

# **BANCADA DIDÁTICA MODULAR PARA AUTOMAÇÃO RESIDENCIAL**

*Thiago de Oliveira, Oziel da Silva*

Programa de Engenharia Eletrônica  
Faculdade Gama - Universidade de Brasília  
13/0038199  
email: thiago\_ocr@hotmail.com,

*Oziel da Silva*

Programa de Engenharia Eletrônica  
Faculdade Gama - Universidade de Brasília  
12/0131536  
email: oziel.service@live.com

## **1. INTRODUÇÃO**

A automação residencial é um ramo da domótica que surgiu por volta dos anos 80 para gestão e monitoramento da iluminação, condições climáticas e segurança[1]. Observou-se que as residências são bens duráveis e que devem proporcionar segurança e conforto. Os sistemas embarcados podem ser utilizados na domótica e na automação residencial para facilitar a interação do usuário com o sistema. Microcontroladores são comumente utilizados para a automação residencial, entretanto, estes não possuem tanta robustez quanto um sistema embarcado pode proporcionar. Pensando nisso, decidiu-se montar um sistema capaz de fazer a automação residencial gerando uma casa inteligente, entretanto, tais funções serão desenvolvidas de forma didática -bancada didática-afim de serem apresentadas na disciplina de Sistemas Embarcados na Universidade de Brasília.

## **2. OBJETIVOS**

Implementar de modo didático e funcional, um módulo de automação residencial, que permite acionamento de cargas, leitura de sensores e supervisão por meio de câmera digital.

Figura 2: Implementação física da Figura 1. Utilizou-se resistências em séries para montar algumas resistências equivalentes.

## **3. REQUISITOS**

### **A. Requisitos Funcionais**

- Leitura de Sensores
- Acionamento de Cargas
- Visualização de Vídeo

### **B. Requisitos Não Funcionais**

- Câmera Digital.
- Módulos de Potência
- Microprocessador
- Microcontrolador
- Computador Móvel Pessoal.

## **4. BENEFÍCIOS**

Sistema em questão pode ser utilizado em duas vertentes sendo elas: Automação Industrial e Automação Residencial. Nesta aplicação será voltado para automação residencial, proporcionando, uma maior comodidade, praticidade e segurança ao usuário.

## **5. REVISÃO BIBLIOGRÁFICA**

Pretende-se montar uma bancada didática para proporcionar um maior entendimento da domótica, automação residencial e o conceito de casas inteligentes. Espera-se que a bancada didática seja capaz de controlar cargas maiores -como lâmpadas e equipamentos eletrônicos-, gerar uma interface gráfica para a comunicação do usuário com o sistema, uma interação das cargas com o usuário de forma unidirecional (o usuário apenas recebe/percebe as cargas e faz o gerenciamento apenas através do sistema embarcado) e possivelmente ser integrado a um sistema supervisor robusto e eficiente. A Figura 1 nos mostra um fluxograma demonstrando o funcionamento do Sistema embarcados aplicado a domótica para o desenvolvimento da bancada didática de automação residencial.

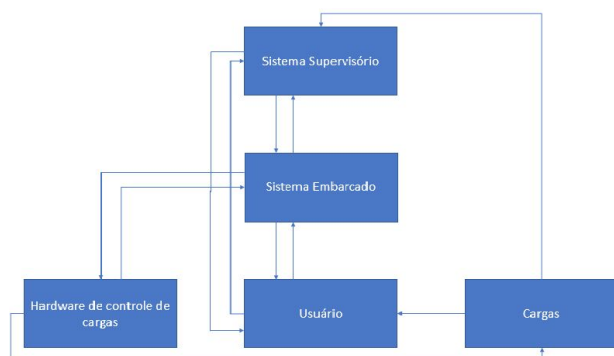


Figura 1: Fluxograma demonstrando o funcionamento do Sistema embarcado aplicado a domótica para o desenvolvimento da bancada didática de automação residencial.

Para a montagem da bancada didática pretende-se utilizar um Raspberry Pi 3 modelo B, pois, é o sistema embarcado possuído pelo grupo. A Raspberry Pi pode operar um sistema operacional baseado no Debian GNU/Linux chamado RASPBIAN. Além disso, este embarcado possui quatro entradas USB, uma entrada Ethernet, uma entrada para câmera, display, pinos de GPIO, entrada para cartão microSD e uma entrada micro USB para alimentação energética da placa [2].

O sistema embarcado será responsável por fazer o controle de cargas maiores -normalmente altas tensões como por exemplo 110 ou 220V- através de sinais com cargas menores -normalmente cargas menores como por exemplo 5 ou 12V-. Para o controle dessas cargas não é necessário apenas um sistema de software embarcado, pois, necessita-se de um hardware capaz de auxiliar no controle das cargas maiores. Este hardware pode ser comprado a um custo baixo e instalado de maneira modular com outros hardwares que possam ser utilizados posteriormente ou pode-se montar este hardware por se tratar de um hardware que pode ser simples -a depender da complexidade exigida no projeto-. Normalmente o hardware responsável pela ativação de cargas maiores utiliza relés e um sistema de proteção contra correntes reversas que podem ser causadas pelas cargas indutivas.

## 6. DESENVOLVIMENTO

Desenvolveu-se um software capaz de controlar o hardware para o controle, acionamento de cargas maiores, comunicação entre o cliente e servidor. Os componentes do hardware em sua maioria foram comprados -uma placa composta por oito relés para

acionamento de cargas-, lâmpadas, Raspberry Pi, câmera, fonte. Montou-se um conversor analógico digital.

### 6.1. DESCRIÇÃO DO SOFTWARE

O Software é robusto e utiliza diversos recursos de sistemas. Dentro do executável desenvolvido é possível fazer o acionamento de cargas maiores através dos pinos de GPIO do Raspberry Pi. O executável está sendo criado de forma modular.

O programa do servidor será chamado de programa principal e o programa do cliente será chamado de programa secundário. O programa principal é rodado no servidor e há um programa que deve ser rodado no cliente para que haja a comunicação entre os dois. O programa principal se comunica com o programa secundário através de um socket -comunicação entre programas-. Este socket é complementado através de diversos recursos do sistemas para um melhor gerenciamento do hardware. Dentro do software secundário já houve a implementação do acionamento de cargas externas através dos pinos de GPIO -esta parte será utilizada em diversas partes do código, pois, a automação irá simular diversos cômodos de uma casa-. Os pinos de GPIO também foram usados para a leitura de dados analógicos, entretanto, como a Raspberry Pi não faz leituras analógicas desenvolveu-se um conversor A/D (conversor analógico digital) para fazer a leitura da temperatura de diversos cômodos (simula-se diversos cômodos de uma residência). O software secundário possui um parte que é executada de forma paralela com os outros processos -entidades ativas-. A inclusão é feita através da biblioteca pthreads.h. A inclusão das bibliotecas necessárias podem ser vistas no Exemplo 1. É importante ter conhecimento das bibliotecas utilizadas para efeitos de compilação, manutenção do código e entendimento. A compilação dos programas é feito através do processo de biblioteca estática, entretanto, caso surja a necessidade de um compartilhamento de memória para melhor gerenciamento dos recursos de software haverá a mudança da compilação para a compilação sobre demanda ou através da utilização de biblioteca dinâmica, mas acredita-se (através das análises feitas até o momento) que a compilação estática seja mais rápida e não irá gerar muita perda de recursos computacionais mesmo sabendo que o hardware utilizado não é tão potente como um desktop ou notebook atual.

*//cliente local*

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <sys/un.h>
#include <unistd.h>
#include <pthread.h>
(a)

```

```

// Servidor Local
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <signal.h>
#include <wiringPi.h>
#include <mcp3004.h>
#include <pthread.h>
(b)

```

Exemplo 1: Descrição das bibliotecas utilizadas pelo cliente (a) e servidor (b).

O programa principal é interrompido através do sinal 2 da SIGINT (Ctrl + c) e quando interrompido acaba-se a comunicação do servidor/cliente.

Utilizou-se o modelo servidor/cliente

## 6.2. DESCRIÇÃO DO HARDWARE

## 7. REFERÊNCIAS

- <sup>[1]</sup> BELEZA, José Eduardo do Monte Moreira et al. Sistema integrado de segurança e domótica. 2012.
- <sup>[2]</sup> PI, Raspberry. Raspberry pi. Raspberry Pi, v. 1, p. 1, 2013.