



UNIVERSIDADE ESTADUAL DO MARANHÃO

Curso de Engenharia Mecânica

JOÃO PEDRO VIEIRA ALVES

**AUTOMAÇÃO RESIDENCIAL COM ARDUINO INTEGRADO AO
APLICATIVO BLYNK**

SÃO LUIS/MA

2021

JOÃO PEDRO VIEIRA ALVES

**AUTOMAÇÃO RESIDENCIAL COM ARDUINO INTEGRADO AO
APLICATIVO BLYNK**

Monografia de graduação apresentada ao Curso de Engenharia Mecânica da Universidade Estadual do Maranhão como parte dos requisitos exigidos para obtenção do título Bacharel em Engenharia Mecânica.

Orientador: Prof. Me. Denner Rodrigues Guilhon

ESTE EXEMPLAR CORRESPONDE À
VERSÃO FINAL DA MONOGRAFIA
DEFENDIDA PELO(A) ALUNO(A) JOÃO
PEDRO VIEIRA ALVES, E ORIENTADO
PELO(A) PROF(A). ME.(A) DENNER
RODRIGUES GUILHON.



ASSINATURA DO(A) ORIENTADOR(A)

SÃO LUIS/MA

2021

FICHA CATALOGRAFICA ELABORADA PELA BIBLIOTECA DA UNIVERSIDADE
ESTADUAL DO MARANHÃO – UEMA

Alves, João Pedro Vieira.

Automação residencial com arduino integrado ao aplicativo blynk /
João Pedro Vieira Alves. – São Luís, 2021.

... f

Monografia (Graduação) – Curso de Engenharia Mecânica,
Universidade Estadual do Maranhão, 2021.

Orientador: Prof. Me. Denner Robert Rodrigues Guilhon.

1.Automação residencial. 2.Arduino. 3.Blynk. 4.Prototipagem. I.Título.

UNIVERSIDADE ESTADUAL DO MARANHÃO
CURSO DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA E PRODUÇÃO

TRABALHO DE CONCLUSÃO DE CURSO

**AUTOMAÇÃO RESIDENCIAL COM ARDUINO INTEGRADO AO
APLICATIVO BLYNK**

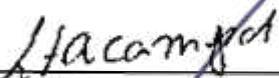
Autor: João Pedro Vieira Alves

Orientador: Denner Rodrigues Guilhon

A Banca examinadora composta pelos membros abaixo aprovou esta Monografia:



Prof. Me. Denner Rodrigues Guilhon
Universidade Estadual do Maranhão - UEMA



Prof. Dr. Lúcio Flávio de Albuquerque Campos
Universidade Estadual do Maranhão - UEMA



Prof. Dr. Mauro Sérgio Silva Pinto.
Universidade Estadual do Maranhão - UEMA

A ata de defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

São Luís, 09 de dezembro de 2021

DEDICATÓRIA

Aos meus pais, os maiores incentivadores e apoiadores dos meus sonhos, a quem dedico todas as minhas vitórias.

AGRADECIMENTOS

Primeiramente agradeço a Deus, por iluminar o caminho até aqui, e dar sabedoria nos momentos mais difíceis.

Aos meus pais e familiares por terem me incentivado e apoiado para que eu chegasse até aqui.

A minha companheira de vida, Girla Zaine da Silva Portela, por sempre está ao meu lado nos momentos mais difíceis

A Universidade Estadual do Maranhão, por proporcionar toda a estrutura e conhecimento que me moldaram durante esses anos, a todo corpo docente do curso de Engenharia Mecânica.

RESUMO

Este trabalho tem caráter descritivo com o intuito de apresentar um modelo de projeto de automação residencial de fácil implementação através de tecnologias integralizadas a placa de prototipagem Arduino juntamente com o aplicativo Blynk, podendo ser implementado com pouca ou nenhuma linha de código, gerando facilidade de acesso e comodidade a seus usuários. Além disso, com avanço dessa tecnologia é possível gerar uma economia no consumo de energia no ambiente em que está instalada, diminuindo os impactos ambientais gerados pela ampliação ou construção de novas usinas hidrelétricas para suprir a demanda nacional.

Palavras-chave: Automação Residencial, Arduino, Blynk, Prototipagem.

ABSTRACT

This work has a descriptive character in order to present a home automation project model of easy implementation of technologies integrated to the Arduino prototyping board together with the Blynk application, which can be implemented with little or no line of code, generating ease of access and presentation to your users. Furthermore, with the advancement of this technology, it is possible to generate savings in energy consumption in the environment in which it is installed, reducing the impacts generated by the expansion or construction of new hydroelectric plants to meet national demand.

Keywords: Home Automation, Arduino, Blynk, Prototyping.

LISTA DE ILUSTRAÇÕES

Figura 1.1 – Níveis dos reservatórios.....	14
Figura 4.1 Ciclo PWM.....	21
Figura 4.2 Sensor de Nível.....	28
Figura 4.3 Sensor de Nível.....	29
Figura 5.1 Criação do projeto blynk.....	16
Figura 5.2 Configuração do teste.....	18
Figura 5.3 Circuito teste de conexão.....	19
Figura 5.4 Planta da casa usada no projeto.....	21
Figura 5.5 Ligação do circuito do relé.....	23
Figura 5.6 Módulo relé 16 canais.....	24
Figura 5.7 Acionamento do LED via internet.....	25
Figura 5.8 Acionamento por relé via internet.....	26
Figura 5.9 Circuito do LDR acionado.....	27
Figura 6.1 Módulo relé integrado ao sensor PIR.....	32
Figura 6.2 Conexão do relé com a sirene.....	32
Figura 6.3 Integração do RFID com a trava elétrica.....	34
Figura 6.4 Adição do eventor no blynk.....	37

LISTA DE TABELAS

Tabela 5.1 Código de integração ao Blynk	16
Tabela 5.2 Lista de equipamentos e módulos	22
Tabela 6.1 Código de acionamento do relé.....	28
Tabela 6.2 Código do sensor de luz com LDR	29
Tabela 6.3 Código do sensor de nível	30
Tabela 6.4 Código do sensor de presença com sirene.....	33
Tabela 6.5 Pinagem do RFID.....	34
Tabela 6.6 Código da fechadura eletrônica integrada ao RFID	34
Tabela 6.7 Código para adicionar pino virtual ao Blynk	36
Tabela 6.8 Orçamento com base no site Aliexpress	38
Tabela 6.9 Orçamento com base no site MercadoLivre.....	38

LISTA DE ABREVIATURAS E SIGLAS

APP – Aplicativo

IDE - Arduino Integrated Development Environment

LDR - Light-Dependent Resistor

LED - Light Emitting Diode

MISO - Master in Slave Out

MOSI - Master Out Slave In

NA – Normalmente Aberto

NOS - Operador Nacional do Sistema Elétrico

NRDC - Natural Resources Defense Council

PLC - Power Line Carrier

PWM - Pulse-Width Modulation

RAM - Random Access Memory

RFID - Radio-Frequency IDentification

ROM - Read Only Memory

SCK - Serial Clock

SPI - Serial Peripheral Interface

SS - Slave Select

SUMÁRIO

1	INTRODUÇÃO	13
2	OBJETIVOS	16
2.1	OBJETIVO GERAL	16
2.2	OBJETIVOS ESPECÍFICOS	16
3	JUSTIFICATIVA	17
4	REVISÃO TEÓRICA	18
4.1	Arduino	18
4.1.1	Sistemas eletrônicos	19
4.1.2	Entradas e saídas digitais	19
4.1.3	Modulação por largura de pulso PWM	20
4.1.4	Comunicação serial	21
4.2	MÓDULOS	22
4.2.1	Módulo relé	22
4.2.2	Sensor de presença hc-sr501	24
4.2.3	Fechadura eletrônica	25
4.2.4	Chave RFID	25
4.2.5	Sensor de nível de água	27
4.2.6	Sensor de luminosidade LDR	28
4.3	APP BLYNK	29
5	MATERIAIS E METODOS	15
5.1	INSTALAÇÃO DO APP BLYNK E CONEXÃO	15
5.2	DEFINIÇÃO DA RESIDÊNCIA E EQUIPAMENTOS AUTOMATIZADOS	20
5.3	MÓDULOS PARA AUTOMATIZAÇÃO	23
5.4	teste de acionamento	25

6	RESULTADOS E DISCUSSÕES	28
6.1	Código puro em arduino para acionamento dos modulos	28
6.1.1	Acionamento do relé	28
6.1.2	Sensor de luz com LDR	29
6.1.3	Sensor de nível.....	30
6.1.4	Sensor de presença.....	31
6.1.5	Fechadura eletrônica integrada ao RFID	33
6.2	integração dos modulos ao blink	36
6.3	Orçamento do projeto	37
7	CONCLUSÕES	39
	REFERÊNCIAS	40

1 INTRODUÇÃO

A automação residencial, também conhecida como domótica, é um assunto cada vez mais presente no cotidiano, quebrando paradigmas sobre o uso da tecnologia. Onde antes era algo específico de uma classe economicamente alta, hoje está se tornando cada vez mais popular.

Os primeiros desenvolvimentos na área de automação residencial datam do final da década de 1970, quando surgiram nos Estados Unidos os primeiros módulos “inteligentes”, cujos comandos eram enviados pela própria rede elétrica da residência, no conceito de Power Line Carrier (PLC). Tratavam-se de soluções simples, praticamente não integradas e que resolviam situações pontuais, como ligar remotamente algum equipamento ou luzes (MURATORI J. E BÓ P., 2011).

Nas economias mais desenvolvidas, o cenário para as chamadas “casas inteligentes” tem evoluído de maneira muito positiva nos últimos anos. Tem contribuído para isso a crescente popularização de diversas tecnologias, seja pelo aspecto educativo do consumidor, seja pelos preços decrescentes (MURATORI J. E BÓ P., 2011).

A automação residencial está relacionada a um conjunto de tecnologias com a capacidade de gerenciar eventos em uma casa ou apartamento, para tornar automático o funcionamento e comando dos equipamentos desejados dentro e fora da residência. Através de um sistema integrado e conexão com a internet, a automação residencial tem o intuito de criar uma casa inteligente. Uma das tecnologias que tornam essa integração possível é a placa de prototipagem Arduino conectado ao Aplicativo (App) Blynk.

O objetivo é oferecer praticidade aos moradores por meio do gerenciamento remoto através da conexão com um smartphone ou tablet, podendo ter o controle dos dispositivos da casa em qualquer lugar. Pode-se também, programar sensores, fechaduras, alarmes, luzes, etc. Além das facilidades oferecidas pela automação residencial, também é possível tornar a residência mais segura com a instalação de sensores de movimento, câmeras de segurança, alarmes de vazamento de gás, entre outros dispositivos.

Outro fato importante a ser observado é a função econômica e sustentável oferecida, pois segundo a NOS - Operador Nacional do Sistema Elétrico - os níveis dos reservatórios apresentam o menor armazenamento médio para o mês de maio desde 2021, ano que o país enfrentou um racionamento de energia, como mostra a figura 1.1 (AMATO ,2021).

Um estudo paralelo sobre o consumo de energia de equipamentos plugados permanentemente nas tomadas realizado em 2015 por NRDC - *Natural Resources Defense*

Council - realizado nos Estados Unidos, mostrou que 23% do consumo de energia elétrica de uma residência é decorrente de cargas ociosas e a tendência é o crescimento desse gasto.

Um exemplo dessa situação é quando uma lâmpada externa de uma casa fica ligada todo o período noturno. Diante disso, a integração de um sistema inteligente a residência auxilia na redução de gastos com as cargas ociosas, reduzindo a demanda por energia elétrica e consequentemente diminuindo de demanda nas usinas do país.

Nível dos reservatórios do Sudeste e Centro-Oeste ao final de maio

Em % do total

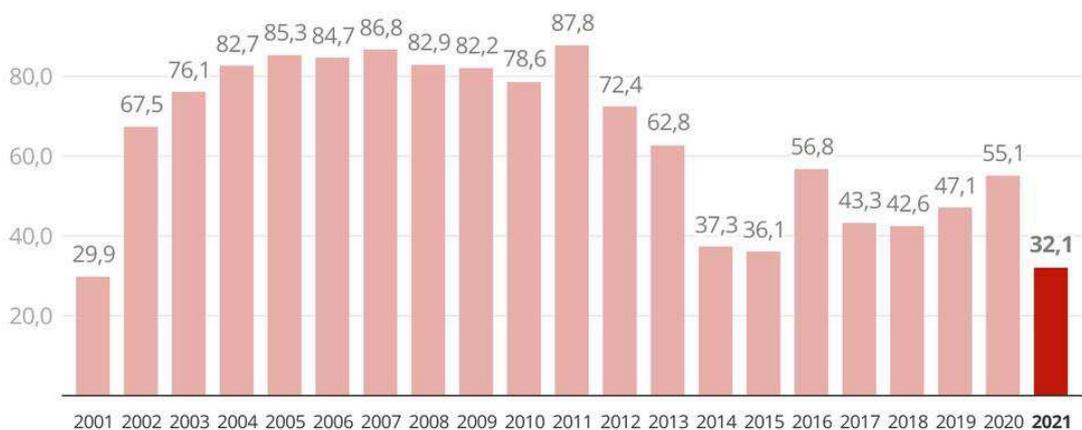


Figura 1.1 – Níveis dos reservatórios

(G1, 2021. Disponível em: <<https://g1.globo.com/economia/noticia/2021/06/02/nivel-dos-reservatorios-de-sudeste-e-centro-oeste-em-maio-e-o-mais-baixo-para-o-mes-desde-2001.ghtml>>).

Este trabalho visa facilitar o acesso aos dispositivos de automação, uma vez que será utilizado um app para a integração com a placa de prototipagem Arduino, que possui um preço acessível comprada com os dispositivos de automação convencionais.

Tendo em vista a facilidade de integração, será utilizado o Blynk - que é um serviço baseado em um aplicativo personalizável que permite controlar remotamente um hardware programável, bem como reportar dados do hardware ao aplicativo. Desta forma, é possível construir interfaces gráficas de controle de forma rápida e intuitiva e que interage com mais de 400 placas de desenvolvimento, em sua maioria baseadas em Arduino.

Nesse contexto, a integração entre o Arduino e o Blynk é intermediado pela internet, podendo ser acionado de qualquer lugar com acesso à rede. A comodidade gerada por essa tecnologia, que vem sendo implantada nas instalações elétricas residenciais, garante mais

conforto e segurança, além de proporcionar um controle que pode ser estabelecido pelo usuário de acordo com a sua necessidade.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Este trabalho tem por objetivo principal descrever um modelo de projeto de automação residencial através do método explicativo para facilitar a implantação com o Arduino e o app Blynk.

2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- Fornecer um projeto de automação residência ao usuário de baixo custo.
- Exemplificar as conexões para os equipamentos em código Arduino;
- Desenvolver a integração para automatizar o funcionamento de lâmpadas, eletrodomésticos, bombas elétricas e portas com comandos via *Internet*.

3 JUSTIFICATIVA

A automação residencial está se tornando cada vez mais popular, assim se faz necessário a integralização e disseminação de tecnologias que auxiliam o desenvolvimento prático desta aplicação. Além da praticidade e comodidade, a automação residencial pode proporcionar economia de energia, uma vez que equipamentos conectados a tomada em modo de stand-by continuam a consumir energia elétrica.

Desse modo, é perceptível a função ambiental que essa tecnologia pode oferecer a seus usuários, já que com a redução do consumo a demanda exigida nas usinas diminui, não sendo necessária a ampliação ou construção de novas hidrelétricas.

Além disso, tecnologias robustas nesse setor possui um valor agregado elevado, mas com a utilização da placa de prototipagem juntamente com o software já desenvolvido o conjunto é barateado tornando mais acessível para o público em geral.

4 REVISÃO TEÓRICA

Neste capítulo será abordado o referencial teórico e bibliográfico, referente a placa de prototipagem Arduino e o app de integração Blynk. Posteriormente, será demonstrado como é feita a conexão do software com hardware, mais precisamente a instalação das bibliotecas para utilizar os dispositivos.

4.1 ARDUINO

Arduino é uma plataforma de eletrônica aberta para a criação de protótipos baseada em software e hardware livres, flexíveis e fáceis de usar. Foi desenvolvida para artistas, designers, hobistas e qualquer pessoa interessada em criar objetos ou ambientes interativos. Ele pode adquirir informação do ambiente através de seus pinos de entrada, para isso uma completa lista de sensores pode ser utilizada. Ademais, o Arduino pode atuar no ambiente controlando luzes, motores ou outros atuadores.

Os campos de atuação para o controle de sistemas são robustos, podendo ter aplicações na área de impressão 3D, robótica, engenharia de transportes, engenharia agrônômica, musical, moda e tantas outras. O microcontrolador da placa é programado mediante a linguagem de programação Arduino, baseada em *Wiring*, e o ambiente de desenvolvimento (IDE) está baseado em *Processing*.

Os projetos desenvolvidos podem ser executados mesmo sem a necessidade de estar conectados a um computador, apesar de que também podem ser feitos comunicando-se com diferentes tipos de software (como *Flash*, *Processing* ou *MaxMSP*).

O hardware de um Arduino é simples, contudo, muito eficiente, baseado nos microcontroladores AVR da Atmel, os modelos ATmega8, ATmega168, ATmega328 e no ATmega1280. O Arduino recebe um codinome em italiano dependendo de cada microcontrolador que é utilizado. (MCROBERTS, 2010).

O núcleo de processamento é um microcontrolador, um computador completo, com memória RAM, ROM, uma unidade de processamento de aritmética e os dispositivos de entrada e saída. Tudo isso em um único chip. (BASCONCELLO FILHO, 2013).

4.1.1 Sistemas eletrônicos

Um sistema eletrônico é um conjunto de circuitos que interagem entre si para obter um resultado. Uma forma de entender os sistemas eletrônicos consiste em dividi-los em **entradas, saídas e processamento de sinais**.

- **Entradas:** As entradas, ou inputs, são sensores eletrônicos ou mecânicos que tomam os sinais (em forma de temperatura, pressão, umidade, contato, luz, movimento etc.) do mundo físico e converte em sinais de corrente ou voltagem. A exemplos de entradas tem-se os sensores de gás, temperatura, pulsadores, fotocélulas, potenciômetros, sensores de movimento, e etc.

- **Saídas:** As saídas, ou outputs, são atuadores, ou outros dispositivos que convertem os sinais de corrente ou voltagem em sinais fisicamente úteis como movimento, luz, som, força ou rotação, entre outros. Exemplos de saídas são motores, LEDs ou sistemas de luzes que acendem automaticamente quando escurece ou um buzzer que gere diversos tons.

- **Processamento de sinais:** O processamento de sinal é realizado mediante circuitos conhecidos como microcontroladores. São circuitos integrados construídos para manipular, interpretar e transformar os sinais de voltagem e corrente vindos dos sensores (entradas) e ativar determinadas ações nas saídas.

4.1.2 Entradas e saídas digitais

As entradas e saídas de um sistema eletrônico serão consideradas como sinais variáveis. Em eletrônica se trabalha com variáveis que são tomadas na forma de tensão ou corrente, que podem simplesmente ser chamados de sinais, que podem ser de dois tipos.

O sinal digital também chamado de variável discreta, se caracterizam por ter dois estados diferentes e, portanto, também podem ser chamadas de binárias. Um exemplo de um sinal digital é o interruptor da campainha de uma residência, já que possui apenas dois estados, ligado e desligado. As entradas e saídas digitais de uma placa de Arduino podem ter 2 níveis de tensão definidos, o nível alto que na maioria dos Arduinos é 5 V, e o nível baixo que é de 0 V.

Já o sinal analógico é aquele que pode tomar um número infinito de valores compreendidos entre dois limites. Grande parte dos fenômenos da vida real são sinais deste tipo. Um exemplo de sistema eletrônico analógico é de uma saída de som de um microfone, que precisa ser amplificado. As ondas de som que são analógicas na sua origem são capturadas e convertidas em uma pequena variação analógica de tensão, denominada sinal de áudio.

4.1.3 Modulação por largura de pulso PWM

A modulação por largura de pulso - mais conhecida pela sigla em inglês PWM (*Pulse-Width Modulation*) - de um sinal ou em fontes de alimentação envolve a modulação de sua razão cíclica (*duty cycle*) para transportar qualquer informação sobre um canal de comunicação ou controlar a quantidade de energia que se envia em uma carga. Por exemplo, ao aplicar o PWM a um LED é possível variar a intensidade do brilho, e o mesmo pode ser aplicado a um motor DC para variar a sua velocidade.

Essa técnica consiste na geração de ondas quadradas em uma frequência muito alta, que pode ser controlada a porcentagem do tempo em que a onda permanece em nível lógico alto. Sua alteração provoca mudança no valor médio da onda, indo desde 0 V (0% de Duty Cycle) a 5 V (100% de Duty Cycle) no caso do Arduino. Conforme visto na Figura 4.1.

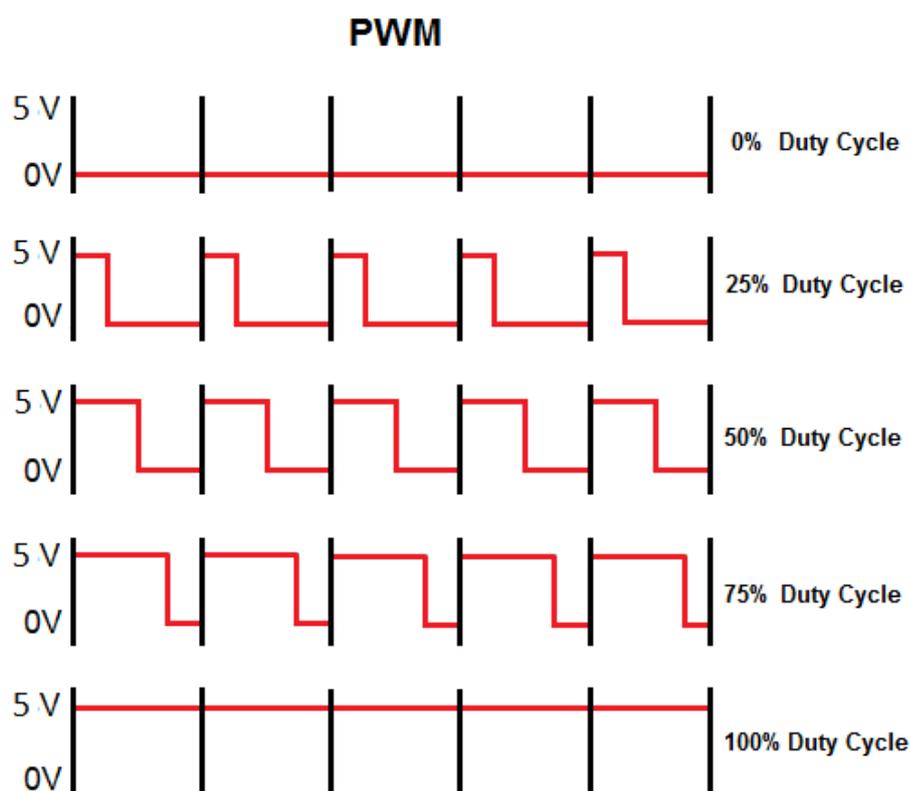


Figura 4.1 Ciclo PWM

(Disponível em: <<https://i1.wp.com/portal.vidadesilicio.com.br/wp-content/uploads/2018/10/PWM.png?resize=527%2C464&ssl=1>>).

As placas de Arduino UNO em seus pinos específicos para saídas PWM são indicados pelo símbolo '~' na frente de seu número. São 6 pinos para saída PWM, são eles: 3,5,6,9,10 e 11.

4.1.4 Comunicação serial

É uma interface de comunicação de dados digitais em que a informação é enviada um bit de cada vez, sequencialmente. É diferente da comunicação paralela, em que todos os bits de cada símbolo são enviados juntos. A comunicação serial é usada em toda comunicação de longo alcance e na maioria das redes de computadores. Um de seus usos é monitorar através da tela do computador o estado de um periférico conectado.

4.2 MÓDULOS

Módulos Arduino são dispositivos eletrônicos formados a partir de placas de circuitos impresso e componentes específicos, a depender da finalidade para o qual foram desenvolvidos. Nos módulos, contém os sensores e outros componentes auxiliares para captar um fenômeno físico específico. Nesse trabalho será abordado apenas os sensores utilizados neste projeto de automação residencial

4.2.1 Módulo relé

O relé é um mecanismo eletromecânico que surgiu no século XIX, é composto por um magneto móvel que se move unido a dois contatos feitos em metal. Foi muito utilizado nos sistemas telefônicos em locais remotos quando se existia as centrais telefônicas analógicas, e são considerados uma espécie de ascendentes dos transistores. Ele é bastante utilizado nos sistemas eletrônicos que necessitam desse dispositivo para controle. (CAMPOS, 2014)

O relé é um dispositivo eletromecânico, formado por um magneto móvel, que se desloca unindo dois contatos metálicos. Quando uma corrente circula pela bobina esta cria um campo magnético que atrai um ou uma série de contatos fechando ou abrindo circuitos – figura 4.2.

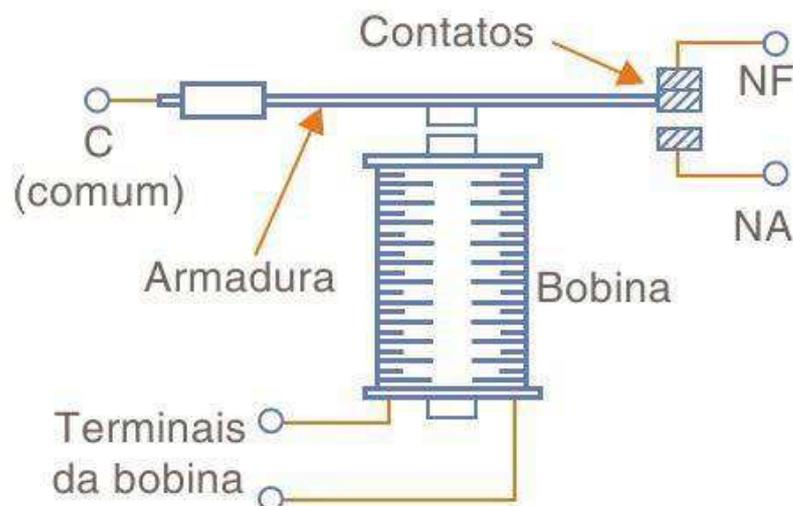


Figura 4.2 Circuito interno

(Disponível em: <<https://www.hardware.com.br/comunidade/circuito-leds/1129187/>>).

Através do módulo relé, pode-se utilizar os pinos digitais do Arduino para chavear sinais de potência e acionar ou desligar qualquer tipo de equipamento. Esse componente é indicado para acionar cargas que utilizam correntes maiores do que a fornecida pelo Arduino. Em geral, liga-se um contato do relé a uma fonte externa de energia, que pode ser o terminal de uma tomada, e o outro terminal é ligado a um dos polos da carga como mostra a figura 4.3.

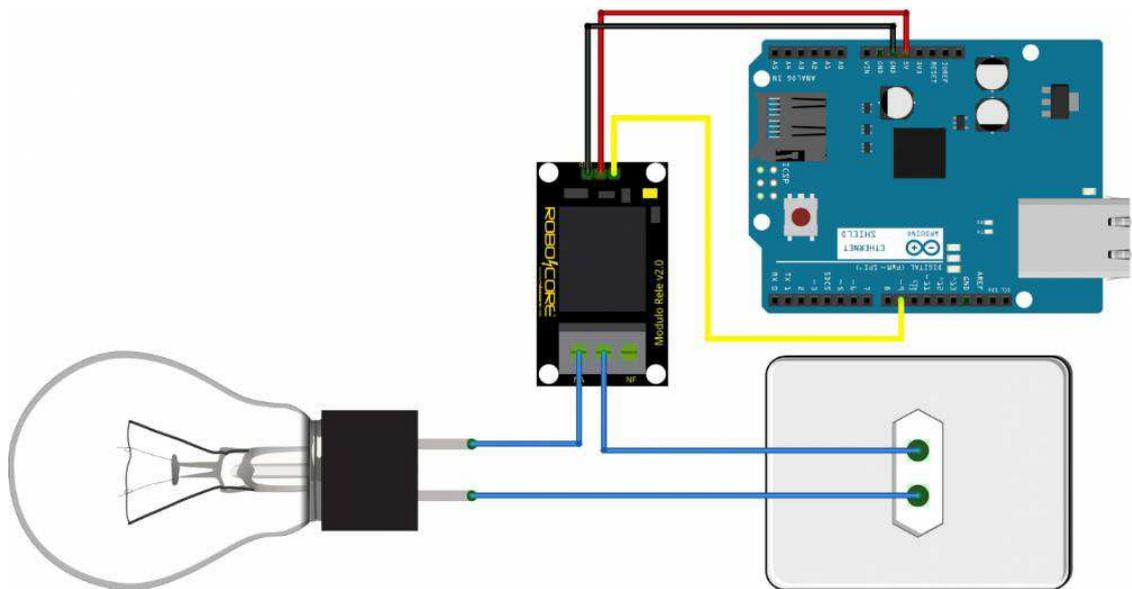


Figura 4.3 Circuito do relé

(Disponível em: <<https://www.robocore.net/tutoriais/acionando-uma-lampada-pela-rede-ethernet>>).

A aplicação mais comum para esse módulo é em lâmpadas em sistemas de automação residencial. Ele funciona como uma chave ou interruptor. No borne de cada relé existem três conexões:

- NA - Normalmente Aberto;
- C - Comum;
- NF - Normalmente Fechado;

4.2.2 Sensor de presença hc-sr501

O sensor de presença é um pequeno dispositivo de fácil integração e comunicação à plataforma Arduino. Ele é responsável pelo monitoramento de pessoas em um ambiente, com um alcance limitado, mas sendo bem posicionado em um local estratégico poderá ser acionado.

Possuindo um sensor piroelétrico de alta sensibilidade (figura 4.4), que consegue detectar a radiação infravermelha emitida pelos corpos, esse sensor é dividido em duas partes, formando dois elementos de medição, de modo que uma variação da radiação infravermelha será detectada por esses elementos piroelétricos.

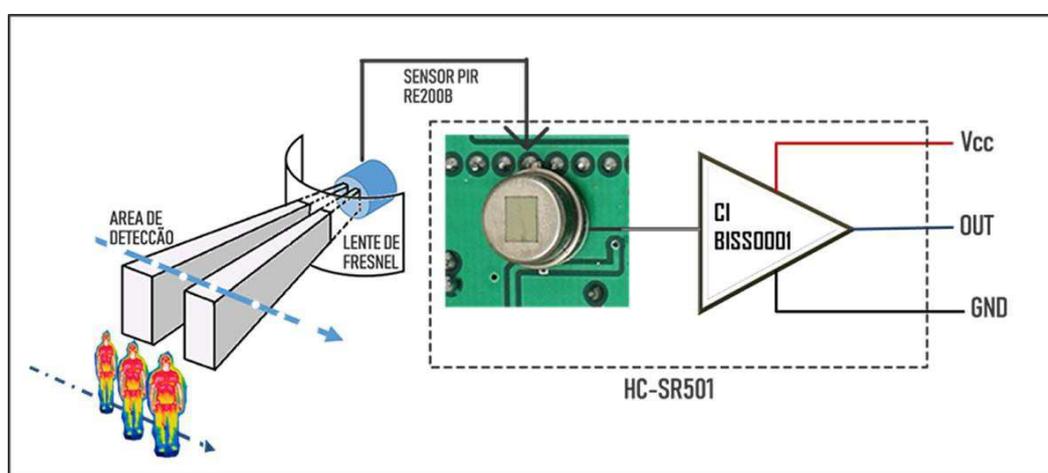


Figura 4.4 Circuito do relé

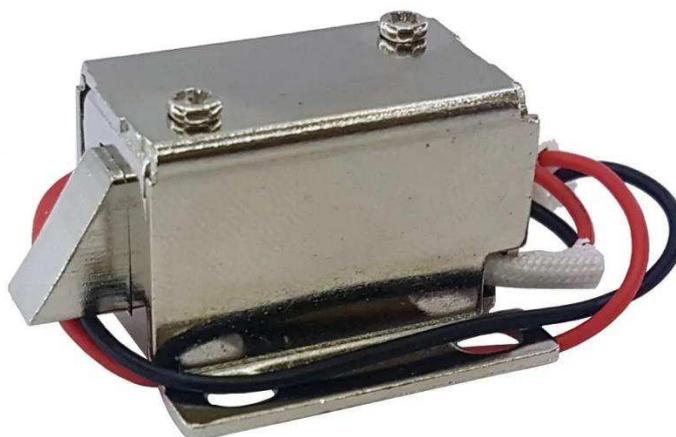
(Disponível em: <<http://blog.moduloeletronica.com.br/detector-de-presenca-com-sensor-pir-hc-sr501/>>).

O HC-SR501 usa o sensor piroelétrico modelo RE200B, encapsulado em um bloco metálico e hermeticamente fechado. Sobre o sensor RE200B é montado uma lente de Fresnel, modelo NL-11NH, com a finalidade de concentrar a radiação para o ponto onde o sensor piroelétrico está localizado na placa do HC-SR501.

Quando um corpo atravessa o campo de medição do sensor PIR HC-SR501, os elementos piroelétricos do RE200B detectam uma mudança no nível de radiação infravermelha e geram um sinal correspondente. Este sinal de saída do RE200B é recebido pelo circuito de detecção e é tratado de forma adequada para gerar o pulso de saída digital. Um exemplo prático da utilização desse sensor é o acionamento de uma lâmpada quando o mesmo identificar algum movimento.

4.2.3 Fechadura eletrônica

A trava elétrica solenoide funciona como uma fechadura convencional, diferenciando-se por ser acionada ao aplicar uma tensão de 12V em seus terminais. Então o pino da trava é recolhido, mantendo-se retraído enquanto a tensão estiver fluindo em seus terminais. Quando não há tensão, o pino volta ao seu estado normal. O Arduino trabalha com 5 volts em seus pinos e uma corrente relativamente baixa (20mA) para o acionamento deste dispositivo, assim, um relé é utilizado para acionar o dispositivo que requer 12V e 600mA para funcionar corretamente.



4.2.4 Chave RFID

O RFID (*Radio Frequency Identification* ou Identificação por Radiofrequência) é um sistema de identificação que captura dados via radiofrequência. Esse é composto por dois dispositivos:

- Um transceptor com decodificador e uma antena;
- Um transponder (Tag).

Quando se aproxima o transponder do leitor RFID, o campo de radiofrequências deste leitor alimenta o circuito do mesmo e assim envia informações para o decodificador. Quando a aproximação é feita o sensor identifica o código do cartão de acesso e a envia para o controlador,

essa informação é conferida com por um código através do Arduino e com essa informação é possível liberar ou barra acesso, o conjunto é ilustrado na figura 4.5.

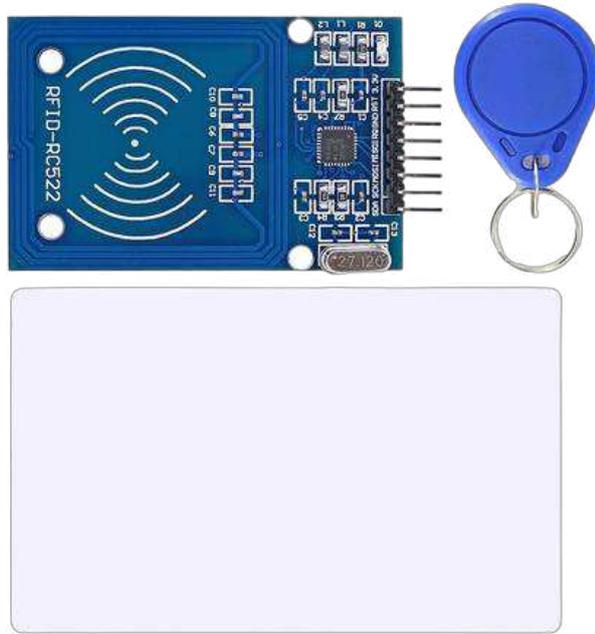


Figura 4.5 Conjunto RFID

(Disponível em:<https://http2.mlstatic.com/D_NQ_NP_921463-MLB44207295170_112020-O.webp>).

Este sensor utiliza o protocolo de comunicação serial SPI (*Serial Peripheral Interface*) descoberta pela Motorola em 1970. Essa é uma conexão full duplex, o que significa que os dados são enviados e recebidos simultaneamente. Em outras palavras, um mestre pode enviar dados ao escravo e um escravo pode enviar dados ao mestre simultaneamente. SPI é uma comunicação serial síncrona, o que significa que o relógio é necessário para fins de comunicação. Abaixo, na figura 4.6, está a representação do diagrama de blocos do SPI Master com Slave.

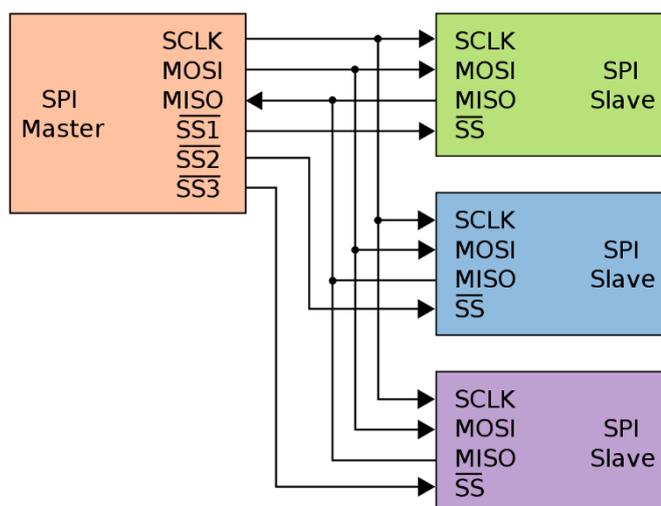


Figura 4.6 Circuito do relé

(Disponível em:<<https://autocorerobotica.blog.br/conhecendo-o-protocolo-spi-com-arduino/>>).

- MISO - A linha Slave para enviar dados ao mestre.
- MOSI - A linha Master para enviar dados para periféricos.
- SCK - Os pulsos de clock que sincronizam a transmissão dos dados gerados pelo mestre.
- SS - O mestre pode usar este pino para habilitar e desabilitar dispositivos específicos.

4.2.5 Sensor de nível de água

Os sensores de nível, que podem ser chamados também de “chave de nível” ou “boia de nível”, funcionam como um contato seco que pode abrir ou fechar, só que isso funciona através do movimento de um flutuador que realiza a mudança de estado. Eles têm a função de detectar o limite de nível dos líquidos em tanques ou reservatórios e enviar sinais de ON ou OFF como saída. São sensores de potência baixa e por isso não podem ser utilizados para acionar bombas de forma direta, necessitando de equipamentos para realizar as ações, um exemplo desse equipamento são os relés auxiliares ou de interface. (EICOS SENSORES, 2016)

Por ficarem em posições fixas e preso nas laterais dos reservatórios, isso faz com que eles não sejam afetados por ondulações ou vibrações, garantindo assim maior confiabilidade e repetibilidade se comparados as antigas “Chaves Boia”. (EICOS SENSORES, 2016)

O sensor de Nível é constituído por uma haste que desliza um cilindro de material flutuante, e no meio dessa haste existe um sensor magnético que através de um imã presente no cilindro, fecha o contato seco dos dois fios. Conforme podemos ver o sensor na Figura 4.3.



Figura 4.2 Sensor de Nível

(Disponível em: <https://http2.mlstatic.com/D_NQ_NP_658906-MLB43626927071_092020-O.webp>).

4.2.6 Sensor de luminosidade LDR

O sensor de luminosidade LDR é um equipamento que funciona variando a resistência de acordo com a intensidade da luz, funcionando basicamente como um fotoresistor, que quanto mais luz incidir no sensor, menor será a resistência elétrica. Para ligação do sensor direto no Arduino, se faz necessário instalá-lo com um resistor em conjunto, já que a corrente máxima que circula pelo sensor é de 20 mA. Conforme visto na Figura 11 temos a foto do sensor de luminosidade. (VIDAL, 2017)

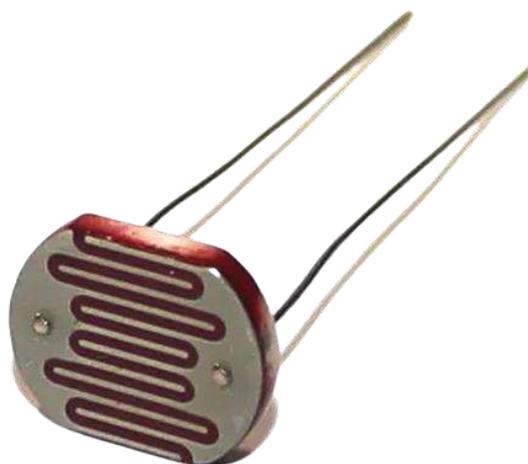


Figura 4.3 Sensor de Nível

(Disponível em: <https://http2.mlstatic.com/D_NQ_NP_925941-MLB44726204334_012021-O.webp>).

4.3 APP BLYNK

O Blynk foi desenvolvido para ser usado em lote de projetos, que é um termo utilizado para definir uma forma como objetos do mundo real permanecidos conectados em rede e podem ser acessados através da internet.

A principal característica do Blynk é permitir que uma comunicação com uma plataforma microcontrolada/embarcada possa ser realizada sabendo o mínimo de programação e até mesmo sem digitar uma linha de código. Ele permite que plataformas sejam controladas remotamente, de forma que dados de sensores e módulos possam ser obtidos e exibidos no aplicativo que fica instalado no dispositivo móvel ou que cargas sejam acionadas, além de muitas outras coisas.

O Blynk é composto basicamente por:

- **App Blynk:** possibilita criar interfaces de controle de forma simples, onde é necessário apenas arrastar os widgets e em poucos passos fazer a configuração.

- **Servidor Blynk:** é responsável por todas as comunicações entre o dispositivo móvel e a plataforma. Você pode usar o Blynk Cloud ou executar um servidor Blynk em sua máquina local. O servidor pode trabalhar com diversos dispositivos, inclusive pode rodar em um Raspberry Pi.

- **Bibliotecas Blynk:** há bibliotecas para todas as plataformas mais populares e compatíveis com o Blynk, permitindo a comunicação como servidor na nuvem ou local, processando todos os comandos de entrada e saída.

A ferramenta é parcialmente Open Source: o servidor e bibliotecas do Blynk Matter possuem código aberto, enquanto o aplicativo mobile é de código fechado. Na figura 4.8 é possível ver a arquitetura de funcionamento.

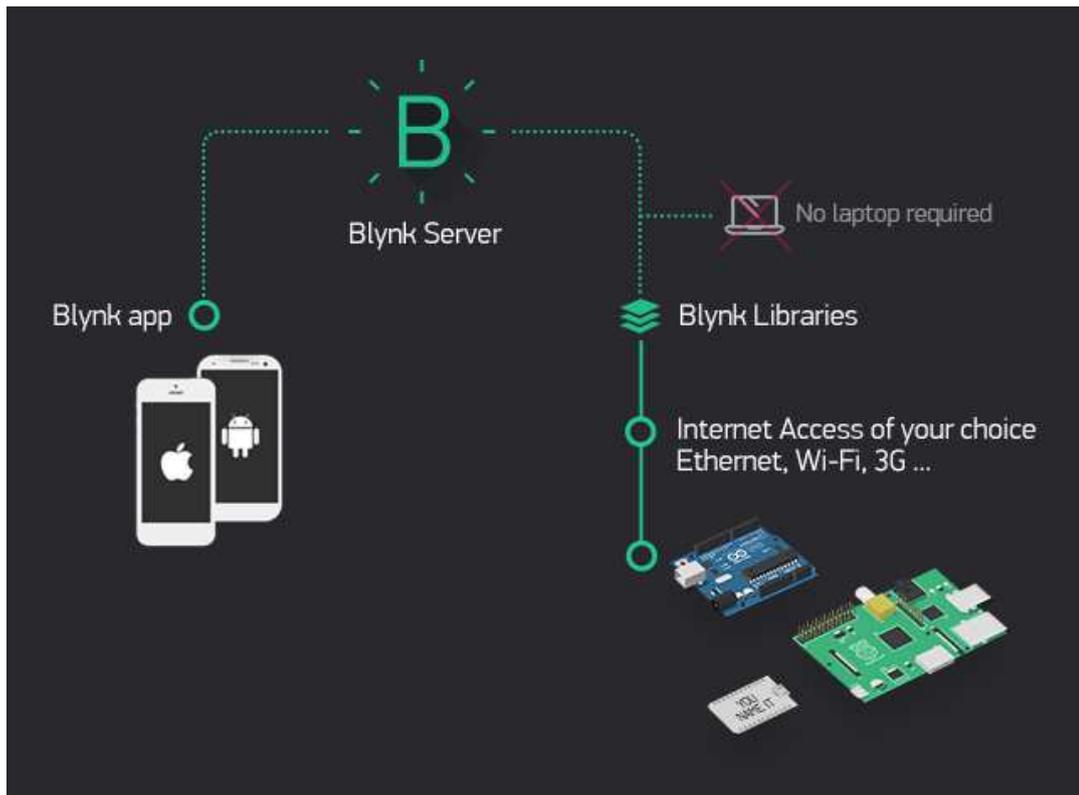


Figura 4.8 Sensor de Nível

(Disponível em: <<https://www.embarcados.com.br/introducao-ao-blynk-app/>>).

5 MATERIAIS E METODOS

De acordo com os objetivos traçados é estabelecido a metodologia de execução do projeto através das etapas a seguir:

- Instalação do aplicativo blynk e conexões
- Definição da residência e equipamentos automatizados
- Módulos para automatização

5.1 INSTALAÇÃO DO APP BLYNK E CONEXÃO

Para fazer a instalação do aplicativo no dispositivo é necessário acessar a loja de apps, Apple Store - em caso de IOS - ou Play Store – em caso de Android. Após o download e instalação do Blynk no dispositivo, será necessário a criação de uma conta de login para criar as credenciais de entrada no aplicativo. Com a conta criada, inicia-se um novo projeto e em seguida um token é gerado e enviado ao e-mail cadastrado, conforme mostra a figura 5.1.

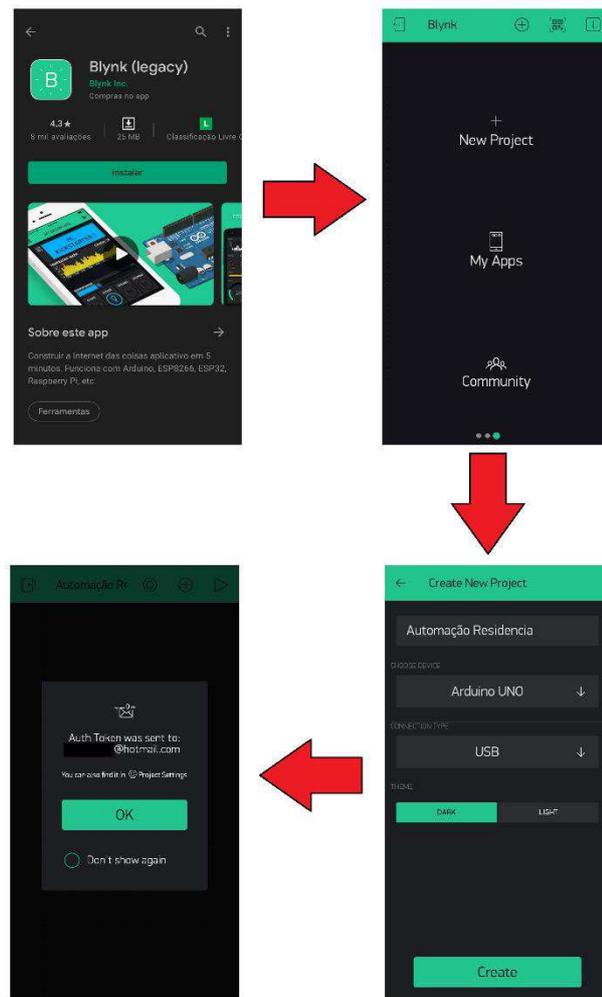


Figura 5.1 Criação do projeto blynk
(autor, 2021).

O Auth Token é uma chave de acesso alfanumérica que possibilita que o servidor Blynk consiga direcionar as informações entre o aplicativo instalado no dispositivo e a plataforma. Esta chave é única para cada projeto e não deve ser compartilhada.

A configuração inicial do Blynk com o Arduino é feita através da inclusão da biblioteca no código e utilizando o token enviado para o e-mail cadastrado através do aplicativo. A tabela 5.1 ilustra o código mencionado.

Tabela 5.1 Código de integração ao Blynk

Código de integração ao Blynk	
1	<code>#include <SoftwareSerial.h> //INCLUSÃO DE BIBLIOTECA</code>
2	<code>SoftwareSerial DebugSerial(2, 3); //OBJETO DO TIPO SoftwareSerial (2 = RX / 3 = TX)</code>
3	
4	<code>#define BLYNK_PRINT DebugSerial</code>
5	<code>#include <BlynkSimpleStream.h> //INCLUSÃO DE BIBLIOTECA</code>

```
6
7 char auth[] = "RrkwRHKhfH2KHJgQXX2nphw7gQbsDoVW"; // (FORNECIDO PELO BLYNK)
8
9 void setup(){
10  DebugSerial.begin(9600); //INICIALIZA O CONSOLE DE DEBUG
11
12  Serial.begin(9600); //INICIALIZA A SERIAL
13  Blynk.begin(Serial, auth); //INICIALIZA A COMUNICAÇÃO BLYNK INFORMANDO OS PARÂMETROS
14 }
15
16 void loop()
17 {
18  Blynk.run(); //INICIALIZA O BLYNK
19  Blynk.virtualWrite(V8, analogRead(A0));
20 }
```

Para fazer e manter a conexão do arduino com o app é necessário executar o Blynk Run USB script, dessa forma o computador faz a ponte de conexão via internet entre o smartphone e placa arduino. Com a conexão estabelecida, é feito um teste para constatar o pareamento.

No aplicativo, na tela do projeto automação residencial, inicia-se a tela de Widget Box para selecionar o botão do tipo on/off. Com o botão posicionado na tela, é dado um click para ajustar as configurações, então é selecionado o pin como digital na porta 7 do arduino e escolhendo o botão como switch. A figura 5.2 ilustra como é escolhido as configurações do botão para o teste inicial.

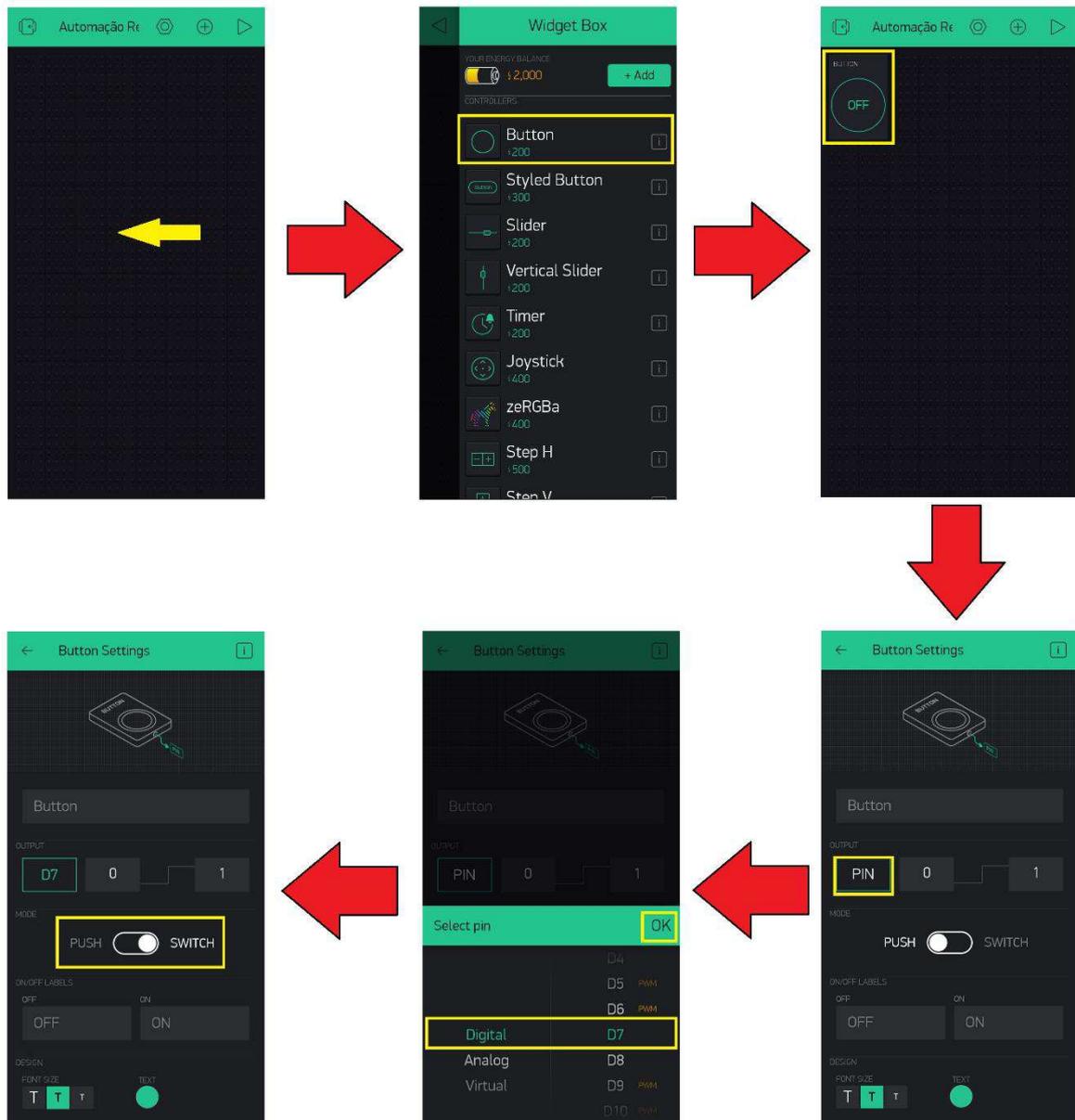


Figura 5.2 Configuração do teste

(autor, 2021).

Após a finalização das configurações mencionadas, será constatado a conexão entre a placa arduino e o smartphone - figura 5.3 - podendo, dessa forma, partir para os próximos componentes a serem utilizados no projeto de automação residencial.

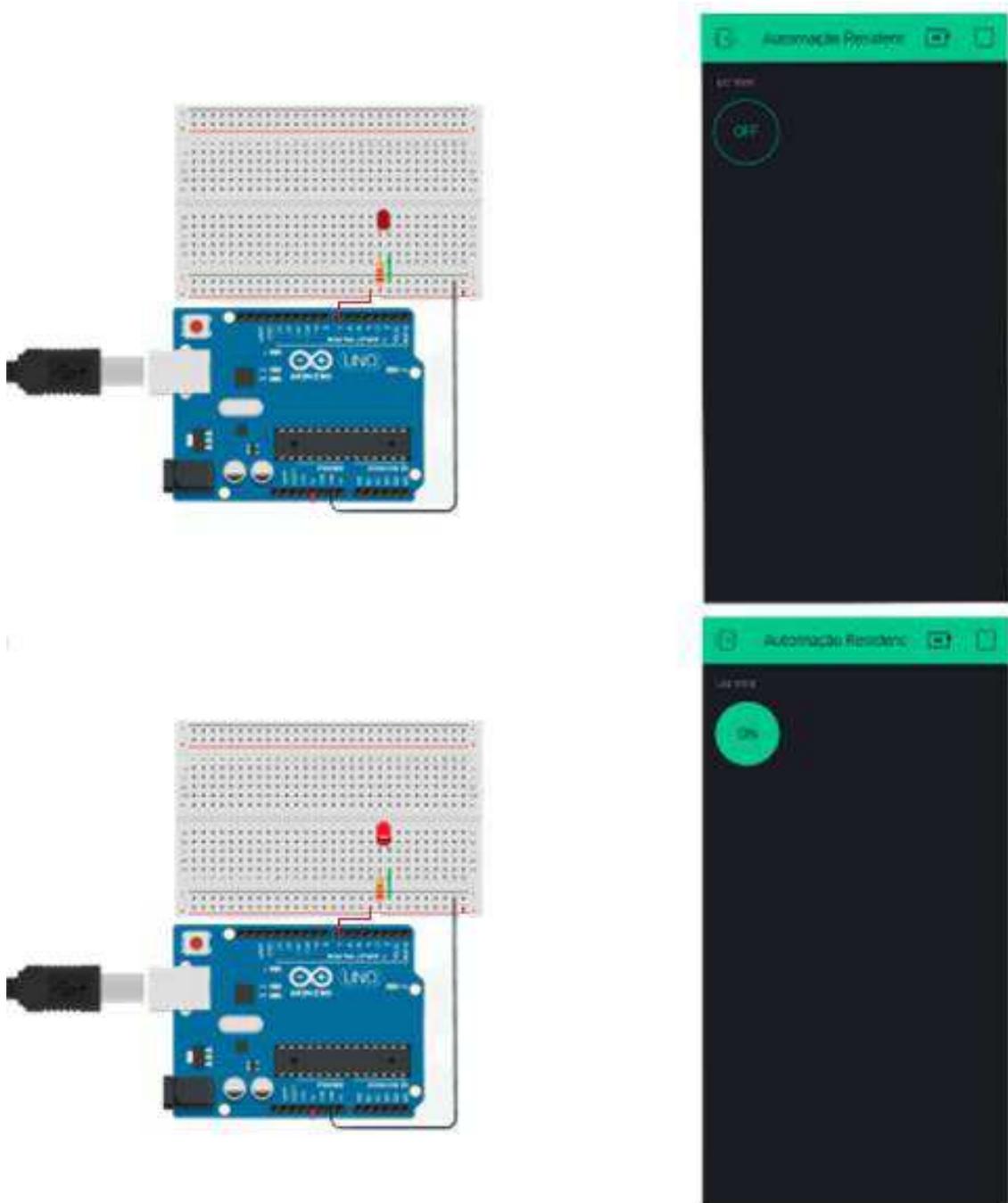


Figura 5.3 Circuito teste de conexão
(autor, 2021).

5.2 DEFINIÇÃO DA RESIDÊNCIA E EQUIPAMENTOS AUTOMATIZADOS

A partir do teste inicial de conexão é estabelecido a configuração básica de automação selecionando os dispositivos mais comuns utilizados no dia a dia de uma residência convencional, com esse intuito os componentes são divididos em:

- **Comodidade:** itens de menor necessidade como lâmpadas e ventiladores;
- **Economia:** são equipamentos que ficam diretamente ligados a tomada em modo de stand-by que possuem consumo relevante como tv's e microondas;
- **Segurança:** equipamentos de monitoramento como sensores de presença.

Para estabelecer os eletrodomésticos e equipamentos que seriam conectados, optou-se por uma residência de poucos cômodos – figura 5.4 - para simplificar a aplicação do projeto, tendo em vista que os mesmos mecanismos utilizados podem ser replicados para estruturas maiores, pois os princípios de funcionamento são os mesmos.

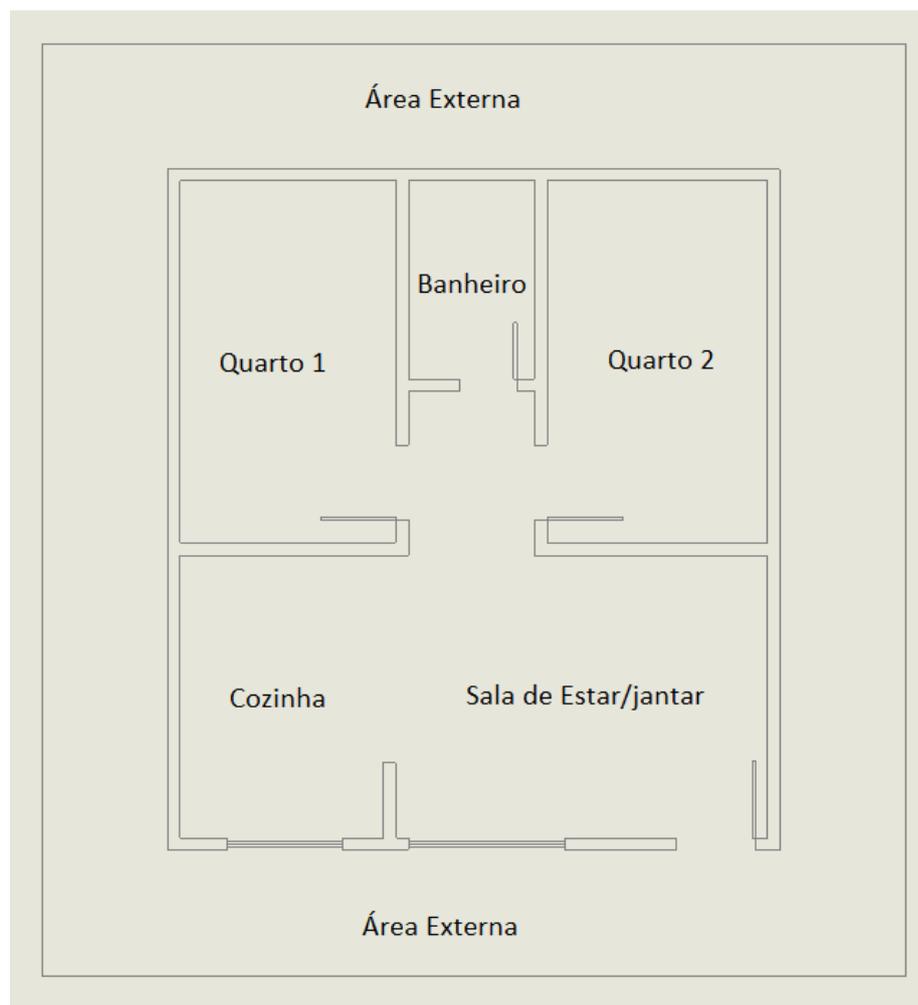


Figura 5.4 Planta da casa usada no projeto

(Autor, 2021)

Com base na 5.4 os componentes de comodidade a serem conectados são:

- Luz cozinha;
- Luz área de serviço;
- Luz sala de jantar/estar;
- Acionamento da caixa d'água;
- Luz quarto casal;
- Luz quarto da criança;

- Luz externa

Pra os equipamentos da categoria economia são selecionados alguns item como:

- Televisão;
- Microondas;
- Ar-condicionado;

E para os equipamentos para segurança será utiizado:

- Alarme;
- Fuchadura.

Em resumo, os itens que serão automatizados são mostrados na tabela 5.2 nela são apresentados os equipamentos e módulos utilizados.

Tabela 5.2 Lista de equipamentos e módulos

TIPO	EQUIPAMENTOS	MODULO
Comodidade	luz do quarto 1	relé
	luz do quarto 2	relé
	luz do banheiro	relé
	luz da cozinha	relé
	luz da sala de jantar/estar	relé
Economia	Microondas	relé
	Televisão	relé
	Ar condicionado 1	relé
	Ar condicionado 2	relé
Segurança	Alarme	relé + PIR
	fechadura eletrônica	relé + RFID
	luzes externas	relé + LDR

5.3 MÓDULOS PARA AUTOMATIZAÇÃO

As ligações de equipamentos do tipo liga/desliga como ventiladores, lâmpadas e ar-condicionados, podem ser controladas através do módulo relé que é conectado de acordo com o diagrama esquemático mostrado na figura 5.5.

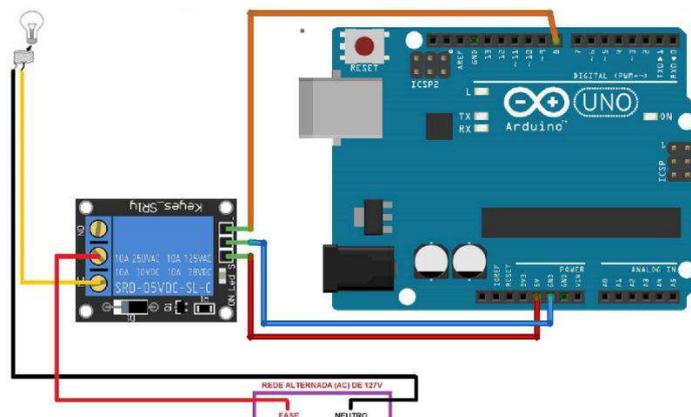


Figura 5. 5 Ligação do circuito do relé

(autor, 2021)

A conexão do relé com o arduino é simples, de forma que nos terminais de conexão com a placa é necessário apenas a alimentação de 5 volts com o retorno e um pino de saída digital para fazer o acionamento do módulo. Já na parte de conexão com o equipamento, o mdulo é alimentado com 110/220 volts no comum e a lâmpada conectada no pino normalmente aberto (NA) do módulo.

Cada relé pode ser responsável pelo acionamneto de um ou um conjunto, de acordo com a quantidade de lâmpadas e equipamentos que serão automatizados. Para este projeto optou-se pelo modulo rele de 16 canais ilustrado na figura 5.6.

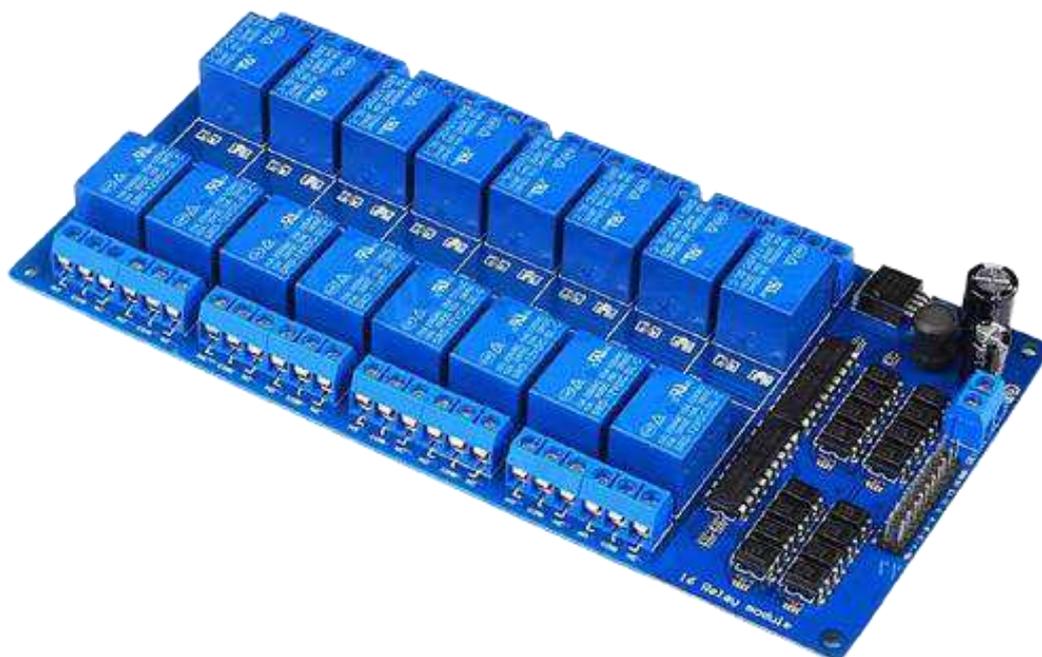


Figura 5.6 Módulo relé 16 canais

(Disponível em:

<https://ae01.alicdn.com/kf/H8d45e7c9cd1642d6a897bade49cdfbccc/M-dulo-de-escudo-de-rel-16-canais-com-optoacoplador-lm2576-m-dulo-de-prote-o.jpg_Q90.jpg_.webp>).

As conexões são feitas nos seguintes equipamentos:

- Quarto 1 e 2;
- Banheiro;
- Cozinha;
- Área de serviço;
- Sala de jantar/estar;
- Ar-condicionado 1 e 2;
- Televisão;
- Microondas;
- Alarme;
- Luzes externas;

- Fechadura.

Para o acionamento do alarme será utilizado o sensor de presença hc-sr501, que receberá o sinal de presença indevida e iniciará o alarme através do módulo relé.

5.4 TESTE DE ACIONAMENTO

A fim de validar a simulação na seção 5.1 e testar a integração proposta neste trabalho é criado um circuito teste. A figura 5.7 retrata o teste de acionamento do LED através de um botão via internet.

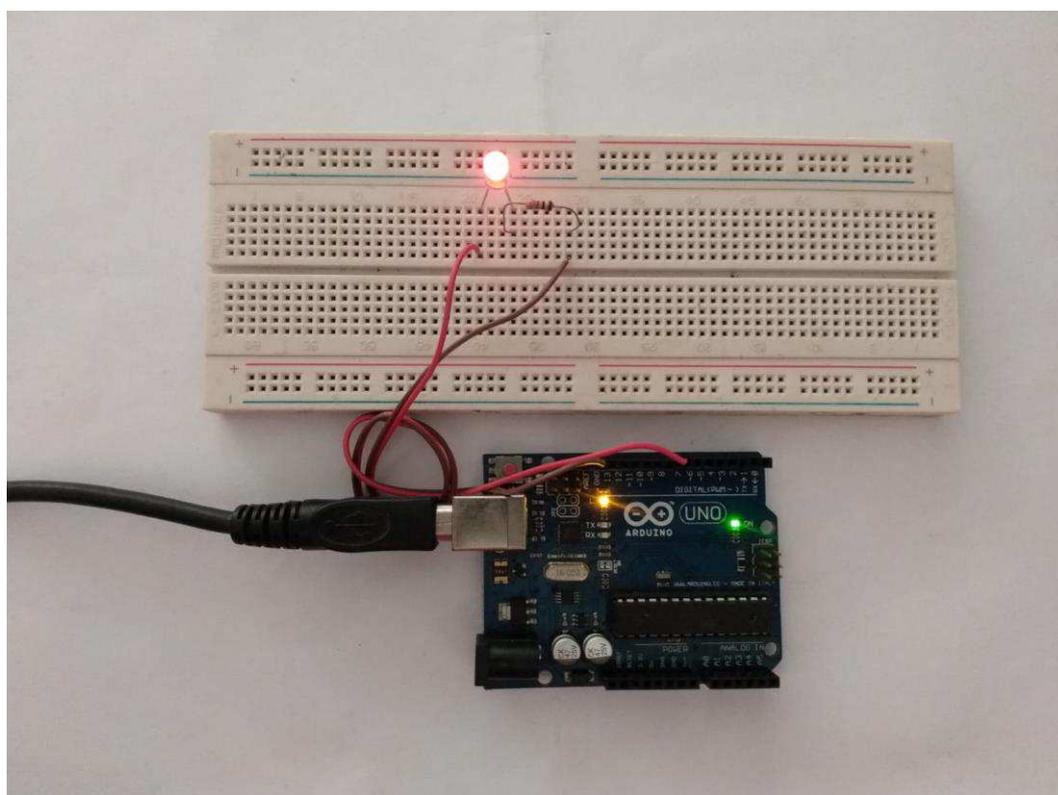


Figura 5.7 Acionamento do LED via internet

Para confirmar o funcionamento de equipamentos conectados via relé é realizado o teste de acionamento de uma lâmpada através do LDR configurado via internet conforme a figura 5.8 e figura 5.9.

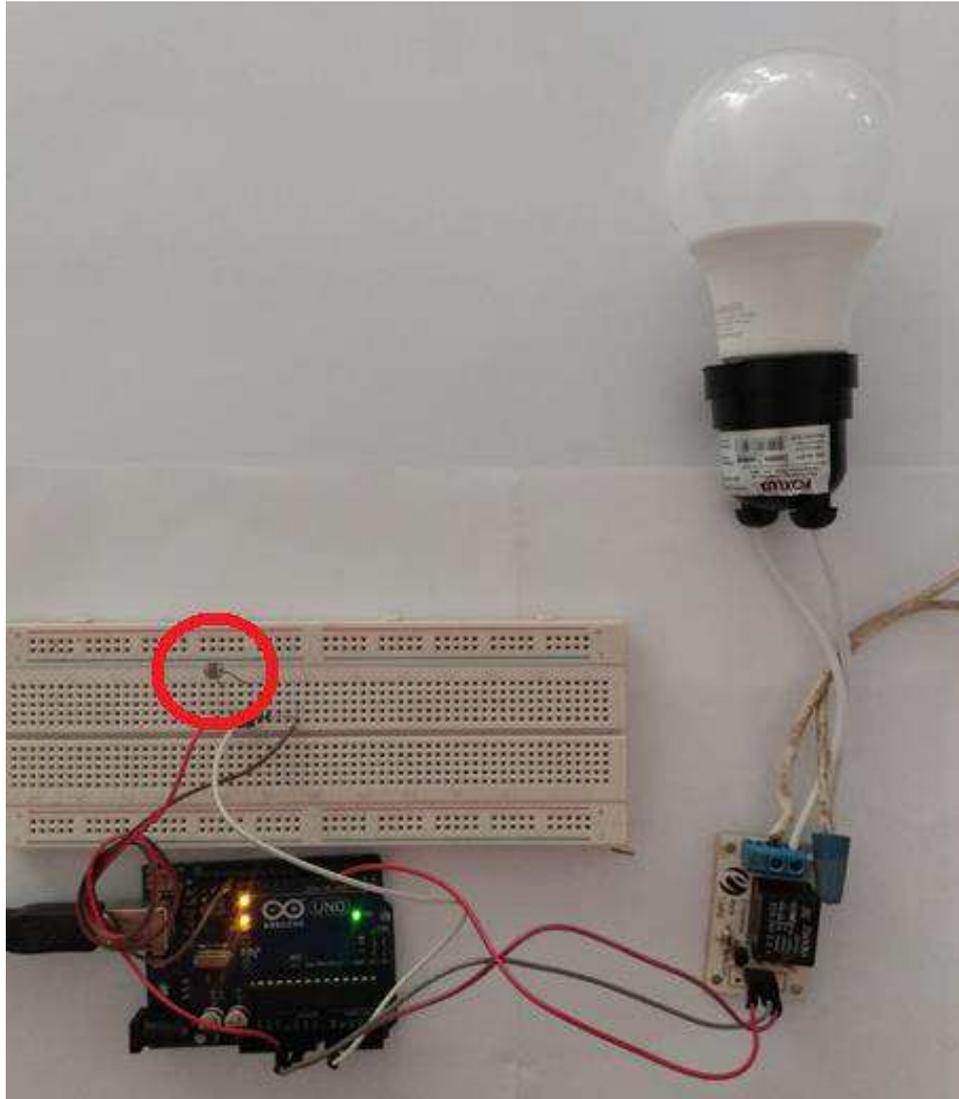


Figura 5.8 Acionamento por relé via internet
(autor, 2021)

Através dos testes iniciais é possível constatar a funcionalidade do aplicativo no que se refere a conexão e acionamentos dos equipamentos sugeridos na tabela 5.2. Na figura 5.9 o circuito está acionado com um tapper cobrindo o sensor para simular a noite.

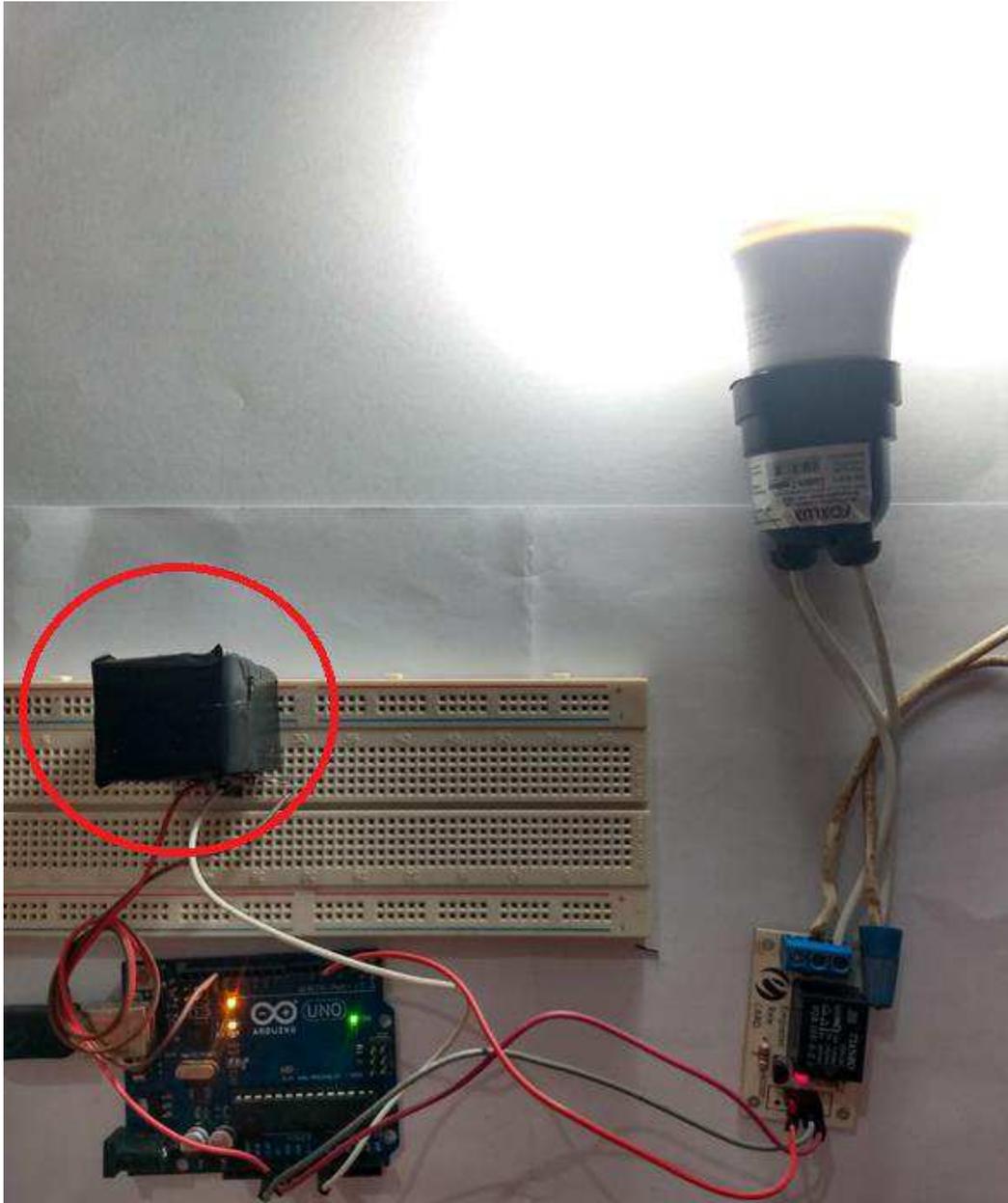


Figura 5.9 Circuito do LDR acionado
(autor, 2021)

6 RESULTADOS E DISCUSSÕES

6.1 CÓDIGO PURO EM ARDUINO PARA ACIONAMENTO DOS MÓDULOS

Esta seção mostra os códigos em Arduino para o acionamento e utilização de cada módulo individualmente. Os códigos são divididos em duas seções, void setup e void loop. O Setup é obrigatório em um programa Arduino, qualquer código que estiver inserido dentro desta seção será executado uma única vez no início do programa configurando, assim, todos os parâmetros iniciais necessários.

Após a execução do setup se inicia o loop, onde a maior parte do código será executado. A função loop será executada repetidas vezes lendo cada linha de código e reproduzidos os comandos inseridos, é nesta seção que deve ser feita toda a lógica de programação e acionamento dos módulos.

6.1.1 Acionamento do relé

Através do modulo relé o Arduino é capaz de controlar equipamentos com cargas maiores que a fornecida pela placa. Dessa forma a conexão do relé é feita com descrita no item 4.2.1 o código utilizado para o acionamento desse modo está exemplificado na tabela 6.1

Tabela 6.1 Código de acionamento do relé

Acionamento do relé
1 //CÓDIGO BASE PARA ACINAMENTO DE UM RELÉ
2 //Porta ligada ao pino IN1 do modulo
3
4 int porta_rele1 = 7;
5 void setup()
6 {
7 //Define pinos para o rele como saida
8 pinMode(porta_rele1, OUTPUT);
9 }
10 void loop()

```

11 {
12   digitalWrite(porta_rele1, LOW); //Liga rele
13   delay(2000);
14   digitalWrite(porta_rele1, HIGH); //Desliga rele
15   delay(2000);
16 }

```

Na linha 4 da tabela 6.1 é definido uma variável para controle da porta 7 da placa. No setup é configurado o modo de como o pino será utilizado, sendo configurado como saída de dados. Já no loop será configurado o que acontecerá na porta definida na linha 4. Inicialmente o relé estará desativado por estar em modo LOW – linha 12 – por 2000 milissegundos (linha 13) e então será acionado por estar em modo HIGH – linha 14 – por 2000 milissegundos (linha 15).

6.1.2 Sensor de luz com LDR

O LDR (Light Dependent Resistor) é um resistor que varia a resistência de acordo com o grau de luminosidade do ambiente, a tabela 6.2 ilustra o código base utilizado para a leitura de dados com este sensor no projeto.

Tabela 6.2 Código do sensor de luz com LDR

Código do sensor de luz com LDR	
1	//Sensor de luz com LDR
2	
3	int ledPin = 7; //Led no pino 7
4	int ldrPin = 0; //LDR no pino analógico 8
5	int ldrValor = 0; //Valor lido do LDR
6	
7	void setup() {
8	pinMode(ledPin,OUTPUT); //define a porta 7 como saída
9	Serial.begin(9600); //Inicia a comunicação serial
10	}
11	
12	void loop() {
13	///ler o valor do LDR
14	ldrValor = analogRead(ldrPin); //O valor lido será entre 0 e 1023
15	
16	//se o valor lido for maior que 600, liga o led

```

17  if (ldrValor >= 600) digitalWrite(ledPin, HIGH);
18  // senão, apaga o led
19  else digitalWrite(ledPin, LOW);
20
21  //imprime o valor lido do LDR no monitor serial
22  Serial.println(ldrValor);
23  delay(100);
24  }

```

O exemplo da tabela 6.2 configura os pinos que serão utilizados no programa. No setup, na linha 8, é definido o modo de trabalho do pino como saída e em seguida é feita a comunicação serial para exibir os dados lidos do LDR. Na linha 14 é feita a leitura do valor analógico fornecido pelo sensor e definido uma condição de acionamento para valores do led maiores ou igual a 500.

6.1.3 Sensor de nível

O sensor de nível é um equipamento do tipo on/off que será utilizado para enviar dados do estado da caixa d'água. Neste projeto será utilizado dois sensores para indicar se a caixa está cheia ou vazia e então acionar a bomba. O sketch base de acionamento está ilustrado na tabela 6.3.

Tabela 6.3 Código do sensor de nível

Código do sensor de nível	
1	<code>#define Sensor1 7</code>
2	<code>#define Sensor2 8</code>
3	<code>#define rele 13</code>
4	
5	<code>int sensor1 = 1, sensor2 = 1;</code>
6	
7	<code>void setup() {</code>
8	<code> Serial.begin(9600);</code>
9	
10	<code> pinMode(Sensor1, INPUT);</code>
11	<code> pinMode(Sensor2, INPUT);</code>
12	<code> pinMode(rele, OUTPUT);</code>
13	
14	<code> Serial.println("Nivel do Reservatorio");</code>

```
15 Serial.println();}
16 void loop() {
17   int sensor1 = digitalRead(Sensor1);
18   if ((sensor1 == 1) && (sensor2 == 1) {
19     Serial.println("Reservatorio Cheio");
20     digitalWrite(rele, LOW); }
21   else if ((sensor1 == 0) && (sensor2 == 0) {
22     Serial.println("Reservatorio Vazio ");
23     digitalWrite(rele, HIGH); }
24   else {
25     digitalWrite(rele, LOW); }
26   delay(1000); }
```

Inicialmente é definido as portas que serão utilizadas no projeto e no setup são definidas as configurações das portas utilizadas. No loop se inicia a leitura dos sensores e dependendo dos seus respectivos valores é definida a ação a ser tomada. Se os dos sensores acusarem nível lógico alto a placa entende que o reservatório de água está completamente cheio e então desliga o relé. Caso os sensores estejam com nível logico baixo, a caixa apresenta baixo nível no reservatório e dessa forma aciona o relé para acionar a bomba. Com os sensores apresentando qualquer outro estado a bomba permanece desligada.

6.1.4 Sensor de presença

O sensor de presença pode ser utilizado como um dispositivo de segurança para acionar alarmes. O esquema de ligação é apresentado na figura 6.1.

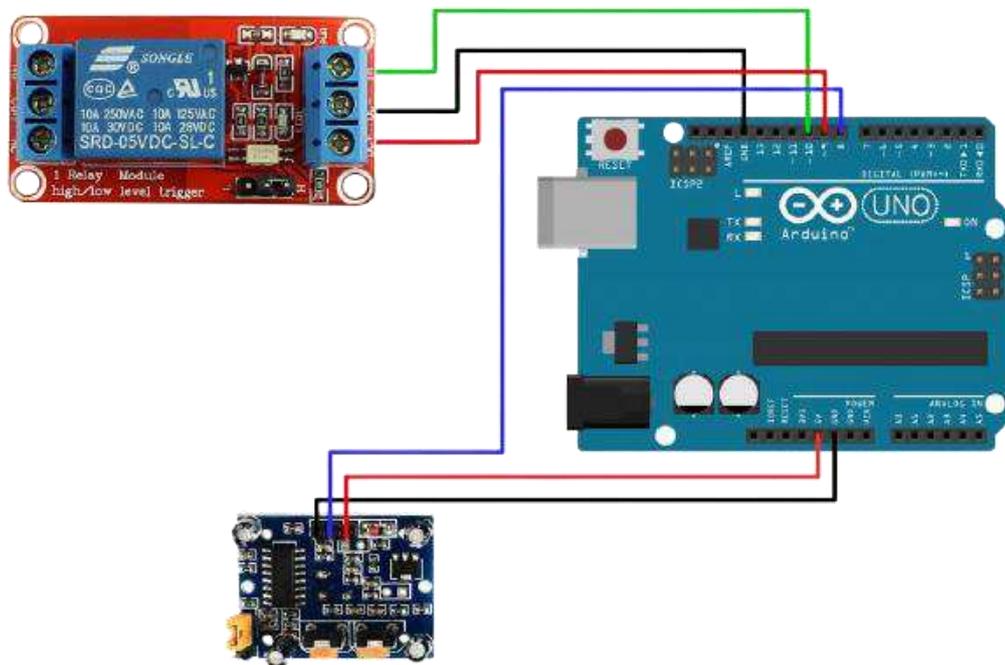


Figura 6.1 Módulo relé integrado ao sensor PIR
(Disponível em: <<https://www.usinainfo.com.br/blog/wp-content/uploads/2019/08/sensor-pir-arduino-rele-1024x683.jpg>>).

Para ser utilizado como alarme o módulo PIR deve ser integrado a uma sirene juntamente com um relé para fazer o acionamento do mesmo. A figura 6.2 mostra a ligação entre eles.

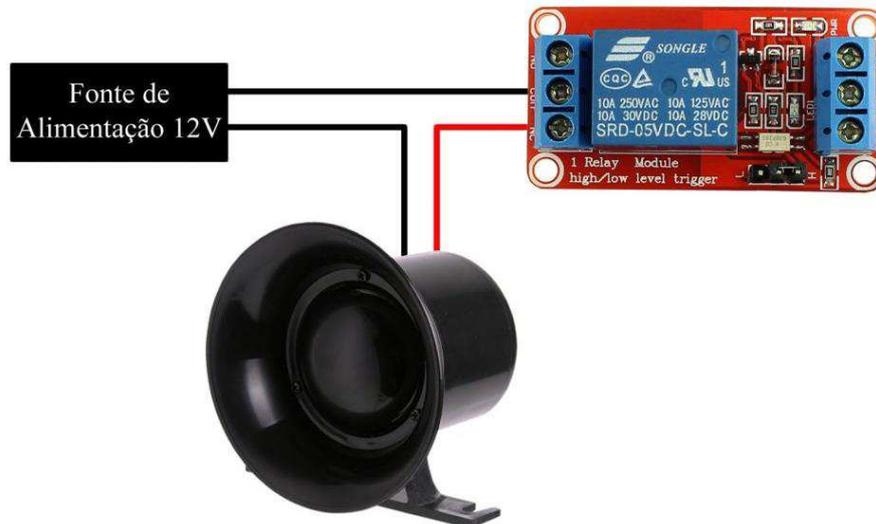


Figura 6.2 Conexão do relé com a sirene
(Disponível em: <<https://www.usinainfo.com.br/blog/wp-content/uploads/2019/08/rele-sirene-alarme-pir-1024x681.jpg>>).

O código utilizado para controle a integração é mostrado na tabela 6.4

Tabela 6.4 Código do sensor de presença com sirene

Código do sensor de presença com sirene	
1	<code>// Projeto de Alarme / Automação Residencial</code>
2	<code>#define pinoPIR 8 // Define o pino 8 como "pinoPIR"</code>
3	<code>#define pino5V 9 // Define o pino 9 como "pino5V"</code>
4	<code>#define pinoRele 10 // Define o pino 10 como "pinoRele"</code>
5	<code>void setup() {</code>
6	
7	<code>Serial.begin(9600); // Declara o BaundRate em 9600</code>
8	<code>pinMode(pinoPIR, INPUT); // Declara o pinoPIR como Entrada</code>
9	<code>pinMode(pino5V, OUTPUT); // Declara o pino5V como Saída</code>
10	<code>pinMode(pinoRele, OUTPUT); // Declara o pinoRele como Saída</code>
11	<code>digitalWrite(pino5V, HIGH); // Põem o pino5V HIGH</code>
12	<code>}</code>
13	<code>void loop() {</code>
14	<code>if (digitalRead(pinoPIR) == LOW) {</code>
15	<code>digitalWrite (pinoRele, LOW); }</code>
16	<code>if (digitalRead(pinoPIR) == HIGH) {</code>
17	<code>digitalWrite (pinoRele, HIGH); // Liga o Relé</code>
18	<code>Serial.println("Alarme Acionado");</code>
19	<code>delay (10000);}}</code>

Da linha 2 a 4 da tabela são definidas as portas a serem utilizadas no projeto. No setup são definidas as configurações de cada pino a ser usado, de modo a ser INPUT, OUTPUT ou HIGH. Já no loop a lógica de programação estabelece as ações de acordo com a resposta do sensor, onde se o estado de leitura for nível lógico baixo o relé permanece desligado e caso o PIR retorne nível lógico alto o relé é acionado e aciona a sirene.

6.1.5 Fechadura eletrônica integrada ao RFID

Nessa integração o RFID é responsável pela leitura e autorização da chave de acesso para o acionamento da fechadura eletrônica através de um módulo relé. A pinagem de conexão entre o Arduino e o sensor é apresentada na figura 6.3 e diagrama esquemático da ligação é ilustrada na tabela 6.5

Tabela 6.5 Pinagem do RFID

Pinagem RFID	
Módulo RFID RC522	Arduino
3.3 v	Pino 3.3v
RST	Pino 9
GND	Pino GND
NC ou IRQ	Não conectado
MISO	pino 12
MOSI	pino 11
SCK	pino 13
DAS	pino 10

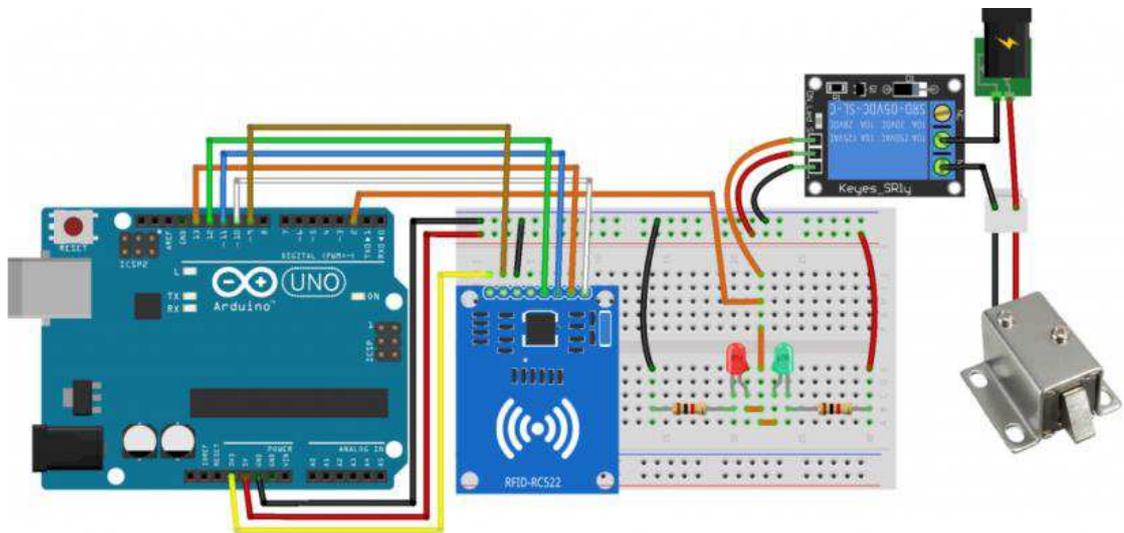


Figura 6.3 Integração do RFID com a trava elétrica
(Disponível em: <<https://blog.fazedores.com/fechadura-eletronica-com-arduino-e-rfid/>>).

O código base com a lógica de programação para integração dos módulos é apresentado na tabela 6.6

Tabela 6.6 Código da fechadura eletrônica integrada ao RFID

Código da fechadura eletrônica integrada ao RFID	
1	//Programa: Sistema de trava eletrica com RFID
2	<code>#include <SPI.h></code>
3	<code>#include <MFRC522.h></code>
4	<code>#define SS_PIN 10</code>
5	<code>#define RST_PIN 9</code>
6	<code>MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.</code>
7	
8	<code>void setup()</code>
9	{

```

10 Serial.begin(9600); // Inicia a serial
11 SPI.begin(); // Inicia SPI bus
12 mfrc522.PCD_Init(); // Inicia MFRC522
13 Serial.println("Aproxime o seu cartao do leitor...");
14 Serial.println();
15 pinMode(2, OUTPUT);
16 }
17 void loop()
18 { // Procura por cartao RFID
19   if ( ! mfrc522.PICC_IsNewCardPresent())
20   {
21     return;
22   }
23   // Seleciona o cartao RFID
24   if ( ! mfrc522.PICC_ReadCardSerial()){
25     return; }
26   //Mostra UID na serial
27   Serial.print("UID da tag :");
28   String conteudo= "";
29   byte letra;
30   for (byte i = 0; i < mfrc522.uid.size; i++)
31   {
32     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
33     Serial.print(mfrc522.uid.uidByte[i], HEX);
34     conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
35     conteudo.concat(String(mfrc522.uid.uidByte[i], HEX)); }
36   Serial.println();
37   Serial.print("Mensagem : ");
38   conteudo.toUpperCase();
39   if (conteudo.substring(1) == "F3 DE 90 55") //UID 1 - Cartao
40   {
41     Serial.println();
42     digitalWrite(2, HIGH); // ativa rele, abre a trava solenoide
43     delay(3000); // espera 3 segundos
44     digitalWrite(2, LOW); // desativa rele
45   }
46 }

```

Inicialmente é feita a importação da biblioteca SPI.h e MFRC522.h para a utilização da comunicação do sensor e Arduino. Na linha 7 é feita a instanciação do objeto e no setup as configurações básicas das bibliotecas e pinos a serem utilizados. No loop é feita a leitura do cartão de acesso e verificado se o UID da TAG é conhecido, caso seja identificado o acesso é liberado acionando o módulo relé para destravar a fechadura.

6.2 INTEGRAÇÃO DOS MÓDULOS AO BLINK

Para fazer a integração dos módulos ao blynk é necessário apenas carregar o código da tabela 6.7 e acrescentar uma linha de comando para cada módulo incrementado com laço condicional. O código de exemplo é mostrado na tabela 6.7.

Tabela 6.7 Código para adicionar pino virtual ao Blynk

Código para adicionar pino virtual ao Blynk	
1	<code>#include <SoftwareSerial.h> //INCLUSÃO DE BIBLIOTECA</code>
2	<code>SoftwareSerial DebugSerial(2, 3); //OBJETO DO TIPO SoftwareSerial (2 = RX / 3 = TX)</code>
3	
4	<code>#define BLYNK_PRINT DebugSerial</code>
5	<code>#include <BlynkSimpleStream.h> //INCLUSÃO DE BIBLIOTECA</code>
6	
7	<code>char auth[] = "RrkwRHKhfH2KHJgQXX2nphw7gQbsDoVW"; //AUTH TOKEN</code>
8	
9	<code>void setup(){</code>
10	<code> DebugSerial.begin(9600); //INICIALIZA O CONSOLE DE DEBUG</code>
11	
12	<code> Serial.begin(9600); //INICIALIZA A SERIAL</code>
13	<code> Blynk.begin(Serial, auth); //INICIALIZA A COMUNICAÇÃO BLYNK</code>
14	<code>}</code>
15	
16	<code>void loop()</code>
17	<code>{</code>
18	<code> Blynk.run(); //INICIALIZA O BLYNK</code>
19	<code> Blynk.virtualWrite(V8, analogRead(A0));</code>
20	<code>}</code>

Na linha 19 da tabela é adicionado “Blynk.virtualWrite” para acessar a porta virtual criada pelo app, por ela pode-se incrementar condições de acionamento e controle dos módulos pelo app. Nessa linha de código são definidos dois parâmetros, V8 (a porta que o blynk deve utilizar) e AnalogRead(A0) – o parâmetro que deve ser inserido como valor no pino digital. A figura 6.4 apresenta o passo a passo de como utilizar o LDR com sensor de luminosidade e a configuração para acionar o relé de acordo com as configurações predefinidas.

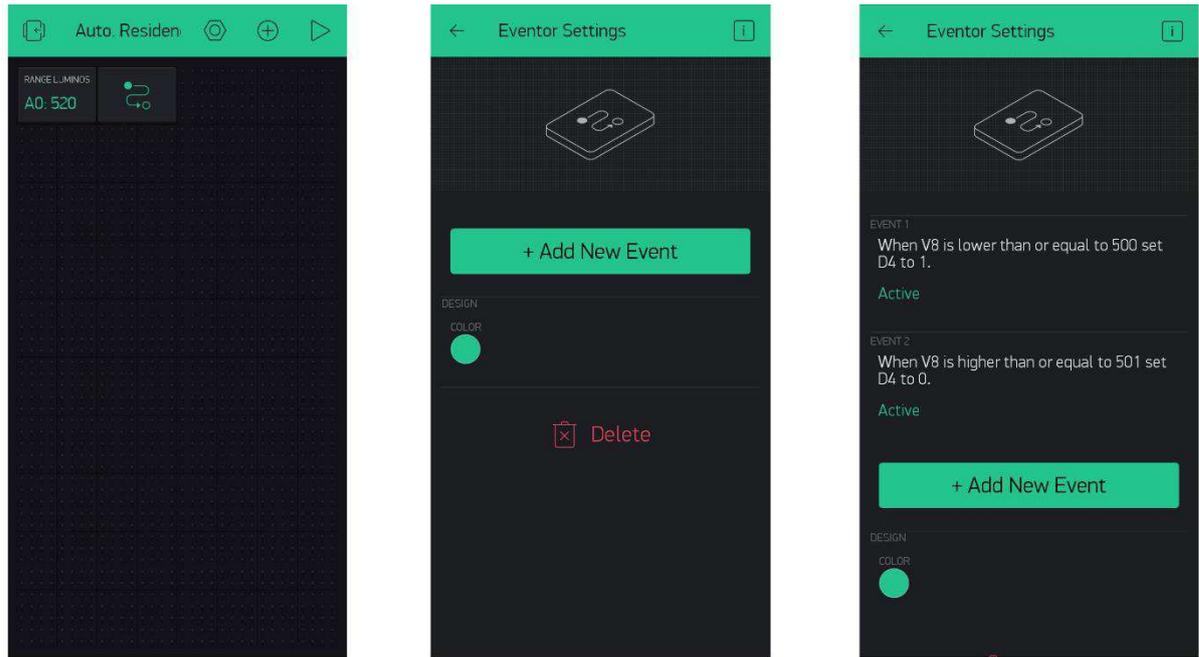


Figura 6.4 Adição do eventor no blynk

(autor, 2021)

Para visualizar os dados fornecidos pelo sensor é necessário adicionar o Value Display ao aplicativo e para fazer o laço condicional é adicionado o Eventor, como mostra a figura 6.4. O display é configurado para ler a porta A0 da placa como estabelecido no sketch. O evento faz a leitura a partir de uma porta virtual, por esse motivo é necessário definir a escrita do pino V8 como sendo o valor lido pela porta analógica A0 e a partir desses valores é feita a tomada de decisão condicional. Para valores de V8 menor ou igual a 500 a porta D4 permanece em nível lógico alto e para valores maior ou igual que 501 a porta D4 permanece em nível lógico baixo. Esse código pode ser replicado para os outros sensores como o sensor de nível d'água e para o PIR.

6.3 ORÇAMENTO DO PROJETO

Para o orçamento foi definidos dois sites base para a compra dos equipamentos necessários para a automação. A tabela 6.8 mostra o orçamento feito pelo site Aliexpress com seus respectivos custos de envio para o Brasil.

Tabela 6.8 Orçamento com base no site Aliexpress

PRODUTO	QUANTIDADE	CUSTO	FRETE	TOTAL DO PRODUTO
Modulo rele 16 canis	1	R\$ 27,97	R\$ 45,58	R\$ 73,55
Sensor PIR	1	R\$ 3,91	R\$ 22,93	R\$ 26,84
Fechadura eletrônica	1	R\$ 13,25	R\$ 26,10	R\$ 39,35
Sensor RFID	1	R\$ 8,49	R\$ 27,69	R\$ 36,18
Sensor de nível de água	2	R\$ 7,19	R\$ 27,69	R\$ 42,07
Sensor LDR	1	R\$ 5,00	R\$ -	R\$ 5,00
Arduino mega	1	R\$ 51,53	R\$ 24,18	R\$ 75,71
TOTAL			R\$	298,70

A tabela 6.9 demonstra o orçamento realizado em um site de revenda nacional. Os custos por cada módulo e suas respectivas taxas de entregas são apresentados na tabela.

Tabela 6.9 Orçamento com base no site MercadoLivre

PRODUTO	QUANTIDADE	CUSTO	FRETE	TOTAL DO PRODUTO
Modulo rele 16 canis	1	R\$ 105,27	R\$ 45,58	R\$ 150,85
Sensor PIR	1	R\$ 17,57	R\$ 40,80	R\$ 58,37
Fechadura eletrônica	1	R\$ 49,90	R\$ 40,80	R\$ 90,70
Sensor RFID	1	R\$ 19,99	R\$ 40,80	R\$ 60,79
Sensor de nível de água	2	R\$ 21,87	R\$ 27,69	R\$ 71,43
Sensor LDR	1	R\$ 5,00	R\$ -	R\$ 5,00
Arduino mega	1	R\$ 140,00	R\$ -	R\$ 140,00
TOTAL			R\$	577,14

7 CONCLUSÕES

Com a integração do Blynk ao Arduino, verificou-se a facilidade de programação uma vez que com apenas o código da tabela 5.1 pode-se estabelecer uma variedade de comandos pelo app sem a necessidade de adicionar códigos específicos para cada módulo ao conjunto de acionamento.

A utilização da placa de prototipagem se torna importante para o barateamento do projeto, já que os equipamentos que se integram ao Arduino para a automação residencial possuem preços mais acessíveis se comparados aos sistemas convencionais de automação. Além disso, o usuário possui a liberdade de associar os equipamentos que desejar ao sistema sem a necessidade de investir em um novo sistema.

Em relação à acessibilidade do projeto, pode-se observar que adquirindo os módulos em sites internacionais gera uma redução de custo de aproximadamente 48% o que viabiliza a implantação do mesmo, já que de acordo com a *NRDC - Natural Resources Defense Council* – cerca de 23% da energia consumida em uma residência é decorrente de aparelhos que ficam conectados diretamente à rede em modo stand-by. Dessa forma a implantação do sistema autônomo é justificada pela economia gerada no consumo de energia elétrica.

O projeto realizado neste trabalho de conclusão de curso, apresenta não apenas um sistema de baixo custo, mas também, uma alternativa simplificada de automação que pode viabilizar projetos complexos com alto custo agregado decorrente de sensores ou microcontroladores com valores expressivos. Para projetos mais robustos, a família dos Arduinos apresenta diversos modelos de placas que podem atender a demanda.

REFERÊNCIAS

BASCONCELLO FILHO, Daniel. **O Hardware do Arduino**. Disponível em <http://www.robotizando.com.br/curso_arduino_hardware_pg1.php>. Acesso em: 23 de julho de 2021.

CAMPOS, R. A. F. **Automação Residencial Utilizando Arduino e Aplicação Web**, Brasília, junho 2014. Disponível em:

<http://repositorio.uniceub.br/bitstream/235/5461/1/Monografia_Roberto.pdf>. MARTINS REBESCHINI, Sauro. **SISTEMA DE SEGURANÇA POR CÂMERAS E SENSORES CONTROLADOS POR DISPOSITIVO REMOTO**. 2012. 58p. Trabalho de Conclusão de Curso Fundação Educacional do Município de Assis – FEMA/Instituto Municipal de Ensino Superior de Assis - IMESA, São Paulo, Assis, 2012. PIR – HC.

MCROBERTS, Michael. **Arduino Básico**. 1. ed, São Paulo. Novatec, 2011. PALMER, Brian. **Evite que seus dispositivos desperdicem energia e dinheiro**. Disponível em: <https://www.nrdc.org/stories/keep-your-devices-wasting-energy-and-money>. Acesso em: 09 de outubro de 2021.

DELFORGE, Pierre. **Carga ociosa doméstica: dispositivos que desperdiçam grandes quantidades de eletricidade quando não estão ativos**. Disponível em: <https://www.nrdc.org/resources/home-idle-load-devices-wasting-huge-amounts-electricity-when-not-active-use>. Acesso em 09 de outubro de 2021.

AMATO, Fabio. **Níveis de reservatórios de Sudeste e Centro-Oeste em maio (...)**. Disponível em: <https://g1.globo.com/economia/noticia/2021/06/02/nivel-dos-reservatorios-de-sudeste-e-centro-oeste-em-maio-e-o-mais-baixo-para-o-mes-desde-2001.ghtml>. Acesso em 10 de outubro de 2021.

MURATORI, José Roberto. BÓ, Paulo Henrique. **Automação residencial: (...)** disponível em http://www.osetoreletrico.com.br/wp-content/uploads/2011/04/Ed62_fasc_automacao_capI.pdf. Acesso em 09 de outubro de 2021.

MÓDULO RELÉ PARA AUTOMOÇÃO RESIDENCIAL COM ARDUINO. Disponível em: <https://blog.eletrogate.com/modulo-rele-para-automacao-residencial-com-arduino/>. Acesso em 11 de outubro de 2021.

MÓDULO. **Detector de presença com sensor PIR HC-SR501**. Disponível em: <http://blog.moduloeletronica.com.br/detector-de-presenca-com-sensor-pir-hc-sr501/>. Acesso em 11 de outubro de 2021.

MODELO DE PLANTA RESIDENCIAL. Disponível em: <https://www.pinterest.pt/pin/611152611907051321/>. Acesso em 09 de outubro de 2021

EICOS SENSORES. **Sensor de Nível: o que é? Blog Eicos**, 2016. Disponível em: <<http://eicos.blog.br/sensor-de-nivel-o-que-e/>>. Acesso em: 10 Julho 2021.

VIDAL, V. **Controle de luminosidade com Arduino e sensor LDR**. Eletrograte, 2017. Disponível em: <<http://blog.eletrogate.com/control-de-luminosidade-com-arduino-e-sensor-ldr/>>. Acesso em: 09 Julho 2021.