

# Desenvolvimento de sistema de controle de presença de alunos usando *smart cards*

## **Resumo**

*A tecnologia de RFID - Radio Frequency Identification (Identificação por rádio-frequência) é amplamente utilizada em diversos setores que necessitam de identificação de usuários ou produtos, como em sistemas de inventário, rastreamento de objetos, controle de acesso, controle de acervo em bibliotecas, passaportes, telemetria, entre outros. Neste projeto, desenvolvemos o protótipo de um sistema de controle digital de presença de alunos às aulas de graduação da UFABC, que tem como objetivo eliminar a necessidade de listas de presença assinadas pelos alunos e sua validação pelos professores. O sistema é composto por um módulo de processamento e armazenamento de dados, um módulo leitor RFID, e um relógio de tempo real (RTC). Os alunos são identificados por meio dos smart cards fornecidos pela UFABC, sendo que o nosso sistema registra o número identificador único (UID) do cartão e o horário da leitura do cartão.*

## **Introdução**

A tecnologia de RFID - Radio Frequency Identification (Identificação por radio frequência) é amplamente utilizada em diversos setores que necessitam de identificação de usuários ou produtos, como em sistemas de inventário, rastreamento de produtos, controle de acesso, passaportes, bibliotecas, telemetria, entre outros. O sistema de identificação RFID usa

basicamente dois componentes: um transceptor de radiofrequência, também conhecido como leitor RFID, e uma tag (etiqueta) de identificação, geralmente embutida em um smart card no formato de um cartão plástico comum, etiqueta autocolante ou chaveiro. O leitor RFID emite uma onda eletromagnética que, ao mesmo tempo que fornece energia para a tag, transmite-lhe informações; a tag, por sua vez, responde ao leitor com os dados nela armazenados (no caso mais simples, apenas um número identificador).

Neste projeto, desenvolvemos o protótipo de um sistema de controle de presença dos alunos às aulas de graduação da UFABC. O sistema é composto por um módulo de processamento e armazenamento de dados, um relógio de tempo real (*RTC*) e um módulo leitor *RFID*, que utiliza os *smart cards* fornecidos pela UFABC como forma automática de identificação de alunos, com o objetivo de eliminar a necessidade de listas de presença assinadas pelos alunos e sua posterior verificação manual pelos professores.

No protótipo, o módulo de processamento usado é um Arduino Uno [1], que possui um microcontrolador programável, facilmente expansível por meio de módulos compatíveis (*shields*), disponíveis a preço acessível no mercado de componentes eletrônicos. Os módulos utilizados neste projeto foram:

- para armazenamento de informações, uma memória EEPROM I2C [14];
- para comunicação com os *smart cards*, um *shield* leitor *RFID* [3] compatível com o padrão *MIFARE* [13];
- para manutenção da data e hora reais, um módulo *RTC* [15]; e
- para exibição de informações ao usuário, um *display* de cristal líquido com módulo controlador [16]

## **Sobre a plataforma *Arduino***

O *Arduino* é uma plataforma de prototipagem para circuitos eletrônicos digitais que possui um microcontrolador programável e diversos conectores de expansão. O hardware e o software do *Arduino* são projetados para facilitar a programação e a interação com dispositivos eletrônicos, ao mesmo tempo em que possibilitam uma grande capacidade de expansão e controle dos dispositivos.

Tanto o projeto dos circuitos eletrônicos, quanto o código fonte são fornecidos de maneira livre e abertos, com licenças permissivas (*open source/open hardware*). Devida à flexibilidade e liberdade de uso, a plataforma *Arduino* é muito empregada por profissionais tanto da área de tecnologia, quanto de outras áreas onde é necessário interagir com dispositivos eletroeletrônicos. Por esses motivos, é possível encontrar uma grande quantidade de material de aprendizado e manuais de uso de boa qualidade, os quais serão necessários neste projeto.

## **Sobre *RFID***

A tecnologia *RFID* é uma tecnologia de sistema de identificação que utiliza ondas eletromagnéticas para transmitir e receber informações, isso elimina a necessidade de contato direto e linha de visão entre a etiqueta ou cartão de identificação (tag) e o leitor *RFID*. Entre os padrões de identificação por radiofrequência existente, usaremos um shield de leitura *RFID* que corresponde ao padrão de leitura e gravação *MIFARE*, esse padrão será utilizado porque foi adotado para identificação dos smart cards da UFABC.

Os smart cards portadores da tecnologia *MIFARE* tem uma grande aplicação dentro do contexto de identificações rápidas, acessíveis economicamente, e segura [13]. Nas grandes cidades metropolitanas os *smart cards* são utilizados para substituir os tradicionais bilhetes e passes de transporte, isso porque a identificação baseada na tecnologia, não requer uma

validação que necessite de impressão de dados, e permite gravar uma grande quantidade de informações do utilizador, sem necessidade de um tempo grande de leitura. Essa tecnologia é muito aplicada também em portarias de empresas, prédios inteligentes, parquímetros, e estações de serviços dos mais variados tipos.

A integração do leitor *MIFARE* com o Arduino será feita por etapas, a primeira etapa será a montagem dos equipamentos e a segunda etapa será a programação do Arduino:

- Montagem: A montagem do leitor *MIFARE* requer atenção do desenvolvedor, para que seja evitados erros de ligações, esses erros de ligações podem causar funcionamento incorreto, ou defeitos nos equipamentos, o que caracteriza essa como uma das principais etapas do projeto.

- Programação: O desenvolvimento de uma programação que possibilite o funcionamento adequado do aparelho e também que seja pratica e funcional é essencial para utilização devida dos recursos disponíveis pelo Arduino e o leitor de cartões.

O *shield* de leitura utilizado será o *RFID/NFC Shield* [3], sua antena foi projetada para trabalhar a uma distância de 1 cm, e atende aos padrões da *ISO 14443A* [11], e tem como base principal de seu funcionamento o circuito integrado *PN525* (CI *PN525*), seu desenvolvimento foi baseado também nos padrões normatizado pela *ISO/IEC 18092* [12].

O CI *PN525* [10] é um modulo de transmissão que realiza contatos e comunicação á 13.56MHz. O CI *PN525* também inclui em sua estrutura as funções de um microcontrolador baseada na série de microcontroladores *80C51*. O circuito integrado também suporta alguns módulos de operação diferentes, dentre eles os que iremos utilizar são o modo de Leitura/Escrita (*R/W – Read/Write*) para esquemas suportados pela *ISO 14443A*, e o modo único de *R/W* suportado pela *ISO 14443B*, todas no padrão *MIFARE*.

O módulo de transmissão do shield utiliza conceitos de modulação (*MUX*) e demodulação (*DEMUX*), integráveis para diferentes tipos de métodos passivos de contato e comunicação, os seus protocolos de transmissão são utilizados na frequência de operação do CI *PN525* (13.26 MHz).

### 1.3. Sobre o sistema de controle de frequência proposto

Propomos um sistema de controle de frequência (*hardware* e *software*) que será implantado para que o professor dentro de sala de aula tenha um controle mais preciso da presença dos alunos em sala de aula.

O sistema de controle de frequência identificará o aluno através do *smart card* que lhe é fornecido pela UFABC, utilizando de algoritmos para a identificação do usuário, após a identificação, as informações geradas serão armazenadas, arquivos de planilhas eletrônicas ou banco de dados dentro de uma memória *EEPROM I2C*.

Há várias vantagens no uso de controle de frequência de alunos por *smart cards*: praticidade no registro das presenças, tanto para os alunos quanto para o professor; menor ocorrência de fraudes (desde que a instituição tenha uma boa política de controle dos *smart cards*); transferência automática dos registros de frequência para arquivos digitais, o que também elimina os erros de transição.

O sistema de integração proposto tem vários tipos de aplicações, por exemplo, na indústria ele é utilizado para identificação e rastreamento de peças e dispositivos [4]. O sistema *RFID* é muito utilizado por que sua praticidade de integração e programação é mais usual, tem uma complexidade de inferior e maior nível de segurança da informação que é transmitida, evitando erros sistemáticos onde outros tipos de rastreamento (mesmo os automatizados), teriam [8].

Um problema constante, em um dos sistemas de identificação atualmente mais utilizados (identificação por meio de código de barras), é o fator contato visual direto necessário entre o identificador e o dispositivo de identificação, essa necessidade de contato direto diminui as vantagens de integração do processo de maneira geral, e também causa diversos tipos de erros durante o processo de identificação ou aquisição dos dados. Os erros gerados nos casos gerais ocorrem na fase intermediária do processo, esses erros de leitura usualmente se aplicam á ambientes rústicos e empoeirados (grande parte da indústria de processos) [4].

A leitura ótica é utilizada em ambientes físicos que agredem de maneira menos direta, os equipamentos de leitura e que causem menor possibilidade de erros no sistema como um todo.

Dentro de ambientes menos rústicos, há uma expectativa de sucesso na leitura e na aquisição de dados maior, isso acaba diminui a possibilidade de erros e tornando viável, especificamente em alguns casos.

## Objetivos e Metodologia

Nosso objetivo final, neste projeto, foi desenvolver um protótipo do sistema (*hardware* e *software*) que possibilite o controle de frequência de alunos por meio de *smart cards*, para ser utilizado pelos professores e alunos da UFABC.

O objetivo de fins práticos do projeto foi evitar situações de falhas, em que o controle de frequência atual apresenta, onde no sistema atual é necessário que o aluno se identifique por meio de assinatura em lista de presença, isso muitas vezes gera fraudes no sistema, diminui a velocidade de execução da tarefa, e causa muitos erros quanto à digitalização e confiabilidade da informação gerada pelo usuário.

Como objetivos intermediários tivemos a compreensão do hardware e a arquitetura do microcontrolador utilizado, e desenvolvemos a programação para o microcontrolador que

possibilitou o registro de presença dos alunos disponibilizando o arquivo para leitura em um *EEPROM I2C*.

Utilizamos a plataforma *Arduino UNO (Atmel 328p)*, um shield de leitura (*RFID*) de *smart cards*, uma memória *EEPROM*, um *RTC* e uma tela de *LCD* para que se tenha a confirmação visual dos dados recebidos pelo microcontrolador.

O projeto desenvolvido baseou-se na aplicação de conceitos básicos de circuitos digitais e de programação, por isso a metodologia consistiu de:

- Compreensão das soluções adotadas para comunicação com *RFID* e de projeto de circuitos com Arduino.
  - Atividades e métodos: (a) Coleta de material de pesquisa necessário através de pesquisa bibliográfica e internet; (b) Estudo dirigido dos materiais em conjunto com o orientador; (c) Projeto de um protótipo com Arduino e *shields* associados.
- Implementação do sistema de controle de presença
  - Atividades e métodos: (a) Programação do controlador Arduino; (b) Testes para ajuste de funcionamento.
- Sistematização dos resultados
  - Atividades e métodos: Elaboração do relatório parcial e do relatório final, com registro documental do desenvolvimento e do produto gerado.
- Divulgação dos resultados
  - Atividades e métodos: (a) Elaboração de pôster e artigo para divulgação; (b) Participação no Congresso de Iniciação Científica da UFABC e eventos relacionados.

# Resultados

## Hardware

Construímos um protótipo usando a plataforma Arduino, com um *shield* que permite a leitura de cartões Mifare, além de outros periféricos, conforme demonstrado na Figura 1.



Figura 1: Protótipo construído.

Na Figura 2, apresentamos o diagrama em blocos do circuito. Note que a memória EEPROM (AT24C256) e o relógio de tempo real (DS3231) compartilham o mesmo barramento I2C, enquanto que o leitor de cartões é conectado ao microcontrolador por meio de seu barramento SPI. O display é ligado a 6 pinos de entrada/saída digitais, enquanto que o teclado é ligado a um circuito formado por divisores resistivos e é ligado a 3 entradas analógicas para economizar pinos de entrada do microcontrolador.



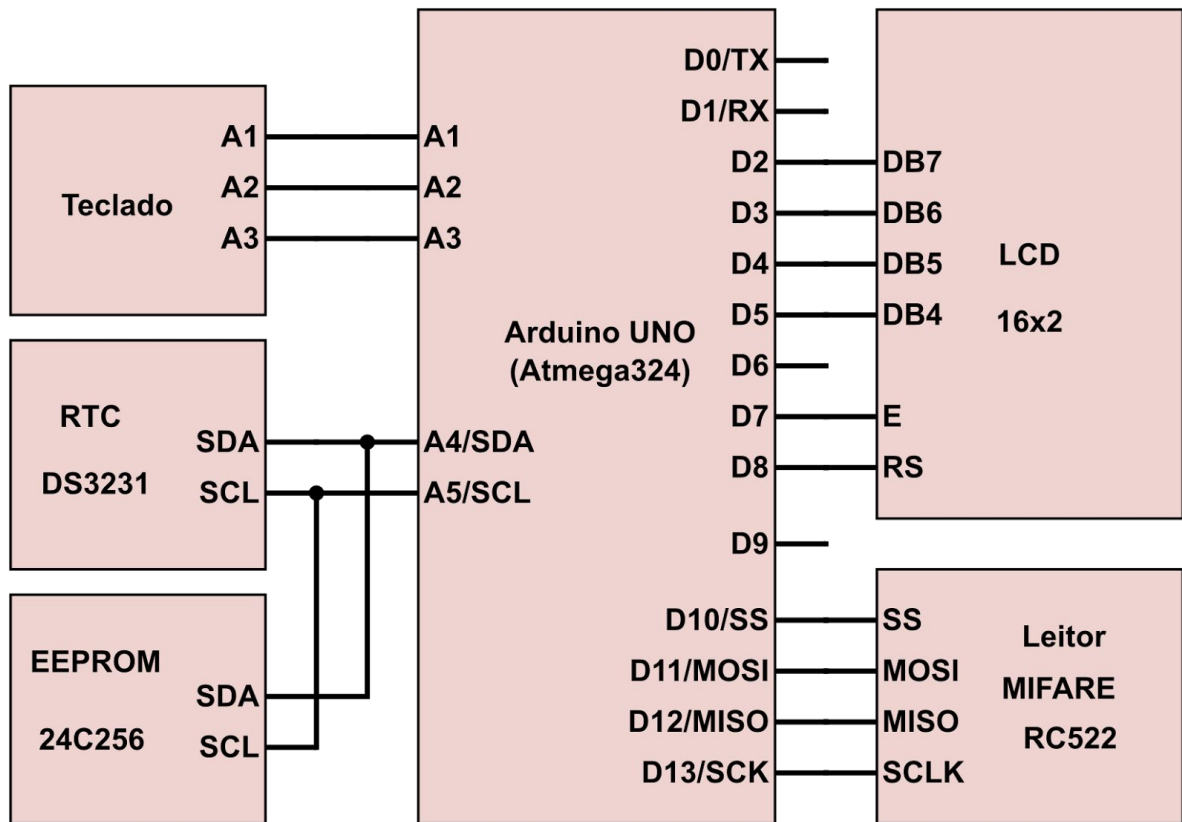


Figura 2: Diagrama em blocos do protótipo.

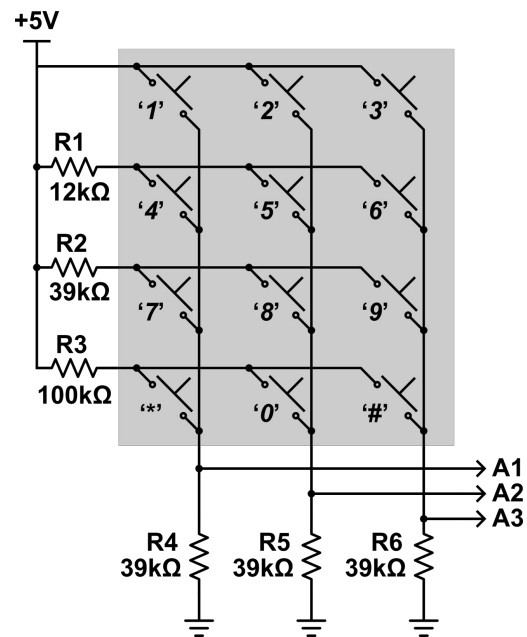


Figura 3: Circuito do teclado matricial

## Software

Foram criadas quatro classes utilizando a linguagem de programação C++, para abstrair o acesso ao hardware. As classes criadas foram nomeadas de *MatrixKeyboard*, *AT24C256*, *TConfig* e *TDBFS*.

A classe *MatrixKeyboard* abstrai o acesso ao hardware de um teclado matricial com estrutura diferenciada aos teclados matriciais existentes. A configuração de *hardware* do teclado utilizado prevê o uso de três portas de leitura analógica do microcontrolador, em contraste a sete portas (I/O) digitais (podendo ser quatro para leitura de dados e três para escrita de dados ou vice-versa) utilizadas em outras configurações mais comuns de hardware para teclados deste tipo (4x3). A classe faz a leitura dos dados pelas portas analógicas e faz referencia ao botão que fora apertado.

A classe *AT24C256* abstrai o acesso à memória EEPROM AT24C256, apresentando métodos para a leitura e escrita de *bytes* e blocos de *bytes* de tamanhos especificados.

A classe *TDBFS* (*Teacher's DataBase File System*) é uma classe que implementa um sistema de arquivos rudimentar de armazenamento das informações relativas ao controle de presença de alunos, provendo uma abstração ao programador para evitar que este tenha que manipular os dados na memória EEPROM em baixo nível no resto do código. A classe associa o RA e a *tag* do cartão *MIFARE*, a data e hora do inicio e termino das aulas e a presença do aluno em cada aula, além de manipular as regiões de acesso a memória *EEPROM* que por sua vez é abstraída pela classe *AT24C256*.

A classe *TConfig* (*Teacher's Configuration*) segue por abstrair informações relativas ao acesso do usuário administrador do sistema, no caso o professor que utiliza o sistema. O usuário administrador do sistema tem privilégios diferentes, como por exemplo o acesso ao

menu de configurações, esta classe abstrai o armazenamento de informações como a senha e a *tag* de identificação do cartão *MIFARE*.

## Problemas encontrados

Nos deparamos um problema de configuração das interfaces do microcontrolador. Ao utilizarmos algum dispositivo que fazia sua comunicação com base no protocolo *SPI* e utilizar também um dispositivo que fazia sua comunicação com base no protocolo *I2C*, havia um ‘congelamento’ na rotina do programa, mais precisamente no comando `Wire.endTransmission()`. Depois alguns dias de acesso a blogs e forúns que abordavam o assunto descobriu-se que a arquitetura interna do *Arduino ATmega328p (Arduino UNO REV3)* tem uma ligação interna entre a porta que é utilizada como *MOSI* pelo protocolo *SPI* e a porta que é utilizada como *SDA* pelo protocolo *I2C*.

O problema foi resolvido acrescentando-se uma função que desabilitava o uso das portas utilizadas pelo protocolo de comunicação *SPI*, toda vez que as funções que utilizavam a comunicação *SPI* finalizavam, alterava-se a configuração *I/O* da porta, utilizando-se por exemplo as seguintes funções:

```
void PN532::enable() {
    pinMode(_ss, OUTPUT);
    pinMode(_clk, OUTPUT);
    pinMode(_mosi, OUTPUT);
    pinMode(_miso, INPUT);
}

void PN532::disable() {
    pinMode(_ss, INPUT);
    pinMode(_clk, INPUT);
    pinMode(_mosi, INPUT);
    pinMode(_miso, OUTPUT);
}
```

Ressaltamos que pode haver outros meios de se eliminar este problema, porém este foi o único meio que encontramos para contornar o problema até o presente momento.

## Conclusão

O trabalho desenvolvido buscou prover uma solução para a necessidade de registro de presenças de alunos na UFABC.

De maneira geral, atingimos a grande maioria dos objetivos previamente estabelecidos, sendo que nosso objetivo final, neste projeto, era desenvolver um sistema (*hardware* e *software*) que possibilitasse o controle de frequência de alunos por meio de *smart cards*, para ser utilizado pelos professores e alunos da UFABC. Algumas etapas da metodologia estão para serem cumpridas, pois é necessário um teste de campo para o dispositivo, fundamental para avaliar e adequar o comportamento do dispositivo na prática. A utilização do dispositivo em um ambiente real ajudaria a identificar futuras necessidades emergentes e estabelecer metas de projeto diferentes para a concepção de um produto.

Os próximos trabalhos serão a realização de testes de campo (sala de aula) do dispositivo de forma a se fazer a análise de desempenho do equipamento identificando possíveis melhorias e correção de *bugs*. Posteriormente aos testes em campo, pode ser desenvolvido um *software* para computadores pessoais que baixe os dados do dispositivo e faça o lançamento de frequência dos alunos automaticamente em um banco de dados. O ultimo passo para desenvolvimento deste projeto seria a concepção de uma placa de circuito impresso para integrar todos os componentes e o desenvolvimento de um invólucro resistente.

## REFERÊNCIAS

- [1] ARDUINO SA. Arduino Uno. Disponível em: <https://www.arduino.cc/en/Main/arduinoBoardUno>. Acesso em 27 de junho de 2015.
- [2] ARDUINO SA. Arduino Ethernet Shield with SD Reader. Disponível em: <http://arduino.cc/en/Main/ArduinoBoardEthernet> >. Acesso em 04 de julho de 2015.
- [3] ELECFREAKS. RFID/NFC Shield. Disponível em: <http://www.electfreaks.com/store/rfidnfc-shield-p-479.html>>. Acesso em 27 de junho de 2015.
- [4] BECKERT, F. A. Rastreamento da produção com RFID usando Arduino para uma indústria moveleira. Trabalho de Conclusão do Curso de Sistemas de Informação da Universidade do Oeste de Santa Catarina – UNOESC. Chapecó, SC, 2012.
- [4] SCHMIDT, M. ARDUINO - A quick-start guide. Editora O'Reilly. Nova Iorque, NY, 2011.
- [5] McROBERTS, MICHAEL. Arduino Básico. Editora Novatec. São Paulo, SP, 2011
- [6] ARDUINO SA. INTERFACING with Hardware. Disponível em: <http://www.arduino.cc/playground/Main/InterfacingWithHardware>>. Acesso em 04 de julho de 2015.

[7] WERNECK, PEDRO. Introdução ao Arduino. Saber Eletrônica, v. 45 n. 437, p. 46-51. Editora Saber. São Paulo, SP, jun. 2009.

[8] SCAVARDA, LUIZ FELIPE; FILHO, CÍCERO NOGUEIRA; KRAEMER, VICTOR. RFID na Logística: Fundamentos e Aplicações. In: XXV Encontro Nacional de Engenharia de Produção. Porto Alegre, RS, 2005.

[10] PHILIPS SEMICONDUCTORS. PN532/C1 NFC controller Datasheet. Eindhoven, Holanda, 2006. Disponível em: <[http://www.nxp.com/documents/short\\_data\\_sheet/PN532\\_C1\\_SDS.pdf](http://www.nxp.com/documents/short_data_sheet/PN532_C1_SDS.pdf)>. Acesso em 04 de julho de 2015.

[11] INTERNATIONAL ORGANIZATION OF STANDARDIZATION (ISO 14443). "ISO/IEC 14443-3:2011(en) - Identification cards; Contactless integrated circuit cards; Proximity cards". Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:14443:-3:ed-2:v1:en>>. Acesso em 04 de julho de 2015.

[12] INTERNATIONAL ORGANIZATION OF STANDARDIZATION (ISO/IEC 18092). "Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)". Disponível em: <[http://read.pudn.com/downloads141/doc/612244/ISO\\_IEC\\_18092\\_2004.pdf](http://read.pudn.com/downloads141/doc/612244/ISO_IEC_18092_2004.pdf)>. Acesso em 04 de julho de 2015.

[13] SOUZA, W. B. Cartão MIFARE classic: ataques e medidas de contorno. Dissertação apresentada ao Instituto de Matemática e Estatística da Universidade de São Paulo para obtenção do título de Mestre em Ciências – São Paulo, nov. 2011.

[14] ATMEL. AT24C256 Two-wire Serial EEPROM Datasheet. Disponível em: <<http://www.atmel.com/Images/doc1116.pdf>>. Acesso em 04 de julho de 2015.

[15] MAXIM INTEGRATED. DS3231 - Extremely Accurate I2C - Integrated RTC/TCXO/Crystal - Datasheet. Disponível em: <<http://www.maximintegrated.com/en/products/digital/real-time-clocks/DS3231.html>>. Acesso em 04 de julho de 2015.

[16] HITACHI. HD44780U (LCD-II) Datasheet. Disponível em: <<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>>. Acesso em 04 de julho de 2015.