
Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

Controle de fluxo de veículos utilizando sensor RFID

Jamerson Costa Teixeira

Prof. MSc. Diogo Fernando Trevisan(Orientador)

Dourados - MS
2015

Controle de fluxo de veículos utilizando sensor RFID

Jamerson Costa Teixeira

Este exemplar corresponde a redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Jamerson Costa Teixeira e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 23 de novembro de 2015.

Prof. MSc Diogo Fernando Trevisan
(Orientador)

Controle de fluxo de veículos utilizando sensor RFID

Jamerson Costa Teixeira

Novembro de 2015

Banca Examinadora:

- Prof. MSc. Diogo Fernando Trevisan (Orientador)
- Profa. MSc. Jessica Bassani de Oliveira
- Prof. Esp. Alcione Ferreira

Resumo

Este trabalho tem como objetivo o desenvolvimento de um sistema capaz de monitorar e controlar o fluxo de veículos que entram e saem em uma determinada instituição, visando sua inteira segurança, levando em conta o custo, eficácia e sua aplicabilidade. Além disso, pretende-se realizar o monitoramento utilizando sensores RFID sob atuação de tecnologia baseada em radio frequência. Os sensores precisam ser adicionados aos veículos a serem monitorados. O sistema varrerá o ambiente com frequência pré-determinada e irá registrar os dados pertinentes ao veículo, gerando um arquivo CSV com as informações extraídas através da leitura dos sensores.

palavras-chaves: *Sensores RFID, Radio frequência, Monitoração.*

Abstract

This work aims to develop a system to monitor and control the flow of vehicles entering and exiting at a given institution, aiming their entire security, taking into account the cost, effectiveness and applicability. Furthermore, we intend to conduct monitoring using RFID sensors under performance-based technology radio frequency. The sensors need to be added to vehicles to be monitored. The system will scan the environment with predetermined frequency and will record the relevant data to the vehicle, generating a CSV file with the extracted information by reading the sensors.

keyword: *RFID sensors, Radio frequency, Monitoring.*

Agradecimentos

A Deus pela oportunidade da vida e pela força que me destes para prosseguir até o final do meu curso.

A minha esposa Katia Regina Pitanga Vieira Teixeira pelo amor, força, apoio, encorajamento, compreensão e, ainda, por acreditar em minha capacidade de chegar vitorioso ao fim desta longa jornada.

A minha filha Érika Vieira Teixeira pelo amor, incentivo, carinho e compreensão.

Ao meu filho Thiago Vieira Teixeira pelo amor, incentivo, carinho e compreensão.

Ao meu pai Jaldson Lima Teixeira por ter ajudado a forjar o meu senso de honestidade, correção e obediência.

A minha mãe Zely Costa Teixeira “in memoriam” por ter estado presente com o amor, dedicação, amizade e ternura em todos os momentos do desenvolvimento de meu caráter e educação.

Aos meus professores do Curso de Sistemas de Informação por suas orientações em meu desenvolvimento acadêmico.

Ao meu coordenador e orientador Professor e Mestre Diogo Fernando Trevisan pelo apoio, competência e imensa boa vontade em buscar soluções para as dificuldades por que passei para continuar o Curso, quando já estava morando na Cidade de Curitiba-PR.

Aos profissionais que trabalham na Divisão de Estágios Curriculares, da Pró-reitoria de Ensino-UEMS, que foram impecáveis e atenciosos durante o trâmite da documentação de celebração de convênio, uma vez que estava em Curitiba-PR e fui atendido com um elevado grau de competência e profissionalismo.

A Universidade Estadual de Mato Grosso do Sul pela oportunidade de me matricular e de demonstrar que fiz o meu melhor para fazer juz ao ensino que recebi.

Sumário

Resumo	vii
Abstract	ix
Agradecimentos	xi
1 Introdução	1
1.1 Objetivos	1
1.2 Justificativa e Motivação	1
1.3 Metodologia	2
1.4 Organização do Texto	2
1.5 Cronograma	3
2 Fundamentação Teórica	4
2.1 Princípios da comunicação sem fio	4
2.2 Sensor RFID	9
2.2.1 Componentes de um sistema RFID	9
2.3 Frequência de funcionamento	14
2.4 Padrões de Protocolo de Comunicação	15
2.5 Arduíno	18
2.5.1 Características	19
2.6 Arquivo CSV	22
3 Implementação do Hardware	23
3.1 Componentes	24
3.2 Ferramentas de apoio	27
3.3 Conexão Arduíno - Módulo RFID YHY502CTG	28
3.4 Conexão Arduíno - Módulo Cartão SD	29
3.5 Dois módulos conectados ao Arduíno	30
4 Implementação do Software	31
4.1 Interface do Arduino	31
4.2 O código	32

Lista de Siglas

CSV	Comma-separated values
FTDI	Future Technology Devices International
GND	Graduated Neutral Density Filter
IEC	International Electrotechnical Commission
LAN	local Área Network
LED	Light Emitting Diode
MHz	- Mega Hertz
MIFARE	Mikron FARE Collection System
MISO	Master In Slave Out
MOSI	Master Out Slave In
PWM	Pulse-Width Modulation
RFID	Radio-Frequency IDentification
SCK	Serial Clock
SCL	Serial Clock
SDA	Serial Data
TTL	Transistor-Transistor Logic
USB	Universal Serial Bus
VIN	Input Voltage

Lista de Tabelas

1.1	Cronograma.	3
2.1	Frequências e aplicações	15
2.2	Classes de tag reconhecidas pela EPC Global	16
2.3	Normas da ISO referentes à RFID	18
2.4	Tabela com as características básicas do Arduino.	22
3.1	Tabela de ligação - HECKEL (2007)	28
3.2	Quadro de padrão UART de comando	28
3.3	Tabela de ligação - HECKEL (2007)	29
4.1	Comando de leitura	37
4.2	Resposta do módulo RFID	37

Lista de Figuras

2.1	Sinal analógico	5
2.2	Sinal digital	6
2.3	Espectro eletromagnético	7
2.4	Ondas de rádio omnidirecionais	7
2.5	Comunicação a laser	8
2.6	Exemplo de Tag	9
2.7	Modelos de Tag	10
2.8	Componentes físicos de um Leitor	12
2.9	Componentes lógicos de um Leitor	12
2.10	Leitor com antena acoplada	13
2.11	Estrutura de um número EPC	16
2.12	Estrutura de um número EPC	17
2.13	Composição de um número EPC	17
2.14	Circuito com Arduíno	19
2.15	Arduino Uno R3	20
3.1	Módulo RFID YHY502CTG	24
3.2	Arduino Uno R3	24
3.3	Módulo de cartão SD	25
3.4	Protobord	25
3.5	TAG	25
3.6	Cabo flet colorido	26
3.7	cabo-jumper	26
3.8	cabo USB	26
3.9	resistores	27
3.10	Lente de aumento	27
3.11	Mutímetro digital	27
3.12	Circuito-Arduino-Modulo RFID	28
3.13	Circuito Modulo-CartaoSD-Arduino	29
3.14	Circuito-divisor-de-tensao	30
3.15	Ligação dos módulos com o Arduino	30
4.1	Botões do Tolbar	31

4.2	Interface do Arduino	32
4.3	Fluxograma	33
4.4	Arquivo dados.txt	39
4.5	Arquivo saida.txt	40

Capítulo 1

Introdução

Este projeto destina-se a fundamentar e definir os processos e atividades necessárias para o desenvolvimento do Trabalho de Conclusão de Curso do Curso de Graduação em Sistemas de Informação, da Universidade do Estado do Mato Grosso do Sul (UEMS). Os trabalhos foram realizados de forma a obter a solução para o problema proposto, monitoramento do fluxo de veículos utilizando sensores, com o intuito de por em prática o conhecimento adquirido no Curso, desenvolvendo um sistema computacional a fim de produzir um benefício ao meio acadêmico e à sociedade. O trabalho foi desenvolvido visando a melhoria da fiscalização, controle e celeridade no processo de reconhecimento e autorização de veículos no interior de uma organização. Leva-se em consideração o alto custo de implantação de um sistema similar se comparado com o custo do projeto desenvolvido.

1.1 Objetivos

Este trabalho tem como objetivo produzir um sistema capaz de mapear veículos utilizando a tecnologia RFID, com eficácia e eficiência, através de sensores com tag passiva ou ativa, de modo a obter pleno controle de entrada e saída de veículos em uma instituição. Buscar desenvolver o sistema de modo a obter o melhor rendimento com segurança e eficiência.

1.2 Justificativa e Motivação

Este projeto visa o desenvolvimento de um sistema que possa colaborar com a segurança no controle de veículos que entram e saem de uma instituição. Objetos e/ou materiais que, se desviados de seus devidos locais de armazenamento e guarda, podem por em risco a segurança, a saúde e o bem estar da sociedade. A tecnologia computacional é uma das formas que poderemos controlar o fluxo dos veículos e conseqüentemente proporcionar uma importante ferramenta que ajudará a evitar o desvio de armamento, munições, explosivos

e produtos químicos. Um eficiente controle de veículos evitará este desvio e, desta forma, protegerá a instituição e a sociedade como um todo.

1.3 Metodologia

O estudo se divide em duas etapas: na primeira etapa foram realizadas pesquisas bibliográficas em artigos, livros e publicações, pertinentes ao conteúdo abordado, com o intuito de obter o máximo de dados necessários para a construção do referencial teórico e ao desenvolvimento do trabalho. Na segunda etapa foi realizado um estudo das necessidades do equipamento: tipo, potência e custo do leitor, antena e tags (sendo utilizada a frequência UHF), para a implantação do sistema de monitoramento veicular (protótipo) no Corpo da Guarda do Comando da 4ª Brigada de Cavalaria Mecanizada, localizado na Avenida Guicurus, Zona Rural, na Cidade de Dourados-MS .

Será levado em conta o sistema de cadastramento de usuários existente (condição obrigatória definida pela Organização Militar), que é controlado através de emissão de crachás plastificados e que ficam pendurados nos para-brisas dos automóveis e também fixados nas motocicletas. São escalados, diariamente, militares para verificar se os veículos estão regulares e em dia com a autorização de entrada no aquartelamento.

O uso do sistema RFID proporcionará mais dinamismo na verificação dos veículos, aumentando a segurança e contribuindo com um maior fluxo de entrada e saída do aquartelamento.

A utilização do Arduino visa obter um sistema blindado e sem uso de um computador para manter o sistema em funcionamento, uma vez que o mesmo é auto-suficiente. Com o uso de um leitor de cartão SD será possível arquivar os dados lidos das tags, acrescentado com o horário de entrada e/ou de saída. Para o armazenamento dos dados serão usados arquivos CSV.

1.4 Organização do Texto

Este trabalho está organizado como se segue: O primeiro capítulo apresenta uma introdução ao trabalho, os objetivos, justificativas que levaram ao desenvolvimento deste projeto e a metodologia utilizada. No segundo capítulo é apresentado o conteúdo teórico utilizado para o seu desenvolvimento, contendo subsídios para o entendimento e acompanhamento do trabalho executado. No terceiro capítulo será apresentada a implementação do hardware. No quarto capítulo será apresentada a implementação do software. Finalizando, o quinto capítulo apresenta os resultados obtidos.

1.5 Cronograma

Para o desenvolvimento do projeto foi seguido o cronograma apresentado na Tabela 1.1 composto pelas seguintes etapas:

1. Revisão Bibliográfica;
2. Implementação do projeto;
3. Realização de testes;
4. Avaliação dos resultados e escrita da versão final da trabalho;
5. Defesa do Projeto de Final de Curso II.

2015												
Atividade	Jan	Fer	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	x	x	x									
2		x	x	x	x	x	x	x				
3						x	x	x	x			
4							x	x	x	x	x	
5											x	

Tabela 1.1: Cronograma.

Capítulo 2

Fundamentação Teórica

Neste capítulo serão abordados os princípios da comunicação sem fio, a definição e a arquitetura do sistema RFID, sua frequência de funcionamento, os padrões de protocolos de comunicação; definição do Arduino e suas características; definição e exemplo de arquivo CSV e, por fim, a metodologia utilizada para realização deste trabalho.

2.1 Princípios da comunicação sem fio

Os elétrons, ao se moverem, criam ondas eletromagnéticas que podem se propagar pelo espaço livre, até mesmo no vácuo. É chamada frequência, f , o número de oscilações por segundo de uma onda eletromagnética e é medida em Hz. A distância entre dois pontos máximos ou mínimos consecutivos é chamada comprimento de onda, representado pela letra grega λ (lambda)(TANENBAUM (2003)).

Ondas eletromagnéticas podem ser transmitidas e recebidas, quando se instala uma antena com o tamanho apropriado, com eficiência por um receptor localizado a uma distância razoável. Toda a comunicação sem fios é baseada nesse princípio (TANENBAUM (2003)).

A onda de rádio, em cima da qual viaja o sinal a ser transmitido, é chamada de portadora. Pode ser senoidal, para o formato analógico, ou assumir os valores discretos 0 ou 1 no formato digital, conforme (Figura 2.1) e (Figura 2.2) respectivamente.

As ondas possuem três características principais:

- amplitude: amplitude é a medida da altura da onda que indica o máximo valor ou força do sinal no tempo.

- frequência: em eletrônica e telecomunicações, é o número de oscilações por segundo de uma onda eletromagnética, sendo medida em Hertz ($\text{Hz} = 1 \text{ ciclo/s}$).

- fase: “é o ângulo de inflexão da onda em um ponto específico no tempo, é medida em graus” (MATEUS;LOUREIRO apud HECKEL (2007)).

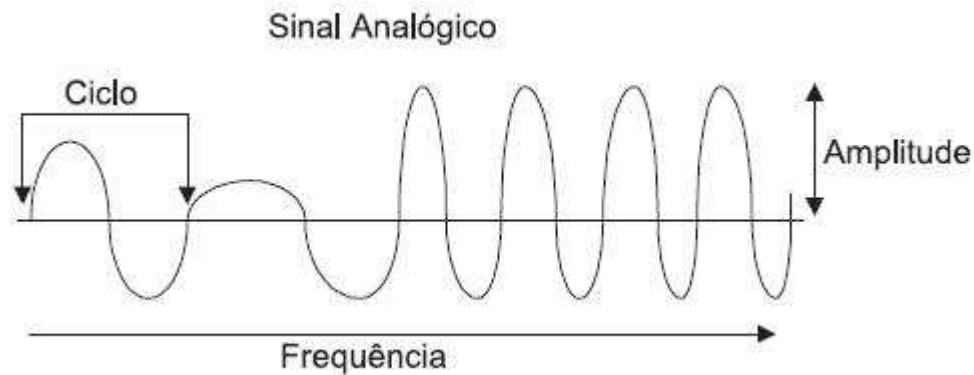


Figura 2.1: Sinal analógico

Fonte: MATEUS; LOUREIRO apud HECKEL (2007).

Outro conceito básico é a modulação:

- seu objetivo é modificar a forma física de um sinal e colocá-lo dentro da faixa de frequência desejada para sua transmissão.

“Modulação é o processo no qual a informação é adicionada às ondas eletromagnéticas. É assim que qualquer tipo de informação, até a voz humana ou transação de dados numa aplicação interativa, é transmitida numa onda eletromagnética.” (PINHEIRO apud HECKEL (2007)).

Ao chegar ao destino essas informações passam pelo processo contrário, chamado de demodulação, a fim de decodificar o que foi transmitido.

Como pode ser observado na (Figura 2.3), do espectro eletromagnético, as frequências mais baixas como as ondas de rádio até a parte da luz visível são as ideais para serem usadas na área de comunicações, pois são possíveis de serem moduladas e não representam perigo para os humanos, diferente das frequências mais altas. (TANENBAUM (2003)).

No vácuo, todas as ondas eletromagnéticas viajam à mesma velocidade, independente de sua frequência. Essa velocidade, geralmente chamada velocidade da luz, é aproximadamente igual a $3 \cdot 10^8$ m/s, ou cerca de 30 cm por nanossegundo. No cobre ou na fibra, a velocidade cai para cerca de $2/3$ desse valor e se torna ligeiramente dependente da frequência. A velocidade da luz é o limite máximo que se pode alcançar. Nenhum objeto ou sinal pode se mover com maior rapidez do que ela (TANENBAUM (2003)).

As porções de rádio, microondas, infravermelho e luz visível do espectro eletromagnético (Figura 2.3) podem ser usadas na transmissão de informações, desde que seja modulada a amplitude, a frequência ou a fase das ondas. A luz ultravioleta, os raios X e os raios gama representariam opções ainda melhores, por terem frequências mais altas, mas são difíceis de produzir e modular, além de serem perigosos para os seres vivos (TANENBAUM (2003)).

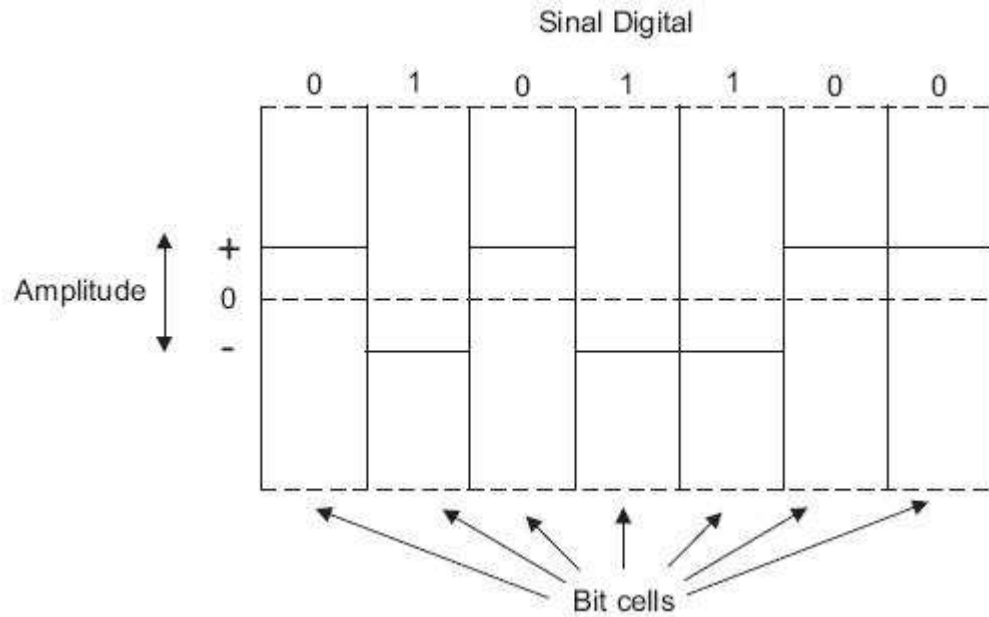


Figura 2.2: Sinal digital

Fonte: MATEUS; LOUREIRO apud HECKEL (2007).

As frequências podem variar da seguinte forma:

- EHF (Extremely High Frequency) - Frequência Extremamente Alta: 30-300 GHz;
- SHF (Super High Frequency) - Frequência Super Alta: 3-30 GHz;
- UHF (Ultra High Frequency) - Frequência Ultra Alta: 0.3-3 GHz;
- VHF (Very High Frequency) - Frequência Muito Alta: 30-300 MHz;
- HF (High Frequency) - Frequência Alta: 3-30 MHz;
- MF (Medium Frequency) - Frequência Média: 0.3-3 MHz;
- LF (Low Frequency) - Frequência Baixa: 30-300 KHz;
- VLF (Very Low Frequency) - Frequência Muito Baixa: 3-30 KHz;
- ELF (Extra Low Frequency) - Frequência Extra Baixa: 3-3000 Hz;
- ULF (Ultra Low Frequency) - Frequência Ultra Baixa: menor que 3 Hz.

O espectro de dispersão de sequência direta, que dispersa o sinal por uma ampla banda de frequências, também está ganhando popularidade no mundo comercial. Em particular, alguns telefones celulares de segunda geração o empregam, e ele se tornará dominante com a terceira geração de telefonia móvel, graças a sua boa eficiência na utilização do espectro e a sua imunidade a ruídos. Algumas LANs sem fios também o utilizam (TANENBAUM (2003)).

Abaixo alguns exemplos de como as diversas partes do espectro eletromagnético são usadas na transmissão sem fio:

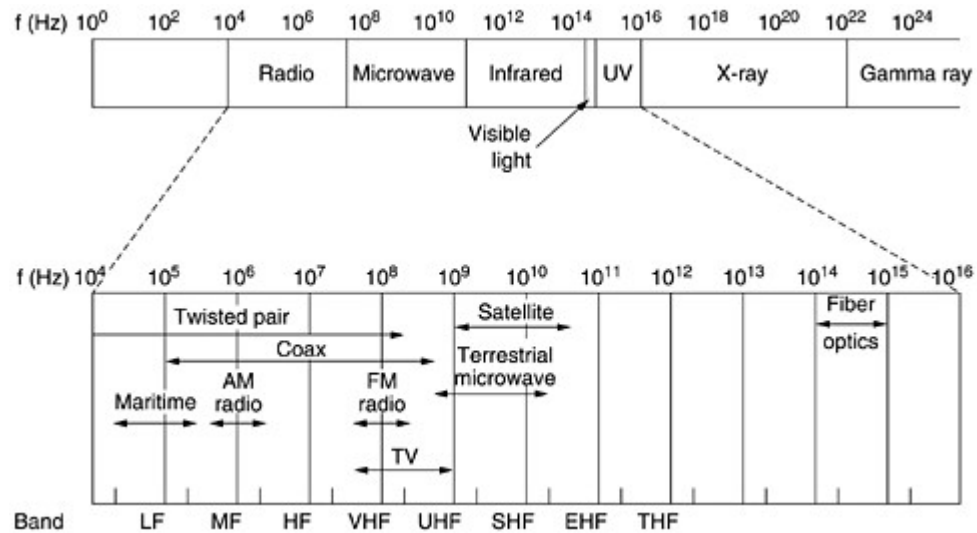


Figura 2.3: Espectro eletromagnético

Fonte: TANENBAUM (2003).

Ondas de rádio:

As ondas de rádio por serem fáceis de gerar, percorrem longas distâncias e penetram facilmente em edificações. São amplamente utilizadas, seja em ambientes fechados ou abertos. Elas viajam em todas as direções a partir da fonte, por este motivo são omnidirecionais, logo, o transmissor e o receptor não precisam estar alinhados (Figura 2.4)(TANENBAUM (2003)).

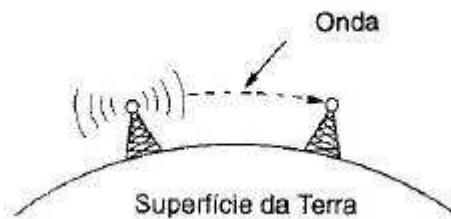


Figura 2.4: Ondas de rádio omnidirecionais

Fonte: TANENBAUM (2003).

Microondas:

Acima de 100 MHz, as ondas trafegam praticamente em linha reta e podem ser concentradas em uma faixa estreita. Esta concentração é possível com o uso de uma antena parabólica, concentrando toda a energia em um pequeno feixe que oferece uma relação sinal/ruído muito mais alta, desde que as antenas de transmissão e recepção estejam alinhadas com o máximo de precisão (TANENBAUM (2003)).

Um problema com as microondas é que viajando em linha reta e com as torres muito

afastadas, a Terra acabará ficando entre elas. Torna-se necessário instalar repetidores a intervalos periódicos. A distância entre os repetidores aumenta de acordo com a raiz quadrada da altura da torre. Torres com 100 m de altura devem ter repetidores a cada 80 km, logo, quanto mais altas são as torres, mais distantes elas podem estar umas das outras (TANENBAUM (2003)).

Infravermelho

São extensamente utilizadas na comunicação de curto alcance. Os controles remotos utilizados nos aparelhos eletrônicos empregam a comunicação por infravermelho. São econômicos, fáceis de montar e relativamente direcionais. Apresentam uma desvantagem: não atravessam objetos sólidos. Em geral, quando nós deslocamos o radio de onda longa em direção a luz visível, perdem sua eficiência por suas ondas assumirem um comportamento cada vez mais parecido com o da luz (TANENBAUM (2003)).

Laser ou Free Space Optics

Por ser unidirecional a comunicação por luz visível ou laser possui as vantagens de ter uma grande largura de banda a um custo baixo e não precisar de regulamentação. O laser pode ser utilizado tanto no espaço livre como nas fibras óticas. O espaço livre é usado como meio de transmissão pelas redes óticas sem fio, o que as torna um dos principais avanços do mercado. Podemos conectar as LANs em dois prédios por meio do laser instalado em seus telhados (Figura 2.5), desde que cada prédio possua seu próprio raio laser e do seu próprio fotodetector (TANENBAUM (2003)).

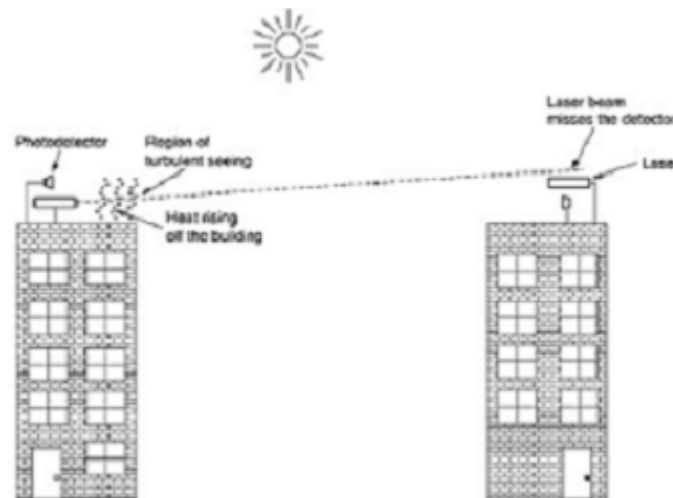


Figura 2.5: Comunicação a laser
Fonte: TANENBAUM (2003).

2.2 Sensor RFID

Sensor é um dispositivo sensível à alguma forma de energia como luz, calor, movimento, que encontram-se no meio ambiente, atribuindo-lhes um valor conforme a grandeza envolvida (WENDLING (2010)).

[...] A identificação por radiofrequência (RFID) é uma tecnologia capaz de captar, gerenciar, analisar e responder aos dados provenientes de sensores eletrônicos. [...] RFID é uma tecnologia de identificação que utiliza a radiofrequência para capturar os dados, permitindo que uma etiqueta RFID seja lida sem a necessidade de contato ou campo visual, através de barreiras e objetos tais como madeira, plástico, papel, entre outros (GREFF (2009), p. 20).

2.2.1 Componentes de um sistema RFID

A arquitetura RFID é dividida em quatro componentes básicos: tag, leitor, antena e *middleware* (TANENBAUM (2003)).

Tag

A tag é formada basicamente por um microchip de silício e uma antena, conforme (Figura 2.6). Também é chamada de transponder, identificador ou etiqueta RFID.



Figura 2.6: Exemplo de Tag

Fonte: (SANTINI apud TEIXEIRA (2011), p. 21)

Normalmente, as tags estão envoltas em plástico, podendo ser encapsuladas em diversos formatos, como mostra a (Figura 2.7). A escolha do formato é fundamental, levando-se em conta a durabilidade, resistência a mudanças de temperatura, acessibilidade e, ainda, a utilização das tags (HECKEL (2007)).

O objetivo de uma tag é identificar o ser vivo, objeto, ou local ao qual está anexada devido ao seu código de identificação, o EPC. Há vários tipos de tags disponíveis no mercado, logo, as tags podem ser classificadas pelo uso de bateria, forma de encapsulamento, frequência, acoplamento e capacidade de armazenamento.



Figura 2.7: Modelos de Tag

Fonte: (RFIDBr (2013) - Portal Brasileiro sobre RFID)

Tag Passiva

A tag passiva contém, normalmente, memória do tipo *Read Only Memory (ROM)* e apenas responde ao sinal emitido pela antena ligada ao leitor. Ela opera sem bateria, sendo que sua alimentação é fornecida pelo próprio leitor através de ondas eletromagnéticas. Este tipo de tag possui alcance médio menor e durabilidade teoricamente infinita, uma vez que sua vida útil só tem como fator limitante o seu bom uso (FAH apud TEIXEIRA (2011), p.22).

Por trabalhar numa frequência mais baixa, essas tags são mais suscetíveis a ruídos e a perdas de sinal em relação a fatores climáticos e barreiras. Contudo, o custo dos modelos passivos é bem inferior, e tem uma vida útil bem mais elevada, se comparado aos modelos ativos (TEIXEIRA (2011)).

Tag Semi-passiva

A tag semi-passiva é muito similar à passiva, porém, incorpora uma pequena bateria que permite que o circuito integrado de leitura seja constantemente alimentado; e tem um tempo de resposta mais rápido, pois é mais potente em seu raio de leitura. Este tipo de tag não possui um transmissor ativo, fato este que o diferencia das tags ativas.

Este tipo de tag é utilizado em sistemas de tempo real para rastreamento de materiais de alto valor ou equipamentos dentro de uma fábrica. Outra aplicação da tag semi-passiva é nos sensores de controle de temperatura, pressão, umidade relativa do ar, aceleração, vibração, movimento e altitude em produtos que exijam esse monitoramento. Ela possui

melhor capacidade de leitura quando anexada à materiais opacos e absorventes (GREFF (2009)).

Tag Ativa

Por ser alimentada por uma bateria interna, seu tempo de vida é limitado. Tem a característica de transmitir o próprio sinal, operando em altas frequências. Uma vez que seu raio de alcance é maior, não é necessário a utilização de várias antenas para cobrir um grande espaço (TEIXEIRA (2011)).

Tem como principais vantagens: realizar processos de escrita e leitura, maior capacidade de memória e tolerância a ruídos e a perdas de sinal. Seu alto custo, em relação às tags passivas, tamanho e tempo de vida finito da bateria, são suas grandes desvantagens.

Leitor

Os leitores emitem ondas de radiofrequência para alimentar as tags, que por sua vez retornam as informações solicitadas. Quando a tag passa pela área de cobertura da antena, o campo magnético é detectado pelo leitor, que decodifica os dados codificados na tag, passando-os para o *middleware* realizar o processamento. A comunicação de dados entre tags e leitores é realizada sem contato físico.

[...]O leitor, também chamado de transceiver, reader ou interrogador, é o componente de comunicação entre o sistema de RFID e os sistemas externos de processamento de informação. A complexidade dos leitores depende do tipo de tag e das funções a serem aplicadas. Os leitores mais sofisticados apresentam funções de verificação de paridade de erro e correção de dados (TEIXEIRA (2011)).

Os leitores possuem duas estruturas gerais: a parte física (Figura 2.8) e a lógica (Figura 2.9) (SANTINI apud HECKEL (2007), p.53).

A parte física é composta pela antena (que não precisa fazer parte do corpo do leitor), pelo controlador que tem como função gerenciar os protocolos de transmissão do identificador. Caso uma informação coletada seja fundamental, o controlador é quem tem a função de analisar e enviar essa mensagem para o próximo componente, a interface de rede. Os leitores externalizam as informações através da interface de rede, por uma interface serial, rede ou até mesmo via *wireless*, recebidas pelo leitor para serem tratadas (HECKEL (2007)).

A parte lógica é definida pelos componentes abaixo descritos (HECKEL (2007)):

- API (*Application Programming Interface*): é responsável por controlar os níveis de bateria das tags e realizar a transição de informações entre o *middleware* e o leitor (SANTINI apud HECKEL (2007)).

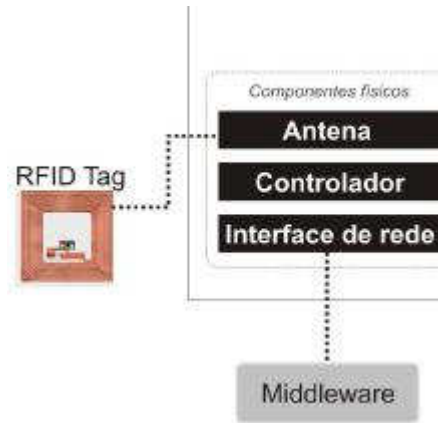


Figura 2.8: Componentes físicos de um Leitor
 Fonte: SANTINI apud HECKEL (2007)

- Parte de comunicações: controla o protocolo de comunicação entre o *middleware* e o leitor. Gerencia a transmissão das informações que caracterizam a API.
- Gerenciador de eventos: é quem identifica se uma tag no espaço de interrogação de um leitor já foi observada ou se é considerada um novo evento. É este componente que avalia a importância de determinado evento e se é relevante para ser transmitido pela interface de rede.
- Parte lógica da antena: é quem controla a comunicação da antena física para com as tags. (GLOVER; BHATT apud HECKEL (2007), p. 54).



Figura 2.9: Componentes lógicos de um Leitor
 Fonte: (SANTINI apud HECKEL (2007), p. 54)

Antena

A antena define como o campo eletromagnético será gerado, realizando a troca de informações entre o transceiver e o transponder. O leitor e as tags devem possuir uma antena para poder realizar a troca de informações. Como a antena não precisa estar no mesmo invólucro do transceiver, pode ser considerada como um elemento a parte nos sistemas de RFID (GLOVER; BHATT apud HECKEL (2007)).

A comunicação entre os leitores RFID e as tags, é realizada através das antenas presentes nos dois componentes. Dois dos fatores que determinam o alcance e a área de cobertura das antenas utilizadas são a potência e a forma de propagação do sinal emitido pela antena. Quanto maior é a potência maior é o alcance/área de cobertura. As formas de propagação mais utilizadas pela tecnologia RFID são circular e linear. A circular realiza uma leitura do ambiente mais abrangente, gerando um campo circular ao seu redor. Já a linear, a leitura é localizada, apenas em uma direção. (BHUPTANI; MORADPOUR apud SENAI-SC (2009)).

Nas aplicações que exigem uma maior mobilidade, as antenas são acopladas aos transceivers, conforme (Figura 2.10), mas isso não é uma regra (HECKEL (2007)).



Figura 2.10: Leitor com antena acoplada
Fonte: (SANTINI apud HECKEL (2007), p. 54)

Middleware

É o dispositivo de interface que controla todo o sistema periférico de RFID (leitor e tags) além da comunicação com os outros componentes do sistema. Por rodar em segundo plano passa despercebido, mas foi desenvolvido para realizar a integração entre as diversas aplicações de RFID. O *middleware* é o responsável por filtrar o grande número de dados coletados pelos leitores, pela depuração das informações recebidas pelas antenas e por converter essas informações em informação útil para atender a demanda que o sistema do usuário necessita (TEIXEIRA (2011)).

Há três motivos para se utilizar *middleware* RFID: encapsular as aplicações das interfaces de dispositivos; processar as informações brutas capturadas pelos leitores de modo que as aplicações só vejam eventos significativos; e para obter uma interface em nível de aplicação para gerenciar leitores e consultar observações do sistema de RFID (BHATT (2007)).

O desenvolvimento do *middleware* varia de acordo com o *hardware* de cada fabricante e exige um alto grau de conhecimento técnico, pois a maioria dos leitores simplesmente capta todos os dados que estão na sua área de interrogação e cabe ao *middleware* organizar esses dados e os transformar em informações (TEIXEIRA (2011)).

2.3 Frequência de funcionamento

Fator fundamental na adoção de um sistema RFID. São regulamentadas como dispositivos de rádio, uma vez que os sistemas RFID transmitirem ondas eletromagnéticas. Sistemas RFID não devem interferir com outras aplicações protegidas, como rádios de serviços de emergência ou transmissões de televisão (BHATT (2007)).

Diferentes frequências são usadas em diferentes aplicações (Tabela 2.1), as principais são:

- LF (Low Frequency): as frequências mais usadas nessa faixa são de 125 e 134.2 KHz, não sendo necessária a regulamentação e penetram na maioria dos materiais, tais como metais, água ou, ainda, no corpo humano. Seu uso em ambientes industriais é limitado por motores elétricos que podem interferir com essa frequência (MOROZ apud HECKEL (2007), p. 46).

- HF (*High Frequency*): As tags de RFID HF operam tipicamente em 13.56 MHz. Transmitem dados mais rapidamente do que sistemas LF, mas consomem mais bateria. Essa frequência também é aceita mundialmente, contudo há algumas diferenças de regulamentação que variam em alguns países quanto à potência das antenas dos leitores. (MOROZ apud HECKEL (2007), p. 46). É mais suscetível à interferências quando os tags RFID estão próximos a metais, porém seu custo é inferior às etiquetas que operam em LF. Exemplo de aplicação: *Smart Cards*.

- UHF (*Ultra High Frequency*): A faixa UHF (300MHz até 3 GHz), é utilizada quando os leitores precisam ler tags a uma distância maior do que as obtidas pelas faixas LF e HF. Essas tags também são facilmente acopladas em diversos tipos de materiais. O processo de fabricação das etiquetas em UHF é relativamente mais simples, contribuindo para baixar o custo dessas tags. Aplicação: identificação de caixas, por exemplo (HECKEL (2007), p. 46).

- Microondas: Frequências acima de 3GHz, as microondas alcançam distâncias maiores e são ideais para a identificação de veículos.

Fonte: (BHATT (2007))

Tabela 2.1: Frequências e aplicações

Nome	Faixa de frequência	Distância Máxima para IDs Passivos	Aplicações Típicas
LF	30-300 KHz	50 centímetros	Leituras em metais, animais e próximas a itens com água
HF	3-30 MHz	3 metros	Controle de acesso
UHF	300 MHz - 3 GHz	9 metros	Caixas e caixotes
Microondas	> 3 GHz	> 10 metros	Identificação de veículos

A faixa de operação RFID, no padrão brasileiro UHF, sintoniza as faixas de 902 a 907,5MHz e de 915 a 928MHz. Na banda intermediária entre 907,5MHz e 915MHz, é uma faixa de caráter secundário na qual o sistema RFID não poderá operar, sendo passível de multa a sua utilização, por parte do órgão de regulamentação local, a Agência Nacional de Telecomunicações (ANATEL). Todo equipamento que trabalhe no Brasil na faixa UHF tem que usar essa faixa de frequência, com potência máxima de saída de até 30 dBm (1W) para que esteja de acordo com a regulamentação nacional vigente (COSTA (2013)).

2.4 Padrões de Protocolo de Comunicação

A necessidade de interoperabilidade de compatibilidade entre diferentes sistemas RFID foi determinante para que fossem adotados padrões para se trabalhar com identificadores de radiofrequência. Os principais identificadores existentes seguem os padrões *Electronic Product Code* (EPC) ou *International Organization for Standardization* (ISO), e cada um deles trabalha com diferentes leitores, o que torna fundamental conhecer os diferentes padrões com o intuito de melhor escolher o leitor conforme cada aplicação (HECKEL (2007); TEIXEIRA (2011)).

A finalidade da padronização e de normas é definir as plataformas em que uma indústria possa operar de forma eficiente e segura. Os maiores fabricantes de RFID oferecem sistemas proprietários, o que resulta numa diversidade de protocolos de sistemas de RFID numa mesma planta industrial (MARTINS (2013)).

EPC

Um EPC estabelece um número único para determinado produto, similar a um *MAC Address* de uma placa de rede.

Com a padronização e a utilização do Código Eletrônico do Produto (EPC), através do órgão regulador internacional EPCglobal, representado no Brasil pela GS1 e responsável pela normalização da informação contida nas tags de RFID, a disseminação dos sistemas de RFID está se tornando realidade e, por isso, a sua padronização mais urgente (COSTA (2013)).

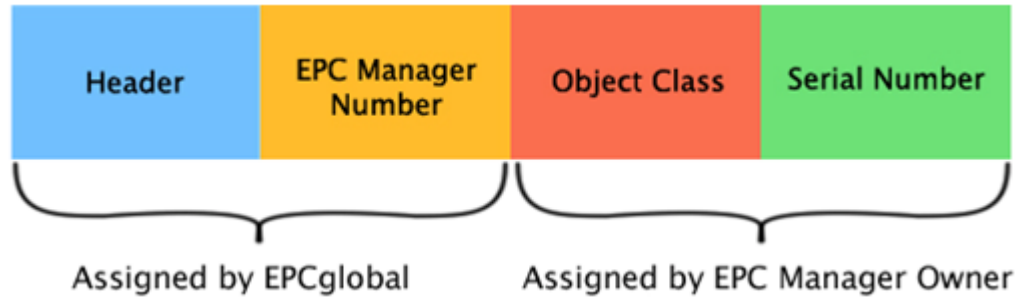


Figura 2.11: Estrutura de um número EPC

Fonte: (EPC-RFID (2013))

O campo *header* (cabeçalho) indica o comprimento, tipo, estrutura, versão e geração do EPC; o campo *EPC Manager Number* (número do gerenciador EPC) é a entidade responsável por manter as partições subsequentes; a *object class* (classe do objeto) identifica a classe do objeto ao qual a tag está anexada (Tabela 2.2); e o *serial number* (número de série) identifica a instância (Figura 2.11).

Tabela 2.2: Classes de tag reconhecidas pela EPC Global

Classe	Descrição
0	Passivas, apenas de leitura
0+	Passiva, grava uma vez, mas usando protocolos da classe 0
I	Passiva, grava uma vez
II	Passiva, grava uma vez com extras, como criptografia
III	Regravável, semi-passiva (chip com bateria, comunicações com energia do leitor), sensores integrados
IV	Regravável, ativa, identificadores podem conversar com outros identificadores, energizando suas próprias comunicações
V	Podem energizar e ler identificadores das Classes I, II e III e ler identificadores das Classes IV e V, assim como atuar como identificadores da classe IV

Fonte: (EPC-RFID (2013))

O EPC que atende às necessidades atuais de mercado é composto de 96 bits, sendo os 8 primeiros bits o cabeçalho, os 28 bits seguintes especificam o fabricante daquele determinado EPC, os próximos 24 bits revelam qual é o tipo do objeto e os últimos 36 bits indicam é o número de série daquele objeto específico (Figura 2.12) (MONTEIRO (2013)).

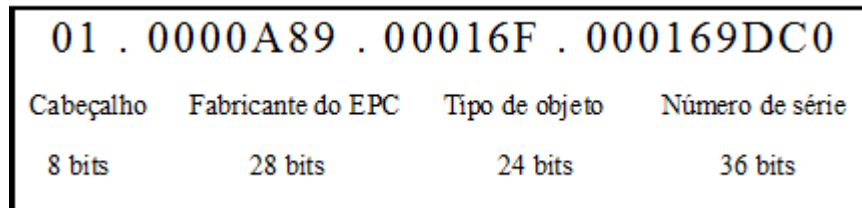


Figura 2.12: Estrutura de um número EPC
Fonte: MONTEIRO (2013)

A transmissão de dados, armazenados em uma tag, deve ser seguro e isento de erros. Para garantir esta integridade é utilizado o CRC (*Cyclic Redundancy Check*) ou verificação cíclica de redundância, que consiste em uma forma de detectar erros de transmissão ou armazenamento (Figura 2.13). O CRC é enviado com a mensagem original e verificado após ser transmitido a fim de comprovar a integridade dos dados. O EPC é o próprio código da tag e o *password* (senha) é o campo responsável por desabilitar a tag. O uso da senha destruirá permanentemente a tag (GLOVER; BHATT apud HECKEL (2007), p. 61).



Figura 2.13: Composição de um número EPC
Fonte: EPC-RFID (2013)

ISO

Como os primeiros sistemas de RFID eram utilizados somente para controle interno, não havia preocupações em relação a uma padronização para a tecnologia. O padrão ISO foi originalmente concebido para uso agrícola para identificar o equipamento agrícola e pecuária (TEIXEIRA (2011); RFID-NEWS (2013)).

Em fevereiro de 2005, a especificação EPC foi submetida a ISO (Tabela 2.3), esperando-se que com isso os alguns conflitos entre as duas abordagens fossem equalizados. (GLOVER; BHATT apud TEIXEIRA (2011), p. 39).

Tabela 2.3: Normas da ISO referentes à RFID

ISO Standard	Título	Status
ISO 11784	RFID para animais - estrutura de código	Publicado - 1996
ISO 11785	RFID para animais - concepção técnica	Publicado - 1996
ISO/IEC 14443	Identificação de cartões - cartões com circuitos integrados sem contato - cartões de proximidade	Publicado - 2000
ISO/IEC 15693	Identificação de cartões - cartões de vizinhança integrados sem contato - cartões com circuitos	Publicado - 2000
ISO/IEC 18001	Tecnologia da Informação - Gerenciamento de Itens de RFID - Perfil de Requisitos de Aplicação	Publicado - 2004
ISO/IEC 18000-1	Parâmetros Gerais para Comunicação por Interface por Ar para Frequências Globalmente Aceitas	Publicado - 2004
ISO/IEC 18000-2	Parâmetros para Comunicação por Interface por Ar abaixo de 135 KHz	Publicado - 2004
ISO/IEC 18000-3	Parâmetros para Comunicação por Interface por Ar em 13,56 MHz	Publicado - 2004
ISO/IEC 18000-4	Parâmetros para Comunicação por Interface por Ar em 2,45 GHz	Em Revisão Final
ISO/IEC 18000-6	Parâmetros para Comunicação por Interface por Ar em 860 a 930 MHz	Publicado - 2004
ISO/IEC 15961	Gerenciamento de Itens de RFID - Protocolo de Dados: Interface de Aplicação	Publicado - 2004
ISO/IEC 15962	Gerenciamento de Itens de RFID - Protocolo: Regras de Codificação de Dados e Funções de Memória Lógica	Publicado - 2004
ISO/IEC 15963	Gerenciamento de Itens de RFID - Identificação única do RF Tag	Em Revisão Final

Fonte: (HECKEL (2007))

2.5 Arduíno

Arduíno (Figura 2.15) é uma plataforma baseada em uma simples placa de Entrada/Saída microcontrolada, de computação física, ou seja, são sistemas digitais ligados a sensores e atuadores, que permitem construir sistemas que percebam a realidade e respondem com ações

físicas, e é desenvolvida sobre uma biblioteca que simplifica a escrita da programação em C/C++ (FONSECA (2010)).

Em termos práticos, um Arduino é um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele[...]. O Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de *hardware e software* (McROBERTS (2011)).

O Arduino possui um microcontrolador (MCU) embarcado que é em essência um computador em um chip. Contém processador, memória e periféricos de entrada/saída. Pode ser programado para funções especiais, em contraste com outros microprocessadores de propósito geral, como os utilizados nos PCs(FONSECA (2010)).

A Figura 2.14 demonstra os elementos principais envolvidos na atuação do Arduino.

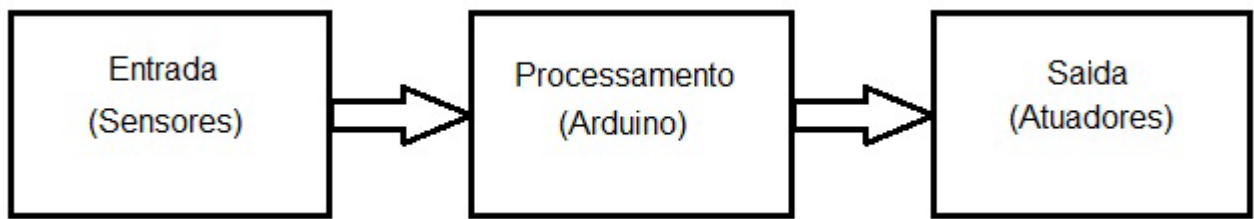


Figura 2.14: Circuito com Arduíno
Fonte: FONSECA (2010)

2.5.1 Características

O Arduino Uno é uma placa de microcontrolador baseado no ATmega328. Ele tem 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB, uma entrada de alimentação uma conexão ICSP (*In-Circuit Serial Programming*) e um botão de *reset*. Ele contém todos os componentes necessários para suportar o microcontrolador (Tabela 2.4) (MULTILÓGICASHOP (2013)).

Existem no mercado diversos tipos de Arduino, cada um com aplicação e características distintas, mais informações no link: <http://www.arduino.cc/en/Main/Boards> (WEBTRONICO (2013)). A seguir é feita a apresentação do Arduino Uno:

Alimentação

O Arduino UNO pode ser alimentado pela conexão USB ou por qualquer fonte de alimentação externa. A fonte de alimentação é selecionada automaticamente. Alimentação externa

(não-USB) pode ser tanto de uma fonte ou de uma bateria. A fonte pode ser conectada com um plug de 2,1mm (centro positivo) no conector de alimentação. Cabos vindos de uma bateria podem ser inseridos nos pinos Gnd (terra) e Vin (entrada de voltagem) do conector de alimentação. A placa pode operar com uma alimentação externa de 6 a 20 volts. A alimentação recomendada é de 7 a 12 volts (WEBTRONICO (2013)).



Figura 2.15: Arduino Uno R3
Fonte: MULTILÓGICASHOP (2013)

Os pinos de alimentação são:

- VIN: Entrada de alimentação para a placa Arduino quando uma fonte externa for utilizada. Você pode fornecer alimentação por este pino ou, se usar o conector de alimentação, acessar a alimentação por este pino.
- 5V. A fonte de alimentação utilizada para o microcontrolador e para outros componentes da placa. Pode ser proveniente do pino Vin através de um regulador on-board ou ser fornecida pelo USB ou outra fonte de 5 volts.
- 3V3. Alimentação de 3,3 volts fornecida pelo chip FTDI. A corrente máxima é de 50 mA.
- GND. Pino terra.

Memória

O ATmega328P tem 32 KB de memória flash para armazenar código (dos quais 2 KB são utilizados pelo *bootloader*), além de 2 KB de SRAM e 1 KB of EEPROM, que pode ser lida e escrita através da biblioteca EEPROM (WEBTRONICO (2013)).

Entrada e Saída

Cada um dos 14 pinos digitais do Arduino UNO pode ser usado como entrada ou saída usando as funções de *pinMode()*, *digitalWrite()*, e *digitalRead()*. Eles operam com 5 volts.

Cada pino pode fornecer ou receber um máximo de 40 mA e tem um resistor pull-up interno (desconectado por padrão) de 20-50 kOhms. Além disso, alguns pinos têm funções especializadas (WEBTRONICO (2013)):

- Serial: 0 (RX) e 1 (TX). Usados para receber (RX) e transmitir (TX) dados seriais TTL. Estes pinos são conectados aos pinos correspondentes do *chip serial FTDI USB-to-TTL*.

- External Interrupts: 2 and 3. Estes pinos podem ser configurados para disparar uma interrupção por um baixo valor, uma elevação ou *falling edge* ou uma mudança de valor. Veja a função *attachInterrupt()* para mais detalhes.

- PWM: 3, 5, 6, 9, 10, e 11. Fornecem uma saída analógica PWM de 8-bit com a função *analogWrite()*.

- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos suportam comunicação SPI, que embora compatível com o *hardware*, não está incluída na linguagem do Arduino.

- LED: 13. Há um LED já montado e conectado ao pino digital 13. Quando o pino está no valor HIGH, o LED acende; quando o valor está em *LOW*, ele apaga.

O Arduino UNO tem 6 entradas analógicas e cada uma delas tem uma resolução de 10 bits (i.e. 1024 valores diferentes). Por padrão, elas medem de 0 a 5 volts, embora seja possível mudar o limite superior usando o pino AREF e um pouco de código de baixo nível. Adicionalmente alguns pinos têm funcionalidades especializadas (WEBTRONICO (2013)):

- I2C: 4 (SDA) and 5 (SCL). Suportam comunicação I2C (TWI) usando a biblioteca *Wire* (documentação no site do *WIRE*).

Há ainda alguns outros pinos na placa:

- *AREF*. Referência de voltagem para entradas analógicas. Usados com *analogReference()*.

- *Reset*. Envie o valor *LOW* para resetar o microcontrolador. Tipicamente utilizados para adicionar um botão de *reset* aos *shields* que bloqueiam o que há na placa.

Comunicação

O ATmega328P permite comunicação serial no padrão UART TTL (5V), que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um chip FTDI FT232RL na placa encaminha esta comunicação serial através do USB e os drives FTDI (incluído no software do Arduino) fornece uma porta COM virtual para o software no computador. O software Arduino inclui um monitor serial que permite que dados simples de texto sejam enviados à placa Arduino (WEBTRONICO (2013)).

A biblioteca *SoftwareSerial* permite comunicação serial por quaisquer dos pinos digitais do Arduino UNO. O ATmega328P também oferece suporte aos padrões de comunicação I2C (TWI) e SPI. O *software* do Arduino inclui uma biblioteca *Wire* para simplificar o uso do *bus* I2C (WEBTRONICO (2013)).

Programação

O Arduino UNO pode ser programado com o *software* Arduino. O ATmega328P no Arduino UNO vem pré-gravado com um *bootloader* que permite enviar novos programas sem o uso de um programador de *hardware* externo. Ele se comunica utilizando o protocolo original STK500 (WEBTRONICO (2013)).

Tabela 2.4: Tabela com as características básicas do Arduino.

Microcontrolador	ATmega328
Voltagem Operacional	5V
Voltagem de entrada (recomendada)	7-12V
Voltagem de entrada (limites)	6-20V
Pinos E/S digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente CC por pino E/S	40 mA
Corrente CC para o pino 3,3V	50 mA
Flash Memory	32 KB (ATmega328) dos quais 0,5KB são utilizados pelo bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidade de Clock	16 MHz

Fonte: WEBTRONICO (2013)

2.6 Arquivo CSV

O CSV é uma implementação de arquivos de texto separados por um delimitador, que usa a vírgula e a quebra de linha para separar os valores. O formato também usa as aspas em campos no qual são usados os caracteres reservados (vírgula e quebra de linha). Essa robustez no formato torna o CSV mais atrativo que outros formatos digitais do mesmo segmento (CAVALCANTE (2013)).

O CSV tornou-se uma alternativa mais estruturada frente aos arquivos textos tradicionais e binários, usados em aplicações legadas, geralmente escritas em C, C++ e outras linguagens da época. Esse formato não segue o formalismo de apresentação de arquivos separados por tabulação, sendo mais direcionado para troca de dados entre sistemas (CAVALCANTE (2013)).

Exemplo de uma linha no ficheiro CSV:

“Título de Artigo para ser importado”,“Um resumo do artigo que desejamos que seja importado com o artigo”,“1292048CVS”,“234576543192”,“23,45”,“1”,“322”.

Capítulo 3

Implementação do Hardware

Neste capítulo será demonstrado o *hardware* escolhido para montar o protótipo e exemplificar o funcionamento do controle de fluxo e algumas ferramentas de apoio.

Foram utilizados os seguintes componentes:

- Módulo de leitura e escrita RFID YHY502CTG de 13.56 MHz;
- Módulo Arduino UNO R3;
- Módulo de leitura e escrita SD;
- *Protobord*;
- TAG de 13.56 MHz;
- Cabo flet colorido para conexão;
- Cabos (*jump*) para conexão;
- cabo USB;
- resistores; e
- Cartão SD 2G.

Foram utilizados as ferramentas de apoio abaixo descritas:

- Lente de aumento;
- Multímetro Digital UT-30B Prata *LOUD*;

3.1 Componentes

Módulo RFID YHY502CTG (Figura 3.1): trabalha a 13.56MHz, compatível com cartões MIFARE. É capaz de ler e escrever em cartões sem necessidade de contato. Possui antena embutida e identifica automaticamente a presença de TAG (cartão).



Figura 3.1: Módulo RFID YHY502CTG

Fonte: Elaborado pelo Autor

Arduíno UNO R3 (Figura 3.2): sendo uma plataforma baseada em uma simples placa de Entrada/Saída microcontrolada, utilizando o *chip* ATMEGA328. Possui quatorze entradas e/ou saídas digitais, das quais seis podem ser usadas como saídas PWM, seis entradas analógicas e um botão de *reset*. É conectado ao *notebook* pelo cabo USB, com 5V de tensão e, também, realiza as ligações lógicas.



Figura 3.2: Arduino Uno R3

Fonte: Elaborado pelo Autor

O Módulo de leitura e escrita SD aceita cartões formatados em FAT16 ou FAT32, e utiliza a comunicação via interface SPI por meio dos pinos MOSI, SCK, MISO e CS. A alimentação do módulo pode ser feita utilizando o pino 3.3V ou o pino 5V, Figura 3.3.

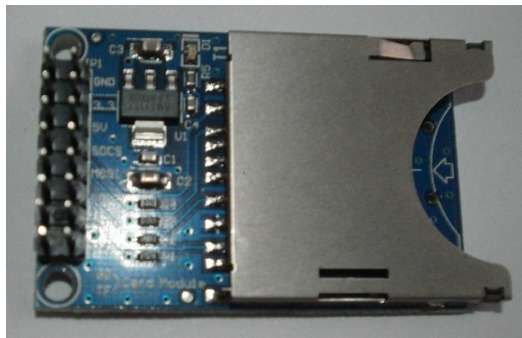


Figura 3.3: Módulo de cartão SD
Fonte: Elaborado pelo Autor

A *protobord* é utilizada para comportar os componentes sem ter que fazer soldagem, facilitando a criação do protótipo, Figura 3.4.

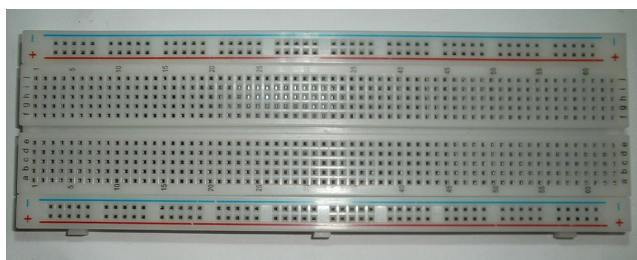


Figura 3.4: Protobord
Fonte: Elaborado pelo Autor

A TAG de 13.56 MHz é um *transponder*. Contém *chip* de silício e antenas que lhe permite responder aos sinais de rádio enviados por uma base transmissora, Figura 3.5.



Figura 3.5: TAG
Fonte: Elaborado pelo Autor

Cabo *flet* colorido é utilizado para conectar os diversos componentes eletrônicos (Figura 3.6).

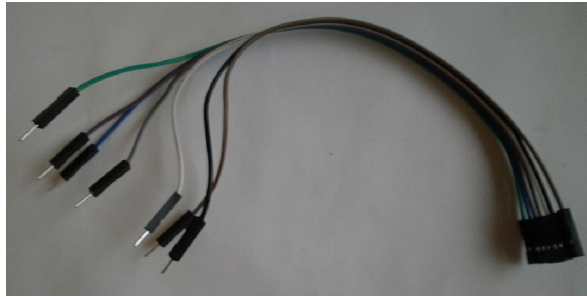


Figura 3.6: Cabo flet colorido
Fonte: Elaborado pelo Autor

Cabo-*jumper* colorido para prototipagem. Utilizado para conectar componentes eletrônicos (Figura 3.7).

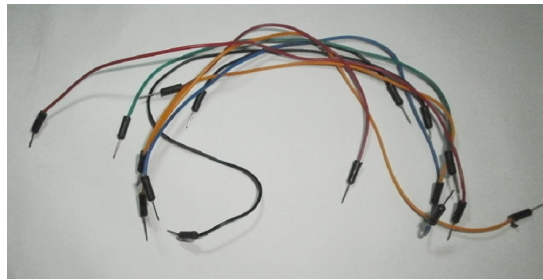


Figura 3.7: cabo-jumper
Fonte: Elaborado pelo Autor

Cabo USB utilizado para conexão serial do *Notebook* com o arduíno (Figura 3.8).



Figura 3.8: cabo USB
Fonte: Elaborado pelo Autor

Os resistores são utilizados para que seja criado um divisor de tensão para as conexões lógicas (Figura 3.9).

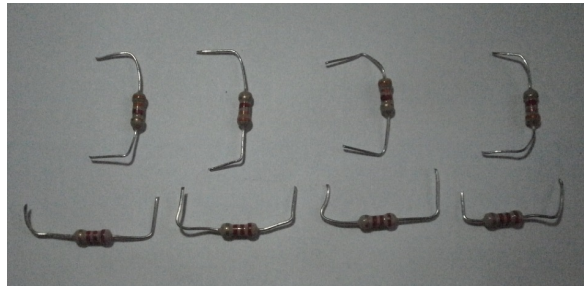


Figura 3.9: resistores
Fonte: Elaborado pelo Autor

3.2 Ferramentas de apoio

Foram necessários uma lente de aumento para viabilizar uma melhor leitura informações impressas nos componentes eletrônicos (Figura 3.10) e um multímetro digital para leituras de tensão, corrente e dos valores dos resistores (Figura 3.11).



Figura 3.10: Lente de aumento
Fonte: Elaborado pelo Autor



Figura 3.11: Mutímetro digital
Fonte: Elaborado pelo Autor

3.3 Conexão Arduino - Módulo RFID YHY502CTG

Para a conexão são utilizados os pinos 2, 3, GND e 5V, do Arduino e J1.1, J1.2, J1.8 e J1.9 do Módulo RFID YHY502CTG. Conexões (Tabela 3.1) e (Figura 3.12), padrão UART de comando (Tabela 3.2).

Tabela 3.1: Tabela de ligação - HECKEL (2007)

Arduino	Módulo RFID
GND	GND
5V	5V
2	RX
3	TX

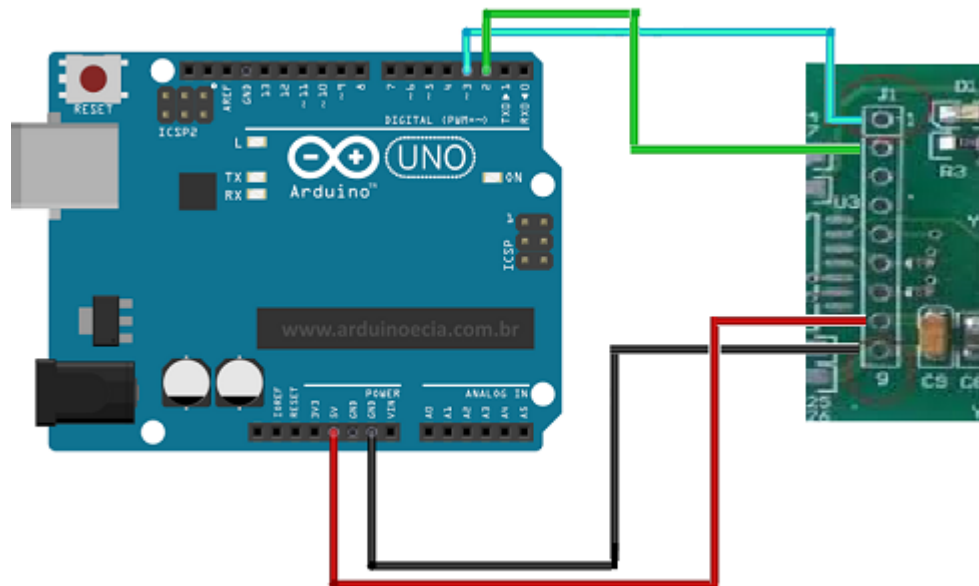


Figura 3.12: Circuito-Arduino-Modulo RFID
Fonte: ARDUINO and CIA (2014)

Protocolo UART: O módulo se comunica a 19200bps.

Tabela 3.2: Quadro de padrão UART de comando

Cabeçalho	Comprimento	Comando	Dados	XOR Checkssum
2 Byte	1 Byte	1 Byte	N Bytes	1 Byte

Fonte: (EHUOYAN (2010))

3.4 Conexão Arduino - Módulo Cartão SD

O módulo SD se conecta ao Arduino (Figura 3.13) utilizando a interface SPI, foram utilizados os pinos 4, 11, 12 e 13. A alimentação do módulo é feita por meio do pino 5V do Arduino (Tabela 3.3).

Tabela 3.3: Tabela de ligação - HECKEL (2007)

Arduino	Módulo SD
GND	GND
3.3V	3.3V
5V	5V
4	CS
11	MOSI
13	SCK
12	MISO

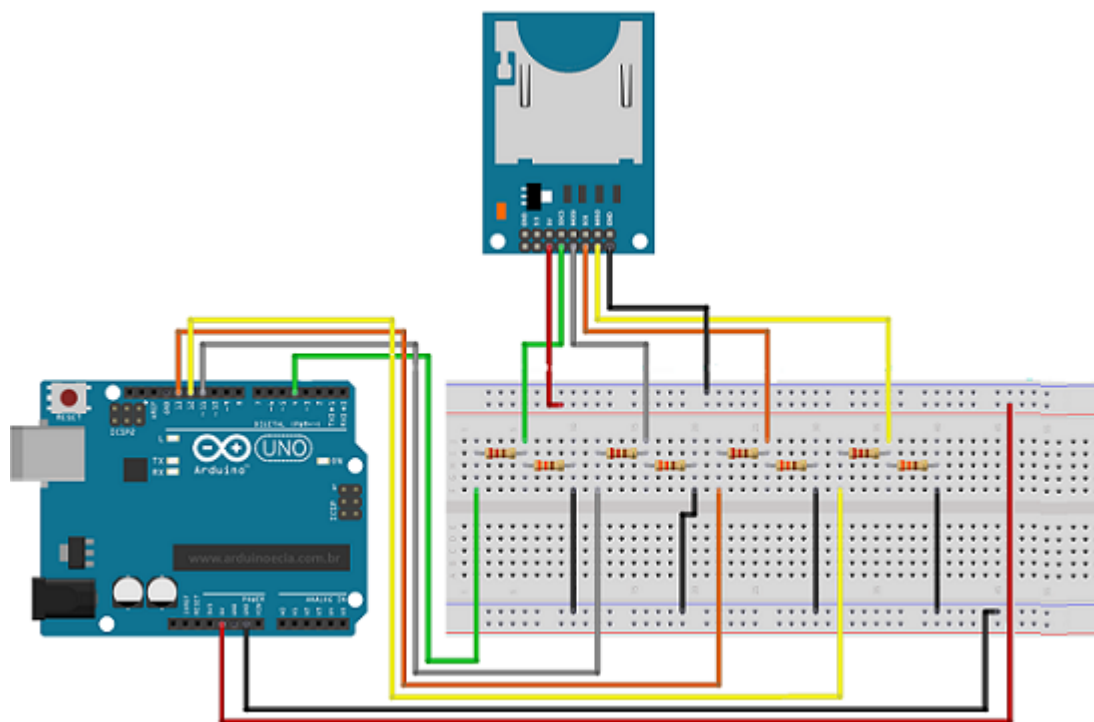


Figura 3.13: Circuito Modulo-CartaoSD-Arduino
Fonte: ARDUINO and CIA (2014)

É preciso cuidado com os pinos de sinal, pois o Módulo SD trabalha com 3.3V, enquanto que o Arduíno funciona com 5V. Devem ser utilizados resistores para criar um divisor de tensão (Figura 3.13). Foram utilizados dois resistores: um de 2.2 (R1) e outro de 3.3 (R2) ohms, que forneceram um nível de sinal de aproximadamente 3V.

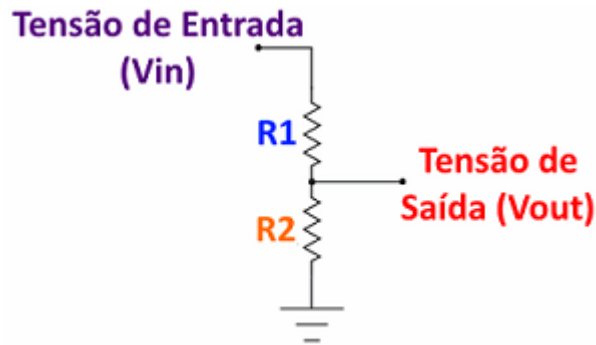


Figura 3.14: Circuito-divisor-de-tensao
Fonte: ARDUINO and CIA (2014)

3.5 Dois módulos conectados ao Arduíno

A (Figura 3.15) demonstra as ligações do módulo RFID e Cartão SD com o arduino. Foi utilizada uma *protobord* para realizar as ligações e montagem do divisor de tensão. A alimentação da *protobord* é realizada pelos pinos 3.3V e 5V do arduino.

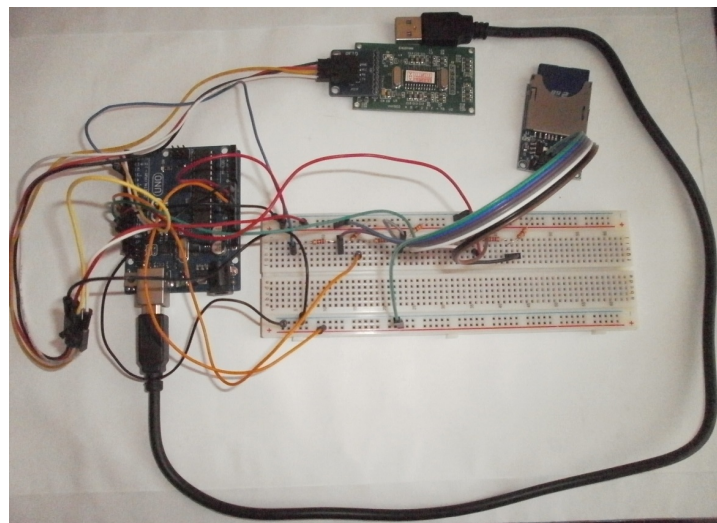


Figura 3.15: Ligação dos módulos com o Arduino
Fonte: Elaborado pelo Autor

Capítulo 4

Implementação do Software

Neste capítulo será apresentada a interface do arduino, a interação desta interface com o Arduino, através do cabo USB. Será demonstrado o código fonte e explicado o seu funcionamento.

O *software* do Arduino pode ser baixado no site www.arduino.cc, é livre e está disponível para Linux, Windows e Mac.

4.1 Interface do Arduino

A interface do Arduino é dividida em três partes: a *Toolbar* no topo, o código ou a *Sketch Window* no centro, e a janela de mensagens na base. A *Toolbar* consiste de sete botões (Figura 4.1). Sob a *Toolbar* há uma guia, ou um conjunto de guias, com o nome do arquivo do *sketch*. Também há um botão posicionado no lado direito.

Ao longo do topo há a barra de menus, com os itens *File*, *Edit*, *Sketch*, *Tools* e *Help*. Os botões na *Toolbar* fornecem acesso conveniente às funções mais utilizadas dentro desses menus.

Verify/Compile	Verifica se há erros no código
Stop	Interrompe o monitor serial, ou desmarca os outros botões
New	Cria um sketch em branco
Open	Mostra uma lista de sketches, em seu Sketchbook, para abrir
Save	Salva o sketch atual em seu sketchbook
Upload	Faz o upload do sketch atual para o Arduino
Serial Monitor	Exibe os dados seriais enviados do Arduino

Figura 4.1: Botões do Tolbar
Fonte: McROBERTS (2011)

É na *Sketch Window* (Figura 4.2) que o código do programa é elaborado e posteriormente verificado e compilado. Caso haja algum erro, será informado na janela de mensagens. Após corrigido o código, este será transferido para o Arduino, através do cabo USB.

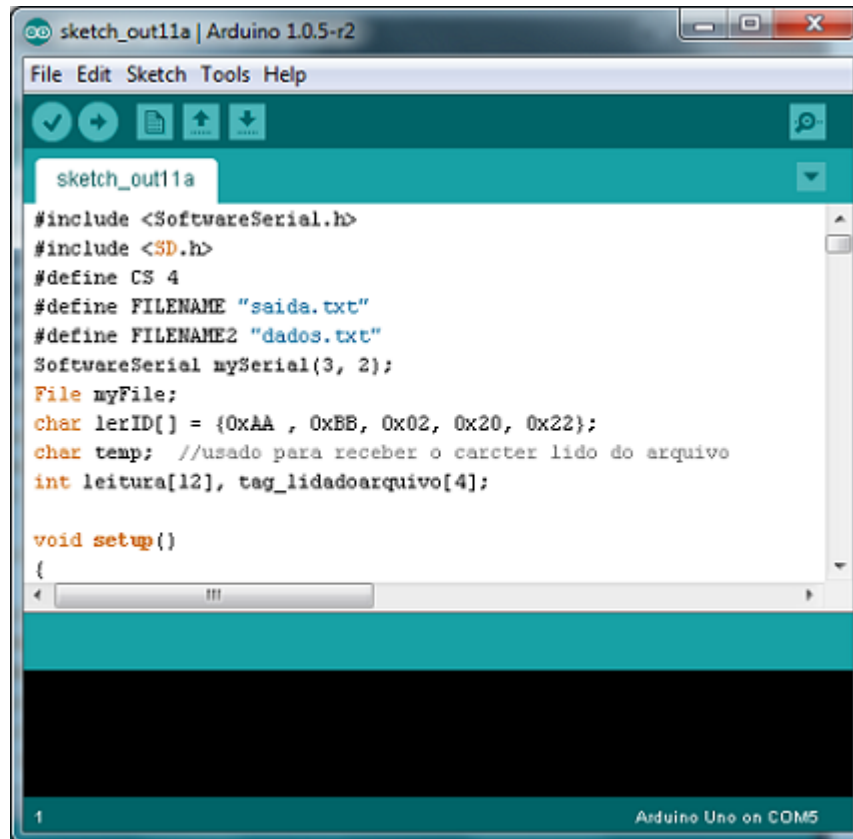


Figura 4.2: Interface do Arduino
Fonte: Elaborado pelo Autor

4.2 O código

Para desenvolver o código foram utilizadas as bibliotecas da linguagem C/C++, `SoftwareSerial.h` e `SD.h` que são disponibilizadas na instalação do software do Arduino.

Um *loop* mantém o leitor buscando uma TAG, através da emissão de ondas de radiofrequência. Enquanto não é detectada uma TAG, o leitor continua a sua busca. Quando a tag passa pela área de cobertura da antena, o campo magnético é detectado pelo leitor, que decodifica os dados codificados na tag e, em seguida, busca no arquivo `dados.txt` a existência do número da TAG.

A busca no arquivo é realizada linha a linha e caso o número da TAG não seja localizado, não é permitida a passagem automática do veículo, sendo o mesmo encaminhado ao atendimento manual. Caso seja localizado o número da TAG, o sistema verifica o dígito de

validade. Para a renovação da validade da TAG são verificados a documentação do veículo, habilitação do condutor e condições de funcionamento do veículo. A validade da TAG é de 365 dias.

Se a TAG estiver na validade, o número da TAG e o dígito de validade são registrados no arquivo saída.txt e em seguida é autorizada a passagem automática do veículo.

Estando a TAG com a validade vencida, também são registrados no arquivo saída.txt o número da TAG, o dígito de validade e, ainda, a informação da irregularidade. Neste caso, o veículo é direcionado para o atendimento manual para fins de verificação. Após os registros serem efetuados, o sistema retorna ao início do processo, com o leitor buscando por uma nova TAG.

A Figura 4.3 exemplifica o funcionamento do código:

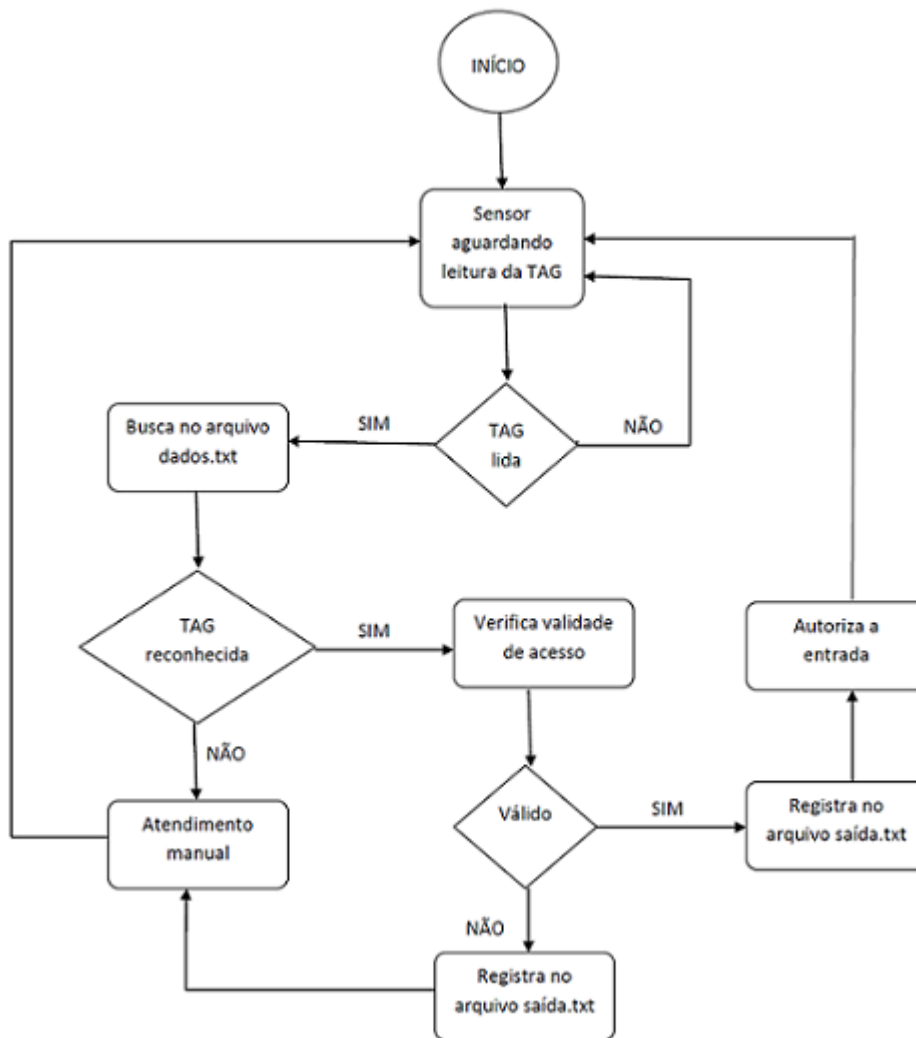


Figura 4.3: Fluxograma
Fonte: Elaborado pelo Autor

Descrição do código utilizado:

```

#include <SoftwareSerial.h>
#include <SD.h>
#define CS 4 //Pino CS do arduino UNO
#define FILENAME "saida.txt"
#define FILENAME2 "dados.txt"
SoftwareSerial mySerial(3,2);//SoftwareSerial mySerial(TX,RX)
File myFile;
char lerID[] = {0xAA , 0xBB, 0x02, 0x20, 0x22};
char tmp; //usado para receber o caracter lido do arquivo
int leitura[12], tag_lidadoarquivo[4];
void setup()
{
  pinMode(10, OUTPUT);
  //Serial.println("Inicializando SD...");
  pinMode(CS, OUTPUT);//Inicializando pino CS como output
  if (!SD.begin(CS)) {
    Serial.println("Falha!");
    return;
  }
  Serial.println("SD iniciado com sucesso!");
  mySerial.begin(19200);
}
void loop()
{
  delay (2000);
  int IDbyte,j,i,checksumTAG,checksumARQ,primeiro_PontoVirgula,primeiro_byte;
  int cod_validade=2;
  char temp;
  IDbyte=j=i=checksumTAG=checksumARQ=primeiro_PontoVirgula=primeiro_byte=0;
  for (j =0 ; j < 5 ; j++){ //Envia comando para leitura do tag
    mySerial.write(lerID[j]); //Serial.println(lerID[j],HEX);
  }
  for (j =0 ; j < 12 ; j++) //Limpa o Buffer de leitura dos tags
    leitura[j] = 0;
  j=0;//inicializa a variavel j para o while
  if(mySerial.available())
  {
    while (mySerial.available())

```

```

    {
        leitura[j]= mySerial.read(); //buffer para armazenar o Serial.print(leitura[j],HEX);
        j++;
    }
    if(j==9)
        { // so entra se for lida uma resposta com sucesso na leitura da TAG
//Abrindo o arquivo para leitura
        myFile = SD.open(FILENAME2, FILE_READ);
        if(!myFile)
            Serial.println("Falha ao abrir o arquivo de dados!");
        while(myFile.available()>0) //Enquanto nao chega no fim do arquivo
        {
            i=0;
            while ((temp = myFile.read()) != '\n')
            {
                if((temp>='0')&&(temp<='9'))// converte o caracter ASCII
                    // para um valor inteiro
                {
                    temp-='0';
                }else if((temp>='A')&&(temp<='F'))
                {
                    temp=(temp+10)-'A';
                }
                if((i&1)==1)//se o numero for impar, junte dois dígitos de 4 bits
                    //em um digito de 8 bits
                {
                    tag_lidadoarquivo[i/2]=(temp | (primeiro_byte << 4));
//movimenta o primeiro digito de 4 bits para a esquerda e adiciona um novo digito
                }
                primeiro_byte=temp;//armazena o primeiro byte lido
//só lê a validade no primeiro ponto e virgula
                if(i==9)
                {
                    cod_validade=temp-'0';//converte para inteiro
                }
                i++;
            } //fim da leitura do cod TAG e validade
            IDbyte=0; checksumARQ=0; checksumTAG=0;
            for (j=4 ; j<8 ; j++)// Retira apenas os Bytes com o ID dos tag
            {
                //Calculo do checksum da TAG

```

```

        checksumTAG^ = leitura[j];
        //Calculo checksum da leitura do arq. dados.txt
        checksumARQ^ = tag_lidadoarquivo[j-4];
        if (leitura[j]==tag_lidadoarquivo[j-4])
            IDbyte++;
    }
//dupla confirmacao: numero de coincidencias e checksum
    if(IDbyte==4 && checksumTAG==checksumARQ)
    {
        myFile.close();//Fechando arquivo dados.txt
//abrindo o arquivo para escrita
        myFile = SD.open(FILENAME, FILE_WRITE);
        if (myFile)
        {
            if(cod_validade==1)
            {
                //gravara os dados no cartao SD
                for (j=4 ; j<8 ; j++)// Retira apenas os Bytes com o ID dos tag
                {
                    myFile.print(leitura[j],HEX);
                }
                myFile.print(";");
                myFile.print(cod_validade);
                myFile.print(";");
                Serial.println("Cartao dentro da validade;");
            }else
// gravará os dados no cartao SD com a informacao da validade vencida
            {
                for (j=4 ; j<8 ; j++)// Retira apenas os Bytes com o ID dos tag
                {
                    myFile.print(leitura[j],HEX);
                }
                myFile.print(";");
                myFile.print(cod_validade);
                myFile.print(";");
                Serial.println("Cartao vencido! Acionar o usuario;");
            }
        } else
        {
            // se o arquivo nao abrir, imprime erro
            Serial.println("erro na abertura do arquivo saida.txt");
        }
        myFile.close(); // Fechando arquivo saida.txt

```



```

        continue;//sai do while pois ja encontrou o TAG
    } // Fim do if(IDbyte==4 && checksumTAG==checksumARQ)
} //fim do while de leitura do arquivo
} //fim do if da TAG lida com sucesso
} // fim do if(mySerial.available())
} //fim do Void loop

```

Funcionamento do código:

Primeiramente são incluídas as bibliotecas *SoftwareSerial.h* e *SD.h*, depois são definidos o pino CS do Arduino como 4, *FILENAME* como *saida.txt* e *FILENAME2* como *dados.txt*. A seguir é instanciado um objeto *SoftwareSerial*, dando-lhe o nome de *myserial*. A esta instância são passados os valores 2 e 3 definindo, assim, os pinos 2 e 3 do Arduino como RX e TX respectivamente. Por padrão RX e TX seriam os pinos 0 e 1 respectivamente.

Definir os pinos 2 e 3 do Arduino como RX e Tx é conveniente, pois seria necessário desconectar os pinos todas as vezes que fosse fazer o upload de algum código para o Arduino.

A seguir são declaradas as variáveis que serão utilizadas no código, um vetor de caracteres `lerID[]` que contém o comando de leitura da TAG, em hexadecimal, a ser enviado ao módulo RFID, uma variável para receber os caracteres que serão lidos do arquivo *dados.txt*, um vetor para armazenar os números de identificação das TAG e um vetor para armazenar a TAG lida do arquivo *dados.txt*.

As Tabelas 4.1 e 4.2 mostram o formato do quadro de dados utilizados na comunicação do Arduino com o módulo RFID:

Tabela 4.1: Comando de leitura

Cabeçalho	Comprimento	Comando	XOR <i>Checksum</i>
AABB	02	20	22

Tabela 4.2: Resposta do módulo RFID

Resultado	Cabeçalho	Comprimento	Comando	Nr TAG	XOR <i>Checksum</i>
Sucesso	AABB	06	20	4 Bytes	XOR <i>Checksum</i>
Falhou	AABB	02	DF	-	DD

Exemplo:

Comando: AABB 02 20 22

Checksum: $02 \oplus 20 = 22$

- Leitura bem sucedida

Resposta: AABB 06 20 92 BF 72 59 20

Número da TAG: 92 BF 72 59 - em hexadecimal

Checksum: $06 \oplus 20 \oplus 92 \oplus BF \oplus 72 \oplus 59 = 20$

- Falha na leitura

Resposta: AABB 02 DF DD

Checksum: $02 \oplus DF = DD$

Após as declarações das variáveis tem-se o primeiro *loop void setup()* que inicia:

- pino 10 do Arduíno como saída (apesar de usar o pino 4-CS como saída é obrigado, pela biblioteca SD, definir o pino 10 como saída);

- a comunicação serial;

- o pino CS como saída;

- o módulo SD card;

- a comunicação serial com o módulo RFID.

Na sequência vem o *loop* principal:

- realiza um *delay* entre as repetições do *loop*;

- inicia as variáveis que serão utilizadas;

- com o primeiro laço *for* envia o comando de leitura para o módulo RFID;

- com o segundo *for* limpa o *buffer* de leitura das TAG;

- com o laço *while* realiza o armazenamento dos valores lidos das TAG no vetor de inteiros *leitura*[12]. Os valores lidos são em hexadecimal;

- abre o arquivo *dados.txt* (Figura 4.4) para leitura;

- o primeiro laço *while* realiza a leitura enquanto não chegar ao fim do arquivo. Os dados são gravados no arquivo um por linha e estarão no formato (HHJJLLUU;N;) sendo que N poderá assumir o valor 0 - para inválido, ou 1 - para válido;

- o segundo laço *while* armazena o carácter lido do arquivo na variável *temp* e converte o carácter ASCII para inteiro. Esta conversão deve ser realizada cuidando para que seja em

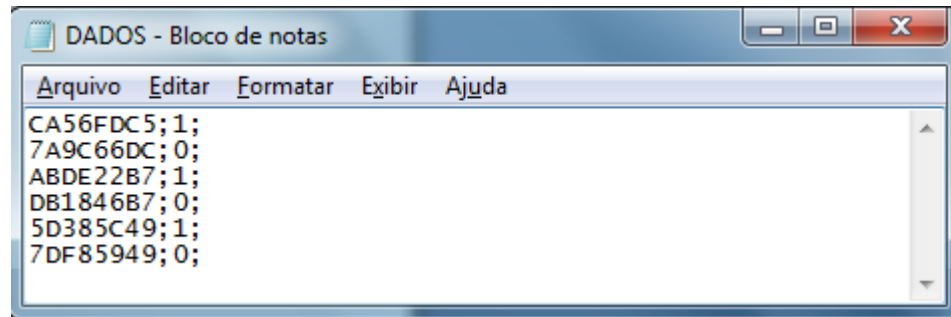


Figura 4.4: Arquivo dados.txt

Fonte: Elaborado pelo Autor

números hexadecimais equivalentes, por isso é usada a instrução *if-else* para determinar se esses caracteres estão entre 0 e 9 ou entre A e F;

Obs: Os valores lidos das TAG são em hexadecimal e representados em um único byte e os caracteres lidos do arquivo são representados cada um com um byte, desta forma, temos que converter os dois caracteres lidos em um único byte.

A variável *i* é iniciada com 0 antes de cada linha lida do arquivo. É utilizada para referenciar o índice do vetor onde estão armazenados cada caracter lido e convertido. Toda vez que o índice do vetor for ímpar, deve ser armazenado no vetor `tag_lidadoarquivo[]` a conversão dos dois últimos caracteres lidos em um único byte.

Exemplo:

Supondo que foram lidos os caracteres 9 e E.

A variável `primeiro_byte` armazena o 9, quando o índice é par. Este valor é deslocado quatro bits para a esquerda e na próxima leitura é realizada uma operação bit a bit OU com o valor atual lido (índice do vetor ímpar).

```
primeiro_byte = 9 = 00001001
00001001 << 4 = 10010000
E = 00001110
10010000 | 00001110 = 10011110
```

A instrução `if(i&1)` verifica se a variável *i* é ímpar ou par com a operação bit a bit (&). Se a variável *i* for par o resultado é zero, se for ímpar o resultado é 1.

Exemplo:

```
8 = 00001000 (par) & 1 = 0
1 = 00000001
```

$00001000 \& 00000001 = 00000000$

$9 = 00001001$ (ímpar) $\& 1 = 1$

$1 = 00000001$

$00001001 \& 00000001 = 00000001$

Observando o vetor `tag_lidadoarquivo[i/2]` o resultado de `i/2` sempre é um inteiro e somente o valor antes da vírgula será mantido. Desta forma garantimos que a cada dois dígitos que a variável `i` incrementa, o valor do índice do vetor `tag_lidadoarquivo` aumenta em uma unidade.

- O segundo laço `for` realiza a comparação dos valores lidos da TAG com os valores lidos do arquivo, a cada coincidência a variável `IDbyte` é incrementada e, realiza o cálculo do checksum;

- Depois é realizada uma dupla confirmação: se variável `IDbyte` for igual a quatro e o valor do checksum for igual ao valor do campo `leitura[8]`, a TAG lida foi encontrada;

- A seguir é fechado o arquivo `dados.txt` e é aberto, para escrita, o arquivo `saida.txt`, Figura 4.5. O *status* da validade será gravado junto com o identificador da TAG;

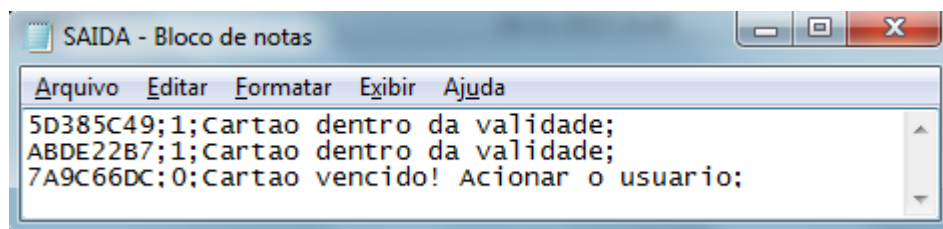


Figura 4.5: Arquivo `saida.txt`

Fonte: Elaborado pelo Autor

- Na sequência é fechado o arquivo `saida.txt`. O sistema volta aguardar por uma nova leitura de TAG.

Capítulo 5

Conclusão

Durante o desenvolvimento deste trabalho foram aplicados os conhecimentos adquiridos durante os quatro anos de formação acadêmica. Foi observado de forma prática como um *software* se relaciona com o *hardware* para a solução de um problema. É com a tecnologia que desvendamos novas soluções para antigos problemas e, assim, favorecemos a sociedade com uma melhor qualidade de vida.

Durante a pesquisa observou-se uma grande dificuldade em encontrar livros publicados que abordam o assunto identificação por radio frequência. Muitas publicações em forma de artigos e tutorias foram utilizadas para atingir o sucesso da experimentação. Também foi necessário estudar eletrônica básica para ser capaz de analisar, encontrar falhas relacionadas à eletrônica e escolher o hardware de acordo com suas especificações

A escolha de utilizar o Arduino Uno foi favorável pela disponibilidade das bibliotecas existentes que puderam ser utilizadas juntamente com o restante do código desenvolvido. Os módulos RFID YHY502CTG e o leitor de cartão SD utilizados demonstraram perfeita integração com o Arduino Uno e funcionaram adequadamente em resposta ao software.

O sistema utilizado no protótipo criado atingiu a sua finalidade, foi capaz identificar a TAG, ler um arquivo de dados, verificar se o identificador da TAG é válido, passar uma informação e, ainda, gerar um arquivo csv com informações relevantes. Estas informações podem ser utilizadas por qualquer aplicação capaz de ler um arquivo de texto.

Devido a motivos alheios a minha vontade, não foi possível instalar o sistema na Guarda do Quartel para teste de avaliação. Os testes foram realizados com base nos leitores e TAG escolhidas para o protótipo. Com certeza é possível desenvolver o sistema, sendo ainda necessário a integração de sensores para o controle do tempo de abertura de portões ou cancelas.

O estudo também permitiu identificar uma infinidade de aplicações para esta tecnologia que, apesar de muito presente em nosso cotidiano, possui uma grande capacidade de proporcionar soluções para problemas que ainda precisam ser resolvidos.

Fica, desta forma, uma lacuna a ser preenchida pelo meio acadêmico. Os futuros alunos do Curso de Sistemas de Informação da UEMS poderão desenvolver e aprimorar seus

conhecimentos, contribuindo, desta forma, para o um melhor aprendizado, desenvolvimento de protótipos, crescimento da segurança e bem estar da sociedade.

Referências Bibliográficas

- ARDUINO and CIA (2014). Conectando o módulo cartão sd ao arduino. <http://www.arduinoecia.com.br/2014/11/gravando-dados-arduino-cartao-sd.html>, acesso em 10 de Outubro de 2015.
- BHATT, B. G. H. (2007). *Fundamentos de RFID*. Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina, Rio de Janeiro, RJ. http://www.linuxmall.com.br/files/_product/429/428823/cap5_leitores_e_impessoras.pdf, acesso em 15 de Agosto de 2013.
- CAVALCANTE (2013). Conversão para arquivo csv. <http://www.cavalcanteassociados.com.br/article.php?id=626>, acesso em 22 de Outubro de 2013.
- COSTA, L. (2013). importância da padronização para o rfid. <http://www.fit-tecnologia.org.br/artigos/a-importancia-da-padronizacao-para-o-rfid>, acesso em 20 de Outubro de 2013.
- EHUOYAN (2010). Datasheet do módulo rfid yhy502ctg. http://www.ehuoyan.com/download/module/DS_YHY502CTGV33.pdf, acesso em 10 de Outubro de 2015.
- EPC-RFID, I. (2013). Epc information. <http://www.epc-rfid.info/>, acesso em 26 de Outubro de 2013.
- FONSECA, E. G. P. (2010). *Apostila Arduino*. Universidade Federal Fluminense - Centro Tecnológico Escola de Engenharia Curso de Engenharia de Telecomunicações - Programa de Educação Tutorial, Niterói, RJ. <http://pt.scribd.com/doc/54716085/Tut-Arduino>, acesso em 09 de Setembro de 2013.
- GREFF, P. A. (2009). *Especificação de um Sistema para Monitoramento de Atividades de Natação usando RFID*. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Santa Catarina, SC. wiki.sj.ifsc.edu.br/wiki/images/0/06/ProjetoFinal_Ponciano.pdf, acesso em 13 de Outubro de 2013.
- HECKEL, A. P. (2007). *Identificação Por Rádio Frequência(RFID)Estudo Teórico e Análise da Viabilidade do Uso de Simulação*. Centro Universitário Feevale Instituto de Ciências Exatas e Tecnológicas Curso de Ciência da Computação, Novo Hamburgo, RS. http://tconline.feevale.br/tc/files/0001_877.pdf, acesso em 10 de Julho de 2013.

- MARTINS, M. B. (2013). Padronização de protocolos de rfid. http://www.teleco.com.br/tutoriais/tutorialrfid/pagina_3.asp, acesso em 21 de Julho de 2013.
- McROBERTS, M. (2011). *Arduino Básico*. Editora Novatec, São Paulo - SP.
- MONTEIRO, F. V. (2013). Radio frequency identification. http://www.gta.ufrj.br/grad/10_1/rfid/index.html, acesso em 17 de Junho de 2013.
- MULTILÓGICASHOP, S. (2013). Arduino uno r3. <http://multilogica-shop.com/Arduino-Uno-R3>, acesso em 27 de Novembro de 2013.
- RFID-NEWS, J. (2013). A discussão da iso standad para rfid: a sua proveniência, viabilidade e limitações. http://www.rfidnews.com/iso_11784.html, acesso em 27 de Novembro de 2013.
- RFIDBr, P. B. (2013). Tags rfid. <http://www.rfidbr.com.br/index.php/tags-rfid.html>, acesso em 15 de Outubro de 2013.
- SENAI-SC (2009). Sistema de gerenciamento de dados coletados pela tecnologia rfid. <http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CDAQFjAA&url=http%3A%2F%2Frevista.ctai.senai.br%2Findex.php%2Feducacao01%2Farticle%2Fdownload%2F109%2F55&ei=Wc54UvnCG6bfsASSs4GwDw&usq=AFQjCNEQdrmPv-Jd7cveWisY8oFqrPS9aQ&bvm=bv.55980276,d.cWc>, acesso em 26 de Outubro de 2013.
- TANENBAUM, A. S. (2003). *Redes de computadores 4 Ed.* Editora Campos, Vrije Universiteit Amsterdam, Holanda, 2003.
- TEIXEIRA, T. (2011). *Controle de Fluxo de Pessoas Usando RFID*. Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina, São José, SC. http://wiki.sj.ifsc.edu.br/wiki/images/f/fa/TCC_TiagoTeixeira.pdf, acesso em 23 de Junho de 2013.
- WEBTRONICO, S. (2013). Arduino uno r3. <http://www.webtronico.com/arduino-uno-r3.html>, acesso em 27 de Novembro de 2013.
- WENDLING, M. (2010). *Sensores*. Colégio Técnico Industrial de Guaratinguetá, Guaratinguetá, SP. <http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>, acesso em 01 de Novembro de 2013.