

FACULDADE DE TECNOLOGIA DE SÃO PAULO – FATEC-SP

DEPARTAMENTO DE SISTEMAS ELETRÔNICOS

DENIMAR SOUZA DE OLIVEIRA

LEO MASAHARU AMAYA KITAGAWA

VICTOR ARTHUR RAVANI

**SISTEMA DE IDENTIFICAÇÃO E CONTROLE DE
CÉDULAS MONETÁRIAS**

SÃO PAULO

2020



DENIMAR SOUZA DE OLIVEIRA
LEO MASAHARU AMAYA KITAGAWA
VICTOR ARTHUR RAVANI

SISTEMA DE IDENTIFICAÇÃO E CONTROLE DE
CÉDULAS MONETÁRIAS

Trabalho de Conclusão de Curso, apresentado ao Departamento de Sistemas Eletrônicos, da Faculdade de Tecnologia de São Paulo – FATEC-SP, como requisito para a conclusão da graduação para o curso de Tecnologia em Eletrônica Industrial.

Orientador: Prof. Me. Maurício Antônio Deffert

SÃO PAULO

2020

Ficha Catalográfica

De Oliveira, Denimar; Kitagawa, Leo; Ravani, Victor.
SISTEMA DE IDENTIFICAÇÃO E CONTROLE DE CÉDULAS
MONETÁRIAS / Denimar Souza de Oliveira, Leo Masaharu
Amaya Kitagawa, Victor Arthur Ravani – 2020.152p. Monografia
– Curso Superior de Tecnologia em Eletrônica Industrial –
Faculdade de Tecnologia de São Paulo, 2020.
Orientador: Prof. Me. Maurício Antonio Deffert
1. Cédulas monetárias 2. Sensor RGB 3. Cofre

DENIMAR SOUZA DE OLIVEIRA
LEO MASAHARU AMAYA KITAGAWA
VICTOR ARTHUR RAVANI

Sistema de identificação e controle de cédulas monetárias

Aprovada em: 18 /12 /2020

Dr. Victor Sonnenberg
Coordenador do curso / Chefe do Departamento de Sistemas Eletrônicos

Me. Maurício Antônio Deffert
Orientador

Dr. Leonardo Frois Hernandez
Avaliador

Me. Ricardo Cardoso Rangel
Avaliador

(DEDICATÓRIA)

Dedicamos este trabalho às nossas famílias e a todos que de alguma forma
contribuíram para a conclusão deste curso

AGRADECIMENTOS

Ao nosso orientador, Prof. Me. Maurício Antônio Deffert, pelo incentivo e auxílio dado ao longo de todo este trabalho.

Aos professores, pelas aulas ministradas e por todo o conhecimento que nos foi oferecido, que certamente contribuíram para nosso crescimento pessoal e profissional.

Aos colegas de classe, pelo conhecimento compartilhado e auxílio nos momentos de dificuldade.

Às nossas famílias, pelo apoio nos momentos difíceis e incentivos dados para a conclusão deste curso.

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis”.

José de Alencar (1829 - 1877)

RESUMO

O dinheiro em espécie é ainda muito utilizado nos tempos atuais. Seja para compras, pagamento de contas ou mesmo para se guardar, a forma física do dinheiro possui grande relevância para boa parte da população brasileira. Saber como poupar, controlar e planejar os gastos é essencial. Por esta razão, propõe-se neste trabalho a elaboração de um protótipo de cofre com sistema microcontrolado, para identificação e controle de cédulas monetárias como forma de educação financeira. Este cofre dará destaque à questão de poupar, permitindo assim desenvolver futuramente uma educação financeira, tanto para crianças como para adultos. Para a identificação das cédulas, utilizou-se um sensor de cor, o RGB TCS 230, que é responsável por verificar individualmente cada cédula inserida e contabilizar o valor na memória EEPROM do microcontrolador. O saldo é apresentado no visor de um *display* LCD. O controle de abertura e fechamento do cofre é feito por meio da digitação de senha em um teclado matricial. Este cofre também foi projetado para ter acesso via interface gráfica em um computador para verificar o saldo e abrir o cofre de forma remota, utilizando um conversor USB serial TTL CH340. A caixa do cofre foi feita com madeira MDF, um material leve e de fácil manuseio, permitindo a realização de cortes com boa precisão. Em testes de confiabilidade da leitura das cédulas com o sensor RGB, houve uma taxa de acerto de aproximadamente 90,5%, com 76 acertos e 8 erros, utilizando três exemplares de cada cédula (2 a 200 reais) e realizando a leitura em quatro pontos distintos das cédulas. A leitura do sensor foi feita em cédulas recentes, sem manchas ou rasgos, pois cédulas mais antigas ou com defeitos dificultam a leitura do sensor utilizado. Nos testes de senhas, a abertura do cofre foi possível somente com a senha programada pelo usuário, reforçando a ideia de segurança do cofre desenvolvido.

Palavras-chave: cofre eletrônico, identificação de cédulas do sistema monetário brasileiro, micro controlador, sensor RGB, sistemas controlados por senha

ABSTRACT

Cash is still widely used today. Whether for purchases, payment of bills or even for storage, the physical form of money has great relevance for a good part of the Brazilian population. Knowing how to save, control and plan spending is essential. For this reason, it is proposed in this work to elaborate a prototype of a safe with microcontrolled system, for identification and control of banknotes as a form of financial education. This safe will highlight the issue of saving, thus allowing you to develop financial education for both children and adults in the future. For the identification of the banknotes, a color sensor, RGB TCS 230 was used, which is responsible for individually checking each inserted banknote and counting the value in the microcontroller EEPROM memory. The balance is shown on the LCD display. The opening and closing control of the safe is done by typing a password on a matrix keyboard. This safe is also designed to be accessed via a graphical interface on a computer to check the balance and open the safe remotely, using a TTL CH340 serial USB converter. The safe box was made with MDF wood, a light material and easy to handle, allowing cuts to be made with good precision. In banknotes' reading reliability tests, with the RGB sensor, there was a hit rate of approximately 90.5%, with 76 hits and 8 errors, using three copies of each banknotes of the Second Real Family (2 to 200 reais) and reading in four different points of the banknotes. The sensor reading was made on recent banknotes, without spots or rips, because older banknotes or with defects make the sensor's reading difficult. In password tests, the safe's opening was only possible with the user's programmed password, reinforcing the safety idea of the safe developed.

Keywords: electronic safe, identification of banknotes of the Brazilian monetary system, microcontroller, RGB sensor, password-controlled systems

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1: Uso do dinheiro: dados da pesquisa Banco do Brasil | 16 |
| Figura 2: Uso do dinheiro: dados da pesquisa Instituto Locomotiva | 17 |
| Figura 3: Arquitetura Von Neumann..... | 22 |
| Figura 4: Arquitetura Harvard..... | 22 |
| Figura 5: Microcontrolador PIC 16F877A e gravador Pickit3 | 23 |
| Figura 6: Comunicação serial PIC 16F877A e computador | 26 |
| Figura 7: Espectro visível da luz..... | 28 |
| Figura 8: Sistemas CMYK e RGB | 29 |
| Figura 9: Cédulas monetárias da Primeira Família do Real | 30 |
| Figura 10: Cédulas monetárias da Segunda Família do Real | 31 |
| Figura 11: Cédula monetária de 2 reais | 31 |
| Figura 12: Cédula monetária de 5 reais | 32 |
| Figura 13: Cédula monetária de 10 reais | 32 |
| Figura 14: Cédula monetária de 20 reais | 32 |
| Figura 15: Cédula monetária de 50 reais | 32 |
| Figura 16: Cédula monetária de 100 reais | 33 |
| Figura 17: Cédula monetária de 200 reais | 33 |
| Figura 18: Item 1 - Marca d'água das cédulas do real..... | 34 |
| Figura 19: Item 2 - Pontos de alto relevo nas cédulas do real..... | 34 |
| Figura 20: Item 3 - Número escondido nas cédulas do real | 35 |
| Figura 21: Item 4 - Fio de segurança nas cédulas do real | 35 |
| Figura 22: Item 5 - Quebra-cabeça nas cédulas do real | 35 |
| Figura 23: Item 6 - Elementos fluorescentes na cédula do real..... | 36 |
| Figura 24: Cofre mecânico | 37 |
| Figura 25: Cofre embutido..... | 37 |
| Figura 26: Cofre biométrico | 38 |
| Figura 27: Cofre eletrônico..... | 38 |
| Figura 28: Sensor RGB TCS230 | 40 |
| Figura 29: Pinagens do sensor RGB TCS230..... | 41 |
| Figura 30: Emissor e receptor infravermelho..... | 42 |
| Figura 31: Display LCD 16x2..... | 43 |
| Figura 32: Circuito elétrico de um teclado matricial..... | 44 |
| Figura 33: Teclado matricial 4x3 | 45 |
| Figura 34: Desenho esquemático de um servo motor..... | 46 |
| Figura 35: Servo motor SG-90 | 47 |
| Figura 36: Conversor USB serial TTL CH340 | 47 |
| Figura 37: Fechadura solenoide eletrônica | 48 |
| Figura 38: Diagrama de blocos do regulador de tensão..... | 49 |
| Figura 39: Circuito regulador de tensão com CI 7812 | 50 |
| Figura 40: Interface do programa MikroC..... | 51 |
| Figura 41: Interface do programa Microsoft Visual Studio..... | 52 |

| | |
|---|----|
| Figura 42: Programas Microsoft Office | 52 |
| Figura 43: Diagrama esquemático sobre o projeto do cofre..... | 54 |
| Figura 44: Circuito elétrico geral do projeto elaborado no programa Proteus .. | 55 |
| Figura 45: Placa desenvolvida no curso de Microcontroladores | 56 |
| Figura 46: Circuito esquemático básico para teste do sensor RGB | 57 |
| Figura 47: Caixa para leitura RGB. | 58 |
| Figura 48: Sistema de reconhecimento de cédulas..... | 59 |
| Figura 49: Interior da caixa para leitura RGB. | 60 |
| Figura 50: Desenho esquemático estimando local de leitura nas cédulas | 60 |
| Figura 51: Protótipo para leitura RGB de cédulas. | 61 |
| Figura 52: Fonte de alimentação 12 V / 3 A | 63 |
| Figura 53: Circuito de regulação 12 V para 5 V..... | 63 |
| Figura 54: Circuito para teste do sistema servo motor | 64 |
| Figura 55: Circuito para teste do teclado matricial | 65 |
| Figura 56: Circuito para teste do sistema com trava eletrônica..... | 66 |
| Figura 57: Menu para senhas..... | 67 |
| Figura 58: Circuito para teste de comunicação serial..... | 68 |
| Figura 59: Circuito elétrico da placa para comunicação serial | 68 |
| Figura 60: <i>Layout</i> da placa para comunicação serial | 69 |
| Figura 61: Imagem 3D da placa para comunicação serial | 69 |
| Figura 62: Placa desenvolvida finalizada | 70 |
| Figura 63: Interface gráfica do cofre identificador | 70 |
| Figura 64: Teste de leitura RGB (cédula de 2 e 5 reais). | 72 |
| Figura 65: Teste de leitura RGB (cédula de 10 e 20 reais). | 72 |
| Figura 66: Teste de leitura RGB (cédula de 50 e 100 reais) | 73 |
| Figura 67: Verificação dos valores RGB. (2, 5, 10, 20, 50 e 100 reais)..... | 75 |
| Figura 68: Calibração da cédula de 2 e 5 reais | 76 |
| Figura 69: Calibração das cédulas de 10 e 20 reais | 77 |
| Figura 70: Calibração da cédula de 50 e 100 reais | 77 |
| Figura 71: Teste do sistema do servo motor | 78 |
| Figura 72: Teste com teclado matricial..... | 79 |
| Figura 73: Testes para salvar dados na memória EEPROM..... | 80 |
| Figura 74: Teste da trava solenoide com senha..... | 81 |
| Figura 75: Teste para leitura na interface gráfica (teste <i>string</i>) | 82 |
| Figura 76: Teste para abrir porta na interface gráfica (teste LED) | 82 |
| Figura 77: Esboço da estrutura do cofre para o projeto | 83 |
| Figura 78: Estrutura física inicial do cofre | 84 |
| Figura 79: Cofre pintado..... | 84 |
| Figura 80: Placa do microcontrolador e placa fonte para o cofre | 85 |
| Figura 81: Parte de circuitos do cofre..... | 86 |
| Figura 82: Parte de armazenagem do cofre | 87 |
| Figura 83: Imagem geral do cofre | 87 |
| Figura 84: Teste de leitura RGB (cédula de 200 reais). | 88 |
| Figura 85: Verificação da leitura RGB (cédula de 200 reais). | 88 |
| Figura 86: Calibração da cédula de 200 reais | 89 |
| Figura 87: Tela de apresentação do cofre..... | 90 |
| Figura 88: Tela de seleção do cofre | 91 |

| | |
|---|-----|
| Figura 89: Tela de menu do cofre | 91 |
| Figura 90: Tela de configuração de senha | 92 |
| Figura 91: Tela de saldo do cofre..... | 92 |
| Figura 92: Tela de configuração de senha | 93 |
| Figura 93: Cédula inserida no cofre | 93 |
| Figura 94: Identificação da cédula no cofre..... | 94 |
| Figura 95: Retirada do dinheiro do cofre | 94 |
| Figura 96: Posições das cédulas para teste de leitura | 95 |
| Figura 97: Erros das cédulas no teste de leitura | 99 |
| Figura 98: Conexão do cofre com um computador | 101 |
| Figura 99: Acesso ao cofre via computador | 101 |
| Figura 100: Tela principal do cofre | 102 |
| Figura 101: Verificação do saldo do cofre na interface gráfica e no cofre | 103 |
| Figura 102: Teste de abertura de porta na interface gráfica | 103 |

LISTA DE TABELAS

| | |
|---|-----|
| Tabela 1: Dimensões das Cédulas da Segunda Família do Real | 33 |
| Tabela 2: Dados técnicos do PIC 16F877A | 39 |
| Tabela 3: Dados técnicos do Sensor TCS 230..... | 41 |
| Tabela 4: Pinagem de um display LCD | 43 |
| Tabela 5: CIs da série 7800 e tensões de saída reguladas | 50 |
| Tabela 6: Dados da leitura RGB das cédulas..... | 73 |
| Tabela 7: Valores corrigidos da leitura RGB nas cédulas | 75 |
| Tabela 8: Testes de leitura de cédulas – 2 reais | 96 |
| Tabela 9: Testes de leitura de cédulas – 5 reais | 96 |
| Tabela 10: Testes de leitura de cédulas – 10 reais | 96 |
| Tabela 11: Testes de leitura de cédulas – 20 reais | 97 |
| Tabela 12: Testes de leitura de cédulas – 50 reais | 97 |
| Tabela 13: Testes de leitura de cédulas – 100 reais | 97 |
| Tabela 14: Testes de leitura de cédulas – 200 reais | 98 |
| Tabela 15: Posições recomendadas para leitura de cédulas | 98 |
| Tabela 16: Testes de senhas | 100 |
| Tabela 17: Lista de materiais dos principais componentes do projeto | 104 |
| Tabela 18: Lista de materiais para montagem do cofre | 105 |
| Tabela 19: Lista de materiais do cofre (componentes secundários) | 106 |
| Tabela 20: Estimativa de tempo gasto com o projeto..... | 108 |

SUMÁRIO

| | |
|--|----|
| RESUMO..... | 7 |
| ABSTRACT | 8 |
| LISTA DE FIGURAS | 9 |
| LISTA DE TABELAS | 13 |
| 1. INTRODUÇÃO..... | 16 |
| 1.1. Objetivo geral | 19 |
| 1.2. Objetivos específicos | 19 |
| 1.3. Justificativa..... | 19 |
| 1.4. Organização deste trabalho | 19 |
| 2. FUNDAMENTAÇÃO TEÓRICA | 21 |
| 2.1. Microcontroladores..... | 21 |
| 2.1.1. Microcontrolador PIC..... | 21 |
| 2.1.1.1. Memórias PIC | 24 |
| 2.1.1.1.1. Memória PROM..... | 24 |
| 2.1.1.1.2. Memória EPROM | 24 |
| 2.1.1.1.3. Memória EEPROM..... | 25 |
| 2.1.1.2. Comunicação serial (USART) | 25 |
| 2.2. Sensores e Atuadores..... | 27 |
| 2.3. Cor | 28 |
| 2.3.1. Sistemas RGB e CMY | 29 |
| 2.4. Cédulas monetárias do sistema financeiro brasileiro | 30 |
| 2.4.1. Cédulas da Segunda Família do Real | 31 |
| 2.4.2. Critérios de segurança das notas da Segunda Família do Real... 34 | |
| 2.5. Cofres..... | 36 |
| 2.5.1. Cofre mecânico..... | 36 |
| 2.5.2. Cofre embutido | 37 |
| 2.5.3. Cofre biométrico | 37 |
| 2.5.4. Cofre eletrônico | 38 |
| 3. MATERIAIS E MÉTODOS | 39 |
| 3.1. MATERIAIS UTILIZADOS NO PROJETO..... | 39 |
| 3.1.1. Microcontrolador PIC 16F877A..... | 39 |
| 3.1.2. Sensor RGB (TCS230) | 40 |
| 3.1.3. Sensor infravermelho (sensor TIL32 e fotodiodo TIL78)..... | 41 |

| | | |
|---------|---|-----|
| 3.1.4. | Display LCD (16x2)..... | 42 |
| 3.1.5. | Teclado matricial (4x3) | 44 |
| 3.1.6. | Servo motor (SG-90) | 45 |
| 3.1.7. | Conversor Serial (TTL CH340) | 47 |
| 3.1.8. | Trava eletrônica (fechadura solenoide 12 V) | 48 |
| 3.1.9. | Regulador de Tensão (CI 7805) | 49 |
| 3.1.10. | Softwares utilizados no projeto | 51 |
| 3.2. | METODOLOGIA UTILIZADA | 53 |
| 4. | TESTES E RESULTADOS PRELIMINARES..... | 54 |
| 4.1. | Procedimentos adotados para testes | 54 |
| 4.1.1. | Projeto inicial e simulações | 54 |
| 4.1.2. | Desenvolvimento de um protótipo básico para leitura RGB | 57 |
| 4.1.3. | Teste para verificação da leitura RGB do sensor TCS230 | 61 |
| 4.1.4. | Identificação dos valores monetários das cédulas..... | 62 |
| 4.1.5. | Desenvolvimento do sistema do servo motor | 62 |
| 4.1.6. | Desenvolvimento do sistema com teclado matricial | 64 |
| 4.1.7. | Desenvolvimento do sistema com trava eletrônica..... | 65 |
| 4.1.8. | Desenvolvimento do sistema para comunicação serial | 67 |
| 4.2. | RESULTADOS PRELIMINARES | 72 |
| 4.2.1. | Resultados do teste de leitura das cédulas com sensor RGB | 72 |
| 4.2.2. | Resultados da identificação do valor das cédulas | 76 |
| 4.2.3. | Resultado do sistema desenvolvido com servo motor | 78 |
| 4.2.4. | Resultado do sistema desenvolvido com teclado matricial | 79 |
| 4.2.5. | Resultado do sistema desenvolvido com trava eletrônica | 81 |
| 4.2.6. | Resultado do sistema para comunicação serial..... | 81 |
| 5. | DESENVOLVIMENTO DO PROTÓTIPO FINALIZADO..... | 83 |
| 5.1. | Projeto do layout do cofre | 83 |
| 5.2. | Calibração da cédula de 200 reais..... | 88 |
| 5.3. | Teste de operação do cofre | 90 |
| 6. | RESULTADOS E DISCUSSÕES..... | 95 |
| 6.1. | Teste de confiabilidade de leitura de cédulas | 95 |
| 6.2. | Testes de confiabilidade do sistema de senhas | 99 |
| 6.3. | Teste da comunicação do cofre com computador..... | 100 |
| 6.4. | Estimativa de custo do projeto do cofre | 104 |
| 6.5. | Comentários gerais e considerações finais sobre o projeto | 107 |

| | |
|---|-----|
| 7. CONCLUSÕES..... | 109 |
| 8. TRABALHOS FUTUROS..... | 110 |
| 9. REFERÊNCIAS BIBLIOGRÁFICAS | 111 |
| ANEXO A – Código fonte do cofre microcontrolado..... | 114 |
| ANEXO B – Código fonte – Visual Studio (tela de login)..... | 148 |
| ANEXO C – Código fonte – Visual Studio (Comunicação)..... | 149 |

1. INTRODUÇÃO

Atualmente, existem diversos meios de utilização do dinheiro. Seja na sua forma física (em espécie) ou na forma de cartão (crédito ou débito), a escolha de utilização de um destes meios depende tanto da situação financeira de cada indivíduo quanto o próprio ambiente econômico do país. Algumas pesquisas realizadas recentemente destacam dados do uso do dinheiro que podem identificar hábitos da população brasileira.

Uma pesquisa do Banco Central do Brasil [1], do ano de 2018, apresenta detalhes referentes à relação dos brasileiros com o dinheiro. Dois gráficos desta pesquisa são apresentados na Figura 1:

Figura 1: Uso do dinheiro: dados da pesquisa Banco do Brasil



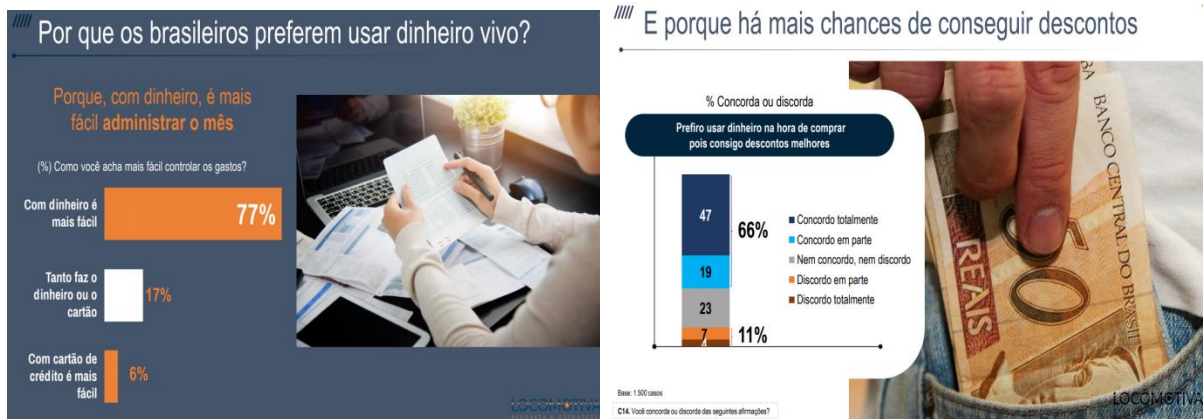
Fonte:

https://www.bcb.gov.br/content/cedulasemoedas/pesquisabrasileirodinheiro/Apresentacao_brasileiro_relacao_dinheiro_2018.pdf. Acesso em 13/11/2020

Do levantamento do Banco do Brasil, destaca-se que 96% dos entrevistados relataram que o dinheiro é a forma preferida para pagamento de contas ou para realizar compras. Outro dado relevante é que o dinheiro é a forma de pagamento mais aceita nos estabelecimentos (99%), em comparação ao cartão de crédito (76%) e débito (74%).

Outra pesquisa mais recente, feita pelo Instituto Locomotiva [2], em setembro de 2019, apresentou outros dados sobre o uso do dinheiro, desta vez voltado à preferência dos brasileiros pelo uso do dinheiro vivo. Dois gráficos desta pesquisa são mostrados na Figura 2:

Figura 2: Uso do dinheiro: dados da pesquisa Instituto Locomotiva



Fonte: <https://static.poder360.com.br/2019/10/Os-brasileiros-e-o-dinheiro-em-especie-Imprensa-Outubro-de-2019.pdf>. Acesso em 13/11/2020

Nesta pesquisa, 77% dos entrevistados relataram que com o dinheiro é mais fácil administrar os gastos. Além disso, 66% concordam totalmente ou em parte com a ideia de que utilizando dinheiro é possível conseguir descontos melhores na hora da compra. [2]

A partir destas pesquisas, é possível entender por que o dinheiro físico é ainda a principal opção na vida de muitos brasileiros. Seja para se fazer compras, fazer o pagamento de contas ou administrar as contas do mês, o uso do dinheiro físico faz parte do dia a dia do brasileiro. No entanto, outra pergunta relevante a se fazer é se poupar dinheiro também é um hábito dos brasileiros.

Segundo a pesquisa do Serviço de Proteção ao Crédito (SPC Brasil) e da Confederação Nacional de Dirigentes Lojistas (CNDL), do ano de 2018, o brasileiro não tem o hábito de poupar. Nesta pesquisa, 73% dos consumidores não pouparam qualquer quantia. Neste mesmo ano, a média de valor poupado foi de R\$498,81. [3]

Na pesquisa do SPC Brasil algumas justificativas foram apresentadas para não se poupar dinheiro. Para 42%, a baixa renda é um fator que determina ser difícil

ter sobras nos rendimentos, 20% afirmam ter enfrentado alguma situação de imprevisto e 18% disseram não ter renda no momento. Nesta pesquisa, 11% admitem ter perdido o controle com gastos excessivos. [3]

Segundo a economista-chefe do SPC Brasil, Marcela Kawauti, pensar na reserva financeira apenas como uma sobra do orçamento é um erro, pois a quantia deve ser guardada com disciplina:

“Os apelos ao consumo estão por toda a parte e, se não houver determinação, dificilmente vai sobrar dinheiro no final do mês. A recomendação é separar a reserva financeira assim que o salário entrar na conta, fazendo disso um compromisso mensal”, comentou a economista.

Tendo este problema em vista, este trabalho busca fornecer um instrumento para a educação financeira, por meio de um cofre microcontrolado. Este, além de armazenar o dinheiro, terá funcionalidades que incentiva o hábito de poupar.

Existem muitos modelos de cofres no mercado, com diferentes tipos de tamanho, peso e material de fabricação. O preço de cada cofre varia principalmente com o nível de segurança desejado pelo usuário. Neste trabalho, decidiu-se pelo modelo eletrônico, que apresenta um bom nível de segurança e praticidade.

A ideia deste trabalho é projetar tanto a parte física do cofre quanto os circuitos eletrônicos para identificação automática do valor das cédulas e controle do cofre utilizando um sistema de segurança com senha. Para a parte de identificação, será abordado neste trabalho um sensor de cor, o sensor RGB.

Este sensor RGB trabalha com a identificação do padrão RGB dos corpos, podendo assim diferenciar uma cédula de outra, programando um microcontrolador para fazer esta comparação. Com relação à parte de segurança, será utilizado um sistema simples de digitação de uma senha em um teclado para abertura do cofre. Neste trabalho, será possível o acesso e controle do cofre também de forma remota, utilizando comunicação serial para fazer a ligação do cofre com um computador.

Assim, este trabalho tem como pergunta problema: como desenvolver um cofre eletrônico, de baixo custo e capaz de identificar cédulas para contabilizar automaticamente o valor armazenado. Observe-se que o uso de um cofre com capacidade de identificação de cédulas também apresenta cunho didático, apresentando ao usuário questões inerente ao modo de consumo.

1.1. Objetivo geral

Projetar e construir um protótipo de cofre para identificação de cédulas monetárias, utilizando um sistema microcontrolado. O *hardware* do projeto deverá ser o mais simples e intuitivo possível, facilitando sua compreensão.

1.2. Objetivos específicos

Deseja-se projetar um cofre que faça a identificação do valor das cédulas monetárias a partir da sua cor, utilizando sensor RGB. Um *display* LCD deverá apresentar o valor acumulado, além de opções de um menu para configuração do cofre. Um sistema com servo motor fará com que a cédula inserida no identificador caia dentro do cofre por ação da gravidade. Um teclado matricial tornará possível a digitação de senhas e configuração do cofre. Uma interface gráfica com comunicação serial será feita para que o cofre possa ser administrado de forma remota. As funcionalidades do cofre serão obtidas via programação em linguagem C. No final do projeto, deverá ser feita uma caixa para o cofre capaz de acomodar todos os circuitos desenvolvidos ao longo do projeto, com sistema de trava eletrônica acionado por senha.

1.3. Justificativa

Este trabalho se justifica pela necessidade de um equipamento eletrônico (cofre) como um instrumento de educação financeira. Conforme observado em pesquisas realizadas com a população brasileira, muitas pessoas encontram dificuldades quando a questão é poupar dinheiro. Isto ocorre geralmente por falta de planejamento, neste caso, o cofre poderia ser usado para estimular o hábito de poupar, acumulando determinada quantia para se planejar gastos futuros.

1.4. Organização deste trabalho

Este trabalho será organizado em capítulos, apresentando todo o processo de desenvolvimento do cofre microcontrolado e os resultados obtidos.

- Capítulo 1 - Introdução:

Este capítulo faz uma discussão sobre o tema deste trabalho, relacionando um problema encontrado nesta discussão com a motivação para desenvolver este projeto. São apresentados os objetivos, justificativa e organização deste trabalho.

- Capítulo 2 – Fundamentação Teórica:

Este capítulo relembra fundamentos teóricos recorrentes no trabalho e importantes para a sua compreensão.

- Capítulo 3 – Materiais e métodos

Este capítulo apresenta os componentes utilizados no trabalho e a metodologia utilizada para o desenvolvimento do cofre microcontrolado.

- Capítulo 4 – Testes e resultados preliminares

Este capítulo apresenta com detalhes os procedimentos para testes dos circuitos do cofre e os resultados obtidos destes testes.

- Capítulo 5 – Protótipo Finalizado

Este capítulo apresenta o processo de construção do protótipo finalizado, desde a etapa de projeto de *layout* até a obtenção do protótipo final.

- Capítulo 6 – Resultados e Discussões

Este capítulo apresenta os resultados obtidos com o protótipo finalizado do projeto. São apresentados os testes de confiabilidade de leitura de cédulas, testes de segurança com senhas, testes de comunicação serial, estimativa de custo do projeto e outras considerações gerais sobre o projeto.

- Capítulo 7– Conclusão

Este capítulo resume o trabalho, discutindo os resultados obtidos.

- Capítulo 8 – Trabalhos futuros

Este capítulo apresenta possibilidades de melhorias para o projeto

- Capítulo 9 – Referências Bibliográficas

Este capítulo apresenta o material utilizado como apoio para o desenvolvimento deste trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão explicados alguns conceitos importantes para a compreensão do trabalho desenvolvido.

2.1. Microcontroladores

Microcontroladores são “pequenos componentes eletrônicos, dotado de uma “inteligência” programável, utilizado no controle de processos lógicos” [4]. Este controle de processos pode ser usado para controlar periféricos diversos, como LEDs, *displays* de cristais líquidos (LCDs), sensores, entre outros periféricos.

Os microcontroladores podem ser programados para executar funções, por meio de um programa estruturado gravado dentro do componente. Quando o microcontrolador é alimentado, então o programa interno é executado. O microcontrolador possui uma Unidade Lógica Aritmética (ULA), que é responsável por realizar todas as operações lógicas e aritméticas do componente.

O microcontrolador é composto por apenas um único circuito integrado (CI). Um microcontrolador possui internamente memória de programa, memória de dados, portas de entrada e/ ou saída paralela, *timers*, comunicação serial, PWM, etc., o que é uma vantagem em relação aos microprocessadores, que apesar de possuírem uma ULA mais poderosa, não possuem todos esses recursos em um único CI.

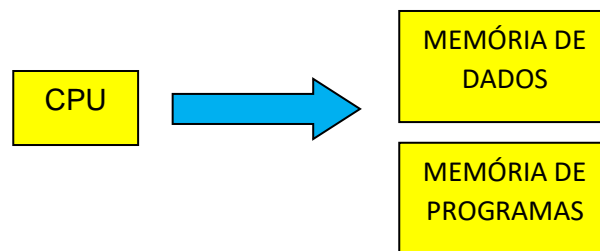
Dentro da classe dos microcontroladores, existem os chamados microcontroladores tradicionais e os da família PIC. Neste trabalho, será detalhado o microcontrolador PIC.

2.1.1. Microcontrolador PIC

Os microcontroladores PIC são circuitos integrados produzidos pela empresa *Microchip Technology Inc.* Este tipo de microcontrolador se destaca pelo tipo de arquitetura utilizada.

No caso dos microcontroladores tradicionais, utiliza-se a arquitetura Von Neumann. Nela, não é possível o acesso simultâneo da CPU a memória de dados e de programas. A Figura 3 apresenta um desenho esquemático representando esta arquitetura:

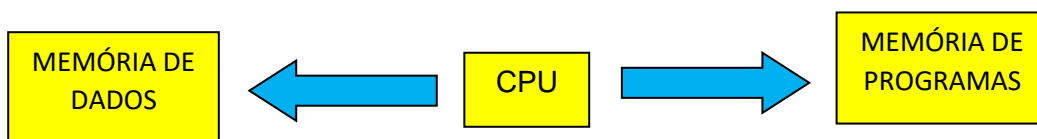
Figura 3: Arquitetura Von Neumann



Fonte: Os autores

Já no caso dos microcontroladores PIC, usa-se a arquitetura Harvard. Nela, há dois barramentos distintos para o acesso à memória de dados e de programas. Logo, enquanto uma instrução é executada, outra pode ser “buscada” da memória, tornando o processamento mais rápido. A Figura 4 apresenta uma representação da arquitetura Harvard:

Figura 4: Arquitetura Harvard



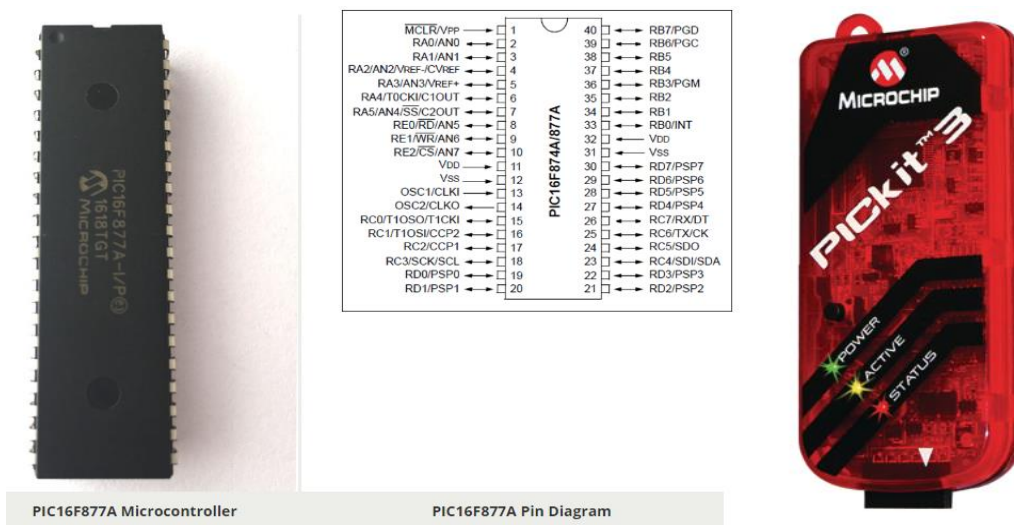
Fonte: Os autores

Os microcontroladores PIC também são do tipo RISC (*Reduced Instruction Set Computer* – computador com set de instruções reduzidas). No PIC, existem apenas 35 instruções, mas isto torna o aprendizado do microcontrolador mais fácil e dinâmico.

Um tipo de PIC muito utilizado em projetos eletrônicos é o 16F877A. Este é um microcontrolador da família 16F, de 8 *bits* e núcleo de 14 *bits* [5]. Este PIC pode ser adquirido na forma de circuito integrado e possui 40 pinos, com encapsulamento PDIP (*Dual In-Line Package*). A primeira letra P do PDIP refere-se ao invólucro de plástico do microcontrolador.

O PIC 16F877A e o gravador Pickit3 são mostrados na Figura 5:

Figura 5: Microcontrolador PIC 16F877A e gravador Pickit3



Fonte: <https://www.baudaeletronica.com.br/microcontrolador-pic16f877a.html>. Acesso em 15/10/2020.

A programação para controle do PIC e seus periféricos é feita via *software*, por meio da geração de um código executável em linguagem C. A transferência deste código para o PIC pode ser feita por meio de um *bootloader* ou um gravador para microcontrolador, como o Pickit 3. Para o funcionamento do PIC 16F877A, normalmente utiliza-se um circuito oscilador com cristal de quartzo e capacitores, ligado aos pinos 13 e 14. Este microcontrolador possui 33 pinos analógicos e digitais para conexão externa de periféricos.

O PIC 16F877A possui ainda 15 fontes de interrupção, conversor AD de 10 *bits* e 8 canais de entrada, 2 módulos CCP (comparação, captura e PWM) e 3 temporizadores/contadores (um de 16 *bits* e dois de 8 *bits*). Com este PIC, é possível também realizar comunicação paralela e serial (USART e MSSP) e usar o *Watchdog Timer*.

2.1.1.1. Memórias PIC

As memórias de um microcontrolador PIC podem ser de programa ou de dados. A memória de dados, também conhecida como RAM (*Random Access Memory* – Memória de Acesso aleatória), possui memória de acesso randômico ou direto e volátil [6]. Ou seja, quando o PIC é desligado, a memória é automaticamente perdida.

Já a memória de programa é a ROM (*Read Only Memory* – Memória apenas de leitura). Esta memória é não volátil e permite apenas a leitura do seu conteúdo e é construída para manter seus dados permanentemente [6].

Algumas categorias das memórias ROM serão detalhadas a seguir.

2.1.1.1.1. Memória PROM

A memória PROM (*Programmable Read-Only Memory* – Memória apenas leitura programável) é uma memória ROM programável apenas uma vez. É construída com ligações a fusível. O usuário pode realizar a gravação no equipamento selecionando uma linha no endereço desejado e inserindo os dados a serem gravados na entrada de dados. Então um pulso no pino específico de programação do CI de memória é aplicado, e o fusível queima, impedindo nova programação

2.1.1.1.2. Memória EPROM

A EPROM (*Erasable Programmable Read-Only Memory* – Memória apenas leitura programável e apagável) pode ser programada, apagada e reprogramada pelo usuário. Sua arquitetura interna é constituída de transistores MOS, sendo que os dados armazenados são apagados com exposição à luz ultravioleta por aproximadamente 15 minutos através de uma janela de quartzo transparente localizada na superfície superior da memória.

2.1.1.1.3. Memória EEPROM

As memórias EEPROM (*Electrically-Erasable Programmable Read-Only Memory* – Memória apenas leitura programável eletricamente e apagável) é uma memória não volátil, sendo que as operações de gravação e apagamento dos dados são realizadas por tensão elétrica. É construída com tecnologia NMOS.

A EEPROM, ou E2PROM, pode ser gravada, apagada e regravada pelo usuário *in circuit*, ou seja, no próprio circuito eletrônico em que a memória está instalada. As operações de programação e as de apagamento podem ser realizadas em posições de memória específicas, não sendo necessário apagar toda a memória para alterar algum dado.

Este tipo de memória é ideal para armazenagem de dados, como os dados do cofre deste trabalho, que devem ser mantidos mesmo se este for desligado.

2.1.1.2. Comunicação serial (USART)

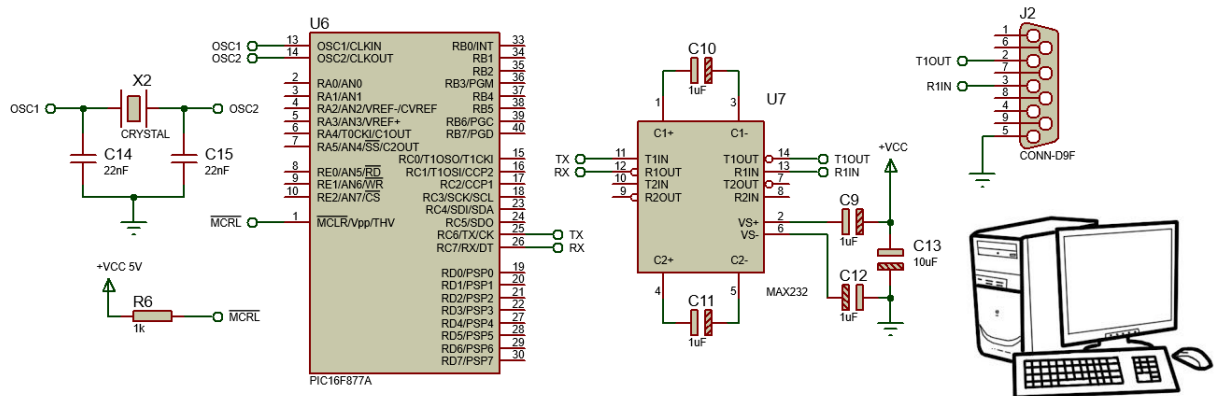
A USART (*Universal Synchronous Asynchronous Receiver Transmitter*), ou interface de comunicação serial, pode ser configurada para trabalhar no modo assíncrono (*full-duplex*) para comunicação com computadores pessoais ou no modo síncrono (*half-duplex*), permitindo a comunicação com conversores A/D, EEPROM serial etc.

A USART é um circuito eletrônico responsável pela interface entre o dispositivo e a porta serial. Utiliza protocolos diferentes para transmissão e recepção. Na transmissão, o dispositivo envia os dados para a USART de forma paralela e a USART se encarrega de transmiti-los *bit a bit* pela comunicação serial. Já na recepção, a USART realiza a operação contrária, recebendo a mensagem *bit a bit* e entregando de forma paralela para o dispositivo.

A comunicação serial pode ser feita nos pinos 25 e 26 do PIC 16F877A. O compilador MikroC do PIC possui uma biblioteca denominada UART, na qual é possível fazer a programação para realizar a comunicação serial.

Uma imagem de comunicação serial entre um microcontrolador PIC 16F877A e um computador é representada na Figura 6:

Figura 6: Comunicação serial PIC 16F877A e computador



Fonte: Os Autores.

Na Figura 6, é apresentado um conversor MAX232, que é responsável por fazer a comunicação entre o PIC e o computador. Este não é o único tipo de conversor, existem outros modelos que também podem desempenhar esta mesma função, como o conversor USB/Seria CH340, utilizado neste trabalho.

A USART pode ser aplicada utilizando a comunicação assíncrona RS-232. Neste protocolo, um pacote é constituído por 10 ou 11 *bits*, dos quais 8 *bits* constituem a mensagem, 1 *bit* de início (*Start Bit*), 1 *bit* de parada (*Stop Bit*) e 1 *bit* de paridade (*Parity Bit*) para o controle de erro.

Após o envio do *start bit*, o transmissor manda a mensagem de 8 *bits*, iniciando pelo *bit* menos significativo (LSB), seguido de um *bit* de paridade e de um *Stop Bit*, sinalizando ao receptor o fim da transmissão. As taxas de transmissão são medidas em *bits* por segundo (bps). Os valores mais comumente utilizados são de 300 bps, 600 bps, 1200 bps, 2400 bps, 4800 bps e 9600 bps.

Desta forma, é possível enviar letras, palavras e comandos de um microcontrolador para uma interface gráfica em computador. Com isso, é possível enviar informações importantes do PIC, como os dados do cofre deste trabalho.

Além disso, a interface gráfica usada pode ser a do próprio compilador MikroC ou outra compatível. Um programa que permite fazer a montagem de uma interface gráfica personalizada é o Visual Studio 2019, que é o utilizado para fazer o menu do cofre deste trabalho.

2.2. Sensores e Atuadores

Sensores, ou transdutores de entrada, fazem a conversão de uma forma de energia em energia elétrica. Já os transdutores de saída fazem o oposto: o sinal eletrônico na saída do circuito é convertido em outra forma de energia, por exemplo, som, luz ou movimento (energia mecânica). [7]

Estes transdutores de saída também podem ser chamados de atuadores, quando realizam a conversão de uma forma de energia em energia mecânica. Exemplos de transdutores de saída são as lâmpadas, LEDs, motores e alto-falantes. [7]

Os sensores são componentes utilizados quando se deseja disparar a operação de um circuito em resposta a algum acontecimento físico (por exemplo, uma variação de temperatura). Esta variação de temperatura pode então ser convertida em energia elétrica para posterior análise.

Existem diversos tipos de sensores, utilizados para análise de diferentes tipos de fenômenos físicos. Um exemplo são os dispositivos sensíveis a luz, como os fotorresistores (ou fotocélulas), fotodiodos e fototransistores. Os fototransistores, por exemplo, são compostos por transistores de junção bipolar, encapsulados com invólucro claro, de forma que a luz polarize a junção base-emissor.

Outro exemplo são sensores que podem detectar variações de temperatura, como o caso dos termopares e termistores e os sensores de pressão, como os manômetro capacitivo, tipo catodo frio ou catodo quente. Estes sensores, de temperatura e pressão, são muito utilizados principalmente nos ambientes industriais, para controle de processos.

De acordo com a necessidade do projeto, podem ser escolhidos diferentes tipos de sensores. No caso deste trabalho, um sensor adequado para detecção de cores é o sensor RGB, que trabalha com a identificação dos corpos a partir da verificação dos seus padrões RGB.

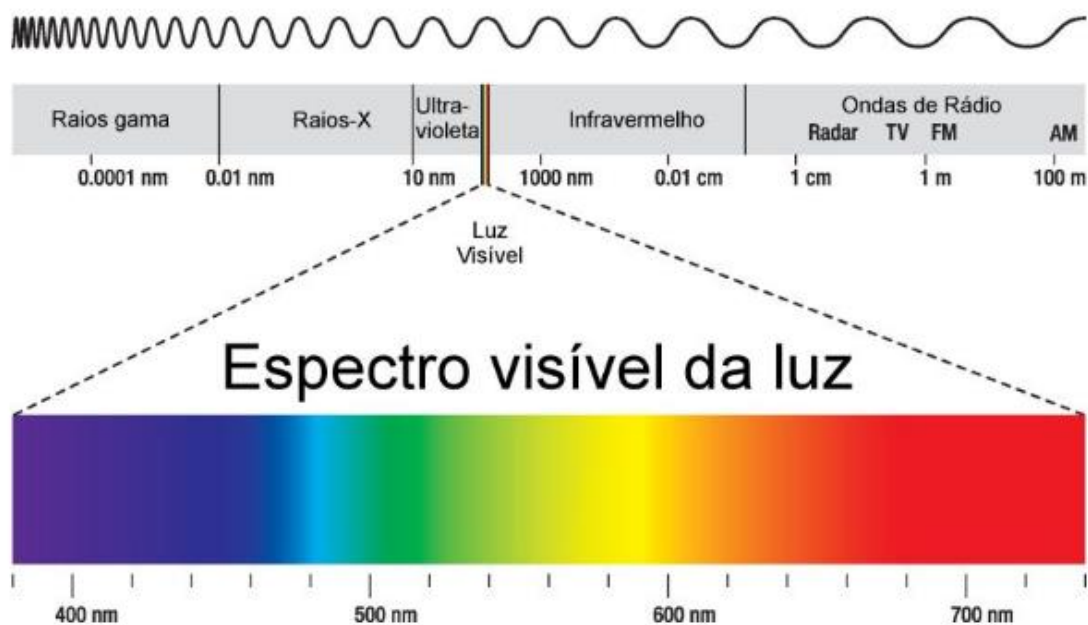
Neste trabalho, será utilizado especificamente o sensor RGB TCS230, um sensor comercial com aplicação em muitos projetos eletrônicos. Este sensor será explicado com detalhes no capítulo 3, materiais utilizados no projeto.

2.3. Cor

A cor é a resultado da atuação de dois tipos de sensores, presentes no interior dos olhos dos seres humanos: os cones e os bastonetes. Os cones são responsáveis pela percepção das cores e os bastonetes pela percepção de tons de cinza. [8]

O ser humano é capaz de enxergar somente em uma limitada faixa de comprimento de onda. Esta faixa, chamada de espectro visível da luz, de aproximadamente 380 a 700 nm, está representada na Figura 7:

Figura 7: Espectro visível da luz



Fonte: <https://www.infoescola.com/fisica/espectro-eletromagnético/>. Acesso em 20/11/2020

A cor de um objeto é determinada pela frequência de onda que ele reflete. Ou seja, um objeto azul absorve frequências fora do azul, e reflete a frequência de onda do azul que será identificada pelo nosso olho.

Existem dois sistemas principais para as cores: o RGB e o CMY. Estes dois tipos de sistemas serão explicados com mais detalhes a seguir.

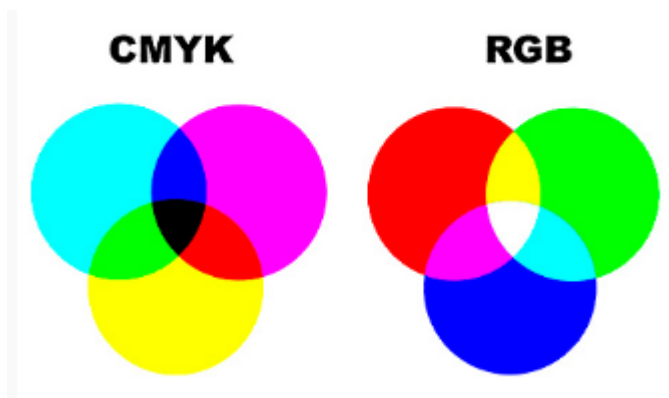
2.3.1. Sistemas RGB e CMY

O sistema RGB (*Red, Green, Blue*) regula as cores dos corpos que emitem luz. É conhecido também como cor luz, e baseia-se na adição, ou seja, se somarmos as três cores básicas nas proporções corretas, teremos branco. O RGB é utilizado nas fotografias, televisores e computadores. [9]

Já o CMY (*Cyan, Magenta e Yellow* – ciano, magenta e amarelo) é o sistema que regula as cores de corpos opacos. Ele também é conhecido como sistema de cor pigmento, e trabalha por subtração, ou seja, somando as três cores na proporção correta, teremos o preto. O CMY é utilizado em impressoras e artes plásticas. Uma variação deste sistema é o CMYK, no qual se acrescenta o preto (K) para variações na escala do cinza. [9]

Na Figura 8, podemos observar uma representação esquemática para cada um destes sistemas:

Figura 8: Sistemas CMYK e RGB



Fonte: <https://imprensa.ufc.br/pt/duvidas-frequentes/padrao-de-cor-rgb-e-cmyk/>. Acesso em 18/11/2020

Na Figura 8, observa-se que nas intersecções entre cada cor forma-se uma cor secundária diferente. A proporção entre cada uma das cores primárias pode gerar cores com tonalidades ligeiramente diferentes.

No caso do CMYK, que possui cores subtrativas, a mistura entre ciano e magenta produz azul, ciano e amarelo produz verde e magenta e amarelo produz vermelho. A mistura das três cores primárias produz preto.

No RGB, cujas cores são aditivas, a mistura de vermelho e verde produz amarelo, vermelho e azul produz magenta e verde e azul produz ciano. A mistura das três cores primárias produz o branco.

Como é possível notar, as cores primárias do CMYK são as cores secundárias do RGB e vice-versa. Neste trabalho, será utilizado somente o sistema RGB, para identificação de corpos com variação de cores, como as cédulas monetárias do sistema financeiro brasileiro. Estas cédulas serão detalhadas a seguir.

2.4. Cédulas monetárias do sistema financeiro brasileiro

O sistema monetário utilizado atualmente no Brasil é o Real, e é composto por cédulas e moedas. No caso das cédulas, que é o foco deste trabalho, a divisão é feita em duas famílias. A primeira família data de 01/07/1994, após a transição do Cruzeiro Real para o Real. [10]

As cédulas desta primeira família podem ser observadas na Figura 9:

Figura 9: Cédulas monetárias da Primeira Família do Real



Fonte: Banco Central do Brasil.

A partir de 2010, foram lançadas as cédulas da Segunda Família do Real, elaborada com novos elementos gráficos e de segurança, para evitar tentativas de falsificação. [11]

As cédulas da Segunda Família do Real podem ser vistas na Figura 10:

Figura 10: Cédulas monetárias da Segunda Família do Real



Fonte: Banco Central do Brasil.

Neste trabalho, serão utilizadas as notas da Segunda Família do Real, que serão detalhadas a seguir.

2.4.1. Cédulas da Segunda Família do Real

As primeiras cédulas da Segunda Família do Real a serem lançadas foram as de 100 e 50 reais, no ano de 2010. Em 2012, foram lançadas as cédulas de 20 e 10 reais. Em 2013, foram lançadas as notas de 5 e 2 reais e em 2020, foi lançada a última nota da família, a de 200 reais. [12]

No corpo das cédulas, a parte frontal é identificada com a efígie da República. Já o verso mostra um animal representativo da fauna brasileira, a numeração da cédula, dentre outros elementos.

As imagens destas cédulas podem ser vistas nas Figuras 11 a 17:

Figura 11: Cédula monetária de 2 reais



Fonte: Cédulas do Brasil.

Figura 12: Cédula monetária de 5 reais



Fonte: Cédulas do Brasil.

Figura 13: Cédula monetária de 10 reais



Fonte: Cédulas do Brasil.

Figura 14: Cédula monetária de 20 reais



Fonte: Cédulas do Brasil.

Figura 15: Cédula monetária de 50 reais



Fonte: Cédulas do Brasil.

Figura 16: Cédula monetária de 100 reais



Fonte: Cédulas do Brasil.

Figura 17: Cédula monetária de 200 reais



Fonte: Cédulas do Brasil.

As dimensões das cédulas (largura x altura) da Segunda Família do Real são apresentadas na Tabela 1:

Tabela 1: Dimensões das Cédulas da Segunda Família do Real

| DIMENSÕES DAS CÉDULAS | | |
|-----------------------|--------------|-------------|
| CÉDULA | Largura (cm) | Altura (cm) |
| 2 REAIS | 12,1 | 6,5 |
| 5 REAIS | 12,8 | 6,5 |
| 10 REAIS | 13,5 | 6,5 |
| 20 REAIS | 14,2 | 6,5 |
| 50 REAIS | 14,9 | 7,0 |
| 100 REAIS | 15,6 | 7,0 |
| 200 REAIS | 14,2 | 6,5 |

Fonte: Os autores.

Como é possível observar na Tabela 1, os valores de altura são praticamente os mesmos para todas as cédulas (6,5 cm), com exceção das cédulas de 50 e 100 reais (7 cm). As larguras das cédulas são ligeiramente diferentes umas das outras.

A cédula de 200 reais possui uma peculiaridade que é ter as mesmas dimensões da cédula de 20 reais. Este foi um fato que gerou críticas negativas à nota, pois dificultou a acessibilidade para pessoas com deficiência visual, que identificam o valor das cédulas pelo seu tamanho [13].

Nas cédulas da Segunda Família do Real, existem diferentes critérios de segurança para verificar a autenticidade destas. Estes critérios serão apresentados a seguir.

2.4.2. Critérios de segurança das notas da Segunda Família do Real

O Banco Central do Brasil possui uma cartilha com diversos critérios para verificar a autenticidade de uma cédula da Segunda Família do Real [14]. Seis desses critérios podem ser usados como base para verificação.

O primeiro é a marca d'água. A nota, observada contra a luz, apresenta a figura do animal e o número do valor da nota, como visto na Figura 18:

Figura 18: Item 1 - Marca d'água das cédulas do real



Fonte: Banco Central do Brasil.

O segundo detalhe é o alto relevo. Pelo tato, algumas regiões apresentam alto relevo, como mostrado na Figura 19:

Figura 19: Item 2 - Pontos de alto relevo nas cédulas do real



Fonte: Banco Central do Brasil.

O terceiro detalhe é um número escondido que é observado quando se inclina levemente a nota. Este detalhe se localiza logo abaixo do número no canto superior direito de cada cédula, como mostrado na Figura 20:

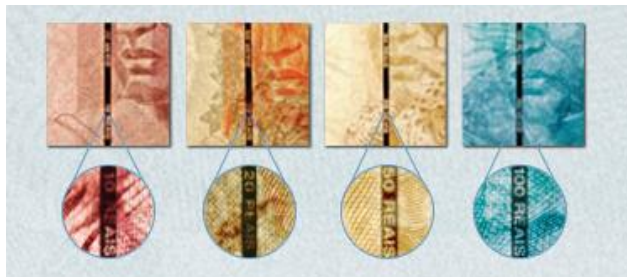
Figura 20: Item 3 - Número escondido nas cédulas do real



Fonte: Banco Central do Brasil.

O quarto detalhe é o fio de segurança. Ele pode ser observado nas notas de 10, 20, 50 e 100 reais colocando a nota contra a luz. Um fio escuro, na área central da nota, aparecerá com o valor da nota e a palavra REAIS escrito, conforme Figura 21:

Figura 21: Item 4 - Fio de segurança nas cédulas do real



Fonte: Banco Central do Brasil.

O quinto detalhe é o quebra-cabeça. Observando a nota contra a luz, aparecerá um desenho que complementa o do verso, formando o número do valor da nota, como mostrado na Figura 22:

Figura 22: Item 5 - Quebra-cabeça nas cédulas do real



Fonte: Banco Central do Brasil.

O sexto detalhe são os elementos fluorescentes. Sob luz ultravioleta, aparecerão os seguintes elementos na Figura 23:

Figura 23: Item 6 - Elementos fluorescentes na cédula do real



Fonte: Banco Central do Brasil.

Neste trabalho, convém ressaltar que não será utilizado nenhum sistema de identificação de autenticidade das cédulas. Este procedimento pode ser feito manualmente, como forma de incentivo à educação financeira.

Para armazenamento de cédulas, podem ser usados cofres. O conceito de cofre e alguns exemplos serão apresentados a seguir.

2.5. Cofres

Cofres são compartimentos normalmente utilizados para guardar objetos de valor, protegidos por algum sistema de segurança. Existem diversos tipos de cofre, com tamanho, peso, material e sistemas de segurança diferentes [15]. A seguir serão apresentados apenas alguns exemplos de cofres disponíveis no mercado, para efeito de exemplificação.

2.5.1. Cofre mecânico

Este tipo de cofre utiliza um sistema manual, utilizando uma fechadura ou um disco com marcações numéricas que vão de 00 até 99. É um cofre simples, prático, e rápido para abertura e fechamento.

Uma imagem de um cofre mecânico é apresentada na Figura 24:

Figura 24: Cofre mecânico



Fonte: <https://www.casadocofre.com/produto/cofre-concretado-cs-40/>. Acesso em 05/11/2020

2.5.2. Cofre embutido

O cofre embutido é normalmente utilizado para guardar objetos de grande valor. Este cofre pode ser colocado em locais mais seguros, como paredes ou fundos falsos. Uma imagem deste cofre é mostrada na Figura 25:

Figura 25: Cofre embutido



Fonte: <https://www.casadocofre.com/produto/cofre-de-embutir-cs-32/>. Acesso em 05/11/2020.

2.5.3. Cofre biométrico

É um tipo de cofre mais sofisticado, utilizando um leitor biométrico que pode identificar o indivíduo pela íris ocular ou digital. Por esta razão, é um cofre com valor mais alto que os demais.

Um exemplo de cofre biométrico por digital é apresentado na Figura 26:

Figura 26: Cofre biométrico



Fonte: https://www.segurancajato.com.br/cofre-yale-standard-compact-bio-com-biometria?gclid=EAlalQobChMliKaZ5cncr7AIVhorlCh3LyAVyEAQYASABEgJbOvD_BwE. Acesso em 05/11/2020.

2.5.4. Cofre eletrônico

Um cofre eletrônico possui um sistema de segurança baseado na digitação de uma senha em um teclado alfanumérico. Os números digitados são apresentados em um *display* LCD. Um exemplo é o apresentado na Figura 27:

Figura 27: Cofre eletrônico



Fonte: <https://www.qualitycofres.com.br/cofre-eletronico-digital-onix>. Acesso em 05/11/2020.

O cofre eletrônico é o que possui as características mais adequadas para este trabalho, como praticidade, custo e possibilidade de realizar alterações no projeto eletrônico. Os materiais utilizados no projeto do cofre e a metodologia do projeto serão apresentados no capítulo a seguir.

3. MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e a metodologia utilizada para o desenvolvimento do cofre deste trabalho.

3.1. MATERIAIS UTILIZADOS NO PROJETO

3.1.1. Microcontrolador PIC 16F877A

O PIC 16F877A foi o microcontrolador escolhido para este trabalho. Alguns dados técnicos relevantes para o projeto são apresentados na Tabela 2:

Tabela 2: Dados técnicos do PIC 16F877A

| DADOS TÉCNICOS: | |
|------------------------------|--|
| Microcontrolador | PIC16F877A |
| Tensão de operação | 5V TTL |
| Tensão de alimentação máxima | 5.5V |
| Entradas e saídas digitais | 33 |
| Corrente de saída | 25 mA |
| Memória Flash | 14,3 KB / 8192 Instruções (para armazenagem de código) |
| Memória SRAM | 368 Bytes |
| Memória EEPROM | 256 Bytes |
| Velocidade do Clock | 0 - 20 MHz |
| Datasheet | PIC16F877A.pdf |

Fonte: <https://www.baudaeletronica.com.br/microcontrolador-pic16f877a.html>. Acesso em 15/10/2020.

Este microcontrolador possui tensão de operação de 5 V, ideal para projetos eletrônicos. Também possui 33 pinos de entradas e saídas digitais, possibilitando a conexão de uma boa quantidade de componentes periféricos. Este PIC pode ser programado via *software*, e o gravador Pickit 3, utilizado para gravação do código também fornece a tensão de 5 V para alimentar o PIC, por meio da conexão em uma porta USB de um computador.

3.1.2. Sensor RGB (TCS230)

O sensor RGB TCS230 é um conversor programável de luz em frequência, que combina fotodiodos de silício configuráveis e um conversor de corrente para frequência em um único circuito integrado monolítico CMOS. A saída é uma onda quadrada (Ciclo de trabalho de 50%) com frequência diretamente proporcional à intensidade da luz (irradiância). [16]

Este sensor funciona emitindo uma luz branca que atinge a superfície analisada e retorna de forma refletida. Este reflexo é processado pelo sensor que gera um padrão RGB (*Red, Green, Blue*) ou vermelho, verde e azul. Caso o valor esteja na variação que esteja configurada na memória do sensor, ele emite uma combinação de sinais discretos ou um sinal via comunicação informando que o produto está dentro da especificação da cor.

Uma imagem do sensor RGB TCS 230 é apresentada na Figura 28:

Figura 28: Sensor RGB TCS230



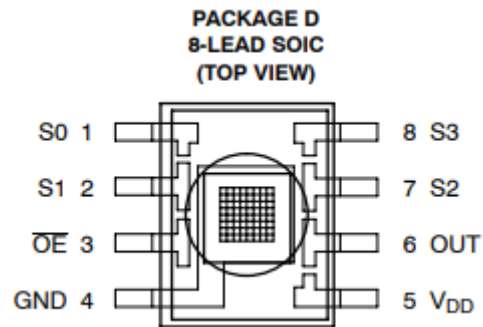
Fonte: <https://daeletrica.com.br/modulo-sensor-de-cor-rgb-tcs230.html>

Acesso em 15/10/2020.

O sensor TCS230 lê uma matriz 8 x 8 de fotodiodos. Dezesesseis fotodiodos possuem filtros azuis, dezesesseis têm filtros verdes, dezesesseis têm filtros vermelhos e dezesesseis são claros sem filtros.

O *datasheet* do TCS 230 [17] apresenta detalhes para configuração deste sensor. O TCS 230 possui oito pinos, sendo S0, S1, S2, S3 usado para configuração e pino OUT para saída. Uma imagem da pinagem do TCS 230 é mostrada na Figura 29:

Figura 29: Pinagens do sensor RGB TCS230



Fonte: Datasheet TAOS: Sensor TCS 230

A configuração do sensor é feita conforme mostrado na Tabela 3:

Tabela 3: Dados técnicos do Sensor TCS 230

| S0 | S1 | OUTPUT FREQUENCY SCALING (f_o) | | S2 | S3 | PHOTODIODE TYPE |
|----|----|------------------------------------|--|----|----|-------------------|
| L | L | Power down | | L | L | Red |
| L | H | 2% | | L | H | Blue |
| H | L | 20% | | H | L | Clear (no filter) |
| H | H | 100% | | H | H | Green |

Fonte: Datasheet TAOS: Sensor TCS230.

Os pinos S0 e S1 são responsáveis por controlar a escala de frequência da saída, enquanto S2 e S3 são usados para utilização de cada um dos quatro conjuntos de filtros de fotodiodos.

Para acionar a leitura do sensor RGB, foi usado um sensor infravermelho para detectar a presença das cédulas. Este sensor é explicado a seguir.

3.1.3. Sensor infravermelho (sensor TIL32 e fotodiodo TIL78)

Os sensores infravermelhos (IR) são comumente utilizados em par, sendo um o emissor infravermelho e o outro o receptor. Estes possuem um alcance limitado, porém podem ser utilizados para diversas aplicações. Entre elas, podem ser citados: sensor de passagem, de presença, de posição, de distância, de rotação e de linha.

O emissor infravermelho emite uma luz cuja radiação está no comprimento de onda acima de 800 nm até 1 mm e frequência de 300 Ghz a 400 Thz. Essa radiação emitida quando refletida por um objeto, alcança o receptor que gera uma tensão proporcional a quantidade de radiação recebida.

Uma imagem de sensor emissor e receptor infravermelho é apresentada na Figura 30:

Figura 30: Emissor e receptor infravermelho



Fonte: <https://arduinobymyself.blogspot.com/2013/01/sensor-ir-projetos.html>. Acesso em 05/12/2020

No projeto do cofre, o sensor infravermelho será usado para detectar ou não a presença de cédula. Neste caso, serão utilizados sensores de dimensões menores que o apresentado na Figura 30, para poderem ser colocados na caixa leitora. O emissor utilizado é um sensor IR TIL32 e o receptor um fotodiodo TIL78.

3.1.4. Display LCD (16x2)

Os LCDs (*Liquid Crystal Displays*) são módulos de caracteres alfanuméricos para interface de comunicação visual. Um display LCD tem de 14 a 16 pinos, sendo dois para ativar o *backlight* (luz de fundo do LCD), três para alimentação, três para controle e oito para dados. [19]

O LCD é composto por um microprocessador de controle, uma RAM interna que mantém a escrita no display (DDRAM) os dados enviados pelo microcontrolador e uma RAM de construção de caracteres especiais (CGRAM).

Para configuração deste *display*, é necessário configurar cada um dos pinos. A Tabela 4 apresenta a função de cada pino do LCD:

Tabela 4: Pinagem de um display LCD

| Pino | Função | Descrição |
|------|-----------------|---|
| 1 | Alimentação | Terra ou GND |
| 2 | Alimentação | VCC ou +5V |
| 3 | V0 | Tensão para ajuste de contraste |
| 4 | RS Seleção: | 1 - Dado, 0 - Instrução |
| 5 | R/W Seleção: | 1 - Leitura, 0 - Escrita |
| 6 | E Chip select | 1 ou (1 → 0) - Habilita, 0 - Desabilitado |
| 7 | B0 LSB | Barramento de Dados |
| 8 | B1 | |
| 9 | B2 | |
| 10 | B3 | |
| 11 | B4 | |
| 12 | B5 | |
| 13 | B6 | |
| 14 | B7 MSB | |
| 15 | A (qdo existir) | Anodo p/ LED backlight |
| 16 | K (qdo existir) | Catodo p/ LED backlight |

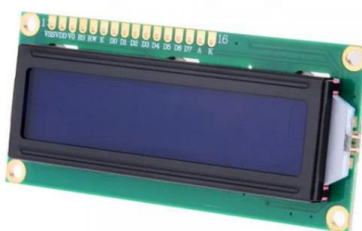
Fonte: <https://www.embarcados.com.br/modulo-de-display-lcd/>

Acesso em 15/10/2020.

Dos pinos, oito deles (8 a 14) são utilizados como barramento de dados, ou seja, para comunicação com o microcontrolador. Existem LCDs de diferentes tamanhos e cor de *backlight*. O tamanho é definido pelo número de linhas e colunas disponíveis para realizar a escrita.

O LCD utilizado neste projeto é o 16x2 (16 colunas e 2 linhas), suficiente para apresentar as informações necessários ao cofre. Este *display* é apresentado na Figura 31:

Figura 31: Display LCD 16x2



Fonte: <https://www.baudaeletronica.com.br/display-lcd-16x2-azul.html>. Acesso em 15/10/2020.

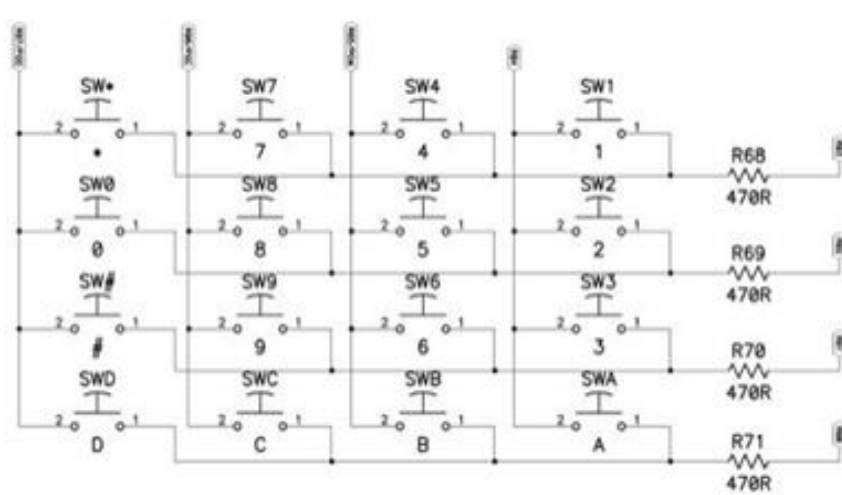
3.1.5. Teclado matricial (4x3)

O teclado matricial utiliza o conceito de multiplexação de teclas para identificar quais teclas foram pressionadas. A multiplexação é a capacidade de enviar mais de um sinal utilizando um mesmo caminho. O teclado é disposto em formato matricial, e com esta configuração, tendo N terminais é possível ler até $(N/2)^2$ chaves, ou seja, 8 terminais fazem a leitura de 16 teclas. [20]

Na leitura matricial, as chaves são separadas em colunas, ligando uma coluna por vez. Isto pode ser feito utilizando terminais de saída para acionamento de cada coluna e terminais de entrada para a leitura das linhas. Normalmente utilizam-se resistores de *pull-up* na entrada do microcontrolador ou os *pull-ups* internos do próprio microcontrolador, caso este possua. Para proteger o sistema de curto entre os terminais, geralmente se utiliza resistores em série nas saídas a serem controladas.

Um exemplo de circuito de teclado matricial é apresentado na Figura 32:

Figura 32: Circuito elétrico de um teclado matricial



Fonte: <https://www.embarcados.com.br/teclado-matricial-e-varredura-de-teclas/>. Acesso em 15/10/2020

Na Figura 32, é apresentado um exemplo de circuito para teclado matricial com 4 linhas e 4 colunas (4x4). Neste circuito, são utilizados resistores com valor de 470 Ω .

Um teclado que atende as especificações do projeto do cofre é o 4x3, com 4 linhas e 3 colunas. Este teclado pode ser observado na Figura 33:

Figura 33: Teclado matricial 4x3



Fonte: <https://produto.mercadolivre.com.br/MLB-959823018-teclado-matricial-12-3x4-botoes-arduino-pic-microcontrolador>. Acesso em 15/10/2020.

Como este teclado utiliza o conceito de multiplexação de teclas, não é necessário conectar cada tecla do teclado a um pino do microcontrolador, reduzindo assim o número de pinos necessários à conexão.

3.1.6. Servo motor (SG-90)

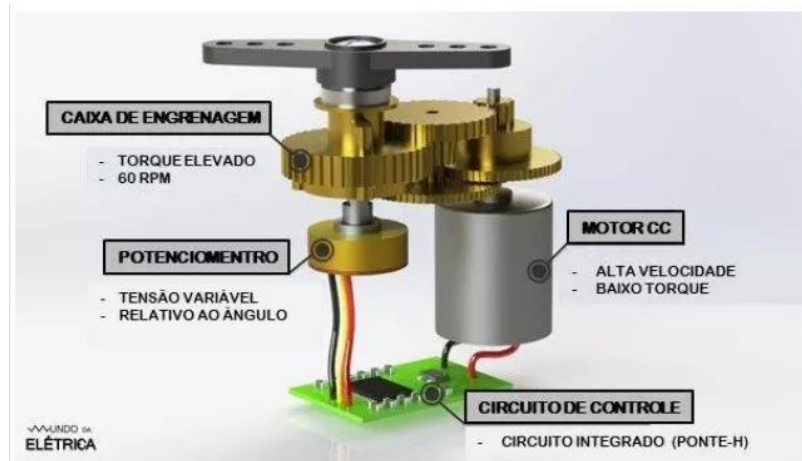
Servos motores são tipos de motores utilizados para controle de movimento com posicionamento de alta precisão, reversão rápida e de alto desempenho. Possuem muitas aplicações em robótica, sistemas automatizados, máquinas CNC, entre outras aplicações. [21]

Para controlar a velocidade e a posição final do motor, trabalha-se com servomecanismo com realimentação de posição, combinando internamente um motor com circuito de realimentação, um controlador e outros circuitos complementares.

Os servos motores podem ser classificados em corrente alternada (CA) e corrente contínua (CC), dependendo da forma de alimentação. Os servos motores CA normalmente são usados em ambientes industriais, pois costuma ser de elevada potência, oferece maior exatidão no seu controle e pouca manutenção.

Os servos motores CC são usados em projetos menores, possuem custo relativamente baixo e são eficientes. Possuem um pequeno motor de corrente contínua, um potenciômetro de realimentação, uma caixa de engrenagens e um circuito eletrônico para acionamento, como é apresentado na Figura 34:

Figura 34: Desenho esquemático de um servo motor



Fonte: <https://www.mundodaeletrica.com.br/o-que-e-servo-motor-e-como-funciona/#:~:text=Para%20controlar%20a%20velocidade%20e,controlador%20e%20outros%20circuitos%20complementares.> Acesso em 18/11/2020

No servo motor CC, uma tensão contínua de referência é regulada para um valor correspondente à saída desejada. Para fazer o controle de forma digital um microcontrolador pode ser utilizado para produzir sinais de controle com maior precisão e assim fornecer pulsos de tensão, que são obtidos pela técnica PWM. Um potenciômetro interno fornece um sinal de tensão correspondente ao ângulo do eixo do motor.

Este sinal de tensão relacionado à posição atual do motor é comparado com a tensão desejada, produzindo um sinal de erro com tensão positiva ou negativa. Se o sinal for positivo, a tensão aplicada na armadura do motor faz com que o rotor gire em uma direção. Se o erro for negativo, o rotor gira na direção oposta.

Um servo motor disponível no mercado e muito utilizado em projetos é o SG-90. Ele é utilizado em projetos de robótica com Arduino, PIC, *Raspberry*, entre outros. Ele possui pequenas dimensões, e foi utilizado para controle da abertura da porta da caixa leitora de cédulas. Seu funcionamento é simples, possuindo apenas três pinos para configuração.

Algumas especificações deste servo motor são apresentadas a seguir [22]:

- Voltagem de operação: 3,0 – 7,2V
- Ângulo de rotação: 180 graus
- Temperatura de operação: -30°C a +60°C
- Dimensões: 32x30x12mm
- Peso: 9g.

Uma imagem do servo motor SG-90 pode ser vista na Figura 35:

Figura 35: Servo motor SG-90



Fonte: <https://www.filipeflop.com/produto/micro-servo-9g-sg90-towerpro/>. Acesso em 17/11/2020

3.1.7. Conversor Serial (TTL CH340)

Um tipo de conversor serial comercial é o adaptador USB-serial TTL CH340. Este conversor é um pequeno módulo de conversão USB para UART equipado com o controlador CH340, que pode ser utilizado em dispositivos e projetos RS-232 para USB [23]. Uma imagem do conversor é apresentada na Figura 36:

Figura 36: Conversor USB serial TTL CH340



Fonte: <https://produto.mercadolivre.com.br/MLB-1721058964-modulo-conversor-usb-para-ttl-rs232-ch340-cnf>. Acesso em 20/11/2020

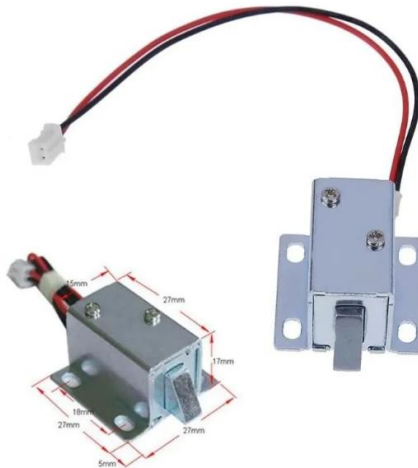
Algumas características técnicas deste conversor são apresentadas a seguir:

- Velocidade: USB 2.0 (12 Mbps) / Até 1024 Bytes / Até 2 Mbps / USB-A
- Tensão: 3,3 V / 5 V. Configurável via jumper
- Temperatura de operação: -40 a +85°C
- Tamanho: Largura: 51 mm, Comprimento: 18 mm, Altura: 8 mm

3.1.8. Trava eletrônico (fechadura solenoide 12 V)

Para a abertura e fechamento do cofre, é utilizada uma trava eletrônica com conexão ao circuito do microcontrolador [24]. Esta trava é uma fechadura solenoide eletrônica, alimentada com 12 V e representada na Figura 37:

Figura 37: Fechadura solenoide eletrônica



Fonte: <https://produto.mercadolivre.com.br/MLB-1421878616-mini-fechadura-solenoide-eletrnica-trava-trinco-automaco>. Acesso em 20/11/2020

Algumas características dessa trava são:

- Comprimento: 27 mm, largura: 15 mm, altura: 17 mm
- Comprimento da língua: 7 mm, altura da língua: 10 mm.
- Comprimento do cabo: 120 mm.

No circuito do cofre, é necessário compatibilizar a tensão de 12 V do solenoide com outros componentes, como o microcontrolador e o servo motor, que usam 5 V. Para isso, utilizam-se os reguladores de tensão, apresentados a seguir.

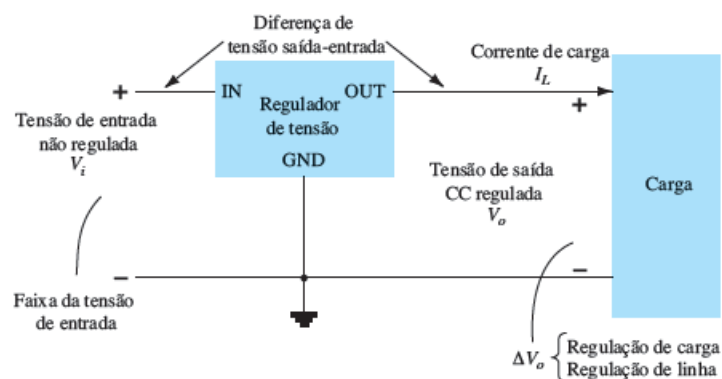
3.1.9. Regulador de Tensão (CI 7805)

Em alguns projetos, é necessário converter um valor de tensão em um valor menor. Para esta função, existem os reguladores de tensão com circuitos integrados (Cis). Estes Cis reguladores contêm circuitos de fonte de referência, amplificador comparador e dispositivo para proteção contra sobrecarga, tudo em uma única pastilha. Os Cis oferecem regulação para uma tensão positiva fixa, uma tensão negativa fixa ou uma tensão ajustável. [25]

Estes reguladores de tensão são compostos normalmente por três terminais conectados a uma carga. O regulador com tensão de saída fixa tem uma tensão CC não regulada aplicada a um terminal de entrada, uma tensão CC de saída regulada em um segundo terminal e um terceiro terminal conectado ao terminal terra.

Esse esquema elétrico é apresentado na Figura 38:

Figura 38: Diagrama de blocos do regulador de tensão

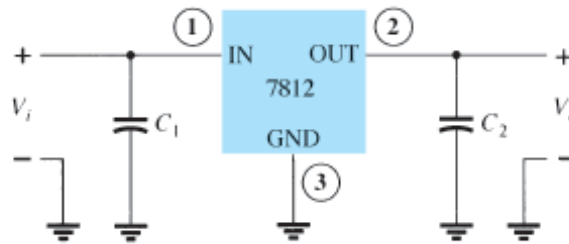


Fonte: Boylestad: Dispositivos Eletrônicos e Teoria de Circuitos.

Para regular tensão positiva fixa, existem os reguladores da série 7800, que fornecem tensões reguladas fixas de 5 V até 24 V. Um CI regulador desta série, o 7812 é utilizado para regular a tensão em +12 V CC. Uma tensão de entrada não regulada é filtrada pelo capacitor C1 e alimenta o CI no terminal IN. O terminal OUT do CI fornece a tensão de +12 V regulada, filtrada pelo capacitor C2 para evitar ruídos.

O esquema elétrico montado com este regulador é mostrado na Figura 39:

Figura 39: Circuito regulador de tensão com CI 7812



Fonte: Boylestad: Dispositivos Eletrônicos e Teoria de Circuitos.

A Tabela 5 apresenta dados de outros tipos de reguladores da série 7800:

Tabela 5: CIs da série 7800 e tensões de saída reguladas

| Componente CI | Tensão de saída (V) | V_i mínima (V) |
|---------------|---------------------|------------------|
| 7805 | +5 | 7,3 |
| 7806 | +6 | 8,3 |
| 7808 | +8 | 10,5 |
| 7810 | +10 | 12,5 |
| 7812 | +12 | 14,6 |
| 7815 | +15 | 17,7 |
| 7818 | +18 | 21,0 |
| 7824 | +24 | 27,1 |

Fonte: Boylestad: Dispositivos Eletrônicos e Teoria de Circuitos.

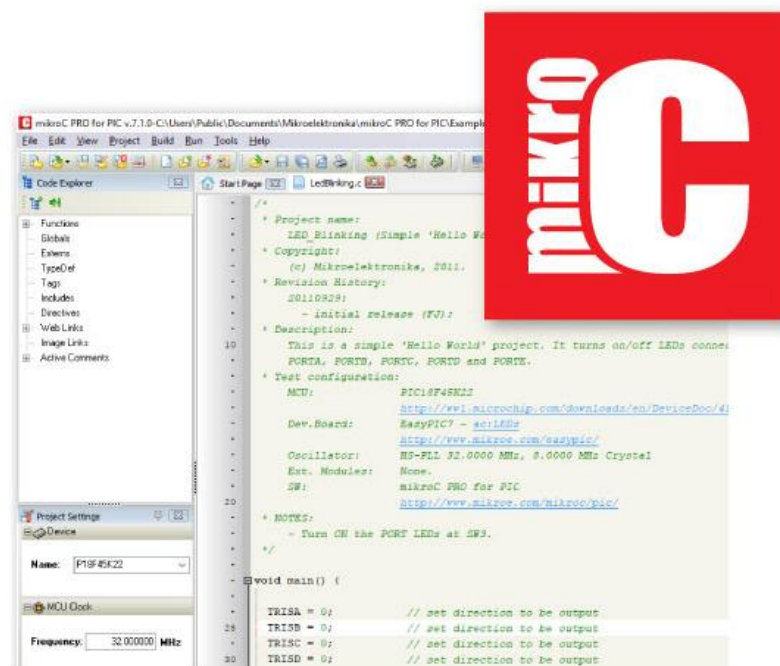
Para regular tensão negativa fixa, existem os reguladores da família 7900. Estes reguladores são semelhantes aos da série positiva 7800. Uma tensão de saída fixa é obtida quando o valor de entrada estiver abaixo de um mínimo especificado. Como exemplo, o CI 7912 fornece uma saída de -12 V se a tensão de entrada do CI tiver um valor mais negativo do que -14,6 V.

No caso do projeto do cofre trabalha-se com tensões positivas. Logo, um CI da família 7800 é utilizado. Nesta situação, é preciso regular uma tensão de 12 V para uma tensão compatível com o PIC (5 V). Desta forma, o regulador 7805 é o escolhido para este trabalho.

3.1.10. Softwares utilizados no projeto

Neste trabalho, foi utilizado o *software* MikroC para programar o microcontrolador. Neste programa, é possível elaborar códigos em linguagem C para configura microcontroladores da família PIC para diversas aplicações. Uma imagem da interface deste programa é apresentada na Figura 40:

Figura 40: Interface do programa MikroC



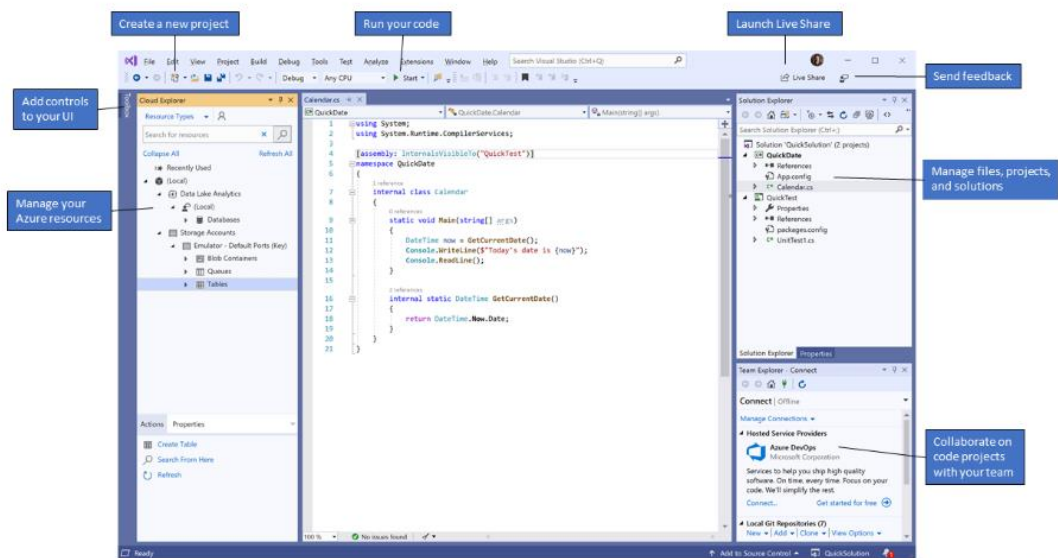
Fonte: <https://www.mikroe.com/mikroc-pic>. Acesso em 20/11/2020

Outro software utilizado neste trabalho é o programa *Microsoft Visual Studio*. O ambiente de desenvolvimento integrado do *Visual Studio* é um painel de inicialização criativo que pode ser utilizado para editar, depurar e compilar um código e, em seguida, publicar um aplicativo.

Este programa foi utilizado para realizar a interface gráfica do cofre microcontrolado em um computador. Nesta interface, além de poder programar funções para botões e caixas de texto, é possível fazer alterações estéticas, como inserir figuras, textos ou outras informações importantes.

Uma imagem da interface deste programa é apresentada na Figura 41:

Figura 41: Interface do programa Microsoft Visual Studio



Fonte: <https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2019>. Acesso em 23/11/2020

Para a documentação deste trabalho, foram utilizados os programas do pacote *Microsoft Office*. Deste pacote, utilizou-se o programa *Microsoft Word* para a escrita da monografia, *Microsoft Excel* para criação de tabelas e *Microsoft Powerpoint* para criação da apresentação do trabalho.

Diferentes versões do *Microsoft Office* foram utilizadas para edição do trabalho. A versão mais utilizada foi a do *Office 2007*, cuja representação é feita na Figura 42:

Figura 42: Programas Microsoft Office



Fonte: Os autores

Na seção seguinte, será descrita a metodologia utilizada no desenvolvimento de projeto.

3.2. METODOLOGIA UTILIZADA

Neste trabalho foi utilizado o método experimental para teste dos circuitos elétricos desenvolvidos. Os experimentos e testes visaram desenvolver um protótipo com um bom nível de confiabilidade na parte de identificação de cédulas, no sistema de segurança do cofre e comunicação serial desenvolvida.

Para a identificação das cédulas, foi desenvolvido um código fonte em linguagem C para diferenciar cada uma das cédulas da Segunda Família do Real. Foram elaboradas rotinas no código fonte para realizar testes lógicos encadeados (testes IF), verificando assim qual a maior ocorrência de cor em uma cédula.

Se for detectada a ocorrência maior da cor azul, então as outras cédulas que não se enquadram neste padrão são descartadas. Então é feito na sequência outro teste lógico, para a cor verde, por exemplo, e assim por diante, até sobrar uma única opção, que será o valor informado pelo microcontrolador no *display*.

Esta metodologia de calibração também foi testada na parte de verificação da confiabilidade da leitura das cédulas. Nesta etapa, foram utilizados três exemplares de cada uma das sete cédulas da Segunda Família do Real. As medições foram feitas considerando as quatro posições em que as cédulas podem ser inseridas, frente e verso. Desta forma, são feitas no total 84 leituras de valor de cédulas e verificada a taxa de acerto para a metodologia de calibração de cédulas desenvolvida.

Para o teste de segurança, foi feito um teste com dez diferentes senhas. Uma senha por vez é programada, e digitam-se no teclado matricial alguns números para verificar se houve ou não a abertura do cofre.

Quanto à parte da comunicação serial do cofre com computador, foi observada a operação da interface gráfica desenvolvida. Nesta parte, foram testadas as funções estabelecidas na interface, como estabelecer conexão com o cofre, verificar saldo e abrir a porta do cofre.

No capítulo a seguir, serão mostrados os processos de desenvolvimento e resultados preliminares obtidos com as montagens iniciais dos circuitos do cofre.

4. TESTES E RESULTADOS PRELIMINARES

Este capítulo mostra os testes preliminares de funcionamento feito com os circuitos do cofre. Uma parte do capítulo descreve os procedimentos adotados para realização dos testes e a outra parte mostra os resultados preliminares obtidos.

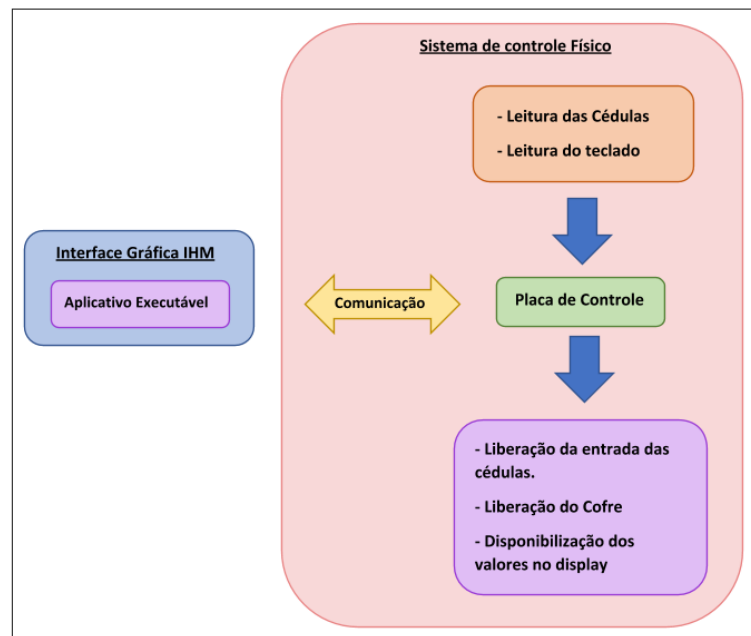
4.1. Procedimentos adotados para testes

4.1.1. Projeto inicial e simulações

O primeiro passo dado neste projeto foi elaborar um diagrama de blocos representando o funcionamento desejado para o cofre. Este diagrama de blocos permitiu ter uma visão geral do que precisaria ser feito, e planejar quais componentes e quais sistemas necessários para o funcionamento do cofre.

Este desenho esquemático é representado na Figura 43:

Figura 43: Diagrama esquemático sobre o projeto do cofre



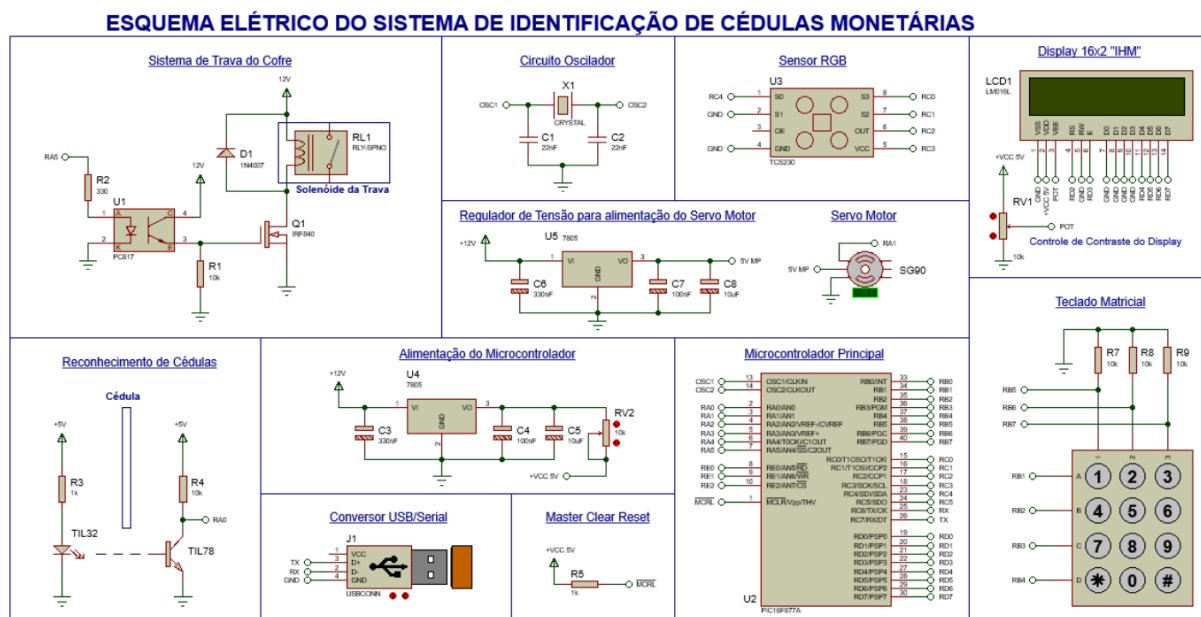
Fonte: Os autores

Na Figura 43, os diagramas de blocos foram organizados seguindo uma lógica de processamento. A leitura das cédulas e do teclado é o *input* (entrada de dados) deste sistema. O processamento é feito com as informações destas leituras, pela placa de controle (microcontrolador), e este fornece uma resposta (*output*), liberando a entrada das cédulas no cofre, abrindo a porta do cofre ou disponibilizando a quantia armazenada no cofre em um *display*. O microcontrolador também controla uma IHM (Interface Humano-Máquina), que será utilizada no projeto para acesso e controle do cofre de forma remota.

Para concretizar este projeto idealizado, foi feita inicialmente a montagem dos circuitos em um simulador para a parte de *hardware* do cofre. O simulador escolhido foi o Proteus, devido a sua facilidade de utilização e familiaridade adquirida durante o curso.

Neste simulador, foi montado o circuito principal do microcontrolador, composto pelo oscilador, a alimentação e o *Master Clear Reset*. Os demais circuitos referem-se aos periféricos utilizados no projeto do cofre. Uma imagem geral do circuito montado para o projeto é apresentada na Figura 44:

Figura 44: Circuito elétrico geral do projeto elaborado no programa Proteus



Fonte: Os autores

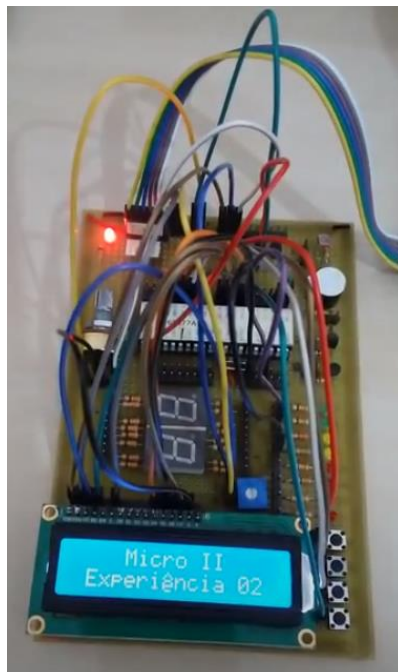
. Esta montagem em simulador permitiu um melhor planejamento e organização das ligações dos componentes ao microcontrolador. A simulação do seu funcionamento, no entanto, não foi possível. Isto ocorreu devido à complexidade de ações desejadas para o cofre e a falta de alguns componentes na biblioteca do Proteus. No circuito da Figura 44, por exemplo, foram criados componentes para representar o sensor RGB e o conversor USB/Serial, pois estes não estavam presentes na biblioteca de componentes do Proteus.

Para o funcionamento dos circuitos apresentados na Figura 44, foi necessário também o desenvolvimento de um código em linguagem C no programa MikroC. Este código é gravado no microcontrolador, para poder fazer as configurações necessárias para o funcionamento dos periféricos ligados a ele. O código completo desenvolvido para este projeto é apresentado no Anexo A deste trabalho.

Para a parte de comunicação serial, foi utilizado o programa Visual Studio para elaborar uma interface gráfica para o cofre em computador. Os códigos-fontes elaborados para esta interface podem ser vistos nos Anexos B e C deste trabalho.

Para a etapa de testes iniciais, foi utilizada uma placa microcontroladora desenvolvida no curso de Microcontroladores. Esta placa está na Figura 45:

Figura 45: Placa desenvolvida no curso de Microcontroladores



Fonte: Os autores

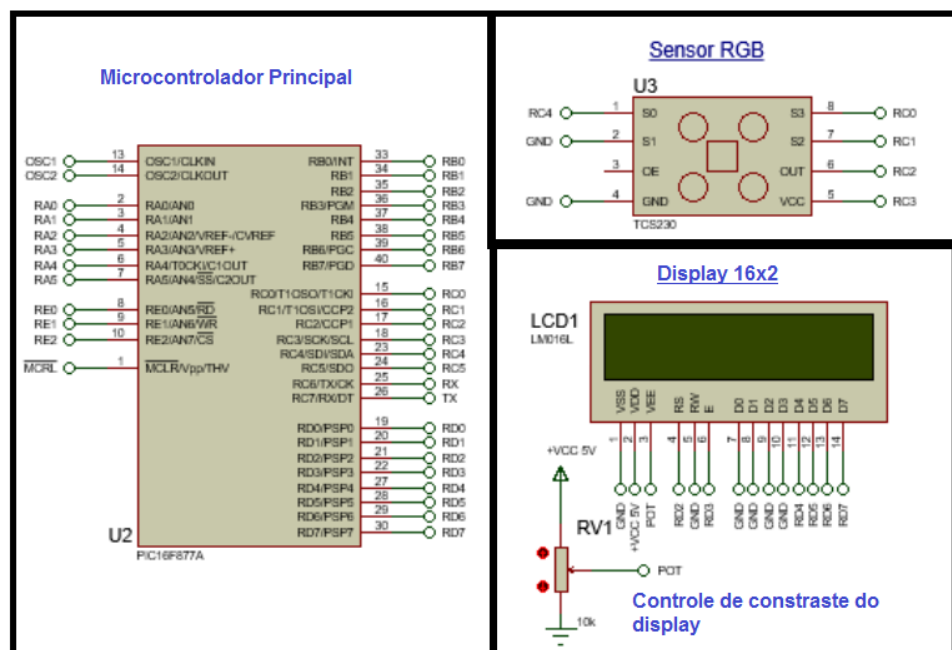
A utilização da placa desenvolvida no curso de Microcontroladores foi conveniente para reduzir o tempo gasto no trabalho. Para esta etapa de testes iniciais, esta placa possui todos os componentes necessários, como o microcontrolador PIC e o *display*. Os demais componentes são ligados ao PIC de forma externa, utilizando a barra de pinos da placa microcontroladora para realizar a conexão.

O primeiro teste com esta placa microcontroladora foi para verificar o funcionamento do sensor RGB, que é descrito a seguir.

4.1.2. Desenvolvimento de um protótipo básico para leitura RGB

Um circuito simples foi montado para verificar a leitura do sensor RGB, ligando diretamente microcontrolador, sensor RGB e display. Como foi utilizada a placa microcontroladora feita em Microcontroladores, foi necessária somente a conexão do sensor RGB. O circuito deste teste é visto na Figura 46:

Figura 46: Circuito esquemático básico para teste do sensor RGB



Fonte: Os autores

No circuito da Figura 46, foram utilizados os pinos RC0 a RC4 para ligação do sensor RGB e pinos RD2 a RD7 do PIC para ligação do *display*. O controle do contraste do *display* é feito por meio de um potenciômetro, para melhorar a visualização das informações.

Neste teste preliminar com o sensor RGB, foi identificada muita interferência da luminosidade durante o processo de leitura, quando ele está à luz ambiente. Como a leitura do sensor precisa ser muito precisa para identificar cédulas com cores parecidas (por exemplo, o caso das cédulas de 2 e 100 reais) foi necessário elaborar um pequeno compartimento fechado para a leitura das cédulas.

Para isso, foi adquirida uma pequena caixa de madeira em MDF (*Medium Density Fiberboard*), com dimensões de 9 cm x 9 cm x 5 cm. A madeira em MDF foi escolhida porque permite facilidade na realização de cortes e possui baixo custo para esta etapa de teste.

Durante os testes, foi verificada ainda uma melhor precisão na leitura do sensor com o interior da caixa pintado com tinta escura. No caso, escolheu-se preto, pois este absorve todas as cores e não reflete nenhuma delas, promovendo assim um ambiente que não interfere tanto na leitura das cédulas.

Nesta caixa foi feito ainda um corte transversal na lateral da caixa, para inserção das cédulas monetárias. Uma imagem do compartimento de leitura das cédulas pode ser vista na Figura 47:

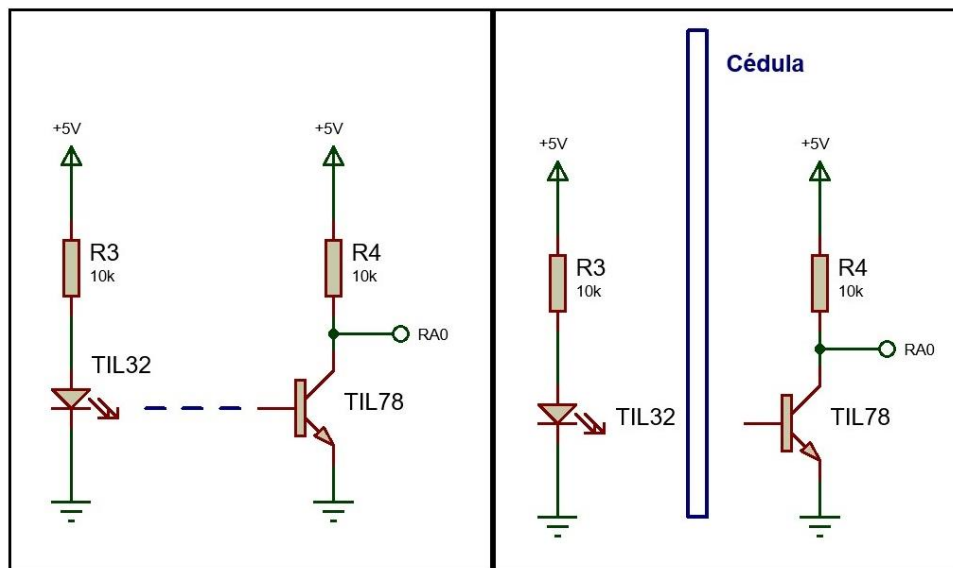
Figura 47: Caixa para leitura RGB.



Fonte: Os autores

Na Figura 47, temos à esquerda uma vista superior da caixa, com interior pintado de preto e à direita a sua vista lateral. Nesta caixa leitora, foi ainda adicionado um circuito para identificar se uma cédula foi inserida e iniciar o processo de leitura. Este sistema simples de reconhecimento de cédulas foi feito com o TIL32 e TIL78, conforme Figura 48:

Figura 48: Sistema de reconhecimento de cédulas.



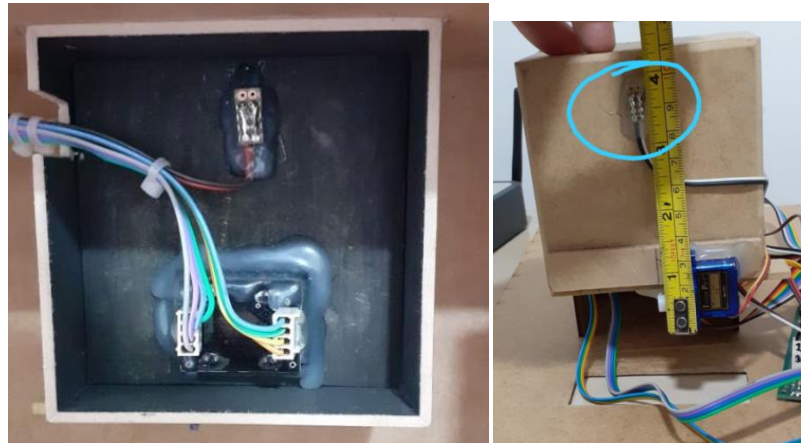
Fonte: Os autores

O sistema de reconhecimento de cédulas é composto por um conjunto de sensor infravermelho e fototransistor (TIL32 e TIL78), posicionados seguindo o esquema da Figura 48. O TIL32 fica sempre aceso, saturando o fototransistor TIL78 e mantendo o pino RA0 com 0 Volts. Quando a cédula é inserida, o fototransistor TIL32 entra em corte, e o pino RA0 fica com 5 V. Neste instante, o microcontrolador aciona a leitura do sensor RGB e o *display* indica que a leitura está ocorrendo.

Para a colocação do sensor RGB, do sensor infravermelho TIL32 e o fototransistor TIL78 numa posição fixa de leitura, foi inserida no interior da caixa leitora uma fina placa de madeira MDF, também pintada de preto, com aberturas para colocação dos componentes. Os componentes então foram colados com cola quente para fixação. Desta forma, deixa-se um pequeno espaço entre a placa e o fundo da caixa para inserir cédula.

Uma imagem da caixa leitora com o sistema de reconhecimento de cédulas e o sensor RGB é apresentada na Figura 49:

Figura 49: Interior da caixa para leitura RGB.

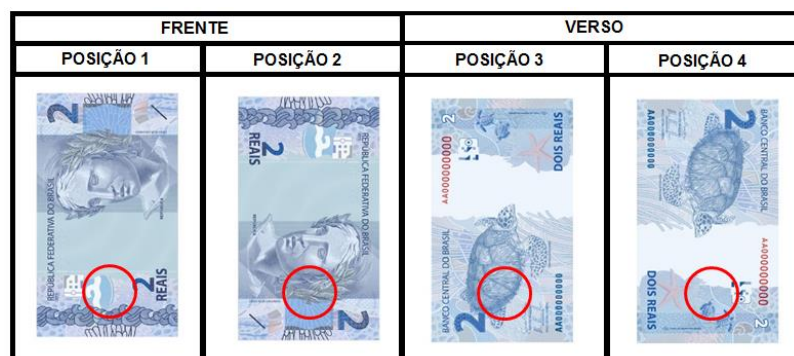


Fonte: Os autores

Na Figura 49, à esquerda e na parte inferior da caixa, está o sensor RGB e logo acima um dos sensores de reconhecimento. O outro sensor se localiza atrás da caixa, conforme destacado na imagem da direita. A cédula é então inserida entre os dois componentes, em uma abertura localizada na parte superior da caixa.

Esta caixa leitora já foi projetada para ser colocada na vertical, para que após cada leitura, a cédula caia por ação da gravidade no cofre. Outro aspecto importante a levar em consideração neste protótipo é o local onde estão sendo feitas as leituras na cédula. Esta estimativa é representada na Figura 50:

Figura 50: Desenho esquemático estimando local de leitura nas cédulas



Fonte: Os autores

Na Figura 50, é possível perceber que a leitura é feita em uma região próxima da borda inferior da nota. Este detalhe é importante para discussão na etapa de resultados da leitura RGB deste trabalho.

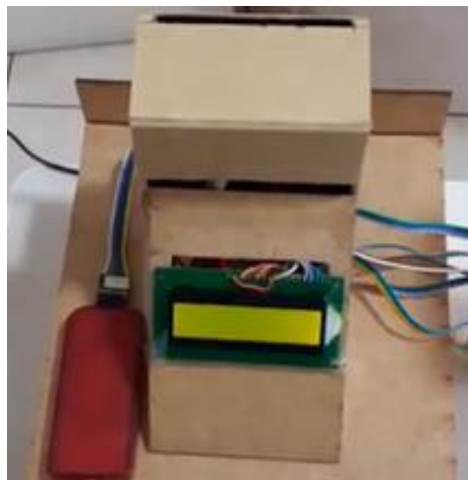
A seguir, será detalhada a montagem do protótipo para fazer a leitura RGB das cédulas e os procedimentos adotados para realizar esta leitura.

4.1.3. Teste para verificação da leitura RGB do sensor TCS230

Este teste foi feito com o protótipo elaborado na etapa anterior, porém com uma alteração. Um *display* foi colocado em uma estrutura em MDF para fácil visualização dos dados na etapa de teste de leitura de cédulas.

A imagem deste protótipo de leitura pode ser observada na Figura 51:

Figura 51: Protótipo para leitura RGB de cédulas.



Fonte: Os autores

Um código-fonte em linguagem C foi elaborado no programa MikroC para mostrar os respectivos valores RGB identificados nas cédulas na tela do *display*. Para elaborar este código, é necessário programar o módulo CCP (Comparação, Captura e PWM) do microcontrolador para fazer as leituras. A leitura de cada valor RGB é feita de forma multiplexada, acionando um conjunto de fotodiodo do sensor por vez, conforme configuração da Tabela 3 apresentada neste trabalho.

Para a leitura, foram utilizadas seis cédulas (2, 5, 10, 20, 50, 100 reais), começando pelos modelos mais recentes, da Segunda Família do Real. A leitura da cédula de 200 reais, como foi adquirida depois desta etapa, será mostrada durante o desenvolvimento do protótipo finalizado, na seção 5.2.

Para a verificação da leitura do sensor RGB nas cédulas, foi feita apenas uma leitura de cada cédula. Esta etapa foi feita visando identificar qual o padrão RGB lido nas cédulas, e comparar na teoria se este valor lido correspondente ao sistema RGB. Para esta comparação, foi utilizando um software online [26], que retorna a cor correspondente ao valor RGB informado e seu código hexadecimal. Os resultados deste teste são apresentados na seção 4.2.1.

4.1.4. Identificação dos valores monetários das cédulas

Nesta etapa, foi elaborado um código-fonte em linguagem C para o microcontrolador identificar o valor monetário respectivo de cada cédula de acordo com o padrão RGB. O valor identificado é então mostrado na tela do *display*.

Esta identificação foi feita realizando sucessivos testes lógicos (IF), verificando sequencialmente qual a cor RGB predominante na cédula (Vermelho, Verde ou Azul), retornando o valor da cédula resultante deste teste no *display*. A calibração destas cédulas foi feita utilizando as quatro posições em que a cédula pode ser inserida. Os resultados deste teste são apresentados na seção 4.2.2.

4.1.5. Desenvolvimento do sistema do servo motor

Nesta etapa, foi desenvolvido o sistema do servo motor. Ele é colocado na parte inferior da caixa leitora, para a cédula cair dentro do cofre após a leitura.

Um sistema para alimentação do cofre foi elaborado nesta etapa, pois até então, a alimentação dos circuitos era feita com o gravador Pickit 3 (5 V). Uma fonte para alimentação, de 12 V / 3 A, foi adquirida para esta etapa do projeto. Este valor de 12 V foi escolhido devido à alimentação da trava eletrônica, que seria desenvolvida em etapas posteriores.

Uma imagem da fonte de 12 V utilizada no projeto é mostrada na Figura 52:

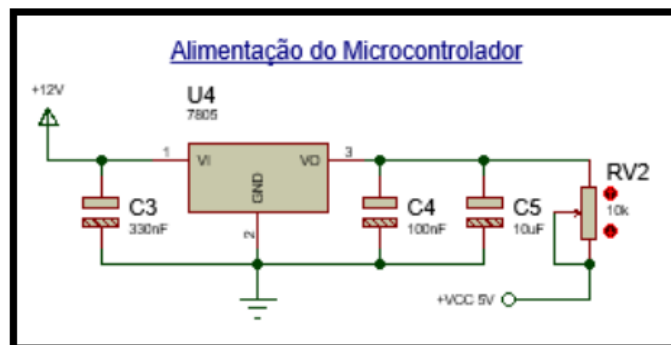
Figura 52: Fonte de alimentação 12 V / 3 A



Fonte: Os autores

Para a alimentação do microcontrolador PIC, que utiliza tensão de 5 V, foi montado um circuito regulador de tensão. Este circuito regulador é obtido com o CI 7805, conforme mostrado na Figura 53:

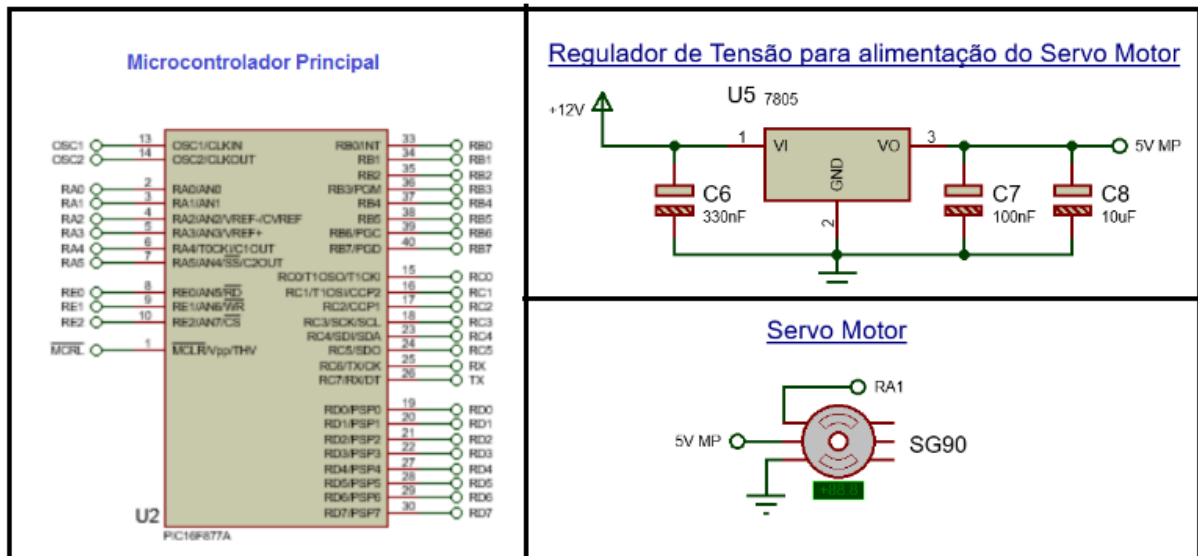
Figura 53: Circuito de regulação 12 V para 5 V



Fonte: Os autores

Como é possível ver na Figura 53, além do CI 7805, utiliza-se capacitor para filtragem do sinal contra possíveis ruídos e um potenciômetro para ajuste fino da tensão em 5 V na saída. Na prática, este valor precisou ser ajustado para 4,7 V, que é o valor real fornecido pelo gravador Pickit3 quando conectado à entrada USB do computador. O sistema com servo motor também foi desenvolvido com regulador de tensão, conforme mostrado na Figura 54:

Figura 54: Circuito para teste do sistema servo motor



Fonte: Os autores

Na Figura 54, é mostrado o sistema do servo motor SG-90 e um regulador de tensão para alimentação. O servo motor é conectado ao pino RA1 do microcontrolador.

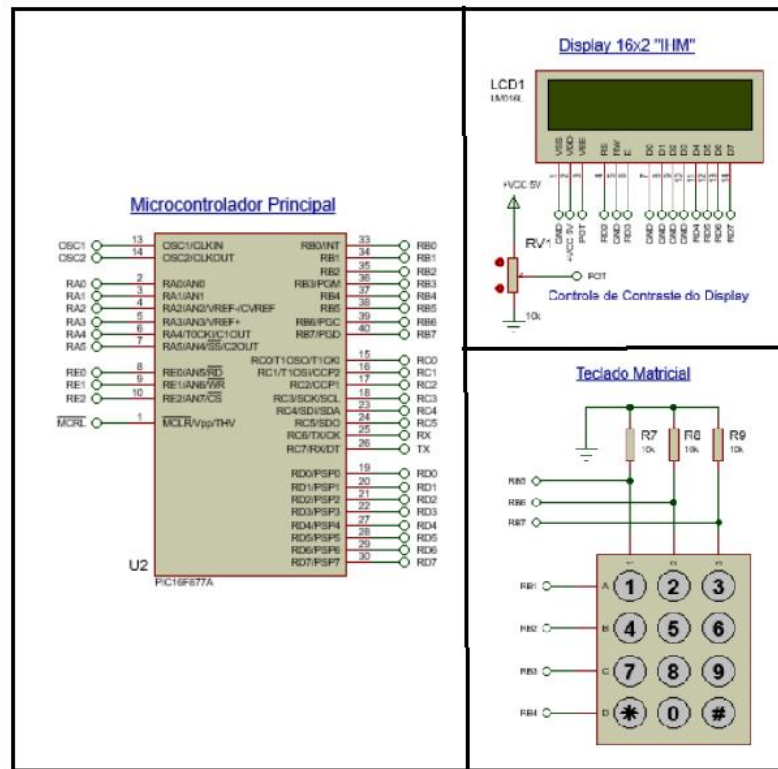
O servo motor possui três ligações: um pino de alimentação ligada na saída do regulador, um pino aterrado e outro ligado ao microcontrolador. Por meio deste pino ligado ao microcontrolador, é dado o comando para o servo motor realizar a ação de abertura da porta da caixa leitora. Um teste de inserção de cédulas na caixa leitora foi feito para verificar o funcionamento do servo motor. Os resultados do teste deste sistema servo motor são mostrados na seção 4.2.3.

4.1.6. Desenvolvimento do sistema com teclado matricial

Para a etapa de digitação de senhas e configuração do cofre, é necessária a elaboração de um sistema com teclado matricial. Neste sistema, além do próprio teclado e do microcontrolador, é necessário também utilizar um *display* para visualizar as informações que estão sendo digitadas.

O circuito para inserção do teclado matricial é mostrado na Figura 55:

Figura 55: Circuito para teste do teclado matricial



Fonte: Os autores

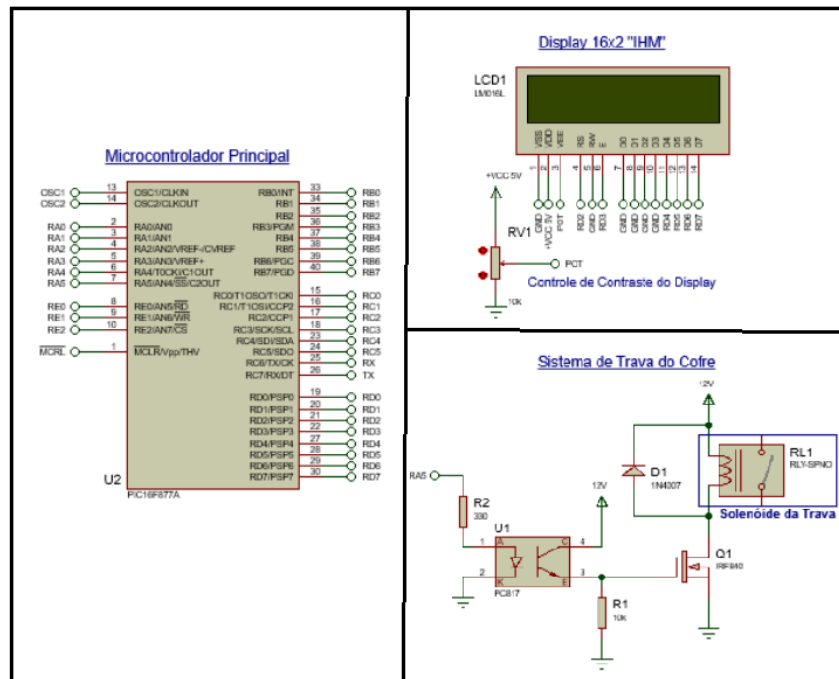
O teclado matricial possui 12 teclas, com números de 0 a 9 e os caracteres * e #. A comunicação do teclado matricial é feita por meio de sete pinos, RB1 a RB7. Neste circuito são utilizados três resistores de *pull-up* de 10kΩ.

Neste teste do teclado, foram verificados se os números digitados correspondem ao valor apresentado em um display. Nesta etapa, foi também feito um teste para salvar os dados digitados na memória EEPROM do microcontrolador, O resultado destes testes é visto na seção 4.2.4.

4.1.7. Desenvolvimento do sistema com trava eletrônica

Para a abertura e fechamento do cofre, foi desenvolvido um sistema com trava eletrônica acionado por senha. O circuito do sistema com trava eletrônica é apresentado na Figura 56:

Figura 56: Circuito para teste do sistema com trava eletrônica



Fonte: Os autores

No circuito com trava eletrônica, um optoacoplador (PC817) é conectado ao pino RA5 do microcontrolador. Este optoacoplador é um diodo emissor de luz infravermelho e um fototransistor de silício NPN, muito utilizado em controle de relés, motores, entre outros dispositivos.

O acoplador óptico ou optoacoplador é um dispositivo na forma de circuito integrado que transmite sinais sem contato direto, apenas utilizando um fototransistor e um sinal luminoso em seu interior. Assim, é possível conectar dois circuitos diferentes mantendo-os completamente isolados entre si, garantindo o bom desempenho e segurança dos dois dispositivos [27].

Por meio do pino RA5, é feito o controle do sistema. Quando é digitada a senha correta no cofre, um sinal é enviado do pino RA5, ligando o LED do PC817. O LED, por sua vez, quando aceso, satura o fototransistor. A saída do PC817 é conectada à base de um transistor IRF540, que gera na saída uma corrente que aciona o solenoide da trava, liberando a porta do cofre. Um diodo em paralelo ao solenoide previne possíveis correntes reversas que podem danificar o circuito.

Para o desenvolvimento do sistema com trava eletrônica, foi elaborado um pequeno menu para digitar uma senha, apresentado na Figura 57:

Figura 57: Menu para senhas



Fonte: Os autores

O menu para senhas possui quatro opções. A primeira opção no menu é para programação de senha. A segunda opção é para acessar o saldo do cofre. A terceira opção é utilizada para abrir a porta. A quarta opção é para sair deste menu.

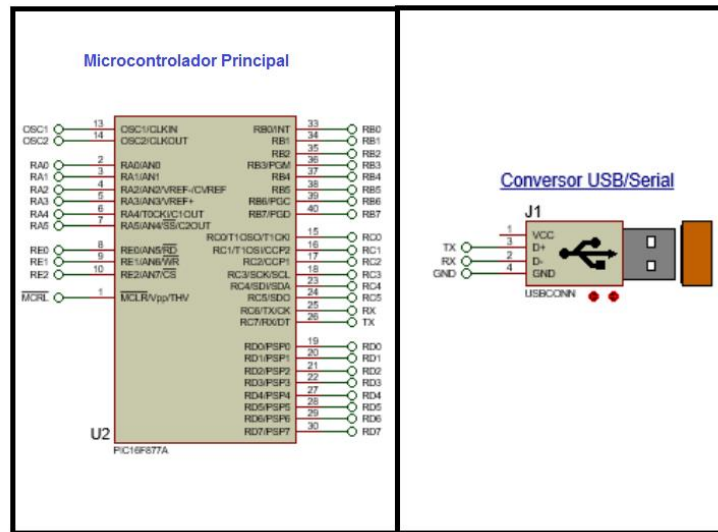
No teste de funcionamento, foi utilizada somente a opção de abrir a porta. Uma senha foi previamente programada. Então, digitada esta senha programada é verificado se a trava recolhe, liberando a porta do cofre. O resultado deste teste é apresentado na seção 4.2.5.

4.1.8. Desenvolvimento do sistema para comunicação serial

Para desenvolver este sistema, foi necessário fazer a parte física do projeto, que é a conexão do conversor USB/serial ao circuito microcontrolador, e a parte de elaboração da interface gráfica, que foi feita no programa Visual Studio.

O circuito físico para comunicação serial é representado na Figura 58:

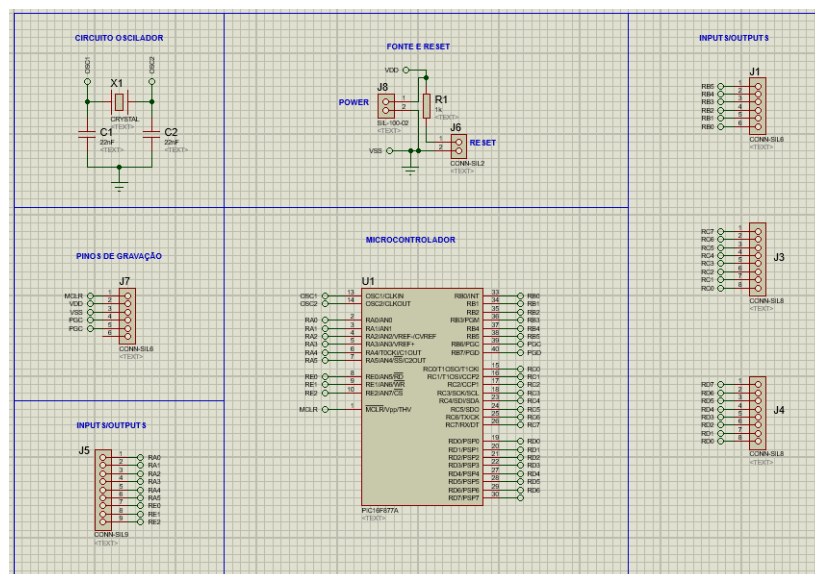
Figura 58: Circuito para teste de comunicação serial



Fonte: Os autores

Neste circuito, é necessária somente a ligação do conversor TTL CH340 aos pinos RX e TX do microcontrolador. A outra entrada do conversor pode ser conectada a uma porta USB de um computador. Nesta etapa, foi desenvolvida uma placa microcontroladora exclusiva para a parte de comunicação. Este projeto, feito no Proteus, é mostrado na Figura 59:

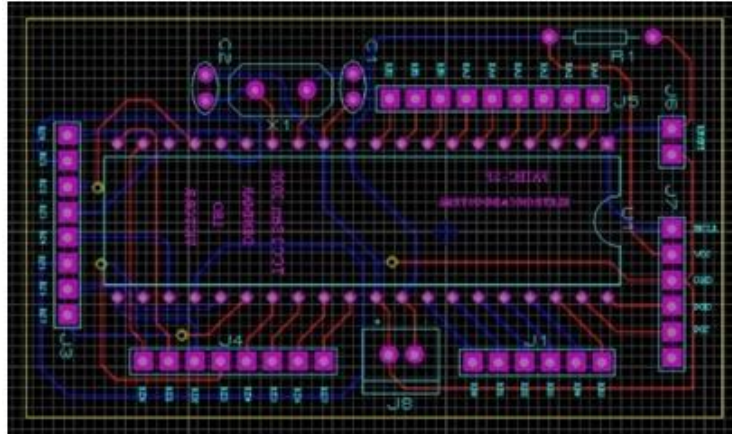
Figura 59: Circuito elétrico da placa para comunicação serial



Fonte: Os autores

O circuito *Schematic* da placa para comunicação serial foi transportado para o programa ARES, cujo *layout* pode ser observado na Figura 60:

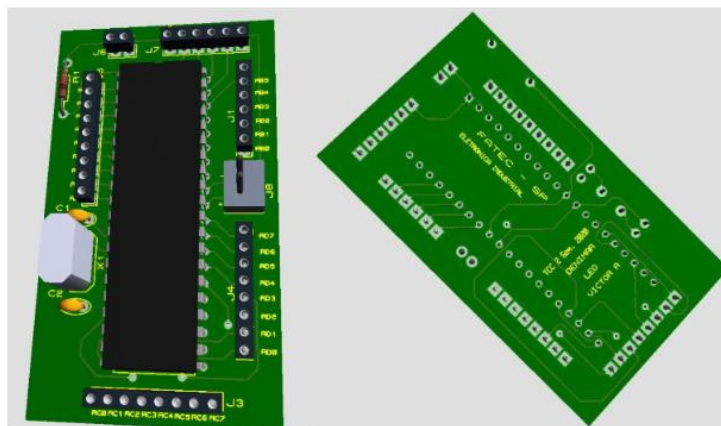
Figura 60: *Layout* da placa para comunicação serial



Fonte: Os autores

A montagem dos componentes foi realizada somente no lado top, e as trilhas nos lados *top* e *bottom* da placa. As conexões na parte *top* da placa são apresentadas em vermelho e na parte *bottom* em azul. A placa possui as dimensões de 68,5 mm por 39,0 mm. No Proteus é possível também ter uma prévia 3D da placa projetada. Esta imagem pode ser vista na Figura 61:

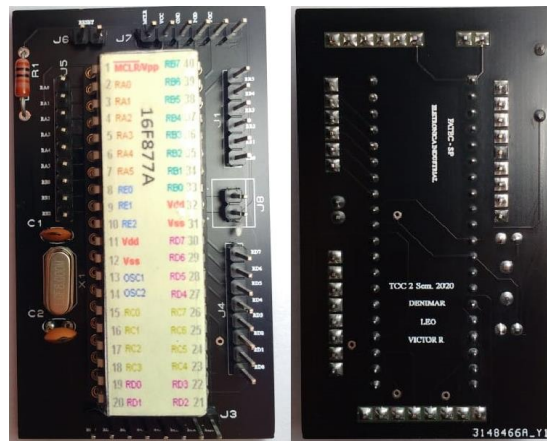
Figura 61: Imagem 3D da placa para comunicação serial



Fonte: Os autores

Após conferência e verificação que a placa não continha erros de layout, o projeto foi encaminhado para fabricação, pela empresa JLCPCB, na China. Uma imagem da placa desenvolvida pode ser vista na Figura 62:

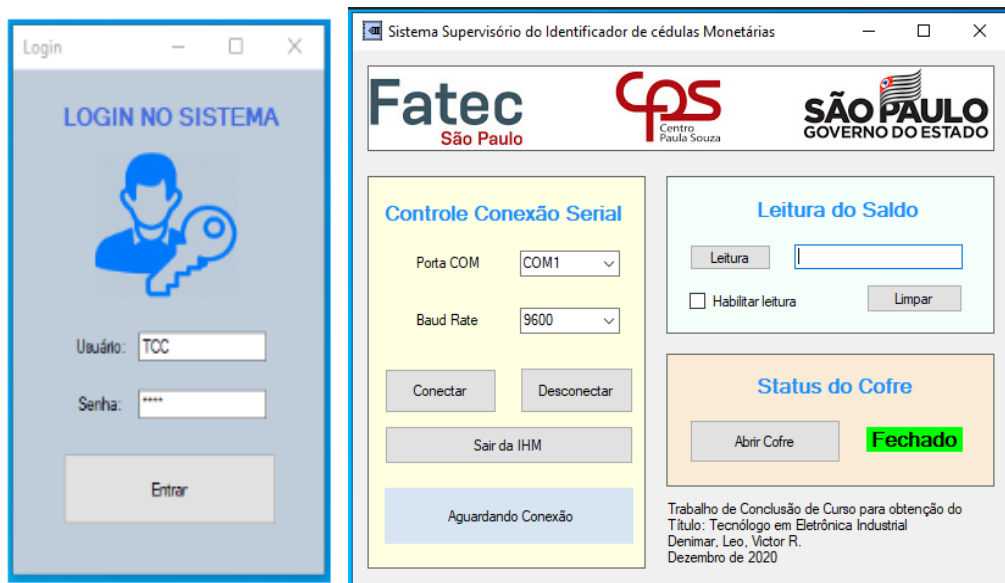
Figura 62: Placa desenvolvida finalizada



Fonte: Os autores

Na parte de elaboração da interface gráfica, foi feita uma tela de *login* com usuário e senha para acesso ao cofre. Somente com a digitação dos dados corretos é feito o acesso a tela do cofre. Estas interfaces podem ser vistas na Figura 63:

Figura 63: Interface gráfica do cofre identificador



Fonte: Os autores

Na Figura 63, temos à esquerda a tela de *login* e à direita a tela do cofre. Os dados iniciais programados desta interface foram usuário: TCC (letras maiúsculas) e senha: 1234, podendo ser alterados depois pelo usuário. À direita, temos a tela do cofre, programada com três opções de seleção.

A primeira opção de seleção é a Conexão Controle Serial. Nela, há uma caixa de seleção “Porta COM” para realizar a conexão com o PIC. Esta porta é identificada automaticamente quando é conectado um microcontrolador. Na caixa de seleção “*Baud Rate*”, é possível selecionar a taxa de transferência de dados. A conexão pode ser encerrada a qualquer momento, clicando em “Desconectar” ou saindo da tela do cofre, clicando em “Sair da IHM”.

A segunda opção é a leitura de saldo. Esta opção foi programada para verificar a quantia armazenada pelo cofre, selecionando a caixa “Habilitar leitura” e depois clicando em leitura. Para atualizar o valor em tempo real, é necessário limpar o valor e depois clicar em leitura.

A terceira opção é o *Status* do cofre. Inicialmente, o cofre é mostrado como “Fechado”, na cor verde, indicando visualmente que o cofre está seguro. Quando é selecionado “Abrir cofre”, o status do cofre muda para “Aberto”, em vermelho, indicando visualmente que o cofre não está seguro, e um comando é enviado para o microcontrolador abrir a porta do cofre.

Nesta etapa de desenvolvimento, esta interface gráfica não foi testada com o cofre. Outros meios foram usados para simular este funcionamento. A leitura do saldo, por exemplo, foi feita escrevendo um valor *string* na memória do microcontrolador para ver se esta informação era recebida pela interface gráfica.

Já o teste de funcionamento da abertura da porta do cofre foi feito com um LED. Quando é enviado o comando de abrir a porta, este LED deve acender, indicando que a comunicação ocorreu com sucesso. Os resultados deste teste são mostrados na seção 4.2.6.

A seguir, serão apresentados os resultados dos testes preliminares descritos nesta parte inicial do Capítulo 4.

4.2. RESULTADOS PRELIMINARES

4.2.1. Resultados do teste de leitura das cédulas com sensor RGB

Seguindo o procedimento descrito na seção 4.1.3, foram feitos os testes de calibração para as cédulas monetárias.

Neste teste, foi feito um vídeo em celular mostrando a calibração de cada uma das seis cédulas. Para cada cédula lida, foi feita uma captura de tela, para verificar quais os valores mostrados no display. A imagem do *display* foi ampliada para melhor visualizar os dados. O resultado deste teste está nas Figuras 64 a 66:

Figura 64: Teste de leitura RGB (cédula de 2 e 5 reais).



Fonte: Os autores

Figura 65: Teste de leitura RGB (cédula de 10 e 20 reais).



Fonte: Os autores

Figura 66: Teste de leitura RGB (cédula de 50 e 100 reais)



Fonte: Os autores

Testes com outras cédulas, como as mais antigas (da Primeira Família do Real), constataram dificuldade do sensor TCS230 em fazer a identificação correta das cédulas. Uma das explicações possíveis para problemas nestas cédulas é que elas estão a mais tempo em circulação, retendo mais sujeira. Cédulas com defeitos, como manchas ou rasgos também apresentaram dificuldade na leitura, tanto da Primeira quanto da Segunda Família do Real.

Sendo assim, a calibração das cédulas foi direcionada à análise somente das cédulas da Segunda Família do Real, em bom estado de conservação (sem defeitos).

Os dados da leitura do sensor RGB das Figuras 64 a 66 foram utilizados para verificar os valores lidos. Estes dados são apresentados na Tabela 6, mostrada a seguir:

Tabela 6: Dados da leitura RGB das cédulas

| CÉDULA | Leitura Sensor RGB | | | |
|-----------|--------------------|-----------|----------|-----------|
| | R (Red) | G (Green) | B (Blue) | C (Clear) |
| 2 reais | 111 | 95 | 66 | 27 |
| 5 reais | 71 | 90 | 65 | 24 |
| 10 reais | 69 | 112 | 92 | 28 |
| 20 reais | 46 | 74 | 77 | 21 |
| 50 reais | 76 | 105 | 101 | 30 |
| 100 reais | 121 | 81 | 59 | 25 |

Fonte: Os autores

Os valores lidos diretamente pelo sensor RGB não representam exatamente o padrão RGB das cédulas. Para entender o porquê disso, fazemos a seguinte análise.

Conforme discutido sobre o funcionamento deste sensor (TCS 230), ele é um conversor de luz em frequência. Para fazer a leitura dos valores RGB, é programado o módulo CCP do microcontrolador. Neste módulo de captura, os dados são obtidos na forma de período (T), que é o inverso da frequência (f), ou seja, $f=1/T$. Ou seja, quando o sensor RGB indica que uma das componentes RGB é menor que as outras, significa que esta é a que está em maior presença no objeto analisado.

Isto pode ser observado nos dados das cédulas de 2 e 100 reais da Tabela 6. Estas cédulas, que são notadamente azuis, possuem a menor componente azul (*Blue*), comparando com os outros valores lidos.

Por este motivo, é necessário fazer a conversão dos valores da Tabela 6 para verificar o padrão RGB das cédulas. De forma prática, sabe-se que o TCS230 ilumina o objeto analisado com uma luz branca, de forma uniforme, conforme já discutido. A luz branca, por sua vez, no sistema RGB, é a somatória de todas as suas cores primárias, ou seja, a soma de vermelho (R), verde (G) e azul (B).

No sensor RGB, foi comentado também que este possui quatro conjuntos de filtros, um para vermelho, um para verde, um para azul, e outro sem filtro. Devido à presença destes filtros, as cores predominantes em cada cédula aparecem com valores menores, ou seja, o valor da leitura é atenuado proporcionalmente à presença da respectiva componente RGB.

Para a conversão dos valores, foi adotada uma simplificação, admitindo que os valores ideais para a luz branca são *Red (R)= 255, Green (G)=255, Blue (B)=255*. O padrão RGB de cada cédula pode então ser obtido fazendo a subtração do valor de 255, por cada um dos valores da Tabela 6.

Os valores de *Clear (C)*, também serão convertidos. Estes valores, no entanto, representam apenas o nível de luminosidade do ambiente de leitura. Como foi utilizado um ambiente controlado, estes valores apresentaram pouca variação entre as medições e não interferem na análise do padrão RGB.

Uma nova tabela com os dados corrigidos da leitura RGB estão representadas na Tabela 7:

Tabela 7: Valores corrigidos da leitura RGB nas cédulas







| CÉDULA | Leitura corrigida do Sensor RGB | | | |
|-----------|---------------------------------|--------------------|-------------------|--------------------|
| | R (<i>Red</i>) | G (<i>Green</i>) | B (<i>Blue</i>) | C (<i>Clear</i>) |
| 2 reais | 144 | 160 | 189 | 228 |
| 5 reais | 184 | 165 | 190 | 231 |
| 10 reais | 186 | 143 | 163 | 227 |
| 20 reais | 209 | 181 | 178 | 234 |
| 50 reais | 179 | 150 | 154 | 225 |
| 100 reais | 134 | 174 | 196 | 230 |

Fonte: Os autores

Na Tabela 7, é possível identificar com maior clareza as componentes com maior presença nas cédulas do sistema monetário brasileiro. Os dados desta tabela podem então ser utilizados para comparação no *software* para análise de valores RGB [26].

Os resultados obtidos com este *software* são apresentados na Figura 67:

Figura 67: Verificação dos valores RGB. (2, 5, 10, 20, 50 e 100 reais)

| | |
|--|--|
| <p>Cédula de 2 reais</p> <p>R = 144 G = 160 B = 189</p> <p>Código RGB: #90a0bd</p>  | <p>Cédula de 5 reais</p> <p>R = 184 G = 165 B = 190</p> <p>Código RGB: #b8a5be</p>  |
| <p>Cédula de 10 reais</p> <p>R = 186 G = 143 B = 163</p> <p>Código RGB: #ba8fa3</p>  | <p>Cédula de 20 reais</p> <p>R = 209 G = 181 B = 178</p> <p>Código RGB: #d1b5b2</p>  |
| <p>Cédula de 50 reais</p> <p>R = 179 G = 150 B = 154</p> <p>Código RGB: #b3969a</p>  | <p>Cédula de 100 reais</p> <p>R = 134 G = 174 B = 196</p> <p>Código RGB: #86aec4</p>  |

Fonte: Os autores

Na Figura 67, observa-se que os resultados gerados são cores que lembram visualmente as observadas nas cédulas monetárias. As cédulas de 2 e 100 reais apresentam tom azul, a de 5 reais, tom roxo, a de 10 reais, tom vermelho e a de 50 reais um tom marrom. A cédula de 20 reais apresentou uma coloração com um tom marrom claro, que pode ter ocorrido devido a alguma mancha ou sujeira.

4.2.2. Resultados da identificação do valor das cédulas

Seguindo o procedimento descrito na seção 4.1.4, foram feitos os testes de identificação dos valores das cédulas monetárias com sensor RGB.

Neste teste, foi feito um vídeo em celular mostrando a calibração de cada uma das seis cédulas. Para cada cédula lida, foi feita uma captura de tela, para verificar quais os valores mostrados no *display*. Nesta etapa, não foi necessário fazer nenhuma edição na imagem, pois os valores apresentados no *display* apresentaram um bom contraste.

O resultado deste teste está nas Figuras 68 a 70:

Figura 68: Calibração da cédula de 2 e 5 reais



Fonte: Os autores

Figura 69: Calibração das cédulas de 10 e 20 reais



Fonte: Os autores

Figura 70: Calibração da cédula de 50 e 100 reais



Fonte: Os autores

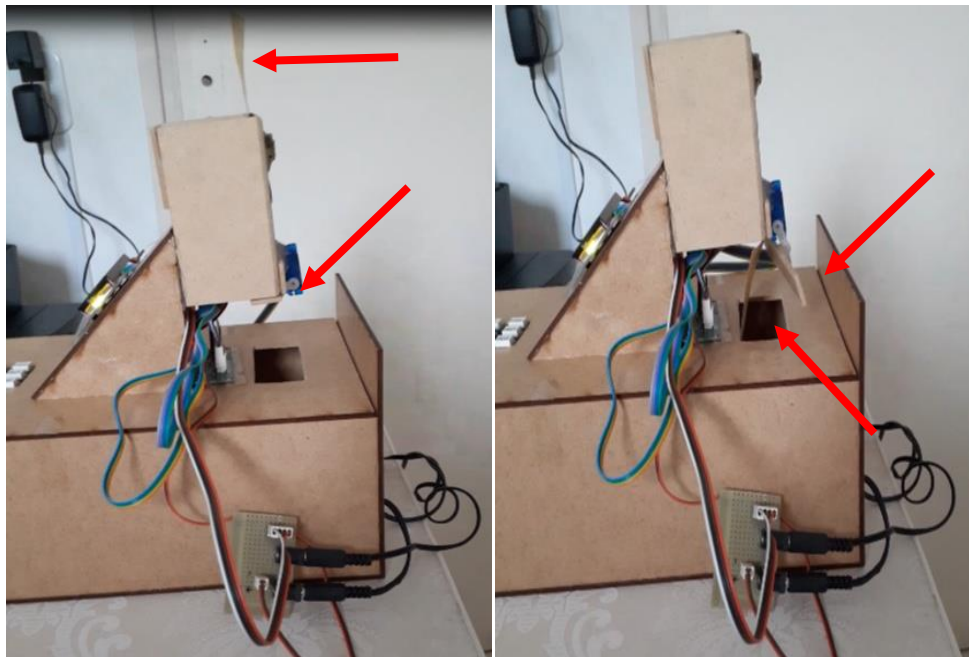
Neste teste, foi possível ver que os dados informados no display correspondem aos valores monetários das cédulas. Neste teste, é possível ver que a identificação do valor das cédulas ocorreu para diferentes posições de inserção, devido ao processo de calibração feito.

4.2.3. Resultado do sistema desenvolvido com servo motor

Seguindo o procedimento descrito na seção 4.1.5, foram feitos os testes de verificação do funcionamento do sistema servo motor.

O funcionamento do sistema do servo motor pode ser visto na Figura 71:

Figura 71: Teste do sistema do servo motor



Fonte: Os autores

Na Figura 71, à esquerda, é indicado com uma seta que a cédula ainda sendo identificada. Logo abaixo, nesta mesma imagem, é mostrado o servo motor. O servo motor possui uma pequena palheta que, enquanto está ocorrendo a leitura, mantém a porta da caixa leitora fechada.

Na imagem à direita, após a leitura, é mostrada a cédula caindo no cofre, após a abertura da porta da caixa leitora. A abertura da caixa é possível porque quando termina a leitura, o microcontrolador manda um comando para a palheta do servo motor rotacionar, liberando a porta que até então estava fechada e a cédula cai no cofre.

Após alguns segundos, o servo motor volta à posição original, fechando a porta, e possibilitando a leitura de uma nova cédula.

4.2.4. Resultado do sistema desenvolvido com teclado matricial

Seguindo o procedimento descrito na seção 4.1.6, foram feitos os testes de verificação do sistema com teclado matricial.

O resultado do teste de digitação de números no teclado matricial pode ser visto na Figura 72:

Figura 72: Teste com teclado matricial



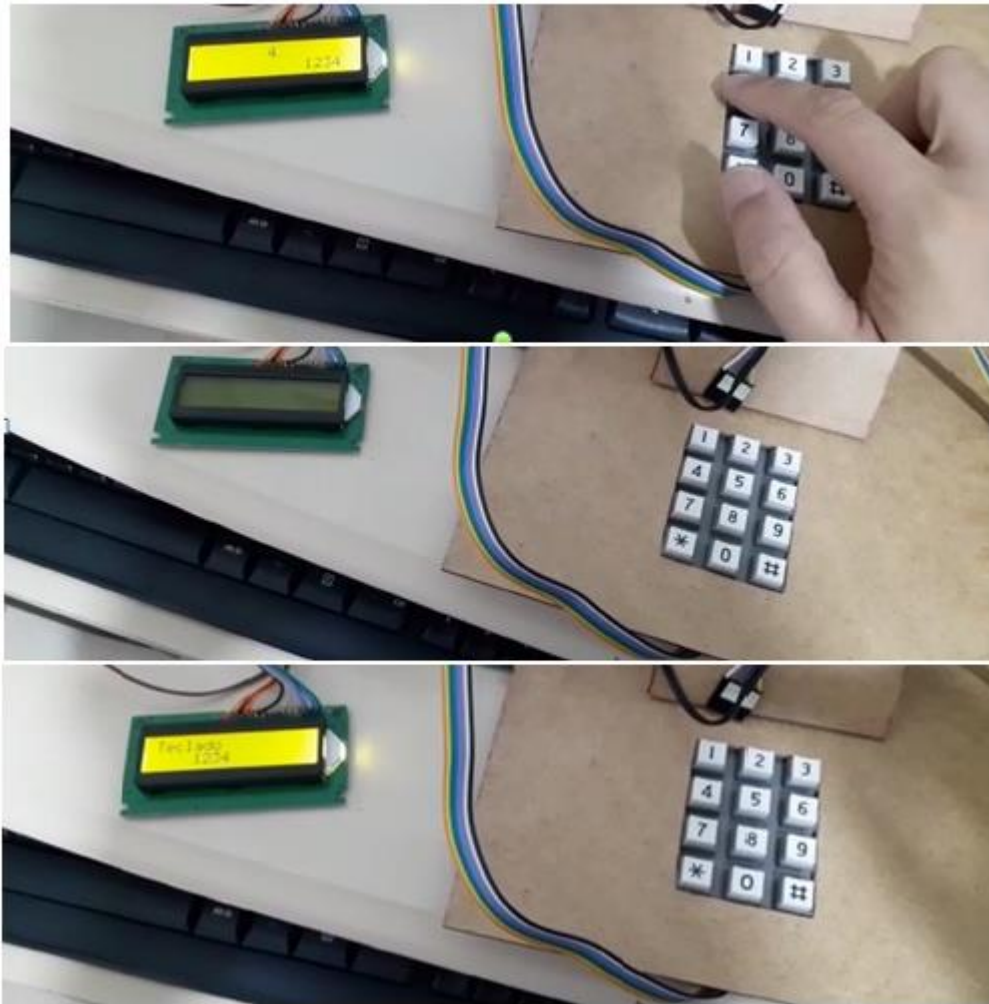
Fonte: Os autores

Neste teste do teclado matricial, foi possível verificar o tempo de resposta e se o número digitado foi apresentado corretamente. Após a digitação da sequência numérica, os respectivos números digitados apareceram na tela do display, de forma rápida, indicando que o teclado estava funcionando de forma correta.

Nesta etapa, foi feito o teste da memória EEPROM. Este teste consistiu na digitação de um número aleatório no teclado matricial. Na sequência, o microcontrolador foi desligado, e depois ligado novamente, para verificar se o número digitado ainda pode ser visualizado.

Este teste pode ser observado na Figura 73:

Figura 73: Testes para salvar dados na memória EEPROM



Fonte: Os autores

Neste teste da memória EEPROM, foi digitado inicialmente um número com quatro algarismos, no caso “1234”. Na sequência, o sistema foi desligado e ao ser religado, o número digitado permaneceu na memória como desejado.

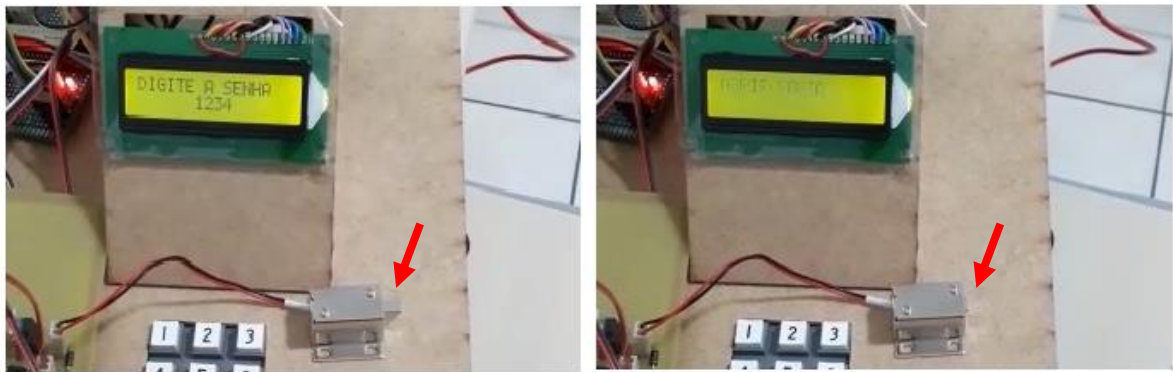
Após esta etapa de testes iniciais, a utilização da memória EEPROM foi aprimorada, configurando o seu funcionamento por meio da elaboração do código-fonte. A memória EEPROM, além de armazenar a senha do cofre, também é importante para armazenar o saldo, ambos devendo ser mantidos mesmo com o desligamento do cofre.

4.2.5. Resultado do sistema desenvolvido com trava eletrônica

Seguindo o procedimento descrito na seção 4.1.7, foram feitos os testes do sistema com trava eletrônica.

A resposta da trava eletrônica com a digitação da senha é vista na Figura 74:

Figura 74: Teste da trava solenoide com senha



Fonte: Os autores

Como pode ser visto na Figura 74, antes da digitação da senha, a trava encontra-se na posição original, ou seja, deixa o cofre fechado. Alguns instantes após a digitação da senha correta, a língua da trava eletrônica recolhe, abrindo assim a porta do cofre.

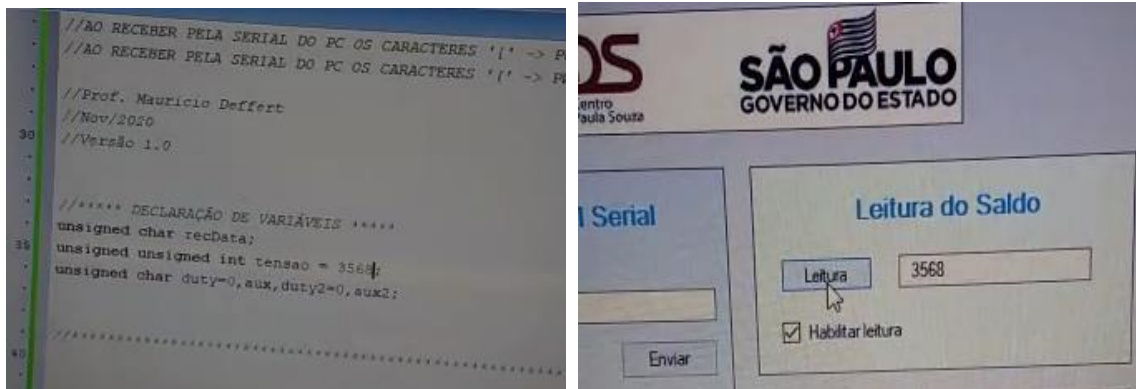
A língua desta trava eletrônica, após alguns segundos, retorna à posição original. Com isso, o fechamento do cofre deve ser manual, o próprio usuário precisa empurrar a porta até travar o cofre.

4.2.6. Resultado do sistema para comunicação serial

Seguindo o procedimento descrito na seção 4.1.8, foram feitos os testes de comunicação serial.

. O resultado do teste de leitura de saldo pode ser visto na Figura 75:

Figura 75: Teste para leitura na interface gráfica (teste *string*)

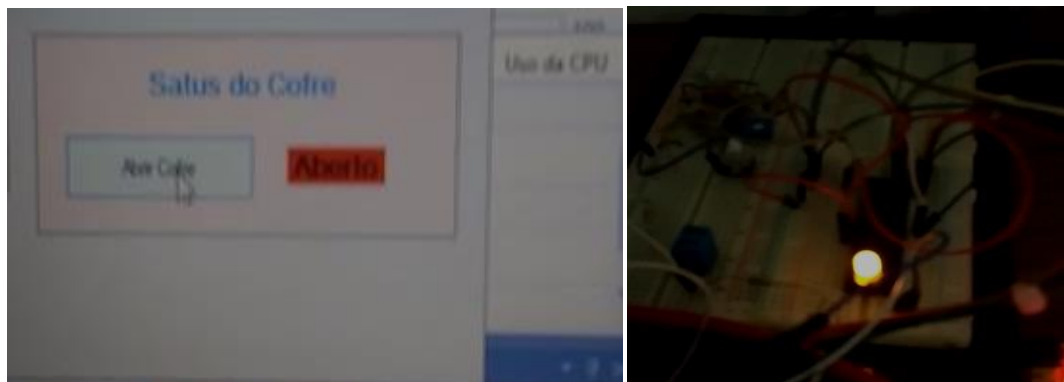


Fonte: Os autores

Como pode ser vista na Figura 75, o valor *string* enviado do microcontrolador para a interface gráfica ocorreu com sucesso. Neste teste, foi possível visualizar o número enviado como *string* na tela de leitura na interface gráfica.

Já o resultado do teste da abertura de porta pode ser visto na Figura 76:

Figura 76: Teste para abrir porta na interface gráfica (teste LED)



Fonte: Os autores

Na Figura 76, observa-se que o comando enviado para abrir o cofre acendeu o LED, indicando que houve comunicação serial entre a interface gráfica e microcontrolador. O capítulo seguinte será dedicado ao desenvolvimento do protótipo finalizado. Nesta etapa, será descrito todos os procedimentos feitos para obter a versão definitiva do cofre.

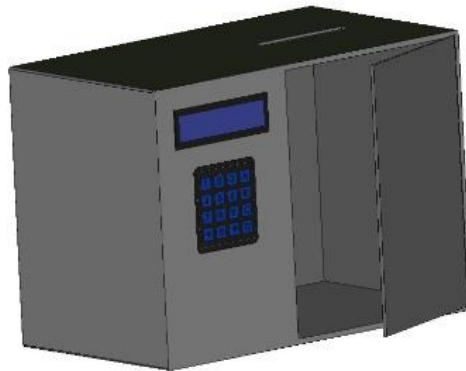
5. DESENVOLVIMENTO DO PROTÓTIPO FINALIZADO

Neste capítulo, serão apresentadas as etapas de desenvolvimento realizadas para obter o protótipo finalizado do cofre, desde o projeto do layout, até a sua montagem completa.

5.1. Projeto do layout do cofre

Para a montagem física da caixa do cofre, foi elaborado inicialmente um esboço de *layout* fazendo um desenho livre. Uma imagem da estrutura desejada para o cofre é apresentada na Figura 77:

Figura 77: Esboço da estrutura do cofre para o projeto



Fonte: Os autores

Por questão de segurança e estética, foi pensado um compartimento reservado para a colocação dos circuitos (à esquerda) e outro para armazenagem das cédulas (à direita). Na parte frontal, à esquerda, seriam posicionados o *display* LCD e logo abaixo o teclado matricial, facilitando a operação do cofre.

Para construção física do cofre, o material escolhido foi madeira MDF. Foram utilizados pedaços de madeira MDF, devidamente dimensionados para a montagem de um cofre com dimensão de 30 cm x 25 cm x 15 cm. Esta caixa foi feita com material resistente para ter um bom suporte mecânico para as montagens.

Após a montagem da caixa externa, foi feito o corte da madeira nas posições onde seriam inseridos o *display* e o teclado. Neste primeiro modelo, foi colocada a porta com um pino na parte superior para permitir o movimento de abertura. Uma primeira imagem do cofre montado é apresentada na Figura 78:

Figura 78: Estrutura física inicial do cofre



Fonte: Os autores

Ainda na parte externa, foi feita pintura do cofre com uma tinta cinza. Neste etapa, também foram acrescentadas as dobradiças na porta, conforme imagem apresentada na Figura 79:

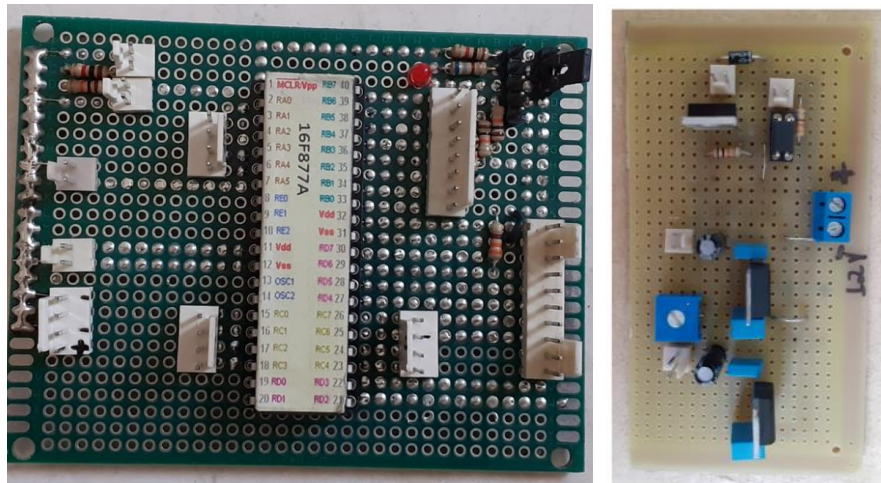
Figura 79: Cofre pintado



Fonte: Os autores

Finalizada a parte externa do cofre, foi pensada então a sua parte interna. Em primeiro lugar, foi realizada a montagem do compartimento de circuitos. Pensando na disposição dos componentes e na forma como seriam feitas as ligações, foram desenvolvidas as placas de controle do cofre apresentados na Figura 80:

Figura 80: Placa do microcontrolador e placa fonte para o cofre



Fonte: Os autores

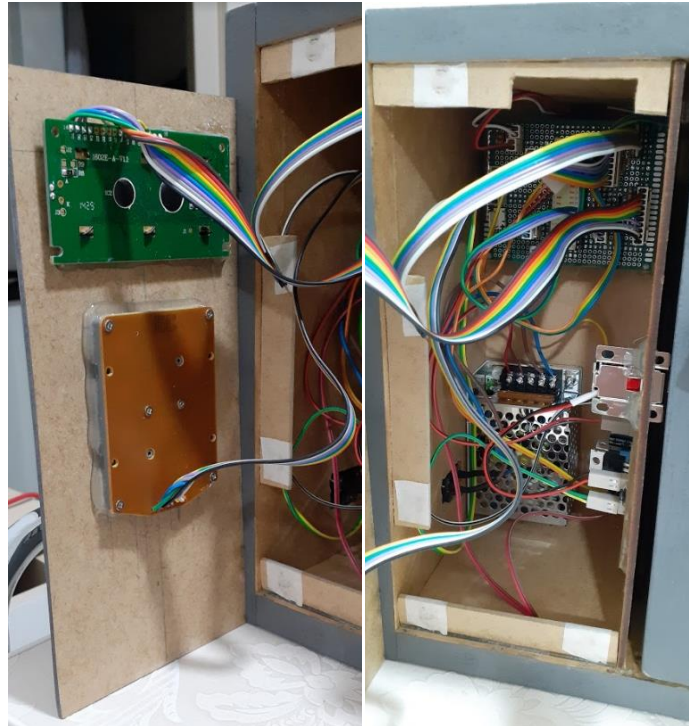
Na Figura 80, à esquerda, podemos ver a placa do microcontrolador com barras de pinos para facilitar a conexões dos demais periféricos. À direita é possível ver a placa fonte do projeto. Esta placa possui os circuitos reguladores e o circuito da trava eletrônica.

Para fazer a divisória, foi cortada uma fina placa para realizar a divisão entre os compartimentos. Esta divisória foi colocada de forma a poder posicionar a placa do microcontrolador na parede do fundo do cofre, na parte superior.

Cabe ressaltar que para essa montagem foi necessário remodelar a parte frontal do cofre apresentada na Figura 79, conforme necessidade de projeto. Na parte de circuitos, foi possível colocar a placa do microcontrolador, a placa fonte, a fonte de 12 V e a trava eletrônica, mais o *display* e o teclado na parte frontal do cofre.

Uma imagem da parte de circuitos montada é mostrada na Figura 81:

Figura 81: Parte de circuitos do cofre



Fonte: Os autores

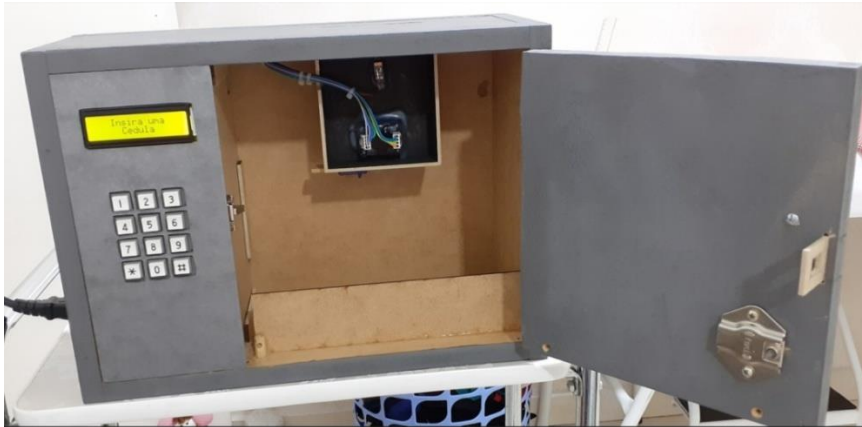
Na Figura 81, observa-se à esquerda a vista traseira do *display* e do teclado matricial fixados à placa frontal. À direita é possível ver a placa do microcontrolador, na parte superior do fundo do cofre e a fonte de 12 V logo abaixo. Para a ligação do cofre na energia, foi feita uma conexão interna da fonte 12 V com uma entrada externa para conectar o cabo de energia. A placa fonte está na parte lateral da divisória, assim como a trava eletrônica.

No lado direito do cofre ficou a parte de armazenagem do dinheiro. Neste espaço, foi feito um corte na parte superior do cofre para inserção das notas. Logo abaixo, foi fixada a caixa leitora para identificação das cédulas. Foi colocada também uma fina placa de madeira MDF para evitar que as cédulas saíssem do cofre quando este fosse aberto.

Com relação à porta, foi feito um encaixe para a língua da trava eletrônica, para permitir fazer uma boa interface mecânica tanto para abrir quanto para fechar o cofre.

Uma imagem da parte de leitura e armazenagem do cofre é apresentada na Figura 82:

Figura 82: Parte de armazenagem do cofre



Fonte: Os autores

Na imagem da Figura 82, é mostrada uma fechadura mecânica na porta. Esta fechadura foi retirada posteriormente na montagem final, pois não tem nenhuma utilidade para o projeto.

Uma imagem geral do cofre, fechado e ligado, é apresentada na Figura 83:

Figura 83: Imagem geral do cofre



Fonte: Os autores

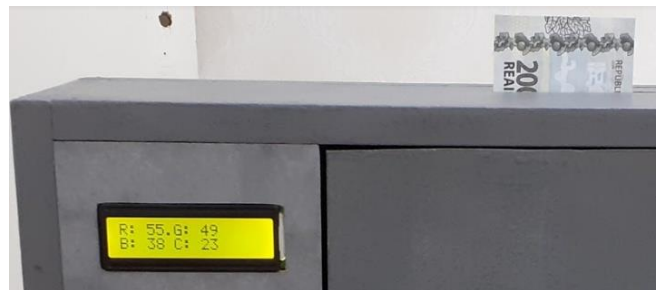
O cofre da Figura 83 foi utilizado para a etapa de testes do capítulo 6. O modelo finalizado do projeto teve ainda a inclusão de uma entrada USB para comunicação serial, entre outros detalhes, que serão mostrados na seção 6.3.

A seguir, será mostrada a calibração da última cédula adquirida, a de 200 reais, feito durante o desenvolvimento deste protótipo finalizado.

5.2. Calibração da cédula de 200 reais

O teste de calibração da cédula de 200 reais seguiu o mesmo procedimento da calibração das demais cédulas, feita na seção 4.1.3. A imagem da calibração desta cédula pode ser vista na Figura 84:

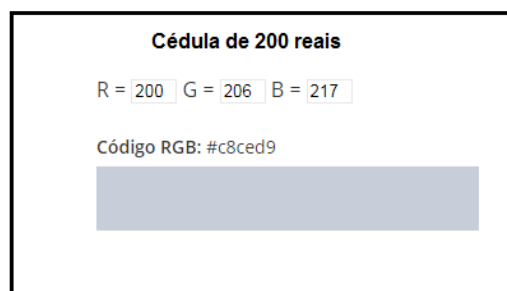
Figura 84: Teste de leitura RGB (cédula de 200 reais).



Fonte: Os autores

Corrigindo os valores RGB da Figura 82, da mesma forma feita na seção 4.2.1, temos R=200, G=206 e B=217. O resultado da verificação da leitura RGB é vista na Figura 85:

Figura 85: Verificação da leitura RGB (cédula de 200 reais).



Fonte: Os autores

Como é possível observar no resultado obtido, o padrão RGB informado retornou uma coloração cinza. Esta coloração ocorre quando os valores de R, G e B são altos (próximos do branco, RGB=255) e são próximos um do outro.

A cédula de 200 reais também foi calibrada para identificação no cofre microcontrolado. O processo de calibração utilizou somente dois exemplares de cédula, por ser uma cédula emitida recentemente e de difícil aquisição.

O processo para identificação da cédula de 200 reais é o desenvolvido na seção 4.2.2, cujo resultado pode ser visto na Figura 86:

Figura 86: Calibração da cédula de 200 reais



Fonte: Os autores

Na Figura 86, observa-se que foi feita identificação do valor da cédula de 200 reais. Com isso, todas as cédulas da Segunda Família do Real foram calibradas, podendo então ser realizados testes de confiabilidade de leitura de cédulas. Estes testes são mostrados na seção 6.1.

Com este protótipo de cofre desenvolvido, foram feitos testes para verificar a sua operação. Esta operação envolveu o acesso e teste dos menus, para inserir cédulas, consultar saldo, trocar a senha e fazer a abertura do cofre.

Na seção seguinte, será apresentado em detalhes como operar cada uma das opções do protótipo do cofre finalizado.

5.3. Teste de operação do cofre

Nesta seção, serão apresentadas várias imagens mostrando a operação do cofre para acessar seus menus. Nesses menus, será visto como inserir cédulas no cofre, alterar senhas e abrir a porta do cofre.

As operações do cofre são feitas basicamente por meio de um teclado matricial e um *display*. No *display*, serão apresentadas informações sobre o cofre e no teclado é possível selecionar algumas opções.

Nesta etapa, serão apresentadas imagens do cofre em funcionamento. Assim que o cofre é ligado, a primeira tela mostrada é a de apresentação do cofre, como representado na Figura 87:

Figura 87: Tela de apresentação do cofre



Fonte: Os autores

Esta tela de apresentação é opcional, podendo ser retirada do cofre alterando o código-fonte apresentado no Anexo A deste trabalho. Outras informações poderão ser incluídas em seu lugar, devendo o usuário ter algum conhecimento básico em programação em linguagem C para configurar corretamente a apresentação das informações no *display*.

Após a tela de apresentação, serão apresentadas as opções de acessar o menu ou inserir cédulas. Estas duas informações alternam na tela do display e para acessá-las é necessário digitar o caractere # na respectiva tela, conforme Figura 88:

Figura 88: Tela de seleção do cofre



Fonte: Os autores

Caso seja escolhida a opção de acessar o menu, as opções mostradas na Figura 89 estarão disponíveis para o usuário:

Figura 89: Tela de menu do cofre



Fonte: Os autores

No menu do cofre é possível selecionar quatro opções. Estas opções serão mostradas a seguir:

- 1 – Senha: Nela é possível configurar a senha. Para isto é necessário digitar a senha antiga, para logo em seguida digitar a nova senha, conforme Figura 90:

Figura 90: Tela de configuração de senha



Fonte: Os autores

- 2 – Saldo: Mostra o valor acumulado pelo cofre, conforme Figura 91:

Figura 91: Tela de saldo do cofre



Fonte: Os autores

- 3 – Abrir Cofre: Nesta opção, é possível abrir o cofre digitando a senha previamente programada. Esta senha é uma sequência de 4 caracteres. Caso a senha digitada esteja correta, a porta é liberada e o display apresentará uma mensagem “Abrir a porta”.

Imagens demonstrando o procedimento para abertura da porta do cofre são mostradas na Figura 92:

Figura 92: Tela de configuração de senha



Fonte: Os autores

- 4 – Sair: Nesta opção, o usuário sai do menu do cofre, podendo então selecionar a opção para inserir cédulas no cofre.

Quando o usuário retorna a tela de seleção do cofre, é possível inserir as cédulas no cofre. Nesta tela de seleção do cofre, deve-se digitar # quando aparecer a informação “Inserir Cédula” para acessar o programa para identificação de cédulas.

A identificação das cédulas segue a rotina mostrada nas Figuras 93 e 94:

Figura 93: Cédula inserida no cofre



Fonte: Os autores

Figura 94: Identificação da cédula no cofre



Fonte: Os autores

Para retirada do dinheiro, deve-se seguir o procedimento explicado em “Abrir Cofre”. Uma imagem desta retirada pode ser vista na Figura 95:

Figura 95: Retirada do dinheiro do cofre



Fonte: Os autores

Na Figura 95, à esquerda, é possível ver o conteúdo do cofre, ou seja, as cédulas que foram lidas pelo sensor RGB e foram liberados para o cofre pelo sistema servo motor. Nesta imagem, verifica-se que é formado um acúmulo de cédulas na porta da caixa leitora. Este problema pode ser alvo de melhoria em trabalhos futuros. Na imagem à direita, é mostrado que após a abertura do cofre, o saldo é zerado.

No capítulo a seguir serão apresentados os resultados do trabalho.

6. RESULTADOS E DISCUSSÕES

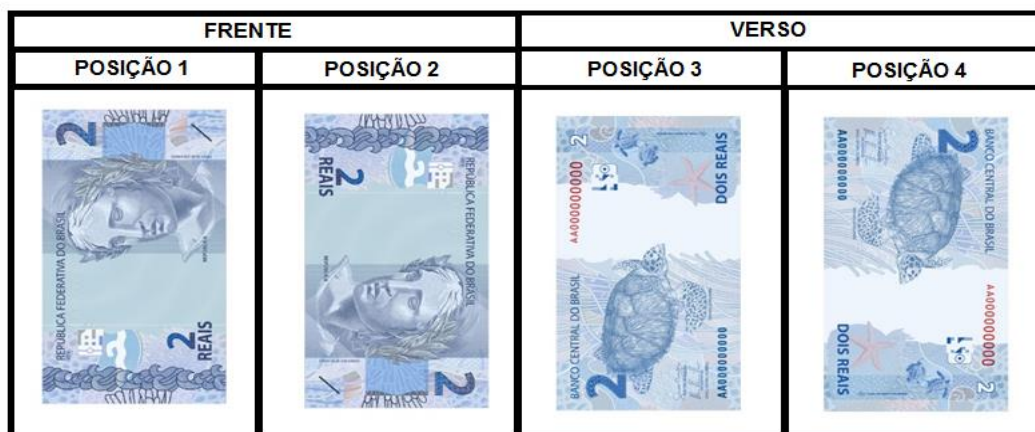
Neste capítulo, serão apresentados os testes de confiabilidade da leitura das cédulas e do sistema de segurança com senha. Além disso, será mostrado o teste de comunicação serial com computador, a estimativa de custo final do projeto e algumas considerações finais sobre o projeto.

6.1. Teste de confiabilidade de leitura de cédulas

Com a finalização do cofre microcontrolado, foram realizados testes para verificar a exatidão na leitura das cédulas, contabilizando os acertos e erros. Para este teste, foram utilizados três conjuntos completos de cédulas, ou seja, três exemplares de cédulas de cada valor. No caso da cédula de 200 reais, foi necessário repetir uma das cédulas, pois só havia dois exemplares para teste.

Para este teste, foram consideradas quatro posições distintas da cédula, representadas na Figura 96.

Figura 96: Posições das cédulas para teste de leitura



Fonte: Os autores

Os resultados destes testes podem ser vistos nas Tabelas 8 a 14:

Tabela 8: Testes de leitura de cédulas – 2 reais

| | POSIÇÃO 1 | | POSIÇÃO 2 | | POSIÇÃO 3 | | POSIÇÃO 4 | |
|-----------|------------------------|------|-----------|------|-----------|------|-----------|------|
| | OK | ERRO | OK | ERRO | OK | ERRO | OK | ERRO |
| 2 REAIS | | | | | | | | |
| 1º GRUPO | x | | x | | x | | x | |
| 2º GRUPO | x | | x | | x | | | x |
| 3º GRUPO | x | | x | | | x | x | |
| RESULTADO | TAXA DE ACERTO = 83,3% | | | | | | | |

Fonte: Os autores

No teste com cédula de 2 reais houve 10 acertos e 2 erros, taxa de acerto de 83,3%. Os erros ocorreram na posição 3 e 4.

Tabela 9: Testes de leitura de cédulas – 5 reais

| | POSIÇÃO 1 | | POSIÇÃO 2 | | POSIÇÃO 3 | | POSIÇÃO 4 | |
|-----------|------------------------|------|-----------|------|-----------|------|-----------|------|
| | OK | ERRO | OK | ERRO | OK | ERRO | OK | ERRO |
| 5 REAIS | | | | | | | | |
| 1º GRUPO | x | | x | | | x | x | |
| 2º GRUPO | x | | x | | x | | x | |
| 3º GRUPO | x | | x | | x | | x | |
| RESULTADO | TAXA DE ACERTO = 91,7% | | | | | | | |

Fonte: Os autores

No teste com cédula de 5 reais houve 11 acertos e 1 erro, taxa de acerto de 91,7%. O erro ocorreu na posição 3.

Tabela 10: Testes de leitura de cédulas – 10 reais

| | POSIÇÃO 1 | | POSIÇÃO 2 | | POSIÇÃO 3 | | POSIÇÃO 4 | |
|-----------|-----------------------|------|-----------|------|-----------|------|-----------|------|
| | OK | ERRO | OK | ERRO | OK | ERRO | OK | ERRO |
| 10 REAIS | | | | | | | | |
| 1º GRUPO | x | | x | | x | | x | |
| 2º GRUPO | x | | x | | x | | x | |
| 3º GRUPO | x | | x | | x | | x | |
| RESULTADO | TAXA DE ACERTO = 100% | | | | | | | |

Fonte: Os autores

No teste com cédula de 10 reais houve 12 acertos e 0 erro, taxa de acerto de 100%. Não ocorreu nenhum erro de leitura.

Tabela 11: Testes de leitura de cédulas – 20 reais

| | POSIÇÃO 1 | | POSIÇÃO 2 | | POSIÇÃO 3 | | POSIÇÃO 4 | |
|-----------|------------------------|------|-----------|------|-----------|------|-----------|------|
| | OK | ERRO | OK | ERRO | OK | ERRO | OK | ERRO |
| 20 REAIS | | | | | | | | |
| 1º GRUPO | x | | | x | x | | x | |
| 2º GRUPO | x | | x | | x | | x | |
| 3º GRUPO | x | | x | | x | | x | |
| RESULTADO | TAXA DE ACERTO = 91,7% | | | | | | | |

Fonte: Os autores

No teste com cédula de 20 reais houve 11 acertos e 1 erro, taxa de acerto de 91,7%. O erro ocorreu na posição 2.

Tabela 12: Testes de leitura de cédulas – 50 reais

| | POSIÇÃO 1 | | POSIÇÃO 2 | | POSIÇÃO 3 | | POSIÇÃO 4 | |
|-----------|------------------------|------|-----------|------|-----------|------|-----------|------|
| | OK | ERRO | OK | ERRO | OK | ERRO | OK | ERRO |
| 50 REAIS | | | | | | | | |
| 1º GRUPO | | x | x | | x | | x | |
| 2º GRUPO | | x | x | | x | | x | |
| 3º GRUPO | x | | x | | x | | x | |
| RESULTADO | TAXA DE ACERTO = 83,3% | | | | | | | |

Fonte: Os autores

No teste com cédula de 50 reais houve 10 acertos e 2 erros, taxa de acerto de 83,3%. Os dois erros ocorreram na posição 1.

Tabela 13: Testes de leitura de cédulas – 100 reais

| | POSIÇÃO 1 | | POSIÇÃO 2 | | POSIÇÃO 3 | | POSIÇÃO 4 | |
|-----------|------------------------|------|-----------|------|-----------|------|-----------|------|
| | OK | ERRO | OK | ERRO | OK | ERRO | OK | ERRO |
| 100 REAIS | | | | | | | | |
| 1º GRUPO | x | | x | | x | | x | |
| 2º GRUPO | | x | x | | x | | x | |
| 3º GRUPO | | x | x | | x | | x | |
| RESULTADO | TAXA DE ACERTO = 83,3% | | | | | | | |

Fonte: Os autores

No teste com cédula de 100 reais houve 10 acertos e 2 erros, taxa de acerto de 83,3%. Os dois erros ocorreram na posição 1.

Tabela 14: Testes de leitura de cédulas – 200 reais

| 200 REAIS | POSIÇÃO 1 | | POSIÇÃO 2 | | POSIÇÃO 3 | | POSIÇÃO 4 | |
|-----------|-----------------------|------|-----------|------|-----------|------|-----------|------|
| | OK | ERRO | OK | ERRO | OK | ERRO | OK | ERRO |
| 1º GRUPO | x | | x | | x | | x | |
| 2º GRUPO | x | | x | | x | | x | |
| 3º GRUPO | x | | x | | x | | x | |
| RESULTADO | TAXA DE ACERTO = 100% | | | | | | | |

Fonte: Os autores

No teste com cédula de 200 reais houve 12 acertos e 0 erro, taxa de acerto de 100%. Não ocorreu nenhum erro de leitura.

Das Tabelas 8 a 14, é importante ressaltar que em todos os casos em pelo menos três das quatro posições a leitura é feita com sucesso. Isto indica que é possível realizar a leitura do valor da cédula, porém, há uma probabilidade, mesmo que baixa, deste valor ser contabilizado errado.

Logo, foi feita uma análise para recomendar as posições das cédulas onde ocorrem as maiores chances de acerto. Estas informações são apresentadas na Tabela 15:

Tabela 15: Posições recomendadas para leitura de cédulas

| CÉDULAS | ACERTOS | ERROS | POSIÇÃO RECOMENDADA |
|-----------|---------|-------|---------------------|
| 2 REAIS | 10 | 2 | 1 e 2 |
| 5 REAIS | 11 | 1 | 1, 2 e 4 |
| 10 REAIS | 12 | 0 | Qualquer posição |
| 20 REAIS | 11 | 1 | 1, 3 e 4 |
| 50 REAIS | 10 | 2 | 2, 3 e 4 |
| 100 REAIS | 10 | 2 | 2, 3 e 4 |
| 200 REAIS | 12 | 0 | Qualquer posição |
| TOTAL | 76 | 8 | |

Fonte: Os autores

Na Tabela 15, são recomendadas as posições onde não ocorreram erros de leitura nos testes. Para análise dos erros de leitura, as posições em que ocorreu erro são apresentadas de forma ilustrativa na Figura 97:

Figura 97: Erros das cédulas no teste de leitura



Fonte: Os autores

Para justificar o erro nestas leituras, foram elaboradas algumas suposições. Cabe ressaltar que a leitura das cédulas é feita na borda inferior da cédula, onde será feita esta análise.

No caso das cédulas de 2 e 5 reais, os erros podem ter ocorrido porque a leitura é feita num local onde o tom de cor das cédulas é mais fraco e o sensor pode confundir os dois valores. Já o caso da cédula de 20 reais pode ter sido causado pelo quadrado vermelho/alaranjado, que é semelhante ao da cédula de 10 reais. No caso das cédulas de 50 e 100 reais, os erros podem ter ocorrido devido à presença da faixa holográfica, comuns a estas duas cédulas.

Em suma, este teste serviu para verificar possíveis fontes de erro na leitura RGB das cédulas, que podem ser aprimoradas em projeto futuros.

6.2. Testes de confiabilidade do sistema de senhas

Este teste foi realizado para verificar a eficiência e segurança do sistema de senhas projetado para este cofre. Neste teste, foi programada uma senha e verificado se digitando esta senha escolhida a porta do cofre seria aberta, conforme esperado. Este teste está representado na Tabela 16:

Tabela 16: Testes de senhas

| Teste de senhas | | | |
|-----------------|-------|-------------|-----|
| N | Senha | Abre porta? | |
| | | Sim | Não |
| 1 | 1234 | X | |
| 2 | 2580 | X | |
| 3 | 3570 | X | |
| 4 | 1937 | X | |
| 5 | 2493 | X | |
| 6 | 4810 | X | |
| 7 | 7825 | X | |
| 8 | 2510 | X | |
| 9 | 5500 | X | |
| 10 | 3478 | X | |

Fonte: Os autores

No teste da Tabela 16, a senha correta permite a abertura da porta. Se for digitada a senha errada, uma informação é mostrada no display, e o usuário deve digitar a senha novamente. Neste cofre, é importante ressaltar que não foi feito nenhum sistema para bloquear o cofre após algumas tentativas, o que pode ser melhorado em projeto futuros.

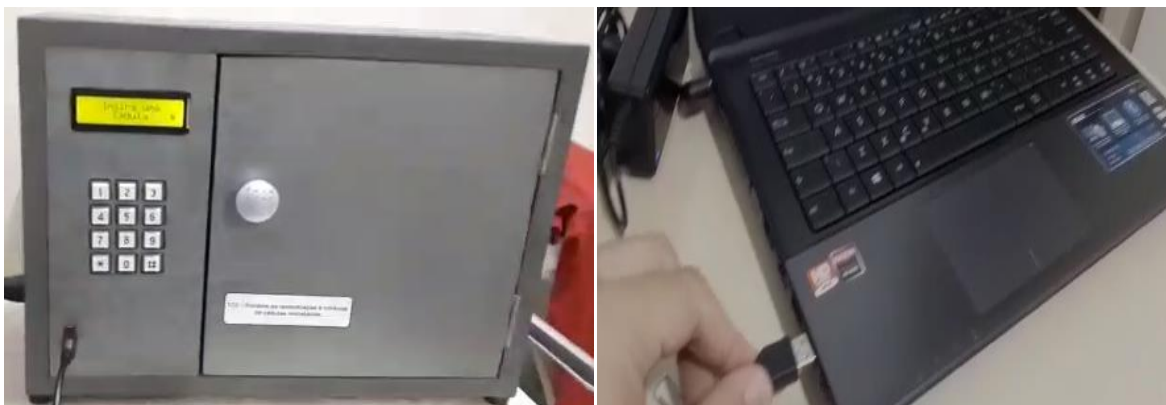
Comentando os resultados da Tabela 16, todos os dez testes de senhas tiveram sucesso. Desta forma é possível dizer que o cofre possui um bom sistema de senhas, abrindo o cofre somente com a digitação da senha correta e possibilitando a alteração da senha conforme necessidade do usuário.

6.3. Teste da comunicação do cofre com computador

Este teste de comunicação serial foi feito no protótipo final do cofre microcontrolado. Nesta parte, foi feito um teste para verificar a comunicação do cofre desenvolvido com um computador. Para fazer esta comunicação, é necessário utilizar um cabo USB para ligar fisicamente o cofre com uma porta USB do computador.

Uma imagem mostrando a conexão física necessária para comunicação é mostrada na Figura 98:

Figura 98: Conexão do cofre com um computador



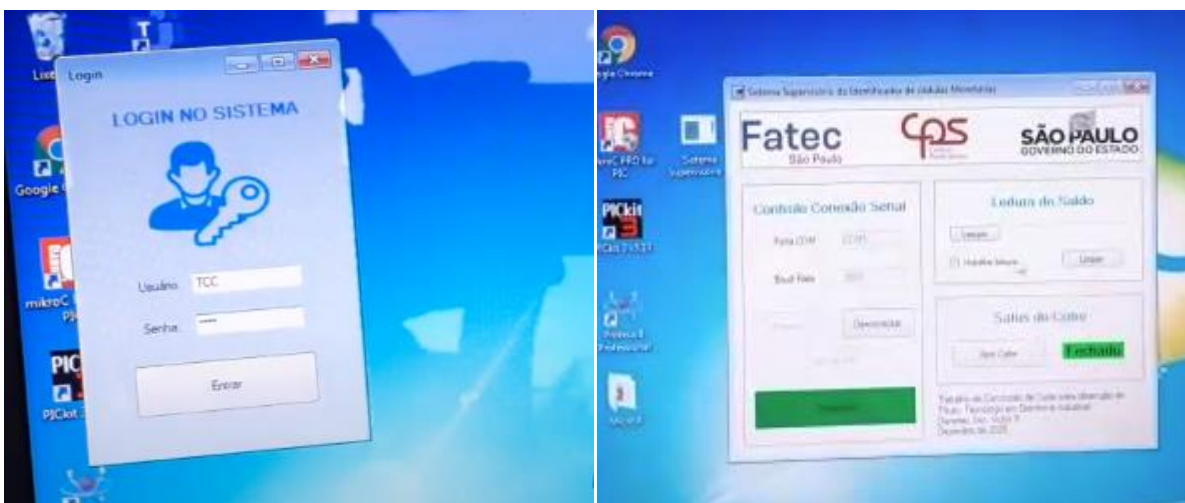
Fonte: Os autores

Na Figura 98, podemos ver à esquerda a entrada USB feita para o cofre. Esta entrada foi feita com uma emenda fêmea, sendo uma das pontas para ligação do cabo USB e a outra ponta está internamente no cofre, ligado ao conversor USB serial. À direita, podemos ver que a outra conexão é feita no computador

Após a ligação física com o cofre, é necessário acessar o executável da interface gráfica. Para isso, é necessário digitar o *login*: TCC e a senha previamente definida no projeto, 1234, para então ter acesso à tela principal do cofre.

Esta interface está representada na Figura 99:

Figura 99: Acesso ao cofre via computador



Fonte: Os autores

Nesta interface gráfica, que é a mesma desenvolvida na seção 4.2.6 deste trabalho, deve-se utilizar a opção “Controle Conexão Serial” para estabelecer a conexão com o cofre. Deve-se selecionar a porta identificada pelo microcontrolador e a taxa de transferência de dados (*baudrate*), conforme mostrado na Figura 100:

Figura 100: Tela principal do cofre



Fonte: Os autores

Na Figura 100, é possível observar que o microcontrolador conseguiu fazer a comunicação na porta COM4, e foi selecionado o *baudrate* (taxa de transferência de dados) de 9600 bps.

Quando a comunicação é realizada com sucesso, é informado que o cofre está “Conectado”. Neste momento, é possível então acessar as opções de leitura do saldo e controlar a porta do cofre.

Para verificar o saldo, deve-se utilizar a opção “Leitura de Saldo”. Então, a caixa “Habilitar leitura” deve ser selecionada e clicar em leitura, para que o valor armazenado no cofre seja disponibilizado. Neste caso, o saldo do cofre é o valor armazenado na memória EEPROM do microcontrolador, que será enviado para a interface gráfico assim que for solicitado sua disponibilização.

Uma imagem mostrando o funcionamento da leitura do saldo, mostrando a tela do *display* do cofre e a tela da interface gráfica é apresentada na Figura 101:

Figura 101: Verificação do saldo do cofre na interface gráfica e no cofre



Fonte: Os autores

Como é possível observar, o valor mostrado fisicamente no cofre (*display*) é o mesmo informado na interface gráfica, pois a comunicação está sendo feita corretamente e o saldo atualizado em tempo real. Logo, esta parte de verificação de saldo também está correta.

A terceira opção presente nesta interface gráfica é o controle da abertura da porta do cofre, na opção “Status do Cofre”. Quando é selecionado na interface “Abrir Cofre”, é enviado um comando para abrir fisicamente a porta do cofre, como mostrado na Figura 102:

Figura 102: Teste de abertura de porta na interface gráfica



Fonte: Os autores

Na Figura 102, é possível ver a abertura da porta do cofre utilizando a interface gráfica. Este foi o último teste do projeto, restando apenas comentar sobre a estimativa de custo do projeto, realizada a seguir.

6.4. Estimativa de custo do projeto do cofre

Uma estimativa de custo foi feita para verificar todos os gastos ao longo do projeto. Para efeito de análise, esta estimativa será dividida em três categorias: componentes principais, da caixa do cofre e secundários.

Um dos objetivos da construção deste cofre era obter um protótipo de baixo custo. Para isto, realiza-se esta análise para discutir se é viável construir um protótipo caseiro de cofre eletrônico. Estes gastos serão mostrados em forma de tabela, com quantidades e valores de cada peça que foi utilizada. A primeira tabela de custos é a dos componentes principais, apresentado na Tabela 17:

Tabela 17: Lista de materiais dos principais componentes do projeto

| LISTA DE MATERIAIS: COMPONENTES PRINCIPAIS CIRCUITOS DO COFRE | | | |
|--|----------------|-----------------------|-------------------|
| TIPO | Unidade | Valor Unitário | CUSTO |
| Conversor USB/Serial CH340 | 1 | R\$ 14,79 | R\$ 14,79 |
| Display LCD 16x2 | 1 | R\$ 21,00 | R\$ 21,00 |
| Fototransistor 5 mm TIL78 | 1 | R\$ 0,76 | R\$ 0,76 |
| Gravador PICKit 3 | 1 | R\$ 87,67 | R\$ 87,67 |
| Led Infravermelho 5 mm TIL32 | 1 | R\$ 0,54 | R\$ 0,54 |
| PIC 16F877A | 1 | R\$ 30,31 | R\$ 30,31 |
| Placa de prototipagem de fibra de vidro | 1 | R\$ 14,16 | R\$ 14,16 |
| Regulador de Tensão 7805 | 2 | R\$ 1,42 | R\$ 2,84 |
| Sensor RGB TCS230 | 1 | R\$ 30,00 | R\$ 30,00 |
| Servo motor SG-90 | 1 | R\$ 26,90 | R\$ 26,90 |
| Trava Solenoide | 1 | R\$ 39,99 | R\$ 39,99 |
| Teclado Matricial 4x3 | 1 | R\$ 8,34 | R\$ 8,34 |
| Fonte Chaveada 40W X 12V Bivolt 3A | 1 | R\$ 28,99 | R\$ 28,99 |
| CUSTO TOTAL | | | R\$ 306,29 |

Fonte: Os autores

Como é possível observar, o valor gasto com os componentes principais é alto, sendo a maior fonte de gastos deste projeto. Somente o gravador Pickit3 custou R\$ 87,67, sendo o item que mais custou para o projeto.

A segunda tabela trata dos gastos feitos com a construção da caixa do cofre. Boa parte dos materiais utilizados nesta construção foram os pedaços de madeira MDF e componentes para fixação. A lista de materiais completa e os custos com a caixa são apresentados na Tabela 18:

Tabela 18: Lista de materiais para montagem do cofre

| LISTA DE MATERIAIS: CAIXA DO COFRE | | | |
|---|----------------|-----------------------|------------------|
| Descrição | Unidade | Valor Unitário | CUSTO |
| MDF 300mm x 150mm x 15mm | 2 | R\$ 3,14 | R\$ 6,28 |
| MDF 250mm x 150mm x 15mm | 2 | R\$ 2,62 | R\$ 5,24 |
| MDF 250mm x 288mm x 15mm | 1 | R\$ 5,00 | R\$ 5,00 |
| MDF 250mm x 100mm x 15mm | 1 | R\$ 1,74 | R\$ 1,74 |
| MDF 246mm x 184mm x 15mm | 1 | R\$ 3,16 | R\$ 3,16 |
| Parafuso 5mm x 50mm | 16 | R\$ 0,15 | R\$ 2,40 |
| Dobradiça | 2 | R\$ 7,44 | R\$ 14,88 |
| Lata de tinta Cinza para madeira | 1 | R\$ 34,00 | R\$ 34,00 |
| CUSTO TOTAL | | | R\$ 72,70 |

Fonte: Os autores

Da Tabela 9, podemos destacar o gasto com a lata de tinta para pintura da parte externa do cofre. Cabe ressaltar que este é um gasto opcional, voltado à parte estética, podendo ser retirado para reduzir o gasto total com a montagem do cofre.

A terceira tabela lista os demais componentes utilizados no projeto. Estes componentes são utilizados para conexão como barra de pinos, conectores, soquete e terminal, que permitem fazer um *layout* de cofre mais enxuto e organizado, facilitando a montagem e a disposição dos componentes principais no interior do cofre.

Os outros componentes, resistores, capacitores, diodos etc. são componentes de circuito, sendo que muitos dos valores de resistência e capacitância utilizados são padrões da Eletrônica, para facilitar a montagem dos circuitos do projeto.

A lista completa com os componentes secundários utilizados e seus custos estão apresentados na Tabela 19:

Tabela 19: Lista de materiais do cofre (componentes secundários)

| LISTA DE MATERIAIS: COMPONENTES SECUNDÁRIOS | | | |
|--|----------------|-----------------------|------------------|
| TIPO | Unidade | Valor Unitário | CUSTO |
| Alojamento 10 vias KK | 1 | R\$ 0,33 | R\$ 0,33 |
| Alojamento 2 vias KK | 4 | R\$ 0,07 | R\$ 0,28 |
| Alojamento 4 vias KK | 6 | R\$ 0,13 | R\$ 0,78 |
| Alojamento 8 vias KK | 1 | R\$ 0,19 | R\$ 0,19 |
| Barra de pinos 40 vias | 1 | R\$ 1,14 | R\$ 1,14 |
| Capacitor Eletrolítico 10 uF 50 V | 2 | R\$ 0,14 | R\$ 0,28 |
| Capacitor Poliéster 22 nF 50 V | 2 | R\$ 0,06 | R\$ 0,12 |
| Capacitor Poliéster 100 nF 50 V | 2 | R\$ 0,09 | R\$ 0,18 |
| Capacitor Poliéster 330 nF 50 V | 2 | R\$ 0,67 | R\$ 1,34 |
| Capacitor Cerâmico de 22 pF | 2 | R\$ 0,11 | R\$ 0,22 |
| Conector 10 vias KK | 1 | R\$ 0,81 | R\$ 0,81 |
| Conector 2 vias KK | 4 | R\$ 0,16 | R\$ 0,64 |
| Conector 4 vias KK | 6 | R\$ 0,29 | R\$ 1,74 |
| Conector 8 vias KK | 1 | R\$ 0,66 | R\$ 0,66 |
| Conector fonte | 2 | R\$ 1,99 | R\$ 3,98 |
| Cristal de quartzo de 8 MHZ | 1 | R\$ 0,79 | R\$ 0,79 |
| Diodo 1N4007 | 1 | R\$ 0,10 | R\$ 0,10 |
| Emenda USB fêmea | 1 | R\$ 5,00 | R\$ 5,00 |
| Jumper 40 peças | 1 | R\$ 8,44 | R\$ 8,44 |
| Led Vermelho 3 mm | 1 | R\$ 0,19 | R\$ 0,19 |
| PC817 "optoacoplador" | 1 | R\$ 0,47 | R\$ 0,47 |
| Potenciômetro 10 K | 2 | R\$ 1,33 | R\$ 2,66 |
| Resistor de 1 kΩ 1/4 W | 2 | R\$ 0,04 | R\$ 0,08 |
| Resistor de 10 kΩ 1/4 W | 5 | R\$ 0,04 | R\$ 0,20 |
| Resistor de 330 kΩ 1/4 W | 1 | R\$ 0,04 | R\$ 0,04 |
| Resistor de 680 kΩ 1/4 W | 1 | R\$ 0,04 | R\$ 0,04 |
| Soquete DIP 40 Pinos Torneado | 1 | R\$ 4,00 | R\$ 4,00 |
| Terminal KK | 100 | R\$ 0,08 | R\$ 8,00 |
| Transistor IRF840 MOSFET ch-n | 1 | R\$ 3,99 | R\$ 3,99 |
| CUSTO TOTAL | | | R\$ 46,69 |

Fonte: Os autores

Somando todos os componentes e materiais utilizados no projeto, o total gasto foi de aproximadamente R\$425,68. Este valor está dentro da média dos cofres encontrados no mercado. O grande diferencial deste protótipo é a identificação dos valores das cédulas. Boa parte dos cofres preza pelo sistema de segurança, o que acabam encarecendo muitos dos modelos encontrados no mercado.

O valor total deste cofre pode ainda ser reduzido em aproximadamente R\$100,00 na confecção de uma nova unidade, se houver a economia na compra do gravador Pickit3 e da lata de tinta para pintura. Neste caso, este cofre ficaria com um preço ainda mais acessível, de aproximadamente R\$325,00.

6.5. Comentários gerais e considerações finais sobre o projeto

Este trabalho foi desenvolvido em dois períodos distintos. A primeira etapa foi desenvolvida no curso de Microcontroladores 2, dedicada somente a identificação de cédulas monetárias. A segunda etapa foi dedicada ao projeto de conclusão de curso (TCC) e envolveu toda a parte de desenvolvimento final do cofre.

A primeira etapa foi feita no período de agosto de 2019 a novembro de 2019. Foram utilizadas o período das aulas de Microcontroladores 2, os laboratórios da FATEC-SP e eventualmente desenvolvimento em ambiente residencial para esta etapa.

A segunda etapa foi feita no período de agosto de 2020 a dezembro de 2020. Esta etapa, devido à pandemia ocorrida no ano de 2020, precisou ser desenvolvida separadamente pelos integrantes do grupo. Um integrante ficou responsável pelo desenvolvimento da parte física e programação dos circuitos do cofre (Denimar), outro para montagem do *layout* do cofre e testes de comunicação serial (Victor) e outro dedicado exclusivamente à escrita da monografia (Leo).

Esta etapa foi dividida em três períodos distintos. No primeiro período (ago/20 a set/20), foi dedicado um dia por semana (2 horas semanais) para o projeto. No segundo período (out/20 a nov/20), foram dedicadas 2 dias por semana (4 horas semanais) para o projeto. Já em dezembro, na véspera da apresentação deste projeto (18/12/20), foram dedicadas diariamente 2 horas para finalizar o cofre.

As estimativas de tempo gasto em cada etapa e do tempo total gasto para este projeto estão representadas na Tabela 20:

Tabela 20: Estimativa de tempo gasto com o projeto

| ETAPA 1 | Horas por semana | Dias utilizados | TOTAL (horas) |
|--|------------------|-----------------|---------------|
| Tempo gasto em aula | 2 | 8 | 16 |
| Tempo gasto fora de aula | 1 | 16 | 16 |
| TOTAL DE HORAS GASTAS (ETAPA 1) | | | 32 |
| ETAPA 2 | Horas por semana | Dias utilizados | TOTAL (horas) |
| Tempo gasto (ago/20 a set/20) | 2 | 8 | 16 |
| Tempo gasto (out/20 a nov/20) | 4 | 8 | 32 |
| Tempo gasto (dez/20) | 2 | 18 | 36 |
| TOTAL DE HORAS GASTAS (ETAPA 2) | | | 84 |
| TOTAL DE HORAS GASTAS COM O PROJETO | | | 116 |

Fonte: Os autores

Como é possível ver na Tabela 20, foram gastas aproximadamente 116 horas para finalizar o projeto. No entanto esta estimativa leva em consideração que a última etapa foi desenvolvida em paralelo pelos três integrantes do grupo. Se uma pessoa decidir escolher fazer todo o projeto individualmente, o tempo total gasto será muito maior.

Sendo assim, fazendo algumas considerações finais sobre o projeto, podemos dizer que são exigidos conhecimentos técnicos de muitas áreas da Eletrônica (Microcontroladores, Circuitos Elétricos, Sensores etc.) para o desenvolvimento dos circuitos do cofre e alguns conhecimentos mecânicos básicos para montagem da interface mecânica do projeto (como o *layout* do cofre).

Além disso, com relação à parte operacional, é necessário muito tempo de dedicação para o desenvolvimento total do projeto. Recomenda-se organização em grupo e divisão das tarefas para obtenção de um bom resultado.

7. CONCLUSÕES

Este trabalho, além de ter um forte aspecto tecnológico, envolvendo conhecimentos de várias áreas da Eletrônica, possui utilização prática no dia a dia. Ainda que outras tecnologias forneçam melhores ferramentas para administrar as finanças, um sistema físico como um cofre é uma solução simples para estimular o hábito de poupar, como armazenar quantias que muitas vezes são desprezadas por boa parte da população, como cédulas de menor valor.

Neste trabalho, a identificação do valor das cédulas monetárias com o sensor RGB TCS 230 teve uma taxa de acerto de 90,5%. Os erros nas leituras das cédulas podem ser críticos para se fazer um planejamento financeiro a curto prazo, porém a longo prazo pode ser usado como uma importante forma de educação financeira.

Como o cofre desenvolvido é simples, mas intuitivo, pode ser utilizado até mesmo para ensinamento de crianças nas escolas, como parte de aulas lúdicas sobre educação financeira. Sobre este tema, podem ser introduzidos assuntos como as novas cédulas do Real, a importância de se conservar as cédulas, os critérios de segurança das notas e inclusive introduzir conceitos básicos de educação financeira. Estes são aspectos que podem ser reforçados também para adultos, que muitas vezes se encontram instáveis financeiramente, acumulando diversas dívidas, por não estarem habituados a lidar com o dinheiro de forma responsável.

Outro aspecto importante abordado neste trabalho é a identificação de objetos pela cor. O sensor TCS 230, por ser um equipamento pequeno e barato, pode ser utilizado em uma boa variedade de projetos, como para deficientes visuais.

No caso dos indivíduos que não possuem qualquer capacidade de visão, poderiam ser usados sistemas sonoros para indicar as cores identificadas pelo sensor. Para indivíduos com capacidade parcial de visão, como os daltônicos, alguns problemas simples podem ser resolvidos, como fazer a identificação de objetos sem auxílio de outra pessoa, evitando constrangimentos.

Em resumo, pode-se dizer que a Eletrônica é uma vasta área que pode ser utilizada para solucionar inúmeros problemas do dia a dia. Mesmo um produto ainda em desenvolvimento encontra aplicações úteis para diferentes segmentos da população.

8. TRABALHOS FUTUROS

Durante a execução deste trabalho, alguns tópicos para melhoria ou sugestões foram discutidos para o cofre identificador de cédulas:

-Melhoria do processo da leitura das cédulas com sensor RGB ou outro sensor de cor, para identificação das cédulas não abordadas neste trabalho.

-Sistema para confirmação do valor da cédula antes deste valor ser contabilizado no saldo do cofre. Este sistema poderia ser feito com a programação de uma rotina para confirmação do próprio usuário no teclado matricial se o valor mostrado no *display* é o correto.

-Verificação da autenticidade de cédulas monetárias, utilizando luz ultravioleta, por exemplo.

-Sistema sonoro para indicar qual cédula está sendo inserida, no caso de deficientes visuais.

-Implantação de um sistema para também identificar valores de moedas. Este sistema pode utilizar sensor indutivo ou outro que trabalhe com elementos metálicos.

-Melhoria do sistema servo motor ou outro sistema para entrada de cédulas. Poderia ser desenvolvido um sistema de esteira para recebimento das cédulas, como o utilizado para puxar as cédulas nos caixas eletrônicos

-Sistema sonoro indicando que a porta está aberta, para evitar tentativa de furto ou que um usuário esqueça de fechar o cofre.

-Alimentação do cofre de forma autônoma, não precisando estar ligado à energia. Poderiam ser utilizadas baterias ou outros que forneçam energia ao cofre.

-Sistema para abertura do cofre em caso de emergência, como queda de energia.

-Aplicativo em celular para poder acessar o cofre de forma remota. Este aplicativo tornaria prático o controle do cofre, podendo ser acessado enquanto se estiver fora de casa.

-Sistema de gerenciamento de dados na interface gráfica, exibição de gráficos, criação de metas, histórico de uso entre outras funções.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BANCO CENTRAL DO BRASIL. O brasileiro e sua relação com o dinheiro – pesquisa 2018. Disponível em: https://www.bcb.gov.br/content/cédulasemoedas/pesquisabrasileirodinheiro/Apresentacao_brasileiro_relacao_dinheiro_2018.pdf. Acesso em 13/11/2020
- [2] LOCOMOTIVA PESQUISA & ESTRATÉGIA. Os brasileiros e o dinheiro em espécie. Disponível em: <https://static.poder360.com.br/2019/10/Os-brasileiros-e-o-dinheiro-em-especie-Imprensa-Outubro-de-2019.pdf>. Acesso em 13/11/2020.
- [3] SPC BRASIL. Apenas 16% dos brasileiros pouparam em fevereiro, o menor percentual em 15 meses, aponta indicador do SPC Brasil e CNDL. Disponível em <https://www.spcbrasil.org.br/pesquisas/indice/4518>. Acesso em 13/11/2020.
- [4] DE SOUZA, David José. Desbravando o PIC: ampliado e atualizado para PIC 16F628A. Saraiva Educação SA, 2005.
- [5] BAU DA ELETRÔNICA. Microcontrolador PIC16F877A. Disponível em <https://www.baudaeletronica.com.br/microcontrolador-pic16f877a.html>. Acesso em 15/10/2020.
- [6] DANTAS, Leandro Poloni. Eletrônica Digital: técnicas digitais e dispositivos lógicos programáveis. São Paulo: Senai-SP Editora, 2014. 557p.
- [7] SHAMIEH, Cathleen. Eletrônica para leigos / Cathleen Shamieh e Gordon McComb; tradutores Fernando Effiori e Roberto Assis Rezende. – 2ed. – Rio de Janeiro: Alta Books, 2011. 416p.
- [8] REVISTA BELAS ARTES. Cor Luz, Cor Pigmento e os sistemas RGB E CMY. Disponível em: <https://www.belasartes.br/revistabelasartes/downloads/artigos/3/cor-luz-cor-pigmento-e-os-sistemas-rgb-e-cmy.pdf>. Acesso em 18/11/2020.
- [9] UNIVERSIDADE FEDERAL DO CEARÁ: IMPRENSA UNIVERSITÁRIA – O que são padrões de cores RGB e CMYK? Disponível em: <https://imprensa.ufc.br/pt/duvidas-frequentes/padrao-de-cor-rgb-e-cmyk/> Acesso em 15/10/2020.

- [10] BANCO CENTRAL DO BRASIL. Cédulas Produzidas – Disponível em: <https://www.bcb.gov.br/cedulasemoedas/cedulasemitidas>. Acesso em 15/10/2020.
- [11] BANCO CENTRAL DO BRASIL. Segunda Família do Real: Por que uma Segunda Família? – Disponível em: <https://www.bcb.gov.br/novasnotas/segunda-familia.html>. Acesso em 15/10/2020.
- [12] CÉDULAS DO BRASIL. Catálogo das cédulas brasileiras: 2011-2020 Reais, 2ª família – Disponível em: http://www.moedasdobrasil.com.br/moedas/c_catalogo.asp?s=9. Acesso em 15/10/2020.
- [13] G1.GLOBO.COM. Nota de R\$200: deficientes visuais relatam dificuldades em diferenciar cédula: “Mesmo tamanho da de R\$20”. Disponível em: <https://g1.globo.com/df/distrito-federal/noticia/2020/09/27/nota-de-r-200-deficientes-visuais-relatam-dificuldades-em-diferenciar-cedula-mesmo-tamanho-da-de-r-20.ghtml>. Acesso em 06/12/2020.
- [14] BANCO CENTRAL DO BRASIL. Segunda Família do Real: cartilha de treinamento – Disponível em: <https://www.bcb.gov.br/novasnotas/assets/downloads/material-apoio/2e5/Cartilha.pdf>. Acesso em 15/10/2020.
- [15] PM COFRES. Conheça alguns tipos de cofres – Disponível em: <https://www.pmcofres.com.br/seguranca/conheca-alguns-tipos-de-cofres/>. Acesso em 15/11/2020
- [16] LOJA DA ELÉTRICA. Módulo sensor de cor RGB TCS230 – Disponível em: <https://daeletrica.com.br/modulo-sensor-de-cor-rgb-tcs230.html>. Acesso em 15/10/2020.
- [17] DATASHEET, T. A. O. S. Tcs230 Programmable Color Light-To-Frequency Converter. 2003. Acesso em 05/11/2020.
- [18] ARDUÍNO BY MYSELF. Sensoriamento com Infravermelho. Disponível em: <https://arduinobymyself.blogspot.com/2013/01/sensor-ir-projetos.html>. Acesso em 05/12/2020.
- [19] EMBARCADOS. Módulo de Display LCD – Disponível em: <https://www.embarcados.com.br/modulo-de-display-lcd/>. Acesso em 05/11/2020.

- [20] EMBARCADOS. Teclado Matricial e Varredura de Teclas – Disponível em: <https://www.embarcados.com.br/teclado-matricial-e-varredura-de-teclas/>. Acesso em 15/10/2020.
- [21] MUNDO DA ELÉTRICA. O que é servo motor e como funciona. Disponível: <https://www.mundodaeletrica.com.br/o-que-e-servo-motor-e-como-funciona/#:~:text=Para%20controlar%20a%20velocidade%20e,controlador%20e%20o%20utros%20circuitos%20complementares>. Acesso em 18/11/2020.
- [22] FILIPE E FLOP. Micro servo 9g SG90 TowerPro – Disponível em: <https://www.filipeflop.com/produto/micro-servo-9g-sg90-towerpro/>. Acesso em 05/11/2020
- [23] MERCADO LIVRE. Módulo Conversor Usb para TTL RS121 CH340 C/nf. Disponível em: https://produto.mercadolivre.com.br/MLB-1721058964-modulo-conversor-usb-para-ttl-rs232-ch340-cnf-_JM. Acesso em 20/11/2020.
- [24] MERCADO LIVRE. Mini Fechadura Solenoide Eletrônica Trava Trinco Automação. Disponível em: <https://produto.mercadolivre.com.br/MLB-1421878616-mini-fechadura-solenoide-eletrnica-trava-trinco-automaco>. Acesso em 20/11/2020.
- [25] BOYLESTAD, Robert L. Dispositivos eletrônicos e teoria de circuitos – 11.ed. – São Paulo: Pearson Education do Brasil, 2013. 766p.
- [26] DICAS DE PROGRAMAÇÃO: Entenda como funcionam os códigos RGB. Disponível em: <https://dicasdeprogramacao.com.br/entenda-como-funcionam-os-codigos-de-cores-rgb/>. Acesso em 24/11/2020
- [27] ATHOS ELECTRONICS. Acoplador óptico – O que é e como funciona. Disponível em: <https://athoselectronics.com/acoplador-optico/>. Acesso em 10/12/2020

ANEXO A – Código fonte do cofre microcontrolado

/*****

*NOME DO PROJETO: Identificador de cédulas com sensor RGB

* AUTORES:

Denimar Souza de Oliveira

Leo Masaharu Amaya Kitagawa

Victor Arthur Ravani

* HISTÓRICO DE REVISÕES:

Versão 1: 22/09/2019 às 15h40 min

Versão 2: 29/09/2019 às 15h30 min

Versão 3: 02/10/2019 às 13h00 min

Versão 4: 23/10/2019 às 21h00 min

Versão 5: 06/11/2019 às 21h00 min

Versão 6: 15/11/2019 às 10h00 min

Versão 7: 01/12/2019 às 18h00 min

Versão 8: 30/08/2020 às 18h00 min

Versão 9: 23/09/2020 às 21h00 min

Versão 10: 14/10/2020 às 21h00 min

Versão 11: 02/11/2020 às 21h00 min

Versão 12: 11/11/2020 às 22h30 min

Versão 13: 18/11/2020 às 22h00 min

Versão 14: 25/11/2020 às 22h40 min

Versão 15: 07/12/2020 às 22h00 min

Versão 16: 18/12/2020 às 13h00 min

* DESCRIÇÃO:

Programa para identificar o valor de cédulas monetárias com o sensor RGB.

O sensor lê as cores presentes nas cédulas, na forma de RGB (Red, Green, Blue).

De acordo com o valor de cada um destes parâmetros, imprime no display o valor correspondente em reais.

O programa é utilizado para controle de periféricos ligados ao microcontrolador: como o teclado matricial para digitação de senhas, servo motor para controle de abertura da caixa leitora de cédulas e trava eletrônica para controle de abertura e fechamento do cofre.

Haverá menus para visualização do saldo, abertura do cofre e para configuração de senhas para o cofre.

Neste programa, será programada a memória EEPROM do microcontrolador para salvar dados. Entre estes dados, há a quantia armazenada no cofre e a senha de segurança do cofre, que devem permanecer na memória mesmo com o desligamento do cofre.

A programação da comunicação serial será mostrada a parte em outra seção de Anexos

```

*****
* CONFIGURAÇÕES:
MCU: PIC16F877A
Dev.Board: Kit PIC 16F (desenvolvido pelos alunos)
Oscilador: Cristal de Quartzo 8 Mhz
Software Compilador: mikroC PRO for PIC 7.2
*****

* NOTAS:
*****/

//Pinagem do Display
sbit LCD_RS at RD2_bit;
sbit LCD_EN at RD3_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at RD7_bit;
sbit LCD_RS_Direction at TRISD2_bit;
sbit LCD_EN_Direction at TRISD3_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;

// ***** Variáveis *****
unsigned int tempo = 0;
char flag = 0b00000001;
int Periodo=0;
char Clear=0;
char Azul=0;
char Verde=0;
char Vermelho=0;
char Cedula=0;
int Notas=0;
char Cont1=0;
char Cont2=0;
char Cont3=0;
char Teclado_Cont=0;
char Teclado_Flag=0;
char Senha0=0;
char Senha_Cont=0;
char SenhaVetor[5];

```

```

char SenhaConfere[5];
char Confere=0;
char SenhaOK=0;
char FlagEeprom=0;
char ImprimeLCD=0;
char Tecla1=0;
char Tecla2=0;
char N_02=0,N_05=0,N_10=0,N_20=0,N_50=0,N_100=0;
char SensorIR=0;
char ServoMotor = 0;
unsigned int Saldo01=0;
unsigned char recData;           // Declaração da variável para receber dados
unsigned int saldocomunicacao = 7452; // Declaração da variável do saldo

//***** Interrupções *****
void interrupt()
{
  if(CCP1IF_bit)           // Flag interrupção modulo CCP
  {
    if(flag.f0)           // Flag para executa sempre para mudar o modo de captura
    {
      TMR1L = 0;           // Zera contador
      TMR1H = 0;
      TMR1ON_bit =1;       // Liga timer 1
      CCP1M0_bit = 0;      // Muda modo de captura de nível alto para nível baixo
      flag.f0 = 0;         // Zera flag
    }
    else                   // se a flag estiver em zero executa o comando
    {
      TMR1ON_bit =0;       // Desliga timer 1
      tempo = (TMR1H<<8)+TMR1L; // junta os valores de TMR1H e TMR1L colocando o
      // valor de TMR1H 8 bits para frente
      CCP1M0_bit = 1;      // Muda modo de captura de nível baixo para nível alto
      flag.f0 = 1;         // Habilita flag para próxima captura
    }
  }
  CCP1IF_bit=0;           // Zera Flag
}

```

```

if(PIR1.TMR2IF==1) // Flag estouro do timer 2
{
  PIR1.TMR2IF=0; // Zera flag
  Teclado_cont++;
  if(SensorIR<2)
  {
    if(adc_read(0)>=100) // Verifica se foi inserido uma cédula
    {
      SensorIR=1; // Aciona logica do sensor RGB
      PORTC.RC3=1; // Liga sensor RGB
      PORTC.RC4=1; // Liga sensor RGB
    }
    if(adc_read(0)<=99) // Verifica se está sem cédulas
    {
      SensorIR=0; // Desliga logica do sensor RGB
      PORTC.RC3=0; // Desliga sensor RGB
      PORTC.RC4=0; // desliga sensor RGB
    }
  }
}

```

```

//***** Interrupção Teclado *****

```

```

Portb=2;
if(Portb.rb1==1)
{
  if(Portb.rb5==1 && Teclado_Flag.f0==0) // Tecla 1
  {
    Teclado_Flag.f0=1;
    Teclado_Cont=0;
    Tecla1=1;
    Tecla2=1;
    Senha_Cont++;
  }
  if(Portb.rb6==1 && Teclado_Flag.f0==0) //Tecla 2
  {
    Teclado_Flag.f0=1;
    Teclado_Cont=0;
    Tecla1=2;
    Tecla2=2;
    Senha_Cont++;
  }
}

```

```
if(Portb.rb7==1 && Teclado_Flag.f0==0) //Tecla 3
{
    Teclado_Flag.f0=1;
    Teclado_Cont=0;
    Tecla1=3;
    Tecla2=3;
    Senha_Cont++;
}
}
Portb=4;
if(Portb.rb2==1)
{
    if(Portb.rb5==1 && Teclado_Flag.f0==0) // Tecla 4
    {
        Teclado_Flag.f0=1;
        Teclado_Cont=0;
        Tecla1=4;
        Tecla2=4;
        Senha_Cont++;
    }
    if(Portb.rb6==1 && Teclado_Flag.f0==0) // Tecla 5
    {
        Teclado_Flag.f0=1;
        Teclado_Cont=0;
        Tecla1=5;
        Tecla2=5;
        Senha_Cont++;
    }
    if(Portb.rb7==1 && Teclado_Flag.f0==0) // Tecla 6
    {
        Teclado_Flag.f0=1;
        Teclado_Cont=0;
        Tecla1=6;
        Tecla2=6;
        Senha_Cont++;
    }
}
Portb=8;
if(Portb.rb3==1)
{
```

```

if(Portb.rb5==1 && Teclado_Flag.f0==0) // Tecla 7
{
    Teclado_Flag.f0=1;
    Teclado_Cont=0;
    Tecla1=7;
    Tecla2=7;
    Senha_Cont++;
}
if(Portb.rb6==1 && Teclado_Flag.f0==0) // Tecla 8
{
    Teclado_Flag.f0=1;
    Teclado_Cont=0;
    Tecla1=8;
    Tecla2=8;
    Senha_Cont++;
}
if(Portb.rb7==1 && Teclado_Flag.f0==0) // Tecla 9
{
    Teclado_Flag.f0=1;
    Teclado_Cont=0;
    Tecla1=9;
    Tecla2=9;
    Senha_Cont++;
}
}
Portb=16;
if(Portb.rb4==1)
{
    if(Portb.rb5==1 && Teclado_Flag.f0==0) //Tecla *
    {
        Teclado_Flag.f0=1;
        Teclado_Cont=0;
        Tecla1=10;
        Tecla2=0;
        Senha_Cont++;
    }
    if(Portb.rb6==1 && Teclado_Flag.f0==0) // Tecla 0
    {
        Teclado_Flag.f0=1;
        Teclado_Cont=0;
    }
}

```



```

    Tecla1=0;
    Tecla2=0;
    Senha_Cont++;
}
if(Portb.rb7==1 && Teclado_Flag.f0==0) // Tecla #
{
    Teclado_Flag.f0=1;
    Teclado_Cont=0;
    Tecla1=11;
    Tecla2=0;
    Senha_Cont++;
}
}
if(Teclado_Cont>=30)
{
    if( Teclado_Flag.f0==1)
    {
        Teclado_Flag.f0=0;
        Teclado_Flag.f1=1;
    }
    Teclado_Cont=0;
}
}
}
//***** Sub-rotinas RGB*****
//          Captura
void captura()
{
    Período= tempo; // Atribui a variável Tempo a Captura
}
//          Sensor RGB - Clear
void Clear_out()
{
    portc.rc1=1; //S2=1; // Configura Fotodiodo Clear
    portc.rc0=0; //S3=0;
    delay_ms(5); // Aguarda 5 milissegundos
    Período=0; // Zera última leitura
    Captura(); // Chama Sub-rotina de captura
    Clear = Período; // Atribui a variável clear
}

```

```

//          Sensor RGB - Vermelho
void Vermelho_out()
{
  portc.rc1=0; //S2=0;          // Configura Fotodiodo Vermelho
  portc.rc0=0; //S3=0;
  delay_ms(5);                // Aguarda 5 milissegundos
  Periodo=0;                  // Zera última leitura
  Captura();                  // Chama Sub-rotina de captura
  Vermelho = Periodo;        // Atribui a variável vermelho
}

//          Sensor RGB - Verde
void Verde_out()
{
  portc.rc1=1; //S2=1;          // Configura Fotodiodo Verde
  portc.rc0=1; //S3=1;
  delay_ms(5);                // Aguarda 5 milissegundos
  Periodo=0;                  // Zera última leitura
  Captura();                  // Chama Sub-rotina de captura
  Verde = Periodo;           // Atribui a variável Verde
}

//          Sensor RGB - Azul
void Azul_out()
{
  portc.rc1=0; //S2=0;          // Configura Fotodiodo Azul
  portc.rc0=1; //S3=1;
  delay_ms(5);                // Aguarda 5 milissegundos
  Periodo=0;                  // Zera última leitura
  Captura();                  // Chama Sub-rotina de captura
  Cont3=1;                    // Chama Sub-rotina de captura
  Azul = Periodo;             // Atribui a variável Azul
}

// ***** Leitura de Cédulas (5, 200) *****
void Cedulas()
{
  char Subtrai=0;             // Variável
  char Subtrai2=0;
  if(cont3==1)                // Inicia leitura

```

```

{
  Cont3++; // contador para iniciar próxima leitura
  if(Azul<Vermelho && Azul<Verde) // Menor valor é a cor lida pelo sensor, lógica
  verifica se a cédula é Azul
  {

    if(Verde>Vermelho) // Verifica Vermelho na Cédula
    {
      Subtrai=(Verde-Azul); // Verifica diferença entre Verde e Azul
      Subtrai2=(Verde-Vermelho);
      if(Verde >=71)
      {
        Cedula=5; // Cédula R$ 5,00
        Cont3=0; // Sai da Sub-rotina de leitura de Cédulas
      }
      if(Verde <=70 )
      {
        if(Azul>=42)
        {
          Cedula=200; // Cédula R$ 200,00
          Cont3=0; // Sai da Sub-rotina de leitura de Cédulas
        }
        if(Azul<=41)
        {
          Cedula=5;
          Cont3=0;
        }
      }
    }
  }
}

```

// ***** Leitura de Cédulas (10, 20, 50) *****

```

void Cedulas2()
{
  char Subtrai=0;
  if(Cont3==2) // Inicia leitura
  {
    Cont3++; // contador para iniciar próxima leitura
  }
}

```

```

if(Vermelho<Azul && Vermelho<Verde ) // Verificar maior presença de Vermelho
{
    if(Azul<Verde) // Verificar maior presença de Azul
    {
        Subtrai=(Verde-Azul); // Verifica a diferença entre Verde e Azul
        if(Subtrai<=5)
        {
            Cedula=50; // Cédula R$ 50,00
            Cont3=0; // Sai da Sub-rotina de leitura de Cédulas
        }
        if(Subtrai>=6)
        {
            Cedula=10; // Cédula R$ 10,00
            Cont3=0; // Sai da Sub-rotina de leitura de Cédulas
        }
    }
    if(Azul>Verde) // Verifica Azul maior que verde
    {
        Cedula=20; // Cédula R$ 20,00
        Cont3=0; // Sai da Sub-rotina de leitura de Cédulas
    }
}
}
}

```

// ***** Leitura de Cédulas (2 e 100) *****

```

void Cedula3()
{
    char Subtrai=0;
    char Subtrai2=0;
    if(Cont3==3) // Inicia leitura
    {
        Cont3=1; // contador para iniciar próxima leitura
        if(Azul<Vermelho && Azul<Verde) // Verifica maior presença de azul
        {
            if (Verde<Vermelho) // Verifica maior presença de verde
            {
                Subtrai=(Verde-Azul); // Verifica a diferença entre Verde e Azul
                Subtrai2=(Vermelho-Verde);
                if(Vermelho >=80)

```

```
{
  if(Subtrai2>=24)
  {
    Cedula=100;           // Cédula R$:100,00
    Cont3=0;
  }
  else if(Subtrai2<=23)
  {
    Cedula=2;           // Cédula R$:2,00
    Cont3=0;           // Sai da Sub-rotina de leitura de Cédulas
  }
}
if(Vermelho <= 79)
{
  if(Subtrai2<=7)
  {
    Cedula=200;         // Cédula R$:200,00
    Cont3=0;           // Sai da Sub-rotina de leitura de Cédulas
  }
  else if(Vermelho >=60 && Vermelho <=69 && Subtrai2>=13)
  {
    Cedula=100;         // Cédula R$:2,00
    Cont3=0;           // Sai da Sub-rotina de leitura de Cédulas
  }
  else if(Subtrai2 >=8 && Cont3==1)
  {
    Cedula=2;           // Cédula R$:2,00
    Cont3=0;           // Sai da Sub-rotina de leitura de Cédulas
  }
}
}
}
}
}
}
}
}
}
}
```

```
//***** Sub-Rotina Servo motor *****
```

```
void Servo_motor()
```

```
{
```

```
int i=0; // Variável looping For
```

```

GIE_bit = 0; // Desabilita interrupções
PEIE_bit = 0;

while(ServoMotor==1) // Looping abre e fecha porta
{
  for(i=0;i<=30;i++) // Abre a porta com servo motor
  {
    porta.ra1=1;
    delay_us(1000);
    porta.ra1=0;
    delay_us(19000);
  }
  delay_ms(1000);
  for(i=0;i<=30;i++) // Fecha porta com Servo motor
  {
    porta.ra1=1;
    delay_us(2200);
    porta.ra1=0;
    delay_us(17800);
  }
  delay_ms(1000);
  GIE_bit = 1; // Habilita interrupções
  PEIE_bit = 1;
  ServoMotor=0; // Sai do Looping
  Flag.f3=1; // Flag para rotina de Saldo
}
}

```

```

//*****Apresentação LCD *****

```

```

// Sub-rotina de mensagem inicial do display

```

```

void Start_Menu()
{
  char Num = 0, Posicao=0;
  const char Apresenta01[] = { ' ', ' ', ' ', 'A', 'P', 'R', 'E', 'S', 'E', 'N', 'T', 'A', 'C', 'A', 'O', ' ', ' ', ' ' }; // Vetor
  const char Apresenta02[] = { ' ', ' ', ' ', ' ', ' ', ' ', 'T', 'C', 'C', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' };
  const char Apresenta03[] = { ' ', ' ', 'D', 'E', 'N', 'I', 'M', 'A', 'R', ' ', ' ', ' ', ' ', 'L', 'E', 'O', ' ', ' ', ' ' };
  const char Apresenta04[] = { ' ', ' ', ' ', ' ', ' ', ' ', 'V', 'I', 'C', 'T', 'O', 'R', ' ', ' ', ' ', ' ', ' ', ' ' };
}

```

```

for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
    Posicao= Num+0; // Centraliza Texto no display LCD
    Lcd_Chr(1,Posicao,Apresenta01[Num]); // Imprime mensagem na primeira linha do display
}
for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
    Posicao= Num+0; // Centraliza Texto
    Lcd_Chr(2,Posicao,Apresenta02[Num]); // Imprime mensagem na segunda linha do display
}
Delay_ms(2000); // Aguarda 2 segundos para trocar a mensagem
for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
    Posicao= Num+0; // Centraliza Texto
    Lcd_Chr(1,Posicao,Apresenta03[Num]); // Imprime mensagem na primeira linha do display
}
for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
    Posicao= Num+0; // Centraliza Texto
    Lcd_Chr(2,Posicao,Apresenta04[Num]); // Imprime mensagem na segunda linha do display
}
delay_ms(2000); // Aguarda 2 segundos para trocar a mensagem
}

// Mensagem Display
void Menu_01()
{
    char Num = 0, Posicao=0;
    const char Menu01[] = {' ',' ',' ','A','C','E','S','S','A','R',' ','M','E','N','U',' ',' '}; // Vetor para Display
    const char Menu02[] = {' ',' ',' ',' ','D','I','G','I','T','E',' ','#',' ',' ',' '}; // Vetor para Display

    for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
    {
        Posicao= Num+0; // Centraliza Texto no display LCD
        Lcd_Chr(1,Posicao,Menu01[Num]); // Imprime mensagem na primeira linha do display
    }
    for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
    {
        Posicao= Num+0; // Centraliza Texto
        Lcd_Chr(2,Posicao,Menu02[Num]); // Imprime mensagem na segunda linha do display
    }
}

```

```

}
Delay_ms(2500); // Aguarda 2 segundos para trocar a mensagem
}

// ***** Sub Menu *****
void Menu_02()
{
char Num = 0, Posicao=0;
const char Msg01[] = {' ','S','E','N','H','A',,,,,,,,,,,,,,'1',,,,,,,,,}; // Vetor para Display
const char Msg02[] = {' ','S','A','L','D','O',,,,,,,,,,,,,,'2',,,,,,,,,}; // Vetor para Display
const char Msg03[] = {' ','A','B','R','I','R',' ','P','O','R','T','A',,,,,,,,,,'3',,,,,,,,,}; // Vetor para Display
const char Msg04[] = {' ','S','A','I','R',,,,,,,,,,,,,,'4',,,,,,,,,}; // Vetor para Display

if(Tecla1==0)
{
for (Num = 1; Num<=16; Num++) // Looping For para imprimir no Display
{
Posicao= Num+0; // Centraliza Texto no display LCD
Lcd_Chr(1,Posicao,Msg01[Num]); // Imprime mensagem na primeira linha do display
}
for (Num = 1; Num<=16; Num++) // Looping For para imprimir no Display
{
Posicao= Num+0; // Centraliza Texto no display LCD
Lcd_Chr(2,Posicao,Msg02[Num]); // Imprime mensagem na primeira linha do display
}

Delay_ms(2000);
}
if(Tecla1==0)
{
for (Num = 1; Num<=16; Num++) // Looping For para imprimir no Display
{
Posicao= Num+0; // Centraliza Texto no display LCD
Lcd_Chr(1,Posicao,Msg03[Num]); // Imprime mensagem na primeira linha do display
}
for (Num = 1; Num<=16; Num++) // Looping For para imprimir no Display
{
Posicao= Num+0; // Centraliza Texto no display LCD
Lcd_Chr(2,Posicao,Msg04[Num]); // Imprime mensagem na primeira linha do display
}
}

```



```

Delay_ms(2000);
}
}

// ***** Sub menu de Opções *****

void Menu_Opcoes()
{
if(Tecla1==1)      // Senha
{
    SensorIR=3;
    Senha_Cont=0;
    Teclado_Flag.f1=0; // Sub Rotina Senha_02
    Teclado_Flag.f2=1; // Sub Rotina Senha_01
}
if(Tecla1==2)     // Saldo
{
    SensorIR=4;
    Flag.f3=1;
    Tecla1=0;
}
if(Tecla1==3)     // Porta
{
    SensorIR=5;
    Teclado_Flag.f2=1;
    Teclado_Flag.f1=0;
    Senha_Cont=0;
}
}

//***** Sub Menu Troca de Senha *****

void Senha_01()
{
char Num = 0, Posicao=0;
const char Menu01[] = {' ','S','E','N','H','A',' ','A','T','U','A','L',':',' ',' ',' '}; // Vetor para Display
const char Menu02[] = {' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}; // Vetor para Display

if(Teclado_Flag.f2==1)
{

```

```

for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
    Posicao= Num+0; // Centraliza Texto no display LCD
    Lcd_Chr(1,Posicao,Menu01[Num]); // Imprime mensagem na primeira linha do display
}
for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
    Posicao= Num+0; // Centraliza Texto
    Lcd_Chr(2,Posicao,Menu02[Num]); // Imprime mensagem na segunda linha do display
}

Senha_Cont=0;
Teclado_Flag.f2=0;
Teclado_Flag.f1=0;
}
}

void Senha_02()
{
    char Num=0;
    char Coluna=0;
    char txt[7];
    char *res;

    if(Teclado_Flag.f1==1)
    {
        SenhaVetor[Senha_Cont]= Tecla2;
        IntToStr(SenhaVetor[Senha_Cont],txt); // Converte inteiro para string
        res = ltrim(txt);
        Coluna=Senha_Cont+6; // Centraliza Texto no display LCD
        Lcd_out(2,Coluna,res);
        Teclado_Flag.f1=0;
        ImprimeLCD++;
    }

    if(Senha_Cont==4 && ImprimeLCD==4 )
    {
        Senha_Cont=0;
        ImprimeLCD=0;
        Confere=1;
    }
}

```

```
if(FlagEeprom.f0==1)
{
    SenhaOk=1;
    Confere=0; //2;
}
delay_ms(2000);
}
}
```

```
void Confere_Senha()
```

```
{
char num=0;
char Cont0=0;

if(Confere==1)
{
    for(Num=1;Num<=4;Num++)
    {
        if(SenhaConfere[Num] == SenhaVetor[Num] )
        {
            Cont0++;
        }
    }
    if(Cont0==4) // senha correta
    {
        FlagEeprom.f0=1;
        Confere=0;
        Teclado_Flag.f3=1;
    }
    if(Cont0<4) // senha incorreta
    {
        FlagEeprom.f0=0;
        Confere=0;
        Teclado_Flag.f4=1;
    }
}
}
```

```

void Senha_03()
{
char Num = 0, Posicao=0;
const char Menu01[] = {' ','N','O','V','A',' ','S','E','N','H','A',':',' ',' ',' ',' ',' '}; // Vetor para Display
const char Menu02[] = {' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}; // Vetor para Display
const char Menu03[] = {' ','S','E','N','H','A',' ','E','R','R','A','D','A',' ',' ',' ',' ',' '}; // Vetor para Display

if(Teclado_Flag.f3==1)
{
for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
Posicao= Num+0; // Centraliza Texto no display LCD
Lcd_Chr(1,Posicao,Menu01[Num]); // Imprime mensagem na primeira linha do display
}
for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
Posicao= Num+0; // Centraliza Texto
Lcd_Chr(2,Posicao,Menu02[Num]); // Imprime mensagem na segunda linha do display
}
Teclado_Flag.f3=0;
}

if(Teclado_Flag.f4==1)
{
for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
Posicao= Num+0; // Centraliza Texto no display LCD
Lcd_Chr(1,Posicao,Menu03[Num]); // Imprime mensagem na primeira linha do display
}
for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
{
Posicao= Num+0; // Centraliza Texto
Lcd_Chr(2,Posicao,Menu02[Num]); // Imprime mensagem na segunda linha do display
}

Tecla1=0;
Tecla2=0;
Senha_Cont=0;
Teclado_Flag.f1=0;
Teclado_Flag.f2=1;

```

```

    Teclado_Flag.f4=0;
    delay_ms(2500);
}
}

void Senha0000()
{
    char GeraZero=0;
    char Num = 0;
    char Local = 0;

    Senha0 = EEPROM_Read(0x04);
    if(Senha0==15)
    {
        GeraZero=1;
    }

    if(GeraZero==0)
    {
        for(Num=1;Num<=4;Num++)
        {
            Local= Num+4;
            EEPROM_Write(Local, 0);
            delay_ms(10);
        }
    }
}

void Le_EEPROM3() // Lógica para ler os dados da eeprom
{
    char Num=0;
    char Local=0;
    char Texto[7]; // Vetor para Imprimir os dados
    char *res; // Ponteiro para retirar caracteres vazios

    for(Num=1;Num<=4;Num++)
    {
        Local= Num+4;
        SenhaVetor[Num] = EEPROM_Read(Local);
        SenhaConfere[Num] = EEPROM_Read(Local);
    }
}

```



```

if(Teclado_Flag.f2==1)
{
    for (Num = 0; Num<=16; Num++)                // Looping For para imprimir no Display
    {
        Posicao= Num+0;                            // Centraliza Texto no display LCD
        Lcd_Chr(1,Posicao,Menu01[Num]);           // Imprime mensagem na primeira linha do display
    }
    for (Num = 0; Num<=16; Num++)                // Looping For para imprimir no Display
    {
        Posicao= Num+0;                            // Centraliza Texto
        Lcd_Chr(2,Posicao,Menu02[Num]);           // Imprime mensagem na segunda linha do display
    }
    Teclado_Flag.f2=0;
}
}

```

```

void Porta_02()
{
    char Num = 0, Posicao=0;
    const char Menu01[] = { ' ','A','B','R','I','R',' ','P','O','R','T','A',' ',' ',' ',' ',' '}; // Vetor para Display
    const char Menu02[] = { ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '}; // Vetor para Display
    const char Menu03[] = { ' ','S','E','N','H','A',' ','E','R','R','A','D','A',' ',' ',' ',' '}; // Vetor para Display

```

```

if(Teclado_Flag.f3==1)
{
    for (Num = 0; Num<=16; Num++)                // Looping For para imprimir no Display
    {
        Posicao= Num+0;                            // Centraliza Texto no display LCD
        Lcd_Chr(1,Posicao,Menu01[Num]);           // Imprime mensagem na primeira linha do display
    }
    for (Num = 0; Num<=16; Num++)                // Looping For para imprimir no Display
    {
        Posicao= Num+0;                            // Centraliza Texto
        Lcd_Chr(2,Posicao,Menu02[Num]);           // Imprime mensagem na segunda linha do display
    }
    //Teclado_Flag.f3=0;
}

```

```

if(Teclado_Flag.f4==1)

```

```

{
  for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
  {
    Posicao= Num+0; // Centraliza Texto no display LCD
    Lcd_Chr(1,Posicao,Menu03[Num]); // Imprime mensagem na primeira linha do display
  }
  for (Num = 0; Num<=16; Num++) // Looping For para imprimir no Display
  {
    Posicao= Num+0; // Centraliza Texto
    Lcd_Chr(2,Posicao,Menu02[Num]); // Imprime mensagem na segunda linha do display
  }
  Senha_Cont=0;
  Teclado_Flag.f1=0;
  Teclado_Flag.f2=1;
  Teclado_Flag.f4=0;
  delay_ms(2500);
}
}

```

```

void Porta_03()
{
  if(Teclado_Flag.f3==1)
  {
    PORTA.RA5 = 1;
    delay_ms(4000);
    PORTA.RA5 = 0;
    Teclado_Flag.f3=0;
    SensorIR=2;
    Tecla1=0;
    Senha_Cont=0;
    ImprimeLCD=0;
    FlagEeprom.f0=0;
    SenhaOk=0;
    Confere=0;
    EEPROM_Write(0x02, 0 );
    delay_ms(10);
  }
}

```



```

// ***** EEPROM *****
void Grava_EEPROM_Saldo()           // Logica para gravar Valor na eeprom
{
    char Dado_H=0;
    char Dado_L=0;
    Dado_H= Saldo01/256;             // Divide o Valor por 256 e salva em Dado_H
    Dado_L= Saldo01%256;            // Resto da divisão de 256 e salva em Dado_L
    EEPROM_Write(0x00, Dado_H);     // Salva Dado_H em uma posição da eeprom
    delay_ms(10);                   // Aguarda 10 ms para salvar
    EEPROM_Write(0x01, Dado_L);     // Salva o Dado_L em outra posição da eeprom
    delay_ms(10);                   // Aguarda 10 ms
    EEPROM_Write(0x02, 15);         // Limpa a variável do eeprom para iniciar com Saldo 0(Zero)
    delay_ms(10);
}

void Le_EEPROM()                   // Logica para ler os dados da eeprom
{
    char Dado_H=0;
    char Dado_L=0;
    char Saldo0=0;
    char GeraZero=0;

    Saldo0 = EEPROM_Read(0x02);
    if(Saldo0==15)
    {
        GeraZero=1;
    }
    if(GeraZero==0)
    {
        EEPROM_Write(0x00, 0 );
        delay_ms(10);
        EEPROM_Write(0x01, 0 );
        delay_ms(10);
    }
    Dado_H = EEPROM_Read(0x00);     // Atribui os dados da eeprom para variável Dado_H
    Dado_L = EEPROM_Read(0x01);     // Atribui os dados da eeprom para variável Dado_L
    Saldo01 = (Dado_H*256)+Dado_L;  // Move a variável Dado_H 8 casas para frente, soma
    // com a variável Dado_L e atribui a variável valor
}

```

```

void Teclado_Menu()
{

if(tecla1==11)
{
  SensorIR=2;
  Tecla1=0;
}
}

void Imprime_Valor(char ValorLCD)
{
  char Num = 0, Posicao=0;
  const char Valor01[] = {' ','R','$',':',' ','2',' ','0','0'};           // Vetor para Display
  const char Valor02[] = {' ','R','$',':',' ','5',' ','0','0'};           // Vetor para Display
  const char Valor03[] = {' ','R','$',':',' ','1','0',' ','0','0'};       // Vetor para Display
  const char Valor04[] = {' ','R','$',':',' ','2','0',' ','0','0'};       // Vetor para Display
  const char Valor05[] = {' ','R','$',':',' ','5','0',' ','0','0'};       // Vetor para Display
  const char Valor06[] = {' ','R','$',':',' ','1','0','0',' ','0','0'};   // Vetor para Display
  const char Valor07[] = {' ','R','$',':',' ','2','0','0',' ','0','0'};   // Vetor para Display
  const char Valor08[] = {' ','E','r','r','o'};                           // Vetor para Display
  const char Valor09[] = {' ','V','i','r','e',' ','a',' ','C','e','d','u','l','a'}; // Vetor para Display

  Lcd_Cmd(_LCD_CLEAR);           // Apaga Display
  if(ValorLCD==2)                 // Verifica nota de R$ 2,00
  {
    for (Num = 1; Num<=9; Num++) // Looping For para imprimir no Display
    {
      Posicao= Num+4;             // Centraliza Texto no display LCD
      Lcd_Chr(1,Posicao,Valor01[Num]); // Imprime mensagem na primeira linha do display
    }
  }
  if(ValorLCD==5)                 // Verifica nota de R$ 5,00
  {
    for (Num = 1; Num<=9; Num++) // Looping For para imprimir no Display
    {
      Posicao= Num+4;             // Centraliza Texto no display LCD
      Lcd_Chr(1,Posicao,Valor02[Num]); // Imprime mensagem na primeira linha do display
    }
  }
}

```

```

if(ValorLCD==10) // Verifica nota de R$ 10,00
{
  for (Num = 1; Num<=9; Num++) // Looping For para imprimir no Display
  {
    Posicao= Num+4; // Centraliza Texto no display LCD
    Lcd_Chr(1,Posicao,Valor03[Num]); // Imprime mensagem na primeira linha do display
  }
}
if(ValorLCD==20) // Verifica nota de R$ 20,00
{
  for (Num = 1; Num<=9; Num++) // Looping For para imprimir no Display
  {
    Posicao= Num+4; // Centraliza Texto no display LCD
    Lcd_Chr(1,Posicao,Valor04[Num]); // Imprime mensagem na primeira linha do display
  }
}
if(ValorLCD==50) // Verifica nota de R$ 50,00
{
  for (Num = 1; Num<=9; Num++) // Looping For para imprimir no Display
  {
    Posicao= Num+4; // Centraliza Texto no display LCD
    Lcd_Chr(1,Posicao,Valor05[Num]); // Imprime mensagem na primeira linha do display
  }
}
if(ValorLCD==100) // Verifica nota de R$ 100,00
{
  for (Num = 1; Num<=9; Num++) // Looping For para imprimir no Display
  {
    Posicao= Num+4; // Centraliza Texto no display LCD
    Lcd_Chr(1,Posicao,Valor06[Num]); // Imprime mensagem na primeira linha do display
  }
}
if(ValorLCD==200) // Verifica nota de R$ 100,00
{
  for (Num = 1; Num<=9; Num++) // Looping For para imprimir no Display
  {
    Posicao= Num+4; // Centraliza Texto no display LCD
    Lcd_Chr(1,Posicao,Valor07[Num]); // Imprime mensagem na primeira linha do display
  }
}

```



```

{
  Imprime_Valor(Notas);           // Imprime Valor
  ServoMotor=1;                   // Flag servo motor
  Servo_motor();                  // Chama Sub-rotina servo motor
  Cont2=0;                         // Zera contador de Erro
  N_05=0;                          // Zera contador de verificação
}
}
if(Notas==10)                      // Logica impressão nota de 10 reais
{
  N_10++;                          // Contador verificação de cédulas
  if(N_10>=1)                       // Após 1 leitura imprime valor
  {
    Imprime_Valor(Notas);           // Imprime Valor
    ServoMotor=1;                   // Flag servo motor
    Servo_motor();                  // Chama Sub-rotina servo motor
    Cont2=0;                         // Zera contador de Erro
    N_10=0;                          // Zera contador de verificação
  }
}
if(Notas==20)                      // Logica impressão nota de 20 reais
{
  Imprime_Valor(Notas);           // Imprime Valor
  ServoMotor=1;                   // Flag servo motor
  Servo_motor();                  // Chama Sub-rotina servo motor
  Cont2=0;                         // Zera contador de Erro
}
if(Notas==50)                      // Logica impressão nota de 50 reais
{
  Imprime_Valor(Notas);           // Imprime Valor
  ServoMotor=1;                   // Flag servo motor
  Servo_motor();                  // Chama Sub-rotina servo motor
  Cont2=0;                         // Zera contador de Erro
}
if(Notas==100)                    // Logica impressão nota de 100 reais
{
  Imprime_Valor(Notas);           // Imprime Valor
  ServoMotor=1;                   // Flag servo motor
  Servo_motor();                  // Chama Sub-rotina servo motor
  Cont2=0;                         // Zera contador de Erro
}

```

```

}
if(Notas==200) // Logica impressão nota de 200 reais
{
  Imprime_Valor(Notas); // Imprime Valor
  ServoMotor=1; // Flag servo motor
  Servo_motor(); // Chama Sub-rotina servo motor
  Cont2=0; // Zera contador de Erro
}
if(Vermelho >150) // Desconsidera leitura em vazio
{
  Cont2=0; // Zera contador de Erro
}
if(Cont2==3) // Após 3 erros de leitura imprime mensagem
{
  Imprime_Valor(Cont2); // Imprime mensagem de erro
  Delay_ms(1000); // Aguarda 1 segundo
  Cont2=0; // Zera contador de Erro
}
Cont1=0; // Zera contador de leitura
Delay_ms(10); // Aguarda 100 milissegundos
Lcd_cmd(_lcd_clear); // Limpa display
}
}

void Comicao()
{
if (UART1_Data_Ready()) // Verifica se chegou dado pela Serial
{
  recData = UART1_Read(); // Lê dado recebido e guarda na variável recData
  delay_ms(1); // Estabiliza canal UART1

if(recData == 'a') //Se chegar "a", Abre a porta
{
  Teclado_Flag.f3=1;
  Porta_02();
  Porta_03();
  Le_EEPROM();
  SensorIR=0;
  recData='0'; // Envia valor 0 para a variável recData
}
}
}

```



```

else if(recData == 'w') //Se chegar "w", envia valor do saldo
{
    char txt[7];
    char *res;
    intToStr(Saldo01,txt); // Converte inteiro para string
    res = ltrim(txt);
    UART1_Write_Text("R$: ");
    UART1_Write_Text(res);
    UART1_Write_Text(",00");
    recData='0'; // Envia valor 0 para a variável recData

} // End do "else if"
} //End ""if da leitura da Uart"
}

// ***** Função Principal *****
void main()
{
    CMCON=7;
    ADCON1= 7; // Converte entrada analógica em digital
    TRISC.RC2 = 1; // Configura pino como entrada
    T1CON = 0b00010000; // Tempo de incremento de 1 us timer 1
    TMR1L = 0; // Zera Variáveis
    TMR1H = 0;

    CCP1M3_bit = 0; // CONFIGURA CCP como captura para borda de subida (L-H)
    CCP1M2_bit = 1;
    CCP1M1_bit = 0;
    CCP1M0_bit = 1;

    GIE_bit = 1; // Habilita interrupções
    PEIE_bit = 1;
    TMR1IE_bit = 0;

    CCP1IE_bit = 1;
    CCP1IF_bit = 0;

```

```

PIE1.TMR2IE= 1;           // Habilita a interrupção do timer2
T2CON= 0b00100110;      // Liga o timer2, Postscaler 1:5, TMR2ON Desligado, Prescaler 1:16
PR2= 250;                // Compara o TMR2 com PR2, PR2 igual a 250
                           // Ciclo de máquina x PR2 x prescaler x postscaler
                           // 5us x 250 x 5 x 16 = 10 ms

PORTA.RA1=0;
PORTA.RA5=0;
TRISA.RA0=1; //sensor presença de notas           // Configuração Portas
TRISA.RA1=0;
TRISA.RA5=0;

Trisb= 0b11100000;
Portb= 0b00000000;

TRISC.rc7 = 1; // RC7 = Entrada -> RX
PORTC.rc7 = 0; // RC7 = Entrada -> RX

TRISC.RC5=0; //S1
TRISC.RC4=0; //S0
PORTC.RC5=0;
PORTC.RC4=0;

TRISC.RC0=0; //S3
TRISC.RC1=0; //S2
PORTC.RC0=0; //S3
PORTC.RC1=0; //S2
TRISC.RC3=0; // Vcc
PORTC.RC3=0;

RCSTA.SPEN = 0; // Desliga Porta Serial habilitada pelo Bootloader
Uart1_init(9600); // CONFIGURA CANAL SERIAL A 9600 bps
Delay_ms(10); // Estabilização circuito UART1

lcd_init(); // Inicialização display
lcd_cmd(_lcd_clear);
lcd_cmd(_lcd_cursor_off);
Senha0000();
Le_EEPROM3();
Delay_ms(2500);

```

```

Start_Menu();
Menu_01();
Le_EEPROM();

while(1)
{
    while(SensorIR==0) // Logica sensor de presença
    {
        Comicacao();
        Msg_01(SensorIR); // Imprime mensagem no display
        Teclado_Menu();
        PORTC.RC3=0; // Desliga sensor RGB
        Flag.f2=0; // Flag para leitura
    }
    while(SensorIR==1)
    {
        PORTC.RC3=1; // Liga Sensor RGB
        Msg_01(SensorIR); // Imprime mensagem no display

        if(SensorIR==1 && Flag.f2==0) // Logica para aguardar 2 segundos antes da leitura
        {
            delay_ms(1500); // Aguarda 2 segundos
            Flag.f2=1; // desabilita logica
        }
        Clear_out(); // Sensor RGB Clear
        Vermelho_out(); // Sensor RGB Vermelho
        Verde_out(); // Sensor RGB Verde
        Azul_out(); // Sensor RGB Azul
        delay_ms(1000);

        Cedulas(); // Leituras de cédulas
        Cedulas2(); // Leituras de cédulas
        Cedulas3(); // Leituras de cédulas
        Valor_Cedulas(); // Imprime valor da Cédulas
        Saldo(); // Imprime Saldo
        Notas=0; // Zera variável notas
    }
    while(SensorIR==2) // Menu
    {

```

```
Menu_02();
Menu_Opcoes();
while(SensorIR==3)    // Menu Senha
{
    Senha_01();
    Senha_02();
    Confere_Senha();
    Senha_03();
    Grava_EEPROM_Senha();
}
while(SensorIR==4)    // Menu Saldo
{
    Saldo();
    SensorIR=2;
}

while(SensorIR==5)    // Menu Porta
{
    Porta_01();
    Senha_02();
    Confere_Senha();
    Porta_02();
    Porta_03();
    Le_EEPROM();
}

if (Tecla1==4)
{
    SensorIR=0;
}
}
}
```

ANEXO B – Código fonte – Visual Studio (tela de login)

O código a seguir foi desenvolvido no programa Visual Studio. Nesta parte, foi configurada uma interface gráfica com uma tela para digitação de um *login* e senha. Caso *login* e senha estejam corretos, será permitido acesso ao cofre, cuja programação da interface gráfica será mostrada no Anexo D.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Threading;

namespace Sistema_Supervisorio_do_Identificador_de_cedulas_monetarias
{
    public partial class Form2 : Form
    {
        Thread nt;

        public Form2()
        {
            InitializeComponent();
        }

        private void btEntrar_Click(object sender, EventArgs e)
        {
            if (textBoxUsuario.Text == "TCC" && textBoxSenha.Text == "1234" )
            {
                this.Close();
                nt = new Thread(novoForm);
                nt.SetApartmentState(ApartmentState.STA);
                nt.Start();
            }
            else
            {
                MessageBox.Show("Login Incorreto!");
            }
        }

        private void novoForm()
        {
            Application.Run(new Form1());
        }
    }
}

```

ANEXO C – Código fonte – Visual Studio (Comunicação)

O código a seguir foi elaborado no programa Visual Studio, para gerar a interface gráfica na qual o usuário poderá controlar o cofre de forma remota. Além disso, esta programação foi feita para realizar a comunicação da interface gráfica com o cofre.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports; // Biblioteca Inserida

namespace Sistema_Supervisorio_do_Identificador_de_cedulas_monetarias
{
    public partial class Form1 : Form
    {
        string bfRecebe = string.Empty; //Variável para receber dados
        public delegate void Fdelegate(string a); //Objeto para referenciar um método (Página 102)

        bool invRDO = true;

        public Form1()
        {
            InitializeComponent();
            VarreCOMs(); // Incluir variável
            serialPort1.DataReceived +=
            new SerialDataReceivedEventHandler(SerialPort1_DataReceived); //Método para recebimento
            de dados
        }

        private void SerialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
        //Função para tratar o recebimento de dados
        {
            //throw new NotImplementedException();

            bfRecebe = serialPort1.ReadExisting(); //Variável para receber dados
            this.BeginInvoke(new Fdelegate(recebe_serial), new object[] { bfRecebe });
        }

        public void recebe_serial(string a)
        {
            if (checkBoxAN0.Checked) //Leitura AN0
            {
                textBoxAN0.Text += a;
            }
        }

        private void VarreCOMs()
        {
            #region Config_Port

```

```

String[] valoresPort = SerialPort.GetPortNames();
for (int i = 0; i <valoresPort.Length; i++)
{
    cbBoxPort.Items.Add(valoresPort[i]);
}
cbBoxPort.Text = "COM1";
#endregion

#region Config_Baud
int[] valoresBaud = { 2400, 4800, 9600, 57600, 115200 };
for (int i = 0; i <valoresBaud.Length; i++)
{
    cbBoxBaud.Items.Add(valoresBaud[i].ToString());
}
cbBoxBaud.Text = "9600";
#endregion
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void btConectar_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == true) serialPort1.Close();
    else
    {
        serialPort1.PortName = cbBoxPort.Text;
        serialPort1.BaudRate = int.Parse(cbBoxBaud.Text);
        serialPort1.DataBits = 8;
        serialPort1.Parity = Parity.None;
        serialPort1.StopBits = StopBits.One;
    }
    try
    {
        serialPort1.Open();
        btConectar.Enabled = false;
        btDesconectar.Enabled = true;
        btSair.Enabled = false;
        cbBoxBaud.Enabled = false;
        cbBoxPort.Enabled = false;
        panelMessageConect.BackColor = Color.Green;
        lbStatusConect.Text = "Conectado";
    }
    catch
    {
        MessageBox.Show("Erro de comunicação com a porta Serial!!!");
        MessageBox.Show("Configure Porta COM e o Baud Rate no Controle de Conexão Serial");
        btConectar.Enabled = true;
        btDesconectar.Enabled = false;
        btSair.Enabled = true;
        cbBoxBaud.Enabled = true;
        cbBoxPort.Enabled = true;
        panelMessageConect.BackColor = Color.Yellow;
        lbStatusConect.Text = "Aguardando Conexão";
    }
}

private void btDesconectar_Click(object sender, EventArgs e)

```

```

{
try
    {
        serialPort1.Close();
        btConectar.Enabled = true;
        btDesconectar.Enabled = false;
        btSair.Enabled = true;
        cbBoxBaud.Enabled = true;
        cbBoxPort.Enabled = true;
        panelMessageConect.BackColor = Color.Yellow;
        lbStatusConect.Text = "Desconectado";
    }
catch
    {
        MessageBox.Show("Erro de comunicação com a porta");
        MessageBox.Show("Configure COM Port e o Baud Rate");
        btConectar.Enabled = false;
        btDesconectar.Enabled = true;
        btSair.Enabled = false;
        cbBoxBaud.Enabled = false;
        cbBoxPort.Enabled = false;
        panelMessageConect.BackColor = Color.Green;
        lbStatusConect.Text = "Aguardando Conexão";
    }
}

private void btSair_Click(object sender, EventArgs e)
{
    Close();
}

private void lbStatusConect_Click(object sender, EventArgs e)
{
}

private void panelConectControl_Paint(object sender, PaintEventArgs e)
{
}

private void btRD0_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == true & invRD0 == true)
    {
        serialPort1.Write("a");
        labelRD0.Text = "Aberto";
        labelRD0.BackColor = Color.Red;
        invRD0 = false;
    }
    elseif (serialPort1.IsOpen == true & invRD0 == false)
    {
        serialPort1.Write("a");
        labelRD0.Text = "Fechado";
        labelRD0.BackColor = Color.Lime;
        invRD0 = true;
    }
    if (serialPort1.IsOpen == false)
    {
        MessageBox.Show("Erro de Comunicação, verifique o Controle de Conexão Serial");
    }
}

```



```
}  
}  
  
privatevoid btAN0_Click(objectsender, EventArgs e)  
{  
if (serialPort1.IsOpen)  
    {  
        serialPort1.WriteLine("w");  
        textBoxAN0.Text = "";  
    }  
}  
  
privatevoid pictureBox1_Click(objectsender, EventArgs e)  
{  
  
}  
  
privatevoid checkBoxAN0_CheckedChanged(objectsender, EventArgs e)  
{  
  
}  
  
privatevoid btLimpaAN0_Click(objectsender, EventArgs e)  
{  
    textBoxAN0.Text = "";  
}  
  
privatevoidtextBoxReceber_TextChanged(objectsender, EventArgs e)  
{  
  
}  
  
privatevoidcheckBoxReception_CheckedChanged(objectsender, EventArgs e)  
{  
  
}  
  
privatevoidtextBoxEnviar_TextChanged(objectsender, EventArgs e)  
{  
  
}  
  
privatevoid label4_Click(objectsender, EventArgs e)  
{  
  
    }  
}
```