https://dx.doi.org/10.22161/ijaers.611.23 *ISSN: 2349-6495(P) | 2456-1908(O)*

Home Automation Using Arduino Platform on an Embedded Server

Bruno Henrique Chagas de Albuquerque[1], Fabiana Rocha Pinto[2], Leonardo de Oliveira Chagas[3]

[1,3]Academic department, University Center FAMETRO, Manaus -AM, Brazil
2Academic department, University FUCAPI, Manaus -AM, Brazil

Abstract*— *Home automation has increased its expansion worldwide, given the available technologies, this study aims to develop a wireless home automation system using the Arduino prototyping platform and the HTML language, where with this technology is elaborated a software for user interface and the wifi technology that the platform makes possible. Creating a platform where the user can monitor their home from anywhere, if there is internet, having more security and control of their home.

Keywords*— *Home Automation, Interface, Monitor, Platform and Security.

I. INTRODUCTION

Since the 1980s, home automation has grown in strength and expression throughout the world, with the advancement of the internet and personal computer, a new culture of access to information has been created, thus enabling the emergence of home automation projects and small and medium sized building [7].

Home automation brings together a set of technologies that have as their main objective the automation of a home and which, besides performing some routine tasks, provide safety, reduce energy consumption, provide comfort and integrate with the user the system functionalities [11].

Given that most people spend most of their day outside their homes, the aim is for the interface to be always available for user access, not just at home, but anywhere that the user need to check and control the state of any device in his home.

The proposal was to develop a low-cost independent system, simple interface to configure, monitor and power devices. The user must connect to the internet by mobile or notebook, communicating from the browser to the platform, which will be connected to the devices, thus being able to control them from anywhere.

II. HEADINGS

To develop a web-based home automation system on an embedded server, we used concepts of home automation, web structure and embedded web server.

2.1 Home Automation

The automation, goes far beyond the simple insertion of electric motors and controllers, refers to the control of lamps, televisions, sockets, remote control doors that have electronic remote control [12].

Home automation systems can be classified as autonomous, integrated and intelligent. Where in an approach seeks to describe autonomous system, since it performs a control on and off of some devices, with a predefined configuration [5].

In this sense, the purpose of this paper is to offer a practical, fast and safe way to control the electrical and electronic components of the residence. The user can still set their preferences for lighting drives and other devices, so the house will always provide a pleasant environment for residents [16].

2.2 Protocol ISO and TCP

With the need to share data, the world wide web (WWW) was created, which addresses specific issues and has a text-only interface information exchange tool, which over time has been improved and made more accessible to millions of people. Thus, arises the need for communication between computational devices and international standardization between protocols, creating the International Standards Organization / Open Systems Interconnection (ISO / OSI) reference model and the Transmission Control Protocol / Internet Protocol (TCP / IP) model. These protocols divide the complex process of communicating into small layers, making tasks less complicated and sub tasks more efficient [10].

2.3 System Operation

Requests from a web system page, a web server is required, a process that constantly waits for the client request. When a customer request arrives, the system

https://dx.doi.org/10.22161/ijaers.611.23 *ISSN: 2349-6495(P) | 2456-1908(O)*

interprets, if valid, executes and sends the response to the customer.

III. MATERIALS AND METHODS

A survey was carried out on hardware platforms in the market with support through the embedded internet to compose the system. The platform that stood out was the Arduino Ethernet Shield Fig.3.1 coupled to an Arduino Mega 2560 Fig. 3.2 meets the needs of the project by having ethernet controller, TCP / IP libraries and low cost, Arduino has 54 input and output pins and 16 pin analog input, can connect multiple devices simultaneously.

Fig. 3.1: Arduino Ethernet Shield

Fig. 3.2: Arduino Mega 2560

It also has a MicroSD slot, which is required for storing files such as web pages, images, settings and scripts. Through the use case diagram Fig 3.3, it is possible to visualize the operation of the process that contemplates the modeling.

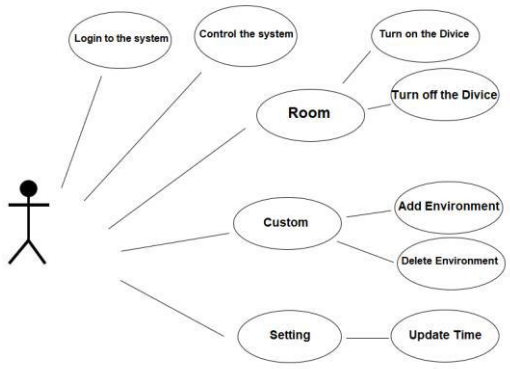

Fig. 3.3: Case Diagram

3.1 Web Server on Board

Arduino is aimed at artists and amateurs alike, or it can enable you to create flexible, easy yet cost-effective designs. Arduino programming uses a programming language similar to C / C ++, making it easy to learn those who already have some knowledge in these languages. In addition, it covers over ninety percent of the sensors on the market [14].

The platform chosen for the web deployment was the Arduino Ethernet Shield coupled to an Arduino Mega 2560. Arduino is an electronic prototyping platform that has a microcontroller and is built on a library that aids in writing or configuring C / C ++ programming. The great differential of this tool is its development and improvement by a community that promotes projects and code, with the intuit that it is an open – source, that is, anyone who has programming knowledge can modify it and share it with others so that it becomes better, increasingly expanding the possibilities it can do [9].

Arduino Mega 2560 is an ATmega2560-based microcontroller board that has 256 KB of flash memory for code storage (of which 8KB is used by the bootloader), 8 KB of SRAM (Static Random Access Memory) and 4 KB of EEPROM (Erasable Programmable Read-Only Memory) [8]. Arduino Ethernet Shield allows an Arduino to be connected to the internet using the Wiznet W5100 ethernet controller, which communicates with both the W5100 and the microSD card via the SPI [3].

It is necessary to store device and environment configuration data in a structure with low processing cost, which can be an embedded system. Arduino has no operating system and there is no possibility of installing a conventional database, the data was structured in XML (Extended Markup Language) and the storage in the micros that Ethernet Shield has.

Using the library provides us with a broader and more diverse programming, for Arduino implementation we used the following Libraries: Webduino, SdFat and

https://dx.doi.org/10.22161/ijaers.611.23 *ISSN: 2349-6495(P) | 2456-1908(O)*

LiquidCrystal. Webduino allows the Arduino card to connect to the internet, and provides some features like: handling of parameters passed by URL; implementation of handles for different resources; Handle the HTTP Methods: GET, HEAD, POST, PUT, DELETE, PATCH, HTTP Basic Authentication, Web encoding shapes and images; among others [10].

SdFat is a library written in C ++ that supports the creation, deletion, reading, writing and truncation of files. The LiquidCrystal library allows the Arduino board to control a liquid crystal display (LCD), showing the system time and signaling the server and the micros to start correctly or not [13].

3.2 Embedded Server Web Page

Over the years, developed Web systems have become increasingly complex. This design pattern has the great advantage of dividing a large project into smaller and less complex parts. Changes to one layer will not affect the others making it clearly simpler to perform maintenance such as layout changes, adding new features [15].

Web pages on the World Wide Web are usually implemented in HTML (Hypertext Markup Language) format. MyHOME has an HTML page, myhome.html, where it added all system screens because it reduces data traffic through Arduino, saving processing.

One of the technologies used in MyHOME was Ajax (Asynchronous JavaScript + XMLs), which merges Web technology, so the user's workflow is not interrupted by the need to communicate with the server as it allows some page features to remain available while the browser collects the new data and then sends it out at once, as Arduino is a low-processing tool, sending modifications one by one at a very high processing cost, delaying the activity.

To facilitate interface development and make it more agile, we opted to use the jQuery JavaScript library, which is supported by virtually all browsers. For user interface (UI) features, the visual part used the jQuery-based library, the jQuery UI [8]. It featured numerous graphical components for web development such as Windows, buttons, animations, advanced and high-level effects, etc. Fig 3.4 shows one of the system windows.

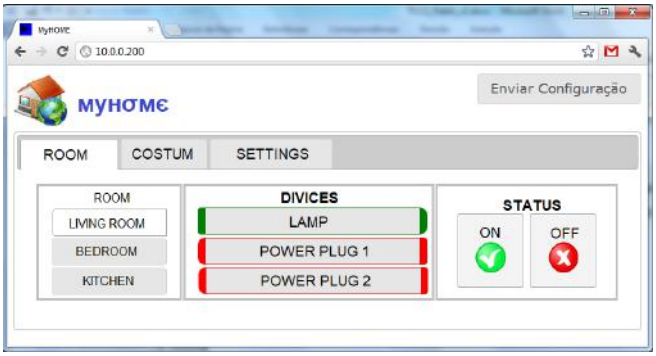

Fig. 3.4: System Room Window

In this window you can control the devices that the rooms have.

3.3 System Operation

The System is basically divided into three functionality topics: devices, environments and scenario.

In the System, a device is an object that needs electric current for its operation, and its power can be connected to a relay, for example: lamps, televisions, computer, router, fan, etc., system acting with the functionality of turning on and hang up.

An environment is a set of devices grouped together to better organize and improve user interaction with the system, so the user can create an environment called a room by grouping all equipment connected to the system.

The scenario is a set of actions with devices, which can be programmed to activate or deactivate devices at certain predefined times, so that some devices from the same or different environments can be set so that they can be activated or deactivated according to time. scheduled, the platform will perform at the specified time.

Thus, implementing a flow to access the System with different execution, by its memory and restricted server processing. After some testing it was identified that the time from reading files in microSD to browser could be improved by Google Code's Project Hosting, which offers a free collaborative development environment for open source projects. It will basically host some static files on the internet (unchanged files), such as the HTML page, images, scripts, and libraries, leaving out only the System database “XMLs”.

IV. SYSTEM TESTS

After the implementation of the new hosting functionality was completed, the system validation tests were performed, as shown in Fig 4.1.

https://dx.doi.org/10.22161/ijaers.611.23 *ISSN: 2349-6495(P) | 2456-1908(O)*

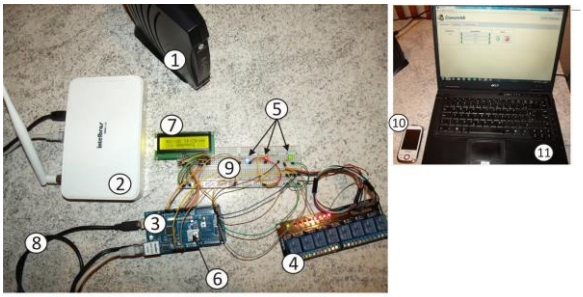

Fig.4.1: Test to validate the system

Fig 4.1 shows the test environment, where a Wireless Router (2) was used, with a Wide Area Network (WAN) port connected to a modem (1) that provides internet; the platform connected to the Router's Local Area Network (LAN) port; a SanDisk 2GB microSD inserted into the Ethernet Shield (6); a USB cable used for firmware transmission and System debugging (8); 8-channel YwRobot relay interface module (AC250V-10A; DC30V-10A), with relay output status indication LEDs (4); three distinct color Light Emitting Diode (LED) (5); 16x2 LCD display with Green Backlight (7); a protoboard with 830 holes (9).

For the browser device that accesses the MyHOME interface, a notebook (11) and a cellphone (10) connected to the router via wireless were used. As it is a Web System, all functionality is based on HTTP requests, and it is necessary to analyze these requests. For this function, the browser tool itself was used. In Google Chrome there is the tool called Developer Tool, it has a set of features to analyze a web page, such as debugging scripts, request monitoring, etc.

Another tool used was Arduino itself, through the Arduino IDE Serial Monitor, which monitors serial communication between Arduino and the computer (COM4 port).

For the functionality tests, such as the activation of the devices, three LEDs with different colors were used as seen in Fig. 4.1 item 5, to distinguish when changing the ports (Arduino output) and to see if it changed its state. After stabilizing the system, the relay drive module was operated with the system. The eight drive ports were connected to the Arduino ports for testing, and each port responded to its particular command, with great response time, both through the notebook connected to the same WI FI network, while on the mobile phone away from home, speed tests also yielded excellent results, sending several commands such as activating and deactivating the same device sequentially to see Arduino's response time, quickly turning on and off different devices. There were few cases where the devices took more than 1 second to switch states, but this was due to the mobile phone's internet having a low network signal, affecting the test results.

V. CONCLUSION

Given the current scenario of home automation, the solution to the problem proposed in this study was to develop a flexible and cost-effective home automation system named “MyHOME”. It allows you to operate on your home devices with control, monitoring and personalization features through an easy and intuitive interface that can be accessed by a browser from anywhere with internet. Featuring extra functionality such as device grouping in environments and scenario creation with programmable on / off times, giving the user the power to monitor and control their home from anywhere.

REFERENCES

[1] Almeida, R., Moraes, C. and Seraphim, T. (2016). *Programação de Sistemas Embarcados*. 1st ed. Rio de Janeiro: Elsevier, pp.13 - 15..

[2] **ARDUINO**. Arduino Ethernet Shield. 2011. Disponível em:<<http://arduino.cc/en/Main/ArduinoEthernetShield>>. Acesso em: 20/09/2019.

[3] **ARDUINO**. Arduino Mega 2560. 2011a. Disponível em:<<http://arduino.cc/en/Main/ArduinoBoardMega2560>>. Acesso em: 20/09/2019.

[4] **ARDUINO**. Arduino Software. 2011. Disponível em: <<http://arduino.cc/en/Main/Software>>. Acesso em: 20/09/2019.

[5] Associação Brasileira de Automação Residencial. Os principais Sistemas de Automação Residencial, 2019. Disponível em << http://www.aureside.org.br/quem-somos>>. Acessado em 18/09/2019.

[6] BATRINU, C. **ESP8266 Home Automation Projects**. 1. ed. São Paulo: Novatec, v. 1, 2018.

[7] BOLZANI, Caio A. M. Residências Inteligentes: um curso de Domótica. São Paulo: Livraria da Física, 2004.

[8] **JQUERY UI**. jQuery User Interface. Demos & Documentation . 2012. Disponível em:<<http://jqueryui.com/demos/>>. Acesso em: 25/09/2019.

[9] Kurose, J. and Ross, K. (2013). *Redes de computadores e a internet*. 6th ed. São Paulo: Pearson, pp.22-27.J.

[10] Leung, L. (2017). *Virtual Ethnicity*. 2nd ed. New York,: Routledge, pp.9 - 12.Retrieved from http://www.oxfordreference.com

[11] **PINTO, F. D. M.** Desenvolvimento de um Protótipo de um Sistema Domótico. 2010. 79f. Dissertação (Mestrado em Engenharia Eletrotécnica e de

https://dx.doi.org/10.22161/ijaers.611.23 *ISSN: 2349-6495(P) | 2456-1908(O)*

Computadores) – Intituto Superior Técnico, Lisboa. Portugal, 2010.

[12] Stevan Jr., S. (2019). *Domótica: Automação residencial e casas inteligentes com Arduino e ESP8266*. 1st ed. São Paulo: Érica, pp.18,19. Perfect, T. J., & Schwartz, B. L. (Eds.) (2002). Applied metacognition Retrieved from http://www.questia.com/read/107598848

[13] **WEBDUINO**. Simple and extensible web server for Arduino and Ethernet Shield. 2019. Disponível em <<http://code.google.com/p/webduino/>>. Acesso em 21/09/2019.

[14] **K. Ghosh, B., Tarn, T. and Xi, N. (1999**). *Control in Robotics and Automation*. 1st ed. San Diego: Academic Press.

[15] Lipovski, G. (2004). *Introduction to microcontrollers*. 2nd ed. Amsterdam: Elsevier Academic Press.

[16] Plonus, M. (2001). *Electronics and Communications for Scientists and Engineers*. 1st ed. San Diego: HARCOURT/ACADEMIC PRESS.