can improve productivity and to achieve remarkable energy savings.

II. Embedded Systems

Embedded system architecture is a generalization of the system. Architecture does not show detailed implementation information such as software source code or hardware circuit design. The hardware and software components in an embedded system are presented as part of composition of interacting elements. Elements are representations of hardware and software, leaving only behavioral and inter-relationship information. A structure is one possible representation of the architecture. A structure is a snapshot of the system's hardware and software at design time or at run-time, given a particular environment and a given set of elements. An embedded systems' architecture is used to resolve challenges early in a project. Without defining or knowing the internal level of implementation the architecture is the first tool used to analyze the system. When designing an embedded system multiple different models can be used to approach the design. The development process is concluded systematically and between steps feedback is obtained and incorporated back to the process.

One of the final phases of embedded system design is the implementation of the design. Design and implementation of control systems are often separated. This causes the development of embedded systems to be highly time

DOI: http://dx.doi.org/10.24018/ejece.2019.3.4.101

consuming and costly. The implementation of the system can be more easy using several tools which have been built for this purpose. The development process of the embedded system's hardware and software layer is made possible with development tools. On the hardware side, one of the tools used is Computer-Aided Design (CAD). CAD is used to simulate circuits at the electrical level. A simulation is implemented in order to study a circuit's behavior before the final circuit. So software is used to test the hardware. Integrated Development Environment (IDE) is used to aid the implementation of the software side in embedded systems. After implementation, functional testing selects tests that assess how well the implementation meets the requirements of the product [4].

The goal of testing is to assure the quality of a system. The tester is trying to determine if the system is operating according to its design. Testing can be used to determine if the error also known as a bug is found from the system. and also to track whether bugs have been fixed [4].

Embedded Systems have the following characteristics. a) They are application specific and specialized. This means that they are designed for a specific application and programs are running repeatedly. b) They are optimized for performance (execution time, latency), energy efficiency, code size, dimensions and cost. c) They are typically designed to meet real-time constraints. This means that they are designed to react to stimuli from the object that they control, within the given time interval. Slower reaction than the time schedule is a problem for real-time systems. d) They are interacting with the environment through sensors and actuators. Therefore they are typically reactive systems. e) They generally have minimal or no user interface.

The elements which comprise an embedded system are Hardware and Software. The core element of an embedded system is the processor (hardware) who is programmed to perform specific tasks using a variety of options. The soul of embedded systems is software. It can be used a variety of languages and libraries according to the tasks that we want to run, depending on the nature of the application. For example, a particular approach (language, library or protocol) which is good for control-dominated applications might not be as good in other applications. Several languages are involved in system design. Hardware Design Languages (HDLs), such as Verilog or VHDL (or even SystemC) are used to describe hardware components. General Programming Languages, especially High Level Languages (C, C++, Java, ADA etc.) or Assembly (symbolic and difficult to understand) are often used for embedded software. Other specialized languages are better for specific application domains.

III. Remote Access Technology

The design of remote monitoring system involves various steps, namely selection of proper sensors to acquire the required physical parameters. In accordance with the conditions in which the system will be applied, it must be chosen the adequate equipment (temperature and humidity sensor, light sensor, CPU, debugging board). It must be analyzed the difference between the most popular and functional debugging boards to determine the most suitable for the system, compare electrical indexes, size and price of boards. Set-up configuration of equipment for its normal working is something that can be done. For the selection of the protocol, it is necessary to find out what are the possible methods of data transfer between sensors and debugging boards. Usually the datasheet of every module declares the description, parameters, product characteristics, type of protocols for information transfer, typical schemes, etc. The method of data transmission must be determined in accordance with the characteristics of equipment. For example the system may assumes the use of GSM / GPRS module as the transfer unit for exchanging information between the selected temperature and humidity sensor and the device to which the SMS message will send with the data sensor. The system designs in such a way that an SMS message sends in those cases when the temperature goes out of the normal functionality. The temperature mode of system sets according to requirements of the operating conditions.

The framework can be designed to be easily adapted and expanded to any kind of sensor, allowing collecting parameters like temperature, humidity, light and others, from analog and digital sensors. The implementation based on the GSM network, using SMS, allowing accessing the local systems from any kind of mobile device. The design and implementation can be also done to assure high energetic autonomy, high reliability and low production cost. With such versatility, the framework can be used in a wide range of contexts and problems, whenever is necessary to monitor systems or environments located on remote areas or simply whenever is necessary to have access, in almost real time, to the conditions of these systems/environments. Besides the advantage of access, in almost real time, to data that is located on remote systems or environments, the solutions that can be developed over the proposed framework can reduce maintenance costs, traveling costs, working time and, contribute to safer system and healthy environments.

IV. Arduino

Arduino boards were originally created in 2005 by Massimo Benzi of IVRAE Institute for the need to learn of the computer and electronic students. Arduino is a development board that integrates a microcontroller and its support circuitry with digital and analog inputs and outputs. It has an open source computing development platform based on an environment for programs creation. The software is written in C or C++ programming language. The Arduino development board is an implementation of wiring, a similar physical computing platform, which is based on the processing multimedia programming environment.

This single chip microcontroller has a microprocessor, which comes from a company called Atmel. The chip is known as an AVR. The AVR chip is running at only 16 MHz with an 8-bit core, and has a very limited amount of available memory, with 32 kilobytes of storage and 2 kilobytes of random access memory. Arduino setup build around Atmel microprocessor causes it to be easy and popular to be used in all different kinds of DIY projects [5, 6].

Arduino IDE is programming environment that allows the user to draft different kind of programs and load them into

the Arduino microcontroller. Arduino uses user-friendly programming language, which is based on programming language called Processing. After the user has written his code, IDE compiles and translates the code to the assembler language. After translating the code, the IDE uploads the program to the Arduino microcontroller. Arduino IDE has a built-in code parser that will check the user written code before sending it to the Arduino. IDE software includes the set of different kind of programs that are ready to be tested on the device. After testing the program it can be uploaded to the Arduino by USB cable that vary in different models [6].

Fig. 1. Arduino Uno microcontroller

Arduino Due is the Arduino Microcontroller family's first development board based on the Atmel SAM3X8E ARM Cortex-M3 CPU that is shown in GRAPH 7. It has 54 digital input/output pins, 12 analog inputs, an 84 MHz clock, an USB OTG capable connection, 2 DAC (digital to analog), 2 TWI, a power jack, an SPI header, a JTAG header, a reset button and an erase button. Arduino Due has extended memory capabilities with 512kb of FLASH memory and 96kb or SRAM. The difference to other Arduino family boards is that the logical level voltage is 3.3v, the most of the Arduino boards run 5v on logical level. Arduino Due is the extended version of the Arduino family and it has all the basic functionalities of an Arduino. The microcontroller does not lack its usability because it has good compatibility with different module boards (shields).

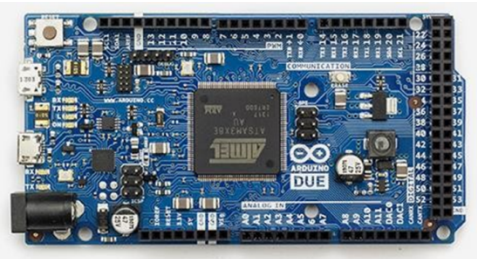

Fig. 2. Arduino Due microcontroller

Arduino Mega is quite similar to Arduino UNO in terms of electronic characteristics but contains many more pins.

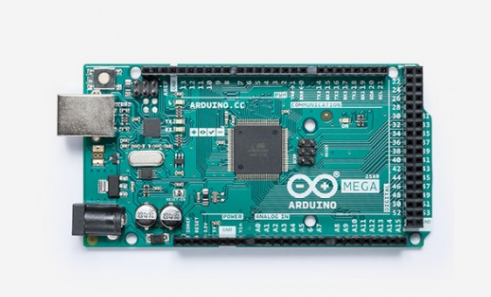

Fig. 3. Arduino Mega microcontroller

Shields are boards that can be stacked on top of the Arduino circuit board extending its capabilities. The different shields follow the same philosophy as the original toolkit: they are easy to mount, and cheap to produce. Arduino Shields are designed to improve the versatility of the simple board. Almost every model of Arduino is compatible with shields designed to it. Shields do not only improve Arduino by giving it more connected sensors or circuits. They also contain code libraries made for the specific usage of the shield. Most common reason to buys shield for Arduino is that the project requires more input or output devices, which default port amount cannot provide. After the community has started developing different shields by themselves, Arduino manufacturers have started to embed shields directly into Arduino circuit boards. Easy install and removal of the shield gives opportunity to use Arduino in all different type of projects [6].

Fig. 4. Ethernet Shield on top

V. Sensors

Sensors are devices which can measure a physical quantity and convert it into a signal which can be read by an observer, such as an electronic instrument. In other words, a sensor is a device which responds to an input physical quantity (e.g. temperature, light) by generating a functionally related output function usually in a form of an electrical or an optical signal. Nowadays, sensors are having numerous applications in our lives. They are used in simple or complicated applications in every embedded system today, such as automotive, train and aerospace industry, computer monitoring systems, fire detection, electricity distribution, industrial automation, automated and smart homes, audio/image/video surveillance, traffic monitoring, medical device monitoring, weather/climate monitoring, air traffic control and robot control. Actually, without sensors, it would be difficult to design or even imagine a control, surveillance or security system. An airplane without sensors will not have the ability to measure the distance between the aircraft and the ground. Also monitoring some environmental variables gives better understanding on how efficiently plants are growing and how to achieve maximal plant growth.

A sensor needs to follow these three rules. 1) The sensor must be sensitive only to the physical quantity for which the sensor has been created. 2) The sensor must not be influenced by any other kind of physical quantity that might influence the system. 3) The sensor must not influence the measured physical quantity, so that in it provides correct values of the measured quantity. The sensor would be ideal if all previously mentioned rules are obeyed. An ideal sensor is typically a linear or logarithmic mathematical function of the measurement. Its output is an analogue signal, related to

the value of the measured quantity. If a sensor needs to be used by a digital system, then its analogue output must be converted into digital, using an Analogue to Digital Converter (ADC). In real sensors, deviations are observed. The resolution of a sensor is the smallest deviation in the measured physical quantity that it can detect. It obviously relates on the accuracy of the sensor.

Sometimes the sensitivity might not be appropriate for the application in which a sensor is used. Other times deviations due to non-linearity, hysteresis, noise and many other conditions e.g., related to the sampling frequency are observed. These deviations can be categorized as systematic errors and random errors. Systematic errors can be reduced by calibrating the sensitivity of the sensors while random errors (e.g. noise) interfering with the hardware can usually be reduced by placing filters.

Sensors are used for monitoring environmental or physical quantities. However, some of these measurements are really difficult to perform, since they must be placed under extreme conditions. This leads us to create sensor networks which can monitor an environment from a safe distance and transmit data far away from their location. Such achievements can help predict physical disasters or understand our planet and our galactic neighborhood even better. A sensor network is a group of specialized transducers with a communication infrastructure intended to monitor and record conditions at diverse locations. Commonly monitored parameters are temperature, humidity, pressure, wind direction and speed, illumination intensity, vibration intensity, sound intensity, power-line voltage, chemical concentrations, pollutant levels, and vital body functions.

Arduino belongs to embedded systems, which means that it communicates with its environment through sensors. As the Arduino became known and spread all over the world, new sensors were designed. At the same time, the need of measuring different physical quantities led to design better control and monitoring systems. That led to the design of a plethora of sensors which, as the years were passing, were appropriate for measuring each physical quantity at different levels of accuracy.

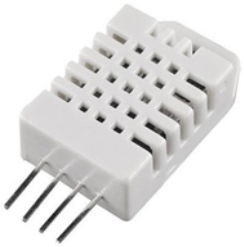

Fig. 5. DHT22, Digital moisture and temperature sensor

VI. Environmental Variables Monitoring Systems

A) First example: Variables of plant growth

Important factors for the quality and productivity of plant growth are temperature, humidity, light and the level of the carbon dioxide. Monitoring of these environmental variables gives better understanding on how efficiently plants are growing and how to achieve maximal plant growth. For example the optimal greenhouse climate adjustment can enable us to improve productivity and to achieve remarkable energy savings - especially during the winter in northern countries. Difference between consumer and corporate level greenhouse monitoring system is that the accuracy of sensing environment is on a small area. Consumer level greenhouses are smaller so total cost of the system can be kept low. It used to be in the past that greenhouses had one cabled measurement point in the middle to measure information from the greenhouse automation system. Modern greenhouses are larger and more adjustable. Lights, ventilation, heating and other support systems can be more precisely controlled, which requires increased amount of sensors and better accuracy of locations. Increased number of measurement points should not dramatically increase the automation system cost [7].

Certain requirements must be set before beginning to design the monitoring system for environmental variables (practical case-application a greenhouse). The system is needed to be easy to use and the user could remotely monitor environmental changes inside the greenhouse. Sensor data required to be collected and stored for showing long period changes in the environment variables. Hardware requirements were set so that the cost of the system would be low as possible. Only three variables can be chosen to be tracked in order to narrow down the cost of the system, for example air humidity, soil moisture and temperature. A good choice for this is Arduino Due and a web-based user interface. Embedded system design is divided into three layers: Hardware, Software and Application layer. Hardware layer consists of electrical specifications of the design. Layer describes wiring of sensors, shield, RTC and Relays to Arduino Due. Software layer has the programming design of the system. Software describes functions to control relays and technique used to read sensor data. Application layer introduces the design of web-based user interface and the way data is transferred between software and the application layer.

Arduino Due microcontroller is the heart of the greenhouse monitoring system. Due provides enough processing power and memory to run the webserver and it can read multiple sensors simultaneously. To add network connectivity to the project, network shield W5100 is added on top of the main board. Arduino network shield enables website hosting to local area network (LAN). W5100 network shield included SD card slot, which was used for long term data storage. Real-time clock module is introduced to the system to keep up time and date, even on power loss. The real time clock module DS1302 is connected to Due through I2C bus. RTC module's data line (SDA) is connected to pin 20 and clock line (SCL) is connected to pin 21 on the Arduino. Digital humidity and temperature sensor DHT22 is connected to digital pin 12 of the microcontroller. Two soil moisture sensors are connected to pins A0 and A1. Current is driven from Arduino to one of the poles in the soil moisture sensor. The second pole on the sensor is connected to Arduino analog pin. Analog pin is used to sense amount of conductivity of the soil. Moisture in a ground increases the conductivity of the soil.

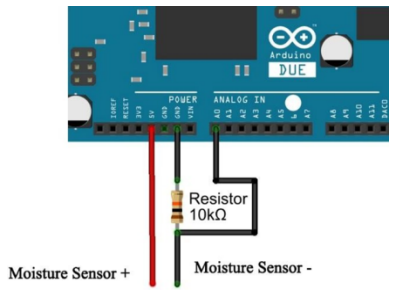

Fig. 6 Moisture sensor circuit diagram [8]

Relay modules are required to separate high current and high voltage devices from logical level devices. The water pump and heater are high voltage and high current devices which are controlled through relay modules. Modules are connected to digital pins 6 and 7.

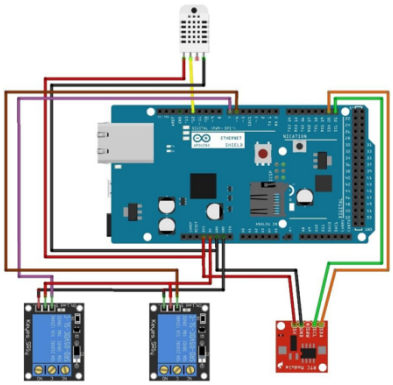
Fig.7 Wiring diagram of the hardware [8]

In Arduino programming there are two main functions. Main functions are setup() and loop(). Setup() function is only operated once when device is booted up, it is mostly used to setup initiation settings. Loop() is ran after the setup() function has finished, loop() function will run repeatedly until power off or reset button is pushed. Arduino programming is supported by wide amount of libraries. Large amount of open-source libraries are available from Arduino community [9].

Programming of the software is first started from the sensors and the real time clock module. Arduino IDE provided libraries to help reading data from DHT22 and RTC modules. Soil moisture sensor data is read directly from analog pins. Data from sensors and RTC is printed out to IDE's serial the monitor for testing purposes. One of the requirements of the system is that there would be long-term data saved for charting purposes. Arduino provides a SD card library, which can be used to create a function, for example sdCardDatalog(). The function saves sensor data and the time to the SD card on the W5100 network shield. Webserver needed to be established, to provide user remotely monitor their greenhouse through webpage. Webserver libraries can be created for Arduino but they did not meet the requirements of the system. Live charting can be designed to display data for the user on the website. Internet connection would have been needed without JavaScript support on the webserver. The new webserver enables the system to be used offline in local area network without internet connection.

Asynchronous JavaScript and XML (AJAX) can be used to transfer sensor data from webserver to a client. The client sends a request of XMLHttpRequest object to communicate with server-side scripts. The server responds to the request by sending the XML file containing sensor data in its HTTP header. The request can send as well as receive information in different formats, including XML, HTML, and even text files. If the webserver cannot access the RTC module data, the system does not save the sensor data to SD card or send it to the web client.

Application layer of the system is the user interface webpage. Web interface can be accessed by hard coded IP address. On the webpage, the user can access relays and view the live data of the sensors. JavaScript library Chart.js can be used to render 24-hour line chart data of temperature and humidity. Line chart data is stored on the SD card on the server. Before the data is sent to the client, the latest timecoded sensor data is verified to make sure duplicate values are not saved on the SD card. In occurrence of sudden power outage system the will load latest data from the SD card and send it to be displayed. Real- time clock module time and date can be displayed under the main heading. Time can be updated every minute together with sensor data and line chart. If webserver does not respond to client data request line chart is shown empty.

Arduino provides tools to assist the implementation of a system. Arduino IDE enables serial communication with Arduino microcontroller through USB. Printing to serial console in the IDE helps to track function events between the hardware and the software layer. Arduino community open source libraries assist the implementation of the sensors and network shield to the Arduino Due. Libraries give access to live sensor data with single function. Another implementing tool can be used for the design is Fritzing [8]. It is an open-source circuit design tool for prototyping. It is used to create wiring diagrams of the project. Graphical presentation of modules, sensors and microcontrollers give the designer free hands to design the prototype.

Testing of the system mainly focuses on the software side testing. During the building process of the monitoring system, dynamic testing must be done constantly together while programming the system itself. Hardware side testing is mainly checking the wiring diagrams of the sensors. After the sensor, web-server, SD card and real time clock functions are tested and diagnosed to be fully operational, software of the system is put together. Arduino being able to print serial data during development process helped debugging of the system.

B) Second example: Home automation-Smart home

To approach the house to a real home automation application the variables under study and control can be interior temperature and lighting, movement around the house and maybe water level of the pool. The house has three main operating modes. In automatic mode, it performs the measurement and executes the control of the variables, regulating itself according to the conditions to which it is exposed. In contrast, the remote mode is achieved using a mobile application that allows user to modify the variables. Finally, in alarm mode it controls the parameters that assure the security of the house when proprietor is away from home. For capturing the signals, temperature, lighting, movement and water level sensors installed, and for the regulation and control it has a fan, some LED, an acoustic

warning device and a water pump. It can be used an Arduino Mega board that allows the application operation and receives, from an Android mobile application, operating modes commands and, if it is operating remotely, orders to individually control the different actuators. For the data transmission from the mobile to the board, can be used communication via Bluetooth. Bluetooth is an industrial specification for WPAN (Wireless Personal Area Network), that enables the transmission of data and voice between different devices through a radiofrequency link in the 2.4GHz ISM band. The main objectives to be achieved with this standard are facilitate communication between mobile devices, remove cables and connectors between them and offer the possibility to create small wireless networks facilitating the synchronization of data between personal equipment. All Bluetooth devices have a unique address of 48 bits and a device name that allows the identification of each other.

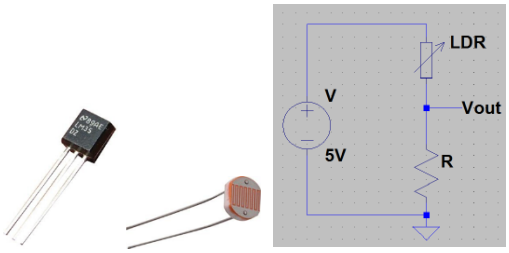

Fig.8. Temperature sensor (LM35), Light sensor (LDR) and circuit (voltage divider) for LDR

Nowadays thanks to the boom of mobile applications there are infinite possibilities to build one. One of them is to use a software in which is not needed to know anything of code. Another choice can be to use Google App Inventor software, an integrated development environment to create mobile applications for Android operating system.

VII. Conclusion

A description of two systems as examples for remote access technologies using Arduino is presented. They are Arduino based embedded systems remotely controlled for environmental variables monitoring.

The system for measurement and control of the three essential parameters for plant growth (temperature, humidity and soil moisture) can successfully reduce the power consumption, complexity and the cost. The results obtained from the measurement have shown that the system performance is quite reliable and accurate. Arduino microcontrollers are evolving development platform. The advantage of this monitoring system is that the user can easily access the data through the web interface and the user can affect the environment inside the greenhouse through the interface. The system could be designed to operate in a wireless environment. Wi-Fi technology would be considerable option to provide free access to user interface. Alternatively, can be 3G or 4G cellular connectivity. Accessing the monitoring data regardless of distance to the location of the greenhouse would make monitoring more efficient and less time consuming. Data could be accessed via internet with the handheld device.

A system is called home automation system if it is capable of automate a house or building including energy management, security, comfort, communication and data accessibility. It can be integrated through wired or wireless communication networks, although nowadays the predominant trend is wireless. The designed framework allows adding more sensors and other equipment's, empowering the solution and allows its use on many other areas and problems. Disadvantage of system is necessity to replenish the balance of SIM card in module for normal operation. Solution to this problem can be the selection of the most suitable mobile phone plan, as there are many convenient plans which had offered by mobile operators. Also, this system can be adopted for mobile applications based on Android or iOS platforms.

Modern control systems for remote monitoring of different variables are an important feature. One of the most convenient solutions is GSM technology, available almost everywhere and requiring no wires. This design has benefits, such as small size, low power consumption, low cost, high accuracy, superior anti-interference performance, and it can be real-time and easy development, human-interactive concisely and clearly. The system suitable for long distance work conditions of wireless data acquisition and transmission.

References

[1] Voudoukis, Nikolaos, Oikonomidis Sarantos, "Inverse square law for light and radiation: A unifying educational approach" *EJERS, European Journal of Engineering Research and Science,* Vol. 2, No. 11, November 2017

[2] G. Pagiatakis, Gerasimos, N. Voudoukis, E. Evangelou, E, "A course module on Green Telecommunications", *IEEE Global Engineering Education Conference (EDUCON),* Proceedings of 2017

[3] Karvinen, T. & Karvinen, K., *Getting Started with Sensors: Measure the World with Electronics, Arduino, and Raspberry Pi.* Maker Media, Inc., 2014

[4] Noergaard, T. Embedded Technology : Embedded Systems Architecture : A Comprehensive Guide for Engineers and Programmers. Burlington, MA, USA:Newnes. ProQuest ebrary. 2005.

[5] Evans, B. *Beginning Arduino Programming*. 2-3. New York: Apress. 2011.

[6] Banzi, M. *Arduino – Getting Started with Arduino*, Maker Media, Inc. 2011

[7] Timmerman, G. J. & Kamp, P. G. H., *Computerized Environmental Control in Greenhouses, PTC.*, 2003.

[8] Fritzing, "Fritzing: a tool for advancing electronic prototyping for designers". Available http://fritz- ing.org/home Accessed: 12 March 2019.

[9] Llemos, J.A., "How does an Arduino sketch work?" Available: http://japaalekhin.llemos.com/wp- content/uploads/2015/10/arduino-software.jpg., 2015. Accessed: 17 February 2019.

Nikolaos Voudoukis received a BSc degree in Physics from Athens National University, Greece, in 1991, a BSc in Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 2012, his MSc degree in Electronics and Telecommunications from Athens National University, in 1993, and his PhD degree from Athens National University, in 2013. He has worked as telecommunication engineer in Greece, as teacher and Assistant Director at a high school and as a part-time Lecturer at the School of Pedagogical & Technological. Education, Athens, Greece. He is IEEE member. Dr. Voudoukis now is Research Associate at School of Electrical and Computer Engineering, National Technical University of Athens, Greece.