

ETEC ALBERT EINSTEIN

**BRUNA FRAGA
BRUNO MUNARÃO FATTE
GUILHERME ALVES DA SILVA**

IDENTIFICADOR DE PRESENÇA RFID

**SÃO PAULO
2020**

ETEC ALBERT EINSTEIN

**BRUNA FRAGA
BRUNO MUNARÃO FATTE
GUILHERME ALVES DA SILVA**

IDENTIFICADOR DE PRESENÇA RFID

Trabalho de conclusão de curso apresentado, para
obtenção do título de técnico em eletrônica.

Orientador: prof. Rosemeire Feracin de Souza

**SÃO PAULO
2020**

ETEC ALBERT EINSTEIN

**BRUNA FRAGA
BRUNO MUNARÃO FATTE
GUILHERME ALVES DA SILVA**

IDENTIFICADOR DE PRESENÇA RFID

Aprovados em ____/____/____ para obtenção do curso Técnico em Eletrônica

Professor Orientador: Edson Possani

Coordenador de Curso: Wilson Carvalho Junior

Coordenador Pedagógico: Fernanda Valli

Diretora: Silvia Petri

ETEC ALBERT EINSTEIN

TERMO DE AUTENTICIDADE

TRABALHO DE CONCLUSÃO DE CURSO – TCC

Nós, Bruna Fraga, Bruno Munarão Fatte e Guilherme Alves da Silva, regularmente matriculados no Curso Técnico de Nível Médio em Eletrônica da Escola Técnica Estadual Albert Einstein, município de São Paulo, declaramos ter pleno conhecimento do regulamento para realização do Trabalho de Conclusão de Curso do Centro Paula Souza. Declaramos, ainda, que o trabalho apresentado é resultado de nosso próprio esforço e que não há cópia de obras impressas ou eletrônicas segundo a lei 9610/98 que dispõe sobre direitos autorais e atribuições e ABNT – NBR 14.724:2011 itens 5.5 que dispõe sobre informação e documentação – citações em documentos apresentadas conforme ABNT NBR 10520/02.

São Paulo, ____ de Novembro de 2020.

Discente(s)	Registro Geral	Assinatura (s)
Bruna Fraga		
Bruno Munarão Fatte		
Guilherme Alves da Silva		

Docente(s) Registro Geral	Registro Geral	Assinatura(s)

DEDICATÓRIA

Dedicamos esse trabalho a todos os
professores, pais e coordenadores que
estiveram presentes em nossa formação como
técnicos.

AGRADECIMENTOS

O sentimento de gratidão e satisfação pela nossa formação é retratado nesse trabalho, agradecemos assim a todos os professores da Etec Albert Einstein por guiar-nos e aos nossos pais que apoiaram o estudo e valorizaram o conhecimento sempre, gerando incentivo para continuar e vencer todos os desafios da pandemia, a fim de que o projeto fosse entregue com resultado esperado.

Deixamos aqui também registrado o agradecimento especial a nosso amigo Vitor Hugo Macieira que nos auxiliou nos momentos de maior complexidade sempre demonstrando apoio.

RESUMO

Identificador de presença *RFID*. 2020. Trabalho de Conclusão de Curso (Tecnologia em eletrônica) – Etec Albert Einstein, São Paulo.

O presente projeto de TCC conta com um sensor de presença *RFID*, que tem como objetivo a especificação e controle de faixa de fluxo de pessoas por um determinado local. Nesse projeto é utilizado não apenas o sensor que detecta a presença, mas o Arduino que recebe o sinal e implementa a programação em *Excel*, concluindo assim de forma rápida e concisa os resultados de todas as entradas e saídas dos respectivos usuários. As pesquisas feitas identificaram a importância de um sistema que consiga efetuar o controle com exatidão, contribuindo por exemplo, para instituições que necessitam de registro em horários de trabalho ou presença de alunos, a favor de cumprir a carga horária estipulada.

Palavras-chave: Detecta a presença. Sensor *RFID*. Controle.

ABSTRACT

RFID presence identifier. 2020. Course Conclusion Paper (Technology in electronics) - Etec Albert Einstein, São Paulo.

This TCC project has an RFID presence sensor, which aims to specify and control the flow of people by a specific location. In this project it is used not only the sensor that detects the presence, but the Arduino that receives the signal and implements the programming in Excel, thus concluding quickly and concisely the results of all the inputs and outputs of the respective users. The surveys carried out identified the importance of a system that can carry out the control with accuracy, contributing, for example, to institutions that need registration in working hours or the presence of students, in favor of meeting the stipulated workload.

Keywords: Detects presence; RFID sensor; Control.

INTRODUÇÃO

Nosso projeto utiliza da eletrônica como aquela que nos auxilia a resolver problemas e é a base da construção e aplicação de todo o contexto de idealização do dispositivo.

O nosso projeto de conclusão de curso tem como objetivo a integração da eletrônica a fim de auxiliar na identificação de presença. Tornando-o conciso na resolução, de forma simples, do controle, precisão em horários de trabalho, ou identificação dos usuários.

A metodologia de construção do projeto baseia-se na implementação da lógica programável, que é um mediador entre o sensor e a linguagem de comunicação transmitida para computador, consolidando o uso do Arduino como protagonista do projeto, sendo viável e versátil a aplicação cotidiana.

SUMÁRIO

1 OBJETIVO	12
2 APLICABILIDADE	13
3 VANTAGENS E DESVANTAGENS DO PROJETO.....	14
3.1 PESQUISAS.....	14
4 IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA.....	18
4.1 SURGIMENTO DO SENSOR.....	18
4.2 CONCEITO BÁSICO E FUNCIONAMENTO	18
4.3 ETIQUETA <i>RFID</i>	19
4.3.1 Componentes.....	20
4.3.2 Antena	20
4.3.3 Microchip, módulos e memória	20
4.3.4 Tags	21
4.4 LEITORES DE ETIQUETA	22
4.5 DISTÂNCIA DE LEITURA	22
4.6 OPÇÕES DE TECNOLOGIA UTILIZADA	23
4.6.1 HDX - Half Duplex.....	23
4.6.2 FDX - Full Duplex	23
4.7 FREQUÊNCIA DE TRANSMISSÃO E RECEPÇÃO	23
4.8 TIPOS DE ACOPLAMENTOS EM FUNÇÃO DA FREQUÊNCIA.....	24
4.8.1 Acoplamento Indutivo.....	25
4.8.2 Acoplamento por Retrodifusão	26
4.9 MÓDULO MIDDLEWARE.....	27
4.10 APLICAÇÕES <i>RFID</i>	27
4.11 <i>RFID</i> ,CÓDIGO DE BARRAS E DIGITAL	28
4.12 MIFARE	28
5 ARDUINO	29
5.1 O QUE É.....	29
5.2 HARDWARE	29
5.3 SOFTWARE.....	30
5.4 ENTRADAS E SAÍDAS	30
5.5 COMPOSIÇÃO ARDUINO UNO.....	32
5.5.1 Microcontrolador.....	32
5.5.2 Pinagem do Microcontrolador.....	32
5.5.3 Interface serial ou USB	33
5.5.4 Comunicação.....	33
5.5.5 Alimentação	34
5.5.6 Shields e Módulos	35

5.5.7 Pinagem para conexão Shields e módulos	35
5.6 APLICAÇÕES ARDUINO	36
6 MÓDULO RFID	37
6.1 ESPECIFICAÇÕES	37
6.1.1 Leitor RFID	37
6.1.2 Tag Cartão 13,56MHz.....	37
6.1.3 Tag Chaveiro 13,56MHz.....	38
6.2 APLICAÇÕES	38
6.3 PINAGEM MÓDULO RFID	38
7 PROGRAMAÇÃO	39
7.1 LINGUAGEM UTILIZADA	39
7.1.1 Características C/C++	39
7.1.2 Biblioteca padrão	39
7.2 BIBLIOTECAS ADICIONADAS	40
7.2.1 <SPI.h>	40
7.2.2 <MFRC522.h>	41
7.2.3 Utilização no projeto	42
7.2.4 <TimeLib.h>	42
7.2.5 Macro da Biblioteca	43
7.2.6 Funções Utilizadas	43
8 UTILIZAÇÃO DO EXCEL.....	45
8.1 PROGRAMA DE SOFTWARE PLX-DAQ	45
9 CIRCUITO	47
9.1 FUNCIONAMENTO BÁSICO.....	47
9.2 ESTRUTURA GERAL	48
9.3 ESPECIFICAÇÕES DOS COMPONENTES	49
9.3.1 Piezo speaker – <i>buzzer</i>	49
9.3.2 Leds	50
10 CONCLUSÃO.....	51
11 REFERÊNCIAS	52
12 APÊNDICE	54
12.1 CÓDIGO DO PROJETO	54
12.2 CIRCUITO FÍSICO	60
12.3 DIAGRAMAS	62
13 ANEXO	63

1 OBJETIVO

O objetivo se dá a partir da solução do gerenciamento eficiente sobre o controle de entrada e saída, deslocamento de pessoas dentro de uma instituição de ensino, e também um sistema que efetue maior controle da frequência do aluno em sala de aula.

Objetivos Específicos:

- Criar um projeto que registre corretamente o horário de entrada e de saída dos usuários.
- Organizar informações relativas aos Alunos e aulas frequentadas.
- Efetuar a frequência, por meio eletrônico, mediante ao registro prévio de presença ou ausência.
- Enviar as informações para o sistema em Excel, possibilitando a visualização e armazenamento para consulta do administrador.

2 APLICABILIDADE

A aplicabilidade do projeto, Sensor de presença *RFID*, se dá em locais como:

- Cenário letivo (dentro de escolas, faculdades e outras instituições de ensino).
- Empresas e fábricas.

No cenário letivo (caso focado no atual trabalho) é rentável e facilita o controle de presença dos alunos em sala de aula, o mesmo tem a capacidade de saber se o aluno está dentro ou fora da sala de aula e quanto tempo ficou presente ou ausente na mesma, sem que o professor tenha preocupação em registrar a frequência dos alunos na aula.

Já no âmbito das empresas e fábricas, também é interessante, sendo similar ao sistema de controle de ponto, deixando o trabalhador sem nenhuma preocupação, pois será executado de forma automática, possibilitando assim a administração das cargas horárias trabalhadas.

3 VANTAGENS E DESVANTAGENS DO PROJETO

Dentre os pontos fortes temos precisão, (presença automática e individual, poupando tempo e até assegurando que o indivíduo específico esteja exatamente no local determinado), armazenamento dos dados (podendo comprovar a presença do indivíduo evitando problemas judiciais caso o mesmo não esteja no local informado), evitando também o esquecimento do registo de presença, e por fim independência do sistema de quem o faz uso, (pois não é preciso nenhuma ação humana para a detecção, torna o processo silencioso e automático, evitando tumultos).

O sistema é rentável pois tem um baixo custo, e possui um uso facilitado, capaz de atender uma demanda variada de clientes, pode ser instalado em vários locais da empresa ou do sistema de ensino, como portas, tendo controle das salas pela qual o indivíduo esteve presente.

Dentre as desvantagens poderíamos citar a clonagem das tags que estabelecem a presença e ausência dos indivíduos, porém hoje em dia muitos sistemas de controle de presença podem apresentar o mesmo problema, em comparação, o sistema de digital não permite a clonagem da mesma, porém ainda assim pode significar um custo muito mais elevado, já que a presença por RFID se dá em cada uma das salas, verificando de forma muito mais específica a atividade do usuário.

3.1 PESQUISAS

Foi realizada uma pesquisa de forma a coletar informações de participantes de instituições de ensino que pudessem colaborar com a compreensão dos benefícios do projeto perante a sociedade e ambiente corporativo.

Utilizamos esse conhecimento a fim de orientar, dar viabilidade ao produto e até assegurar a ideia como um auxiliar em processos como: chamadas em instituições de ensino de forma ampla, ponto de presença de trabalhadores, e até controle de fluxo de pessoas, demonstrando ter uma boa repercussão mediante aos entrevistados pois poupa o tempo e inviabiliza o esquecimento na realização da chamada, o que gera maior comprometimento dos alunos com a frequência.

Gráfico 1 – Tempo para realização da chamada (Autor, 2020)



Gráfico 2 – Problemas para realizar a chamada (Autor, 2020)

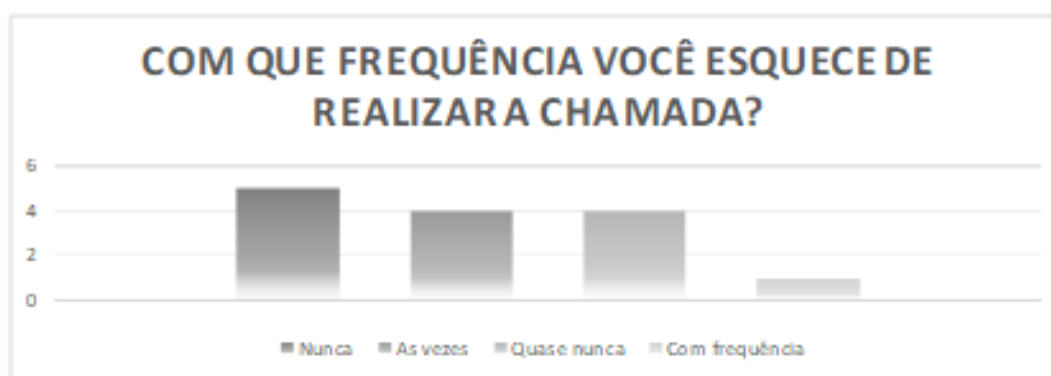


Gráfico 3 – Novos meios para realizar a chamada (Autor, 2020)

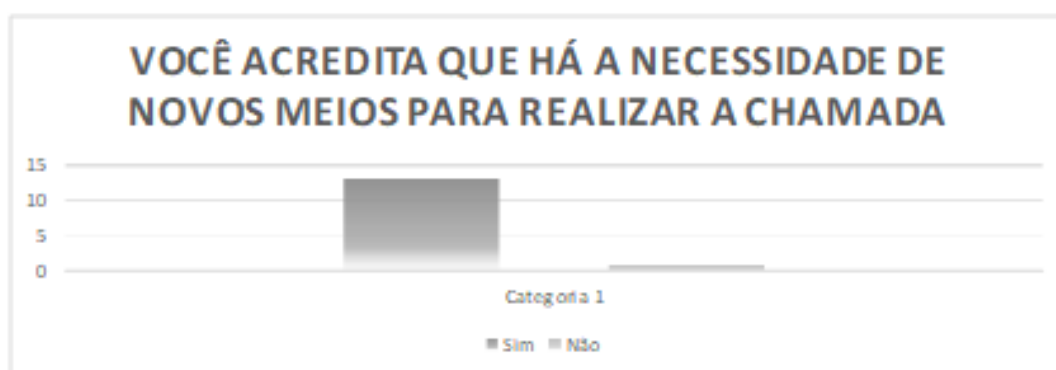


Gráfico 4 – Importância da Frequência (Autor, 2020)



Gráfico 5 – Problemas de Alunos com a chamada (Autor, 2020)

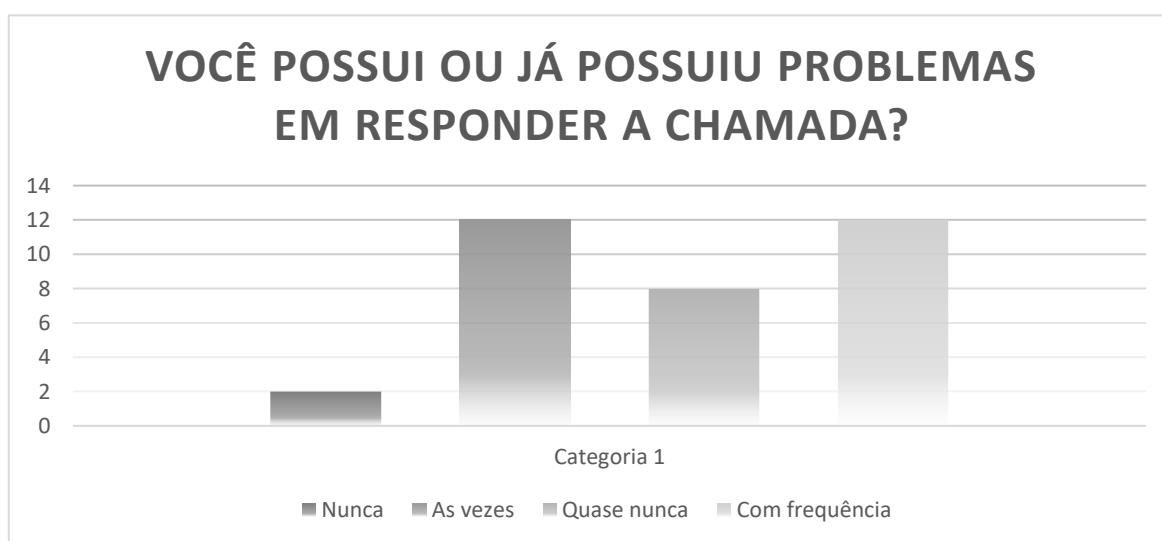


Gráfico 6 – Comprometimento com a aula mediante a frequência (Autor, 2020)

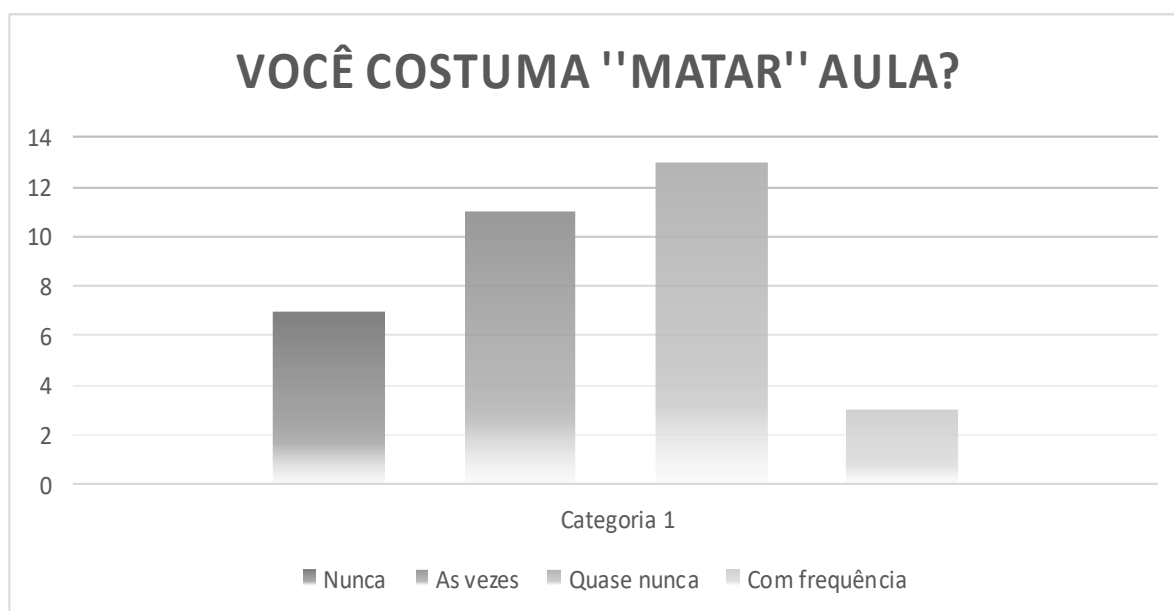
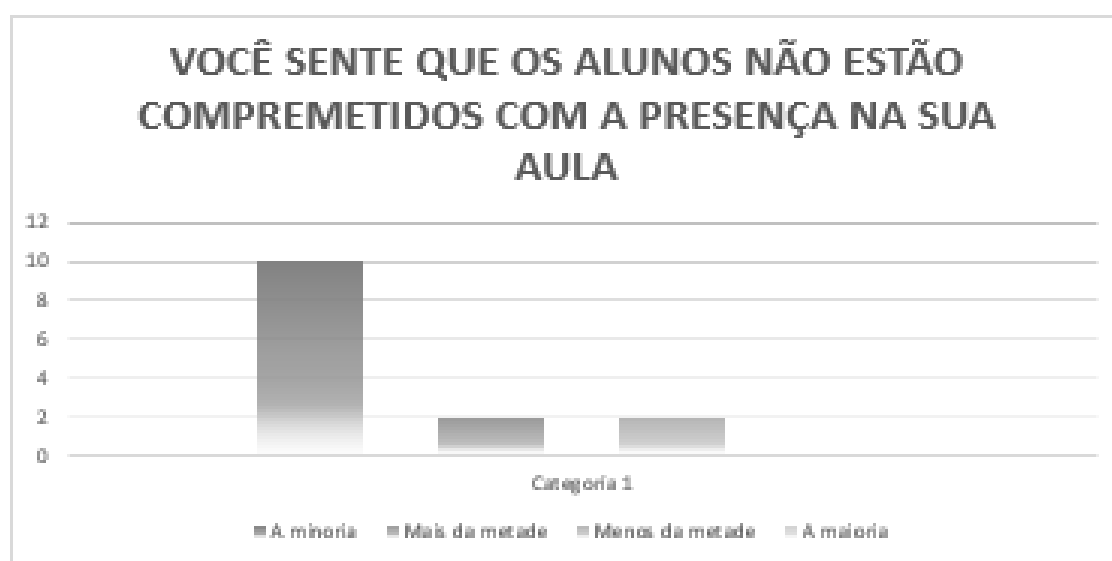


Gráfico 7 – Incentivo a presença (Autor, 2020)



De forma geral o projeto teve sucesso nas pesquisas, demonstrando boas respostas de aprovação, e assim, colaborativo, não sendo de apenas fácil uso e instalação, como grande benefício e aplicação da eletrônica no dia a dia.

4 IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA

O Identificador por radiofrequência ou *RFID* será abordado a fim da sua compreensão de forma ampla, a partir de sua aplicação no projeto como um método de identificação automática através de sinais de rádio transmitidos pelas tags, e posteriormente armazenados.

4.1 SURGIMENTO DO SENSOR

A tecnologia de identificação por radiofrequência surge a partir de avanços científicos feitos no século XIX, que confirmaram afirmações de Faraday e Maxwell.

Durante a segunda guerra mundial, os países em conflito usavam o radar para alertar a aproximação de aviões. Porém ainda não era possível identificar, por meio destes radares, quais aviões estavam se aproximando, não sendo previsto se representavam ameaças ou não. Assim os alemães desenvolveram uma técnica que ao girar os aviões o sinal era alterado, e, portanto, os responsáveis poderiam identificá-los como aviões alemães. Dessa forma, se deu a sua primeira utilização.

Posteriormente, Sir Robert Alexander Watson-Watt desenvolveu um transmissor, usado nos aviões, que determinassem uma resposta, identificando os aviões amigos rapidamente. Sendo esse sistema utilizado até os dias de hoje. (ACADÊMICO, 2013)

4.2 CONCEITO BÁSICO E FUNCIONAMENTO

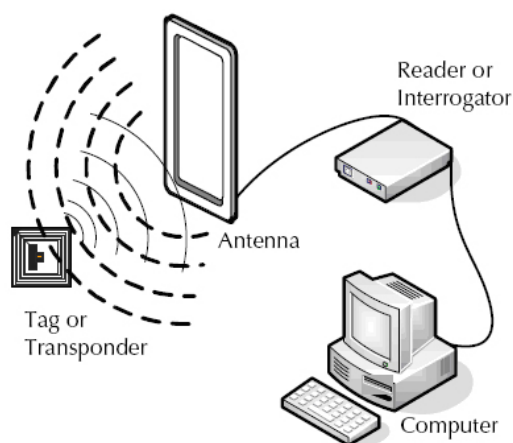


Figura 1 - Componentes básicos *RFID*

Fonte: GTA.UFRJ, 2020

RFID é um método de identificação automática através de sinais de rádio, recuperando e armazenando dados para comunicação entre dispositivos, sendo que essas informações ficam nas etiquetas *RFIDs*. Os sistemas *RFID* são compostos por uma antena ou bobina; um

transceptor (leitor ou programador), um *transponder* (RF tag, ou simplesmente tag).

Cada dispositivo contém informações e, o sinal é transmitido a partir do *transponder* para o *transceptor* que faz a leitura do sinal emitido e transfere para o dispositivo leitor que é acessado, conectado a um sistema computacional que gerencia as informações do sistema *RFID*, mostrando toda a informação contida no dispositivo *transponder* ou etiqueta *RFID*, com isso torna-se possível o uso dos dados para as mais diversas finalidades, entre elas a implementação de um banco de dados, como mostra a imagem ao lado.

4.3 ETIQUETA *RFID*

Dentro da composição da tag que contém um microchip com uma antena, o componente cumpre com uma função, a de transmitir e responder pelas informações, de onde a expressão *TRANSmitter / resPONDER* faz surgir o nome *transponder*.

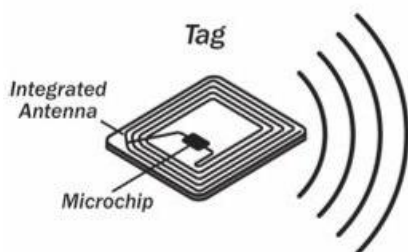
São aplicados diferentes formatos as etiquetas devido a suas diversas aplicações, embora todas possuam a mesma funcionalidade, a escolha do tipo de invólucro que a etiqueta é inserida torna-se essencial para a demanda específica de utilizações. A seguir possíveis tipos de encapsulamentos usados para exemplificação.



Figura 2 - Diferentes tipos de encapsulamento podem ser encontrados

Fonte: GTA.UFRJ, 2020

No caso do projeto utilizaremos o encapsulamento por chaveiro e cartão, para demonstração, porém pode ser adaptado a outros formatos visando a versatilidade para uso dos funcionários ou alunos, (público ao qual será direcionado o projeto).



O sistema tag (linguagem de marcação, também chamado de *transponder*) é formado por uma antena, um *CI* (circuito integrado, microchip) e seu respectivo encapsulamento, como a figura ao lado.

Figura 3 - RFID tag - Fonte: wikia.org, 2020

4.3.1 Componentes

O *transponder* possui um chip, uma antena e pode também conter memória permanente ou regravável, bateria e etc. Veremos agora os princípios dos componentes primordiais dentro da tag para que possa ser estabelecida a comunicação sem contato físico.

4.3.2 Antena

Antena *RFID* é o que possibilita o recebimento e transmissão de sinais sob modulação, a partir da frequência utilizada, o tamanho da antena pode ser alterado mediante o comprimento de onda (λ).

4.3.3 Microchip, módulos e memória

O microchip trata-se de um componente miniatura, o qual decodifica o sinal de entrada e encaminha esse sinal modulado e único para a antena, possuem basicamente os seguintes módulos:

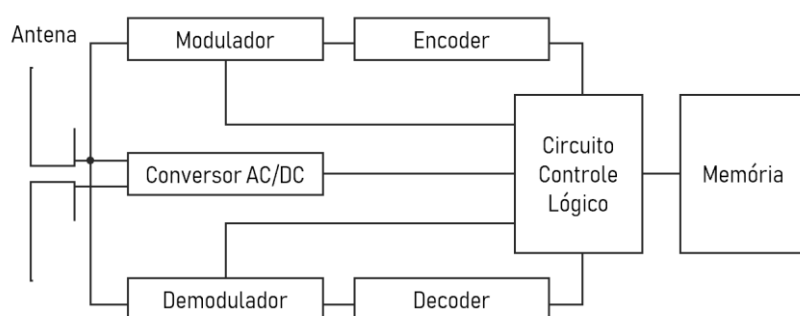


Figura 4 – Detalhamento do circuito interno tag passiva
Fonte: GTA.UFRJ, 2020

Segue abaixo a explicação do circuito:

- Conversor *AC/DC*: circuito de conversão do sinal com características da corrente alternada vinda da antena em corrente contínua, que vai para alimentação do circuito lógico.
- Modulador: modula um sinal digital em uma onda analógica, pronta a ser transmitida.
- Codificador ou *Encoder*: recebe o sinal modulado e o codifica para preparar o sinal que será enviado ao circuito de controle lógico.
- Memória: Onde ficam armazenados os dados da tag. Podendo ser do tipo somente leitura (onde armazena uma única informação) ou regravável (onde armazena um conjunto de informações).
- Circuito Controle Lógico: Armazena os dados recebidos na memória, e modula o sinal para que seja enviado como resposta que seguirá para o decodificador.
- Decodificador ou *Decoder*: modula o sinal recebido do controle lógico e o transforma em uma forma única.
- Demodulador: demodula o sinal analógico e o reconverte para o formato digital. (*RFID*)

4.3.4 Tags

O sinal emitido pela tag que se encontra dentro da pulseira possui um *ID*, (identidade única), que será posteriormente registrada, afinal a frequência de operação e o tipo de etiqueta influenciam diretamente se a mesma é capaz de iniciar a comunicação e se tem fonte de energia própria. Podemos separar assim as etiquetas mediante suas características:

A tag ativa possui bateria interna e utiliza essa alimentação em seu transmissor, que não necessita, portanto, do requerimento de seus dados pelo leitor, enviando-os de forma própria, com grande capacidade de transmissão e armazenamento de dados.

A tag semi-passiva usa sua fonte de energia interna somente para o funcionamento do microchip, que responde com a energia das ondas de requerimento enviadas pelo leitor. Possuem um alcance médio de 30 metros.

A tag passiva não possui alimentação interna e utiliza parte da energia transmitida pelo leitor para retransmitir os dados armazenados, portanto refletem a milímetros a 6 metros. (Caso que usaremos no projeto).

4.4 LEITORES DE ETIQUETA

Os leitores de etiquetas (também chamados de *transceptors*), efetuam o requerimento de informações sob as etiquetas enviando ondas de rádio, e a recepção de informações enviadas por elas. Responsável por decodificar o sinal recebido e enviar para um sistema de computador que possa administrar as informações, ou até mesmo um servidor. (Bases do RFID, 2018)

Os leitores podem apenas ler as informações (*RO*) dos *transponders* (etiquetas) ou podem sobrescreve-las, gravando dados na memória dos tags desde que esse seja do tipo leitura/gravação (*RW*). (RFID FAQ, 2017)

4.5 DISTÂNCIA DE LEITURA

A distância da leitura, a potência da leitora, tamanho da antena e direção são muito importantes pois são fatores que influenciam o cálculo certo da distância entre o leitor e a tag, a fim do bom funcionamento do sistema. Porém, a principal característica que pode determinar a distância para a comunicação correta é estabelecida pela frequência dos sinais transmitidos. (BOALIM, 2015)

FREQUÊNCIA	DISTÂNCIA	VANTAGENS	DESVANTAGENS
125 kHz a 134.3 kHz (LW)	Ate 30 cm	Boa operação próximo a metais e água	Alcance entre o leitor e a etiqueta curto e baixa taxa de leitura
13.56 MHz (HF)	Ate 1,5 m	Baixo custo das etiquetas; boa interação e boa qualidade de transmissão	Necessita de uma potência elevada nos leitores
860 MHz a 960 MHz (UHF)	De 1m ate 12m	Baixo custo, etiquetas com tamanho reduzido	Não opera bem próximo de metais e líquidos
2.45 GHz (SHF)	Mais que 100 metros	Velocidade de transmissão de dados	Não opera bem próximo de metais e líquidos; custo elevado

Tabela 1 – Frequências e suas características sob leitura

Fonte: GTA.UFRJ, 2020

4.6 OPÇÕES DE TECNOLOGIA UTILIZADA

De maneira simplificada, a tecnologia *RFID* se utiliza de uma comunicação dispersa no ar através de ondas de rádio que transmitem informações que portam os dados uni ou bidireccionalmente, o que dependerá da tecnologia que estiver sendo utilizada no momento, pois a transferência de dados entre leitor e tag depende da transferência de dados que ocorre de acordo com um dos dois processos principais que veremos a seguir.

4.6.1 HDX - Half Duplex

Onde a transmissão pode ser alternada, da tag para o leitor e do leitor para a tag, mas não de forma simultânea, sendo assim o processo unidirecional de transmissão.

4.6.2 FDX - Full Duplex

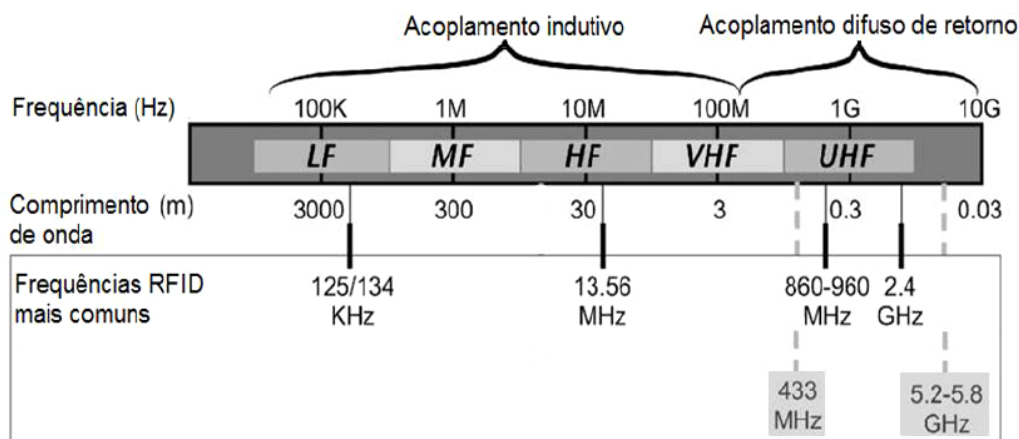
A transmissão ocorre nos dois sentidos simultaneamente. Onde o leitor é capaz de enviar para tag e vice versa, havendo uma conversa, portanto um processo bidirecional de transmissão. (Modos de Comunicação)

4.7 FREQUÊNCIA DE TRANSMISSÃO E RECEPÇÃO

O sensor *RFID* utiliza uma faixa de frequência que determina um maior ou menor raio de alcance, sofrendo influência sobre a distância máxima de leitura entre a etiqueta e o leitor, o tipo de material do objeto a ser etiquetado, requisitos dos clientes e/ou fornecedores, normas e regulamentações a serem atendidas, sendo previamente determinada para que não ocorra interferência no sinal, a frequência também define a taxa de transferência de dados.

Pode-se dizer que cada frequência tem características diferentes e, portanto, aplicações diferentes. Assim, as faixas de frequência disponíveis para a utilização do sistema *RFID* são as chamadas ISM (*Industrial-Scientific-Medical*).

Segue abaixo as frequências de operação do sensor.

Figura 5 – Frequências *RFID* - Fonte: GTA.UFRJ, 2020

Baixa Frequência, *Low Frequency* (LF) – Faixa de operação de 125 kHz até 134 kHz. Capaz de ler através de objetos metálicos, são ondas que compõe uma faixa de curta a média de distância de leitura e de baixo custo de operação. Nesta faixa o principal uso é para equipamentos militares e marítimos.

Alta Frequência, *High Frequency* (HF) – faixa de operação de 13,56 MHz para equipamentos *RFID*. Utilizada em curta e média distância, podem ser realizadas conexões transcontinentais, ou também sistemas de controle remoto e sistemas de *RFID* indutivo, controle de acesso e *smart cards*.

Ultra Alta Frequência, *Ultra High Frequency* (UHF) – faixa de operação de 300MHz até 3 GHz. A banda de 902 a 928 MHz é muito utilizada, conta com grande alcance, alta velocidade, porém custo mais elevado. Frequências que podem ser utilizadas por SRDs (*Short Range Devices*), portões automáticos, TV, militar, telefonia celular, comunicação com satélites, transmissão de dados via *WIFI*, *Bluetooth* entre outros. (GUILHERME DANTAS COUTO E TARSÍUS SÉRGIO MALAFAIA, 2019)

4.8 TIPOS DE ACOPLAMENTOS EM FUNÇÃO DA FREQUÊNCIA

A comunicação entre os tags e o *transceptor* é feita por uma antena que emite um sinal para ativar o tag e lê ou escreve um dado no tag como já vimos anteriormente, formando um sistema de propagação, pois os sistemas *RFID* utilizam dois métodos que distinguem e caracterizam os mesmos, um baseado em proximidade por acoplamento difuso de retorno (retrodifusão) e outro na propagação de ondas eletromagnéticas, o acoplamento eletromagnético.

4.8.1 Acoplamento Indutivo

De forma simplificada, no acoplamento indutivo (eletromagnético) tanto o leitor quanto a etiqueta utilizam bobinas que criam campos magnéticos o qual transfere dados entre o leitor e as etiquetas, como antenas. Segue abaixo a imagem do circuito e explicação de seu

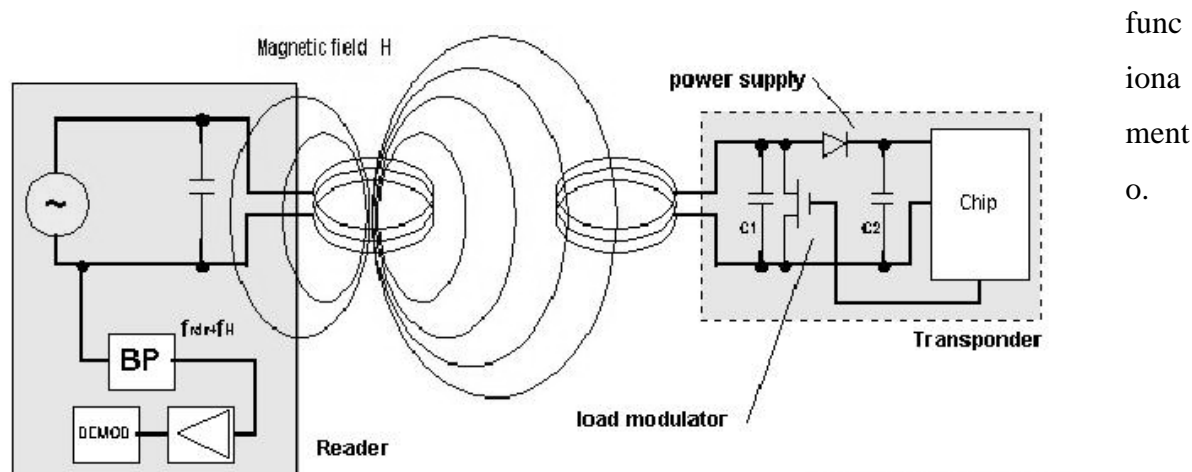


Figura 6 - Circuito de Acoplamento Indutivo – Fonte: WIKI.IFSC, 2020

Para a transmissão de energia, a bobina da antena do leitor gera um campo eletromagnético forte, de alta frequência, que penetra a seção transversal tanto da bobina como da área a sua volta. Uma vez que a escala de frequência da onda que se usa é várias vezes maior do que a distância entre a antena do leitor e o *transponder*, o campo eletromagnético pode ser tratado como um campo magnético em alternância (Como Funciona - Componentes RFID)

A bobina da antena do *transponder* interage com a bobina do leitor. Assim a tensão gerada na bobina da antena do *transponder* (por indução magnética) é retificada e serve como a fonte de alimentação para o microchip. Um capacitor seleciona sua indutância, que estando conectado paralelamente à bobina do leitor, influencia na indutância da mesma para gerar um circuito ressonante paralelo, portanto a frequência ressonante corresponda com a frequência da transmissão do leitor. Correntes muito elevadas são geradas pela bobina da antena do leitor que pode ser usado para gerar a energia requerida do campo para a operação do *transponder* remoto. Assim o acoplamento indutivo deve ser feito com proximidade de leitura por volta de 30 cm para *LF* (Frequências Baixa) e de 1m para *HF* (Frequências Altas).

4.8.2 Acoplamento por Retrodifusão

O Acoplamento por retrodifusão (também chamado de *backscatter*, ou acoplamento difuso de retorno) geralmente é utilizado por etiquetas passivas operando em *UHF* e permite maiores áreas de leitura. Nesse caso há o uso de ondas eletromagnéticas, e não campo. De forma mais completa analisaremos o princípio fundamental deste acoplamento com o funcionamento do circuito, assim como feito no acoplamento indutivo.

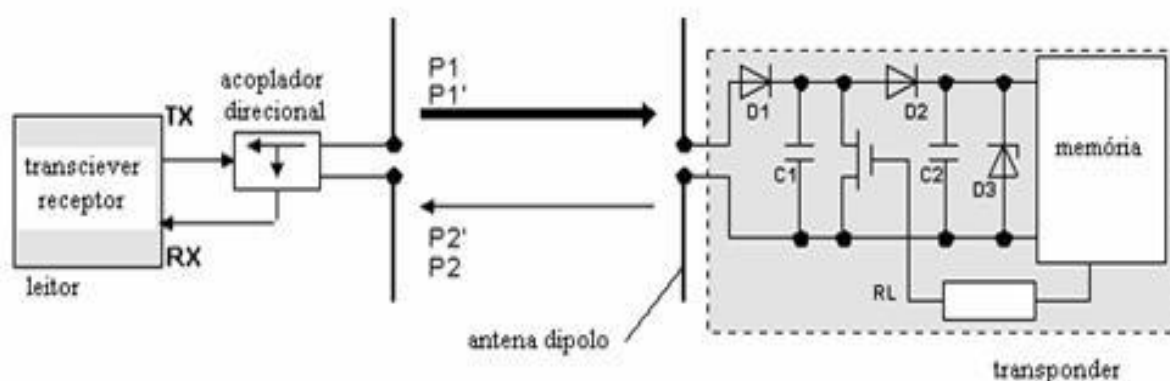


Figura 7 – Circuito de Acoplamento por Retrodifusão - Fonte: GTA.UFRJ, 2020

A energia $P1$ é emitida pela antena do leitor, mas somente uma pequena parte desta energia alcança a antena do *transponder*. A energia $P1'$ ($P1' < P1$) é fornecida às conexões da antena como a tensão e após a retificação feita pelos diodos $D1$ e $D2$, esta pode ser usada como voltagem inicial tanto para a desativação como para a ativação do modo econômico de energia. Os diodos usados são os diodos de barreira fraca de Schottky, que têm uma tensão particularmente baixa no ponto inicial. A tensão obtida pode também ser suficiente para servir como uma fonte de alimentação para distâncias curtas. Uma parte da energia entrante $P1'$ é refletida pela antena e retornada como o sinal $P2$. As características da reflexão da antena podem ser influenciadas alterando a carga conectada à antena.

A fim transmitir dados do *transponder* ao leitor, um resistor RL é ligado conectado em paralelo com a antena, trabalhando de modo alternado em sincronia com a transmissão dos dados. A amplitude de $P2$ refletida do *transponder* pode assim ser modulada. O sinal $P2$ refletido pelo *transponder* é irradiado no espaço livre. Uma parte pequena deste é captado pela antena do leitor. O sinal refletido viaja através da conexão da antena do leitor no sentido contrário, e pode ser desacoplado usando um acoplador direcional para depois ser transferido ao receptor de um leitor. Em média, o sinal do transmissor é aproximadamente dez vezes mais forte do que o recebido de um *transponder* passivo. (Como funciona *RFID*)

4.9 MÓDULO MIDDLEWARE

O *Middleware RFID*, nada mais é do que um *software* para identificação de tags *RFID*, que intermedia a comunicação entre o sistema da empresa e *hardware* (antena ou leitora e *transponders*) enviados ao banco de dados e assim podendo ser administrado em sistemas de computadores. Para que as empresas que decidam trabalhar com o *RFID* possam utiliza-lo sem a criação de um *software*, o *middleware* vem como uma solução, afinal seu desenvolvimento exige alto grau de conhecimento técnico.

Em nosso projeto a aplicação do *middleware* se da por conta própria, com a criação de um banco de dados a partir da utilização do Arduino. (Identificação por radiofrequência, 2013)

4.10 APLICAÇÕES *RFID*

O *RFID* tem diversas aplicações no mercado, a seguir algumas delas: Marcação de animal, bagagem de aviões, controle automático de veículos em pedágios, gerenciamento de estoques, acompanhamento de encomendas, *Checkout* automatizado. (Bases do RFID, 2018) (RFID: Princípios Gerais da Tecnologia e Aplicações, 2020). Segue abaixo exemplo de aplicação RFID:

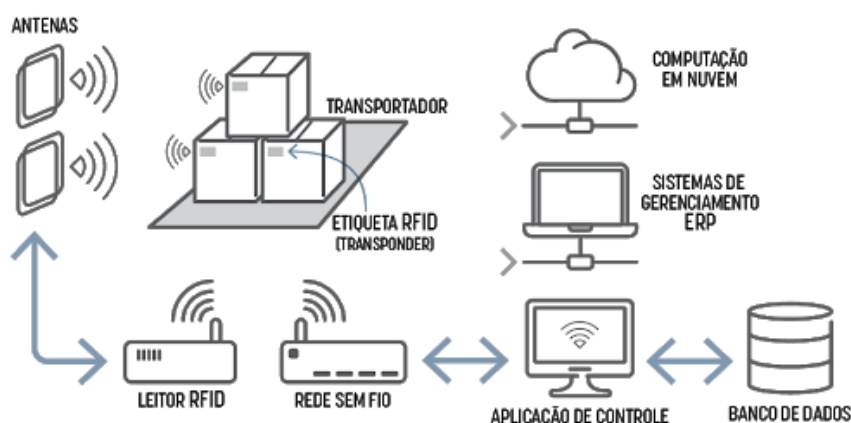


Figura 8 – Exemplo de Aplicação *RFID* - Fonte: Parson, 2020

4.11 *RFID* ,CÓDIGO DE BARRAS E DIGITAL

A aplicação do *RFID* se dá no atual trabalho como um identificador de pessoas e funcionários, viabilizando assim uma utilização paralela do sensor a fim de que não seja necessário o uso da digital e nem de código de barras que seriam soluções alternativas, cujos sensores sofrem mais interferências que as transmissões por *RFID* no caso da aplicação alvo. Seguem abaixo algumas das vantagens:

Possibilidade de atualização de dados e armazenamento, não uso de leitores de códigos e nem de digitais que podem gerar perturbações nos sinais de comunicação (o que gera maior praticidade em sua utilização no dia a dia), reutilização e durabilidade do sistema, além de não haver interferência na quantidade de tags lidas em um mesmo tempo (evitando filas e tornando o processo instantâneo).

(Solução de RFID para logística interna e externa, 2020)

4.12 MIFARE

MIFARE é uma marca que produz chips de circuito integrado usados em cartões inteligentes sem contato e cartões de proximidade, sendo a tecnologia MIFARE utilizada no trabalho e caracterizada pela alta frequência de 13,56MHz. A seguir mais detalhes sobre a tecnologia retirada de sites comerciais.

“Consiste em um cartão de PVC com um chip de capacidade pequena de memória e uma antena interna que percebe a aproximação do leitor através de campo magnético identificado pela frequência do cartão (*RFID*).

O cartão pode ser utilizado para várias aplicações, sua criptografia segura impede o acesso não autorizado a informações armazenadas, isto torna extremamente difícil copiar um cartão MIFARE. Não necessita de contato, sendo reconhecido até 10 cm do leitor, sofre menos desgaste que um cartão de contato, também não precisa ser inserido, como é com um cartão magnético com chip, basta só aproximá-lo.

Dentro da ISO 14443, existem algumas variações do MIFARE, de acordo com a qualidade das informações que armazenam.” (CALANDRIELLO, 2012)

5 ARDUINO

Utilizamos o Arduino no projeto como mediador entre as informações coletadas e o sensor *RFID*, a fim de que o mesmo possa desempenhar como a plataforma de código aberto a qual é programada e leva a informação ao *Excel* no sistema de computador, o qual administra o banco de dados a partir das comunicações feitas entre transponder e transceptor. (Arduino, 2020)

5.1 OQUE É

Arduino é uma plataforma de teste e projeção eletrônica, que concede liberdade ao usuário para executar, acessar e modificar o código fonte, e redistribuir cópias com ou sem modificações e de placa única. Pode servir tanto para o desenvolvimento de projetos interativos como ser conectada a um outro computador. (Arduino, 2020)

5.2 HARDWARE

Existem vários modelos de microcontroladores com diferentes quantidades de memória e configuração dos módulos de entrada e saída disponíveis. Abaixo segue um diagrama em blocos da arquitetura do hardware do Arduino, que será explicado detalhadamente ao decorrer da monografia. (Arduino, 2020)

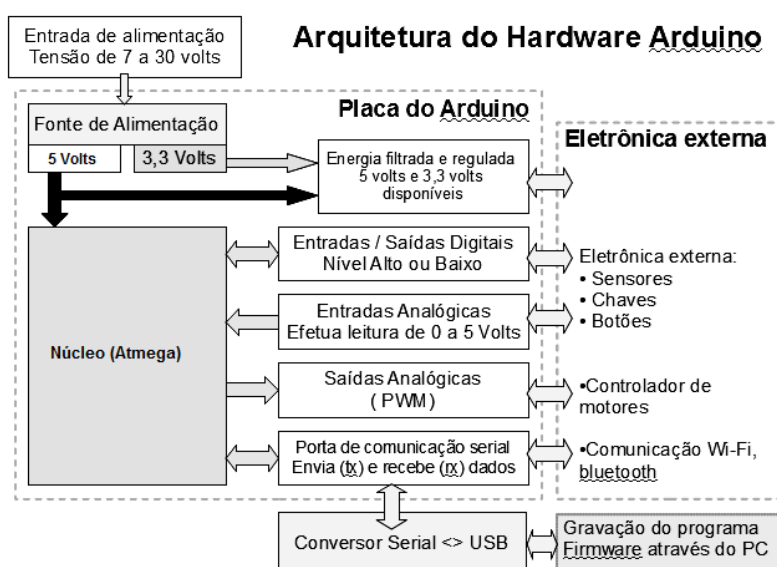


Figura 9 - Arquitetura do Hardware Arduino – Fonte: RESEARCHGATE, 2020

5.3 SOFTWARE

A IDE (Ambiente de Desenvolvimento Integrado) Arduino é uma ferramenta escrita em Java que reúne características e ferramentas para agilizar o processo de desenvolvimento de software com uma interface gráfica simples de usar, trata-se de um compilador de linguagens de programação GCC derivada dos projetos *Processing* e *Wiring* (programável em C e C++), que compila por Firmware o programa para o Arduino utilizando a conexão USB.

(Arduino – O que é, tipos e aplicações, 2020) (Arduino, 2020)

5.4 ENTRADAS E SAÍDAS

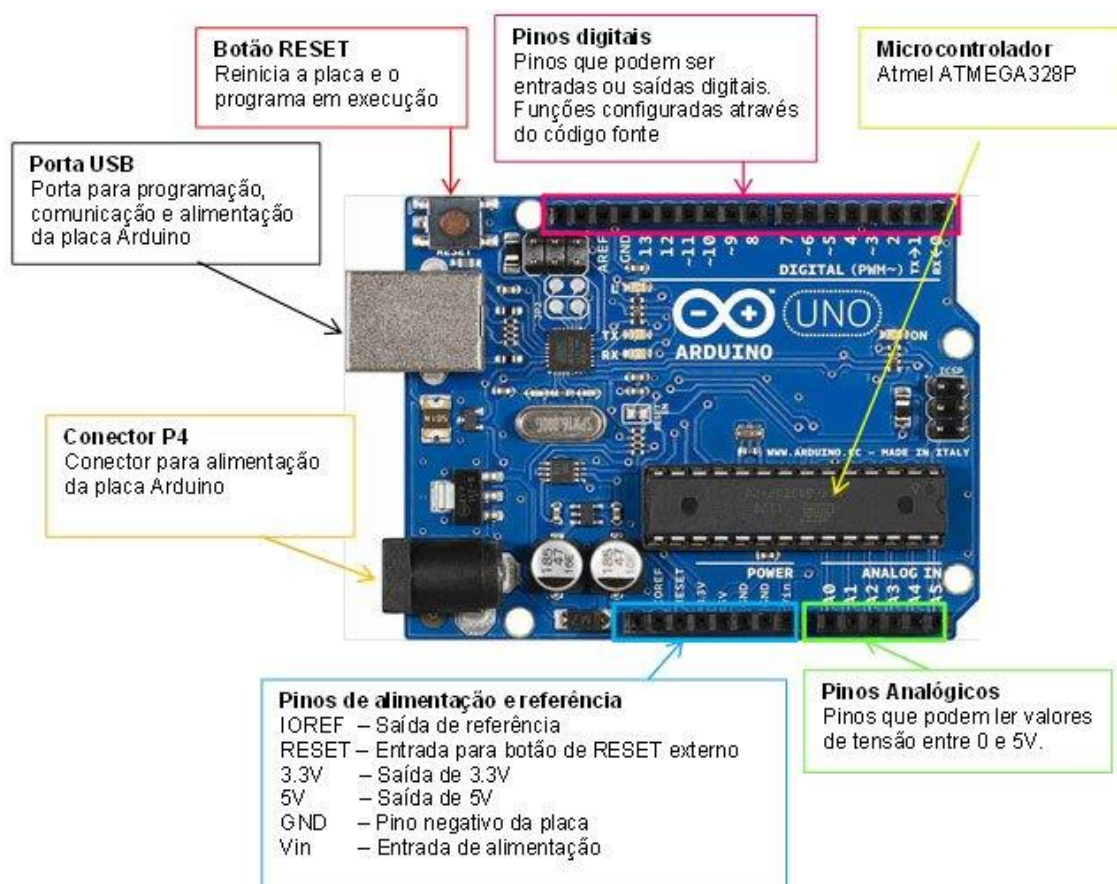


Figura 10 – Fonte: BAUDAELETRONICA, 2020

O desenho proporciona uma visualização geral do Arduino Uno e a localização dos respectivos componentes, como podemos observar o mesmo conta com:

- Pinos digitais (Os pinos digitais podem assumir duas posições, ON - HIGH (5V) e OFF - LOW (0V). Os pinos digitais do Arduino podem ser configurados como entradas ou saídas, os pinos digitais podem fornecer até 40 mA (miliampères) de corrente para outros dispositivos ou circuitos).
- Pinos analógicos (Para medir o valor de sinais analógicos, o Arduino utiliza de um conversor analógico-digital (ADC) interno, já que ele não possui apenas dois estados como nos pinos digitais. O ADC transforma a tensão analógica num valor digital).

Segue exemplificação:

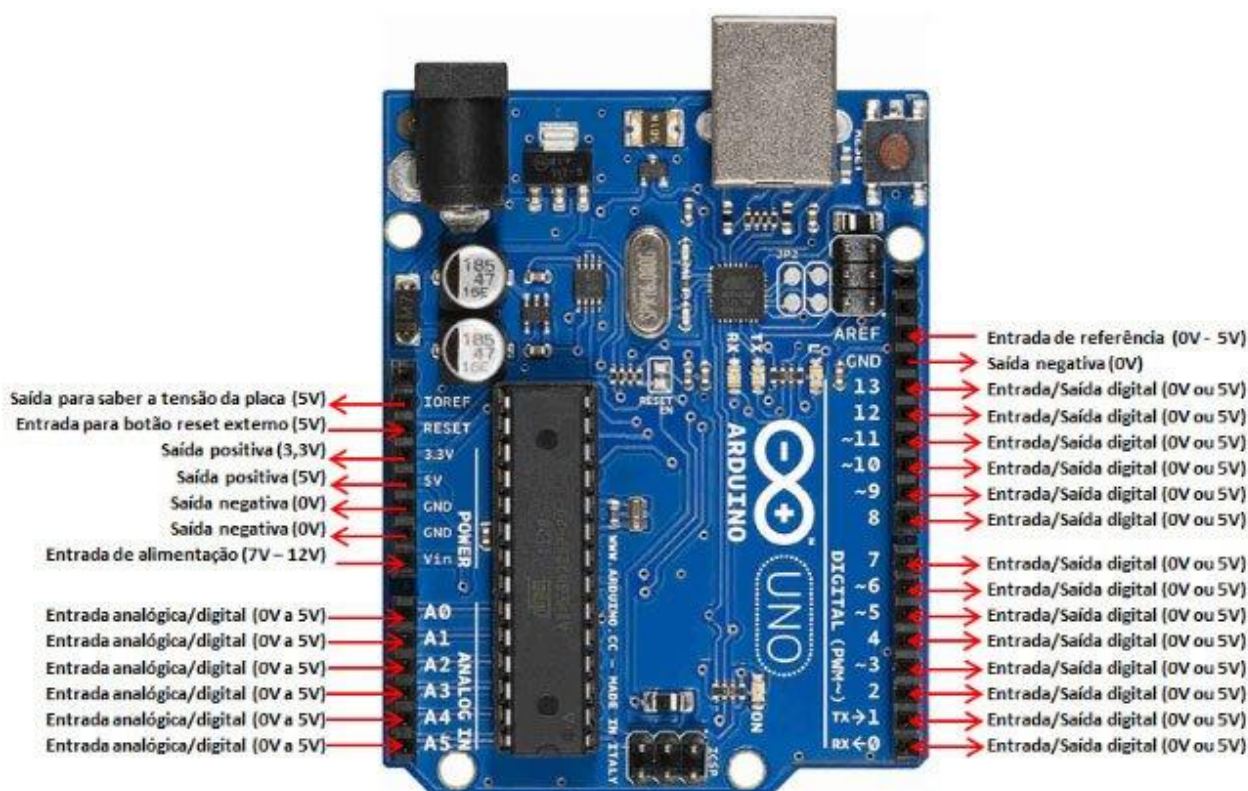


Figura 11 – Fonte: BAUDAELETRONICA, 2020

Na imagem a seguir é possível identificar cada um dos pinos do Arduino levando em consideração se é uma entrada, saída, voltagem, e o sinal respectivo.

5.5 COMPOSIÇÃO ARDUINO UNO

Uma placa Arduino é composta por um microcontrolador e alguns componentes complementares que facilitam a programação e incorporação para outros circuitos. No atual projeto é utilizado a placa Arduino UNO, e, portanto, será explicada com foco em suas características específicas de fabricação.

5.5.1 Microcontrolador

O Arduino possui um microcontrolador Atmel megaAVR (ATmega328) de chip único com uma arquitetura de 32KB (com recurso de leitura durante gravação, sendo 512 Bytes utilizados para *bootloader*) , e classificada como dispositivos de 8-bits RISC (apesar de serem de 8 bits avançada e com encapsulamento DIP28, cada instrução consiste de palavras de 16 bits, onde as informações residem no núcleo), utiliza memória flash não voláteis (em que os dados são conservados e portanto não há perda ao se retirar a fonte de energia), além de 1 KB de EEPROM, 2 KB de SRAM, 23 linhas de Entradas e Saídas de uso geral, 32 registros de trabalho de uso geral, três temporizador flexível / contadores com modos de comparação, interrupções internas e externas, USART programável serial, uma interface serial de 2 fios orientada a bytes, porta serial SPI, conversor A / D de 6 canais de 10 bits (8 canais em pacotes TQFP e QFN / MLF) , temporizador de *watchdog* programável com oscilador interno (ao executar instruções em um único ciclo de *clock* o dispositivo consegue equilibrar o consumo de energia e a velocidade de processamento) e pode operar a até 20 MHz, porém na placa Arduino UNO opera em 16 MHz e 1,8-5,5 volts. (SOUZA, 2013) (ARDUINO UNO REV3, 2020)

5.5.2 Pinagem do Microcontrolador

Possui 28 pinos, sendo que 23 desses podem ser utilizados como entradas e saídas. Segue a imagem referente ao Datasheet:

PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
Vcc	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Figura 11 - Pinagem ATmega328PDIP – Fonte: Atmel Images datasheet, 2020

5.5.3 Interface serial ou USB

A placa ainda conta com um conector USB para envio do código de programação e alimentação da placa. Para cada uma das funções segue explicação específica:

5.5.4 Comunicação

A comunicação serve para interligar-se ao hospedeiro (computador), que é usado para programá-la e interagir com a mesma em tempo real, e é feita nos pinos 0 e 1, que podem ser utilizados para comunicação serial (sendo que estes pinos são ligados ao microcontrolador responsável pela comunicação USB com o PC).

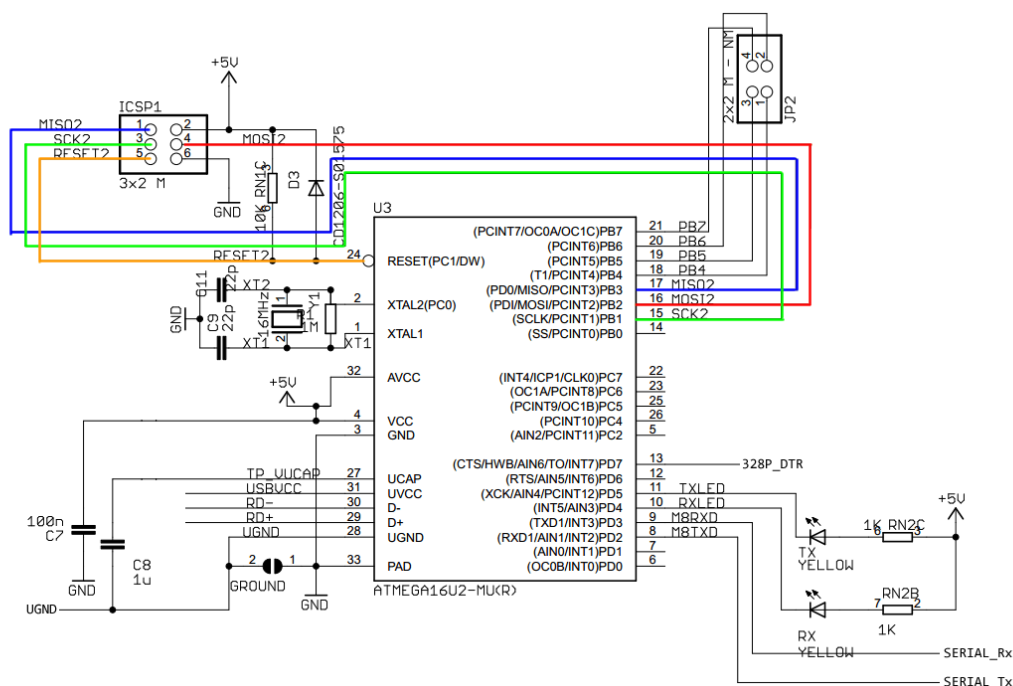


Figura 12 – ATmega16U2 - Fonte: Rheingoldheavy Subsystem, 2020

O microcontrolador ATMEL ATMEGA16U2 da imagem a cima é o responsável pelo upload do código binário feito pelo usuário, ou seja, a comunicação da programação compilada para o Arduino. Possui um conector ICSP para gravação de firmware através de um programador ATMEL, para atualizações futuras.

Nesse microcontrolador também estão conectados dois *leds* (TX de transmissão e), RX de recepção), controlados pelo software do microcontrolador, que indicam o envio e recepção de dados da placa para o computador. (SOUZA, 2013)

5.5.5 Alimentação

A placa pode ser alimentada pela conexão USB ou por uma fonte de alimentação externa possibilitando a conexão de baterias ou fontes para alimentar a placa quando ela não está conectada no USB.

A alimentação externa pode receber de 7V a 12V, que será regulada pelo CI NCP1117, e o diodo D1 protetor do circuito para casos onde a tensão seja invertida, o circuito segue abaixo.

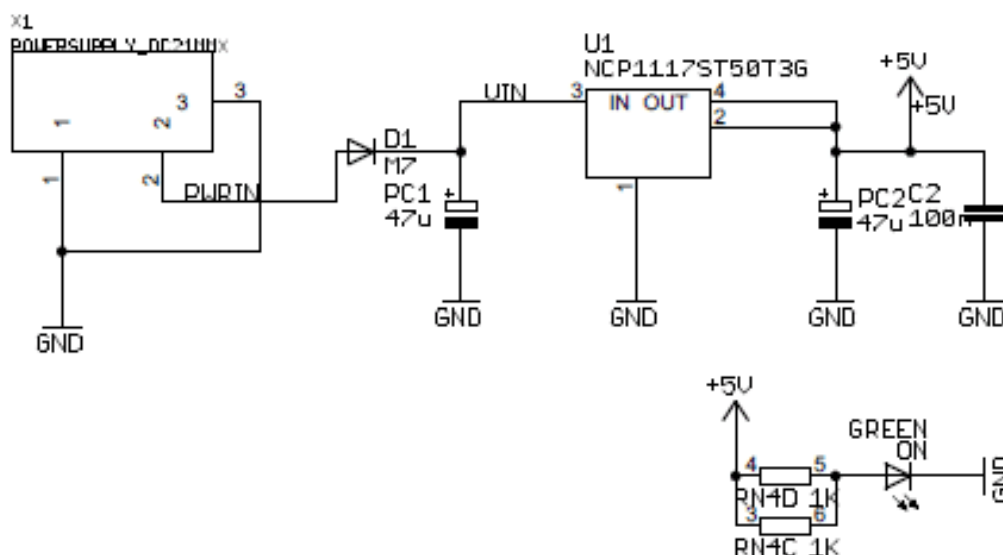


Figura 13 - Circuito Regulador de tensão – Fonte: Embarcados Arduino-uno, 2020

5.5.6 Shields e Módulos

Os Shields são placas desenvolvidas para atuarem em conjunto com o Arduino que estendem a capacidade de Hardware do mesmo, adicionando funções.

Os módulos são pequenas placas que contém sensores, resistores, capacitores ou LEDs, que podem ser acopladas, portanto sendo harmônicas eletricamente e mecanicamente. (OLIVEIRA, 2020)

Seguem alguns exemplos de Shields e módulos:

Shield Ethernet, data logger, XB, multifunções, impressora 3D, groove, etc.

Módulo relé de 1 canal, sensor de som, botão, sensor de choque, codificador rotativo, sensor de inclinação, sensor de movimento, etc.

5.5.7 Pinagem para conexão Shields e módulos

Como já havia sido observado o Arduino possui pinagem específica para suportar os *shields* e módulos.

Sendo:

IOREF – Tensão de referência para que *shields* possam selecionar o tipo de interface apropriada (3,3V ou 5V).

RESET – Reset externo da placa Arduino.

3,3 V. – Fornece tensão de 3,3V e corrente máxima de 50 mA.

5 V – Fornece tensão de 5 V para alimentação de *shields* e circuitos externos.

GND – Pinos de referência, terra.

VIN – Pino para alimentar a placa através de *shield* ou bateria externa.

(SOUZA, 2013)

5.6 APLICAÇÕES ARDUINO

Já que o Arduino pode ser conectado a diversos módulos e *shields* as possibilidades de aplicações são inúmeras, por conter potencial de automatização por aprendizado, as bibliotecas e suporte pedagógico do Arduino possibilitam que se torne uma plataforma de uso livre e incentivo a aplicação da lógica programável e da eletrônica, o que contribui com a divulgação das ideias e gera grandes potenciais de utilização da tecnologia para todos.

Dentre as formas finais de aplicação podemos citar alguns dos projetos: Automação básica via celular, aplicações em robótica, biometria digital, comunicação wireless, comunicação entre dispositivos remotamente, entre outros projetos que podem sempre ter acessório adicionados. (EDUARDO HENRIQUE FERRONI)

6 MÓDULO *RFID*

O sensor *RFID* também pode ser acoplado no Arduino utilizando o módulo leitor *RFID* baseado no chip MFRC522. Este chip, de baixo consumo e pequeno tamanho, permite sem contato ler e escrever em cartões que seguem o padrão *Mifare*, muito usado no mercado.

6.1 ESPECIFICAÇÕES

6.1.1 Leitor *RFID*

- Corrente de trabalho: 13-26mA / DC 3.3V
- Corrente ociosa: 10-13mA / 3.3V
- Corrente *Slep*: <80uA – Pico de corrente: <30mA
- Frequência de operação: 13,56MHz
- Tipos de cartões suportados: *Mifare1 S50, S70 Mifare1, Mifare UltraLight, Mifare Pro, Mifare Desfire*
- Temperatura de operação: -20 a 80 graus Celsius
- Temperatura ambiente: -40 a 85 graus Celsius
- Umidade relativa: 5% – 95%
- Parâmetro de Interface SPI
- Taxa de transferência: 10 Mbit/s
- Dimensões: 8,5 x 5,5 x 1,0cm (ELETRÔNICOS, 2020)

6.1.2 Tag Cartão 13,56MHz

- Frequência de operação: 13,56MHz
- Taxa de transferência: 106Kbaud
- Capacidade: 8Kbit / 16 partições
- Alcance: 2 a 10cm
- Tempo de leitura e escrita: 1 a 2ms
- Temperatura de operação: -20° a 55° celsius
- Material: PVC

6.1.3 Tag Chaveiro 13,56MHz

- Frequência de operação: 13,56MHz
- Taxa de transferência: 106Kbaud
- Capacidade: 8Kbit / 16 partições
- Alcance: 2 a 10cm
- Tempo de leitura e escrita: 1 a 2ms
- Temperatura de operação: -20° a 80° celsius
- Material: ABS (OLIVEIRA, 2020)

6.2 APLICAÇÕES

A partir das transmissões de informações feitas pela tag é possível criar interações que após identificados os sinais que acontecem em consequência da operação, como abertura de uma cancela eletrônica, abertura de um portão, liberação de roleta na portaria de uma empresa, ou até mesmo a aplicação do projeto atual, que envia a informação para o Arduino e mediante programação contata a hora e presença de cada tag por meio do ID único contido dentro da mesma. (OLIVEIRA, 2020)

6.3 PINAGEM MÓDULO RFID

O módulo RFID exige pinagem específica, possuindo 8 pinos que seguem a seguinte sequência de ligação:

- Pino SDA ligado na porta 10 do Arduino
- Pino SCK ligado na porta 13 do Arduino
- Pino MOSI ligado na porta 11 do Arduino
- Pino MISO ligado na porta 12 do Arduino
- Pino NC – Não conectado
- Pino GND ligado no pino GND do Arduino
- Pino RST ligado na porta 9 do Arduino
- Pino 3.3 – ligado ao pino 3.3 V do Arduino

7 PROGRAMAÇÃO

7.1 LINGUAGEM UTILIZADA

Como já vimos o Arduino trabalha com as linguagens de programação C/C++ (é uma linguagem de programação compilada multi-paradigma, seu suporte inclui linguagem imperativa, orientada a objetos e genérica e de uso geral), por isso o projeto conta com a utilização da IDE, a fim de que seja estabelecidas as instruções para que os sinais transmitidos pelas tags possam chegar ao computador de forma a estabelecer a identificação únicas de cada tag e sua respectiva frequência.

7.1.1 Características C/C++

Dentre as características temos algumas regras que regem a lógica C++:

Sendo desenvolvido para ser uma linguagem tipada estaticamente e de propósito geral que é tão eficiente e portátil quanto o C, com suporte para múltiplos paradigmas, e fornecer ao programador escolhas, mesmo que seja possível ao programador escolher a opção errada, dando mais liberdade ao código, que para ser o mais compatível com C possível, fornece transições simples e evita fornecer facilidades que são específicas a certas plataformas ou a certos grupos de desenvolvedores, além de não exigir *overhead* para facilidades que não são utilizadas. (Wikipédia, 2020)

7.1.2 Biblioteca padrão

A biblioteca padrão do C++ é constituída pela biblioteca C (com algumas pequenas modificações), o gabaritos STL (fornece ferramentas úteis como *containers*, vetores, listas, algoritmos, entre outros) e iteradores (ponteiros inteligentes genéricos para acessar tais *containers* e interligá-los aos algoritmos).

O projeto *Boost* fornece elementos adicionais à STL, dos quais alguns já são considerados a serem parte da biblioteca padrão no futuro. (Wikipédia, 2020)

7.2 BIBLIOTECAS ADICIONADAS

7.2.1 <SPI.h>

Interface Periférica Serial (SPI) é um protocolo de dados seriais para se comunicar com um ou mais dispositivos periféricos rapidamente em curtas distâncias. Com uma conexão SPI, há sempre um dispositivo mestre (geralmente um microcontrolador) que controla os dispositivos periféricos.

- **MISO** (*Master In Slave Out*) - A linha *Slave* para enviar dados ao mestre,
- **MOSI** (*Master Out Slave In*) - A linha *Master* para o envio de dados aos periféricos,
- **SCK** (*Serial Clock*) - Os pulsos de *clock* que sincronizam a transmissão de dados gerados pelo mestre

e uma linha específica para cada dispositivo:

- **SS** (*Slave Select*) - o pino em cada dispositivo que o mestre pode usar para habilitar e desabilitar dispositivos específicos.

Quando o pino de seleção de escravo de um dispositivo está baixo, ele se comunica com o mestre. Quando está alto, ele ignora o mestre. Isso permite que você tenha vários dispositivos SPI compartilhando as mesmas linhas MISO, MOSI e CLK.

De modo geral, existem quatro modos de transmissão. Esses modos controlam se os dados são deslocados para dentro e para fora na borda ascendente ou descendente do sinal do *clock* de dados (chamada de fase do *clock*) e se o *clock* está ocioso quando alto ou baixo (chamada de polaridade do *clock*). Os quatro modos combinam polaridade e fase de acordo com esta tabela:

Modo	Polaridade do relógio (CPOL)	Fase do relógio (CPHA)	Borda de saída	Captura de dados
SPI_MODE0	0	0	Queda	Aumentar
SPI_MODE1	0	1	Aumentar	Queda
SPI_MODE2	1	0	Aumentar	Queda
SPI_MODE3	1	1	Queda	Aumentar

Figura 14 –
Polaridade e Fases
do Clock – Fonte:
SPI library, 2020

Depois dos parâmetros SPI, é usada a *SPI.beginTransaction* (para começar a usar a porta SPI), que é então configurada, sendo a maneira mais simples e eficiente de usar o *SPISettings*. Por exemplo:

```
SPI.beginTransaction (SPISettings (14000000, MSBFIRST, SPI_MODE0));  
(SPI library, 2020)
```

7.2.2 <MFRC522.h>

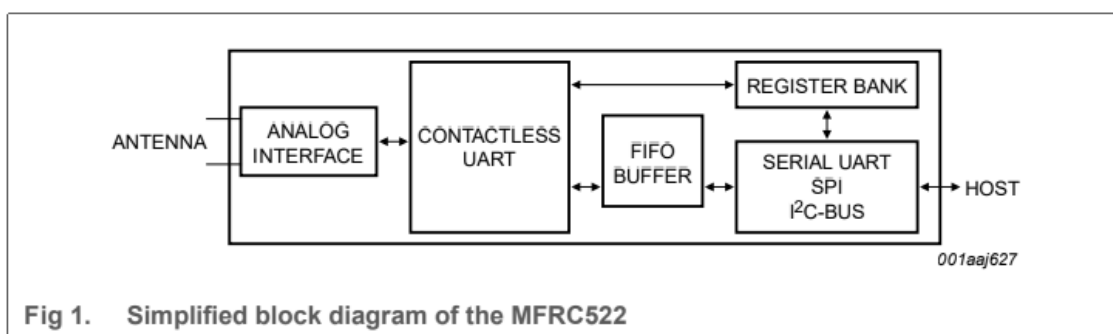
O MFRC522 é uma solução de leitor / gravador IC, que trabalha em 3V, utilizando o padrão MIFARE de 13,56 MHz. O leitor suporta o modo ISO / IEC 14443 A / MIFARE.

O transmissor interno é capaz de acionar uma antena de leitor / gravador projetada para se comunicar com os cartões e transponders sem circuitos ativos adicionais e o módulo receptor fornece uma implementação robusta e eficiente para demodulação e decodificação de sinais de placas e transponders. O módulo digital gerencia o enquadramento completo e a funcionalidade de detecção de erro (paridade e CRC) e é compatível com os produtos MF1xxS20, MF1xxS70 e MF1xxS50, suporta comunicação sem contato e usa velocidades de transferência MIFARE mais altas de até 848 kBd em ambas as direções.

As seguintes interfaces de host são fornecidas:

- Interface Periférica Serial (SPI) – explicada anteriormente
- UART serial (semelhante ao RS232 com níveis de tensão dependentes da alimentação de tensão do pino)
- Interface de barramento I2C

Onde: Figura 15 – Diagrama de Blocos MFRC522 - Fonte: Data-sheet MFRC522, 2020



- A interface analógica é lida com a modulação e demodulação dos sinais analógicos.
- O UART gerencia os requisitos de protocolo para a comunicação.
- Os protocolos trabalham em cooperação com o host. O buffer FIFO garante dados rápidos e convenientes.
- São transferidos de *and* para o host e o para o UART, e vice-versa.

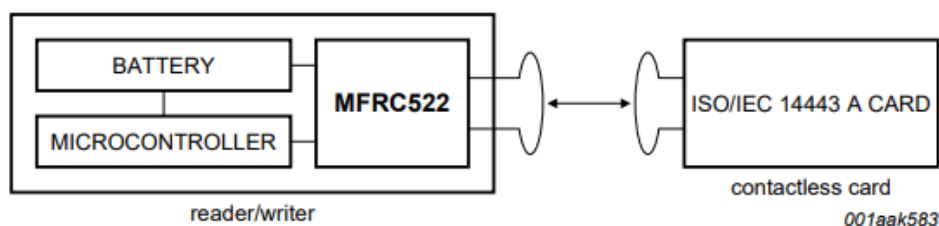


Figura16 – Visão Geral MRFC522 - Fonte: Data-sheet MFRC522, 2020

Na imagem acima é possível identificar a A tag compatível com a ISO/IEC 14443 A CARD e o leitor com o chip MFRC522, a bateria e o microcontrolador, tendo velocidade de transferência de 106 kBd a 848 kBd.

7.2.3 Utilização no projeto

Por fim a biblioteca MFRC522 atua sobre a biblioteca SPI.h, simplificando o processo de comunicação, facilitando o uso do Módulo Leitor RFID-RC522. (Datasheet MFRC522)

7.2.4 <TimeLib.h>

Time.h é um arquivo cabeçalho que define quatro tipos de variáveis, duas macros e várias funções para manipulação de data e hora, contendo a timelib.h dentro de seu arquivo zipado adicionado ao IDE do Arduino. Abaixo seguem as variáveis e sua respectiva função:

size_t Este é o tipo integral sem sinal e é o resultado da palavra-chave <i>sizeof</i> .
clock_t Este é um tipo adequado para armazenar o tempo do processador.
time_t is Este é um tipo adequado para armazenar a hora do calendário.
struct tm Esta é uma estrutura usada para armazenar a hora e a data.

Tabela 2 - Variáveis da biblioteca Time.h – Fonte:

https://www.tutorialspoint.com/c_standard_library/time_h.htm

7.2.5 Macro da Biblioteca

<p>NULO</p> <p>Esta macro é o valor de uma constante de ponteiro nulo.</p>
<p>CLOCKS_PER_SEC</p> <p>Esta macro representa o número de <i>clocks</i> do processador por segundo.</p>

Tabela 3 – Macros da biblioteca Time.h – Fonte: tutorialspoint standard library, 2020

7.2.6 Funções Utilizadas

Descrição da função
<p><u>char * asctime (const struct tm * timeptr)</u></p> <p>Retorna um ponteiro para uma string que representa o dia e a hora da estrutura timeptr.</p>
<p><u>clock_t clock (vazio)</u></p> <p>Retorna o tempo do clock do processador usado desde o início de uma era definida pela implementação (normalmente o início do programa).</p>
<p><u>char * ctime (const time_t * timer)</u></p> <p>Retorna uma string que representa a hora local com base no cronômetro do argumento.</p>
<p><u>double difftime (time_t time1, time_t time2)</u></p> <p>Retorna a diferença de segundos entre time1 e time2 (time1-time2).</p>
<p><u>struct tm * gmtime (const time_t * timer)</u></p> <p>O valor do cronômetro é dividido na estrutura tm e expresso em Coordinated Universal Time (UTC), também conhecido como Greenwich Mean Time (GMT).</p>
<p><u>struct tm * localtime (const time_t * timer)</u></p> <p>O valor do temporizador é dividido na estrutura tm e expresso no fuso horário local.</p>

time_t mktime (struct tm * timeptr)

Converte a estrutura apontada por timeptr em um valor time_t de acordo com o fuso horário local.

size_t strftime (char * str, size_t maxsize, const char * formato, const struct tm * timeptr)

Formata a hora representada na estrutura timeptr de acordo com as regras de formatação definidas em format e armazenadas em str.

time_t time (time_t * timer)

Calcula a hora atual do calendário e a codifica no formato time_t.

Tabela 4 –Funções da biblioteca Time.h – Fonte: tutorialspoint standard_library, 2020

8 UTILIZAÇÃO DO EXCEL

O Microsoft Excel é software editor de planilhas eletrônicas, feito pela Microsoft para computadores e dispositivos móveis, contendo uma interface que possibilita a construção das tabelas de modo a ser fácil visualizar todos os dados transmitidos pelas tags. Por isso ele será utilizado para registrar os dados que podem ser salvos e até modificados caso haja necessidade.

8.1 PROGRAMA DE SOFTWARE PLX-DAQ

Esse programa foi o que nos permitiu o a possibilidade de um componente mediador da programação e da escritura dos dados obtidos a partir do circuito.

“PLX-DAQ é uma ferramenta complementar de aquisição de dados por microcontrolador da Parallax para Microsoft Excel. Qualquer um de nossos microcontroladores conectados a qualquer sensor e à porta serial de um PC agora pode enviar dados diretamente para o Excel.”
(Parallax.INC)

Segue abaixo a aparência exemplificada do funcionamento do PLX-DAQ conectada a porta 5do Arduino em funcionamento:

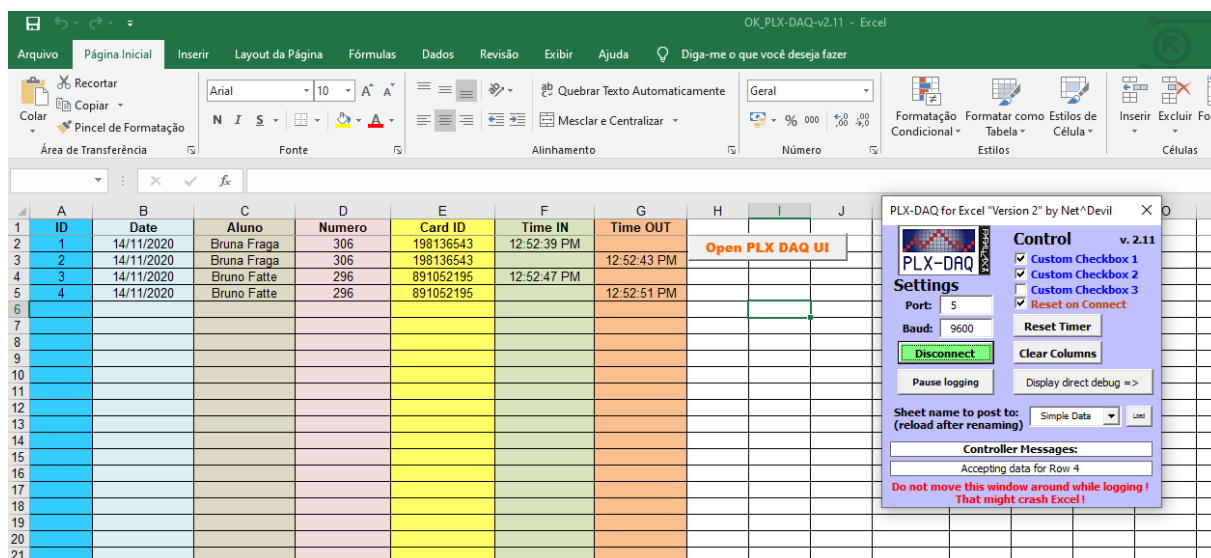


Figura 17 – Captura de Tela da Visualização da Frequência no Excel - (Autor, 2020)

No Arduino o comando a seguir aplica as funções do PLX-DAQ em conjunto com as bibliotecas:

```
void loop() {  
  Serial.println(("LABEL, ID,Date,Name,Number,Card ID,Time IN,Time OUT"));  
  delay(100);  
}
```

- **Id:** Numero de vezes que foi feita a leitura de um cartão.
- **Date:** Data da coleta de dados baseada no dia do ano.
- **Número:** Número de referência da Tag.
- **Card ID:** número do ID do cartão.
- **Time IN:** Horário de entrada (passagem da tag pelo sensor).
- **Time OUT:** Horário de saída (Segunda passagem da mesma tag pelo sensor).

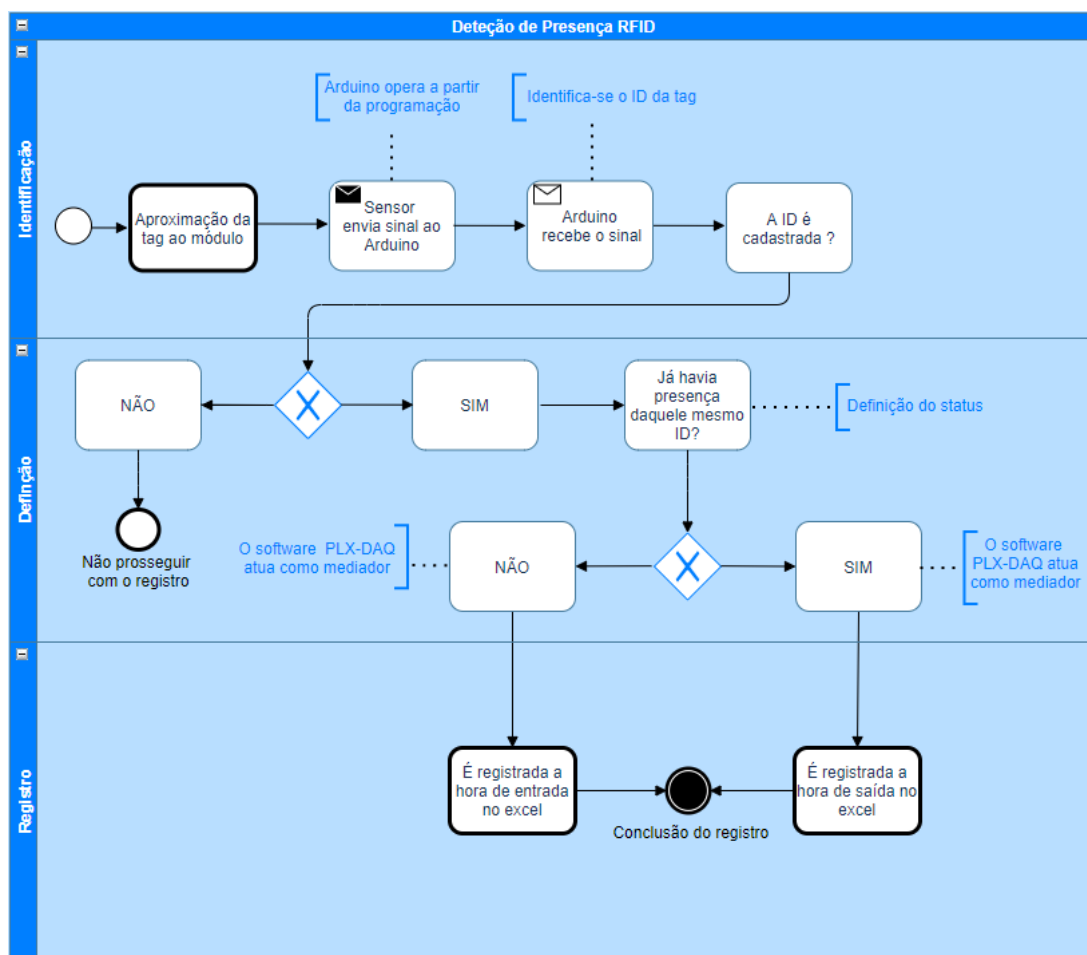
9 CIRCUITO

A seguir veremos o circuito criado virtualmente e mais detalhes sobre cada componente, segue também anexo de figura do circuito físico criado para teste e apresentação do trabalho de conclusão de curso.

9.1 FUNCIONAMENTO BÁSICO

Para melhor compreensão do objetivo geral do projeto o fluxograma nos auxilia a visualizar de forma simplificada os processos que devem acotecer para o funcionamento do circuito, para que posteriormente seja introduzido a estrutura geral do mesmo.

Figura 18 - Fluxograma Geral – Fonte: (Autor, 2020)



O fluxograma é separado em 3 etapas (identificação, definição e registro).

O usuário terá um código pessoal no chip, que será identificado a partir da aproximação da tag pessoal no local de instalação do módulo, onde de acordo com a programação, podem ser tomadas 3 decisões:

1. Se o ID não estava presente, se registra a entrada
2. Se o ID já estava presente, se registra a saída
3. Se o ID não é cadastrado, não é registrado

Posteriormente registrando as informações no Excel (para visualização em planilhas, que será conectada pelo programa mediador PLX-DAQ) as respectivas informações de cada código pessoal (registrando o ID da tag, nome do usuário, hora de entrada e saída e assim a permanência do mesmo).

9.2 ESTRUTURA GERAL

O circuito conta com o módulo RFID que envia as informações ao Arduino que foi conectado com o mesmo segundo pinagem informada pelo fornecedor do MFRC522, e portanto segue operando com a programação já citada, de forma a conectar-se respectivamente com computador, por meio da serial. O circuito também contém 3 Leds:

- Verde (aceso quando o cartão é identificado como previamente cadastrado e registrado, caso fosse acoplado um sistema de abertura de fechadura por exemplo, o led verde seria substituído por aquele responsável por abrir a tranca).

- Vermelho (aceso quando o cartão não é identificado como previamente cadastrado).

- Azul (aceso quando o cartão é aproximado do leitor de forma a expressar resposta de funcionamento a qualquer uma das ações feitas quando o cartão já é definido como previamente cadastrado).

Portanto também são usados 3 resistores para limitar a tensão de cada um dos Leds, utilizamos o protoboard pequeno para realizar as ligações necessárias conforme polarizações e melhor conexão do buzzer que emite os sinais respectivos acompanhando o funcionamento dos Leds.

9.3 ESPECIFICAÇÕES DOS COMPONENTES

Como já foi explicado o Arduino e toda sua configuração, bem como o módulo RFID, não incluiremos esse tópico dentre os componentes seguintes.

Lista de componentes:

Label	Part Type	Properties
Componente1	Arduino Uno (Rev3)	tipo Arduino UNO (Rev3)
Componente2	RFID-RC522	variante variant 1; número do componente 101218
J1	Piezo Speaker	
LED1	Red LED (633nm)	cor Red (633nm); leg yes; pacote 5 mm [THT]
LED2	Blue LED (525nm)	cor Blue (525nm); leg yes; pacote 5 mm [THT]
LED3	Green LED (570nm)	cor Green (570nm); leg yes; pacote 5 mm [THT]
R1	220 Ω Resistor	bands 4; Espaço entre pinos 400 mil; tolerância $\pm 5\%$; resistência 220 Ω ; pacote THT
R2	220 Ω Resistor	bands 4; Espaço entre pinos 400 mil; tolerância $\pm 5\%$; resistência 220 Ω ; pacote THT
R3	220 Ω Resistor	bands 4; Espaço entre pinos 400 mil; tolerância $\pm 5\%$; resistência 220 Ω ; pacote THT

Tabela 5 – Componentes Usados - Fonte: Fritzing, 2020

9.3.1 Piezo speaker – *buzzer*

- Pino positivo – Identificado pela leitura do símbolo positivo (+) ou terminal maior alimentado por 6 V DC
- Pino negativo - Identificado pelo terminal menor (terra)
- Tensão nominal: 6V DC
- Tensão operacional: 4-8V DC
- Corrente nominal: <30mA
- Tipo de som: bipe contínuo

- Frequência de ressonância: ~ 2300 Hz
- Pacote selado pequeno e organizado
- Breadboard e Perf board amigável (Active Passive Buzzer, 2017)

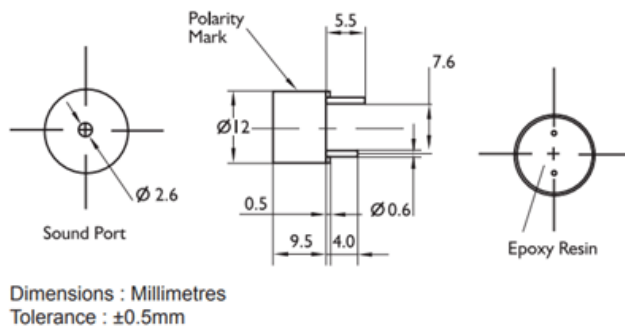


Figura 19 - Dimensões Buzzer – Fonte: Components101 Buzzer, 2020

9.3.2 Leds

- LED de Alto Brilho emissor de luz;
- Componente bipolar (ânodo + e cátodo -);
- Ordem de conexão específica;
- Bulbo incolor;
- Terminal de maior comprimento positivo;
- Terminal de menor comprimento negativo;
- Chip Semicondutor integrado;

Vermelho:

- Tensão de alimentação: 2,0 - 2,2V;
- Corrente máxima: 20mA;
- Resistor usado para limitar corrente: 220 ohms

Azul:

- Tensão de alimentação: 3 a 3.2V;
- Corrente máxima: 30mA;
- Resistor usado para limitar corrente: 220 ohms

Verde:

- Tensão de alimentação: 2 a 2.2V DC;
- Corrente máxima: 20mA;
- Resistor usado para limitar corrente: 220 ohms (USINAINFO, 2020)

10 CONCLUSÃO

Foram feitos, no presente projeto de detecção de frequência via sensor RFID para registro de presença de alunos nas instituições de ensino, diversas pesquisas que acrescentaram muito no conhecimento, além do que já havia sido aprendido em decorrer do curso técnico em eletrônica.

No transcorrer da monografia foram desenvolvidos diversos testes com o Arduino, os quais nos possibilitaram sempre a identificação das operações feitas mediante a resposta física do circuito e suas interações. Por assim dizer o método de ABP (aprendizado baseado em problemas) se aplicou nessa etapa, além de também estar diretamente ligado a complementação de ideais em interdisciplinaridade, trabalhando e aplicando várias áreas do curso em um mesmo projeto e também da habilidade de lidar com os outros integrantes do grupo.

Dentre essa experiência o projeto foi finalizado com sucesso de forma a viabilizar o registro de entrada e saída dos Alunos no Excel com data e hora, cumprindo com todos os objetivos estabelecidos na idealização do mesmo.

Vale ressaltar que também pode ser futuramente integrado ao projeto a função Wireless que envia os dados do Excel para um computador que integre todos os módulos posicionados nas salas a fim de que haja apenas 1 gestão de computador para todo o sistema.

11 REFERÊNCIAS

- ACADÊMICO. Trabalho de Redes de Computadores. **A História do RFID**, 2013. Disponível em:
<https://www.gta.ufrj.br/grad/13_1/rfid/cap2_1.html#:~:text=A%20tecnologia%20de%20identifica%C3%A7%C3%A3o%20por,eletromagnetismo%20de%20James%20Clerk%20Maxwell%2C>. Acesso em: 15 out. 2020.
- ACTIVE Passive Buzzer. **Componentes101**, 2017. Disponível em:
<<https://components101.com/buzzer-pinout-working-datasheet>>. Acesso em: 2 nov. 2020.
- ALCÂNTARA, J. RFID: o que é e como a tecnologia funciona. **Linx**, 2020. Disponível em:
<<https://e-millennium.com.br/rfid-o-que-e-e-como-a-tecnologia-funciona/>>. Acesso em: 31 out. 2020.
- ARDUINO – O que é, tipos e aplicações. **Athos Eletronics**, 2020. Disponível em:
<<https://athoselectronics.com/o-que-e-arduino/>>. Acesso em: 1 nov. 2020.
- ARDUINO. **Wikipédia, a enciclopédia livre**, 2020. Disponível em:
<<https://pt.wikipedia.org/wiki/Arduino>>. Acesso em: 1 nov. 2020.
- ARDUINO UNO REV3. **Arduino.CC store**, 2020. Disponível em:
<<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 1 nov. 2020.
- BASES do RFID. **IFSC**, 2018. Disponível em:
<https://wiki.sj.ifsc.edu.br/index.php/Bases_do_RFID>. Acesso em: 17 out. 2020.
- BOALIM, T. As frequências de operação das etiquetas e leitores RFID. **Moura Informática**, 2015. Disponível em: <<https://rfidmoura.wordpress.com/2015/07/21/as-frequencias-de-operacao-das-etiquetas-e-leitores-rfid/>>. Acesso em: 10 out. 2020.
- CALANDRIELLO, L. O que é MIFARE. **MGITECH**, 2012. Disponível em:
<<https://blog.mgitech.com.br/blog/o-que-mifare>>. Acesso em: 1 out. 2020.
- COMO Funciona - Componentes RFID. **Universidade Federal do Rio de Janeiro**. Disponível em:
<https://www.gta.ufrj.br/grad/07_1/rfid/RFID_arquivos/como%20funciona.htm>. Acesso em: 13 out. 2020.
- DATASHEET MFRC522. **https://blogmasterwalkershop**. Disponível em:
<<https://blogmasterwalkershop.com.br/arquivos/datasheet/Datasheet%20MFRC522.pdf>>.
- DESCONHECIDO. C++. **Origem: Wikipédia, a enciclopédia livre.**, 2020. Disponível em:
<<https://pt.wikipedia.org/wiki/C%2B%2B>>. Acesso em: 2 nov. 2020.
- EDUARDO HENRIQUE FERRONI, H. R. V. J. H. N. R. K. A PLATAFORMA ARDUÍNO E SUAS APLICAÇÕES. **Arquivo PDF**. Disponível em:
<<file:///C:/Users/bruna/Downloads/14354-Texto%20do%20Trabalho-44895-1-10-20180522.pdf>>. Acesso em: 1 nov. 2020.
- ELETRÔNICOS, F. C. Kit Módulo Leitor Rfid Mfrc522 Mifare. **FelipeFlop**, 2020. Disponível em: <<https://www.filipeflop.com/produto/kit-modulo-leitor-rfid-mfrc522-mifare/>>. Acesso em: 1 nov. 2020.
- FATTE, B. **Pesquisas RFID**. Etec Albert Einstein. Sp. 2020.
- GUILHERME DANTAS COUTO E TARSÍUS SÉRGIO MALAFAIA. Departamento de Redes de Computadores. **Universidade Federal do Rio de Janeiro**, 2019. Disponível em:
<<https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/rfid/>>. Acesso em: 20 out. 2020.
- IDENTIFICAÇÃO por radiofrequência. **Wikipédia**, 2013. Disponível em:
<https://pt.wikipedia.org/wiki/Identifica%C3%A7%C3%A3o_por_radiofrequ%C3%Aancia>. Acesso em: 20 out. 2020.

INC, P. PLX-DAQ. **Parallax Inc.** Disponível em: <<https://www.parallax.com/downloads/plx-daq>>. Acesso em: 1 out. 2020.

LEDS. **USINAINFO**, 2020. Disponível em: <<https://www.usinainfo.com.br/>>. Acesso em: 2 nov. 2020.

MODOS de Comunicação. **RFID**. Disponível em: <https://www.gta.ufrj.br/grad/12_1/rfid/links/modos_de_comunicacao.html>. Acesso em: 11 nov. 2020.

OLIVEIRA, E. Como usar com Arduino – KIT RFID MFRC522. **Master Walker**, 2020. Disponível em: <<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-kit-rfid-mfrc522/>>. Acesso em: 1 nov. 2020.

OLIVEIRA, G. Um pouco sobre o Arduino. **Master Walker**, 2020. Disponível em: <<https://blogmasterwalkershop.com.br/arduino/um-pouco-sobre-o-arduino/>>. Acesso em: 1 nov. 2020.

RFID. **Universidade Federal do Rio de Janeiro**. Disponível em: <<https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/rfid/>>. Acesso em: 13 out. 2020.

RFID FAQ. **RFIDSystems**, 2017. Disponível em: <<http://www.rfidsystems.com.br/faq.html>>. Acesso em: 15 out. 2020.

RFID: Princípios Gerais da Tecnologia e Aplicações. **Teleco**, 2020. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialrfid/pagina_2.asp>. Acesso em: 17 out. 2020.

SILVEIRA, G. ENTENDA COMO FUNCIONA O ALCANCE DA TECNOLOGIA RFID. **RFIDBrasil**, 2017. Disponível em: <<https://rfidbrasil.com/blog/alcance-da-tecnologia-rfid/>>. Acesso em: 20 out. 2020.

SOLUÇÃO de RFID para logística interna e extena. **Parson Serviços em Tecnologia**, 2020. Disponível em: <<http://parson.com.br/>>. Acesso em: 15 out. 2020.

SOUZA, F. Arduino UNO. **Embarcados**, 2013. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em: 1 nov. 2020.

SPI library. **Arduino.CC**, 2020. Disponível em: <https://www.arduino.cc/en/Reference/SPI?__cf_chl_jschl_tk__=8fc512b01ac95e0806b9295556a1e84f6b921d2b-1604449494-0-AbN1KEzu5FOsaQ2RODdqLJs1KeCbdF3kblniXwpG4oms4cCQW4Ty3caCp-xAZgwyM8ZDeYx-puISiS4ecUXLevidDH0oAGxQYkfhrX7DeZMu-Rwq-cEtMw5rmDEKLD7aXxsn7x_KiNS4h>. Acesso em: 1 nov. 2020.

12 APÊNDICE

12.1 CÓDIGO DO PROJETO

Segue abaixo o código e a explicação barrada (em verde) ao lado das linhas.

```

/* Layout de pinos usados:
* -----
*           MFRC522      Arduino
*           Leitor/PCD    Uno
* Sinal     Pinos        Pinos
* -----
* RST/Reset  RST          9
* SPI SS     SDA(SS)      10
* SPI MOSI   MOSI         11 / ICSP-4
* SPI MISO   MISO         12 / ICSP-1
* SPI SCK    SCK          13 / ICSP-3

SPI - (Biblioteca) Comunicação SPI (Serial Peripheral Interface) com
Arduino.
A Serial Peripheral Interface é um protocolo de dados seriais
síncronos utilizado em microcontroladores para comunicação
entre o microcontrolador e um ou mais periféricos.
MFRC522 - (Biblioteca) módulo RFID usa o chip MFRC522 da empresa NXP,
que por sua vez, pode ser usado em comunicações a uma frequência
de 13,56MHz, permitindo, por exemplo, sem contato, a leitura e
escrita em cartões que seguem o padrão Mifare
TimeLib - A biblioteca Time adiciona funcionalidade de cronometragem ao
Arduino com ou sem hardware externo de cronometragem. Permite
um esboço para obter a hora e a data como: segundo, minuto,
hora, dia, mês e ano. Ele também fornece o tempo como um time_t C
padrão para que os tempos decorridos possam ser facilmente calculados
e os valores de tempo compartilhados entre diferentes plataformas.
PICC - Proximity Integrated Circuit Chip (cartão ou chaveiro)

*/
#include <SPI.h>           // Inclui a Biblioteca SPI (Serial Peripheral
Interface)- responsável pela comunicação
#include <MFRC522.h>       //Inclui a Biblioteca MFRC522
#include <TimeLib.h>       //Inclui a Biblioteca TimeLib

#define SS_PIN 10          //Define que o Pino Slave Select (SDA-SS) do Modulo
RFID esta conectado ao pino 10 do Arduino
#define RST_PIN 9          //Define que o Pino RST (RESET) do Modulo RFID esta
conectado ao pino 9 do Arduino
#define som 8              //Define que o Pino 8 do Arduino tera o parâmetro
som

MFRC522 mfrc522(SS_PIN, RST_PIN); // Definição do objeto mfrc522 da qual
se passa os parâmetros SS_PIN e RST_PIN
String card_ID="";        //Informa que a variavel card_ID é uma
string (cadeia de caracteres)e atribui o valor vazio

//Adicionar os cartoes abaixo
String Aluno1="891052195"; // Tipo da varivel do Aluno1 é uma
string (cadeia de caracteres)
String Aluno2="198136543"; // Tipo da varivel do Aluno2 é uma
string (cadeia de caracteres)
/*String Aluno3="15353114169";

```

```
String Aluno4="13937143185";  
String Aluno5="79 DD 81 2F";  
String Aluno6="89 48 C9 2F";*/  
  
int NumbCard[2];  
int j=0;  
int statu[2];  
int s=0;  
  
int const LedVermelho=6;  
int const LedVerde=5;  
int const LedAzul=7;  
  
String Aluno;  
long Numero;  
int ID=1;  
  
void setup() {  
    setTime(20,00,00,1,31,2020);           // ajusta a data e hora para 20:00:00,  
31 de janeiro de 2020  
    Serial.begin(9600);                   // Inicia a porta serial com  
velocidade de 9600 bounds  
    SPI.begin();                           // Inicia as vias de comunicação com o  
protocolo SPI  
    mfrc522.PCD_Init();                     // Inicia o Modulo leitor PCD  
(Proximity Coupling Device)  
  
    Serial.println("CLEAR SHEET");          //Limpa a página do Excel  
    Serial.println("LABEL,ID,Date,Aluno,Numero,Card ID,Time IN,Time OUT");  
//Cria as colunas no Excel  
    pinMode(LedVermelho,OUTPUT);           // Declara que o LedVermelho esta  
ligado numa porta Saída do Arduino  
    pinMode(LedVerde,OUTPUT);              // Declara que o LedVerde esta ligado  
numa porta Saída do Arduino  
    pinMode(LedAzul,OUTPUT);               // Declara que o LedAzul esta ligado  
numa porta Saída do Arduino  
    pinMode(som, OUTPUT);                  // Declara que o som esta ligado numa  
porta Saída do Arduino  
    setTime(20,00,00,1,31,2020);           // ajusta a data e hora para 20:00:00,  
31 de janeiro de 2020  
  
    delay(200);                             //Espera 200 milisegundos  
}  
  
void loop() {  
  
    if ( ! mfrc522.PICC_IsNewCardPresent()) {           // Se um novo cartão  
foi apresentado ao leitor RFID, laia e continue  
        return;  
    }  
  
    if ( ! mfrc522.PICC_ReadCardSerial()) {             // Uma vez que um PICC  
colocado, obtenha o Serial e continue  
        return;  
    }  
  
    for (byte i = 0; i < mfrc522.uid.size; i++) {       // Byte = 0 a 255 -  
Loop 4 vezes para obter os 4 bytes (size do cartão)      // incremento de 1  
        (i++)
```

```

    card_ID += mfrc522.uid.uidByte[i];           // Soma o valor da
    variavel card_ID com o valor de i

}

Serial.println(card_ID);                         //imprime na porta
serial as informações contidas no cartão "card_ID"

    if(card_ID==Aluno1){                         //Se a variavel card_ID
é igual ao Aluno1, então o Aluno1 é o           //Bruno Fatte e o
numero dele é 891052195                          //Atribui para a
    Aluno="Bruno Fatte";                         //Atribui para a
    Numero="891052195";
    j=0;
variavel j o valor zero
    s=0;
variavel s o valor zero
    }
    else if(card_ID==Aluno2){                   //Se a variavel card_ID
é igual ao Aluno2, então o Aluno2 é a           //Bruna Fraga e o
numero dela é 198136543                          //Atribui para a
    Aluno="Bruna Fraga";                         //Atribui para a
    Numero="198136543";
    j=1;
variavel j o valor um
    s=1;
variavel s o valor um
    }
    /*else if(card_ID==Aluno3){
        Aluno="Third employee";
        Numero=789101;
        j=2;
        s=2;
    }
    else if(card_ID==Aluno4){
        Aluno="Fourth employee";
        Numero=789101;
        j=3;
        s=3;
    }
    else if(card_ID==Aluno5){
        Aluno="Fiveth employee";
        Numero=789101;
        j=4;
        s=4;
    }
    else if(card_ID==Aluno6){
        Aluno="Sixth employee";
        Numero=789101;
        j=5;
        s=5;
    }
    */
    else{
        digitalWrite(LedAzul,LOW);               //LedAzul desligado
        digitalWrite(LedVerde,LOW);              //LedVerde desligado
        digitalWrite(LedVermelho,HIGH);          //LedVermelho ligado
        tone (som,600,100);                       // ALERTA ID do
cartao DESCONHECIDO - 600hz é a frequencia de saída do som

```



```

// 100 milisegundos
é o tempo de duração do som
    delay(100);
    tone (som,300,100);
    delay(100);
    tone (som,600,100);
    delay(100);
    tone (som,300,100);
    delay(100);
    tone (som,600,300);
    delay(30);

    goto cont; // Pula o
algoritmo para a parte que esta escrito cont
}

    if(NumbCard[j] == 1 && statu[s] == 0){ //Faz a comparação
se j é igual a 1 e s = 0
    statu[s]=1; // atribui o valor 1
para a variavel(s) --> vai imprimir no excel a saída

    Serial.print("DATA,"); // printa na
serial/Excel os dados ID, DATA, Aluno, Numero, card_ID e Horas (saída)
    Serial.print(ID);
    Serial.print(",");
    Serial.print("DATE");
    Serial.print(",");
    Serial.print(Aluno);
    Serial.print(",");
    Serial.print(Numero);
    Serial.print(",");
    Serial.print(card_ID);
    Serial.print(",");
    Serial.print("");
    Serial.print(",");
    Serial.println("TIME");
    ID=ID+1; //Incrementa o valor
do ID com mais 1 --> índice que aparece na coluna ID do excel
    digitalWrite(LedVerde,HIGH); //LedVerde ligado
    digitalWrite(LedVermelho,LOW); //LedVermelho
desligado
    digitalWrite(LedAzul,HIGH); //LedAzul acompanha
o Alerta

    tone (som,1200,100); // ALERTA SAÍDA -
1200hz é a frequencia de saída do som
// 100 é o tempo de
duração do som
    delay(100);
    tone (som,900,100);
    delay(100);
    tone (som,700,100);
    delay(100);
    tone (som,500,100);
    delay(100);
    tone (som,300,300);
    delay(500);
    digitalWrite(LedAzul,LOW); //LedAzul acompanha
o Alerta, desligado
    delay(30);
}

```

```

        if(NumbCard[j] == 0){ //Quando NumbCard[j]
= 0 --> Foi lido o cartão mas não printa, //quando ele vai pra
1 printa entrada //Informa a entrada
        NumbCard[j] = 1; //do cartão --> NumbCard[j] = 1
do cartão --> NumbCard[j] = 1 // (printa o horário
da entrada no excel)

        Serial.print("DATA,"); // printa na
serial/Excel os dados ID, DATA, Aluno, Numero, card_ID e Horas (entrada)
        Serial.print(ID);
        Serial.print(",");
        Serial.print("DATE");
        Serial.print(",");
        Serial.print(Aluno);
        Serial.print(",");
        Serial.print(Numero);
        Serial.print(",");
        Serial.print(card_ID);
        Serial.print(",");
        Serial.print("TIME");
        Serial.print(",");
        Serial.println("");
        digitalWrite(LedVerde,HIGH); //LedVerde ligado
        digitalWrite(LedVermelho,LOW); //LedVermelho
desligado
        digitalWrite(LedAzul,HIGH); //LedAzul acompanha
o Alerta
        tone (som,300,100); // ALERTA ENTRADA -
300hz é a frequencia de saída do som // 100 é o tempo de
duração do som
        delay(100);
        tone (som,500,100);
        delay(100);
        tone (som,700,100);
        delay(100);
        tone (som,800,100);
        delay(100);
        tone (som,1200,300);
        delay(500);
        digitalWrite(LedAzul,LOW);
        delay(30);

        ID=ID+1; // Incrementa o
valor do ID com mais 1 --> índice que aparece na coluna ID do excel
    }

    else if((statu[s] == 1)&&(NumbCard[j] == 1)){ //Se ja foi
verificado que o cartão ja entrou = NumbCard[j] = 1 //e o cartão ja saiu
= statu[s] = 1 //então zera a
entrada = NumbCard[j]=0 //então zera a saída
= statu[s]=0
        statu[s]=0;
        NumbCard[j]=0;

```

```

    }
    delay(1000);           //espera 1000
    milisegundos

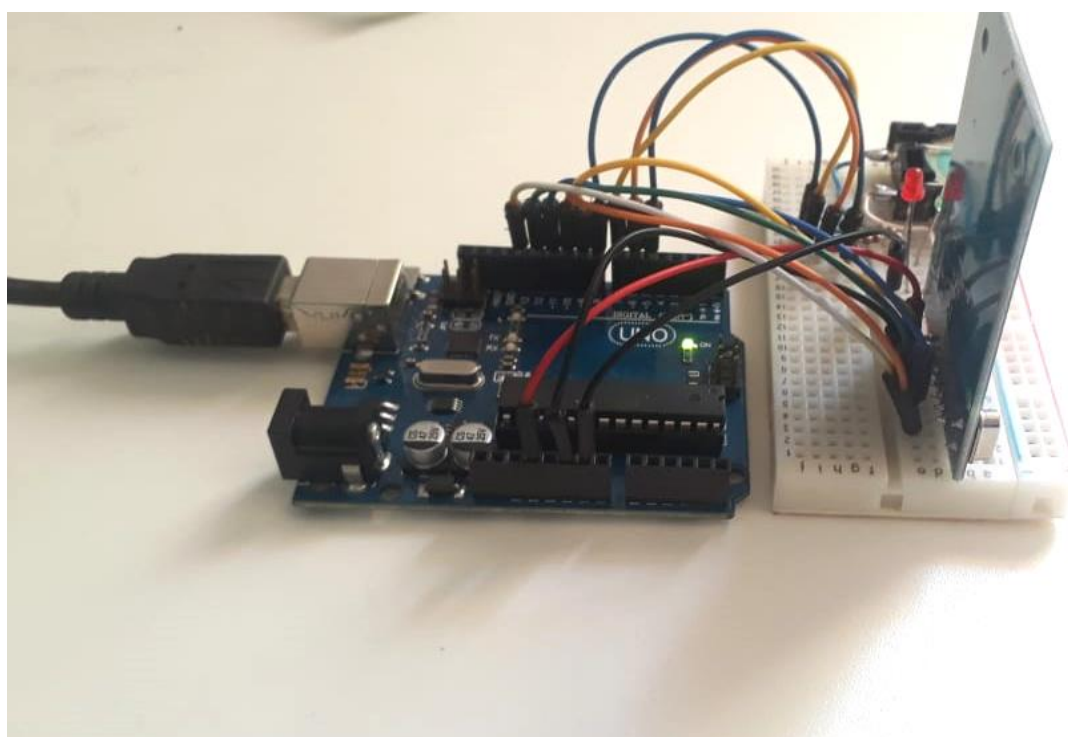
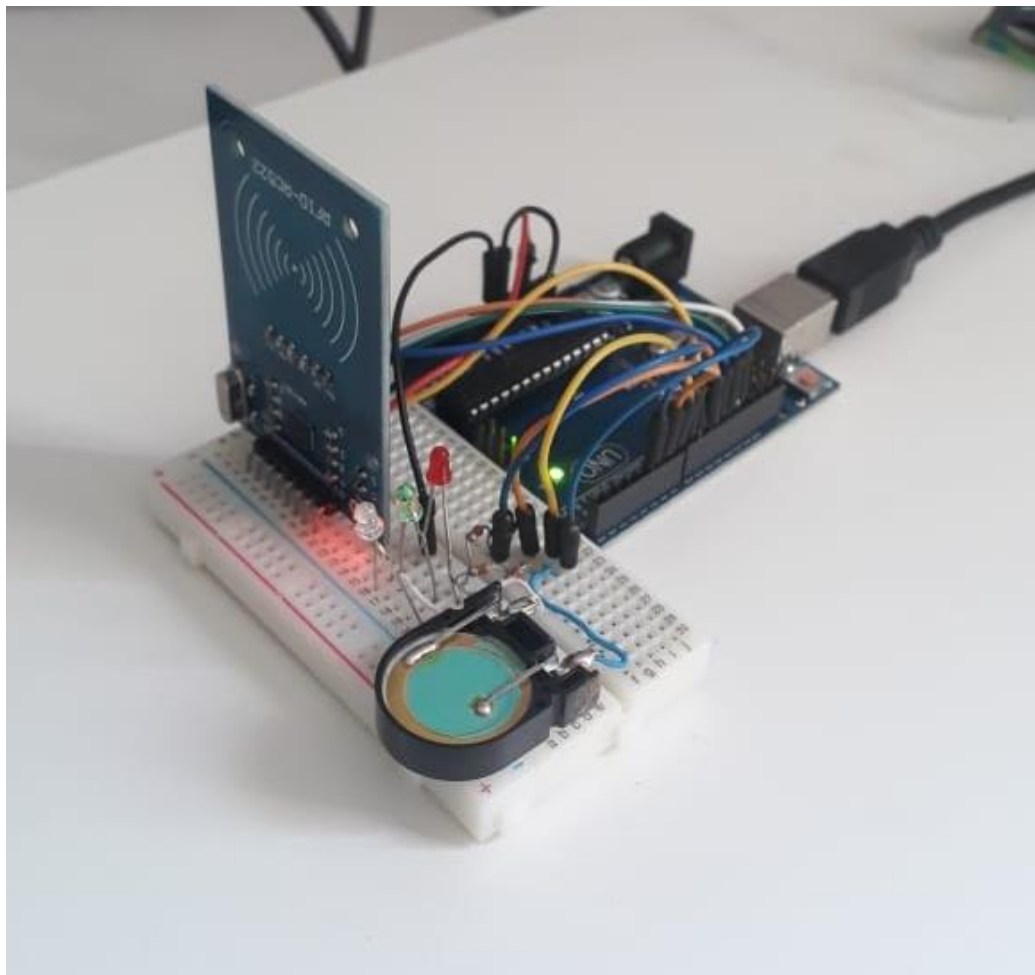
    cont:                  //o algoritimo de
    cima pula pra cá, que é o cont
    delay(2000);           //espera 2000
    milisegundos
    digitalWrite(LedVerde, LOW); //LedVerde desligado
    digitalWrite(LedVermelho, LOW); //LedVermelho
    desligado

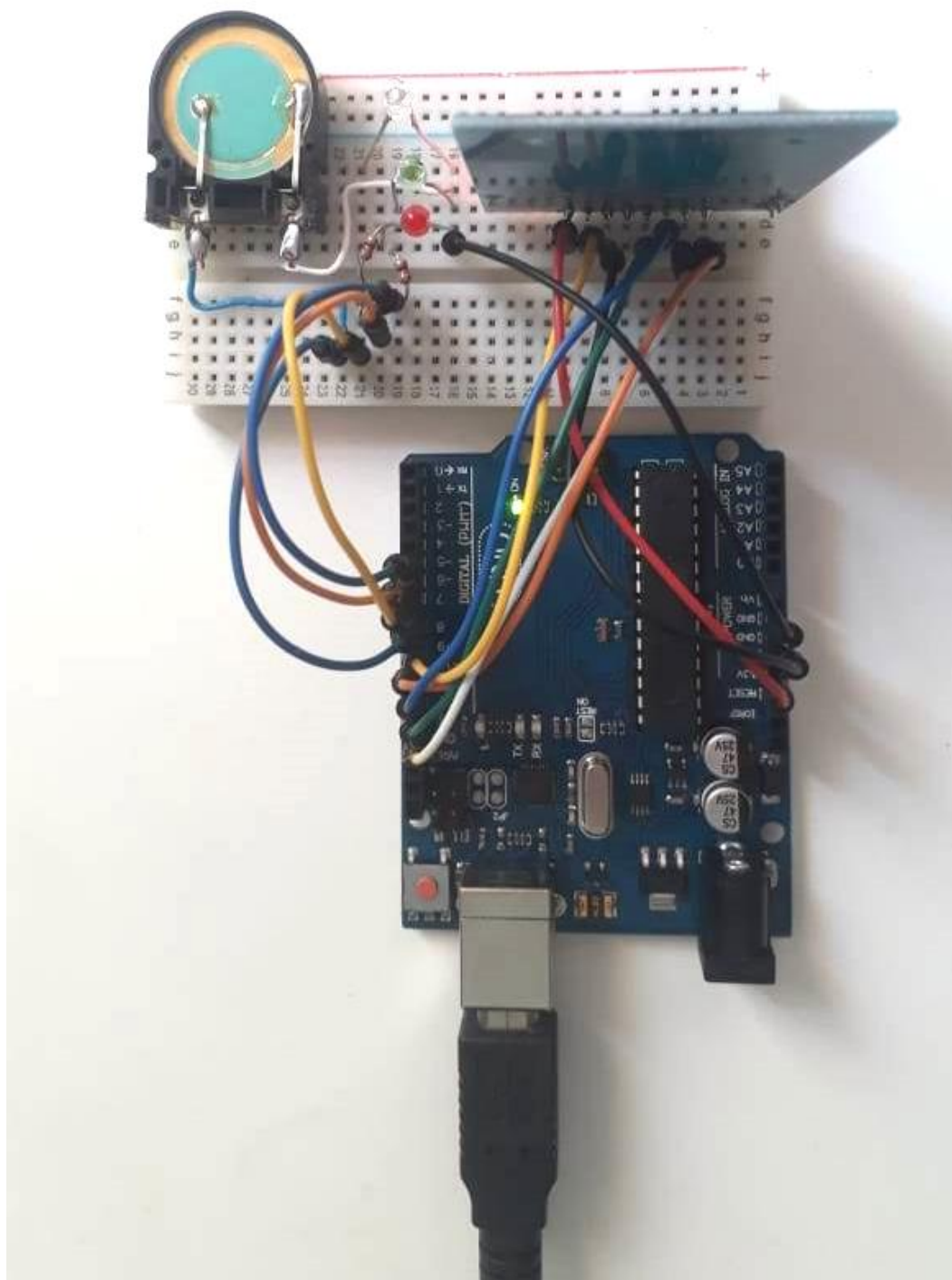
    card_ID="";           //Atribui o valor da
    variavel card_ID para vazio

}
    void printTime() {    //chama a função
printTime
    Serial.print(hour()); //imprime as horas,
    minutos e segundos no excel
    Serial.print(":");
    Serial.print(minute());
    Serial.print(":");
    Serial.print(second());
    Serial.print(",");
}

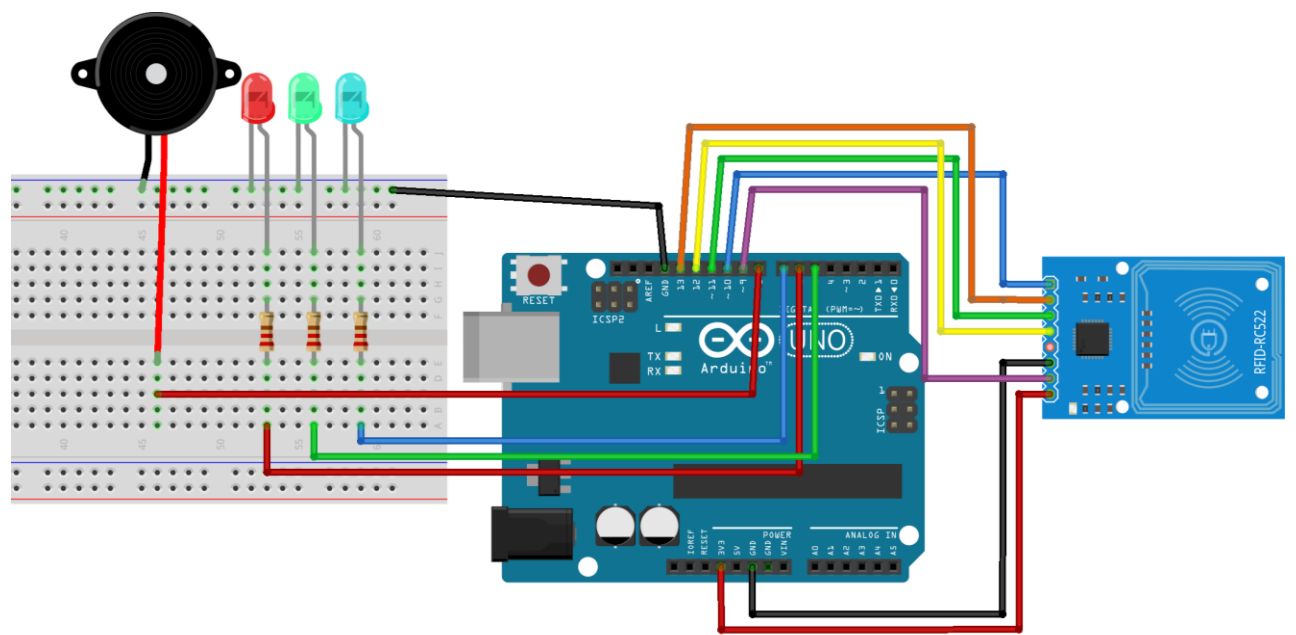
```

12.2 CIRCUITO FÍSICO

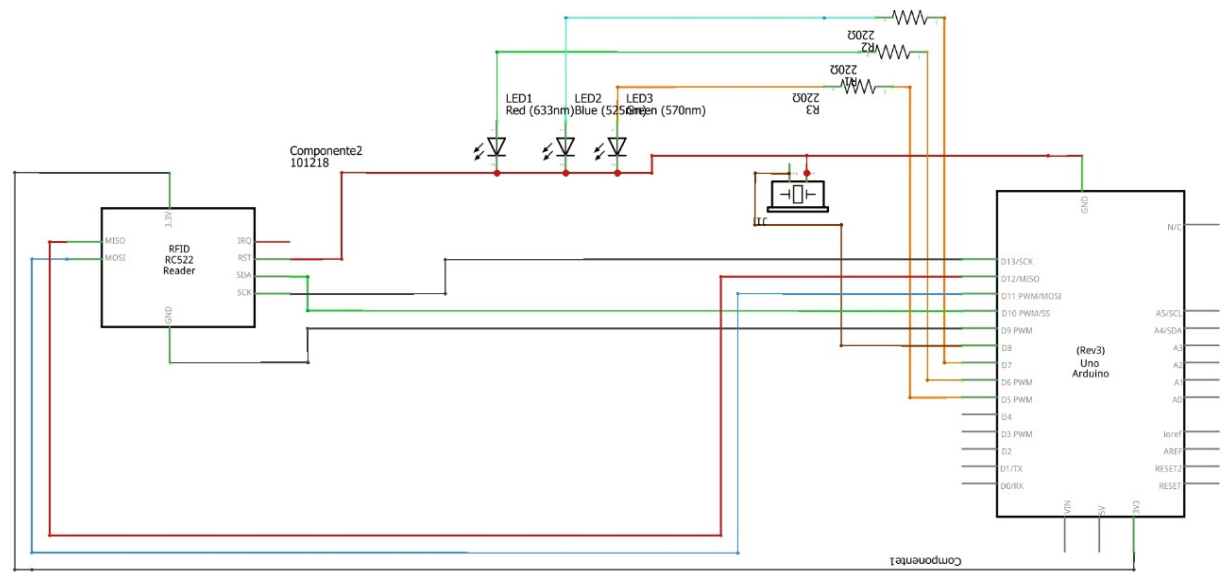




12.3 DIAGRAMAS



fritzing



fritzing

13 ANEXO

Datasheet



Piezoelectronic Buzzers(without circuit) PS Series(Pin Terminal/Lead)

Conformity to RoHS Directive

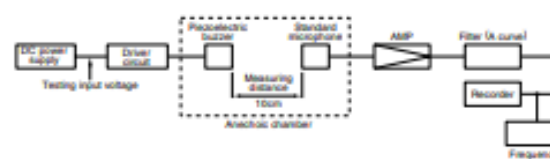
FEATURES

- The PS series are high-performance buzzers that employ unimorph piezoelectric elements and are designed for easy incorporation into various circuits.
- They feature extremely low power consumption in comparison to electromagnetic units.
- Because these buzzers are designed for external excitation, the same part can serve as both a musical tone oscillator and a buzzer.
- They can be used with automated inserters. Moisture-resistant models are also available.
- The lead wire type(PS1550L40N) with both-sided adhesive tape installed easily is prepared.

APPLICATIONS

Electric ranges, washing machines, computer terminals, various devices that require speech synthesis output.

SOUND MEASURING METHOD



SPECIFICATIONS AND CHARACTERISTICS

Type	Part No.	External dimensions		Pitch (mm)	Characteristics		
		Outer diameter (mm)	Height (mm)		Sound pressure (dB(A)/10cm)	Frequency (kHz)	Input voltage (V _{o-p})(Rectangular wave)
PS12 Type	PS1240P02BT	φ12.2	6.5	5	70 min.	4	3
	PS1240P02CT3	φ12.2	3.5	5	60 min.	4	3
PS14 Type	PS1440P02BT	φ14	8	5	75 min.	4	3
	PS1420P02CT	φ14	11	5	70 min.	2	5
PS17 Type	PS1720P02	φ17	8	10	70 min.	2	3
	PS1740P02E	φ17	7.5	10	75 min.	4	3
	PS1740P02CE	φ17	4.6	10	60 min.	4	3
PS19 Type	PS1927P02	φ19	10.5 [excluding terminal]	20	90 min.	2.7	10
	PS1920P02	φ19	10.5 [excluding terminal]	20	80 min.	2	10
Others	PS1550L40N	φ15	1.6	—	Depend on the installation condition		

Type	Part No.	Applications	Features
PS12 Type	PS1240P02BT	For warning and alarm sounds of home appliances(air conditioners, refrigerators, fan forced heaters, cordless telephones, etc.)	• Compact • Automatic mountable • 12.7mm pitch radial taping
	PS1240P02CT3		• Thin type • Automatic mountable • 12.7mm pitch radial taping
PS14 Type	PS1440P02BT		• High sound pressure • Automatic mountable • 15mm pitch radial taping
	PS1420P02CT		• Low frequency tone • Automatic mountable • 15mm pitch radial taping
PS17 Type	PS1720P02	For potted circuit (washing machines, drying machines, hot water supply systems, etc.)	• High sound pressure
	PS1740P02E		• High sound pressure
	PS1740P02CE		• Thin type
PS19 Type	PS1927P02	Digital camera	• High sound pressure • Water-proof processing element
	PS1920P02		• Low frequency tone • Water-proof processing element
Others	PS1550L40N		• Compact, Thin type • Fix in both-sided adhesive tape



MFRC522

Standard performance MIFARE and NTAG frontend

Rev. 3.9 — 27 April 2016
112139

Product data sheet
COMPANY PUBLIC

1. Introduction

This document describes the functionality and electrical specifications of the contactless reader/writer MFRC522.

Remark: The MFRC522 supports all variants of the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus RF identification protocols. To aid readability throughout this data sheet, the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus products and protocols have the generic name MIFARE.

1.1 Differences between version 1.0 and 2.0

The MFRC522 is available in two versions:

- MFRC52201HN1, hereafter referred to version 1.0 and
- MFRC52202HN1, hereafter referred to version 2.0.

The MFRC522 version 2.0 is fully compatible to version 1.0 and offers in addition the following features and improvements:

- Increased stability of the reader IC in rough conditions
- An additional timer prescaler, see [Section 8.5](#).
- A corrected CRC handling when RX Multiple is set to 1

This data sheet version covers both versions of the MFRC522 and describes the differences between the versions if applicable.

2. General description

The MFRC522 is a highly integrated reader/writer IC for contactless communication at 13.56 MHz. The MFRC522 reader supports ISO/IEC 14443 A/MIFARE and NTAG.

The MFRC522's internal transmitter is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443 A/MIFARE cards and transponders without additional active circuitry. The receiver module provides a robust and efficient implementation for demodulating and decoding signals from ISO/IEC 14443 A/MIFARE compatible cards and transponders. The digital module manages the complete ISO/IEC 14443 A framing and error detection (parity and CRC) functionality.

The MFRC522 supports MF1xxS20, MF1xxS70 and MF1xxS50 products. The MFRC522 supports contactless communication and uses MIFARE higher transfer speeds up to 848 kBd in both directions.



The following host interfaces are provided:

- Serial Peripheral Interface (SPI)
- Serial UART (similar to RS232 with voltage levels dependant on pin voltage supply)
- I²C-bus interface

3. Features and benefits

- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers for connecting an antenna with the minimum number of external components
- Supports ISO/IEC 14443 A/MIFARE and NTAG
- Typical operating distance in Read/Write mode up to 50 mm depending on the antenna size and tuning
- Supports MF1xxS20, MF1xxS70 and MF1xxS50 encryption in Read/Write mode
- Supports ISO/IEC 14443 A higher transfer speed communication up to 848 kBd
- Supports MFIN/MFOUT
- Additional internal power supply to the smart card IC connected via MFIN/MFOUT
- Supported host interfaces
 - ◆ SPI up to 10 Mbit/s
 - ◆ I²C-bus interface up to 400 kBd in Fast mode, up to 3400 kBd in High-speed mode
 - ◆ RS232 Serial UART up to 1228.8 kBd, with voltage levels dependant on pin voltage supply
- FIFO buffer handles 64 byte send and receive
- Flexible interrupt modes
- Hard reset with low power function
- Power-down by software mode
- Programmable timer
- Internal oscillator for connection to 27.12 MHz quartz crystal
- 2.5 V to 3.3 V power supply
- CRC coprocessor
- Programmable I/O pins
- Internal self-test

4. Quick reference data

Table 1. Quick reference data

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V _{DDA}	analog supply voltage	V _{DD(PVDD)} ≤ V _{DDA} = V _{DDD} = V _{DD(TVDD)} ; V _{SSA} = V _{SSD} = V _{SS(PVSS)} = V _{SS(TVSS)} = 0 V	[1][2]	2.5	3.3	3.6	V
V _{DDD}	digital supply voltage			2.5	3.3	3.6	V
V _{DD(TVDD)}	TVDD supply voltage			2.5	3.3	3.6	V
V _{DD(PVDD)}	PVDD supply voltage		[3]	1.6	1.8	3.6	V
V _{DD(SVDD)}	SVDD supply voltage	V _{SSA} = V _{SSD} = V _{SS(PVSS)} = V _{SS(TVSS)} = 0 V		1.6	-	3.6	V

Table 1. Quick reference data ...continued

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
I_{pd}	power-down current	$V_{DDA} = V_{DDD} = V_{DD(TVDD)} = V_{DD(PVDD)} = 3\text{ V}$					
		hard power-down; pin NRSTPD set LOW	[4]	-	-	5	μA
		soft power-down; RF level detector on	[4]	-	-	10	μA
I_{DD}	digital supply current	pin DVDD; $V_{DDD} = 3\text{ V}$		-	6.5	9	mA
I_{DDA}	analog supply current	pin AVDD; $V_{DDA} = 3\text{ V}$, CommandReg register's RcvOff bit = 0		-	7	10	mA
		pin AVDD; receiver switched off; $V_{DDA} = 3\text{ V}$, CommandReg register's RcvOff bit = 1		-	3	5	mA
$I_{DD(PVDD)}$	PVDD supply current	pin PVDD	[5]	-	-	40	mA
$I_{DD(TVDD)}$	TVDD supply current	pin TVDD; continuous wave	[6][7][8]	-	60	100	mA
T_{amb}	ambient temperature	HVQFN32		-25	-	+85	$^{\circ}\text{C}$

[1] Supply voltages below 3 V reduce the performance in, for example, the achievable operating distance.

[2] V_{DDA} , V_{DDD} and $V_{DD(TVDD)}$ must always be the same voltage.

[3] $V_{DD(PVDD)}$ must always be the same or lower voltage than V_{DDD} .

[4] I_{pd} is the total current for all supplies.

[5] $I_{DD(PVDD)}$ depends on the overall load at the digital pins.

[6] $I_{DD(TVDD)}$ depends on $V_{DD(TVDD)}$ and the external circuit connected to pins TX1 and TX2.

[7] During typical circuit operation, the overall current is below 100 mA.

[8] Typical value using a complementary driver configuration and an antenna matched to $40\ \Omega$ between pins TX1 and TX2 at 13.56 MHz.

5. Ordering information

Table 2. Ordering information

Type number	Package		
	Name	Description	Version
MFRC52201HN1/TRAYB[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body $5 \times 5 \times 0.85\text{ mm}$	SOT617-1
MFRC52201HN1/TRAYBM[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body $5 \times 5 \times 0.85\text{ mm}$	SOT617-1
MFRC52202HN1/TRAYB[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body $5 \times 5 \times 0.85\text{ mm}$	SOT617-1
MFRC52202HN1/TRAYBM[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body $5 \times 5 \times 0.85\text{ mm}$	SOT617-1

[1] Delivered in one tray.

[2] Delivered in five trays.

6. Block diagram

The analog interface handles the modulation and demodulation of the analog signals.

The contactless UART manages the protocol requirements for the communication protocols in cooperation with the host. The FIFO buffer ensures fast and convenient data transfer to and from the host and the contactless UART and vice versa.

Various host interfaces are implemented to meet different customer requirements.

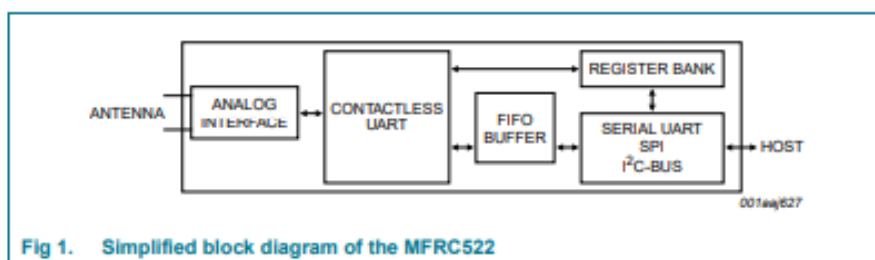


Fig 1. Simplified block diagram of the MFRC522

Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory
 - 256/512/512/1K Bytes EEPROM
 - 512/1K/1K/2K Bytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 4 MHz@1.8 - 5.5V, 0 - 10 MHz@2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



**8-bit AVR[®]
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash**

**ATmega48A
ATmega48PA
ATmega88A
ATmega88PA
ATmega168A
ATmega168PA
ATmega328
ATmega328P**

Summary

Rev. 8271CS-AVR-08/10



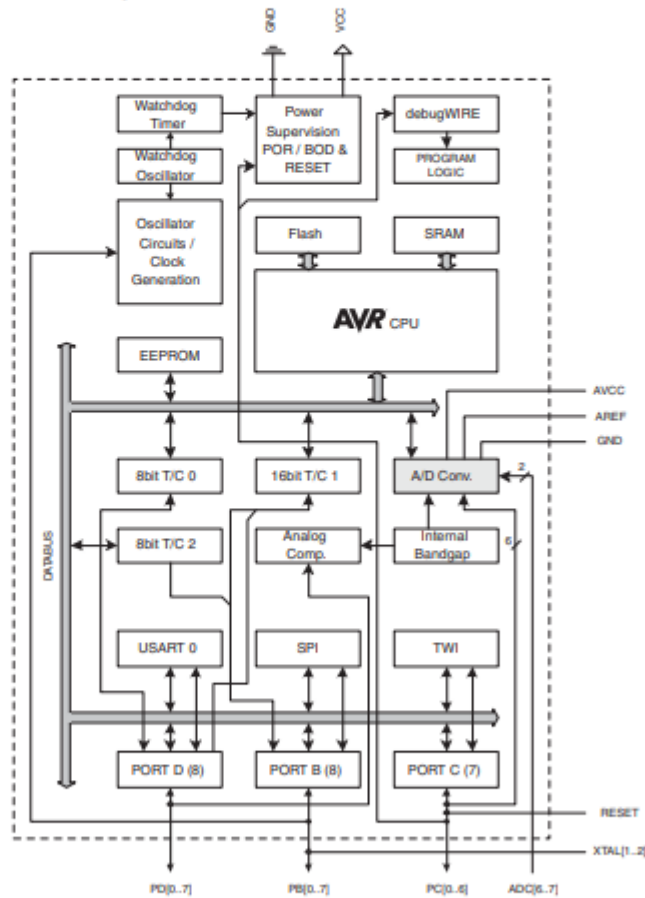
ATmega48A/48PA/88A/88PA/168A/168PA/328/328P

2. Overview

The ATmega48A/48PA/88A/88PA/168A/168PA/328/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48A/48PA/88A/88PA/168A/168PA/328/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 125 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
- Non-volatile Program and Data Memories
 - 8K/16K/32K Bytes of In-System Self-Programmable Flash
 - 512/512/1024 EEPROM
 - 512/512/1024 Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/ 100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
- In-System Programming by on-chip Boot Program hardware-activated after reset
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- USB 2.0 Full-speed Device Module with Interrupt on Transfer Completion
 - Complies fully with Universal Serial Bus Specification REV 2.0
 - 48 MHz PLL for Full-speed Bus Operation : data transfer rates at 12 Mbit/s
 - Fully independent 176 bytes USB DPRAM for endpoint memory allocation
 - Endpoint 0 for Control Transfers: from 8 up to 64-bytes
 - 4 Programmable Endpoints:
 - IN or Out Directions
 - Bulk, Interrupt and Isochronous Transfers
 - Programmable maximum packet size from 8 to 64 bytes
 - Programmable single or double buffer
 - Suspend/Resume Interrupts
 - Microcontroller reset on USB Bus Reset without detach
 - USB Bus Disconnection on Microcontroller Request
- Peripheral Features
 - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode (two 8-bit PWM channels)
 - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Mode (three 8-bit PWM channels)
 - USART with SPI master only mode and hardware flow control (RTS/CTS)
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- On Chip Debug Interface (debugWIRE)
- Special Microcontroller Features
 - Power-On Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 22 Programmable I/O Lines
 - QFN32 (5x5mm) / TQFP32 packages
- Operating Voltages
 - 2.7 - 5.5V
- Operating temperature
 - Industrial (-40°C to +85°C)
- Maximum Frequency
 - 8 MHz at 2.7V - Industrial range
 - 16 MHz at 4.5V - Industrial range

Note: 1. See "Data Retention" on page 6 for details.



8-bit AVR[®]
Microcontroller
 with
8/16/32K Bytes
of ISP Flash
and USB
Controller

ATmega8U2

ATmega16U2

ATmega32U2

7799E-AVR-09/2012

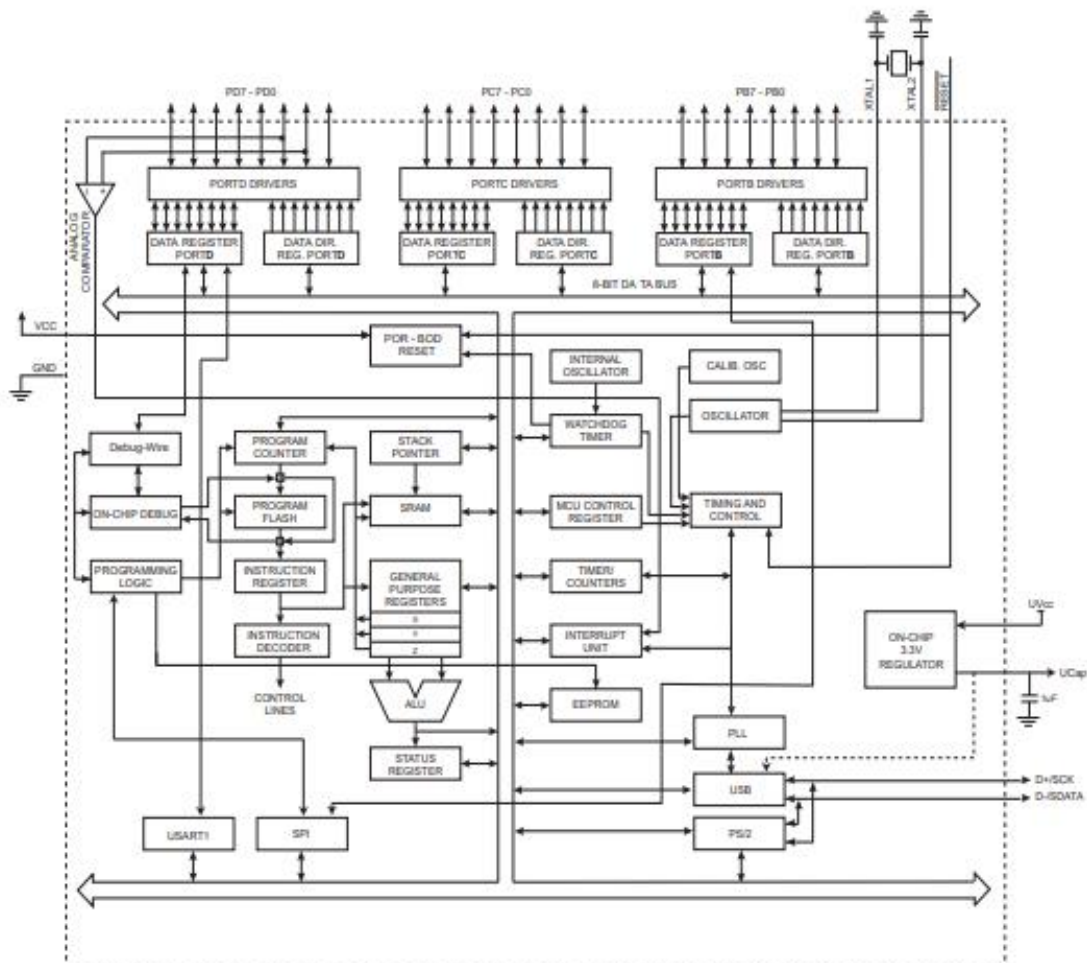


2. Overview

The ATmega8U2/16U2/32U2 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8U2/16U2/32U2 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting