

**CENTRO PAULA SOUZA
ETEC GILDO MARÇAL BEZERRA BRANDÃO**

**ABNER PEREIRA SILVA
GABRIEL CARDOSO DE PAULA
MARCOS GOMES CORREIA**

CURSO DE ELETRÔNICA

PROJETO CIE – CAIXA DE CORRESPONDÊNCIA INTELIGENTE ELETRÔNICA

**SÃO PAULO
2014**

ABNER PEREIRA SILVA Nº: 01
GABRIEL CARDOSO DE PAULA Nº: 11
MARCOS GOMES CORREIA Nº: 18

PROJETO CIE – CAIXA DE CORRESPONDÊNCIA INTELIGENTE ELETRÔNICA

MONOGRAFIA APRESENTADA COMO
EXIGÊNCIA PARA OBTENÇÃO DO
GRAU DE TECNOLOGIA EM
ELETRÔNICA DA ETEC GILDO MARÇAL
BEZERRA BRANDÃO.

ORIENTADOR: PROFESSOR RONALDO
OLIVEIRA

SÃO PAULO
2014

FOLHA DE APROVAÇÃO

ABNER PEREIRA SILVA
GABRIEL CARDOSO DE PAULA
MARCOS GOMES CORREIA

PROJETO CIE – CAIXA DE CORRESPONDÊNCIA INTELIGENTE ELETRÔNICA

Trabalho de Conclusão de Curso apresentado à ETEC Gildo
Marçal Bezerra Brandão como exigência parcial para
obtenção do título de Técnico em Eletrônica.

Aprovado em __/__/__

BANCA EXAMINADORA

Prof. Ronaldo Oliveira

Prof. Diogo Farias

Prof. Pablo Urbano

DEDICATÓRIAS

Dedicamos este projeto ao nossos familiares que sempre nos apoiaram a concluir este curso, a todos os educadores que estiveram com a gente nessa trajetória de aprendizado e a todos que nos ajudaram e apoiaram.

AGRADECIMENTOS

Agradecemos primeiramente a Deus pela força e sabedoria que tem nos dado, a todos os colegas de classe, ao Welligton José colega de trabalho do integrante Gabriel Cardoso pela sua contribuição técnica, ao Vello Charles Deak, familiar do integrante Gabriel Cardoso pelos conselhos e pela contribuição com as ferramentas, ao nosso orientador do curso e a todos os professores e colaboradores envolvidos direta ou indiretamente a nossa turma de formandos em Eletrônica.

EPIGRAFE

*“Teria maior confiança no desempenho
de um homem que espera ter uma
grande recompensa do que no daquele que já a recebeu.”*

Voltaire

RESUMO

Apesar de hoje em dia grande parte das pessoas receberem correspondências por e-mail, algumas coisas ainda são entregues pelos correios em residências e pensando nisso foi criado a CIE – Caixa de Correspondência Inteligente Eletrônica. Visando facilitar e integrar cada vez mais a vida de pessoas que frequentemente se encontram ausentes em suas casas com as tecnologias móveis, o projeto CIE - Correspondência Inteligente Eletrônica tem como proposta manter sempre o usuário atualizado sobre o recebimento cartas e correspondências que chegam em sua residência. Ela é basicamente uma Caixa de Correspondência Residencial que irá funcionar da seguinte maneira: Quando chegada uma carta na caixa de correspondência inteligente, será acionado um sensor que por sua vez acionará um circuito, este mandará a seguinte mensagem de texto para o aparelho celular do usuário via GSM: “Há carta disponível em sua Caixa de Correspondências, retire-a o mais breve possível.”. Após o mesmo receber e ler a mensagem SMS irá retirar a carta, tudo com segurança e privacidade. Com isso o usuário terá com o controle das suas correspondências evitando assim, possíveis furtos ou enganos.

Palavras-chave: Caixa de Correspondência, Correspondência, GSM, SMS.

ABSTRACT

Today, although most people receive their correspondences by e-mail, there are still many things that could only be sent into homes in general and with this in mind that this project was created CIE – Caixa de Correspondência Inteligente Eletrônica.

Aiming to facilitate and integrate people's lives who often find themselves absentee in their homes making use of the mobile technology increasingly, the Project CIE - Correspondencia Inteligente Electronica has the goal of making the user always updated concerning the letters and correspondences that get into their homes. It is basically a Residential MailBox that will work as follows: when a letter is received in the "Intelligent Correspondence", a sensor will be activated. The former will activate a circuit that will send the following SMS message to the user's cell phone by GSM: "There is a letter available in your MailBox and you have to take it out as soon as possible." After the user receive and read the text message, he will take out the letter safely and privacy. Therefore, the user will have the control of his correspondences by avoiding a possible theft or an error.

Key-words: MailBox, Correspondence, GSM, SMS.

LISTA DE FIGURAS

- Fig. 01 - Os primeiros telefones GSM (1991) Pg. 17
- Fig. 02 – Diagrama em blocos do projeto Pg. 19
- Fig. 03 – Esquema elétrico do Display LCD no ISIS Pg. 20
- Fig. 04 – Esquema elétrico da MCU Pg. 21
- Fig. 05 – Esquema elétrico do teclado Pg. 22
- Fig. 06 Esquema elétrico do Modulo GSM Pg. 23
- Fig. 07 - Arduino Shield GSM GPRS SIM900 Pg. 26
- Fig. 08 – Display LCD LM016L 16x2 Pg. 27
- Fig. 09 – Teclado numérico com push buttons Pg. 28
- Fig.10 – Fluxograma Projeto CIE Pg. 30
- Fig. 11 – Simulador de Periféricos USART (USART Terminal) Pg. 32
- Fig. 12 – Simulação do projeto no software ISIS, Proteus 7 Pg. 34
- Fig. 13 – Keypad-Phone (componente do ISIS, Proteus 7) Pg. 35
- Fig. 14 - Kit de Desenvolvimento MultiPIC para PICS Pg. 38
- Fig. 15 - Gravador de PIC Multiprog da Microchip Pg. 39
- Fig. 16 – Vista superior da estrutura da caixa de correspondência Pg. 63
- Fig. 17 – Vista traseira da estrutura da caixa de correspondência Pg. 63

LISTA DE TABELAS

Tab. 01 – Tab.01 – Mapa de portas Pg. 29

Tab. 02 – Tab. 02 – Lista de materiais Pg. 31

Tab. 03 – Tab. 03 – Especificações do produto Pg. 40

LISTA DE ABREVIATURAS E SIGLAS

PIC – Programmable Interface Controller (Controlador de Interface Programável)

GSM – Global System for Mobile Communications (Sistema Global para Comunicações Móveis)

SMS - Short Message Service (Serviço de Mensagens Curtas)

2G – 2rd Generation (2ª Geração)

3G – 3rd Generation (3ª Geração)

3GPP – 3rd Generation Partnership Project (

GPRS – General packet radio servisse (*Serviço de Rádio de Pacote Geral*)

EDGE – Enhanced Data Rates *For GSM Evolution* (*Taxas de Dados Ampliadas para a Evolução do GSM*)

LCD - Liquid Crystal Display (Display de Cristal Líquido)

MCU – Microcontroller Unit (Unidade Microcontrolada)

EEPROM – Electrically-Erasable Programmable Read-Only (Memória Somente de Leitura Programável Apagável Eletricamente).

SIM – Subscriber Identity Module (Módulo de Identificação do Assinante)

SUMÁRIO

1. INTRODUÇÃO	14
2. OBJETIVOS.....	14
2.1. Objetivos gerais	14
2.2. Objetivos específicos	15
3. FUNDAMENTAÇÃO TEÓRICA	16
4. TECNOLOGIA	17
4.1. Tecnologia GSM.....	17
4.1.1. Como funciona	18
5. DESENVOLVIMENTO	20
5.1. Diagrama em Blocos.....	20
5.2. Esquema Elétrico	20
5.2.1. Esquema do display	21
5.2.2. Esquema do MCU.....	22
5.2.3. Esquema sensores óticos	23
5.2.4. Esquema modulo GSM	24
5.3. Principais Componentes	24
5.3.1. Microcontrolador PIC16F877A.....	24
5.3.1.1. Características básicas	25
5.3.2. Modulo GSM	27
5.3.3. Sensor ótico	27
5.3.4. Display LCD	28
5.3.5. Keypad (Teclado)	29
5.4. Mapa de portas/pinos	30
5.5. Fluxograma.....	31
5.6. Lista de Materiais	32

6. TESTES E SIMULAÇÕES	33
6.1. Etapas de testes	33
6.1.1. Testes com Modulo GSM.....	33
6.1.2. Ligação elétrica	34
6.1.3. Simulação virtual	35
6.2. Ferramentas utilizadas	38
6.2.1. Proteus 7.....	38
6.2.2. MikroC Por for PIC	38
6.2.3. CCS Compiler	38
6.2.4. Kit de Desenvolvimento MultiPIC	39
6.2.5. Gravador de PIC Multiprog.....	40
7. ESPECIFICAÇÕES DO PRODUTO	41
8. PROBLEMAS E POSSÍVEIS SOLUÇÕES	42
9. CONCLUSÃO	43
10. REFERÊNCIAS.....	44

1. INTRODUÇÃO

Voltado para fins acadêmicos, o Projeto CIE propõe facilitar a vida de seu usuário conectando-o a sua caixa de correspondência residencial através do seu aparelho celular. Com isso ele(a) não perdera tempo olhando em sua caixa, pra saber se tem ou não tem cartas. Antigamente as pessoas não tinham controle de quantas correspondências que chegavam em sua caixa, facilitando a violação do mesmo. Este projeto e voltado para usuários que passam a maior parte do tempo fora da sua casa. Com a facilidade de receber faturas e documentos pelo e-mail

Projetado com sensores de presença, Display LCD, teclado numérico, sistema elétrico microcontrolado e com um modulo de alta tecnologia GSM.

2. OBJETIVOS

2.1 Objetivos Gerais

O Projeto CIE é uma caixa de correspondência residencial aparentemente comum, porem seu diferencial é que quando chegar alguma correspondência na caixa uma mensagem de texto será enviada para o numero de celular que o usuário.

Não importa onde esteja, no trabalho ou viajando com a família, o usuário sempre terá o controle de suas correspondências pelo celular.

2.2. Objetivos Específicos

Pensando em facilitar a utilização do produto para o usuário, a caixinha do Projeto CIE foi projetada com um Display LCD que exibi as informações do produto como quantidade de correspondências e configuração do numero de celular em sua parte de trás, um teclado numérico com função de Backspace (apagar caractere anterior) e Enter (confirma a digitação do numero) ao lado do Display e uma chave de fácil acesso para inserir o numero de celular do usuário no lado esquerdo próximo do teclado.

A abertura da entrada de cartas é um pouco maior do que caixas de correspondências comuns para evitar enroscamentos e sua antena de recepção de sinal GSM fica exposta ao lado externo para melhor captação de sinal.

3. FUNDAMENTAÇÃO TEÓRICA

3.1. Origem das correspondências

Correio é um sistema de comunicação que envolve o envio de documentos (cartas, faturas) e encomendas entre um remetente e um destinatário, que podem estar numa mesma cidade ou em lugares muito distantes entre si.

A princípio, o serviço postal pode ser privado ou público. Governos podem instituir restrições para que empresas privadas assumam o sistema postal.

Desde a metade do século XIX, o sistema postal nacional passou geralmente a ser estabelecido por monopólios governamentais através de um papel pré-pago, que era na forma de estampas adesivas, os selos. Em geral, os monopólios governamentais apenas entregavam as encomendas para prestadoras de serviços, e estas responsáveis pela entrega da encomenda até o endereço correto. Comunicação através de documentos escritos carregados por um intermediário até outro lugar certamente data-se anteriormente a invenção da escrita. Contudo, o desenvolvimento formal do sistema postal ocorreu muito depois.

4. TECNOLOGIA

4.1. Tecnologia GSM

Global System for Mobile Communications, ou Sistema Global para Comunicações Móveis (GSM: originalmente, *Groupe Special Mobile*) é uma tecnologia móvel e o padrão mais popular para telefones celulares do mundo. Telefones GSM são usados por mais de um bilhão de pessoas em mais de 200 países. A onipresença do sistema GSM faz com que o *roaming* internacional seja muito comum através de "acordos de *roaming*" entre operadoras de telefonia móvel. O GSM diferencia-se muito de seus antecessores sendo que o sinal e os canais de voz são digitais, o que significa que o GSM é visto como um sistema de telefone celular de *segunda geração* (2G). Este fato também significa que a comunicação de dados foi acoplada ao sistema logo no início. O GSM é um padrão aberto desenvolvido pela 3GPP.

O GSM possui uma série de características que o distinguem dentro do universo das comunicações móveis. Nascido nos anos 80 e fruto de uma cooperação sem precedentes dentro da Europa¹, o sistema partilha elementos comuns com outras tecnologias utilizadas em telemóveis, como a transmissão ser feita de forma digital e a utilizar células (como funciona um telemóvel). Este artigo irá apresentar as características fundamentais do sistema, assim como as suas capacidades.

Do ponto de vista do consumidor, a vantagem-chave do GSM são os serviços novos com baixos custos. Por exemplo, a troca de mensagens de texto foi originalmente desenvolvida para o GSM. A vantagem para as operadoras tem sido o baixo custo de infra-estrutura causada por competição aberta. A principal desvantagem é que o sistema GSM é baseado na rede TDMA, que é considerada menos avançada que a concorrente CDMA. A performance dos celulares é muito similar, mas apesar disso o sistema GSM tem mantido compatibilidade com os telefones GSM originais. No mesmo tempo, o sistema GSM continua a desenvolver-se com o lançamento do sistema GPRS. Além disso, a transmissão de dados em alta velocidade foi adicionada no novo esquema de modulação EDGE. A versão de 1999 do padrão introduziu índices relativamente altos de transmissão de dados, e é normalmente referida como 3G.

4.1.1. Como funciona

O sistema GSM 900 utiliza dois conjuntos de frequências na banda dos 900 MHz: o primeiro nos 890-915MHz, utilizado para as transmissões do terminal, e o segundo nos 935-960MHz, para as transmissões da rede.

O método utilizado pelo GSM para gerir as frequências é uma combinação de duas tecnologias: o TDMA (*Time Division Multiple Access*) e o FDMA (*Frequency Division Multiple Access*). O FDMA divide os 25 MHz disponíveis de frequência em 124 canais com uma largura de 200 kHz e uma capacidade de transmissão de dados na ordem dos 270 Kbps. Uma ou mais destas frequências é atribuída a cada estação-base e dividida novamente, em termos de tempo, utilizando o TDMA, em oito espaços de tempo (*timeslots*). O terminal utiliza um *timeslot* para recepção e outro para emissão. Eles encontram-se separados temporalmente para que o telemóvel não se encontre a receber e transmitir ao mesmo tempo. Esta divisão de tempo também é chamada de *full rate*. As redes também podem dividir as frequências em 16 espaços, processo designado como *half-rate*, mas a qualidade da transmissão é inferior.



Fig. 01 - Os primeiros telefones GSM (1991)

A voz é codificada de uma forma complexa, de forma que erros na transmissão possam ser detectados e corrigidos. Em seguida, a codificação digital da voz é

enviada nos *timeslots*, cada um com uma duração de 577 milisegundos e uma capacidade de 116 bits codificados. Cada terminal deve possuir uma agilidade de frequência, podendo deslocar-se entre os *timeslots* utilizados para envio, recepção e controle dentro de um *frame* completo. Ao mesmo tempo, um telemóvel verifica outros canais para determinar se o sinal é mais forte e mandar a transmissão para eles, caso a resposta seja afirmativa.

5. DESENVOLVIMENTO

5.1. Diagrama em Blocos

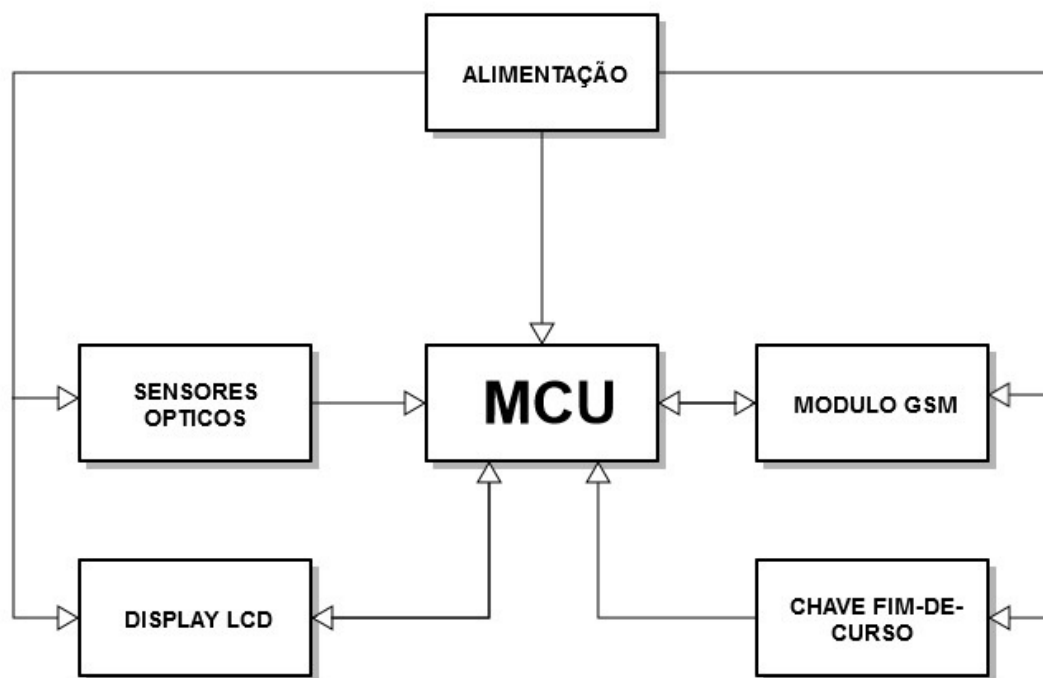


Fig. 02 – Diagrama em blocos do projeto

5.2 Esquema Elétrico

O Projeto CIE é um aparelho eletrônico com um sistema microcontrolado. No desenvolvimento do seu esquema elétrico, para facilitar foi dividido em 4 partes (esquema completo consulte Figura ??) descritas nos capítulos a seguir.

5.2.1. Esquema do display

O Display LCD que foi utilizado é um modelo LM016L, nele será exibido as informações de quantidade de correspondências e digitação do numero de celular. Segue abaixo o esquema elétrico do Display LCD:

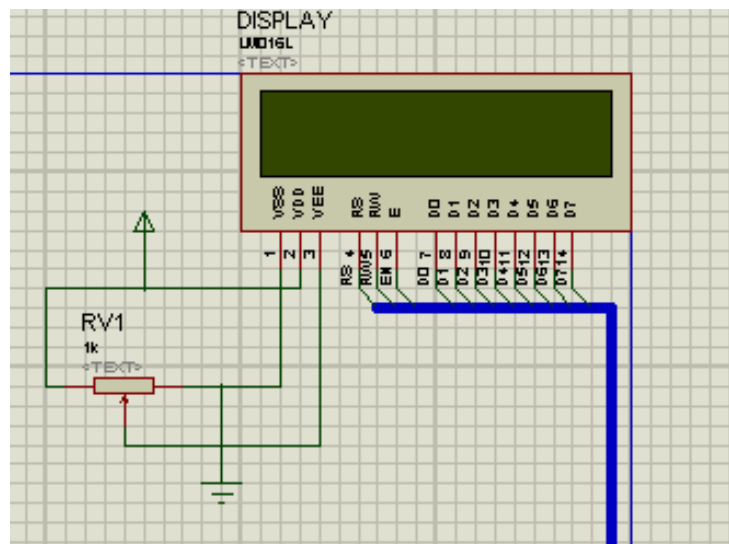


Fig. 03 – Esquema elétrico do Display LCD no ISIS

Os pinos D0 ate o D7 são os pinos de barramentos de dados onde são transmitidos os bits. Estes pinos estão conectados no Port D do microcontrolador (melhores detalhes da pinagem consulte o Mapa de portas, capítulo 5.4).

Temos o pino de clock E (Enable) habilitado, o R/W que é o pino de seleção de leitura (nível logico 1) ou escrita (nível logico 0) que esta selecionado como escrita, RS é o pino de seleção de dados (nível logico 1) ou instruções (nível logico 0) habilitado como dados e temos os pinos de contraste VSS, VDD e VEE. Utiliza-se um potenciômetro para fazer o contraste do LCD.

5.2.2. Esquema do MCU

Como CPU do projeto foi usado o microcontrolador PIC16F877A que contem 40 pinos sendo 33 I/O. Nele esta todo a programação logica (código fonte da programação consulte apêndices).

Abaixo esta o esquema elétrico do PIC:

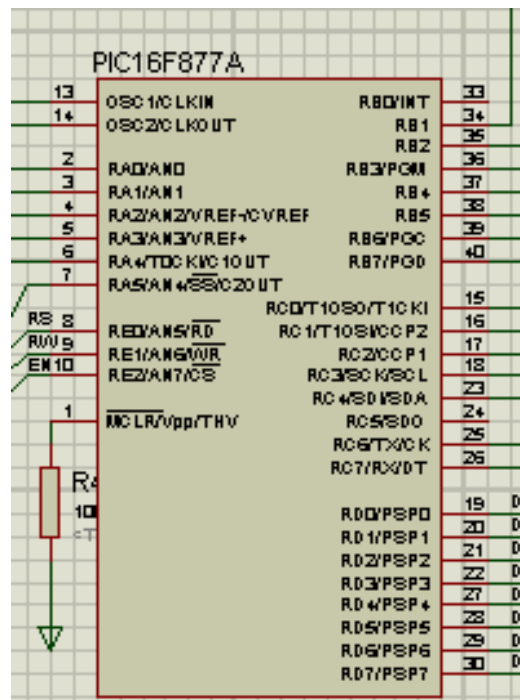


Fig. 04 – Esquema elétrico da MCU

5.2.3. Esquema sensores óticos

Como no Proteus não tem os componentes do sensor ótico para desenvolver o esquema elétricos utilizamos push buttons.

São 3 pares de sensores óticos (emissor com IRLED e receptor com Fototransistor) que detectam as cartas. Os resistores conectados neles são para pull up.

Neste esquema também estão a chave fim-de-curso para reset e a chave de configuração do numero de celular.

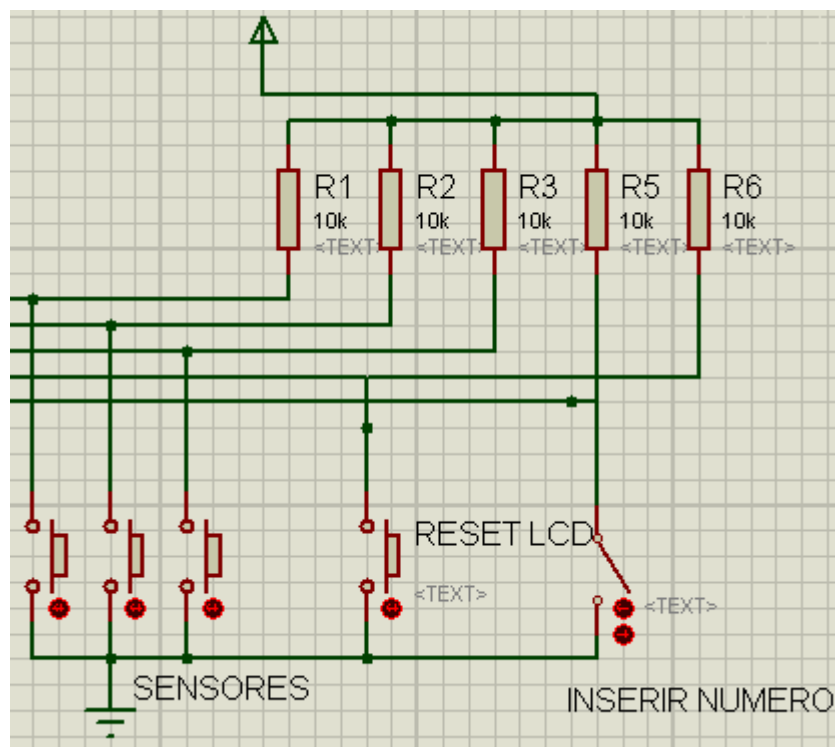


Fig. 05 – Esquema elétrico do teclado

5.2.4. Esquema modulo GSM

Apesar do modulo GSM ter varias funções, para o projeto foi necessário apenas a de envio de mensagem de texto.

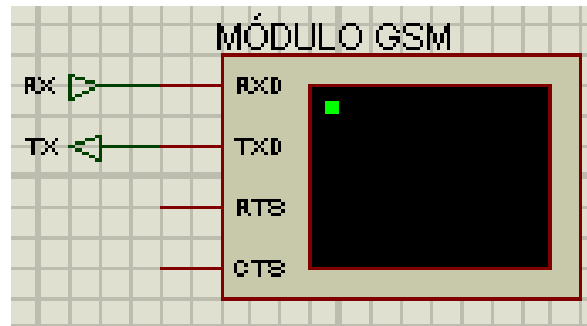


Fig. 06 Esquema elétrico do Modulo GSM

Para a função desejada, foi utilizado apenas os pinos de comunicação serial TX, RX e os de alimentação.

5.3 Principais Componentes

Cada componente deste projeto foi escolhido visando facilitar ao máximo sua elaboração.

Apesar do projeto ter vários componentes, o principal deles é o Modulo GSM, pois o grande diferencial do projeto CIE é o fato de que o usuário sempre receba mensagens de textos SMS onde quer que esteja.

5.3.1 Microcontrolador PIC16F877A

A MCU do projeto é o microcontrolador PIC16F877A da família 16F da empresa Microchip.

Devido a grande quantidade de componentes que seria conectados a nossa MCU, necessitaríamos de um microcontrolador com muitos pinos, por este motivo utilizamos o PIC16F877A que possui 40 pinos sendo 33 deles I/O.

5.3.1.1. Características básicas

- Alta - Desempenho RISC CPU:
- Apenas 35 instruções de uma única palavra para aprender
- Todas as instruções de ciclo único , exceto para os programas ramificados , que são de dois ciclos
- Velocidade de operação: DC - entrada de clock 20 MHz
DC - 200 ns ciclo de instrução
- Até 8K x 14 palavras de flash de memória do programa ,
Até 368 x 8 bytes de memória de dados (RAM) ,
Até 256 x 8 bytes de memória de dados EEPROM
- Pinagem compatível com outros 28 pinos ou 40/44 pinos
PIC16CXXX e PIC16FXXX microcontroladores

Características dos periféricos:

- Timer0 : 8-bit temporizador / contagem com 8 bits prescaler
- Timer1 : 16-bit temporizador / divisor de contagem ,
pode ser aumentado durante o sono via cristal / relógio externa
- Timer2 : 8-bit do temporizador / contador com o período de 8-bit
registrar, prescaler e postscaler
- Dois captura , comparar módulos PWM
 - Capture é de 16 bits , máx. resolução é de 12,5 ns
 - Comparar é de 16 bits , máx. resolução é de 200 ns
 - PWM max . resolução é de 10 bits
- Porta Serial Sincrona (SSP) com SPI TM
(Modo Mestre) e I2C TM (Mestre / Escravo)
- Transmissor e Receptor Universal Sincrono e Assíncrono (USART / SCI) com
detecção de endereço com 9 bits
- Porta Paralela Escrava (PSP) - 8 bits de largura, com
controles externos RD, WR e CS (apenas 40/44 pinos)
- Brown -out circuito de detecção para
Brown -out Reset (BOR)

CMOS Tecnologia:

- Baixo consumo de energia , alta velocidade Flash / EEPROM tecnologia
- Design totalmente estática
- faixa de tensão de funcionamento larga (2.0V a 5.5V)
- faixas de temperatura Comerciais e Industriais
- Baixo consumo de energia

5.3.2. Modulo GSM

O componente utilizado para realizar o envio de mensagens de texto do projeto é o Arduino Shield GSM GPRS SIM900. Com ele é possível executar funções como ligações telefônicas, envio de mensagens de texto e mais algumas outras.

Por este Modulo GSM ter periféricos de comunicação fáceis de se achar (Internet e Livros) o utilizamos.



Fig. 07 - Arduino Shield GSM GPRS SIM900

5.3.3. Sensor ótico

No projeto foi necessário um sensor que pudesse identificar uma carta quando entregue, por isto foi usado um sensor ótico onde um feixe de luz será inibido quando a carta entrar na caixa de correspondência.

Devido ao tamanho padrão de uma carta de correspondência, nenhum sensor ótico com baixo custo atenderia as necessidades do projeto. Por este motivo foi utilizado como sensor ótico um IRLED como transmissor que ficaria de frente com um Fototransistor como receptor, este que é o mesmo esquema de uma Chave Ótica, porem com um campo de sensoriamento bem melhor.

Assim quando a carta de correspondência cortar o feixe de luz do IRLED o Fototransistor mandara um sinal para o microcontrolador.

5.3.4. Display LCD

Foi utilizado no projeto um Display LCD alfanumérico 16x2 devido a quantidade de informações que seria preciso ser exibidas, como era apenas a mensagem “Você possui mensagens = 0” este atendeu perfeitamente.

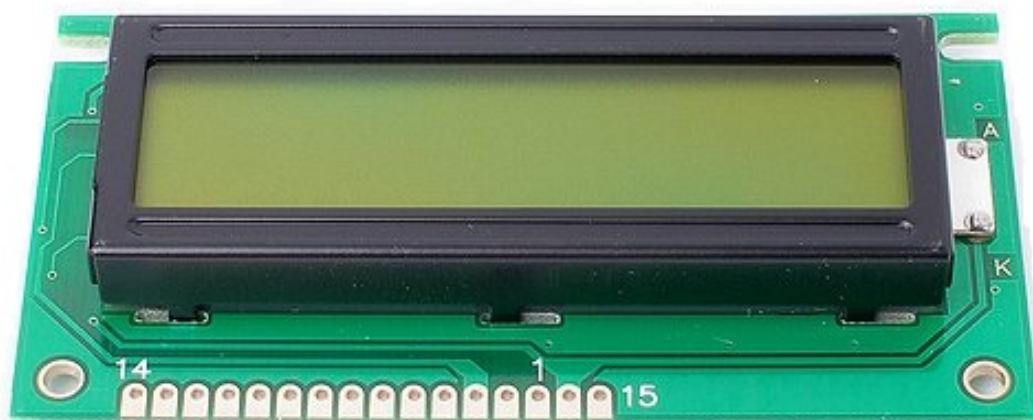


Fig. 08 – Display LCD LM016L 16x2

5.3.5. Keypad (Teclado)

A keypad será utilizada no projeto para a inserção do numero de celular desejado pelo usuário a receber as mensagens de texto.

Para a keypad foram usados 12 Push Button representando cada algarismo do teclado(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, Enter e Backspace).

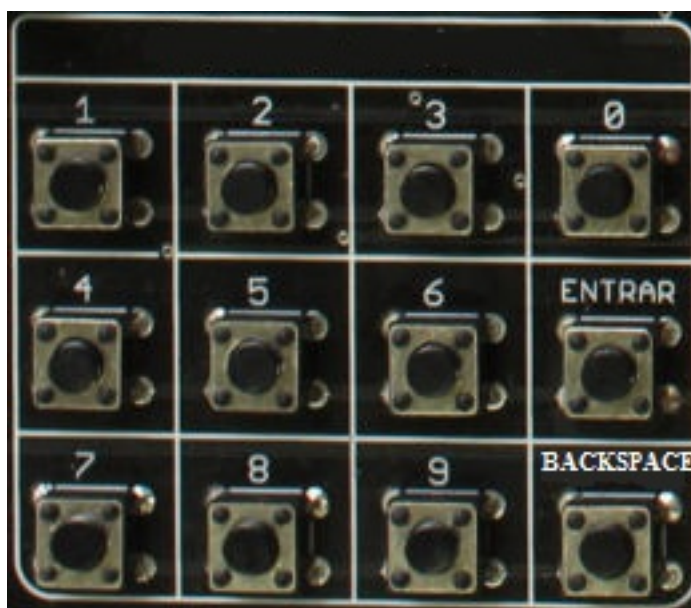


Fig. 09 – Teclado numérico com push buttons

5.4. Mapa de portas/pinos

PIC16F877A		
PINO	DESCRIÇÃO	FUNÇÃO
1	MCLR/VPP	CLEAR
2	RA0/AN0	NADA
3	RA1/AN1	NADA
4	RA2/AN2/VREF-/CVREF	NADA
5	RA3/AN3/VREF+	NADA
6	RA4/T0CKI/C1OUT	NADA
7	RA5/AN4/SS/C2OUT	NADA
8	RE0/RD/AN5	RS DO DISPLAY
9	RE1/WR/AN6	RW DO DISPLAY
10	RE2/CS/AN7	EM DO DISPLAY
11	VDD	VDD
12	VSS	VSS
13	OSC1/CLKI	CRYSTAL
14	OSC2/CLKO	CRYSTAL
15	RC0/T1OSO/T1CKI	ENTRADA DO SENSOR
16	RC1/T1OSI/CCP2	ENTRADA DO SENSOR
17	RC2/CCP1	ENTRADA DO SENSOR
18	RC3/SCK/SCL	RESER
19	RD0/PSP0	D0 DO DISPLAY
20	RD1/PSP1	D1 DO DISPLAY
21	RD2/PSP2	D2 DO DISPLAY
22	RD3/PSP3	D3 DO DISPLAY
23	RC4/SDI/DAS	CONFIGURAÇÃO
24	RC5/SDO	NADA
25	RC6/TX/CK	TX DO MODULO GSM
26	RC7/RX/DT	RX DO MODULO GSM
27	RD4/PSP4	D4 DO DISPLAY
28	RD5/PSP5	D5 DO DISPLAY
29	RD6/PSP6	D6 DO DISPLAY
30	RD7/PSP7	D7 DO DISPLAY
31	VSS	VSS
32	VDD	VDD
33	RB0/INT	NADA
34	RB1	TECLADO
35	RB2	TECLADO
36	RB3/PGM	TECLADO
37	RB4	TECLADO
38	RB5	TECLADO
39	RB6/PGC	TECLADO
40	RB7/PGD	TECLADO

Tab.01 – Mapa de portas

5.5. Fluxograma

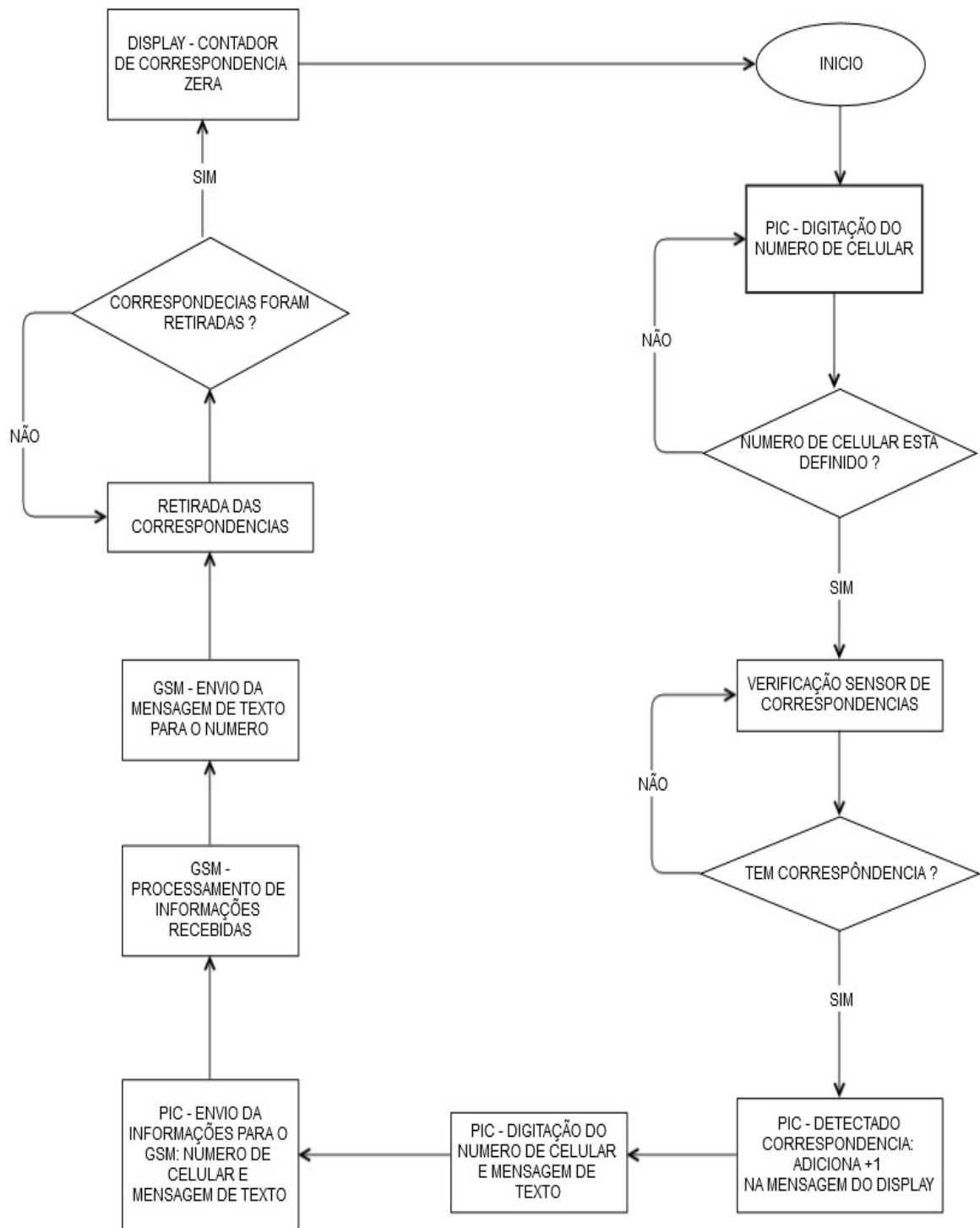


Fig.10 – Fluxograma Projeto CIE

5.6. Lista de Materiais

PRODUTO(S) COMPRADO(S)	QUANT.	VALOR UNIT.	VALOR TOTAL
Arduino Shield Gsm Gprs Sim900 + Antena	1	R\$ 251,40	R\$ 251,40
IRLED	3	R\$ 2,60	R\$ 7,80
Fototransistor	3	R\$ 2,60	R\$ 7,80
Resistor de 10kΩ	12	DOAÇÃO	-
Capacitor de 22pF	2	DOAÇÃO	-
PIC16F877A	1	R\$ 13,20	R\$ 13,20
Soquete p/ PIC16F877	1	R\$ 2,20	R\$ 2,20
LED alto brilho	1	R\$ 1,00	R\$ 1,00
Trimpot de 10kΩ	1	R\$ 1,00	R\$ 1,00
Placa de fenolite 10x10	2	R\$ 0,50	R\$ 1,00
Placa de fenolite 20x30	1	R\$ 1,20	R\$ 1,20
Fototransistor	4	R\$ 1,00	R\$ 4,00
Display LCD 16x2	1	R\$ 28,00	R\$ 28,00
Placa de Acrílico 1m x 70cm	1	R\$ 50,00	R\$ 50,00
Pacotes Cantoneiras c/6	4	R\$ 3,00	R\$ 12,00
Parafusos com porcas	6	R\$ 2,00	R\$ 12,00
Push Button	12	R\$ 0,15	R\$ 1,80
Flat Wire Up 1m	2	R\$ 3,00	R\$ 6,00
Crystal Oscilador 20MHz	1	R\$ 1,25	R\$ 1,25
Fonte de Alimentação	1	DOAÇÃO	-
Mão de obra	1	R\$ 400,00	R\$ 400,00
Total			R\$ 1.001,65

Tab. 02 – Lista de materiais

6. SIMULAÇÕES E TESTES

6.1. Etapas de testes

6.1.1 Testes com Modulo GSM

O Modulo GSM SIM 900 escolhido para o projeto vem em uma placa de Arduino onde é possível realizar diversas montagens e aplicações com o modulo, mais no projeto utilizamos somente os pinos de comunicação serial TX e RX.

O software de programação em Linguagem C MikroC Pro for PIC é um compilador que disponibiliza diversas ferramentas de simulação de programas como Editor de memoria EEPROM, Simulador de Display 7 segmentos, entre outros. Porem o que merece destaque é o Simulador de Periféricos USART, o USART Terminal.

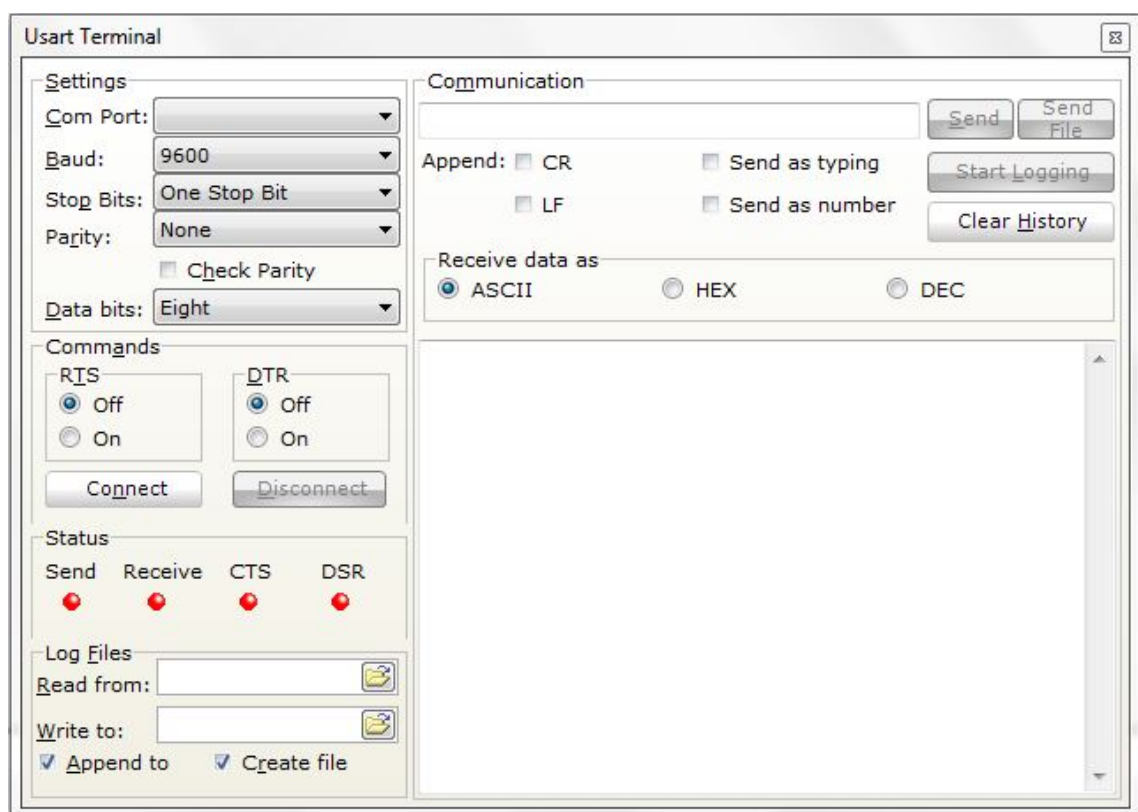


Fig. 11 – Simulador de Periféricos USART (USART Terminal)

Com ele foi feito os testes de comunicação com o Modulo GSM no Computador descritos nos tópicos a seguir:

6.1.2. Ligação elétrica

Utilizando a keypad e as saída USB de comunicação do Kit de desenvolvimento MultiPIC com o computador e alguns componentes no protoboard foi montado o circuito para testes. Para fazer a comunicação serial do PIC com o Modulo foi usado o MAX232, um componente de comunicação serial.

Através do USART Terminal, inserindo a programação pesquisada e desenvolvida, foi possível de inicio fazer uma ligação telefônica com o Modulo, mais logo foi desenvolvido uma programação para o envio de mensagens de texto pelo Modulo GSM.

Concluído os testes utilizando o computador, foi feito uma programação especifica para o microcontrolador PIC onde foi executada com sucesso.

6.1.3. Simulação virtual

O software Proteus foi o utilizado no projeto para a simulação do circuito elétrico.

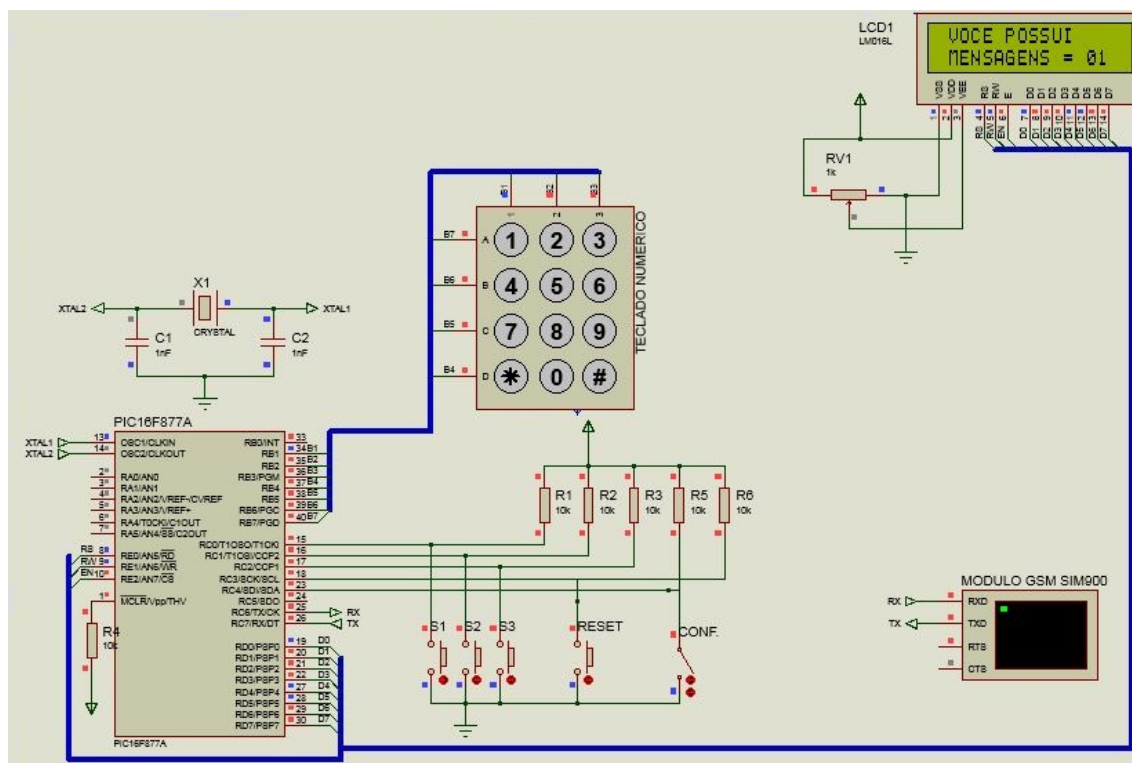


Fig. 12 – Simulação do projeto no software ISIS, Proteus 7

Os botões S1, S2 e S3 serão utilizados como nosso sensor ótico pois não há como simular o sensor do projeto.

Apertando qualquer um dos três botões o microcontrolador recebera um sinal de tensão. Com isso o microcontrolador mandara os comandos AT para o Terminal Virtual (simulador do Modulo GSM) com o numero de telefone do usuário e texto da mensagem “Você possui 1 nova(s) correspondência(s)” e enviara o SMS.

Quando o SMS for enviado para o celular do usuário aparecera ao mesmo tempo uma mensagem no display informando se a correspondência e a quantidade “Você possui mensagens = 1”.

O botão RESET esta simulando uma chave fim de curso que será instalada na porta da caixa de correspondência. Quando acionada, ela mandara um sinal de tensão pro microcontrolador que iniciara o comando que limpava todos as mensagens enviadas do display LCD e a programação do PIC zerando tudo.

O botão CONF. é o botão de configuração para o Modulo. Pressionando ele será acionado a programação de edição do numero de celular em que será enviado as mensagens de texto.

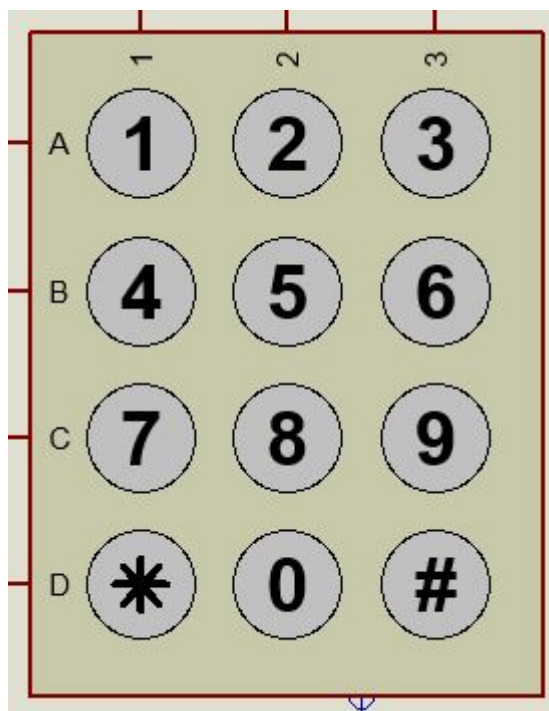


Fig. 13 – Keypad-Phone (componente do ISIS, Proteus 7)

Utilizando o teclado para inserir os numero, serão possível digitar 9 dígitos, (que corresponde ao numero de dígitos de um numero de celular de São Paulo) o botão do teclado * (estrela) servira como Backspace (apagar digito anterior) para corrigir se algum numero for digitado errado e o botão # (jogo da velha) servira como botão Enter para confirmar o numero de telefone.

Logo após uma mensagem de confirmação aparecer a “OK”, o botão CONF devera ser pressionado novamente e assim o microcontrolador gravara o novo numero na programação.

Os resistores R1,R2,R3,R4,R5 e R6 são os resistores de pull p, garantindo um nível alto de tensão quando os botões correspondentes a cada resistor for pressionado.

Temos então o circuito de clock externo do microcontrolador PIC com um Crystal (X1) e os Capacitores (C1 e C2).

Concluído os testes de simulação e no protoboard, foi desenvolvido a primeira placa eletrônica do projeto.

O layout da placa foi desenvolvido no ARES, uma ferramenta do Proteus para desenvolver layouts PCB para placas eletrônicas.

Após a confecção do layout e logo impresso na placa, foi iniciado os testes de todos os circuitos.

Com isso foi concluído todos os testes.

6.2 Ferramentas utilizadas

6.2.1. Proteus 7

O Proteus é um software de simulação virtual de circuitos eletrônicos.

Este software conta com uma grande variedade de componentes e ferramentas para o desenvolvimento de circuitos elétricos. No Projeto CIE foram utilizados as ferramentas ISIS para a simulação dos circuitos eletrônicos e o ARES para o desenvolvimento do layout PCB para ser impresso na placa de Fenolite.

6.2.2. MikroC Por for PIC

O MikroC é um software compilador de códigos em Linguagem C. Esta que é utilizada para microcontroladores da linha PIC da empresa Microchip.

Este software foi utilizados apenas para realizar os testes com o Modulo GSM através da ferramenta USART Terminal para simulações de comunicação serial TX e RX com o PIC.

6.2.3 CCS Compiler

Assim como o MikroC, o CCS Compiler é um software compilador de códigos em Linguagem C. Este foi usado para desenvolver todo o código fonte do projeto CIE por acharmos mais simples de se programar.

6.2.4. Kit de Desenvolvimento MultiPIC

O Kit de Desenvolvimento MultiPIC da Microchip é um versátil sistema de aprendizado e desenvolvimento para microcontroladores PIC.

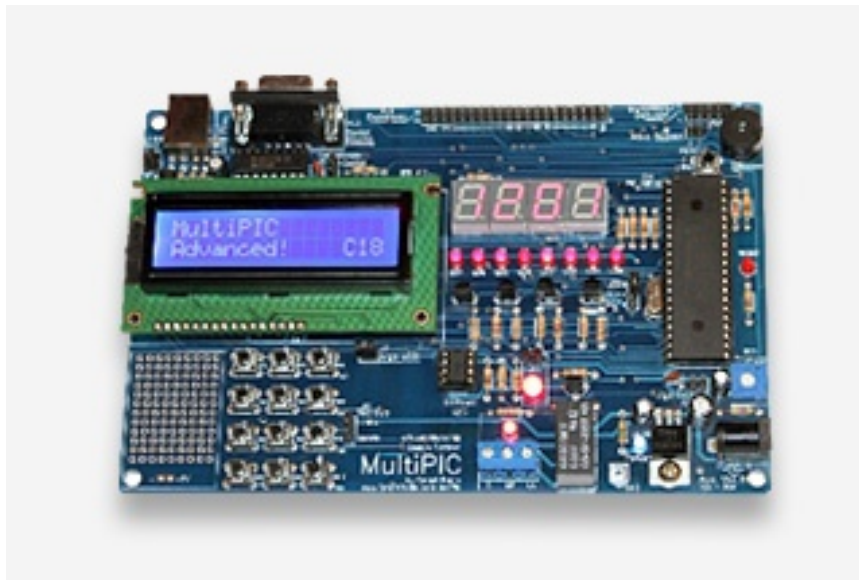


Fig. 14 - Kit de Desenvolvimento MultiPIC para PICS

Sua principal característica que levou a utiliza-lo no processo de desenvolvimento do projeto foi seu hardware de suporte à comunicação serial USB quando usados com PICs que tem este recurso. Por esta característica que esta Kit de Desenvolvimento foi utilizado para fazer os testes de comunicação do Modulo GSM.

6.2.5 Gravador de PIC Multiprog

Para a gravação da nossa MCU foi utilizado esse hardware de gravação de microcontroladores PIC das famílias 10F, 12F, 16F, 18F e 32F.



Fig.15 - Gravador de PIC Multiprog da Microchip

7. ESPECIFICAÇÕES DO PRODUTO

ESPECIFICAÇÕES DO PRODUTO	
Corrente	1,5A
Pico de Corrente	2A
Alimentação	5VDC
Fonte de Alimentação	127VAC
Ambiente Apropriado	Protegido de Sol e Chuva
Potencia	7,5W
Dimensões	Superior e Inferior = 30x20cm
	Lateral Direita e Esquerda = 12x20cm
	Frente = 14,35cm
	Traseira = 30x12cm
Peso	1,2kg
Material	Acrílico

Tab. 03 – Especificações do produto

8 PROBLEMAS E POSSÍVEIS SOLUÇÕES

O Display LCD foi instalado e posicionado na porta móvel de trás da caixa fazendo com que o flat do display se estique podendo rompê-lo. O adequado seria o Display ficar em uma parte fixa da caixa.

Por a Caixa de Correspondência ser feita com acrílico transparente, os sensores óticos responsáveis pela detecção de cartas sofrem muita influencia da luz externa podendo não acionar o sistema quando uma carta entrar. O adequado é que a Caixa seja totalmente opaca.

Poderia ser implementado uma trava elétrica por senha na Caixa para evitar que outras pessoas retirem as correspondências do usuário.

9. CONCLUSÃO

Com os conhecimentos adquiridos no decorrer do curso, concluímos nosso projeto com êxito alcançando todos os objetivos.

Apesar das possíveis implementações que poderia ser inseridas no projeto, ele ficou como o planejado no início.

Com a junção dos conhecimentos adquiridos em todas as disciplinas, fomos capacitados a desenvolver este projeto com uma boa qualidade técnica.

Foi uma ótima experiência desenvolvê-lo nos estimulando a pesquisar conteúdos que não fazem parte da grade do curso como a tecnologia GSM utilizada no projeto.

O Projeto CIE teve uma ótima aceitação na feira de tecnologia da ETEC e elogiada por muitos professores.

10. REFERÊNCIAS BIBLIOGRAFICAS

Historia da Correspondência. Disponível em:

<<http://historiadacorrespondencia.blogspot.com.br/>> Acesso em: 26 out. 2013

Historia do SMS. Disponível em:

<<http://www.mobilepronto.org/historia-sms.html>> Acesso em: 26 out. 2013

Historia Postal. Disponível em:

<<http://www.correios.com.br/sobreCorreios/empresa/historia/>> Acesso em: 26 out. 2013

Programação para Display LCD. Disponível em:

<http://mecatronicadegaragem.blogspot.com.br/2011/01/apostila-usando-lcd-16x2-nos.html> Acesso em: 5 abr. 2014

Tecnologia GSM. Disponível em: < <http://pt.wikipedia.org/wiki/GSM> > Acesso em: 06 abr. 2014

Tabela ASCII. Disponível em:

<<http://web.cs.mun.ca/~michael/c/ascii-table.html>> Acesso em: 2 mar. 2014

Programação em Linguagem C. Disponível em:

Livro - Microcontroladores Pic Programação Em C por Fabio Pereira, Editora Érica.
Consultado em: 8 mar. 2014

Gravar EEPROM interna. Disponível em:

<<http://www.mikroe.com/forum/viewtopic.php?p=83341>> Acesso em: 19 abr. 2014

APÊNDICES

APÊNDICE A. Cronograma do projeto de TCC

	Início	Duração
Escolha do grupo de TCC	16/ago/13	14
Escolha do projeto de TCC	30/ago/13	14
Dicas e orientações dos professores sobre o TCC	13/set/13	17
Diagrama de blocos	30/set/13	7
Paper do projeto	30/set/13	7
Desenvolvimento do Pré-Projeto	12/out/13	20
Entrega do Pré-TCC	01/nov/13	1
Compra da Caixa de Correspondencia	04/nov/13	4
Ensaio da apresentação do Pré-TCC	08/nov/13	1
Modelação da Caixa de Correspondencia	11/nov/13	10
Apresentação do Pré-TCC	22/nov/13	1
Entrega do Pré-TCC final e apresentação do mecanismo do projeto	29/nov/13	1
Feira de projetos TCC	30/nov/13	1
Apresentação dos TCCs de 2013	06/dez/13	1
Compra do modulo GSM	07/dez/13	1
Elaboração de esquema eletrico e do fluxograma	09/fev/14	6
Desenvolvimento da monografia	15/fev/14	6
Programação do PIC para o modulo GSM e montagem do esquema eletrico	16/fev/14	6
Desenvolvimento da monografia	22/fev/14	6
Inserir programação do PIC no modulo GSM e fazer testes	23/fev/14	6
Desenvolvimento da monografia	01/mar/14	6
Desenvolvimento do esquema do circuito GSM e adaptação do sensor	02/mar/14	6
Desenvolvimento da monografia	08/mar/14	6
Instalação do circuito GSM na Caixa de Correspondencia	09/mar/14	6
Desenvolvimento da monografia	15/mar/14	6
Testes iniciais do Projeto	16/mar/14	6
Desenvolvimento da monografia	22/mar/14	6
Posiveis reparos caso haja defeitos	23/mar/14	6
Finalização do Projeto e testes finais	30/mar/14	6
Acasos e imprevistos do Projeto	06/abr/14	6
Feira de projetos TCC 2014	24/mai/14	1
Apresentação do TCC na feira	30/mai/14	1

Correspondência Inteligente

Escola Técnica Estadual Gildo Marçal Bezerra Brandão.

- ✓ Abner Pereira Silva
- ✓ Gabriel Cardoso de Paula
- ✓ Marcos Gomes Correia

Resumo

O projeto Caixa de Correspondência Inteligente Eletrônica é uma caixa de correspondência residencial aparentemente normal, seu diferencial é que ela te avisa quando chegar qualquer carta por SMS. Não importa onde você esteja, no trabalho ou viajando com a família, você sempre terá o controle de suas correspondências pelo celular e seguro que somente você ou alguém de sua confiança terá acesso as suas correspondência.

Introdução

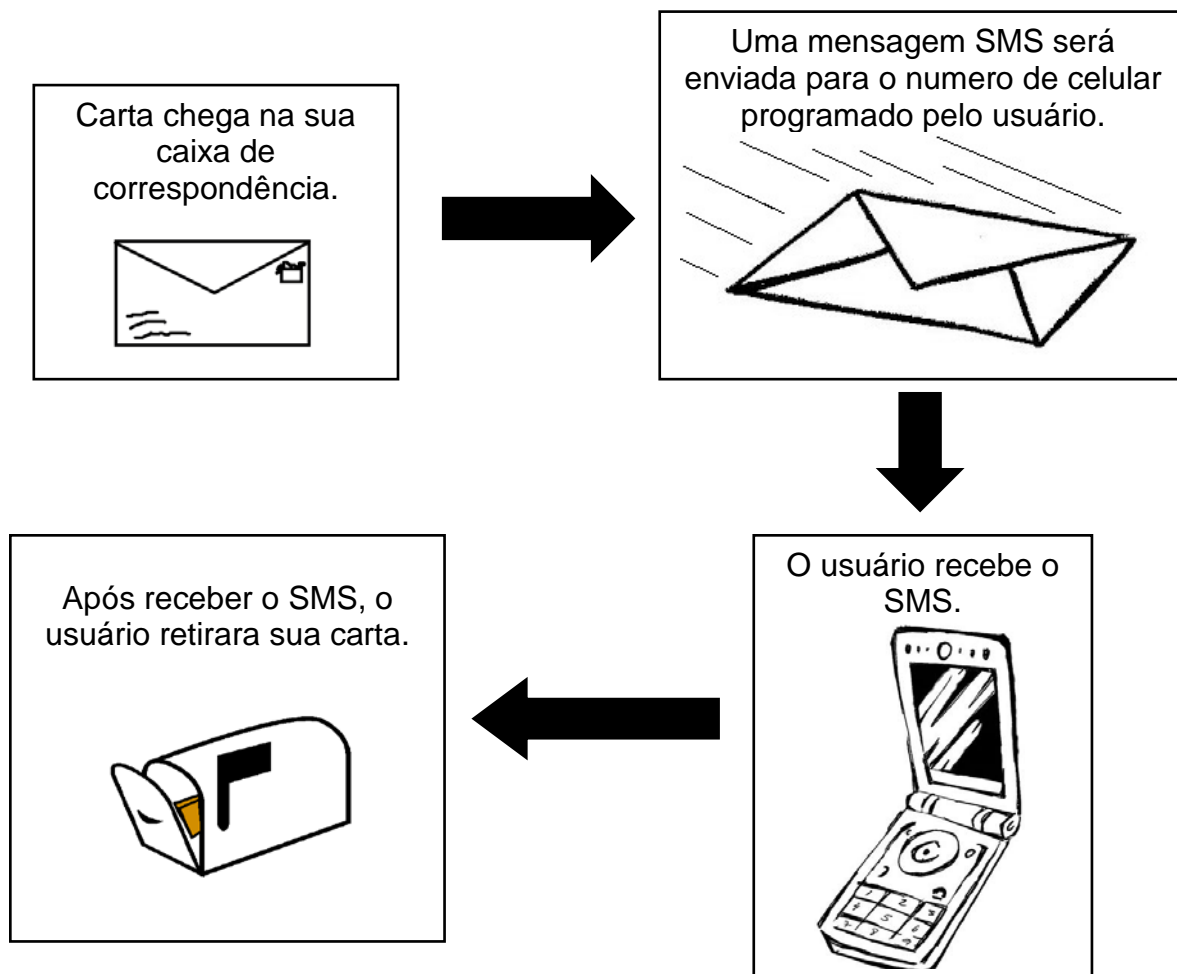
Hoje em dia andamos tão ocupados com trabalhos e estudos que mal paramos em nossas casas. Com isso não temos tempo para verificar nossas correspondências que aguardamos como faturas de cartão de credito, cartas de faculdades, multas de transito, etc. Pensando nisso, desenvolvemos uma caixa de correspondência inteligente para te avisar quando chegar algum em sua caixa de correspondência e te dar segurança e privacidade garantindo que apenas você ou alguém de sua confiança veja suas cartas.

Funcionamento

O processo de funcionamento do nosso projeto é simples e seguro. Para explicar o funcionamento suponhamos que o usuário esteja no trabalho. Quando o carteiro colocar a carta em sua caixa de correspondência, um sensor que se encontra na entrada da caixa será cortado acionando o circuito do GSM. Uma mensagem será escrita por uma CPU (Há uma correspondência disponível em sua caixa de correspondência, verifique-a assim que possível). Essa mensagem será enviada pelo circuito GSM como SMS para o usuário.

Quando o usuário receber o sms e for verificar sua correspondência, terá um painel com senha numérica que apenas o usuário terá conhecimento na caixa de correspondência, com a senha ele poderá destravar e retirar sua/suas carta/cartas da caixa.

Veja o esquema do processo de funcionamento da caixa de correspondência inteligente.



Aplicação

A Caixa de Correspondência Inteligente pode ser aplicada em diversos casos a partir da necessidade do usuário como em casa residências onde os donos ficam muito tempo ausentes ou para usuários que necessitam ter um controle de correspondência que entram e saem de um local(casa residenciais, empresas, etc.).

Referencias Bibliográficas.

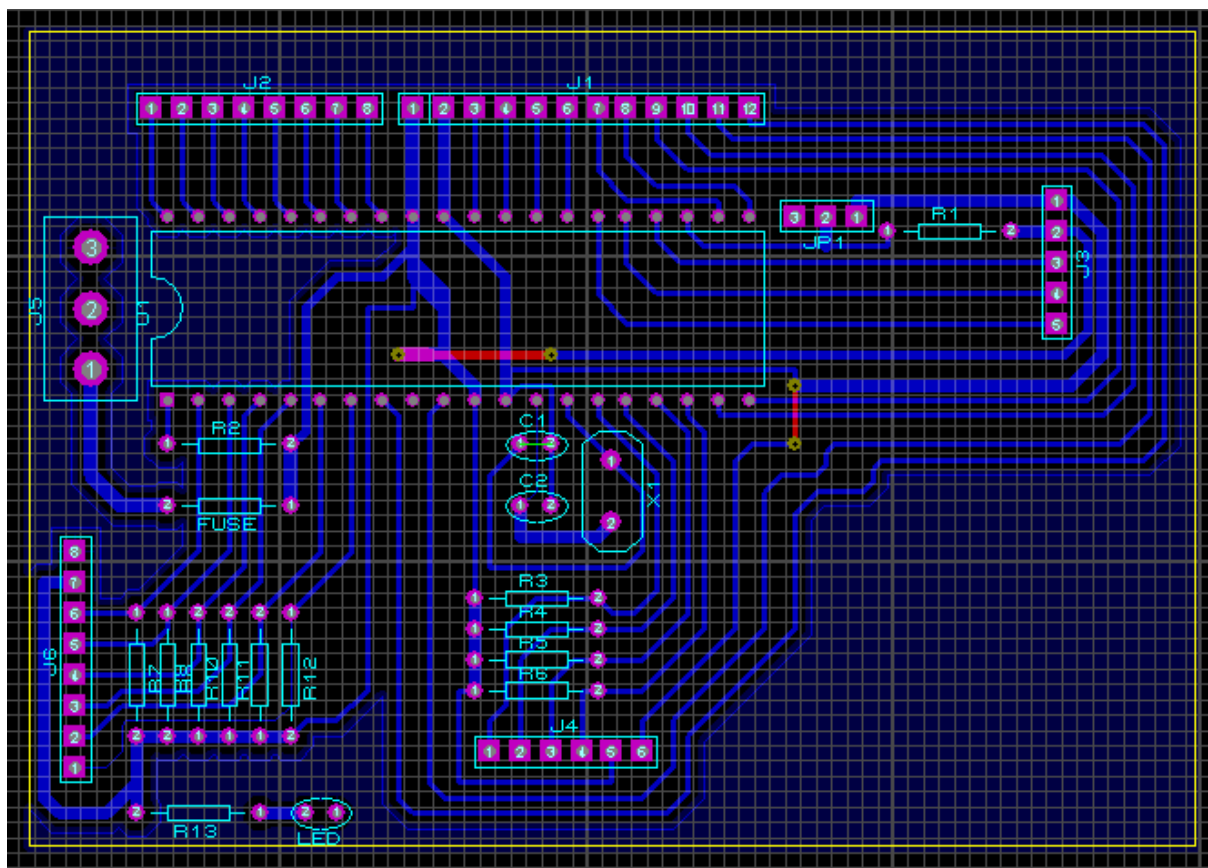
Transferência de SMS por GSM:

http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/ricardo/2_2.html

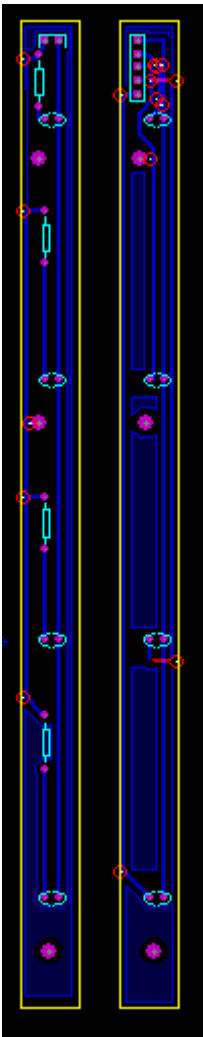
APÊNDICE C. LAYOUTS DAS PLACAS

Todos os layouts foram desenvolvidos no ARES, uma ferramenta de desenvolvimento de layouts PCB do software Proteus.

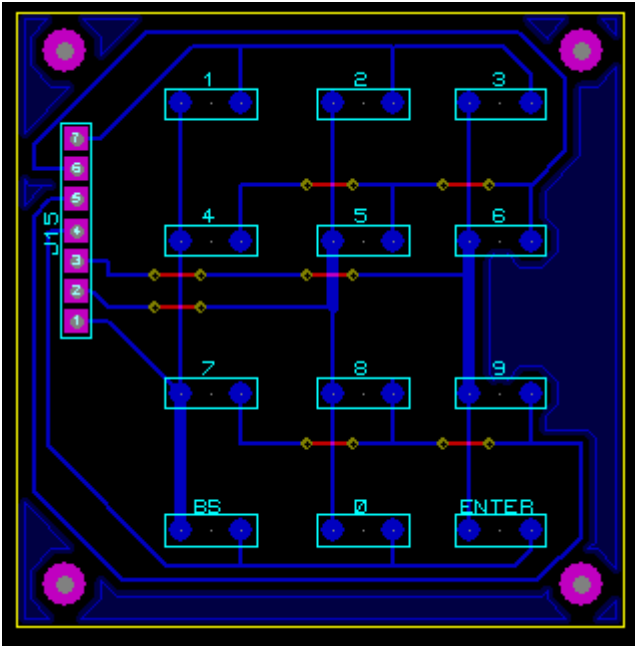
Layout da CPU



Layout dos sensores óticos



Layout do teclado



APÊNDICE D. CODIGO FONTE

```
////////////////////////////////////
////////Integrantes: Gabriel Cardoso, Abner Pereira e Marcos.////////
////////ETEC  GILDO  MARÇAL  BEZERRA  BRANDÃO,  ELETRONICA
4ºZ////////
////////Este é o programa da nossa Caixa Inteligente de Correspondência////////
////////////////////////////////////

//#include <18f4520.h>

#include <16f877a.h>// Definição do microcontrolador que será usado
#include <delay.h>// Frequência de operação (20MHz)
#include <rs232.h>// Definindo RX e TX

//fuses hs,noprotect,nobrownout,nolvp,nowdt,nocpd//Proteção e configurações do PIC

//#Fuses
HS,NOWDT,NOPUT,NOBROWNOUT,NOLVP,NOCPD,NOWRT,NODEBUG,NOPROTECT

#define sn1 PIN_C0//Entrada do sensor 1
#define sn2 PIN_C1//entrada do sensor 2
#define sn3 PIN_C2//entrada do sensor 3
#define rst PIN_C3//RESET lcd
#define RS PIN_E0//RS do lcd
#define RW PIN_E1//RW do lcd
#define EN PIN_E2//E do lcd
#define prog PIN_C4//Tecla de programação
#define B1 PIN_B1//Botão 1
#define B2 PIN_B2//Botão 2
#define B3 PIN_B3//Botão 3
#define B4 PIN_B4//Botão 4
#define B5 PIN_B5//Botão 5
#define B6 PIN_B6//Botão 6
#define B7 PIN_B7//Botão 7
#define pwrkey PIN_C5//Liga módulo GSM

//#define msg int

char msg1[17]=" ETEC PERUS ";//Informação que irá aparecer na primeira linha do display
char msg2[17]=" MENSAGENS =";//Informação que irá aparecer na segunda linha do display
char msg3[17]=" PROGRAMACAO";//Informação que irá aparecer na primeira linha do display
char msg4[17]=" TEL: ";//Informação que irá aparecer na segunda linha do display
/*char Aluno[17]=" Alunos";
```

```

char Gabriel[17]={"Gabriel"};
char Abner[17]={"Abner"};
char Marcos[17]={"Marcos"};
char escola1[17]={"ETEC GILDO"};
char escola2[17]={"M. B. BRANDAO"};
char Curso[17]={"ELETRONICA 4°Z"},*/

int indice1=0, indice2=0;//Variaveis de controle
int carta = 0;//Variavel "carta"

char telefone[9];//Variável telefone
int y,x,w = 0;//variável

int msg;//variável

void escrita(int valor)//função que executa o envio de bytes ao barramento
{
    output_high(EN);//pino definido como EN no estado "alto"
    delay_ms(1);//Aguarda 1 ms
    output_d(valor);//envia byte ao barramento de comunicação
    delay_ms(1);//Aguarda 1 ms
    output_low(EN);//gera borda de descida (dado armazenado na memória)
}

void teclado()//Declara que teclado sera uma sub-rotina
{
    output_low(RS);//Habilitando para receber instrução
    escrita(0x80);//Desloca o cursor para a posição inicial da primeira linha
    escrita(0x0f);//Display aceso com cursor intermitente
    output_high(RS);//Habilitando para reber dados

    while(msg3[indice1]!=0)//
    {
        escrita(msg3[indice1]);//
        indice1++;//indice1= indice1+1
    }

    output_low(RS);
    escrita(0xc0);//
    output_high(RS);

    while(msg4[indice2]!=0)//
    {
        escrita(msg4[indice2]);//
        indice2++;// indice2= indice2+1
    }

    x=0x20;
    output_low(RS);//Habilitando para receber instrução
    escrita(0xC6);//Declara que este "local" do display irá aparecer as funções digitadas

```

```

no teclado
output_high(RS); //Habilitando para receber dados
while (!input(prog)) //Faça enquanto for verdadeiro
{
    indice1=0, indice2=0;

    delay_ms(2); //Aguarda 2 ms //
    output_high(B1); //Não há verificações na coluna 1
    output_high(B2); //Não há verificações na coluna 2
    output_low(B3); //Varredura da coluna 3

    if(!input(B7)&&(y<9)) //Se for pressionado o B7
    {
        while(!input(B7)){}
        x=0x33; //B7 corresponde ao numero 3 se for pressionado
        telefone[y]=x;
        y++;
        escrita(x); //tudo que for digitado no teclado, este comando será enviado para o
        display
    }
    if(!input(B6)&&(y<9)) // Se for pressionado
    {
        while(!input(B6)){}
        x=0x36; //B6 corresponde ao numero 6 se for pressionado
        telefone[y]=x;
        y++;
        escrita(x); //tudo que for digitado no teclado, este comando será enviado para o
        display
    }
    if(!input(B5)&&(y<9)) //Se for pressionado o B5
    {
        while(!input(B5)){}
        x=0x39; //Se for pressionado, B5 será o numero 9
        telefone[y]=x;
        y++;
        escrita(x); //tudo que for digitado no teclado, este comando será enviado para o
        display
    }
    if(!input(b4)) //Se for pressionado o B4
    {
        while(!input(B4)){} //pressionado enter
        {
            y=0;
            while(y<9)
            {
                write_eeprom(y,telefone[y]);
                y++;
            }
            y=0;
            output_low(RS);
        }
    }
}

```

```

escrita(0x01);//
escrita(0xc7);
output_high(RS);
escrita("O");
escrita("K");
while(!input(prog))
{
}
}

```

```

while(telefone[y]!=0)
{
escrita(telefone[y]);
y++;
}

```

```

y=0;
}
delay_ms(2);//

```

```

////////////////////

```

```

output_high(B1);//Não há verificações na coluna 1
output_low(B2);//Varredura da coluna 2
output_high(B3);//Não há verificações na coluna 3

```

```

if(!input(b7)&&(y<9))//Se for pressionado o B7
{
while(!input(B7)){
x=0x32;//Se for pressionado, B7 será o numero 2
telefone[y]=x;
y++;
escrita(x);//tudo que for digitado no teclado, este comando será enviado para o
display
}
if(!input(b6)&&(y<9))//Se for pressionado o B6
{
while(!input(B6)){
x=0x35;//Se for pressionado, B6 será o numero5
telefone[y]=x;
y++;
escrita(x);//tudo que for digitado no teclado, este comando será enviado para o
display
}
if(!input(b5)&&(y<9))//Se for pressionado o B5
{
while(!input(B5)){
x=0x38;//Se for pressionado, B5 será o numero 8
telefone[y]=x;

```

```

y++;
escrita(x);//tudo que for digitado no teclado, este comando será enviado para o
display
}
if(!input(b4)&&(y<9))//Se for pressionado o B4
{
while(!input(B4)){
x=0x30;//Se for pressionado, B4 será o numero 0
telefone[y]=x;
y++;
escrita(x);//tudo que for digitado no teclado, este comando será enviado para o
display
}
delay_ms(2);//Aguarda 2 ms

////////////////////////////////////

output_low(B1);//Varredura da coluna 2
output_high(B2);//Não há verificações na coluna 2
output_high(B3);//Não há verificações na coluna 3

if(!input(b7)&&(y<9))//Se for pressionado o B7
{
while(!input(B7)){
x=0x31;//Se for pressionado, B7 será o numero 1
telefone[y]=x;
y++;
escrita(x);//tudo que for digitado no teclado, este comando será enviado para o
display
}
if(!input(b6)&&(y<9))//Se for pressionado o B6
{
while(!input(B6)){
x=0x34;//Se for pressionado, B6 será o numero 4
telefone[y]=x;
y++;
escrita(x);//tudo que for digitado no teclado, este comando será enviado para o
display
}
if(!input(b5)&&(y<9))//Se for pressionado o B5
{
while(!input(B5)){
x=0x37;//Se for pressionado, B5 será o numero 7
telefone[y]=x;
y++;
escrita(x);//tudo que for digitado no teclado, este comando será enviado para o
display
}
if(!input(b4)&&(y<10)&&(y>0))//Se for pressionado o Backspace
{

```

```

while(!input(B4)){
output_low(RS);//
escrita(0x04);//
escrita(0x0f);//
output_high(RS);//
escrita(" ");
y--;
output_low(RS);//
escrita(0x06);//
// escrita(0x0c);//
output_high(RS);//
}
}
}
void inicializa();//Inicialização do display
{
output_low(RS);//Habilitando para receber instrução
escrita(0x38);//Configura display para 2 linhas 5x7 8 bits
escrita(0x06);//Escreve deslocando o cursor para a direita
escrita(0x0c);//Display aceso sem cursor
escrita(0x01);//Limpa display
output_high(RS);//Habilita para mandar dados
delay_ms(3);//Aguarda 3 ms
}

void reset();//Função de reset da contagem no LCD
{
msg=0;//
write_eeprom(20,msg);
output_low(RS);//configura o LCD para receber uma instrução RS=0
escrita(0xcd);//
output_high(RS);//configura o LCD para receber um dado RS=1
escrita(0x20);
}
void valor_lcd(long int w)
{
int dezena=0x30,unidade=0x30;
while(w>=10)
{
w=w-10;
dezena++;
}
while(w>=1)
{
w=w-1;
unidade++;
}
output_low(RS);//configura o LCD para receber uma instrução RS=0
escrita(0xcd);//Posição que a contagem aparecerá no display
output_high(RS);//configura o LCD para receber um dado RS=1

```

```
escrita(dezena);//
escrita(unidade);//
}
```

```
void main()
{
  SETUP_CCP1(CCP_OFF);
  PORT_b_PULLUPS(true); //Ativa pullup interno do portB
  output_low(RW); //Manda nível lógico baixo neste pino do display, pois ele não será
  utilizado
  inicializa(); //Chamada de função inicializa
  y=0;
```

```
  output_low(pwrkey);
  delay_ms(2500);
  output_high(pwrkey);
  msg=read_eeprom(20);
  valor_lcd(msg);
  while(y<9)
  {
    telefone[y]=read_eeprom(y);
    y++;
  }
  y=0;
  while (true)//
  {
    output_low(RS);
    escrita(0x80); //
    output_high(RS);
    while(msg1[indice1]!=0)//
    {
      escrita(msg1[indice1]); //
      indice1++; //indice1= indice1+1
    }
  }
```

```
  output_low(RS);
  escrita(0xc0); //
  output_high(RS);
```

```
  while(msg2[indice2]!=0)//
  {
    escrita(msg2[indice2]); //
    indice2++; // indice2= indice2+1
  }
```

```
  output_low(RS);
  escrita(0xce); //
  output_high(RS);
```



```

//valor_lcd(msg);

if (((input(sn1))||(input(sn2))||(input(sn3)))&&(carta == 0))//Se for acionado alguns desses sensores
{
// while((input(sn1))||(input(sn2))||(input(sn3)))//só ira acionar o sensor depois
// que a carta passar pelo mesmo.
// {
while (carta == 0)
{

delay_ms(200);// Aguarda 2 segundos
carta = 1;//A variável carta é verdadeira e os comando do display e do GSM serão enviados
}
// }
msg++;// Incrementa mais 1 (+1) no valor do display, conforme for acionado os sensores
write_eeprom(20,msg);//Grava o número de telefone na EEPROM interna do MCU
valor_lcd(msg);// escreve valor no lcd


delay_ms(250);//Aguarda 250 ms
printf("AT\r\n");//Comando AT informa ao modem que será enviado um comando
delay_ms(200);//Aguarda 20ms para o próximo comando
printf("AT+CMGF=1\r\n");//Função para mandar sms
delay_ms(200);//Aguarda 20ms para o próximo comando


printf("AT+CMGS=\"%c%c%c%c%c%c%c%c%c%c\r\n",telefone[0],
telefone[1],telefone[2],telefone[3],telefone[4],
telefone[5],telefone[6],telefone[7],telefone[8]); //Inserção do numero de telefone
delay_ms(200);//Aguarda 20ms para o próximo comando
printf("PROJETO CIE:Voce possui %u nova(s)
correspondencia(s)\r\n",msg);//Mensagem a ser enviada
delay_ms(250);//Aguarda 25ms para o próximo comando
putc(0x1a); //Comando de envio de mensagem
delay_ms(250);
}


if(input(rst))
{
reset();//Reset da contagem no LCD
valor_lcd(msg);
}
if (!(input(sn2))&&(input(sn3))&&(input(sn1))) carta = 0;// Se não estiver acionado os sensores
//a variável carta voltará a ser falsa e irá aguardar a entrada dos sensores novamente.

while (!input(prog))//Faça enquanto atecla PROG for Acionada

```

```
{
inicializa();//Inicializa display para o modo teclado
indice1=0, indice2=0;//Zera os valores dos indices
teclado();//Vai para a função teclado
inicializa();//Volta para a mensagem inicial do display
valor_lcd(msg);//E guarda a contagem que estava anterior da quantidade das
mensagens

}
}
}
```

APÊNDICE E. Relatórios de Atividades Individuais

RELATÓRIO INDIVIDUAL DIA 16/02 – por Abner Pereira Silva

OBJETIVO

O objetivo da semana foi desenvolver e testar programações para o Modulo GSM e para o Display de LCD.

DESENVOLVIMENTO

Graças há um colega de trabalho do integrante do projeto Gabriel Cardoso conseguimos uma grande ajuda na parte de programação. Com isso conseguimos simular o Modulo GSM mandar uma mensagem SMS.

Após feita a simulação do modulo pesquisamos algumas programações para fazer um Display LCD 16X2 mostrar as informações do Modulo GSM como nível do sinal de transmissão e o texto que será exibido quando houver uma carta.

Em testes conseguimos fazer o display mostrar a mensagem de testes “ETEC GILDO MARCAL BEZERRA BRANDAO” em um circuito individual, porém quando incluímos a programação do display na programação do Modulo GSM o display não funcionou.

Dia 21/02

Quando fomos testar a programação no hardware do Modulo GSM descobrimos que infelizmente este modulo não faz comunicação com o microcontrolador PIC.

CONCLUSÃO

Concluimos que a programação não funcionou por termos incluído a do display de forma inadequada na do Modulo GSM.

Iremos elaborar e organizar meu as funções da programação e buscar mais informações sobre o nosso Modulo GSM para fazer a comunicação com o PIC.

RELATÓRIO INDIVIDUAL DIA 27/02 e 28/02 – por Abner Pereira Silva

OBJETIVO

O objetivo da semana foi pensar em uma forma de fazer a comunicação com o nosso Modulo GSM.

DESENVOLVIMENTO

Como na semana passada nossos testes no protoboard com o Modulo GSM falharam, pensamos que o modulo que compramos não fazia comunicação com o microcontrolador PIC.

Porem o integrante do projeto Gabriel realizou testes com o Modulo GSM e através do Compilador MikroC no Computador conseguiu fazer a comunicação onde conseguiu enviar um SMS e fazer uma ligação utilizando seu chip da operadora no modulo GSM.

CONCLUSÃO

Concluimos com os testes que realizados que os testes iniciais falharam por não termos testado pelo MikroC e por causa de algumas configurações que não definimos.

RELATÓRIO INDIVIDUAL DIA 03/02 – por Gabriel Cardoso de Paula

OBJETIVO

O nosso objetivo essa semana foi fazer os testes com o módulo GSM e algumas modificações na monografia.

CONTEÚDO

No domingo (23 de fevereiro), nos reunimos em minha casa para fazer as modificações na monografia do nosso projeto, que foi a mudança do micro controlador que seria utilizado no projeto dentre outras coisas. Essa semana toda eu estive fazendo testes com o módulo GSM, mais só hoje (28 de fevereiro) consegui fazer ligação para outro celular e o que era esperado, mandar mensagem de texto através do modem GSM, porém foi via PC que consegui mandar SMS, já a chamada foi totalmente pelo PIC, onde obtive êxito nesta parte.

CONCLUSÃO

Conclui que, esta semana conseguimos elaborar os testes que encontramos dificuldades na semana passada, e foi muito proveitosa para o nosso projeto e adiantamos uma boa parte do projeto

RELATÓRIO INDIVIDUAL DIA 02/03 – por Gabriel Cardoso de Paula

OBJETIVO

O nosso objetivo essa semana foi fazer os testes com o módulo GSM e a inserção da programação do display

CONTEÚDO

Essa semana eu estive fazendo alguns testes com o módulo GSM e consegui fazer

o envio da mensagem através do microcontrolador, e o teste foi efetuado com sucesso. Na mesma semana consegui fazer a inserção da programação do display.

CONCLUSÃO

Conclui que, esta semana conseguimos adiantar uns 50% do nosso projeto.

RELATÓRIO INDIVIDUAL DIA 06/04 – por Gabriel Cardoso de Paula

OBJETIVO

Elaborar o layout da placa

CONTEÚDO

Essa semana eu alguns testes à respeito do layout da placa, porém não terminei.

CONCLUSÃO

Conclui que falta apenas o layout da placa, para montarmos os circuitos

APÊNDICE F. IMAGENS DA ESTRUTURA PRONTA

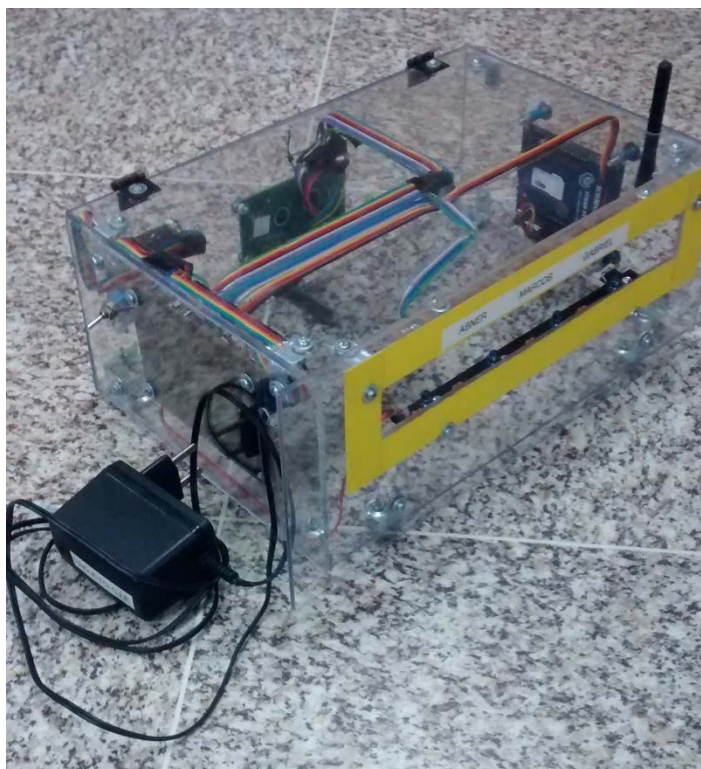


Fig. 16 – Vista superior da estrutura da caixa de correspondência

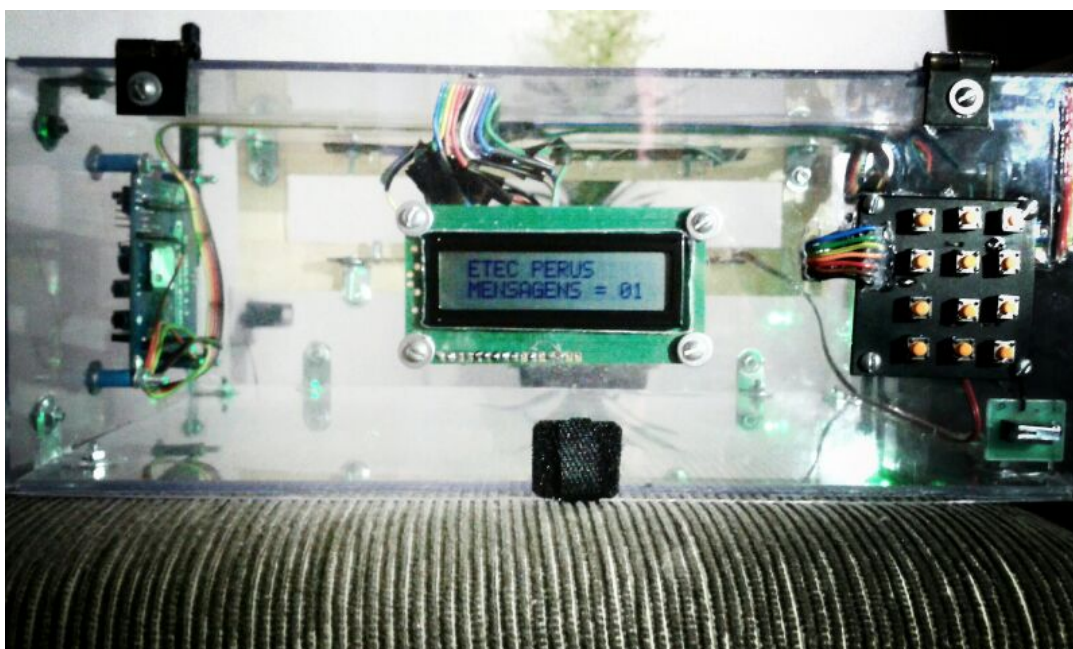


Fig. 17 – Vista traseira da caixa de correspondência

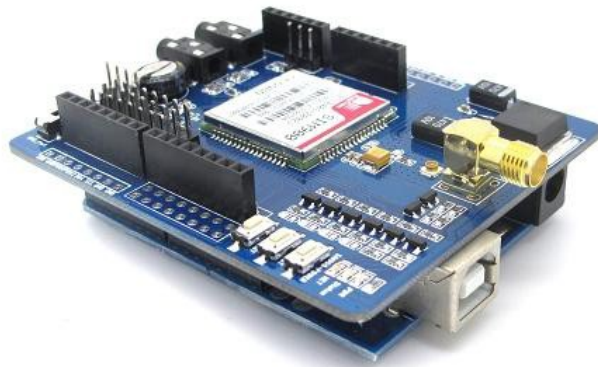
ANEXOS

ANEXO A. Datasheet Arduino Shield GSM GPRS SIM900

IComSat v1.1

-SIM900 GSM/GPRS shield

Overview



IComsat is a GSM/GPRS shield for Arduino and based on the SIM900 Quad-band GSM/GPRS module. It is controlled via AT commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands), and fully compatible with Arduino / Iteduino and Mega.

Features

- ☐ Quad-Band 850/900/1800/1900MHz
- GPRS multi-slot class 10/8
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- ☐ Class 4 (2W @ 850/900MHz)
- ☐ Class 1 (1W @ 1800/1900MHz)

- Control via commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands)
- Short message service
- Free serial port selection
- All SIM900 pins breakout
- RTC supported with Super Cap
- Power on/off and reset function supported by Arduino interface

Specifications

PCB size	77.2mm X 66.0mm X 1.6mm
Indicators	PWR, status LED, net status LED
Power supply	9~20V, compatible with Arduino
Communication Protocol	UART
RoHS	Yes

Electrical Characteristics

Specification	Min	Type	Max	Unit
Power Voltage(Vlogic)	4.5	5	5.5	VDC
Power Voltage(Vsupply)	9	-	20	VDC
Input Voltage VH:	4.5	5	5.5	V
Input Voltage VL:	-0.3	0	0.5	V
Current Consumption(pulse)	-	-	2000	mA
Current Consumption(Continues)			500	mA
Baud rate		9600		bps

Hardware

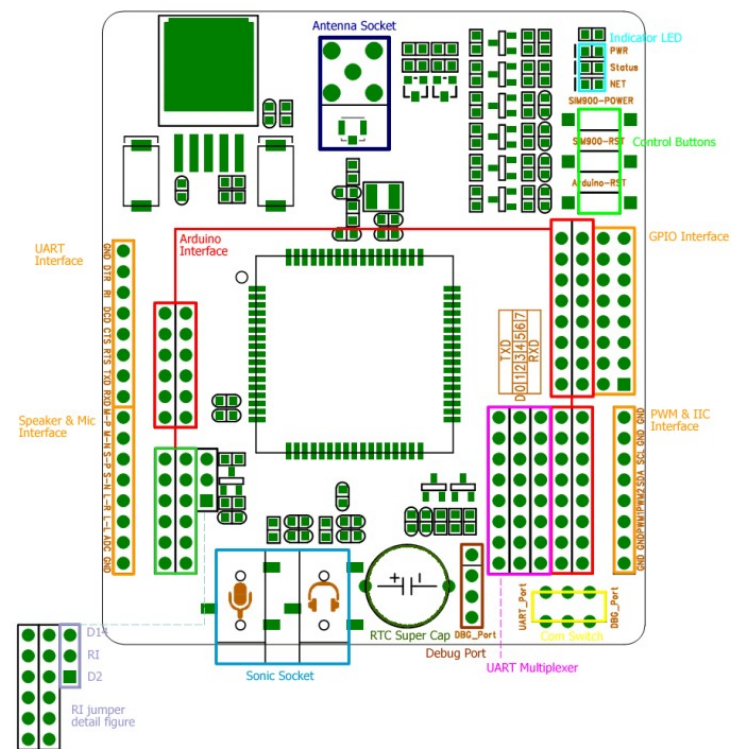


Figure 1 Top Map

Interface	Pin	Descripti
GPIOs	1	VDD*
	2	GN
	3	GPIO1
	4	GPIO2
	5	GPIO3
	6	GPIO4
	7	GPIO5
	8	GPIO6
	9	GPIO7
	1	GPIO8
	1	GPIO9
	1	GPIO10
	1	GPIO
	1	GPIO12

	1	GND
	1	VDD*
UART	1	GN
	2	DTR
	3	RI
	4	DCD
	5	CT
	6	RT
	7	TXD
	8	RXD
IIC&PWM	1	GN
	2	GN
	3	IIC_SCL
	4	IIC_SDA
	5	PWM
	6	PWM1
	7	GN
	8	GN
Debug_Port	1	GN
	2	PERKEY
	3	DBG_R
	4	DBG_TXD

VDD* = 3.0V

Note 1: the operation level of the port is 3.0V

Installation

UART Multiplexer (For free UART connection setting)

You can use the jumper to connect the TXD and RXD pins on SIM to any pins of Arduino D0

—
D7.



Figure 3 UART Multiplexer

When using the connection as Figure 4, the SIM900 connect to the ATmega328 chip on board.

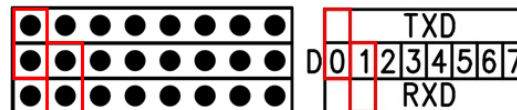


Figure 4 Connect the Arduino board

When you using the connection as Figure 5, the SIM900 connect with the FT232RL chip, and the FT232RL connect to PC by USB. Whit this configuration you can use the serial software on PC to control or configure the SIM900 module.

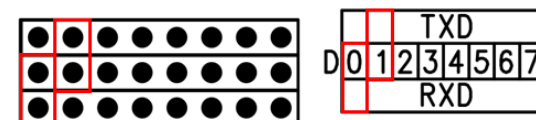


Figure 5 Connect the UART Interface as FT232

Except the 2 configurations above, you can connect the TXD and RXD to any other pins from D0-D7, and using the software-serial library to control the SIM900 module.

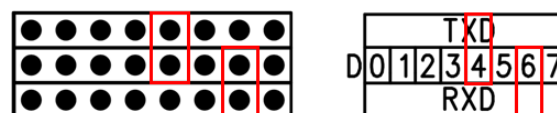


Figure 6 an example for software-serial connection

Com Switch

There is a com switch for selecting the communicate port to UART port or Debug port of SIM900 module as the following figure 7. When connect to the SIM900 debug port, the UART multiplexer just can be set as Figure 4 or 5.

RI Behavior Output

There is a Jumper for configuration of RI, one to connect D2, another to connect D14.

Statu	RI response
Standby	High
Voice call	The pin is changed to low. When any of the following events occur, the pin will be changed to high:
Data call	The pin is changed to low. When any of the following events occur, the pin will be changed to high:
SMS	The pin is changed to low, and kept low for 120ms
URC	The pin is changed to low, and kept low for 120ms when some URCs are reported. Then

The behavior of the RI pin is shown in the following figure when SIM900 module is used as a receiver.

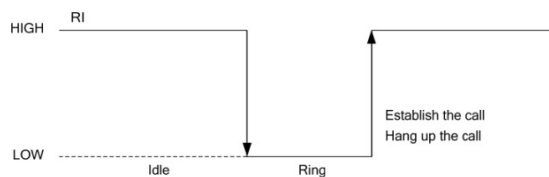


Figure 8 RI behavior of voice calling as a receiver

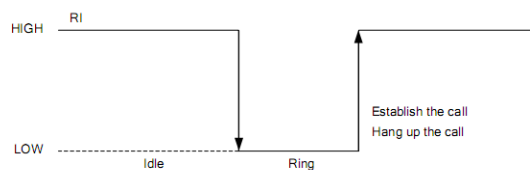


Figure 9 RI behaviors of data calling as a receiver

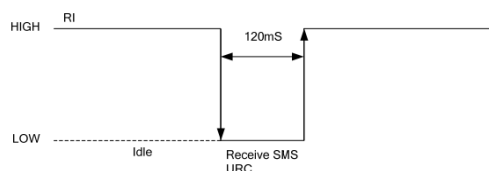


Figure 10 RI behavior of URC or receive SMS When the IComSat is used as caller, the RI will remain high.

Indicator LED and Buttons:

NETSTATUS: The status of the NETSTATUS LED is listed in

following table:

Statu	Description
Off	SIM900 is not running
64ms On/800ms	SIM900 not registered the network
64ms On/3000ms	SIM900 registered to the network
64ms On/300ms	GPRS communication is established

STATUS: Power status of SIM900.

PWR: Power status of IcomSat.

SIM900-POWER: After the IcomSat power on, you need to press the SIM900-POWER button for a moment to power on the SIM900 module. The pin 9 of Arduino interface is connecting to PWRKEY, and a high pulse with 400us wide can power on/off it.

Sim900-RST: Reset the SIM900. The pin 10 of Arduino interface is connecting to RESRT of SIM900, and a high pulse with 400us wide can power on/off it.

Arduino-RST:
Reset the Arduino.

Software Notes:

1. After module power on , please DO NOT operate the SIM card until get the URC "CALL READY"(with fixed baud rate) (User can use AT+CCALR to inquire the states of "CALL READY")
2. When meets error 512 or error 515:
 - Confirm that have not operated the SIM card before "CALL READY" response
 - Confirm that the memory of SIM card is not full

REVISION HISTORY

Rev.	Description	Release date
v1.0	Initial version	2011-5-27

v1.1	Add the RI description, and fix some typo	2011-7-22
v1.2	Update for IComsat v1.1	2012-1-17
V1.3	Add Software Notes	2012-04-21

ANEXO B. Datasheet Display LCD MDLS16265

VARITRONIX LIMITED

LCM Design Engineering

SPECIFICATION

FOR

LCD MODULE TYPE ITEM NO.:

MDLS16265-04

MDLS16265-LV-G-LED04G (BB) (DOC.

REVISION 1.0)

DEPARTMENT	NAME	SIGNATURE	EFFECTIVE DATE
PREPARED BY	PHILIP CHENG		1999.10.14
APPROVED BY	ANDY LEUNG		1999.10.14

1. Electrical Specifications

1.1 Interface signals

Table 4

Pin No.	Symbol	Description
1	Vss	Ground
2	VDD	Power supply for logic (+5V)
3	VO	Power supply for LCD driver
4	RS	Register Select Input: "High" for Data register (for read and write) "Low" for Instruction register (for write),
5	R/W	Read/Write signal: 'High' for Read mode. 'Low' for
6	E	Enable . Start signal for data read /write.
7	DB0	Data input/output (LSB)
8	DB1	Data input/output
9	DB2	Data input/output
10	DB3	Data input/output
11	DB4	Data input/output
12	DB5	Data input/output
13	DB6	Data input/output
14	DB7	Data input/output (MSB)
A	LED(+)	Anode of LED Backlight.
K	LED(-)	Cathode of LED Backlight.

1.1 Typical Electrical Characteristics at Ta = 25 °C, VDD = 5V+5%, VSS = 0V.

Table 5

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply voltage (Logic)	V _{DD} - V _{SS}		4.75	5.00	5.25	V
Supply voltage (LCD)	V _{DD} - V ₀	V _{DD} = 5V, Note (1)	4.4	4.65	4.9	V
Input signal voltage 1 for E,DB0-	V _{IH1}	“High” level	2.2	-	V _{DD}	V
	V _{IL1}	“Low” level	0	-	0.6	V
Input signal voltage 2 for OSC1	V _{IH2}	“High” level	V _{DD} -	-	V _{DD}	V
	V _{IL2}	“Low” level	0	-	1.0	V
Supply Current	I _{DD}	Note(1)	-	1.7	2.3	mA
Supply Current (LCD)	I ₀	Note(1)	-	0.2	0.3	mA
Supply Voltage of LED04		Forward current = 18 x5= 90mA	3.8	4.1	4.5	V

Note (1):

There is tolerance in optimum LCD driving voltage during production and it will be within the specified range.



PIC16F87XA

Data Sheet

**28/40/44-Pin Enhanced Flash
Microcontrollers**

PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873 A
- PIC16F874 A
- PIC16F876 A
- PIC16F877 A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM),
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)

- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (VREF) module
 - Programmable input multiplexing from device inputs and internal voltage reference
 - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control

- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

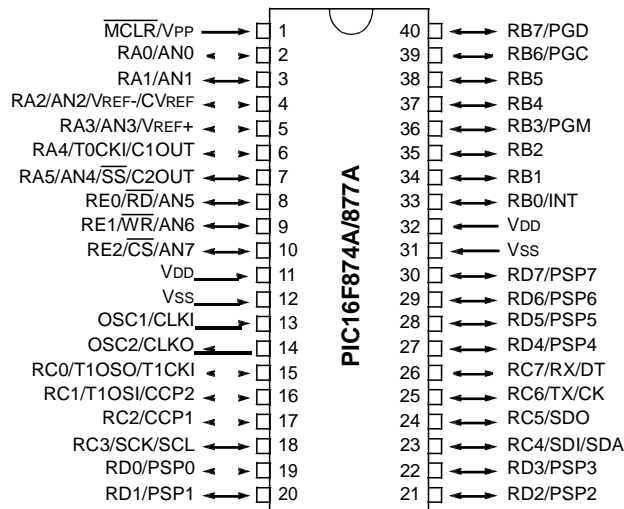
CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

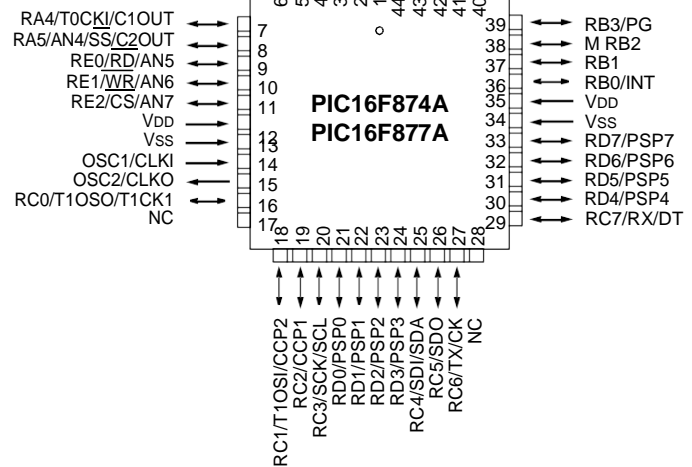
Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

Pin Diagrams (Continued)

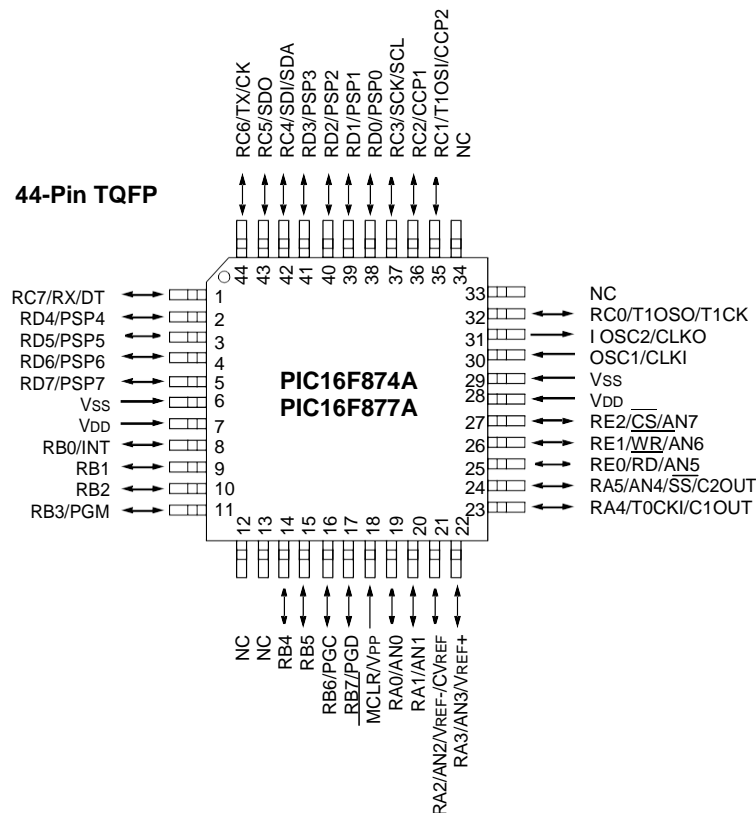
40-Pin PDIP



44-Pin PLCC



44-Pin TQFP



10.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI.) The USART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The USART can be configured in the following modes:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

Bit SPEN (RCSTA<7>) and bits TRISC<7:6> have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

The USART module also has a multi-processor communication capability using 9-bit address detection.

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D

bit 7

bit 0

- bit 7 **CSRC:** Clock Source Select
 bit Asynchronous mode:
 Don't care.
Synchronous mode:
 1 = Master mode (clock generated internally from BRG)
 0 = Slave mode (clock from external source)
- bit 6 **TX9:** 9-bit Transmit Enable bit
 1 = Selects 9-bit transmission
 0 = Selects 8-bit transmission
- bit 5 **TXEN:** Transmit Enable bit
 1 = Transmit enabled
 0 = Transmit disabled
Note: SREN/CREN overrides TXEN in Sync mode.
- bit 4 **SYNC:** USART Mode Select bit
 1 = Synchronous mode
 0 = Asynchronous mode
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **BRGH:** High Baud Rate Select
 bit Asynchronous mode:
 1 = High speed
 0 = Low speed
Synchronous mode:
 Unused in this

mode.

bit 1 **TRMT:** Transmit Shift Register Status bit

1 = TSR empty

0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data, can be Parity bit

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D

bit 7bit 0

- bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled
- bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception
- bit 5 **SREN:** Single Receive Enable
bit Asynchronous mode:
Don't care.
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave:
Don't care.
- bit 4 **CREN:** Continuous Receive Enable bit
bit Asynchronous mode:
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive
- bit 3 **ADDEN:** Address Detect Enable
bit Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
- bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error
- bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error
- bit 0 **RX9D:** 9th bit of Received Data (can be parity bit but must be calculated by user firmware)

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

ANEXO D.**TERMO DE AUTORIZAÇÃO DE DIVULGAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO – TCC**

Nós, alunos abaixo assinados, regularmente matriculados no Curso Técnico de _____, na qualidade de titulares dos direitos morais e patrimoniais de autores da obra “ _____ ”,

Trabalho de Conclusão de Curso apresentado na ETEC Gildo Marçal Bezerra Brandão, no bairro de Perus, município de São Paulo, em ____ de Maio de 2014, autorizamos o Centro Paula Souza a reproduzir integral ou parcialmente o trabalho e/ou disponibilizá-lo em ambientes virtuais.

São Paulo, ____ de Maio de 2014.

NOME	RG	ASSINATURA

Ciência do Professor Responsável

Nome	Assinatura	Data

ANEXO E.

TERMO DE AUTENTICIDADE

TRABALHO DE CONCLUSÃO DE CURSO – TCC

Nós, alunos abaixo assinados, regularmente matriculados no Curso Técnico de _____ da ETEC Gildo Marçal Bezerra Brandão, situada a Rua Presidente Vargas, s/nº em Perus, município de São Paulo, declaramos ter pleno conhecimento do Regulamento para a realização do Trabalho de Conclusão de Curso do Centro Paula Souza. Declaramos ainda, que o trabalho apresentado é resultado do nosso próprio esforço e que não há cópia de obras impressas e eletrônicas.

São Paulo, ____ de Maio de 2014.

[illegible]