

Development of a Low-Cost Monitoring and Warning System for a Data Processing Room

Oğuzhan TİMUR*¹ ORCID 0000-0002-6537-7840
Ahmet TEKE¹ ORCID 0000-0003-2610-9576

¹ Cukurova University, Faculty of Engineering, Department of Electrical & Electronics Engineering, Adana, Türkiye

Geliş tarihi: 20.05.2023

Kabul tarihi: 23.06.2023

Atıf şekli/ How to cite: TİMUR, O., TEKE, A., (2023). Development of a Low-Cost Monitoring and Warning System for a Data Processing Room. Cukurova University, Journal of the Faculty of Engineering, 38(2), 375-390.

Abstract

Continuity is crucial in systems that provide uninterrupted service. When errors or warnings occur, they should be reported to the responsible people as quickly as possible and fixed right away. Since manual control will cause errors, monitoring and control should be performed automatically. In order to fill this need, expensive hardware and software are used in the market. The purchased package includes a lot of extra features that are not required. Additionally, the majority of these systems use brand-specific sensors and other hardware components. In this study, the identical operation is carried out with significantly greater cost efficiency and brand independence by using microcontroller-based hardware. The designed system can be operated in desktop, web and mobile environments thanks to the computer program that is written. In the data processing center of a university hospital, the proposed cost-effective system has been used, and it has been discovered that it is capable of carrying out the same tasks as the more expensive systems.

Keywords: Smart systems, Monitoring and warning systems, Embedded systems, Software

Bilgi İşlem Merkezi için Ucuz Maliyetli Görüntüleme ve İkaz Sisteminin Geliştirilmesi

Öz

Kesintisiz hizmet veren sistemlerde, süreklilik büyük önem taşımaktadır. Meydana gelecek olan hataların veya uyarıların en kısa zamanda sorumlu kişilere bildirilmesi ve mümkün olan en kısa zamanda çözülmesi gerekir. Manuel kontrol hatalara sebebiyet vereceği için, görüntüleme ve kontrol otomatik olarak gerçekleştirilmelidir. Piyasada bu ihtiyacı karşılamak için kullanılan yazılım ve donanımlar, yüksek maliyetlere sahiptir. İhtiyaç duyulmayan birçok özellik, satın alınan ürünle birlikte fazladan gelmektedir. Ek olarak bu sistemlerin çoğunda, kullanılan algılama cihazları (sensörler) ve diğer donanım ürünleri marka bağımlıdır. Bu çalışmada aynı görevi çok daha uygun maliyetle ve marka bağımsız olarak gerçekleştirecek mikrodenetleyici tabanlı donanım kullanılmıştır. Tasarlanan sistem, yazılan bilgisayar programı sayesinde, masaüstü, web ve mobil ortamında çalıştırılabilmektedir. Önerilen uygun maliyetli uyarı ve kontrol sistemi,

*Sorumlu yazar (Corresponding Author): Oğuzhan TİMUR, otimur@cu.edu.tr

bir üniversite hastanesinin bilgi işlem merkezinde kullanılmış olup, pahalı olan sistemlerle aynı işleri gerçekleştirildiği tespit edilmiştir.

Anahtar Kelimeler: Akıllı sistemler, Görüntüleme ve uyarı sistemleri, Gömülü sistemler, Yazılım

1. INTRODUCTION

Monitoring, analysis, control and management of continuous systems are vitally important for prevention of inefficient energy usage and waste of time. Because of this, appliances which consume energy should be monitored 7 days 24 hours continuously. Measured values and warnings in the system should be saved. In addition, saved data should be analyzed by statistical and machine learning methods.

Generally, smart systems are used in buildings for energy efficiency, fire detection, and safety systems. Smart and computer-aided automatic control systems should present more comfortable life to residents and the other living creatures. In addition, smart systems can be monitored and managed by remote access and management tools in mobile phones, laptop computers, and other mobile devices.

There are a lot of studies interested in monitoring, analyzing and optimization. One of the best studies is realized by Gokce and Gokce's friends in 2014. They have developed a new energy monitoring, analysis and optimization tool for energy efficient buildings. Their study can be grouped into four sections as data mining and monitoring, data analysis, user awareness and optimization. Java programming platform is preferred to develop software in their study. When it is focused on the energy efficiency section, three main sections are attracted attention as energy consumption monitoring, the heating system controlling and lighting system controlling. As a result of their study, their system achieves to improving building performance, developing intelligent control routines and implementing fault diagnosis measures [1].

In the study of Santis and his friends have proposed smart monitoring system to monitor low voltage (LV) electrical grids. The proposed project has been

tested at four different cases. In the proposed scenario, real-time measurements have been obtained. Measure current and voltage transducers are connected to the Arduino DUE controller to measure three-phase line currents and voltages. In this scenario, current and voltage values of the grid lines are measured phase by phase in every 0.5 second continuously. These measured values are compared to real values measured by using Fluke 287 multi-meter. As a result of this comparison, maximum less than 1.30% difference is reported. According to measured current and voltage values, active and reactive power values are calculated. Wi-Fi and GPS module are also used in the proposed scenario at the same time. GPS module is used for location of the measurement point and Wi-Fi is used for communication. All measurements and calculated values are displayed on a web page. One of the advantages of this system is that it can be seen from anywhere by web browser easily. Another important advantage is the low cost of devices. The entire system only costs €200 approximately [2].

Behan and Krejcar have realized web-based smart home system. In their paper, they have described the common problem of house automation systems primarily. According to them, this type of home automation system has closed based solutions which means there is not certain standard for sensors, controllers, and user interface development softwares. They have proposed a new type infrastructure based on both software and hardware side is open sourced. In this project, Linux based open source minicomputer RaspberryPi is used as home controller named as commander and arduino which has similar features with RaspberryPi is used as room controller. Ethernet shields which have Power over Ethernet (POE) feature are used instead of the using external power supply in the proposed project. At the same time, another duty of ethernet cards is to provide communication among the devices. In their solution proposal, the physical house is divided into logical units. Every logical unit is equivalent to a room or a part of a house.

Every logical unit has its own controller which connects to the commander via Unshielded Twisted Pair (UTP) cable in a star topology. Data is transmitted and received over ethernet and all device powers are supplied by PoE standard named as 802.3af by The Institute of Electrical and Electronics Engineers (IEEE). The power of PoE depends on supply voltage and it is confined with a maximum of 48 V or 400 mA. After the installation of hardware, web-based management software can be used through any mobile device that has Android, IOS or Windows operating system. In summary, they have realized web-based controlled smart home automation system based on open electronic and open-source software solution in this project. This new system will provide a connection between sensors and web interface with controlling, monitoring, and visualizing. The most important innovation of this study is that the implemented system can be redesigned, improved, upgraded, or tuned by anyone under an open-source license [3].

In 2014, Hussein and his friends realized smart home prototype for disabled people. In the proposed project, all systems are almost backed up. Two internal networks are used for connection among devices used in disabled people's house. One of them is used to ZigBee protocol and other is used to Power Line Communication (PLC) protocol. Normally, the system runs by using ZigBee but if any problem occurs, PLC network will be substituted automatically. Similarly, two external networks are used for communication between responsible people and proposed system in the house. Thanks to these external networks, remote access to the system can be provided by the authorities. In their experiments, Neural Networks (NNs) methods have been used to predict future events. Firstly, in their smart fire detection system, perceptron based feed-forward neural network has been used. Five parameters have been used as input parameters. These parameters are carbon monoxide level, oxygen concentration, smoke detection, heat level, and flame detection. With the help of these parameters, they will be able to determine if there is a fire or not and according to changing values of parameters, they will be able to guess fires which can be occurred. Another NN study is interested in users' habits and forecasting their movements. This

NN type is a recurrent neural network. In this study, Arduino microcontroller is used for switching ON/OFF of devices such as coffee machine, television, lights, and so forth. As a power measurement device, a differential amplifier circuit is also used. Used NN methods, Arduino controller and other electronic circuits like differential amplifiers have been operated in relation to each other. For example, if someone wakes up in the morning, when he/she goes into the living room, prediction mechanism will be started by NN methods. Because of this, the lights of the room and television will be turned on automatically [4].

Arduino and the other embedded cards have broad application areas such as energy, health, industry, safety, transportation and so forth. Gad and Gad have developed a new temperature data acquisition system for solar energy applications. When this system is compared to classical data acquisition systems, the newly developed system has a lot of significant advantages. The main advantages of the system are its cheap and flexible structure. Because of the flexible structure, used temperature sensors can be changed with other type and mark sensors easily. In this study, by using 16 temperature sensors and taking 5 minutes sampling rate, ambient temperatures were measured and recorded on Secure Digital (SD) memory card with Comma Secured Values (CSV) file format. In classical systems, recorded data on Arduino or other embedded system boards wait until system operator or administrator downloads data to a computer. When data are downloaded into the computer, recorded data are transferred to computer from the embedded system board. The operation of described system is quite simple. By the new technology, sampling rate defines recording time periods and measured temperature values can be saved on the SD card without the need of any computer with CSV file format. A new CSV file is created and recorded on SD card every new day. Starting and ending time values are defined in the system by using the PIC and demonstrated on Liquid Crystal Display (LCD) screen. Used Arduino board on this system is responsible for evaluating of voltage values and writing data on the SD card. LM35DZ analog temperature sensor is used on the sensor-based designed card. It can

measure temperature values between $-55\text{ }^{\circ}\text{C}$ and $+150\text{ }^{\circ}\text{C}$. Its accuracy is 0.25% of the real value. Its internal temperature is about $0.1\text{ }^{\circ}\text{C}$. In this study, both PIC and Arduino are programmed respectively. PIC software consists of three sections named as initialization, waiting and recording. Starting and ending times of recording time are defined in the initialization section by users. During the execution of the software, the starting and ending times are compared with Real Time Clock (RTC) until the RTC is smaller than the ending time. Data packet is defined in PIC software and sent to Arduino board [5].

Rahman et al. were realized Smart Energy Meter by using Arduino and GSM module. Owing to this smart meter, they have measured energy consumption and calculated the electricity bill. Firstly, they have simulated this study in Proteus and then conducted in a laboratory environment successfully. In this work, they have used energy meter Integrated Circuit (IC), Arduino microcontroller, GSM modem and Arduino relay shield. After power on, Arduino has turned on the relay and connects the load and energy meters to the line. Energy meter sends an impulse to Arduino according to energy consumption. So Arduino increases the value and shows on its display. At the same time, GSM modem checks whether there is a new SMS or not. And then if the SMS text is "DATA", it sends consumption data measured by the Arduino and it stores in EPROM to electric provider company. If the text is "LINE CUT", GSM modem sends command to Arduino to turn-off relay and deenergize the line energy. In contrast, if the text is "LINE OK", Arduino turns on the relay again and the load will be connected to line. In this study, they have calculated 1 kWh consumption for every 1600 impulse by using energy meter IC. A Light Emitting Diode (LED) is connected to Arduino to show the impulse visually, so the LED flashes with each impulse. If the provider company wants to read consumption value from the meter, they can send a SMS to a smart meter. Afterwards, smart meter has replied to taken message by sending the consumer ID and consumed energy value. (Rahman et al., 2015) can also control the line status. This project can be turned on or off the line energy which customer connects by using GSM wireless network

protocol. One of the disadvantages of this system is high material cost. It may be reduced by mass production. In this study, the authors show that realized smart energy meter saves from time, money, human power, and labor costs [6].

Ismail and his friends have developed a door lock Android application named "LockIt Door" for disabled people in their study. This application has been designed by using Eclipse and Java softwares. This study has been especially realized for disabled people who use a wheelchair. The disabled users can use Android application without any human help easily. In this realized project, bluetooth communication protocol is preferred instead of Wi-Fi, ZigBee, NFC, and others. The working principle of the implemented circuit can be briefly summarized as follows; written Android application connects to arduino controller through bluetooth module. According to command coming from the android software, the relay contact which is connected to arduino either opens or closes door. In order to find an effective range of bluetooth, the realized project has been tested for three different environments such as indoors with obstacles, indoors without obstacles, and outdoors without obstacles. After these tests, maximum range radius has been found as 20 meters without any problems at outdoor without obstacle area. Wi-Fi communication protocol can be used instead of bluetooth to improve the effective range in the future works. [7]

Wang and Chi have designed a new temperature and humidity measurement circuit which can communicate wirelessly. The designed system consists of two parts. In the first part, temperature and humidity sensor named DHT11 is connected to the Arduino controller which has a wireless transmitter. In the second part, arduino controller that has a wireless transmitter is connected to LCD, buzzer and setting module. As understood from the devices, the first part is used for input and second part is used for output. Read values with the helping of the first part is sent to the second part with the helping of wireless. Wireless modules communicate with each other by serial communication protocols. Taken values are employed in the second part and displayed on the

LCD screen. In this project, normal temperature and humidity are defined by using potentiometer named as setting module in the project. If the range values are exceeded or fell down, the buzzer will be active. According to Wang and Chi, it is an advantage that this project consists of two parts. Because the location where the sensors are located and the location where the results are displayed may be different. The realized project has been tested for the indoor and outdoor environment. While wireless communication is provided up to 40 meters in an outdoor environment, the maximum effective range is up to 20 meters in the indoor environment because of the barriers. As a result of this study, the advantages of this system can be summarized as low cost, low power consumption, high accuracy, and so forth [8].

Ramadhani and his friends have designed a web-based air quality monitoring system that can measure and display the types of gases like carbon monoxide and sulfur monoxide. In the realized prototype, gas sensor has been developed by Ramadhani et al. in 2013. Zinc Oxide thin film is used as a gas sensor material in order to for measuring the amount of gas. Monitoring system mainly consists of head and body modules. While head module has a gas sensor, arduino board, and ethernet shield, body module has server and database. In this study, gas sensor is used for sensing of the amount of gas, ethernet shield is used for communication and arduino microcontroller is used as Analog to Digital Converter (ADC). In addition, server and database are used for data acquisition and data management. The implemented system runs briefly as follows: when designed gas sensor perceives the amount of gas in the environment, the resistance of gas sensor will change. These resistance values will be analog input values of arduino. Arduino will convert these analog values to the digital values and then digital values will be transferred to the server through the ethernet shield. Data which arrive the server will be saved to the database as raw data by using simple PHP (Personel Home Page) interface. This process will continue periodically. The whole raw data will be converted to understandable data by software and graphical plot diagram. Besides, all data will be saved to MYSQL database tables. Consequently,

the users can monitor graphic and gas values by using this web-based software easily. In the realized prototype, in order to increase system sensitivities, the amount of carbon dioxide added into the sensor material should be used for different concentration [9].

As seen in studies in the selected journals, used systems have been realized by using electronic cards with microcontrollers. Later, the sensors selected according to the requirements differed. After the installation and programming of the embedded cards, the warning and display system can be set up by using computer programming language.

2. MATERIAL AND METHOD

The realized study mainly consists of two main parts. These parts have been named as embedded system part and computer programming part. While, embedded system part occurs from input devices, output devices, and microcontroller, computer programming part also consists of two parts as administrative part, and client part.

Administrative part occurs from server computer and desktop management software. Moreover, client part occurs from users and desktop, web, and Android based software solutions. The system block diagram of the realized project is generally shown in Figure 1.

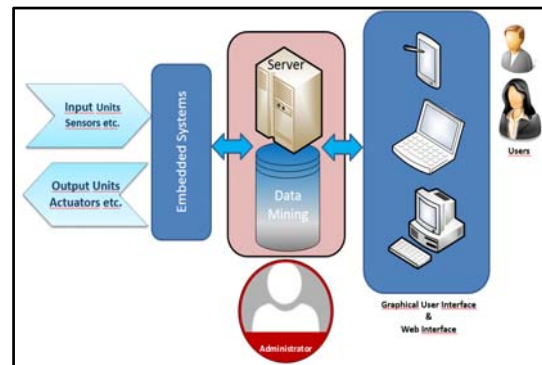


Figure 1. Block diagram of the realized project

Arduino board is constructed around a straightforward microcontroller board [10]. A

variety of lights, fans, and other physical outputs can be controlled by the Arduino board to construct interactive devices that can accept input from various sensors or switches [11]. The Arduino board has a lot of benefits when compared to other microcontroller platforms. Among the benefits are low cost, ease of use, cross-platform compatibility, a clear programming environment, extensible and open source hardware, and extensible and open source software [12]. The sensor based input layer is the hardware framework of the smart systems.

2.1. Embedded System Part

In this study, embedded systems were established by ARDUINO. ATMEL microprocessor on the Arduino was programmed by C+ compiler. Thus, the connection between sensors and actuators was provided. Arduino cards used as end devices were connected to router cards using ZigBee wireless protocol. Similarly, Arduino cards used as router devices were connected to coordinator card with using the same wireless protocol. The realized embedded system part comprises of three sections energy management section (EMS), security check section (SCS), and a fire detection control section (FDCS) mainly. The realized system was named as Smart Building Management System (SBMS). The general view of the SBMS is shown in Figure 2.

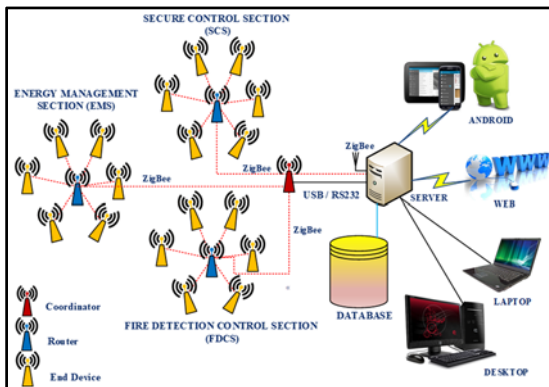


Figure 2. Schematic view of the realized project

In the EMS, studies were actualized interested in smart plug which was named as iDev (intelligent Device) briefly and split type air conditioner control. The electrical quantities such as voltage,

current and power factor were measured with iDev and split air conditioners in the data processing room were controlled with the microcontroller based split air conditioner control card [13,14].

In addition to EMS, SCS, and FDCS were occurred in this study. SCS is made up of password entering and fingerprint reading parts respectively. password entry was realized with keypad, timing control with Real Time Clock (RTC), and SMS sending applications with GPRS module were realized as shown in Figure 3a.

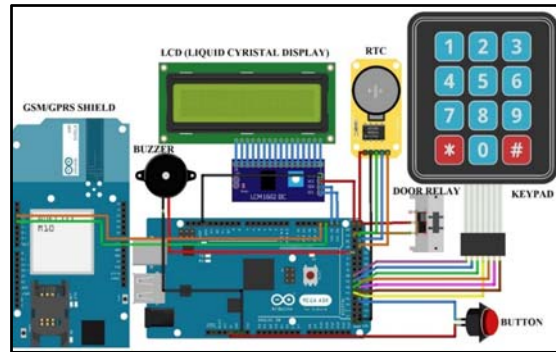


Figure 3a. Password Entering Part of SCS

The fingerprint reading part was actualized with a fingerprint sensor and a relay mainly as shown in Figure 3b. The relay was used to control the opening and closing of the door.

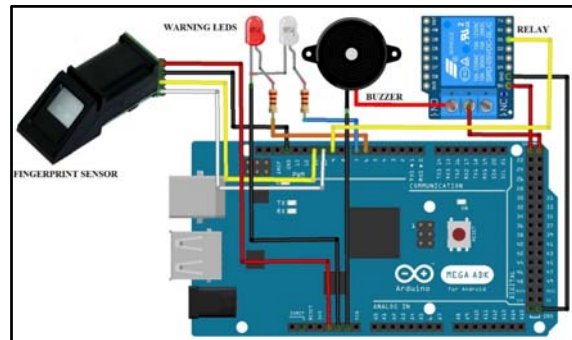


Figure 3b. Fingerprint Reading Part of SCS

Finally, in the FDCS, fire detection, MMS sending, IP camera control, and fire extinguishing (if desired) applications are carried out as shown in Figure 4.

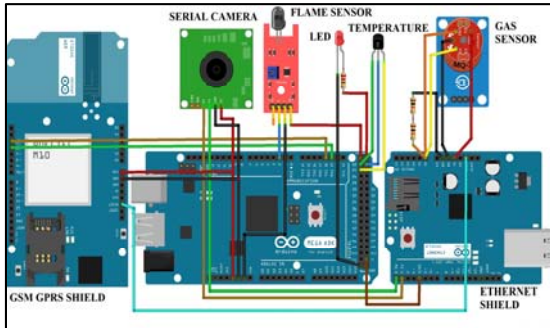


Figure 4. Fire detection control system

After the hardware installation, Microcontroller units were programmed by Arduino IDE. Arduino IDE software was used to write codes for microcontroller with ATME1 integrated circuit. All systems and the working principle of each part have been given with software codes together in [15].

2.2. Computer Programming Part

When the microcontroller software completed, server established to connect between the microcontroller and Android software. Windows Server 2012 R2 operating system was established to Server Computer. Furthermore, Firebird database software was installed on the Windows Server 2012 R2 operating system. Server management software was written by using Delphi compiler. This software helped to the administrator to adjust all parameters. The server application is like a bridge between plug and users. Moreover, desktop and web based software were written for users at the client side. Finally, the Android based application was written for the users by using Delphi compiler.

2.2.1 Writing of Software Codes

The software codes for the completed project are divided into server and client sides. The server software is the major structural component of the entire software and operates on a Windows-based server environment. Data from databases that run on servers is used by all client software programs. On a server running Windows, client software can also function. To provide access to the completed project from anywhere, Web and Android-based applications are created in addition to the more

dated Windows applications. The steps of the designed project are depicted in below.

- Database engines were researched and compared with each other. Firebird database engine was selected.
- Database and other database objects were created. Tables and records were also designed.
- Firstly, server program and then client programs were realized. All screens were designed respectively.
- All components were placed to the forms. Edit and combo boxes were linked to the database objects. Navigation and operation buttons were placed.
- List grids and Graph components were placed and linked to database objects. All graphics were adjusted to print.
- Report screens were designed and tested. All program codes were debugged and tested.

Besides, general features of all software platforms are explained as follows,

- For communication between the server computer and the ZigBee coordinator, specially developed server software is used. Between the server and the router/end node devices, the ZigBee coordinator device functions as a bridge. All sensor data are transmitted to the server computer, and data from the server computer are transmitted to all router/end node devices via the coordinator node device. Every device is capable of bidirectional communication.
- Sensor data is recorded to database tables, and this data needs to be routinely inspected.
- The screen displays, analysis and monitoring of data from sensors and controller circuits. The table and graphical charts can evaluate and monitor data in the database servers if necessary. For the measured data and the energy usage, a comprehensive report is created with tables and graphs. Client software allows users to examine the energy usage of any device.

Running systems can be automatically controlled in accordance with the established critical values. In addition, the web interface and Android devices can be used to remotely control all SBMS.

2.2.2. Choosing of the Right Software

The Server-Client architecture is the foundation of this software project. This kind of structures are supported by the majority of well-known integrated development environments (IDEs). In this regard, the server operating system, on which the server software is installed, is crucial. Server software can be operated under several operating systems. Windows Server environment was selected for this study because to its management, simplicity of installation, support for a wide range of technical documents, and compatibility with hardware. Client apps, on the other hand, ought to be used in desktop, mobile, and web browser settings. There isn't a development method that is supported by every environment. Investigation of IDEs reveals that the majority of development platforms only support desktop settings. At most two of these environments are supported by few of the others. Both PC and Android mobile platforms are supported by the RAD studio environment created by Embercadero. Delphi Builder and C++ are both included in it. The Pascal programming language, developed by Borland Software Company, is the foundation of Delphi. It contains a sophisticated debugger and a very quick compiler. Because to the aforementioned features, this study's recommended IDE was Delphi.

Applications based on Windows, Linux, and MacOS on the server and desktop client sides as well as IOS and Android-based applications on the mobile side can be produced using the Delphi IDE. Also, every platform can support the development of domestic, full-performance, and platform-specific client applications. The most powerful specification of this IDE is the ability to produce a single source file and build it for use in multiple domestic operating environments. This development environment offers a strong text editor but also a straightforward, user-friendly interface. The environment has numerous component repositories and integrated debugging tools in its text editor. Also, it contains a very quick compiler as well as a viewer for CPU registers and local variables. A million lines can be processed by this

compiler almost per minute. The DATASNAP technology was created by the Embercadero Company and implemented into their development environments. This technology offers server-to-client data transfer techniques as an object or object specification. TCP/IP is used to run this technology. As a result, thanks to the Internet, its connection potential is practically infinite. The data requests made by the client are translated into a procedure and function on the server side and run on the server in order to increase the efficiency of the built application. The collected data are then sent in plain text format. Applications improve in simplicity and performance using this approach. This data can, if desired, be subjected to filtering procedures including compression and encryption. The object-oriented programming method is supported by application development languages to speed up development and enhance code quality. The IDE's debugger may also locally or remotely debug source code. Delphi compiler is chosen for this study due to its exceptional qualities with the integrated development environment.

Firstly, the server software is created, and then desktop client software is created in a subsequent order. The software was created using Delphi XE 10.2, a developer version that can create outstanding programs for the 32-bit and 64-bit Windows desktop environments. Android mobile software is created utilizing components created for Android applications after the desktop application. Android source codes may be utilized without modification in MacOS applications upon request. Lastly, Cybele Software's Thinfinity VirtualUI program is employed to transform the desktop application into a Web application. Delphi XE 10.2 has robust code development facilities. Oracle, PostgreSQL, Interbase/Firebird, MySQL, Microsoft SQL Server, SQLite, and other well-known databases are among those that Delphi is permitted to connect to.

2.2.3 Choosing of a Database Engine

A database is typically required by computer software to store data. To store measured sensor values like current and voltage, the implemented

software also need a database. Similarly, all additional data, including calculated values and report data, should be kept in the database too. Databases should process data relatively quickly due to the large amount of data they store. Also, it should be portable and use a smaller amount of disk space. The chosen database should be installable and portable, and it should be able to run on Windows, Linux, and MacOS. It needs to be used with either 32-bit or 64-bit operating systems. The majority of enterprise class databases are capable of meeting these needs, however they are also very expensive and immobile solutions. Embedded databases are portable, fast and easy manageable products. They also require less disk space and don't require installation. Several open source and cost-free embedded databases are available.

It is discovered that some embedded databases don't support Structure Query Language (SQL) when the embedded databases are contrasted with one another. All of the data in a database may be examined, filtered, and managed with the help of SQL. Application processing speed is the most crucial component for processing enormous amounts of data. SQL is quick and effective because it was created for big data. A database in the finished project must therefore support SQL.

Firebird is a quick, portable, scalable, open source database. Moreover, it supports SQL. It can be used with a variety of operating systems, including Windows, Linux, Macintosh, and Solaris. It is possible to move Firebird data from one platform to another. There are three main versions of Firebird. Super Server version for Windows environments is the first of them. The second is a multi-user version of Super Classic. It is compatible with various operating systems, including Windows and Linux. The third iteration of Firebird is the portable, compact, and powerful database engine known as the embedded version. The Firebird Embedded Database Engine fully satisfies the necessary characteristics for the project as it was accomplished.

2.2.4 User Interface Design

The user interface in the completed project was created using page logic. The majority of people have used Microsoft Office programs. Applications for Excel often feature a single main window and numerous tabs. The design of these tabs provides facility to manage a lot of different page at the same time. The accessibility and usability have improved with this design. Nothing is more than a mouse click away. An objective-based form design appearance manager is available in Delphi. With the appearance manager, users can select and modify the application's views.

2.2.5 Miscellaneous Programs

At the development stage, a few different miscellaneous programs were employed. One of them is IBExpert software. A database can be created using the program IBExpert. It is utilized for managing and maintaining the Firebird database engine. Database objects can be created such as a database, table, index, view, etc. IBExpert may be used to modify database object properties concurrently. The maintenance of the database is executed by this program. Creating and maintaining the database structure on the disk were necessary for the proposed project, to put it briefly.

Unidac is a component of Delphi. It is provided most known database engine connection components. Every database object has logical components in Delphi environments.

Quick Report is a report builder component in Delphi. From the database table, reports in great detail can be generated automatically. With this component, reports can be printed to a printer, saved to disk, or previewed on the screen.

Teechart is a graphic drawing component in Delphi. This component allows for the direct graphic rendering of database-stored data. There are options for graphic styles, color series, and more. It is possible to preview, print, or save a graph.

After choosing the development environment, database engine, and other development tools for this project, the database design is prepared.

2.2.6 Database Design Basics

Designing of a database comprises two stages as logical database design and physical database design. The variables that are represented and the database regions that are utilized throughout the design stage of the logical database determine the design stage. Variable types and sizes are decided simultaneously. The creation of logical objects occurs in the physical stage.

In the realized project, database object SmartControl.FDB was created. There are five database tables in the database file. Device table is one of them. It keeps track of the devices' data. The SENSORS table is the following table. It keeps track of device sensor data. It stores the sensor information of devices. The third table is SENSORLOGS table. It stores the sensor values periodically. The fourth table is VALUES table. By computing the program based on the sensor table data, it records the measured values. A MEM_VALUE table makes up the final table. By reading frequently, it records the various sensor log values for use in reports and graphs. SmartControl is the primary database file. As seen in Figure 5, FDB was produced via the IBExpert program.

Using this procedure, the design of database tables and other objects such as domains, indexes, and so on was initiated. The table's column names, sizes, types, and other specifications were established and generated as shown in Figure 6.

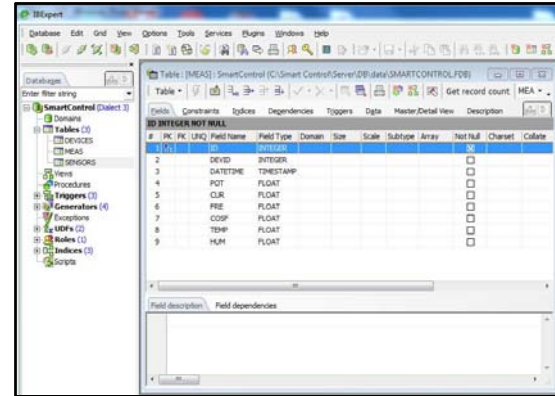


Figure 6. Table creation stage

Indexing should be utilized in the table for fast access to the data. The main key in a table can only be one. On the other hand, a foreign key can be used to link the primary key of one table to another. Hence, two tables will be associated to one another. The startup screen was primarily made for applications, as seen in Figure 7. This screen is a login screen, and it has header section, login section, and status bar section. The program name is listed as Smart Building Management System in the header screen.

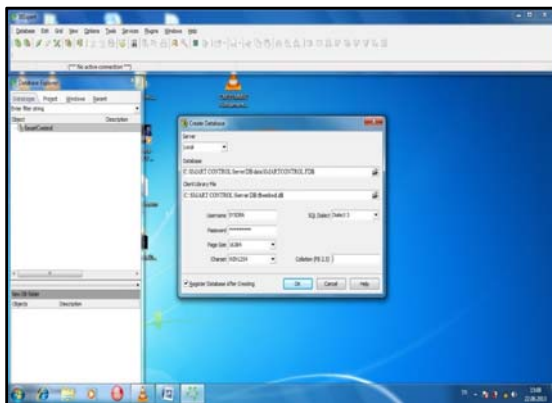


Figure 5. Database creation stage with IBExpert



Figure 7. Welcome screen of the realized software

In the login section, user name and password should be known in order to enter the application. Each person is given a special username and password, which are then recorded in the database as a password table while the allowed individuals are identified. The software will be entered successfully if the username and password match. The user cannot access the program in any other case. Lastly, certain program-related information is provided in the status bar area. In particular, this section can show if the COM port is connected or not, whether the correct password is known or not, and other information. Also, the status's date and time values are recorded to the database's logging table. Hence, if needed, the achieved status can be traced back step by step.

Following this phase, a database form was built to hold all database objects. This form has a database element called SmartControl. FDB stands for a physical database file on a disk. Four database components serve as representations for each database table. A table component, which is the first part, symbolizes the actual database table in the database file.

The second one manages each read or write activity to the table object through a transaction component. This component is used to either write newly entered or modified data to a physical disk file or to undo an operation.

The third part, an update part, is in charge of keeping track of the SQL statements for each action, including Insert, Update, Cancel, Delete, Save, and Refresh operations. When the user clicks a button on a form to begin or end a database action, such as beginning data entry, changing a record, deleting a record, and so forth. The database engine processes the SQL statements from the table components by adding some arguments and processing them.

The data source component, which makes connections between the table and all other database components, is the fourth component. It serves as a link between table objects and their component parts. All elements connected to database objects are depicted as shown in Figure 8. This screen also named as appearance screen is where the background image can be added. The appearance

list box also allows the user to choose among the default views. On each program screen, a chosen view or an additional image will be displayed.



Figure 8. Database objects screen

Following this phase, the main screen, displays for reading sensor information, logs, reports, and additional screens were built. Figure 9 displays an example of a value reading screen. Sensor reading values are graphically displayed on this screen. By double clicking the mouse, users can zoom in on these images and view them in full-screen mode, as well as print them off if they so choose.

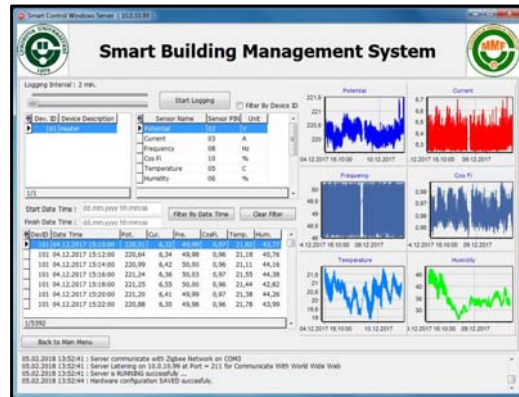


Figure 9. Sensor values reading screen

There are four tabs on the realized project screen. The home screen is the first tab. The screen for this tab comprises four sections. The first section is Hardware Configuration section. The second section is appearance setting area. The second section has the Windows program appearance

settings. The appearance and color settings of the whole program can be changed. All of the fundamental operations of smart systems can be examined in the third section. Periodically checking the fundamental values is possible. Graphics are located in the fourth area. The sensor values can be viewed in this area as a graph by day as well as in database values.

Zigbee devices and computer interface communication parameters can be changed in the hardware configuration section. Moreover, the sensors' maximum and minimum reading values are determined as warning values. The application will issue a warning when the read values fall below or rise above the warning values. If it is desired, the system will be shut down automatically. In other words, the crucial sensor values and serial port connection parameters can both be configured simply using this screen. Figure 10 displays the warning limits adjustment screen and hardware settings. Use the Return to Main Menu button to move past the previous screen.



Figure 10. Hardware configuration and warning limits adjustment screen

Functional Test is the name of the second tab. The device number can be provided in this page, after which the desired pin number can be tested. Analog and digital write and read functions, as well as basic microcontroller functions, are also controllable. The values that the functions returned after being executed can also be seen. Online checking is available for the reading sensor values, including potential, current, efficiency, and frequency. Functional test screen is depicted in Figure 11.



Figure 11. Functional test screen

Each function may be periodically read as well as separately in the functional test screen. The user can specify the period time. Each reading value is a collection of actions that can be read. All integrated cards can be tested here prior to commissioning. As a result, the likelihood of an error will be reduced. The monitoring of reading values, whether manually or automatically, is done on this screen concurrently. Selecting the Auto Updating Check box and choosing the interval time are prerequisites for the automatic mode.

Appearance Set Screen is the name of the third tab. In this screen, the background image can be set. Also, the software chosen from the list's appearance can be modified. Appearance settings screen is shown in Figure 12.



Figure 12. Appearance settings screen

The saved modifications should be saved by clicking the Save Configuration button before leaving this screen. Otherwise, all modifications will be invalid.

The program coding stage can begin after the screen design and component connectivity stages are complete. Delphi code editor for writing program is illustrated in Figure 13. The code editor has menus for each component's properties and events. All screens, including the logging screen, report screen, and their associated codes, were effortlessly accomplished with the aid of all components. The code editor includes a lot of useful features. Code compilation is one of them. When writing the code, the code editor highlights the incorrect code.

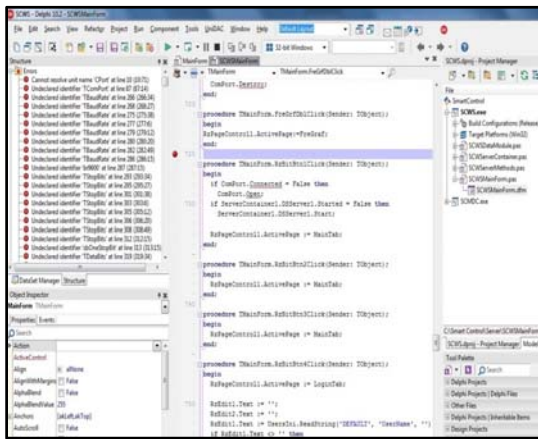


Figure 13. Delphi code editor and debugger

In Delphi, a breakpoint can be added to desired lines by the development environment. The program debugger can be run. The program debugger tracks all writing codes as well as lines with breakpoints, which cause the compiler to halt an operation and display the breakpoint lines or writing errors in the code editor. The code can be followed and examined variables' values in Delphi debugger. Additionally, the program can run the codes line by line and step by step, changing the codes based on the values of the variables. It is a simple approach to identify incorrect code or illogical design flaws. Once the errors have been found, they must be fixed before the updated codes may be compiled. Finally, software without any errors should be converted into an application.

After the code was built and the complete form was designed, the software for the completed project ran smoothly. As a result, both the desktop client program and the server application were largely finished. Android based mobile application is developed by Delphi code editor. Android based components are different from the desktop components. For software based on Android, new screen designs should be implemented due to this distinction. According to mobile device sizes, design screen sizes can be selected. In android software, three different screens are depicted together. The first screen is the connection to the main server. The IP address of the primary server computer should be supplied on this page. In the second screen, main menus of the Android software are shown. The reading of data from all sensors is done on the third screen. Once the desired coordinator or end node number has been entered, access to the desired data will be available. Android based software screen samples are depicted in Figure 14.

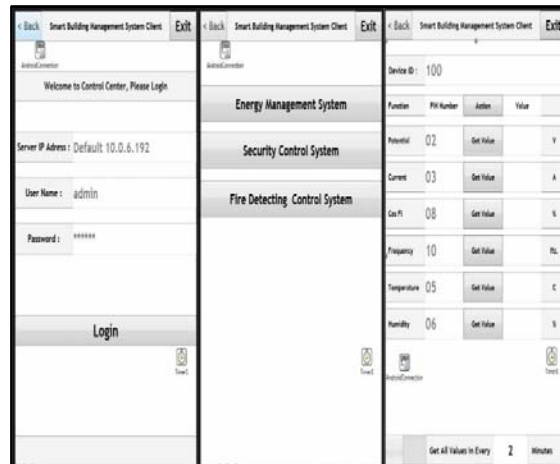


Figure 14. Realized Android based software screen samples

Finally, after some small revisions, the application is turned into Web-based by Thinfinity VirtualUI software written by Cybele Software Company as shown in Figure 15. Thanks to this software, the existing software has been made web-based without the need for a different programming language.

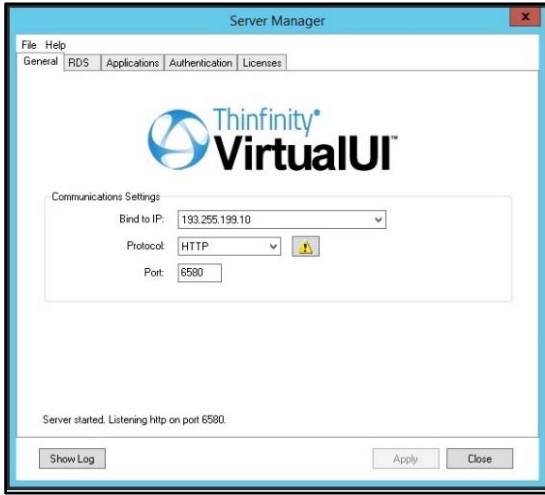


Figure 15. Thinfinity VirtualUI software screen

3. RESULTS AND DISCUSSION

In this study, all of the embedded cards are classified into two groups with respect to the purpose of use. As a consequence of the classification, a single software program code for each class has refined for standardization. If there is a need for additional new cards to the system later, one of the standardized code lines will be loaded and executed according to the intended use of the card. Standardized codes and load over an embedded card with a limited resource has been lightened by transferring to the server. The limited resources that embedded cards have too small in comparison with the resources that advanced computers have. This study is the first of its kind in the literature when similar studies in the past are investigated.

After physical installation and coding of embedding cards are done, a Windows based desktop application operating on the server computer has been developed for reading data on these cards, storing the read data, analyzing the stored data, reporting, and management. Created software by using Delphi programming language is the main software program of SBMS. All cards and all parameters belong to these cards are identified by using this software program. Furthermore, all of the obtained data are stored via this software program

in a database file on the same server. FireBird database has been chosen as being open source, free, and compatible with Delphi. Software of the server is not only used for all identifications and parameter settings by people who have a high level of administrative authorization but also holds all data and reports that users can access at the same time. Users can remotely access and control devices by connecting to server software. Users can either connect to SBMS via a desktop application or different applications developed for web-based or mobile devices. Instead of using separate languages for user platforms, the Delphi compiler, which can be converted to an Android-based application by making minor changes in the desktop application, has been chosen as the programming language. Additionally, the application has been turned into web-based by Thinfinity VirtualUI software of Cybele Software after some small revisions. Hence, users can govern devices at the remotest point according to their pre-defined authorizations in the system via three different graphical interfaces in order to access SBMS and database. Obtained data from measurement and monitoring in the SBMS have been analyzed.

Realized SBMS is parametric and has flexible structure. The conducted study is expendable without depending on a specific brand. This project can be utilized in different projects for diversified purposes. At the end of the study, it was determined that the responsible personnel were informed by sending an sms in case the determined limit values were exceeded or below. Relevant personnel can control the system in real time with mobile, web or desktop application.

4. CONCLUSION

In this study, information related to programmed server and client-side software is given. Client-side software is based on Windows-based operating system and it is executed on Windows 2012 Server. FireBird database is chosen instead of SQL Server for being open source. Rather than using separate languages such as C#, ASP.NET and MIT App Inverter for user platforms, the Delphi compiler, which can be converted to an Android-based

application by making minor changes in the desktop application, is chosen as the programming language. Therefore, users can manage devices at the remotest point in the system via three different graphical interfaces in order to access SBMS.

Developed SBMS has a flexible structure. Any device having analog or digital inputs can be included in the system by making small electronic modifications. It should be noted that input and output voltages of embedded card used as microcontroller are $5 V_{dc}$. The conducted study is expandable without depending on a specific brand.

This project can be utilized in different for diversified purposes. A part of the developed project is commissioned in the main data processing room at a University Hospital.

Consequently, in this study, the implementation of a wireless network for a low costed monitoring, warning, and analysis in data processing room is carried out. As a result of the study, it has been seen that the established system can perform the same task much cheaper than the ready-made applications that do the same job in the market.

5. REFERENCES

1. Gokce, H.U., Gokce, K.U., 2014. Multidimensional Energy Monitoring, Analysis and Optimization System for Energy Efficient Building Operation. *Sustainable Cities and Society*, 10, 161-173.
2. Santis, DD., Giampetruzzi, A., Abbatantuono, G., Scala M.L., 2016. Smart Metering for Low Voltage Electrical Distribution System using Arduino Due. 2016 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS), 13-16 June 2016, Bari, Italy, 1-6.
3. Behan, M., Krejcar, O., 2013. Vision of Smart Home Point Solution as Sustainable Intelligent House Concept. 12th IFAC Conference on Programmable Devices and Embedded Systems: The International Federation of Automatic Control (IFAC), Ostrava, Czech Republic, 25-27.
4. Hussein, A., Adda, M., Atieh, M., Fahs, W., 2014. Smart Home Design for Disabled People based on Neural Networks. Elsevier: *Procedia Computer Science*, 37, 117-126.
5. Gad, H.E., Gad, H.I.E., 2015. Development of a New Temperature Data Acquisition System for Solar Energy Applications, Elsevier: *Renewable Energy*, 74(C), 337-343.
6. Rahman, M.M., Jannat, N.E., Islam, M.O., Salakin, M.S., 2015. Arduino and GSM Based Smart Energy Meter for Advanced Metering and Billing System. IEEE 2015: 2nd International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) May 2015, Dhaka, Bangladesh.
7. Ismail, N.H., Tukiran, Z., Shamsuddin, N.N., Saadon, E.I.S., 2014. Android-based Home Door Locks Application via Bluetooth for Disabled People. IEEE 2014: 2014 IEEE International Conference on Control System, Computing and Engineering, November 2014, Penang, Malaysia, 191-195.
8. Wang, Y., Chi, Z., 2016. System of Wireless Temperature and Humidity Monitoring Based on Arduino Uno Platform. IEEE 2016. 2016 Sixty International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), Harbin, China, 770-773.
9. Ramadhani, M.F., Yulianto, B., Nugraha, 2013. Design and Fabrication of Real Time Air Quality Monitoring System Based on Web Application. 2013 3rd International Conference on Instrumentation Control and Automation (ICA) Bali, Indonesia, 28-30.
10. A. Mackey, 2010. *Introducing the .NET 4.0 with Visual Studio*, Apress, ISBN: 978-1-4302-2455-6.
11. Griffiths, I., Adams, M., and Liberty, J., *Programming C#, 6th Edition*, 28 Aug 2010.
12. Arduino, *Guide and Introduction*, 2016., <http://arduino.cc/en/Guide/Introduction>
13. Timur, O., Zor, K., Celik, O., Teke, A., 2018. Enhancement of a Low-Cost Intelligent Device for Improving Energy Efficiency in Buildings, *Commun. Fac. Sci. Univ. Ank. Series A2-A3*, 60(2), 103-128.

14. Timur, O., Teke, A., 2022. Design and Implementation of Microcontroller Based Split Air Conditioner Control System, Commun.Fac.Sci.Univ.Ank.Series A2-A3, 64 (2), 117-128.
15. Timur, O.,2018. Design and Implementation of a Wireless Sensor Network for Energy Monitoring, Analysis and Management in Smart Buildings. PhD Thesis, Çukurova University, Institute of Natural and Applied Science, Electrical&Electronics Engineering, Adana, 215.