International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 16-19
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

CUSTOM MADE EMBEDDED AUTOMATION SYSTEMS FOR SMART HOMES - PART 2: THE IMPLEMENTATION

M. Papoutsidakis
Dept. of Automation Engineering, Piraeus University A.S., Athens, Greece

Rajneesh Tanwar
Dept. of Information Technology, Amity University, Noida, India

A. Chatzopoulos
Dept. of Automation Engineering, Piraeus University A.S., Athens, Greece

D. Tseles
Dept. of Automation Engineering, Piraeus University A.S., Athens, Greece

***Abstract*— The subject for the current paper was the completion and implementation of creation of an remote controlled environment based on a microcontroller. A template of a house was built and upon it the user can control various functions and tasks of a house like the temperature the lights and many more all that remotely.to achieve that a GSM module was used in combination with the microcontroller giving the ability to the user by SMS's to have full control of his house. The main element used was an ARDUINO microcontroller an ADAFRUIT GPS module and various sensors and electronic elements. Following the already done preliminary work, the steps we took for the development of the project are summarized as, the creation of a house template where we adjusted the microcontroller, the GPS module, the sensors and all the various electronics components and the wiring together with the microcontroller, the GSM, the sensors and the electronic components. The final step was the programming of the microcontroller, which was done using C language.**

Keywords— **Smart home, Sensor Network, Embedded Solutions, Microcontrollers Applications**

I. INTRODUCTION

In recent years the development of telecommunications and telematics world follows exponential rate giving man the ability to simultaneously save resources and time without compromising on service quality.

An example of this concept is the human need to check the status of particular applications and then to manage, but without necessary to be physically present.

The solution to this need given the progress made over the years in the field of remote control and hence to telemetry and development in the field of microelectronics and more specifically in the field of microcontrollers. Although this project has a wide range of applications, it was selected to simulate a " smart home " for which the demand is constantly increasing.

In this work is given the opportunity to the reader to first understand the meaning of telemetry as described by the theoretical perspective and then gradually move on to detailed descriptions of both the implementation of our application and to use and upgrade. In an attempt to present an overall idea of the technologies that were included in a this project, some typical examples are provided below.

Telemetry is an automated communications process by which measurements and other data are collected at remote or inaccessible points and transmitted to receiving equipment for monitoring. The word is derived from Greek roots: *tele* = remote, and *metron* = measure. Systems that need external instructions and data to operate require the counterpart of telemetry, telecommand. Although the term commonly refers to wireless data transfer mechanisms (e.g., using radio, ultrasonic, or infrared systems), it also encompasses data transferred over other media such as a telephone or computer network, optical link or other wired communications like power line carriers. Many modern telemetry systems take advantage of the low cost and ubiquity of GSM networks by using SMS to receive and transmit telemetry data. A telemeter is a device used to remotely measure any quantity. It consists of a sensor, a transmission path, and a display, recording, or control device. Telemeters are the physical devices used in telemetry. Electronic devices are widely used in telemetry and can be wireless or hard-wired, analog or digital. Other technologies are also possible, such as mechanical, hydraulic and optical. Telemetry may be commutated to allow the transmission of multiple data streams in a fixed frame.

A programming language is a formal computer language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs to control the behavior of a machine or to express algorithms. The earliest known programmable machine preceded the invention of the digital computer and is

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 14-19
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

the automatic flute player described in the 9th century by the brothers Musa in Baghdad, "during the Islamic Golden Age". As found in [1], from the early 1800s, "programs" were used to direct the behavior of machines such as Jacquard looms and player pianos [2]. Thousands of different programming languages have been created, mainly in the computer field, and many more still are being created every year. Many programming languages require computation to be specified in an imperative form (i.e., as a sequence of operations to perform), while other languages use other forms of program specification such as the declarative form (i.e. the desired result is specified, not how to achieve it). The description of a programming language is usually split into the two components of syntax (form) and semantics (meaning). Some languages are defined by a specification document (for example, the C programming language is specified by an ISO Standard), while other languages (such as Perl) have a dominant implementation that is treated as a reference. Some languages have both, with the basic language defined by a standard and extensions taken from the dominant implementation being common.

A microprocessor is a computer processor which incorporates the functions of a computer's central processing unit (CPU) on a single integrated circuit (IC), like [3] or at most a few integrated circuits like [4]. The microprocessor is a multipurpose, clock driven, register based, programmable electronic device which accepts digital or binary data as input, processes it according to instructions stored in its memory, and provides results as output. Microprocessors contain both combinational logic and sequential digital logic. Microprocessors operate on numbers and symbols represented in the binary numeral system. The integration of a whole CPU onto a single chip or on a few chips greatly reduced the cost of processing power. Integrated circuit processors are produced in large numbers by highly automated processes resulting in a low per unit cost. Single-chip processors increase reliability as there are many fewer electrical connections to fail. As microprocessor designs get faster, the cost of manufacturing a chip (with smaller components built on a semiconductor chip the same size) generally stays the same. Before microprocessors, small computers had been built using racks of circuit boards with many medium- and small-scale integrated circuits. Microprocessors combined this into one or a few large-scale ICs. Continued increases in microprocessor capacity have since rendered other forms of computers almost completely obsolete (see history of computing hardware), with one or more microprocessors used in everything from the smallest embedded systems and handheld devices to the largest mainframes and supercomputers.

A microcontroller (or MCU for microcontroller unit) is a small computer on a single integrated circuit. In modern terminology, it is a System on a chip or SoC. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory in the form of Ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems.

Figure 1: A typical microcontroller

By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz, for low power consumption (single-digit milliwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

GSM (Global System for Mobile Communications, originally Groupe SpécialMobile), is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe the protocols for second-generation (2G) digital cellular networks used by mobile phones, first deployed in Finland in July 1991. As of 2014 it has become the de facto global standard for mobile communications – with over 90% market share, operating in over 219 countries and territories. 2G networks developed as a replacement for first generation

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 14-19
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

(1G) analog cellular networks, and the GSM standard originally described as a digital, circuit-switched network optimized for full duplex voice telephony. This expanded over time to include data communications, first by circuit-switched transport, then by packet data transport via GPRS (General Packet Radio Services) and EDGE (Enhanced Data rates for GSM Evolution or EGPRS). Subsequently, the 3GPP developed third-generation (3G) UMTS standards followed by fourth-generation (4G) LTE Advanced standards, which do not form part of the ETSI GSM standard.

In the broadest definition, a sensor is an object whose purpose is to detect events or changes in its environment and sends the information to the computer which then tells the actuator (output devices) to provide the corresponding output. A sensor is a device that converts real world data (Analog) into data that a computer can understand using ADC (Analog to Digital converter). Sensors are used in everyday objects such as touch-sensitive elevator buttons (tactile sensor) and lamps which dim or brighten by touching the base, besides innumerable applications of which most people are never aware. With advances in micromachinery and easy-to-use micro controller platforms, the uses of sensors have expanded beyond the most traditional fields of temperature, pressure or flow measurement, as in [5], for example into MARG sensors. Moreover, analog sensors such as potentiometers and force-sensing resistors are still widely used. Applications include manufacturing and machinery, airplanes and aerospace, cars, medicine, robotics and many other aspects of our day-to-day life. A sensor's sensitivity indicates how much the sensor's output changes when the input quantity being measured changes. For instance, if the mercury in a thermometer moves 1 cm when the temperature changes by 1 °C, the sensitivity is 1cm/°C (it is basically the slope Dy/Dx assuming a linear characteristic). Some sensors can also affect what they measure; for instance, a room temperature thermometer inserted into a hot cup of liquid cools the liquid while the liquid heats the thermometer. Sensors need to be designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages.[citation needed] Technological progress allows more and more sensors to be manufactured on a microscopic scale as microsensors using MEMS technology. In most cases, a microsensor reaches a significantly higher speed and sensitivity compared with macroscopic approaches.

II. SYSTEM METHODOLOGY

The system consists of custom designed software that is hosted and supported by the appropriate software. A brief summary of the equipment is provided below.

A. **Microcontroller**

For the construction of the model we initially studied various housing standards and having chosen the most suitable for this study began the design of the model. The low cost though powerful enough microcontroller Arduino UNO was chosen for the task, which is shown in the following Figure.

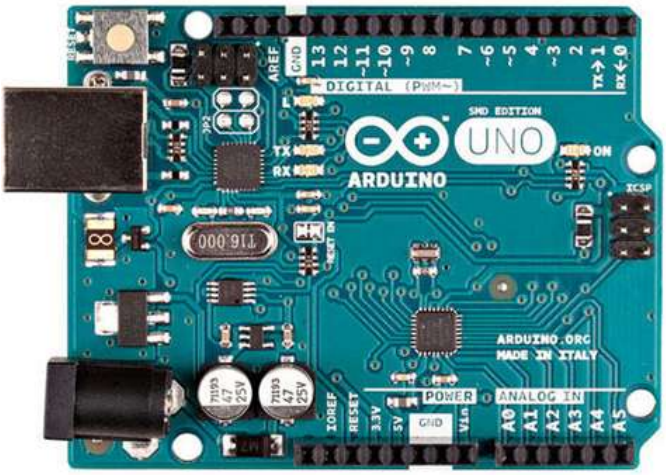

Figure 2: The microcontroller board

The microcontroller architecture design was based on the common logical diagram that follows:

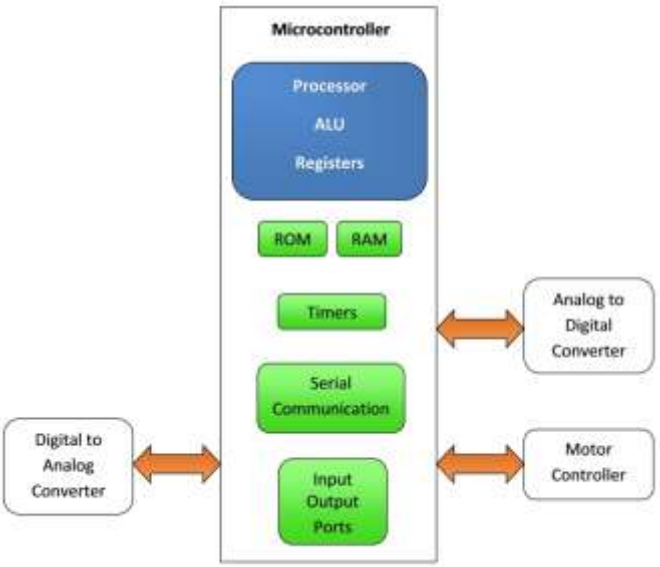

Figure 3: The microcontroller architecture

The model was constructed with cardboard maquette Neofoam 10mm, which created pieces that are joined together so they formulated the model as a home environment. After the construction of the model, we examined the method and the position of how each component will be placed. Initially the

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 14-19
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

microcontroller Arduino was placed then the GSM module and after that all the other electronic components positioned and placed. To position the motion sensor we selected an area that represents the living room,where the entrance door is. Also the temperature sensor was placed there as this happens in most homes. In the lounge there is also a fan, which is used to simulate a cooling / heating system, which is placed next to the temperature sensor. All rooms are fitted with light emitting diodes (LED) for lighting simulation and an LED in the bathroom for the boiler indicator. Finally in the garage we adapted the servomotor to open the garage door, and in the same area there is the board of the microcontroller, the GSM module and the auxiliary board. Finishing the installation of the components, we performed the necessary wiring components together.

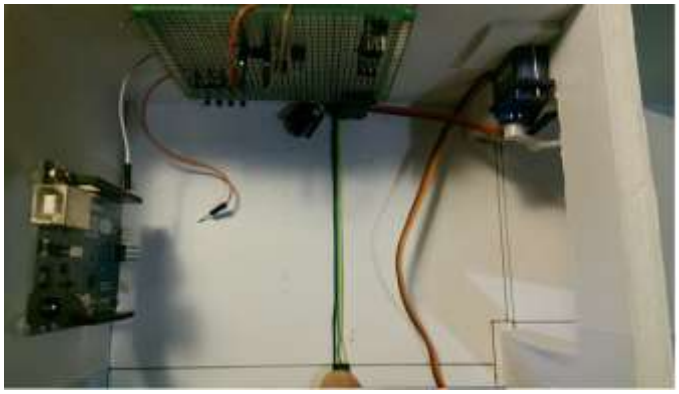

Figure 4: The microcontroller's placement

B. Wiring and Networking

All necessary wiring and inter-connections between the microcontroller and its peripherals can be found in Figure 5. At this point it must be stated that all connections were tested for many operating hours, multiple experiments took place to prove the stability of the project. In Figure 6, one can find all the wiring in details, the sensor's placement and the ports of the microcontroller that operate them as well as the power management and handling of the system.

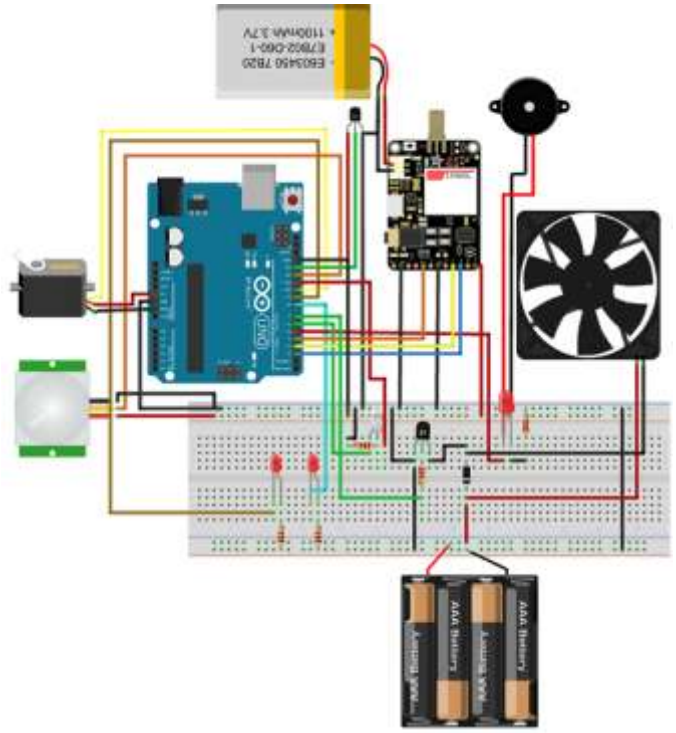

Figure 5: The microcontroller's peripherals

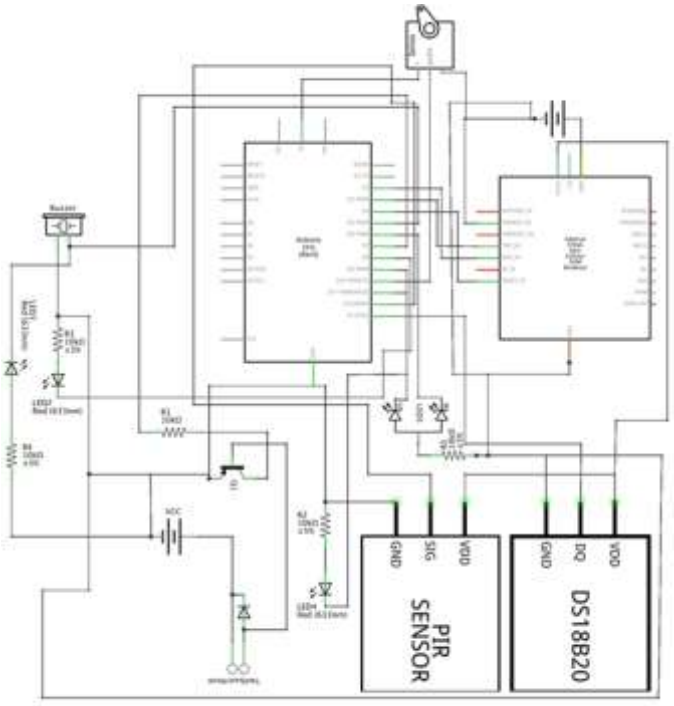

Figure 6: The system's electronic circuitry

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 14-19
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)

C. **C Code**

Last step for completion of the work was to develop the code. This is an automated system responsible for control and protection of a house environment. All manipulations are done via mobile phone and in fact by sending-receiving text messages via the GSM network. It is compatible with all mobile operators and the connection is simple with the use of a SIM card. This network was chosen because it is more stable and reliable than an Internet connection, which may at any time be interrupted. It can be checked from anywhere, with condition to have ability to send a text message. In the first part we declared the necessary libraries for the operation of the code and then we declare the variables used. In the second part, which is the main part of the program, the code is developed using variables and functions that are available and depending on the inputs it receives it makes the necessary actions.

```
1.  #include <Servo.h>
2.  #include <OneWire.h>
3.  #include <DallasTemperature.h>
4.  #include<stdio.h>
5.  #include<stdlib.h>
6.  #include <SoftwareSerial.h>
7.  #include "Adafruit_FONA.h"
8.  #define FONA_RX 2
9.  #define FONA_TX 3
10. #define FONA_RST 4
11. #define ONE_WIRE_BUS 13
12. Servo myservo;
13. OneWire oneWire(ONE_WIRE_BUS);
14. DallasTemperature sensors(&oneWire);
15. DeviceAddress insideThermometer;
16. char replybuffer[255];
17. int buzzer=5;              // output pin for buzzer
18. int greenledflush=6;      // green indicator led
19. int ac=7;                 //  output pin for A/C
20. int lights=8;             //  output pin for led lights
21. int boiler=9;             // output pin for boiler
22. int redledflush=11;       // red indicator led
23. int sensor=12;            // input pin for P|R signal
24. int alarm=0;              //  Variable to anable/desable the alarm system
25. int RTemp;                //  Real temperature
26. int DTemp;                //  Desired temperature
27. int tempflag1=0;          //  Variable to control the temperature system
28. int tempMode=0;    // Cooling/Heating mode
29. int minSecsBetweenTextsAlarm = 15; // 1 min
30. long lastSend =-minSecsBetweenTextsAlarm* 1000l;
31. int minSecsBetweenTemp = 15;
32. long lastSent = -minSecsBetweenTemp * 1000l;
33. SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
34. Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
35. uint8_t readline(char *buff, uint8_t maxbuff, uint16_t timeout = 0)
36. void setup(void) {
37. while (!Serial);
38. Serial.begin(115200);
39. Serial.println
40. Serial.println(F("FONA basic test"));
41. Serial.println(F);
42. fonaSS.begin(4800);
43. if (! fona.begin(fonaSS)) {
44. Serial.println(F("Couldn't find FONA"));
45. while (1);
46. }
47. Serial.println(F("FONA is OK"));
48. pinMode(ac,OUTPUT);
49. pinMode(sensor,INPUT);
50. pinMode(lights, OUTPUT);
51. pinMode(boiler, OUTPUT);
52. pinMode(greenledflush, OUTPUT);
53. pinMode(redledflush, OUTPUT);
54. digitalWrite(greenledflush,HIGH);
55. digitalWrite(redledflush,HIGH);
```

International Journal of Engineering Applied Sciences and Technology, 2017
Vol. 2, Issue 5, ISSN No. 2455-2143, Pages 14-19
Published Online March-April 2017 in IJEAST (http://www.ijeast.com)


```
56. pinMode(buzzer, OUTPUT);
57. sensors.begin();
58. Serial.print(F("Locating devices..."));
59. Serial.print(F("Found "));
60. Serial.print(sensors.getDeviceCount(), DEC);
61. Serial.println(F(" devices."));
62. if (!sensors.getAddress(insideThermometer, 0))
63. Serial.println(F("Unable to find address Device 0"));
64. sensors.setResolution(insideThermometer, 9);
65. }
66. int printTemperature(DeviceAddress deviceAddress)
67. {
68. float tempC = sensors.getTempC(deviceAddress);
69. Serial.print(F("Temp C: "));
70. Serial.println(tempC);
71. return tempC;
72. }
73. void loop()
74. {
75. int sms=fona.getNumSMS();
76. if (sms>0 )  // Checks if there is an SMS available
77. {
78. int8_t smsnum = fona.getNumSMS();
79. uint16_t smslen;
80. fona.readSMS(smsnum, replybuffer, 250, &smslen);
81. String smstext = String(replybuffer);
82. Serial.println(smstext);
//prints the sms text in the serial
```

III. CONCLUSION

This paper represents the second part of a research work that started earlier by the research team and actually concludes successfully the first attempt to research the potentials to automate many home facilities. Given the fact that a model of a house was actually implemented and thoroughly tested, this work can be considered as an overall premise for anybody that can implement low cost but reliable microcontroller applications to automate home facilities. was successful given the fact that all equipment was custom made with on the shelf electronics. There is another research work following this that is on the way to be presented that includes more sensors and moreover the necessary C code programming to make the whole system operative.

IV. ACKNOWLEDGEMENTS

All authors would like to express their gratitude to the Piraeus University of Applied Sciences for providing the required data and funding in order to undertake and complete this research project as part of "Automation of Production and Services" Postgraduate Program of Studies.

V. REFERENCE

[1] T. Denning, T. Kohno, and H. M. Levy, "Computer security and the modern home," Commun. ACM, vol. 56, no. 1, pp. 94–103, Jan. 2013.

[2] B. Ur, J. Jung, and S. Schechter, "The current state of access control for smart devices in homes," in Workshop on Home Usable Privacy and Security (HUPS) . HUPS 2014, July 2013.

[3] B. Ur, E. McManus, M. Pak Yong Ho, and M. L. Littman, "Practical trigger-action programming in the smart home," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 803–812.

[4] F. Roesner, T. Kohno, A. Moshchuk, B. Parno, H. J. Wang, and C. Cowan, "User-driven access control: Rethinking permission granting
in modern operating systems," in Proceedings of the 2012 IEEE Symposium on Security and Privacy , ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 224–238.

[5] Allseen Alliance, "AllJoyn Data Exchange," https://allseenalliance.org/ framework/documentation/learn/core/system-description/data-exchange, Accessed: Nov 2015.