Instituto Federal de Educação Ciência e Tecnologia de São Paulo

Curso de Graduação em Engenharia Eletrônica

Módulo de Fechadura

RELATÓRIO DA DISCIPLINA INTRODUÇÃO AO DESENVOLVIMENTO DE PROJETOS COM O PROF. RICARDO PIRES E PROF. CÉSAR DA COSTA

Alessandro Silvério da Silva Júnior	SP3037177
Gustavo Senzaki Lucente	SP303724X
Igor Galdeano Rodrigues	SP3037223
Luana Mitiko Chagas Iwamura	SP3037151
Luís Otávio Lopes Amorim	SP3034178

Outubro

São Paulo

SUMÁRIO

1 INTRODUÇÃO 5
1.1 Objetivos 6
1.2 Justificativa 7
1.3 Metodologia 7

2 VALORES E CRONOGRAMA 8
2.1 Orçamento 8
2.2 Cronograma 8

3 SENSORES 9
3.1 Módulo RFID RC522 9
3.2 Teste do sensor 10

4 ATUADORES 11

5 INTEGRAÇÃO 12
5.1 Memória Flash 12
5.2 Módulo RTC 14

6 CONCLUSÃO 16

REFERÊNCIAS 17

A CÓDIGO PARA TESTE DOS SENSORES 20

B CÓDIGO PARA TESTE DOS SENSORES 23

C CÓDIGO FINALIZADO 26

LISTA DE FIGURAS

Figura 1 – Fechadura egípcia 5
Figura 2 – Fechadura de Yale 5
Figura 3 – Fechadura Elétrica 6
Figura 4 – Fechadura Biométrica 6
Figura 5 – Módulo RFID RC522 9
Figura 6 – Esquema elétrico do teste dos sensores 10
Figura 7 – Esquema elétrico do teste do atuador 11
Figura 8 – Árvore de diretórios 13
Figura 9 – Socket para cartão microSD 13
Figura 10 – Esquema com cartão microSD 14
Figura 11 – Esquema finalizado do projeto 15

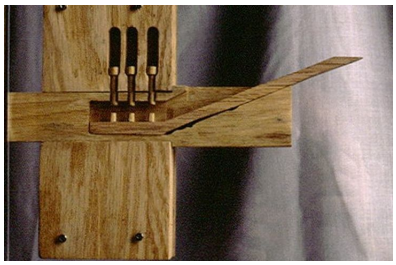
LISTA DE TABELAS

Tabela 1 – Orçamento 8
Tabela 2 – Cronograma 8
Tabela 3 – Conexões do RC522 10
Tabela 4 – Conexões do LCD 10
Tabela 5 – Conexões do socket 14
Tabela 6 – Conexões do módulo RTC DS3231 15
Tabela 7 – Componentes utilizados 16

1 INTRODUÇÃO

A primeira fechadura de que se tem notícia (Figura 1) data de 4000 A.C e foi criada no Egito. Se tratavam de dispositivos de madeira (seu maior defeito) que podiam ser abertos por grandes chaves também feitas de madeira. O funcionamento também era parecido com o de hoje em dia, a chave movia pequenos pistões que ficavam dentro da fechadura. O grande problema era que o material era muito fácil de ser rompido, diminuindo assim a segurança (CORDEIRO, 2018)

Figura 1 – Fechadura egípcia



Fonte: Incrível

Por isso, com a habilidade no manuseio de metais, como ferro e bronze, os romanos utilizaram a mesma ideia e a adaptaram para serem feitas tanto as chaves quanto as fechaduras de metais isso aumentou ainda mais a segurança e permitiu uma diminuição no tamanho de ambos (REPRIZZO, 2018).

Ainda assim, a primeira patente de uma fechadura foi realizada no século XIX pelo médico Abraham Stransbury. E modelo de fechaduras utilizado hoje (Figura 2) em dia, com a chave plana e dentada, foi criado por Linus Yale Jr. em 1861 (CANABARRO, 2019).

Figura 2 – Fechadura de Yale

Fonte: Wikipédia

Hoje em dia, por mais que o modelo de Yale ainda seja utilizado, devido ao avanço da tecnologia, principalmente da eletrônica, o uso de fechaduras mais modernas se torna comum. Assim surgem os modelos elétricos e eletrônicos.

A fechadura elétrica (Figura 3) é mais simples, controlada por um botão que a abre devido a passagem de corrente elétrica por um solenoide. Por outro lado, a eletrônica é mais complexa e pode ser feita de vários jeitos dentre eles com abertura por senha, sensor RFID, impressão digital (Figura 4) ou até mesmo leitura de íris (PIRES, 2020).

Figura 3 – Fechadura Elétrica

Fonte: Leroy Merlin

Figura 4 – Fechadura Biométrica

Fonte: Madeira Madeira

1.1 Objetivos

O objetivo deste projeto é desenvolver uma fechadura eletrônica utilizando sensor de RFID visando menor custo de produção e maior aproveitamento dos componentes utilizados. A fechadura deverá manter salvo os usuários e possuir um usuário administrador que pode cadastrar ou remover usuários.

Além disso, o projeto busca incentivar nos participantes a busca por conhecimentos necessários de forma autônoma, sem que essa informação seja passada a eles de forma passiva.

1.2 Justificativa

Essa montagem foi escolhida pelo grupo devido à falta de segurança das fechaduras comuns e alto preço de fechaduras eletrônicas no mercado. Então a busca por materiais de baixo custo para tornar o produto mais acessível para o consumidor final é parte determinante para o sucesso do projeto

1.3 Metodologia

O projeto ocorrerá principalmente em duas etapas: pesquisa e montagem.

Na parte de pesquisa os conhecimentos necessários para a criação da fechadura serão buscados pelos alunos sendo utilizada a ajuda de livros, internet e dos professores. Além disso, será necessário buscar pelos melhores componentes para serem utilizados, para garantir assim o melhor custo-benefício.

Na etapa de montagem serão feitos dois protótipos e uma montagem final. Os protótipos serão feitos para o teste e melhor conhecimento do sensor e do atuador e serão remontados até que funcionem perfeitamente.

- Protótipo 1: Tem como objetivo a verificação do funcionamento do microcontrolador (ATMEGA 328p) aliado a forma de abertura da fechadura (RFID)
- Protótipo 2: O atuador (eletroímã) será adicionado ao protótipo e a fechadura será apresentada.
- Projeto final: A fechadura pronta será apresentada com todas as suas funcionalidades e interfaces.

2 VALORES E CRONOGRAMA

Antes de iniciarmos o projeto buscamos fazer o orçamento total necessário para sua conclusão e, criamos um cronograma a ser seguido, para assim termos prazos de conclusão que incentivem ainda mais o desenvolvimento do trabalho.

2.1 Orçamento

A tabela 1 é uma relação de todos os componentes utilizados e os preços encontrados no mercado. Os produtos foram procurados na internet em sites como AliExpress e Mercado Livre, sempre levando em conta eficiência e custo, para que o produto final tenha o maior custo-benefício.

Tabela 1 – Orçamento

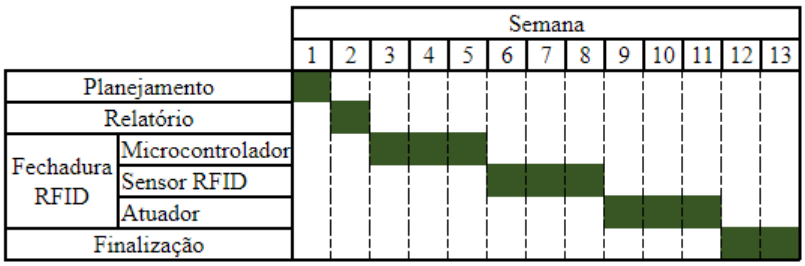
Componente	Valor	Quantidade	Total
ATMEGA 328p	R$ 5,55	1	R$ 5,55
Conector Borne 2 vias	R$ 0,82	12	R$ 9,84
Display LCD	R$ 13,20	1	R$ 13,20
Fonte 12 V 1 A	R$ 5,81	1	R$ 5,81
Módulo RFID	R$ 5,12	1	R$ 5,12
Placa de Fenolite	R$ 1,81	1	R$ 1,81
Soquete 28 pinos	R$ 2,20	4	R$ 8,80
Suporte LED 5mm	R$ 0,33	2	R$ 0,66
Frete	R$ 68,47	1	R$ 68,47
Total			R$ 119,26

Fonte: Elaborada pelos autores

2.2 Cronograma

A tabela 2 indica o planejamento do projeto em semanas, o que é esperado que seja feito e o tempo levado por cada etapa do processo.

Tabela 2 – Cronograma



Fonte: Elaborada pelos autores

3 SENSORES

A fechadura utilizará apenas um tipo de sensor, o sensor de RFID que auxiliará na autenticação.

O termo RFID é a sigla para identificação por radiofrequências (Radio Frequency Identification), ou seja, é uma forma de por meio de ondas de rádio para identificação de algo (ROUSE, 2019).

Um sistema RFID possui 3 componentes: uma antena, um transceptor e um transponder. O transponder (etiqueta) é a identificação em si, cada transponder emite uma frequência diferente. A antena tem a função de receber essa frequência do transponder e repassá-la para o transceptor que converterá essa frequência para um sinal digital, que será tratado por um outro componente, no nosso caso, o ATMEGA328p (CIRIACO, 2019).

O transponder, também chamado de tag RFID, pode ser de dois tipos: ativo ou passivo. Uma tag passiva é aquela que emite um sinal apenas como resposta ao sinal da antena, já as tags ativas emitem seu próprio sinal, mas para isso precisam de uma bateria interna.

3.1 Módulo RFID RC522

O módulo RC522 (Figura 5) que utilizaremos é uma placa que contém a antena e o transceptor. Ele se comunicará com o microcontrolador utilizando o protocolo ISP, por isso precisa ser conectado conforme a tabela 3 (GBUR, 2017).

Figura 5 – Módulo RFID RC522

Fonte: Project Shop

Tabela 3 – Conexões do RC522

Sensor	Conexão
NSS	Pino 10
SCK	Pino 13
MOSI	Pino 11
MISO	Pino 12
IRQ	Não Conecta
GND	GND
RST	Pino 9
VCC	3.3V

Fonte: Elaborada pelos autores

3.2 Teste do sensor

O teste foi feito utilizando além do sensor, uma tela LCD, que foi ligada ao circuito conforme a tabela 4 (COMPONENTS101, 2017). A tela exibiu o texto “Acesso liberado” quando o sensor leu uma frequência aceita, caso a frequência lida tenha sido de um cartão bloqueado a tela exibiu o texto “Bloqueado”, e por fim, no caso de um cartão desconhecido, o texto exibido foi “Acesso negado”.

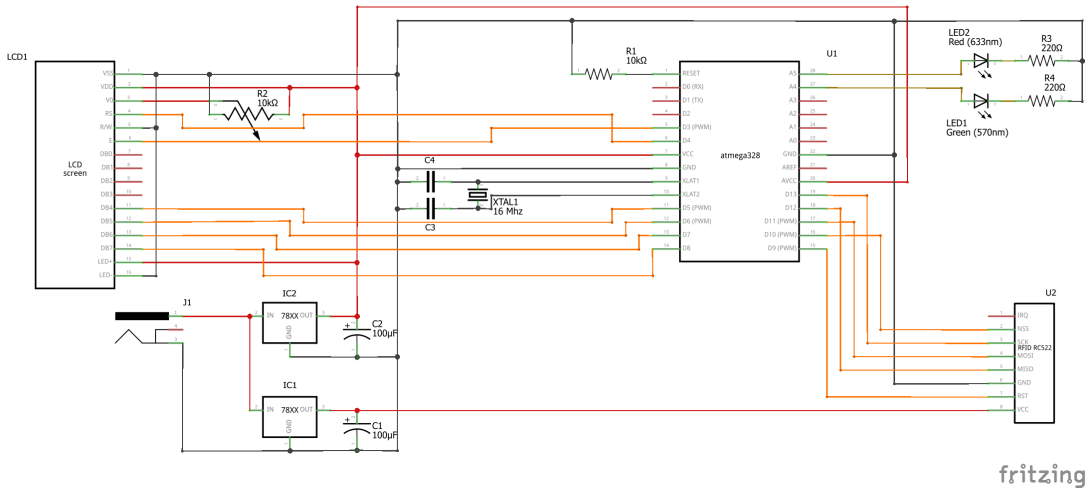
Tabela 4 – Conexões do LCD

LCD	Conexão
VSS	GND
VDD	5V
V0	Potenciometro -> GND
RS	Pino 4
R/W	GND
E	Pino 3
DB0 - DB3	Não Conecta
DB4 - DB7	Pinos 5 - 8
LED+	5V
LED-	GND

Fonte: Elaborada pelos autores

O código utilizado para o teste do sensor está no Apêndice A, e a figura 6 representa o esquema da montagem final para o teste.

Figura 6 – Esquema elétrico do teste dos sensores



Fonte: Elaborado pelos autores

4 ATUADORES

Por se tratar de um módulo de fechadura eletrônica, o projeto apresentará apenas a conexão para que a trava seja colocada. Dessa forma é possível utilizar qualquer tipo de trava, seja utilizando solenoides ou eletroímãs (INTELBRAS, 2019). Ainda assim, o circuito de apoio à trava foi feito pelo grupo.

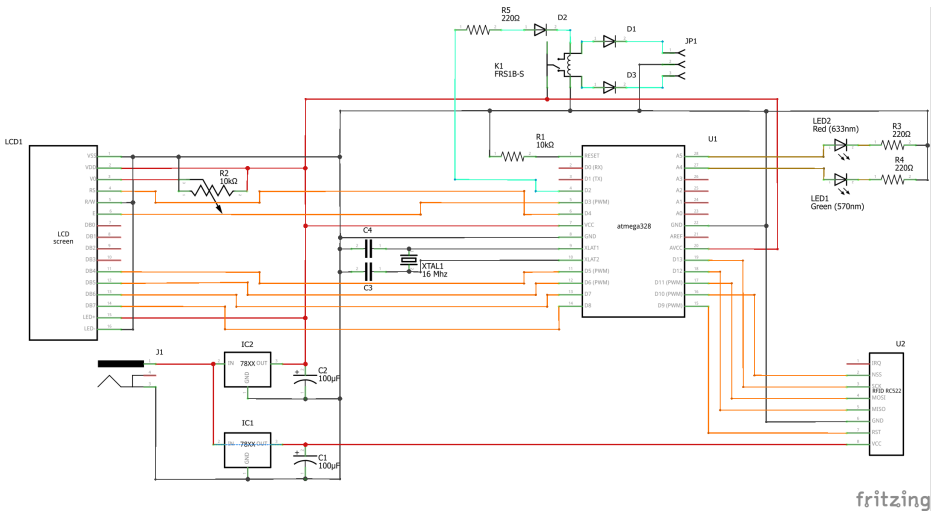
Em geral, há dois tipos de travas, aquelas que estão sempre fechadas e, ao receber energia elétrica se abrem (MADEIRA, 2018), e as que funcionam ao contrário, ficam sempre abertas e se trancam ao receber a energia (DESTERRO, 2018).

Como o circuito auxiliar precisa lidar com os dois casos, utilizaremos um relé, assim travas que necessitam de energia para serem fechadas, devem ser ligadas ao terminal normalmente fechado do relé, e as que se abrem ao receber energia são conectadas ao terminal normalmente aberto.

Para impedir que o relé queime o controlador são necessários um resistor e alguns diodos. O resistor diminui a corrente elétrica que o relé recebe, não permitindo o ATmega enviar mais corrente do que ele suporta. Já os diodos impedem que a corrente induzida pelo relé chegue ao Arduino.

Para finalizar o circuito adicionamos um terminal de três conectores para a trava. Um dos conectores está ligado no GND e os outros dois estão ligados um em cada saída do relé, assim o consumidor pode simplesmente conectar o tipo de trava que deseja que o módulo acione. O circuito finalizado pode ser visto na figura 7 e o código para seu funcionamento foi colocado no Apêndice B.

Figura 7 – Esquema elétrico do teste do atuador



Fonte: Elaborado pelos autores

5 INTEGRAÇÃO

A parte final do projeto trata da criação e administração de múltiplos usuários que têm acesso à fechadura. Para isso, foi preciso de armazenar, de alguma forma, quais as UID's que podem abrir a fechadura e quais são as UID's banidas.

Armazenar todos esses dados na memória do controlador seria um problema. Precisaríamos de uma estrutura de dados dinâmica para cada categoria de cartão (aceito e bloqueado). Essas estruturas ficariam armazenadas na memória RAM do ATMEGA, isso poderia esgotar rápidamente essa memória caso a fechadura seja instalada em um prédio empresarial com vários funcionários por exemplo.

Por isso, a decisão tomada pelo grupo foi adicionar uma memória flash para armazenar esses valores.

Além disso, adicionamos um terceiro diretório para criação de arquivos de log diários contendo informações de todas as tentativas de abertura, seus horários, o cartão utilizado e se a fechadura foi realmente aberta. Para fazer esse log, precisamos de uma forma de informar ao processador o horário, fizemos isso utilizando um módulo RTC

5.1 Memória Flash

Uma memória flash é um tipo específico de EEPROM (sigla em inglês para Memória Somente de Leitura Programável Apagável Elétricamente) muito utilizada em dispositivos portáteis como smartphones e cartões de memória (HAMMERSCHMIDT, 2012).

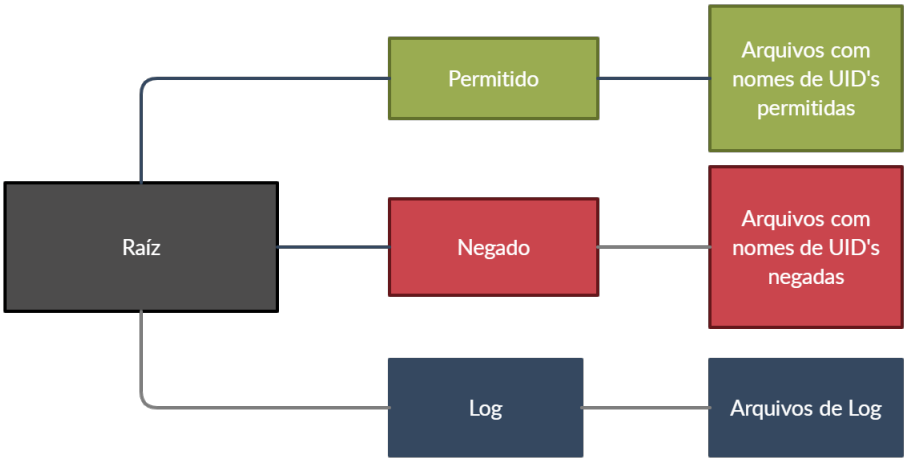
A propriedade importante das memórias Flash que utilizaremos é que elas permitem o armazenamento de dados por muito tempo, mesmo que a alimentação elétrica seja interrompida (ALENCAR, 2012).

Optamos por utilizar como memória Flash um cartão microSD no qual ficarão as UID's dos cartões que podem abrir a porta, dos cartões bloqueados e uma pasta para os arquivos de log.

Quando um arquivo é lido pelo processador, ele é colocado em sua memória RAM, assim, possuindo um arquivo com todas as UID's ainda teriamos o problema da falta de memória do ATMEGA 328p. Portanto, o caminho escolhido foi de criar um diretório no cartão microSD e neste diretório colocar arquivos vazios porém que o nome seja igual à UID de um cartão, desta forma precisamos apenas checar se existe um arquivo cujo nome é igual à UID do cartão aproximado e, se esse arquivo existir, ver se ele está no diretório de cartões bloqueados ou no de cartões permitidos.

Fazendo isso, não lemos nenhum arquivo na memória RAM do processador, e poupamos também armazenamento do microSD, já que esses arquivos uitilizarão no máximo alguns bytes. Assim, a figura 8 representa a árvore de diretórios dentro do cartão microSD.

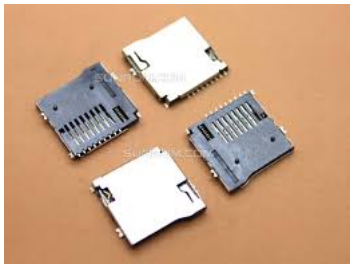
Figura 8 – Árvore de diretórios



Fonte: Elaborado pelos autores

Parga adicionar o microSD ao circuito, precisamos de um socket para este tipo de cartão (Figura 9). O socket será o intermediário entre o cartão e o ATMEGA328p. A comunicação entre este socket também é feita utilizando o protocolo ISP, assim, é necessário conectá-lo utilizando quase o mesmo esquema utilizado para o sensor. A tabela 5 mostra as conexões necessáiras.

Figura 9 – Socket para cartão microSD



Fonte: Sunrom

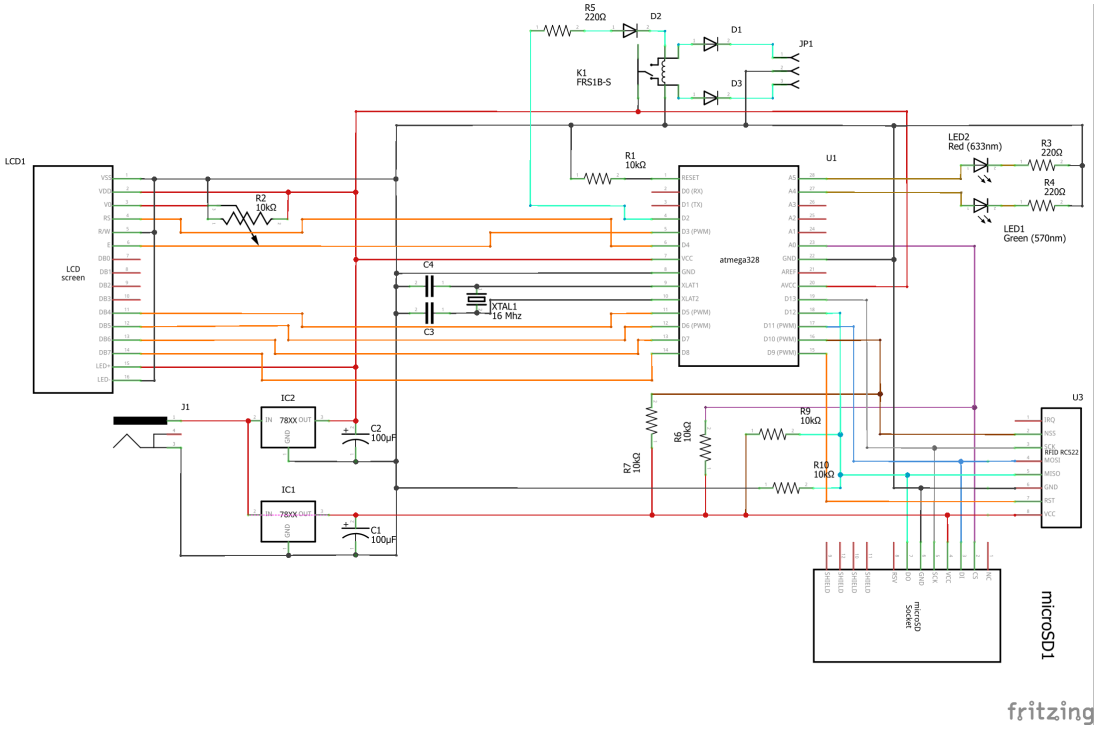
Tabela 5 – Conexões do socket

Socket microSD	Conexão
NC	Não Conecta
CS	A0
D1	Pino 11
VCC	3.3B
SCK	Pino 13
GND	GND
D0	Pino 12
RSV	Não Conecta

Fonte: Elaborada pelos autores

Como precisamos utilizar alguns pinos do arduino tanto para o sensor quanto para o socket, precisaremos ligar os dois em paralelo, além disso, precisamos adicionar resistores de pull-up no CS do socket e no NSS do sensor. Pro fim, é necessário ligar resistores de $10K\Omega$ no fio de MISO e D0, conectando-os no GND e 3.3V (PAUL, 2014). Assim, a figura 10 mostra o circuito com o microSD.

Figura 10 – Esquema com cartão microSD



Fonte: Elaborada pelos autores

5.2 Módulo RTC

Um módulo RTC (Real-Time Clock) é um componente eletrônico capaz de armazenar informações de tempo. No nosso projeto optamos por utilizar o RTCDS3231.

Esse módulo se comunica com o ATMEGA328p por meio do protocolo I2C, assim, precisamos de apenas 2 pinos analógicos, além de sua alimentação (MALLARI, 2020). Suas conexões estão na tabela 6.

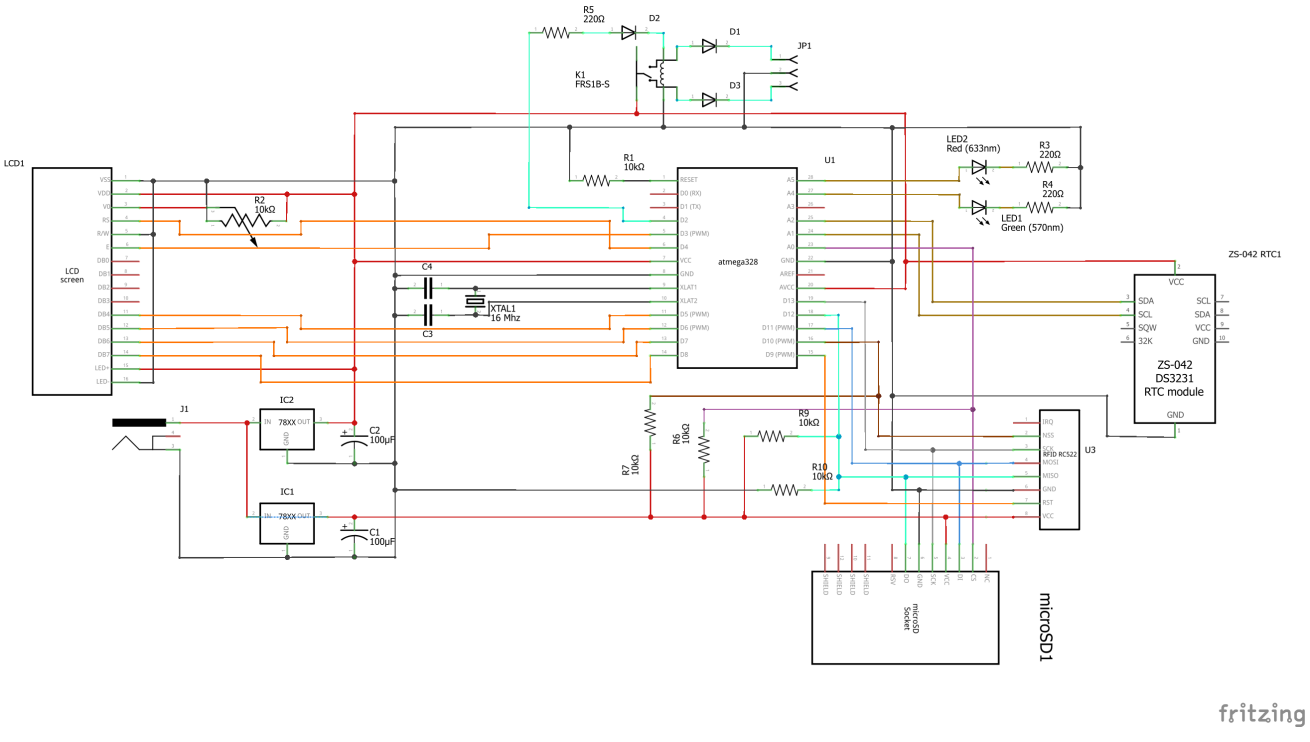
Tabela 6 – Conexões do módulo RTC DS3231

DS3231	Conexão
SDA	A2
SCL	A1
SQW	Não Conecta
32K	Não Conecta
VCC	5V
GND	GND

Fonte: Elaborada pelos autores

Colocando este módulo no circuito, chegamos ao esquema da figura 11. Além disso, o apêndice C contém o código finalizado.

Figura 11 – Esquema finalizado do projeto



Fonte: Produzida pelos autores.

6 CONCLUSÃO

O projeto pôde ser concluído dentro do prazo, porém com algumas limitações. Uma delas, foi que não conseguimos encontrar uma forma simples e eficiente de apagar arquivos de log muito antigos. Isso pode fazer com que a capacidade do microSD se esgote. Uma forma de impedir isso seria caso o consumidor final, de tempos em tempos, removesse o microSD, fizesse um backup desses arquivos e os apagasse. Isso é uma grande limitação, já que a solução do problema depende do consumidor final, que muitas vezes não sabe fazer isso de forma correta.

Além disso ocorreu uma mudançanos componentes utilizados. Inicialmente havíamos montado uma lista de compras, porém, no decorrer do projeto alguns desses componentes não foram utilizados, e outros precisaram ser comprados. No final, a lista de componentes necessários para a criação do projeto está na tabela 7.

Tabela 7 – Componentes utilizados

Componente	Quantidade
ATMEGA 32P	1
Display LCD	1
Módulo RFID RC522	1
Socket microSd	1
Power Jack DC	1
Regulador de tensão 5 V	1
Regulador de tensão 3.3V	1
Capacitor eletrolítico 100µF	2
Capacitor cerâmico 22µF	2
Potenciometro 10KΩ	1
Cristal 16MHz	1
Diodo 1n4007	3
Resistor 10kΩ	5
Resistor 220Ω	2

Fonte: Produzida pelos autores

REFERÊNCIAS

ALENCAR, F. **Entenda como funcionam as memórias Flash, o coração dos seus eletrônicos**. 2012. Disponível em: <https://www.guiadopc.com.br/artigos/22397/entenda-como-funciona-memorias-flash-coracao-dos-seus-eletronicos.html>. Citado na página 12.

CANABARRO, A. **Quem inventou a fechadura?** 2019. Disponível em: <https://www.tricurioso.com/2019/01/22/quem-inventou-a-fechadura/>. Acesso em: 23 de fev. de 2020. Citado na página 5.

CIRIACO, D. **Como funciona a RFID?** 2019. Disponível em: <https://www.tecmundo.com.br/tendencias/2601-como-funciona-a-rfid-.htm>. Acesso em: 06 de set. de 2020. Citado na página 9.

COMPONENTS101. **16X2 LCD Module**. 2017. Disponível em: <https://components101.com/16x2-lcd-pinout-datasheet>. Acesso em: 06 de set. de 2020. Citado na página 10.

CORDEIRO, T. **Como surgiu a chave?** 2018. Disponível em: <http://www.superabril.com.br/mundo-estranho/>. Acesso em: 23 de fev. de 2020. Citado na página 5.

DESTERRO. **Fechadura Eletromagnética Trava Eletroímã**. 2018. Disponível em: <https://www.desterroeletricidade.com.br/blog/sistema-de-seguranca/fechadura-eletromagnetica-trava-eletroima/>. Acesso em: 23 de set. de 2020. Citado na página 11.

GBUR, F. **Módulo RFID RC522 Mifare com Arduino**. 2017. Disponível em: <https://portal.vidadesilicio.com.br/modulo-rfid-rc522-mifare/>. Acesso em: 06 de set. de 2020. Citado na página 9.

HAMMERSCHMIDT, R. **O que é memória Flash?** 2012. Disponível em: <https://www.tecmundo.com.br/hardware/198-o-que-e-memoria-flash-.htm>. Acesso em: 07 de out. de 2020. Citado na página 12.

INTELBRAS. **Conheça os tipos de fechaduras para condomínio e suas aplicações.** 2019. Disponível em: <http://blog.intelbras.com.br/conheca-os-tipos-de-fechaduras-paracondominios-e-suas-aplicacoes/>. Acesso em: 23 de set. de 2020. Citado na página 11.

MADEIRA, D. **Trava elétrica solenoide com Arduino.** 2018. Disponível em: <https://portal.vidadesilicio.com.br/trava-eletrica-solenoide/#A_trava_eletrica_solenoide>. Acesso em: 23 de set. de 2020. Citado na página 11.

MALLARI, J. **HOW TO USE A REAL-TIME CLOCK MODULE WITH THE ARDUINO**. 2020. Disponível em: <https://www.circuitbasics.com/how-to-use-a-real-time-clock-module-with-the-arduino/>. Acesso em: 09 de out. de 2020. Citado na página 15.

PAUL. **Better SPI Bus Design in 3 Steps**. 2014. Disponível em: <https://dorkbotpdx.org/blog/paul/better_spi_bus_design_in_3_steps/>. Citado na página 14.

PIRES, C. **Fechaduras Eletrônicas ou Elétricas - Como Escolher?** 2020. Disponível em: <https://reprizzo.com.br/2018/12/17/historia-das-chaves-e-fechaduras/>. Acesso em: 23 de fev. de 2020. Citado na página 6.

REPRIZZO. **História das chaves e fechaduras.** 2018. Disponível em: <https://reprizzo.com.br/2018/12/17/historia-das-chaves-e-fechaduras/>. Acesso em: 23 de fev. de 2020. Citado na página 5.

ROUSE, M. **RFID (radio frequency identification)**. 2019. Disponível em: <https://internetofthingsagenda.techtarget.com/definition/RFID-radio-frequencyidentification>. Citado na página 9.

APÊNDICES

A CÓDIGO PARA TESTE DOS SENSORES

```
/*Pinagem
  *LCD RS - pino D4
  *LCD EN - pino D4
  *LCD D4 - pino D4
  *LCD D5 - pino D4
  *LCD D6 - pino D4
  *LCD D7 - pino D4
  *
  *RFID NSS - pino D4
  *RFID SCK - pino D4
  *RFID MOSI - pino D4
  *RFID MISO - pino D4
  *
  *LED Vermelho - pino D4
  *LED Verde - pino D4
*
*/

#include <SPI.h>            // Comunicacao com o modulo RFID
#include <MFRC522.h>        // Biblioteca do modulo RFID
#include <LiquidCrystal.h>  // Biblioteca da tela

#define SS_PIN 10
#define RST_PIN 9

// Instanciando o modulo RFID e LCD
MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal lcd(4,3,5,6,7,8);

String UID = "";

void setup() {
  SPI.begin();        // Inicia comunicacao SPI
  mfrc522.PCD_Init(); // Inicia o modulo RFID
  lcd.begin(16,2);    // Inicializa o display LCD
  boot();             // Rotina de texto inicial
}

void boot() {
  lcd.clear();
  lcd.print("Aproxime o seu");
  lcd.setCursor(0, 1);
  lcd.print("cartao no leitor");
```

```
44 }
45
46 void ler_cartao() {
47   // Procurar cartao
48   if (!mfrc522.PICC_IsNewCardPresent()) {
49     return;
50   }
51
52   // Ler dados do cartao
53   if (!mfrc522.PICC_ReadCardSerial()) {
54     return;
55   }
56
57   // Receber UID do cartao
58   for (byte i = 0; i < mfrc522.uid.size; i++) {
59     UID.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
60     UID.concat(String(mfrc522.uid.uidByte[i], HEX));
61   }
62 }
63
64 void resposta (){
65   UID.toUpperCase();
66
67   //UID esperada do cartao liberado
68   if (UID.substring(1) == "FF FF FF FF") {
69     lcd.clear();
70     lcd.setCursor(0, 0);
71     lcd.print("Bem vindo");
72     lcd.setCursor(0, 1);
73     lcd.print("Acesso liberado!");
74  }
75
76   //UID esperada do cartao bloqueado
77   else if (UID.substring(1) == "00 00 00 00") {
78     lcd.clear();
79     lcd.setCursor(0, 0);
80     lcd.print("Usuario");
81     lcd.setCursor(0, 1);
82     lcd.print("Bloqueado");
83   }
84   else if (UID.substring(1) != ""){
85     lcd.clear();
86     lcd.setCursor(0, 0);
87     lcd.print("Acesso negado");
88   }
89 }
90
```

```
91 void loop() {
92   ler_cartao();
93   resposta();
94 }
```

B CÓDIGO PARA TESTE DOS SENSORES

```
/*Pinagem
  *LCD RS - pino D4
  *LCD EN - pino D4
  *LCD D4 - pino D4
  *LCD D5 - pino D4
  *LCD D6 - pino D4
  *LCD D7 - pino D4
  *
  *RFID NSS - pino D4
  *RFID SCK - pino D4
  *RFID MOSI - pino D4
  *RFID MISO - pino D4
  *
  *LED Vermelho - pino D4
  *LED Verde - pino D4
  *
  * Rele - pino 12
*
*/

#include <SPI.h>            // Comunicacao com o modulo RFID
#include <MFRC522.h>        // Biblioteca do modulo RFID
#include <LiquidCrystal.h>  // Biblioteca da tela

#define SS_PIN 10
#define RST_PIN 9
#define atuador 2

int intervalo = 5000; // Tempo que a trava ficara aberta em ms

// Instanciando o modulo RFID e LCD
MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal lcd(4,3,5,6,7,8);

String UID = "";

void setup() {
  SPI.begin();         // Inicia comunicacao SPI
  mfrc522.PCD_Init(); // Inicia o modulo RFID
  lcd.begin(16,2);    // Inicializa o display LCD
  boot();             // Rotina de texto inicial
}

```

```
44 void boot() {
45   lcd.clear();
46   lcd.print("Aproxime o seu");
47   lcd.setCursor(0, 1);
48   lcd.print("cartao no leitor");
49
50   // Garante que o pino do atuador seja iniciado sem energia
51   digitalWrite(atuador, LOW);
52 }
53
54 void abrir() {
55   // Abre a fechadura e depois a fecha
56   digitalWrite(atuador, HIGH);
57   delay(intervalo);
58   digitalWrite(atuador, LOW);
59 }
60
61 void ler_cartao() {
62   // Procurar cartao
63   if (!mfrc522.PICC_IsNewCardPresent()) {
64     return;
65   }
66
67   // Ler dados do cartao
68   if (!mfrc522.PICC_ReadCardSerial()) {
69     return;
70   }
71
72   // Receber UID do cartao
73   for (byte i = 0; i < mfrc522.uid.size; i++) {
74     UID.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
75     UID.concat(String(mfrc522.uid.uidByte[i], HEX));
76   }
77 }
78
79 void resposta (){
80   UID.toUpperCase();
81
82   //UID esperada do cartao liberado
83   if (UID.substring(1) == "FF FF FF FF") {
84     lcd.clear();
85     lcd.setCursor(0, 0);
86     lcd.print("Bem vindo");
87     lcd.setCursor(0, 1);
88     lcd.print("Acesso liberado!");
89  }
90
```

```
91    //UID esperada do cartao bloqueado
92    else if (UID.substring(1) == "00 00 00 00") {
93      lcd.clear();
94      lcd.setCursor(0, 0);
95      lcd.print("Usuario");
96      lcd.setCursor(0, 1);
97      lcd.print("Bloqueado");
98    }
99    else if (UID.substring(1) != ""){
100     lcd.clear();
101     lcd.setCursor(0, 0);
102     lcd.print("Acesso negado");
103   }
104 }
105
106 void loop() {
107   ler_cartao();
108   resposta();
109 }
```

C CÓDIGO FINALIZADO

```
/*Pinagem
   LCD RS - pino D4
   LCD EN - pino D4
   LCD D4 - pino D4
   LCD D5 - pino D4
   LCD D6 - pino D4
   LCD D7 - pino D4

   RFID CS - pino D4
   SD CS - pino A0
   SCK - pino D13
   MOSI - pino D11
   MISO - pino D12

   LED Vermelho - pino D4
   LED Verde - pino D4

    Rele - pino 12

*/

#include <SPI.h>            // Comunicacao com o modulo RFID
#include <MFRC522.h>        // Biblioteca do modulo RFID
#include <LiquidCrystal.h>  // Biblioteca da tela
#include <SD.h>             // Biblioteca cartao SD
#include <Wire.h>           // Biblioteca do RTC
#include <RTClib.h>         // Biblioteca do RTC


#define SS_PIN 10
#define RST_PIN 9
#define atuador 2
#define MICRO_SD_PIN A0
#define VERMELHO A5
#define VERDE A4

int intervalo = 5000; // Tempo que a trava ficara aberta em ms

// Instanciando modulos
MFRC522 mfrc522(SS_PIN, RST_PIN);
LiquidCrystal lcd(4, 3, 5, 6, 7, 8);
RTC_DS3231 rtc;

```

```
44 String admin = "";
45 String UID = "";
46 bool adicionar = false;
47
48 void setup() {
49   SPI.begin();          // Inicia comunicacao SPI
50   mfrc522.PCD_Init();   // Inicia o modulo RFID
51   lcd.begin(16, 2);     // Inicializa o display LCD
52   rtc.begin();          // Inicializa modulo RTC
53   SD.begin();           // Inicializa modulo microSD
54   boot();               // Rotina de texto inicial
55 }
56
57 void boot() {
58   lcd.clear();
59   lcd.print("Aproxime o seu");
60   lcd.setCursor(0, 1);
61   lcd.print("cartao no leitor");
62
63   //Inicializa conexoes SPI
64   pinMode(SS_PIN, OUTPUT);
65   pinMode(MICRO_SD_PIN, OUTPUT);
66
67   // Garante que o pino do atuador seja iniciado sem energia
68   digitalWrite(atuador, LOW);
69
70   // Cria diretorios se necessario
71   digitalWrite(MICRO_SD_PIN, LOW);
72   digitalWrite(SS_PIN, HIGH);
73   if (!SD.exists("Permitido")) {
74     SD.mkdir("Permitido");
75   }
76   if (!SD.exists("Negado")) {
77     SD.mkdir("Negado");
78   }
79   if (!SD.exists("Log")) {
80     SD.mkdir("Log");
81   }
82
83   // Garante que os pinos de chip select sejam inicializados
       corretamente
84   digitalWrite(SS_PIN, LOW);
85   digitalWrite(MICRO_SD_PIN, HIGH);
86 }
87
88 void abrir() {
89   // Abre a fechadura e depois a fecha
```

```
90    digitalWrite(atuador, HIGH);
91    delay(intervalo);
92    digitalWrite(atuador, LOW);
93  }
94
95  void ler_cartao() {
96    // Procurar cartao
97    if (!mfrc522.PICC_IsNewCardPresent()) {
98      return;
99    }
100
101   // Ler dados do cartao
102   if (!mfrc522.PICC_ReadCardSerial()) {
103     return;
104   }
105
106   // Receber UID do cartao
107   for (byte i = 0; i < mfrc522.uid.size; i++) {
108     UID.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
109     UID.concat(String(mfrc522.uid.uidByte[i], HEX));
110   }
111   UID.toUpperCase();
112
113   if (admin == "") {
114     admin = UID;
115     return;
116   }
117
118   if (UID == admin) {
119     adicionar = true;
120   } else if (adicionar) {
121     SD.open("/Permitido/" + UID, FILE_WRITE);
122   }
123 }
124
125 void resposta () {
126   digitalWrite(SS_PIN, HIGH);
127   digitalWrite(MICRO_SD_PIN, LOW);
128
129   //UID esperada do cartao liberado
130   if (SD.exists("Permitido/" + UID)) {
131     lcd.clear();
132     lcd.setCursor(0, 0);
133     lcd.print("Bem vindo");
134     lcd.setCursor(0, 1);
135     lcd.print("Acesso liberado!");
136     digitalWrite(VERDE, HIGH);
```

```
137       abrir();
138       digitalWrite(VERDE, LOW);
139
140       //Adiciona ao log do dia
141       escreve_log(true);
142       return;
143     }
144
145     //UID esperada do cartao bloqueado
146     else if (SD.exists("Negado/" + UID)) {
147       lcd.clear();
148       lcd.setCursor(0, 0);
149       lcd.print("Usuario");
150       lcd.setCursor(0, 1);
151       lcd.print("Bloqueado");
152       digitalWrite(VERMELHO, HIGH);
153       delay(intervalo);
154       digitalWrite(VERMELHO, LOW);
155     }
156     else {
157       lcd.clear();
158       lcd.setCursor(0, 0);
159       lcd.print("Acesso negado");
160       digitalWrite(VERMELHO, HIGH);
161       delay(intervalo);
162       digitalWrite(VERMELHO, LOW);
163     }
164     // Adicionar ao log
165     escreve_log(false);
166     digitalWrite(SS_PIN, LOW);
167     digitalWrite(MICRO_SD_PIN, HIGH);
168   }
169
170   void adicionar_cartao() {
171     digitalWrite(SS_PIN, HIGH);
172     digitalWrite(MICRO_SD_PIN, LOW);
173
174     File arquivo_UID;
175
176     if (SD.exists("Permitido/" + UID)) {
177       SD.remove("Permitido/" + UID);
178       arquivo_UID = SD.open("/Permitido/" + UID, FILE_WRITE);
179       arquivo_UID.close();
180     }
181     else if (SD.exists("Negado/" + UID)) {
182       SD.remove("Negado/" + UID);
183       arquivo_UID = SD.open("/Permitido/" + UID, FILE_WRITE);
```

```
184       arquivo_UID.close();
185     } else {
186       arquivo_UID = SD.open("/Permitido/" + UID, FILE_WRITE);
187       arquivo_UID.close();
188     }
189
190     digitalWrite(SS_PIN, LOW);
191     digitalWrite(MICRO_SD_PIN, HIGH);
192   }
193
194
195   void escreve_log(bool aberto) {
196     // Abre ou cria o arquivo de log do dia
197     DateTime agora = rtc.now();
198     File log = SD.open("/Log/" + agora.day() + agora.month(), FILE_WRITE);
199     log.seek(log.size());
200
201     log.write("HORA: ");
202     log.write(agora.hour());
203     log.write(":");
204     log.write(agora.minute());
205     log.write(":");
206     log.write(agora.second());
207     log.write("    UID:");
208     log.print(UID);
209     log.write("Aberto: ");
210     if (aberto) {
211       log.write("SIM");
212     } else {
213       log.write("NAO");
214     }
215     log.println();
216   }
217
218   void loop() {
219     ler_cartao();
220     resposta();
221   }
```