

AN APPROACH TO SMART HOME SECURITY SYSTEM USING ARDUINO

Abel A. Zandamela

The Sirindhorn International Thai-German Graduate School of Engineering
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand

ABSTRACT

Recent advancements in the field of Internet of Things (IoT) have turned this area in one of the most promising research topics to bring innovative and useful technologies in the future. Solutions for a low-cost and efficient Home Automation model have always been a widely addressed problem; This study focuses on the architecture and design of wireless, flexible and inexpensive smart home system. Using Mega 2560 Arduino Board Platform, we develop an intelligent system able to send real-time video and GSM-based information of a break-in, fire and motion detection, along with an advanced temperature and humidity monitoring system. A prototype is designed, and tested on China Mobile Network, in 20 samples the prototype shows a 9.4s average time of response.

KEYWORDS

Smart Home System; Arduino; IoT; Mega 2560; GSM; wireless

1. INTRODUCTION

Internet of Things has emerged as one of the most promising technologies for the future [1,2]. This field is actively researched, and different solutions have been proposed to address the challenges in this area, such as limited amount of energy and cost-efficiency [3]. One of the most discussed topics in IoT is the Home Automation, developing an inexpensive and safe system for indoor use has been a widely researched area which has brought advances in technology and availability of small, flexible, and smart systems. Guided by the concepts of IoT we develop an intelligent security system designed to give surveillance at all time [4,5,6,7].

Mobility is a crucial part of the development of monitoring systems. Therefore, this system provides security reports even when the user is away from home, using GSM mobile technology which performs remote communication wherever the user is located, this is achieved by using a GSM module-SIM900A. A Wiznet 5100 Arduino Ethernet Shield is used to establish a Network Connection between the Arduino Board and the internet, which enables us to set up a website for system monitoring. Temperature and Humidity control system is accomplished by adopting DHT11 sensor which is a digital signal-acquisition temperature and humidity sensor which is attached to the Mega 2560 microcontroller. Also, an Android-based application is implemented for the overall intelligent home system control.

SYSTEM DESIGN

Figure 1 shows the overall system block diagram and the proposed functionalities of the design; we divide the system into two main units: data acquisition and system control. The data acquisition is modeled using the ATmega2560 microcontroller, DHT11 sensor, PIR sensors, ASR-38B magnetics sensors, flame sensor, Photocell and a matrix array keyboard. Control system is through single-chip microcomputer ATmega2560 controlling the GSM Module SIM900A, Wiznet 5100, IP Camera, then the real-time display of the received information is achieved using LCD1602 and the web interface for the real-time video.

The power supply part of the system is shown in the square through the PCB. Also, a 9V power supply is used for the SIM900A module.

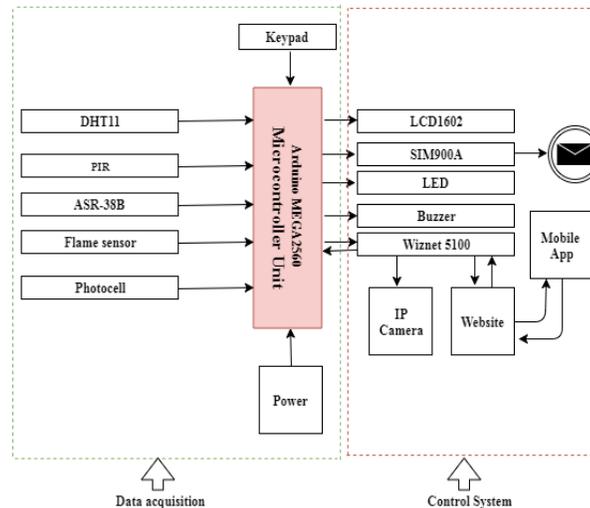


Fig.1. Block diagram of the Smart Home Security System

HARDWARE DESIGN

A. Arduino Mega 2560 microcontroller

The Arduino Mega 2560 is a microcontroller board based on Atmel ATmega2560 microcontroller. Mega 2560 is a low-cost and highly flexible board platform, these characteristics along with the existence of open-source projects for Arduino boards simulations such as Fritzing, place Arduino boards as one of the most widely used microcontrollers for research proposes, with particular focus in prototypes of sensor systems.

Arduino Mega 2560 is a microcontroller consisting of 54 digital input and output pins, a 16 MHz crystal oscillator HC-49S, a USB type A to B connector, an audio power jack, an ICSP (in circuit serial programming) header, and a reset button. Its operating Voltage is 5V; the recommended input Voltage ranges from 7 to 12V; nevertheless, the limits of the input can be placed between 6 to 20V. The DC Current per I/O Pin is 40mA, and the DC Current for 3.3V Pin is 50mA. Its internal flash memory is of 256 KB of which 8KB is reserved for the bootloader, the SRAM, EEPROM are 8KB and 4KB, respectively.

This board can be programmed using the Arduino Language, which is a high-level language, a straightforward description places Arduino as being merely a set of C/C++ functions that can be called from Arduino Code, then passed to a C/C++ compiler (avr-g++) after undergoing minor changes.

One of the crucial key factors in the design of an Arduino prototype is a thorough understanding of the platform interconnections, therefore is important to understand the overall pin distribution. In the 54 digital pins on the Mega 2560, 14 can be used as pulse wave modulation (PWM) outputs, 16 are reserved for analog inputs and four adopted for universal asynchronous receiver-transmitter UARTs, i.e., hardware serial ports. The whole board pin particular function can be summarized as:

- Serial pins: 0 used to receive TTL (Time-to-live) serial data often represented by RX; 1 used to transmit TTL serial data and expressed by(TX); 1:19 reserved for (RX) and 18 for (TX); 2: 17 for (RX) and 16 for (TX); Serial 3: 15 (RX) and 14 (TX).
 - External Interrupts pins: four interrupt modes are supported by the Arduino Mega 2560; Rising (which activates on rising edge), Falling (which activates on falling edge), Change (its responds to any change in the interrupt pins value) and Low (triggered by digital low); The interrupt pins are: 2,3,21,20,19 and 18, for interrupt 0, interrupt 1, interrupt 2, interrupt 3, interrupt 4, and interrupt 5, respectively.
 - PWM pins: 0 to 13. They provide 8-bit PWM output.
 - Serial Peripheral Interface (SPI) pins: these pins provide synchronous serial data protocol for the microcontroller communication peripheral devices. The pins are 50 MISO (Master In Slave Out), 51 MOSI (Master Out Slave In), 52 SCK (Serial Clock) and 53 SS (Slave Select).
 - LED: Built-in LED connected to digital pin 13
 - I2C (Inter-Integrated Circuit): pin 20 for Serial Data (SDA) and pin 21 for Serial Clock (SCL).
- The figure below shows a schematic of pin distribution on the Arduino Mega 2560 board.

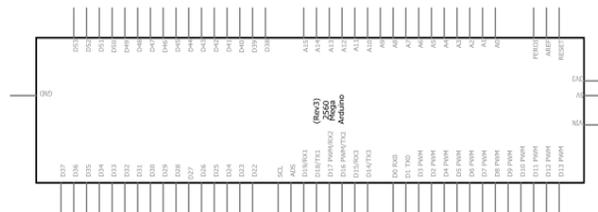


Fig.2. Arduino Mega 2560 Pin distributions

B. GSM Module SIM900A

The SIM900A is a complete Dual-band GSM/GPRS module in an SMT (Surface-mount technology) type. Is small and cost-effective, which features an RS232 interface, allows

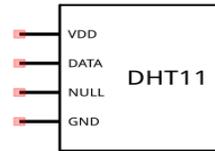


Fig. 4. DHT11 sensor overall pins distribution

E. PIR & Flame sensor

Passive Infra-Red or Pyroelectric sensor emits digital output pulse using the BISS0001 micro power PIR detector IC. Designed of a crystalline material which generates a surface electric charge when exposed to infrared heat, this charge changes in response to the amount of radiation on the crystal, and is measured using a sensitive J-FET (junction gate field-effect transistor) device built into the sensor. The digital output is 3V (high) in case of motion detection and idles otherwise. With an accuracy of 0.1 up to 6 meters, it requires an operating voltage of 3-10V, it is a low-cost sensor module with a working wavelength of 7um - 14um and an operating temperature placed between -20° C and 70° C.

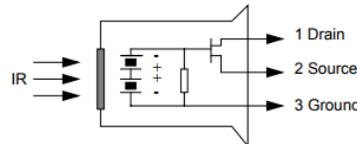


Fig. 5., PIR circuit configuration extracted from RE200B datasheet.

For fire detection, we use a flame sensor embedded with an LM393 voltage comparator chip and infrared sensing probe, which is sensitive to ordinary light. Low-priced with an exposure wavelength of 760 – 1100nm and an operating voltage of 3.3V – 5.3 V. The detection angle is ranged between 0° to 60° with the interface of 4 pins for digital output, analog output, power ground and positive power supply.

F. 1602A-1 LCD module

1602A-1 Liquid Crystal Display is usually based on an SPLC780D1 microcontroller, offers a display of 16x2 characters and 8-bit parallel interface. Inexpensive with a white backlight and a power supply of 5V and operating temperature of -10° C - 60° C. The supply voltage for logic (VDD) ranges from -0.3V – 7.0V, on the other hand, the supply voltage for LCD (V0), starts from 0 and the maximum is obtained by adding 0.3 to VDD. The input voltage minimum value is -0.3V, and its maximum can be obtained by adding 0.3 to the supply voltage for logic. See below the pin distribution of 1602A-1 LCD.

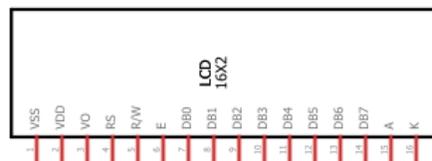


Fig. 6. 1602A-1 LCD pin distribution.

2. SOFTWARE DESIGN

The Software design of this smart security system is based on the Arduino development environment. Arduino Project is an open source code. The software programming uses modular programming ideas. Hence, the central functional module in the system is defined independently and later on blended together by the main program.

The overall flow diagram of the proposed model is shown in Figure 7, where it is assumed that the user already set up the credentials for the system activation and the mobile address for GSM communications.

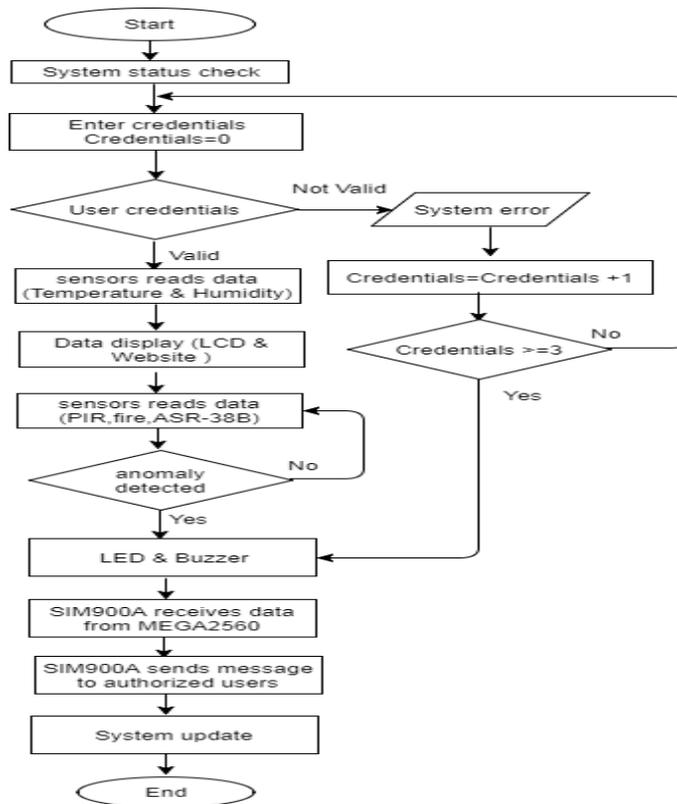


Fig.7. Flowchart of the Smart Home Security System

G. 1602A-1 LCD & Keypad configurations code

We start by defining the interconnections of the LCD and the Keypad. For the LCD connections we reserve the pins 7,8,9,10,11,12 and the code is as follows:

```

#include <LiquidCrystal.h>
lcd.begin(16,2);
Serial.begin(9600);
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
void setup ()
{

```

```
lcd.begin(16,2);  
lcd.setCursor(1,0);  
lcd.print(" SMART HOME");  
lcd.setCursor(1,1);  
lcd.print("SYSTEM");  
delay(3000);  
lcd.clear();  
lcd.print(F("STARTING GSM COM"));  
delay(50); }
```

To display any text during the code execution we just need to call the lcd.print("-----").

Next, we define the keypad communications:

```
#include <Password.h>  
#include <Keypad.h>  
Password password = Password( "1234" );  
const byte ROWS = 4;  
const byte COLS = 4;  
char keys[ROWS][COLS] = {  
  {'1','2','3','A'},  
  {'4','5','6','B'},  
  {'7','8','9','C'},  
  {'*','0','#','D'}};  
byte rowPins[ROWS] = {44,45,47,46};  
byte colPins[COLS] = {39,41,49,21};  
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );  
void setup()  
{ keypad.addEventListener(keypadEvent); }  
void loop(){ keypad.getKey();}
```

With the above code, the keypad can show the pressed digits by calling the lcd.print("----"). Nevertheless, we want to be able to receive the pressed number and store it in the micro-controller. Therefore, we create two additional functions, one to store the keypad events and the second to verify those events with a predefined password.

KeypadEvent Function

```
int x=0  
void keypadEvent(KeypadEvent eKey){  
  switch (keypad.getState()){  
    case PRESSED:  
      if (passwd_pos - 15 >= 5) { return ;}  
      Serial.print("Enter: ");  
      lcd.clear();  
      lcd.print("Enter:");  
      lcd.print(eKey);  
      Serial.println(eKey);  
      delay(10);  
      switch (eKey){  
        case 'A': checkPassword(); delay(1); break;
```

```
case 'C': checkPassword(); delay(1); break;
case 'D': checkPassword(); delay(1); break;
case 'B': password.reset(); delay(1); break;
case '*': checkPassword(); break;
case '#': password.reset(); break;
default: password.append(eKey); delay(1);}}
void checkPassword(){
if (password.evaluate()){
  Serial.println("Accepted");
  lcd.clear();
  lcd.print("Accepted");
  delay(500);
  lcd.clear();
  lcd.print(" SYSTEM ACTIVE");
  delay(80);
  digitalWrite(3, HIGH);
  delay(3000);
  digitalWrite(3, LOW);

//code to run if it works
  }
else
  {
    Serial.println("Denied");
    lcd.clear();
    lcd.print("Denied");
    password.reset();
    delay(10);
    x++;
if (x==3)
  {digitalWrite(35, HIGH);
  delay(1000);
  digitalWrite(35, LOW);
  lcd.clear();
  lcd.print("USER WARNED");
  delay(500);
  lcd.clear();
  digitalWrite(35, HIGH);
  delay(10000);
  digitalWrite(35,LOW);
  lcd.clear();
  lcd.print("Try Again!!!");
password.reset();}}}
```

As we previously separated the architecture of the Smart Home System into two main units, we can now separately describe the software design approach for each of the main parts, data acquisition and system control, respectively.

H. Software design for Data acquisition unit

Next, we describe the interconnection of the most relevant components in the data acquisition unit.

1. DHT11

The DHT11 Data pin (2) is connected to the pin A0(analog input 0). The code is as follows:

```
#include "DHT.h"
#define DHTPIN A0
#define DHTTYPE DHT11
DHT dht (DHTPIN, DHTTYPE);
void setup ()
{dht.begin();}
void dhtsensor () {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!"); return;}
  float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false);
  Serial.println("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.println("Temperature: ");
  Serial.print(t);Serial.print(" *C ");}
```

2. PIR

```
int calibrationTime = 30;
long unsigned int lowIn;
long unsigned int pause = 5000;
boolean lockLow = true;
boolean takeLowTime;
  int pirPin = 2; int ledPin = 4;
void setup() {sensor1();delay(200);
  pinMode(4, OUTPUT);
  pinMode(led, OUTPUT); }
void loop(){sensor();}
void sensor1() {
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(pirPin, LOW);
  lcd.clear();
  lcd.setCursor(0,0);}
```

```
lcd.print("SENSOR 1 ACTIVE");
delay(2000);
Serial.println("SENSOR 1 ACTIVE");}
void sensor() {
if (password.evaluate() {
if(digitalRead(pirPin1) == HIGH){
digitalWrite(ledPin1, HIGH);
digitalWrite(35,HIGH);
Serial.println("sensor2");
lcd.clear();
lcd.print("sensor 2");
delay(500);
(sms.SendSMS("MobileNumber","Break-in, Sensor 1 detected motion!!!"));
if(lockLow){ :
lockLow = false;
delay(50);}
takeLowTime = true; }
if(digitalRead(pirPin1) == LOW){
digitalWrite(ledPin1, LOW);
digitalWrite(35,LOW);
if(takeLowTime){ lowIn = millis();
takeLowTime = false; } } }
```

I. System Control unit software design

For the system control unit, we analyze the codes of the SIM900A, Wiznet5100 and the website set up.

1.SIM900A

```
#include "SIM900.h"
#include <SoftwareSerial.h>
#include "sms.h"
MSGSMS sms;
int numdata;
boolean started=false;
char smsbuffer[160];
char n[20];
void setup()
{ Serial.begin(9600);
if (gsm.begin(9600))
{ lcd.clear();
lcd.setCursor(1,1);
lcd.print(F("GOOD TO GO!!"));
delay(1000);}
else{ lcd.clear();
lcd.print(F("GSM NOT CONNECTED"));
delay(2000);
lcd.print(F("Try Again"));} }
```

The “sms.h” library offers quick and easy commands for GPRS communications, using a function such as sms.SendSMS("--","--") which allow users to send SMS. In the smart home design, this function can be used to warn the user when a password verification is made:

```
void checkPassword(){
  if (password.evaluate()){
    Serial.println("Accepted");
    lcd.clear();
    lcd.print("Accepted");
    delay(500);
    lcd.clear();
    lcd.print(" SYSTEM ACTIVE");
    delay(80);
    (sms.SendSMS("xxxxxx","Your Code was accepted!"));
  }
}
```

2. Wiznet5100

```
#include <SPI.h>
#include <Ethernet.h>
byte mac[ ] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[ ] = {x,x,x,x}; //ip address
byte gateway[ ] = { 10,0,1,20 };
byte subnet[ ] = { 255,255,255,0 };
EthernetServer server(80);
String readString;
void setup(){
  Ethernet.begin(mac,ip,gateway,subnet);
  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}
```

Using this code, we can establish an internet connection with the microcontroller. For online monitoring, we can have all the access using the IP defined in the code above; this address varies according to the local network.

We create a function called **net()** and place it under void loop() so that the website will be available at any time. Next, we request the **password.evaluate()** function; this will allow online users previously registered to have access to the online monitoring.

```
void loop(){
  net();
  net(){
    if (password.evaluate()){
      EthernetClient client = server.available();
      if (client) {
        while (client.connected()) {
          if (client.available()) {
            char c = client.read();
            if (readString.length() < 100) { readString += c;}
            if (c == '\n') {

```

```
Serial.println(readString);
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();
client.println("<HTML>");
client.println("<HEAD>");
client.println("<meta name='apple-mobile-web-app-capable' content='yes' />");
client.println("<meta name='apple-mobile-web-app-status-bar-style' content='black-
translucent' />");
client.println("<link rel='stylesheet' type='text/css'
href='http://randomnerdtutorials.com/ethernetcss.css' />");
client.println("<TITLE>Arduino Smart Home System</TITLE>");
client.println("</HEAD>");
client.println("<BODY>");
client.println("<H1>Smart Home</H1>");
client.println("<hr />");
client.println("<br />");
client.println("<H2>Arduino Home Security V1.0</H2>");
client.println("<br />");
client.println("<a href='http://admin:0000000AA@10.0.1.5/image.jpg'
target=_blank'><input type='button' value='Take Photo'></a>");
client.println("<a href='\"/?button1\"' target=_blank'><input type='button' value='Show
Temperature'></a>");
client.println("<a href='\"/?button1\"' target=_blank'><input type='button' value='Show
Humidity'></a>");
client.println("<br />");
client.println("<br />");
client.println("<a href='\"/?button2\"'>Rotate Left</a>");
client.println("<a href='\"/?button3\"'>Rotate Right</a><br />");
client.println("<p>Smart Home Design!</p>");
client.println("<br />");
client.println("</BODY>");
client.println("</HTML>");
delay(1);
client.stop();
}readString=""; } } } }
```

3. SIULATIONS AND RESULTS

The Test of the system needs to follow the proposed hardware and software design. Given that for an Intelligent Home System, a significant number of interconnections is required for binding the components together into the two referred units: data acquisition and system control, before the actual implementation the authors used the open-source project fritzing to test the elements of the two parts separately. Furthermore, the linking of the circuit of main modules was also simulated using the fritzing platform, which is a highly flexible and intuitive tool.

The system was tested on a broad range of conditions, covering outdoor and indoor environments (the range of the PIR, flame and DHT11 sensors) and demonstrated good accuracy according to

the description given in the hardware design. Different weather exposition was also used for the system reliability tests and poor bandwidth areas. We conducted 20 samples, and the system was consistent and highly responsive. The results of the response delay of the DHT11 sensor is shown in Figure 8 and 9. The response time needed during credentials authentication can be seen in figures 10 and 11, for wrong credentials and accepted credentials, respectively. Flame sensor response time is shown in figure 12. For visualization of the results, a prototype was built, and the images of the working system are shown in Figure 13 and 14. A GSM Network experiment was conducted, similar to the one described in [8]. In 20 samples the average response time for the flame sensor is of 9.4, contrasted with 9.16 obtained over 25 Samples. Both systems show a quick response time, although additional parameters should be considered to influence the system such as the Local Network efficiency.

The Average duration for the feedback in case of wrong credentials is of 5.5s, on the other hand, the response time for the accepted credentials is of 7s, this time can be further improved by placing the GSM response code before the sensors activated when the credentials are in place.

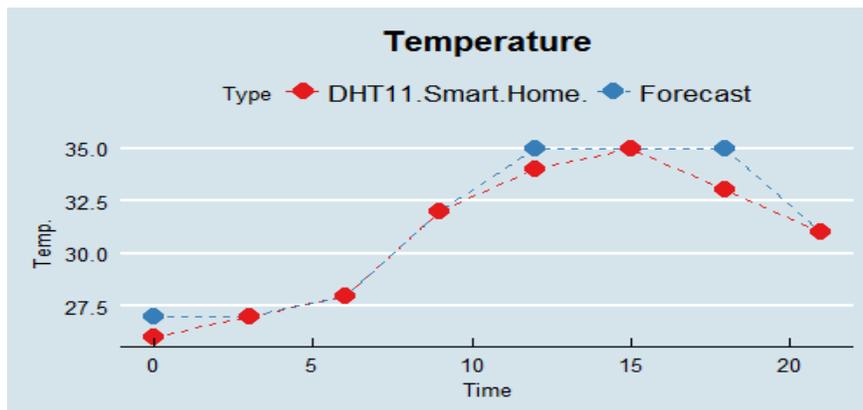


Fig. 8. Evaluation of temperature during different times of the day, the red dots represents the values registered by the DHT11 and the blue ones, the values forecasted in the same day.

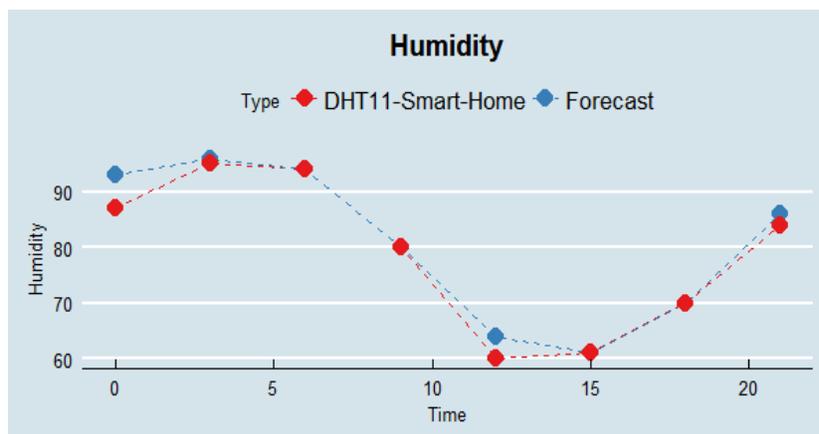


Fig.9. Analysis of the humidity during different times of the day. Red dots represent the readings of the DHT11 and the blue dots the values of the forecasted values.

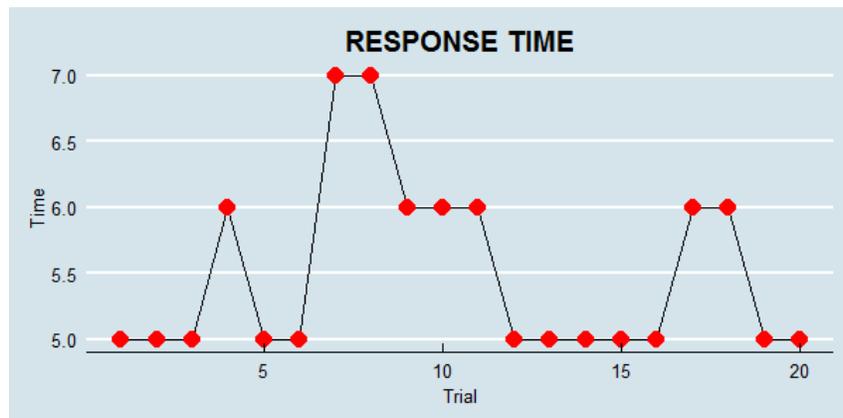


Fig. 10. Response delay for wrong credentials during user authentication, using China Mobile. The time is given in seconds.

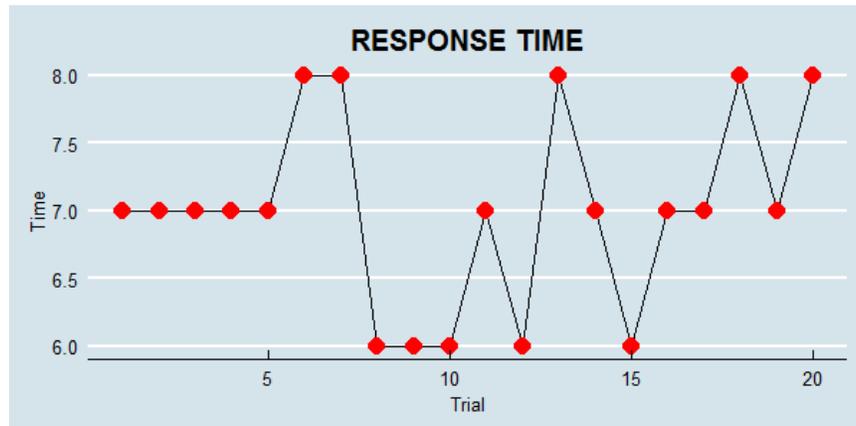


Fig. 11. Response delay for correct credentials during user authentication, using China Mobile Network.

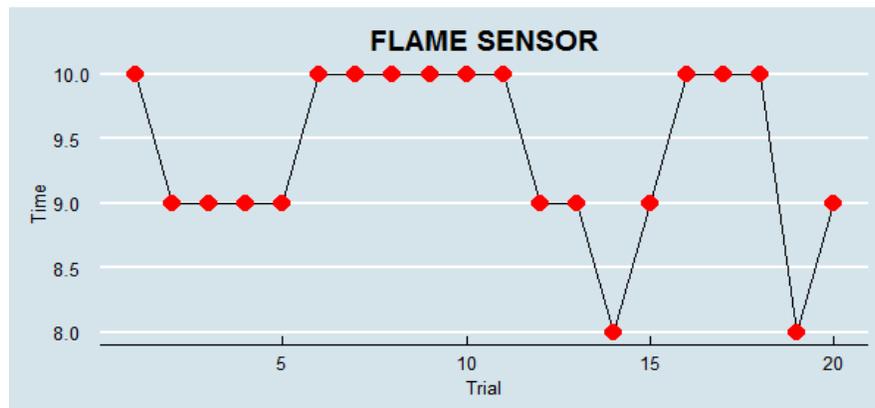


Fig. 12. Response delay for the flame sensor.



Fig. 13. Using android application for monitoring the system.

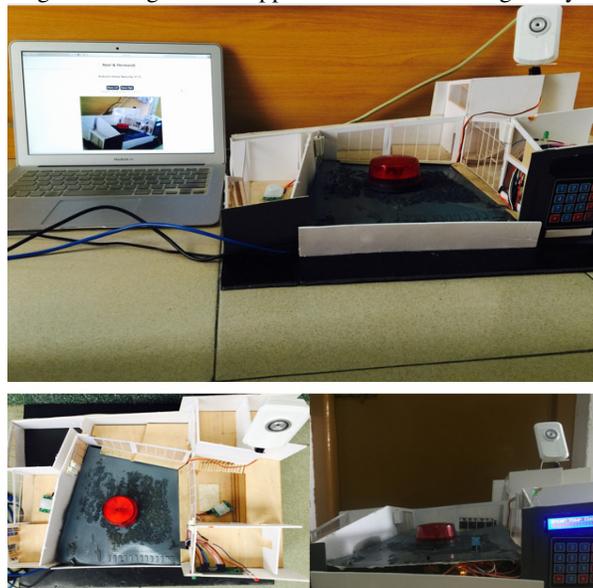


Fig. 14. A prototype of the system along with the website monitoring page.

4. CONCLUSION

We developed a smart home security system, using an ATmega2560 microcontroller. The performances of the system were tested in which the Temperature and Humidity sensors integrated were exposed for 24 hour period on 2nd June 2016. The system is controlled using the Internet communications. We designed an Android Application which can monitor the system; GSM is integrated for response in break-in situations.

We schematized a prototype, and its test results show excellent interconnectivity. The system offers low-cost, low power consumption, real-time surveillance, high accuracy, high flexibility with a good anti-interference performance and an average time of 9.4s for Network response in flame sensor detection. Modularity is used in response to the need for flexibility in modern systems design. In this case, the user can add more sensors without having to modify the whole system. This system can be implemented in many applications requiring low-cost and short distance working conditions.

REFERENCES

- [1] L. Atzori and A. Iera and G. Morabito. The Internet of Things: A survey. *Computer networks*, 54(15), 2010.
- [2] Miorandi D., Sicari S., Pellegrini F. De, Chlamtac I. Internet of things: Vision, applications, and research challenges *Ad Hoc Netw.*, 10 (7) (2012), pp. 1497-1516
- [3] F. Samie, L. Bauer, and J. Henkel, "IoT Technologies for Embedded Computing: A Survey," in *CODES+ISSS*, 2016.
- [4] L. Ophir, "802.11 Over Coax – A Hybrid Coax – Wireless Home Network Using 802.11 Technology", *Consumer Communications and Networking Conference*, Jan. 2004, pp. 13-18,
- [5] B. Yuksekkaya, A.A. Kayalar, M.B. Tonsun, M.K. Ozean, and A.Z. Alkar, "A GSM, Internet, and Speech Controlled Wireless Interactive Home Automation System," *IEEE Transactions on Consumer Electronics*, Vol.52, No.3, (2006), pp. 837-843.
- [6] M. Khan, B. N. Silva, K. Han, "Internet of Things based energy-aware smart home control system," *IEEE Access*, vol. 4, pp. 7556-7566, 2016.
- [7] Sivanathan A., Sherratt D., Gharakheili H., Sivaraman V., Vishwanath A., "Low-cost flow-based security solutions for smart-home IoT devices," <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7947781>
- [8] S. I. Azid and S. Kumar. (2011). Analysis and Performance of a Low-Cost SMS Based Home Security System. *International Journal of Smart Home*, 5(3), pp.15-24. [6] H. ElKamchouchi and A. ElShafee, "Design and prototype implementation of SMS based home automation system," in *Proc. 2012 IEEE International Conference on Electronics Design, Systems and Applications (ICEDSA)*, Kuala Lumpur, pp.162-167
- [9] Yanping Wang and Zongtao Chi, "System of Wireless Temperature and Humidity Monitoring Based on Arduino Uno platform," 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication, and Control.
- [10] Rajani US, Kadar A.A, "GSM Based Home Security System using PIR Sensor," *IJECT* Vol.8, Issue 2, April-June 2017.
- [11] Abid Rahim, Zeeshan Ali, Raushan Bharti, Syed and Mrs. Ramya B.K, "Design and Implementation of a Low-Cost Wireless Sensor Network using Arduino and nRF24L01(+)", *International Journal of Scientific Research Engineering & Technology (IJSRET)*, May 2016.
- [12] Sheikh Izzal Azid, Bibhya Sharma, *Intelligent Home: SMS Based Home Security System with Immediate Feedback*, *World Academy of Science, Engineering and Technology*, Vol. 72, 2012.
- [13] Mahendran.N, Geo Joe Mathai, Veenesh. M.U, *Multiple Sensor Feeding Supported Building Automation System Using Arduino Platform With Exposure of 802.15.4 Functionalities*, *International Journal of Engineering Trends and Technology*, Vol. 4 Issue2, 2013.
- [14] Nwankwo Prince, Nsionu Ifeanyi and E C Joseph, 2013 "Design and Implementation of Microcontroller Based Security Door System (Using Mobile Phone & Computer Set)" *Automation and Control Engineering*, Vol. 1, No. 1.
- [15] Hemant Chaudhary, Prateek Bansal and B.Valarmathi, 2017 "Advanced CAR Parking System using Arduino," *International Conference on Advanced Computing and Communication Systems (ICACCS-2017)*.
- [16] Vinnay Chadda: *Electronics for you "Arduino Based Vehicle Parking Counter"* (2013)
- [17] Alan G Smith "Introduction to Arduino." (2011)

- [18] N. S. Kumar, B.Vuayalakshmi, R.J Prarthana and A.Shankar, 2017 "IOT based smart garbage alert system using Arduino UNO," TENCON 2016-2016IEEE Region 10 Conference.
- [19] J. Dongjiu Geng, Yue Suo, Yu Chen, Jun Wen, Yongqing Lu, Remote Access and Control System Based on Android Mobil Phone, vol.2. Journal of Computer Applications, 2011, pp. 560-562
- [20] Ness., 2007. "Ness Quantum Dual Detector 100-204", PIR Datasheet. Ness Security Products, Ver.1.0
- [21] G. Gu and G. Peng, "The survey of GSM wireless communication system," in Proc. 2010 International Conference on Computer and Information Application (ICCIA), Tianjin, pp.121 – 124.
- [22] Naijia Ye. Principle and application of series of Atmega single-chip microcomputer. China Machine Press,2008
- [23] Tinsharp Industrial Co., Ltd. "LCM Module" TC1602A-01T datasheet. Sept.2009
- [24] Shenzhen EONE ELECTRONICS CO., LTD "Specification for LCD Module" 1602A-1 (V1.2)
- [25] Waveshare, Flame-Sensor-UserManual
- [26] Adafruit Industries, "Pyroelectric Passive Infrared Sensor,"PIR Sensor RE200B, 7.4.99
- [27] Murata Manufacturing Co., Ltd. "Pyroelectric Infrared Sensors for Reflow Soldering,"SMD-Dual Type Pyroelectric Infrared Sensor IRS-A200ST01 Series.0.6.24
- [28] Adafruit Industries, "BISS001" Micro Power PIR Motion Detector IC.
- [29] Adafruit Industries, "DHTxx Sensors," Updated[2016-12-26]
- [30] D-Robotics UK, "DHT11 Humidity & Temperature Sensor",7/30/2010
- [31] Aosong Electronics Co., Ltd.,"Temperature and humidity module DHT11 Product Manual."
- [32] J. M. Hughes, Arduino in a Nutshell: A Desktop Quick Reference, O'Reilly, 2016.
- [33] RobotShop," Arduino Mega 2560 Datasheet"
- [34] Adafruit Industries, "PIR Motion Sensor," Updated[2016-09-10]
- [35] Wiznet Co., Inc. "W5100 Datasheet" Version 1.1.6 Research Design Lab, "GPRS/GSM SIM900A MODEM" USER MANUAL, REV1.0
- [36] <http://www.mantech.co.za/datasheets/products/A000047.pdf>
- [37] P S. Bangare, A Pote, S L. Bangare, P Kurhekar and D Patil, 2013, "The Online Home Security System: Ways to Protect Home from Intruders & Thefts," International Journal of Innovative Technology and Exploring Engineering (IJITEE)ISSN: 2278- 3075, Volume-2, Issue-3.
- [38] IEEE Code of Ethics http://www.ieeeusa.org/documents/CAREER/CAREER_LIBRARY/ethics.html
- [39] Connecting a Photocell
<https://learn.adafruit.com/photocells/wiring-a-photocell>
- [40] Web view <https://developer.android.com/reference/android/webkit/WebView.html>
- [41] HTML <iframe> http://www.w3schools.com/tags/tag_iframe.asp
- [42] <https://www.arduino.cc/en/Main/arduinoBoardMega>
- [43] <https://www.allaboutcircuits.com/technical-articles/using-interrupts-on-arduino/>
- [44] <http://fritzing.org/learning/tutorials>
- [45] <http://www.weather.com.cn/>
- [46] AndroidLoginScreen http://www.tutorialspoint.com/android/android_login_screen.htm
- [47] Seedstudio PIR sensor website http://www.seedstudio.com/wiki/File:Twig_BISS0001.pdf
- [48] Arduino Ethernet Shield Version 5.0 <http://shieldlist.org/arduino/ethernet-v5>
- [49] Simcom GSM modem website, <http://www.simcom.eu/index.php?m=letoltes>
- [50] <http://www.mantech.co.za/datasheets/products/A000047.pdf>
- [51] <https://www.arduino.cc/en/Guide/HomePage>
- [52] RS Components, "Arduino Mega 2560"
- [53] <http://www.nl-electronics.com/?q=en/node/1284>
- [54] <https://www.arduino.cc/en/Reference/AttachInterrupt>
- [55] Atmel, "Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V", 2549Q-AVR-02/2014
- [56] <https://pastebin.com/bAuQGWay>
- [57] <https://playground.arduino.cc/Main/DHT11Lib>
- [58] <https://www.adafruit.com/products/386>
- [59] https://wiki.eprolabs.com/index.php?title=SIM_900A_GSM_GPRS_Module
- [60] <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>
- [61] <http://www.glolab.com/pirparts/infrared.ht>

AUTHORS

Abel A. Zandamela was born on 2nd May 1993 in Maputo, Mozambique. He received the B.E Degree in Information Engineering from the South China University of Technology. He is now pursuing his M.Eng Degree in Communication Engineering from King Mongkut's University of Technology North Bangkok.

