



**CENTRO UNIVERSITÁRIO DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS**  
**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**JEFFERSON SILVA SANTOS**

**DETECTOR DE VAZAMENTO DE GÁS COM AVISO POR SMS**

**Orientadora: Prof<sup>a</sup>. M.C. Maria Marony de Sousa Farias**

**BRASÍLIA**

**2º SEMESTRE DE 2012**

**JEFFERSON SILVA SANTOS**

**DETECTOR DE VAZAMENTO DE GÁS COM AVISO POR SMS**

**Trabalho apresentado ao UniCEUB – Centro  
Universitário de Brasília como pré-requisito pa-  
ra obtenção de Certificação de Conclusão do  
Curso de Engenharia de Computação.  
Orientadora: Prof<sup>a</sup>. M.C. Maria Marony Sousa  
Farias.**

**BRASÍLIA**

**2º SEMESTRE DE 2012**

**JEFFERSON SILVA SANTOS**

**DETECTOR DE VAZAMENTO DE GÁS COM AVISO POR SMS**

Trabalho apresentado ao UniCEUB – Centro  
Universitário de Brasília como pré-requisito pa-  
ra obtenção de Certificação de Conclusão do  
Curso de Engenharia de Computação.  
Orientadora: Prof<sup>a</sup>. M.C. Maria Marony Sousa  
Farias.

**Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro  
de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia  
e Ciências Sociais Aplicadas - FATECS.**

---

**Prof. Abiezer Amarília Fernandes**  
**Coordenador do Curso**

**Banca Examinadora:**

---

**Prof<sup>a</sup>. Maria Marony Sousa Farias, mestre  
em Engenharia Elétrica – UFPB – PB.**  
**Orientadora**

---

**Prof. Luciano Henrique Duque, mestre em  
Engenharia Elétrica.**

---

**Prof. Doutor Miguel Archanjo Bacellar Goes**  
**Telles Junior**

---

**Prof<sup>a</sup>. Doutora Karin Astrid Marques Santos**

*Dedico a todas as pessoas que contribuíram para a conclusão desta grande etapa.*

## **AGRADECIMENTOS**

Primeiramente a Deus. A toda minha família, pela motivação ao longo deste projeto, pela dedicação sempre buscando o melhor para mim. A todos os colegas e grandes amigos com quem estudei e principalmente aqueles que contribuíram diretamente ou indiretamente para o desenvolvimento deste projeto.

E principalmente a algumas pessoas especiais que mesmo tudo dando errado não pararam de me incentivar a continuar e a concluir esta grande etapa da minha vida.

Muito Obrigado!

*“Comece fazendo o que é necessário, depois o que é possível e de repente estará fazendo o impossível.”* **São Francisco de Assis.**

## RESUMO

Neste trabalho é apresentado um dispositivo de detecção de vazamento de gás com aviso por SMS. O sensor utilizado é o MQ-9, que faz a leitura da quantidade de gás no ambiente. O projeto integra um módulo GSM, microcontrolador PIC16F877A, o sensor de gás e display. No projeto, a detecção de gás é dividida em três níveis: O Primeiro é denominado “Nível Normal”, onde o valor medido no ambiente é exibido no display e aciona um Led verde. O segundo é denominado de “Vazamento Médio” onde além da exibição no display, do acionamento de um Led amarelo, um aviso sonoro é emitido. O terceiro é denominado como “Perigo! Gas!” que exibe no display, aciona um Led vermelho, um aviso sonoro e o envio de SMS a um celular previamente cadastrado, notificando a irregularidade na concentração de gás inflamável.

Palavras-chave: Vazamento, GSM, microcontrolador PIC16F877A, SMS.

## ABSTRACT

This work presents a sensing device with gas leak alert by SMS. The sensor used is a MQ-9, which reads the amount of gas in the environment. The project integrates a GSM module, microcontroller PIC16F877A, the gas sensor and display. In the project, the gas detection is divided into three levels: The first is called "Nível Normal", where the measured value in environment is shown in the display and activates a green LED. The second is called "Vazamento Médio" where besides the display the display, the activation of a yellow LED, an audible warning is issued. The third is termed as "Perigo! Gas!" showing on the display, activates a red LED, a buzzer and sending SMS to a mobile phone previously registered, notify the irregularity in the concentration of flammable gas.

Key-words: Leak, GSM, microcontroller PIC16F877A, Text messaging.



## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>XII</b>
<b>LISTA DE TABELAS E QUADROS.....</b>	<b>XIV</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>XV</b>
<b>CAPÍTULO 1 – INTRODUÇÃO .....</b>	<b>16</b>
1.1 Introdução ao Tema Proposto.....	16
1.2 Motivação .....	17
1.3 Objetivos.....	18
1.5 Estrutura da Monografia.....	19
<b>CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA.....</b>	<b>20</b>
2.1 Problemas de Saúde .....	20
2.2 Segurança Doméstica.....	20
2.3 Soluções Existentes.....	21
2.4 Benefícios do Dispositivo Proposto e Suas Restrições .....	21
<b>CAPÍTULO 3 – REFERENCIAL TEÓRICO.....</b>	<b>23</b>
3.1 Microcontroladores .....	23
3.1.1 Microcontroladores da família PIC .....	24
3.2 Sensor de Gás .....	25
3.3 Rede GSM.....	26
3.3.1 SMS ( <i>Short Message Service</i> ) .....	26

3.3.2 SIM Card .....	26
----------------------	----

<b>3.4 Visão Geral do Projeto .....</b>	<b>28</b>
---	-----------

## **CAPÍTULO 4 – DESCRIÇÃO DO HARDWARE E SOFTWARE ..... 30**

<b>4.1 Microcontrolador PIC 16F877A .....</b>	<b>30</b>
---	-----------

4.1.1 Especificações .....	31
----------------------------	----

4.1.2 Pinagem do PIC 16F877A .....	31
------------------------------------	----

4.1.3 Kit de Gravação PICkit 2 .....	34
--------------------------------------	----

<b>4.1.3.1 Utilização do Kit de Gravação PICKit 2.....</b>	<b>36</b>
--	-----------

<b>4.2 Sensor de Gás MQ-9 .....</b>	<b>37</b>
-------------------------------------	-----------

4.2.1 Especificações .....	38
----------------------------	----

4.2.2 Pinagem.....	39
--------------------	----

<b>4.3 Módulo GSM SIM900 .....</b>	<b>39</b>
------------------------------------	-----------

4.3.1 Especificações .....	40
----------------------------	----

4.3.2 Pinagem.....	42
--------------------	----

4.3.3 Comandos AT .....	44
-------------------------	----

<b>4.4 Comunicação Serial RS-232 .....</b>	<b>44</b>
--	-----------

4.4.1 Circuito Integrado MAX232.....	46
--------------------------------------	----

4.4.2 Pinagem do CI MAX232.....	46
---------------------------------	----

<b>4.5 PIC C Compiler .....</b>	<b>48</b>
---------------------------------	-----------

<b>4.6 Display LCD.....</b>	<b>48</b>
-----------------------------	-----------

## **CAPÍTULO 5 – IMPLEMENTAÇÃO ..... 50**

<b>5.1 Modelagem do Sistema .....</b>	<b>50</b>
---------------------------------------	-----------

5.1.1 – Fluxograma Geral do Sistema.....	52
--	----

<b>5.2 Elaboração do Circuito .....</b>	<b>52</b>
---	-----------

5.2.1 Proteus ISIS 7.7 Professional.....	53
--	----

5.2.2 Proteus ARES 7.7 Professional .....	53
<b>5.3 Elaboração do Código Fonte para o Microcontrolador PIC .....</b>	<b>54</b>
5.3.1 Escrita do Código Fonte .....	55
5.3.2 Compilação e Gravação do Código Fonte .....	59
<b>5.4 Montagem do Protótipo .....</b>	<b>59</b>
<b>CAPÍTULO 6 – RESULTADOS OBTIDOS.....</b>	<b>61</b>
6.1 Simulações .....	61
6.2 Problemas Encontrados .....	62
6.4 Orçamento do Projeto .....	62
6.5 Protótipo Final.....	65
<b>CAPÍTULO 7 – CONSIDERAÇÕES FINAIS .....</b>	<b>66</b>
7.1 Conclusões .....	66
7.2 Propostas para Trabalhos Futuros .....	66
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>68</b>
<b>APÊNDICE A – CÓDIGO FONTE DO DISPOSITIVO .....</b>	<b>71</b>
<b>APÊNDICE B – BIBLIOTECA DISPLAY LCD .....</b>	<b>80</b>
<b>APÊNDICE C – FUNÇÃO DO MODEM GSM .....</b>	<b>88</b>
<b>APÊNDICE D - MONTAGEM DO CIRCUITO NA PLACA .....</b>	<b>90</b>

## LISTA DE FIGURAS

Figura 1.1- Métodos de Detecção de Vazamento de Gás com os Sentidos Humanos .....	17
Figura 3.1– Dimensionamento das famílias PIC e dsPIC. ....	24
Figura 3.2 - Cartão SIM utilizado no projeto .....	27
Figura 3.3 – Diagrama em blocos do projeto .....	28
Figura 3.4 – Diagrama Esquemático do Projeto .....	29
Figura 4.1– PIC 16F877A utilizado no projeto .....	30
Figura 4.2 - Pinagem do PIC 16F877A .....	32
Figura 4.3– Pinagem do PIC 16F877A utilizada no projeto .....	34
Figura 4.4 – PICKit 2.....	35
Figura 4.5 – Forma de Encaixe do PICKit 2 .....	35
Figura 4.6 – Software PICKit 2 identificando o microcontrolador .....	36
Figura 4.7 – PICKit 2 Gravação Realizada com Sucesso.....	37
Figura 4.8 – Sensor de Gás MQ-9 .....	37
Figura 4.11 – Módulo GSM SIM900 .....	40
Figura 4.12 – Pinagem e Ligações Utilizadas .....	42
Figura 4.13 – Modificação no Modem .....	43
Figura 4.14 – Conector DB9 Fêmea com pinagem .....	45
Figura 4.15 – CI MAX232.....	46
Figura 4.16 - Pinagem MAX232 .....	47
Figura 4.17 – Pinagem do MAX232 utilizada no projeto.....	47
Figura 4.18 – Display LCD utilizado no projeto.....	49
Figura 5.1 – Protótipo Concluído.....	50
Figura 5.2 – Esboço do Projeto .....	51
Figura 5.3 – Simulação do projeto no software ISIS .....	53
Figura 5.4 – Placa projetada no software ARES.....	54
Figura 5.5 – Código Fonte Parte I .....	55
Figura 5.6 – Código Fonte Parte II .....	55
Figura 5.7 – Código Fonte Parte III .....	56
Figura 5.8 – Código Fonte Parte IV .....	57

<b>Figura 5.9 – Código Fonte do Modem GSM .....</b>	<b>58</b>
<b>Figura 5.10 – Tela do PIC C Compiler .....</b>	<b>59</b>
<b>Figura 6.1 – Componentes do Projeto .....</b>	<b>64</b>
<b>Figura 6.2 – Projeto Completo .....</b>	<b>65</b>

## LISTA DE TABELAS E QUADROS

Tabela 4.1 – Significado das nomenclaturas dos pinos do PIC16F877A.....	32
Tabela 4.2 –Pinagem módulo GSM SIM900 .....	42
Tabela 4.3 – Comando AT Utilizados .....	44
Tabela 4.4– Conector DB9 com direção e função dos pinos .....	45
Tabela 4.5– Níveis de tensão – lógica TTL e padrão RS-232 .....	46
Tabela 4.6 – Pinos do Display LCD e ligações no projeto .....	49
Quadro 6.1 – Custo do Projeto .....	63

## LISTA DE ABREVIATURAS E SIGLAS

<b><math>\Omega</math></b>	Ohms
<b>A/D</b>	Analógico / Digital
<b>AT</b>	Hayes AT Commands
<b>bps</b>	bit por segundo
<b>CDMA</b>	Code Division Multiple Access
<b>DB9</b>	Data Bus 9
<b>EEPROM</b>	Erasable Electronically Programmable Read Only Memory
<b><math>\mu\text{F}</math></b>	Micro-Farad
<b>GLP</b>	Gás Liquefeito de Petróleo
<b>GSM</b>	Global System for Mobile Communications
<b>I2C</b>	Inter-Integrated Circuit
<b>LCD</b>	Liquid Cristal Display
<b>LED</b>	Diodo Emissor de Luz
<b>PIC</b>	Programmable Interface Controller
<b>SIM</b>	Subscriber Identification Module
<b>SnO2</b>	Dióxido de Estanho
<b>SMS</b>	Short Message Service
<b>SPI</b>	Serial Peripheral Interface
<b>V</b>	Volts

## CAPÍTULO 1 – INTRODUÇÃO

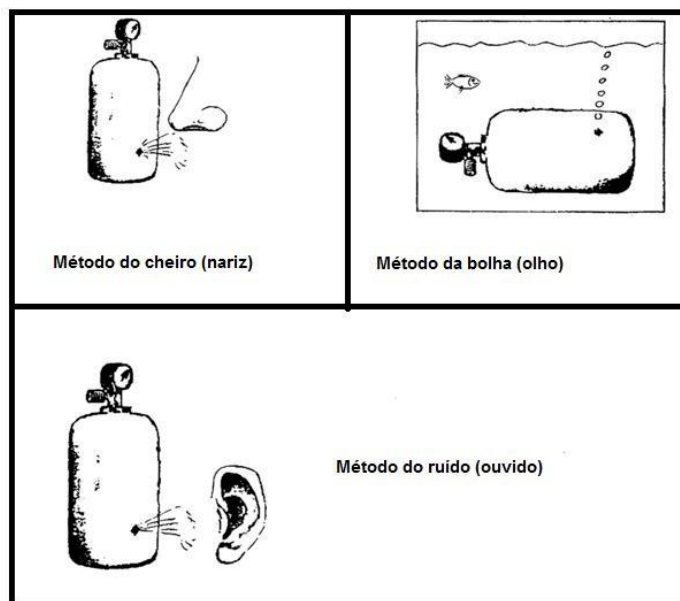
### 1.1 Introdução ao Tema Proposto

A história da localização de vazamento de gases é longa, 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 antiga 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.

Existem diversos métodos simples para a localização de um vazamento que são bastantes conhecidos. Na Figura 1.1 são apresentados alguns exemplos de inspeções realizadas com estes métodos sem o emprego de equipamentos e/ou instrumentos auxiliares, isto é, métodos feitos simplesmente utilizando-se alguns sentidos humanos. (IDEM).





**Figura 1.1- Métodos de Detecção de Vazamento de Gás com os Sentidos Humanos**  
(FONTE: Técnicas Preditivas Ligadas à Inspeção, 2012, modificada)

O desenvolvimento tecnológico, entretanto, fez com que estes métodos de detecção de vazamento de gás mencionados anteriormente 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).

Há uma crença popular de que quando se instala um sistema de detecção de gás o local se torna seguro. Esta afirmação é errada e perigosa. O sistema de detecção de vazamento de gás faz o papel de assistente e aumenta a segurança do ambiente. Uma residência não pode estar protegida por uma ação simples de instalar um detector de gás, a segurança residencial se inicia na fase de concepção, da engenharia e dos materiais de boa qualidade.

## 1.2 Motivação

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 do usuário, a utilização de um modem GSM para o envio de uma mensagem de texto avisando a irregularidade poderá evitar grandes explosões e salvar vidas.

Em 11 de junho de 1996, véspera do Dia dos Namorados, um vazamento nos dutos de gás no Osasco Plaza Shopping, na Grande São Paulo, provocou a explosão que deixou 42 mortos no local. O acidente aconteceu na hora do almoço, na praça de alimentação. O "Jornal Nacional" do dia reconstituiu o acidente, mostrou as cenas de destruição e a busca por feridos. (G1 Notícias, 2007)

Embora os casos de explosões sejam raros, o Corpo de Bombeiros recomendam algumas precauções que todos devem ter em caso de vazamento de gás. Este tipo de acidente é quase sempre causado pela falta de cuidado com o chamado "kit" de gás, composto pelos reguladores de pressão, mangueiras e abraçadeiras. Este kit deve ser trocado de 5 em 5 anos, sempre após o vencimento. Geralmente não é o botijão que explode e sim a instalação. (SP Notícias, 2011)

### **1.3 Objetivos**

A finalidade deste projeto é apresentar um dispositivo de detecção de vazamento de gás GLP com aviso por SMS, utilizando para isso equipamentos computacionais simples e com custo baixo.

- Apresentar um protótipo de um dispositivo capaz de identificar vazamento de gás inflamável.
- Integrar o microcontrolador com o módulo GSM.
- Emitir um aviso sonoro, exibir uma mensagem no display LCD e enviar uma mensagem de texto para um celular previamente cadastrado quando ocorrer vazamento de gás.

## 1.5 Estrutura da Monografia

Esta monografia é dividida em sete capítulos, incluindo a INTRODUÇÃO, que trata da introdução ao tema proposto, a motivação do projeto e os principais objetivos. É essa seção que descreve toda a estrutura da monografia.

No segundo capítulo, APRESENTAÇÃO DO PROBLEMA, é apresentada uma descrição aprofundada sobre as questões motivacionais do trabalho, as soluções existentes atualmente sobre o tema proposto. E a última seção deste capítulo aborda de forma resumida os benefícios do dispositivo proposto e suas restrições.

No terceiro capítulo, REFERENCIAL TEÓRICO, trata de assuntos como sensor, microcontroladores, GSM de forma conceitual. Nesse capítulo é apresentada também uma visão geral do projeto.

No quarto capítulo, DESCRIÇÃO DO HARDWARE E SOFTWARE, aborda as especificações dos dispositivos utilizados, é detalhada a especificação dos componentes de controle de comunicação serial, do microcontrolador PIC utilizado, detalhamento do sensor, bem como a configuração do módulo GSM, dentre outros dispositivos e *softwares*.

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

No sétimo capítulo, CONSIDERAÇÕES FINAIS, são apresentadas a conclusão e as sugestões para trabalhos futuros.

## CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA

Este capítulo tem o objetivo de detalhar as questões que motivaram o trabalho, como: evitar acidentes, que foi apresentado resumidamente na seção — Motivação — do capítulo anterior.

São apresentadas também algumas soluções existentes para estas questões, no que diz respeito à detecção de vazamento de gás. E por fim, são apresentados os benefícios do dispositivo proposto pelo trabalho e suas restrições.

### 2.1 Problemas de Saúde

A exposição ao gás através de vazamento pode ser prejudicial à saúde. O gás pode levar a explosões e representar riscos graves para a saúde que às vezes são até mesmo fatais. O vazamento pequeno acumula ao longo de um período de tempo e adiciona uma quantidade significativa de poluentes que forçam o sistema imunitário e outras funções corporais. De acordo com a *National Library of Medicine*, se um vazamento de gás ocorreu e é grave, o oxigênio pode ser reduzido, causando tonturas, cansaço, náuseas, dor de cabeça e respiração irregular. Vazamento de gás pode causar sérios danos às plantas, afetando assim o ecossistema ambiental também. (Online Directory, 2012).

O gás de cozinha é mais pesado do que o ar e, quando há vazamentos, vai se acumulando a partir do piso, expulsando o oxigênio e preenchendo todo o ambiente. Ele não é tóxico, mas tem efeito anestésico. Dependendo da quantidade e do local onde ocorrer o vazamento, pode levar à asfixia. O gás quando está sobre pressão no botijão e ocorre vazamento, há absorção do calor na vaporização do gás e o contato com o GLP nesta fase pode causar queimaduras.

### 2.2 Segurança Doméstica

Um dos mais comuns acidentes domésticos são os acidentes envolvendo vazamento de gás de cozinha e até explosões. Quando ocorre, a chama de um fósforo, a brasa de um cigarro, a faísca produzida pelo relé da geladeira ou pelo

interruptor de luz elétrica são suficientes para provocar uma explosão causando deslocamento de ar, que pode destruir paredes e arremessar objetos a grandes distancias.

### **2.3 Soluções Existentes**

Por ser um produto inodoro (sem cheiro), o GLP possui um composto à base de enxofre para caracterizar seu cheiro. Dessa forma, é possível detectar eventuais vazamentos. Para verificar se ocorreram vazamentos de gás, é recomendado passar uma esponja com água e sabão sobre a conexão da válvula/regulador de pressão de gás. Se houver vazamento, aparecerão bolhas no local.

É encontrado no mercado um Detector de Gás, que detecta a concentração de gás GLP. Ao detectar a concentração de gás emite um sinal sonoro e um sinal visual do próprio aparelho. Que pode ser instalado em cozinhas residenciais ou comerciais, depósitos de GLP e outros locais onde podem ocorrer eventuais vazamentos. (Minulight, 2012).

### **2.4 Benefícios do Dispositivo Proposto e Suas Restrições**

A função do projeto é apresentar um dispositivo de detecção de vazamento de gás GLP com aviso por SMS, integrando o sensor de gás com o microcontrolador e com o modem GSM.

O principal benefício do dispositivo será a possibilidade do cliente ser avisado de um vazamento de gás por um telefone e ter o *feedback*<sup>1</sup> do dispositivo, informando através de uma mensagem SMS se ocorreu um grande vazamento de gás no local onde o dispositivo foi instalado, e de certo modo ele será avisado mesmo não estando presente no local.

---

<sup>1</sup> *Feedback* é o procedimento que consiste no provimento de informação à uma pessoa sobre uma ação executada por esta, objetivando informar sobre as ações executadas anteriormente.

O projeto não tem como funcionalidade de tratar erros do sensor ou até armazenar os dados recebidos pelo mesmo. O protótipo somente fará a leitura do nível do gás e dependendo deste valor terá um comportamento.

Por questão de melhor demonstração do funcionamento do protótipo e por questões de segurança o vazamento do gás será feito através de um isqueiro que será posicionado próximo ao sensor.

## CAPÍTULO 3 – REFERENCIAL TEÓRICO

Neste capítulo é exibida uma visão de forma geral do projeto proposto. Por isso é essencial para o entendimento do mesmo a compreensão dos assuntos abordados neste capítulo.

Para o melhor desenvolvimento deste trabalho, é necessário o estudo de conceitos teóricos relacionados ao projeto desenvolvido. Embora alguns dos conceitos citados exijam um detalhamento mais amplo do tema para sua total compreensão, ter em vista manter o foco principal, apenas as características mais incluídas ao projeto serão apresentadas.

### 3.1 Microcontroladores

O microcontrolador é um dispositivo semicondutor em forma de circuito integrado, que integra as partes básicas de um microcomputador - microprocessador, memórias não-voláteis e voláteis e portas de entrada e saída. Geralmente, é limitado em termos de quantidade de memória, principalmente no que diz respeito à memória de dados, é utilizada em aplicações específicas, ou seja, naquelas que não necessitam armazenar grandes quantidades de dados. Apresenta um custo bastante baixo e possui vários fornecedores. (GIMENEZ, 2002, p.4).

Alguns dos fornecedores mencionados acima são: Hitachi, Intel, Microchip, Mitsubishi, Motorola, Philips, Toshiba etc.

Foi utilizado neste projeto o microcontrolador PIC16F877A, por ter uma abundância de pinos de entrada e saída (33 pinos dos 40 pinos do microcontrolador), por ter um fácil acesso no comércio local e principalmente porque os microcontroladores da série 16F têm conversor analógico/digital.

### 3.1.1 Microcontroladores da família PIC

O microcontrolador PIC produzido pela *Microchip Technology Inc.*, é um circuito integrado que em um único dispositivo possui todos os circuitos necessários para realizar um sistema programável completo.

Os microcontroladores da família PIC possuem modelos variados com características análogas e que podem ser operados por um parecido set de instruções. Embora apresentando características principais parecidas, eles se distinguem quanto a seus periféricos (memórias, comparadores, conversores) e quantidade de pinos. Dessa forma, há uma diversidade de microcontroladores e funcionalidades. (SOUZA, 2005)

A denominação família PIC é devido aos vários microcontroladores fornecidos, que variam de 8-bits, 16-bits até 32-bits, conforme mostrada na Figura 3.1.

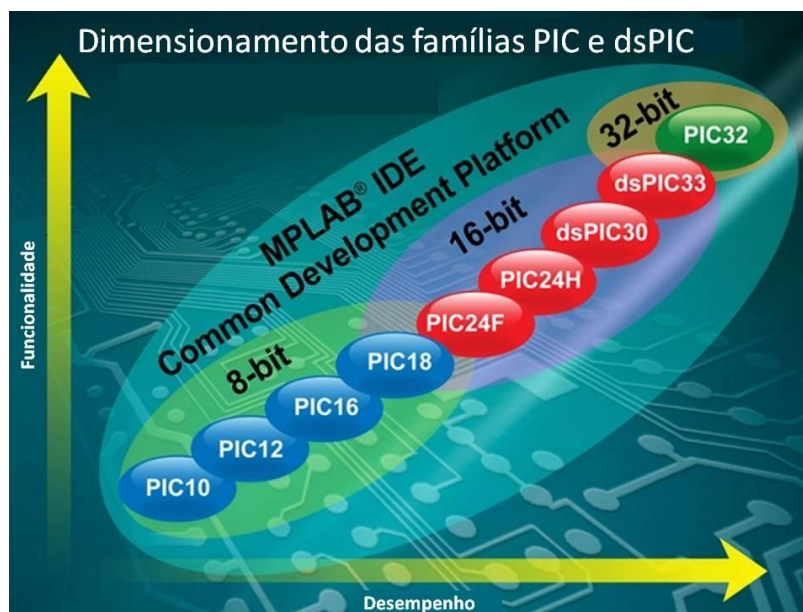


Figura 3.1– Dimensionamento das famílias PIC e dsPIC.

(Fonte: MICROCHIP TECHNOLOGY INC, adaptada).

As principais características técnicas do microcontrolador PIC18F877A, serão abordadas na seção – Microcontrolador PIC16F877A- no capítulo a seguir.



### 3.2 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. (IDEM).

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 homogênea. (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).

O sensor utilizado no projeto utiliza o dióxido de estanho na estrutura rutila ( $\text{SnO}_2$ ) que é um semicondutor que faz parte da classe dos óxidos condutores transparentes (TCO). Possui gap direto de 3,6 eV e condutividade do tipo n, mesmo quando não dopado intencionalmente. Mais especificações sobre o sensor são dadas no Capítulo 4 especificamente no 4.2 Sensor de Gás MQ-9.

### 3.3 Rede GSM<sup>1</sup>

A rede GSM foi criada com o objetivo de padronizar o sistema de telefonia móvel acabando com a incompatibilidade de redes. Que ocorreu devido ao crescimento desenfreado da produção dos celulares analógicos durante os anos 80 onde cada país tinha seu próprio sistema de envio de dados, protocolos e frequência de comunicação. Um grupo de estudos Europeu foi criado para desenvolver um sistema móvel padrão o *Group Special Mobile (GSM)*.

A rede GSM é uma tecnologia digital para celulares usada para transmissão de voz e serviços de dados móveis. Suporta chamadas de voz e dados como o envio de SMS, com velocidade de transferência de até 9,6 Kbps. (GSM WORD, 2012, traduzido).

#### 3.3.1 SMS (*Short Message Service*)

O Serviço de Mensagens Curtas (*Short Message Service*) permite aos usuários enviarem e receberem mensagens a partir de um aparelho de telefone celular. Cada mensagem pode ser de até 160 caracteres e ser enviada e recebida por usuários de diferentes redes de operadoras. (CAMPOS, 2010).

Este serviço foi criado para funcionar com a tecnologia digital GSM, seu principal objetivo era um sistema simples de mensagens que funcionasse mesmo quando os aparelhos móveis dos usuários estivessem fora da área de cobertura ou desligados.

#### 3.3.2 SIM<sup>2</sup> Card

O SIM Card é um chip de memória portátil usado principalmente em telefones celulares que operam em redes GSM. Estes cartões SIM possuem memória

---

<sup>1</sup> GSM é a sigla em inglês de *Global System for Mobile Communications* (Sistema Global para Comunicações Móveis), antiga *Group Special Mobile*.

<sup>1</sup> SIM é a sigla de *Subscriber Identity Module* (Módulo de Identificação do Assinante).

do tipo EEPROM<sup>1</sup>, onde são guardadas informações pessoais do titular da conta de telefone, incluindo o número de telefone, contatos, mensagens de texto e outros dados. Quando um usuário quer mudar de aparelho, ele pode facilmente remover o chip de um aparelho e inseri-lo em outro.

Os cartões SIM são uma parte fundamental do desenvolvimento da tecnologia dos celulares, pois antigamente somente existia a rede CDMA onde o aparelho que guardava as informações do usuário e para realizar a troca de aparelho, o usuário tinha que ir à loja da operadora para transferir o número para outro aparelho. (Wise Geek, 2012, traduzido e modificado).

A Figura 3.2 ilustra o cartão SIM utilizado no projeto. Cartão SIM da operadora TIM (escolhido aleatoriamente), com 128 KB de memória e 25 mm por 15 mm de tamanho.



Figura 3.2 - Cartão SIM utilizado no projeto (FONTE: Autor)

---

<sup>1</sup>EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) – é um *chip* de armazenamento não-volátil, podendo ser programada e apagada várias vezes, eletricamente.

### 3.4 Visão Geral do Projeto

O projeto basicamente é constituído por módulo GSM (SIM900), microcontrolador (PIC do modelo 16F877A), conversor de sinais de porta serial (circuito Integrado MAX232), sensor de gás (MQ-9), regulador de tensão (7805) 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 3.3.

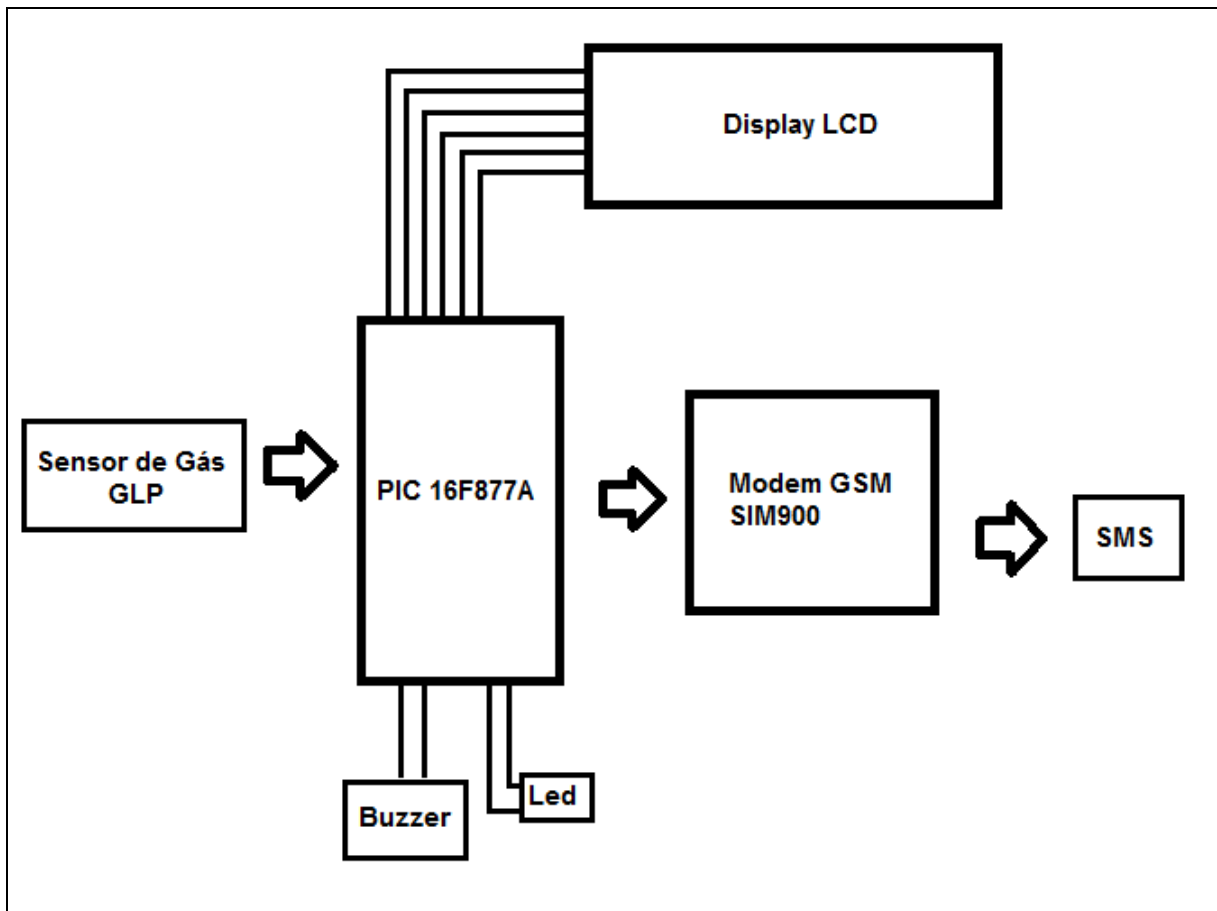
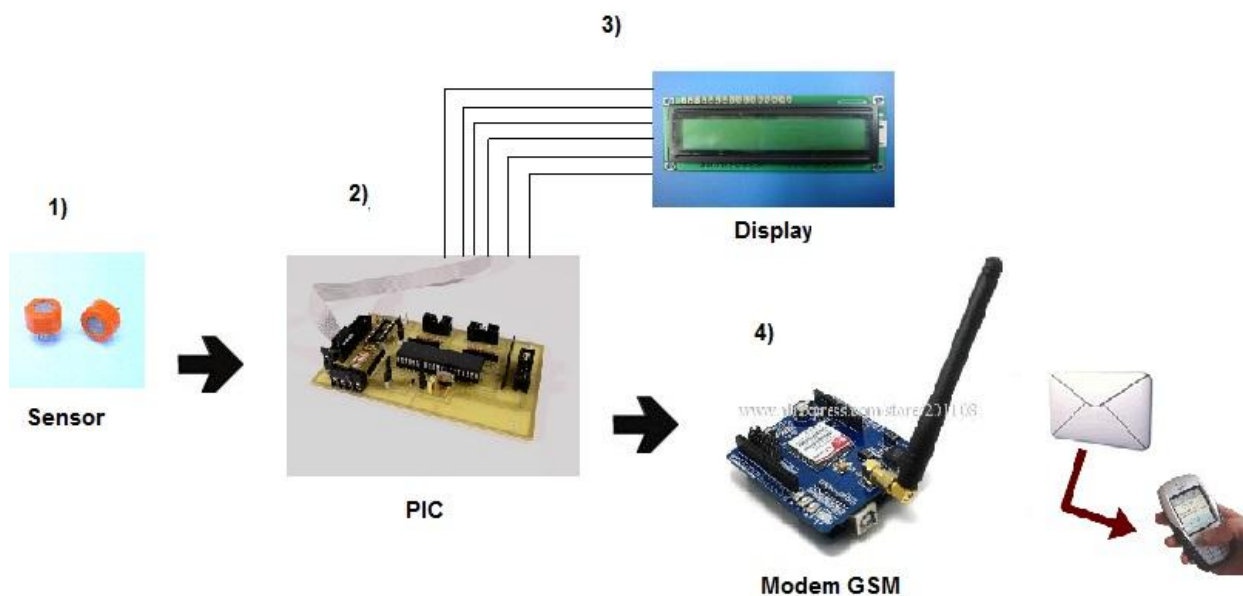


Figura 3.3 – Diagrama em blocos do projeto (FONTE: Autor)

O diagrama esquemático do projeto proposto pode ser observado na Figura 3.4, na qual representa de forma objetiva a composição geral do projeto.



**Figura 3.4 – Diagrama Esquemático do Projeto (FONTE: Autor)**

- 1) O sensor de gás manda o valor da leitura para o microcontrolador PIC;
- 2) O PIC recebe o sinal, caso for irregular aciona o modem e informa no Display;
- 3) O Display exibe a mensagem;
- 4) Modem envia o SMS

Detalhes sobre o *hardware* e o *software* utilizados e seus funcionamentos encontram-se no **CAPÍTULO 4 – DESCRIÇÃO DE HARDWARE E SOFTWARE**. A implementação do projeto pode ser observada no **CAPÍTULO 5 – IMPLEMENTAÇÃO**.

## CAPÍTULO 4 – 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.

### 4.1 Microcontrolador PIC 16F877A

O microcontrolador utilizado nesse projeto foi o PIC 16F877A, conforme mencionado no capítulo anterior. Ele é o componente responsável em fazer a comunicação com o circuito, onde estão ligados os outros componentes do projeto.

Por ser um modelo com 40 pinos, possibilita a montagem de um *hardware* complexo e capaz de interagir com diversos recursos ao mesmo tempo. (SOUZA, 2005, p.18).

A Figura 4.1 ilustra o PIC 16F877A utilizado neste projeto.

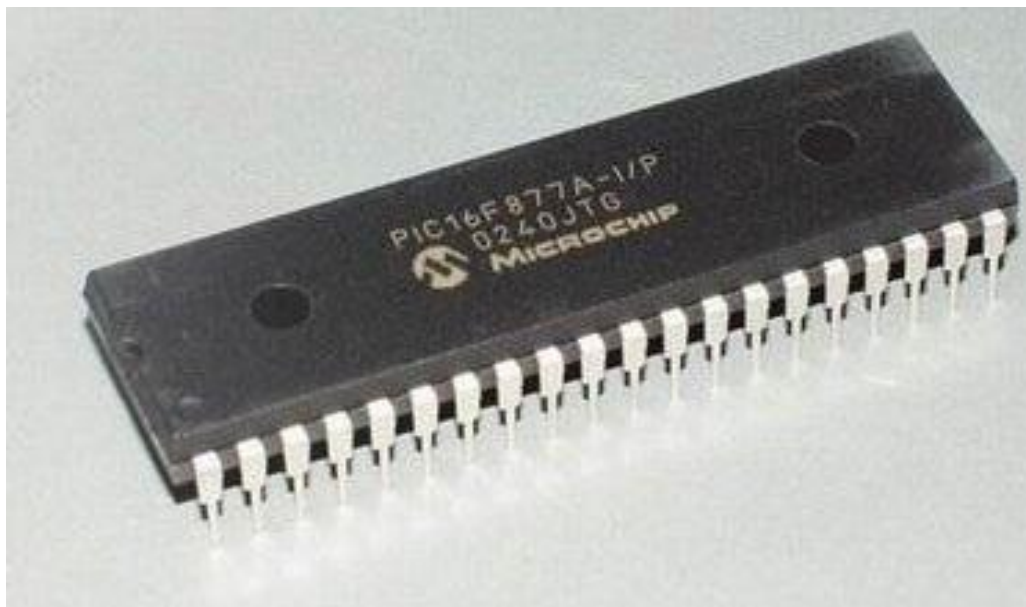


Figura 4.1– PIC 16F877A utilizado no projeto (Fonte:Multilogica)

### 4.1.1 Especificações

As principais especificações técnicas do PIC 16F877A são: (SOUZA, 2005).

- Microcontrolador de 40 pinos;
- Via de programação com 14 bits e 35 instruções;
- Programação *in-circuit* (alta e baixa tensão);
- Frequência de operação de até 20 MHz;
- Memória *flash* de programa de 8kwords;
- 33 portas configuráveis como entrada ou saída;
- 15 interrupções disponíveis;
- Memória EEPROM interna com 256 bytes;
- Memória RAM com 368 bytes;
- Conversores A/D (analógico-digital) de 10 bits (8x) e comparadores analógicos (2x);
- Comunicações seriais: SPI, I2C e USART.

Este modelo da família PIC foi escolhido devido ao fato de possuir 33 portas configuráveis de entrada e saída, comunicação serial e conversor A/D, conforme estas características a elaboração deste dispositivo foi possível.

### 4.1.2 Pinagem do PIC 16F877A

A pinagem do PIC 16F877A é mostrada na Figura 4.2 e o significado das nomenclaturas utilizadas na identificação desses pinos na Tabela 4.1.

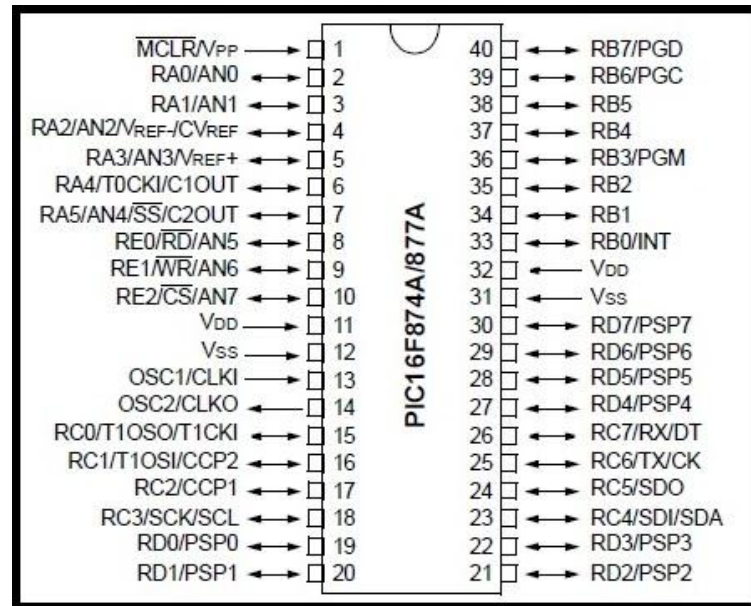


Figura 4.2 - Pinagem do PIC 16F877A

(Fonte: Microchip Technology Datasheet, 2003, p.3)

Tabela 4.1 – Significado das nomenclaturas dos pinos do PIC16F877A.

N°	Pino	Descrição
1	MCLR/Vpp	Master Clear (reset). O microcontrolador funciona quando este pino está em nível alto.
2	RA0 AN0	Entrada e saída digital. Entrada analógica.
3	RA1 AN1	Entrada e saída digital. Entrada analógica.
4	RA2 AN2 V <sub>REF-</sub> /CV <sub>REF</sub>	Entrada e saída digital. Entrada analógica. Tensão negativa de referência analógica.
5	RA3 AN3 V <sub>REF+</sub>	Entrada e saída digital. Entrada analógica. Tensão positiva de referência analógica.
6	RA4 T0CKI C1OUT	Entrada e saída digital. Open-drain quando configurado como saída. Entrada externa do contador TMR0. Saída do comparador 1.
7	RA5 AN4 SS C2OUT	Entrada e saída digital. Entrada analógica. Slave para a comunicação SPI. Saída do comparador 2.
8	RE0 RD AN5	Entrada e saída digital. Controle de leitura da comunicação paralela. Entrada analógica.
9	RE1 WR AN6	Entrada e saída digital Controle de escrita da comunicação paralela. Entrada analógica.
10	RE2 CS AN7	Entrada e saída digital. Habilitação externa para comunicação paralela. Entrada analógica.
11/32	VDD	Alimentação positiva.



12/31	VSS	GND.
13	OSC1/CLKIN	Oscilador cristal ou entrada de osciladores externos.
14	OSC2/CLKOUT	Saída para oscilador cristal.
15	RC0 T1OSO T1CKI	Entrada e saída digital. Saída do oscilador externo para TMR1. Entrada de incremento para TMR1.
16	RC1 T1OSI CCP2	Entrada e saída digital. Entrada do oscilador externo para TMR1. Entrada do Capture2 ou Saída para Compare2/PWM2.
17	RC2 CCP1	Entrada e saída digital. Entrada do Capture1 ou Saída para Compare1/PWM1.
18	RC3 SCK SCL	Entrada e saída digital. Entrada/Saída do clock para comunicação SPI. Entrada/Saída do clock para comunicação I2C.
19	RD0 PSP0	Entrada e saída digital. Comunicação paralela.
20	RD1 PSP1	Entrada e saída digital. Comunicação paralela.
21	RD2 PSP2	Entrada e saída digital. Comunicação paralela.
22	RD3 PSP3	Entrada e saída digital. Comunicação paralela.
23	RC4 SDI DAS	Entrada e saída digital. Entrada de dados para comunicação SPI. Entrada/Saída de dados para comunicação I2C.
24	RC5 SDO	Entrada e saída digital. Saída de dados para comunicação SPI.
25	RC6 TX CK	Entrada e saída digital. Transmissão para comunicação assíncrona USART. Clock para comunicação síncrona USART.
26	RC7 RX DT	Entrada e saída digital. Recepção para comunicação assíncrona USART. Dados para comunicação síncrona USART.
27	RD4 PSP4	Entrada e saída digital. Comunicação paralela.
28	RD5 PSP5	Entrada e saída digital. Comunicação paralela.
29	RD6 PSP6	Entrada e saída digital. Comunicação paralela.
30	RD7 PSP7	Entrada e saída digital. Comunicação paralela.
33	RB0 INT	Entrada e saída digital. Interrupção externa.
34	RB1	Entrada e saída digital.
35	RB2	Entrada e saída digital.
36	RB3 PGM	Entrada e saída digital. Entrada para programação de baixa tensão.
37	RB4	Entrada e saída digital.
38	RB5	Entrada e saída digital.
39	RB6 PGC	Entrada e saída digital. Clock de programação serial ou pino de in-circuit debugger.
40	RB7 PGD	Entrada e saída digital. Dado de programação serial ou pino de in-circuit debugger.

A pinagem escolhida no projeto pode ser observada na Figura 4.3. Essa configuração atende às necessidades de cada componente utilizado no projeto

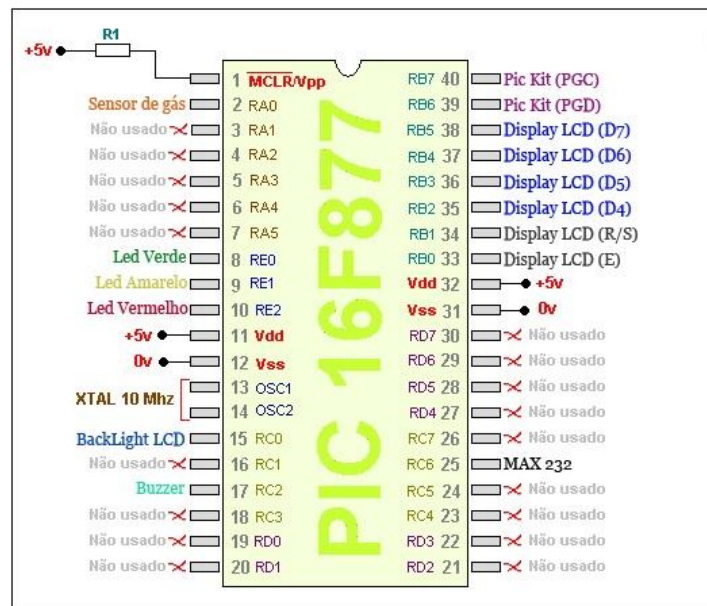


Figura 4.3– Pinagem do PIC 16F877A utilizada no projeto.

(Fonte: Rogercom, modificada)

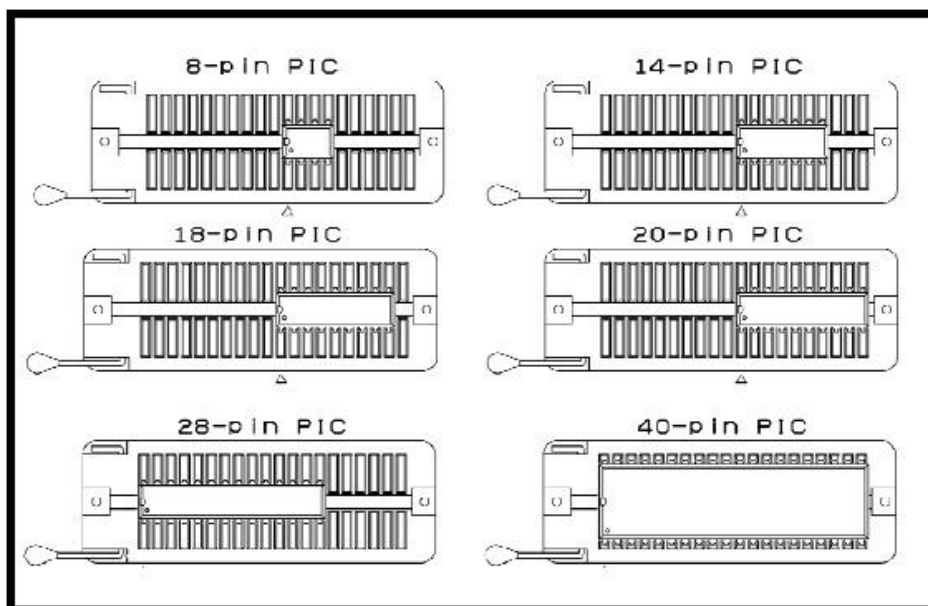
### 4.1.3 Kit de Gravação PICkit 2

O kit de gravação PICkit 2 é uma ferramenta de desenvolvimento que permite comunicar o computador diretamente com o microcontrolador PIC. É composto do conector para gravação em microcontroladores PIC de 8, 14, 18, 20, 28 e 40 pinos, conforme é mostrado na Figura 4.4.



**Figura 4.4 – PICKit 2 (Fonte: Autor)**

Na Figura 4.5 é mostrada como os microcontroladores devem ser inseridos no Soquete ZIF de 40 pinos de acordo com sua quantidade de pinos.



**Figura 4.5 – Forma de Encaixe do PICKit 2 (Fonte: Manual PICKit2)**

Além do gravador e o cabo USB, o PICKit 2 possui também um CD com o *software (PICKit 2 Programmer)* que identifica o microcontrolador a ser gravado e pode ser utilizado em qualquer computador com uma porta USB 1.1 ou 2.0. A Figura 4.6 ilustra a tela do *software* identificando a conexão com o microcontrolador.

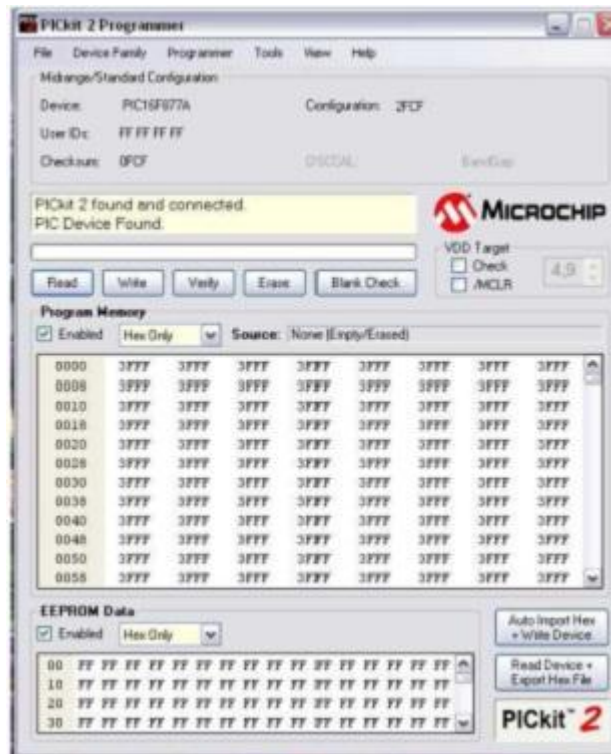


Figura 4.6 – Software PICKit 2 identificando o microcontrolador (Fonte: Autor)

#### 4.1.3.1 Utilização do Kit de Gravação PICKit 2

Após criar o arquivo com o PIC C Compiler, de extensão “.hex”, o passo seguinte foi abrir o *software* onde foi mostrado em “Device” o dispositivo utilizado (Conforme Figura 4.6, PIC16F887A), clicando no botão “File” e em seguida em “Import Hex” uma janela se abrirá onde deverá ser localizado o arquivo de extensão “.hex”, com isso o *software* está pronto para realizar a gravação no microcontrolador. Clicando em “Write” o *software* PICKit 2 iniciará a gravação e conforme é mostrado na Figura 4.7 onde o status “Programming Sucessfull” (gravação realizada com sucesso) é apresentado.

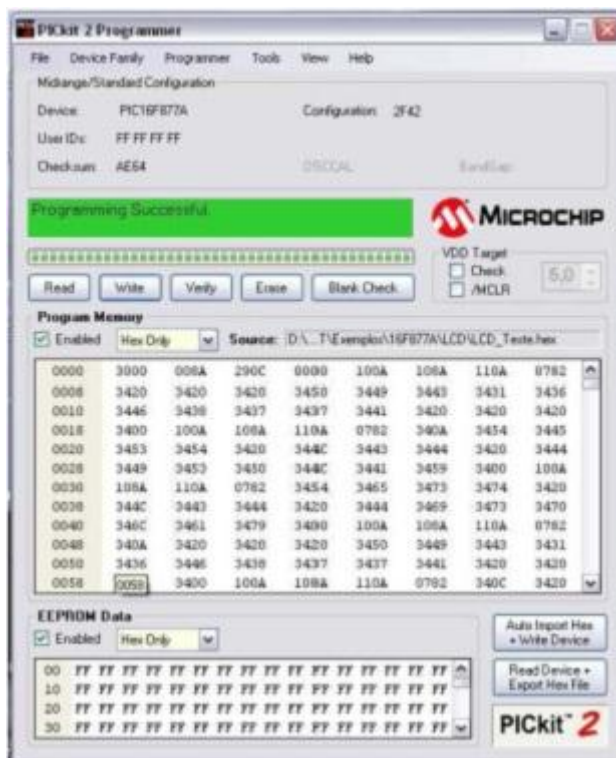


Figura 4.7 – PICKit 2 Gravação Realizada com Sucesso (Fonte: Autor)

## 4.2 Sensor de Gás MQ-9

O MQ-9 mostrado na Figura 4.6 tem um sensor de gás metano ( $\text{CH}_4$ ) com alta sensibilidade ao monóxido de carbono (CO) e GLP. O sensor é de baixo custo e adequado para diferentes aplicações.



Figura 4.8 – Sensor de Gás MQ-9 (Fonte: MQ-9 Datasheet)

### 4.2.1 Especificações

O material sensível do sensor de gás MQ-9 é o  $\text{SnO}_2$ , que possui menor condutividade em ar limpo. A detecção é feita pelo método da temperatura no ciclo alto e baixo, detecta CO quando a temperatura é baixa (aquecido por 1,5 V). A condutividade do sensor é mais elevada juntamente com a concentração de gás ascendente. Quando a temperatura é elevada (aquecido por 5.0V), detecta metano ( $\text{CH}_4$ ), propano ( $\text{C}_3\text{H}_8$ ), GLP, etc. (MQ-9 Datasheet, 2012, traduzido e adaptado).

O MQ-9 utiliza um circuito elétrico simples. Converte a mudança de condutividade para corresponder ao sinal de saída de concentração de gás.

Características:

- 1 - Alta sensibilidade ao CO e gás combustível;
- 2 - Alta sensibilidade ao Propano ( $\text{C}_3\text{H}_8$ ), metano ( $\text{CH}_4$ ) e CO;
- 3 - Resposta rápida;
- 4 - Ampla faixa de detecção;
- 5 - Desempenho e estabilidade, longa vida, baixo custo;
- 6 - Simples circuito de acionamento.

A descrição acima é circuito de teste básico do sensor conforme mostrado na Figura 4.9. 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. A pinagem utilizada é ilustrada na Figura 4.10.

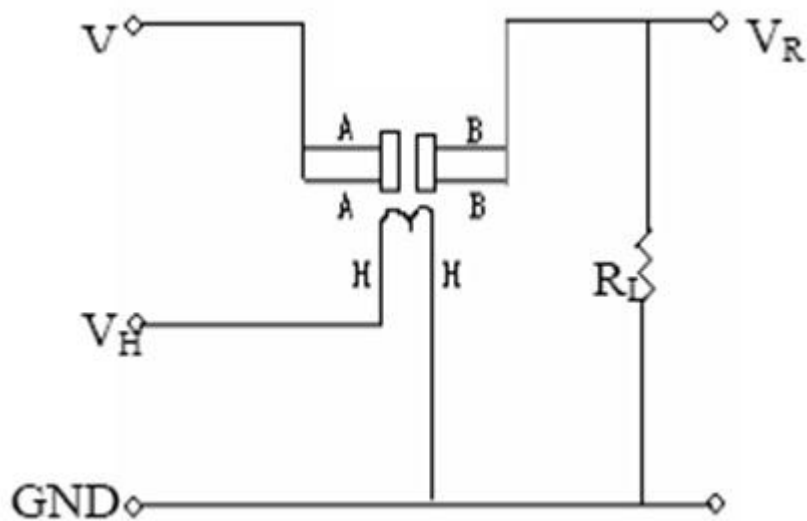


Figura 4.9 - Circuito de teste básico. (Fonte: MQ-9 Data sheet)

## 4.2.2 Pinagem

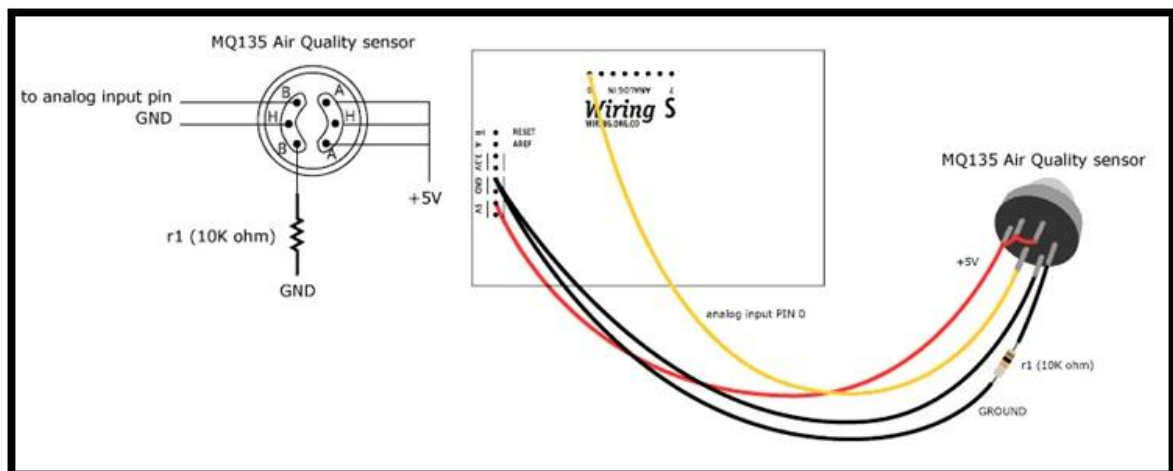


Figura 4.10 – Pinagem utilizada (Fonte: MQ-9 Data Sheet)

## 4.3 Módulo GSM SIM900

O IComSat é um shield GSM/GPRS baseado no módulo Quad-band GSM/GPRS SIM900. Ele é controlado via comando AT (GSM 07.07, 07.05 e coman-

do AT melhorado SIMCOM<sup>1</sup>), é um módulo totalmente compatível com vários microcontroladores, porém foi produzido para uso fácil com Arduino. (SIM 900 Data Sheet, IComSat SIM 900). A Figura 4.11 ilustra o modem GSM SIM900.



Figura 4.11 – Módulo GSM SIM900 (Fonte: SIM 900 Data Sheet, IComSat SIM 900)

### 4.3.1 Especificações

O SIM900 é um módulo GSM/GPRS quadband com pilha TCP/IP e pode ser usado em aplicações onde a transmissão via tecnologia GSM/GPRS se faça necessária; seja, transmissão de voz, SMS - mesmo assim, consome pouca energia. Seu reduzido tamanho o torna ideal para os mais exigentes requisitos das aplicações industriais, como M2M<sup>2</sup>, telemetria ou qualquer outra forma de comunicação móvel. O módulo é fabricado pela SIMCOM e distribuído no Brasil pela ME Compo-

---

<sup>1</sup> SIMCOM *Wireless Solutions* – é líder na indústria fornecendo módulos sem fio de alta qualidade para diferentes plataformas de tecnologias em redes. (Página na Web: <http://wm.sim.com>).

<sup>2</sup> Máquina a máquina (M2M) – é a utilização um dispositivo para capturar um evento que é retransmitido para um aplicativo que traduz o evento capturado em informações significativas.



nentes<sup>1</sup>. As principais características do módulo SIM900 são: (SIM 900 Data Sheet, IComSat SIM 900).

- Quad-Band 850/900/1800/1900MHz;
- GPRS classe multi-slot 10/8;
- GPRS estação móvel classe B;
- Compatível com o GSM fase 2/2+;
- Classe 4 (2W@850/900MHz);
- Classe 1 (1W@1800/1900MHz);
- Controle via comandos (GSM 07.07, 07.05 e SIMCOM Comandos AT melhorado);
- SMS(Short Message Service);
- Porta serial livre para seleção;
- Todos os módulos SIM900 conexões para pinos de saídas;
- RTC<sup>2</sup> suportado com Super Capacitor;
- Power on/off e reset de funções suportadas pela interface Arduino;
- Indicadores de PWR, LED de estado e LED de estado de rede;
- Tamanho da PCB: 77,2mm X 66 mm X 1,6mm;
- Alimentação Externa: 9 a 20 volts DC(Compatível com Arduino);
- Protocolo de comunicação: UART<sup>3</sup>;

---

<sup>1</sup> ME Componentes e Equipamentos – é uma empresa fundada em 1997, que trabalha com todos os tipos e com as melhores marcas de componentes eletrônicos, tais como módulos GPS, módulos GSM/GPRS, circuitos integrados. (Página na Web: <http://www.mecomp.com.br>).

<sup>2</sup> RTC (*Real-time clock*) – Relógio de Tempo Real é um circuito integrado que mantém o controle do tempo presente, utilizado para evitar confusão com um outro “clock”.

<sup>3</sup> UART (*Universal Asynchronous Receiver/Transmitter*) – tem a finalidade de converter dados paralelos para a forma serial e vice versa.

- RoHS: Sim.

### 4.3.2 Pinagem

Na Figura 4.12 é mostrada a pinagem do modem GSM/GPRS SIM900 e a troca de “jumpers” para realizar a configuração do modem. A Tabela 4.2 descreve todos os pinos do modem.

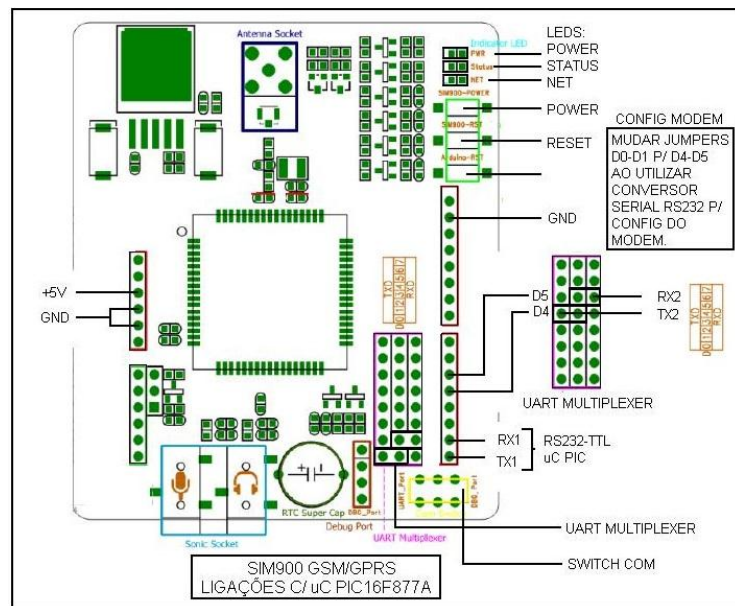


Figura 4.12 – Pinagem e Ligações Utilizadas (Fonte: SIM 900 Data Sheet, adaptado)

Tabela 4.2 – Pinagem módulo GSM SIM900

Interface	Pino	Descrição
GPIOs	1	VDD*
	2	GND
	3	GPIO1
	4	GPIO2
	5	GPIO3
	6	GPIO4
	7	GPIO5
	8	GPIO6
	9	GPIO7
	10	GPIO8
	11	GPIO9
	12	GPIO10
	13	GPIO11
	14	GPIO12
	15	GND

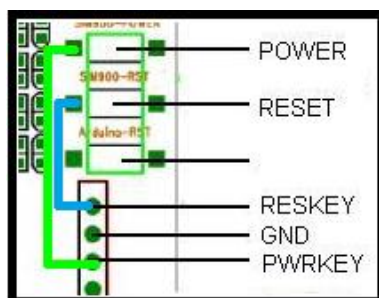
	16	VDD*
UART	1	GND
	2	DTR
	3	RI
	4	DCD
	5	CTS
	6	RTS
	7	TXD
	8	RXD
IIC&PWM	1	GND
	2	GND
	3	IIC_SCL
	4	IIC_SDA
	5	PWM2
	6	PWM1
	7	GND
	8	GND
Debug_Port	1	GND
	2	PERKEY
	3	DBG_RXD
	4	DBG_TXD

VDD\* = 3.0 V

Nota: O nível de operação da porta? É de 3.0V.

**FONTE: ME COMPONENTES**

Foram feitas algumas modificações no modem GSM SIM900, para que ele possa ser iniciado e reiniciado com comandos vindos diretamente do microcontrolador. Na Figura 4.13 é mostrada estas modificações, onde estão os dois fios soldados nos pinos.



**Figura 4.13 – Modificação no Modem (Fonte: SIM 900 Data Sheet, adaptado)**

### 4.3.3 Comandos AT

O comando AT é um conjunto de linhas de comando que se comunica com o módulo GSM e é utilizado para configurar este módulo. Um dos serviços do módulo GSM que foram configurados neste projeto é o envio de SMS através da rede de celular com tecnologia GSM. Os comandos AT utilizados para a configuração do modem GSM SIM 900 e suas descrições são apresentados na Tabela 4.3.

**Tabela 4.3 – Comando AT Utilizados**

Comando	Descrição
AT <ENTER>	Verifica se o modem está conectado corretamente. Receberá a resposta OK na tela.
AT+IPR=<RATE>	Define a taxa de transferência de dados. No lugar do <RATE> é colocado a taxa de bauds por segundo.
AT+CPIN="2727" <ENTER>	Libera o uso do SIM CARD no modem. Usado somente caso o chip possua PIN, O número "2727" deverá ser substituído pelo PIN referente ao SIM CARD.
AT+CMGF=1 <ENTER>	Estabelece a condição de enviar mensagens de texto simples SMS. Receberá resposta OK.
AT+CSCS="+55XXXXXXXXXX"	Cadastra o número de telefone celular em que se deseja enviar a mensagem de texto. O número "+55XXXXXXXXXX" deverá ser substituído.
TESTE DO MODEM <CTRL-Z>	Deve-se digitar o texto da mensagem que deseja enviar e em seguida digitar o comando <CTRL-Z> para finalizar e concluir o envio.

FONTE: SIMCOM, 2010b, p. 6.

## 4.4 Comunicação Serial RS-232

A comunicação serial do tipo RS232 é muito empregada para permitir a comunicação entre dispositivos. Dispositivos como modem são apenas uns dos diversos equipamentos que podem ser conectados à porta serial, que realiza a comunicação entre um equipamento de dados terminal e um equipamento de comunicação de dados. Foi utilizado o conector DB9 neste projeto conforme mostrado na Figura 4.14 e a pinagem é apresentada pela Tabela 4.4. (CANZIAN, 2012; ZANCO, 2010, p.262).

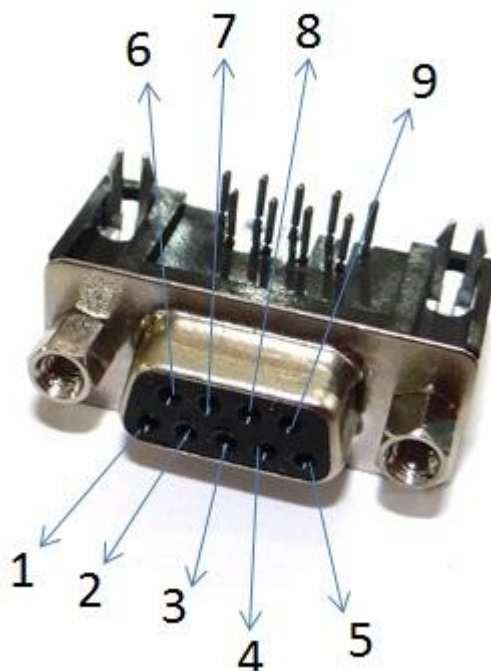


Figura 4.14 – Conector DB9 Fêmea com pinagem (Fonte: Autor)

Tabela 4.4– Conector DB9 com direção e função dos pinos.

Número	Nome	Direção	Função
1	DCD – Data Carrier Detect	Entrada	Detectar Portadora
2	RXD – Receive Data	Entrada	Recepção de Dados
3	TXD – Transmit Data	Saída	Transmissão de Dados
4	DTR – Data Terminal Ready	Saída	Terminal de Dados Prontos
5	GND – Signal Ground	-	Terra
6	DSR – Data Set Ready	Entrada	Conjunto de Dados Prontos
7	RTS – Request to Send	Saída	Pronto para Enviar (computador)
8	CTS – Clear to Send	Entrada	Envie os Dados (modem)
9	RI – Ring Indicator	Entrada	Indicador de Campanha Elétrica

FONTE: ZANCO, 2010, p. 263 (alterada).

Foi utilizado o Circuito Impresso MAX232 para fazer a comunicação entre o PIC 16F877A e o módulo GSM SIM900, pois é um conversor de nível TTL<sup>1</sup> para o padrão RS-232. Somente três pinos do conector DB9 foram utilizados: O pino 2 para receber os dados do modem (RXD), o pino 3 para transmitir os dados para o modem (TXD) e o pino 5 que é o terra (GND).

<sup>1</sup> TTL (*Transistor-Transistor Logic*) – é um padrão elétrico que foi estabelecido para permitir que componentes de diferentes fabricantes possam se comunicar sem a preocupação de que o nível lógico zero ou um varie entre os componentes.

#### 4.4.1 Circuito Integrado MAX232

O CI MAX232 é um circuito integrado que converte níveis de tensão correspondentes à lógica TTL no padrão RS-232 e vice versa. Foi utilizado devido ao microcontrolador PIC16F877A e o módulo GSM SIM900 terem níveis de tensão diferentes. (ZANCO, 2010, p.267).

A Figura 4.15 ilustra o CI MAX232 utilizado neste projeto e a Tabela 4.5 apresenta os níveis de tensão correspondentes aos níveis lógicos zero (0) e um (1) do padrão RS-232 e do PIC16F877A.



Figura 4.15 – CI MAX232 (Fonte: Autor)

Tabela 4.5– Níveis de tensão – lógica TTL e padrão RS-232

Nível Lógico	PIC16F877A	Padrão RS-232
0	$\leq 0,6$	+5V a +15V
1	$\geq 5V - 0,6V$	-5V a -15V

FONTE: ZANCO, 2010, p. 267.

#### 4.4.2 Pinagem do CI MAX232

Na Figura 4.16 é mostrado a o CI MAX232 com sua pinagem, nele foram ligados cinco capacitores de 10 $\mu$ F conectados externamente a alguns dos seus pinos.

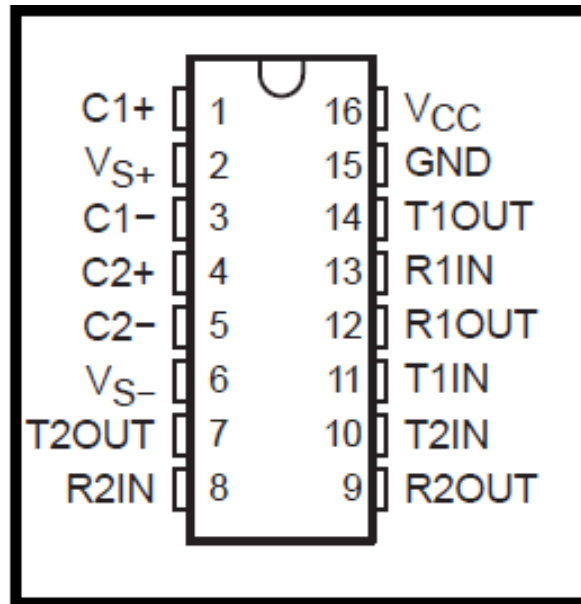


Figura 4.16 - Pinagem MAX232 (Fonte: Datasheet MAX232)

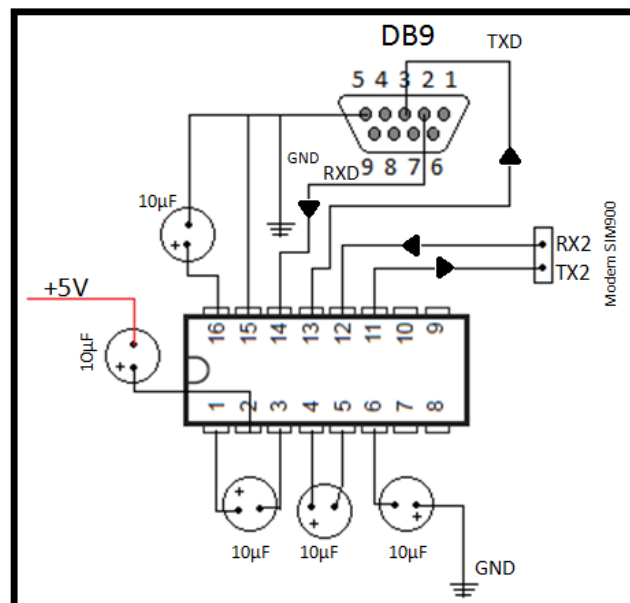


Figura 4.17 – Pinagem do MAX232 utilizada no projeto (Fonte: Autor)

A pinagem optada para o circuito integrado MAX232 no projeto, como mostrada na Figura 4.17 foi:

- Pino 1 e 3: capacitor de 10µF;
- Pino 2 e Fonte de alimentação: capacitor de 10µF;
- Pino 4 e 5: capacitor de 10µF;
- Pino 6 e Terra: capacitor de 10µF;

- Pinos 7, 8, 9 e 10: não são utilizados;
- Pino 11: ligação com modem GSM SIM900 no pino TX2;
- Pino 12: ligação com modem GSM SIM900 no pino RX2;
- Pino 13: ligação com a porta serial (DB9) no pino 3 (TXD);
- Pino 14: ligação com a porta serial (DB9) no pino 2 (RXD);
- Pino 15: ligação com o GND (terra);
- Pino 16: Fonte de alimentação e capacitor de 10 $\mu$ F aterrado.

#### 4.5 PIC C Compiler

Foi utilizado para a elaboração da programação do projeto 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.

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.

#### 4.6 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 4.18 ilustra o módulo utilizado.





Figura 4.18 – Display LCD utilizado no projeto (Fonte: Autor)

A Tabela 4.6 apresenta a descrição dos pinos do módulo LCD e o interfacçamento no projeto.

Tabela 4.6 – Pinos do Display LCD e ligações no projeto

Pino	Símbolo	Função	Interfacçamento no projeto
16	K	Cátodo Backlight	Aterrado
15	A	Anodo Backlight	Pino 15 do PIC16F877A
1	VSS	Terra	Aterrado
2	VDD	+5 V	Fonte de Alimentação
3	V <sub>0</sub>	Ajuste de contraste	Resistor de 4k7
4	R/S	Seleção de registro	Pino 34 do PIC16F877A
5	R/W	Leitura / Escrita	Aterrado
6	E	Inicia ciclo Leitura / Escrita	Pino 33 do PIC16F877A
7	DB0	Dado	NÃO UTILIZADO
8	DB1	Dado	NÃO UTILIZADO
9	DB2	Dado	NÃO UTILIZADO
10	DB3	Dado	NÃO UTILIZADO
11	DB4	Dado	Pino 35 do PIC16F877A
12	DB5	Dado	Pino 36 do PIC16F877A
13	DB6	Dado	Pino 37 do PIC16F877A
14	DB7	Dado	Pino 38 do PIC16F877A

FONTE: Datasheet 1602h.

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.

## CAPÍTULO 5 – 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;
- Elaboração do código fonte para o microcontrolador PIC;
- Montagem dos circuitos nas placas;
- Montagem do protótipo;

Na Figura 5.1 é mostrado o protótipo em sua fase final.



Figura 2.1 – Protótipo Concluído (Fonte: Autor)

### 5.1 Modelagem do Sistema

Primeiramente foi definido o que o projeto deveria ter de forma geral e quais os componentes necessários para fazer a integração de algumas de suas partes. A Figura 5.2 ilustra o esquema inicial que deu origem ao modelo para a simulação no *software* ISIS.

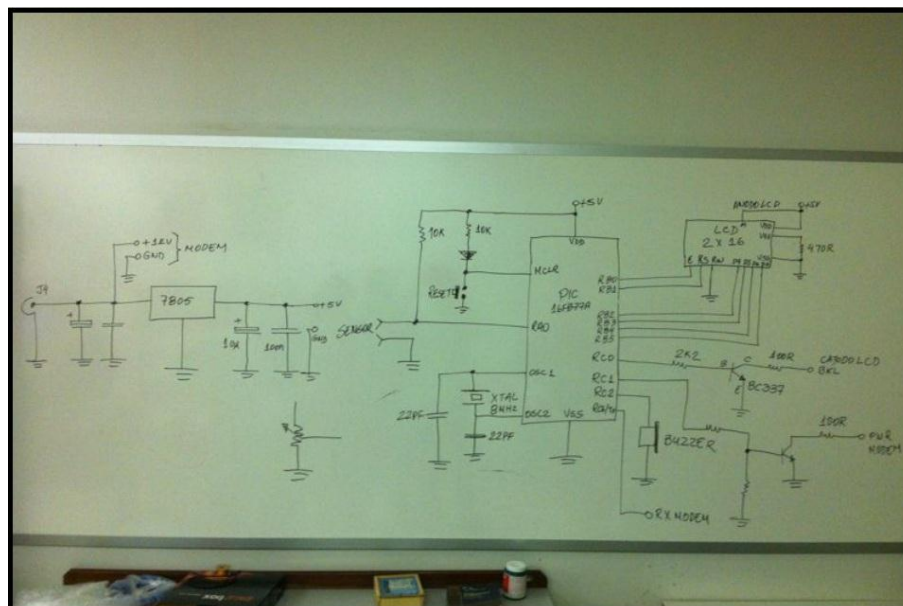


Figura 5.2 – Esboço do Projeto (Fonte: Autor)

Na Figura 5.2 todas as ligações com o PIC16F877A estão esquematizadas para seu devido funcionamento com os componentes determinados no projeto.

O projeto é formado por uma placa de 10 cm x 15 cm que são encontrados todos os componentes. O regulador de tensão foi colocado em um dissipador de calor e conectado através de um cabo a placa, pois ao ser ligado esquenta bastante e a sua localização na placa fica muito próxima do modem GSM SIM900. O display LCD, o buzzer, o sensor e os Led's são ligados à placa através de cabos também.

### 5.1.1 – Fluxograma Geral do Sistema

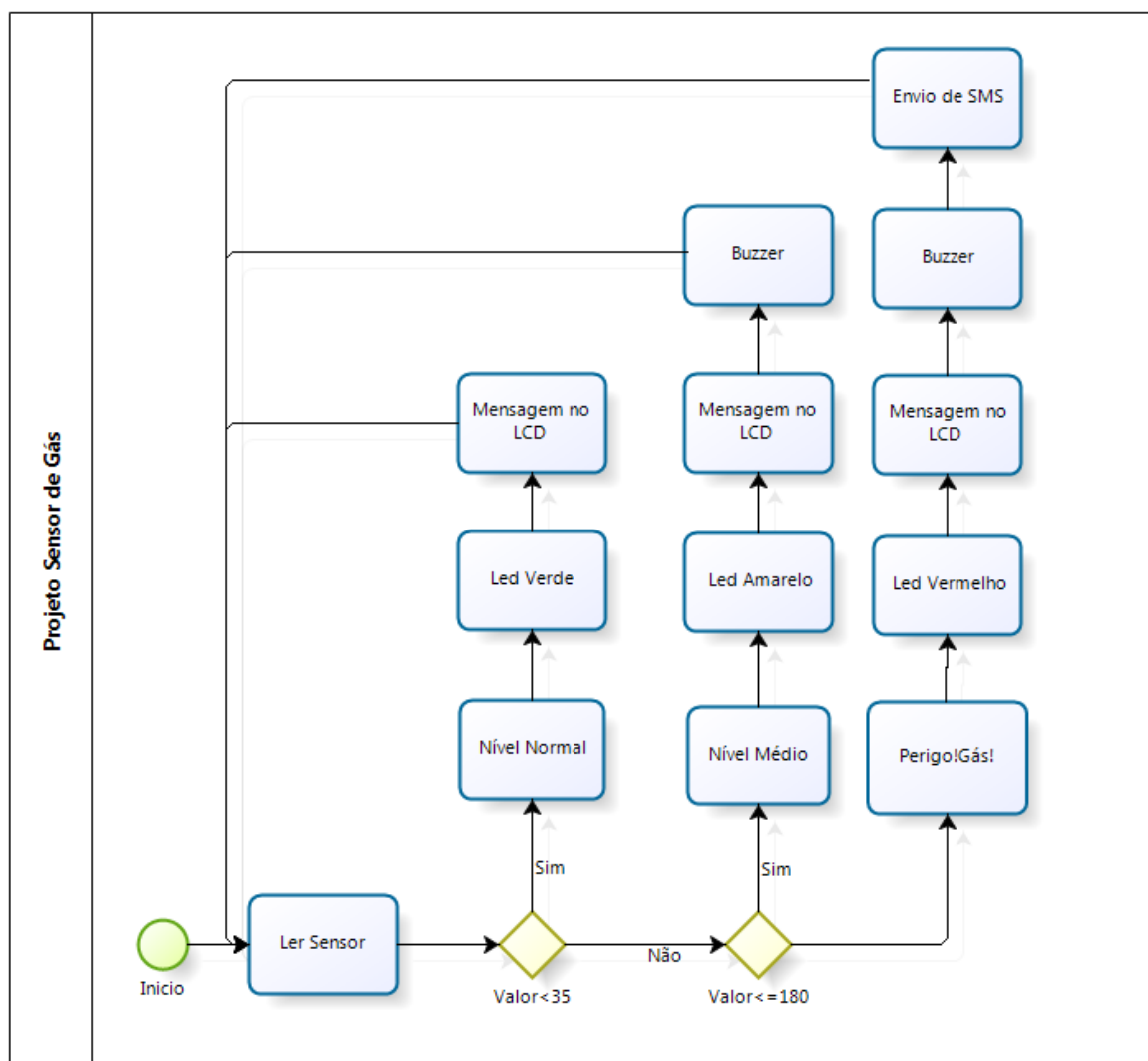


Figura 5.3 – Fluxograma do Projeto (Fonte: Autor)

## 5.2 Elaboração do Circuito

Após determinar o que terá 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.

## 5.2.1 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. Nelo foi montado o circuito do projeto contendo praticamente todos os componentes, com exceção do sensor de gás que foi simulado por um potenciômetro e o módulo GSM que foi simulado a partir de um Led. A Figura 5.3 ilustra a simulação realizada no *software* ISIS.

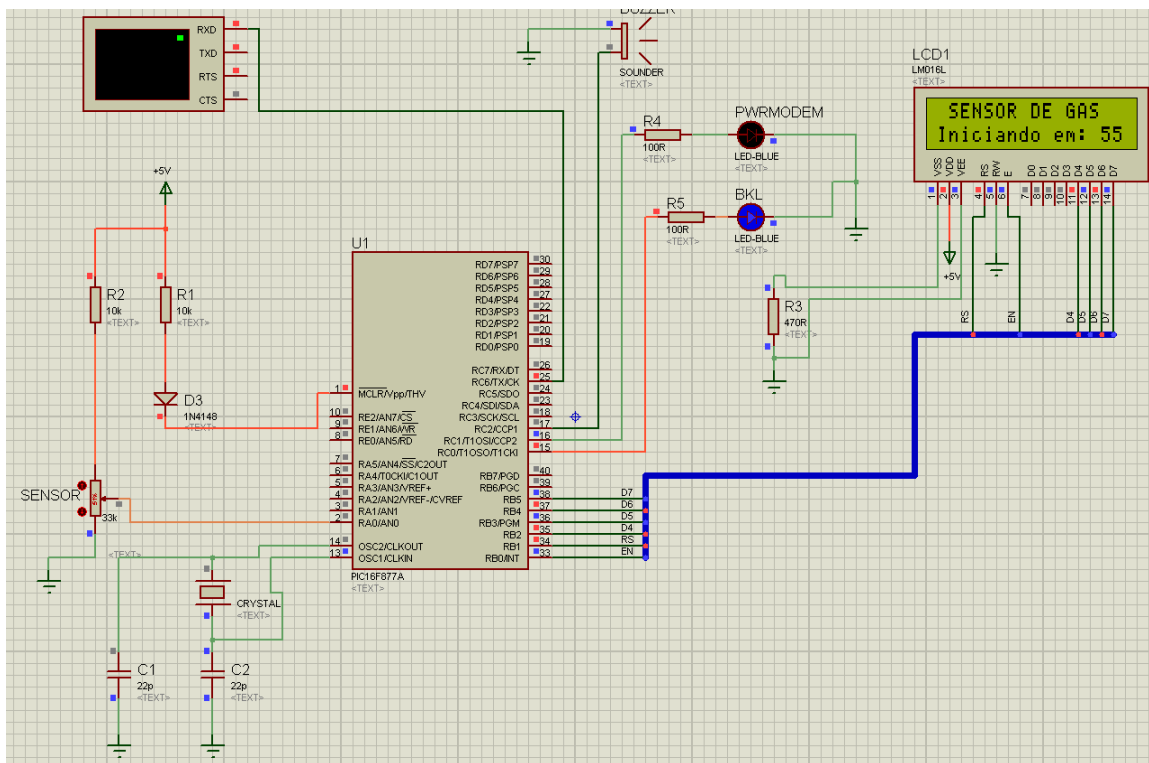


Figura 5.3 – Simulação do projeto no software ISIS (Fonte: Autor)

Para realizar a simulação no *software* ISIS foi necessário realizar a programação do microcontrolador PIC16F877A que será apresentada a seguir no 5.3 Escrita do Código Fonte.

## 5.2.2 Proteus ARES 7.7 Professional

Com o *software* ARES 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 5.4 é mostrada a placa projetada no *software* ARES.

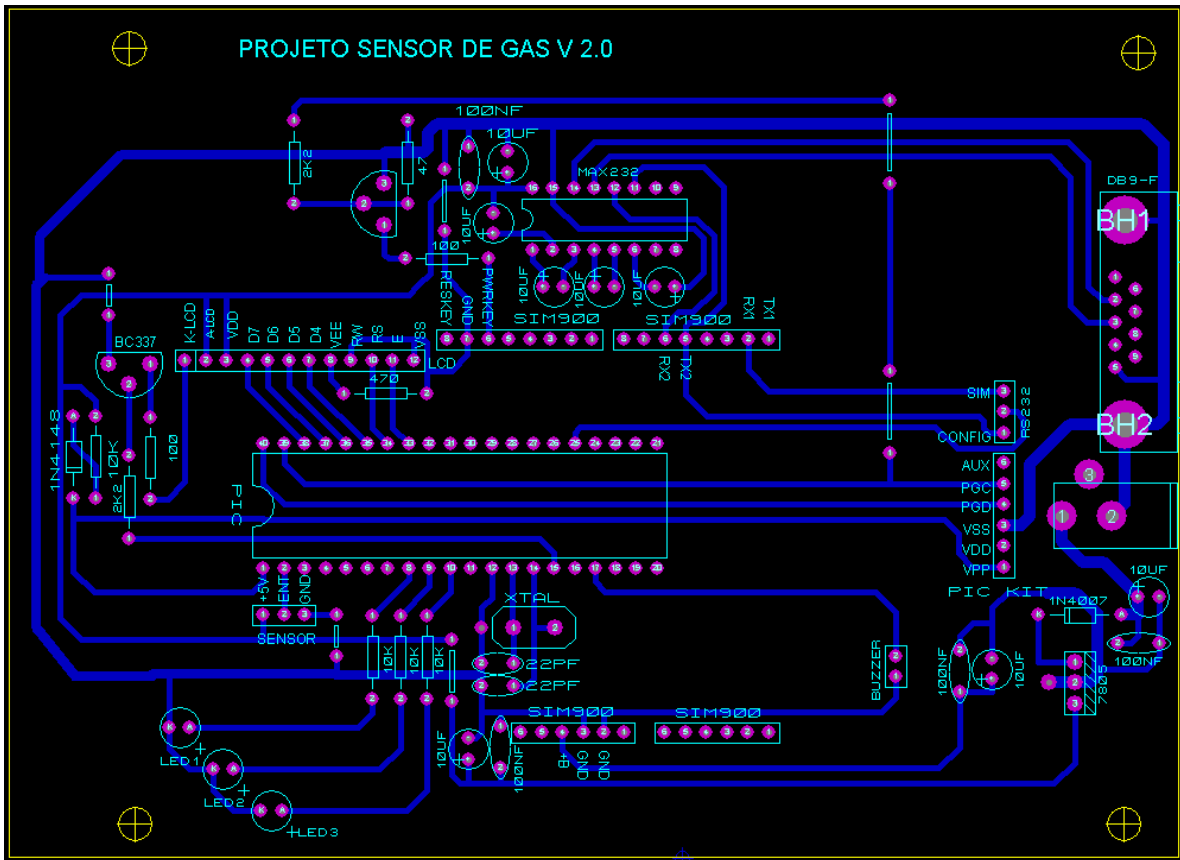


Figura 5.4 – Placa projetada no software ARES (Fonte: Autor)

### 5.3 Elaboração do Código Fonte para o Microcontrolador PIC

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 PIC16F877A pelo PICKit 2. Para gerar o código em “.hex” foi utilizado o PIC C Compiler, aos poucos o código fonte foi sendo gerado para o devido funcionamento dos componentes.

### 5.3.1 Escrita do Código Fonte

A seguir será explicada a escrita do código fonte, com as funções do programa para o devido funcionamento do projeto com todos seus componentes.

```

*****/
#include <16F877A.h> //Inclui a biblioteca p/ o microcontrolador PIC16F877A.
#define adc=8 //Utiliza conversor AD com resolução p/ 8 bits.
#define delay(clock=1000000) //Cristal oscilador 10MHz.
#define fuses hs, nowdt, put, brownout, nolvp
#define rs232(baud=9600, xmit=PIN_C6) //velocidade de transmissão 9600 bps, sem paridade, 8 bits de
//dados, pino TX RC6.
#include <lcd-v2.0.h> //Biblioteca p/ o módulo LCD (Fabio Pereira). Esta biblioteca sofreu
//alteração para utilizar os pinos R/W, RS e Enable do LCD nos Ports C e D em vez do Port E do PIC.
#include <Modem.h> // Funcao para ativar o modem GSM/GPRS SIM900
#define nivel_normal 35
#define nivel_perigoso 180

```

Figura 5.5 – Código Fonte Parte I (Fonte: Autor)

Na Figura 5.5 é mostrado o início da escrita do código fonte, onde as bibliotecas necessárias para o funcionamento do programa foram declaradas. Biblioteca do microcontrolador PIC16F877A, a biblioteca para o display LCD criada pelo Fabio Pereira com alterações do autor para utilização dos pinos do microcontrolador, a função criada para a ativação do modem GSM SIM900. O uso da diretiva #fuses que serve para definir as configurações do PIC. Por exemplo, a configuração da serial que ele usará ou a velocidade do clock usado. E a definição do valor do nível normal e do nível perigoso, estes valores foram definidos em testes onde o protótipo foi ligado e verificado o nível do gás no ambiente normal e no ambiente com vazamento de gás.

```

*****/
#define PWR_MODEM pin_b6 //RB6 pino de saída p/ modem GSM/GPRS.
#define BUZZER pin_c2 //Pino RC2 conectado a um buzzer.
#define BK_LIGHT pin_c0 //RC0 Led do backlight do LCD.
#define LED_VERDE pin_e0 //RE0 Led VERDE
#define LED_AMARELO pin_e1 //RE0 Led AMARELO
#define LED_VERMELHO pin_e2 //RE0 Led VERMELHO

/*****Inicialização das variáveis.*****/
int Valor=0; //Variável global p/ leitura do sensor de gás no canal ADC.

/*****Função p/ leitura do ADC*****/
void Canal_ADC()
{
    set_adc_channel(0); //Leitura do sensor de gás em RA0.
    delay_ms(10); //Tempo necessário p/ estabilizar as configurações ADC.
    read_adc(adc_start_only);
    Valor = read_adc(adc_read_only); //Atribui à variável a leitura do canal ADC 0.
    delay_ms(100); //Tempo entre as leituras do sensor de gás.
}

/*****Função do backlight*****/
void Back_Light()
{
    output_high(BK_LIGHT); //Ativa pino RC0 do backlight.
    return; //retorna.
}
//fim void Back_Light.

```

Figura 5.6 – Código Fonte Parte II (Fonte: Autor)

Na Figura 5.6 é mostrada a parte do código onde se definem alguns pinos do microcontrolador onde estão conectados os componentes: modem GSM SIM900, o buzzer, o backlight do display LCD e os Leds. Mostra a declaração da variável “valor” que foi usada para a leitura do sensor de gás no canal ADC do PIC16F877A. E também mostra as funções para a leitura do canal ADC e a função do backlight do LCD.

```

/*****Função Tempo*****/
void Tempo()
//Função para contagem regressiva de 1 minuto para o aquecimento do sensor de gás
// tempo necessário para estabilizar o sensor para medição de GLP
{
int Conta_Seg=00; //Declara e inicializa a variável p/ timer = 60s
while(true) //Enquanto verdadeiro...
{
    Conta_Seg--; //Decrementa segundos.
    if (Conta_Seg > 59)
    {
        Conta_Seg = 59;
    }
    printf("SENSOR DE GAS\r\nIniciando em:%02u", Conta_Seg);

    lcd_escreve('\f');
    lcd_pos_xy(2,1);
    printf(lcd_escreve,"SENSOR DE GAS\r\nIniciando em: %02u", Conta_Seg);

    if(Conta_Seg == 0)
    {
        return;
    }

    delay_ms(1000); //Decrementa contador de segundos a cada 1s.
} //fim do while
} //fim void Tempo.

```

**Figura 5.7 – Código Fonte Parte III (Fonte: Autor)**

A Figura 5.7 ilustra a função do tempo que foi criada para realizar uma contagem regressiva de 1 minuto para o aquecimento do sensor de gás e estabilização para a leitura do GLP, este tempo é obrigatório para o funcionamento correto do sensor de gás. E a Figura 5.8 abaixo mostra o programa principal do projeto, onde está a configuração do canal ADC, a rotina do backlight, a rotina do LCD, a rotina do tempo, a ativação dos pinos do modem GSM.

O programa após a contagem regressiva de 1 minuto faz a leitura do sensor e exibe este valor no display LCD, se o valor medido for menor que o nível normal (menor que 35 ppm) o programa entra na primeira condição que acionará o LED verde e exibirá mensagem no display. Se o valor for entre o nível normal e o nível perigoso (maior que 180 ppm) a segunda condição irá acionar o LED amarelo, exibirá mensagem no display e emitirá um aviso sonoro. E por ultimo se o valor for maior que o nível perigoso, o programa entrará na terceira condição acionando o LED vermelho, exibindo mensagem no display, emitindo aviso sonoro e enviando SMS para o celular previamente cadastrado na função do modem GSM explicada na Figura 5.9.



```

/*****Programa Principal*****/
set_tris_b(0b00000000);

void main()
{
  int x;
  setup_adc_ports(RA0_ANALOG); //Configura o ADC p/ RA0 como pino analógico, demais pinos digitais.
  setup_adc(ADC_CLOCK_INTERNAL); //ADC com clock interno.
  setup_comparator(NC_NC_NC_NC); //Desliga comparadores.
  setup_vref(FALSE); //Referência de tensão desligada.

  Back_light();// //Chama a rotina para acionar o BACK LIGHT do display
  lcd_ini(); //Chama a rotina de inicialização do display
  Tempo(); //Chama rotina de tempo=60s p/ estabilização do aquecimento da resistência interna do sensor.

  output_high(PWR_MODEM); //Ativa saída p/ modem GSM/GPRS.
  delay_ms(1000); // Tempo para estabilização da rede GSM
  output_low(PWR_MODEM); // Desativa saída do modem GSM/GPRS
  delay_ms(3000); // Tempo necessário para o envio do SMS

  do
  {
    Canal_ADC(); // Ler sensor
    //Leitura da conversão ADC.

    lcd_pos_xy(2,1); //Posiciona o cursor na linha 1 coluna 2.
    printf(lcd_escreve,"SENSOR DE GAS 2.0\r\n"); //Mostra na linha 1 do LCD "SENSOR DE GAS" e na linha 2...
    printf(lcd_escreve,"\r\nVALOR MEDIDO:%u", Valor); //...o valor da leitura do sensor de gás.
    delay_ms(1000); //Espera 1s.

    if(Valor < nivel_normal)
    {
      output_low(LED_AMARELO); //Desativa LED amarelo pino RE1
      output_low(LED_VERMELHO); //Desativa LED vermelho pino RE2
      output_high(LED_VERDE); //Ativa LED verde pino RE0
      Canal_ADC(); //Leitura da conversão ADC.
      lcd_escreve('\f'); //Limpa o LCD.
      lcd_pos_xy(2,1); //Posiciona o cursor na linha 1 coluna 2.
      printf(lcd_escreve,"Nivel Normal\r\n"); //Mostra na linha 1 do LCD "NIVEL NORMAL" e na linha 2...
      printf(lcd_escreve,"\r\nVALOR MEDIDO:%u", Valor); //...o valor da leitura do sensor de gás.
      delay_ms(1000); //Espera 1s p/ tempo entre as leituras.
    }
    else
    if(Valor >= nivel_normal && Valor < nivel_perigoso) //Valor entre nivel normal e perigoso
    {
      output_high(LED_AMARELO); //Ativa LED amarelo pino RE1
      output_low(LED_VERMELHO); //Desativa LED vermelho pino RE2
      output_low(LED_VERDE); //Desativa LED verde pino RE0

      output_high(BUZZER); //Ativa Buzzer.
      delay_ms(200); //Desativa Buzzer.
      output_low(BUZZER);
      delay_ms(200);

      output_high(BUZZER); //Ativa Buzzer.
      delay_ms(200); //Desativa Buzzer
      output_low(BUZZER);
      delay_ms(100);

      Canal_ADC(); //Leitura da conversão ADC.
      lcd_escreve('\f'); //Limpa o LCD.
      lcd_pos_xy(2,1); //Posiciona o cursor na linha 1 coluna 2.
      printf(lcd_escreve,"Vazamento Medio\r\n"); //Mostra na linha 1 do LCD "NIVEL ALARMANTE" e na linha 2...
      printf(lcd_escreve,"\r\nVALOR MEDIDO:%u", Valor); //...o valor da leitura do sensor de gás.
      delay_ms(1000); //Espera 1s p/ tempo entre as leituras.
    }

    else
    {
      output_low(LED_AMARELO); //Desativa LED amarelo pino RE1
      output_high(LED_VERMELHO); //Ativa LED vermelho pino RE2
      output_low(LED_VERDE); //Desativa LED verde pino RE0

      output_high(BUZZER); //Ativa Buzzer.
      delay_ms(100); //Desativa Buzzer.
      output_low(BUZZER);
      delay_ms(100);

      output_high(BUZZER); //Ativa Buzzer.
      delay_ms(100); //Desativa Buzzer.
      output_low(BUZZER);
      delay_ms(75);

      modemgsn(); //Chama função do modem para o envio do SMS

      Canal_ADC(); //Leitura da conversão ADC.
      lcd_escreve('\f'); //Limpa o LCD.
      lcd_pos_xy(2,1); //Posiciona o cursor na linha 1 coluna 2.
      printf(lcd_escreve,"PERIGO!GAS!\r\n"); //Mostra na linha 1 do LCD "Nivel PERIGOSO" e na linha 2...
      printf(lcd_escreve,"\r\nVALOR MEDIDO:%u", Valor); //...o valor da leitura do sensor de gás.
      delay_ms(1000); //Espera 1s p/ tempo entre as leituras.
    }
  }while(true);
}
//Fin do void main.
/*****

```

Figura 5.8 – Código Fonte Parte IV (Fonte: Autor)

Conforme mostrado na Figura 5.6 o código do programa principal é onde todas as funções necessárias são chamadas e juntas fazem com que o programa funcione conforme é mostrado na Figura 5.3 o fluxograma.

A função criada para o acionamento do modem GSM SIM900 é mostrada na Figura 5.9, nesta função é onde os comandos AT para o funcionamento do modem são dados, o comando principal desta função é o “AT+CMGF=1” que é o comando para o envio do SMS, logo em seguida é utilizado o comando “AT+CMGS=+556181087555” que determina o número do celular para onde a mensagem será enviada. E por ultimo é inserida a mensagem a ser enviada.

```

/*****Função p/ envio de SMS pelo Modem SIM900*****/

Nome do Arquivo: Modem.h
Versão: 1.1.2

Descrição: Função para teste do modem SIM900 GSM/GPRS da ICosat v1.0 comercializado pela
ITEAD STUDIO (www.iteadstudio.com).

Autor: Jefferson Silva Santos
Compilador: PIC COMPILER Versão 3.43.
Ambiente de simulação: Proteus 7.8 ISIS Professional.
Microcontrolador utilizado: PIC16F877A.
Data: Outubro de 2012.

*****/

void modemgsm()
{
    printf("AT\n\r"); //Verifica se o modem está conectado.
    delay_ms(500);

    printf("AT&K0\n\r"); //Desativa a compactação de dados.
    delay_ms(500);

    printf("ATE0\n\r"); //Desliga o modo eco.
    delay_ms(500);

    printf("AT+CMGF=1\n\r"); //Envio de SMS modo texto (CMGF=1), modo PDU (CMGF=0).
    delay_ms(500);

    printf("AT+CMGS=\"+556181087555\"\n\r"); //Número do telefone da Jefferson no qual a msg será enviad.
    delay_ms(1000);

    printf("Alerta! Gas Uazando!\n\r"); //Mensagem a ser enviada.
    delay_ms(1000);

    putc(0x1a);putc(0x1a);putc(0x1a); //Finaliza o envio de SMS.
    delay_ms(500);
} //Fim.

/*****
|

```

**Figura 5.9 – Código Fonte do Modem GSM (Fonte: Autor)**

### 5.3.2 Compilação e Gravação do Código Fonte

A compilação do código fonte foi realizada no *software* PIC C Compiler, a Figura 5.10 ilustra o momento da compilação do código onde o *software* dá algumas informações sobre a compilação. É mostrado o tempo de compilação, o número de linhas do código, se ocorreu erros, e a porcentagem da memória do microcontrolador que o programa irá utilizar de memória ROM e RAM.

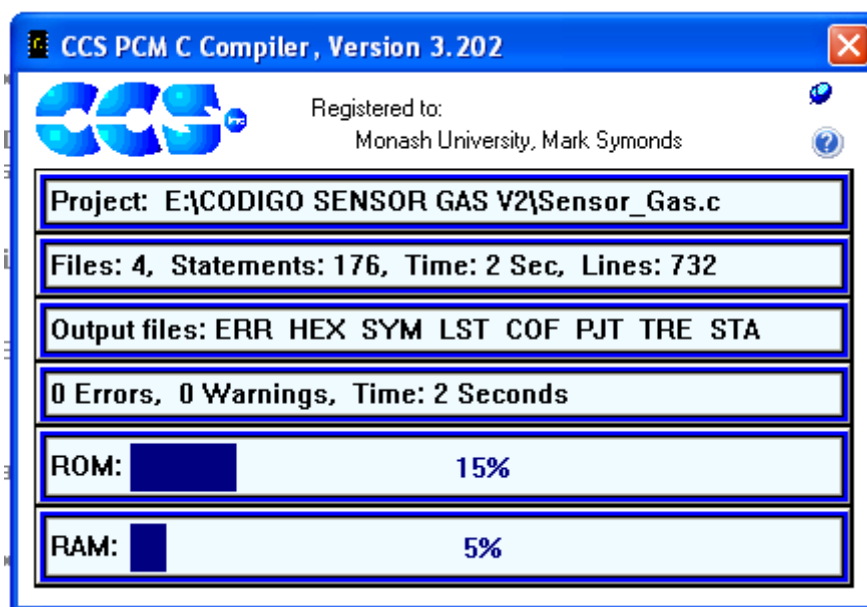


Figura 5.10 – Tela do PIC C Compiler (Fonte: Autor)

A gravação do código fonte foi realizada com a utilização do PICKit 2 que é descrita no 4.1.3 Utilização do Kit de gravação PICKit2. É um procedimento simples, com o PIC16F877A encaixado no soquete ZIF é só conectar o cabo USB no computador e executar o *software* do PICKit 2, onde é carregado o código “.hex” e gravado no microcontrolador.

## 5.4 Montagem do Protótipo

A montagem do protótipo foi feita em uma caixa de acrílico que foi reutilizada de uma caixa de bombons, nesta caixa foram feitos os furos para os encaixes dos componentes do projeto e para parafusar a placa e o display LCD na caixa. Foram feitos alguns pares de cabos para as ligações do display LCD, do buzzer, dos LEDs e do regulador de tensão. Este regulador de tensão foi colocado fora da placa,

pois ele aquece bastante com o uso e poderia danificar o funcionamento do projeto. O protótipo é alimentado por uma fonte de 12 V e ligada na placa através do conector tipo jack P4.

O circuito foi projetado para o encaixe perfeito do modem GSM SIM900 que já possuía uma extensão para a antena GSM que foi utilizada também e feito um furo na caixa para a colocação da antena.

A caixa onde o protótipo foi colocada tem um tamanho de 22 cm x 12 cm x 3 cm, ela já possui parafusos para a fixação no local onde o detector poderá ser instalado. O melhor local para a instalação do protótipo é perto do botijão de gás aproximadamente na mesma altura do botijão, pois o gás GLP é mais pesado que o ar e tem a tendência de descer sendo assim possível a detecção do vazamento.

## CAPÍTULO 6 – 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.

### 6.1 Simulações

Como dito no item 5.2.1, do capítulo 5, 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 2 para a gravação do PIC. O código fonte foi alterado em vários momentos até conseguir o funcionamento correto do modem, dos avisos sonoros e dos acionamentos dos LEDs no momento correto. O modem foi testado e configurado através da conexão serial RS232 da placa e não apresentou problema algum.

No código fonte do projeto foram definidos dois níveis de gás, o nível normal é quando o sensor detecta menos de 35 ppm do GLP no ambiente e o nível perigoso onde o valor medido pelo sensor supera 180 ppm. Estes valores foram definidos com a realização de testes em laboratório e em uma cozinha, as medições do nível do gás GLP no ambiente normal não variaram acima de 30 ppm sendo assim escolhido 35ppm como um nível normal de GLP no ambiente. Com o teste do nível o ambiente foi liberado gás GLP e o valor medido no sensor para muito gás no ambiente foi de aproximadamente 175 ppm sendo desta forma acima 180 ppm foi determinado um nível muito perigoso de vazamento de gás. O valor do nível médio ficou entre o nível normal e o perigoso. Estes níveis podem ser simplesmente alterados na programação para se adequarem melhor a um ambiente real, estes valores foram definidos para a demonstração do protótipo para a banca que será utilizado um isqueiro para simular o vazamento do gás.

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.

## **6.2 Problemas Encontrados**

Devido o problema inesperado descrito acima a placa foi refeita em menos de uma semana e com isso alguns pequenos problemas não foram detectados no momento de projetar a placa no ARES. Algumas trilhas foram esquecidas, mas com a placa já confeccionada os ajustes necessários para o funcionamento correto do projeto foram realizados.

A aquisição dos componentes também não foi tão fácil, pois o fornecedor dos componentes não possuía todos eles, e com isso alguns componentes foram comprados fora de Brasília.

Com a falta de uma trilha na ligação do pino de alimentação do PIC não foi possível gravar o microcontrolador, pois é necessário 5 V para isso e sem a ligação correta foi medido com um multímetro 3,8 V, com isso foi descoberto a falta desta trilha e foi soldado um fio para realizar esta ligação e assim a gravação do PIC funcionar corretamente.

Após realizar a solda da placa a trilha foi testada e todas as ligações funcionaram corretamente, este teste foi feito com o auxílio de um multímetro. Com as trilhas funcionando e o microcontrolador gravado, o protótipo foi ligado com a fonte de 12 V e demonstrou funcionamento conforme o planejado e proposto para o projeto.

## **6.4 Orçamento do Projeto**

A aquisição dos componentes eletrônicos para o projeto foi realizada e foi fundamental para o projeto. Abaixo o Quadro 6.1 detalha dos componentes e o seus devidos custos.

Quadro 6.1 – Custo do Projeto

Item	Custo Unitário	Quantidade	Custo Total
GSM/GPRS ICom Sat – SIM900	R\$ 185,00	1	R\$ 185,00
PIC16F877A - MicroChip	R\$ 14,30	3	R\$ 42,90
Sensor de Gases Inflamáveis MQ-9	R\$ 37,00	1	R\$ 37,00
Soquete CI 40 Pinos	R\$ 0,41	1	R\$ 0,41
MAX232	R\$ 1,65	1	R\$ 1,65
Soquete CI 16 Pinos	R\$ 0,18	1	R\$ 0,18
Conector Barra de Pinos Macho (1x40x11,2 – 180°)	R\$ 0,50	2	R\$ 1,00
Conector Barra de Pinos Macho (1x40x11,2 – 90°)	R\$ 0,50	2	R\$ 1,00
Capacitor Cerâmico 22pF	R\$ 0,07	2	R\$ 0,14
Capacitor Cerâmico 100nF	R\$ 0,07	4	R\$ 0,28
Capacitor Eletrolítico 10µF	R\$ 0,08	8	R\$ 0,64
Cristal Quartzo 10MHz	R\$ 0,50	1	R\$ 0,50
Regulador de Tensão LM7805	R\$ 0,79	1	R\$ 0,79
Diodo 1N4007	R\$ 0,05	1	R\$ 0,05
Diodo 1N4148	R\$ 0,04	1	R\$ 0,04
Conector Jack DC – J4 (PCB)	R\$ 0,55	1	R\$ 0,55
Conector DB9 Fêmea 90° (PCB)	R\$ 1,65	1	R\$ 1,65
Buzzer 5v	R\$ 1,38	1	R\$ 1,38
Transistor BC337 (NPN)	R\$ 0,11	2	R\$ 0,22
Resistor de Filme de Carbono 10k – 1/4W	R\$ 0,04	4	R\$ 0,16
Resistor de Filme de Carbono 100R – 1/4W	R\$ 0,04	2	R\$ 0,08
Resistor de Filme de Carbono 2k2 – 1/4W	R\$ 0,04	2	R\$ 0,08
Resistor de Filme de Carbono 47K – 1/4W	R\$ 0,04	1	R\$ 0,04
LED Transparente de Alto Brilho Verde 5mm	R\$ 0,66	1	R\$ 0,66
LED Transparente de Alto Brilho Amarelo 5mm	R\$ 0,28	1	R\$ 0,28

LED Transparente de Alto Brilho Vermelho 5mm	R\$ 0,50	1	R\$ 0,50
Cabo Manga Sem Blindagem 16x26 (1m)	R\$ 3,30	1	R\$ 3,30
Display Cristal Líquido (LCD 16x02)	R\$ 19,95	1	R\$ 19,95
Terminal Modu Fêmea 1 T	R\$ 0,06	100	R\$ 6,00
Conector Modu Fêmea (Alojamento 01x02 – 180°)	R\$ 0,06	10	R\$ 0,60
Conector Modu Fêmea (Alojamento 01x04 – 180°)	R\$ 0,07	20	R\$ 1,40
Placa Fenolite Virgem (Face simples 30x30cm)	R\$ 13,20	1	R\$ 13,20
Conector Barra de Pinos Fêmea (1x6x11,2 – 180°)	R\$ 0,25	2	R\$ 0,50
Conector Barra de Pinos Fêmea (1x8x11,2 – 180°)	R\$ 0,31	2	R\$ 0,31
Fonte de Alimentação 12 V / 1A	R\$ 25,00	1	R\$ 25,00
<b>TOTAL</b>	-	-	<b>R\$ 347,44</b>

FONTE: Autor.

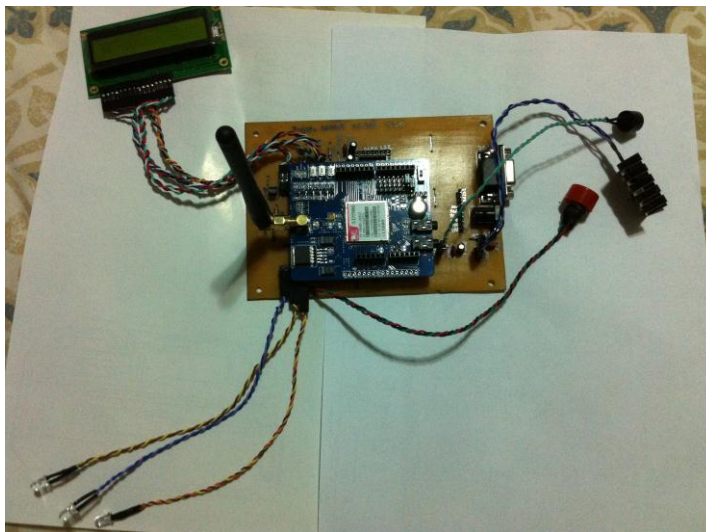
O custo do projeto foi R\$347,44 (US\$165,45) e na Figura 6.1 são mostrados os componentes adquiridos para a confecção do projeto.



Figura 6.1 – Componentes do Projeto (Fonte: Autor)



## 6.5 Protótipo Final



**Figura 6.2 – Projeto Completo (Fonte: Autor)**

Conforme mostrado na Figura 6.2, o projeto com todos seus componentes ligados.

## **CAPÍTULO 7 – CONSIDERAÇÕES FINAIS**

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

### **7.1 Conclusões**

Foi desenvolvido neste trabalho um dispositivo de detecção de vazamento de gás, com a função de envio de um SMS para um celular cadastrado. 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, detecção de vazamento de gás GLP em ambientes domiciliares ou até industriais.

A missão do sistema de detecção de gás é de uma sentinela 24 horas, monitorando o ambiente onde o projeto está instalado, alertando a ocorrência de um evento anormal que indique um vazamento de gás GLP.

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 microcontrolador PIC16F877A 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 início do projeto. A integração entre o modem GSM SIM900 e o microcontrolador realizada através do MAX232 foi funcional também funcionando conforme o planejado.

### **7.2 Propostas para Trabalhos Futuros**

Uma boa modificação no projeto seria adicionar mais um sensor para a detecção de monóxido de carbono, papel que o sensor utilizado, o MQ-9, pode realizar, mas seria necessária uma nova programação porque o funcionamento do sensor para esta situação é bem diferente.

O modem GSM SIM900 possui várias outras funcionalidades que poderiam ser utilizadas, como: caso o usuário queira saber o valor medido no ambiente em um determinado momento enviaria um SMS para o dispositivo e receberia a resposta do valor medido. Ou até poderia ser configurado para enviar um status do ambiente de tempo em tempo para um melhor acompanhamento do usuário.

O PIC16F877A possui vários pinos e este foi um dos motivos da escolha deste microcontrolador, mas não foram utilizados todos os pinos do mesmo. Sendo assim podem ser adicionados outros componentes para complementar o protótipo e para aumentar o leque de funções que o microcontrolador pode assumir.

Seria uma boa opção a inclusão de um atuador mecânico para a válvula de gás, caso seja identificado o vazamento este atuador mecânico fecharia a válvula evitando assim que o gás continue vazando.

## REFERÊNCIAS BIBLIOGRÁFICAS

BERNAL, P. S. M. **Comunicações Móveis: Tecnologias e Aplicações**. 1ª. Ed. São Paulo: Érica, 2010.

CALENTE, A; PIO, A; MENEZES, R. T. **Técnicas Preditivas Ligadas à Inspeção**. Disponível em: < <http://www.abraman.org.br/Arquivos/50/50.pdf>> Acessado em: 08 de out. 2012.

CAMPOS, D. A. L. **Sistema de Alarme de Intrusão utilizando o SMS**. Brasília: Faculdade de Tecnologia e Ciências Sociais Aplicada, UniCEUB, 2010.

CANZIAN, E. **Comunicação Serial RS-232**. Disponível em < [http://www.capriconsultorios.com/Aula4-Comun\\_serial.pdf](http://www.capriconsultorios.com/Aula4-Comun_serial.pdf)> Acessado em: 26 de out. 2012.

G1 Notícias. **Arquivo G1: Explosão no shopping de Osasco**. Publicado em 11 de jun 2007. Disponível em < <http://g1.globo.com/Noticias/SaoPaulo/0,,MUL49524-5605,00-ARQUIVO+G+EXPLOSAO+NO+SHOPPING+DE+OSASCO.html>>. Acesso em 08 de out 2012.

GIMENEZ, S. P. **Microcontroladores 8051**. 1ª ed. São Paulo: Pearson, 2005.

GSM World. **GSM**. 2012. Disponível em: < <http://www.gsma.com/aboutus/gsm-technology/gsm/>> Acesso em: 10 de out. 2012.

HENAN HANWEI ELECTRONICS. **MQ-9 DataSheet**: Semiconductor Sensor fot CO/Combustible Gas. Disponível em: <<http://www.roboweb.net/media/downloadable/rw-pl-1483/MQ9.pdf>> Acessado em: 20 set. 2012.

ICOMSAT SIM 900. **SIM 900 DataSheet.** Disponível em: <  
[http://iteadstudio.com/store/images/produce/Shield/IComSat/icomsat\\_DS1.2.pdf](http://iteadstudio.com/store/images/produce/Shield/IComSat/icomsat_DS1.2.pdf)  
>  
Acessado em: 26 out. 2012.

ME COMPONENTES. **Módulo GSM/GPRS,** SIM900D. Disponível em:  
<<http://www.mecomp.com.br/sim900d.html>> Acesso em 26 out 2012.

MICROCHIP TECHNOLOGY INC. **PIC16F87XA.** DataSheet: 28/40/44-Pin Enhanced  
Flash Microcontrollers. 2003. Disponível em: <[http://ww1.microchip.com/downloads/  
en/devicedoc/39582b.pdf](http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf)> Acesso em.

Minulight. **Detector de Vazamento de Gás.** Disponível em: <  
[http://www.minulight.com.br/novo/gas/detectores-de-gas/detector-de-vazamento-de-  
gas-glp-gn/](http://www.minulight.com.br/novo/gas/detectores-de-gas/detector-de-vazamento-de-gas-glp-gn/)>. Acessado em 13 de out. 2012.

Online Directory. **Problemas de Saúde Relacionados a Vazamentos de Gás Natu-  
ral.** Disponível em: < [http://www.fuguitang.com/problemas-de-saude-relacionados-a-  
vazamentos-de-gas-natural.html](http://www.fuguitang.com/problemas-de-saude-relacionados-a-vazamentos-de-gas-natural.html)>. Acessado em 08 de set. 2012.

PEREIRA, F. **Microcontroladores PIC:** Programação em C. 7ª. Ed. São Paulo: Éri-  
ca, 2007.

POLITO, R. F. S. **Prevenção de Acidentes com GLP.** Publicado em 06 de março  
de 2008 Disponível em: <  
[http://www.dgst.cbmerj.rj.gov.br/modules.php?name=Content&pa=showpage&pid=14  
1](http://www.dgst.cbmerj.rj.gov.br/modules.php?name=Content&pa=showpage&pid=141)> Acessado em : 08 de set. 2012

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

SOUZA, D. J. de; LAVINIA, N. C. **Conectando o PIC 16F877A:** Recursos Avança-  
dos. 2ª. Ed. São Paulo: Érica, 2005.

SP Notícias. **Corpo de Bombeiros dá dicas contra o vazamento de gás**. Publicado em 15 de outubro de 2011. Disponível em: <<http://www.saopaulo.sp.gov.br/spnoticias/lenoticia.php?id=216466>>. Acesso em 08 de out. 2012.

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.

Wise Geek. **What Is a SIM Card**. Disponível em: < <http://www.wisegeek.com/what-is-a-sim-card.htm> > Acessado em 9 de out 2012.

ZANCO, W. da S. **Microcontroladores PIC com Base no PIC16F877A: Técnicas de Software e Hardware para Projetos de Circuitos Eletrônicos**. 2ª. Ed. São Paulo: Érica, 2010.

## APÊNDICE A – CÓDIGO FONTE DO DISPOSITIVO

/\*\*\*\*\*Sensor de Gas\*\*\*\*\*/

Descrição:

Nome do arquivo: Sensor\_Gas.c;

Versão: 2.0

Autor: Jefferson Silva Santos;

Micro utilizado: PIC16F877A;

Compilador: IDE PCW C Compiler versão 3.43 CCS;

Ambiente de simulação: ISIS Proteus 7.8 SP2;

Data de criação: Outubro 2012-Versão: 1.0;

\*\*\*\*\*/

```
#include <16f877A.h>           //Inclui a biblioteca p/ o microcontrolador
PIC16F877A.
```

```
#device adc=8                 //Utiliza conversor AD com resolução p/ 8 bits.
```

```
#use delay(clock=10000000)    //Cristal oscilador 10Mhz.
```

```
#fuses hs, nowdt, put, brownout, nolvp
```

```
#use rs232(baud=9600, xmit=PIN_C6) //velocidade de transmissão 9600 bps,
sem paridade, 8 bits dedados, pino TX RC6.
```

```
#include <lcd-v2.0.h>         //Biblioteca p/ o módulo LCD (Fabio Pereira). Esta
biblioteca sofreu alteração para utilizar os
pinos R/W, RS e Enable do LCD nos Ports
C e D em vez do Port E do PIC.
```

```
#include <Modem.h>           // Funcao para ativar o modem GSM/GPRS SIM900
```

```
#define nivel_normal 35      //Constante definida como nivel normal de gas
no ambiente.
```

```
#define nivel_perigoso 180           //Constante definida como nivel perigoso de
                                     gas no ambiente.
```

```
/******Definições dos pinos do PIC*****
```

RC0 Led backlight do LCD.

RC1 pino de saída p/ ativação do módulo GSM/GPRS.

RC6 pino de transmissão serial TTL-RS232 para o PC.

RC7 pino de recepção dos dados no formato TTL232 (não utilizado neste projeto).

RB5 ligado ao pino 14 (DB7) do LCD.

RB4 ligado ao pino 13 (DB6) do LCD.

RB3 ligado ao pino 12 (DB5) do LCD.

RB2 ligado ao pino 11 (DB4) do LCD.

O pino 5 (R/W) do LCD foi ligado ao GND.

RB1 ligado ao pino 4 (RS) do LCD.

RB0 ligado ao pino 6 (E = Enable) do LCD.

O pino 1 (Vss) do LCD ligado ao GND.

O pino 2 (Vdd) do LCD ligado ao +5V.

O pino 3 (Vee) do LCD ligado ao trimpot de contraste.

OBS: Na biblioteca "lcd-v2.0.h" as definições já estão declaradas.

```
*****/
```

```
#define PWR_MODEM          pin_b6      //RB6 pino de saída p/ modem
                                     GSM/GPRS.
```

```
#define BUZZER            pin_c2      //Pino RC2 conectado a um buzzer.
```

```
#define BK_LIGHT          pin_c0      //RC0 Led do backlight do LCD.
```



```

#define LED_VERDE      pin_e0    //RE0 Led VERDE
#define LED_AMARELO   pin_e1    //RE0 Led AMARELO
#define LED_VERMELHO  pin_e2    //RE0 Led VERMELHO

/*****Inicialização das variáveis.*****/

int Valor=0; //Variável global p/ leitura do sensor de gás no canal ADC.

/*****Função p/ leitura do ADC*****/

void Canal_ADC()
{
set_adc_channel(0);      //Leitura do sensor de gás em RA0.
delay_ms(10);           //Tempo necessário p/ estabilizar as configurações
                        ADC.

read_adc(adc_start_only);

Valor = read_adc(adc_read_only); //Atribui à variável a leitura do canal ADC 0.
delay_ms(100);         //Tempo entre as leituras do sensor de gás.
}

/*****Função do backlight*****/

void Back_Light()
{
output_high(BK_LIGHT); //Ativa pino RC0 do backlight.
}

```

```

return;          //retorna.
}                //fim void Back_Light.

/*****Função Tempo*****/

void Tempo()     //Função para contagem regressiva de 1 minuto para o
                 //aquecimento do sensor de gás tempo necessário para
                 //estabilizar o sensor para medição de GLP
{
int Conta_Seg=00;    //Declara e inicializa a variável p/ timer = 60s
    while(true)     //Enquanto verdadeiro...
    {
        Conta_Seg --; //Decrementa segundos.
        if (Conta_Seg > 59)
        {
            Conta_Seg = 59;
        }
        printf("SENSOR DE GAS\r\nIniciando em:%02u", Conta_Seg);
            //Mostra no terminal serial.
        lcd_escreve('\f'); //Limpa o LCD.
        lcd_pos_xy(2,1); //Posiciona o cursor na
            //linha 1 coluna 2.
        printf(lcd_escreve,"SENSOR DE GAS\r\nIniciando em: %02u", Con-
ta_Seg); //Mostra no LCD linha 1:
            //"SENSOR DE GAS" e na linha 2:"Iniciando em: " e a
            //quantidade de tempo em segundos.

```

```

        if(Conta_Seg == 0)                                //Se valor de Conta_Seg = 0 ...
        {
            return;                                       //retorna.
        }

        delay_ms(1000); //Decrementa contador de segundos a cada 1s.

    } //fim do while
} //fim void Tempo.

/*****Programa Principal*****/

set_tris_b(0b00000000);

void main()
{
    int x;

    setup_adc_ports(RA0_ANALOG);           //Configura o ADC p/ RA0 como pino analógico, demais pinos digitais.

    setup_adc(ADC_CLOCK_INTERNAL);        //ADC com clock interno.

    setup_comparator(NC_NC_NC_NC);        //Desliga comparadores.

    setup_vref(FALSE);                     //Referência de tensão desligada.

    Back_Light();//                            //Chama a rotina para acionar o BACK LIGHT do display

    lcd_ini();                               //Chama a rotina de inicialização do display

```



```

output_low(LED_VERMELHO);           //Desativa LED vermelho
                                     pino RE2

output_high(LED_VERDE);             //Ativa LED verde pino
                                     RE0

Canal_ADC();                        //Leitura da conversão ADC.

lcd_escreve("\f");                  //Limpa o LCD.

lcd_pos_xy(2,1);                    //Posiciona o cursor na linha 1
                                     coluna 2.

printf(lcd_escreve,"Nivel Normal\r\n"); //Mostra na linha 1 do
                                     LCD "NIVEL NORMAL" e na linha 2...

printf(lcd_escreve,"\r\nVALOR MEDIDO:%u", Valor); //...o valor da
                                     leitura do sensor de gás.

delay_ms(1000);                     //Espera 1s p/ tempo entre as leituras.
}

else

if(Valor >= nivel_normal && Valor < nivel_perigoso) //Valor entre nivel
normal e perigoso

{

output_high(LED_AMARELO);           //Ativa LED amarelo
                                     pino RE1

output_low(LED_VERMELHO);           //Desativa LED ver-
                                     melho pino RE2

output_low(LED_VERDE);              //Desativa LED verde pi-
                                     no RE0

output_high(BUZZER);                //Ativa Buzzer.

delay_ms(200);

```

```

output_low(BUZZER);           //Desativa Buzzer.
delay_ms(200);

output_high(BUZZER);         //Ativa Buzzer.
delay_ms(200);
output_low(BUZZER);         //Desativa Buzzer
delay_ms(100);

Canal_ADC();                 //Leitura da conversão ADC.
lcd_escreve('\f');          //Limpa o LCD.
lcd_pos_xy(2,1);            //Posiciona o cursor na linha 1
                             coluna 2.
printf(lcd_escreve,"Vazamento Medio\r\n"); //Mostra na linha 1 do
                             LCD "NIVEL ALARMANTE" e na linha 2...
printf(lcd_escreve,"\r\nVALOR MEDIDO:%u", Valor); //...o valor da
                             leitura do sensor de gás.
delay_ms(1000);             //Espera 1s p/ tempo entre as
                             leituras.
}

else
{
output_low(LED_AMARELO);    //Desativa LED amarelo pino RE1
output_high(LED_VERMELHO);  //Ativa LED vermelho pino RE2
output_low(LED_VERDE);      //Desativa LED verde pino RE0

```

```

output_high(BUZZER);    //Ativa Buzzer.

delay_ms(100);

output_low(BUZZER);    //Desativa Buzzer.

delay_ms(100);

output_high(BUZZER);    //Ativa Buzzer.

delay_ms(100);

output_low(BUZZER);    //Desativa Buzzer.

delay_ms(75);

modemgsm();            //Chama função do modem para o envio do SMS

Canal_ADC();            //Leitura da conversão ADC.

lcd_escreve('\f');      //Limpa o LCD.

lcd_pos_xy(2,1);        //Posiciona o cursor na linha 1
                        //                coluna 2.

printf(lcd_escreve,"PERIGO!GAS!\r\n");    //Mostra na linha 1 do
                        //                LCD "Nivel PERIGOSO" e na linha 2...

printf(lcd_escreve,"\r\nVALOR MEDIDO:%u", Valor); //...o valor da
                        //                leitura do sensor de gás.

delay_ms(1000);        //Espera 1s p/ tempo entre as
                        //                leituras.

    }

}while(true);

}                        //Fim do void main.

/*****/

```

## APÊNDICE B – BIBLIOTECA DISPLAY LCD

Esta biblioteca foi criada pelo Fábio Pereira e alterada pelo autor, para o funcionamento do display alterando os pinos do microcontrolador PIC16F877A para os pinos utilizados no projeto.

```

/*****/

/*MOD_LCD.C - Biblioteca de manipulação de módulo LCD          */

/*Autor: Fábio Pereira

/*Esta biblioteca foi modificada para utilizar os pinos do Port D do PIC*/

/*para o LCD:

/*Pino Enable (E) ligado ao pino RB0

/*Pino RS ligado ao pino RB1

/*Pino RW ligado ao pino RD6 (não utilizado)

/*****/

// As definições a seguir são utilizadas para acesso aos pinos do display
// caso o pino RW não seja utilizado, comente a definição lcd_rw

#ifndef lcd_enable

#define lcd_enable    pin_b0           // pino enable do LCD

#define lcd_rs        pin_b1           // pino rs do LCD

//#define lcd_rw      pin_d6           // pino rw do LCD

#define lcd_d4        pin_b2           // pino de dados
                                        d4 do LCD

```



```
#define lcd_d5          pin_b3          // pino de dados
                                   d5 do LCD

#define lcd_d6          pin_b4          // pino de dados
                                   d6 do LCD

#define lcd_d7          pin_b5          // pino de dados
                                   d7 do LCD

#endif

#define lcd_type 2          // 0=5x7, 1=5x10, 2=2 linhas

#define lcd_seg_lin 0x40    // Endereço da segunda linha na RAM do LCD

// a constante abaixo define a seqüência de inicialização do módulo LCD
byte CONST INI_LCD[4] = {0x20 | (lcd_type << 2), 0xf, 1, 6};

byte lcd_le_byte()

// lê um byte do LCD (somente com pino RW)
{
    byte dado;

    // configura os pinos de dados como entradas

    input(lcd_d4);

    input(lcd_d5);

    input(lcd_d6);

    input(lcd_d7);

    // se o pino rw for utilizado, coloca em 1

#ifdef lcd_rw
```

```
        output_high(lcd_rw);

    #endif

    output_high(lcd_enable); // habilita display

    dado = 0;    // zera a variável de leitura

    // lê os quatro bits mais significativos

    if (input(lcd_d7)) bit_set(dado,7);

    if (input(lcd_d6)) bit_set(dado,6);

    if (input(lcd_d5)) bit_set(dado,5);

    if (input(lcd_d4)) bit_set(dado,4);

    // dá um pulso na linha enable

    output_low(lcd_enable);

    output_high(lcd_enable);

    // lê os quatro bits menos significativos

    if (input(lcd_d7)) bit_set(dado,3);

    if (input(lcd_d6)) bit_set(dado,2);

    if (input(lcd_d5)) bit_set(dado,1);

    if (input(lcd_d4)) bit_set(dado,0);

    output_low(lcd_enable); // desabilita o display

    return dado; // retorna o byte lido

}
```

```
void lcd_envia_nibble( byte dado )

// envia um dado de quatro bits para o display

{
```

```

// coloca os quatro bits nas saidas
output_bit(lcd_d4,bit_test(dado,0));
output_bit(lcd_d5,bit_test(dado,1));
output_bit(lcd_d6,bit_test(dado,2));
output_bit(lcd_d7,bit_test(dado,3));

// dá um pulso na linha enable
output_high(lcd_enable);
output_low(lcd_enable);
}

void lcd_envia_byte( boolean endereco, byte dado )
{
// coloca a linha rs em 0
output_low(lcd_rs);

// aguarda o display ficar desocupado
//while ( bit_test(lcd_le_byte(),7) ) ;

// configura a linha rs dependendo do modo selecionado
output_bit(lcd_rs,endereco);

delay_us(100); // aguarda 100 us

// caso a linha rw esteja definida, coloca em 0
#ifdef lcd_rw
output_low(lcd_rw);
#endif
}

```

```
// desativa linha enable
output_low(lcd_enable);

// envia a primeira parte do byte
lcd_envia_nibble(dado >> 4);

// envia a segunda parte do byte
lcd_envia_nibble(dado & 0x0f);
}

void lcd_ini()
// rotina de inicialização do display
{
    byte conta;

    output_low(lcd_d4);
    output_low(lcd_d5);
    output_low(lcd_d6);
    output_low(lcd_d7);
    output_low(lcd_rs);

    #ifdef lcd_rw
        output_high(lcd_rw);
    #endif

    output_low(lcd_enable);

    delay_ms(15);

    // envia uma seqüência de 3 vezes 0x03
```

```
// e depois 0x02 para configurar o módulo
// para modo de 4 bits
for(conta=1;conta<=3;++conta)
{
    lcd_envia_nibble(3);
    delay_ms(5);
}
lcd_envia_nibble(2);
// envia string de inicialização do display
for(conta=0;conta<=3;++conta) lcd_envia_byte(0,INI_LCD[conta]);
}
```

```
void lcd_pos_xy( byte x, byte y)
{
    byte endereco;
    if(y!=1)
        endereco = lcd_seg_lin;
    else
        endereco = 0;
    endereco += x-1;
    lcd_envia_byte(0,0x80|endereco);
}
```

```
void lcd_escreve( char c)
```

```
// envia caractere para o display
{
  switch (c)
  {
    case '\f'      :    lcd_envia_byte(0,1);
                       delay_ms(2);
                       break;

    case '\n'      :

    case '\r'      :    lcd_pos_xy(1,2);
                       break;

    case '\b'      :    lcd_envia_byte(0,0x10);
                       break;

    default        :    lcd_envia_byte(1,c);
                       break;

  }
}
```

```
char lcd_le( byte x, byte y)
// le caractere do display
{
  char valor;

  // seleciona a posição do caractere
  lcd_pos_xy(x,y);

  // ativa rs
```

```
output_high(lcd_rs);  
  
// lê o caractere  
valor = lcd_le_byte();  
  
// desativa rs  
output_low(lcd_rs);  
  
// retorna o valor do caractere  
return valor;  
}
```

## APÊNDICE C – FUNÇÃO DO MODEM GSM

Esta função foi separada do código fonte do projeto pois é uma função que poderá ser utilizada em trabalhos futuros e pode servir para outros projetos.

/\*\*\*\*\*\*Função p/ envio de SMS pelo Modem SIM900\*\*\*\*\*

Nome do Arquivo: Modem.h

Versão: 2.0

Descrição: Função para teste do modem SIM900 GSM/GPRS da IComsat v1.0 comercializado pela ITEAD STUDIO ([www.iteadstudio.com](http://www.iteadstudio.com)).

Autor: Jefferson Silva Santos

Compilador: PIC COMPILER Versão 3.43.

Ambiente de simulação: Proteus 7.8 ISIS Professional.

Microcontrolador utilizado: PIC16F877A.

Data: Outubro de 2012.

\*\*\*\*\*/

```
void modemgsm()
```

```
{
```

```
    printf("AT\n\r"); //Verifica se o modem está conectado.
```

```
    delay_ms(500);
```

```
    printf("AT&K0\n\r"); //Desativa a compactação de dados.
```

```
    delay_ms(500);
```





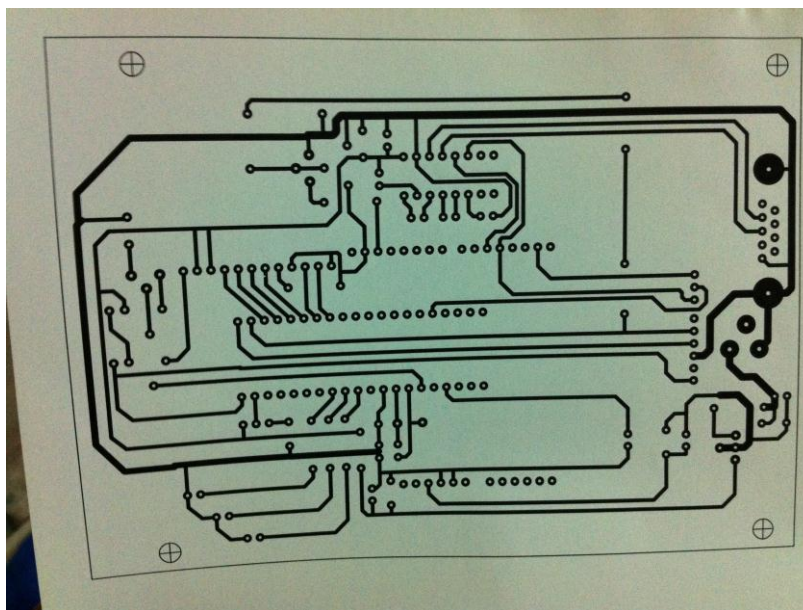
## APÊNDICE D - MONTAGEM DO CIRCUITO NA PLACA

Depois da simulação feita no ISIS e a placa projetada no ARES foi realizado a montagem da placa do projeto. Para confeccionar esta placa foram utilizados os seguintes materiais:

- Placa de fenolite (10 cm x 15 cm)
- Papel Couché 90g (com a impressão da trilha feita no ARES impresso em impressora a Laser)
- Percloroeto de Ferro
- Água
- Sabão
- Recipiente de plástico
- Fita crepe
- Ferro de Passar
- Palha de Aço
- Tyner
- Estilete
- Tesoura
- Papel Toalha
- Fio de arame (encontrado em cabos de rede)
- Caneta para Retroprojektor
- Mini furador

Para se iniciar este processo de confecção a placa de fenolite foi limpa com palha de aço, utilizando a tesoura o papel couché onde está a trilha do circuito que iremos fazer foi recortado, em seguida utilizando a fita crepe a trilha na placa foi colada do lado cobreado prendendo todas suas bordas, tendo todo o cuidado de não

enrugar o papel e deixando a superfície lisa. Na Figura 1 é mostrada a trilha impressa no papel couché.



**Figura 1 – Trilha Impressa (Fonte: Autor)**

Com o auxílio do ferro de passar a trilha foi passada para a placa, com um pouco de pressão e com o calor do ferro o toner impresso no papel couché passa para a placa formando assim a trilha na placa. Após este procedimento a placa foi colocada no recipiente de plástico contendo uma solução de água e sabão, necessária para retirar o papel couché da placa deixando somente a trilha.

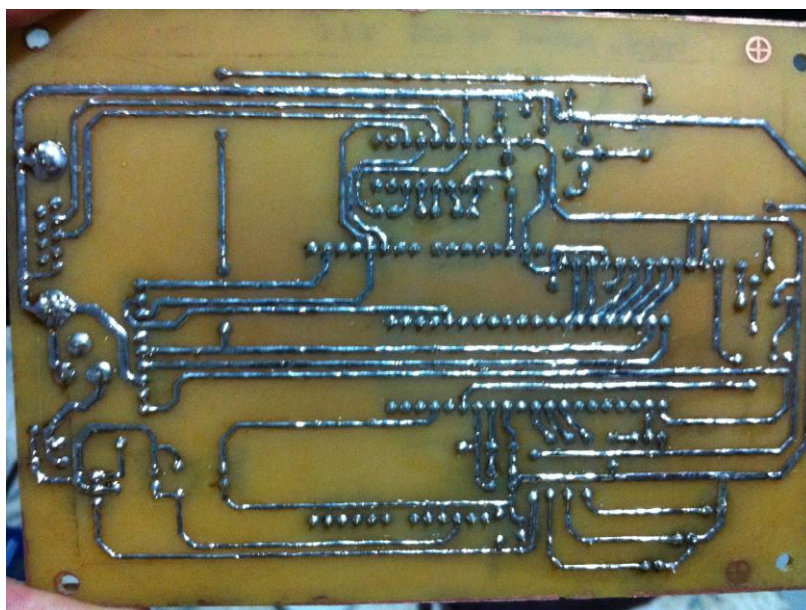
Com a placa preparada, todas as trilhas foram verificadas e corrigidas utilizando uma caneta para retroprojektor caso ocorra algumas falhas de ligação da trilha causada pelo processo de passar a imagem do papel para a placa. O fio de arame foi amarrado na placa e a placa mergulhada na solução de percloro (percloro de ferro e água). Após alguns minutos o percloro cumpriu o seu papel corroendo todo o cobre da placa exceto a parte coberta com o toner (a trilha). A placa foi lavada em água corrente para retirar a solução do percloro e secada com papel toalha.

Utilizando o mini furador conforme mostrado na Figura 2, todos os locais onde serão soldados os componentes do projeto foram furados.



**Figura 2 – Mini Furador (Fonte: Autor)**

Com a placa já furada, foram iniciadas as soldas dos componentes. A Figura 3 ilustra os componentes soldados na placa e na Figura 4 é mostrada a máscara da placa feita à mão utilizando a caneta para retroprojektor.



**Figura 3 – Componentes Soldados (Fonte: Autor)**

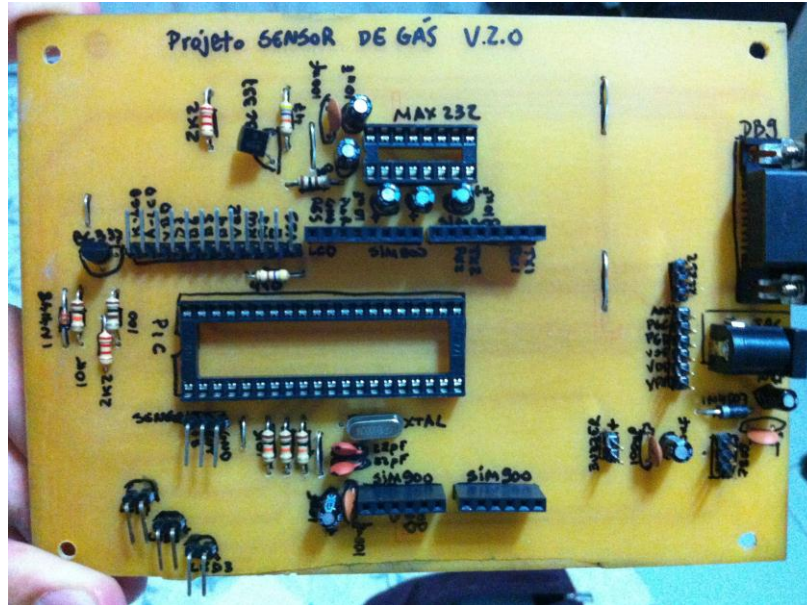


Figura 4 – Componentes da Placa e Máscara (Fonte: Autor)