CENTRO UNIVERSITÁRIO UNIFACVEST
CIÊNCIAS DA COMPUTAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO
MOISÉS MARTINS OZÓRIO

Desenvolvimento de robô detector de gases

LAGES,
2014.

MOISÉS MARTINS OZÓRIO

Desenvolvimento de robô detector de gases

Projeto apresentado à banca examinadora de trabalhos do curso de Ciência da computação para análise e aprovação.

LAGES ,
2014.

MOISÉS MARTINS OZÓRIO

Desenvolvimento de robô detector de gases

Trabalho de conclusão do Curso de Ciência da Computação apresentado ao Centro Universitário UNIFACVEST como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação.
Prof. MSc. Márcio José Sembay.

Lages, SC ____/____/2014. Nota_____ _____

_____

_____

_____

LAGES,
2014

RESUMO

Este trabalho tem como finalidade descrever o desenvolvimento de um robô para detecção de diversos tipos de gases, em ambientes industriais. Com o desenvolvimento científico de uma forma geral, criaram-se cada vez mais novos processos industriais. Tais processos, muitas vezes criaram nas indústrias, ambientes em que a atmosfera contém muitos gases potencialmente explosivos. Essas áreas requerem, na maioria dos casos, o uso frequente de equipamentos elétricos, que somados a esta atmosfera, constituem uma das principais fontes de ignição para explosões, uma vez que podem produzir centelhas ou arcos, seja em seu funcionamento normal, ou devido a falhas específicas. Nesse contexto, o robô surge como um equipamento mais confiável e preciso, e um aliado na busca pela segurança humana.

Palavras-chave: Robô, Segurança, Detecção de gases.

ABSTRACT

This work aims to describe the development of a robot to detect various types of gases in industrial environments. With the development of science in general , they have created increasingly new industrial processes .Such processes often created in industries, environments where potentially explosive atmosphere contains many gases. These areas require, in most cases, the frequent use of electrical equipment, which added to this atmosphere, constitute a major source of ignition for explosions, since they can produce sparks or arcs, either in normal operation or due to specific failures. In this context, the robot emerges as a more reliable and accurate equipment and an ally in the quest for human security.

Keywords: Robot, Security, Detection of gases.

RESUMEN

Este trabajo tiene como objetivo describir el desarrollo de un robot para detectar diversos tipos de gases en ambientes industriales . Con el desarrollo de la ciencia en general , han creado cada vez nuevos procesos industriales. Estos procesos a menudo creados en industrias , entornos en los que potencialmente atmósfera explosiva contiene muchos gases . Estas áreas requieren , en la mayoría de los casos , el uso frecuente de los equipos eléctricos , que añade a la atmósfera de ensayo constituyen una importante fuente de estallidos de ignición , ya que pueden producir chispas o arcos , ya sea en el funcionamiento normal o debido a fallas específicas . En este contexto , el robot se perfila como un equipo más fiable y preciso y un aliado en la búsqueda de la seguridad humana .

Palabras clave : robot , Seguridad , Detección de gases .

LISTA DE SIGLAS

CC – *Corrente Contínua*

CPU – *Unidade Central de Processamento*

DDNS – *Sistema de Nomes de Domínios Dinâmico*

DHCP – *Protocolo de configuração dinâmica de host*

DNS – *Sistema de Nomes de Domínios*

EEPROM – *Memória somente de leitura programável apagável eletricamente*

GLP – *Licença Pública Geral*

GND – *Terra*

HZ – *Hertz*

IDE – *Ambiente de desenvolvimento integrado*

IE – *Internet Explorer*

IEEE – *Instituto de Engenheiros Eletricistas e Eletrônicos*

IP – *Protocolo de Internet*

ISM – *Industrial, Científica e Médica*

Kbps – *Kilo Bytes por segundo*

LAN – *Rede de área local*

LCD – *Display de Cristal Líquido*

LED – *Diodo emissor de luz*

PC – *Computador Pessoal*

PIC – *Controlador de Interface Programável*

PPM **--***Parte por Milhão*

RAM – *Memória de Acesso Randômico*

RF – *Rádio Frequência*

ROM – *Memória apenas de leitura*

VCC – *Tensão corrente contínua*

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

AGRADECIMENTOS

Agradeço aos meus pais que sempre me incentivaram e me apoiaram, e pelas broncas e permissões negadas, pois só assim não desvirtuei do caminho certo.

Agradeço principalmente a minha esposa por tentar me entender, e pela pessoa maravilhosa que é.

Agradeço a minhas irmãs pelas brigas e brincadeiras que vivenciamos.

Agradeço ao meu cunhado pela força, e pela fé que sempre depositou em mim.

Agradeço aos meus amigos que chegaram até a este momento, os que não chegaram também, os que me ajudaram, e os que compartilharam as dificuldades da faculdade.

Agradeço aos professores que se dedicaram a transmitir seus conhecimentos.

Agradeço ao Adilson pelo tempo e ambientes que proporcionou para que este trabalho fosse desenvolvido, e pela parceria sempre.

Agradeço ao Jaison pela colaboração e dedicação no projeto, e pelos conhecimentos compartilhados.

Agradeço ao Guilherme pela força.

Agradeço a todas as pessoas que contribuíram para meu sucesso e para meu crescimento como pessoa. Sou o resultado da confiança e da força de cada um de vocês.

E por fim obrigado a Deus, por estar vivo e poder viver mais esta conquista.

I. INTRODUÇÃO

A história da localização de vazamento de gases é antiga, visto que os romanos já realizavam na antiguidade estas inspeções em barris de vinho. Os barris eram imersos em um tanque com água, se fosse constatado a presença de bolhas de ar saindo de uma determinada junta deste barril, então era porque ele não estava bem vedado e estava vazando ar. Esta mesma técnica ainda é utilizada nos dias de hoje por mecânicos para localizar vazamento em pneus e a partir dela foram se desenvolvendo outras formas para detecção do vazamento de gás. (Calante, 2012).

Antigamente uma lamparina era utilizada na mineração como uma forma de detecção de gás em minas de carvão subterrâneas e sistemas de esgoto. Apesar de ter sido originalmente criada como fonte de luz, o aparato também podia ser usado para identificar níveis de gases combustíveis, com uma precisão aproximada entre os 25% e 50%, dependendo da experiência do usuário, do seu treinamento, idade e identificação de cores. Já os detectores de gás modernos precisam ser muito mais confiáveis e com uma maior precisão.

O desenvolvimento tecnológico, fez com que os métodos de detecção de vazamento de gás não ficassem restritos somente à utilização dos sentidos humanos. A utilização de equipamentos complexos e específicos permitiu ampliar bastante a faixa de medição e melhorar muito a precisão destes métodos de ensaio. (Calante, 2012).

Nesse sentido esta monografia consiste em descrever o desenvolvimento de um protótipo de robô detector de gases para ambientes industriais, onde muitas vezes se torna perigoso a presença humana, utilizando inicialmente sensores específicos para detecção de gases Metano (CH4), Butano (C4H10) e Propano (C3H8).

I.1 Justificativa

O vazamento de gás hoje em dia é um problema que sem os cuidados necessários podem gerar grandes tragédias. A detecção moderna tem a capacidade de identificar vazamentos de acordo com sua proporção em relação ao oxigênio, mas quase todos estes equipamentos para a detecção do vazamento somente avisam as pessoas que estão próximas ao local onde estão instalados, com avisos sonoros.

Tendo em vista a comodidade e a segurança do usuário, a utilização de um robô facilmente controlável, com sistema de detecção e dados em tempo real, avisando a irregularidade poderá evitar grandes explosões e salvar vidas.

> Uma explosão na academia Tem Esportes, em São Bernardo do Campo, no ABC, deixou ao menos dois mortos e nove pessoas feridas na manhã deste sábado (17), segundo os bombeiros. Os feridos, entre eles uma criança, foram levados a hospitais da região. Os bombeiros afirmam que a principal hipótese é que um vazamento de gás tenha causado a explosão. Vizinhos relataram um forte cheiro de gás no entorno da academia. (G1 Notícias, 2014).

Vazamentos contínuos de gases industriais são deixados a responsabilidades do ‘nariz’ dos trabalhadores, que acostumados com a atmosfera do ambiente de trabalho são prejudicados pela diminuição sensitiva e gradual do olfato. Níveis, que nenhum trabalhador pode mensurar ou diagnosticar quantidades, para um pronto atendimento a um vazamento a tempo de evitar uma tragédia; casos assim, de acidentes citados em várias matérias editadas por jornais e redes de televisão.

1.2 Objetivos do trabalho

1.2.1 Objetivo geral

• Desenvolver um protótipo de robô, para verificar o nível de gás no ambiente.

1.2.2 Objetivos específicos

• Apresentar um protótipo de um robô capaz de identificar vazamento de gás inflamável.

Controlar o Robô em distâncias de até 1 km, utilizando controle remoto com transmissores de alto desempenho.

• Obter dados de níveis de gases, através de sensores específicos, e apresentar em display LCD.

• Obter imagens através de câmera IP instalada no robô, a partir de qualquer rede conectada com a internet.

1.3 Metodologia

1.3.1 Caracterizações da pesquisa

Segundo Silva e Menezes (2001, p. 20), existem diversas formas de classificar as pesquisas, porém as formas clássicas de classificação são: quanto aos objetivos, quanto à forma de abordagem, quanto à natureza, e quanto aos procedimentos adotados.

Quanto aos objetivos, a presente pesquisa pode ser classificada como exploratória, pois tem como finalidade desenvolver, esclarecer e explorar o tema escolhido. Segundo Gil (1999, p. 43), "pode-se dizer que estas pesquisas têm como objetivo principal o aprimoramento de ideias ou a descoberta de intuições".

Ainda de acordo com Gil (2002, p. 43), as pesquisas exploratórias são desenvolvidas com o objetivo de proporcionar visão geral, de tipo aproximativo acerca de determinado fato.

Quanto à abordagem, a presente pesquisa pode ser classificada, segundo Silva e Menezes (2001, p. 20), como qualitativa e quantitativa, pois "a interpretação dos fenômenos e a atribuição de significados são básicos no processo [...]. E requerem o uso de métodos e técnicas estatísticas".

Quanto à natureza, pode ser considerada uma pesquisa aplicada, que de acordo com Marconi e Lakatos (2002, p. 20) "caracteriza-se por seu interesse prático, isto é, que os resultados sejam aplicados ou utilizados, imediatamente, na solução de problemas que ocorreram na realidade".

Quanto aos procedimentos a pesquisa é bibliográfica, utilizando-se de livros e artigos revisados relacionados com termologia. A maior fonte de conteúdo é os datasheets, manuais e catálogos dos equipamentos. Segundo Gil (1999) a pesquisa bibliográfica é elaborada através de material já publicado com livros e artigos de periódicos estando ou não em meios eletrônicos.

1.3.2 Estrutura do trabalho

O trabalho está organizado da seguinte forma:

O primeiro capítulo traz a INTRODUÇÃO, apresentando o tema da pesquisa, as questões de estudo, os objetivos, bem como, as justificativas e a sistematização do trabalho.

No segundo capítulo, REVISÃO BIBLIOGRÁFICA, trata de assuntos como sensor, microcontroladores, transmissores, câmeras IP, motores CC, de forma conceitual. Nesse capítulo é apresentada também uma visão geral do projeto.

No terceiro capítulo, DESCRIÇÃO DO HARDWARE E SOFTWARE, aborda as especificações dos dispositivos utilizados, é detalhada a especificação dos componentes de controle, do microcontrolador PIC utilizado, do Arduino, detalhamento do sensor, bem como do Display LCD, dentre outros dispositivos e *softwares*.

No quarto capítulo, IMPLEMENTAÇÃO, é apresentado o desenvolvimento do projeto, como a integração entre o *hardware* e *software*, desenvolvimento do código na linguagem C e outras informações.

No quinto capítulo, RESULTADOS OBTIDOS abordam toda a parte de teste do protótipo, ajustes de problemas encontrados e suas dificuldades.

No sexto capítulo, CONSIDERAÇÕES FINAIS, são apresentadas as sugestões para trabalhos futuros.

1.3.3 Cronograma

Atividades Realizadas	Jul.	Ago.	Set	Out	Nov.	Dez
Pesquisa	x					
Apresentação da pré-proprosta		x				
Revisão Bibliográfica		x				
Desenvolvimento do Robô		x	x	x		
Testes e modificações			x	x		
Entrega do TCC a banca examinadora					x	
Correções					x	
Entrega do TCC a banca examinadora						x

Figura 01 - Cronograma

Fonte: O Autor.

II REVISÃO BIBLIOGRÁFICA

2.1 Arduino

> Arduino e um projeto que engloba software e hardware e tem como objetivo fornecer uma plataforma fácil para prototipação de projetos interativos, utilizando um microcontrolador. Ele faz parte do que chamamos de computação física: área da computação em que o software interage diretamente com o hardware, tornando possível integração fácil com sensores, motores e outros dispositivos eletrônicos. (Justen, 2014, p.8).

A parte de hardware do projeto, uma placa que cabe na palma da mão, é um computador como qualquer outro: possui microprocessador, memoria RAM, memoria flash (para guardar o software), temporizadores, contadores, dentre outras funcionalidades.

Neste projeto, foi utilizada a versão UNO R3 pela programação e componentes utilizados serem considerados básicos, porem existe muitos Arduinos como o MEGA, ADK e LEONARDO que possuem o processamento mais rápido, além de possuir mais entradas digitais e analógicas.

Figura 02 - Arduino UNO R3

Fonte: Robocore.net

A principal diferença entre um Arduino e um computador convencional é que, além de ter menor porte (tanto no tamanho quanto no poder de processamento), o Arduino utiliza dispositivos diferentes para entrada e saída em geral. Por exemplo: em um PC utiliza-se teclado e mouse como dispositivos de entrada e monitores e caixa de som como dispositivos de saída; já em projetos com o Arduino os dispositivos de entrada e saída são circuitos elétricos/eletrônicos. Como a interface do Arduino com outros dispositivos esta mais perto do meio físico que a de um PC, pode-se ler dados de sensores (temperatura, luz, pressão etc.) e controlar outros circuitos (lâmpadas, motores, eletrodomésticos etc.), dentre outras coisas que não conseguíramos diretamente com um PC. A grande diferença com relação ao uso desses dispositivos, no caso do Arduino, e que, na maior parte das vezes, constrói-se os circuitos que são utilizados, ou seja, não se limita apenas a produtos existentes no mercado: o limite é dado

pelo conhecimento e criatividade do usuário. O melhor de tudo nesse projeto é que seu software, hardware e documentação são abertos. O software é livre (GNU GPL), o hardware é totalmente especificado (basta entrar no site e baixar os esquemas) e a documentação esta disponível.

2.2 Sensores

2.2.1 Sensor de Gás

Sensor é um termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente. Um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Geralmente o sinal de saída é manipulado antes de sua leitura no sistema de controle. (Thomazini e Albuquerque, 2005, p.17).

É um termo utilizado para indicar dispositivos sensíveis a alguma forma de energia do ambiente que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza que pode ser medida, como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc. (Thomazini e Albuquerque, 2005, p.17).

As principais características físicas dos gases são a sua grande compressibilidade e extraordinária capacidade de expansão. Os gases não apresentam um volume fixo, pois sempre ocupam o volume total do recipiente em que estão confinados. Outra propriedade inerente aos gases é que eles são miscíveis entre si em qualquer proporção, ou seja, formam uma mistura homogenia. (Usberco e Salvador, 2006, p.374).

Assim como existem materiais condutores (cobre, alumínio, ouro, prata) e materiais isolantes (borracha, vidro), existe um tipo de material que é um meio termo entre esses dois primeiros. Esse material é o semicondutor, ou seja, um quase condutor de eletricidade. O semicondutor possui um nível de condutividade entre os extremos de um isolante e um condutor. Os dispositivos semicondutores são considerados a peça mais importante na revolução ocorrida na microeletrônica que tanto tem influenciado as nossas vidas. Os materiais semicondutores mais usados na indústria eletrônica são o Germânio (Ge) e o Silício (Si). (Schmidt, 2010).

2.2.2 Sensor ultrassônico

Outro sensor utilizado foi o sensor ultrassônico que serviu para evitar colisões com paredes e objetos, pois detecta uma barreira através do uso de ultrassons.

Figura 03 - Sensor Ultrassônico

Fonte: Robocore.net

Os sensores que usam ultrassons encontram uma grande variedade de utilizações na indústria e mesmo em outros campos de atividades.

Esses sensores se caracterizam por operar por um tipo de radiação não sujeita a interferência eletromagnética e totalmente limpa, o que pode ser muito importante para determinados tipos de aplicações, podendo operar de modo eficiente detectando objetos em distâncias que variam entre milímetros até vários metros, eles podem ser usados para detectar os mais variados tipos de objetos e substâncias.

Segundo Newton Braga (2012), O princípio de operação desses sensores é exatamente o mesmo do sonar, usado pelo morcego para detectar objetos e presas em seu voo cego. O pequeno comprimento de onda das vibrações ultrassônicas faz com que elas reflitam em pequenos objetos, podendo ser captadas por um sensor colocado em posição adequada.

O comprimento de onda usado e a frequência são muito importantes nesse tipo de sensor, pois ele determina as dimensões mínimas do objeto que pode ser detectado.

Na verdade, conforme mostra a figura 4, só ocorre reflexão em intensidade suficiente para se atingir um bom sinal, quando o objeto tem dimensões que se aproximam do comprimento de onda do sinal, ou seja, maior.

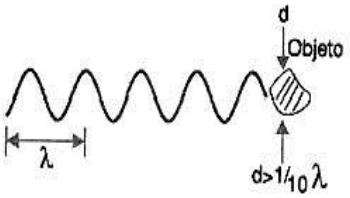

Figura 04 - Reflexo das vibrações

Fonte: O Autor

Os sinais passam através de objetos cujas dimensões sejam muito menores do que o comprimento de onda. Por esse motivo é que sons comuns não podem ser usados nesse tipo de detector.

Um sinal de 1000 Hz, por exemplo, teria 34 cm de comprimento de onda, sendo teoricamente esse o tamanho do menor objeto que poderia ser detectado por essa frequência, considerando-se uma velocidade aproximada do som de 340 m/s. (Braga, 2012).

Na prática um sensor ultrassônico é formado por um emissor e um receptor, tanto fixados num mesmo conjunto como separados, dependendo do posicionamento relativo desejado, conforme mostra a figura 5.

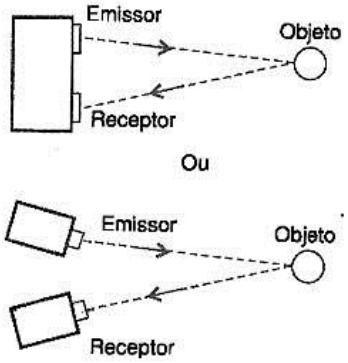

Figura 05 - Tipos de Sensores Ultrassônicos

Fonte: O autor.

Levando-se em conta a velocidade do som, pode-se determinar com precisão a velocidade de aproximação ou afastamento do objeto pela medida da alteração de sua frequência, portanto o princípio de funcionamento é o mesmo dos radares usados nas rodovias, mas naquele caso utilizam micro-ondas.

2.3 Câmeras IP

Uma câmera IP é uma câmera de vídeo que pode ser acessada e controlada através de qualquer rede IP, como a LAN, Intranet ou Internet. Usando apenas um navegador web e uma conexão de internet, usuários podem devidamente ter acesso ao vídeo de uma câmera e, em alguns casos, até áudio, de qualquer local que esteja. Os modelos atuais são compatíveis com as tecnologias Ethernet e Wi-Fi e são separadas em categorias como Pan/Tilt/Zoom que é um modelo que permite movimentos horizontais e verticais da câmera e também o zoom, além de possibilitar ao usuário mudar o ângulo das câmeras, habilitar áudio, controlar uso de luz infravermelha para uso noturno, entre outros.

> Uma das vantagens da Câmera IP é o fato de que ela pode funcionar sem a necessidade de um microcomputador no local onde será feito o monitoramento bastando existir uma rede WI-FI com acesso a Internet. Outro ponto importante é o fato dela possuir mecanismos que permitem sua rotação 320° na horizontal e 120° na vertical e LEDs infravermelho que permitem a captação de imagens em completa escuridão, possibilitando a visualização detalhada do local. Isto possibilita que possamos posicionar a câmera em qualquer direção remotamente através de um browser (IE, Firefox, Chrome, etc….) ou de um smartphone , Iphone ou Tablet, visualizando o que ocorre no local em tempo real e mesmo na ausência de luminosidade ambiente. (Saulo, 2013, p. 14).

A instalação e a configuração da câmera de segurança IP sem fio é relativamente simples. É necessário apenas configurar um roteador com servidor DHCP e um computador, que receberá as imagens. Após a configuração do endereço do IP e do DNS, o usuário terá o número de identificação do seu computador e dos outros equipamentos conectados em sua rede local e a internet. Apenas com o endereço de IP o usuário irá conseguir monitorar a sua devida câmera pela internet.

2.4 Módulo XBee

Os módulos XBee, são módulos RF que fazem comunicações no padrão ZigBee IEEE 802.15.4. O Protocolo ZigBee permite comunicações robustas e opera na frequência ISM (Industrial, Scientific and Medical), sendo aqui no Brasil 2,4 GHz (16 canais) e em outras partes do mundo, e não requerem licença para funcionamento.

De acordo com Messias(2008)

> Os módulos RF padrão ZigBee foram criados para economizar o máximo de energia possível. Com isso, é possível criar aplicações onde é possível ler sensores em campo remotamente, apenas utilizando pilhas ou baterias comuns, que durarão meses ou mesmo anos sem precisarem ser substituídas. Isso porque, os módulos ZigBee quando não estão transmitindo/recebendo dados, entram num estado de dormência ou em "Sleep", consumindo o mínimo de energia.

As Redes ZigBee oferecem uma excelente imunidade contra interferências, e a capacidade de hospedar milhares de dispositivos numa Rede (mais que 65.000), com taxas de transferências de dados variando entre 20Kbps a 250Kbps. O Protocolo ZigBee é destinado a aplicações industriais, portanto, o fator velocidade não é crítico numa implementação ZigBee.

2.5 Motores CC

Os motores de corrente contínua (CC) ou motores DC (Direct Current), como também são chamados, são equipamentos que operam aproveitando as forças de atração e repulsão geradas por eletroímãs e imãs permanentes. (Braga, 2014).

O princípio de funcionamento se baseia na ideia de que se passar correntes elétricas por duas bobinas próximas, os campos magnéticos criados poderão fazer com que surjam forças de atração ou repulsão. A ideia básica de um motor é montar uma bobina entre os polos de um imã permanente ou então de uma bobina fixa que funcione como tal.

Partindo da posição inicial, em que os polos da bobina móvel (rotor), ao ser percorrido por uma corrente, estão alinhados com o imã permanente tem-se a manifestação de uma força de repulsão. Esta força de repulsão faz o conjunto móvel mudar de posição. A tendência do rotor é dar meia volta para que seu polo Norte se aproxime do polo Sul do imã permanente. Da mesma forma, seu polo Sul se aproximará do polo Norte pelo qual será atraído. No entanto, no eixo do rotor, por onde passa a corrente que circula pela bobina, existe um comutador. A finalidade deste comutador é inverter o sentido da circulação da corrente na bobina, fazendo com que os polo mudem. O resultado disso será uma transformação da força de atração em repulsão, o que fará com que o rotor continue seu movimento, passando "direto" pela posição que seria de equilíbrio. Sua nova posição de equilíbrio seria obtida com mais volta, de modo que os polos do rotor se defrontassem com os de nome oposto do imã fixa. Mais meia volta, e quando isso poderia ocorrer, a nova posição faz com que o comutador entre em ação e tem-se nova comutação da corrente. Com isso os polos se invertem, O resultado disso é que o rotor

não para, pois deve continuar em busca de sua posição de equilíbrio.

Estes motores possuem um rendimento e custo razoável quando usados em projetos de Robótica e Mecatrônica, sendo por este motivo os preferidos de muitos projetistas. Eles podem ser encontrados numa ampla faixa de tensões nominais, tipicamente entre 1,5 e 48 volts.

2.6 Gases

2.6.1 Propano

Segundo a Petrobrás (2014, p.2), as substâncias desta categoria contêm principalmente e moléculas de hidrocarbonetos de baixo peso molecular, as quais são o perigo dominante nos gases de hidrocarbonetos de petróleo. Suas características físicas e químicas exigem que sejam mantidos dentro de sistemas rigorosamente fechados. Ao contrário de gases de refinaria, gases de hidrocarbonetos de petróleo não contêm compostos inorgânicos (por exemplo sulfeto de hidrogênio, amônia, monóxido de carbono).

2.6.2 Butano

De acordo com a Colband(2012),O butano é um hidrocarboneto gasoso, composto orgânico cujos átomos de carbono se encontram dispostos em cadeias lineares. É um gás altamente inflamável, portanto deve ser manuseado de maneira correta. Este é o conhecido gás de cozinha utilizado no ambiente doméstico. O gás propano pode emitir mais energia do que o butano, mas o gás butano tem uma determinada propriedade que o torna ideal para o confinamento.

2.6.3 Metano

O metano (CH_4) é um gás incolor e sem cheiro, possui pouca solubilidade em água e, quando adicionado ao ar, pode ser altamente explosivo. Ele é muito conhecido por suas propriedades energéticas e por ser proveniente das vacas, mas há diversas outras fontes de CH_4, que pode ser muito prejudicial à saúde humana.

2.6.4 Limite de tolerância a exposição

De acordo com a NR 15, Nas atividades ou operações nas quais os trabalhadores ficam expostos a agentes químicos, a caracterização de insalubridade ocorrerá quando forem ultrapassados os limites de tolerância. No Quadro 1 é apresentado o limite de exposição para alguns gases.

Quadro 1- Limite de tolerânica

Limite de tolerância		
Gás	até 48 horas por semana	
	p.p.m	mg/m^3
Butano	470	1090
Metano	asfixiante	simples
Propano	asfixiante	simples

Conforme mostrado no quadro, a NR 15 considera o propano e o metano como asfixiante simples e não impõe limites de exposição, entretanto, no ambiente de trabalho, deve-se garantir que a concentração mínima de oxigênio seja de 18% em volume. As situações na qual a concentração de oxigênio estiver abaixo deste valor serão consideradas de risco grave e iminente.

Nos Estados Unidos, a OSHA "Occupational Safety & Helth Administration" do "U.S. Department of Labor", determina que a concentração máxima de propano no ambiente de trabalho seja de 1000 ppm (em volume a 25ºC e 1 atm).

2.7 PROTEUS

A ferramenta Proteus é composta por duas ferramentas principais, tais são: ISIS, responsável pela construção, modelagem e simulação de circuitos digitais e analógicos em tempo real; e ARES, responsável pela criação dos layouts dos circuitos projetados. Dessa forma é possível simular o funcionamento dos circuitos antes que sejam construídos. A Figura 06 e a Figura 07 apresentam as telas principais das ferramentas ISIS e ARES, respectivamente.

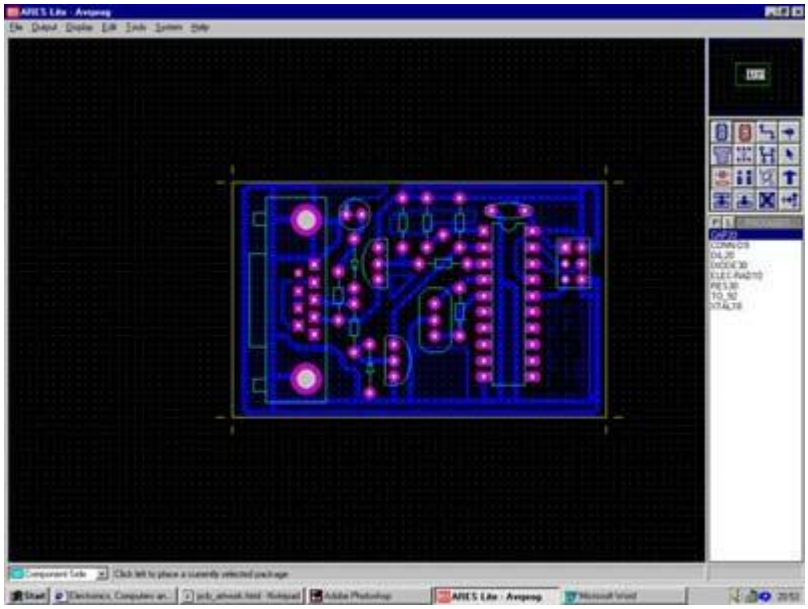

Figura 6 – Tela principal do software Ares

Fonte: O Autor

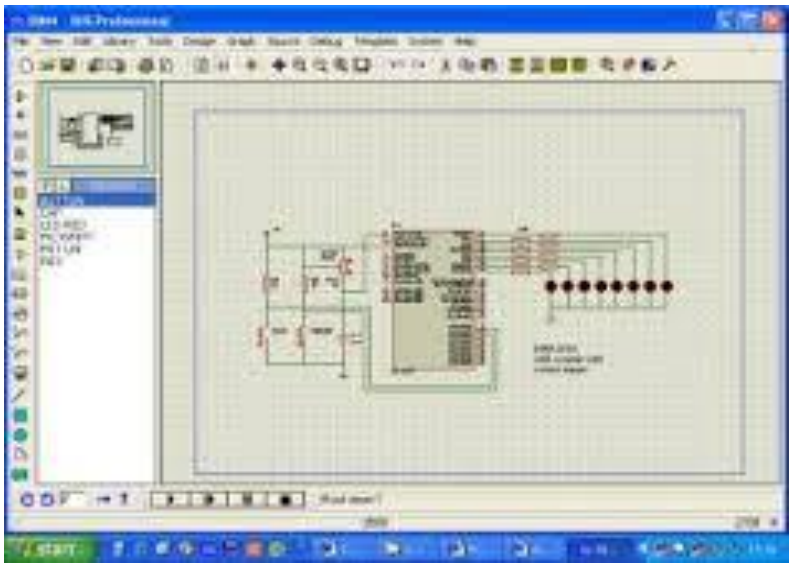

Figura 7 - Tela principal do software ISIS

Fonte: O Autor

O programa Proteus apresenta recursos importantes, a saber, coleção de bibliotecas; compatibilidade entre ISIS e ARES; desenho de placas em mais de uma camada; componentes animados; e cadastro com vários microcontroladores da família PIC.

.

III DESCRIÇÃO DO HARDWARE E SOFTWARE

Este capítulo explica as especificações, pinagens de forma detalhada, dos dispositivos usados e seu devido funcionamento nesse projeto, tanto a parte física, ou seja, o hardware, como a parte lógica, os softwares utilizados.

3.1 Microcontrolador PIC 18F23K20

Um dos microcontroladores utilizados para esse projeto foi o PIC 18F23K20, conforme mencionado no capítulo anterior. Ele é o componente responsável por receber os comandos do controle e enviar para o robô, assim como receber os dados do sensor de gás e mostra-los em um display LCD.

A Figura 8 ilustra o PIC 18F23K20 utilizado neste projeto.

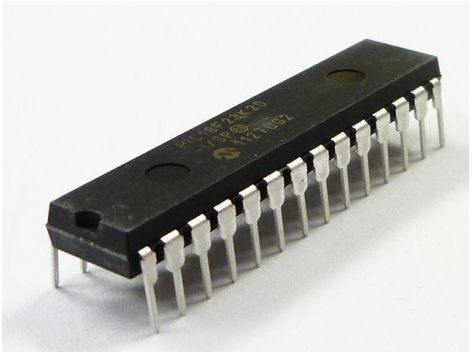

Figura 08 - PIC 18F23K20

Fonte: Newark.com

3.1.1 Especificações

O PIC 18F23K20 é um microcontrolador fabricado pela Microchip Technology, com as seguintes características:

- Composto por 28 pinos;
- Possui 75 instruções no seu microcódigo;
- Sinal de clock de frequência até 64MHz;
- Memória de programa do tipo flash de 8192 words.
- 512 bytes de memória RAM para dados;
- 256 bytes de memória EEPROM para dados;

- Dados de 8 bits por endereço de memória;
- 4 timers;
- 24 pinos, os quais podem ser configurados como entradas e/ou saídas;
- Outras características especiais como programação in-circuit serial, proteção por código, watchdog timer, módulo CPP, e comparador interno.

Este modelo da família PIC foi escolhido devido ao fato de ser facilmente encontrado além de possuir um número de 24 pinos configuráveis de entrada e saída, atendendo plenamente ao projeto.

3.1.2 Pinagem do PIC 18F23K20

A pinagem do PIC 18F23K20 é mostrada na Figura 9 e o significado das nomenclaturas utilizadas na identificação desses pinos no quadro 1.

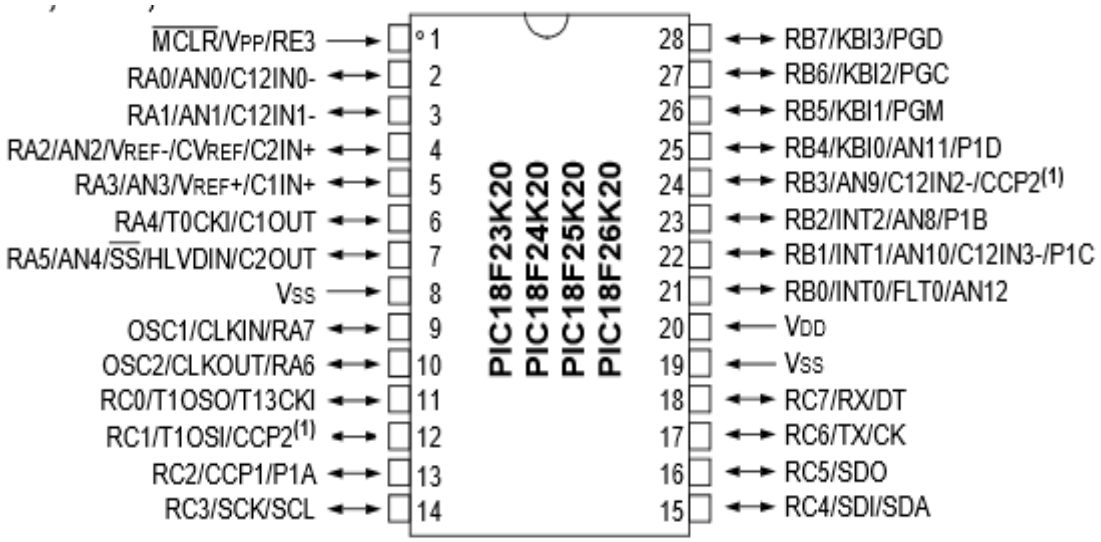

Figura 09 - Pinagem do PIC 18F23K20

Fonte:Microchip Technology Datasheet, 2009, p.4)

Quadro 2- Significado das nomenclaturas dos pinos do PIC18F23K20

Nº	Pino	Descrição
1	MCLR	Master Clear (reset). O microcontrolador funciona quando este pino está em nível alto.
	VPP	Entrada de tensão de programação
	RE3	Entrada Digital
2	RA0	Entrada e saída digital.
	AN0	Entrada analógica.
	C12IN0	Comparadores C1 e C2 entrada inversora
3	RA1	Entrada e saída digital.
	AN1	Entrada analógica.
	C12IN1	Comparadores C1 e C2 entrada inversora
4	RA2	Entrada digital.
	AN2	Entrada analógica.
	VREF -	Referência de tensão negativa
	CVREF	Saida de tensão de referência do comparador
	C2IN+	Comparador C2 entrada não inversora
5	RA3	Entrada e saída digital.
	AN3	Entrada analógica canal 3
	VREF+	Referência de tensão positiva
	C1IN+	Comparador C1 entrada não inversora
6	RA4	Entrada e saída digital.
	TOCKI	Entrada de clock externo timer 0
	C1OUT	Saída comparador C1
7	RA5	Entrada e saída digital.
	AN4	Entrada analógica canal 4
	SS	SPI entrada de seleção de escravo
	HLVDIN	Entrada de detecção de tensão positiva e negativa
	C2OUT	Saída comparador C2
8	VSS	Referencia de terra para logica de pinos de entrada e saida
9	OSC1	Oscilador de cristal
	CLKIN	Entrada de osciladores externos
	RA7	Entrada e saída digital.
10	OSC2	Oscilador de cristal
	CLKOUT	Saída para oscilador de cristal
	RA6	Entrada e saída digital.
11	RC0	Entrada e saída digital.
	T1OSO	Saida do oscilador externo para TMR1.
	T13CK1	Entrada de clock externo timer 1 e timer 3.
12	RC1	Entrada e saída digital.
	T10SI	Entrada do oscilador externo para TMR1.
	CCP2	Entrada de captura e saída de comparador e PWM 2.
13	RC2	Entrada e saída digital.
	CCP1	Entrada de captura e saída de comparador
	P1A	Saída reforçada de PWM CCP1
14	RC3	Entrada e saída digital.
	SCK	Relógio de série de entrada e saída para modo SPI

	SCL	Relógio de série de entrada e saída para modo I² C
15	RC4	Entrada e saída digital.
	SDI	Entrada de dados SPI
	SDA	Entrada e saída de dados I² C
16	RC5	Entrada e saída digital.
	SD0	Saída de dados SPI
17	RC6	Entrada e saída digital.
	TX	Transmissão de dados
	CK	Relógio relacionado a RX
18	RC7	Entrada e saída digital.
	RX	Recepção de dados
	DT	Dados relacionados a transmissão
19	VSS	Referencia de terra
20	VDD	Referencia positiva
21	RB0	Entrada e saída digital.
	INT0	Interrupção externa 0
	FLT0	Entrada PWM falhas para CCP1
	AN12	Entrada analógica canal 12
22	RB1	Entrada e saída digital.
	INT1	Interrupção externa 1
	AN10	Entrada analógica canal 10
	C12IN3-	Comparadores C1 e C2 entradas inversoras
	P1C	Saída reforçada PWM CCP1
23	RB2	Entrada e saída digital.
	INT2	Interrupção externa 2
	AN8	Entrada analógica canal 8
	P1B	Saída reforçada PWM CCP1
24	RB3	Entrada e saída digital.
	AN9	Entrada analógica canal 9
	C12IN2-	Comparadores C1 e C2 entradas inversoras
	CCP2	Entrada de captura , saída de comparador e PWM
25	RB4	Entrada e saída digital.
	KBl0	Interrupção mudança de pino
	AN11	Entrada analógica canal 11
	P1D	Saída reforçada PWM CCP1
26	RB5	Entrada e saída digital.
	KBl1	Interrupção mudança de pino
	PGM	Referencia de tensão negativa programação ICSP
27	RB6	Entrada e saída digital.
	KBl2	Interrupção mudança de pino
	PGC	Debugger e ICSP, programação pino do relógio
28	RB7	Entrada e saída digital.
	KBl3	Interrupção mudança de pino
	PGD	Debugger e ICSP, programação de dados

3.2 Display LCD

O display LCD escolhido para este projeto foi um LCD 16x2, é um módulo que possui duas linhas, cada linha exibe até dezesseis caracteres. A Figura 10 ilustra o módulo utilizado.

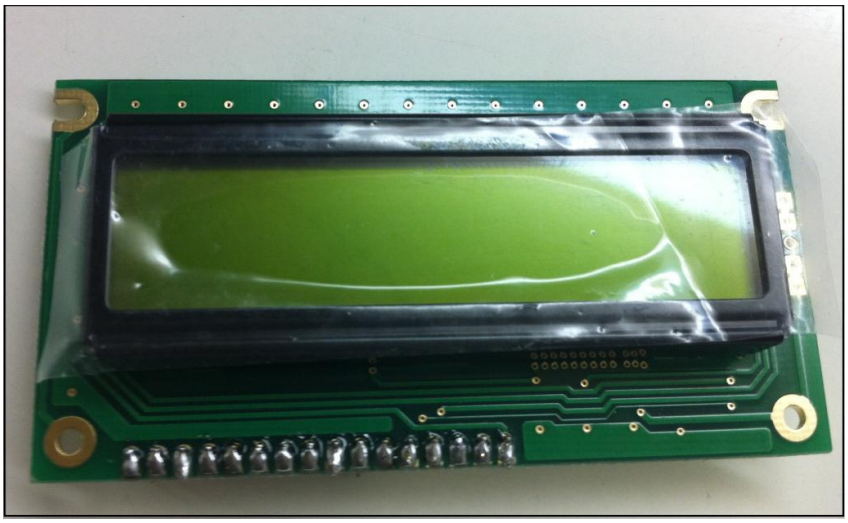

Figura 10 - Display LCD utilizado no projeto

Fonte: O Autor

O quadro 3 apresenta a descrição dos pinos do módulo LCD e o interfaceamento no projeto.

Quadro 3 - Pinos do Display LCD e ligações no projeto

Pino	Símbolo	Função	Interfaceamento no projeto
1	VSS	Terra	Aterrado
2	VDD	5V	Alimentação
3	Vo	Ajuste de Contraste	Conectado ao trimpot
4	R/S	Seleção de Registro	Pino RC1 do PIC23K20
5	R/W	Leitura/Escrita	Pino RC3 do PIC23K20
6	E	Inicio ciclo Leitura/Escrita	Pino RC0 do PIC23K20
7	DB0	Dado	Pino RB0 do PIC23K20
8	DB1	Dado	Pino RB1 do PIC23K20
9	DB2	Dado	Pino RB2 do PIC23K20
10	DB3	Dado	Pino RB3 do PIC23K20
11	DB4	Dado	Pino RB4 do PIC23K20
12	DB5	Dado	Pino RB5 do PIC23K20
13	DB6	Dado	Pino RB6 do PIC23K20
14	DB7	Dado	Pino RB7 do PIC23K20
15	A	Anodo Backlight	Vcc
16	K	Cátodo Backlight	Aterrado

Foram utilizados apenas quatro bits de dados, as quatro linhas mais significativas (DB7: DB4), com isso foram necessários apenas seis pinos do microcontrolador para que a comunicação com o LCD fosse viabilizada e um pino para controlar o Backlight do Display.

3.3 Sensor de Gás MQ-4

O principal sensor utilizado foi o sensor de gases MQ-4. Este sensor de gás detecta a concentração de metano, Butano, e Propano no ar e dá a saída como uma voltagem analógica. A faixa de sensibilidade é de concentrações de 200 a 10000ppm sendo apropriado para detecção de vazamentos. Este sensor pode operar em temperaturas de -10 a 50ºC e consome menos de 150mA a 5V. A carga da resistência deve ser calibrada para a sua particular aplicação utilizando as equações do datasheet mas um bom valor para início de calibragem é 20kΩ. (MQ-4 Datasheet, 2012, traduzido e adaptado).

Características:

1. Boa sensibilidade para gás combustível
2. Alta sensibilidade para gás natural
3. Vida longa e baixo custo
4. Circuito simples

A descrição acima é circuito de teste básico do sensor conforme mostrado na Figura 9. O sensor necessita de 2 tensões, tensão de aquecimento, (VH) e tensão de teste (VC). VH é usado para fornecer temperatura de trabalho certificado para o sensor, enquanto VC usado para detectar a tensão (VRL) na carga resistiva (RL) que está em série com o sensor. VC precisa de tensão DC. VC e VH podem usar o mesmo circuito elétrico com pré-condição para garantir desempenho do sensor.

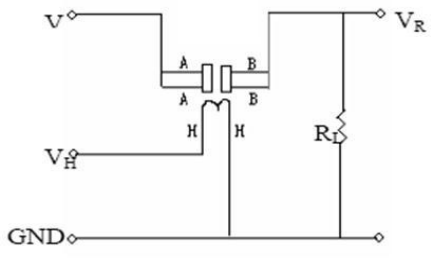

Figura 11 - Circuito de teste básico

Fonte: MQ4 Data Sheet

3.4 Arduino UNO R3

3.4.1 O Hardware

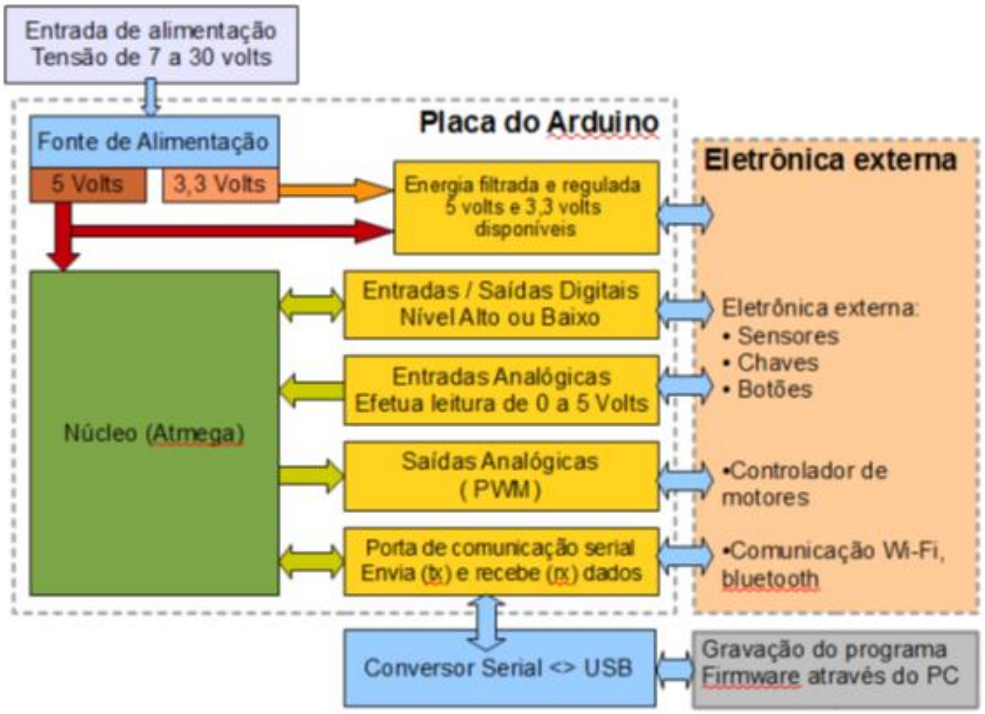

Figura 12 - Arquitetura do Arduino

Fonte: O Autor

O hardware do Arduino é muito simples, porém muito eficiente. Esse hardware é composto dos seguintes blocos:

Figura 13 - Blocos do Arduino

Fonte:Arduino.cc modificada

3.4.1.1 Fonte de Alimentação

Responsável por receber a energia de alimentação externa, que pode ter uma tensão de no mínimo 7 Volts e no máximo 35 Volts e uma corrente mínima de 300mA. A fonte filtra e depois regula a tensão de entrada para duas saídas: 5 Volts e 3,3 Volts. O requisito deste bloco é entregar as tensões de 5 e 3,3 Volts para que a CPU e os demais circuitos funcionem.

3.4.1.2 Núcleo CPU

O núcleo de processamento de uma placa Arduino é um micro controlador, uma CPU, um computador completo, com memória RAM, memória de programa (ROM), uma unidade de processamento de aritmética e os dispositivos de entrada e saída. Tudo em um chip só. E é esse chip que possui todo hardware para obter dados externos, processar esses dados e devolver para o mundo externo.

3.4.1.3 Entradas e Saídas

O microcontrolador usado no Arduino UNO é o ATMega328 que possui 28 pinos de conexões elétricas, 14 de cada lado. É através desses pinos que se pode acessar as funções do microcontrolador, enviar dados para dentro de sua memória e acionar dispositivos externos.

No Arduino, os 28 pinos deste micro controlador são divididos da seguinte maneira:

- 14 pinos digitais de entrada ou saída (programáveis)
- 6 pinos de entrada analógica ou entrada/saída digital (programáveis)
- 5 pinos de alimentação (gnd, 5V, ref analógica)
- 1 pino de reset
- 2 pinos para conectar o cristal oscilador

Os dois primeiros itens da lista são os pinos úteis, disponíveis para o usuário utilizar. Através destes pinos que o Arduino é acoplado à eletrônica externa. Entre os 14 pinos de entrada/saída digitais têm dois pinos que correspondem ao módulo de comunicação serial USART. Esse módulo permite comunicação entre um computador (por exemplo) e o chip.

Todos os pinos digitais e os analógicos possuem mais de uma função. Os pinos podem ser de entrada ou de saída, alguns podem servir para leituras analógicas e também como

entrada digital. As funções são escolhidas pelo programador, quando escreve um programa para a sua placa.

Na placa do Arduino, os pinos úteis do micro controlador são expostos ao usuário através de conectores fêmea onde podem ser encaixados conectores para construir o circuito externo à placa do Arduino.

3.4.1.4 Pinos com funções especiais

De acordo com Justen (2014), existem pinos do Arduino que possuem características especiais, que podem ser usadas efetuando as configurações adequadas através da programação. São eles:

PWM: Tratado como saída analógica, na verdade é uma saída digital que gera um sinal alternado (0 e 1) onde o tempo que o pino fica em nível 1 (ligado) é controlado. É usado para controlar velocidade de motores, ou gerar tensões com valores controlados pelo programa. **Pinos 3, 5, 6, 9, 10 e 11.**

Porta Serial USART: Pode-se usar um pino para transmitir e um pino para receber dados no formato serial assíncrono (USART). Pode-se conectar um módulo de transmissão de dados via bluetooth por exemplo e comunicar com o Arduino remotamente. **Pinos 0 (rx recebe dados) e pino 1 (tx envia dados).**

Comparador analógico: Pode-se usar dois pinos para comparar duas tensões externas, sem precisar fazer um programa que leia essas tensões e as compare. Essa é uma forma muito rápida de comparar tensões e é feita pelo hardware sem envolver programação. **Pinos 6 e 7.**

Interrupção Externa: Pode-se programar um pino para avisar o software sobre alguma mudança em seu estado. Usado para detectar eventos externos à placa. **Pinos 2 e 3.**

Porta SPI: É um padrão de comunicação serial Síncrono, bem mais rápido que a USART. É nessa porta que conecta-se cartões de memória (SD) e muitas outras coisas. **Pinos 10 (SS), 11 (MOSI), 12 (MISO) e 13 (SCK).**

3.4.1.5 Firmware

É simplesmente um software que é carregado dentro da memória do micro controlador. Tecnicamente o firmware é a combinação de uma memória ROM, somente para leitura, e um programa que fica gravado neste tipo de memória. E esse é o caso do micro controlador que a placa Arduino usa.

3.4.2 O software

O ambiente de desenvolvimento do Arduino é um compilador gcc (C e C++) que usa uma interface gráfica construída em Java. Basicamente se resume a um programa IDE muito simples de se utilizar e de estender com bibliotecas que podem ser facilmente encontradas. As funções da IDE do Arduino são basicamente duas: Permitir o desenvolvimento de um software e enviá-lo à placa para que possa ser executado.

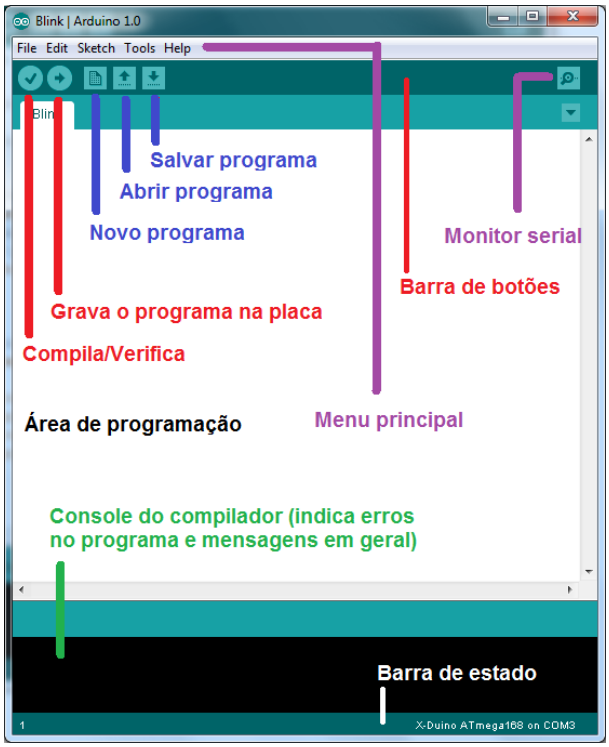

Figura 14 - Ambiente de programação

Fonte: O autor

3.4.2.1 Estrutura do programa

O programa para o Arduino é dividido em duas partes principais: **Setup** e **Loop**.

A função **setup** serve para inicialização da placa e do programa. Esta sessão é executada uma vez quando a placa é ligada ou resetada através do botão.

A função **loop** é como se fosse a main () da placa. O programa escrito dentro da função **loop** é executado indefinidamente, ou seja, ao terminar a execução da última linha desta função, o programa inicia novamente a partir da primeira linha da função **loop** e continua a executar até que a placa seja desligada ou o botão de reset seja pressionado.

A linguagem base para a programar um Arduino é C. Logo, suas estruturas de controle (if, else, while, for...), seus elementos de sintaxe (#define, #include, {}...), operadores aritméticos (+, -, *, ^ ...), operadores de comparação (==, !=, <, > ...), enfim, todos são utilizados no IDE. Portanto, saber C é primordial para programar o Arduino em alto nível.

Abaixo segue as principais funções que segundo Neto (2013, p. 10 e 11), servem para controlar o arduino:

pinMode (pin, mode): Configura o pino especificado para que se comporte como entrada ou saída, sendo Pin = número do pino e mode = INPUT ou OUTPUT

digitalWrite (pin,value): escreve um valor HIGH ou LOW em um pino digital. Se o pino foi configurado como saída sua voltagem será determinada ao valor correspondente: 5V para HIGH e 0V para LOW. Se o pino estiver configurado como entrada escrever um HIGH levantará o resistor interno de 20kΩ. Escrever um LOW rebaixará o resistor. Obviamente pin = numero do pino e valor = HIGH ou LOW.

int **digitalRead (pin):** Lê o valor de um pino digital especificado, HIGH ou LOW. Pin = numero do pino. Retorna HIGH ou LOW.

Int **analogRead (pin):** Lê o valor de um pino analógico especificado. Pode mapear voltagens entre 0 a 5v, sendo 4,9mV por unidade.

analogWrite (pin, value): Escreve um valor analógico. Pode ser utilizada para ligar um LED variando o brilho ou girar um motor a velocidade variável. Após realizar essa função o pino

vai gerar uma onda quadrada estável com ciclo de rendimento especificado até que o próximo analogWrite() seja realizado (ou que seja realizado um digitalRead() ou digitalWrite() no mesmo pino).

3.4.2.2 Serial Monitor

Esse monitor é usado para que seja possível comunicar a placa com o computador, mas também é muito útil para a depuração do programa. Basicamente conecta-se a placa no computador e através desta tela pode-se ver as informações enviadas pela placa.

Figura 15 - Exemplo de serial monitor

Fonte: O Autor

No exemplo, o comando **Serial.begin(9600)** inicializa a comunicação com uma taxa de 9600 bauds (taxa de bits). O comando **Serial.println ('argumento')** envia a mensagem para o computador.

Após compilar e enviar o programa para placa deve-se abrir o serial monitor. As informações enviadas pela placa Arduino aparecem no console.

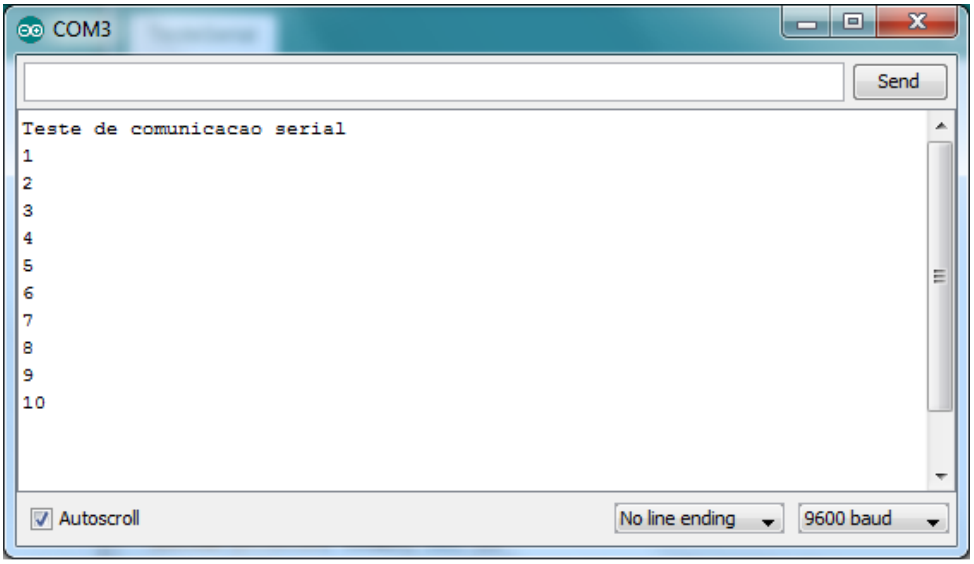

Figura 16 - Saída na serial monitor

Fonte: O Autor

3.5 Shield Ponte H L293d

Esse shield é compatível com Arduino Uno e Arduino Mega, possui dois chips L293D, cada um composto por 2 pontes H. Tudo isso para controlar até 4 motores DC, 2 Servos (alimentados por 5v) ou 2 motores de passo. A corrente máxima suportada por cada L293D é de 600 mA, com picos de 1.2A.

Nas laterais da placa encontra-se os terminais (com parafusos) para conexão dos motores DC ou motores de passo. Na parte superior esquerda, conectores de 3 pinos permitem a conexão de até 2 servos. Um led na parte central da placa indica não só o funcionamento do shield como também que há alimentação para os motores.

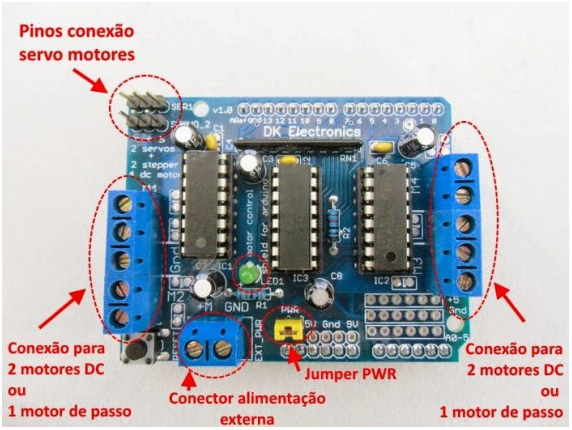

Figura 17 - Detalhes Shield Ponte H L293d

Fonte: Arduinoecia.com.br

A tensão de entrada pode variar de 4,5 à 25 Vcc. Para utilização de alimentação externa, deve-se retirar o jumper **PWR**.

Como a maioria dos shields, tem-se à disposição os pinos que sobram quando você não está controlando motores, obviamente, e também alguns pinos que estão sempre disponíveis :

- **Pinos disponíveis :** os pinos analógicos de A0 a A5, que também podem ser utilizados como pinos digitais 14 a 19. Também estão disponíveis os pinos 2 e 13.
- **Pinos utilizados para controle de motores DC e motores de passo :** 11,3,5 e 6, além dos pinos 4,7, 8 e 12.
- **Pinos utilizados para controle de servo motores :** pinos 9 (servo 1) e 10 (servo 2)

Nesse projeto foram utilizados apenas as conexões para os 4 motores CC, e o shield foi alimentado através de 2 baterias 6V/4Ah, ligadas em série.

3.6 Sensor Ultrassônico HC- SR04

Este sensor foi utilizado para evitar a colisão da plataforma robótica com obstáculos. É capaz de medir distânicas de 2cm a 4m. O módulo possui um circuito pronto, com emissor e receptor ultrassônico. Os parâmetros elétricos são descritos no quadro 4.

Quadro 4 - Parâmetros Elétricos do Sensor Ultrassônico

Parâmetros Elétricos	
Tensão de Trabalho	DC 5 V
Corrente de Trabalho	15mA
Frequência de Trabalho	40Hz
Alcance Máximo	4m
Alcance Mínimo	2cm
Ângulo medido	15º
Gatilho sinal de entrada	10uS TTL pulso
Saída de sinal de eco	Sinal alavanca TTL de entrada e o intervalo em proporção
Dimensão	45*20*15mm

O módulo possui 4 pinos : VCC , Trig, Echo, GND . Sendo assim, é muito simples utiliza-lo. O processo de medição se resume em: colocar o pino TRIG em nível ALTO por mais de 10us, e a medição começa; Quando o ultrassom emitido bater em um obstáculo e voltar, o pino ECHO ficará em nível ALTO pelo tempo correspondente a distância medida. Assim, a distância do obstáculo pode ser medida pelo tempo (largura do pulso) que o pino ECHO ficar em nível alto usando a seguinte regra:

1. Distância em CENTIMETROS: (largura do pulso em uS) / 58;
2. Distância em POLEGADAS: (largura do pulso em uS) / 148;
3. Se a largura do pulso for igual a 38ms, então não tem obstáculo.

Figura 18 - Sensor Ultrassônico HC-SR04

Fonte: Data sheet(2013)

3.7 Módulo XBee

O módulo XBee foi acoplado ao arduino através de placa desenvolvida, que permite a instalação do módulo XBee e comunicação sem a perda de funcionalidades da placa arduino. Para configuração do módulo XBee foi utilizado o software X-CTU disponibilizado pela Digi em conjunto com a unidade XBee Explorer USB, que através de um cabo mini USB permite a comunicação com o módulo.

3.8 IDE MPLAB

O ambiente de desenvolvimento para a programação do PIC utilizado foi o MPLAB.

De acordo com Vidal (2012, p. 1)

> O MpLab é um ambiente integrado para o estudo e desenvolvimento com a família PIC de microcontroladores. Sua principal característica é a total integração de seus módulos com o ambiente Windows, permitindo a fácil cópia de arquivos e trechos de arquivos de um aplicativo para outro.

Para o desenvolvimento da programação foi utilizado a linguagem C, por ser de fácil entendimento para programar se comparado com o assembly.

MPLAB IDE foi projetado para trabalhar com vários modelos de Microchip e flexibilidade para uso de ferramentas de linguagem de programação de terceiros. Estas ferramentas de programação podem ser usadas para projetos escritos em linguagem de programação em assembly, C ou linguagem BASIC; compilados em linguagem hexadecimal o código executável pode ser gravado no dispositivo em uma memória flash ou eprom. Possui debugger por software por ferramentas internas como o MPLAB SIM ou plugins externos como o Proteus (programa de computador) VSM® e Matlab, ou ainda por modulos externos como PICKIT 3 utilizado neste projeto, que faz o debugger no proprio microcontrolador enviando os valores dos registradores para o computador possibilitando a analise dos valores na memória RAM e avaliação do desempenho do programa. Para programação dos microprocessadores, o MPLAB possui interface de programação possibilitando a programação diretamente do MPLAB para a memoria flash do micro controlador através da usb pela interface PICKIT 3.

3.9 PIC C Compiler

Foi utilizado para a elaboração da programação do microcontrolador PIC o compilador PIC C Compiler da CSS,Inc. É um programa que apresenta uma grande variedade de ferramentas para o desenvolvimento e depuração de aplicativos embutidos em execução nos microcontroladores PIC.

De acordo com Corteletti (2013), o PIC C Compiler compila códigos gravados com a extensão ‘.c’. Após compilação oito arquivos com o mesmo nome são gerados no mesmo local, com as seguintes extensões: ‘.cof’, ‘.err’, ‘.hex’, ‘.lst’, ‘.pjt’, ‘.sta’, ‘.sym’, ‘.tre’. O programa além de compilar, mostra avisos e erros e a previsão do uso da memória RAM e ROM do microcontrolador. Após compilação, o arquivo ‘.hex’ será utilizado, tanto na simulação no programa ISIS Proteus, como na gravação no microcontrolador PIC.

3.10 PICKIT 3

Para fazer o debugger do programa neste projeto foi utilizado o PICKit 3,mostrado na figura 19.

Figura 19 - PICKit 3

Fonte: Sagitron.com

O PICKit 3 é a ferramenta de depuração e programação mais simples e de menor custo da Microchip. É totalmente suportada pelo ambiente de desenvolvimento MPLAB IDE e possui uma simples conexão USB Full Speed com o computador que não só permite programar e depurar, como também fazer atualização do firmware interno do PICKit3. (Microchip, 2009, p. 10).

Possui circuitos de proteção de sobre tensão e de curto-circuito, permite a execução em tempo real e suporta tensões desde 2.0V. Respeitando a norma USB, o PICKit3 pode fornecer 100mA ao circuito onde esta ligado e possui LEDS de informação rápida para o utilizador.

Além destas funcionalidades, existe uma muito interessante e em alguns ambientes extremamente útil do PICKit3 que é a funcionalidade “Programmer-To-Go”. A funcionalidade “Programmer-To-Go” permite de uma forma muito simples programar qualquer microcontrolador das famílias PIC16, PIC18, PIC24, dsPIC33F e PIC32 da Microchip sem necessitar de um computador, o que é muito útil se está em um ambiente onde levar um computador não é prático ou totalmente impossível.

O PICKit3 permite guardar um código de até 512KB na sua flash interna a partir do MPLAB IDE e depois permite levar o PICKit3 e de forma autónoma, apenas alimentando o PICKit3 desde uma conexão USB.

3.11 Fonte de Alimentação

Para alimentação elétrica da plataforma robótica foram utilizadas 2 baterias de 6V/ 4Ah, ligadas em série, que ligadas em série geram uma tensão de 12V.

A figura 20 mostra o modelo de baterias utilizadas.

Figura 20 - Bateria selada 6V/ 4Ah

Fonte: Proesi.com

A grande vantagem de se utilizar as duas baterias é a estabilidade da plataforma devido ao peso.

Para o circuito do controle utilizou-se uma bateria alcalina de 9V.

IV IMPLEMENTAÇÃO

Neste capítulo são apresentados tópicos fundamentais para compreensão geral da implementação do projeto, foram definidas as etapas necessárias para isso, que são:

- Modelagem do sistema;
- Elaboração do circuito para o módulo Xbee do Robô;
- Elaboração do circuito para o controle remoto;
- Elaboração do código fonte para o microcontrolador PIC;
- Elaboração do código fonte para o arduino;
- Montagem dos circuitos nas placas;
- Montagem do protótipo;

Na figura 21 é mostrado o protótipo em sua fase final.

Figura 21 - Robô em Fase Final

Fonte: O Autor

4.1 Visão Geral do Projeto

O projeto basicamente é constituído por sensor de gás (MQ-4), sensor ultrassônico, ponte H(L293d), motores CC, e um módulo Xbee que são conectados diretamente no Arduino Uno R3, com microcontrolador ATMega 328, assim como um Display de 16x2 e um módulo XBee que são conectados diretamente no PIC 18F23K20. Possui ainda baterias, regulador de tensão e demais componentes eletrônicos, tais como resistores, capacitores, oscilador de cristal, led's, diodos e outros.

O diagrama em blocos do projeto é ilustrado na Figura 22.

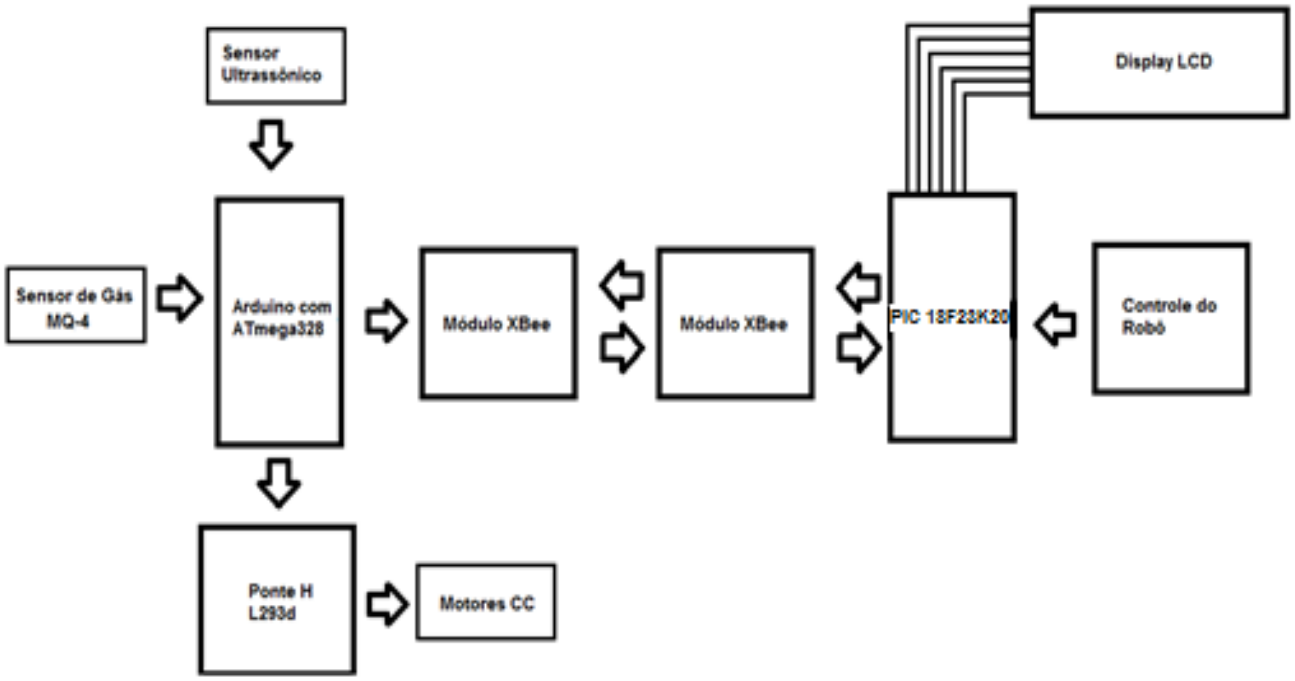

Figura 22 - Diagrama em Blocos do Projeto

Fonte: O autor

O diagrama esquemático do projeto proposto pode ser observado na Figura 23, na qual representa de forma objetiva a composição geral do projeto.

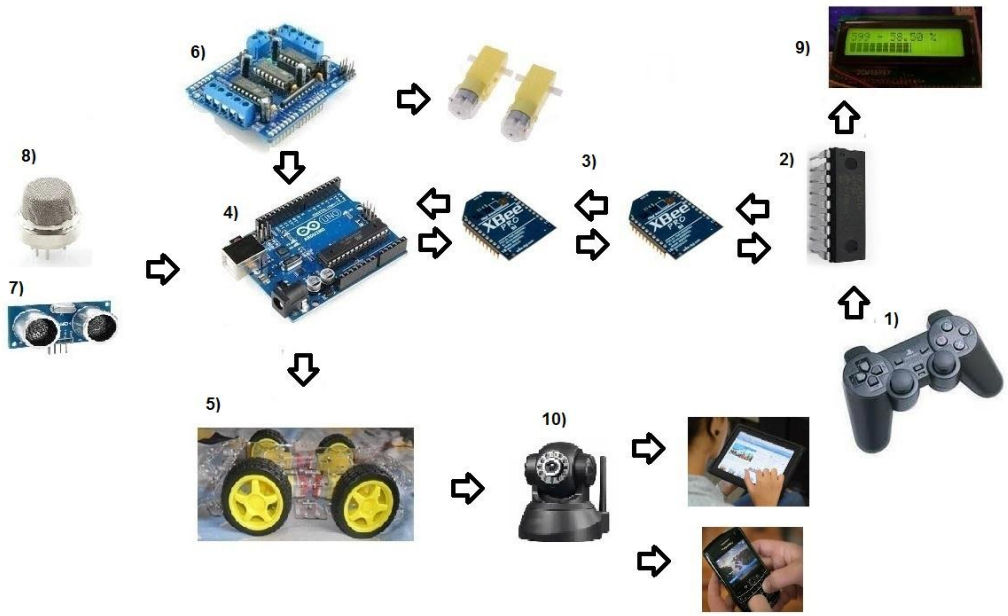

Figura 23 - Diagrama Esquemático do Projeto

Fonte: O autor.

1) Através do controle sem fio, manda-se sinal para o microcontrolador PIC, que comunica-se com o Arduino através de módulo ZigBee, para controlar o robô.

2) O PIC recebe o sinal para controle do robô , e transmite para o microcontrolador do Arduino, ao mesmo tempo que recebe dados do Arduino referente aos dados do sensor de gás.

3) Os Módulos ZigBee recebem o sinal do controle e enviam para o Arduino e vice e versa.

4) O Arduino é responsável por receber os dados do sensor de temperatura e enviar a informação para o display, e receber também a informação do controle, assim como receber sinal do sensor ultrassónico para se guiar no espaço.

5)A plataforma abriga o Arduino, a ponte H, os sensores ultrassônico e de gases, os motores CC, e a câmera IP.

6)A ponte H é responsável por fazer o controle dos motores.

7) O sensor ultrassônico serve para prevenir possíveis colisões.

8) O sensor de gases é o principal sensor do projeto e tem como função detectar gases no ambiente.

9) O display recebe a informação em porcentagem do nível de gás detectado no ambiente.

10) A câmera IP pode ser acessada de qualquer rede, via celular, tablet, computador, para obter imagens em tempo real do trajeto do robô.

4.2 Elaboração dos circuitos

Após determinar o que teria no projeto e ser feito o esquema do projeto o *software* Proteus foi utilizado para simulação do funcionamento do circuito e para projetar a placa.

A figura 24 mostra a simulação feita no software ISIS.

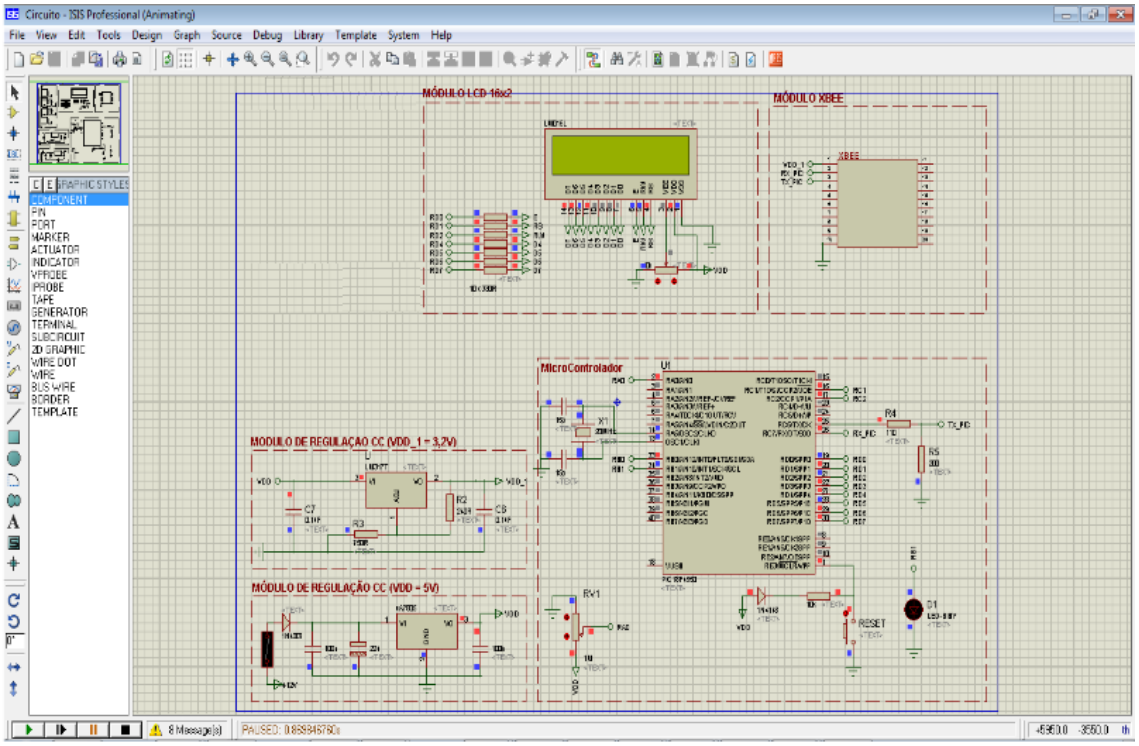

Figura 24 - Simulação no Proteus ISIS

Fonte: O Autor

4.2.1 Simulação do circuito do controle no software Proteus ISIS 7.7 Professional

O *software* ISIS permite o desenho de circuitos empregando um entorno gráfico no qual é possível simular os componentes e realizar a simulação do seu funcionamento sem correr o risco de ocasionar danos aos circuitos. Nele foi montado o circuito do projeto contendo todos os componentes do controle, com exceção do sensor de gás e sensor ultrassônico que foi simulado diretamente no ARDUINO.

Para realizar a simulação no *software* ISIS foi necessário realizar a programação do microcontrolador PIC1823k20 que será apresentada a seguir no 4.3, que trata da elaboração do código fonte.

4.2.2 Projeto das placas de circuito impresso (PCB)

Com o *software* Proteus ARES Professional é possível projetar o circuito e as trilhas da placa de circuito impresso (PCB). O principal cuidado que se deve ter no momento de planejar a placa são as ligações das trilhas dos componentes para não ocorrer um curto circuito e ocasionar o mau funcionamento da placa. Na Figura 25 é mostrada a placa projetada no *software* ARES para o controle do robô.

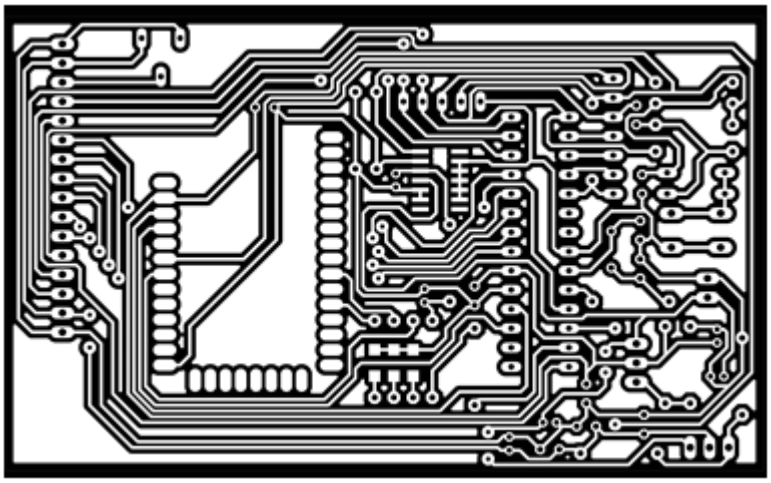

Figura 25 - Circuito impresso do controle do robô

Fonte: O Autor

O circuito impresso do controle do robô contém as trilhas e conexões necessárias para a integração e comunicação entre o módulo Xbee, e o microcontrolador PIC, além dos reguladores de tensão, resistores e outros componentes necessários.

Na Figura 26 é mostrada a placa projetada no *software* ARES para o módulo Xbee que faz parte do robô.

Este módulo faz a comunicação entre o Arduino e o controle, através do outro módulo Xbee presente no controle do robô.

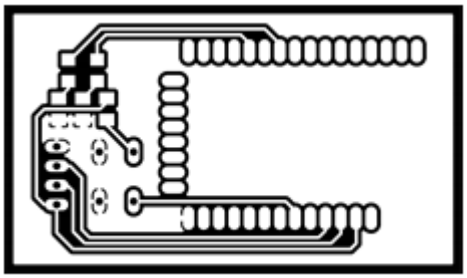

Figura 26 - Circuito impresso módulo Xbee

Fonte: O Autor

4.3 Elaboração do Código Fonte

Para realizar a simulação no *software* ISIS foi necessário elaborar o código fonte do projeto, esta programação contém todas as funções do protótipo, e foi gravada no PIC18F23k20 pelo PICKit 3. Para gerar o código em ".hex" foi utilizado o PIC C compiler e aos poucos o código fonte foi sendo gerado para o devido funcionamento dos componentes.

Já para a o código para o hardware da plataforma robótica foi escrito e feito o debugger no ambiente de programação do Arduino.

4.3.1 Código Fonte escrito no ARDUINO

A seguir será explicada a escrita do código fonte, com as funções do programa escrito no Arduino para o devido funcionamento do projeto com todos seus componentes.

```
//*********************************************************************************************
//*                        código fonte da plataforma Robótica                                  *
//*********************************************************************************************
#include <AFMotor.h>
#include <Ultrasonic.h>

#define echoPin 13 // pino 13 recebe o pulso do echo do sensor ultrassônico
#define trigPin 12 // pino 12 envio o pulso para gerar o echo do sensor ultrassônico

// Declaração da variáveis
int valor=0, velocidade=0, cont=0, vetor[20]={0}, check=0, cont2=0, nivel_gas=0, checksum=0;
int vet[25]={0}, dado[6]={0};
long duracao=0, distancia=0, valor_sensor=0;

AF_DCMotor motor1(1); //Seleciona o motor 1
AF_DCMotor motor2(2); //Seleciona o motor 2
AF_DCMotor motor3(3); //Seleciona o motor 1
AF_DCMotor motor4(4); //Seleciona o motor 2
```

Figura 27 - Código Fonte ARDUINO parte I

Fonte: O Autor

Na Figura 27 é mostrado o início da escrita do código fonte, onde as bibliotecas

necessárias para o funcionamento do programa foram declaradas, foram definidos quais seriam os pinos utilizados para o sensor ultrassônico, as variáveis foram declaradas, e os motores definidos.

```
//*************************************************************************************************************
//*                                          Sensor Ultrassônico                                              *
//*************************************************************************************************************
void setup()
{
  Serial.begin(9600);
  pinMode(echoPin, INPUT); // Define o pino 13 como entrada (recebe)sinal
  pinMode(trigPin, OUTPUT); // Define o pino 12 como saida (envia) sinal
}
void loop()
{
     //Seta o pino 12 com um pulso baixo "LOW" ou desligado ou ainda 0
    digitalWrite(trigPin, LOW);
    // delay de 2 microsegundos
    delayMicroseconds(2);
    //Seta o pino 12 com pulso alto "HIGH" ou ligado aou ainda 1
    digitalWrite(trigPin, HIGH);
    // delay de 10 microsegundos
    delayMicroseconds(10);
    // Seta o pino 12 como baixo novamente
    digitalWrite(trigPin, LOW);
    //pulseIN lê o tempo entre a chamada e o pino entrar em high
    duracao= pulseIn(echoPin, HIGH);
    // esse cálculo é baseado na seguinte fórmula: Distânica= tempo echo em nível alto * velocidade do som / 2,
    //sendo que a divisão ja foi feita previamente, restando ja o valor, lembrando que o tempo vem dobrado porque
    // tem que se considerar o tempo de ida e volta
    distancia= duracao*0.017;
    //espera 1 segundo para fazer a leitura novamente
    delay(1000);
```

Figura 28 - Código fonte ARDUINO parte II

Fonte: O Autor

Na figura 28 é mostrado a parte do código que trata o sinal do sensor ultrassônico, levando-se em consideração que para começar a medição é necessário alimentar o módulo e colocar o pino Trigger em nível alto por mais de 10us. Assim o sensor emitirá uma onda sonora que ao encontrar um obstáculo rebaterá de volta em direção ao módulo, sendo que o neste tempo de emissão e recebimento do sinal o pino Echo ficará em nivel alto. Logo o calculo da distância pode ser feito de acordo com o tempo em que o pino Echo permaneceu em nível alto após o pino Trigger ter sido colocado em nível alto.

```
// ***************************************************************************************************************
// *                                                sensor de gás                                                *
// ***************************************************************************************************************
// rebebe o valor do sensor de gás
valor_sensor = analogRead(0);
// guarda o valor do sensor em uma variavel, sabendo-se que a resolução do sensor vai até 1024,convertendo em um
//valor proporcional.
 nivel_gas=((valor_sensor*100)/1024);
```

Figura 29 - Código Fonte ARDUINO parte III

Fonte: O Autor

Na figura 29 é mostrado a aquisição do valor do sensor de gás que será enviado posteriormente ao microcontrolador do controle onde será tratado como um valor proporcional, para exibi-lo no display LCD.

```
//***************************************************************************************************************
//*                                  Aquisição dos dados recebidos do controle                                  *
//***************************************************************************************************************
// aguarda o recebimento dos pacotes de dados enviados pelo controle
if(Serial.available())
{
    //armazena os dados que estão no buffer da serial para posteriormente tratar essa informação
    for(cont=0; cont<=24; cont++)
    {
        vetor[cont]=Serial.read();
    }
    if(vetor[0]==0x7E)
    {
      // verifica se os dados foram recebidos corretamente
      for(cont2=3;cont2<=vetor[2];cont2++)
      {
        check=check+vetor[cont2];
      }
      if (check==0xff)
      {    //armazena os dados do buffer
          dado[0]=vetor[17];
          dado[1]=vetor[18];
          dado[2]=vetor[19];
          dado[3]=vetor[20];
          dado[4]=vetor[21];
          dado[5]=vetor[22];

          checksum=(0xFF - (0x10 + 0x13 + 0xA2 + 0x40 + 0x66 + 0x9E + 0x6F + 0xFF + 0xFE + nivel gas)); // Definição do Valor de checksum
```

Figura 30 - Código Fonte do ARDUINO parte IV

Fonte: O Autor

Na figura 30 é mostrada a parte do código que faz a aquisição dos dados recebidos do controle e guarda em um vetor para posteriormente passar como parâmetro o valor da velocidade e direção dos motores.

A figura 31 contém o código de envio do pacote contendo o valor do sensor de gases que posteriormente será tratado no código do controle.

```
//Envio do Pacote de transmissão ao controle contendo o valor do sensor de gas
Serial.write(0x7E);
Serial.write(0x00);
Serial.write(0x0F);
Serial.write(0x10);
Serial.write(0x01);
Serial.write(0x00);
Serial.write(0x13);
Serial.write(0xA2);
Serial.write(0x00);
Serial.write(0x40);
Serial.write(0x66);
Serial.write(0x9E);
Serial.write(0x6F);
Serial.write(0xFF);
Serial.write(0xFE);
Serial.write(0x00);
Serial.write(0x00);
Serial.write(nivel_gas);
Serial.write(checksum);
```

Figura 31 - Código Fonte do ARDUINO parte V

Fonte: O Autor

```
//*************************************************************************************************
//*                                  Controle dos Motores                                         *
//*************************************************************************************************
if(distancia<20) //Verificação da distancia atraves do sensor ultrassonico para bloqueio do veiculo
{
  velocidade=0;
  motor1.setSpeed(velocidade); //Define a velocidade do motor 1
  motor2.setSpeed(velocidade); //Define a velocidade do motor 2
  motor3.setSpeed(velocidade); //Define a velocidade do motor 3
  motor4.setSpeed(velocidade); //Define a velocidade do moror 4

  motor1.run(RELEASE); // motor pára
  motor2.run(RELEASE); // motor pára
  motor3.run(RELEASE); // motor pára
  motor4.run(RELEASE); // motor pára
  if(dado[2]<120)
  {
    motor1.run(FORWARD); //Gira o motor sentido horario
    motor4.run(FORWARD); //Gira o motor sentido horario
    motor3.run(BACKWARD); //Gira o motor sentido anti-horario
    motor2.run(BACKWARD); //Gira o motor sentido anti-horario
  }else if(dado[2]<136)
  {
    motor2.run(FORWARD); //Gira o motor sentido horario
    motor3.run(FORWARD); //Gira o motor sentido horario
    motor1.run(BACKWARD); //Gira o motor sentido anti-horario
    motor4.run(BACKWARD); //Gira o motor sentido anti-horario
  }else
  {
```

```
        motor1.run(FORWARD); //Gira o motor sentido horario
        motor2.run(FORWARD); //Gira o motor sentido horario
        motor3.run(FORWARD); //Gira o motor sentido horario
        motor4.run(FORWARD); //Gira o motor sentido horario
      }
      velocidade=125-dado[2]; //Define a velocidade em função da posição do controle de comando

      motor1.setSpeed(velocidade); //Define a velocidade do motor 1
      motor2.setSpeed(velocidade); //Define a velocidade do motor 2
      motor3.setSpeed(velocidade); //Define a velocidade do motor 3
      motor4.setSpeed(velocidade); //Define a velocidade do moror 4
    }else if(dado[3]>131)
    {
      if(dado[2]<120)
      {
        motor2.run(FORWARD); //Gira o motor sentido horario
        motor3.run(FORWARD); //Gira o motor sentido horario
        motor1.run(BACKWARD); //Gira o motor sentido anti-horario
        motor4.run(BACKWARD); //Gira o motor sentido anti-horario
      }else if(dado[2]<136)
      {
       motor1.run(FORWARD); //Gira o motor sentido horario
       motor4.run(FORWARD); //Gira o motor sentido horario
       motor3.run(BACKWARD); //Gira o motor sentido anti-horario
       motor2.run(BACKWARD); //Gira o motor sentido anti-horario
     }else
     {
       motor1.run(BACKWARD); //Gira o motor sentido anti-horario
       motor2.run(BACKWARD); //Gira o motor sentido anti-horario
       motor3.run(BACKWARD); //Gira o motor sentido anti-horario
       motor4.run(BACKWARD); //Gira o motor sentido anti-horario
     }
     velocidade=dado[2]-131; //Define a velocidade em função da posição do controle de comando

     motor1.setSpeed(velocidade); //Define a velocidade do motor 1
     motor2.setSpeed(velocidade); //Define a velocidade do motor 2
     motor3.setSpeed(velocidade); //Define a velocidade do motor 3
     motor4.setSpeed(velocidade); //Define a velocidade do moror 4
   }else
   {
        velocidade=0;
        motor1.setSpeed(velocidade); //Define a velocidade do motor 1
        motor2.setSpeed(velocidade); //Define a velocidade do motor 2
        motor3.setSpeed(velocidade); //Define a velocidade do motor 3
        motor4.setSpeed(velocidade); //Define a velocidade do moror 4

        motor1.run(RELEASE); //motor pára
        motor2.run(RELEASE); //motor pára
        motor3.run(RELEASE); //motor pára
        motor4.run(RELEASE); //motor pára
      }
    }
}
```

Figura 32 - Código Fonte ARDUINO parte VI

Fonte: O Autor

A figura 32 contém todo o código para o controle da direção e velocidade dos motores, a partir dos dados recebidos do controle.

4.3.2 Código Fonte escrito no microcontrolador PIC 18F23K20

A seguir será explicada a escrita do código fonte no microcontrolador PIC , com as funções do programa para o devido funcionamento do projeto com todos seus componentes.

```
//*****************************************************************************
//*                        Cóigo Fonte do controle do robô                    *
//*****************************************************************************

#include "D:\Globalteck\Project\Projetos\Carrinho Moises\Controle\V0.1.h"

//Declaração das Variávies do Tipo Bit
intl M0=0, M1=0, comunica=0;

//Declaração das Variávies do Tipo Inteiro
int pacote[5]={0x01,0x42,0xFF,0xFF,0xFF}, API[25]={0}, dados[9]={0};
int cont=0, cont_1=0, API_1=0, checksum=0, checksum_1=0;
int mascara=0bllll0000;

//Atribuição ao pino de saída à um nome de Referência
#define lcdrs pin_c1
#define lcdrw pin_c3
#define lcden pin_c0

//Atribuição a um valor um nome de Referência
#define apaga   0x01
#define esp     0x14
#define segunda 0xc0

//Rotina de Tratamento da Interrupção do Temporizador TIMER0
#int_TIMER0
void  TIMER0_isr(void)
{
//Limpar o Bit do Registrador de Interrupção do TIMER0
   clear_interrupt(INT_TIMER0);
}

//Rotina de Tratamento da Interrupção da Comunicação Serial
#int_RDA
void  RDA_isr(void)
{
   API[API_1]=fgetc(PORT1);
   clear_interrupt(int_RDA);
   API_1++;
}

//Rotina de envio dos dados para habilitar o LCD
void envia (void){
   delay_us(2);
   output_high(lcden);
   delay_us(2);
   output_low(lcden);
   delay_us(2);
}
```

Figura 33 - Código fonte PIC parte I

Fonte: O Autor

A figura 33 contém parte do código onde são declaradas as variáveis, e os tratamentos necessários para o funcionamento do display LCD.

```
//Rotina de envio dos dados para a inicialização do LCD
void init (void )
{
   output_low(lcdrs);
   output_b(0b00111000);
   envia();
   output_high(lcdrs);
   delay_ms(5);

   output_low(lcdrs);
   output_b(0b00111000);
   envia();
   output_high(lcdrs);
   delay_ms(5);

   output_low(lcdrs);
   output_b(0b00111000);
   envia();
   output_high(lcdrs);
   delay_ms(5);

   output_low(lcdrs);
   output_b(0b00111000);
   envia();
   output_high(lcdrs);
   delay_us(2);

   output_low(lcdrs);
   output_b(0b00001100);
   envia();
   output_high(lcdrs);
   delay_us(2);

   output_low(lcdrs);
   output_b(0b00000001);
   envia();
   output_high(lcdrs);
   delay_us(2);

   output_low(lcdrs);
   output_b(0b00000111);
   envia();
   output_high(lcdrs);

 }
 //Rotina de envio dos comandos para o LCD
void enviac (unsigned int comando){

   output_low(lcdrs);
   output_low(lcdrw);
   OUTPUT_B(comando);
   envia();
   //output_high(lcdrs);
 }

//Rotina de envio dos dados para o LCD
void enviad (unsigned int dado){
   output_high(lcdrs);
   OUTPUT_B(dado);
   envia();
   //output_low(lcdrs);
   //OUTPUT_B((dado<<4)&mascara);
  // output_high(lcdrs);
   //envia();
}
```

Figura 34 - Código Fonte do PIC parte II

Fonte: O Autor

Na figura 34 é mostrado o código necessário para a inicialização do display LCD, bem como o envio de comandos e dados.

```
//Rotina Principal
void main()
{
   //Definição das Entradas Analógicas (Sem entradas analógicas | Valor de referência de 0 a VCC)
   setup_adc_ports(NO_ANALOGS|VSS_VDD);

//Definição do Conversor ADC (Conversor Desabilitado | Tempo de Aquisição do Conversor no minímo)
   setup_adc(ADC_OFF|ADC_TAD_MUL_0);

//Comunicação SPI Desabilitada
   setup_spi(SPI_SS_DISABLED);

//Timer Watch Dog Desabilitado
   setup_wdt(WDT_OFF);

//Configuração do Temporizador TIMER0 (Temporizador Interno | Resolução de 0,2uS | Contagem em 8 bits) Overflow 51,2uS
   setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1|RTCC_8_bit);

//Configuração do Temporizador TIMER1 (TIMER1 Desabilitado)
   setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);      //104 ms overflow


   //Configuração do Temporizador TIMER2 (Resolução de | Overflow em 400uS | Interrupção em 400uS)
   setup_timer_2(T2_DIV_BY_16,124,1);


   //Configuração dos Módulos de Compardação - Desabilitado
      setup_comparator(NC_NC_NC_NC);// This device COMP currently not supported by the PICWizard

   //Habilita Interrupção do TIMER0
   enable_interrupts(INT_TIMER0);

   //Habilita Interrupção do TIMER1
   enable_interrupts(INT_TIMER1);

   //Habilita Interrupção da Comunicação Serial
   enable_interrupts(INT_RDA);


      //Habilita as Interrupção
   enable_interrupts(GLOBAL);

   Config tab
      delay_ms(20);
      output_low(lcden);
      output_low(lcdrs);
      output_low(lcdrw);
      init();
      enviac (0x02);
      enviac (0x01);
```

Figura 35 - Código fonte do PIC parte III

Fonte: O autor

A figura 35 mostra o código da rotina principal onde são feitas as configurações do microcontrolador e do display LCD.

```
//*************************************************************************************************
//*                    Aquisição dos dados recebidos do sensor de gases                   *
//*************************************************************************************************
 While(true)
 {
    if(API[0]==0x7E)
    {
       if(API_l>(API[2]+3) && API[2]!=0)
       {
          for(check=3; check<=(API[2]+3); check++)
          {
             checksum=checksum+API[check];
          }
          if(checksum==0xFF)
          {
             switch(API[3])
             {
                case 0x90:
                {
                   valor_sensor=API[17];
                }break;
             }
          }else
          {
             checksum=0;
             API_l=0;
          }
       }

       }else
       {
          API_l=0;
       }
```

Figura 36 - Codigo fonte do PIC parte IV

Fonte: O Autor

Na figura 36 pode-se verificar o código que é responsável pela aquisição dos dados recebidos do sensor de gases que fica na plataforma robótica. Os dados já são recebidos em porcentagem, devido ser tratados anteriormente no código do Arduino.

```
//*****************************************************************************************
//*          Rotina de Gerenciamento e Aquisição dos dados do Controle                    *
//*****************************************************************************************

output_low(pin_a1);
for(cont=0;cont<9;cont++)
{
   set_timer0(0);
   clear_interrupt(INT_TIMER0);
   enable_interrupts(INT_TIMER0);
   cont_1=0;
   for(cont_1; cont_1<8;)
   {
      if(!input_state(pin_a0) && !M0)
      {
         output_bit(pin_a2,bit_test(pacote[cont],cont_1));
         if(input(pin_a3))
         {
            bit_set(dados[cont],cont_1);
         }else
         {
            bit_clear(dados[cont],cont_1);
         }
         M0=1;
      }else if(input_state(pin_a0) && M0)
      {
         cont_1++;
         M0=0;
         output_high(pin_a2);
         disable_interrupts(INT_TIMER0);
         //delay_ms(1);
      }
```

Figura 37 - Código fonte PIC parte V

Fonte: O Autor

A figura 37 mostra o código para aquisição dos dados recebidos do controle. Como só foi usado um botão analógico do controle, o microcontrolador recebe os dados desse botão e envia através do módulo xbee para o Arduino.

```
// envia o texto da primeira linha para o display LCD
output_high(pin_a1);

enviac (0x02);
enviac (apaga);
enviad ('n');
enviad ('i');
enviad ('v');
enviad ('e');
enviad ('l');
enviac (esp);
enviad ('d');
enviad ('e');
enviac (esp);
enviad ('g');
enviad ('a');
enviad ('s');
```

Figura 38 - código fonte PIC parte VI

Fonte: O autor

Na figura 38 está a parte do texto no display LCD, a primeira linha terá sempre o mesmo texto.

```
    // envia o texto da primeira linha para o display LCD
    output_high(pin_a1);

    enviac (0x02);
    enviac (apaga);
    enviad ('n');
    enviad ('i');
    enviad ('v');
    enviad ('e');
    enviad ('l');
    enviac (esp);
    enviad ('d');
    enviad ('e');
    enviac (esp);
    enviad ('g');
    enviad ('a');
    enviad ('s');
 // envia o texto da segunda linha do display se o valor do sensor for menor ou igual a 200 p.p.m. (valor ja tratado no arduino)
 enviac (segunda);
 if(valor_sensor<=2)
 {
    enviac (esp);
    enviac (esp);
    enviac (esp);
    enviac (esp);
    enviac (esp);
    enviad ('N');
    enviad ('o');
    enviac ('r');
    enviad ('m');
    enviad ('a');
    enviad ('l');
   // envia o texto da segunda linha do display se o valor do sensor for menor ou igual a 1000 p.p.m. (valor ja tratado no arduino)
}else if(2<valor_sensor<=10)
{
   enviac (esp);
   enviac (esp);
   enviac (esp);
   enviad ('T');
   enviad ('o');
   enviac ('l');
   enviad ('e');
   enviad ('r');
   enviad ('a');
   enviad ('v');
   enviad ('e');
   enviad ('l');
    // envia o texto da segunda linha do display se o valor do sensor for menor que 2000 p.p.m. (valor ja tratado no arduino)
 }else if(10<valor_sensor<20)
 {
    enviac (esp);
    enviac (esp);
    enviac (esp);
    enviac (esp);
    enviad ('P');
    enviad ('e');
    enviac ('r');
    enviad ('i');
    enviad ('g');
    enviad ('o');
    enviad ('s');
    enviad ('o');
   // envia o texto da segunda linha do display se o valor do sensor for maior ou igual a 2000 p.p.m. (valor ja tratado no arduino)
}else if(valor_sensor>=20)
{
   enviac (esp);
   enviac (esp);
   enviad ('P');
   enviad ('e');
   enviac ('r');
   enviad ('i');
   enviad ('g');
   enviad ('o');
   enviad ('s');
   enviad ('i');
   enviad ('s');
   enviad ('s');
   enviad ('i');
   enviad ('m');
   enviad ('o');
}
```

Figura 39 - Código fonte do PIC parte VII

Fonte: O Autor

Na figura 39, onde encontra-se o código para envio do texto para a segunda linha do display, depende do valor lido no sensor de gases. Se o nível de gás for menor ou igual a 200 p.p.m então escreve “Normal”, se o nível de gás for menor ou igual a 1000 p.p.m. então escreve “Tolerável”, se for menor que 2000 p.p.m então escreve “perigoso”, se for maior ou igual a 2000 p.p.m então escreve “perigosíssimo”.

```
// envia dados do controle para a plataforma robótica
checksum=0xFF-(0x10+0x01+0x13+0xA2+0x40+0x66+0x9F+0x02+0xFF+0xFE+dados[3]+dados[4]+dados[5]+dados[6]+dados[7]+dados[8]);
if(comunica)
{
   putc(0x7E);
   putc(0x00);
   putc(0x14);
   putc(0x10);
   putc(0x01);
   putc(0x00);
   putc(0x13);
   putc(0xA2);
   putc(0x00);
   putc(0x40);
   putc(0x66);
   putc(0x9F);
   putc(0x02);
   putc(0xFF);
   putc(0xFE);
   putc(0x00);
   putc(0x00);
   putc(dados[3]);
   putc(dados[4]);
   putc(dados[5]);
   putc(dados[6]);
   putc(dados[7]);
   putc(dados[8]);
   putc(checksum);
```

Figura 40 - Código Fonte PIC parte VIII

Fonte: O Autor

Na figura 40 está o código de envio dos dados do controle para o Arduino, através do módulo Xbee. O Arduino por sua vez receberá estes dados e passará como parâmetro para a referencia de velocidade e direção dos motores.

4.4 Montagem do protótipo

A plataforma robótica já foi comprada no formato atual com duas chapas de acrílico, sendo possível criar dois andares, no de baixo foi acondicionado as baterias de 6V / 4Ah e em cima a câmera IP, o ARDUINO, o módulo Xbee, e o sensor ultrassônico que foi instalado na frente, assim como o sensor de gases que foi instalado na parte traseira. Na plataforma foi ainda montado o conjuntos de 4 motores CC com redução.

Para o controle foi utilizado um controle de PlayStation 2, sendo necessário descobrir o seu funcionamento na parte de comunicação e desenvolver uma placa para a comunicação com o robô, que foi encaixada na parte de trás do controle. Nessa placa também foi colocado o display LCD que recebe o nível de gás detectado. A figura 41 mostra o controle em sua fase final.

Figura 41 - Controle do Robô em fase final

Fonte: O Autor

O analógico do controle quando está no centro sempre enviará um valor de 127 para o microcontrolador, se estiver totalmente para a direita enviará 0, e se estiver totalmente para a esquerda enviará 255, assim também para o controle da direção, sendo 0 para baixo, 127 no centro e 255 totalmente para cima.

Esses valores foram transmitidos para o Arduino através do xbee e referenciam a velocidade e direção no código do Arduino.

V RESULTADOS OBTIDOS

Neste capítulo é apresentado todo o desenvolvimento das simulações feitas com o projeto, suas funções e todos os problemas ocorridos durante o período de desenvolvimento.

5.1 Simulações

Como dito no item 4.2.1, do capítulo 4, onde é explicado o *software* ISIS, inicialmente a simulação do funcionamento do projeto foi realizada neste simulador para evitar danos à placa devido a conexões erradas ou instalação de componentes inadequados. Após a simulação feita no ISIS e a placa confeccionada, os testes reais no protótipo foram realizados e pequenos problemas foram ajustados.

A programação foi testada inúmeras vezes, sendo necessário o uso do PICKit 3 para a gravação do PIC. O código fonte foi alterado em vários momentos até conseguir o funcionamento correto do controle, da comunicação entre os módulos Xbee, da detecção dos gases e do controle correto dos motores do robô, utilizando também o sensor ultrassônico para evitar colisões com outros objetos. Os testes da comunicação com o ARDUINO foram realizadas através do SERIAL MONITOR do ARDUINO, assim como os testes de mobilidade do robô e detecção de gases foram possíveis através da IDE do ARDUINO.

No código fonte do projeto foram definidos a porcentagem de gases para a apresentação de cada barra do Bar Graph criado no Display LCD, sendo que no display sempre ficará o nível real de gás detectado.

O dispositivo foi testado em uma cozinha residencial, e colocado perto do botijão de gás e funcionou conforme o esperado. Para a realização deste teste foi necessário folgar um pouco o regulador de gás para ocorrer um vazamento e assim poder testar o funcionamento do dispositivo.

A câmera IP apresentou um ótimo funcionamento quando acessada da rede local, tanto em ambiente claro quanto em escuro, e a distância máxima que alcançou foi de aproximadamente 50m sem obstáculos e aproximadamente 30m com obstáculos, devendo ter no ambiente repetidores wireless para não perder o sinal da câmera.

As baterias de 6V/4Ah não atingiram o objetivo quando ligadas em paralelo, pois a tensão não foi suficiente para o deslocamento do robô devido ao seu peso, sendo necessário fazer a alteração colocando as duas em série e conectando na entrada de tensão externa da

ponte H. Com essa alteração foi solucionado o problema, e os testes demonstraram que a autonomia do robô com a câmera ligada e o robô em deslocamento é de aproximadamente 24 horas.

A bateria de 9V do controle do robô não apresentou a autonomia desejada, devendo ser estudada a possibilidade de colocar uma bateria recarregável.

A comunicação entre o controle e o robô não apresentou problemas com relação a distância, pois isso já havia sido previsto no começo do projeto, sendo utilizado módulos com alcance de 1,5km.

5.2 Problemas Encontrados

O primeiro obstáculo encontrado foi a definição da plataforma. Primeiramente tentou-se a construção de uma plataforma de alumínio do tipo esteira, devido ao peso e a estabilidade para se manobrar o robô em ambientes adversos, porém mudou-se a ideia devido a grande complexidade mecânica de fabricação. Testou-se então o uso de rodas. Rodas são de longe o método mais popular de providenciar a locomoção de um robô e são devidamente usadas em muitos tamanhos e formas diferentes, rodas podem assumir qualquer tamanho, é bem fácil adaptá-las A plataforma escolhida, usa 4 rodas e têm a vantagem de usar múltiplos drivers para os motores, um driver conectado à cada roda proporcionando um bom desempenho, uma boa tração e um baixo coeficiente de deslizamento. Para mudar o sentido de deslocamento da plataforma, basta controlar via programação para que girem apenas as rodas do lado oposto ao que se quer andar.

Na confecção das placas de circuito impresso houve um problema na transferência das trilhas da impressão para a placa devido ao método de transferência utilizado. O método utilizado foi o de transferência térmica e acabou não funcionando muito bem, pois as trilhas da placa projetada ficaram muito próximas e com a pressão necessária do ferro de passar roupa, as trilhas se juntavam ocasionando curto circuito na placa. A solução foi um método mais atual de confecção de placas através de tinta fotossensível.

A aquisição dos componentes também não foi tão fácil, pois os fornecedores dos componentes foram todos de outro estado, devido à cidade ter um potencial muito pequeno na área, demorando na entrega, e ocasionando um atraso no projeto.

A placa do controle remoto também precisou ser refeita devido não ter sido previsto

um regulador de tensão para alimentação, para se obter uma tensão ideal do display LCD.

5.3 Orçamento do Projeto

Abaixo o Quadro 5 detalha os componentes e o seus devidos custos.

Quadro 5 – Orçamento do projeto

Item	Custo Unitário	Quantidade	Custo Total
Arduino UNO R3	R$ 49,89	1	R$ 49,89
Tinta Fotosensivel P/ Circuito Impresso (mascara) 50 Gr	R$ 24,50	1	R$ 24,50
Câmera Ip Ir Wireless Visão Noturna	R$ 124,90	1	R$ 124,90
Display Lcd 16x2 Escrita Preta Backlight Verde	R$ 12,99	1	R$ 12,99
Sensor Mq-4 Gás / Metano / Propano / Butano -	R$ 19,99	1	R$ 19,99
Sensor De Distância Ultrassônico Hc-sr04	R$ 9,99	1	R$ 9,99
Arduino L293d Motor Shield Ponte H Driver	R$ 29,99	1	R$ 29,99
Carrinho Chassi 4 Motores Robotica	R$ 128, 50	1	R$ 128, 50
Bateria Selada 6v 4ah	R$ 25,00	2	R$ 50,00
Bateria 9V	R$ 6,00	1	R$ 6,00
Placa de Fenolite cobreada	R$ 3,00	2	R$ 6,00
Microcontrolador PIC18F23k20	R$ 14,30	1	R$ 14,30
Capacitor cerâmico 100nF	R$ 0,07	6	R$ 0,42
Capacitor eletrolítico 10uF	R$ 0,08	2	R$ 0,16
Capacitor Cerâmico 18pF	R$ 0,07	2	R$ 0,14
Soquete CI 28 Pinos	R$ 0,41	1	R$ 0,41
Conector Barra de Pinos Macho	R$ 1,00	1	R$ 1,00
Conector Barra de Pinos Fêmea	R$ 1,00	1	R$ 1,00
Módulo Digi XBee ZB (S2C) - Antena RF Pad - 1 Mbps	R$ 89,90	2	R$ 179,80
Resistor de Filme de Carbono 10k – 1/4W	R$ 0,04	2	R$ 0,08
Resistor de Filme de Carbono 20k – 1/4W	R$ 0,04	2	R$ 0,08
Resistor de Filme de Carbono 1k – 1/4W	R$ 0,04	3	R$ 0,12
Resistor de Filme de Carbono 3k9 – 1/4W	R$ 0,04	1	R$ 0,04
Resistor de Filme de Carbono 33k – 1/4W	R$ 0,04	1	R$ 0,04
Resistor de Filme de Carbono 680 – 1/4W	R$ 0,04	1	R$ 0,04
Resistor de Filme de Carbono 68– 1/4W	R$ 0,04	3	R$ 0,12
Resistor de Filme de Carbono 220– 1/4W	R$ 0,04	2	R$ 0,08
Diodo N4148	R$ 0,03	2	R$ 0,06
CI LM324D	R$ 2,00	1	R$ 2,00
Regulador de tensão Lm317T	R$ 1,50	2	R$ 3,00
Cristal Quartzo 10MHz	R$ 1,00	1	R$ 1,00
borne Fêmea	R$ 0,50	2	R$ 1,00
Clip para bateria 9V	R$ 0,83	1	R$ 0,83
Resistor trimpot 5k	R$ 0,65	1	R$ 0,65
		Custo Total	R$ 540,70

VI CONSIDERAÇÕES FINAIS

Neste capítulo são apresentadas as conclusões e as propostas para trabalhos futuros, que podem utilizar este projeto como base.

6.1 Conclusões

Foi desenvolvido neste trabalho um dispositivo de detecção de vazamento de gás, com o diferencial de não estar fixo no ambiente, sendo possível monitorar todas as áreas através de um robô controlado remotamente, com imagens em tempo real do ambiente através de câmera que pode ter sua imagem acessada de qualquer computador conectado a rede, e com a função de um display no controle para registrar o nível de gás daquele ambiente. Com os testes realizados em situações de vazamento de gás de cozinha o projeto funcionou conforme o previsto, e desta forma pode ser usado para o devido fim, mas pode também detectar outros tipos de gases industriais através de sensores que podem ser instalados no robô.

O objetivo do projeto foi alcançado, o protótipo funciona conforme o planejado. Ocorreram alguns imprevistos durante o processo de confecção, mas isso foi superado. O hardware desenvolvido atendeu completamente as funções do protótipo e ainda poderiam ser adicionados outros componentes caso futuramente seja incluído ao projeto. O circuito foi projetado com sucesso e o funcionamento do mesmo foi excelente atendendo as expectativas criadas no inicio do projeto.

6.2 Propostas para trabalhos futuros

Uma boa modificação no projeto seria adicionar outros sensores para a detecção de outros tipos de gases industriais como gás amônia por exemplo.

Outro fato crucial se for necessário a visualização da imagem da câmera IP através de celular, ou de outra rede, é a aquisição de um IP fixo, e o redirecionamento no roteador. Muitas câmeras já possuem inclusive a função DDNS que poderá ser configurada.

A característica mais importante que deverá ser modificada no robô é a blindagem de

todos os componentes eletroeletrônicos, pois em um ambiente com risco de explosão, qualquer faísca proveniente dos circuitos poderia ser uma catástrofe.

Como o projeto é somente um protótipo não foi considerado a altura nem o tamanho da plataforma, sendo necessário dependendo do caso ter a distância exata a que os gases ficam e o posicionamento da câmera.

8. REFERÊNCIAS

[ANÔNIMO] **câmera IP como funciona utilização e funcionalidades.** Disponível em: <http://blog.projseg.com.br/index.php/2014/02/camera-ip-definicoes-funcionalidades-instalacao-utilizacao>. Acesso em: 12 de Outubro de 2014.

BRAGA, Newton. **Como funcionam os sensores ultrassônicos.** Disponível em: <http://www.newtoncbraga.com.br/index.php/como-funciona/5273-art691>. acesso em : 10 de outubro de 2014.

BRAGA, Newton. **Como funcionam os motores de corrente contínua.** Disponível em: <http://www.newtoncbraga.com.br/index.php/como-funciona/3414-art476a>. acesso em: 13 de outubro de 2014.

CALENTE, A; PIO, A; MENEZES, R. T. **técnicas preditivas ligadas à inspeção.** Disponível em: < http://www.abraman.org.br/arquivos/50/50.pdf> Acesso em: 13 de outubro de 2014.

COLBAND. **Butano C4H10.** Disponível em: <http://quimica.colband.net.br/files/2012/09/franco-1C4-butano.pdf> Acesso em: 14 de novembro de 2014.

CORTELETTI, DANIEL. **Manual de uso do compilador PCW.** Disponível em: < http://www.mecatronica.org.br/disciplinas/programacao/004/manual_ccs.pdf> Acesso em 15 de Outubro de 2014.

COSTA, Sandro. **O ensino através da pesquisa: uma proposta prática em base multidisciplinar.** Rio de janeiro, 2013. Disponível em: < http://www.if.ufrj.br/~pef/producao_academica/dissertacoes/2013_Sandro_Costa/dissertacao_Sandro_Costa.pdf >. Acesso em: 06 de Outubro de 2014.

G1 Notícias. Arquivo G1**: Explosão em academia no ABC deixa mortos e feridos, dizem bombeiros.** Publicado em 17 de Maio de 2014. Disponível em < http://g1.globo.com/sao-paulo/noticia/2014/05/explosao-em-academia-no-abc-deixa-feridos-e-morto-diz-bombeiro.html>. Acesso em 09 de setembro de 2014.

GIL, A, C. **Métodos e técnicas em pesquisa social**. 5.ed. São Paulo: Atlas, 1999.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4.ed. São Paulo: Atlas, 2002.

JUSTEN, Álvaro. **Curso de Arduino**. Rio de Janeiro, 2014. Disponível em:< http://www.cursodearduino.com.br/apostila/apostila-rev4.pdf>. Acesso em: 29 de Setembro de 2014.

NETO, MANOEL. **Práticas Arduino.** Disponível em:< www.wiki.ifba.edu.br/gsort/tiki-

download_file.php?fileId=301>. Acesso em: 02 de Outubro de 2014.

MARCONI, M. A. de.; LAKATOS, E. M. **Técnicas de Pesquisa**. 5. ed. São Paulo: Atlas, 2002.

MESSIAS, ANTÔNIO ROGÉRIO. **Controle remoto e aquisição de dados via xbee/zigbee (ieee 802.15.4).** Disponível em: < http://www.rogercom.com/zigbee/zigbee.htm>. Acesso em: 12 de Outubro de 2014.

NEWARK. **Microchip pic18f23k20-i/sp.** Disponível em: http://www.newark.com/microchip/pic18f23k20-i-sp/mcu-8bit-pic18-64mhz-dip-28/dp/27M0507?CMP=AFC-OP.> Acesso em :13 de Outubro de 2014.

NR-15. **atividades e operações insalubres** . Disponível em:< http://portal.mte.gov.br/data/files/FF8080812BE87BF5012BE893E4C762EF/nr_15_anexo11.pdf> Acesso em: 23 de outubro de 2014.

PETROBRÁS. **ficha de informação de segurança de produto químico – fispq.** Disponível em: <http://www.br.com.br/wps/wcm/connect/6c28fb0043e78342aec2efd1ce9b7c51/fispq-comb-gas propano.pdf?MOD=AJPERES> Acesso em: 20/11/2014.

ROBOCORE. **Sensor ultrassônico – HC-SR04.** Disponível em:< https://www.robocore.net/modules.php?name=GR_LojaVirtual&prod=620>. Acesso em: 10 de Setembro de 2014.

ROBOCORE. **Arduino UNO R3**. Disponível em:< https://www.robocore.net/modules.php?name=GR_LojaVirtual&prod=120>. Acesso em: 10 de Setembro de 2014.

SCHMIDT W. Materiais Elétricos Volume 1 – **Condutores e Semicondutores.** 3ª Ed. Blucher, 2010.

SILVA, E. L. MENEZES, E. M. Metodologia da pesquisa e elaboração de dissertação. 3. ed. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

THOMAZINI, D. ; ALBURQUERQUE, P. U. B. de. **sensores industriais: fundamentos e aplicações.** 4ª ed. Érica, 2005.

USBERCO & SALVADOR. **Química geral Ensino Médio** – volume 1. 12ª ed. Saraiva, 2006.