

---

# AUTOMATIZAÇÃO RESIDENCIAL: QUARTO INTEGRADO VIA ARDUINO

---

**Bruno Bronzeri**  
Eng<sup>a</sup> de Controle e Automação  
UFSC - Blumenau

**Gabriel Mariano**  
Eng<sup>a</sup> de Controle e Automação  
UFSC - Blumenau

**Leonardo Schmitt**  
Eng<sup>a</sup> de Controle e Automação  
UFSC - Blumenau

**Prof<sup>o</sup> Orientador: Carlos Roberto Moratelli**  
Universidade Federal de Santa Catarina  
Campus Blumenau

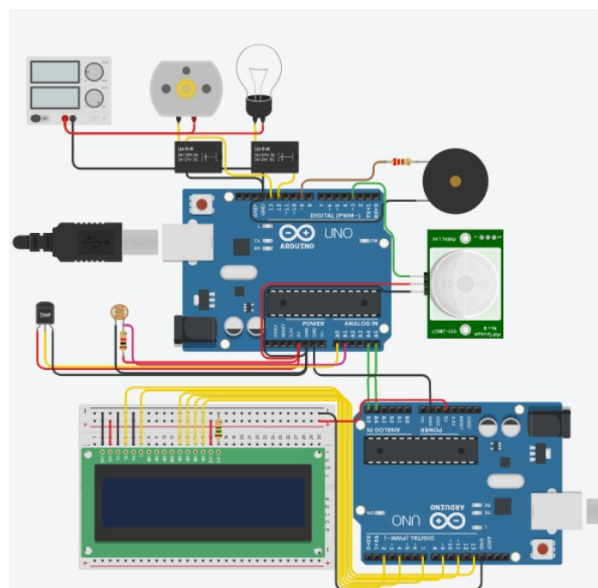
14 de Março, 2022

## 1 Introdução

A tecnologia por definição, tem como finalidade, trazer uma solução para auxiliar um processo ou um método da atividade humana. Na contemporaneidade, a utilização de hardwares e softwares para automatização vem se tornando cada vez mais recorrente. Por esse motivo, este trabalho traz consigo, a ideia/protótipo de uma automatização residencial, feita em Arduino.

O propósito é, fundamentalmente, transformar um processo humano em um processo automático, criando maior acessibilidade às pessoas, devido ao menor custo, e tornando o dia-a-dia mais confortável e otimizado. Isso se dá, justamente pelo enfoque trazido à tona, em cima do projeto de obter tecnologias a favor da sociedade, efetuando tarefas e trazendo benefícios à ela.

Figure 1: Circuito TinkerCad



Fonte: Autor

## 2 Trabalhos Relacionados

Analogamente, encontra-se também, outras formas de automatização residencial, como a Alexa e Google Home, por exemplo - inteligência artificial da Amazon e da Google, respectivamente -. Ambas baseadas no conceito de IOT e são proporcionados apenas o aparelho de comunicação direta IA-usuário. A IAsmin, por sua vez assemelha-se muito com este conceito, trazendo novidades como uma redução do preço, além de diversas funcionalidades que podem ser facilmente instalados. Por se tratar de um protótipo feito em cima da plataforma TinkerCad, infelizmente a comunicação sonora torna-se limitada.

## 3 Desenvolvimento

A automatização residencial feita com Arduino, a priori, consiste em entregar ao usuário uma experiência já completa juntamente com os periféricos. Dessa forma é possível ao usuário controlar e programar funções básicas de sua casa - como: lâmpadas, ar condicionado, alarmes e outros - por meio de uma comunicação feita entre duas placas Arduino UNO.

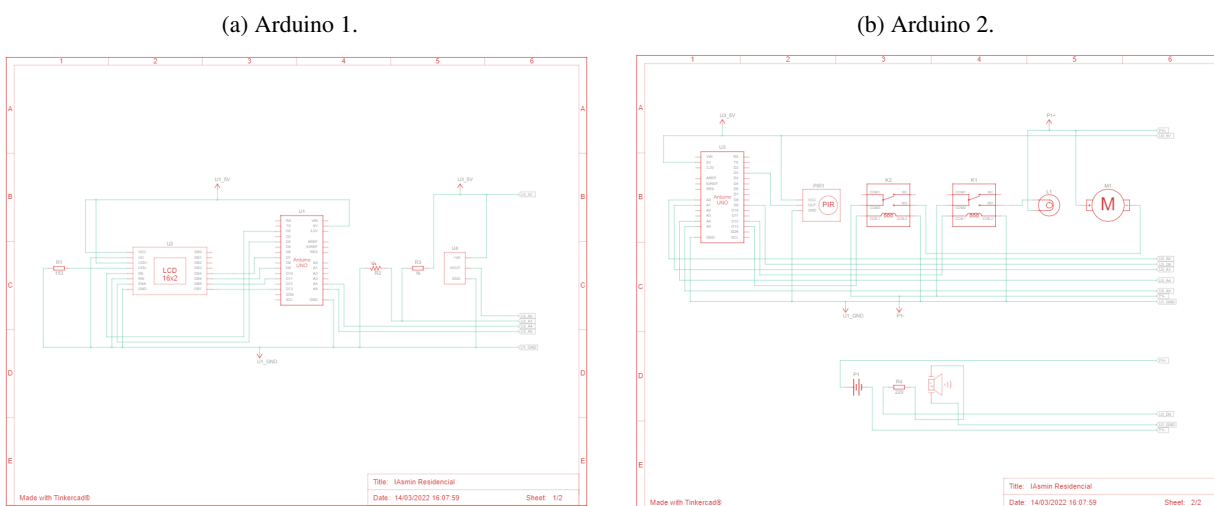
O projeto consiste entre essas duas placas Arduino UNO, conectadas por um protocolo de comunicação I2C, em que na placa mestre está localizado apenas a central de comunicação direta com o usuário, e na escravo a ligação com os periféricos. Ou seja, o usuário interage com o Arduino 1 (mestre), que simultaneamente se comunica com o Arduino 2 (escravo), passando parâmetros que funcionam como comandos para execução das tarefas respectivas desejadas pelo usuário.

### 3.1 Circuito elétrico

Componentes conectados em cada placa Arduino UNO:

Arduino 1: Display LCD 16x2 e um Resistor 160  $\Omega$ ; & Arduino 2: Sensor de Temperatura TMP36, Fotorresistor PIR, Piezo, Motor CC, dois Relês, uma Lâmpada, dois Resistores (um de 220  $\Omega$  e outro de 1000  $\Omega$ ) e uma Fonte 5V. Podemos estar analisando o circuito elétrico e os respectivos componentes, conforme mostra a figura 1.

Figure 2: Esquemas Elétricos



Fonte: Autor

### 3.2 Funcionamento Arduino 1

No Arduino 1, é onde acontece a comunicação direta com o usuário pelo monitor serial, de modo a simular a interação do protótipo com a voz do usuário. No display LCD é possível interagir com todas as opções fornecidas pela IAsmin. Logo, o funcionamento se dá invocando a IAsmin digitando qualquer letra no Monitor Serial do Arduino 1, simulando mais uma vez a interação com a voz do usuário, baseando-se na ideia de iniciar a Alexa da Amazon, ou a Siri da Apple. Ao fazê-lo, o display LCD exhibe todas as funções disponíveis: [1] Ar condicionado; [2] Luz; [3] Presença; [4] Cancelar;

Cada opção selecionada pelo usuário são enviados os parâmetros para o Arduino 2 de modo com que este execute as tarefas nos periféricos desejados. Ao selecionar a opção [3]Pres, é possível ligar e desligar o alarme de presença digitando [1] ou [2] respectivamente, além de atribuímos para a opção [3] a ação de voltar para a tela inicial. Voltando à tela inicial, ao selecionar a opção [1]Ar ou [2]Luz, são exibidas novas possibilidades de comando: [1] On; [2] Off; [3] Configuração; [4] Voltar;

Ao optar pela opção [3], o usuário tem acesso a configuração do periférico, ou seja, no caso do ar condicionado, o usuário escolhe a temperatura para o ambiente, medida pelo sensor de temperatura e no caso da luz, ele escolhe a luminosidade mínima para acender/apagar a luz, medida pelo fotoresistor. E para ambos os casos, as configurações são mantidas, sendo possível navegar entre outras funções sem perder a configuração vigente e é possível alterá-las sempre que desejado. Um ponto a ser destacado é que os números devem ser digitados um de cada vez.

No caso, para a opção [4]Cncl da página inicial, esta tem a função de desativar as luzes e o ar condicionado, no entanto mantendo o alarme ainda funcional, caso ele tenha sido previamente ligado. Para fins práticos, significa que o usuário liga o alarme e sai de sua residência, por isso, ao cancelar, o alarme é o único que se mantém ativo.

### 3.3 Funcionamento Arduino 2

O funcionamento do Arduino 2 por sua vez, diferencia-se do Arduino 1, justamente por se tratar de ser um escravo e obdecer a ordens do seu mestre (Arduino 1). O protocolo de comunicação I2C, permite que o mestre mande a informação para o escravo em forma de byte, com isso o Arduino 2 recebe determinados bytes, para cada tipo de função que deve realizar.

Podemos citar como exemplo ligar o ar condicionado, caso o usuário desejar. O Arduino 1, escreve byte 1 no endereço onde está localizado o escravo, logo após o escravo lê esse byte, alterando o estado do ar condicionado, a figura 2, mostra o código fonte do exemplo em questão. Vale ressaltar ainda, que este método de funcionamento, se dá para todos os periféricos que estão conectados no Arduino 2, contudo, existe um byte diferente para cada dispositivo. Nesse sentido o Arduino 2 é responsável pelo controle dos periféricos, este recebe dados fornecidos pelo usuário por meio do Serial, com o auxílio da comunicação I2C, que necessita a utilização da biblioteca Wire.h, cuja tem funções que facilitam a comunicação entre as duas placas.

Figure 3: Código Fonte

(a) Código Mestre.

```
switch (n){
  case 49: // liga o ar
    Serial.println("Ar ON");
    Wire.beginTransaction(0x08);
    Wire.write(1);
    Wire.endTransmission();
    n=-1;
    break;
```

(b) Código Escravo.

```
if(a==1) { // Estrutura if para ligar o ar
  digitalWrite(pino_ar, HIGH);
  contador_ar = 0;
}
```

Fonte: Autor

## 4 Testes e Validação

O prazo de desenvolvimento do projeto foi de 7 dias. Logo no primeiro já foram debatidas ideias e possibilidades e feitos alguns testes referentes à formas diferentes de comunicação entre os Arduinos. No segundo dia foi decidida a implementação de um despertador que pudesse ser configurado também pelo usuário para um horário desejado e se abriria ou não cortinas juntamente. Porém após tentativas a ideia foi abandonada por falta de registradores e inviabilidade de trabalhar com o timer juntamente com tantos outros periféricos. Além disso, pelo fato de ser uma simulação que ocorre no TinkerCad, diversas limitações de processamento nos afastam das condições reais do Arduino.

Contudo, todos os dias houve investimento no código, testes frequentes e conserto de *bugs*, além de sempre buscar reduzir a complexidade ciclomática do código. Indentação e aprimoramento do algoritmo de forma com que fosse possível aumentar a eficiência, mesclando tanto a utilização de comentários em todas as linhas possíveis como boas práticas de programação. Até enfim, alcançar um projeto satisfatório e sem bugs que passou em todos os testes efetuados pelos desenvolvedores.

---

## 5 Conclusão

Este projeto trouxe à tona a prática e a semelhança da realidade de um projeto de automação. As dificuldades presentes em um projeto, foram contornadas, portanto, com base em métodos, pesquisas e testes. Visando sempre resolver qualquer problema de maneira completamente racional e viável. Destarte, a obtenção de um algoritmo extremamente otimizado, facilita o entendimento e livra a dificuldade de falhas de testes. Nesse caso, por meio da intersecção entre conceitos, fundamentos e ideias, tornou-se plausível ao usuário desfrutar de um programa original e personalizado, por já se tratar de um projeto integrado com a residência e passível de expansões futuras.

Em anexo link do projeto na plataforma TinkerCad:

<https://www.tinkercad.com/things/7b6jarsqriu-exquisite-gaaris/editel?sharecode=7bJSq6OwvFec5UB-TZOwC7ZYyq864tBWmzvzpPbdpH8>

## 6 Referências

Flávio. Canal brincando com ideias. In [://www.youtube.com/c/BrincandocomIdeias](https://www.youtube.com/c/BrincandocomIdeias)

Daniel Madeira. Protocolo I2C - Comunicação entre dois arduinos. In <https://portal.vidadesilicio.com.br/i2c-comunicacao-entre-arduinos/>

Arduino Team. Serial - Arduino Reference. In <https://www.arduino.cc/reference/en/language/functions/communication/serial/>

Unkown. Tabela ascii. In <https://web.fe.up.pt/ee96100/projecto/Tabela%20ascii.htm>