UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

Fábio de Oliveira Pires

SISTEMA DE CONTROLE PARA FECHADURAS DE PORTAS COM O ARDUINO

Santa Maria, RS
2020

Fábio de Oliveira Pires

SISTEMA DE CONTROLE PARA FECHADURAS DE PORTAS COM O ARDUINO

Trabalho de Conclusão do Curso apresentado ao Curso de Engenharia de Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Engenheiro de Computação.**

Orientador: Prof. Dr. Fábio Ecke Bisogno

Santa Maria, RS
2020

Fábio de Oliveira Pires

SISTEMA DE CONTROLE PARA FECHADURAS DE PORTAS COM O ARDUINO

Trabalho de Conclusão do Curso apresentado ao Curso de Engenharia de Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Engenheiro de Computação.**

Aprovado em 06 de janeiro de 2021:

Fábio Ecke Bisogno, Dr. (UFSM)
(Presidente/Orientador)

Lucas Sangoi Mendonça, Msc. (UFSM)

Jorge Rodrigo Massing, Dr. (UFSM)

Santa Maria, RS
2020

RESUMO

SISTEMA DE CONTROLE PARA FECHADURAS DE PORTAS COM O ARDUINO

AUTOR: Fábio de Oliveira Pires
ORIENTADOR: Prof. Dr. Fábio Ecke Bisogno

Essa pesquisa, decorrente de um Trabalho de Conclusão de Curso, teve como objetivo principal desenvolver um circuito de baixo custo para fechaduras de portas residenciais com a plataforma Arduino, que possa fornecer aos usuários a comodidade de saber se a porta está aberta ou fechada (chaveada) através de um aplicativo de smartphone. Para contemplar esse objetivo, realizamos alguns estudos teóricos para nos auxiliar no desenvolvimento do projeto. Para o desenvolvimento da pesquisa elencamos seis principais etapas: estágio do levantamento de requisitos; configuração do microcontrolador; elaboração do sistema elétrico no Arduino; escolha dos componentes; programação na IDE e sincronização com o aplicativo e a montagem do projeto final bem como a parte prática para o seu funcionamento. Como resultado da pesquisa, o projeto possibilitou detectar com precisão a situação da fechadura. Porém ressaltamos que o aplicativo Blynk no qual foi usado neste trabalho, não foi desenvolvido pelo pesquisador, o que nos deixa ainda algumas alternativas para melhorar nesse aspecto, desde o desenvolvimento próprio de um aplicativo e um sistema de segurança mais confiável.

Palavras-chave: Arduino. Blynk. Microcontolador.

ABSTRACT

CONTROL SYSTEM FOR DOOR LOCKS USING ARDUINO

AUTHOR: Fábio de Oliveira Pires
MENTOR: Prof. Dr. Fábio Ecke Bisogno

This research, resulting from a Final Paper, had as main objective to develop a low-cost circuit for residential door locks with the Arduino platform, which can provide users with the convenience of knowing whether the door is open or closed by means a smartphone app. In order to achieve this objective, theoretical studies were carried out to assist the development of the project. For the development of the research, six main stages are listed: stage of requirements gathering; configuration of the microcontroller; elaboration of the electrical system on Arduino; choice of components; programming in the IDE and synchronization with the application and the assembly of the final project as well as the practical part for its operation. As a result of the research, we concluded that in general the project worked as designed. However, we emphasize that the Blynk application in which it was used in this work, was not developed by the researcher, which still leaves us some alternatives to improve in this aspect, since the development of an application and a more reliable security system.

Keywords: Arduino. Blynk. Microcontroller.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ABREVIATURAS

API	Aplication Programming Interface.
APP	Aplicativo.
CA	Corrente Alternada.
CC	Corrente Contínua.
HTTP	Hyper Text Transfer Protocol.
IDE	Ambiente de Desenvolvimento Integrado.
LED	Light Emitting Diode.
NPN	Negativo-Positivo-Negativo
PLC	Power Line Communication.
PNP	Positivo-Negativo-Positivo.
PWM	Pulse Width Modulation.
R	Resistor.
USB	Universal Serial Bus.
WI-FI	Wireless Fidelity.

SUMÁRIO

1 INTRODUÇÃO

Atualmente a tecnologia vem avançando numa rapidez gigantesca, e assim, devido ao avanço tecnológico o consumo de produtos eletrônicos tem aumentado continuamente, o que tem estimulado cada vez mais o surgimento de sistemas eletrônicos na vida das pessoas modernas. À medida que cresce o consumo de produtos eletrônicos, computadores, telefones celulares, automação residencial e industrial, é necessário produzir equipamentos mais complexos e resistentes a custos mais baixos de produção.

> O uso do raciocínio tem garantido ao homem um processo crescente de inovações. Os conhecimentos daí derivados, quando colocados em pratica, dão origem a diferentes equipamentos, instrumentos, recursos, processos, ferramentas, enfim, a tecnologias. (KENSKI, 2007, p.15).

As tecnologias cada vez mais fazem parte e invadem nossas vidas, desenvolvem nossas memórias, garantem uma nova sensação de bem-estar e fragilizam as habilidades naturais dos seres humanos. Com isso as novas gerações evoluem muito rápido, e como já podemos perceber, somos muito diferentes de outros antepassados, e estamos acostumados a alguns confortos técnicos, como por exemplo, água encanada, luz elétrica, telefones, e nem podemos imaginar como seria a vida sem eles.

> A automação residencial, ou também conhecida como demótica, é uma tecnologia que vem sendo estudada cada vez mais, sendo valorizada e muito explorada, já que visa trazer ao proprietário da casa cada vez mais comodidade e conforto. Ela permite que diferentes ambientes de uma casa seja controlada e monitorada, facilitando o acesso e praticidade, além de aumentar a segurança (BOLZANI, 2004).

Um outro exemplo de conforto, cômodo e segurança, são as fechaduras elétricas. Uma das maiores preocupações das pessoas nos dias de hoje, certamente é a segurança, e assim elas são atualmente, um item fundamental para aumentar a segurança de ambientes, como condomínios, residências e empresas. Elas já são realidade no mercado de trabalho, estando presente nos mais diferentes tipos de projetos, as quais podem ser controladas por senha, trazendo maior comodidade por dispensar o porte de chaves.

Também sabemos que existem fechaduras elétricas com conexões à smartphones, que informam ao usuário se a porta está fechada ou aberta, um exemplo seriam os modelos de fechaduras da Utraloq (2019), algumas fechaduras com mais de seis funções. Porém o custo destas é muito alto e inviável para a grande maioria da população.

Pensando na segurança e no custo benefício para essas pessoas, além da comodidade, e também a parir de experiências do cotidiano em que os sujeitos após saírem de casa e em determinado tempo lembram se a fechadura da porta foi realmente fechada e precisam retornar para verificar, surge o interesse e motivação em pesquisar um sistema de baixo custo, sem o uso de fechaduras elétricas e fazer um projeto relacionado a este tema.

A problematização do trabalho surge nas seguintes necessidades:

- 1. Evitar retornar à residência para a verificação se a porta estava realmente fechada;
- 2. Evitar deslocamentos desnecessários causados por insegurança;
- 3. Garantir checagem automática da situação das portas.

É a partir dessas questões que foi formulado o objetivo desse trabalho, que é desenvolver um circuito de baixo custo para fechaduras de portas residenciais com a plataforma Arduino, que possa fornecer aos usuários a comodidade de saber se a porta está aberta ou fechada através de um aplicativo de smartphone. Para isso, definimos os seguintes objetivos específicos:

- Criar um circuito base;
- Realizar a conexão do sensor indutivo com o Arduino;
- Implementar um aplicativo livre pelo smartphone que possa se comunicar com o Arduino e realizar o controle de verificação do sensor.

Dessa forma, o presente trabalho estrutura-se em cinco capítulos. O primeiro apresenta a pesquisa, explicitando o interesse e motivação da escolha do tema do pesquisador, bem como o objetivo geral e os objetivos específicos. O segundo aborda a fundamentação teórica do trabalho a partir de conceitos e assuntos que foram estudados e pesquisados em diversas fontes para a constituição e implementação do projeto.

A metodologia da pesquisa é apresentada no terceiro capítulo, no qual é discorrido sobre os procedimentos metodológicos aderidos para a construção da mesma.

No quarto capítulo, expomos as explicações das etapas que foram realizadas para que a montagem do projeto se tornasse possível.

Para finalizar, no quinto capítulo algumas considerações finais acerca do trabalho, e, posteriormente, as referências bibliográficas utilizadas.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresentará a nossa fundamentação teórica a partir de conceitos e assuntos que foram estudados e pesquisados em diversas fontes, bem como os conhecimentos adquiridos durante o curso de engenharia de computação, o que possibilitou a realização desse trabalho. Será discorrido sobre alguns conceitos essenciais, o qual se faz necessário compreendê-los para entender o desenvolvimento do projeto, conceitos como, a fechadura convencional, funções e componentes dos materiais que foram utilizados.

2.1 FECHADURA

De acordo com o site Yale (2020), a fechadura tem sua história desde as ruinas do Palácio de Khorsabad, no Iraque, e seus princípios básicos de tambor de pinos podem ter sido datados por volta de 4000 A.C. no Egito. Porém, em 1805 a primeira patente de fechadura a tambor, foi historicamente concedida na Inglaterra ao médico estadunidense Abraham O. Stansbury, da qual seguia os princípios básicos das fechaduras egípcias. Mas em torno de 1848, Linus Yale desenhou e fabricou uma série de fechaduras de alta segurança para sua loja, mas foi seu filho Linus Yale Jr perito da época, isto em 1840, que ao espelhar na fechadura de seu pai que resolveu inventar e adaptar uma chave menor e dentada a fechadura de tambor, transformando fechaduras com inúmeras invenções de sucesso, a qual segue seu modelo até os dias atuais.

Figura 1 – Fechadura de tambor de pinos

Fonte: https://pt.wikipedia.org/wiki/Fechadura_de_tambor_de_pinos

2.2 ARDUINO

Arduino UNO (UNO, 2013) é uma plataforma de prototipagem formada por uma placa eletrônica (*open-source)* com um microprocessador e um ambiente de programação integrado. Recentemente, muito se tem discutido sobre seu uso em projetos de tecnologia, em sua versão UNO, possui o microcontrolador ATMega328 de 8 bits, e através dele pode receber sinais elétricos por vários sensores e assim com sua interface controlar luzes, motores, relês, leds, entre outros atuadores. Além disso, ele possui muitas vantagens em que,

> A maior vantagem do Arduino em relação a outras plataformas de desenvolvimento de micro controladores é a sua facilidade de utilização, o que permite que pessoas que não sejam de áreas técnicas possam aprender o básico e criar seus próprios projetos em um período relativamente curto. (MCROBERTS, 2011, p.24).

O Arduino possui uma IDE (*Integrated Development Environment*) que é uma aplicação escrita em linguagem Java onde pode-se desenvolver códigos em linguagem C/C++ para controlar os pinos do microcontrolador. Essa IDE está disponível para download gratuitamente no site do fornecedor (Arduino, 2013), sendo assim,

> O Arduino é uma plataforma de computação física de fonte aberta, com base em uma placa simples de entrada/saída (input/output, ou I/O), assim como em um ambiente de desenvolvimento que implementa a linguagem Processing (www.processing.org)" (BANZI, 2011, p. 17).

Ao presente trabalho usamos o Arduino UNO como mostrado a seguir na figura 2, e suas características no quadro abaixo, o qual será centro de todo nosso trabalho.

Quadro 1 – Características do Arduino Uno

Componentes	**Detalhes**
Microcontrolador	ATMega328
Tensão de operação	5V
Tensão de entrada (recomendado)	7-12V
Pinos digitais de I/O	14 (6 com saída PWM)
Pinos de entrada analógica	6
Corrente por pino de I/O	40 mA
Corrente pelo pino de 3,3V	50 mA
Memória Flash	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidade do Clock	16 MHz

Fonte: Organizado a partir de Silva (2014).

O Arduino UNO é uma versão melhorada e mais atual do que sua antecessora, a versão Duemilenove, claro que ambos são similares, mas o Arduino UNO como dito anteriormente possui o microcontrolador ATMega328, que disponibiliza 20 pinos de I/O, possui também um cristal de 16 MHz, conector de potência – *jack*, possui conexão USB (*Universal Serial Bus*), além disso também possui um botão *reset*, sem esquecer dos pinos de gravação *In-System*, e como mostrado no quadro 1, pode ser alimentado pelo USB ou por uma bateria/fonte de tensão, mas ambos fornecendo a tensão de entrada recomendada. Podemos analisar melhor a sua estrutura/circuito no anexo A.

Figura 2 – Arduino Uno Rev3

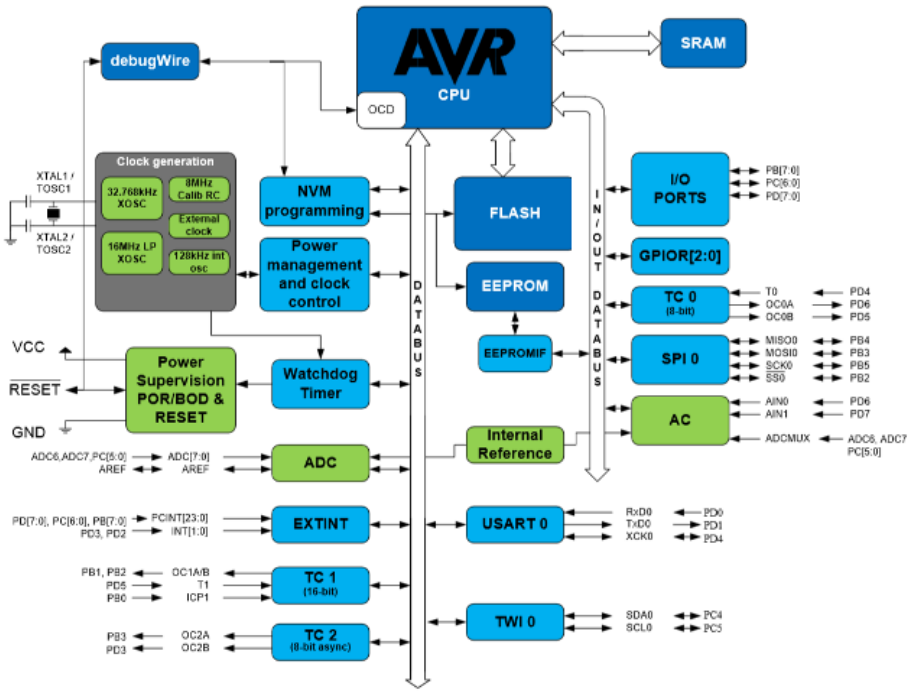
Fonte: www.arduino.cc

2.3 ATMEGA328

Um micro controlador define-se como um sistema micro processado, a partir dele, também pode se construir um computador em um único circuito integrado (Chip), podendo ser desde um desktop até um Mainframe. Ele pode também ser considerado um *CPU* (Unidade Central de Processamento), *single-chip computer* (computador em um único chip), *microcomputer*, ou *como embedded controller*. Nesse mesmo chip tem integrado uma CPU, da qual chamada de core ou (núcleo), que possui circuitos auxiliares (periféricos), como memória programada, memória de dados, clocks, interface de comunicação serial, temporizadores/controladores, portas de I/O e entre outros. Porém, as suas funções podem variar de fabricante para fabricante, dependendo de os desenvolvedores especificarem os seus micros controladores e exemplificando adequadamente cada uma de suas aplicações.

Como as placas Arduino Uno, possuem o micro controlador ATMega328, que tem uma alta performance e que pode executar instruções com um ciclo de clock e alcance 1 MIPS/MHz (1 Milhão de Instruções por segundo por Mega Hertz), do qual nos possibilitará velocidade no processamento e consumo de potência. O ATMega328 pertence ao fabricante Atmel, com o núcleo e arquitetura da família RISC AVR, possibilitando um alto desempenho com um custo baixíssimo e uma ótima optimização para compiladores C/C++. A figura 3 mostrará um pouco dessa arquitetura.

Figura 3 – Diagrama de blocos do micro controlador



Fonte: https://www.newtoncbraga.com.br

O microcontrolador AVR surgiu na década de 90, o qual podemos perceber o seu avanço tecnológico até os dias atuais. Assim, conforme Lima (2012),

> A história dos microcontroladores AVR começa em 1992 na Norwegian University of Science and Technology (NTNU), na Noruega, onde dois estudantes de doutorado, Alf-Egil-Boden e Vegard Wollan, defenderam uma tese sobre um microcontrolador de 8 bits com memória de programa flash e arquitetura RISC avançada. Dos seus nomes surgiu a sigla AVR: Alf – Vegard – RISC. Percebendo as vantagens do microcontrolador que haviam 13 criado, Alf e Vegard refinaram o seu projeto por dois anos. Com o AVR aperfeiçoado, foram para a Califórnia (EUA) para encontrar um patrocinador para a ideia, a qual foi comprada pela Atmel, que lançou comercialmente o primeiro AVR em meados de 1997, o AT90S1200. Desde

> então, o aperfeiçoamento do AVR é crescente, juntamente com a sua popularidade. (p. 12-13).

O ATMega328 possui alguns dados técnicos que serão apresentados no quadro a seguir.

Quadro 2 – Dados técnicos

Fabricante	**Atmel**
Núcleo	AVR
Largura de Barramento de Dados	8 bits
Frequência de operação máxima	20 MHz
Tamanho da memória do programa	32 KB
Tamanho RAM dos dados	2 KB
Caixa / Gabinete	PDIP-28
Tamanho de bit A/D	10 bits
Canais A/D disponíveis	6
Tipo de interface	2-Wire, SPI, USART
Número de I/Os programáveis	23
Tensão de alimentação - Máx	5.5V
Tensão de alimentação - Min	1.8V

Fonte: Organizado pelos autores.

Como o ATMega328 tem várias de suas características da família AVR, e também por ser um micro controlador compacto, isso facilita no fator decisivo atual para o desenvolvimento na plataforma Arduino UNO e sendo assim uma fácil aquisição. De acordo com Lima (2012), as características do ATMega328 são:

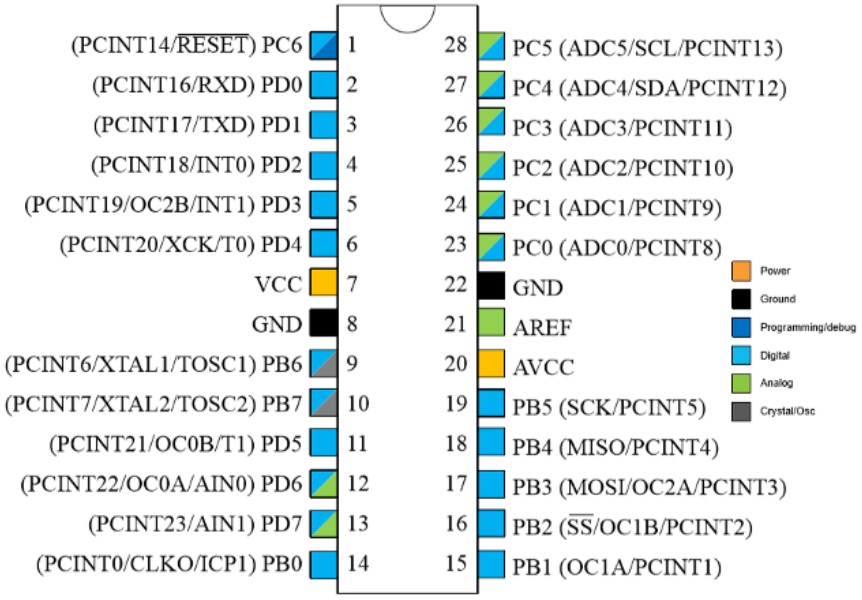
> - Microcontrolador de baixa potência, com arquitetura RISC avançada.
> - 131 instruções, a maior parte executada em 1 ou 2 ciclos de clock (poucas em 3 ou 4 ciclos).
> - 32 registradores de trabalho de propósito geral (8 bits cada). Alguns trabalham em par para endereçamentos de 16 bits.
> - Operação de até 20 MIPS a 20 MHz.
> - Multiplicação por hardware em 2 ciclos de clock.
> - 32 kbytes de memória de programa flash de autoprogramação In-System (8 k, 16 k, nos respectivos ATmega88 e ATmega168).
> - 1 kbytes de memória EEPROM.
> - 2 kbytes de memória SRAM.
> - Ciclos de escrita e apagamento: memória flash 10 mil vezes, EEPROM 100 mil vezes.
> - Seção opcional para código de boot para programação In-System por boot loader6.
> - Bits de bloqueio para proteção contra a cópia do firmware.
> - Possui os seguintes periféricos: 23 entradas e saídas (I/Os) programáveis. 2 Temporizadores/Contadores de 8 bits com Prescaler separado, com modo de comparação. 1 Temporizador/Controlador de 16 bitz com Prescaler separado, com modo de comparação e captura. Contador de tempo real (com um cristal externo de 32,768 kHz conta precisamente 1s). 6 canais PWM. 8 canais AD com resolução

de 10 bits na versão TQFP (Thin profile plastic Quad Flat Package) e 6 canais na versão PDIP (Plastic Dual Inline Package). Interface serial para dois fios orientada a byte (TWI), compatível com o protocolo I2C. Interface serial USART. Interface serial SPI Master/Slave. Watchdog Timer com oscilador interno separado. 1 comparador analógico.

- Características especiais: Power-on Reset e detecção Brown-out programável. Oscilador interno RC (não há a necessidade do uso de cristal externo ou de outra fonte de clock). Fontes de interrupções internas e externas (em todos os pinos de I/O). Saída de clock em um pino de I/O (PB0). Pull-up habilitáveis em todos os pinos de I/O. 6 modos de Sleep: Idle, Redução de ruído do ADC, Power-down, Powersave, Standby e Extended Standby. Medição de temperatura do encapsulamento.
- Tensão de operação: 1,8 - 5,5 V.
- Consumo de corrente a 1 MHz (1,8 V, 25 ºC): modo ativo = 0,2 mA e modo Power-down = 0,1 µA. (p. 17-18).

Isso explicita um pouco da sua capacidade e o comportamento do funcionamento do ATMega328, e na figura 4 ilustramos o micro controlador, bem como no quadro 3 a descrição de seus pinos e portas.

Figura 4 – Portas e pinos de tensão do ATMega328



Fonte: https://www.newtoncbraga.com.br

Quadro 3 – Descrição dos pinos do ATMega328

Pinos de Alimentação	
VCC	Tensão de alimentação.
AVCC	Pino para a tensão de alimentação do conversor AD. Deve ser externamente conectado ao VCC, mesmo se o ADC não estiver sendo utilizado.
AREF	Pino para a tensão de referência analógica do conversor AD.
GND	Terra.
PORTB	
PB0	ICP1 – entrada de captura para o Temporizador/Contador 1. CLKO – saída de clock do sistema. PCINT0 – interrupção 0 por mudança no pino.
PB1	OC1A – saída da igualdade de comparação A do Temporizador/Contador 1 (PWM). PCINT1 – interrupção 1 por mudança no pino.
PB2	SS – Pino de seleção de escravo da SPI (Serial Peripheral Interface). OC1B – saída da igualdade de comparação B do Temporizador/Contador 1 (PWM). PCINT2 – interrupção 2 por mudança no pino.
PB3	MOSI – pino mestre de saída e escravo de entrada da SPI. OC2A – saída da igualdade de comparação A do Temporizador/Contador 2 (PWM). PCINT3 – interrupção 3 por mudança no pino.
PB4	MISO – pino mestre de entrada e escravo de saída da SPI. PCINT4 – interrupção 4 por mudança no pino.
PB5	SCK – pino de clock da SPI. PCINT5 – interrupção 5 por mudança no pino.
PB6	XTAL1 – entrada 1 do oscilador ou entrada de clock externa. TOSC1 – entrada 1 para o oscilador do temporizador (RTC). PCINT6 – interrupção 6 por mudança no pino.
PB7	XTAL2 – entrada 2 do oscilador. TOSC2 – entrada 2 para o oscilador do temporizador (RTC). PCINT7 – interrupção 7 por mudança no pino.
PORTC	
PC0	ADC0 – canal 0 de entrada do conversor AD. PCINT8 – interrupção 8 por mudança no pino.
PC1	ADC1 – canal 1 de entrada do conversor AD. PCINT9 – interrupção 9 por mudança no pino.
PC2	ADC2 – canal 2 de entrada do conversor AD. PCINT10 – interrupção 10 por mudança no pino.
PC3	ADC3 – canal 3 de entrada do conversor AD. PCINT11 – interrupção 11 por mudança no pino.
PC4	ADC4 – canal 4 de entrada do conversor AD. SDA – entrada e saída de dados da interface a 2 fios (TWI – I2C). PCINT12 – interrupção 12 por mudança no pino.
PC5	ADC5 – canal 5 de entrada do conversor AD. SCL – clock da interface a 2 fios (TWI – I2C). PCINT13 – interrupção 13 por mudança no pino.
PC6	RESET – pino de inicialização. PCINT14 – interrupção 14 por mudança no pino.
PORTD	
PD0	RXD – pino de entrada (leitura) da USART. PCINT16 – interrupção 16 por mudança no pino.
PD1	TXD – pino de saída (escrita) da USART. PCINT17 – interrupção 17 por mudança no pino.
PD2	INT0 – entrada da interrupção externa 0. PCINT18 – interrupção 18 por mudança no pino.
PD3	INT1 – entrada da interrupção externa 1.

	OC2B – saída da igualdade de comparação B do Temporizador/Contador 2 (PWM) PCINT19 – interrupção 19 por mudança no pino.
PD4	XCK – clock externo de entrada e saída da USART. T0 – entrada de contagem externa para o Temporizador/Contador 0. PCINT 20 – interrupção 20 por mudança no pino.
PD5	T1 – entrada de contagem externa para o Temporizador/Contador 1. OC0B – saída da igualdade de comparação B do Temporizador/Contador 0 (PWM). PCINT 21 – interrupção 21 por mudança no pino.
PD6	AIN0 – entrada positiva do comparador analógico. OC0A – saída da igualdade de comparação A do Temporizador/Contador 0 (PWM). PCINT 22 – interrupção 22 por mudança no pino.
PD7	AIN1 – entrada negativa do comparador analógico. PCINT 23 – interrupção 23 por mudança no pino.

Fonte: Organizado de acordo com Lima (2012).

Como especificado no quadro acima, mostramos como o ATMega está organizado, cada um deles com 8 bits (PORTB, PORTD) e (exceto o PORTC), todos trabalham e se organizam em conjunto. Cada PORT se caracteriza por uma porta bidirecional de I/O com 8 bits e que possuem resistores internos de *pull-up* que selecionam cada bit. Também tem seus registradores simétricos e com uma capacidade de receber e fornecer uma corrente de 40 mA, e seus pinos tem pelo menos duas funções distintas, incluindo até o pino de *reset*, tornando-se assim sua principal e importante característica.

2.4 APLICATIVO BLYNK

Como o projeto tem como base o Arduino UNO, o conveniente seria usar um aplicativo livre como o Blynk. Esse aplicativo é personalizável, tendo como função o controle de um hardware programável, ou por uso de shields ou placas de wi-fi (Wireless Fidelity) e bluetooth. Como o aplicativo recebe dados desses hardwares, fica fácil ter um gráfico rápido e intuitivo do controle, mas ele não se baseia só em Arduino, assim como também interage com mais de 400 placas de desenvolvimento atualmente. O Blynk é um aplicativo gratuito, tem o acesso ilimitado aos seus servidores e possui widgets. Porém, existem também widgets que custam *energy*, que é um tipo de moeda virtual da aplicação e de baixo custo, mas que nem sempre são necessários utilizá-los em determinados projetos, pois o limite de *energy* disponibilizado na *energy box* é gratuitamente.

De acordo com (Brito, 2016), “o Blynk foi um projeto desenvolvido pela *Massachusetts Institute of Technology* (MIT), com a finalidade de contribuir para o ensino e a aderência de mais pessoas para o conceito de Internet das Coisas (IOT), aonde qualquer

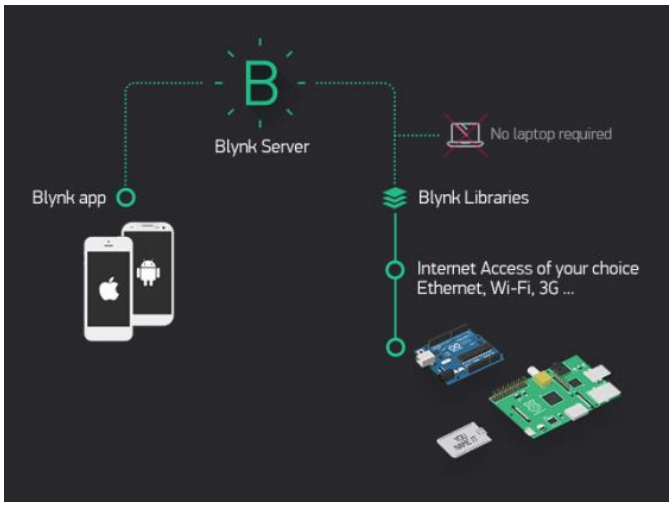
pessoa com um mínimo de noção de programação, consegue enviar e apresentar dados para o Blynk" (2016, p.53).

O Blynk funciona basicamente em três partes, o Blynk App (aplicativo), o Blynk *Server* (servidor) e o Blynk *Library* (biblioteca).

- Blynk App: aplicativo disponível na plataforma IOS e Android, o qual tem como finalidade proporcionar ao usuário a criação de aplicações que interagem com o hardware, sendo assim o usuário tem controle das funções pela a inserção de widgets (botões, sliders e chaves) e de tal forma receber a leitura dos dados do hardware usado.
- Blynk Server: é o que permite toda comunicação entre o hardware e o aplicativo através do cloud Blynk. Esse Cloud é responsável pelo servidor, de transmitir seus dados com o hardware, assim armazenando estados/dados do aplicativo e do hardware que serão usados no momento. Mas todos esses dados armazenados no servidor Blynk podem ser acessados pelo usuário através de um API (*Application Programming Interface*) HTTP (*HyperText Transfer Protocol*).
- Blynk Libraries: tem como finalidade proporcionar o hardware todas as bibliotecas do Blynk, isso tudo para várias/plataformas de desenvolvimento. Essa Libraries gera a conexão do servidor Blynk com o hardware, e assim tendo o controle de todas entradas/saídas de comandos e dados.

Na imagem a seguir, vejamos um diagrama do funcionamento básico das três partes do Blynk.

Figura 5 – Funcionamento do Blynk



Fonte: Retirado de http://docs.blynk.cc/

Como dito anteriormente o aplicativo Blynk interage com a placa Arduino UNO, coração do projeto, mas para isso acontecer é necessário instalar a biblioteca Blynk no IDE do Arduino, e realizar o download da biblioteca pelo acesso desse link https://github.com/blynkkk/blynk-library/releases/tag/v0.5.2. A partir disso, e já com os *sketchbook* já instalados, teremos o controle do aplicativo com o Arduino, deixando toda a parte de programação para a IDE do Arduino.

2.5 MÓDULO WI-FI ESP8266

O módulo wi-fi ESP8266 é de baixo custo e foi desenvolvido especialmente para que o usuário possa se conectar facilmente ao micro controlador através de uma conexão wi-fi. Esse módulo wi-fi tem uma característica que pode ser pequeno, porém é muito poderoso, suportando redes 802.11 b/g/n, as quais são muito utilizadas atualmente, e também permite trabalhar como ponto de acesso (*Acess Point*) ou uma estação (*Station*) que envia e recebe dados. Suas especificações podem ser analisadas no quadro a seguir.

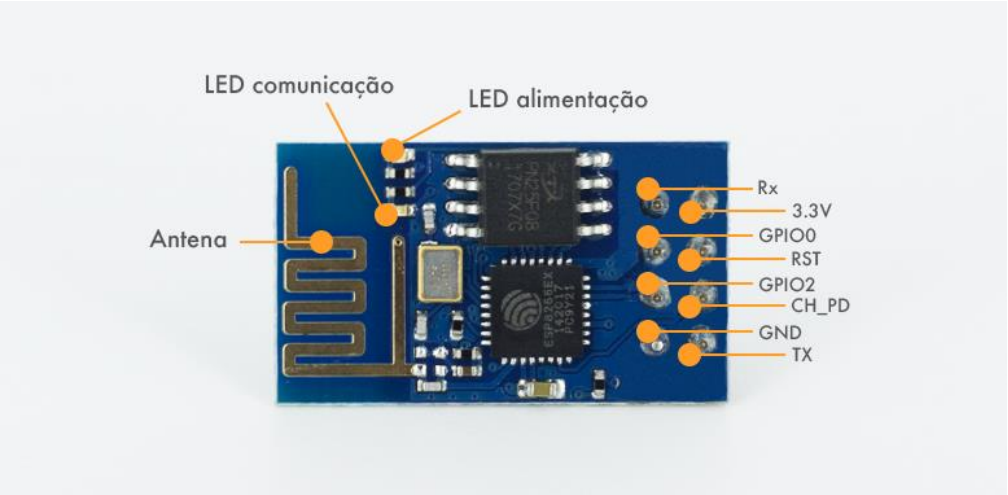
Quadro 4 – Especificações do módulo ESP8266

ESPECIFICAÇÕES	
Chip	ESP8266
Modelo	ESP-01
Tensão de operação	3,3V
Suporte à redes	802.11 b/g/n
Alcance	90m aprox..
Comunicação	Serial (TX/RX)
Suporta comunicação	TCP e UDP
Conectores	GPIO, I2C, SPI, UART, Entrada ADC, Saída PWM e Sensor de Temperatura interno
Modo de segurança	OPEN/WEP/WPA_PSK/WPA2_PSK/WPA_WPA2_PSK
Dimensões	25 x 14 x 1mm
Peso	7g

Fonte: Organizado pelos autores.

O módulo ESP8266 possui antena embutida e também possui um conector de 8 pinos, e interage seu funcionamento com o usuário através dos leds, em funcionamento (vermelho) e com comunicação (azul). Na figura a seguir, vejamos um breve detalhamento de seus pinos.

Figura 6 – Detalhamento dos pinos



Fonte: Retirado de https://www.filipeflop.com/blog/esp8266-arduino-tutorial/

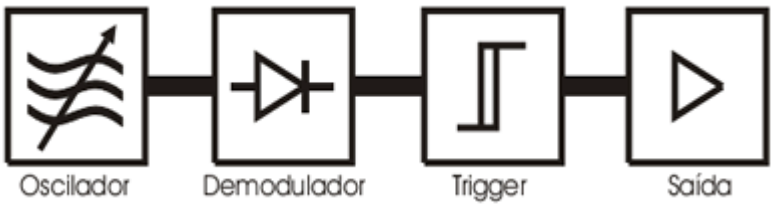
2.6 SENSOR INDUTIVO LJ12A3-4-Z

Os sensores indutivos são transmissores de sinais, capazes de detectar elementos metálicos que passam pelo campo magnético sem contato direto e convertê-los em sinais elétricos compreensíveis. Esses sensores consistem basicamente em uma bobina ao redor do núcleo. Quando existem objetos com magnetismo (como ímãs, materiais ferrosos e até materiais diamagnéticos e as linhas de força que dispersam o campo magnético) as características da bobina mudam, porque interferem no campo magnético gerado pelo oscilador conectado à bobina. Essa mudança é percebida (a uma distância predeterminada) e o sensor muda.

> A presença de um objeto altera esse campo e o circuito eletrônico do sensor pode descobrir a alteração. Um sensor de proximidade indutivo inclui um circuito oscilador LC, um comparador de sinal e um chaveador. A bobina desse circuito oscilador gera um campo eletromagnético de alta frequência. Esse campo é emitido à face do sensor. (THOMAZINI, 2011, p. 33)

De acordo com o site (MECAWEB), sensores de proximidade indutivos são elementos que podem executar comutação elétrica sem tocar em um corpo de metal. Como será apresentado no diagrama de blocos da figura a seguir.

Figura 7 – Diagrama de blocos do sensor indutivo LJ12A3-4-Z



Fonte: Retirado de http://www.mecaweb.com.br/eletronica/content/e_sensor_indutivo

O sensor usa o oscilador como seu "coração". Quando um objeto metálico é introduzido no campo magnético da bobina interna do sensor, essa oscilação muda e, quando o objeto é removido, a oscilação volta ao normal.

O gatilho explicará a mudança no comportamento do oscilador para obter a saída do sinal ALTO-BAIXO (onda quadrada), que pode excitar o circuito de potência, como um transistor, para obter uma chave liga-desliga. Estado sólido, capaz de alternar bobinas de relé, contatores pequenos e até circuitos lógicos. Todo o dispositivo eletrônico é montado através de uma tecnologia do mercado atual, instalado em um gabinete de plástico ou metal e encapsulado em resina de alta densidade para formar um forte bloco à prova d'água que pode resistir a vibrações e intempéries Corrente Alternada (CA), Corrente Contínua (CC), Saída normalmente aberta, normalmente fechada e até transistores negativo-positivo-negativo (NPN) ou positivo-negativo-positivo (PNP) que são fáceis de integrar com controladores lógicos programáveis (Power Line Communication /PLC).

O sensor que foi usado nesse projeto será o Sensor indutivo PNP de proximidade LJ12A3-4-Z ilustrado na figura abaixo, porque ele é muito aplicável em projetos com Arduino ou com outras plataformas micro controlada. Este sensor indutivo além de vantajoso é usualmente viável e também aplicado na indústria, essas vantagens vão de sua vida útil estendida, e se comparar a outros sensores sua confiabilidade é melhor, como os mecânicos por exemplo.

Figura 8 – Sensor indutivo LJ12A3-4-Z

Fonte: Retirado de https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-indutivo-pnp-de-proximidade-lj12a3-4-z-by/

Como descrito anteriormente, esse sensor tem as características básicas como qualquer outro sensor indutivo (componente eletrônico que detecta aproximação de objetos metálicos). No quadro a seguir apresentaremos suas especificações.

Quadro 5 – Especificações do sensor indutivo LJ12A3-4-Z

ESPECIFICAÇÕES	
Modelo	LJ12A3-4-Z/BY
Tensão de operação	6 - 36VDC
Corrente de saída	300mA
Conexão	3 fios
Polaridade	PNP
Distância de detecção	4mm / 2mm para alumínio
Estado da saída	NA (normalmente aberto)
Comprimento do cabo	110cm
Comprimento do sensor	65mm
Diâmetro	10mm
Peso	40g

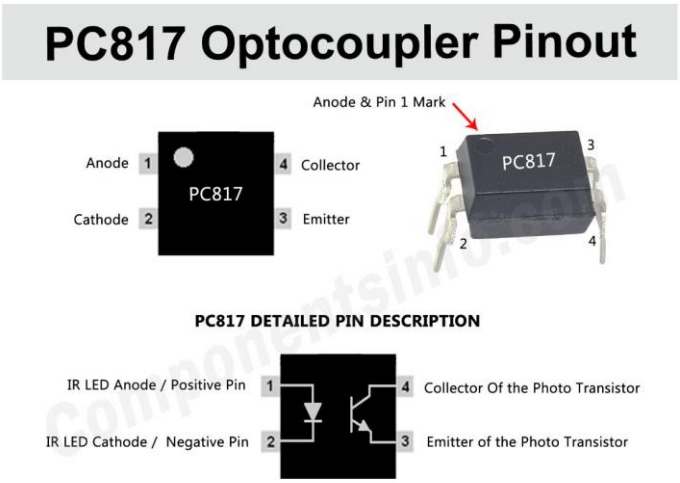
Fonte: Organizado pelos autores.

2.7 PC817

Optoacoplador PC817 (constituído de um diodo emissor de luz infravermelho e um fototransistor de silício) é um componente eletrônico, que tem a função de impedir que exista um contato elétrico entre dois circuitos, e é capaz de fazer o isolamento de algumas partes do circuito. São muito utilizados em controles de relés, motores e outros dispositivos, por

necessitar de baixa potência de operação e ser muito mais rápido possibilitando aprova de interferência, sendo que cada vez mais está ganhando espaço no mercado de automação com o passar dos anos.

Figura 9 – Pinagem do PC817



Fonte: Retirado de https://www.componentsinfo.com/pc817-optocoupler-pinout-datasheet/

Conectado a um tiristor ele assegura que, mesmo em caso de uma grande descarga elétrica o circuito eletrônico lógico continue operando, e isso limita o estrago a alguns poucos componentes. No quadro a seguir e de acordo com o (*datasheet*) suas especificações.

Quadro 6 - Especificações do PC817

ESPECIFICAÇÕES	
Fabricante	SHARP
Tipo de elemento semicondutor	Acoplador ótico
Montagem	THT
Número de canais	1
Tipo de saída	transistor
Tensão de isolamento	5kV
CTR@If	50-600%@5mA
Tensão coletor-emissor	35V
Carcaça	DIP4
Peso bruto	0,43g

Fonte: Organizado pelos autores.

2.8 MICRO SERVO 9G SG90 TOWERPRO

O Micro Servo 9g Tower Pro é excelente para o controle de posição angular e aplicações com o Arduino, é também um servo de alta qualidade e que atende as necessidades do nosso projeto, porem este Micro Servo Tower Pro não se limita apenas a esse projeto ou ao Arduino em si, ele também tem suporte para outro micro controlador, como por exemplo, o PIC, Raspberry e entre outros. O Micro Servo 9g Tower Pro pode ser pequeno, mas possui um torque muito bom para aplicações educacionais. E na figura a seguir apresentaremos suas dimensões e o modelo utilizado.

Figura 10 – Dimensões e modelo

Fonte: Retirado de http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

O servo motor é conhecido apenas como *servo*, de acordo com Santos (2007, p. 2) ‘’o servo é um dispositivo eletromecânico que, a partir de um sinal elétrico em sua entrada, pode ter seu eixo posicionado em uma determinada posição angular. Por serem pequenos e compactos, além de permitir um posicionamento preciso de seu eixo, os servos motores são largamente utilizados em robótica e modelismo’’. Ele é excelente para controle de posição angular (-90 a 90) possuindo um grau de liberdade de 180 através de *Pulse Width Modulation* (PWM). A seguir apresentaremos no quadro as especificações do Micro Servo motor SG90 - 9g Tower Pro.

Quadro 7 – Especificações do Micro Servo 9g SG90 TowerPro

ESPECIFICAÇÕES	
Tensão de operação	3 a 7,2V
Modulação	PWM
Faixa de Rotação	180º
Velocidade (4.8V)	0.12 s / 60º
Torque (4.8V)	1,5kg/cm
Dimensões	22x12x19mm
Peso	9g
Tamanho do cabo	24cm
PINAGEM	
Fio Laranja	Sinal PWM
Fio Vermelho	+VCC
Fio Marrom	GND

Fonte: Organizado pelos autores.

Neste capítulo foi abordado de forma breve um conteúdo sobre fechadura, microcontrolador, Arduino e periféricos do mesmo, claro que, não podemos esquecer do app Blynk. Claramente e com uma maior visibilidade a aplicação dos mesmos vem crescendo significativamente e evidenciando que a utilização destes aplicativos e sistemas, é de forma visível, e promissora para o desenvolvimento de projetos de ampla aplicabilidade e baixo custo. Logo, e com o atual cenário é motivador para a introdução da utilização destes sistemas fora e dentro do ambiente acadêmico.

No próximo capítulo, serão descritos a metodologia e desenvolvimento do projeto, juntamente com os componentes usados para esse trabalho.

3 METODOLOGIA E DESENVOLVIMENTO DO PROJETO

Esse trabalho teve como objetivo desenvolver um circuito de baixo custo para fechaduras de portas residenciais com a plataforma Arduino, que possa fornecer aos usuários a comodidade de saber se a porta está aberta ou fechada através de um aplicativo de smartphone.

Assim, a metodologia do trabalho apresenta o desenvolvimento de uma interface de programação embarcada servida via aplicação web, com uso de duas plataformas: wi-fi e bluetooth em conjunto com um Arduino. A fim de contemplar o objetivo, o trabalho foi desenvolvido por meio de algumas etapas, sendo: programação no IDE do Arduino; parte elétrica e montagem no Arduino; conexão com o aplicativo e por fim uma parte prática do funcionamento.

Deste modo, o presente trabalho visa um projeto para controle de porta ou qualquer outro formato semelhante, sendo um protótipo para gerenciamento através do smartphone via uma aplicação web. Um sistema criado para enviar uma mensagem para o aplicativo no smartphone quando a porta residencial for chaveada, assim o smartphone receberá um aviso através de uma mensagem, avisando que a porta foi chaveada. Caso o usuário no momento em que chavear a porta esquecer-se de visualizar a mensagem, também poderá mais tarde conferir pelo aplicativo se a porta foi realmente chaveada. O aplicativo consiste em uma interface simples, facilitando o manuseio do mesmo, sem muitos procedimentos.

Apresento um exemplo para ilustrar o seu manuseio. Se o usuário tenha esquecido-se de chavear a porta de sua residência, ele pode conferir pelo aplicativo utilizado no procedimento. No aplicativo foi posto um Widget *Light Emitting Diode* (LED) que servirá para informar quando a porta estará chaveada ou aberta, ou seja, quando o Widget LED estiver no estado “on”, a porta estará chaveada, e quando estiver no estado “off” a porta estará destrancada.

Também foi implementado outros comandos, como uma ferramenta no aplicativo, ou seja, um botão slider que aciona um servo motor, o qual é responsável por ativar ou desativar uma tranca, similar às portas elétricas de hoje em dia. Assim o usuário não precisa se deslocar até sua residência ou local que tenha sido instalado o sistema, podendo muito facilmente trancar sua porta através do acionamento deste botão.

Claro que para a pesquisa foi utilizada a programação em C, pois o código ficaria de melhor entendimento e com uma fácil compreensão, além de proporcionar uma implementação futura. Com o procedimento do próprio Arduino e a inserção de pacotes e

recursos do aplicativo Blynk, propicia uma melhor comunicação entre ambos. Já na parte elétrica e montagem do projeto como já citado, foi usado um Arduino e os conhecimentos adquiridos no decorrer do curso de engenharia, mais especificamente sobre as ligações e quais tensões seriam desejadas.

O desenvolvimento desse projeto foi realizado em algumas etapas como já citado anteriormente, sendo:

- O levantamento de requisitos;
- A configuração do microcontrolador;
- A elaboração do sistema elétrico no Arduino;
- A escolha dos componentes;
- Programação na IDE e sincronização com o aplicativo, e por fim;
- A montagem do projeto final bem como a parte prática para o seu funcionamento.

Para tanto, será discorrido sobre cada etapa.

3.1 LEVANTAMENTO DE REQUISITOS

Nesta primeira etapa, foi realizado o levantado de requisitos para definir o que deveria ou não, fazer parte do projeto. Essa etapa foi importante para estabelecer o escopo que o produto final deveria atender, além do desenvolvimento das estratégias.

Para ser capaz de interpretar e executar a lógica desenvolvida, o hardware necessita ser micro processado e, por questões de custo e quantidade de periféricos, o microcontrolador usado foi o Atmega328. Ainda que os microcontroladores atuais estejam cada vez mais rápidos, menores e eficientes e com níveis de tensão muito menores, como o padrão de 3,3V e 5V, isso permite o desenvolvimento de hardwares com menor necessidade de resfriamento e com fontes de alimentação mais simples e com menor potência. O ATMEGA328 é um chip de alto desempenho como já citado no referencial teórico, possuindo 8 bits AVR RISC, o que combina o alto desempenho com a memória flash ISP de 32 KB, com recursos de leitura durante a gravação, entre outros complementares desse microcontrolador.

A fonte de alimentação foi definida no período de teste com uma fonte controlada com 9V no sensor indutivo LJ12A, pois ele opera entre 6V a 36V, e o Arduino foi alimentado com 5V, fornecidos pelo próprio Universal Serial Bus **(**USB). Já no servo motor por exigir uma quantidade maior de corrente, foi utilizado quatro baterias de 1,5V em série, pois, se o servo motor ligado ao Arduino, ele vibraria muito devido a ligação com baixar corrente, fornecida pelo próprio Arduino, e assim, o uso dessas baterias separadas deixaria o sistema mais estável.

Claro que nesse caso a maior parte dos componentes internos faz uso dos 3.3V, fornecidos pelos pinos digitais do Arduino.

Como os dispositivos de entrada (sensores) e saída (atuadores) pode transmitir e receber sinais digitais e analógicos é de suma importância que as interfaces de entrada e saída (I/O) do hardware sejam capazes, de gerar e interpretar sinais desse tipo. O protótipo do projeto no modelo WI-FI ou Bluetooth, atribuiu o uso de três entradas digitais, sendo duas para o módulo wi-fi ou bluetooth, uma para o sensor indutivo e outra para a saída digital para o servo motor.

Para o teste, diagnóstico, identificação de erros e/ou remoção de erros ou problemas no microcontrolador, foi realizado testes práticos para encontrar a melhor maneira de adequar as tensões e conexões, e assim um funcionamento mais estável e com menor custo de energia.

Por último, é preciso reconhecer e aceitar as limitações de fabricação e garantir que o resultado final do projeto seja de qualidade e, ao mesmo tempo, de baixo custo. Considerando os fatores acima foi determinado o uso de um optoacoplador, para que as entradas digitais pudessem receber as tensões necessárias e não danificar o Arduino ou até mesmo o microcontrolador ATMEGA328.

3.2 CONFIGURAÇÃO DO MICROCONTROLADOR

A configuração dos 28 pinos do microcontrolador ATMEGA328, foi realizada de uma forma que fosse melhor se adequar ao uso. Essa configuração, em detalhes no quadro a seguir, visa promover condições para a elaboração do esquema elétrico, ao indicar e determinar as respectivas funções de cada um dos pinos usados no microcontrolador.

Quadro 8 – Especificações dos pinos configurados no microcontrolador ATMEGA328

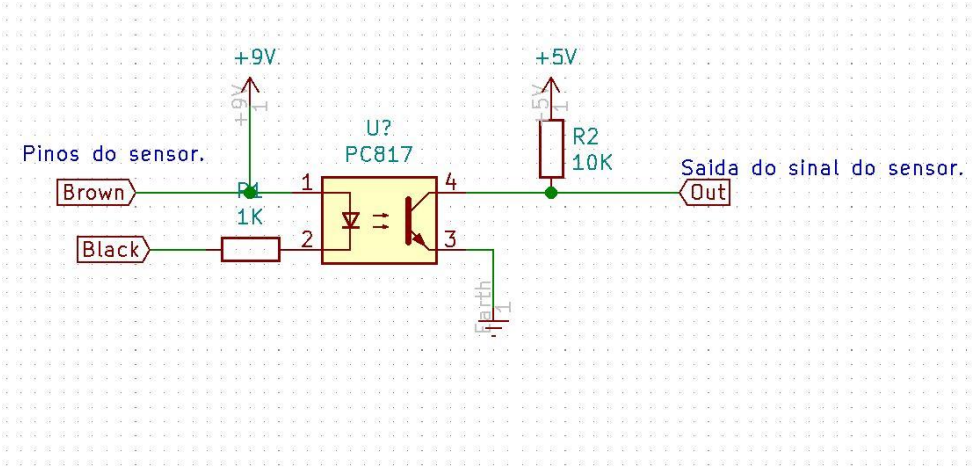
Pin06	**Saída do sinal para o servo motor**
Pin07	VCC
Pin08	GND
Pin14	Entrada do sensor indutivo
Pin16	Entrada do TXH do modulo ESP8266
Pin17	Entrada do RXH do modulo ESP8266

Fonte: Organizado pelos autores.

3.3 ELABORAÇÂO DO ESQUEMA ELÉTRICO

A elaboração do esquema elétrico foi realizada com o auxílio software Kicad, e assim mostrando as conexões junto ao sensor indutivo LJ12A, foi usado um PC817, com a montagem na figura a seguir. Claro que existe outro modo de usar o LJ12A sem a necessidade do PC817, como por exemplo, utilizando um divisor de tensão entre Resistor (R1) e Resistor (R2) na entrada do sinal LJ12A, sendo uma ótima solução, pois assim a tensão que o Arduino receberia ficaria entre seu limitador de 5V.

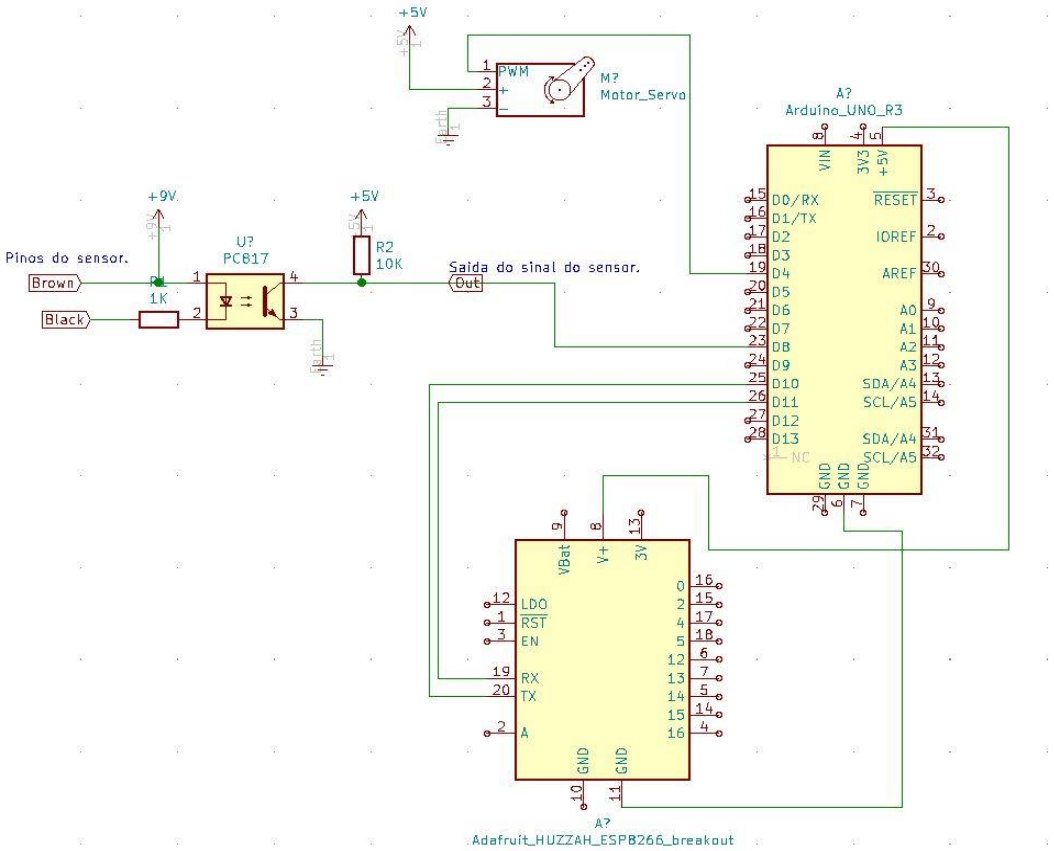
Figura 11 – Esquema elétrico do PC817



Fonte: Autores.

Na parte do modelo wi-fi foi usado o módulo ESP8266, pois a conexão com o modelo bluetooth ficaria muito limitado, ou seja, a poucos metros, e como o projeto é de baixo custo, não teria como usar modelos de transmissão bluetooth mais modernas. Como atualmente a maioria das pessoas tem internet em seus smartphones, o modelo wi-fi ficaria adequado para controles mais longos. Na figura abaixo, visualizamos o esquema elétrico completo de como funciona esse modelo proposto.

Figura 12 – Esquema elétrico do PC817

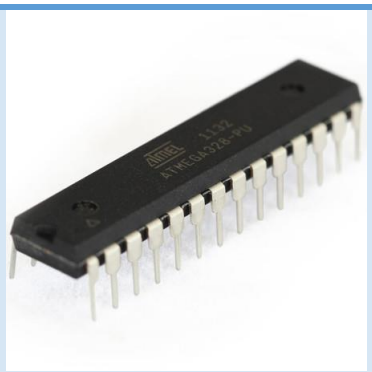
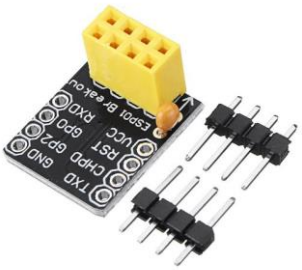


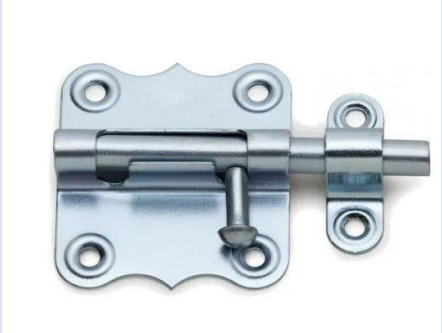
Fonte: Autores.

3.4 ESCOLHA DOS COMPONENTES

A partir da elaboração do esquema elétrico, fez se necessário decidir os componentes para a construção do protótipo do projeto. Desse modo, são apresentados a seguir os componentes escolhidos em ambos os modelos, com seus respectivos números de série, uma ilustração e uma breve descrição.

Quadro 9 – Listagem dos componentes usados no projeto

Número de série	Foto	Descrição
ATMEGA328		Microcontrolador da família Atmel, com núcleo AVR e com uma largura de banda de 8 bits, junto com uma memória de 32KB e mais alguns periféricos.
PC817		Optoacoplador de uso geral, com encapsulamento DIP-4.
ESP8266		Um pequeno módulo wi-fi que permite o Arduino se comunicar a rede. Com duas portas GPIO, um processador RISK de 80 MHz, com fonte de alimentação e níveis lógicos de 3,3V.
Adaptador módulo ESP8266		Para a prototipagem do modulo ESP a protoboard.

LJ12A3-4-Z/BX		Sensor indutivo NPN de proximidade. Com tensão de 6 – 36V e corrente de saída de 300mA.
Micro Servo Towerpro 9G Sg90		Um servo de qualidade que possui ângulo de rotação de 180 graus. Com voltagem de operação de 3 - 7,2V, velocidade de 0,12 seg/60graus (4,8V) sem carga. Entre outros periféricos.
Resistor 1K		Resistor PTH comum de 1/4W, +/- 5% de tolerância, excelentes pull-ups, pull-down e limitador de corrente.
Resistor 10K		Resistor PTH comum de 1/4W, +/- 5% de tolerância, excelentes pull-ups, pull-down e limitador de corrente.
Trinco		Trinco tarjeta de três polegadas, usado normalmente para trancar todo tipo de portas.

Modulo Bluetooth HC06		Possui protocolo V2.0 + EDR, com frequência de 2.4 Ghz e banda ISM e modulação GFSK, entre outros periféricos.

Fonte: Organizado pelos autores.

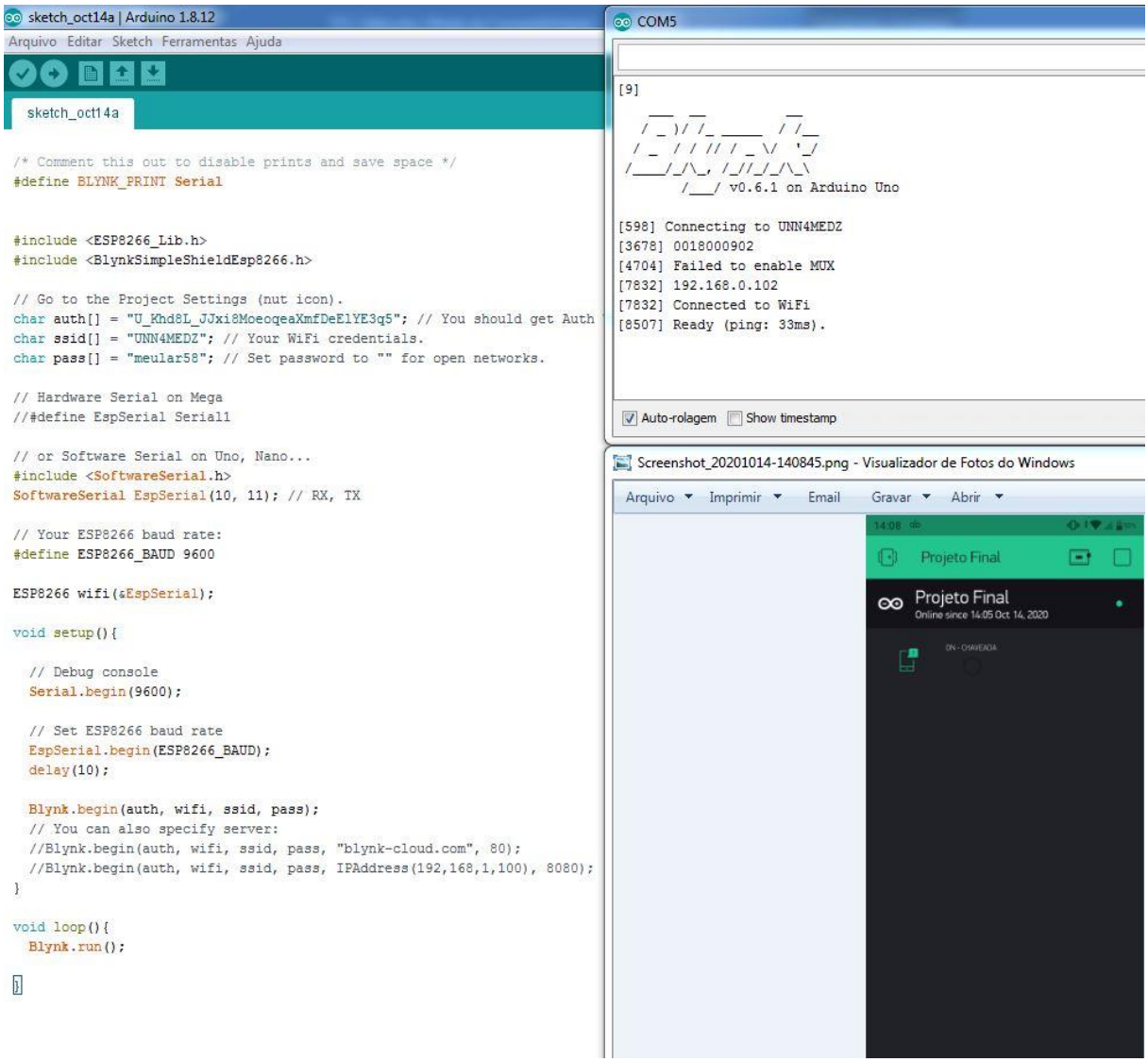
3.5 PROGRAMAÇÃO NA IDE DO ARDUINO E SINCRONIZAÇÃO

Na parte de programação, onde será o centro de todo o processo, foi usado a própria IDE do Arduino, junto com o pacote de bibliotecas do aplicativo Blynk, e com o uso do mesmo uma implementação de bom entendimento e com fácil modificação para a inserção de outros dispositivos.

O intuito foi fazer com que o código inserido no Arduino fosse reconhecido através da plataforma wi-fi e bluetooth com os widget do aplicativo Blynk, no smartphone.

A parte de conexão com a placa ESP8266 deu-se pelo código na figura a seguir junto a um Token para a melhor segurança, assim sem ter que se preocupar com conexões de outras pessoas ao aplicativo, e com isso melhorando a segurança do usuário. Foram usados os pinos digitais 10 e 11 do Arduino para a conexão com o sensor wi-fi, ficando responsável do RX e o outro pelo TX da placa ESP8266.

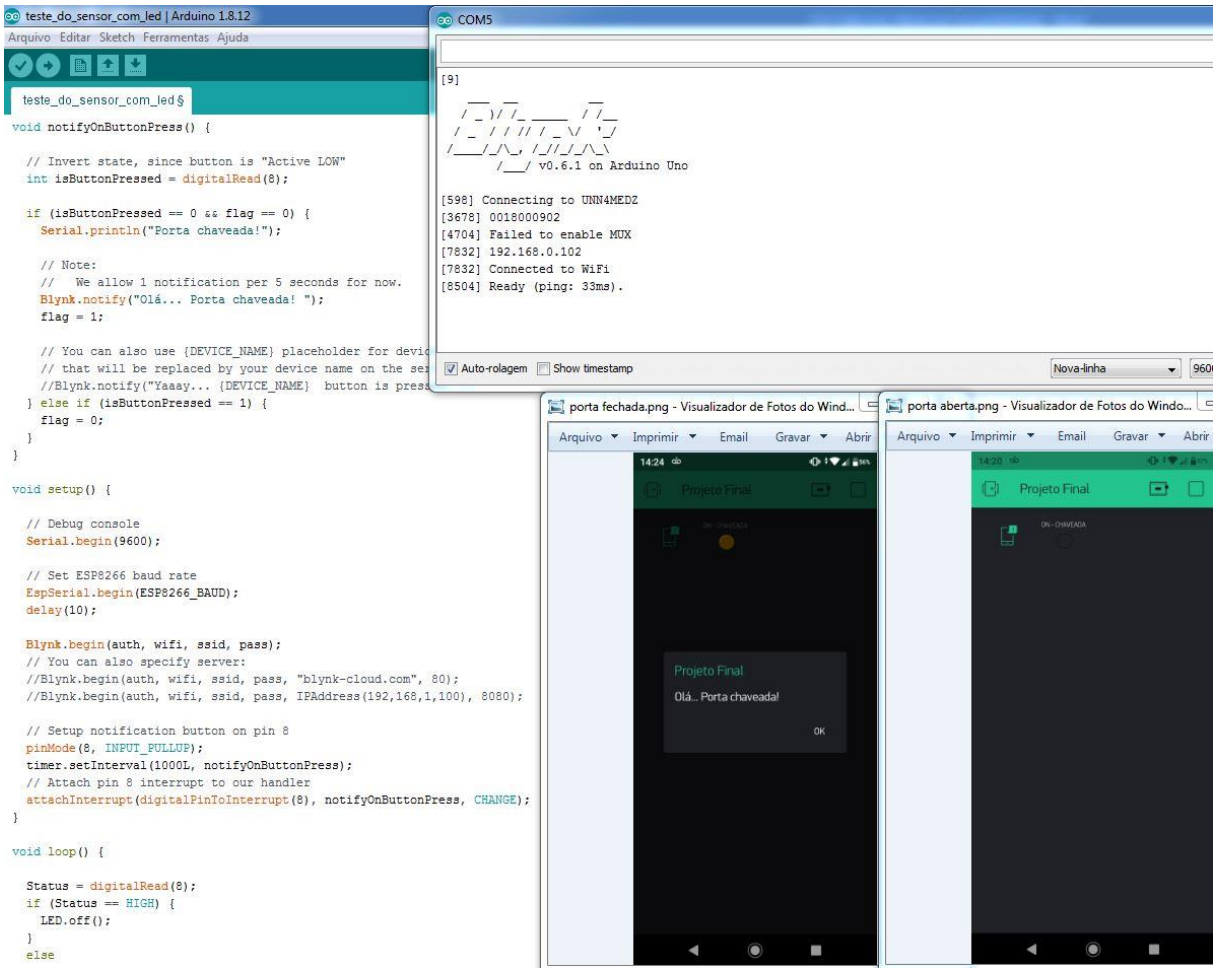
Figura 13 – Código de conexão com ESP8266



Fonte: Autores.

Já para a parte da configuração do sensor indutivo, foi desenvolvido o código de maneira que a entrada do sinal estivesse na seleção do pino digital 8 do Arduino, e quando o lingote da fechadura fosse chaveado, esse metal ficaria em contato com o sensor, assim enviando uma mensagem para o aplicativo do smartphone informando que a porta foi chaveada. Também foi posto um widget led em um pino virtual V8 do próprio aplicativo, para informar se a porta realmente está chaveada. Em caso de led “on” a porta está chaveada e a mensagem será enviada no exato momento, e em caso do led “off”, isso informa ao usuário que a porta está destrancada, ocasionada por um eventual esquecimento de ter chaveado. Na figura a seguir a implementação do código e a sincronização com o aplicativo.

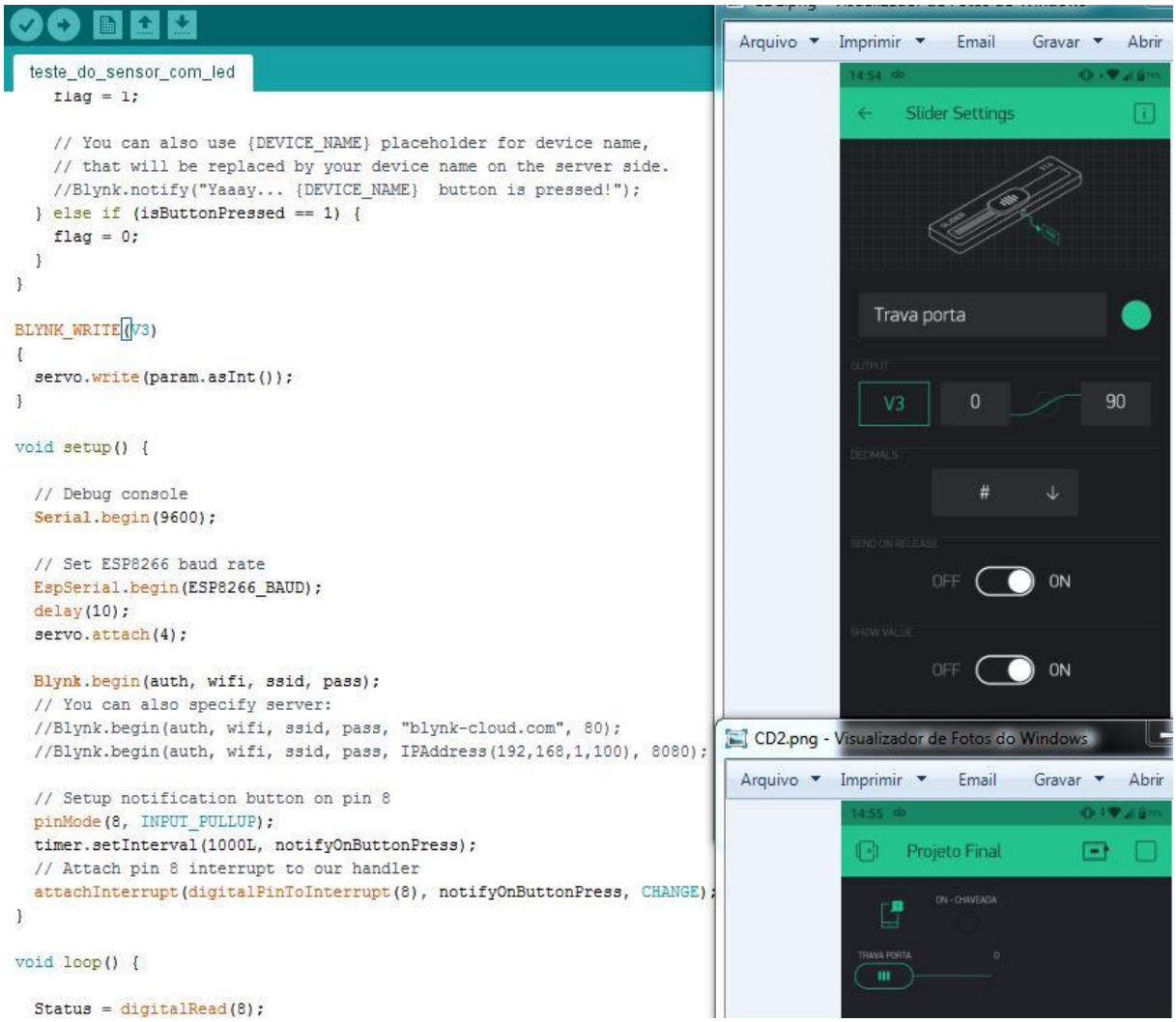
Figura 14 – Código de conexão com ESP8266 e Sensor



Fonte: Autores.

Porém, pensando no usuário que se esqueceu de chavear a porta e na sua comodidade de ter que se deslocar de onde estiver naquele momento para poder assim chavear a sua porta, surgiu a ideia de implementar um controlador para um servo motor. Assim com a adição de outro widget, um slider no pino virtual V3 do aplicativo para controlar esse servo entre 0º a 90 º graus, com a leitura/escrita desse sinal no pino digital 4 do Arduino, e através dessa conexão o acionamento de uma tranca, sem a necessidade de usar uma trava elétrica, tornando o projeto como o princípio, que é baixo custo. Na figura a parte de implementação do código para essa conexão com o widget do aplicativo.

Figura 15 – Código de implementação do servo motor



Fonte: Autores.

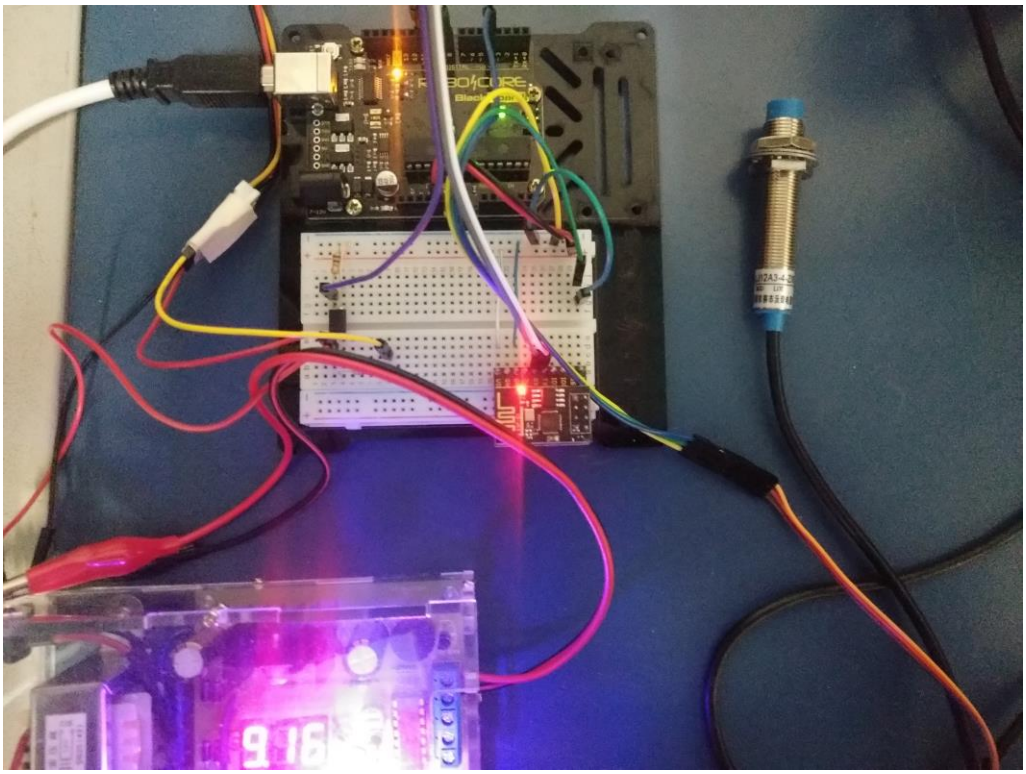
E por fim, o código final que será adicionado no apêndice desse trabalho, finalizando a implementação entre o Arduino e o aplicativo Blynk. Claro que o modelo apresentado acima foi baseado na implementação de modelo wi-fi. No apêndice do trabalho como já citado, também será inserido o código do modelo bluetooth. Assim o usuário poderá escolher o modelo que melhor lhe convém, pois ambos são voltados para um projeto de baixo custo.

3.6 MONTAGEM E PARTE PRÁTICA DO PROJETO

Para a montagem prática foi utilizado um protótipo de porta, construído para que se referisse a uma porta convencional, como na maioria das residências hoje em dia, com um

modelo de fechadura comum. Assim que posicionado o lingote próximo ao sensor, e no momento que a porta fosse chaveada, o lingote que possui material ferroso, ficaria no campo magnético gerado pelo sensor e com isso entrando em condução, saturando o led e acionando o transistor fotoelétrico, isso tudo no PC815. Já na tensão de 5V foi reduzido com o resistor R2 a tensão na entrada do sinal no pino digital do Arduino. Mesmo que fosse aumentar a tensão no sensor até seu limite, não teria problema de danificar o microcontrolador, e com esse sinal foi trabalhado através da implementação já citada. O sinal vai ser transmitido para o aplicativo, no caso da porta chaveada uma mensagem vai ser enviada para o usuário e um LED "on", informando que a porta está realmente chaveada. E em caso do LED "off" a porta estará destrancada. Claro que fica a critério do usuário usar o slider do aplicativo para acionar a tranca ou não, ou se no caso quer se deslocar até onde o sistema foi instalado.

Figura 16 – Montagem no Arduino do modelo wi-fi



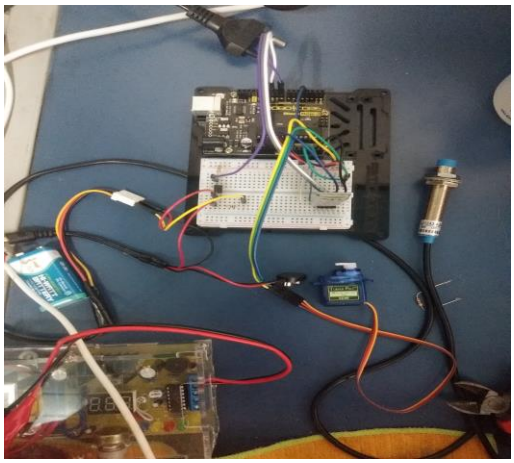
Fonte: Autores.

Figura 17 – Posicionamento do sensor

Fonte: Autores.

Após a montagem no Arduino e o sensor já posto no local definido, os dados coletados foram como o esperado. O sensor reagiu bem ao lingote da fechadura, o microcontrolador e o aplicativo se portaram de maneira esperada e satisfatória, isso no modelo wi-fi, pois o intuito vai ser o acesso do aplicativo a longa e curta distância. Porém para testes, a inserção do outro modelo, como na figura a seguir, ou seja, o modelo bluetooth, ficou limitado só para pequenas distâncias, aproximadamente 10m, e isso dificultaria para o usuário da conexão a longa distância, tornado o projeto mais custoso, pois iria necessitar a implementação de outros componentes, assim não tornando esse projeto viável com proposta de baixo custo.

Figura 18 – Montagem Arduino modelo bluetooth



Fonte: Autores.

Já na etapa final e para uma apresentação mais estética do projeto, foi adequada a ele uma pequena caixa, tirando aquela visibilidade mais grosseira. Posicionado sobre a porta, dentro dele o Arduino alimentado por uma fonte externa de 9V/1A, uma bateria interna de 9V para o sensor indutivo e mais quatro baterias de 1,5V em série para o micro servo. Com isso alimentando o sistema por partes, cada um com a tensão e corrente necessária.

Figura 19 – Visão externa do projeto

Fonte: Autores.

Figura 20 – Visão interna do projeto

Fonte: Autores.

4 CONSIDERAÇÕES FINAIS

Nessa etapa final da pesquisa, entendemos que este foi um processo que nos possibilitou muitos estudos e aprendizagens. Entendemos, também, que estamos sempre em movimento de aprendizagem e, assim, nos apropriamos de conhecimentos novos e evoluímos a cada instante.

Este trabalho tinha como objetivo desenvolver um circuito de baixo custo para fechaduras de portas residenciais com a plataforma Arduino, que possa fornecer aos usuários a comodidade de saber se a porta está aberta ou fechada através de um aplicativo de smartphone. Com isso, de forma que contemplasse o objetivo geral, traçamos os objetivos específicos: criar um circuito base; realizar a conexão do sensor indutivo com o Arduino; implementar um aplicativo livre pelo smartphone que possa se comunicar com o Arduino e realizar o controle de verificação do sensor. Assim o trabalho foi desenvolvido em seis etapas.

A partir das etapas desenvolvidas, concluímos que em geral o projeto funcionou como projetado. Porém ressaltamos que o aplicativo no qual foi usado neste trabalho, não foi desenvolvido pelo pesquisador, o que nos deixa ainda algumas alternativas para melhorar nesse aspecto, desde o desenvolvimento próprio de um aplicativo e um sistema de segurança mais confiável. Mas, como o projeto foi se baseado em baixo custo e com os mesmos limitados, e para estudo e desenvolvimento prático deste trabalho, o aplicativo livre mostrou-se necessário/benéfico e de ótima utilidade.

Em relação ao Hardware, concluímos que para uma melhoria futura, ainda seria necessário a troca de alguns componentes, como o Arduino Uno R3 por um NodeMCU V3, pois nele já vem integrado a placa wi-fi, e também pelo fato da placa ser menor, isso nos possibilitaria uma adaptação mais compacta.

Já na parte do sensor indutivo uma melhora seria trocar o sensor LJ12A por um sensor menor, ou um sensor chave fim de curso, no qual nos informará em qual das duas etapas o lingote se encontra, e com isso nos possibilitaria a cada fase do lingote uma nova função.

Contudo ainda poderíamos ajustar outras melhoras, além dessas já citadas, em vez de usar um micro servo de acionamento de tranca, poderíamos construir uma trava já dentro da porta, acionadas por um servo mais forte, uma trava elétrica ou até mesmo pelo uso da digital integrando um sistema RFID.

E por fim, uma melhoria na distribuição de energia, em vez de fontes separadas, uma bateria de 12V, ajustada com controladores de tensão, assim cada parte do sistema seguira recebendo a tensão e corrente necessária.

Com relação ao software, os maiores aprendizados com o mesmo, no caso a IDE do próprio Arduino, nos possibilitou uma ampla biblioteca e um fácil entendimento para quem for interpretar o código, ou pelo simples fato de implementações futuras, como por exemplo, a inserção de outros componentes elétricos.

Destacamos que a elaboração desse projeto foi bastante desafiadora e promoveu um ganho significativo de conhecimento teórico-prático, conhecimentos adquiridos durante o decorrer do curso de Engenharia de Computação. Lembramos que o trabalho foi o desenvolvimento de um sistema automatizado e controle para as portas, mas não limitando apenas a isso, visando atender usuários que esquecem certas coisas facilmente, assim amenizando preocupações desnecessárias.

É importante ressaltar que os objetivos propostos nesse referido trabalho foram atingidos na medida do possível, como o fato do desenvolvimento desse projeto, que levou mais tempo do que o necessário devido à circunstância atual e também por falta de alguns recursos, mesmo assim tentamos apresentar um trabalho que possa contribuir para outros estudos e com possibilidades de melhorias futuras e aberta a sugestões.

Contudo, a elaboração do trabalho exigiu um alto rigor técnico e metodológico, presentes desde o tema até o levantamento de requisitos, elaboração de um esquema elétrico compacto e escolha dos componentes até o projeto no estado final. Essas etapas foram fundamentais para apresentar um produto de relevância na área da tecnologia, e com isso contribuindo para novos estudos em sala de aula ou até mesmo em seu lar, possibilitando aos estudantes usar essas aplicações em trabalhos futuros. E também a partir dele, deixo como sugestão para o curso de Engenharia de Computação, que promova e estimule esse tipo de desenvolvimento, ou até mesmo outras tecnologias similares aos alunos, porém ligados a um maior estudo prático para um aperfeiçoamento dos trabalhos acadêmicos.

Por fim, espera-se que esse trabalho possa contribuir de alguma forma para uma maior difusão a respeito do tema proposto, possibilitando o surgimento de novas ideias e sugestões de melhorias. Também deixo aqui o meu agradecimento final a todos os educadores, amigos e colegas que passaram por mim e todo conhecimento construído no decorrer da minha graduação, meu muito obrigado.

REFERÊNCIAS

ALLDATASHEET. **PC817**. Disponível em: <https://www.alldatasheet.com/view.jsp?Searchword=Pc817&gclid=Cj0KCQjwzZj2BRDVARIsABs3l9JpaEffA53M_w_KA76frBh0v3IosHbo8NMrNzl08RlPHPWxQsMdwQsaAoa-EALw_wcB>. Acesso em 02 ago. 2020.

ALLDATASHEET. **Servo Motor SG90**. Disponível em: <http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf>. Acesso em 02 ago. 2020.

ARDUINO. **Software**. 2020. Disponível em: < http://arduino.cc/en/main/software>. Acesso em: 1 mai. 2020.

BANZI, M. **Primeiros passos com Arduino**. São Paulo: Novatec Editora, 2011.

BLYNK. **Intro**. 2020. Disponível em: < http://docs.blynk.cc/>. Acesso em: 3 mai. 2020.

BOLZANI, C. A. M. **Residências Inteligentes**. Rio de Janeiro: Livraria da Física, 2004.

BRITO, J. L. G. de. **Sistema para monitoramento de consumo de energia elétrica particular, em tempo real e não invasivo utilizando a tecnologia Arduino**. 2016. 106 p. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade Estadual de Londrina, Londrina, PR, 2016.

EMBARCADOS. **Introdução ao Blynk App**. Disponível em: <https://www.embarcados.com.br/introducao-ao-blynk-app/>. Acesso em: 26 jul. 2020.

FLOP, F. **Adaptador Módulo Wifi Esp8266 Esp-01 para Protoboard.** Disponível em: <https://www.filipeflop.com/produto/adaptador-modulo-wifi-esp8266-esp-01-para-protoboard/#tab-description>. Acesso em: 27 jul. 2020.

FLOP, F. **Tutorial Módulo Wireless ESP8266 com Arduino.** Disponível em: <https://www.filipeflop.com/blog/esp8266-arduino-tutorial/>. Acesso em: 27 jul. 2020.

INSTITUTO DIGITAL. **Microcontrolador ATmega328p-PU. 2020**. Disponível em: <https://www.institutodigital.com.br/pd-5666bf-microcontrolador-atmega328p-pu.html>. Acesso em: 15 abr. 2020.

KENSKI, V. M. **Educação e tecnologias:** O novo ritmo da informação. 3. ed. Campinas, SP: Papirus, 2008.

LIMA, C. B. de. **AVR e Arduino**: técnicas de projeto. 2. ed. Florianópolis: Ed. dos autores, 2012.

MCROBERTS, M. **Arduíno básico**. São Paulo: Novatec Editora, 2011.

MECAWEB. **Sensores Indutivos**. Disponível em: <http://www.mecaweb.com.br/eletronica/content/e_sensor_indutivo>. Acesso em 30 jul. 2020.

NCB. **Conhecendo o Cerne do Microcontrolador Atmega328p Arduino Uno**. Disponível em: <https://www.newtoncbraga.com.br/index.php/microcontrolador/138-atmel/14863-conhecendo-o-cerne-do-microcontrolador-atmega328p-arduino-uno-mic165>. Acesso em: 12 jun. 2020.

SANTOS, A. **Servomotores**. Disponível em: <http://www.pictronics.com.br/downloads/apostilas/servomotores.pdf>. Acesso em: 04 ago. 2020.

SILVA, L. R. da. **Regulador de temperaturas para chuveiros elétricos**. 2014. 63 p. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Universidade Federal de Santa Maria, Santa Maria, RS, 2014.

THOMAZINI, D.; ALBUQUERQUE, P. U. B. de. **Sensores Industriais:** Fundamentos e Aplicações. 8. ed. São Paulo, SP, 2011.

YALE. **A história da Yale**. Disponível em: <https://www.yale.com.br/pt-br/sobre-a-yale/nossa-marca/>. Acesso em: 26 mar. 2020.

APÊNDICES

Apêndice A – Código do Modelo Wi-fi

teste_do_sensor_com_led

```
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial


#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
#include <Servo.h>

// Go to the Project Settings (nut icon).
char auth[] = "U_Khd8L_JJxi8MoeoqeaXmfDeElYE3q5"; // You should get Auth Token in the Blynk App.
char ssid[] = "UNN4MEDZ"; // Your WiFi credentials.
char pass[] = "meular58"; // Set password to "" for open networks.

// Hardware Serial on Mega
//#define EspSerial Serial1

// or Software Serial on Uno, Nano...
#include <SoftwareSerial.h>
SoftwareSerial EspSerial(10, 11); // RX, TX

// Your ESP8266 baud rate:
#define ESP8266_BAUD 9600

ESP8266 wifi(&EspSerial);
Servo servo;
WidgetLED LED(V8);
BlynkTimer timer;
int flag = 0;
int Status = 0;

void notifyOnButtonPress() {

  // Invert state, since button is "Active LOW"
  int isButtonPressed = digitalRead(8);

  if (isButtonPressed == 0 && flag == 0) {
    Serial.println("Porta chaveada!");

    // Note:
    //   We allow 1 notification per 5 seconds for now.
    Blynk.notify("Olá... Porta chaveada! ");
    flag = 1;

    // You can also use {DEVICE_NAME} placeholder for device name,
    // that will be replaced by your device name on the server side.
    //Blynk.notify("Yaaay... {DEVICE_NAME}  button is pressed!");
  } else if (isButtonPressed == 1) {
    flag = 0;
```

```
  }
}

BLYNK_WRITE(V3)
{
  servo.write(param.asInt());
}

void setup() {

  // Debug console
  Serial.begin(9600);

  // Set ESP8266 baud rate
  EspSerial.begin(ESP8266_BAUD);
  delay(10);
  servo.attach(4);

  Blynk.begin(auth, wifi, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, wifi, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, wifi, ssid, pass, IPAddress(192,168,1,100), 8080);

  // Setup notification button on pin 8
  pinMode(8, INPUT_PULLUP);
  timer.setInterval(1000L, notifyOnButtonPress);
  // Attach pin 8 interrupt to our handler
  attachInterrupt(digitalPinToInterrupt(8), notifyOnButtonPress, CHANGE);
}

void loop() {

  Status = digitalRead(8);
  if (Status == HIGH) {
    LED.off();
  }
  else
  {
    LED.on();
  }
  Blynk.run();
  timer.run();
}
```

Apêndice B – Código do Modelo Bluetooth

sketch_oct25a §

```
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial



#include <Servo.h>

// Go to the Project Settings (nut icon).
char auth[] = "U_Khd8L_JJxi8MoeoqeaXmfDeElYE3q5"; // You should get Auth Token in the Blynk App.


// Hardware Serial on Mega
//#define EspSerial Serial1

// or Software Serial on Uno, Nano...
#include <SoftwareSerial.h>
SoftwareSerial SwSerial(10, 11); // RX, TX

Servo servo;
WidgetLED LED(V8);
BlynkTimer timer;
int flag = 0;
int Status = 0;

void notifyOnButtonPress() {

  // Invert state, since button is "Active LOW"
  int isButtonPressed = digitalRead(8);

  if (isButtonPressed == 0 && flag == 0) {
    Serial.println("Porta chaveada!");

    // Note:
    //   We allow 1 notification per 5 seconds for now.
    Blynk.notify("Olá... Porta chaveada! ");
    flag = 1;

    // You can also use {DEVICE_NAME} placeholder for device name,
    // that will be replaced by your device name on the server side.
    //Blynk.notify("Yaaay... {DEVICE_NAME}  button is pressed!");
  } else if (isButtonPressed == 1) {
    flag = 0;
  }
}

BLYNK_WRITE(V3)
{
  servo.write(param.asInt());
}
```

```
void setup() {

  // Debug console
  Serial.begin(9600);

  // Set ESP8266 baud rate
  EspSerial.begin(SerialBLE, auth);
  delay(10);
  servo.attach(4);



  // Setup notification button on pin 8
  pinMode(8, INPUT_PULLUP);
  timer.setInterval(1000L, notifyOnButtonPress);
  // Attach pin 8 interrupt to our handler
  attachInterrupt(digitalPinToInterrupt(8), notifyOnButtonPress, CHANGE);
}

void loop() {

  Status = digitalRead(8);
  if (Status == HIGH) {
    LED.off();
  }
  else
  {
    LED.on();
  }
  Blynk.run();
  timer.run();
}
```