SYSTEM DESIGN FOR IMPROVING OUR HOME COMFORT

M.D. Stroia, *Babeș-Bolyai University of Cluj-Napoca, Reșița, ROMÂNIA*
C. Hațiegan (corresponding author), *Babeș-Bolyai University of Cluj-Napoca, Reșița, ROMÂNIA*
Cristinel Popescu, *Universitatea Constantin Brâncuși, Târgu-Jiu, ROMANIA*
S.V. Bugar, *Atos it solutions and services, Timișoara, ROMÂNIA*

ABSTRACT: We live in times where everything we do is on fast-forward, from our most simple to our most complex activity. More than that, we are all into multi-tasking thinking and doing, therefore a system that would ease all up would be more than welcome into our lives. With that in mind, we thought considering a system design that could be helpful with enjoying simple routines right in the comfort of our own homes, where we can escape this rush century. The system we implemented is a design, hardware and software, which can be adjusted, with required architectural changes, for a real home. For proposed design we used the "fashionable" IoT principles, electronic components and a software application.

KEY WORDS: smart device, IoT, Arduino, home comfort, automation.

1. INTRODUCTION

"Comfort" has become a rare and desired aspect of our nowadays, whether we speak in terms of comfort at our jobs, other outside activities or in our homes.
Home comfort systems are high technologies which transform a home into a "smart" house. In general, a smart house involves automatic lighting systems, temperature control, safety, etc., which integrates electronic devices, device controllers and a telecommunication system [1-6].
In present times, system home comfort has come to developing more and more high performance and reliable automatic system technologies, such as wireless, Bluetooth, satellites, radio.
With this in mind, we considered developing and implementing a system design intended to assure a minimum comfort in our homes, all the more so as, in our country, such type of system is far from a priority.

In designing the home comfort system, we considered two aspects: hardware and software.
For the hardware part we have built a house model where we implemented needed electronic components, based on Arduino technology, which were connected to an Android system mobile [7-11].
The software part consists in an application installed and functioning on an Android system mobile. By its mean we can activate different functions of home comfort system. Both components design will be described in the following.

2. HARDWARE PLATFORM DESIGN

First step in hardware platform design was to create a home environment. Walls were built from Styrofoam material, for furniture we used wood, on the courtyard synthetic carpet was set and appliances, like TV, audio system, cooler, sprinklers, were executed with a 3D printer, as shown in figure 1.

For simulating the functionality of appliances, various electronic components were use as solution. For example, function of turning on the TV is implemented using a led, function of starting on the cooler system is implemented using a general purpose engine.

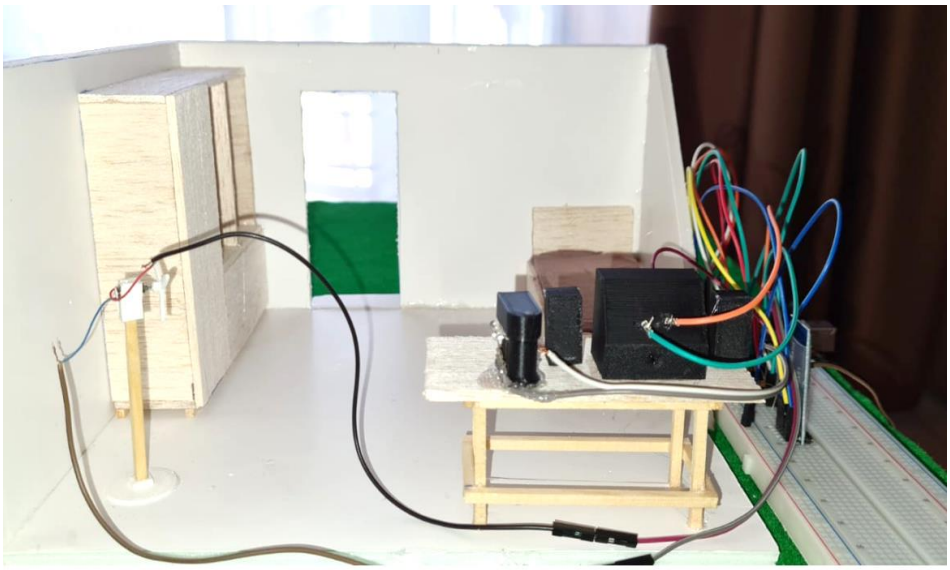


Figure 1. Hardware platform-home environment

For wiring appliances to Arduino Uno board we used an intermediate component, a breadboard [11-13]. Communication between devices was made by means of Bluetooth module HC-05 mounted on breadboard (figure 2). We also used jumpers for controlling opening/closing system circuits.

Arduino Uno board is provided with ATmega328P microcontroller and has available 14 I/O digital pins and 6 I/O analog pins which can be programmed through Arduino IDE [12, 14-16]. For supply we used an USB cable, but a 9 voltage battery can also be used.Wiring connections can be seen in figure 2.

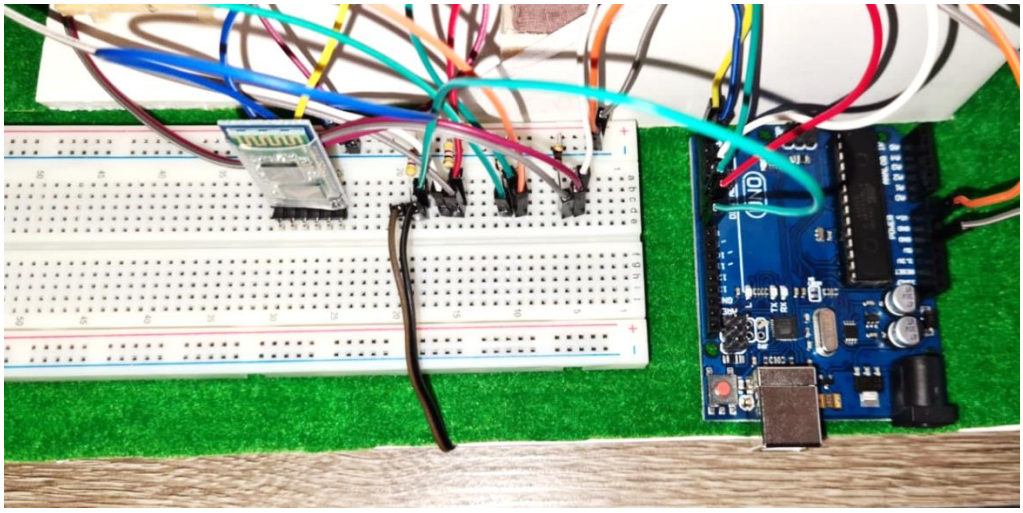


Figure 2. Hardware platform-wiring connections

2. SOFTWARE IMPLEMENTATION

The software for system home comfort is divided in two programing codes, a code for Android application and a code for Arduino board.

Android application is in fact the interface between front user and home system. It is used to establish connection between smartphone

and Bluetooth on breadboard for communication part, as well as to make actions which generates events for appliances control [17-21]. For each appliance there is a command button in the application that has a functionality and generates a corresponding action. Created buttons have the name of the appliances. For this system design we have built 6 buttons that can be in one of two states, on or off, meaning activated or deactivated [21-23]. So, if one leaves home and forgets the light on, for instance, it can be turned off using the smartphone.

Figure 3 highlights the code used in buttons programming.

```
<Button
    android:id="@+id/btOn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="On"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/btOff"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.121" />

<Button
    android:id="@+id/btOff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Off"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.987"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.121" />
```

Figure 3. Android interface – code sequence

bluetooth_smarthome_3LED

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

int livingroom = 5;
int bedroom = 6;
int diningroom = 7;
int sprinker = 8;

SoftwareSerial Bluetooth(0, 1);
char Data;
void sendData(String transmitData){
Bluetooth.println(transmitData);}

void setup(){
    Bluetooth.begin(9600);
    pinMode(livingroom,OUTPUT);
    pinMode(bedroom,OUTPUT);
    pinMode(diningroom,OUTPUT);
    pinMode(sprinker,OUTPUT);
}

void loop(){
    if(Bluetooth.available()){
        Data=Bluetooth.read();
        if(Data==('7')){
            digitalWrite(sprinker,1);
            sendData("sprinker ON");
        }
        if(Data==('8')){
            digitalWrite(sprinker
            ,0);
            sendData("sprinker OFF");
        }
        if(Data==('4')){
```

Figure 4. Arduino Uno – code sequence

The programming code for Arduino Uno is written in C++ language, due to its flexibility and scalability. Arduino IDE offers the facility to error check the code before is load on board [13, 21]. Another important step is disconnecting Rx and Tx pins from Arduino board before loading the programming code [23].

A code sequence for Arduino Uno is presented in figure 4. We have chosen to activate the Bluetooth module through Android application, so if we do not execute this step first of all, the system will not function.

For each function of the system home comfort there is an activation button on Android user interface which generates an event sent to Arduino Uno which will send a command to the appliance we select [6, 12,13]. For every on/off command we set a numeric value to trigger the event [1, 6], for example the cooling system (named Ventilator) activation button has associated number 4 for "ON" state and number 1 for "OFF" state.

3. USING THE APPLICATION AND FUNCTIONING TESTS

The Android application has a user-friendly interface. In figure 5 we are already inside the application, at the button controls screen. At this point, as it can be seen, we have 4 appliances connected to the system home comfort: a TV, a cooling system, a coffee machine (Espressor) and a sprinkler. We have the option to activate each one at a time as well as activate them all at once. We have Button 6 designed for system extensions.

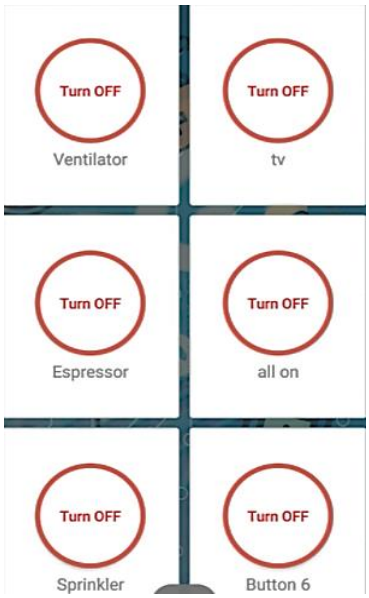
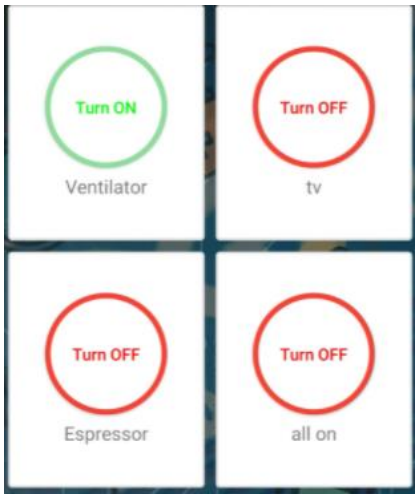


Figure 5. Android application – button screen

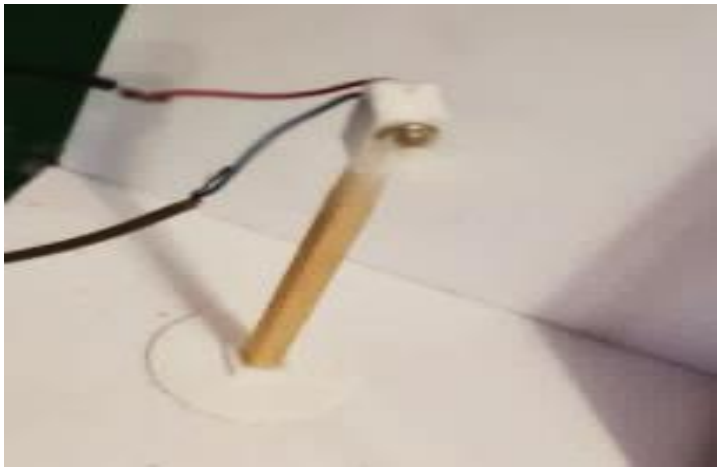
Now we will show how the Android application is used and how the designed system home comfort functions.

When we touch a button from application window shown in figure 5, let's say, for instance, button corresponding to cooling system, the button will turn green and will display "Turn ON". That means a signal was sent from smartphone, through Bluetooth device (figure 2) to the Arduino Uno board which triggers a command signal sent to the cooling system. The cooler will start functioning, in this case being simulated by the 3D part representing a cooler spinning.

As we prefer, we can now turn off the cooler or let it in function, and touch another button corresponding to a different appliance, in this case let us choose the TV. So we touch the second button (tv), which turns green, being in "ON" state, action that starts a similar series of action-event states, signals following the line Smartphone-Bluetooth-Arduino-TV. All the rest of the buttons function in the same way. What we have just exposed above is represented in figures 6-a, b and 7-a, b, where we activated one button at a time on the application and we can see the functioning results of appliances.

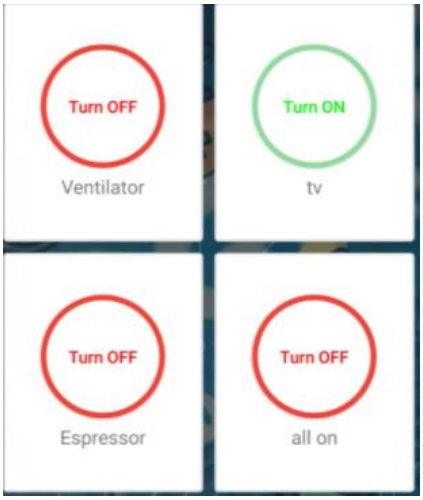


a) cooler button activated

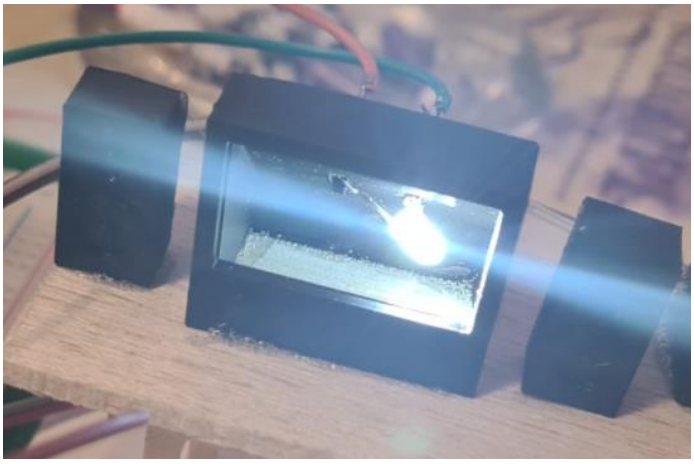


b) cooler blades spinning

Figure 7. Application – Turning on the cooling system



a) TV button activated



b) TV led functioning

Figure 7. Application – Turning on the TV

4. CONCLUSION

The design we implemented and tested was functional, even though is not a 1:1 model for house or appliances. We plan to expand the present design in terms of adding other possible to simulate functions like door locking and unlocking, window opening, as well as updating the programming code for the system to work on voice commands or with timers for function deactivation on delay.

The true challenge is to implement such a system on a real house. This would imply definitely higher costs, and more performant devices for communication, an active internet connection and changes in appliances electrical schemes.

REFERENCES

[1]Anghel-Drugărin, C.V, Stroia, M.D., Drăghici, S., Relative Research Upon Data Transmission Channels, The 2nd year of Global Virtual Conference, 2014

[2] Stroia Mihaela-Dorica, Hatiegan Cornel, Popescu Cristinel, Virtual Instrument Designed for Detecting Distortion Regime Caused by Frequency Variation, Analele Universităţii Constantin Brâncuşi din Târgu-Jiu - Seria Inginerie, Nr. 4, 2020.

[3] Stroia Mihaela-Dorica, Hatiegan Cornel, Popescu Cristinel, Virtual instrument designed for data acquisition, Studia Universitatis Babes-Bolyai Engineering, Vol. 65, Nr. 1, 2020.

[4] Eugen Raduca, Dan Ungureanu-Anghel, Nicolina Pop, Sorin Florea, Mihaela Raduca, Cornel Hatiegan, M Ayaz Ahmad, *Determination of prime implicants of a logic function trough the implementation of Quine Mccluskey method in LabVIEW,* Advances in Environmental Biology, 10(3) March 2016, Pages: 186-191, 2016.

[5] Stair, R. M., Reynolds, G.W., Principles of Information Systems. A Managerial Approach-Ninth Edition, Course Technology, Cengage Learning, 2010.

[6] Stroia M.D., Anghel, D., Moșteanu D.E., Hațiegan, C., Communication Interface Prototype Used for Data Transmission at Electric Systems, International Conference Knowledge-Based Organization, Vol. XXV, No 3, 2019

[7] E Răduca, D Ungureanu-Anghel, L Nistor, C Haţiegan, S Drăghici, C Chioncel, E Spunei, R Lolea, *Web server with ATMEGA 2560 microcontroller*, International Conference on Applied Sciences 2015 (ICAS2015), IOP Conf. Series: Materials Science and Engineering 106, 012018, 2016

[8] Eugen Raduca, Lucian Nistor, Cornel Hatiegan, Mihaela Raduca, Ioan Padureanu, Silviu Draghici, Web server for command, control and monitoring of industrial equipment,2015 9th International Symposium on (ATEE), 7-9 May Bucharest, 2015.

[9] Ioan Pădureanu, Marcel Jurcu, Ladislau Augustinov, Cornel Haţiegan,Eugen Răduca, Laurenţiu Pădeanu, Optimisation of the Start-up and Operation Regimes of Cooling Water Pumps of a High-Power Hydro Generator, Analele Universităţii "Eftimie Murgu", Nr. 1, 2015.

[10] Ioan Pădureanu, Marcel Jurcu, Ladislau Augustinov, Cornel Haţiegan, Eugen Răduca, Implementation of an Automatic System for the Monitoring of Start-up and Operating Regimes of the Cooling Water Installations of a Hydro Generator, Analele Universităţii "Eftimie Murgu", Nr. 1, 2015.

[11] Stroia Mihaela-Dorica, Hatiegan Cornel, Popescu Cristinel, Virtual instrument designed for data acquisition, Studia Universitatis Babes-Bolyai Engineering, Vol. 65, Nr. 1, 2020.

[12] Blum, J., Exploring Arduino: Tools and Techniques for Engineering, California: Wiley Publishing, 2019.

[13] Boxall, J, Arduino workshop, California: No Starch press, 2013.

[14] Pocatilu, P. et. all, Programarea aplicațiilor Android, ASE publishing, 2015.

[15] Stroia M.D., Derbac D., Hațiegan C., Cîndea L., Thermostat model with Arduino Uno board for controlling a cooling system, Annals of "Constantin Brâncuși" University, Târgu Jiu, 2018

[16] M. Stroia, D. Moşteanu, I. Virca, E.Răduca, C. Popescu, C. Haţiegan, Case studies for automotive components using CAD and CAE techniques, International Conference on Applied Sciences ICAS 2019, May 9-11, Hunedoara, Romania, 2019.

[17] Gerber, A., Craig, C., Learn Android Studio, Apress Publishing, 2015.

[18] Stroia Mihaela-Dorica, Hatiegan Cornel, Muscai Cristian, Simulating an improved algorithm for propagation of transverse oscillations through a string, Vol. 65, Nr. 1, 2020.

[19] Hatiegan Cornel, Stroia Mihaela-Dorica, Popescu Cristinel, Muscai Cristain-Mircea, Application for Simulating and Analysis of a Serial R-L-C Circuit, Analele Universităţii Constantin Brâncuşi din Târgu-Jiu - Seria Inginerie, Nr. 3, 2020.

[20] Cristinel Popescu, Cornel Hatiegan, Aspects regarding the Calculation of the Dielectric Loss Angle Tangent between the Windings of a Rated 40 MVA Transformer, Analele Universităţii "Eftimie Murgu", 2, Reşiţa, 2015.

[21] Kushnir, V., *Safe C++*, United States: O'Reilly Media Publishing, 2012

[22] Loizides, F et all, Human Computer Interaction and Emerging Technologies: Adjunct Proceedings from the INTERACT 2019 Workshops, Cardiff University Press, 2020.

[23] https://www.android.com/

[24] https://www.arduino.cc/