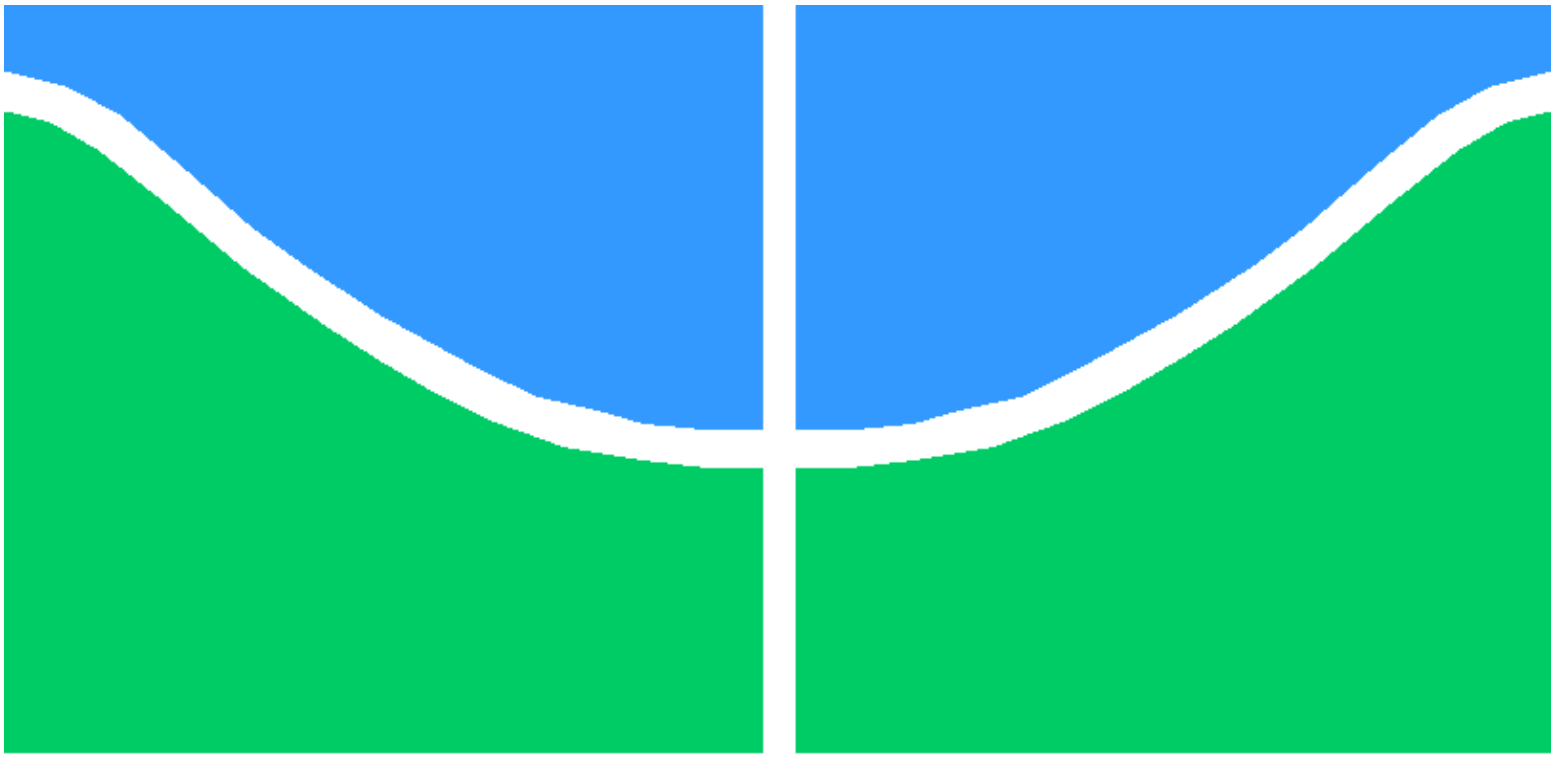


PROJETO DE UM KIT ARM PARA SIMULAÇÃO DE UM CLP RESIDENCIAL DE BAIXO CUSTO COM PLACA DE EXPANSÃO DE RELÊS SEM FIO

ILTON PEREIRA DE OLIVEIRA

DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS

DEPARTAMENTO DE ENGENHARIA MECÂNICA

Brasília Dezembro de 2014

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA MECÂNICA

PROJETO DE UM KIT ARM PARA SIMULAÇÃO DE UM CLP RESIDENCIAL DE BAIXO CUSTO COM PLACA DE EXPANSÃO DE RELÊS SEM FIO

ILTON PEREIRA DE OLIVEIRA

ORIENTADOR: CARLOS HUMBERTO LLANOS QUINTERO

DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS

PUBLICAÇÃO ENM – DM – 84/14

Brasília 14 de Dezembro de 2014

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA MECÂNICA

PROJETO DE UM KIT ARM PARA SIMULAÇÃO DE UM CLP RESIDENCIAL DE BAIXO CUSTO COM PLACA DE EXPANSÃO DE RELÊS SEM FIO

ILTON PEREIRA DE OLIVEIRA

DISSERTAÇÃO A SER SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISÍTOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM SISTEMAS MECATRÔNICOS.

APROVADA POR:

_____

Prof. Dr. Carlos Humberto Llanos Quintero (ENM-UnB)
(ENM-UnB, orientador)

_____

Prof. Rudi van Els (FGA-UnB)
(membro interno ao PPMEC)

_____

Prof. Pedro de Azevedo Berger
(CIC-UnB) (membro externo)

_____

Prof. Daniel Mauricio Muñoz Arboleda
(FGA-UnB, membro suplente)

Brasília Dezembro de 2014

FICHA CATALOGRÁFICA

OLIVEIRA ,ILTON PEREIRA DE

Projeto de um kit ARM para simulação de um CLP residencial de baixo custo com placa de expansão de relês sem fio [Distrito Federal] 2014

xxi, 1pg., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2014).

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Mecânica.

1. Sistemas Embarcados 2. Controlador Lógico Programável

3. ARM CORTEX M3 4. Wireless

I. ENM/FT/UnB II. Título (serie)

REFERÊNCIA BIBLIOGRÁFICA

Oliveira, Ilton Pereira. (2014). Projeto de um kit ARM para simulação de um CLP residencial de baixo custo com placa de expansão de relês sem fio. Dissertação de Mestrado em sistemas mecatrônicos, Publicação ENM – DM– 84/14, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, p. 1.

CESSÃO DE DIREITOS

AUTOR: Ilton Pereira de Oliveira.

TÍTULO: Projeto de um kit ARM para simulação de um CLP residencial de baixo custo com placa de expansão de relês sem fio.

GRAU: Mestre ANO: 2014

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

_____

Ilton Pereira de Oliveira

Ilton.oliveiradf@gmail.com.

Brasília– DF – Brasil.

À minha esposa Cintia e meus Filhos Luan e Brunna, a quem tanto amo, pelo inestimável apoio e compreensão durante toda a minha caminhada.

Aos meus pais, pela educação que me deram e por tudo que fizeram por mim durante toda minha vida acadêmica

AGRADESCIMENTOS

Acima de tudo a Deus e aos meus mentores espirituais que me orientam e zelam pela minha existência.

Agradeço, de forma muito especial ao Prof. Dr. Carlos Humberto Llanos Quintero que, com toda sua competência me orientou generosamente, tendo a paciência digna dos grandes mestres durante todo o processo.

Ao Coordenador do curso Prof. Dr.-Ing. Edson Paulo da Silva que nos últimos momentos deu todo o suporte e acompanhamento para que eu pudesse concluir meu mestrado .

À minha esposa, Cintia, meus Filhos Luan e Brunna , que, com muito amor e paciência meu ajudaram a vencer mais esta batalha.

À Universidade de Brasília por tornar possível a realização desta conquista, não só por oferecer o curso, mas também por toda a estrutura concedida.

A todos os professores e amigos que contribuíram de alguma forma para o desenvolvimento deste trabalho.

RESUMO

PROJETO DE UM KIT ARM PARA SIMULAÇÃO DE UM CLP RESIDENCIAL DE BAIXO CUSTO COM PLACA DE EXPANSÃO DE RELÊS SEM FIO

Autor: Ilton Pereira de Oliveira

Orientador: Carlos Humberto Llanos Quintero

Programa de Pós-Graduação em Sistemas Mecatrônicos

Brasília, Dezembro de 2014

O objetivo deste trabalho é projetar um sistema embarcado baseado em um microcontrolador da família ARM, com o objetivo de simular um CLP (Controlador Lógico Programável) para aplicação em automação residencial, levando em conta vários requisitos, seguindo normas de segurança brasileiras e internacionais, programação simples, usando uma ferramenta gratuita de desenvolvimento. Neste contexto, há uma grande variedade aplicações que podem ser desenvolvidas usando um CLP residencial, tais como, controle de iluminação, controle básico de equipamentos eletrônicos, climatização de ambientes, acionamento do movimentador de portão, irrigação de jardins, abertura de persianas, alarme de segurança por meio da leitura de sensores na casa, entre outros. A proposta deste trabalho é que o CLP residencial torne-se um projeto de código aberto, incluindo todos os códigos fontes, esquemáticos e manuais de usuário, disponibilizando-os em uma página web disponível para receber modificações para novas aplicações. O sistema embarcado desenvolvido é baseado em um ARM CORTEX M3 LPC1768, incluindo também uma porta Ethernet e interfaces CAN, RS232, RS485, USBe e XBee. O sistema também conta com um RTC (relógio em tempo real), memoria interna de 512k, assim como memórias externas do tipo SD-card e EPROM. Além disso, foi desenvolvido um módulo de expansão para controle de cargas, que pode ser controlado a partir da placa-mãe usando tanto conexão XBee quanto via cabo serial. Este módulo é composto de 8 relés que oferecem conexões opto-isoladas que são dirigidas para as tarefas de automação residencial. Adicionalmente, o sistema desenvolvido foi projetado para ser aplicado em cursos de graduação com foco em disciplinas voltadas para microcontroladores e sistemas embarcados para automação e controle. Neste trabalho o quesito de interface, implícito nos CLPs comerciais, envolvendo linguagens de programação tais como Ladder (Logic Diagram Programming), FBD (Function Block Diagram), LD (Ladder Diagram), ST (Structured Text), SFC (Sequential Function Chart), entre outros, é deixado como trabalho futuro.

Palavras-chave: Sistemas Embarcados , ARM CORTEX M3 , Controlador Lógico Programável. Solução eletrônica. Baixo custo. Wireless.

ABSTRACT

The purpose of this work is to design an ARM-based embedded system, addressed to achieve the simulation of a PLC (Programmable Logic Controller) for home automation; taking into account several requirements such as low-cost, international safety standards, following the Brazilian rules, and offering simple programming by using a free tool development. In this context, there are a wide variety of applications that can be developed using a residential PLC, such as lighting control, basic control electronic equipment, air conditioning environments, gate mover driver, irrigation of gardens, opening blinds, security alarm by reading sensors in the home, among others. The purpose of this work is that the proposed residential CLP becomes an open source design, including all software files, data-sheets and user manuals, making available them on an online site, and thus available for receiving modifications for new applications. The developed embedded system is based on an ARM-3 (16-68C), and includes also an Ethernet port, as well as CAN, RS232, RS485, USB, and XBee interfaces. Additionally, the system also has a real time clock, as well as internal 512K memory and SD-card and EPROM external memories. Additionally, a daughter board has been also developed which can be accessed from the mother board using both XBee connection and via parallel cable. This daughter board is composed of 8 relays offering opto-isolated connections which are addressed for home automation tasks. Finally, the developed system has been designed to be applied in undergraduate courses focused in microcontroller application, and automation and control lectures. In this work the interface issue, implicit in commercial PLCs, involving programming languages such as Ladder (Logic Diagram Programming), FBD (Function Block Diagram), LD (Ladder Diagram), ST (Structured Text), SFC (Sequential Function Chart) among others, it is proposed as a future work.

Key words: PLC (programmable logic controller.), microcontroller, home automation, Xbee technology, TCP -IP communication protocols

LISTA DE FIGURAS

FIGURA 1 - *ARDUINO* UNO 8
FIGURA 2 - *SHIELD* INTERFACE *ETHERNET* 8
FIGURA 3 - *SHIELD* INTERFACE RS 485 8
FIGURA 4 - *SHIELD* INTERFACE USB 8
FIGURA 5 - INTERFACES MICROCONTROLADOR ARM LPC 1768 9
FIGURA 6 - CASA INTELIGENTE 17
FIGURA 7 - ARQUITETURA CENTRALIZADA 20
FIGURA 8 - ARQUITETURA DESCENTRALIZADA 22
FIGURA 9 - ESTRUTURA INTERNA DE UM CLP 26
FIGURA 15 - MBED 39
FIGURA 16 -MBED 40
FIGURA 17 - CIRCUITO 40
FIGURA 18 - DIAGRAMA DE BLOCOS 45
FIGURA 19 - *XBEE* APLICAÇÕES 50
FIGURA 20 - REDES *XBEE* 52
FIGURA 21 - MODELOS DE ANTENAS 53
FIGURA 22 - PLACA MBED 56
FIGURA 23 - PLACA LPCXPRESSO 56
FIGURA 24 - ESQUEMÁTICO 59
FIGURA 25 - *LAYOUT* DA PLACA 60
FIGURA 26 - PLACA CLP 1.0 61
FIGURA 27 – ESQUEMÁTICO DA PLACA CLP 1.1 63
FIGURA 28 - *LAYOUT* DA PLACA CLP 1.1 64
FIGURA 29 - PLACA CLP 1.1 MONTADA 65
FIGURA 30 - MBED PERIFÉRICOS 67
FIGURA 31 - RELAY *SHIELD* 68
FIGURA 32 - GSM *SHIELD* FONTE: SPARKFUN.COM 68
FIGURA 33 - *SHIELD* PARA CONTROLE DE MOTORES 69
FIGURA 34 - SENSOR DE TEMPERATURA E UMIDADE 70
FIGURA 35 - SENSOR DE TEMPERATURA E UMIDADE – LIGAÇÃO TÍPICA 70
FIGURA 36 - *SHIELD* SENSOR DE TEMPERATURA E UMIDADE MONTADO 70
FIGURA 37 - PARTE 1 ESQUEMÁTICO – CLP VERSÃO 1.2 71
FIGURA 38 - PARTE 2 ESQUEMÁTICO – CLP VERSÃO 1.2 72
FIGURA 39 - PARTE 3 ESQUEMÁTICO – CLP VERSÃO 1.2 73
FIGURA 40 - *LAYOUT* DA PLACA – VERSÃO 1.2 74

FIGURA 41 - FOTO DA PLACA CLP – VERSÃO 1.2 77
FIGURA 42 - FOTO DA PLACA CLP – VERSÃO 1.2 COM MÓDULO DE REDE 78
FIGURA 43 - CIRCUITO DE ALIMENTAÇÃO 83
FIGURA 44 - KIT LPCXPRESSO 84
FIGURA 45 - DIAGRAMA EM BLOCOS DO LPC1769 86
FIGURA 46 - DIAGRAMA EM BLOCOS DO LPC1769 87
FIGURA 47 – CIRCUITO *XBEE* 89
FIGURA 48 - CIRCUITO DO RTR 90
FIGURA 49 - CIRCUITO TRANSCEIVER PRESENTE NA PLACA LPCXPRESSO 92
FIGURA 50 - BOBINA DE ACOPLAMENTO EMBUTIDA NO CONECTOR J9 93
FIGURA 51 - CIRCUITO USB PRESENTE NA PLACA LPCXPRESSO 94
FIGURA 52 - CONECTOR USB DA PLACA BASE 94
FIGURA 53 - CIRCUITO TRANSCEIVER RS232 95
FIGURA 54 - CIRCUITO TRANSCEIVER RS485 96
FIGURA 55 - CIRCUITO TRANSCEIVER CAN 97
FIGURA 56 - ISOLADORES ÓPTICOS DAS LINHAS DIGITAIS 98
FIGURA 57 - ENTRADAS ANALÓGICAS 98
FIGURA 58 - CIRCUITO DE CONTROLE DOS RELÉS 99
FIGURA 59 - LINHA DE CONTROLE DE CARGA AC 100
FIGURA 60 - CIRCUITO DE INTERFACE COM A EEPROM 101
FIGURA 61 - INTERFACE COM O CARTÃO SD 101
FIGURA 62 - *LAYOUT* FACE DE CIMA (DIMENSÕES NÃO-REAIS) 106
FIGURA 63 - *LAYOUT* FACE DE BAIXO (DIMENSÕES NÃO-REAIS) 107
FIGURA 64 - *LAYOUT* FACE DE CIMA (DIMENSÕES NÃO-REAIS) 108
FIGURA 65 - *LAYOUT* FACE DE BAIXO (DIMENSÕES NÃO-REAIS) 108
FIGURA 66 - TELA INICIO DA IDE 116
FIGURA 67 – IDE FONTE:ELABORADO PELO AUTOR 117
FIGURA 68 - IDE 119
FIGURA 69 - RESULTADO LISTAGEM 1 120
FIGURA 70 - IDE TESTE PLACA DE RELES 121
FIGURA 71 - RESULTADO PLACAS DE RELES 123
FIGURA 72 - MODULOS *XBEE* 124
FIGURA 73 - XCTU 125
FIGURA 74 - PINAGEM *XBEE* 126
FIGURA 75 - ESTRUTURA DO FRAME DE DADOS NO MODO API 126
FIGURA 76 - PLACA MBED COM MICROCONTROLADOR LPC1768 139
FIGURA 77 - MICROCONTROLADOR LPC1768 139
FIGURA 78 - LPCXPRESSO 139

FIGURA 79 - LPC 1768 PERIFÉRICOS 140
FIGURA 80 - CIRCUITO DA FONTE DA PLATAFORMA 141
FIGURA 81 - EXEMPLO LIGAÇÃO LÂMPADA COM RELE 143
FIGURA 82 - LIGAÇÃO TRIAC 145
FIGURA 83 - CIRCUITO EXPANSÃO CN2 147
FIGURA 84 - EXPANSÃO CN2 147
FIGURA 85 - CONFIGURAÇÃO JUMPERS CN2 148
FIGURA 86 - PLACA DE RELES 148
FIGURA 87 - KIT+PLACA DE RELE 149
FIGURA 88 - ENDEREÇO ÚNICO *XBEE* 151
FIGURA 89 - PLACA RELE COM *XBEE* 152
FIGURA 90 - BARRAMENTO EXPANSÃO 155
FIGURA 91 - CIRCUITO BARRAMENTO EXPANSÃO 155
FIGURA 92 - CIRCUITO LIGAÇÃO RS-232 156
FIGURA 93 - PINAGEM RS232 156
FIGURA 94 -CIRCUITO LIGAÇÃO RS-485 157
FIGURA 95 - INTERFACES SERIAIS 157
FIGURA 96 - CIRCUITO LIGAÇÃO CAN 158
FIGURA 97 - INTERFACES SERIAIS 158
FIGURA 98 - CIRCUITOS I2C 159
FIGURA 99 -INTERFACES I2C DA PLATAFORMA 160
FIGURA 100 - SDCARD 160
FIGURA 101 - MAGJACK 161
FIGURA 102 - CIRCUITO *ETHERNET* 161
FIGURA 103 - CIRCUITO CN1 162
FIGURA 104 - BARRAMENTO CN1 162
FIGURA 105 -FUNÇÕES BARRAMENTO 163
FIGURA 106 - OPTO KTIR0621DS 164
FIGURA 107 - CIRCUITO DE LIGAÇÃO 164
FIGURA 108– MÓDULOS *XBEE* 167
FIGURA 109 - ENDEREÇO ÚNICO *XBEE* 170
FIGURA 110 -FORMATO FRAME MODO API 171
FIGURA 111 - TELA INICIAL X-CTU 174
FIGURA 112 - TELA COM PARÂMETROS DO MÓDULO 175
FIGURA 113 -TELA COM RESULTADO DO MÓDULO 176
FIGURA 114 - RESULTADO COMANDO ND 177
FIGURA 115 - ESQUEMÁTICO DA PLACA DE RELE COM *XBEE* 180
FIGURA 116 - *LAYOUT* SUPERIOR DA PLACA 180

FIGURA 117 - *LAYOUT* INFERIOR DA PLACA 181
FIGURA 118 - PLACA COM COMPONENTES 181
FIGURA 119 - CIRCUITO *XBEE* 182
FIGURA 120 - CIRCUITO 182
FIGURA 121 - LIGAÇÃO DA PLACA COM CABO 183
FIGURA 122 - TELA INICIO DA IDE 183
FIGURA 123 - PLACA DE RELE ACIONADA 185
FIGURA 124 - RESULTADO TESTE PLACAS RELES *XBEE* 189

LISTA DE TABELAS

TABELA 1 - SELEÇÃO DA ALIMENTAÇÃO....81
TABELA 2 – PINAGEM E SUAS FUNÇÕES....102
TABELA 3 – CONFIGURAÇÃO E FUNCIONALIDADE DOS PINOS....113
TABELA 4: ELABORADO PELO AUTOR....142
TABELA 5: ELABORADO PELO AUTOR....143
TABELA 6– DESCRIÇÃO PINOS TRIACS....145
TABELA 7– DESCRIÇÃO DOS PINOS DA PLACA DE RELE....149
TABELA 8- FUNÇÕES DOS PINOS DO MODULO *XBEE SERIE 1*....168
TABELA 9- ALGUNS COMANDOS AT DA CONFIGURAÇÃO *XBEE*....172
TABELA 10– COMANDOS AT UTILIZADOS NO EXEMPLO TESTE....176

CÓDIGOS DE PROGRAMAS

LISTAGEM 1 : PROGRAMA 01 TESTE DE I/O....118
LISTAGEM 2: CÓDIGO PROGRAMA TESTE PLACA RELE CONEXÃO COM CABO....122
LISTAGEM 3 : CÓDIGO PROGRAMA TESTE PLACA RELE *XBEE*....128
LISTAGEM 4– EXEMPLO USO DOS RELES....144
LISTAGEM 5– EXEMPLO USO DOS TRIACS....146
LISTAGEM 6– EXEMPLO USO DOS RELES DA PLACA EXTERNA....151
LISTAGEM 7: CÓDIGO PROGRAMA TESTE PLACA RELE *XBEE*....154
LISTAGEM 8 – PROGRAMA TESTE DO OPT-SENSOR....164
LISTAGEM 9: CÓDIGO PROGRAMA TESTE PLACA RELE CONEXÃO COM CABO....185
LISTAGEM 10: CÓDIGO PROGRAMA TESTE PLACA RELE *XBEE*....188

LISTA DE ABREVIATURAS E SIGLAS

Ni - Ca	Nickel - Cadmium
NO	Normally Open
NTC	Negative Temperature Coefficient
PLC	Programmable Logic Control
PTC	Positive Temperature Coeficient
PROM	Programmable Read-Only Memory
RAM	Random Access Memory
SCR	Silicon Controled Rectifier
SFC	System Function Chart
TRIAC	Triode for Alternating Current
VCA	Volt Corrente Alternada
VCC	Volt Corrente Contínua
LED	Light Emitting Diode
LDR	Light Dependent Resistor
NA	Normalmente Aberto
NC	Normally Close
NF	Normalmente Fechado

SUMÁRIO

CESSÃO DE DIREITOS 3
RESUMO iii
ABSTRACT iv
LISTA DE FIGURAS v
LISTA DE TABELAS viii
CÓDIGOS DE PROGRAMAS viii
LISTA DE ABREVIATURAS E SIGLAS ix
1 INTRODUÇÃO 5
1.1 Motivação do trabalho 7
1.2 Justificativa do trabalho 10
1.3 Objetivos do trabalho 12
1.3.1 Objetivo geral 12
1.3.2 Objetivos específicos 12
1.4 Metodologia 13
1.5 Estrutura deste trabalho 14
2 FUNDAMENTAÇÃO TEÓRICA 16
2.1 Domótica como conceito de automação residencial 16
2.2 Classificação dos sistemas domóticos 20
2.3 Arquitetura centralizada 20
2.4 Arquitetura descentralizada 21
2.5 Segmento de automação residencial 23
2.6 Benefícios oferecidos pelos sistemas domóticos 24
2.7 Aspectos gerais sobre CLPs (Controladores Lógico Programáveis) 25
2.8 História das tecnologias baseadas em CLPs 27
2.9 Vantagens do uso de CLPs 29
2.10 Desvantagens de uso de CLPs 29
2.11 Módulos de entradas e saídas para tecnologia CLPs 29
2.12 Estrutura básica de um CLP 31
2.13 Tipos de CLPs 32

2.13.1 CLPs de grande porte *32*
2.13.2 CLPs de médio porte *33*
2.13.3 CLPs de pequeno porte *33*
2.13.4 Arquitetura compacta X modular *34*
2.14 Entradas e saídas 34
2.14.1 Entradas digitais *35*
2.14.2 Entradas analógicas *35*
2.14.3 Saídas *35*
2.14.4 Saídas digitais *36*
2.14.5 Saídas analógicas *36*
2.16 Sistemas embarcados no contexto da domótica 37
2.17 Plataforma de prototipagem MBED para desenvolvimento de aplicações de sistemas embarcados (usada neste projeto) 39
2.17.1 O microcontrolador LPC1768 ..... *42*
2.18 – Tecnologia de comunicação sem fio *ZigBee* 46
2.18.1 – Topologia das redes ZigBee *51*
2.18.2 Módulos de comunicação ZigBee *52*
2.19 Conclusões do capítulo 53
CAPÍTULO 3: METODOLOGIA PROPOSTA PARA O TESTE E ELABORAÇÃO DAS PLACAS DE CIRCUITO IMPRESSO 55
3.1 Levantamento dos requisitos e descrição do *hardware* 55
3.2 Versões das placas 57
3.2.1 Versão 1.0 ..... *57*
3.2.3 Placa CLP versão 1.1 ..... *62*
3.2.4 Versão 1.2 ..... *66*
3.2.5 PLACA VERSÃO 1.4 – VERSÃO FINAL *79*
3.3 A estrutura final da plataforma do projeto desta dissertação é composta por três placas 80
3.3.1 Alimentação *81*
3.3.2 Alimentação externa de backup *83*
3.3.3 LPCxpresso *84*
3.3.4 Recursos disponíveis no LPCxpresso *85*
3.3.5 MBED *87*
3.3.6 A interface XBEE *89*
3.3.7 Relógio de Tempo Real *90*
3.3.8 O projeto de comunicação da placa *92*
3.3.8.1 A comunicação Ethernet *92*
3.3.8.2 A interface USB *93*

3.3.8.3 A interface RS232 ..... *95*
3.3.8.4 A interface RS485 ..... *95*
3.3.8.5 A interface CAN *96*
3.3.9 Linhas de leitura e atuação da placa base *97*
3.3.10 O projeto de memória da placa base *100*
3.3.11 Mapa de IO *102*
3.4 O projeto da placa de reles 105
3.5 LEIAUTE 105
da placa-base e da placa-relés 105
3.5.1 Apresentação e requisitos do LEIAUTE *106*
3.6. Outros aspectos do projeto da placa-base 109
3.6.1 Sinais Ethernet e USB *109*
3.6.2 A carcaça dos conectores *109*
3.6.3 O plano de terra *109*
3.6.4 A espessura das trilhas de alta corrente *110*
3.6.5 A fabricação da placa *110*
3.7 Testes placa-base e placa-reles 111
3.8 Conclusão 112
CAPíTULO 4: PROJETO DE *SOFTWARE* EMBARCADO PARA VALIDAÇÃO E TESTE DO SISTEMA 113
4.1 O mapeamento dos pinos a serem utilizados 113
4.2 O Compilador IDE (*Integrated Development Environment*) 116
4.3 TESTE DE I/O 117
4.4 Teste de acionamento de cargas com relés 120
4.5 Teste da placa de relés com conexão XBEE 124
Conclusão 129
5.1 Considerações gerais 130
5.2 Sugestões para trabalhões futuros 132
REFERÊNCIAS BIBLIOGRÁFICAS 134
APÊNDICE A – MANUAL DE APRESENTAÇÃO DA PLATAFORMA GRACO 1.4 139
A1 - Visão Geral 139
A1.1– Microcontroladores 139
A1.2- Fonte 141
A1.3 – Relês 143
A1.4 Triacs 145
A2 Usando placa porta expansão para Reles 147

A3 Placa rele externa funcionando remotamente via Xbee 151
A4 Expansão pinos dos módulos XBEE 155
A5 Comunicação serial RS-232 156
A6 Comunicação serial RS-485 157
A7 Comunicação serial CAN 158
A8 Comunicação I2C 159
A9 Comunicação ETHERNET 161
A10 Barramento de expansão CN1 162
A11 Barramento de expansão J2 E J3 162
Conclusão do Apêndice A 165
APÊNDICE B – CONFIGURAÇÃO MODULOS XBEE 166
B1 - Visão Geral 166
B2 – Características: 167
B3 - Endereçamento 170
B4 – Forma de operação dos módulos 171
B5 – Configurando os Módulos 172
Conclusão do Apêndice B 178
APÊNDICE C – APRESENTAÇÃO DA PLACA DE RELES 179
C1 - Visão Geral 179
C2 – Esquemático da placa 179
C3 – Layout da placa 180
C4 Circuitos da placa 182
C5 Teste da placa 183
Conclusão sobre a apresentação da placa de reles 190

1 INTRODUÇÃO

Ainda hoje o conceito de automação predial e/ou residencial ainda está sendo desenvolvido, pois, em um projeto de automação, é preciso dar condições para que todos os pequenos sistemas do ambiente, tais como iluminação, segurança, ar condicionado, controle de energia, incêndio, entre outros, possam trabalhar em conjunto de forma otimizada e harmônica, dentro da instalação, o que nem sempre ocorre nos sistemas que atualmente assim se intitulam (PAIVA, 2007).

O termo *domótica* é usado para designar residências que empregam serviços automatizados. Tecnicamente falando, uma rede domótica pode ser representada por um conjunto de serviços interligados que realizam diversas funções de gerenciamento e atuação, podendo, assim, estar conectados entre si por meio de uma rede de comunicação interna e/ou externa (MARIOTONI, 2002).

Uma casa não se torna “inteligente” simplesmente pelo fato de se utilizar de qualquer equipamento de domótica. Neste sentido, a “inteligência” de uma casa está na integração adequada dos sistemas escolhidos com os diversos dispositivos e sistemas existentes, tais como:

a) segurança;
b) gestão de energia;
c) automação de tarefas domésticas;
d) formação cultural e de entretenimento;
e) escritório de trabalho em casa;
f) monitoramento do conforto ambiental;
g) operação e supervisão de instalações (DIAS, César 2004).

Os sistemas de automação atuais podem ser de tecnologias centralizadas ou distribuídas. Os sistemas centralizados, como o nome sugere, são aqueles que dispõem de uma unidade central de controle pela qual todos os dispositivos da instalação são conectados, tanto para o recebimento dos sinais dos sensores, como para enviar os comandos e ajustes aos dispositivos receptores para que executem as operações (após o processamento dos sinais). Já os sistemas com tecnologias distribuídas ou descentralizadas são constituídos de diversos dispositivos com

processamento inteligente próprio, cada um com função específica dentro das inúmeras necessidades do sistema de automação, sendo distribuídos por toda a extensão da instalação, interligados por uma rede, comunicando-se e enviando sinais entre sensores e atuadores que podem se encontrar próximos ou integrados ao ponto de controle e monitoração (DIAS, César 2004).

Nos projetos de domótica atuais, as tecnologias sem fio apresentam-se como uma grande promessa à facilitação da disseminação da domótica, porém é fundamental que os projetistas estejam atentos e preparados para que a concepção de seus projetos habitacionais deem abrigo a essas tecnologias da automação, integrando-as aos demais componentes e sistemas residenciais, permitindo, assim, que os usuários possam desfrutar da melhor forma de seus benefícios.

Neste contexto, o papel da automação vai muito além do aumento da praticidade e qualidade de vida, pois o mesmo está intrinsecamente ligado ao uso racional da energia, água e aos demais recursos naturais (DIAS, César 2004).

O número de empresas que fornecem soluções para automação tem aumentado ao longo dos anos, e novos sistemas mais acessíveis vêm permitindo muitas possibilidades que antes eram apenas ficção tornando-se viáveis do ponto de vista econômico. Adicionalmente, deve-se levar em consideração que o mercado ainda é dominado por sistemas proprietários fechados, com soluções modulares que não se integram com sistemas de outros fabricantes, o que dificulta a criação de projetos que atendam as necessidades em relação a dos custos existentes para implantação de tais sistemas.

Dentro do contexto deste trabalho, foi percebida uma carência de soluções práticas de baixo custo para CLPs residenciais de pequeno porte, que tivessem todo o seu conteúdo (tanto de *hardware* como de *software*) seguindo os preceitos da *GNU (General Public License)* que é uma forma de *copyleft* criada para uso em manuais, livros-texto ou outros documentos, para garantir que qualquer um tenha a real liberdade de copiá-los e redistribuí-los, com ou sem modificações, tanto comercial quanto não comercialmente. A versão mais recente é a 1.3 (www.gnu.org, 2014).

A proposta deste trabalho é de tornar mais acessível a um grande número de pessoas a possibilidade de utilizarem um CLP residencial, em conjunto com sensores e atuadores de fácil aquisição junto ao mercado, que viabilizem a supervisão, o controle e o monitoramento de equipamentos, a fim de se criar um

ambiente inteligente em residências, com a opção da operação remota através da internet, bastando para isso pequenas adaptações em um ambiente já existente.

1.1 Motivação do trabalho

As aplicações da domótica em uma residência são enormes, tais como em gerenciamento do consumo de energia, controle de acesso, iluminação, climatização, consumo de água, sistemas de câmeras, limpeza automatizada, tornando o ambiente residencial mais saudável, eficiente e mais confortável.

Para que essas aplicações saiam do campo da necessidade e passem para a aplicação real, é necessário seguir algumas etapas, por exemplo:

- Criação de projeto de automação com a escolha detalhada dos itens da residência que deverão ser gerenciados;
- Escolha dos sensores e atuadores que farão a interface de comunicação com os CLPs, por exemplo:
    - Módulo sensor de temperatura;
    - Módulo sensor de umidade;
    - Módulo sensor de fumaça;
    - Módulo de rele para aplicações gerais (como abrir e fechar uma porta; abrir um portão eletrônico; ligar e desligar um equipamento);
    - Módulo sensor de alarmes;
    - Módulo controle de acesso.
- Escolha de um CLP (Controlador Lógico Programável) para atuar junto aos sensores e atuadores escolhidos;
- Projeto do *software* da aplicação de gerenciamento do projeto.

No contexto de plataformas que possam ser utilizadas para domótica, existem no mercado algumas soluções de baixo custo; entretanto, as mesmas possuem limitações para aplicações mais elaboradas. Nesta direção, pode-se exemplificar a plataforma *ARDUINO* (Figura 1), que possui *hardware* e *software* com licença *open source*, os quais foram desenvolvidos utilizando um microcontrolador Altmel AVR de

8 *bits*. A plataforma desenvolvida não possui interfaces de comunicação adequadas, precisando assim de vários *shields* (placas de expansão) para inserir, por exemplo, uma interface de *Ethernet* (Figura 2), RS485(Figura 4), rs232, usb, etc (vide Figura 4).

Figura 1 - *ARDUINO* Uno

Fonte : http://*ARDUINO*.cc

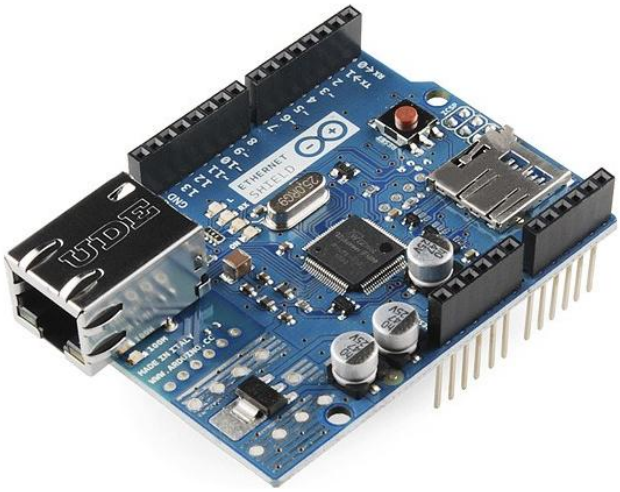


Figura 2 - *Shield* interface *Ethernet*

Fonte: http://www.labdegaragem.org

Figura 3 - *Shield* interface RS 485

Fonte: http://www.labdegaragem.org

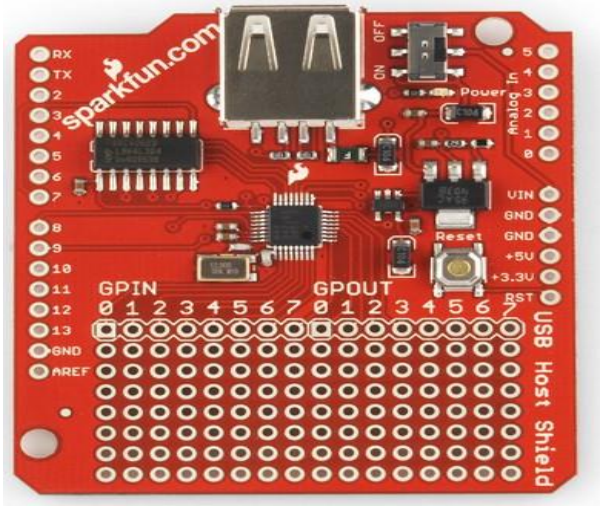


Figura 4 - *Shield* Interface USB

Fonte: http://www.labdegaragem.org

Neste sentido, este trabalho tem como proposta desenvolver uma plataforma mais completa, usando um microcontrolador de 32 *bits*, que possua os *hardware*s básicos encapsulados. Após várias pesquisas, escolheu-se o microcontrolador ARM NXP LPC1768 de 32 *bits* com 100 MHZ, o qual, além de incluir 512kb de memória FLASH, 32KB RAM, tem a qualidade de possuir uma série de interfaces do tipo Ethernet, USB CAN, SPI, I2C, ADC, DAC e PWM.

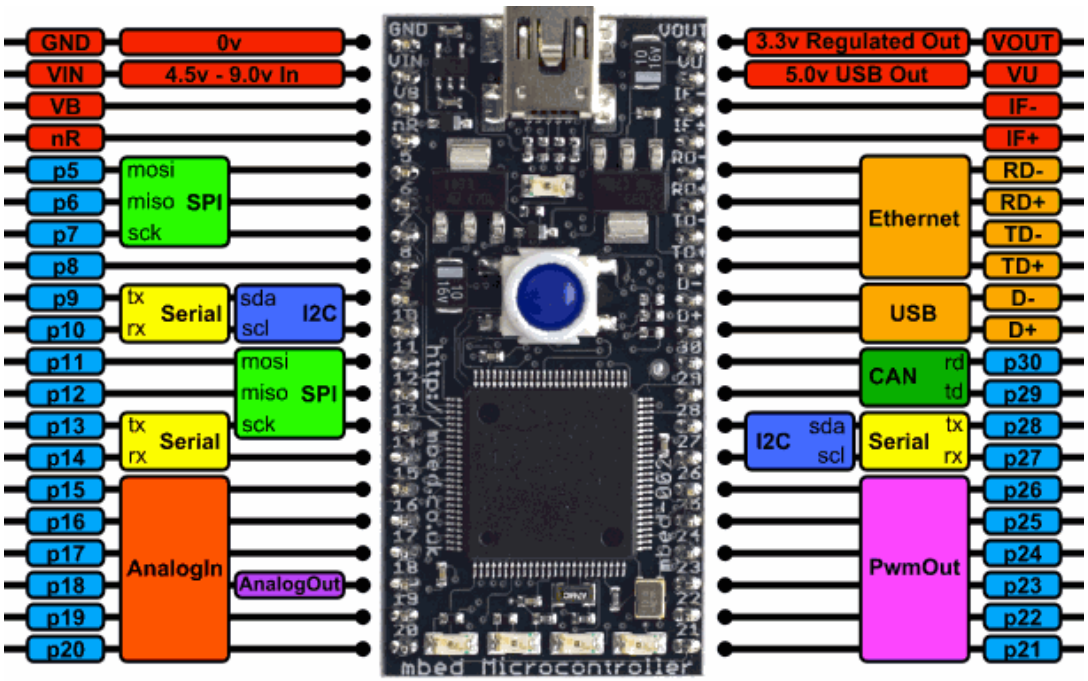


Figura 5 - Interfaces Microcontrolador ARM LPC1768

Fonte: MBED.org

Com a escolha do microcontrolador (vide Figura 5) foi possível o desenvolvimento de um *hardware* que possui, além da função de um CLP para trabalhos de automação residencial, a capacidade de ser um *kit* de desenvolvimento que poderá ser utilizado para fins acadêmicos de forma a promover uma ferramenta de simulação de outras aplicações dentro da área de domótica. Essa capacidade do projeto desenvolvido vem a cobrir uma lacuna dentro da área de Mecatrônica da UnB, trazendo um potencial de popularizar o uso de plataformas ARM dentro da comunidade de estudantes de Graduação e de Pós-Graduação dessa Universidade.

1.2 Justificativa do trabalho

Existem hoje no mercado muitas opções de equipamentos visando a automação de fábricas, shoppings, prédios, estádios e residências. Entretanto, esses equipamentos, em sua grande maioria, têm alto custo e uma arquitetura fechada. A proposta deste trabalho é a de reunir uma série de conceitos e tecnologias já existentes para desenvolver um *kit* de desenvolvimento de aplicações de automação com todas as características de um CLP, atingindo as normas básicas para esse tipo de aplicações.

Alguns itens específicos foram os principais motivadores para a criação desse equipamento, sendo incorporados à especificação básica do sistema a ser projetado:

a) Baixo custo;

b) Possibilidade de acrescentar módulos já existentes no mercado;

c) Formas de comunicação com outros equipamentos, *Ethernet*, CAN, *Xbee* e RS485;

d) Desenvolvimento fácil de *software* para aplicativos .

Historicamente, a automação residencial e a predial se restringiram a sistemas autônomos, destinados a comandar um dispositivo (ou outro sistema). Por exemplo, o sistema de ventilação, refrigeração, exaustão e aquecimento ambiental com ajustes predefinidos e sem qualquer integração com demais elementos das instalações prediais. Posteriormente, a mesma evoluiu para sistemas integrados, que podem ser definidos como aqueles que (por meio de um dispositivo) integram e controlam alguns equipamentos ou sistemas de uma instalação. Nesse sentido, esses sistemas apresentam uma atuação limitada pelo controlador (previamente determinada pelo fabricante), a exemplo das centrais de comando de *home theater*, em que sua atuação alcança, em geral, apenas os comandos de cortinas, iluminação de ambiente, além do comando de áudio e vídeo. (DIAS, César; PIZZOLATO, Nélio 2004).

Atualmente, as residências já podem dispor dos sistemas denominados complexos, os quais atuam não somente como controladores remotos, mas também como gerenciadores, com comunicação em “mão dupla” e retroalimentação de *status* entre os sistemas que compõem as instalações automatizadas, podendo, assim, ser personalizados de acordo com as exigências do cliente (LUIZ, Domingues Nélio).

1.3 Objetivos do trabalho

1.3.1 Objetivo geral

O objetivo deste trabalho é descrever o desenvolvimento de um CLP residencial de baixo custo, que vise atender uma demanda por equipamentos que façam o gerenciamento de projetos de automação residencial, de uma forma simples e eficiente.

1.3.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

1. Fazer a escolha de um microcontrolador que atenda os requisitos gerais de acordo com a concepção do projeto;
2. Desenvolver uma plataforma flexível a ser utilizada como CLP residencial de baixo custo, para aplicações em automação residencial e, ao mesmo tempo, desenvolver um *kit* de estudo para testes em aplicações de domótica, para o Laboratório de Eletrônica do GRACO-UNB;
3. Elaborar uma *placa-mãe* que possua os periféricos necessários para utilização dos principais interfaces do microcontrolador escolhido;
4. Permitir que o CLP desenvolvido seja compatível com vários *shields* existentes no mercado, que já possibilitem a expansão da capacidade de aplicações, tais como sensores de temperatura, pressão, movimento, fumaça, umidade entre outros;
5. Desenvolver uma placa de relés com interface *Xbee,* que permita ser utilizada remotamente para controlar equipamentos em várias partes de uma casa, além de demonstrar as possibilidades de criação de placas de expansão para o CLP desenvolvido;

6. Atender as principais normas técnicas para criação de equipamentos eletrônicos;
7. Criar do esquemático da placa base e placa de relés;
8. Escolher os componentes eletrônicos;
9. Seguir uma metodologia de elaboração do projeto baseada em prototipagem. Dessa maneira, serão elaborados vários protótipos do CLP a serem avaliados, para assim permitir a melhora do projeto, ou mesmo aprimorar a especificação do mesmo. Essa metodologia usada está baseada no modelo em espiral, usado em engenharia de *software* (PRESSMAN, Roger S. 2011).
10. Elaborar do *leiaute* das placas de circuito impresso;
11. Confecção da placa base e placa de relés;
12. Montagem dos componentes;
13. Testes eletrônicos dos *hardwares*;
14. Criação dos *softwares* para teste geral das interfaces, RS485, *ETHERNET*, CAN, RS232, RELÉS, *XBEE*.

1.4 Metodologia

Este projeto foi inicialmente idealizado para ser um CLP simples que tivesse as funções básicas para controle de uma residência, e, para tal tarefa, seguiu alguns critérios em sua concepção:

a) Estudo do problema;
b) Estudo sobre a domótica e suas utilizações no mercado atual;
c) Escolha do escopo do *hardware*;
d) Escolha do processador;
e) Escolha das interfaces de comunicação;
f) Criação do esquemático (para cada protótipo);
g) Criação do *layout* da placa (para cada protótipo);
h) Soldagem dos componentes (para cada protótipo);

i) Teste eletrônico (para cada protótipo);

j) Criação dos *softwares*;

k) Teste dos *softwares*.

Durante o processo de criação, o conceito de um CLP simples foi estendido para o desenvolvimento de uma plataforma de “sistema embarcado”, dedicado a tarefas específicas, como para otimizar o projeto reduzindo tamanho, recursos computacionais e custo do produto para que não tivesse somente a funcionalidade de um CLP e sim fosse um *kit* de desenvolvimento ARM 32 *bits*, que simulasse um CLP com todas suas funcionalidades e que permitisse várias outras aplicações e modularidades.

Com este conceito foram desenvolvidas quatro versões do projeto com protótipos que foram sendo alterados conforme novas solicitações de recursos. Essas versões serão detalhadamente explicadas em um capítulo específico. Tendo em conta que a tarefa de se desenvolver um CLP é altamente complexa, o projeto de interface através de uma linguagem de programação do tipo SFC (Função Gráfica de Sequenciamento), IL (Lista de Instruções), ST (Texto Estruturado), FBD (Diagrama Gráfico de Funções) e LD (Diagrama Ladder) foi deixada para um trabalho futuro.

1.5 Estrutura deste trabalho

Este trabalho se divide em quatro capítulos, mais referências bibliográficas e três anexos gerais.

No capítulo 2, é mostrada uma revisão bibliográfica, contemplando os estudos e seus autores, sem a necessidade de aprofundar-se no detalhamento dos conceitos e na descrição dos equipamentos.

O capítulo 3 destacará a metodologia utilizada para o teste-projeto.

Já no capítulo 4, entrar-se-á na construção eletrônica do CLP residencial, na qual será detalhada toda a criação dos circuitos eletrônicos, dos componentes

escolhidos, e haverá uma breve explicação sobre a arquitetura 32 *bits* dos microcontroladores ARM que será usada como núcleo do projeto.

O capítulo 5 abordará as concussões e propostas para trabalhos futuros.

Já o apêndice A apresenta um breve manual da plataforma Graco 1.4.

Temos no apêndice B uma breve visão geral sobre a configuração dos módulos XBEE.

E, finalizando, no apêndice C teremos uma apresentação mais detalhada da placa de relés.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Domótica como conceito de automação residencial

Em geral, a *domótica* trata da utilização simultânea da eletricidade, da eletrônica e das tecnologias da informação no ambiente residencial, permitindo assim realizar a sua gestão, local ou remota, oferecendo uma vasta gama de aplicações nas áreas de segurança, conforto, comunicações e gestão de energia (MARIOTONI; ANDRADE, 2007).

Uma definição bastante interessante e que a domótica agrega também a ideia de integração de sistemas residenciais, a qual pode ser resumida pela seguinte frase: “É um processo que, usando diferentes soluções e equipamentos, possibilita ao usuário usufruir o máximo de qualidade de vida na sua habitação” (Muratori, José Roberto 2013).

A forma como a domótica está evoluindo rapidamente nos permite prever que, na sequência, a próxima fase de desenvolvimento, é a junção com o conceito atual “da internet das coisas” (IOT – *Internet of Things*). Essa transformação levará à maior expansão da domótica, trazendo efeitos abrangentes, em todos os setores e em todas as vidas cotidianas (Miori, V.; Russo, D. 2014).

Embora haja o atual crescimento da domótica, a missão de difundir os benefícios da automação residencial ainda não é uma tarefa fácil, principalmente para os mais céticos. Muitos daqueles conceitos de automação vistos como futuristas são hoje utilizados com naturalidade por muitas famílias brasileiras. Os sistemas estão cada vez mais acessíveis e as pessoas, no mínimo, já ouviram falar a respeito, seja por meio da mídia ou por algum conhecido que já possui um sistema instalado em sua residência. (Muratori, José Roberto 2013).

Uma residência inteligente contém um sistema para gerenciar todo tráfego de informação, bem como um outro de controle dos equipamentos, permitindo maior conforto com menor gasto de energia (BOLZANI, 2004b), agregando, assim, vários ambientes inteligentes, sendo que os mesmos não necessariamente correspondem

ao espaço físico delimitado pelas paredes que cercam o ambiente em si. Neste contexto, sala de estar, quartos, cozinha e sala de jantar, por exemplo, são ambientes que promovem muita interação, devido ao fato de que os mesmos têm um alto tráfego de pessoas, agregando muito sensores e atuadores. Desta forma, para se otimizar a relação entre usuário e equipamentos, é importante dividir o espaço físico em subespaços, proporcionando assim setores diferenciados com relação ao controle dos sistemas e padrões de sensoriamento (BOLZANI, 2004).

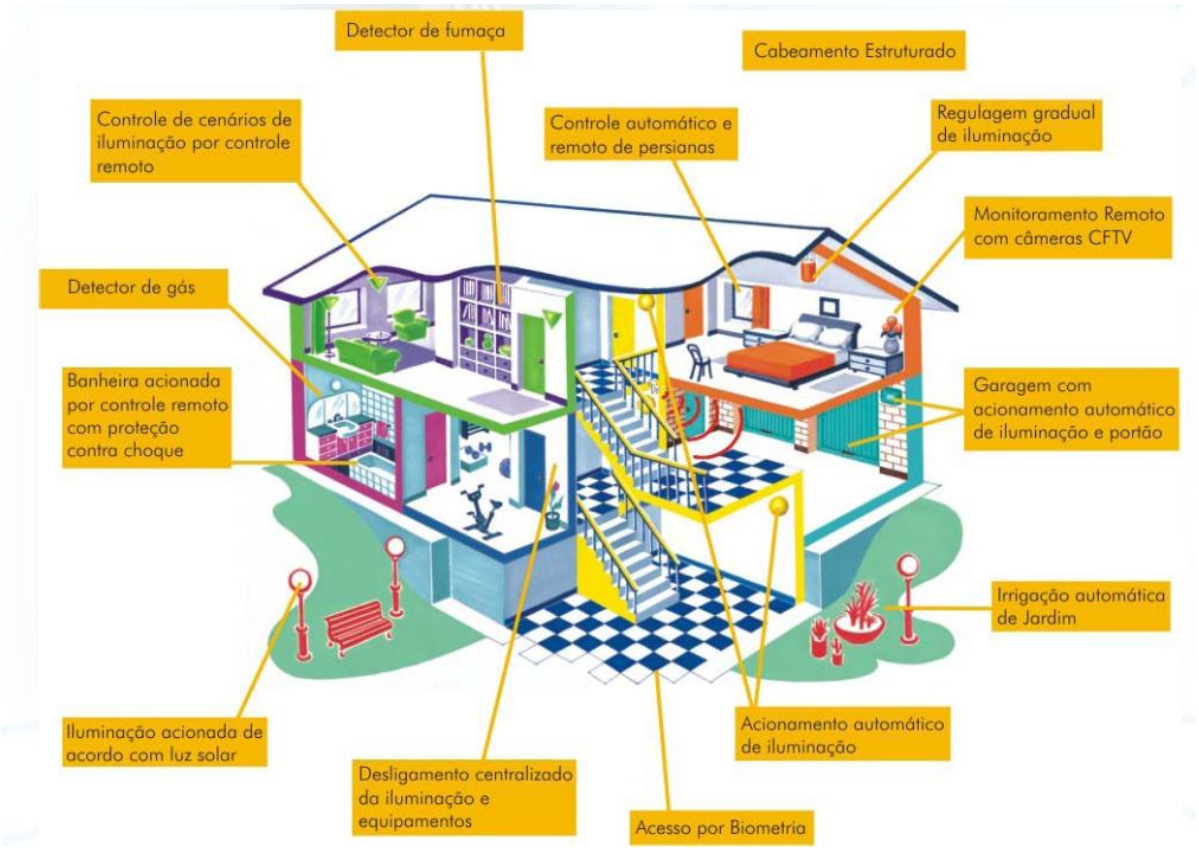


Figura 6 - Casa Inteligente

Fonte: Finder (2013)

Na figura: Palavras com iniciais minúsculas, nos retângulos laranja => **estruturado**, **remoto**, **jardim**, **biometria**.

Na Figura 6, pode ser observado um exemplo de possibilidades de automatização de uma residência através de um CLP, onde poderia haver as seguintes opções:

a) Sensores de incêndio distribuídos em todos os cômodos;
b) Sensores de presença em áreas principais, trabalhando em conjunto com o sistema de alarme da casa;

c) Atuadores na banheira que poderiam, através de um simples relé, ligar o equipamento em uma programação já pré-definida;
d) Sensores e atuadores que poderiam simular a presença de pessoas na casa em momento de viagem, através da ligação da iluminação em cômodos pré-determinados, ligação de aparelhos eletrônicos na cozinha;
e) Sensores e atuadores que trabalhariam em conjunto para irrigação do jardim;
f) Acesso à casa através da biometria ;
g) Acesso controlado da garagem com ativação remota;
h) Fechamento automático das persianas com a detecção de cenários, como a presença do sol e uma determinada temperatura desejada.

Um conceito ainda em construção, chamado de “Internet das coisas” (IoT), vem criando várias possibilidades de novos cenários, agregando o fato de cada dispositivo de uma casa (que se conecta à Internet) tem atribuído um endereço IP. Neste contexto, foi previsto que, até o ano 2020, haverá milhares de dispositivos habilitados para a Internet, em um mundo com uma população mundial prevista, de 7,6 bilhões de pessoas. Considerando-se que, de acordo com uma pesquisa de ambientes urbanos, cada ser humano é cercado de 1000 a 5000 objetos rastreáveis (J. B. Waldner, 2007), a mudança para o IPv6 é necessária, uma vez que fornece um número quase inimaginável de endereços IP (18 quintilhões de blocos de 18 quintilhões de possíveis endereços). Neste cenário, a domótica deve envolver o controle e o acompanhamento de eletrodomésticos em um sistema unificado. Tais sistemas de controle incluem iluminação, climatização, sistemas de segurança e até mesmo eletrodomésticos (Miori, V.; Russo, D. 2014).

Há muitos tipos diferentes de sistemas de automação residencial disponível. Esses sistemas são normalmente concebidos e adquiridos para fins diferentes. De fato, um dos principais problemas é que tais sistemas diferentes não são nem interoperáveis nem interligados. Estes vão desde simples interruptores de luz de controle remoto aos dispositivos de rede totalmente integrados, que controlam todos os aparelhos de uma casa inteira.

Neste momento, a domótica é uma aplicação extremamente interessante para o conceito de “internet das coisas”. Este fato prevê um ambiente futuro em que os sensores embutidos e atuadores serão autoconfiguráveis, podendo ser controlados remotamente através da Internet e permitindo, assim, uma variedade de aplicações de monitoramento e controle. Tais capacidades de comunicação muitas vezes são oferecidas pelos fabricantes de sistemas de domótica, embora sem compatibilidade com a Internet. Na verdade, cada fabricante produz o seu próprio dispositivo sem seguir um padrão que permita a comunicação com outros dispositivos de outros fabricantes. (Miori, V.; Russo, D. 2014)

Hoje em dia, muitos sistemas de domótica raramente são interoperáveis e, portanto, não permitem que os consumidores escolham os dispositivos de acordo com as suas necessidades ou outros critérios relevantes, tais como custo, desempenho, tendências e confiança. A atual falta de interoperabilidade é devida ao fato de que a prática de mercado atual liga eficazmente os consumidores a tecnologias proprietárias, forçando-os a comprar apenas os dispositivos que estejam em conformidade com o sistema de um fabricante específico. Até o momento, nenhuma tecnologia domótica surgiu, de fato, como um padrão na indústria (Miori, V.; Russo, D. 2014).

2.2 Classificação dos sistemas domóticos

A arquitetura de um sistema domótico, como em qualquer sistema de controle, especifica a forma como os diferentes elementos do sistema vão se localizar. Neste sentido, existem duas arquiteturas básicas: (a) a arquitetura centralizada; e (b) a arquitetura distribuída, as quais serão descritas a seguir.

2.3 Arquitetura centralizada

No sistema centralizado, existe um equipamento que controla todas as entradas e saídas dos sensores e atuadores, seguindo uma programação determinada (vide Figura 7).

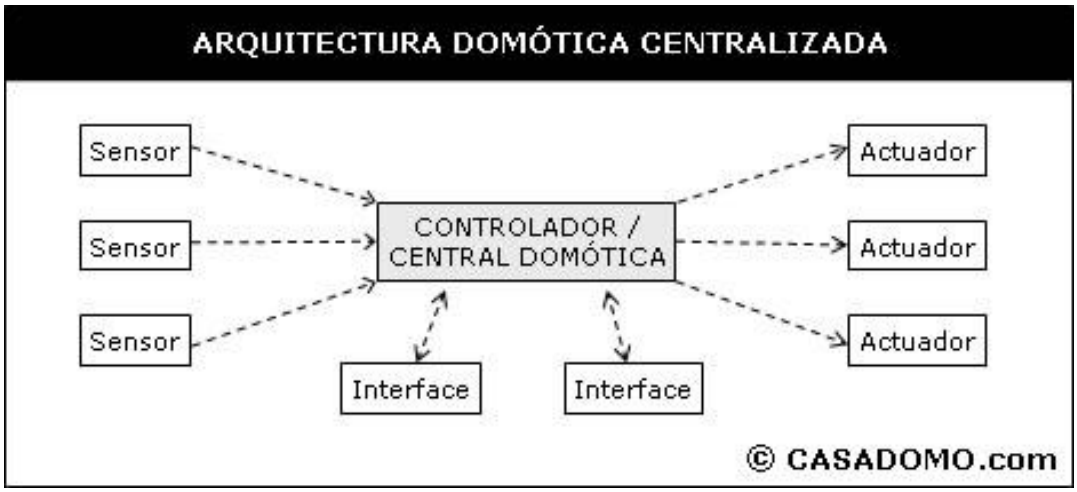


Figura 7 – Arquitetura Centralizada
Fonte Casadomo.com

As vantagens das arquiteturas centralizadas estão relacionadas principalmente ao custo, sendo um fator a considerar nos casos em que uma residência é preparada para receber um projeto de domótica. Nesse cenário, o projeto já consta de um cabeamento estruturado e de pontos de passagem de cabos, cujo custo de instalação de sensores é baixo. Entretanto, se for necessário

criar toda essa estrutura, o projeto pode ficar caro. Nesse contexto adverso, o projeto proposto nesta dissertação pode ser aplicado a uma arquitetura centralizada, usando uma rede sem fio para melhorar os custos de implantação do mesmo.

Entretanto, as arquiteturas centralizadas têm algumas desvantagens. Por exemplo, a centralização das funções pode ser um problema no caso do CLP falhar. Nesse caso, todas as operações controladas por ele ficam comprometidas, além da necessidade de uma grande quantidade de cabeamento, fato este que está sendo aos poucos superado pela utilização de sensores e atuadores *wireless,* tais como aqueles que utilizam a tecnologia *zIgbee* (LUGLI, Alexandre; SOBRINHO, Darlan 2013).

2.4 Arquitetura descentralizada

Em um sistema domótico, com arquitetura descentralizada, existem vários controladores interconectados por uma rede, que trocam informação entre eles enviando sinais de controle aos atuadores e informações às interfaces conectadas aos controladores (dependendo do aplicativo). Adicionalmente, o conjunto de controladores define estratégias e ações, dependendo da informação que recebe dos sensores, do conjunto de sistemas interconectados e das ações dos usuários (CASADOMO, 2008). Um exemplo dessa arquitetura pode ser visto na Figura 8.

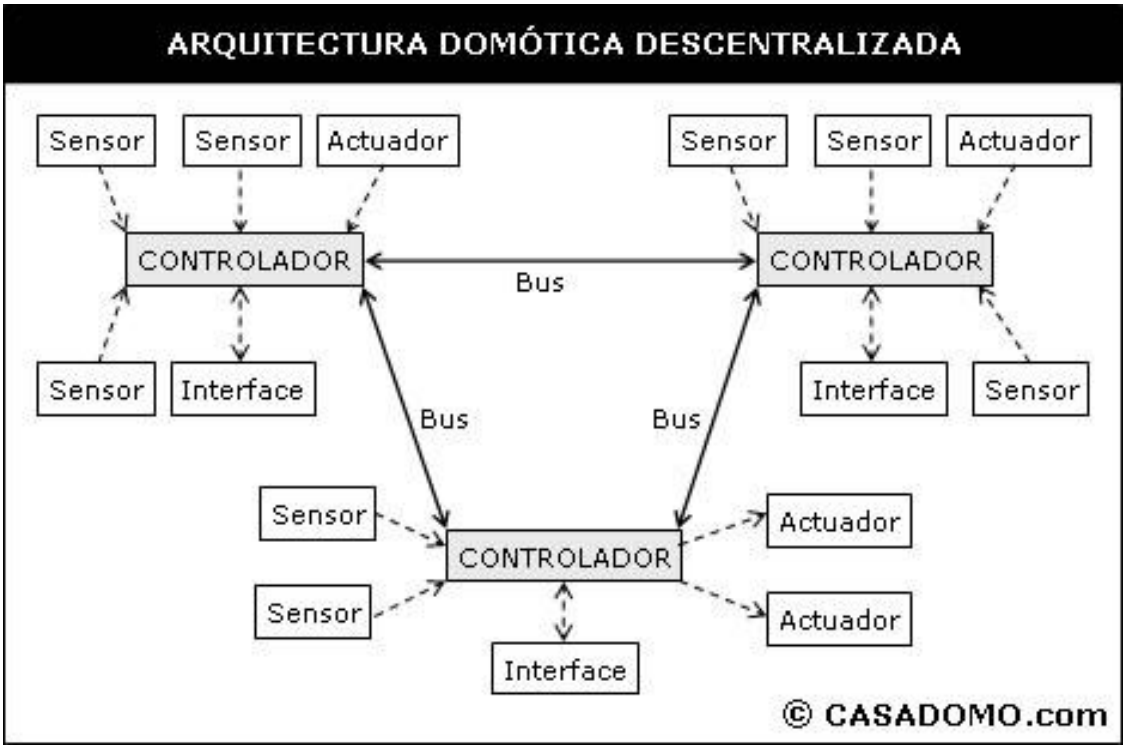


Figura 8 – Arquitetura Descentralizada

Fonte: CASADOMO (2008)

As arquiteturas descentralizadas têm algumas vantagens, se comparadas com as centralizadas. Neste caso, pode-se facilmente identificar e isolar possíveis falhas no sistema. Por exemplo, se o controlador das câmeras parar de funcionar, o mesmo não altera a continuidade do funcionamento dos outros sistemas controlados. Também nesse caso, vários CLPs podem trabalhar em rede e facilitar o uso do sistema, assim como sua robustez e sua confiabilidade. Além disso, os sistemas descentralizados têm outras desvantagens. Dentre elas, podem ser citadas:

a) Vários CLPS instalados têm impacto direto no custo do projeto;
b) Em caso de falhas, é preciso procurá-las em vários pontos para isolar o problema;
c) Programação complexa para direcionar uma entrada/saída para um novo CLP no sistema.

2.5 Segmento de automação residencial

Os projetos de automação residencial e empresarial valorizam o estilo de vida do indivíduo, proporcionando, assim, maiores conforto e comodidade. Aliando tecnologia na fronteira do conhecimento com a automatização de sistemas, a automação residencial pode superar as expectativas na elaboração de projetos especiais personalizados, de acordo com a necessidade de cada pessoa (ABDI, 2009).

O impacto do segmento de automação residencial no mercado pode ser considerado ainda recente no Brasil, sendo ainda muito ligado ao setor de serviços, com um número reduzido de empresas nacionais associadas, com poucos valores divulgados para uma análise de mercado, mas em crescente e forte evolução (ABDI, 2009).

Como acontece com qualquer processo de inovação tecnológica, concepção, produção e operação dos sistemas de automação residencial ainda se ressentem de profissionais adequadamente formados e de regras e práticas claras, assim como nas relações projetista-cliente e fabricante-cliente. Os possíveis usuários estão cada vez mais exigentes, exigindo cada vez mais qualidade dos produtos. Em geral, os sistemas vendidos na maior parte das residências têm muito mais *marketing* do que eficiência e utilidade (ABDI, 2009).

Os profissionais que atuam em projeto e manutenção de sistemas ou são egressos da área comercial das empresas fabricantes com foco nos aspectos comerciais, ou da área de automação industrial, com foco nos aspectos técnicos do problema (ABDI, 2009). Tudo isso é devido à inexistência de cursos de formação específica, os quais deveriam integrar conhecimentos de mecânica, instalações prediais, elétrica, eletrônica e instrumentação.

Finamente, do ponto do vista de manutenção de sistemas de automação residencial, pode ser observado que, na maioria das casas "inteligentes", a manutenção dos sistemas de automação acaba sendo realizada pelo próprio fabricante (ABDI, 2009).

2.6 Benefícios oferecidos pelos sistemas domóticos

No contexto atual da domótica, podem ser citadas as seguintes vantagens (ABDI, 2009):

a) **Conservação de Energia:** Através da melhor precisão no sistema de comando, com uso da eletrônica e da automatização dos procedimentos, tem-se a eliminação de desperdícios. Além disto, rotinas de tarefas e novos métodos de controle permitem processos mais eficientes;

b) **Conforto Ambiental:** A própria otimização das rotinas de controle e o aumento da precisão do sistema possibilita redução dos tempos de resposta, o que proporciona a manutenção dos ambientes dentro dos parâmetros de controle;

c) **Supervisão Residencial:** Toda sua instalação pode ser resumida na tela de um computador (*on-line*), com telas gráficas, alarmes e relatórios emitidos e/ou arquivados automaticamente. Isso constitui poderosa ferramenta para diagnose preventiva e corretiva de problemas;

d) **Confiabilidade:** A operação, o controle dos sistemas, a identificação e a visualização de defeitos passam a ser automáticos, ficando independentes de falha humana.

As instalações domóticas automatizadas apresentam inúmeras vantagens sobre as convencionais, como, por exemplo:

a) **Flexibilidade:** Possibilidade de alteração das características operacionais do sistema por mudanças nos *software*s, em vez de correções ou atualizações, tornando possíveis alterações substanciais nas maneiras de funcionamento, com baixíssimo ou até nenhum custo (ABDI, 2009);

b) **Dimensões:** Compactação dos elementos de comando e acionamento de sistemas elétricos, através do uso da microeletrônica e da eletrônica de potência, valorizando o espaço e sua ocupação, bem como minimizando possibilidades de interferência física entre as diversas instalações (ABDI, 2009);

c) **Confiabilidade:** Menor desgaste dos elementos devido à substituição de mecanismos eletromecânicos por eletrônicos (estáticos – ou chaves estáticas);

d) **Funções Intrínsecas:** Reaproveitamento de experiências de mercado e otimização do tempo de desenvolvimento de soluções, graças à utilização de soluções pré-formatadas e à facilidade de reuso de métodos computacionais (ABDI, 2009);

e) **Precisão:** Sistemas eletrônicos e com *feedback* (operando em malha fechada), permitindo maior precisão e confirmação das ações, resultando em eficiência energética, qualidade do produto final, conforto, entre outros aspectos (ABDI, 2009);

f) **Supervisão e Gerenciamento Centralizado:** Utilização das ferramentas de teleinformática (interfaces microprocessadas, métodos computacionais e redes de dados) e de processamento distribuído, proporcionando o total gerenciamento das instalações através de computadores (ABDI, 2009).

2.7 Aspectos gerais sobre CLPs (Controladores Lógico Programáveis)

Um controlador lógico programável, segundo a ABNT (Associação Brasileira de Normas Técnicas), é um equipamento eletrônico digital com *hardware* e *software* compatíveis com aplicações industriais. Já segundo a NEMA (*National Electrical Manufacturers Association*), um CLP é um aparelho eletrônico digital que utiliza uma memória programável para armazenar internamente instruções, implementando funções específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, controlando, por meio de módulos de entradas e saídas, vários tipos de máquinas ou processos, vide Figura 9.

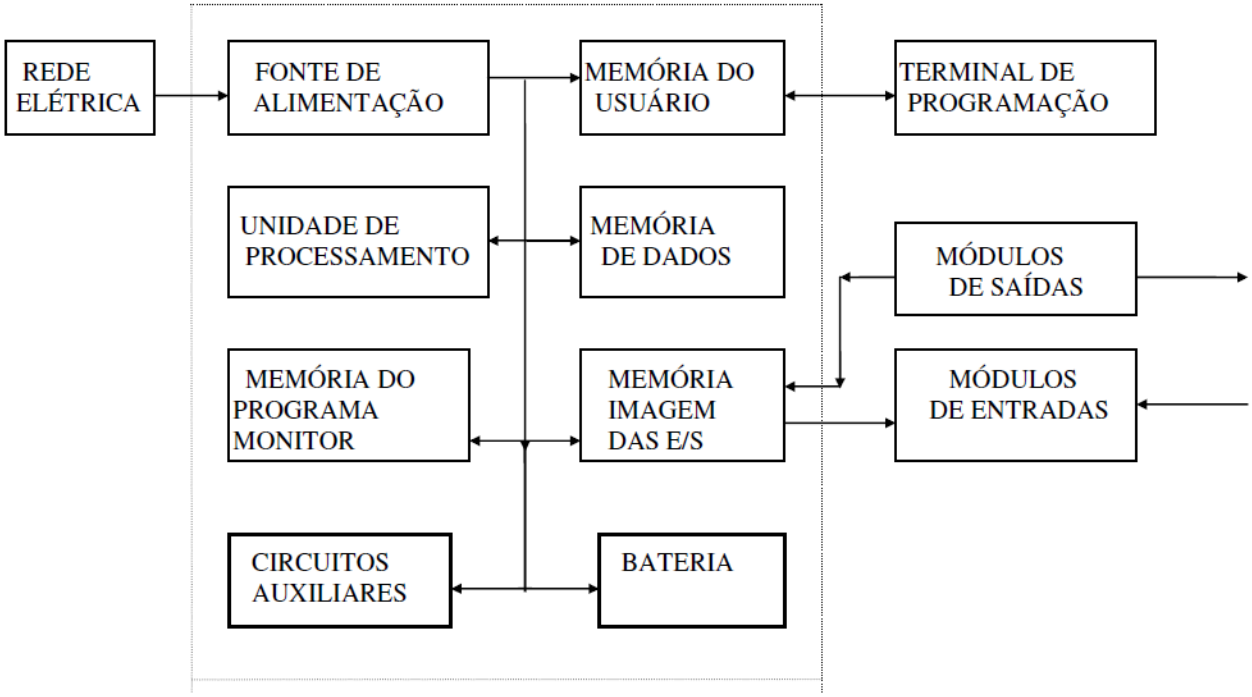


Figura 9 - Estrutura interna de um CLP

Fonte: CASADOMO (2008)

Historicamente, os CLPs foram projetados para uso em automação de fábricas, quando a operação requeria tarefas muito rápidas, repetitivas, tais como acontecem nas linhas de montagem. Um CLP é eficiente para executar operações de sequenciamento, atividades de alarme e de intertravamento. O controle em tempo real para intertravar motores e equipamentos relativos tornou-se muito prático dentro do uso de CLPs no contexto de controle de processos (Ribeiro, 2005).

CLPs estão muito difundidos nas áreas de controle de processos ou de automação industrial. No primeiro caso, a aplicação se dá nas indústrias do tipo contínuo, produtoras de líquidos, materiais gasosos e outros produtos. No segundo, ocorre nas áreas relacionadas com a produção em linhas de montagem, por exemplo, na indústria automotiva (Ribeiro, 2003). Em um sistema de automação típico, toda a informação dos sensores é concentrada no controlador (CLP) que, de acordo com o programa em memória, define o estado dos pontos de saída conectados a atuadores.

2.8 História das tecnologias baseadas em CLPs

A ideia inicial do CLP foi a de ser de um equipamento com as seguintes características resumidas:

a) Facilidade de programação;
b) Facilidade de manutenção com conceito *plug-in*;
c) Alta confiabilidade;
d) Dimensões menores que painéis de relês, para redução de custos;
e) Envio de dados para processamento centralizado;
f) Preço competitivo;
g) Expansão em módulos;
h) Mínimo de 4000 palavras na memória.

Os CLPs podem ser divididos historicamente, de acordo com o sistema de programação por eles utilizado:

1ª Geração: Os CLPs de primeira geração se caracterizam pela programação intimamente ligada ao *hardware* do equipamento. A linguagem utilizada era o *assembly*, que variava de acordo com o processador utilizado no projeto do CLP; ou seja, para poder programar era necessário conhecer a eletrônica do projeto do CLP (SENAI –ES,1999).

Assim a tarefa de programação era desenvolvida por uma equipe técnica altamente qualificada, gravando-se o programa em memória EPROM, sendo realizada normalmente no laboratório junto com a construção do CLP.

2ª Geração: Aparecem as primeiras “Linguagens de Programação” não tão dependentes do *hardware* do equipamento, possíveis pela inclusão de um “programa monitor” no CLP, o qual converte as instruções do programa em comandos de baixo nível, verifica o estado das entradas, compara com as instruções do programa do usuário, e altera o estado das saídas.

Os “terminais de programação” (ou maletas, como eram conhecidas) eram, na verdade, programadores de memória EPROM. As memórias, depois de programadas, eram colocadas no CLP para que o programa do usuário fosse executado (SENAI –ES,1999).

3ª Geração: Os CLPs passam a ter uma “entrada de programação”, na qual um teclado ou programador portátil é conectado, podendo assim alterar, apagar, gravar o programa do usuário, além de realizar testes (*debug*) no equipamento e no programa. A estrutura física também sofre alterações sendo a tendência para os “sistemas modulares com bastidores” ou *racks* (SENAI–ES,1999).

4ª Geração: Com a popularização e a diminuição dos preços dos microcomputadores (normalmente clones do IBM PC), os CLPs passaram a incluir uma entrada para a comunicação serial. Com o auxílio dos microcomputadores, a tarefa de programação passou a ser realizada nestes. As vantagens eram a utilização de várias representações das linguagens, possibilidades de simulações e testes, treinamento e ajuda por parte do *software* de programação, possibilidade de armazenamento de vários programas no micro, etc (SENAI –ES,1999).

5ª Geração: Atualmente, existe uma preocupação em padronizar protocolos de comunicação para os CLPs de modo a proporcionar que o equipamento de um fabricante “converse” com o equipamento de outro fabricante, não só CLPs, como Controladores de Processos, Sistemas Supervisórios, Redes Internas de Comunicação e etc, proporcionando uma integração que facilite automação, gerenciamento e desenvolvimento de plantas industriais mais flexíveis e normalizadas, fruto da chamada globalização (SENAI–ES,1999).

2.9 Vantagens do uso de CLPs

São inúmeras as vantagens da utilização de um CLP (controlador lógico programável). Entre elas, pode-se citar (RIBEIRO, 2005):

a) São relativamente fáceis de serem programados;
b) Possuem uma boa confiabilidade de operação;
c) Permitem uma maior velocidade na elaboração de projetos de utilização do sistema;
d) Possuem boa flexibilidade;
e) Consomem pouca energia.

2.10 Desvantagens de uso de CLPs

Algumas desvantagens dos CLPs são:

a) Necessidade do conhecimento técnico elaborado para sua programação;
b) Apesar da possibilidade de seus sensores e atuadores se comunicarem via *wireless*, devido ao alto custo ainda se usa muito o tradicional cabeamento;
c) Dependendo da arquitetura escolhida, a centralização em um único ponto de controle dá margens de risco de toda operação parar, devido a falhas eletrônicas no equipamento (RIBEIRO, 2005).

2.11 Módulos de entradas e saídas para tecnologia CLPs

Os CLPs são classificados pelo seu porte em função do número de pontos de entrada e saída. Os de pequeno porte possuem até 128 pontos de entrada e saída;

os de médio porte, entre 128 e 512 pontos de entrada e saída; e os de grande porte possuem mais de 512 interfaces de entrada e saída (BRUNE, 2005).

Uma interface de entrada é considerada aquela em que um sinal é recebido a partir de um dispositivo ou componente externo (SILVEIRA e SANTOS, 2002). O ponto de entrada pode ser digital ou analógico. Entradas digitais são aquelas que possuem apenas dois estados possíveis: ligado ou desligado (SILVEIRA; SANTOS, 1998). Alguns dos exemplos de dispositivos que podem ser ligados a elas são.

a) Botoeiras;
b) Chaves (ou micro) fim de curso;
c) Sensores de proximidade indutivos ou capacitivos;
d) Chaves comutadoras;
e) Termostatos;
f) Pressostatos;
g) Controle de nível (boia).

As entradas digitais são fáceis de manipular por possuírem apenas dois estados (ligado e desligado). As analógicas, no entanto, podem ter infinitos estados dentro de uma faixa determinada. Existe uma grande gama de sinais de entradas padronizados: +/- 12,5 mV; +/- 50 mV; +/- 500 mV; +/- IV; +/- 5 V; +/- 10 V; l a 5V; 0 a 5 V; 0 a 10 V; 0 a 20 mA; 4 a 20 mA (KOPELVSKI, 2010).

Um ponto de saída pode ser analógico ou digital, sendo que as saídas digitais possuem apenas dois estados, enquanto as saídas analógicas possuem mais de dois estados. O número de estados depende do número de *bits* usado pelo conversor D/A. Por exemplo, um conversor D/A de 8 *bits* permite 256 estados (níveis de tensão) na saída. Os pontos de saída digitais podem ser implementados por relés, transistores, ou, ainda, por SCR e TRIAC. Esses dispositivos são usados para acionar lâmpadas, motores, solenoides, válvulas, etc. Os pontos analógicos de saída podem fornecer vários níveis de corrente como, por exemplo, de 4 a 20mA, bem como de tensões, citando o valor de 0 a 10V (MORRISS, 1995).

2.12 Estrutura básica de um CLP

Conforme (SILVA, 2008), a estrutura básica de um CLP é a seguinte:

a) **Fonte de alimentação:** Converte a tensão da rede de 110 ou 220 VCA em +5VCC, +12VCC ou +24 VCC para alimentar os circuitos eletrônicos, as entradas e as saídas;

b) **Unidade de processamento:** Também conhecida por CPU, é composta por microcontroladores ou microprocessadores (Intel 80xx, Motorola 68xx, PIC 16xx). Endereçamento de memória de até 1MB, velocidades de *clock* de 4 a 30 MHz, manipulação de dados decimais, octais e hexadecimais;

c) **Bateria:** Utilizada para manter o circuito do relógio em tempo real. Normalmente são utilizadas baterias recarregáveis do tipo Ni - Ca;

d) **Memória do programa supervisor:** O programa supervisor é responsável pelo gerenciamento de todas as atividades do CLP. Não pode ser modificado pelo usuário e fica normalmente em memórias do tipo PROM, EPROM, EEPROM;

e) **Memória do usuário:** Espaço reservado ao programa do usuário. Constituída por memórias do tipo RAM ou EEPROM. Também é possível a utilização de cartuchos de memória para proporcionar agilidade e flexibilidade;

f) **Memória de dados:** Armazena valores do programa do usuário, tais como valores de temporizadores, contadores, códigos de erros, senhas, etc. Nessa região se encontram também as memórias de imagem de entrada e a saída. Esta funciona como uma tabela virtual em que a CPU busca informações para o processo decisório;

g) Os circuitos auxiliares atuam em caso de falha do CLP, e são os seguintes:

   a. ***POWER ON RESET*:** desliga todas as saídas assim que o equipamento é ligado, isso evita que possíveis danos venham a acontecer;

b. ***POWER DOWN*:** monitora a tensão de alimentação salvando o conteúdo das memórias antes que alguma queda de energia possa acontecer;
c. ***WATCH DOG TIMER*:** deve ser acionado em intervalos periódicos, isso evita que o programa entre em *loop* ou trave.

2.13 Tipos de CLPs

De acordo com a sua necessidade e complexidade de utilização os CLPs se dividem em de grandes, médios e pequeno porte, e possuem uma variedade grande de fabricantes.

2.13.1 CLPs de grande porte

Plataformas ideais para as mais sofisticadas soluções em automação, tais como controle de processos com grande volume de dados ou gerenciamento de sistemas de manufatura, que exijam um alto nível de desempenho, velocidade e capacidade de memória, aliando processadores de alta performance, permitindo as mais variadas e complexas soluções em automação (WEG, 2008). Esses tipos de equipamentos possuem grande capacidade de armazenamento de programa e interfaces de comunicação integradas. Alguns desses CLPs podem operar em regime de multiprocessamento, trabalhando juntos para garantir maior desempenho. Em alguns desses equipamentos, os módulos podem ser substituídos, com o sistema em operação, garantido assim uma grande disponibilidade do sistema (SIEMENS, 2008).

2.13.2 CLPs de médio porte

Os CLPs desta categoria caracterizam-se em geral por suas dimensões menores que os da categoria anterior e pela excelente relação custo-benefício. Sendo, sobretudo, equipamentos idealizados para aplicações de pequeno e médio porte em tarefas de intertravamento, temporização, contagem e operação matemáticas, substituem com vantagens contadores auxiliares, temporizadores e contadores eletromecânicos, reduzindo o espaço necessário e facilitando significativamente as atividades de manutenção. Em muitos desses equipamentos, é possível a comunicação via *modem*, permitindo que seja feita de forma rápida e fácil, a assistência técnica remota (SIEMENS, 2008a).

Esses dispositivos representam CLPs rápidos, oferecendo um excelente comportamento em tempo real, garantindo maior qualidade, eficiência e confiabilidade ao processo. Adicionalmente, o mesmos têm uma concepção modular, permitindo assim que as soluções possam ser desenvolvidas sob medida e ampliadas conforme a demanda. Os mesmos possuem diferentes níveis de memória e diferentes números de entradas e saídas integradas, tendo disponíveis uma vasta gama de módulos de expansão para diversas funções.

2.13.3 CLPs de pequeno porte

Os CLPs de pequeno porte têm crescido em importância no mercado, principalmente em aplicações simples nas quais há demanda por baixo custo. Também podem ser utilizados como E/S remoto (inteligente ou não) em aplicações distribuídas, mas quais cada ilha de automação possui baixa complexidade, embora o processo como um todo possa ter complexidade elevada.

Em alguns casos, CLPs de pequeno porte podem ter recursos bastante avançados em termos de desempenho e capacidade de comunicação. Existem

aplicações que, embora requeiram um baixo número de pontos de E/S, exigem tais recursos avançados.

2.13.4 Arquitetura compacta X modular

CLPs de grande e médio porte geralmente possuem arquiteturas modulares, ou seja, os pontos de E/S podem ser expandidos na forma de módulos de E/S distribuídos localmente (E/S local) ou através de redes de campo (E/S remoto). Entretanto, nada impede que existam CLPs de pequeno porte modulares, e de fato eles podem ser encontrados no mercado. CLPs de pequeno porte, não obstante, para atingir metas de baixo custo e dimensões reduzidas, muitas vezes são oferecidos em arquiteturas compactas. Isto é, com todos os elementos em um único chassi (por exemplo, fonte de alimentação, CPU, interfaces de comunicação e programação e pontos de E/S). Neste caso, o CLP é acompanhado de um número fixo e não expansível de pontos de E/S.

2.14 Entradas e saídas

Para um CLP se comunicar com seus sensores e atuadores, o mesmo precisa de meios de receber ou enviar sinais para serem processados pela CPU, sinais estes que podem ser de entrada ou saída (KOPELVSKI, 2010). Neste caso, a CPU só consegue trabalhar com sinais digitais, que são sequências de *bits* ligados e desligados. Isso é representado por 0 para um *bit* desligado e 1 para um *bit* ligado. Através destes dois estados, pode-se, com o uso de codificação, representar qualquer número, caractere ou função (KOPELVSKI, 2010).

2.14.1 Entradas digitais

As entradas digitais convertem cada elemento do campo em um *bit*. Existem entradas digitais de dois tipos: com alimentação interna e externa. As entradas com alimentação interna fornecem uma tensão de alimentação que deve passar pelo elemento do processo e retornar à entrada. Para acionar este tipo de entrada basta um contato não alimentado (KOPELVSKI, 2010).

2.14.2 Entradas analógicas

As entradas digitais são fáceis de manipular por possuírem apenas dois estados (ligado e desligado). As analógicas, no entanto podem ter infinitos estados dentro de uma faixa determinada. Existe uma grande gama de sinais de entradas padronizados: +/- 12,5 mV; +/- 50 mV; +/- 500 mV; + /- IV; +/- 5 V; +/- 10 V; l a 5V; 0 a 5 V; 0 a 10 V; 0 a 20 mA; 4 a 20 mA. (KOPELVSKI ,2010)

2.14.3 Saídas

As unidades de saída fornecem as conexões entre os dispositivos de campo e a unidade central de processamento. Essas interfaces podem ter um ou mais canais, fornecendo sinais digitais ou analógicos (ATOS, 2010). Da mesma forma, as saídas também são divididas em duas categorias, de acordo com os tipos de sinais que manipulam: digitais e analógicas, e também possuem uma representação na tabela de imagens de entradas e saídas.

2.14.4 Saídas digitais

Os tipos e níveis de sinais das saídas digitais são os mesmos daqueles das entradas. As saídas digitais convertem os sinais digitais fornecidos pela CPU em sinais capazes de energizar cargas, tais como:

a) Contatores;
b) Solenoides;
c) Relés;
d) Lâmpadas;
e) Sirenes.

A comutação executada por uma unidade de saída pode ser através de transistores (em corrente contínua), através de TRIACs (em corrente alternada) ou através de relés (corrente contínua ou alternada) (ATOS, 2010).

2.14.5 Saídas analógicas

A interface para saídas analógicas recebe do processador dados numéricos que são convertidos em valores proporcionais de corrente ou tensão e aplicados nos dispositivos de campo. A interface contém um conversor digital-analógico (D/A). O valor analógico é geralmente expresso como um valor decimal (BCD). Os conversores D/A normalmente são de 10 ou 12 *bits*. (ATOS,2010). As faixas de valores de tensão e corrente para saídas analógicas mais utilizadas na indústria são as seguintes:

a) 0 a 20mA;
b) 4 a 20mA;
c) 0 a 10Vdc.

2.16 Sistemas embarcados no contexto da domótica

Computadores do tipo *laptop* ou *desktop* são de uso geral, podendo ser usados em diferente tipo de tarefas. A base desses sistemas é um processador de uso geral, tal como um microprocessador (GISLASON, 2007).

Neste sentido, um sistema embarcado é caracterizado por ser um sistema digital/analógico direcionado para resolver um problema específico, tendo para isto que cumprir com um série de requisitos em desempenho, consumo de potência, custo, tamanho, entre outros. A base de processamento de um sistema embarcado pode ser um microprocessador ou um microcontrolador.

Microcontroladores tendem a ser usados em tarefas de controle e automação, oferecendo uma gama de dispositivos tais como geradores de PWM, interfaces para redes industriais, contadores, *timers*, entradas analógicas e digitais, entre outros (GISLASON, 2007).

Sistemas embarcados vêm sendo usados na indústria automotiva, levando a níveis de complexidade alta, envolvendo até centenas de processadores ligados através de redes de alta segurança (FALUDI, 2011).

Geralmente sistema é classificado como “embarcado” quando está dedicado a uma única tarefa e interage continuamente com o ambiente à sua volta por meio de sensores e atuadores. Por exigir uma interação contínua com o ambiente, esse tipo de sistema requer do projetista um conhecimento em programação, sistemas digitais, noções de controle de processos, sistemas de tempo real, tecnologias de aquisição de dados (conversão analógico/digital e sensores) e de atuadores (conversão digital/analógico, acionamento eletromecânico e PWM), e cuidados especiais na eficiência de estruturação do projeto e do código produzido (CHASE, OTÁVIO, 2007).

As principais características de classificação de sistemas embarcados são a sua capacidade computacional e a sua independência de operação. Outros aspectos relevantes dependem dos tipos de sistemas, dos modos de funcionamento e dos itens desejados em aplicações embarcadas.

Há uma expectativa de crescimento de utilização de sistemas embarcados. De acordo com Gallassi e Martins (2014), nos últimos anos tem crescido muito o volume de *software*s desenvolvidos para sistemas embarcados. Com a diminuição

do custo dos processadores de 32 *bits*, muitas aplicações embarcadas começaram a utilizar esse tipo de processador, o que possibilita o desenvolvimento de *software*s mais complexos e sofisticados. Com os processadores de 32 *bits*, o uso de sistemas operacionais embarcados se tornou uma realidade, principalmente a partir dos sistemas operacionais padronizados para aplicações embarcadas.

2.17 Plataforma de prototipagem MBED para desenvolvimento de aplicações de sistemas embarcados (usada neste projeto)

Em termos gerais, o MBED leva um microcontrolador, tal como vimos na **Figura 10**, e tem na placa alguns circuitos de apoio muito úteis. Ele coloca isso tudo em uma placa de circuito impresso (PCB) de tamanho pequeno e tem todo suporte de um compilador *on-line*, de uma biblioteca de programas e de um manual. Isso gera um ambiente completo de desenvolvimento de sistemas embarcados, permitindo aos usuários desenvolver protótipos de sistemas embarcados de forma simples, eficiente e rápida (MBED.ORG, 2014).

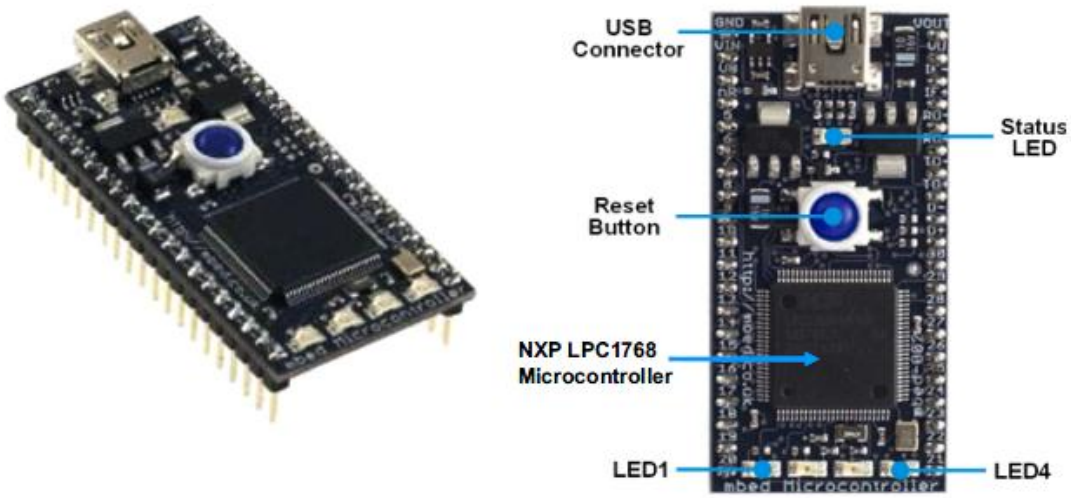


Figura 10 - MBED
Fonte www.MBED.og

O MBED tem uma placa de tamanho pequeno (53 mm x 26 mm), com 40 pinos dispostos em duas filas de 20, com o espaçamento de 0,1 mm entre os pinos, sendo esse espaçamento um padrão em muitos componentes eletrônicos. Olhando para as principais características descritas na **Figura 11**, vemos que o MBED é baseado em um microcontrolador LPC1768 que uma ARM Cortex-M3 fez pela empresa NXP semicondutores (TOUSON, 2012).

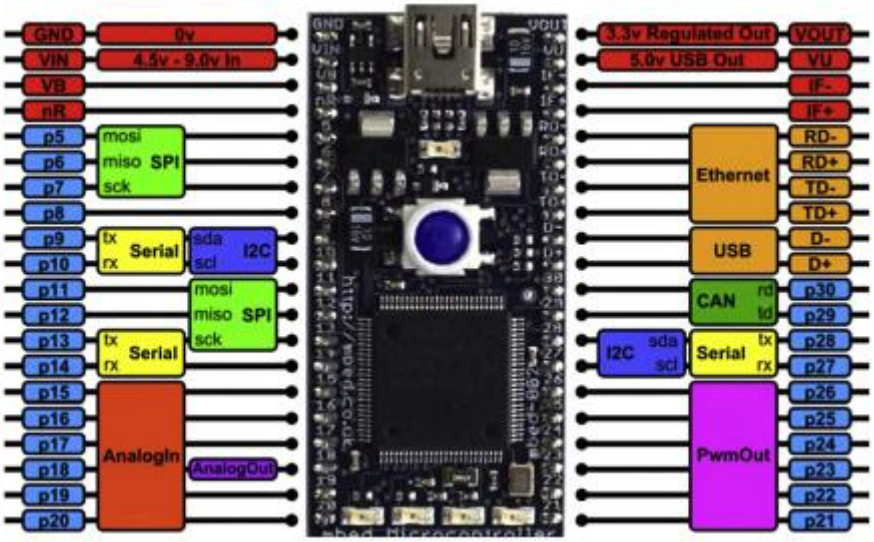


Figura 11 - MBED

Fonte www.MBED.og

Uma representação em diagrama de blocos da arquitetura de MBED é mostrada na Figura 12. No coração do MBED existe um microcontrolador LPC1768, onde se ligam a ele os terminais de sinal do MBED.

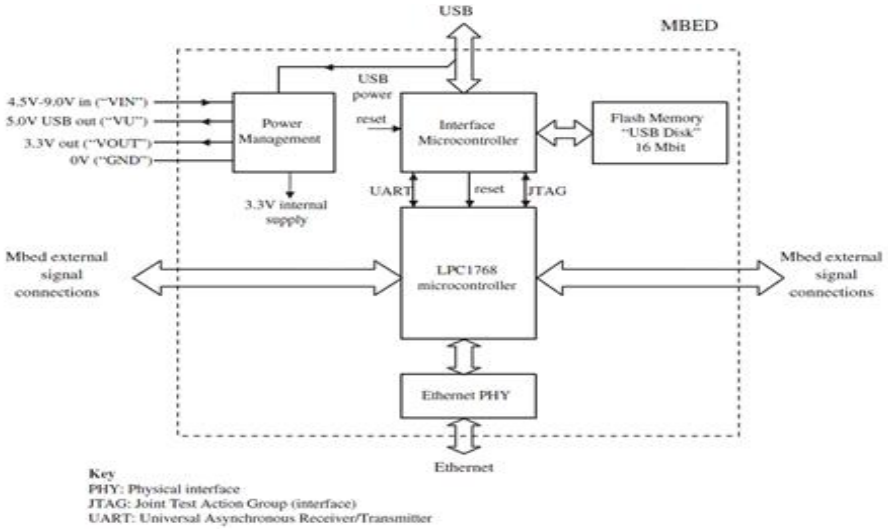


Figura 12 - Circuito

Fonte www.MBED.og

A unidade de “Gestão de Energia" nesse tipo de sistema é composta de dois reguladores de tensão, que se encontram de cada lado do *LED* de *status*. Há também um limitador de corrente, que fica no canto superior esquerdo da *MBED*. O *MBED* pode ser alimentado a partir da *USB*, e essa é uma forma comum de usá-lo, especialmente para aplicações simples. Para aplicações que necessitam de uma

tensão mais alta, ele também pode ser alimentado a partir de uma fonte externa 4,5-9,0volts. Fornecida ao pino 2, a fonte também pode ser proveniente de pinos MBED 39 e 40 (rotulado VU e VOUT, respectivamente) (TOUSON, 2012).

2.17.1 O microcontrolador LPC1768

O microcontrolador utilizado no projeto proposto nesta dissertação é o ARM Cortex-M3 LPC1768, que possui todas as características para atender as aplicações a serem aqui suportadas. Nesta seção, serão descritas as principais características desse microprocessador.

Um diagrama de blocos do microcontrolador LPC1768 é mostrado na **Figura 13**, na qual algumas características são mostradas:

a. Processador ARM Cortex-M3, funcionando em frequências de até 100 MHz. A Unidade de Proteção de Memória (MPU).
b. Até 512 kB no *chip* de memória flash de programação. Aprimorado acelerador de memória *flash*, que permite a operação de alta velocidade de 100 MHz.

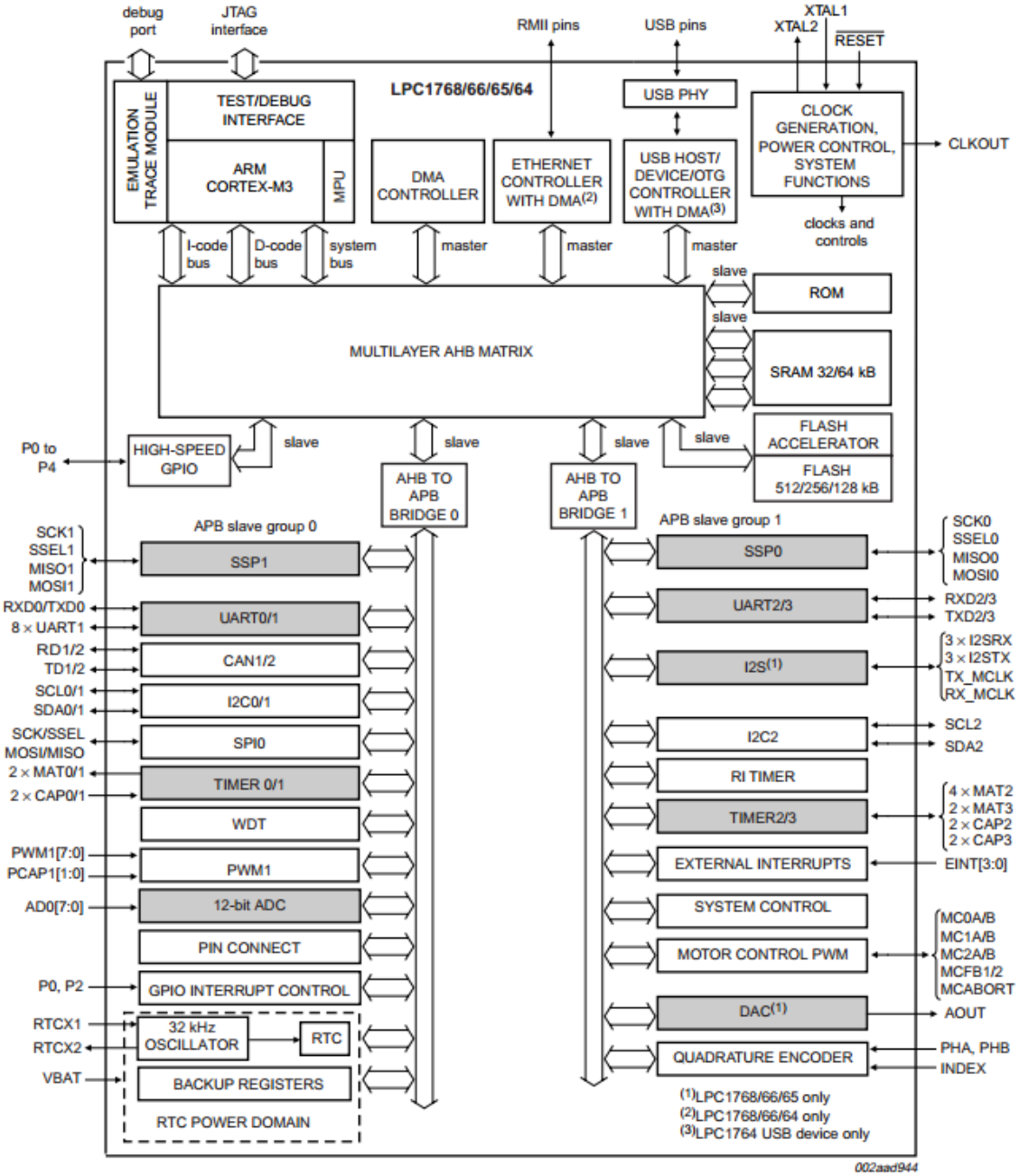
b) Interfaces seriais:

a. MAC *Ethernet* com interface RMII e dedicado controlador de DMA;
b. USB 2.0 dispositivo / *host* / OTG controlador de velocidade máxima com dedicado controlador DMA e PHY no *chip* para as funções do dispositivo, *host* e OTG. O LPC1764 inclui apenas um controlador de dispositivo;
c. Quatro UARTs, FIFO interna, suporte a DMA e apoio RS-485;
d. Um UART tem controle modem I/O, e uma UART tem suporte IrDA;
e. CAN 2.0B com dois canais;
f. SPI controlador síncrona, serial, comunicação *full duplex* e comprimento de dados programável;
g. Dois controladores de SSP com FIFO e capacidades multi-protocolo. As interfaces do SSP podem ser usadas com o controlador GPDMA.
h. Duas interfaces I2C-bus, com uma taxa de 400kbits/s com reconhecimento de endereço múltiplo e modo de monitor de dados;

i. Uma interface de I2C-*bus* completo especificação I2C-*bus* em modo rápido com uma taxa de 1 Mbit/s com reconhecimento de endereço múltiplo e modo de monitor de dados.

c) Outros periféricos:
   a. 70 I/O (GPIO) de uso geral, pinos com resistores *pull-up*/*down* configuráveis e um novo modo de operação-dreno aberto configurável;
   b. 12-*bit* Analog-to-Digital Converter (ADC) com multiplexação de entrada entre os oito pinos, taxas de conversão de até 1 MHz e múltiplos registos de resultados. O ADC de 12 *bits* pode ser usado com o controlador GPDMA;
   c. 10-*bit* Digital-Analógico (DAC) com temporizador de conversão dedicado e suporte a DMA (LPC1768/66/65 apenas);
   d. Quatro temporizadores / contadores de propósito, com um total de oito entradas de captura e dez saídas. Cada bloco temporizador tem uma entrada de conta externa e suporte a DMA;
   e. Um PWM controle de motor com suporte para controle de motor trifásico;
   f. Um bloco PWM / *timer* padrão com entrada de contagem externa;
   g. RTC com um domínio Poder Separado e Dedicado oscilador RTC. O bloco RTC, incluídos 64 *bytes* de registros de *backup* movidos à bateria;
   h. Watchdog Timer (WDT) redefine o microcontrolador dentro de um período razoável de tempo;
   i. Temporizador do sistema, incluindo uma opção de entrada de *clock* externo;
   j. Temporizador de interrupção repetitiva fornece interrupções programadas programáveis e de repetição;
   k. Cada periférico tem o seu próprio divisor de relógio para obter mais economia de energia;
   l. JTAG interface de teste / *debug* padrão para compatibilidade com as ferramentas existentes . fio de série

m. Unidade de Gerenciamento de Energia integrada que ajusta automaticamente reguladores internos para minimizar o consumo de energia;
n. Quatro modos de redução de energia;
o. Única fonte de alimentação de 3,3V (2,4V a 3,6V);
p. Quatro entradas de interrupção externas configuráveis;
q. Interrupção (NMI) de entrada;
r. *Power-On Reset* (POR);
s. Oscilador de cristal com um alcance de 1 MHz a 25 MHz;
t. 4 MHz RC interno oscilador aparadas com uma precisão de 1%, que pode, opcionalmente, ser utilizado como um relógio do sistema.



Grey-shaded blocks represent peripherals with connection to the GPDMA.

Figura 13 - Diagrama de Blocos

Fonte: MBED.org

2.18 – Tecnologia de comunicação sem fio *ZigBee*

Uma das necessidades do projeto proposto nesta dissertação é que seus atuadores e sensores tenham a possibilidade de se comunicar com a placa central usando uma tecnologia sem fio. Para esse fim, foi escolhida uma solução baseada em *ZigBee*, pela facilidade em sua utilização, o que facilmente permite que possam migrar os vários sensores e atuadores existentes no mercado.

A comunicação sem fio (ou *wireless*) já está inclusa no mercado há anos. Entretanto, são poucas as redes *wireless* destinadas exclusivamente ao controle de dispositivos como relês, trancas eletromagnéticas, ventilação, aquecimento, motores, eletrodomésticos, brinquedos, aquisição de dados de sensores, como temperatura, luminosidade, umidade, pressão etc.

Dentre as Redes *WPAN* (Wireless Personal Area Network) existentes, a mais recente e promissora é a que usa o padrão *ZigBee* IEEE 802.15.4. A *ZigBee* Alliance é quem desenvolveu o padrão *ZigBee* junto ao IEEE (Institute of Electrical and Eletronics Engineers), através da associação de várias empresas que, juntas, trabalham em conjunto para proporcionar e desenvolver tecnologias para criar um padrão de baixo consumo de energia, baixo custo, segurança, confiabilidade, e com funcionamento em rede sem fios baseado em uma norma aberta global (MESSIAS, 2008).

Com base na Tabela 3, observa-se que *XBee* (*ZigBee*) se destaca em várias das características acima, em comparação com os seus concorrentes. Uma característica que faz distinguir *XBee* o dispositivo sem fio mais adequado para automação residencial é o seu baixo consumo de energia. Com isso, os aparelhos com fio com *XBee* não exigem a substituição da bateria frequente em comparação com o resto dos dispositivos sem fio (SEONG, Peng; Gik Hong 2011).

Characteristics	Bluetooth	Wireless USB	WiFi
Standard/Reference	IEEE 802.15.1	Certified USB-IF compliance	IEEE 802.11 B-G and others
Distance (max)	100m (class1)	10m (110Mbps)	100m
Data rate (max)	3Mbps	480Mbps (3m)	54Mbps
Connections Line of sight	Ad hoc, max 8 devices No	127 devices (max) No	Point-to-hub No
Relative power consumption	Medium	Low	High
Typical application	Voice and data applications for consumer electronics and personal computing devices	Data applications for consumer electronics and PC peripherals	Data and voice applications, wireless LAN, broadband internet access
Security	Authentication and encryption	Authentication and encryption	Authentication and encryption (WEP/WPA)

Characteristics	WiMax	ZigBee	IrDA Infrared
Standard/Reference	IEEE 802.16a	IEEE 802.15.4	IrDA specifications
Distance (max)	50km	100m nominal	1m
Data rate (max)	70Mbps	250kbps (2.4Ghz)	16Mbps
Connections Line of sight	Point-to-hub, mesh No	Ad hoc, Peer-to-peer, star No	Point-to-point Yes
Relative power consumption	High	Very low	Low
Typical application	Wireless metropolitan area network or last mile technology for rural areas	Sensing and controlling applications in home automation	Remote controls, mobile phones
Security	Authentication and encryption	Several measures, 128- bit AES encryption	n/a

Tabela 3 – Comparação entre soluções sem fio

Fonte : Humanities, Science and Engineering (CHUSER), 2011 IEEE Colloquium on

Atualmente, a *ZigBee* Alliance está incluindo novos e mais abrangentes recursos, possibilitando assim que os fabricantes aumentem significativamente a capacidade da *ZigBee*, fazendo com que sua posição de liderança continue firme e crescente no mercado de redes para controle de dispositivos sem fio. Atualmente, existem mais de 300 empresas associadas à *ZigBee* Alliance em vários países, tendo esta um crescimento expansivo (MESSIAS, 2008).

Sensores sem fio baseados no ZigBee têm dois aspectos no contexto de uso de protocolos: protocolo IEEE 802.15.4 ou protocolo padrão *ZigBee*. Geralmente, a pilha de sensores sem fio inclui quatro camadas. Estas são camada física, camada MAC, camada de rede e camada de aplicação. 802.15.4 (Al, Sharbaty, 2014).

A *ZigBee* permite comunicações robustas, operando na frequência *ISM* (Industrial, Scientific and Medical), sendo, na Europa, de 868 MHz (1 canal), 915 MHz (10 canais) nos Estados Unidos e 2,4 GHz (16 canais) em outras partes do mundo, não requerendo licença para funcionamento. As redes *ZigBee* oferecem uma excelente imunidade contra interferências, e a capacidade de hospedar milhares de dispositivos numa só rede (mais que 65.000), com taxas de transferências de dados variando entre 20Kbps e 250Kbps. O protocolo *ZigBee* é destinado a aplicações industriais, portanto, o fator velocidade não é crítico numa implementação *ZigBee* (MESSIAS, 2008).

Os módulos RF padrão *ZigBee* foram criados para economizar ao máximo energia. Com isso, é possível criar dispositivos sensores remotos alimentados com pilhas ou baterias comuns, que durarão meses ou mesmo anos sem precisar ser substituídas. Isso acontece porque os módulos *ZigBee,* quando não estão transmitindo/recebendo dados, entram num estado de dormência ou em *sleep*, consumindo o mínimo de energia (MESSIAS,2008). Numa rede ZigBee, são identificados dois tipos de dispositivos: FFD e RFD, sendo descritos a seguir.

a) FFD - *Full Function Device* (Dispositivos de Funções Completas): São dispositivos mais complexos e precisam de um *hardware* mais potente para a implantação da pilha de protocolos, consequentemente, consomem mais energia. Numa topologia de Rede *ZigBee,* eles podem assumir o papel de Coordenador, Roteador ou mesmo de um dispositivo final (*End Divice*). Dispositivos FFDs podem se comunicar com quaisquer membros da Rede. São implementados em microcontroladores com, no mínimo, 32KB de memória de programa e tem uma certa quantidade de memória RAM, para implementações de tabelas de rotas e configurações de parâmetros (MESSIAS, 2008).

b) RFD - *Reduced Function Device* (Dispositivos de Funções Reduzidas): São dispositivos mais simples, cuja pilha de protocolo pode ser implementada usando os mínimos recursos possíveis de *hardware*,

como, por exemplo, em microcontroladores de 8 *bits* com memória de programa próxima a 6KB, mas só podem se comunicar com dispositivos FFDs (Coordenador ou Roteador). Numa topologia de rede *ZigBee*, eles assumem o papel de End Device (dispositivo final). Na prática, podem ser: interruptores de iluminação, *dimmers*, controle de relês, sensores, entre outros. No padrão *ZigBee,* existem três classes de dispositivos lógicos (Coordenador, Roteador e Dispositivo final) que definem a rede (MESSIAS, 2008).

Figura 14 - *Xbee* Aplicações

Fonte: http://www.rogercom.com

Algumas aplicações (**Figura 14**) de sensores usando *ZigBee:*

a. Sensor de umidade;
b. Sensor de temperatura;
c. Sensor de velocidade do vento;
d. Sensor de direção do vento;
e. Sensor de pressão atmosférica;
f. Controle de iluminação;
g. Controle de aquecimento;
h. Controle de ventilação;
i. Controle de irrigação;
j. Alarmes;
k. Controle de cancelas;
l. Controle de portas e portões;
m. Aplicações automotivas.

2.18.1 – Topologia das redes *ZigBee*

Podem-se destacar as principais topologias de redes *Zigbee* (vide **Figura 15**):

a) *Mash* (Malha ou Ponto a Ponto): Na topologia *Mesh*, a rede pode se ajustar automaticamente, tanto na sua inicialização como na entrada ou saídas de dispositivos na rede. A rede se auto-organiza para otimizar o tráfego de dados. Com vários caminhos possíveis para a comunicação entre os nós, esse tipo de rede pode abranger em extensão, uma longa área geográfica, podendo ser implementada numa fábrica com vários galpões distantes, controle de irrigação ou mesmo num prédio com vários andares (FALUDI, 2006);
b) *Cluster Tree* (Árvore): Semelhante à topologia de “Malha”, uma rede em árvore tem uma hierarquia muito maior, e o coordenador assume o papel de nó-mestre para a troca de informação entre os nós *Router* e *End Device* (FALUDI, 2006);
c) *Star* (Estrela): É uma das topologias de Rede *ZigBee* mais simples de serem implantadas, é composta de um nó-coordenador, e quantos nós *End Device* forem precisos. Esse tipo de rede deve ser instalada em locais com poucos obstáculos para a transmissão e a recepção dos sinais, como, por exemplo, em uma sala sem muitas paredes ou locais abertos (FALUDI, 2006).

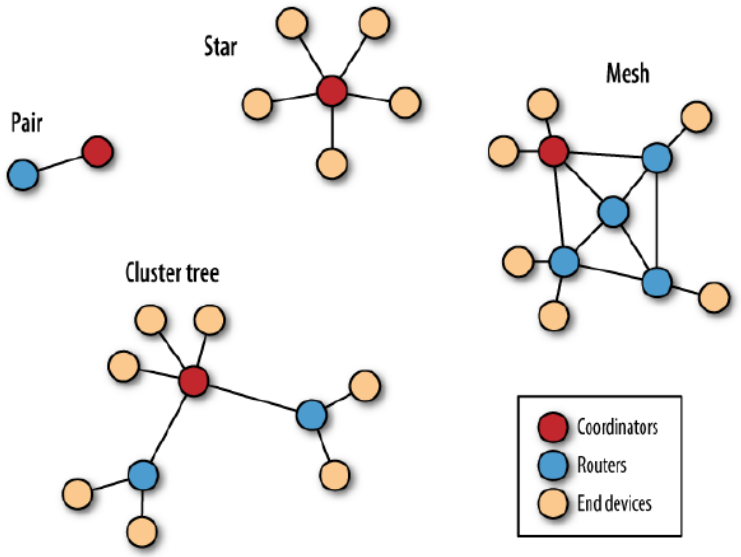


Figura 15 - Redes *Xbee*

Fonte : Faludi, 2006 Pag, 28

2.18.2 Módulos de comunicação *ZigBee*

Existem muitas empresas que fabricam seus produtos baseados na pilha de protocolo *ZigBee*, como os utilizados no projeto proposto nesta dissertação (vide **Figura 16**). Os *Xbee* e *Xbee*-pro 802.15.4 são módulos embarcados que fornecem uma comunicação ponto a ponto sem fio entre dispositivos. Esses módulos utilizam o protocolo de rede IEEE 802.15.4 em redes ponto a ponto, ponto a multiponto ou *peer-to-peer*. Os mesmos são projetados para aplicações de alto rendimento que requerem baixa latência e sincronismos de comunicação previsíveis. Esses dispositivos são ideais para aplicações de baixa potência e de baixo custo. Os mesmos podem ser utilizados para substituir cabos de comunicação serial ou, numa aplicação mais complexa, atuar como um *hub* de uma rede de sensores sem fio.

Os módulos 802.15.4 fazem parte da família de produtos *RF Xbee* da *Digi International*, são fáceis de usar, compartilham um único fator de forma e são totalmente interoperáveis com outros produtos *Xbee* que utilizam a tecnologia 802.15.4. Os usuários podem substituir um módulo *Xbee* por outro modelo com o mínimo esforço de desenvolvimento e de risco (Albacore, 2014).

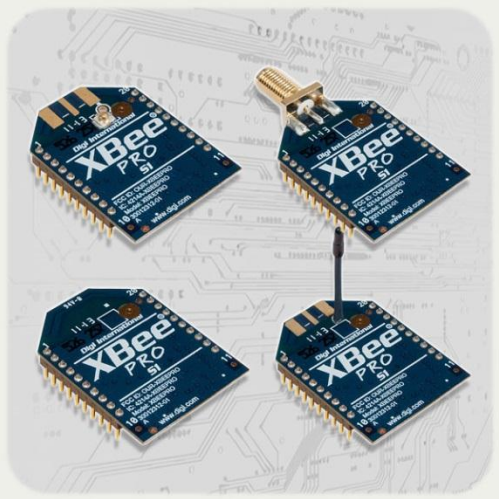


Figura 16 - Modelos de Antenas
Fonte: http://www.albacore.com.br

2.19 Conclusões do capítulo

Neste capítulo de revisão bibliográfica, foram apresentadas as principais informações que compõem o projeto, abrangendo as definições e as características de sistemas domóticos, o que possibilitou adquirir o conhecimento necessário para aplicar melhorias na plataforma. Um exemplo é que, somente após a terceira versão da placa, foi verificada a necessidade de melhorias para atender o mercado atual, que converge para a comunicação de todos os equipamentos usando tecnologias sem fio.

Em seguida, foram apresentados conceitos e características de um CLP, já que o objetivo principal do projeto é possuir todas as características e funcionalidades de um CLP residencial, permitindo assim que seus usuários possam facilmente criar um ambiente residencial controlado através de uma série de atuadores e sensores.

O estudo da conceituação e as principais características de sistemas embarcados também foram apresentados, pois, apesar do foco na aplicação de um CLP, o projeto também é uma poderosa plataforma para emulação de sistemas embarcados, podendo facilmente ser usado nos laboratórios para estudos e testes de aplicações que necessitem de configurações diversas já que a plataforma criada

possui periféricos específicos para aplicação de um CLP. O projeto proposto também possui alguns periféricos para testes de aplicações de sistemas embarcados, além de acesso a todos os pinos do microcontrolador, o que estende as possibilidades de utilização do mesmo.

Foram apresentadas as características do microcontrolador ARM Cortex-M3 LPC1768, que é um dos mais avançados microcontroladeres de 32 *bits* da família ARM. Esse microcontrolador possui integrados periféricos como USB, CAN, RS485, *ETHERNET*, entre outros, o que permite abrangentes possibilidades de criação de aplicações usando apenas um microcontrolador, sem a necessidade de outros circuitos auxiliares.

Umas das características do projeto apresentado nesta dissertação é a fácil reprodução dessa plataforma por qualquer um que se interesse. Entretanto, inicialmente foi verificada uma grande dificuldade na criação de uma placa com soldagem de um microcontrolador tipo SMD, o que tornaria inacessível para a maioria dos usuários a reprodução do mesmo, já que haveria necessidade de equipamentos profissionais sofisticados. Neste caso, optou-se por utilizar uma plataforma de prototipagem já existente, que utilizasse o microcontrolador escolhido. Essa plataforma é a MBED, que é uma placa que já possui o microcontrolador ARM LPC1768 soldado com todas as saídas disponibilizadas em um sistema de pinos.

Isso permitiu criar uma plataforma modular, onde, espetando o MBED na placa-mãe projetada, o sistema estaria em condições operacionais de desenvolvimento de aplicativos típicos de um CLP.

CAPÍTULO 3: METODOLOGIA PROPOSTA PARA O TESTE E ELABORAÇÃO DAS PLACAS DE CIRCUITO IMPRESSO

Este capítulo tem por objetivo demonstrar a metodologia empregada no desenvolvimento do *hardware* proposto nesta dissertação, bem como todos os testes de validação das placas desenvolvidas. Os testes dos aplicativos desenvolvidos, por sua vez, serão descritos no Capítulo 5.

Neste contexto, a metodologia de desenvolvimento das placas de circuito impressa do projeto seguiu os procedimentos descritos:

a) Levantamento dos requisitos;
b) Definição dos componentes a serem utilizados;
c) Desenvolvimento esquemático do *software*;
d) Criação dos *layouts* em *software*;
e) Fabricação da placa de circuito impresso protótipo;
f) Montagem dos componentes;
g) Testes eletrônicos;
h) Criação dos *softwares* para embarcar no microcontrolador;
i) Teste dos *softwares*;
j) Simulação de um cenário de um CLP residencial.

3.1 Levantamento dos requisitos e descrição do *hardware*

O sistema eletroeletrônico do projeto proposto nesta dissertação foi baseado em uma arquitetura que tem como principal processador um ARM CORTEX-M3, produzido pela NXP *Semiconductors*. Essa plataforma está preparada para trabalhar com duas placas com microcontroladores já embarcados, a MBED (Figura 17) e a LPCxpresso (vide Figura 18). Essa linha da família ARM é especialmente voltada para aplicações de alto desempenho, baixo consumo e baixo custo. Sua grande

variedade de periféricos e linhas de comunicação disponíveis possibilitam a interação com muitos equipamentos encontrados no mercado.

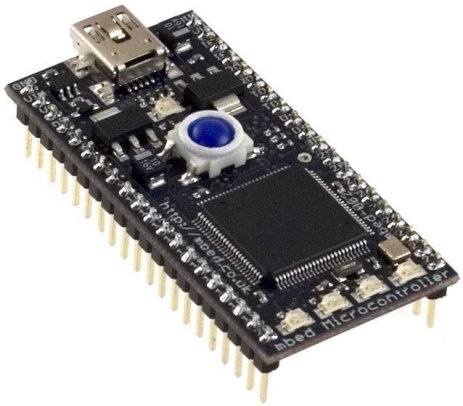


Figura 17 - Placa MBED
Fonte: www.MBED.org (2013)

Figura 18 - Placa LPCxpresso
Fonte: www.MBED.org (2013)

3.2 Versões das placas

Durante as fases de desenvolvimento e testes, foram desenvolvidas quatro versões da plataforma proposta nesta dissertação. Nesta seção, cada versão será discutida em detalhe.

3.2.1 Versão 1.0

Na versão 1.0 do *hardware* (Figura 21), foi definido um escopo inicial da plataforma de desenvolvimento com todas as funcionalidades de um CLP (Controlador Lógico Programável), possuindo as seguintes características:

a) Controle de *displays* LCD alfanumérico 16X4 (16 colunas por 4 linhas), nos modos 4 e 8 *bits*;
b) Matriz de teclado com 4 teclas;
c) 4 *leds* para controle lógico visual;
d) 2 relés NA/NF para acionamento de cargas externas de 10A / 220V;
e) RTC – relógio de tempo real com bateria, para o desenvolvimento de aplicações que necessitassem de temporizadores, acionamentos programados, calendários, entre outras;
f) Canal Serial RS232: canal para comunicação serial com PC ou outras máquinas;
g) Canal USB 2.0;
h) Sensor de temperatura LM35;
i) Memória serial E2PROM via I2C 24C04;
j) Canal de comunicação RS485;
k) Microcontrolador PIC18F452 DIP, com 32Kbyte de Flash;
l) Regulador de tensão;
m) Alimentação: AC ou DC, por transformador ou fonte de 12V 1A;

n) 04 entradas analógicas: entradas de 0 a 5V DC;
o) 04 entradas digitais;
p) 02 saídas a triac.

Na apresentação do escopo da primeira versão, foram identificadas algumas restrições com o microcontrolador PIC, sendo definida a troca deste por um ARM de 32 *bits*. Após a aprovação do escopo, foram iniciadas as seguintes etapas, referentes ao projeto:

a) Escolha dos componentes do projeto;
b) Criação do esquemático do projeto;
c) Criação do *layout* do projeto;
d) Confecção da primeira placa;
e) Montagem e teste dos componentes na placa.

A Figura 19 demonstra como ficou o esquemático da primeira versão do projeto

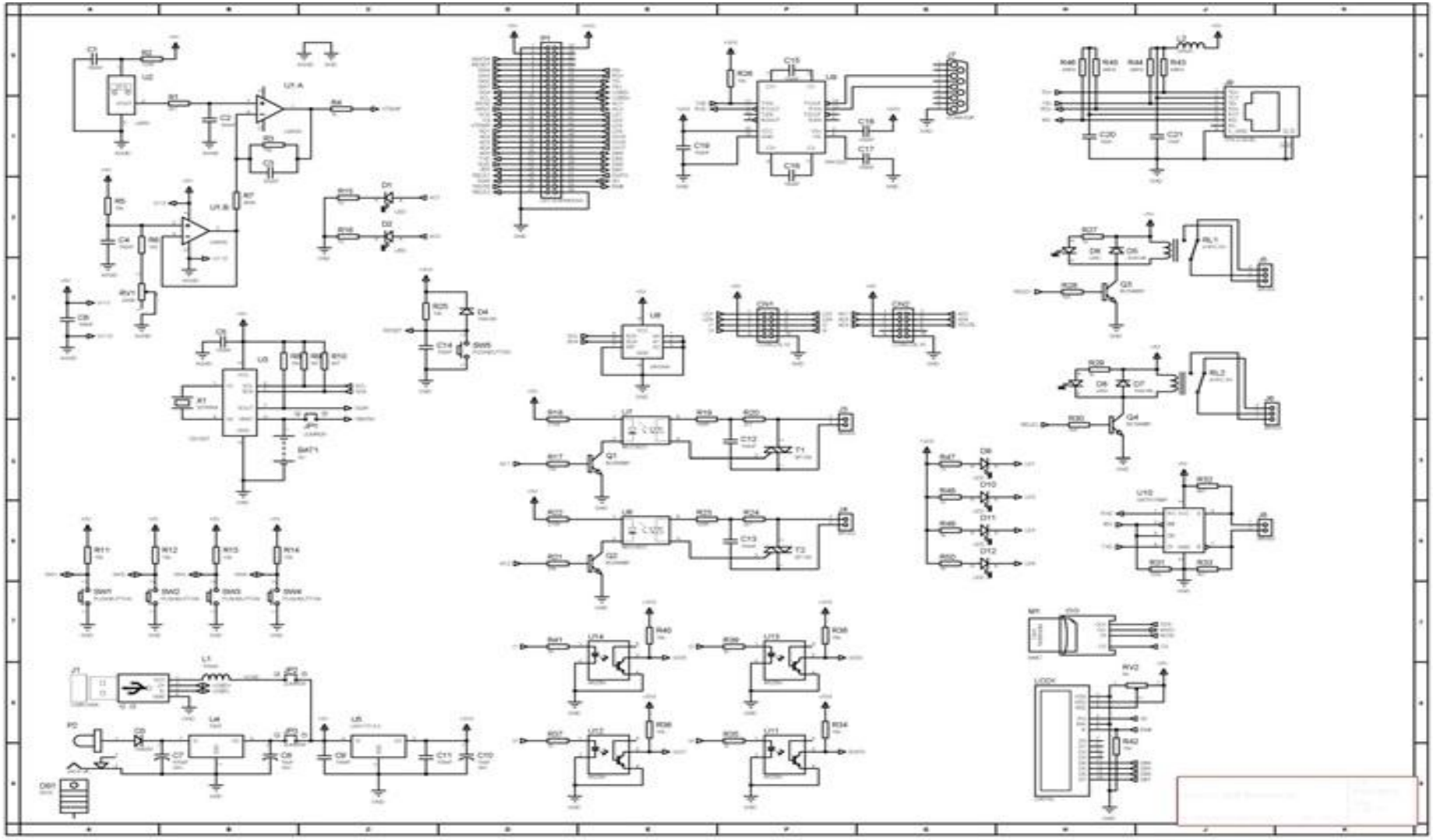


Figura 19 – Esquemático da Placa Versão 1.0

Fonte: Elaborado pelo autor.

O *layout* da placa 1.0 é mostrado na Figura 20, e a Figura 21 mostra o resultado completo do projeto da placa.

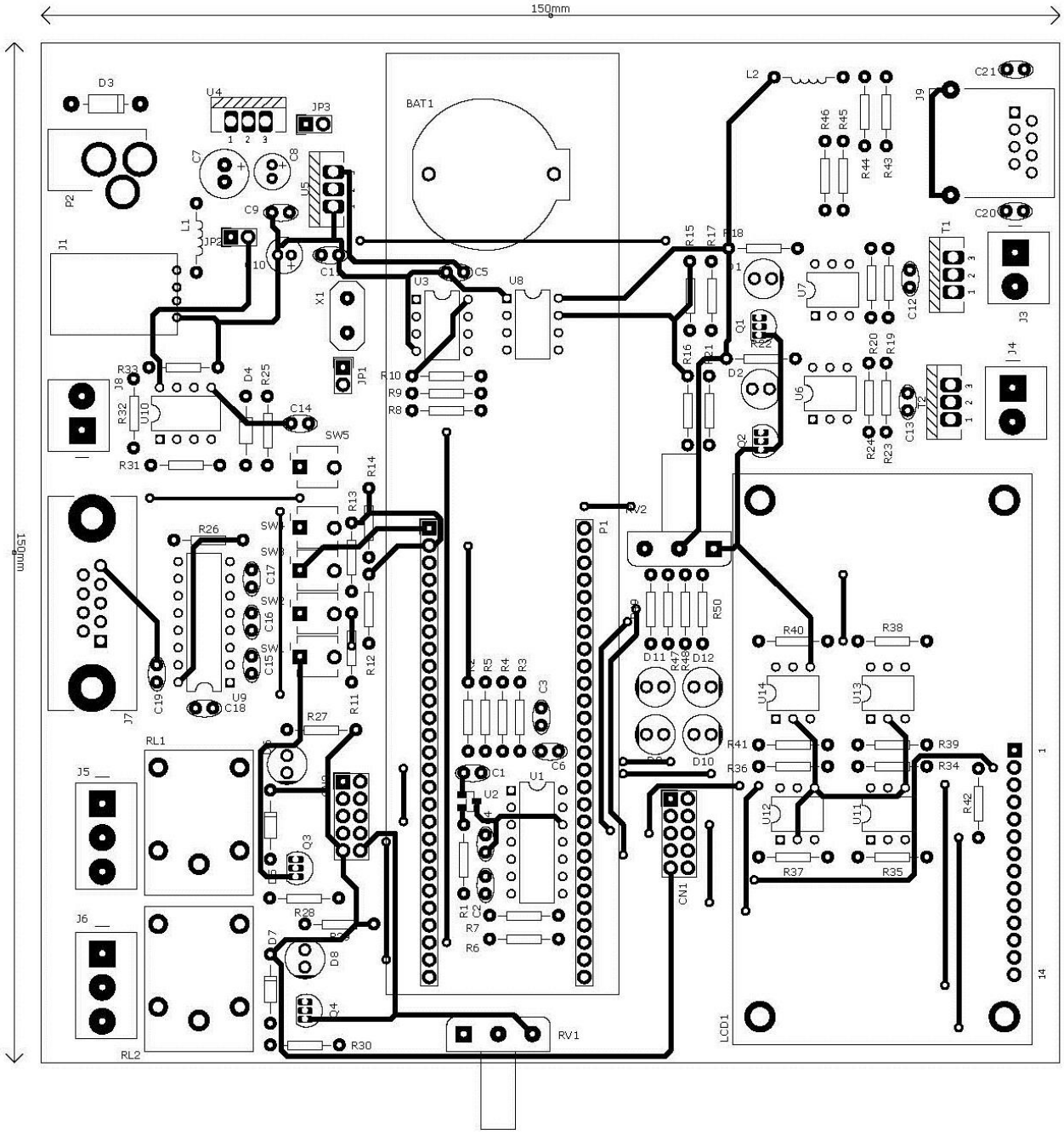


Figura 20 - *Layout* da Placa

Fonte: Elaborado pelo autor.

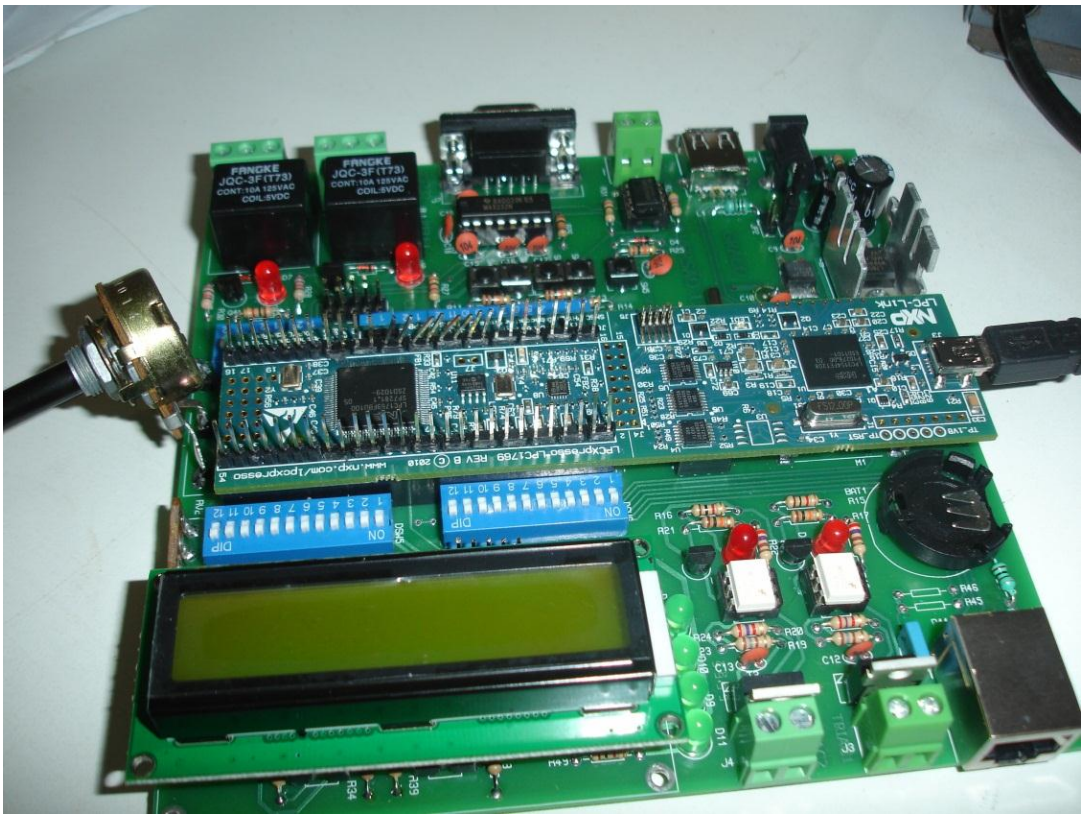


Figura 21 - Placa CLP 1.0

Fonte: E laborado pelo autor.

Após alguns testes, na primeira apresentação da placa, foi necessário realizar alguns ajustes oriundos de observações feitas pelo professor orientador do projeto:

a) Foi solicitada uma interface para comunicação de uma rede CAN, já que a plataforma possuía interfaces para *Ethernet*, RS485 e RS232;
b) Foi solicitada a troca do acesso aos pinos que estavam em cima da placa do LPCxpresso para a lateral da placa, como uma barra de pinos fêmea, ligado a cada pino do LPCxpresso;
c) Foi solicitada a retirada do sensor de temperatura, deixando esses pinos disponíveis para outras funcionalidades.

3.2.3 Placa CLP versão 1.1

Com a adição da etapa do circuito CAN, houve alguns conflitos entre os pinos do LPCxpresso/MBED e a placa projetada, os quais são descritos a seguir.

Para o projeto físico da interface do protocolo CAN, era necessário o uso dos pinos correspondentes do *datasheet*. Neste sentido, só existiam as seguintes possibilidades:

a) P0.1 e P0.2 → pinos 9 e 10 do conector P1 da placa base, já utilizados para a interface I2C (interface com o RTR);
b) P0.4 e P0.5 → pinos 38 e 39 do conector P1 da placa-base, já utilizados para as linhas de controle das cargas AC (AC1 e AC2);
c) Segundo *DATASHEET* do LPCxpresso/MBED, o CAN_rx2 e o can_tx2 são habilitados pelos pinos P0.4 e P0.5 → pinos 38 e 39;
d) P0.21 e P0.22 → pinos 23 e 24 do conector P1 da Placa Base, já utilizados para as linhas de R/T e RELE 1;

Com isso, foi necessário sacrificar algumas expansões extras de IO's comuns ou, na pior das hipóteses, retirar algum circuito da placa-base. Neste caso, a solução obtida foi o uso dos pinos P0.4 e P0.5 para o CAN, e definida a transferência das linhas de controle de cargas AC1 e AC2 para onde estão os sinais LD3 e LD4 (pinos 42 e 43). Neste sentido, sacrificaram-se duas linhas de interface que se julgaram menos importantes nesse momento. A outra opção era retirar duas entradas analógicas (AD5 e AD6 pinos 19 e 20) e colocar no lugar os controles AC1 e AC2.

Entretanto, todos os pinos estavam sendo utilizados para alguma função, sendo inevitável o sacrifício de alguma interface com a adição do circuito CAN. Com as modificações acima, o esquemático modificado ficou como mostrado na Figura 22

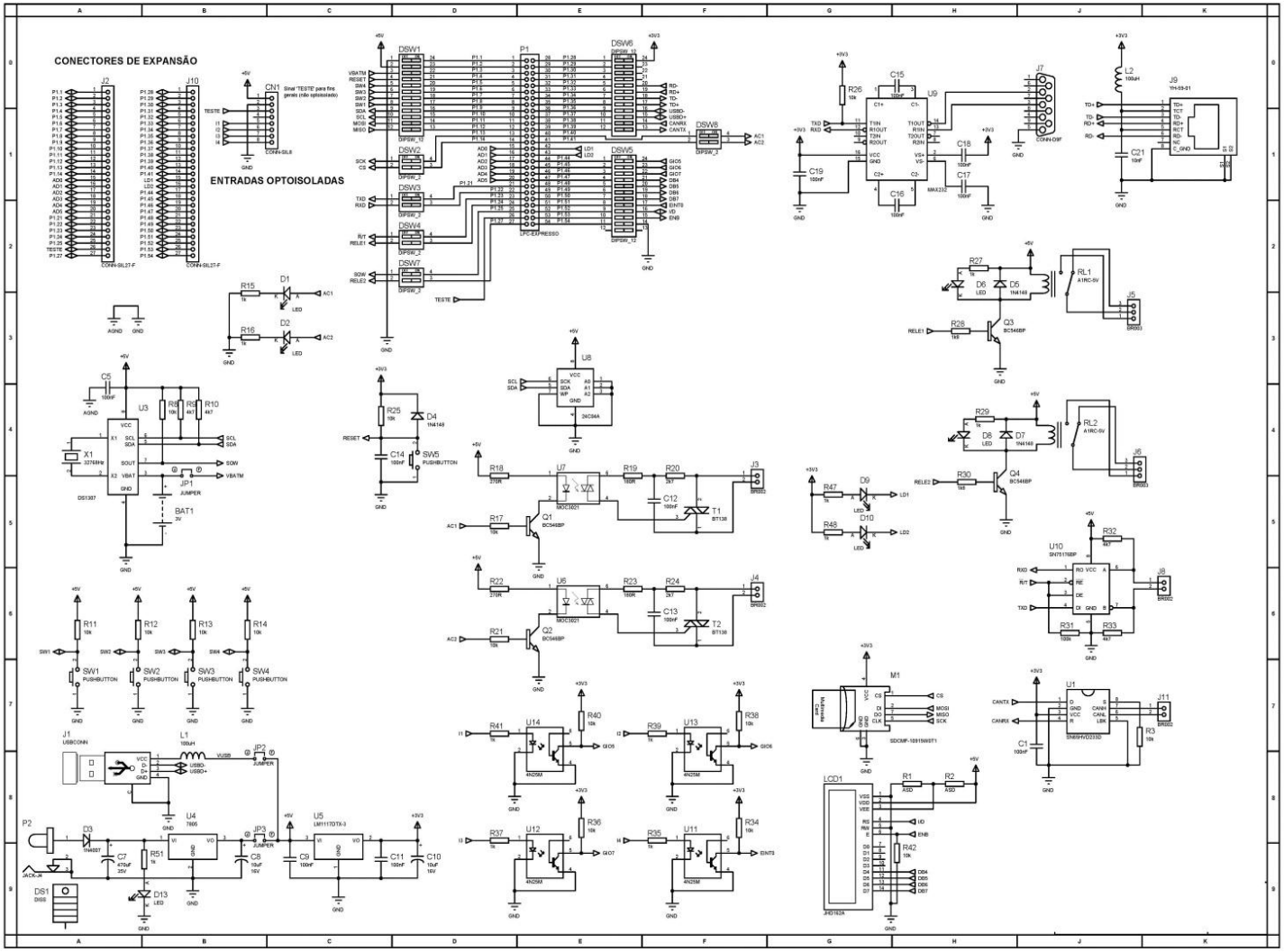


Figura 22 – Esquemático da Placa CLP 1.1

Fonte: elaborado pelo autor.

O *layout* da placa da versão 1.1 ficou mostrado na Figura 23.

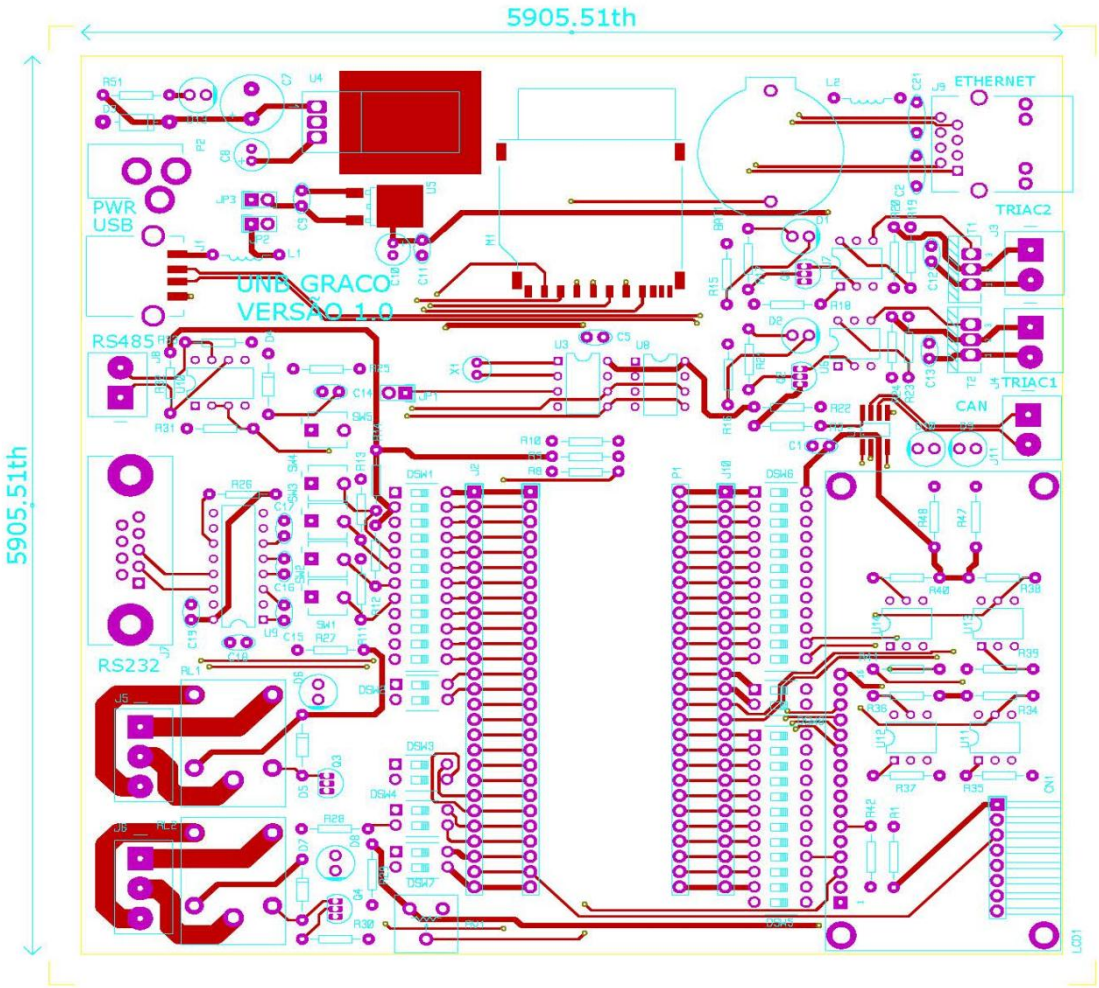


Figura 23 - *Layout* da Placa CLP 1.1

Fonte: elaborado pelo autor.

Após a definição de todos os novos componentes e a compra dos mesmos, passou-se ao processo de montagem da nova placa. Nessa tarefa, houve alguns problemas com a importação de componentes, como o RJ45 Magicjack, que foi importado da S*parkfun*[1], pois este já continha todos os componentes embarcados no próprio corpo do RJ45; nesse caso, a funcionalidade adequada do circuito *Ethernet*.

[1] Sparkfun loja de eletrônicos - https://www.sparkfun.com/products/8534

Após a montagem, a aparência da placa ficou conforme Figura 24:

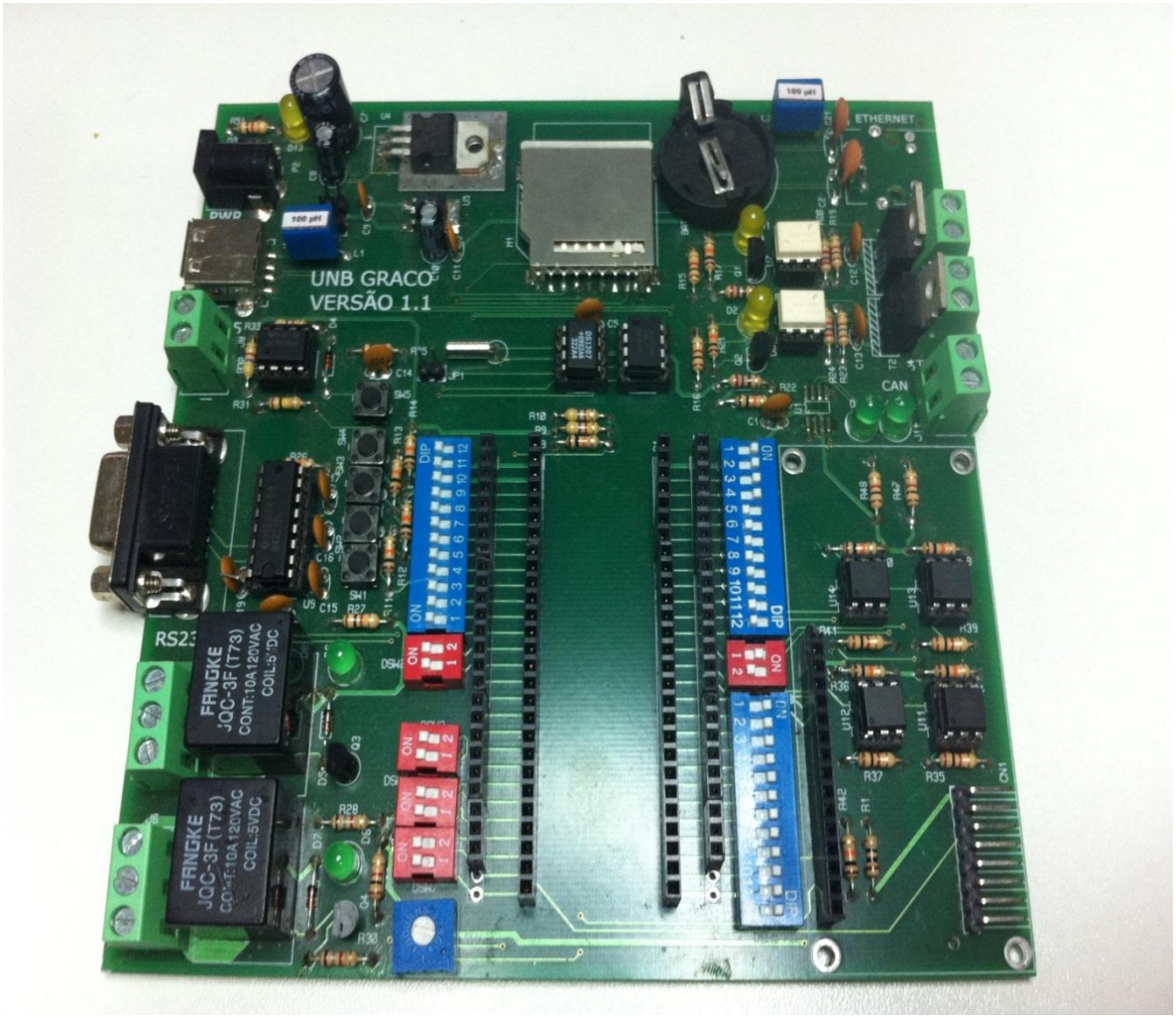


Figura 24 - Placa CLP 1.1 montada

Fonte: elaborado pelo autor.

3.2.4 Versão 1.2

Após uma série de testes e a apresentação da versão 1.1 da plataforma, foram solicitadas novas modificações, descritas a seguir:

a) A placa estava adequada, mas era grande e dever-se-ia pensar em usar componentes SMD na confecção da mesma;
b) Mudança da expansão dos pinos para a lateral da placa, para facilitar a conexão de outros periféricos;
c) Incluir furos para fixação da placa em uma caixa qualquer;
d) Devia ser incluída uma entrada na fonte para uma bateria 12v externa;
e) Como a placa só tem quatro reles, isso era um limitador para o controle do CLP; assim, foi solicitado que o *hardware* pudesse receber módulos de expansão com outros reles.,

Entretanto, essas modificações eram bem significativas e geraram dúvidas. Após muitas análises, deparou-se com alguns problemas:

a) Faltava entendimento acerca dos componentes SMD;
b) Como escolher e onde achar esses componentes?
c) Como soldar esses componentes na placa?
d) Como seriam o funcionamento e a interface com a placa de rele externa?
e) Qual seria o custo desse processo?
f) Desenvolver uma placa de rele ou usar um *shield* já pronto?

Para tanto, realizaram-se muitas pesquisas e estudos bibliográficos sobre a utilização de componentes SMD nos projetos eletrônicos e resolveu-se modificar toda a placa para essa nova concepção. Neste caso, encontraram-se muitas informações sobre placas como módulos auxiliares (que já existiam) e utilizou-se a mesma estratégia de projeto de uma placa muito utilizada (chamada de *ARDUINO*[2]).

No *ARDUINO*, existe um *kit* único, com o microcontrolador e os pinos de acesso para as expansões necessárias. Dessa forma, é possível ter muitas opções de configurações, assim como acessos para vários tipos de criações de protótipos de circuitos eletrônicos.

[2] http://www.arduino.cc/

Na plataforma criada (e conforme solicitado), dever-se-ia ter todas as funcionalidades básicas de um CLP, assim como oferecer a capacidade de poder ser utilizado para o aprendizado como *kit* de desenvolvimento. Após algumas pesquisas, chegou-se a um novo *kit* com microcontrolador ARM igual ao que se vinha utilizando até o momento. Este *kit* é o MBED (www.MBED.org), que segue a mesma arquitetura do *ARDUINO*, possuindo vasta literatura, assim como um compilador *online* gratuito e uma comunidade dedicada ao aperfeiçoamento e à total compatibilidade com os *Shields* já existentes para o *ARDUINO*.

A principal vantagem do MBED (mostrado na Figura 25) é a sua expressa prototipagem, sendo muito rápido fazer um projeto com o MBED, além da disponibilidade de vários *shields*. Os *shields* são placas de espação de funcionalidades criadas para o ARDUINO, e podem ser facilmente adaptadas para serem conectadas no MBED. Cada *shield* possui uma finalidade diferente.

Existem *shields* para *displays*, conexão com a internet, controle de motores, controle de relés, transmissão *bluetooth*, mp3 e vários outros. A seguir, serão mostrados alguns exemplos desses *shields*:

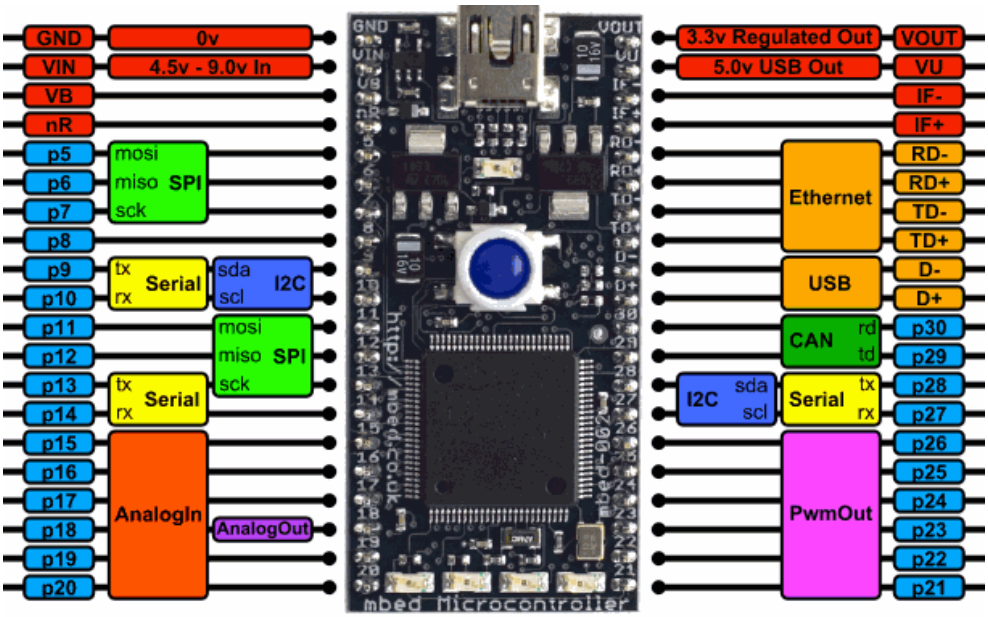


Figura 25 - MBED periféricos

Fonte: MBED.org

a) Relay-*shield* (mostrado na Figura 26) é uma placa muito versátil, pois permite acionar cargas em outras tensões de operação (por exemplo,

lâmpadas, motores, eletrodomésticos, máquinas, etc). Nesse caso, a mesma funciona como um interruptor eletrônico, no qual pode-se ligar e desligar cargas.

Figura 26 - *Relay Shield*
Fonte: sparkfun.com

b) GSM/GPRS-s*hield* (mostrado na Figura 27) é um módulo que pode ser utilizado no projeto desenvolvido e possibilita operar na rede móvel GSM/GPRS de celulares, capaz de efetuar todas as funções de um aparelho convencional, tais como efetuar e receber chamadas, enviar e receber SMS, conectar-se à *internet* por meio de conexão GPRS e enviar FAX.

Figura 27 - GSM *Shield*
Fonte: sparkfun.com

c) O *shield* (mostrado na Figura 28) pode controlar até quatro motores de corrente contínua e uma carga resistiva extra. Para usá-lo, basta conectar uma fonte de alimentação de até 46V (até 1A por motor), ligando nos terminais Vcc e GND. A placa possui indicação de positivo e negativo para cada motor, que pode puxar até 1A, na tensão máxima de alimentação da fonte (Vcc), mas que pode variar por PWM (controle disponível na biblioteca). A carga resistiva extra pode puxar até 30A em 60V.

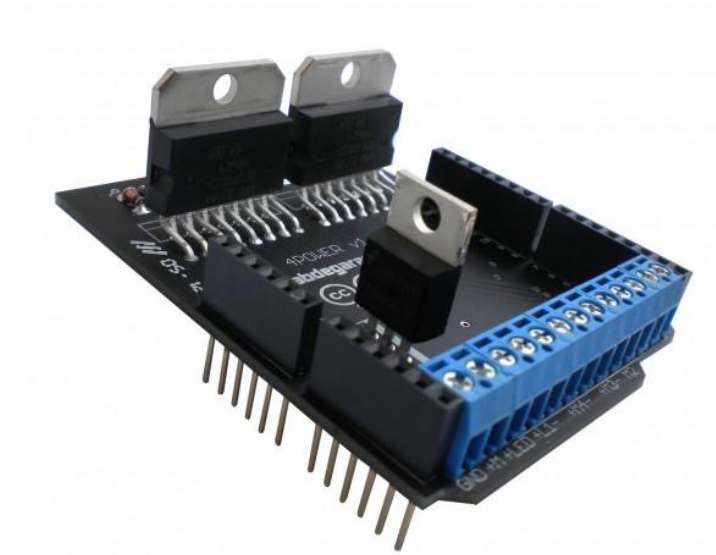
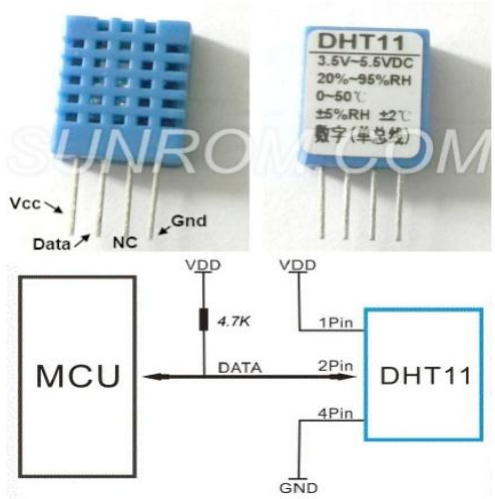


Figura 28 - *Shield* para controle de motores
Fonte: sparkfun.com

Adicionalmente, existem muitos outros *shields* para diversas aplicações, sendo que, para fins didáticos, pode-se projetar um *shield* facilmente, como por exemplo, um sensor de temperatura (vide Figura 29).



:

Figura 29 - Sensor de Temperatura e umidade

Fonte: unrom.com

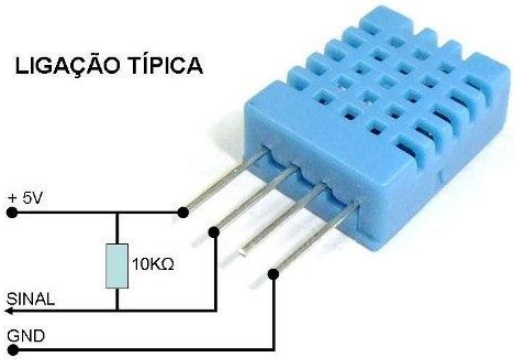


Figura 30 - Sensor de temperatura e umidade – ligação típica

Fonte: elaborado pelo autor

Um *shield* montado comercialmente fica como mostrado na Figura 31.

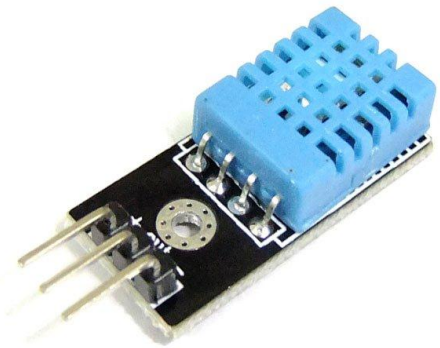


Figura 31 - *Shield* Sensor de temperatura e umidade montado

Fonte: sparkfun.com

Nesta nova versão, o esquemático para a placa de CLP ficou conforme a Figura 35:

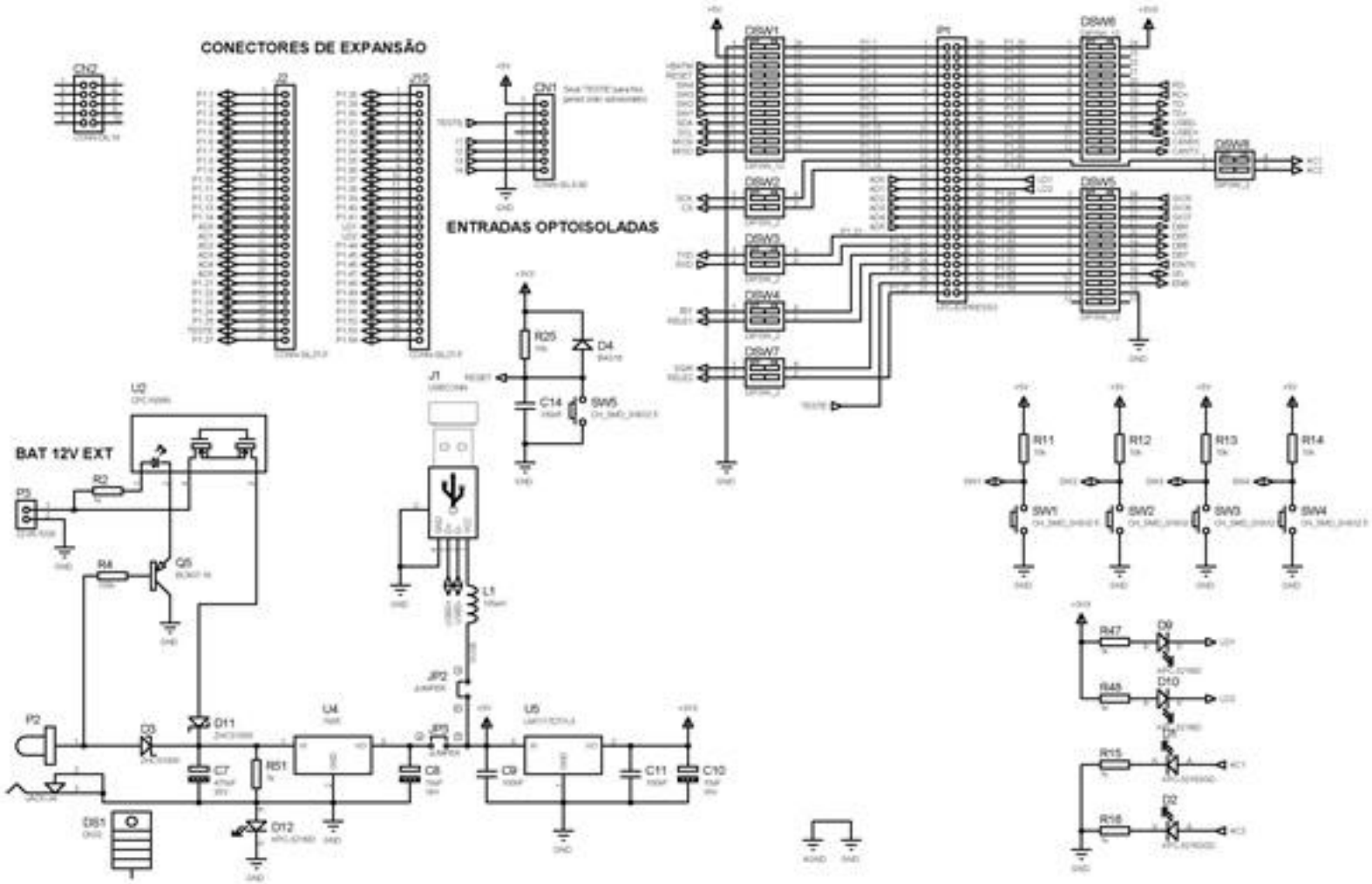


Figura 32 - Parte 1 Esquemático – CLP versão 1.2

Fonte: elaborado pelo autor

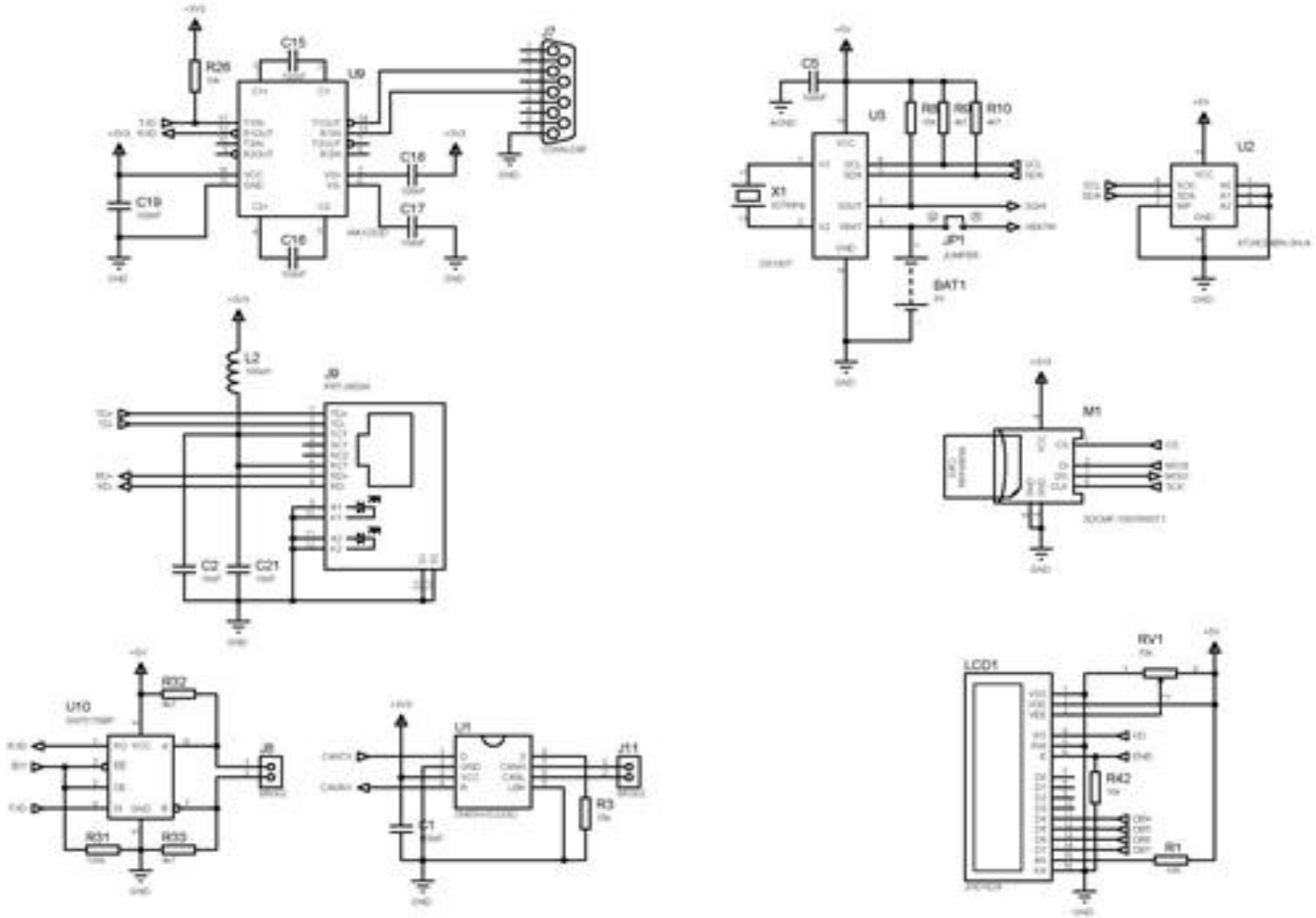


Figura 33 - Parte 2 Esquemático – CLP versão 1.2

Fonte: elaborado pelo autor

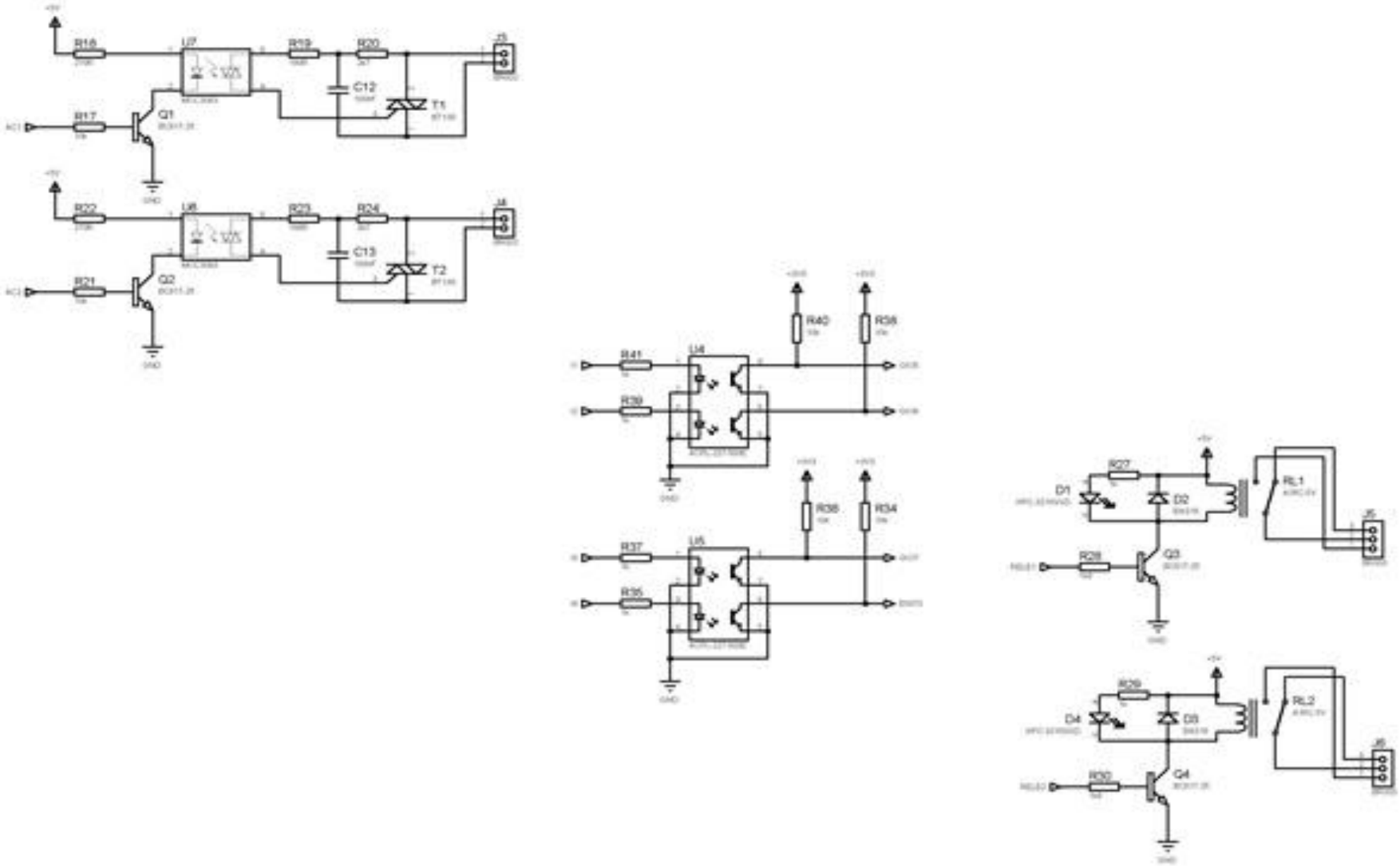


Figura 34 - Parte 3 Esquemático – CLP versão 1.2

Fonte: elaborado pelo autor

O *layout* desta versão da placa ficou conforme a Figura 37.

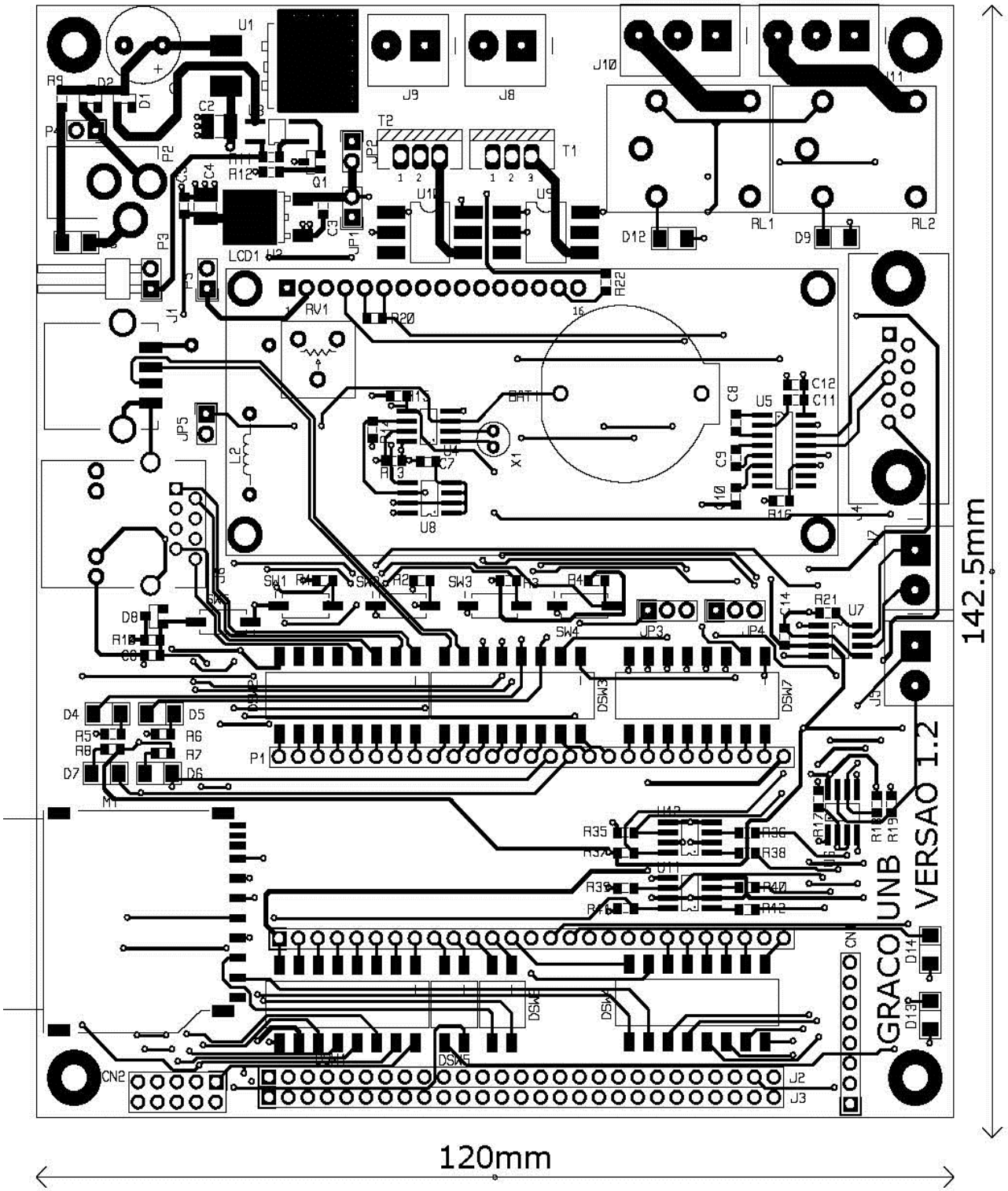


Figura 35 - *Layout* da placa – versão 1.2

Fonte: elaborado pelo autor

Após serem fabricadas as placas de fenolite. não se sabia como fazer a soldagem dos componentes SMD e houve muitas dificuldades com os testes. Neste caso, optou-se por fazer um curso de soldagem SMD na empresa Tecnoponta[3], com o seguinte conteúdo programático:

- Introdução à solda convencional;
- Tipos de componentes THTs;
- Ferramentas necessárias;
- Ferro de solda;
- Solda;
- Sugador;
- Camisinha;
- Pano de limpeza;
- Suporte;
- Esponja vegetal;
- Pasta de solda;
- Álcool Isopropílico;
- Escova antiestética;
- Tipos de soldas e suas aplicações;
- Tipos de ferros de solda (estação de trabalho) e suas aplicações;
- Tipos de sugadores;
- Limpeza e manutenção do ferro de solda;
- Trocando a ponta e a resistência;
- Limpeza e manutenção do sugador;
- Dessoldando o componente;
- Executando procedimento correto com sugador e ferro de solda;
- Removendo o componente;
- Preparando a área da placa para colocação do componente;
- Localizando os terminais do componente;
- Preparando os terminais do componente;
- Soldando o componente;
- Realizando limpeza da placa;
- Introdução à solda SMD;

[3] Disponível em: http://www.tecnoponta.com.br/cursos/solda-smd/

- Tipos de componentes SMDs;
- Ferramentas necessárias;
- Estação de retrabalho;
- Solda em pasta;
- Fita dessoldadora;
- Ferro de solda ponta grossa;
- Pinça;
- Fluxo;
- Álcool Isopropílico;
- Pincel;
- Tipos de estações de retrabalho e suas aplicações;
- Estação de retrabalho analógica;
- Estação de retrabalho digital;
- Tipos de bicos e suas aplicações;
- Cuidados de uso das estações de retrabalho;
- Dessoldando o componente;
- Preparando o componente para dessolda;
- Definindo o bico para dessolda;
- Acertando a temperatura e o fluxo de ar da estação de retrabalho;
- Cuidados na remoção do componente;
- Removendo o componente;
- Preparando a área da placa para colocação do componente;
- Retirando a solda remanescente da placa;
- Localizando os terminais do componente;
- Preparando os terminais do componente;
- Soldando o componente;
- Aplicação correta da solda em pasta;
- Realizando limpeza da placa.

Após o curso, iniciou-se o processo de soldagem. O processo de soldagem levou, aproximadamente, três (3) semanas, e a placa ficou conforme as Figura 36 e 37.

Figura 36 - Foto da placa CLP – versão 1.2

Fonte: elaborado pelo autor

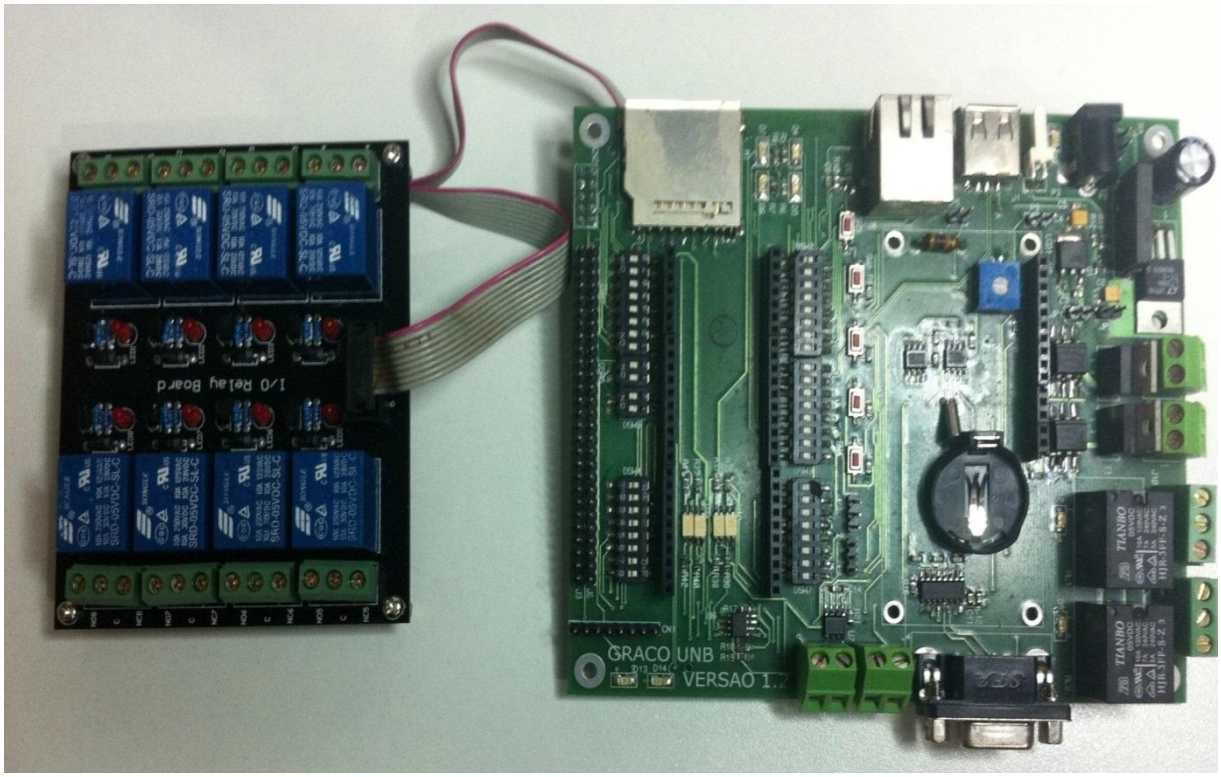


Figura 37 - Foto da placa CLP – versão 1.2 com módulo de rede

Fonte: elaborado pelo autor

Ao fazer os testes e apresentar o resultado ao orientador, foram feitas novas observações:

a) A placa estava agora com todas as modificações;
b) Era preciso validar os *hardware*s com os *software*s;
c) A placa de rele era proprietária e talvez devesse projetar uma como parte do projeto;
d) A placa de relé tinha uma limitação de distância;
e) A placa de relé não tinha alimentação própria.

3.2.5 PLACA VERSÃO 1.4 – VERSÃO FINAL

Depois dos testes e validações da versão anterior, as observações eram críticas e necessárias para este projeto, pois, realmente, apesar de ter mais oito (8) relés, a limitação de distância era um complicador para o uso do projeto como um CLP. Neste caso, foi preciso voltar às pesquisas e achar uma forma de controle à distância, sem cabos para os relés.

Após pesquisas, foram encontradas algumas opções tais como módulos RF, mas esses não tinham nenhum padrão. Então, chegou-se a um *hardware* do tipo *XBEE, que usa um padrão de protocolo chamado de* Zigbee *utilizado em aplicações de sensores sem fio*, que permite que a placa de CLP possa controlar várias placas de relés ao mesmo tempo.

ZigBee é um padrão que define um conjunto de protocolos de comunicação para dispositivos sem fio baseados em *ZigBee,* operando em 868 MHz, 915 MHz e 2,4 GHz bandas de frequência. A taxa de dados máxima é de 250 K *bits* por segundo.

Os módulos *Xbee* usados neste projeto foram criados pela *Digi International* usando como base no protocolo *ZigBEE* e possuem várias características que são abordadas em detalhes no apêndice B desta dissertação

Nesta fase do projeto, a plataforma, com a inclusão do modulo *Xbee*, abriu-se à possibilidade da inclusão de vários sensores sem fio se comunicando, criando uma rede de sensores com várias aplicações, tais como:

a) Sensor de umidade;
b) Sensor de temperatura;
c) Sensor de velocidade do vento;
d) Sensor de direção do vento;
e) Sensor de pressão atmosférica;
f) Controle de iluminação;
g) Controle de aquecimento;
h) Controle de Ventilação;
i) Controle de Irrigação;

j) Alarmes;
k) Controle de cancelas;
l) Controle de portas e portões; .
m) Aplicações automotivas.

3.3 A estrutura final da plataforma do projeto desta dissertação é composta por três placas

a) Placa *MBED.*
b) Placa Base.
c) Placa de Relés.

Definiu-se que as placas seriam *empilhadas* por conectores de empilhamento, sendo que os sinais das placas MBED e *LPCxpresso* estarão disponíveis para o usuário. Na placa base, a funcionalidade de cada pino é selecionada por interruptores*,* os quais podem ser configuráveis manualmente, tornando opcional o uso ou não dos sinais do microcontrolador na placa base, ou no meio externo.

Para melhor explicação do sistema, os seguintes tópicos serão apresentados:

a) Alimentação;
b) *LPCxpresso*;
c) *Xbee*;
d) Relógio de Tempo Real;
e) Comunicações;
f) Linhas de leitura e atuação;
g) Armazenamento em memória.

3.3.1 Alimentação

O sistema de alimentação do projeto foi pensado usando uma fonte DC 12V, conectada ao conector *Power Jack* P2. Sua alimentação é regulada para 5V e, em seguida, para 3,3V. Pode-se, também, selecionar a alimentação de 5V, através do conector USB disponível na placa. A seleção é feita através dos *jumpers* JP2 e JP3. A Tabela 1 a seguir mostra a configuração dos *jumpers*:

JP2	JP3	Alimentação
Aberto	Aberto	Sem alimentação
Aberto	Fechado	+5V através do regulador
Fechado	Aberto	+5V através da USB
Fechado	Fechado	Proibido*

*** Esta opção danificará o circuito, portanto nunca deverá ser selecionada.**

Tabela 1 - Seleção da alimentação

Fonte: elaborado pelo autor.

Para a regulação de 5V (quando selecionado), o regulador escolhido foi o LT1084CT-5, produzido pela *Linear Tech,* que possui as seguintes características:

a) Corrente máxima de saída (Imáx. 5.0A);
b) Corrente quiescente (5mA);
c) Rejeição de *ripple* na entrada: 68dB @ 120Hz;
d) Queda de tensão mínima (*Dropout Voltage)*: 1,5V @ 5A;
e) Resistência térmica: 65º/W.
f) Encapsulamento: TO-220.

Dado o consumo estimado de 150mA nesse regulador, a potência dissipada será de:

a) Pd=Vdrop*Ic*Rt;
b) Pd=(12-5)*0,15;
c) Pd=1,05W.

A potência dissipada necessita de um dissipador mecânico para auxiliar na transferência de calor e evitar superaquecimento do componente. O dissipador foi adicionado na placa. Para a alimentação de 3,3V, necessária ao *LPCxpresso e MBED*, foi selecionado o regulador LM1117DTX-3.3, produzido pela *National Semiconductor,* que possui as seguintes características:

a) Corrente máxima de saída: 800mA;
b) Corrente quiescente: 10mA;
c) Rejeição de *ripple* na entrada: 75dB @ 120Hz;
d) Queda de tensão mínima (*Dropout Voltage)*: 1.3V @ 0.8A;
e) Resistência térmica: 55º/W;
f) Encapsulamento: TO-263.

Dado o consumo estimado de 80mA nesse regulador, a potência dissipada será de:

a) Pd=Vdrop*Ic;
b) Pd=(5-3,3)*0,08;
c) Pd=0,136W.

Para esse caso, a potência dissipada não necessita de um dissipador mecânico para auxiliar na transferência de calor.

3.3.2 Alimentação externa de *backup*

O circuito de alimentação externa foi adicionado para uma possível queda de tensão não comprometer o funcionamento da placa-base, bem como as interfaces conectadas a ela. Para tanto, utilizou-se um rele de estado sólido, que chaveia a alimentação da fonte usual para uma bateria externa de *backup*. O tempo de chaveamento de alguns microssegundos é incapaz de reiniciar o microcontrolador, evitando quaisquer *glitches* na ativação dos periféricos externos. O diagrama completo da etapa de alimentação é apresentado na Figura 38.

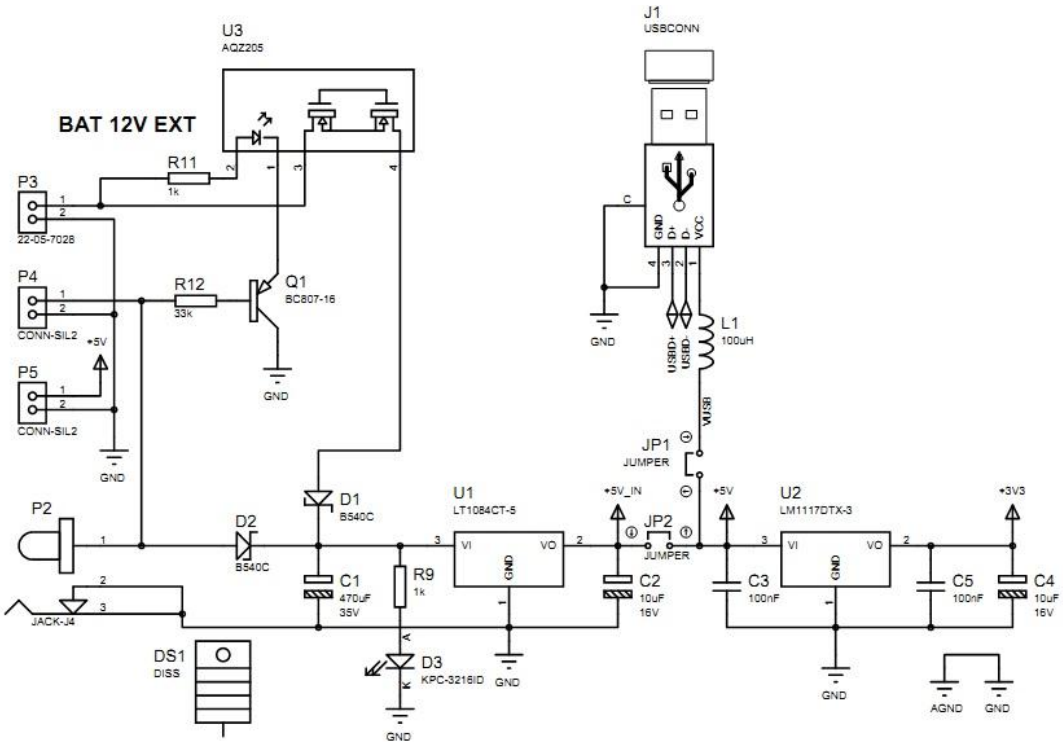


Figura 38 - Circuito de alimentação

Fonte: elaborado pelo autor

3.3.3 *LPCxpresso*

O *kit LPCxpresso* (vide Figura 39) é um placa de desenvolvimento para aplicações que utilizam o microcontrolador LPC1769, baseada no núcleo ARM *Cortex*-M3. O *kit* didático inclui um microcontrolador ARM LPC3154, que faz a interface USB com o LPC1769, sendo também utilizado como um *JTAG/Debbuger* em caso de uso do *kit* sem qualquer outra placa de apoio.

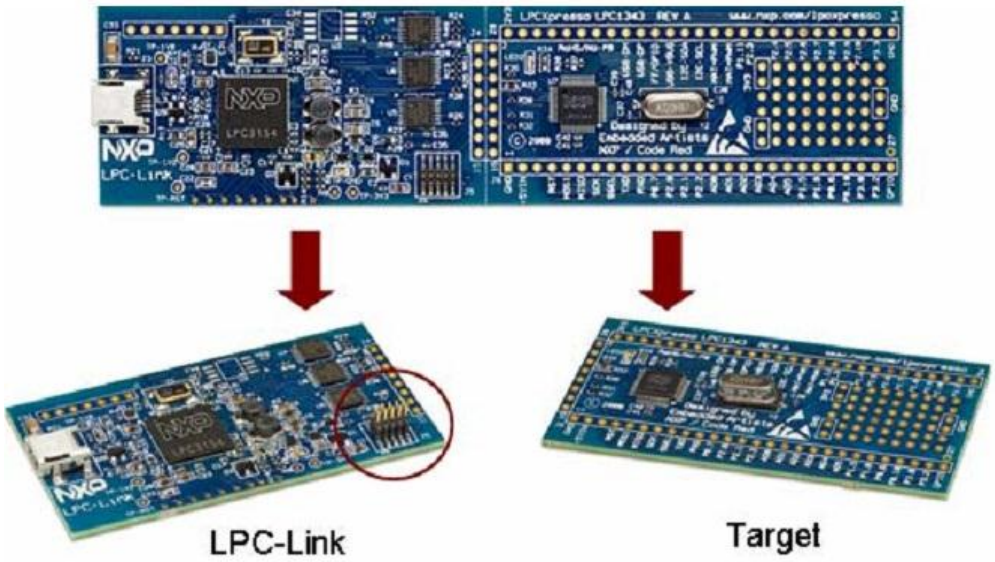


Figura 39 - *Kit* LPCxpresso

Fonte: elaborado pelo autor

Para esse caso, foi definido que seriam utilizadas as interfaces disponíveis apenas no microcontrolador "*Target"* (LPC1769) e deixar-se-ia a parte o *"LPC-Link"* para futuras aplicações.

3.3.4 Recursos disponíveis no *LPCxpresso*

O *LPCxpresso* possui os seguintes recursos:

- Microcontrolador de 32bits *NXP LPC1769*;
- Memória *Flash* para programa: 512kB;
- Memória de Dados: 64kB;
- Frequência de operação: 12MHz;
- Interfaces disponíveis:
  - 1 *Ethernet*;
  - 1 USB 2.0 F*ull-SpeedDevice/Host*;
  - 4 *UART's* (podendo ser 4 RS232 ou 3 RS232 + 1 RS422/RS485);
  - 1 CAN 2.0B;
  - 1 Interface SPI e 2 SSP (podendo ser configuráveis para SPI);
  - 3 Interfaces I2C;
  - 3 Interface I2S;
  - 2 Interfaces de Quadratura de Encoder;
  - 2 Interfaces de PWM;
- 70 pinos de GPIO;
- 8 canais de ADC de 12-bits
- 1 saída DAC de 10-bits;
- *Watchdog;*
- Relógio de Tempo Real (RTR);
- Consumo: 7mA @ 12MHz;
- Modos de operação:
  - *Sleep*: 2 mA;
  - *Deep Sleep:* 240 uA;
  - *Power Down:* 31uA;
  - *Deep Power Down* (apenas RTR operando): 630nA;
- Dimensões: 35 mm x 140mm.

Para tanto, um *software* embarcado foi desenvolvido, o qual comunica com todas as interfaces, armazena valores e processa quaisquer dados inseridos pelo usuário. A Figura 40 mostra o diagrama em blocos do LPC1769.

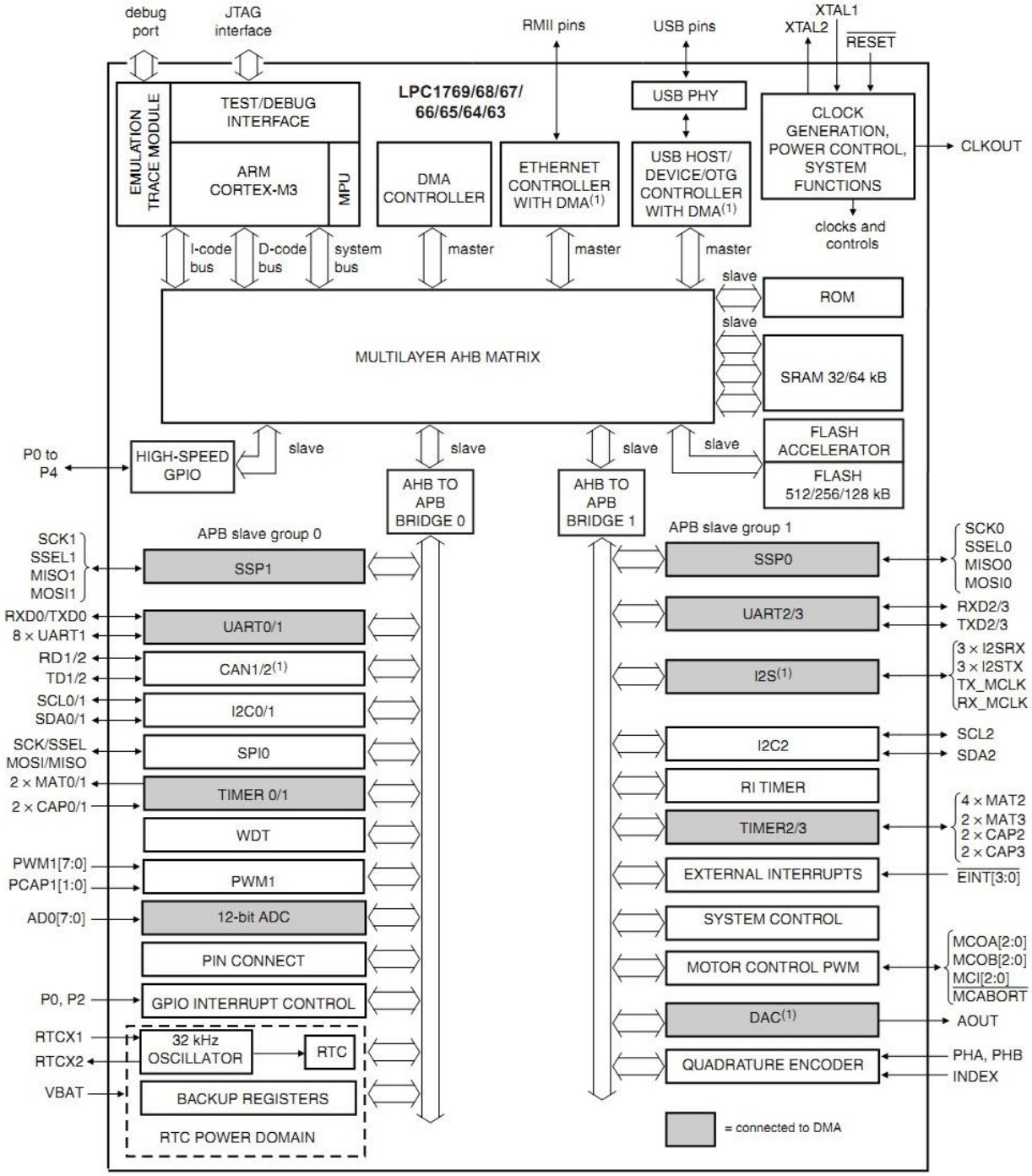


Figura 40 - Diagrama em blocos do LPC1769

Fonte: www.lpcware.com/

3.3.5 MBED

O Kit MBED é uma placa de desenvolvimento de microcontroladores ARM, projetado para prototipagem rápida, sendo projetada para todos os tipos de protótipos de dispositivos, especialmente aqueles incluindo *Ethernet* e USB.

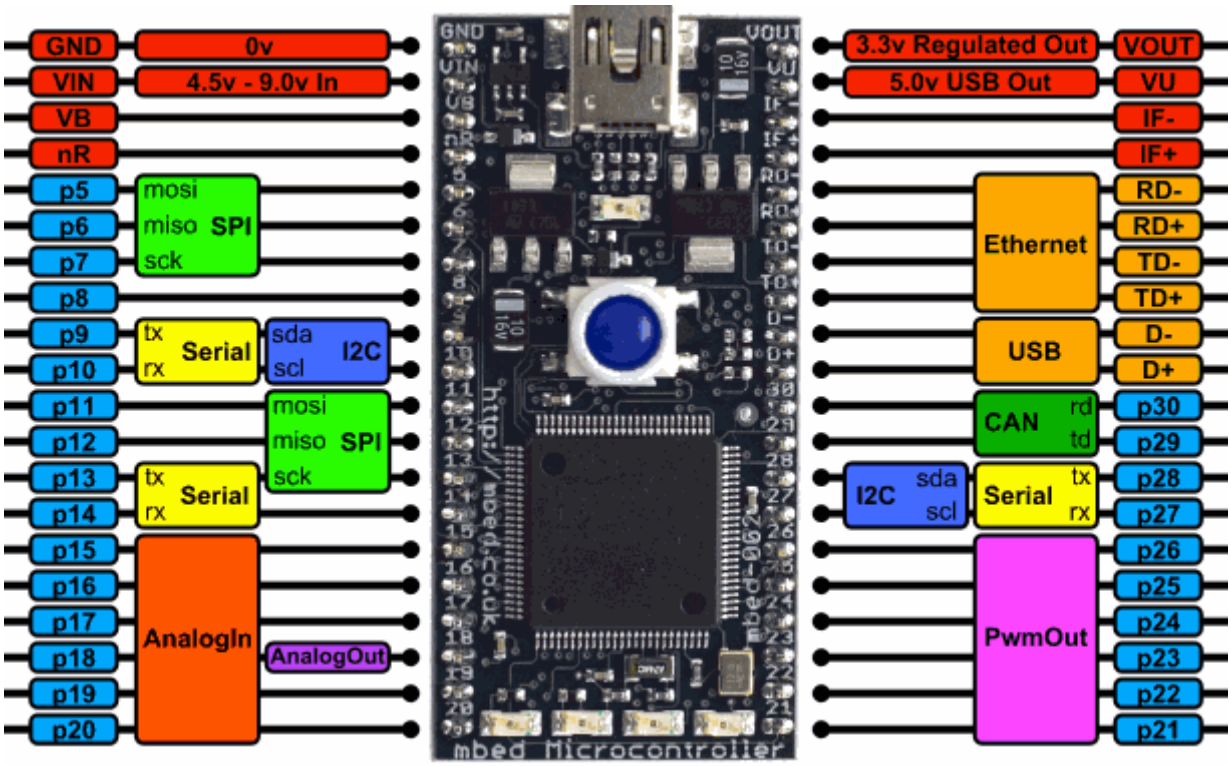


Figura 41 - Diagrama em blocos do LPC1769

Fonte: MBED.ORG

A placa baseia-se no microncontrolador NXP LPC1768, com um núcleo ARM Cortex-M3 32 *bits* rodando a 96MHz. A mesma inclui 512KB FLASH, 32KB de RAM e varias interfaces, incluindo *Ethernet*, USB Host, CAN, SPI, I2C, ADC, DAC, PWM e outras interfaces de I/O. A pinagem acima mostra as interfaces comumente usadas e suas localizações. Pode ser observado que todos os pinos numerados (p5-p30) podem também ser usados como *digital-in* e *digital-out*.

O *MBED* possui os seguintes recursos:

- Microcontrolador de 32bits *NXP LPC1768*;
- Memória *Flash* para programa: 512kB;
- Memória de Dados: 64kB;
- Frequência de operação: 12MHz;
- Interfaces disponíveis:
  - 1 *Ethernet*;
  - 1 USB 2.0 F*ull-SpeedDevice/Host*;
  - 4 *UART's* (podendo ser 4 RS232 ou 3 RS232 + 1 RS422/RS485)
  - 1 CAN 2.0B;
  - 1 Interface SPI e 2 SSP (podendo ser configuráveis para SPI);
  - 3 Interfaces I2C;
  - 3 Interface I2S;
  - 2 Interfaces de Quadratura de Encoder;
  - 2 Interfaces de PWM;
- 70 pinos de GPIO;
- 8 canais de ADC de 12-bits
- 1 saída DAC de 10-bits;
- *Watchdog;*
- Relógio de Tempo Real (RTR);
- Consumo: 7mA @ 12MHz;
- Modos de operação:
  - *Sleep*: 2 mA;
  - *Deep Sleep:* 240 uA;
  - *Power Down:* 31uA;
  - *Deep Power Down* (Apenas RTR operando): 630nA;
- Dimensões: 35 mm x 140mm;

3.3.6 A interface *XBEE*

Neste projeto, a interface *Xbee* é responsável por comunicar o módulo do *Xbee* com o módulo *Xbee* da *placa-reles* (vide **Figura 42**). O módulo *Xbee* possui uma linha serial UART para interfacear com o processador, tornando a comunicação *wireless* transparente na linha. Abaixo são descritas algumas características do módulo *Xbee*:

- Alcance de 30m a 3 km ;
- Taxa de transmissão/recepção em RF: 250kbps;
- Taxa de transmissão/recepção serial: 1200bps – 250kbps;
- Consumo:
  - 45mA @ 3,3V (transmissão);
  - 50mA @ 3,3V (recepção);
  - <10uA (*Power-down*);
- IOs de uso geral;
- Dimensões: ~2,4 x 2,7 cm

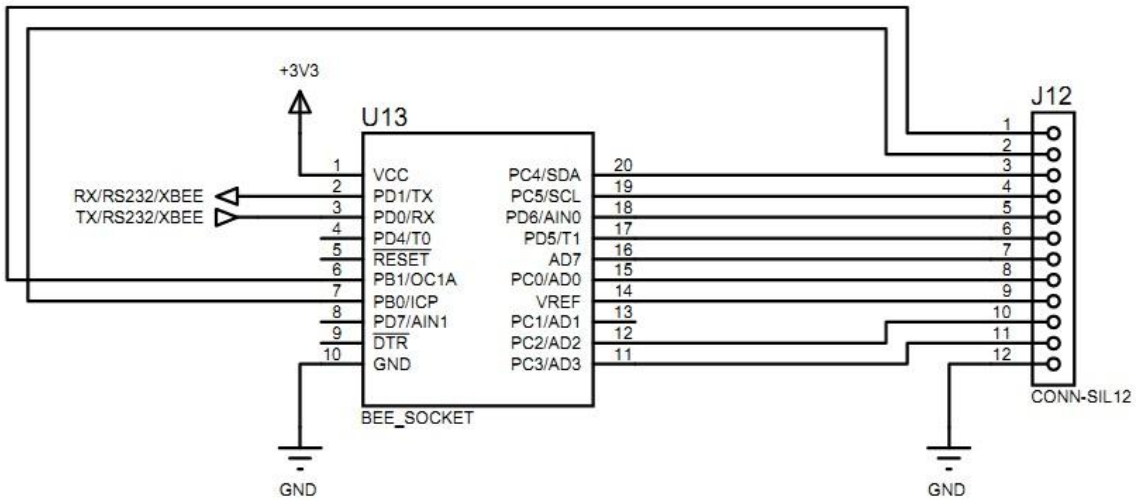


Figura 42 – circuito *Xbee*

Fonte: elaborado pelo autor

3.3.7 Relógio de Tempo Real

Para fins de contagem do tempo real e economia de consumo da bateria, utiliza-se o Relógio de Tempo Real DS1307 produzido pela *Dallas Semiconductor,* que possui as seguintes características:

- Contagem real em segundos, horas, minutos, dias, meses e anos (inclusive ano bissexto);
- 56 bytes de memória não volátil;
- Consumo < 500nA;
- Interface I2C.

Neste caso, foi inserido um cristal de 32,768 kHz 20ppm para temporização interna do RTR. A comunicação I2C será feita através dos pinos 9 e 10 do conector P1 do barramento (pinos da interface I2C1). Uma bateria do tipo “moeda” de 3V de lítio (modelo CR2032) foi utilizada para alimentar o *chip* e efetuar a contagem do tempo mesmo com a alimentação principal desligada. Como o *LPCxpresso,* possui um RTR também, foi adicionado um *jumper* para a linha “VBAT” do conector P1, no caso da opção de uso deste RTR. O diagrama do circuito é apresentado na Figura 43.

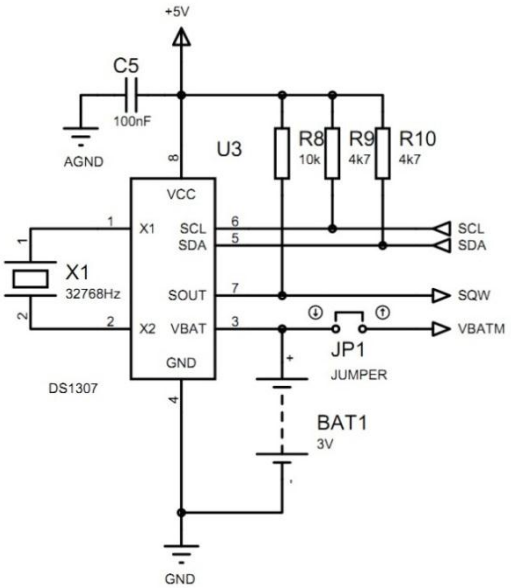


Figura 43 - Circuito do RTR

Fonte: elaborado pelo autor

A opção de utilização desse circuito RTR, ao invés do circuito já presente nas placas *LPCxpresso e MBED,* garante as vantagens de maior estabilidade.

Finalmente, tem-se um circuito dedicado a essa tarefa, diferentemente das placas *LPCxpresso* e *MBED,* em que o RTR é integrado ao processador e, portanto, depende da ativação e do controle através do *software* embarcado, podendo ocasionar instabilidade. Além disso, obteve-se uma maior autonomia da bateria, visto que o consumo é duas vezes menor que o do *LPCxpresso* (indicado no *datasheet* em 1.1uA com os reguladores de tensão desalimentados). Por outro lado, a placa utilizada possui aproximadamente três vezes mais memória para armazenamento de tempo decorrido do que o RTR presente na placa *LPCxpresso* (56 *bytes* do circuito contra 20 *bytes* do *LPCxpresso*).

3.3.8 O projeto de comunicação da placa

3.3.8.1 A comunicação *Ethernet*

A comunicação *Ethernet* é disponibilizada a fim de criação de redes locais. Baseada no envio de pacotes e padronização da IEEE como 802.3, a mesma é capaz de se comunicar com protocolos 10/100 Base-T. Com isto, pode-se conectar a placa base a uma rede de computadores, na qual todos podem acessar, alterar e visualizar quaisquer dados disponibilizados pela placa. A implementação da comunicação *Ethernet* está parcialmente feita no *LPCxpressoe e MBED* (vide Figura 44), a qual possui um *transceiver* e seus sinais são ligados ao conector. Por sua vez, estes são ligados a um conector padrão RJ45 na placa-base, com uma bobina de acoplamento embutida, a qual elimina interferências geradas pelo cabo.

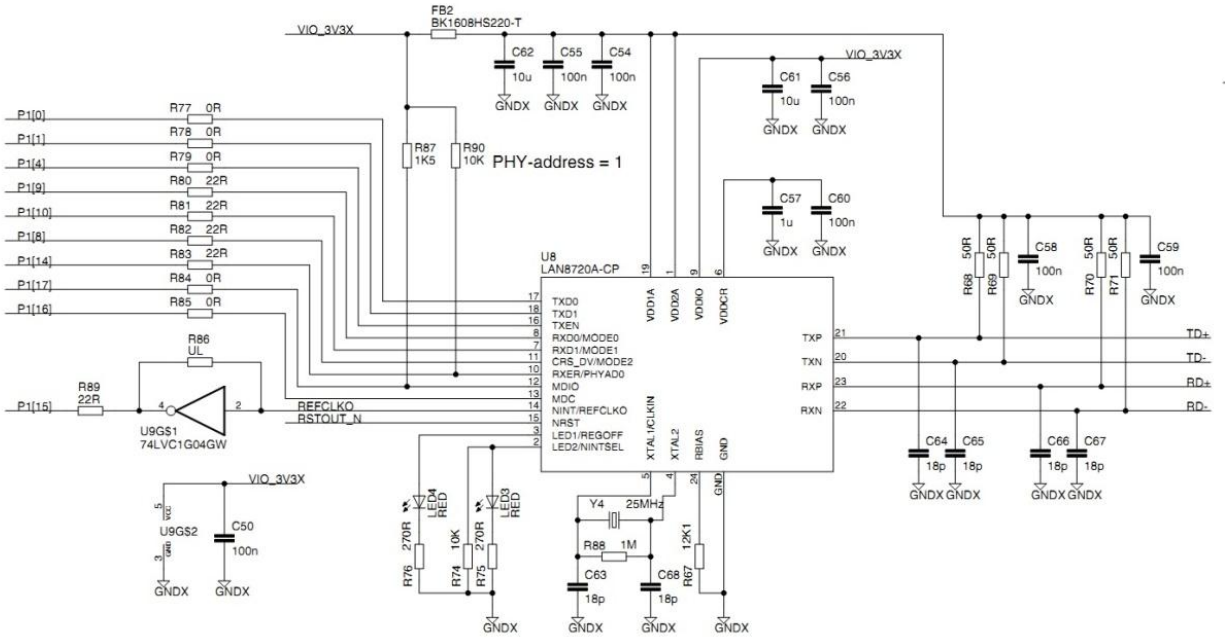


Figura 44 - Circuito *transceiver* presente na Placa LPCxpresso

Fonte: elaborado pelo autor

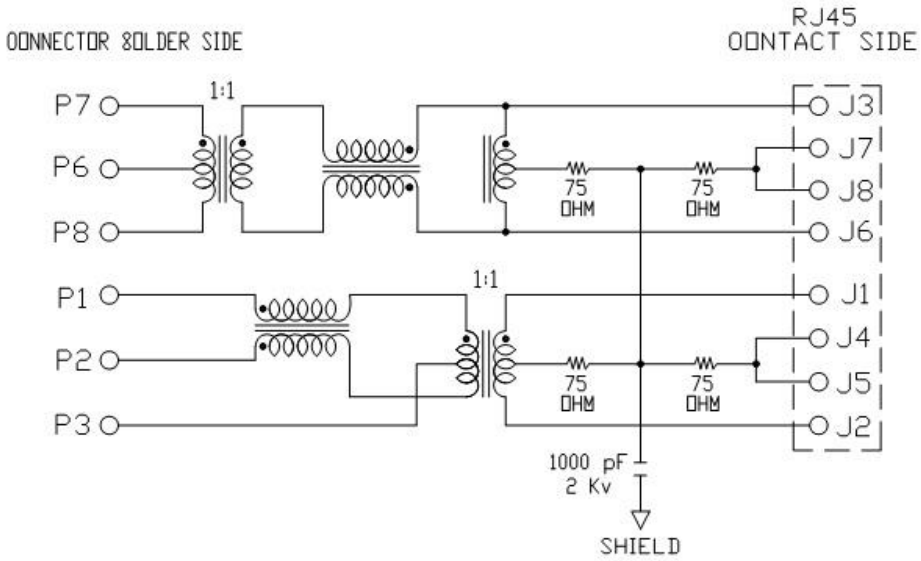


Figura 45 - Bobina de acoplamento embutida no conector J9

Fonte: sparkfun.com

3.3.8.2 A interface USB

Através desta USB, pode-se conectar um computador como hospedeiro e a placa como dispositivo, no qual se podem acessar recursos da placa e seus periféricos. Adicionalmente, é possível atualizar o *firmware* da placa através desta interface.

A porta USB pode ser utilizada como padrão USB 1.1 *Low Speed* com taxa de transferência de até 1,5 Mbps ou como padrão USB 2.0 *Full Speed* com taxa de transferência de até 12 Mbps. Como essa função é integrada ao *LPCxpresso e MBED*, a figura 45 mostra a etapa e os sinais aos quais são conectadas as linhas USB no *LPC-Link*, observando que já existe um conector mini-usb presente na *LPCxpresso*. Apenas os sinais “USB_DM” e “USB_DP” seguem para a placa base. A Figura 46 demonstra as conexões ligadas ao conector J1 tamanho padrão presente na placa-base.

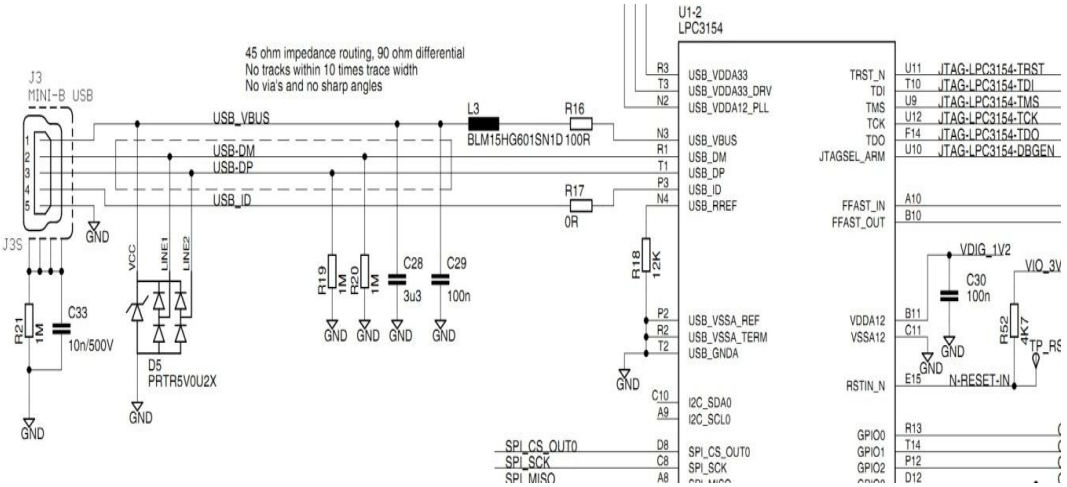


Figura 46 - Circuito USB presente na placa *LPCxpresso*

Fonte: elaborado pelo autor

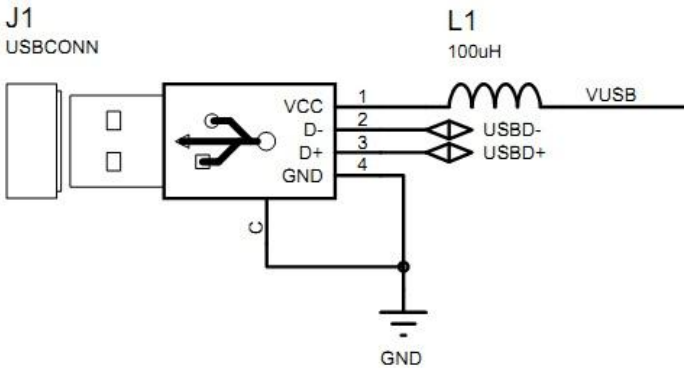


Figura 47 - Conector USB da Placa Base

Fonte: elaborado pelo autor

Um conector USB, em tamanho padrão, provê maior facilidade de utilização por parte do usuário.

3.3.8.3 A interface RS232

O padrão *Recommended Standard* 232 (RS232) é uma comunicação serial assíncrona, capaz de trafegar informações de transmissão e recepção simultaneamente, a uma taxa de até 115200 bps. Uma comunicação prática para o caso de outras placas microcontroladas, por exemplo. Neste caso, foi utilizada uma das saídas UART do *LPCxpresso/MBED*, um *transceiver* MAX232, produzido pela Maxim. O diagrama é mostrado na Figura 51.

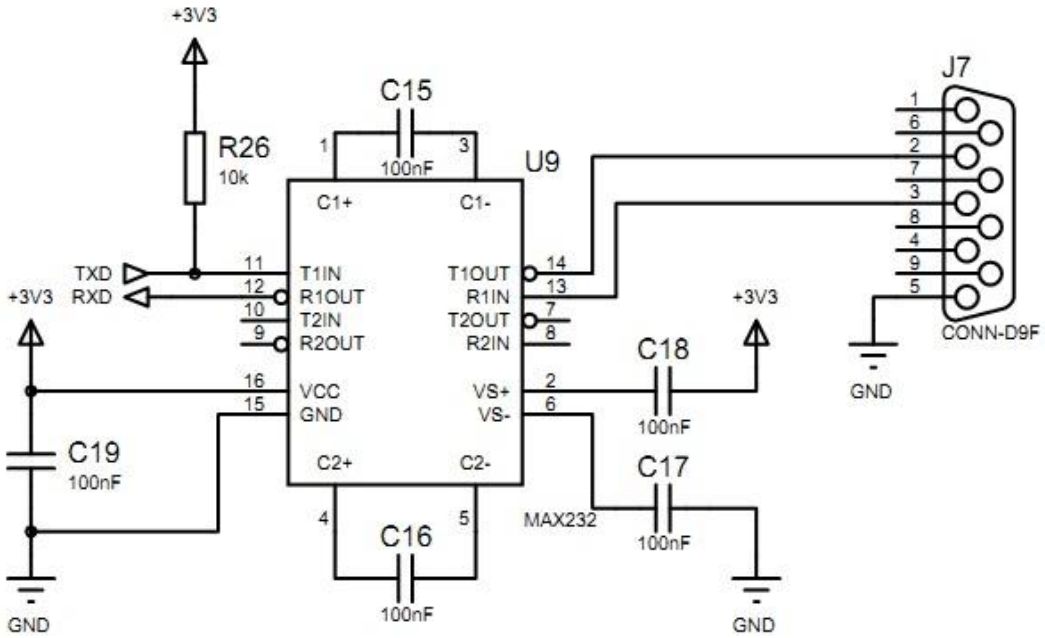


Figura 48 - Circuito *transceiver* RS232

Fonte: elaborado pelo autor

3.3.8.4 A interface RS485

Similar ao padrão RS232, a comunicação RS485 está presente na placa para aplicações de equipamentos mais robustos. Por permitir uma distância de 1500m de extensão do cabo, sua utilização pode ser específica para aplicações críticas, na qual o acesso humano torna-se complexo ou mesmo impossível quando operante.

Também é utilizada uma das UART's disponíveis no *LPCxpresso/MBED* para comunicação com o *transceiver*. O *transceiver* SN75176BP (produzido pela *Texas Instrument)* foi utilizado para o projeto da placa-base, conforme ilustra o diagrama elétrico mostrado na Figura 49.

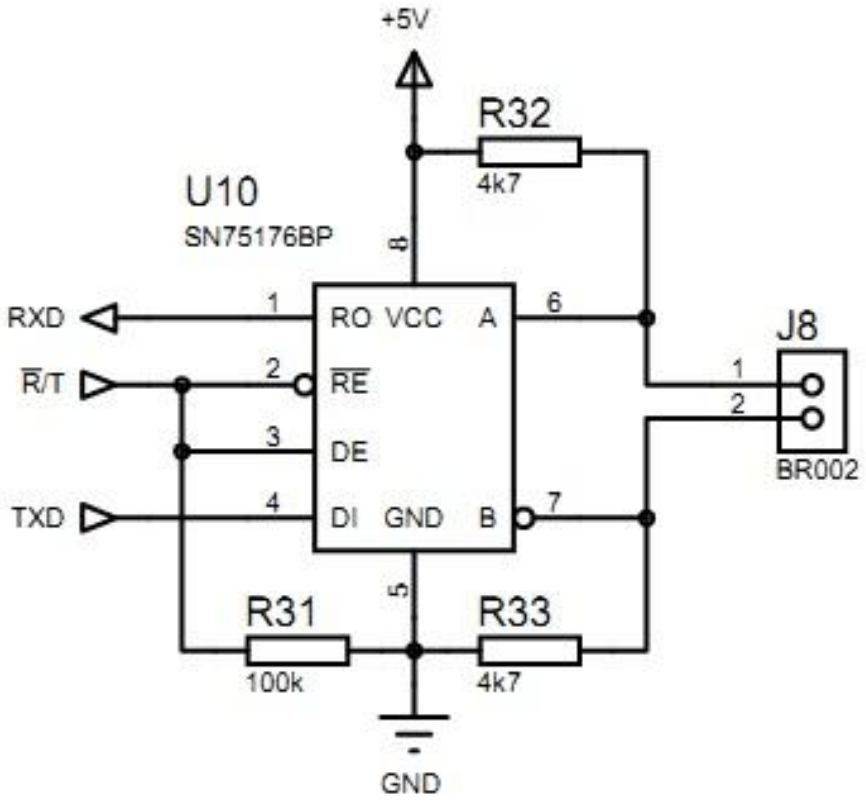


Figura 49 - Circuito *transceiver* RS485

Fonte: elaborado pelo autor

3.3.8.5 A interface CAN

Uma interface *Controlled Area Network* (CAN) foi adicionada ao sistema, para interface com dispositivos ou sistemas completos que adotam essa comunicação. Utilizada no meio automotivo, torna-se fácil, rápida e resistente a ruídos, além de não necessitar de um dispositivo *host* (dado que a mensagem de maior prioridade na rede será dada pelo alto valor numérico).

O *LPCxpresso* e o MBED já possui integrado em seu microcontrolador um controlador CAN 2.0, capaz de se comunicar com velocidades de até 1 Mbps e até 29 *bits* de comprimento de cada pacote. No entanto, para adequar os níveis de tensão do microcontrolador para o meio externo, foi adicionado um *transceiver* na placa base, conforme diagrama mostrado na Figura 50.

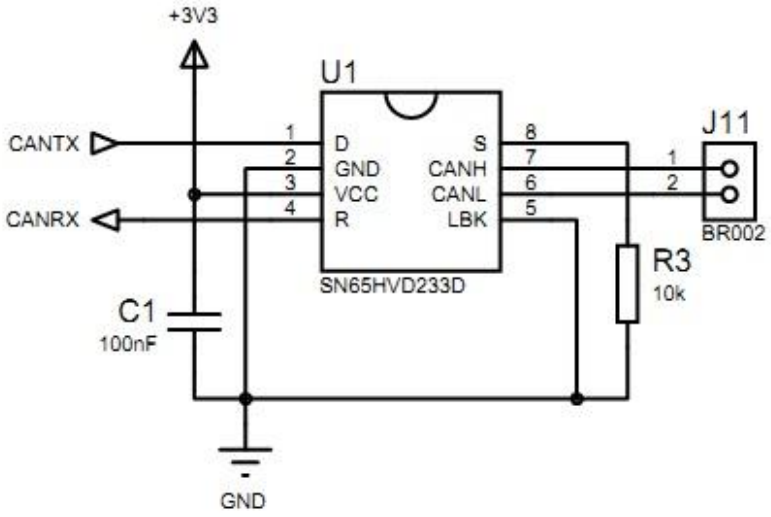


Figura 50 - Circuito *transceiver* CAN

Fonte: elaborado pelo autor

3.3.9 Linhas de leitura e atuação da placa base

As linhas de leitura e atuação do sistema são todas aquelas linhas capazes de interfacear diretamente com algum periférico, sem a necessidade de um *transceiver*, como é feito nas comunicações. As linhas de leitura são as seguintes:

a) 4 linhas de entrada digitais: entradas discretas digitais (0 e 5V), de até 100kHz. As entradas possuem um resistor de *pull-down* de 10k. Todas as entradas são optoisoladas pelo optoisolador ACPL-227, para atender as normas de segurança exigidas pela NBR-5410. Sendo assim, trata-se de um sistema PELV (do inglês "*protected extra-low voltage*"). O diagrama do circuito é mostrado na Figura 51.

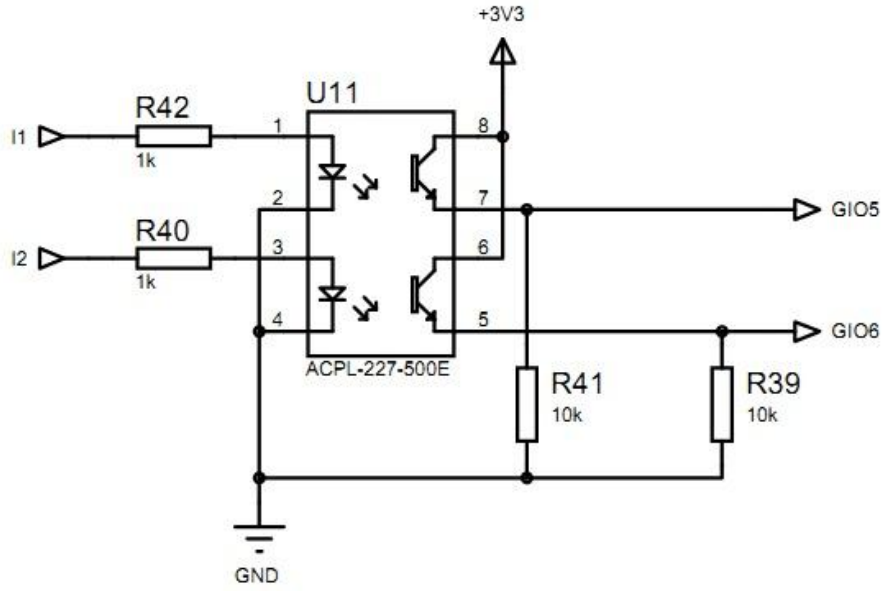


Figura 51 - Isoladores ópticos das linhas digitais

Fonte: elaborado pelo autor

b) 5 canais de leitura analógica: Canais disponibilizados para leitura do ADC de 12-*bits* do microcontrolador. Suporta tensões de 0 a 3Vdc à frequência máxima de 200kHz. As entradas analógicas são feitas através do conector de expansão, conforme mostrado na Figura 52.

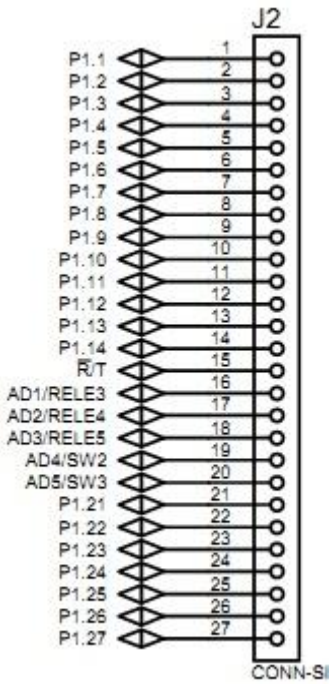


Figura 52 - Entradas analógicas

Fonte: elaborado pelo autor

As linhas de atuação e de saída discreta são duas linhas de controle de cargas DC de até 10A através de relés. A comutação dos sinais é feita através de um transistor ligado à bobina, que comuta o estado ao transistor ao atingir o nível de saturação. Dada a resistência elétrica da bobina, de aproximadamente 300 ohms, os cálculos para a corrente de base do transistor são os seguintes:

a) Vbe = 0,7 @ 10mA;
b) R_{bobina} = 400Ω;
c) h_{fe} = 200;
d) Ic = (5 – 0,7) / 400 = 10,75 mA + I consumo do led = 15,75mA;
e) Ib = Ic / h_{fe} =15,75m / 200 = 78,75uA.

Para uma corrente mínima de base de 78,75uA, a resistência máxima será de: R = Vb / Ib = 3 / 78,75u = 38,09 kΩ. Para tanto, foi adotada uma resistência de 1k8, capaz de garantir uma corrente muito acima da corrente de saturação do transistor. Os *leds* D6 e D8 indicam o estado de cada carga. O diagrama da Figura 53 mostra uma das linhas.

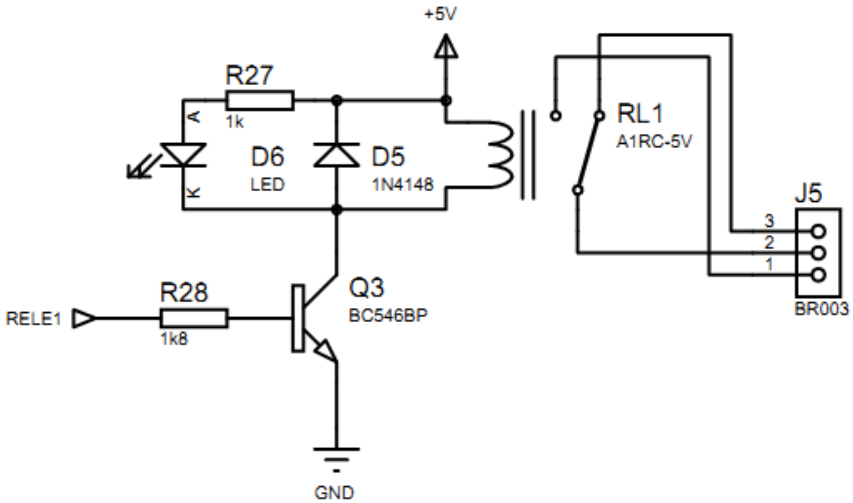


Figura 53 - Circuito de controle dos relés

Fonte: elaborado pelo autor

Duas linhas de controle de cargas AC de até 12A. Por se tratarem de cargas de corrente alternada e suscetíveis a danos, tanto no circuito de entrada quanto ao operador, a linha está opto-isolada através do *chip* MOC3021 e sua parte é isolada do comando vindo do microcontrolador. Sendo assim, a solução obtida atende às normas de segurança da NBR-5410, isolando qualquer interferência de alguma das fases da rede elétrica.

Assim como nas cargas DC, as linhas AC possuem *leds* sinalizadores de estados (D1 e D2). O diagrama de uma das linhas de controle é apresentado na Figura 54.

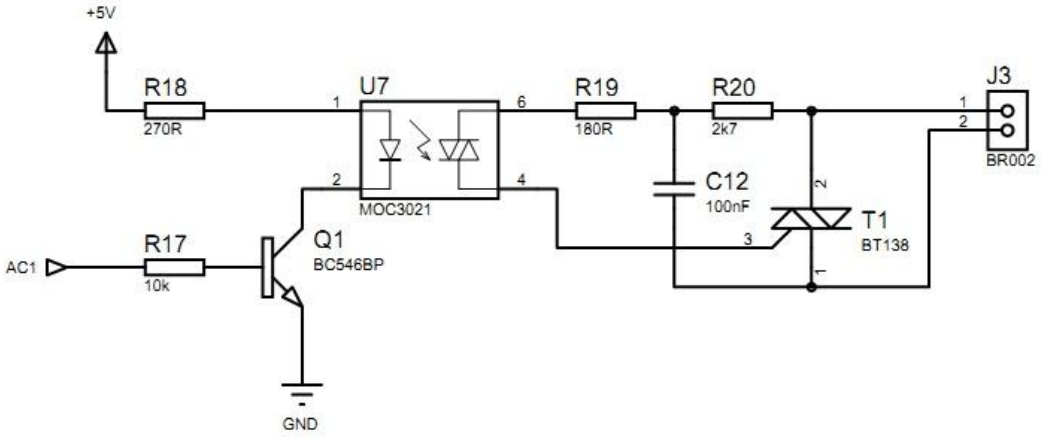


Figura 54 - Linha de controle de Carga AC

Fonte: elaborado pelo autor

Quatro (4) linhas de saída discretas são disponibilizadas para uso geral externo. São linhas digitais (0 a 3,3V), diretamente controladas pelas IO's do microcontrolador. Estão disponibilizadas no conector de expansão CN1.

3.3.10 O projeto de memória da placa base

O sistema projetado de memória consta de uma memória EEPROM 24C04A, produzida pela *Microchip*, que possui as seguintes características:

- 4Kb de dados armazenamento não volátil;
- Alimentação de 5V;
- Interface de comunicação I2C;
- Possibilidade de até 1 milhão de regravações;
- Encapsulamento: DIP-8.

Essa memória pode servir para armazenar algum *firmware* do microcontrolador, configurações pré-determinadas, etc. O diagrama elétrico é mostrado na Figura 55.

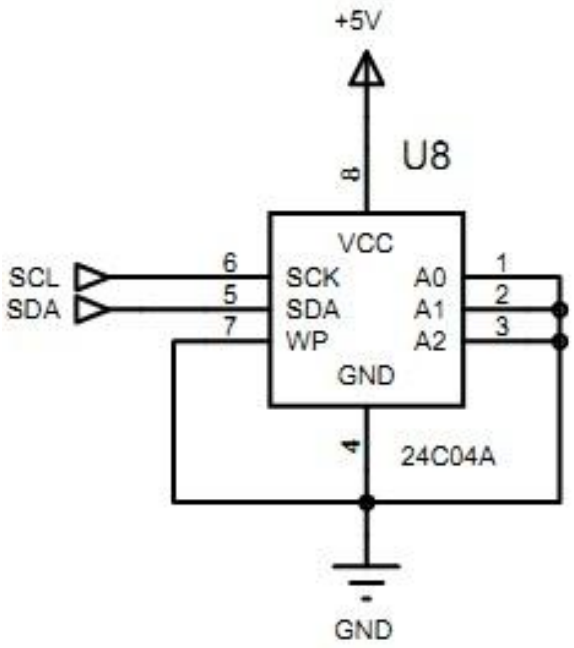


Figura 55 - Circuito de interface com a EEPROM

Fonte: elaborado pelo autor

A segunda memória é disponibilizada através de um cartão padrão SD (*Secure Digital*), capaz de armazenar dados em massa. Sua disponibilidade no mercado varia de unidades de MB a dezenas de GB. Portanto, a mesma serve tanto para aplicações de armazenamento de grandes arquivos, como aquisições de longa duração.

A placa-base possui um conector SD que recebe os cartões. O cartão fica a critério de o usuário utilizá-lo ou não. Sua comunicação é através da linha SPI disponibilizada pelo microcontrolador. O diagrama do conector é mostrado na Figura 56.

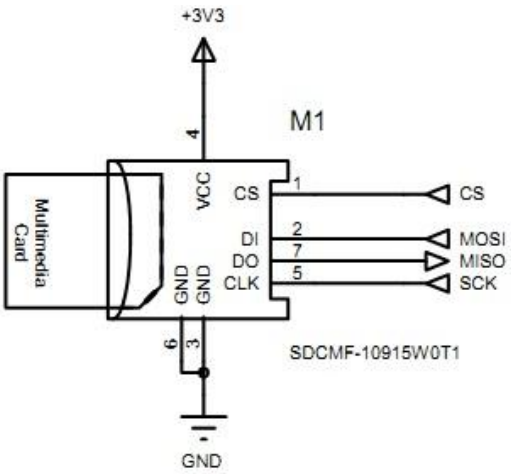


Figura 56 - Interface com o Cartão SD

Fonte: elaborado pelo autor

3.3.11 Mapa de IO

A Tabela 2 – Pinagem e suas funções apresenta todas as conexões realizadas entre a placa *LPCxpresso/MBED* e a placa-base, bem como a descrição e a funcionalidade dos pinos. Todos os pinos possuem a seleção por interruptores, localizados imediatamente à lateral de cada pino, sendo assim, na posição ON, o pino assumirá a função da placa (3ª coluna), na posição OFF, ele assumirá a função original da placa *LPCxpresso/MBED* (última coluna).

Tabela 2 – Pinagem e suas funções

Pino	**Sinal**	**Função**	**Descrição**	**Pino *LPCxpresso/MBED***
1	GND	Alimentação	Sinal negativo da alim.	GNDX
2	+5V	Alimentação	Sinal positivo da alim.	EXT_POWX
3	VBATM	Alimentação	Alimentação Bateria RTR	VB
4	RESET	Controle	Sinal de *reset* do microcontrolador	RESET_N
5	SQW	Controle	Onda quadrada oriunda do RTR da placa-base	P0[9]
6	CS	Comunicação	Sinal CS da linha SPI0	P0[8]
7	SCK	Comunicação	Sinal SCK da linha SPI0	P0[7]
8	SW1	Controle	*Switch* 1 de uso geral do usuário	P0[6]
9	SDA	Comunicação	Sinal SDA da linha I2C1	P0[0]
10	SCL	Comunicação	Sinal SCL da linha I2C1	P0[1]
11	MOSI	Comunicação	Sinal MOSI da linha SPI0	P0[18]
12	MISO	Comunicação	Sinal MISO da linha SPI0	P0[17]
13	TX/RS232/*XBEE*	Comunicação	Sinal de recepção (em relação ao *LPCxpresso*) da linha serial RS-232	P0[15]

14	RX/RS232/*XBEE*	Comunicação	Sinal de transmissão (em relação ao *LPCxpresso*) da linha serial RS-232	P0[16]
15	R/T	Comunicação	Sinal de controle de direção da linha serial RS-485	P0[23]
16	AD1/RELE3	Analógica	Canal 1 de entrada analógica/Saída de controle do Rele 3	P0[24]
17	AD2/RELE4	Analógica	Canal 2 de entrada analógica/Saída de controle do Rele 4	P0[25]
18	AD3/RELE5	Analógica	Canal 3 de entrada analógica/Saída de controle do Rele 5	P0[26]
19	AD4/SW2	Analógica	Canal 4 de entrada analógica	P0[30]
20	AD5/SW3	Analógica	Canal 5 de entrada analógica	P0[31]
21	GIO7	Controle	Entrada optoacoplada	P0[2]
22	GIO8	Controle	Entrada optoacoplada	P0[3]
23			Aberto	P0[21]
24	TESTE1	Controle	Sinal de teste geral	P0[22]
25	TESTE2	Controle	Sinal de teste geral	P0[27]
26			Aberto	P0[28]
27			Aberto	P2[13]
28	+3V3	Alimentação	Sinal positivo da alim.	VIO_3V3X
29	-			
30	-			
31	-			
32	RD-	Comunicação	Sinal negativo de recepção do par trançado da linha *Ethernet*	RD-
33	RD+	Comunicação	Sinal positivo de recepção do par trançado da linha *Ethernet*	RD+
34	TD-	Comunicação	Sinal negativo de transmissão do par trançado da linha *Ethernet*	TD-
35	TD+	Comunicação	Sinal positivo de transmissão do	TD+

			par trançado da linha *Ethernet*	
36	USBD-	Comunicação	Sinal negativo de dados da linha USB	USB-D-
37	USBD+	Comunicação	Sinal positivo de dados da linha USB	USB-D+
38	CANRX	Comunicação	Sinal de recepção da linha CAN	P0[4]
39	CANTX	Comunicação	Sinal de transmissão da linha CAN	P0[5]
40	AC1/RELE1	Controle	Sinal de controle de acionamento do Triac 1 (Carga AC A) /Saída de controle do Rele 1	P0[10]
41	AC2/RELE2	Controle	Sinal de controle de acionamento do Triac 2 (Carga AC B) /Saída de controle do Rele 2	P0[11]
42	LED1/RELE6	Controle	Sinal de controle de saída discreta /Saída de controle do Rele 6	P2[0]
43	LED1/RELE7	Controle	Sinal de controle de saída discreta /Saída de controle do Rele 7	P2[1]
44	GIO5/RELE8	Controle	Entrada optoacoplada/Saída de controle do Rele 8	P2[2]
45	GIO6	Controle	Entrada optoacoplada	P2[3]
46	RELE_A	Controle	Saída de controle do Rele A	P2[4]
47	RELE_B	Controle	Saída de controle do Rele B	P2[5]
48			Aberto	P2[6]
49			Aberto	P2[7]
50			Aberto	P2[8]
51			Aberto	P2[10]
52			Aberto	P2[11]
53			Aberto	P2[12]
54	GND	Alimentação	Sinal negativo da alim.	GNDX

3.4 O projeto da placa de reles

A placa-rele é um sistema de expansão para sinais que controlam a alimentação de dispositivos externos. A placa-reles possui os seguintes circuitos:

- Alimentação de 5V e 3,3V;
- Conector para módulo *Xbee*;
- Opto-isolamento das saídas do *Xbee*;
- Ativação dos relés;
- Conectores de alimentação e cabo de dados (quando necessário).

Todos os circuitos acima são idênticos aos da placa-base, com as devidas modificações. Essa expansão possibilita o uso de mais oito (8) controles de dispositivos utilizando a placa base. Os sinais são acessíveis através de dois modos: (a) um deles é através da comunicação sem fio *Xbee;* ou (b) através do cabo de dados, ligado através de um cabo *flat* à placa-base..

3.5 Leiaute da placa-base e da placa-relés

Para maior facilidade de manuseio da placa, foram consideradas possibilidades de fácil acesso ao usuário aos conectores das linhas de atuação digitais/analógicas, linhas de comando DC e AC, comunicações e alimentação. Também, a placa possui indicações na serigrafia para instruir o operador que irá manusear a placa.

3.5.1 Apresentação e requisitos do Lelaute

O leiaute do circuito impresso da Placa Base são mostradas na Figura 57 e 64.

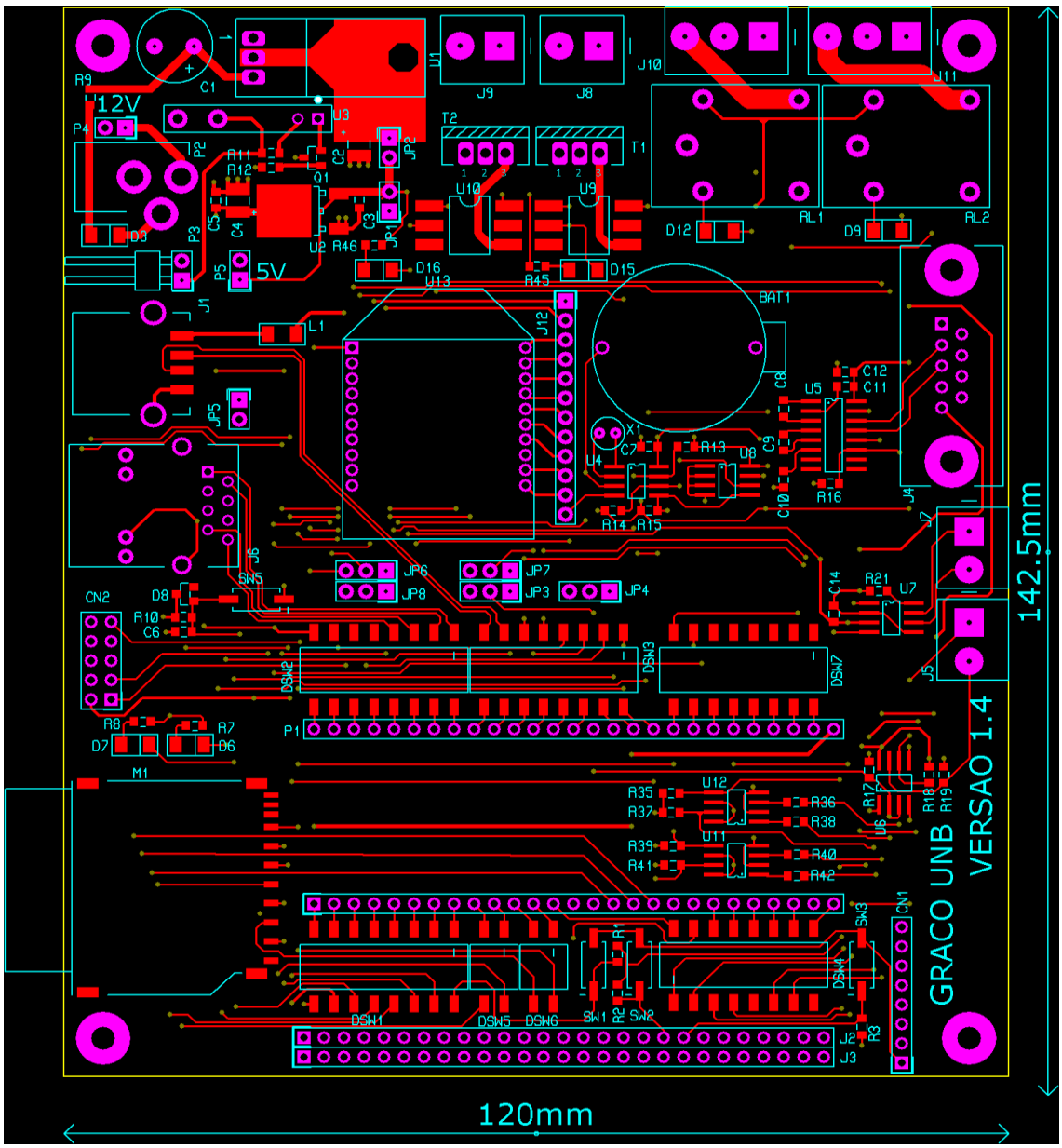


Figura 57 – Leiaute, face de cima (dimensões não reais)

Fonte: elaborado pelo autor

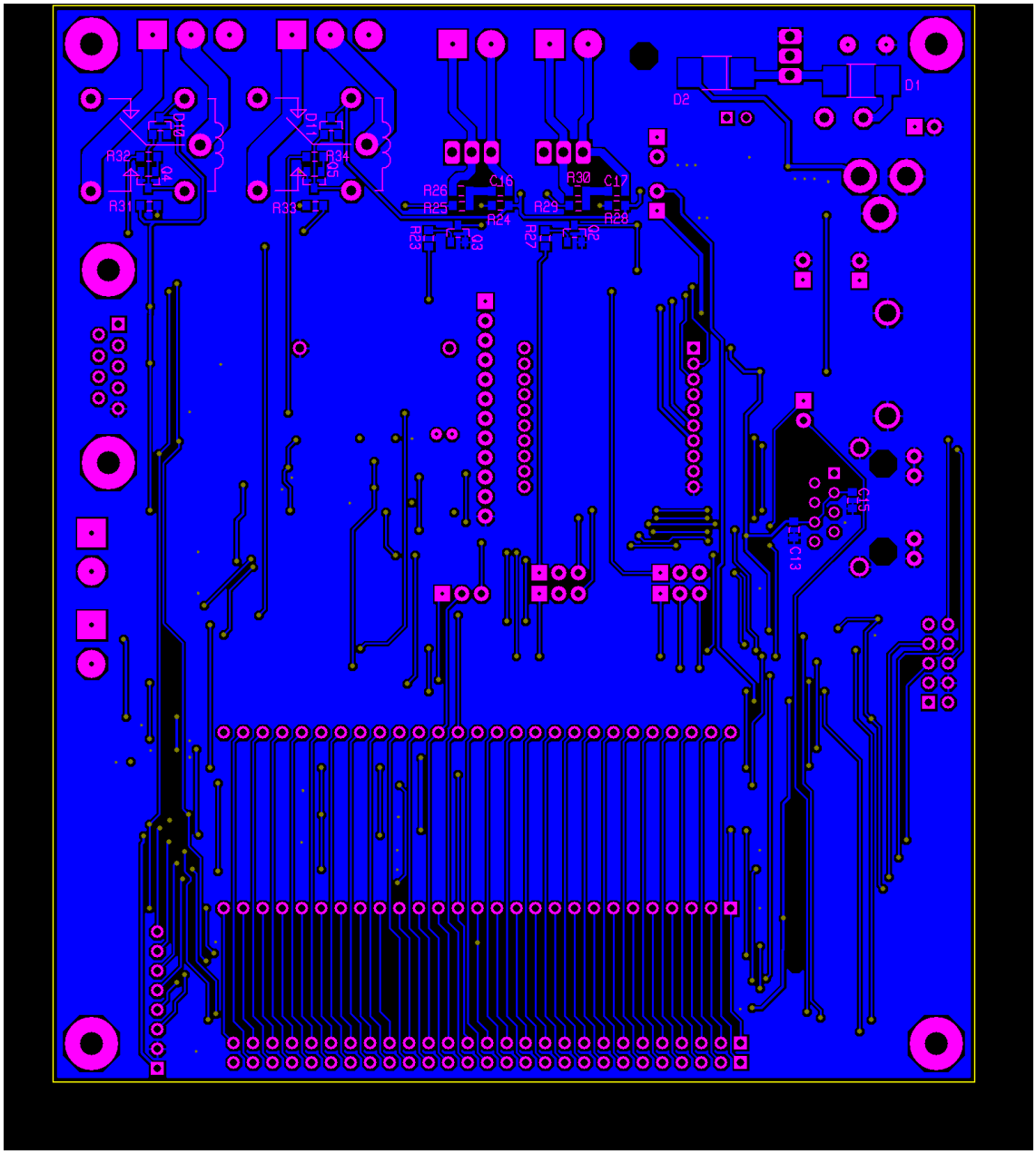


Figura 58 - Leiaute face de baixo (dimensões não reais)

Fonte: elaborado pelo autor

O *layout* da placa-rele é mostrado nas figuras 65 e 66

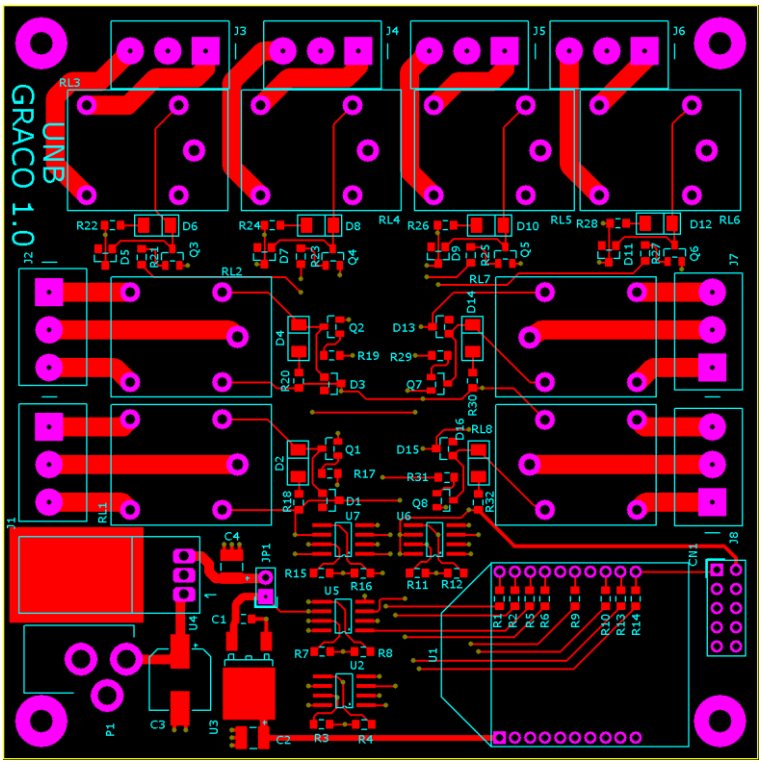


Figura 59 - Leiaute face de cima (dimensões não reais)

Fonte: elaborado pelo autor

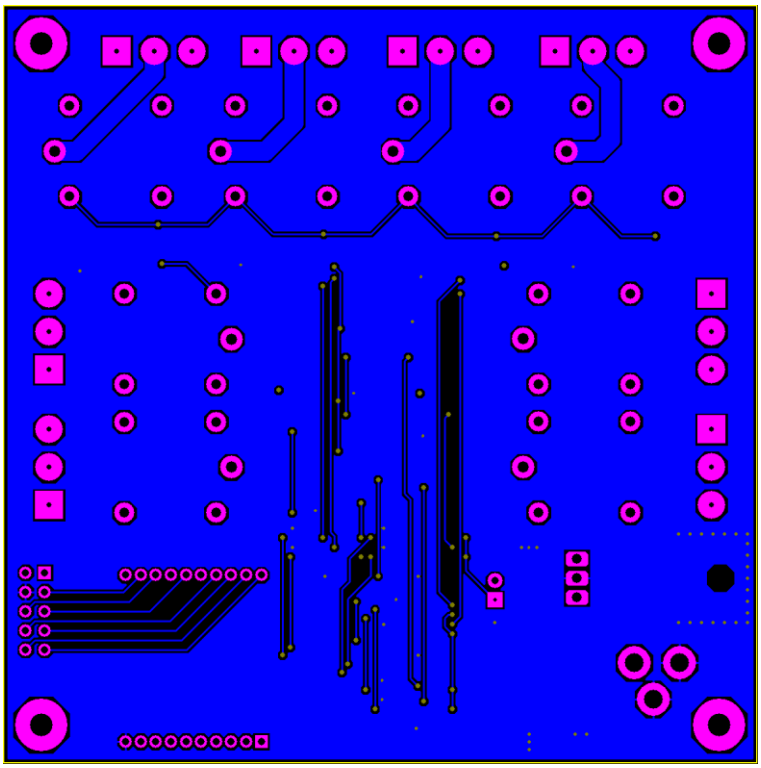


Figura 60 - Leiaute face de baixo (dimensões não reais)

Fonte: elaborado pelo autor

3.6. Outros aspectos do projeto da placa-base

3.6.1 Sinais *Ethernet* e USB

Nos sinais *Ethernet* e USB, por se tratarem de sinais diferenciais (a discretização de cada *bit* será feita através da diferença de tensões existente entre os dois pinos), as trilhas foram desenhadas paralelamente uma a outra, e possuem aproximadamente o mesmo comprimento. Isso faz com que quaisquer interferências de EMI, temperatura ou ambientes ruidosos sejam canceladas.

3.6.2 A carcaça dos conectores

Para normas de segurança e garantia de funcionalidade do circuito, todas as carcaças metálicas de conectores foram devidamente aterradas ao potencial negativo da placa. Isso faz com que qualquer contato do ser humano não danifique circuito nem o operador.

3.6.3 O plano de terra

Foi adicionado um plano de terra maciço na face de baixo da placa, a fim de facilitar o leiaute e também servir como um isolador de interferências ao ambiente externo. Por exemplo, pode-se ligar uma carga indutiva próxima as faces de baixo das placas e estas não sofrerão interferências entre si.

3.6.4 A espessura das trilhas de alta corrente

Nos circuitos de carga DC, foram especificadas trilhas que suportam alta corrente contínua. Dada a possibilidade de aumento da temperatura da placa, todas as trilhas que conectam a cargas de alta corrente suportam mais que 10A, mesmo que o circuito em si esteja projetado para o máximo de 10A, foi considerada uma faixa de segurança.

3.6.5 A fabricação da placa

O leiaute da placa base utilizou recursos de cobre nas duas faces da placa e furos metalizados. Seguem algumas especificações da placa:

a) Dimensões: 120 x 142,5 mm // 150 x 150 mm;
b) Nº de camadas: 2;
c) Material do laminado: FR-4 (fibra de vidro);
d) Componentes PTH (*Pin Throught Hole*) e SMD (*Surface Mounted Device*);
e) Furos metalizados;
f) Espessura do cobre: 1/2 oz.;
g) Máscara de solda.

De acordo com as características, a fabricação da placa exigiu um nível profissional de construção e acabamento, bem como uma grande confiabilidade em suas conexões elétricas (com o objetivo de controle de equipamentos em uma residência). Portanto, a placa foi fabricada em uma empresa especializada em fabricação de placas de circuito impresso

3.7 Testes placa-base e placa-reles

Os testes realizados na placa-base e na placa-reles consistiram em testes básicos de:

- Alimentação;
- Simulação de ativações de IO’s;
- Recepção de sinais através das entradas optoacopladas;
- Simulação de dados através dos *trasceivers* CAN, RS232 e RS485;
- Sinais de *reset* e botões de ativação dos *switch’s* de entrada.

3.8 Conclusão

Este capítulo teve por objetivo demonstrar a metodologia empregada no desenvolvimento do *hardware* proposto nesta dissertação, em que foram expostos as características do *hardware* criado.

Ao longo do desenvolvimento de todas as etapas de criação do projeto, foram criadas várias versões que demonstraram a melhoria qualitativa das novas funções e tecnologias empregadas a fim de criar um protótipo que pudesse atender da melhor forma possível os objetivos propostos.

É importante salientar que a área de automação residencial tem evoluído muito rápido e a cada dia tem aparecido novas tecnologias que até pouco tempo atrás eram impensáveis. Neste contexto, o projeto aqui proposto procurou atender algumas características fundamentais, como a possibilidade de adicionamento de vários sensores sem fio gerenciáveis, o que abre grandes possibilidades de novas melhorias servindo como base de estudos para implementações em trabalhos futuros.

CAPÍTULO 4: PROJETO DE *SOFTWARE* EMBARCADO PARA VALIDAÇÃO E TESTE DO SISTEMA

Este capítulo trata do desenvolvimento dos programas embarcados para validação e teste da plataforma desenvolvida. Os programas que serão embarcados no microcontrolador têm como finalidade o teste e a verificação completa do projeto, criando assim, as condições para que todos os sensores, atuadores e demais componentes troquem informações entre si, e executem os cenários desejados.

4.1 O mapeamento dos pinos a serem utilizados

A Tabela 3 apresenta todas as conexões realizadas entre a placa *LPCxpresso*/MBED e a placa-base, bem como a descrição e a funcionalidade dos pinos. Todos os pinos possuem a seleção por interruptores, localizados imediatamente à lateral de cada pino, sendo assim, na posição *ON,* o pino assumirá a função da placa (3ª coluna); na posição *OFF*, ele assumirá a função original da Placa LPCxpresso/MBED (vide última coluna):

Tabela 3 – configuração e funcionalidade dos pinos

Pino	**Sinal**	**Função**	**Descrição**	**Pino *LPCxpresso/MBED***
1	GND	Alimentação	Sinal negativo da alim.	GNDX
2	+5V	Alimentação	Sinal positivo da alim.	EXT_POWX
3	VBATM	Alimentação	Alimentação Bateria RTR	VB
4	RESET	Controle	Sinal de *reset* do microcontrolador	*RESET*_N
5	SQW	Controle	Onda quadrada oriunda do RTR da Placa-Base	P0[9]
6	CS	Comunicação	Sinal CS da linha SPI0	P0[8]
7	SCK	Comunicação	Sinal SCK da linha SPI0	P0[7]
8	SW1	Controle	*Switch* 1 de uso geral do usuário	P0[6]
9	SDA	Comunicação	Sinal SDA da linha I2C1	P0[0]

10	SCL	Comunicação	Sinal SCL da linha I2C1	P0[1]
11	MOSI	Comunicação	Sinal MOSI da linha SPI0	P0[18]
12	MISO	Comunicação	Sinal MISO da linha SPI0	P0[17]
13	TX/RS232/*XBEE*	Comunicação	Sinal de recepção (em relação ao *LPCxpresso*) da linha serial RS-232	P0[15]
14	RX/RS232/*XBEE*	Comunicação	Sinal de transmissão (em relação ao *LPCxpresso*) da linha serial RS-232	P0[16]
15	R/T	Comunicação	Sinal de controle de direção da linha serial RS-485	P0[23]
16	AD1/RELE3	Analógica	Canal 1 de entrada analógica/saída de controle do Rele 3	P0[24]
17	AD2/RELE4	Analógica	Canal 2 de entrada analógica/Saída de controle do Rele 4	P0[25]
18	AD3/RELE5	Analógica	Canal 3 de entrada analógica/Saída de controle do Rele 5	P0[26]
19	AD4/SW2	Analógica	Canal 4 de entrada analógica	P0[30]
20	AD5/SW3	Analógica	Canal 5 de entrada analógica	P0[31]
21	GIO7	Controle	Entrada optoacoplada	P0[2]
22	GIO8	Controle	Entrada optoacoplada	P0[3]
23			Aberto	P0[21]
24	TESTE1	Controle	Sinal de teste geral	P0[22]
25	TESTE2	Controle	Sinal de teste geral	P0[27]
26			Aberto	P0[28]
27			Aberto	P2[13]
28	+3V3	Alimentação	Sinal positivo da alim.	VIO_3V3X
29	-			
30	-			
31	-			
32	RD-	Comunicação	Sinal negativo de recepção do par trançado da linha *Ethernet*	RD-
33	RD+	Comunicação	Sinal positivo de recepção do par trançado da linha *Ethernet*	RD+
34	TD-	Comunicação	Sinal negativo de transmissão do par trançado da linha *Ethernet*	TD-
35	TD+	Comunicação	Sinal positivo de transmissão do par trançado da linha	TD+

			Ethernet	
36	USBD-	Comunicação	Sinal negativo de dados da linha USB	USB-D-
37	USBD+	Comunicação	Sinal positivo de dados da linha USB	USB-D+
38	CANRX	Comunicação	Sinal de recepção da linha CAN	P0[4]
39	CANTX	Comunicação	Sinal de transmissão da linha CAN	P0[5]
40	AC1/RELE1	Controle	Sinal de controle de acionamento do Triac 1 (Carga AC A) /Saída de controle do Rele 1	P0[10]
41	AC2/RELE2	Controle	Sinal de controle de acionamento do Triac 2 (Carga AC B) /Saída de controle do Rele 2	P0[11]
42	LED1/RELE6	Controle	Sinal de controle de saída discreta /Saída de controle do Rele 6	P2[0]
43	LED1/RELE7	Controle	Sinal de controle de saída discreta /Saída de controle do Rele 7	P2[1]
44	GIO5/RELE8	Controle	Entrada optoacoplada/Saída de controle do Rele 8	P2[2]
45	GIO6	Controle	Entrada optoacoplada	P2[3]
46	RELE_A	Controle	Saída de controle do Rele A	P2[4]
47	RELE_B	Controle	Saída de controle do Rele B	P2[5]
48			Aberto	P2[6]
49			Aberto	P2[7]
50			Aberto	P2[8]
51			Aberto	P2[10]
52			Aberto	P2[11]
53			Aberto	P2[12]
54	GND	Alimentação	Sinal negativo da alim.	GNDX

4.2 O COMPILADOR IDE (*INTEGRATED DEVELOPMENT ENVIRONMENT*)

Este trabalho envolveu dois (2) aspectos referentes ao *software*: (a) o ambiente de desenvolvimento integrado para configurar o *hardware*; e (b) o *software* desenvolvido para ser embarcado na placa. O ambiente de desenvolvimento é um compilador gcc (C e C++), que usa uma interface gráfica construída em *Java.* As funções da IDE são basicamente duas: (a) permitir o desenvolvimento de um *software*; e (b) enviar o mesmo para o microcontrolador da placa, a fim de ser executado.

A plataforma IDE utilizada (*Integrated Development Environment*) é uma plataforma *online* que é disponibilizada pela ARM para utilização com o microcontrolador que foi usado no projeto. Essa IDE pode ser acessada gratuitamente no *site* www.mbed.org, e não existe restrição de tamanho de código a ser desenvolvido.Na Figura 61, mostra-se a tela inicial do compilador descrevendo as informações básicas dos projetos que se tem no *Workspace.*

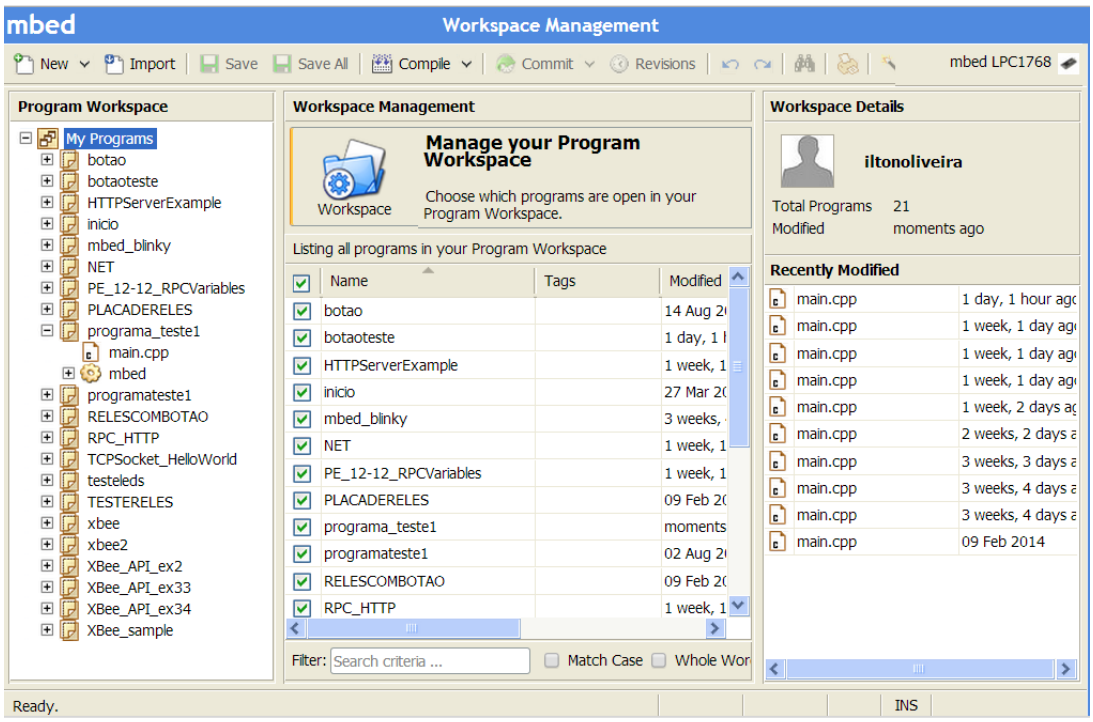


Figura 61 - Tela início da IDE

4.3 TESTE DE I/O

Para se criar um projeto inicial, a tela da Figura 62 solicita um nome para o projeto que vai ser gerado. No exemplo, serão acionados 02 reles da placa, através de 02 botões.

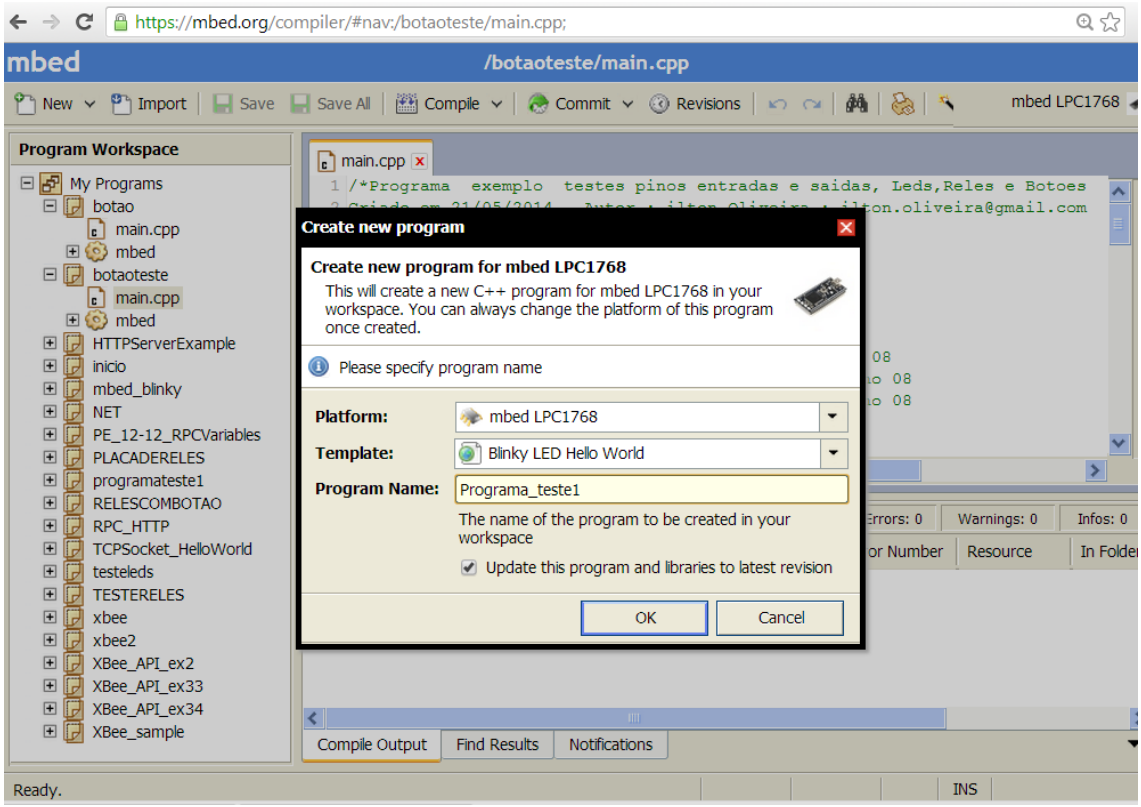


Figura 62 – IDE

Fonte: elaborado pelo autor

O código do programa-exemplo é mostrado na Listagem 1.

```
        /*Programa  exemplo  testes pinos entradas e saidas, Leds,Reles e Bot
oes
Criado em 21/05/2014 , Autor : ilton Oliveira : ilton.oliveira@gmail.com
*/
#include "MBED.h"
DigitalOut myled(LED1); // led da placa MBED
DigitalOut redled(p26); // led que esta no pino 26
DigitalOut rele1(p22); // rele1 ligado ao pino 22
DigitalOut rele2(p21); // rele 2 ligado ao pino 21
DigitalIn botao(p8); // SW1 - botao ligado ao pino 08
DigitalIn botao2(p20); // SW2 - botao ligado ao pino 20
DigitalIn botao3(p19); // SW3 - botao ligado ao pino 19
int main()
{
    while(1) {
        if (botao==1) { //verifica se o botao foi ativado,
            rele1 = 1; //rele1 é ligado
            redled = 1; // LIGA O LED
```

```
16              myled = 1; //liga o led da placa MBED
17              wait(0.2); // espera ,2 segundos
18              redled = 0; // desliga o led pino 26
19              wait(0.2); // espera 0.2 segundos
20          } //end of if
21          else { //caso o botao SW1 não esteja precionado
22              myled = 0; // desliga do led da placa MBED
23              redled = 0; //desliga o  Led ligado ao Pino 21
24              rele1 = 0; // desliga o rele1
25              wait(0.2);
26              rele1 = 0;
27              wait(0.2);
28
29              if (botao2==1) { //verifica se o botao foi ativado
30                  rele2 = 1; //rele1 é ligado
31                  redled = 1; // LIGA O LED
32                  myled = 1; //liga o led da placa MBED
33                  wait(0.2); // espera ,2 segundos
34                  redled = 0; // desliga o led pino 26
35                  wait(0.2); // espera 0.2 segundos
36              } //end of if
37              else { //caso o botao SW1 não esteja precionado
38                  myled = 0; // desliga do led da placa MBED
39                  redled = 0; //desliga o  Led ligado ao Pino 21
40                  rele2 = 0; // desliga o rele1
41                  wait(0.2);
42                  rele2 = 0;
43                  wait(0.2);
44              }
45              if (botao3==1) { //verifica se o botao foi ativado
46                  rele1 = 1; //rele1 é ligado
47                  redled = 1; // LIGA O LED
48                  myled = 1; //liga o led da placa MBED
49                  wait(0.2); // espera ,2 segundos
50                  redled = 0; // desliga o led pino 26
51                  wait(0.2); // espera 0.2 segundos
52              } //end of if
53              else { //caso o botao SW1 não esteja precionado
54                  myled = 0; // desliga do led da placa MBED
55                  redled = 0; //desliga o  Led ligado ao Pino 21
56                  rele1 = 0; // desliga o rele1
57                  wait(0.2);
58                  rele1 = 0;
59                  wait(0.2);
60              }
61          }
62      }
63  }
```

Listagem 1: Programa 01 teste de I/O

Para um melhor entendimento da listagem 1, os pinos 26, 22 e 21 do microcontrolador são ativados como saída digital nas linhas 6, 7 e 8 usando a função *DigitalOut*. Os pinos 8, 20 e 19 do microcontrolador foram ligados fisicamente na placa aos botões SW1, SW2 e SW3 e, conforme demonstrado nas linhas 8, 9 e 10, foram configurados como entrada digital usando a função interna DigitalIn. No restante do programa, entre as linhas 12 e 55, através de simples instruções

condicionais IF e Else, o programa verifica se foi apertado algum dos botões e liga o *led* correspondente.

Após a criação do código, deve-se compilar o programa, e a IDE gerará um arquivo binário, a ser embarcado no microcontrolador, conforme Figura 63.

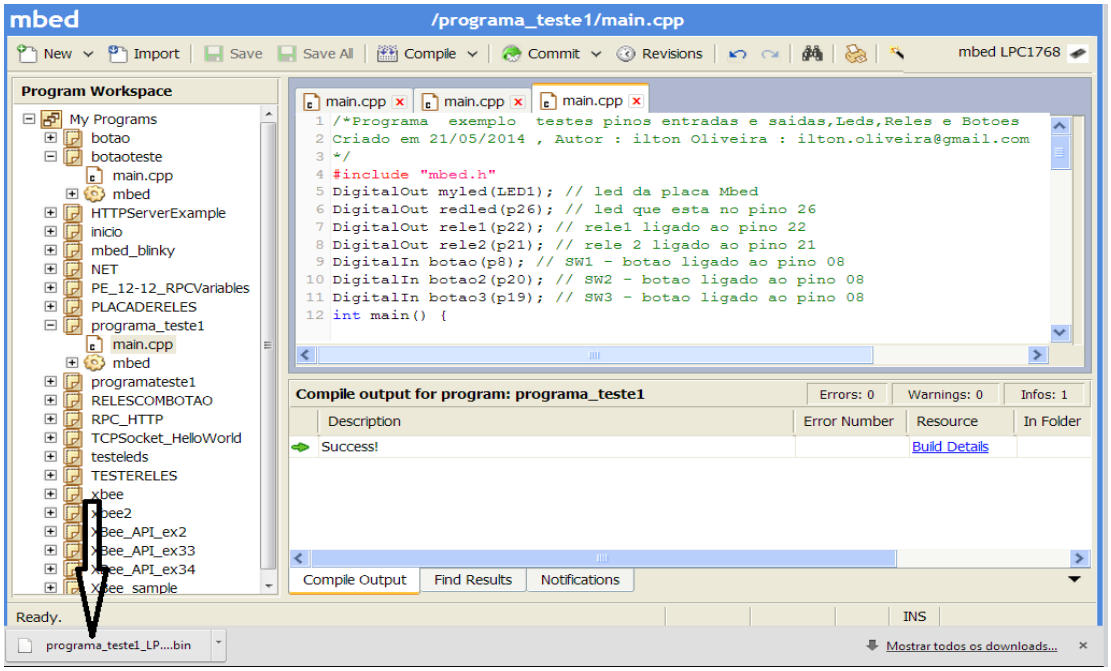


Figura 63 - IDE

Fonte: elaborado pelo autor

Após copiar o arquivo binário para o microcontrolador, a placa passa a funcionar conforme o *software* criado. Na Figura 64, pode-se ver que o *led* do rele 2 está ligado ao acionar o botão SW2 da placa.

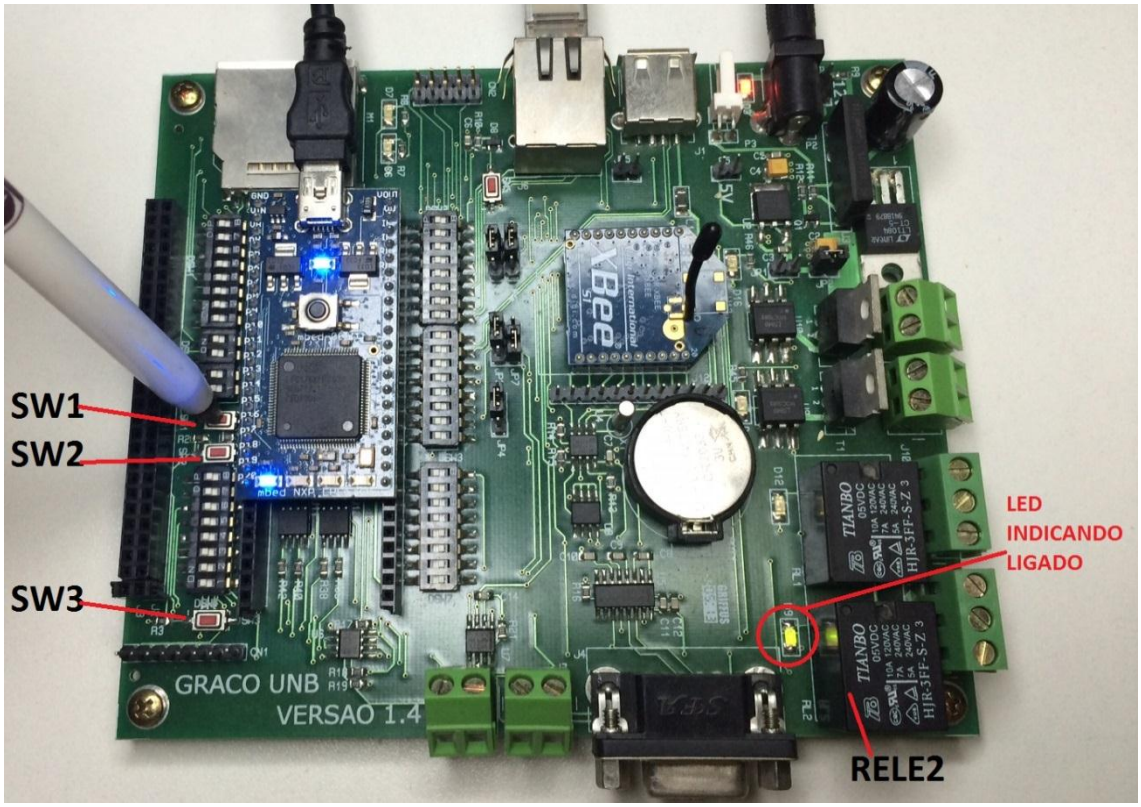


Figura 64 - Resultado listagem 1

Fonte: Elaborada pelo autor

4.4 Teste de acionamento de cargas com relés

No exemplo a seguir é demonstrada a simulação do controle de acionamento de cargas através dos reles da placa de reles externa, usando o cabo de comunicação; e depois com o uso de uma conexão remota, usando o *Xbee.*

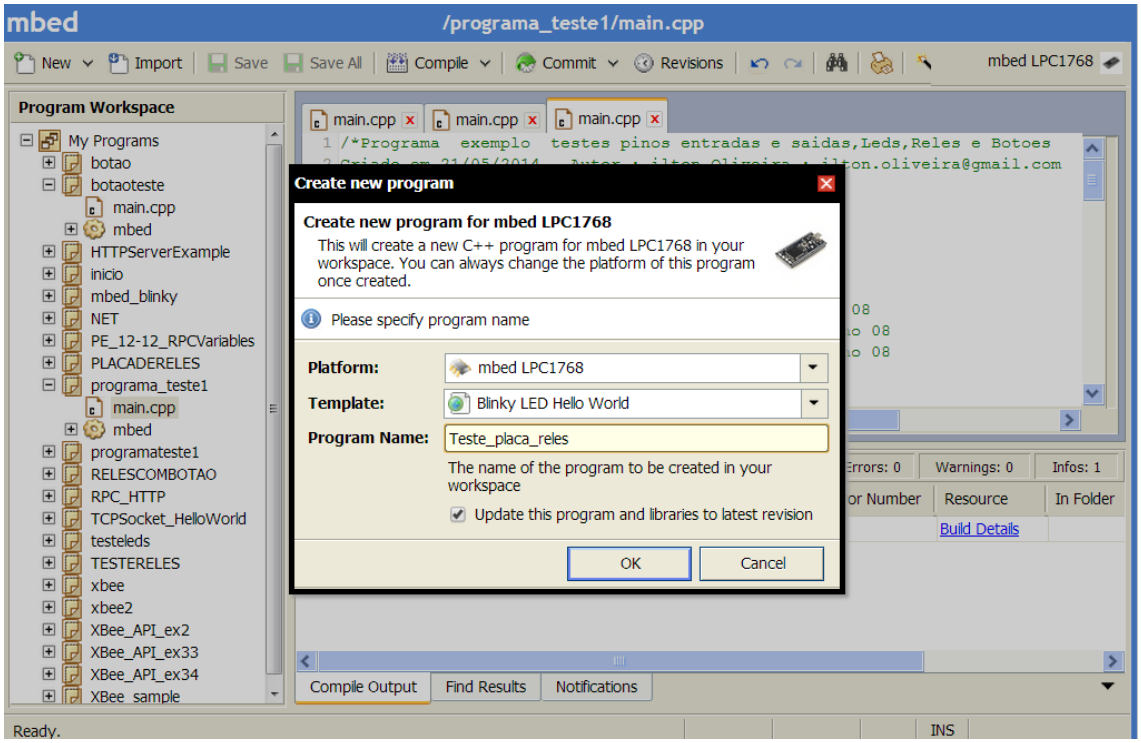


Figura 65 - IDE teste placa de reles

Fonte: Elaborada Pelo Autor

O Código do programa é mostrado na Listagem **2**.

```
        /* BRASILIA 09-02-2014 16H10
AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
CURSO : MESTRADO EM SISTEMAS MECATRONICOS
COMPILADOR : https://MBED.org/compiler
PROGRAMA DA PLACA  GRACO V. 1.4 - TESTE RELES 1 A 8 E BOTOES sw1,sw2 e sw3
OBS: O PROGRAMA FOI BAIXADO NO MEBED E TESTADO NA PLACA GRACO 1.4

#include "MBED.h"
DigitalOut myled(LED1); // led da placa MBED
DigitalOut rele1(p28); // rele1 ligado ao pino 28
DigitalOut rele2(p27); // rele2 ligado ao pino 27
DigitalOut rele3(p15); // rele3 ligado ao pino 15
DigitalOut rele4(p16); // rele4 ligado ao pino 16
DigitalOut rele5(p17); // rele5 ligado ao pino 17
DigitalOut rele6(p26); // rele6 ligado ao pino 26
DigitalOut rele7(p25); // rele7 ligado ao pino 25
DigitalOut rele8(p24); // rele8 ligado ao pino 24
DigitalIn botao(p8); // SW1 - botao ligado ao pino 08
DigitalIn botao2(p20); // SW2 - botao ligado ao pino 08
DigitalIn botao3(p19); // SW3 - botao ligado ao pino 08
int main() {
while(1) {
if (botao==1) { //verifica se o botao foi ativado
rele1 = 1; //rele1 é ligado
rele2 = 1; //rele1 é ligado
rele3 = 1; //rele1 é ligado
myled = 1; //liga o led da placa MBED
wait(0.2); // espera ,2 segundos
}
else { //caso o botao SW1 não esteja precionado
rele1 = 0; //rele1 é ligado
rele2 = 0; //rele1 é ligado
```

```
29 rele3 = 0; //rele1 é ligado
30 myled = 0; //liga o led da placa MBED
31 wait(0.2); // espera ,2 segundos
32
33 if (botao2==1) { //verifica se o botao foi ativado
34 rele4 = 1; //rele1 é ligado
35 rele5 = 1; //rele1 é ligado
36 rele6 = 1; //rele1 é ligado
37 myled = 1; //liga o led da placa MBED
38 wait(0.2); // espera ,2 segundos
39 }
40 else { //caso o botao SW1 não esteja precionado
41 rele4 = 0; //rele1 é ligado
42 rele5 = 0; //rele1 é ligado
43 rele6 = 0; //rele1 é ligado
44 myled = 0; //liga o led da placa MBED
45 wait(0.2); // espera ,2 segundos
46
47 if (botao3==1) { //verifica se o botao foi ativado
48 rele7 = 1; //rele1 é ligado
49 rele8 = 1; //rele1 é ligado
50 myled = 1; //liga o led da placa MBED
51 wait(0.2); // espera ,2 segundos
52 }
53 else { //caso o botao SW1 não esteja precionado
54 rele7 = 0; //rele1 é ligado
55 rele8 = 0; //rele1 é ligado
56 myled = 0; //liga o led da placa MBED
57 wait(0.2); // espera ,2 segundos
58      }
59      }
60 }
   }
   }
```

Listagem 2: Código programa teste placa Rele conexão com cabo

Após fazer o *download* do programa para o microcontrolador da placa, pode-se ver (na Figura 66) que foram acionados os reles 7 e 8 da placa de reles (que pode ser visualizado através dos *leds* dos reles ligados).

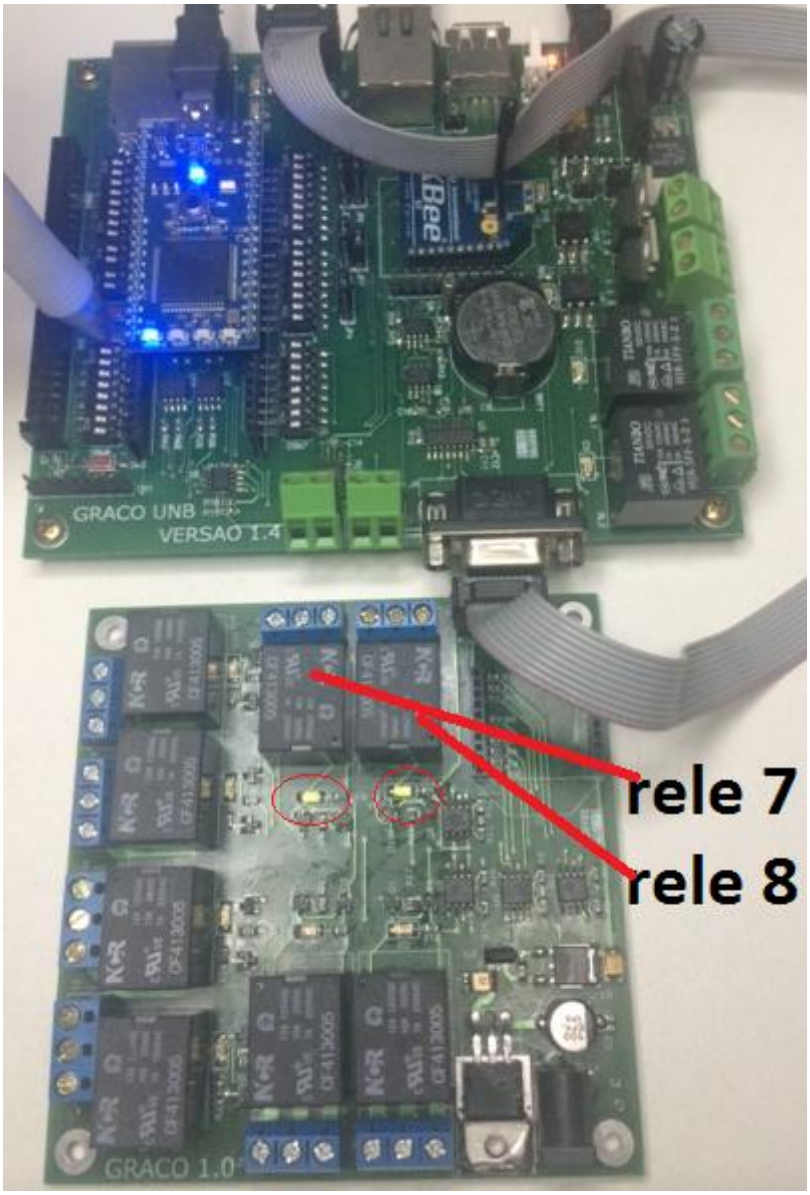


Figura 66 - Resultado Placas de Reles

Fonte: elaborada pelo autor

4.5 Teste da placa de relés com conexão XBEE

Na listagem 3, é exposto um exemplo que demonstra o funcionamento da plataforma criada e a placa de reles com acesso remoto, através de uma conexão utilizando os módulos *Xbees* instalados nas placas. A placa *ZigBee* permite comunicações robustas e opera na frequência ISM (*Industrial, Scientific and Medical*), sendo, na Europa, de 868 MHz (1 canal), 915 MHz (10 canais) nos Estados Unidos e 2,4 GHz (16 canais) em outras partes do mundo, e não requerem licença para funcionamento. As redes *ZigBee* oferecem uma excelente imunidade contra interferências, e a capacidade de hospedar milhares de dispositivos numa Rede (mais que 65.000), com taxas de transferências de dados variando entre 20Kbps e 250Kbps. XCTU : (Messias, 2011), vide figura 79.

Nos módulos *Xbee*/*Xbee*-Pro™ da figura 50, há três opções de antenas: (a) tipo chicote (um pedaço de fio de ~2,5 cm); (b) tipo conector (para antena externa); e (c) tipo *chip*, a mais compacta. Com os dois tipos, chicote e externa, é possível direcionar o feixe de sinal, e, assim, melhorar o desempenho da rede (Messias, 2011).

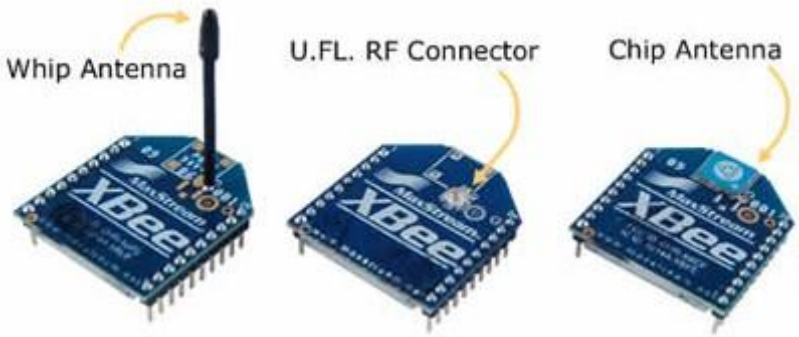


Figura 67 - Modulos *Xbee*

Fonte: Digi.com

O modulo que é utilizado na plataforma é um modulo *Xbee* ZB serie 1, e toda a sua programação pode ser feita através de um programa desenvolvido na IDE ou através de um programa disponibilizado gratuitamente pela MaxStream, que se chama *X-CTU* conforme Figura 68.

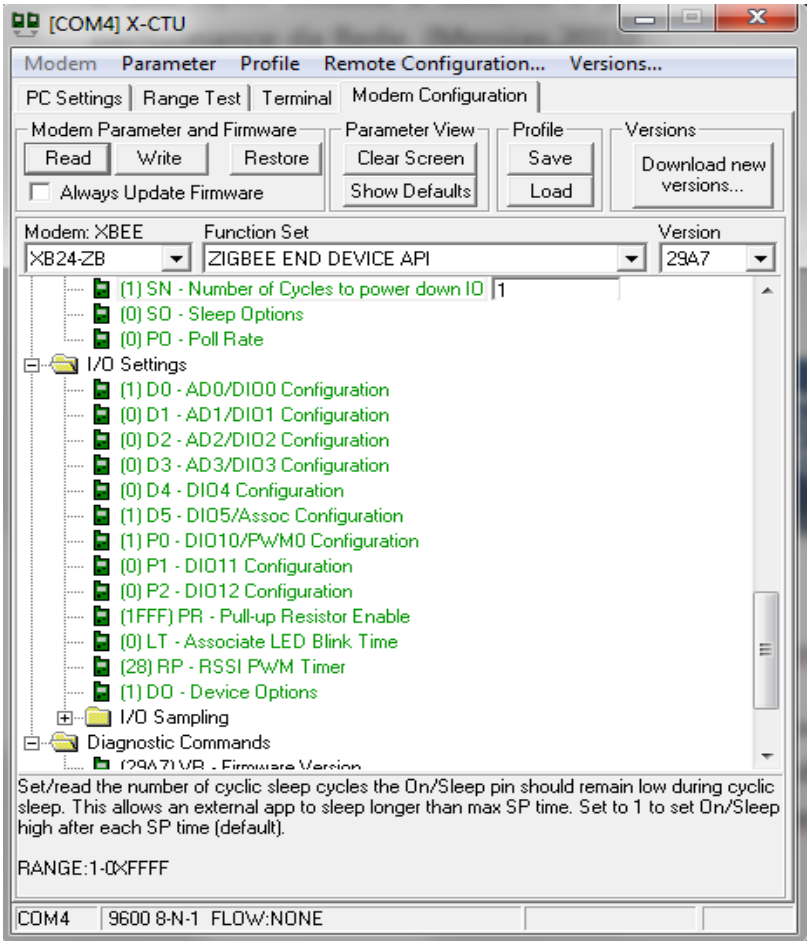


Figura 68 - XCTU

Fonte: elaborada pelo autor

Neste programa, podem-se configurar todos os parâmetros dos módulos *Xbee*, bem como a utilização dos pinos da (mostrados na Figura 69), podendo determinar, por exemplo, que o pino 18 será iniciado como sinal de entrada digital em estado lógico 05 HIGH, que informa ao módulo que esse pino inicia ligado.

Pino #	Nome	Direção	Descrição
1	VCC	-	Alimentação 3,3v
2	DOUT	Saída	Saída de dados da UART
3	DIN / CONFIG	Entrada	Entrada de dados da UART
4	DO8*	Saída	Saída digital 8
5	RESET	Entrada	Inicializa módulo (um pulso nível 0 de pelo menos 200ms)
6	PWM0 / RSSI	Saída	Saída do PWM 0 / Indicador de Força do sinal de RF (RX)
7	PWM1	Saída	Saída do PWM 1
8	(Reservado)	-	Ainda não tem uma função definida (futura implementação)
9	DTR / SLEEP_IRQ / DI8	Entrada	Linha de Controle da Função Sleep ou Entrada digital 8
10	GND	-	Terra
11	AD4 / DIO4	Entrada/Saída	Só Entrada Analógica 4 ou Entrada/Saída Digital 4
12	CTS / DIO7	Entrada/Saída	Controle de Fluxo CTS ou Entrada/Saída Digital 7
13	ON / SLEEP	Saída	Indicador de Estado do Módulo
14	VREF	Entrada	Voltagem de Referência para as Entradas A/D
15	Associação / AD5 / DIO5	Entrada/Saída	Indicador de Associação, só Entrada Analógica 5 ou Entrada/Saída Digital 5
16	RTS / AD6 / DIO6	Entrada/Saída	Controle de Fluxo RTS, só Entrada Analógica 6 ou Entrada/Saída Digital 6
17	AD3 / DIO3	Entrada/Saída	Só Entrada Analógica 3 ou Entrada/Saída Digital 3
18	AD2 / DIO2	Entrada/Saída	Só Entrada Analógica 2 ou Entrada/Saída Digital 2
19	AD1 / DIO1	Entrada/Saída	Só Entrada Analógica 1 ou Entrada/Saída Digital 1
20	AD0 /DIO0	Entrada/Saída	Só Entrada Analógica 0 ou Entrada/Saída Digital 0

Figura 69 - Pinagem *Xbee*

Fonte: ROGECOM

Na figura: linha 6 => força com inicial minúscula;

Linha 9: iniciais minúsculas e "*sleep*" em itálico.

Os módulos *Xbee* são configurados através de simples comandos AT, bem parecidos com aqueles usados para configurar *modems*.

Nos programas de teste da plataforma, vamos utilizar a programação utilizando o modo *API (Application Programming Interface).* No modo API, os dados transmitidos e recebidos estão contidos em *frames*, que definem operações ou eventos dentro do módulo. Através desse modo de operação, é possível um determinado módulo enviar endereço-fonte, endereço-destino, nome de um determinado nó, sinal RSSI, estado e outros parâmetros (MESSIAS, 2011).

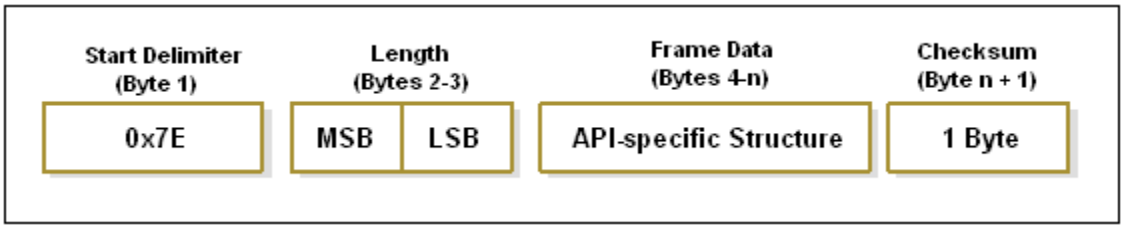


Figura 70 - Estrutura do Frame de Dados no modo API

Na listagem 3, linhas 30 a 43, é enviado via *serial*, pelos pinos 13 e 14, para o modulo *XBEE,* um *frame* com endereço único, linhas 36 a 42, e nos *bytes* das linhas 71 a 73 informa-se que pino será usado e que estado ele deve ter ao apertar o botão SW1.

```
        /* BRASILIA 19-03-2014 13H00
 9 AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
10 CURSO : MESTRADO EM SISTEMAS MECATRONICOS
11 COMPILADOR : https://MBED.org/compiler
12 PROGRAMA DA PLACA  GRACO V. 1.4 E PLACA DE RELE 1.0 - TESTE XBEE
13 */
14
15 #include "MBED.h"
16 Serial Xbee1(p13,p14); //TX E RX do MBED
17 DigitalOut myled(LED3);//Cria uma variavel para Led 3
18 DigitalIn botao(p8); // SW1 - botao ligado ao pino 08
19 DigitalIn botao2(p20); // SW2 - botao ligado ao pino 20
20
21 int main()
22 {
23     myled = 0;
24
25     while (1) {
26     if (botao==1) { //verifica se o botao foi ativado
27         myled = 1; // Led 3 Off
28         // Inicio do FRAME XBEE 1 PRIMEIRA PLACA DE RELAY
29         Xbee1.putc(0x7E);  //inicia byte API frame
30         Xbee1.putc(0x00);
31         Xbee1.putc(0x10);
32         Xbee1.putc(0x17);
33         Xbee1.putc(0x05);
34         Xbee1.putc(0x00);
35         Xbee1.putc(0x13); //XBEE Endereço
36         Xbee1.putc(0xA2); //XBEE Endereço
37         Xbee1.putc(0x00); //XBEE Endereço
38         Xbee1.putc(0x40); //XBEE Endereço
39         Xbee1.putc(0xA1); //XBEE Endereço
40         Xbee1.putc(0x3A); //XBEE Endereço
41         Xbee1.putc(0xB6); //XBEE Endereço
42         Xbee1.putc(0xFF);
43         Xbee1.putc(0xFE);
44         Xbee1.putc(0x02);
45         Xbee1.putc(0x44);     // commando ASCII caracter "D" PINO NO XBEE
46         Xbee1.putc(0x34);     // commando ASCII caracter "4" PINO NO XBEE
47         Xbee1.putc(0x05);     // D 4 HIGH ATIVA O PINO D4 EM HIGH LIGA RELAY
48         Xbee1.putc(0xE1);     //CHECKSUM
49         wait(1);
50         myled = 0; //Desliga Led 3
51         wait(1);
52         }
53         else {
54         Xbee1.putc(0x7E);         //inicia byte API frame
55         Xbee1.putc(0x00);
56         Xbee1.putc(0x10);
57         Xbee1.putc(0x17);
58         Xbee1.putc(0x05);
59         Xbee1.putc(0x00);
60         Xbee1.putc(0x13); //XBEE Endereço
61         Xbee1.putc(0xA2); //XBEE Endereço
62         Xbee1.putc(0x00); //XBEE Endereço
63         Xbee1.putc(0x40); //XBEE Endereço
64         Xbee1.putc(0xA1); //XBEE Endereço
65         Xbee1.putc(0x3A); //XBEE Endereço
66         Xbee1.putc(0xB6); //XBEE Endereço
67         Xbee1.putc(0xFF);
68         Xbee1.putc(0xFE);
```

```
69          Xbee1.putc(0x02);
70          Xbee1.putc(0x44);    // commando ASCII caracter "D" PINO NO XBEE
71          Xbee1.putc(0x34);    // commando ASCII caracter "4" PINO NO XBEE
72          Xbee1.putc(0x04);    // D 4 LOW DESATIVA O PINO D4 EM LOW DESLIGA RELAY
73          Xbee1.putc(0xE2);//CHECKSUM
74          wait(1);
75           }
76          if (botao2==1) { //verifica se o botao foi ativado
77                // Inicio do FRAME XBEE 2 SEGUNDA PLACA DE RELAY
78          Xbee1.putc(0x7E);
79          Xbee1.putc(0x00);
80          Xbee1.putc(0x10);
81          Xbee1.putc(0x17);
82          Xbee1.putc(0x05);
83          Xbee1.putc(0x00);
84          Xbee1.putc(0x13);
85          Xbee1.putc(0xA2);
86          Xbee1.putc(0x00);
87          Xbee1.putc(0x40);
88          Xbee1.putc(0xA0);
89          Xbee1.putc(0x40);
90          Xbee1.putc(0x37);
91          Xbee1.putc(0xFF);
92          Xbee1.putc(0xFE);
93          Xbee1.putc(0x02);
94          Xbee1.putc(0x44);
95          Xbee1.putc(0x34);
96          Xbee1.putc(0x05);
97          Xbee1.putc(0x5B);
98          wait(1);
99          }
100         else {
101         Xbee1.putc(0x7E);
102         Xbee1.putc(0x00);
103         Xbee1.putc(0x10);
104         Xbee1.putc(0x17);
105         Xbee1.putc(0x05);
106         Xbee1.putc(0x00);
107         Xbee1.putc(0x13);
108         Xbee1.putc(0xA2);
109         Xbee1.putc(0x00);
110         Xbee1.putc(0x40);
111         Xbee1.putc(0xA0);
112         Xbee1.putc(0x40);
113         Xbee1.putc(0x37);
114         Xbee1.putc(0xFF);
115         Xbee1.putc(0xFE);
116         Xbee1.putc(0x02);
117         Xbee1.putc(0x44);
118         Xbee1.putc(0x34);
119         Xbee1.putc(0x04);
            Xbee1.putc(0x5C);

   wait(1);
            }
       }
       }
```

Listagem 3: Código programa teste placa Rele *XBEE*

Conclusão

Este capítulo teve por objetivo apresentar o funcionamento do projeto criado, demonstrando as várias possibilidades de aplicações que podem ser desenvolvidas. Foram apresentadas as ferramentas necessárias para a criação de aplicações a serem embarcadas no microcontrolador. O uso de uma *IDE online* para a criação das aplicações permite, por exemplo, que projetos mais elaborados possam ser criados por uma equipe multidisciplinar, em que os participantes possam acessar cada módulo que está sendo desenvolvido de qualquer lugar que possua uma conexão com a internet.

A quantidade de bibliotecas e exemplos de aplicativos existentes para funcionalidades, como Ethernet, USB, LCD, I2C, entre outros, são facilitadores que permitem um rápido aprendizado para criação de novas possibilidades de uso desta plataforma.

Como demonstrado nos exemplos deste capítulo pode-se evidenciar que a plataforma criada é totalmente funcional e possui várias possibilidades de configurações, o que permite uma vasta gama de aplicações dentro da área de automação residencial.

5 CONCLUSÕES

5.1 Considerações gerais

O objetivo inicial deste trabalho era projetar e desenvolver uma plataforma que tivesse a funcionalidade de um CLP (Controlador Lógico Programável) que servisse para os propósitos de testes acadêmicos no laboratório do LEIA-GRACO-Unb. Com o desenvolvimento da primeira versão, observou-se que suas funcionalidades poderiam ir muito além da implementação de um CLP (Controlador Lógico Programável), podendo se criar uma plataforma de desenvolvimento de aplicações variadas no universo de automação.

A criação de uma plataforma para servir de *KIT* de desenvolvimento ARM com a simulação funcional de um CLP (Controlador Lógico Programável) se mostrou mais adequada para este projeto de dissertação, pois, dessa forma, ter-se-ia a possibilidade de disponibilizar uma ferramenta *open-source* para a criação de aplicações tais como CLPs, *scanners* automotivos, projetos com a utilização de sensores remotos para captação de dados, robótica, entre outros.

Para que a plataforma pudesse ter várias funcionalidades, era necessária a escolha de um microcontrolador que, tecnica e financeiramente, adequasse-se à escolha do microcontrolador ARM de 32 *bits*, que se mostrou o mais viável dentre as opções que existiam no mercado.

O microcontrolador ARM Cortex M3 LPC1768 foi um dos vários da família ARM a ser estudado e foi escolhido pela sua gama de recursos embutidos, tais como 100 MHz de frequência, comunicação ETHERNET, CAN, USB, I2C, SPI, UART e saídas PWM. Um dos fatores mais importante para a escolha desta família de microcontrolador ARM foi o IDE de desenvolvimento de aplicações, disponibilizado gratuitamente *online* e possui todo um ecossistema em www.mbed.org, que facilita a implementação de várias aplicações, disponibilizando fóruns de discursão, códigos, exemplos, bibliotecas diversas que permitem uma melhor interação dos usuários com a plataforma.

Foi pensado desde o início que a plataforma não deveria ser do tipo proprietário, sendo que dependeria de um manual extenso para seu entendimento. Desta maneira, a escolha desse ARM LPC1768 se encaixou perfeitamente aos objetivos propostos, pois, além de possuir embarcados vários periféricos importantes, para as aplicações atuais, o mesmo contempla vasta documentação e interações na internet, permitindo assim um rápido aprendizado da plataforma.

A confecção profissional da placa de circuito impresso foi uma necessidade devido à complexidade da plataforma (seria pouco produtiva a confecção caseira da placa, além de comprometer a qualidade do produto final).

Após a segunda versão, foi escolhida a utilização de componentes SMD, o que deu um aspecto mais profissional à plataforma e permitiu a compactação da placa, mas estes componentes necessitam de um conhecimento mais apurado dos desenvolvedores que quiserem reproduzir o *hardware*, pois a soldagem dos componentes não é tão trivial como nos componentes tradicionais.

Como observado, a plataforma possui muitos recursos de expansibilidade, o que era uma das prioridades do projeto. A possibilidade de desenvolvimento de outras aplicação facilmente é permitido através das portas de expansão que permitem acesso a todos os pinos do microcontrolador.

Com o objetivo original de simular uma aplicação de um CLP, a plataforma possui, em seu corpo, periféricos que foram disponibilizados exclusivamente para essa funcionalidade, tais como reles, *triacs*, porta ethernet e *XBee*.

A criação da placa de reles com *XBee* foi projetada especificamente para atender a aplicação do CLP, permitindo, assim, que qualquer carga em uma residência possa ser controlada sem a necessidade de se criar um cabeamento específico para isso .

A placa de reles *XBee* pode facilmente ser modificada para criação de qualquer tipo de sensor remoto, bastando uma reprogramação de *software* e uma criação de circuitos simples, como um sensor de temperatura por exemplo.

Após vários testes, o objetivo do projeto foi concluído com sucesso, pois o mesmo atingiu os resultados esperados, demonstrando ser uma poderosa ferramenta acadêmica para o desenvolvimento de várias aplicações de sistema

embarcados, além de possuir funcionalidade exclusivas não encontradas em vários *kits* de desenvolvimento existentes no mercado como, por exemplo, o uso da comunicação *XBee* sem fio, a qual permitiu um ganho de escala na expansão de outras aplicações, trabalhos futuros, servindo de base para a criação de outras plataformas similares para fins acadêmicos .

5.2 Sugestões para trabalhões futuros

Devido às grandes possibilidades de desenvolvimento de aplicações utilizando a plataforma criada, trabalhos futuros podem ter como referência este projeto, que pode guiar novas ideias para criação de outras plataformas que tenham como objetivo automação residencial, industrial, robótica, *scanners* para veículos, utilizando rede CAN, entre outros.

No decorrer do desenvolvimento de cada versão do trabalho, vàrias possibilidades de novas modificações na plataforma foram surgindo e, após a versão 1.4, foi resolvido que essas novas modificações deveriam ser expostas nesse capítulo como sugestões para trabalhos futuros. Visando ajudar no direcionamento de novas melhorias para o projeto, apresentaremos algumas sugestões que poderão servir de parâmetros para futuras modificações.

Um ponto importante a ser salientado é que toda parte de interface de programação fica como um trabalho futuro a ser desenvolvido pelo grupo de pesquisa do LEIA-GRACO.

As melhorias no circuito principal da plataforma podem ser as seguintes:

a) Após alguns testes, verificou-se que um ponto crítico na plataforma era a fonte, e seria uma boa implementação uma fonte removível que pudesse ser trocada facilmente em caso de um defeito;
b) Os botões e os *leds* de trabalho de uso geral poderiam ficar melhor identificados e disponibilizados juntos para facilitar em testes como *kit* de desenvolvimento;

c) A divisão dos componentes em ilhas identificadas na placa facilitaria o uso da placa;
d) Inserir uma chave liga/desliga para iniciar o uso da placa;
e) Apesar da disponibilização de todos os pinos do microcontrolador na expansão da placa ser possível, uma melhoria como a criação de um encaixe universal para empilhamento de módulos tais como é feito nos Shields dos *Arduinos* ;
f) A impressão das funcionalidades dos *switches* que ligam e desligam o acesso dos pinos do microcontrolador aos periféricos da placa;

As melhorias no circuito principal da placa de reles seriam as seguintes:

a) Fonte removível que pudesse ser trocada facilmente em caso de um defeito;
b) Circuito na fonte para uma bateria reserva como usado na plataforma para o caso de falta de energia;
c) No circuito do módulo *Xbee*, poderia existir uma expansão empalhável para cada acesso aos pinos do modulo *Xbee*, o que permitiria a implementação de outros sensores junto à placa.

REFERÊNCIAS BIBLIOGRÁFICAS

Miori, Vittorio ; Russo, Dario. **Advanced Information Networking and Applications Workshops (WAINA)**, 2014 28th International Conference , Disponível em: http://ieeexplore.ieee.org.ez54.periodicos.capes.gov.br/xpls/icp.jsp?arnumber=6844739&tag=1, Acesso em: 2014.

Miori, Vittorio; Russo,Dario; Aliberti, Massimo .**Domotic technologies incompatibility becomes user transparent**, Commun. ACM 53, 1 January 2010, Disponível em: http://cacm.acm.org.ez54.periodicos.capes.gov.br/magazines/2010/1/55764-domotic-technologies-incompatibility-becomes-user-transparent/fulltext Acesso em 2014.

GALASSI, T. T.; MARTINS, L. E. G. **Um estudo exploratório sobre sistemas operacionais embarcados**. Relat. Inic. Cient., Americana, v.1, n.1, p.78-105, set.2013 / mar.2014.

SEONG, Peng; Gik Hong . **Centralised Smart Home Control System via XBee transceivers**, Publicado em : Humanities, Science and Engineering (CHUSER), 2011 IEEE Colloquium on , Publicado em: 5-6 Dezembro de. 2011

Al,Sharbaty . **Wireless sensors network and acknowledgement technique based ZigBee system** . Publicado em : Computer Science and Information Technology (CSIT), 2014 6th International Conference on, publicado em 26-27 Março de 2014

FONSECA, Marcos. **Apresentação sobre a IEC 61131**. Disponível em: http://www.iec61131.com.br Acesso em: 1 de maio de 2010.

SENAI, 2002. **Eletricidade Instalações Industriais.** Sapucaia do Sul, RS. Julho de 2002

BOLZANI, **C.A.M. Desenvolvimento de um simulador de controle de dispositivos residenciais inteligentes : uma introdução aos sistemas domóticos**. Dissertação (mestrado), Universidade de São Paulo, São Paulo 2004.

ALIEVE , César . **Automação Residencial com Utilização de Controlador Lógico Programável. Dissertação** , Centro Universitário Feevale, 2008

FARAHANI, Shahin. **ZigBee Wireless Networks and Transceivers,** Ed. Elsevier , 2008.

HWANG, Soyoung. **Remote Monitoring and Controlling System Based on ZigBee Networks,** Artigo publicado pelo International Journal of Software Engineering and Its Applications Vol. 6, No. 3, July, 2012

D. Yan and Z. Dan, **ZigBee-based Smart Home System Design**, Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, 2010.

IEEE 802.15.4, **Wireless medium access control (MAC) and physical layer (PHY) specification for low-rate wireless personal area networks** (WPANs), 2006.

SOMANI Nisha; PATEL Yask. **Zigbee: A Low Power Wireless Technology For Industrial Applications**, artigo publicado no International Journal of Control Theory and Computer Modelling (IJCTCM) Vol.2, No.3, Maio de 2012.

Instrument, Texas. **Z-Stack-ZigBee protocol stack.** Acessado em: http://www. ti. com/tool/z-stack 2013.

CANZIAN, **E. Comunicação Serial - RS232**, [S.d.]. Disponível em: http://www.professores.aedb.br/arlei/AEDB/Arquivos/rs232.pdf

GEORGINI, M. Automação **Aplicada Descrição e Implementação de Sistemas Seqüenciais com CLPs**. 6. ed. São Paulo: Editora Erica 2006.

OLIVEIRA, Júlio César P. **Controlador Programável**, São Paulo: Makron Books, 1993.

Documentação Técnica da norma IEC 61131. Dados obtidos em 06 de abril de 2008 por meio da url:http://www.iec61131.com.br

José Maurício Santos Pinheiro -, **As Redes Com ZigBee**, no. Redes sem fios, Julho 2004.

YIU, Joseph. **The definitive guide to the ARM Cortex-M3** , Editora Elsevier

Brown, Marty. **Power supply cookbook** , Editora Butterworth–Heinemann 2001

SURELL, Kirk. **C Programming for Embedded Systems** , Editora R&D Books 2008

Kuphaldt ,Tony R. **Lessons In Electric Circuits**, Volume I 2003

Zurawsk, Richard . **Embedded systems handbook** Editora CRC Press 2009

Eady, Fred . **Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers** Editora Elsevier 2007

Gislason, Drew. **Zigbee Wireless Networking** , Editora Newnes 1999

Faludi Robert**. Building Wireless Sensor Networks** , Editora O´REILLY

Ribeiro , Antônio Marco . **Fundamentos Da Automação** , 2003

Paiva, Leonardo Silveira . **Metodologia para Implantação de Automação Residencial**, Monografia UFOP setembro 2007

ROGERCOM. - **Controle remoto e aquisição de dados via *Xbee*/ZigBee (IEEE 802.15.4)** Junho de 2013, Disponivel em : http://www.rogercom.com/

ARDUINO. ***Arduino*** **Duemilanove.Disponível** em: <http://www.*ARDUINO*.cc>. Acesso em: 03 out. 2012.

BIGNELL, J.; DONOVAN, R. Eletrônica Digital - **Lógica Sequencial**. 3. ed. São Paulo: Makron Books, 1995.

CASTILHO, M. P. **Projeto de Protocolos Considerando as Características não Lineares das Baterias em Redes de Sensores Sem Fio**. Dissertação (Mestrado) — Pontifícia Universidade Católica de Minas Gerais Minas, 2009.

YIU Joseph, **The definitive guide to the ARM Cortex-M3** . Ed Elsevier, 2010.

SANTOS, S. T., **Redes de Sensores Sem Fio em Monitoramento e Controle**, Dissertação de Mestrado, COPPE/UFRJ, Engenharia Elétrica, 2007.

MONIZ, L.M.F.F. SSAA: **Sistema Para Simulação de Agentes e Ambientes.** Dissertação (mestrado). Universidade Técnica de Lisboa. Portugal. 1993.

REZENDE, S.O. **Sistemas Inteligentes. Fundamentos e Aplicações**. Editora Manole, São Paulo. 2002.

PRESSMAN, Roger S, **Engenharia de Software**. Editora Bookman, 2011.

LUGLI, Alexandre; SOBRINHO, Darlan 2013 . Tecnologias Wireless Para Automação Industrial. INATEL 2013 Disponível em : http://www.inatel.br/pos/artigos-publicados-ai/cursos/automacao-industrial/artigos-publicados, Acesso em Dezembro de 2014

DIAS, César Luiz de Azevedo. **Domótica: Aplicabilidade Às Edificações Residenciais**. Dissertação de mestrado, Universidade Federal Fluminense. Niterói, 2004.

RUIZ, L. B. (2003). MANNA: **Uma Arquitetura para o Gerenciamento de Redes de Sensores Sem Fio**. Tese de doutorado, Universidade Federal de Minas Gerais.

VERDONE, R.; Dardari, D.; Mazzini, G.; Conti, A.; **Wireless Sensor and Actuator Networks: Technologies**, Analises and Design. Burlington: Elsevier, 2007.

CASADOMO. **Domótica - Introducción**. Disponível em: https://www.casadomo.com/biblioteca . Acesso em 12 de Maio de 2010.

DIGI International, **X-CTU Configuration & Test Utility *Software***, Abril de 2013 . Disponível em : http://ftp1.digi.com/support/documentation/90001003_A.pdf

DIGI International, **Product Manual v1.xEx - 802.15.4 Protocol**, Abril de 2013 . Disponível em : http://ftp1.digi.com/support/documentation/90000982_P.pdf

DIGI International, ***Xbee* Drop-in-Networking Accessories User's Guide**, Abril de 2013 . Disponível em : http://ftp1.digi.com/support/documentation/90000891_H.pdf

DIGI International, ***Xbee*® 802.15.4 Starter Kit Getting Started Guide**, Abril de 2013 . Disponível em : http://ftp1.digi.com/support/documentation/90002160_A.pdf

DIGI International, ***Xbee*®/*Xbee*-PRO® RF Modules**, Abril de 2013 . Disponível em : http://ftp1.digi.com/support/documentation/90000982_P.pdf

DIGI International, ***Xbee* Development Examples and Guides**. Disponível em : http://examples.digi.com/

MBED.org, **Exemplos gerais de bibliotecas e exemplos de aplicações**. Disponível em : https://MBED.org/cookbook/Homepage

MBED.org, **Manual de uso geral** . Disponível em : https://MBED.org/handbook/Homepage

MBED.org, Repositório geral de códigos para varias aplicações . Disponível em : https://MBED.org/code/

APÊNDICE A – MANUAL DE APRESENTAÇÃO DA PLATAFORMA GRACO 1.4

A1 - Visão Geral

A1.1– Microcontroladores

A *plataforma Graco 1.4* possui varias opções de utilização de microcontroladores ARMs, possuindo barramento para uso do LPC EXPRESSO e do MBED (Vide figuras 77 e 78) que podem vir com diferentes microcontroladores, à plataforma que esta sendo descrita neste breve manual utiliza um MBED com microcontrolador ARM 32 bits LPC1768FBD100 mostrado na Figura 73.

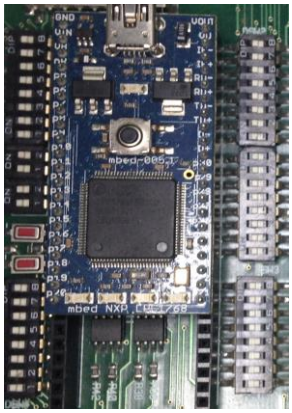


Figura 71 - Placa MBED com Microcontrolador LPC1768

Figura 72 - Microcontrolador LPC1768

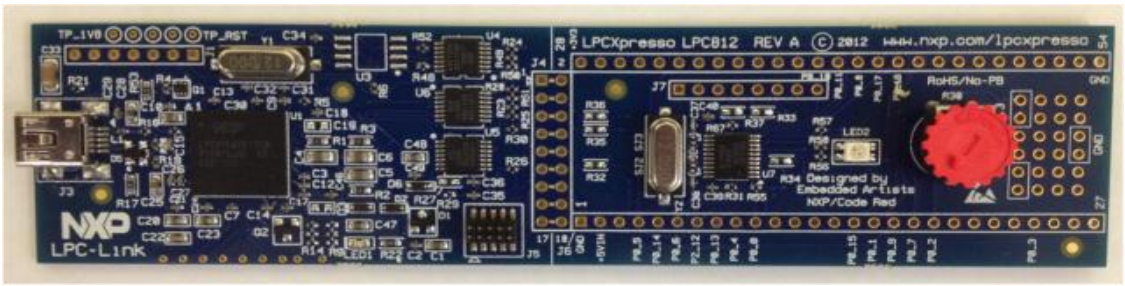


Figura 73 - LPCXPRESSO

Com o uso do LPCXPRESSO também é possível utilizar o seguintes microcontroladores ARMS de 32 bits:

a) LPCxpresso812 ;
b) LPCxpresso1104;
c) LPCxpresso1114;
d) LPCxpresso1115;
e) LPCxpresso11C24;
f) LPCxpresso1227;
g) LPCxpresso11U14;
h) LPCxpresso11U24;
i) LPCxpresso1343;
j) LPCxpresso1347;
k) LPCxpresso1768;
l) LPCxpresso1769;

Cada um dos microcontroladores acima tem especificações diferentes e podem ser utilizadas em varias aplicações , na *plataforma GRACO 1.4* usa-se o LPC 1768 que é um dois mais completos da família *ARM CORTEX M3* e possui vários periféricos já embutidos conforme Figura 74.

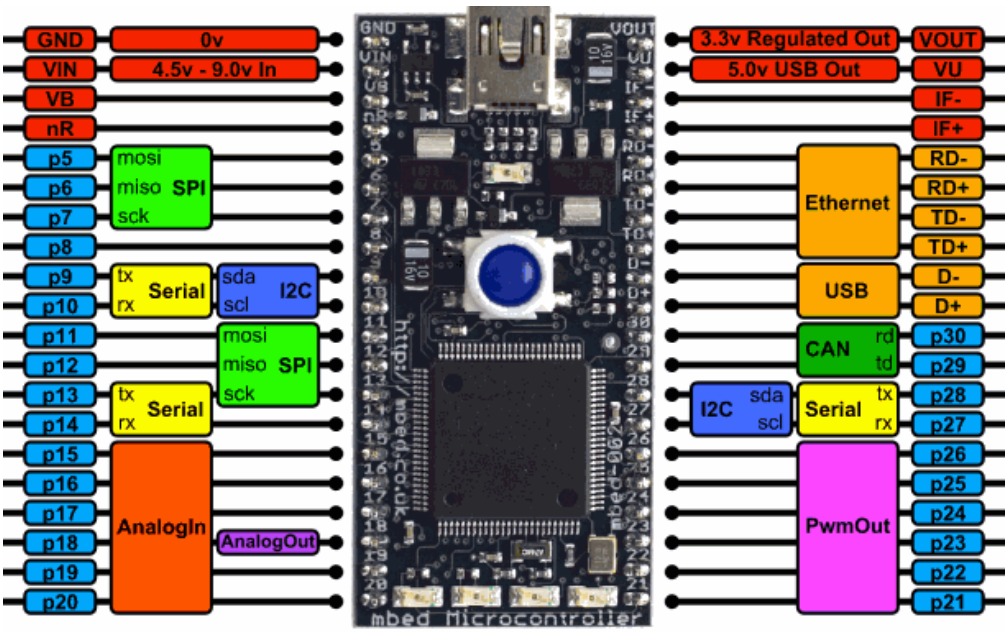


Figura 74 - LPC 1768 Periféricos

A1.2- Fonte

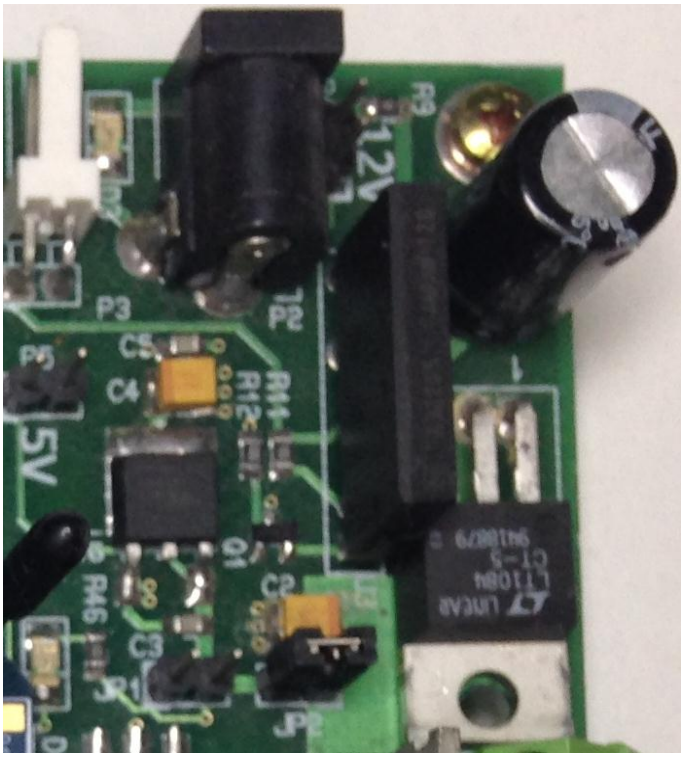


Figura 75 - Circuito da Fonte da plataforma

A fonte da plataforma *GRACO 1.4* (mostrado na Figura 75), foi criada para alimentar os circuitos da placa que possuem voltagens diferentes para o microcontrolador LPC1768 que é alimentado com +5v e o modulo de comunicação sem fio *Xbee* que é alimentado com +3v, ela possui um recurso de bateria reserva que aceita uma bateria de 12v externa e caso a energia elétrica falhe o sistema automaticamente passa a operar com a reserva , impedindo que todo o sistema fique inoperante, sua configuração pode ser vista na tabela 8.

É importante lembrar que a bateria reserva alimenta somente a placa principal, logo é necessário algum recurso para alimentação do restante do sistema como os sensores e atuadores.

Tabela 4: Elaborado pelo Autor- Configurações das entradas do circuito da fonte da plataforma

Entradas	Descrição
P2	Entrada de 12 V que alimenta a placa, deve ser utilizada uma fonte com no mínimo 1 ampere para permitir a utilização dos periféricos da placa como os relês
P3	Entrada auxiliar de 12v que é acionado automaticamente caso a energia elétrica fique inoperante, pode-se utilizar uma bateria selada de 7 amperes . O sistema não recarrega a bateria, sendo necessária a recarga ou substituição em algum momento
P4	É uma saída de 12v para uso geral
P5	É uma saída de 5V para uso geral
JUMPER 1	
JUMPER 2	

Tabela 4: Elaborado pelo Autor

A1.3 – Relês

Na placa existem 02 Reles independes NA/NF que permitem acionar motores, maquinas, ou outros tipos de cargas

Tabela 5: Elaborado pelo Autor– Descrição Pinos Reles

PINO MBED	Descrição
22	RELE1
21	RELE2

Tabela 5: Elaborado pelo Autor

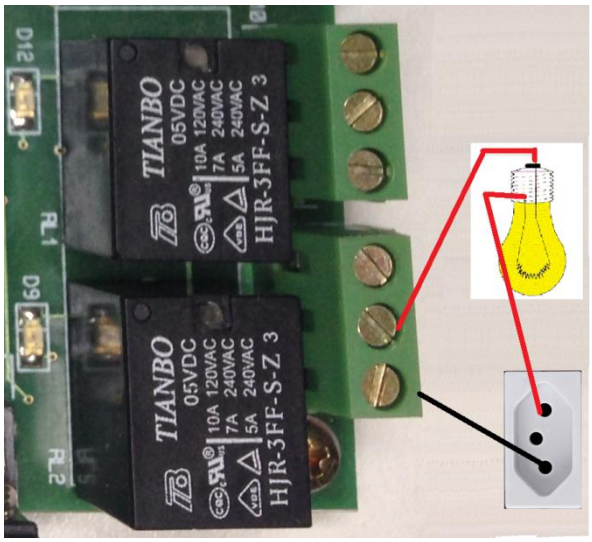


Figura 76 - Exemplo ligação lâmpada com Rele

Na Tabela 5 abaixo tem-se um exemplo de ligação dos Reles usando dois botões da placa, observe que deve-se primeiramente definir as funções dos pinos como saida digital nas linhas 14,15 e entrada digital para os botões SW2, SW3 pinos16 e 17, após isso é só compilar o programa no IDE e fazer o download para a placa.

```
1 /* BRASILIA 09-02-2014 14H06
2 AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
3 CURSO : MESTRADO EM SISTEMAS MECATRONICOS
4 COMPILADOR : https://MBED.org/compiler
5 PROGRAMA DA PLACA  GRACO V. 1.4 -
6  TESTE RELES NA/NF USANDO BOTOES SW2 E SW3
7 OS PINOS USADOS SÃO PINOS 21 E 22 DOS LEDS 1 E 2 , PINOS 19 E 20 DOS BOTOES
8 SW2 E SW3
9 OBS: O PROGRAMA FOI BAIXADO NO MEBED E TESTADO NA PLACA GRACO 1.4 */
10
```

```
11
12 #include "MBED.h"
13
14 DigitalOut rele1(p22); // COLOCA O PINO 21 COMO SAIDA DIGITAL
15 DigitalOut rele2(p21);// COLOCA O PINO 21 COMO SAIDA DIGITAL
16 DigitalIn botao1(p19);// COLOCA O PINO 19 COMO ENTRADA DIGITAL BOTÃO SW1
17 DigitalIn botao2(p20); // COLOCA O PINO 20 COMO ENTRADA DIGITAL BOTÃO SW3
18
19 int main() {
20     while(1) {
21     if (botao1==1){ //VERIFICA O ESTADO DO BOTAO SW1
22         rele1 = 1; // LIGA O RELE
23         wait(0.2); // ESPERA 0.2 MILESEGUNDOS
24
25     }
26     else {
27         rele1 = 0; // DESLIGA O RELE
28         wait(0.2);
29          }
30    if (botao2==1){ //VERIFICA O ESTADO DO BOTAO SW2
31         rele2 = 1; // LIGA O RELE
32         wait(0.2); // ESPERA 0.2 MILESEGUNDOS
33
34                  }
35     else {
36         rele2 = 0; // DESLIGA O RELE 2
37         wait(0.2);
38     }
39        }
40
  }
```

Listagem 4– Exemplo uso dos Reles

A1.4 Triacs

Para o controle de cargas alternadas de alta intensidade tem-se na placa 02 Triacs que possuem 02 opto-acopladores MOC3020 para proteção do circuito, permitindo maior segurança para o sistema, nos Triacs em comparação com os reles comuns não há repique, quando os contatos de um relé abrem ou fecham, eles levam uma fração de segundo para completar esta operação, e durante este intervalo fortes variações da corrente podem ser geradas. Em cargas fortemente indutivas, estes repiques podem causar a geração de pulsos de alta tensão, e em muitos circuitos também são geradas interferências eletromagnéticas (EMI) Num TRIAC o estabelecimento da corrente ou sua interrupção ocorrem de forma constante.

Na Figura 77 um exemplo de ligação para qualquer dispositivo a ser controlado pelos Triacs.

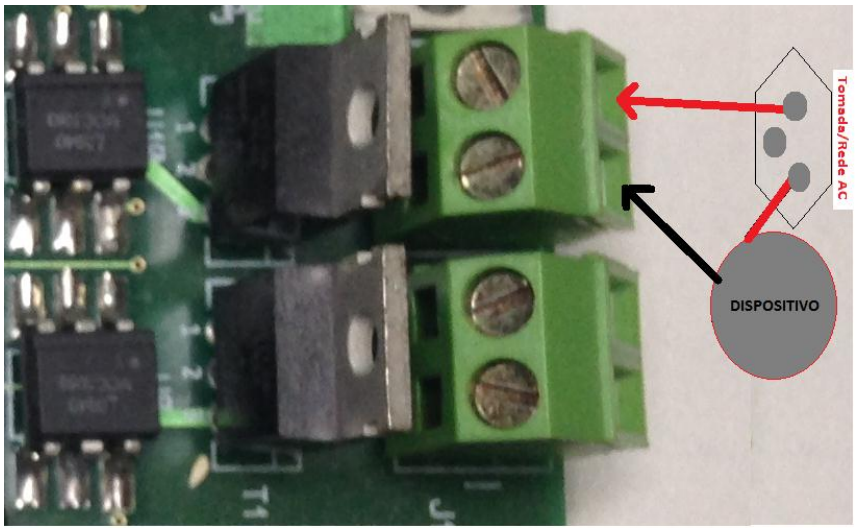


Figura 77 - Ligação Triac

Tabela 6– Descrição Pinos TRIACS

PINO MBED	Descrição
27	TRIAC1
28	TRIAC2

Fonte : Elaborado pelo Autor

```
 1
 2 /* BRASILIA 09-02-2014
 3 AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
 4 CURSO : MESTRADO EM SISTEMAS MECATRONICOS
 5 COMPILADOR : https://MBED.org/compiler
 6 PROGRAMA DA PLACA  GRACO V. 1.4 - TESTE TRIACS USANDO BOTOES SW2 E SW3
 7 OS PINOS USADOS SÃO PINOS 27 E 28 DOS TRIACS 1 E 2 , PINOS 19 E 20 DOS
 8 BOTOES SW2 E SW3
 9 OBS: O PROGRAMA FOI BAIXADO NO MEBED E TESTADO NA PLACA GRACO 1.4 */
10
11 #include "MBED.h"
12
13 DigitalOut triac1(p27); // COLOCA O PINO 27 COMO SAIDA DIGITAL
14 DigitalOut triac2(p28);// COLOCA O PINO 28 COMO SAIDA DIGITAL
15 DigitalIn botao1(p19);// COLOCA O PINO 19 COMO ENTRADA DIGITAL BOTÃO SW2
16 DigitalIn botao2(p20); // COLOCA O PINO 20 COMO ENTRADA DIGITAL BOTÃO SW3
17
18 int main() {
19     while(1) {
20     if (botao1==1){ //VERIFICA O ESTADO DO BOTAO SW1
21         triac1 = 1; // LIGA O TRIAC1
22         wait(0.2); // ESPERA 0.2 MILESEGUNDOS
23
24     }
25     else {
26         triac1 = 0; // DESLIGA O TRIAC1
27         wait(0.2);
28          }
29    if (botao2==1){ //VERIFICA O ESTADO DO BOTAO SW2
30         triac2 = 1; // LIGA O TRIAC2
31         wait(0.2); // ESPERA 0.2 MILESEGUNDOS
32
33                  }
34     else {
35         triac2 = 0; // DESLIGA O TRIAC2
36         wait(0.2);
37     }
38        }
39
40 }
```

Listagem 5– Exemplo uso dos TRIACS

ATENÇÃO: Os Triacs contidos na *PLATAFORMA GRACO 1.4* foram projetado para conexão direta à rede elétrica, e usá-lo de maneira inadequada gera riscos de CHOQUE ELÉTRICO e INCÊNDIO. Só faça projetos usando este recurso caso você já tenha experiência prévia em instalações elétricas.

A2 Usando placa porta expansão para Reles

Na plataforma existe uma expansão CN1 conforme Figura 78 que serve para fazer comunicação via cabo com a placa de reles Graco 1.0.

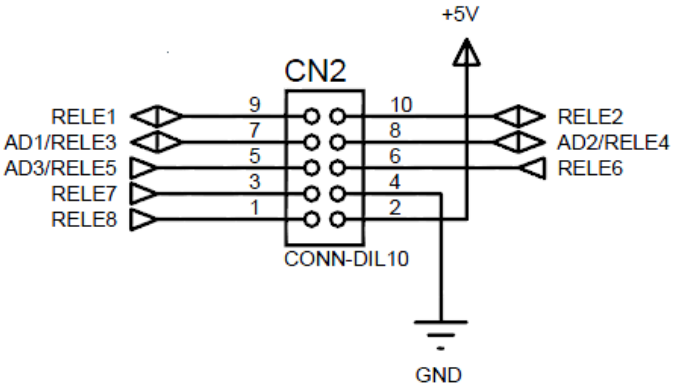


Figura 78 - Circuito Expansão CN2

Figura 79 - Expansão CN2

A placa de reles Graco 1.0 mostrada na figura 94 foi criada para trabalhar via cabo exclusivamente com o *Plataforma Graco 1.4*, mas pode ser utilizada em aplicações sem fio utilizando o modulo *XBee* em qualquer outra plataforma.

Para utilizar os 8 reles da placa usando a expansão CN1, é necessário 8 pinos do microcontrolador para acionar cada um dos reles, para isso usa-se as configurações dos JUMPERS 3,4,6,7 e 8 conforme figura 93

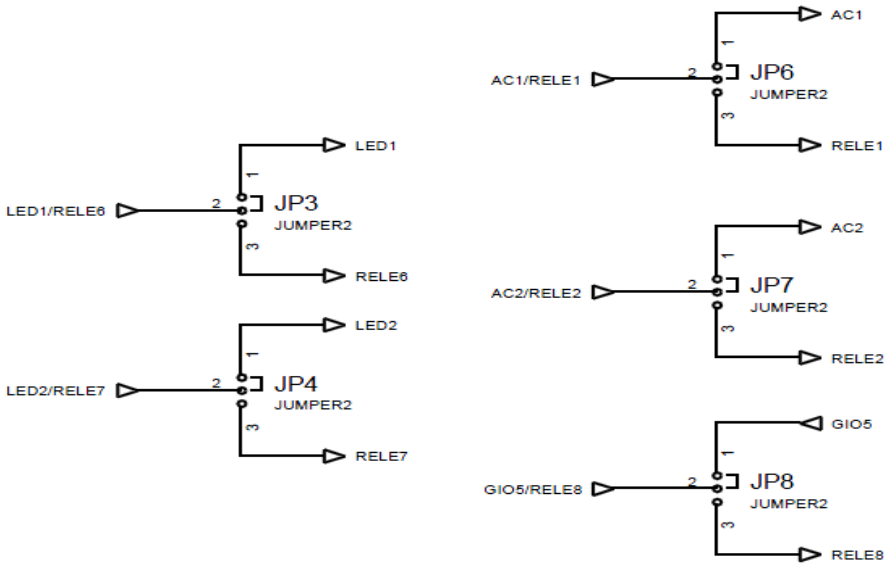


Figura 80 - Configuração Jumpers CN2

A placa de rele (figura 94) pode ser alimentada pelo cabo que usa as saídas de energia da expansão CN2 ou pode ser ligada diretamente em uma fonte de 12v

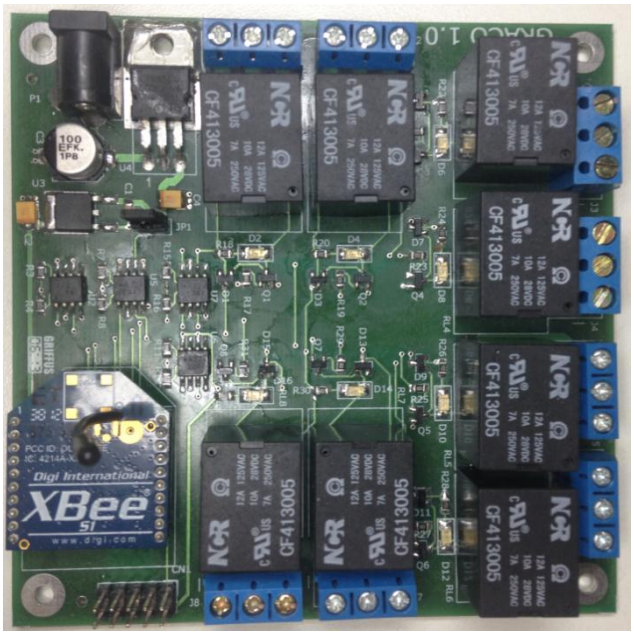


Figura 81 - Placa de Reles

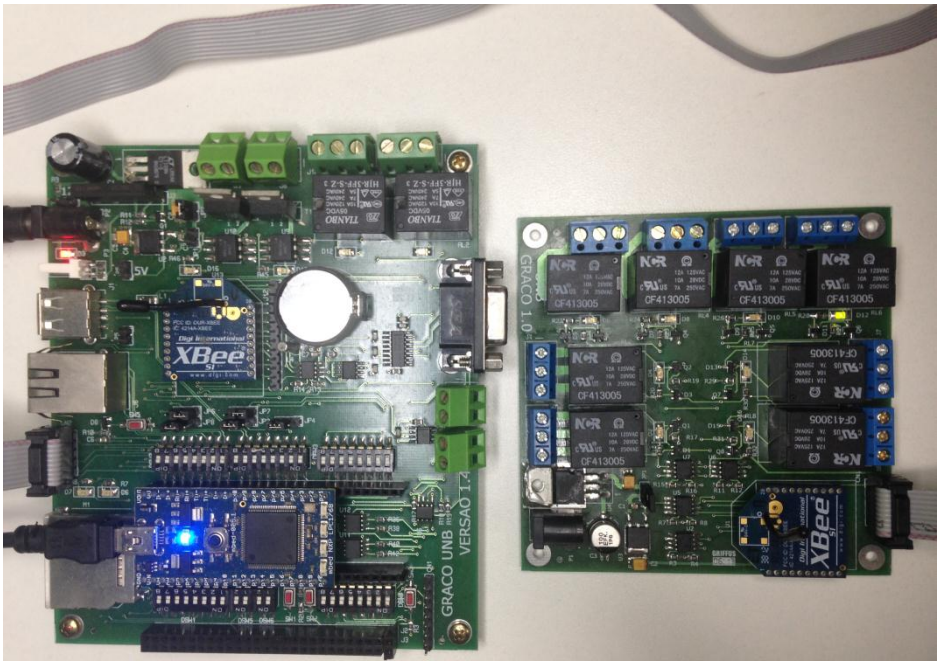


Figura 82 - Kit+placa de Rele

Tabela 7– Descrição dos pinos da placa de rele

PINO MBED	Descrição
28	RELE1
27	RELE2
16	RELE3
17	RELE4
18	RELE5
26	RELE6
25	RELE7
24	RELE8

Fonte: Elaborado pelo Autor

```
/* BRASILIA 12-02-2014 16H10
AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
CURSO : MESTRADO EM SISTEMAS MECATRONICOS
COMPILADOR : https://MBED.org/compiler
PROGRAMA DA PLACA  GRACO V. 1.4 -
 TESTE PLAPA RELES 1 A 8 E BOTOES sw1,sw2 e sw3
OBS: O PROGRAMA FOI BAIXADO NO MEBED E TESTADO NA PLACA GRACO 1.4  e
Placa de Rele Graco 1.0*/

#include "MBED.h"
DigitalOut myled(LED1); // led da placa MBED
DigitalOut rele1(p28); // rele1 ligado ao pino 28
DigitalOut rele2(p27); // rele2 ligado ao pino 27
DigitalOut rele3(p16); // rele3 ligado ao pino 16
DigitalOut rele4(p17); // rele4 ligado ao pino 17
DigitalOut rele5(p18); // rele5 ligado ao pino 18
DigitalOut rele6(p26); // rele6 ligado ao pino 26
DigitalOut rele7(p25); // rele7 ligado ao pino 25
DigitalOut rele8(p24); // rele8 ligado ao pino 24
DigitalIn botao(p8); // SW1 - botao ligado ao pino 08
DigitalIn botao2(p19); // SW2 - botao ligado ao pino 19
DigitalIn botao3(p20); // SW3 - botao ligado ao pino 20
int main() {
while(1) {
    if (botao==1) { //verifica se o botao foi ativado
        rele1 = 1; //rele1 é ligado
        rele2 = 1; //rele2 é ligado
        rele3 = 1; //rele3 é ligado
        myled = 1; //liga o led da placa MBED
        wait(0.2); // espera ,2 segundos
    }
    else { //caso o botao SW1 não esteja precionado
    rele1 = 0; //rele1 é ligado
    rele2 = 0; //rele2 é ligado
    rele3 = 0; //rele3 é ligado
    myled = 0; //liga o led da placa MBED
    wait(0.2); // espera ,2 segundos
    }
    if (botao2==1) { //verifica se o botao foi ativado
        rele4 = 1; //rele4 é ligado
        rele5 = 1; //rele5 é ligado
        rele6 = 1; //rele6 é ligado
        myled = 1; //liga o led da placa MBED
        wait(0.2); // espera ,2 segundos
        }
    else { //caso o botao SW1 não esteja precionado
    rele4 = 0; //rele4 é ligado
    rele5 = 0; //rele5 é ligado
    rele6 = 0; //rele6 é ligado
    myled = 0; //liga o led da placa MBED
    wait(0.2); // espera ,2 segundos
    }
        if (botao3==1) { //verifica se o botao foi ativado
        rele7 = 1; //rele7 é ligado
        rele8 = 1; //rele8 é ligado
        myled = 1; //liga o led da placa MBED
        wait(0.2); // espera ,2 segundos
```

```
58          }
59      else { //caso o botao SW1 não esteja precionado
60      rele7 = 0; //rele7 é ligado
61      rele8 = 0; //rele8 é ligado
62      myled = 0; //liga o led da placa MBED
63      wait(0.2); // espera ,2 segundos
64      }
65
66 }
67       }
```

Listagem 6– Exemplo uso dos Reles da placa externa

A3 Placa rele externa funcionando remotamente via Xbee

Na Listagem 6– Exemplo uso dos Reles da placa externa, o programa aciona a placa de reles de uma forma eficiente, mas, existe uma limitação de distancia, devido a necessidade da utilização de um cabo para fazer as conexões com os pinos do microcontrolador da placa base. Na placa de reles Graco 1.0 foi implementado uma interface de comunicação sem fio com a possibilidade de comunicação serial para ativação de ate 65 mil outras placas de reles, este numero é um limitador de uma rede utilizando módulos *Xbee*.

Para configuração da placa usando o *Xbee* precisa-se de conhecimentos da configuração dos módulos *Xbee*, esta configuração é detalhada no apêndice B desta dissertação, no exemplo da listagem 04 foram utilizados módulos da serie 1 dos *Xbees* da Digi International *e os mesmos foram utilizados com a configuração padrão.*

A Listagem 03 aciona reles em 02 placas distintas e conforme figura 96 pode-se notar que cada modulo Xbee possui um endereço único que deve ser utilizado para o acionamento de cada rele em placas distintas sem a necessidade de ter vários pinos disponíveis para isso, utilizando apenas os Pinos 13 e 14 que são os pinos RX E TX que permitem a comunicação serial .

Figura 83 - Endereço único *Xbee*

A Figura 84 mostra a placa de reles com 8 reles NA/NF com o modulo *Xbee* devidamente instalado, para facilitar a instalação remota cada modulo tem sua própria fonte interna facilitando o acionamento de até 8 equipamentos próximos a placa remotamente via a interface *Xbee*.

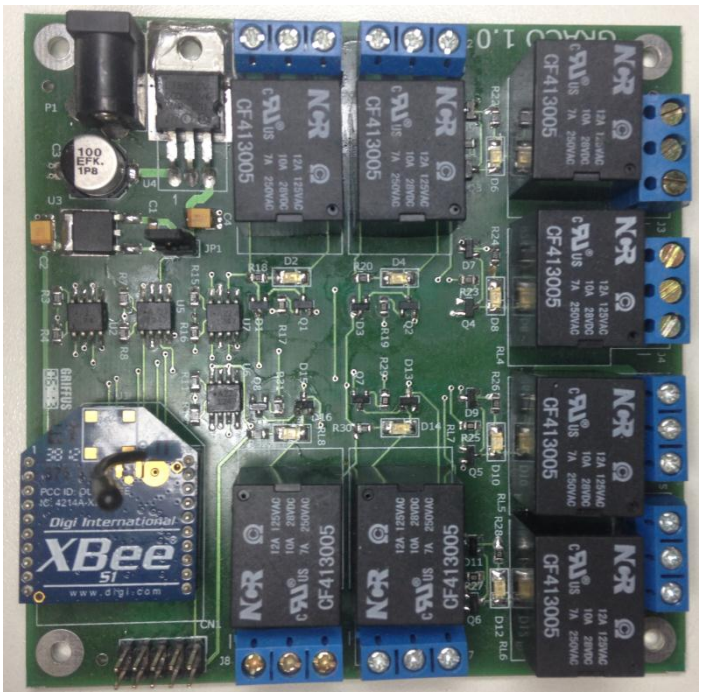


Figura 84 - Placa Rele com *Xbee*

Na *plataforma Graco 1.4* não existe a necessidade de nenhuma configuração física na placa, é somente necessário o desenvolvimento do *software* de controle para a comunicação serial das placas de rele.

Na listagem 4 é importante observar que das linhas 29 a 48 possui o frame de configuração do modulo *Xbee* instalado na placa de rele 1 e das linhas 78 a 97 é a configuração do frame que se comunica com o modulo instalado na placa 02, para reprodução deste exemplo é necessário a troca do endereço do módulos contido nas linhas 35 a 41 e o byte do frame da linha 48 que é o Checksum responsável ela validação do frame.

```
        /* BRASILIA 19-03-2014 13H00
 9 AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
10 CURSO : MESTRADO EM SISTEMAS MECATRONICOS
11 COMPILADOR : https://MBED.org/compiler
12 PROGRAMA DA PLACA  GRACO V. 1.4 E PLACA DE RELE 1.0 - TESTE XBEE
13 */
14
15 #include "MBED.h"
16 Serial Xbee1(p13,p14); //TX E RX do MBED
17 DigitalOut myled(LED3);//Cria uma variavel para Led 3
18 DigitalIn botao(p8); // SW1 - botao ligado ao pino 08
19 DigitalIn botao2(p20); // SW2 - botao ligado ao pino 20
20
21 int main()
22 {
23     myled = 0;
24
25     while (1) {
26     if (botao==1) { //verifica se o botao foi ativado
27         myled = 1; // Led 3 Off
28         // Inicio do FRAME XBEE 1 PRIMEIRA PLACA DE RELAY
29         Xbee1.putc(0x7E);  //inicia byte API frame
30         Xbee1.putc(0x00);
31         Xbee1.putc(0x10);
32         Xbee1.putc(0x17);
33         Xbee1.putc(0x05);
34         Xbee1.putc(0x00);
35         Xbee1.putc(0x13); //XBEE Endereço
36         Xbee1.putc(0xA2); //XBEE Endereço
37         Xbee1.putc(0x00); //XBEE Endereço
38         Xbee1.putc(0x40); //XBEE Endereço
39         Xbee1.putc(0xA1); //XBEE Endereço
40         Xbee1.putc(0x3A); //XBEE Endereço
41         Xbee1.putc(0xB6); //XBEE Endereço
42         Xbee1.putc(0xFF);
43         Xbee1.putc(0xFE);
44         Xbee1.putc(0x02);
45         Xbee1.putc(0x44);    // commando ASCII caracter "D" PINO NO XBEE
46         Xbee1.putc(0x34);    // commando ASCII caracter "4" PINO NO XBEE
47         Xbee1.putc(0x05);    // D 4 HIGH ATIVA O PINO D4 EM HIGH LIGA RELAY
48         Xbee1.putc(0xE1);    //CHECKSUM
49         wait(1);
50         myled = 0; //Desliga Led 3
51         wait(1);
52         }
53         else {
54         Xbee1.putc(0x7E);        //inicia byte API frame
55         Xbee1.putc(0x00);
56         Xbee1.putc(0x10);
57         Xbee1.putc(0x17);
58         Xbee1.putc(0x05);
59         Xbee1.putc(0x00);
60         Xbee1.putc(0x13); //XBEE Endereço
61         Xbee1.putc(0xA2); //XBEE Endereço
62         Xbee1.putc(0x00); //XBEE Endereço
63         Xbee1.putc(0x40); //XBEE Endereço
64         Xbee1.putc(0xA1); //XBEE Endereço
65         Xbee1.putc(0x3A); //XBEE Endereço
66         Xbee1.putc(0xB6); //XBEE Endereço
67         Xbee1.putc(0xFF);
68         Xbee1.putc(0xFE);
```

```
69          Xbee1.putc(0x02);
70          Xbee1.putc(0x44);    // commando ASCII caracter "D" PINO NO XBEE
71          Xbee1.putc(0x34);    // commando ASCII caracter "4" PINO NO XBEE
72          Xbee1.putc(0x04);    // D 4 LOW DESATIVA O PINO D4 EM LOW DESLIGA RELAY
73          Xbee1.putc(0xE2);//CHECKSUM
74          wait(1);
75           }
76          if (botao2==1) { //verifica se o botao foi ativado
77                // Inicio do FRAME XBEE 2 SEGUNDA PLACA DE RELAY
78          Xbee1.putc(0x7E);
79          Xbee1.putc(0x00);
80          Xbee1.putc(0x10);
81          Xbee1.putc(0x17);
82          Xbee1.putc(0x05);
83          Xbee1.putc(0x00);
84          Xbee1.putc(0x13);
85          Xbee1.putc(0xA2);
86          Xbee1.putc(0x00);
87          Xbee1.putc(0x40);
88          Xbee1.putc(0xA0);
89          Xbee1.putc(0x40);
90          Xbee1.putc(0x37);
91          Xbee1.putc(0xFF);
92          Xbee1.putc(0xFE);
93          Xbee1.putc(0x02);
94          Xbee1.putc(0x44);
95          Xbee1.putc(0x34);
96          Xbee1.putc(0x05);
97          Xbee1.putc(0x5B);
98          wait(1);
99          }
100         else {
101         Xbee1.putc(0x7E);
102         Xbee1.putc(0x00);
103         Xbee1.putc(0x10);
104         Xbee1.putc(0x17);
105         Xbee1.putc(0x05);
106         Xbee1.putc(0x00);
107         Xbee1.putc(0x13);
108         Xbee1.putc(0xA2);
109         Xbee1.putc(0x00);
110         Xbee1.putc(0x40);
111         Xbee1.putc(0xA0);
112         Xbee1.putc(0x40);
113         Xbee1.putc(0x37);
114         Xbee1.putc(0xFF);
115         Xbee1.putc(0xFE);
116         Xbee1.putc(0x02);
117         Xbee1.putc(0x44);
118         Xbee1.putc(0x34);
119         Xbee1.putc(0x04);
            Xbee1.putc(0x5C);

   wait(1);
            }
       }
       }
```

Listagem 7: Código programa teste placa Rele *XBEE*

A4 Expansão pinos dos módulos XBEE

Na figura 98 pode-se observar que a plataforma disponibiliza acesso a todos os pinos do modulo *XBEE* através de um barramento de expansão isso permite a simulação de uma infinidade de aplicações utilizando acesso remoto com a tecnologia zigbee.

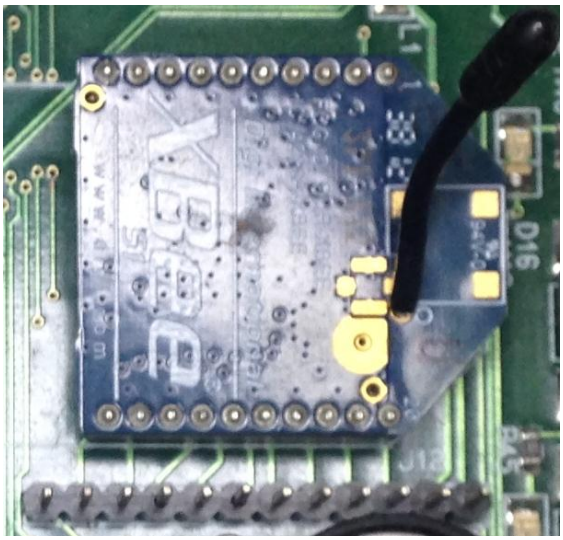


Figura 85 - Barramento expansão

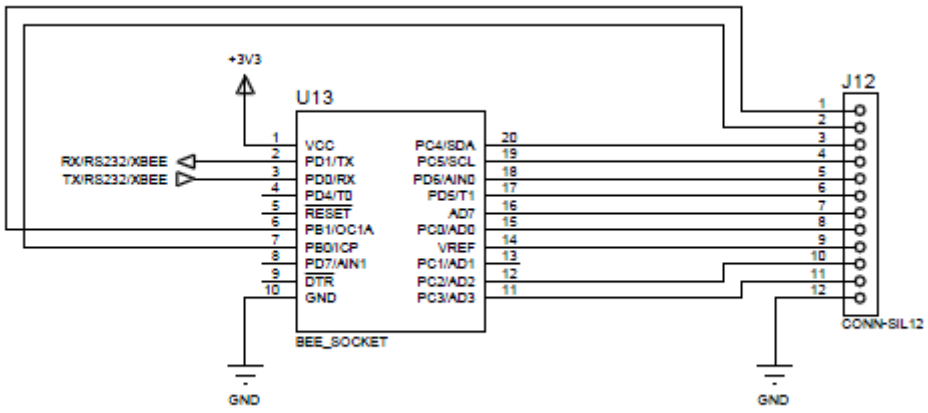


Figura 86 - Circuito Barramento expansão

A5 Comunicação serial RS-232

A comunicação RS-232 usa uma comunicação bidirecional utilizando o *driver/receiver MAX232D é seu circuito é mostrado na* Figura 87*.*

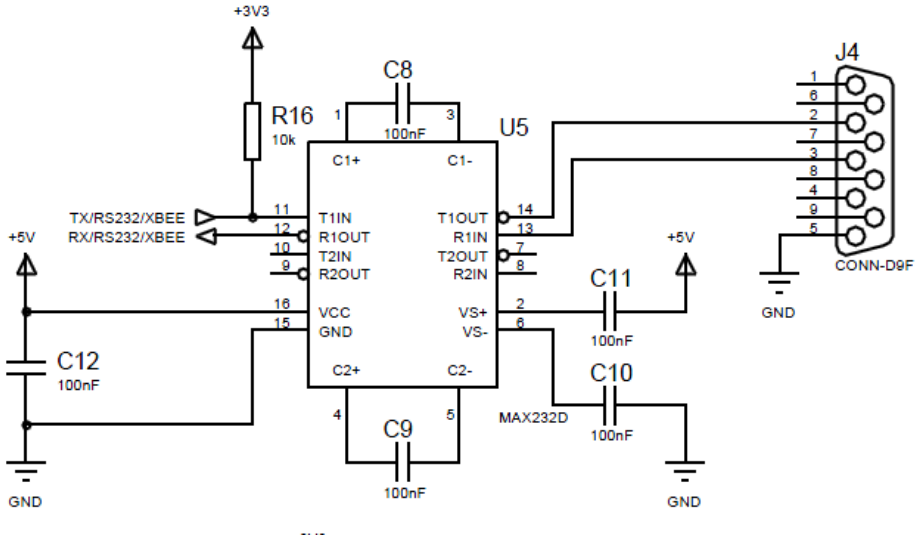


Figura 87 - Circuito ligação RS-232

Para um melhor entendimento na Figura 88 é mostrado a descrição da *interface de conexão.*

Pino	Descrição	IO pino
1		NC
2	TXD	UART1_TX
3	RXD	UART1_RX
4	DRT	DRT(Control RESET)
5	GND	GND
6		NC
7	RTS	RTC
8		NC
9		NC
Shield	*Shield*	GND

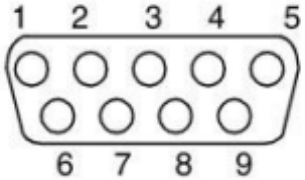


Figura 88 - Pinagem RS232

A6 Comunicação serial RS-485

A comunicação RS-485 usa uma comunicação bidirecional utilizando o *transceiver SN75176BP é seu circuito é mostrado na* Figura 89*.*

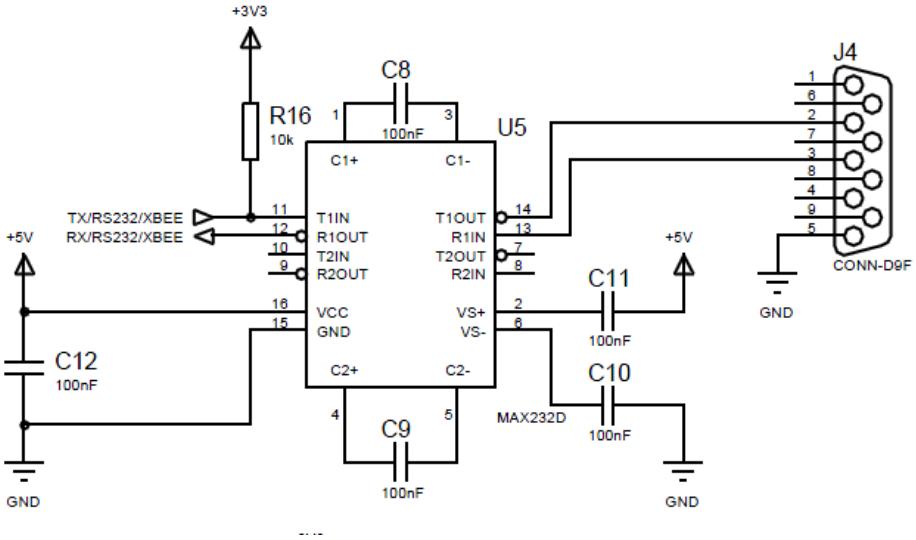


Figura 89 -Circuito ligação RS-485

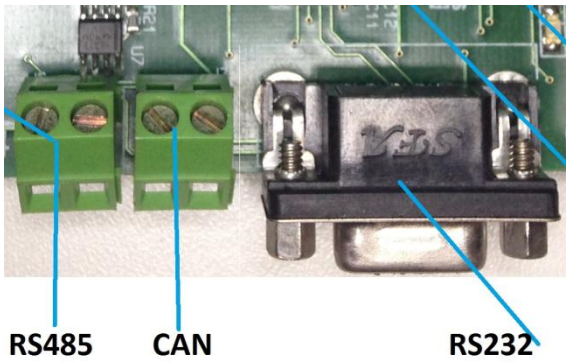


Figura 90 - Interfaces Seriais

A7 Comunicação serial **CAN**

A comunicação CAN usa uma comunicação utilizando o *transceiver N65HVD233D é seu circuito é mostrado na* Figura 91 *e 105.*

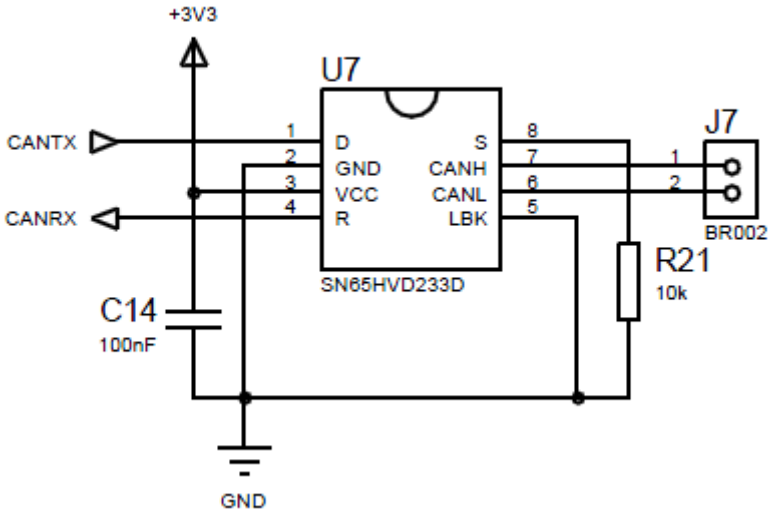


Figura 91 - Circuito ligação CAN

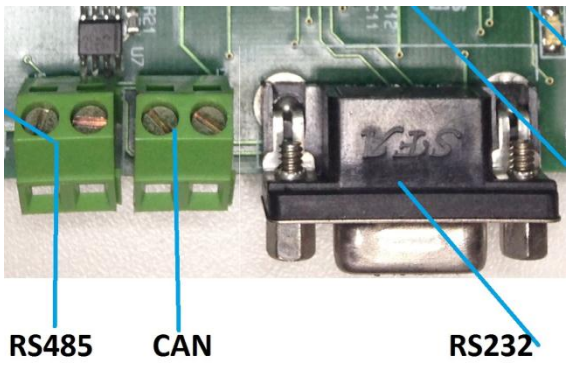


Figura 92 - Interfaces Seriais

A8 Comunicação **I2C**

A memória serial AT24C04B, o relógio de tempo real Ds1307 e o SD CARD estão ligados no mesmo barramento I2C (SCL e SDA) conforme Figura 93.

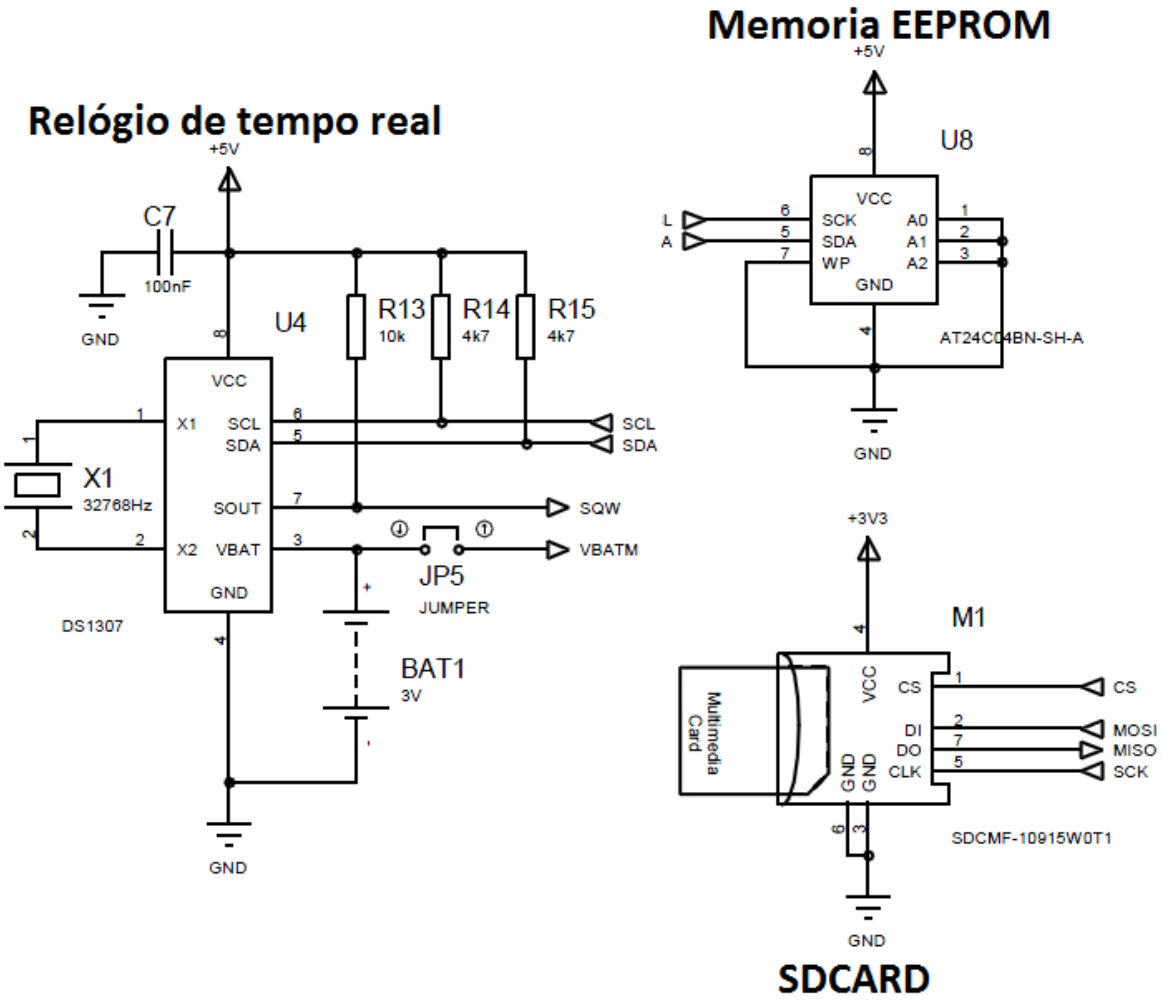


Figura 93 - circuitos I2C

Para utilização destes recursos é necessário a criação de *software* especifico para cada aplicação utilizando as bibliotecas de utilização da comunicação I2C.

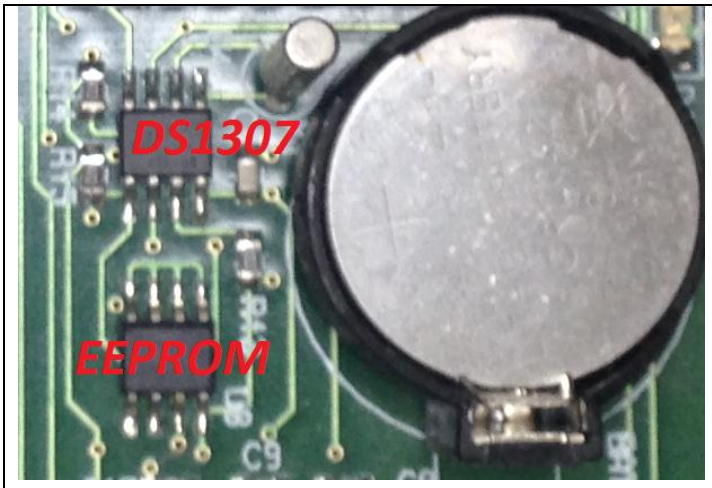


Figura 94 -Interfaces I2C da plataforma

Figura 95 - SDCARD

A9 Comunicação **ETHERNET**

O Plataforma Graco 1.4 tem onboard suporte para 10/100 Mbit / s Comunicação Ethernet. O chip LPC1768 suporta a interface RMII. O conector RJ45 tem LEDs integrados para indicar a conectividade de link e colisões , ele possui magnetismo internos e os filtros de rede conforme Figura 96

Figura 96 - MagJack

Fonte: Sparkfun.com

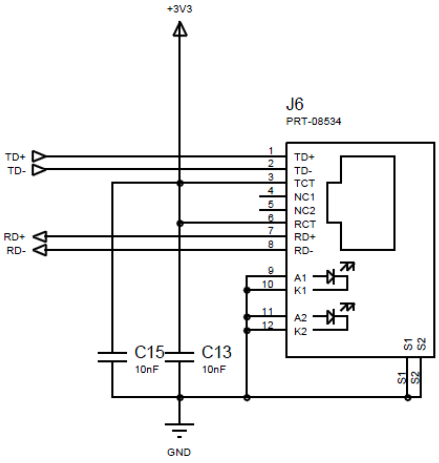


Figura 97 - Circuito *Ethernet*

A utilização da *Ethernet* possibilita que as aplicações desenvolvidas na plataforma possam ser acessadas em uma rede interna ou de qualquer lugar do planeta pela internet.

A configuração física do circuito na placa não precisa de alteração, necessitando-se apenas do desenvolvimento do *software*.

Na *IDE* do *MBED* já existe bibliotecas prontas para transforma a plataforma em um servidor *Ethernet* , necessitando apenas do desenvolvimento do *software* para a comunicação dos pinos de *I/O* .

A10 Barramento de **expansão CN1**

Existe na plataforma um barramento nomeado CN1 que possui 4 pinos analógicos de uso geral , sendo I1 e I2 entrada analógica pino 19 e 20 do MBED e os pinos I3 e I4 somente podem ser usado quando estive usando uma placa LPCxpresso , este barramento possui ainda uma alimentação e aterramento de 5v , pinos 01 e 02 respectivamente conforme circuito demostrado na Figura 98

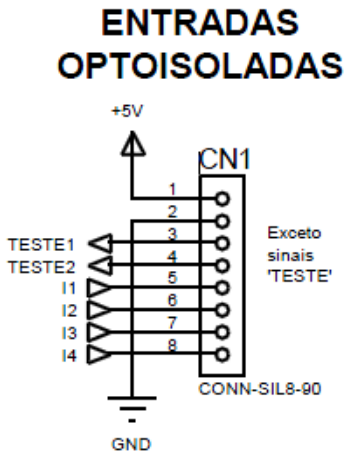


Figura 98 - Circuito CN1

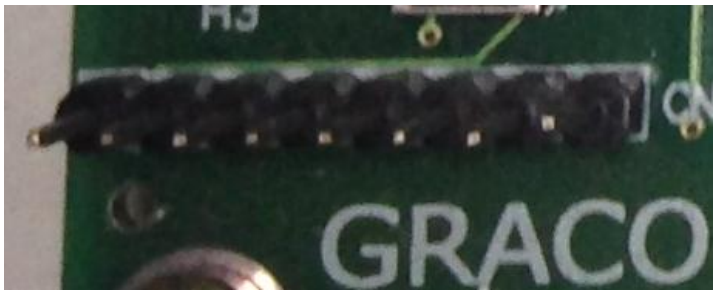


Figura 99 - Barramento CN1

A11 Barramento **de expansão J2 E J3**

Para permitir a integração com outros hadwares e possibilidade de teste dos mais variados tipos de aplicações que necessitem o controle de um microcontrolador eficiente como o ARM de 32 bits LPC 1768 utilizado na plataforma, (conforme Figura 100) existe um barramento com acesso a todas os pinos do MBED ou de qualquer LPCxpresso a ser utilizado .

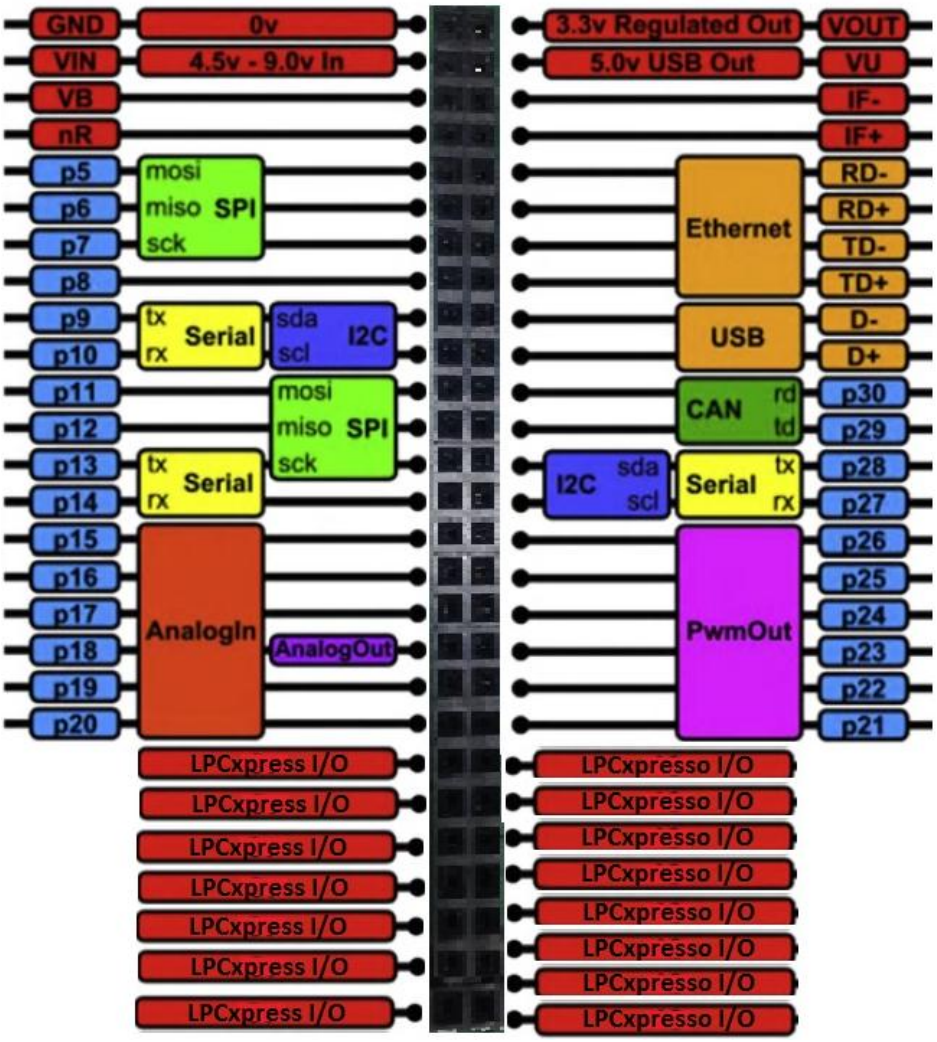


Figura 100 -Funções barramento

Neste barramento pode-se utilizar a grande maioria dos Shields existente no mercado, além disso, é possível o desenvolvimento de uma vasta gama de novos sensores que podem ser testados usando esta opção.

No exemplo 01 da Figura 101 há um circuito para comunicação de um opto-sensor que pode ser simulado em uma breadboard e as conexões com os pinos do barramento de expansão da Figura 100.

Na listagem 5 tem-se um programa que pode ser compilado na IDE e testado para comprovar a funcionalidade.

Figura 101 - Opto KTIR0621DS

Fonte: http://uk.farnell.com

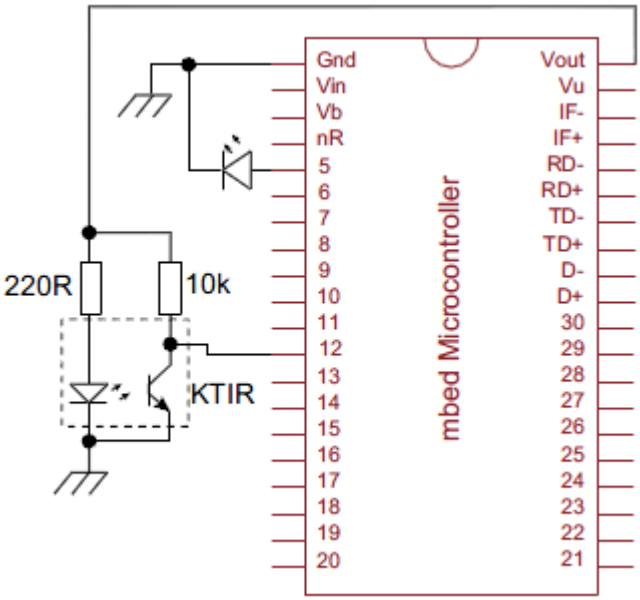


Figura 102 - Circuito de ligação

Fonte: MBED book

No circuito da Figura 102 a saída do sensor é ligado ao pino 12, que é configurado no programa da listagem 5 linha 12 como uma entrada digital. Quando não há nenhum objeto no sensor, então a luz incide sobre a opto-sensor, a saída do sensor está no estado lógico 0. Quando o feixe for interrompido, então a saída é a lógica 1.,então o programa liga o LED quando o feixe é interrompido, ou seja, um objeto foi detectado.

```
1 /* BRASILIA 22-06-2014
2 AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
3 CURSO : MESTRADO EM SISTEMAS MECATRONICOS
4 COMPILADOR : https://MBED.org/compiler
5 PROGRAMA DA PLACA  GRACO V. 1.4 - TESTE DO BARRAMENTO DE EXPANSÃO
6 OS PINOS USADOS SÃO PINOS 5 LIGAÇÃO DE UM LED E PINO 12 PARA LEITURA DO
7 OPTO-SENSOR
8 OBS: O PROGRAMA FOI BAIXADO NO MEBED E TESTADO NA PLACA GRACO 1.4 */
9
10 #include "MBED.h"
11 DigitalOut led(p5);
12 DigitalIn opto(p12);
13 int main() {
14 while(1) {
15 if (opto==1) //Entrada = 1 se o feixe for interrompida
16 led = 1; //liga o led se o feixe for interrompido
17 else
18 led = 0; //o led fica desligado caso o feixe esteja normal
19 }
  }
```

Listagem 8 – Programa teste do Opt-sensor

Conclusão do Apêndice **A**

Este apêndice é apenas um manual básico da utilização da plataforma Graco 1.4, onde permite através de vários exemplos que os futuros usuários possam ter uma compreensão do funcionamento físico e da programação necessária para implantação de varias aplicações.

Para aprofundar os estudos e testar varias possibilidades sugere-se acessar o link da pagina oficial do MBED dentro do link https://MBED.org/cookbook/Homepage, neste existem exemplos dos mais variados tipos de aplicações, tais como, áudio, câmeras, inclinometros , compasso, GPS, sensores de temperatura, processamento de sinais, wireless e outros.

Sugere-se a aqueles que irão utilizar as placas de reles controladas pelos módulos *XBee* ler o apêndice B que descreve como configurar e trabalhar com os módulos existentes nas placas.

É importante ainda o estudo da linguagem C++ utilizados na maioria das plataformas de sistemas embarcados, pois a correta aplicação dos algoritmos pode facilitar a implementação da aplicação a ser testada na plataforma.

APÊNDICE B – CONFIGURAÇÃO MODULOS XBEE

B1 - Visão Geral

Muitas pessoas pensam que *ZigBee* e *Xbee* são a mesma coisa. Isso não é verdade. *ZigBee* é um protocolo de comunicação padrão para baixo consumo de energia, redes *mesh* sem fio. *Xbee* é uma marca de rádio que suporta uma variedade de protocolos de comunicação, incluindo o *Zig-Bee*, 802.15.4, e Wi-Fi, entre outros.

O protocolo *ZigBee* é um padrão da mesma maneira que o *Bluetooth* é um padrão, qualquer dispositivo do fabricante que suporta totalmente o padrão *ZigBee* podem se comunicar com dispositivo *ZigBee* de qualquer outra empresa., portanto assim como o fone de ouvido Bluetooth da Motorola pode comunicar com o um *iPhone* da *Apple*, um interruptor de luz *CentralLite ZigBee* podem se comunicar com uma fechadura *Black & Decker* (FALUDI Robert, 2011)

Algumas Aplicações de sensores usando *os rádios XBee que pode-se agregar em varias aplicações:*

n. - Sensor de umidade;
o. - Sensor de temperatura;
p. - Sensor de velocidade do vento;
q. - Sensor de direção do vento;
r. - Sensor de pressão atmosférica;
s. - Controle de iluminação; ;
t. - Controle de aquecimento;
u. - Controle de Ventilação;
v. - Controle de Irrigação;

w. - Alarmes;
x. - Controle de cancelas;
y. - Controle de portas e portões; .
z. - Aplicações automotivas;

B2 – Características:

Na Figura 103 pode-se ver as 03 formas diferentes de antenas disponíveis nos módulos e abaixo a configuração do módulo *XBee* serie 1 da *Digi International* que é usado na plataforma Graco 1.4 e nas placas de reles Graco 1.0 .

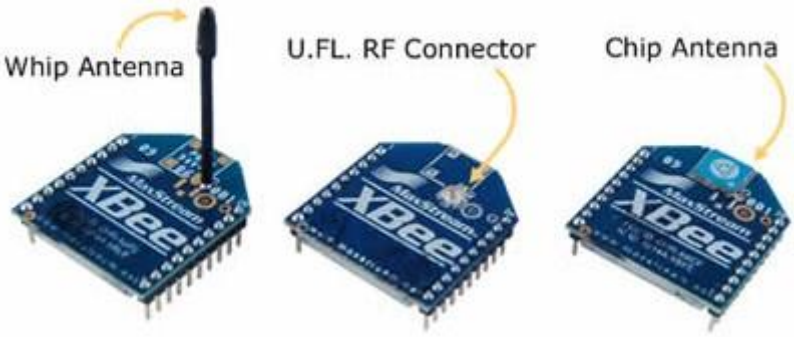


Figura 103– Módulos *Xbee*

Performance

a) Rendimento da Potência de saída: 1 mW (0 dBm);
b) Alcance em ambientes internos/zonas urbanas: 30m;
c) Alcance de RF em linha visível para ambientes externos: 100m;
d) Sensibilidade do receptor: -92 dBm;
e) Freqüência de operação: ISM 2.4 GHz;
f) Taxa de dados de RF: 250.000 bps;
g) Taxa de dados da Interface (Data Rate): 115.200 bps;

Alimentação

a) Tensão de alimentação: 2.8 à 3.4v;
b) Corrente de transmissão (típico): 45 mA @ 3.3 V;
c) Corrente de Recepção (típico): 50 mA @ 3.3 V;

d) Corrente de Power-down Sleep: <10 µA;

Propriedades físicas

a) Dimensões: (2.438cm x 2.761cm);
b) Peso: 0.10 oz (3g);
c) Temperatura de operação: -40 to 85º C (industrial);
d) Opções de antena: Conector U.FL RF, Chip ou Chicote (whip);

Rede

a) Tipo de espalhamento espectral: DSSS (Direct Sequence Spread Spectrum);
b) Manipulação de erro: Retransmite novamente (Retries) & reconhecimento;
c) Topologia de Rede: Peer-to-peer(Par-a-par), ponto-a-ponto, ponto-a-multiponto e malha;
d) Endereçamento: 65.000 endereços de rede disponíveis para cada canal;
e) Opções de filtros: PAN ID, canais e endereços;
f) Criptografia: 128-bit AES;
g) Número de canais selecionáveis via *software*: 16 canais de sequência direta;

Geral

a) - Faixa de freqüência: 2.4000 - 2.4835 GHz;

Tabela 8- Funções dos pinos do modulo *XBee serie 1*

Pino #	Nome	Direção	Descrição
1	VCC	-	Alimentação 3,3v
2	DOUT	Saída	Saída de dados da UART
3	DIN / $\overline{\text{CONFIG}}$	Entrada	Entrada de dados da UART
4	DO8*	Saída	Saída digital 8
5	$\overline{\text{RESET}}$	Entrada	Inicializa módulo (um pulso nível 0 de pelo menos 200ms)
6	PWM0 / RSSI	Saída	Saída do PWM 0 / Indicador de Força do sinal de RF (RX)
7	PWM1	Saída	Saída do PWM 1
8	(Reservado)	-	Ainda não tem uma função definida (futura implementação)

9	$\overline{\text{DTR}}$ / SLEEP_IRQ / DI8	Entrada	Linha de Controle da Função Sleep ou Entrada digital 8
10	GND	-	Terra
11	AD4 / DIO4	Entrada/Saída	Só Entrada Analógica 4 ou Entrada/Saída Digital 4
12	$\overline{\text{CTS}}$ / DIO7	Entrada/Saída	Controle de Fluxo CTS ou Entrada/Saída Digital 7
13	ON / SLEEP	Saída	Indicador de Estado do Módulo
14	VREF	Entrada	Voltagem de Referência para as Entradas A/D
15	Associação / AD5 / DIO5	Entrada/Saída	Indicador de Associação, só Entrada Analógica 5 ou Entrada/Saída Digital 5
16	$\overline{\text{RTS}}$ / AD6 / DIO6	Entrada/Saída	Controle de Fluxo RTS, só Entrada Analógica 6 ou Entrada/Saída Digital 6
17	AD3 / DIO3	Entrada/Saída	Só Entrada Analógica 3 ou Entrada/Saída Digital 3
18	AD2 / DIO2	Entrada/Saída	Só Entrada Analógica 2 ou Entrada/Saída Digital 2
19	AD1 / DIO1	Entrada/Saída	Só Entrada Analógica 1 ou Entrada/Saída Digital 1
20	AD0 /DIO0	Entrada/Saída	Só Entrada Analógica 0 ou Entrada/Saída Digital 0

O protocolo de comunicação ZigBee define três tipos de papéis para os dispositivos em sua rede, sendo eles: Coordenador, Roteador e Dispositivo final. Segue a explicação de cada um:

a) Coordenador: é o único responsável por formar uma rede *Xbee*, portanto sua presença é obrigatória em todas as situações. O coordenador, também conhecido como ZC (ZigBee Coordinator), estabelece parâmetros (canal de operação e número lógico) para formar uma rede. Depois de estabelecidos os parâmetros, os roteadores e dispositivos finais podem se integrar à rede formada. Após essa formação, o ZC(ZigBee Coordinator) funciona como um roteador, podendo participar no redirecionamento de pacotes de dados e ser uma fonte de destino para os mesmos.
b) Roteador: é um nó que cria e/ou mantém as informações da rede, utilizando-as para determinar a melhor rota para um determinado pacote de dados. Os roteadores, também conhecidos como ZR (ZigBee Router), devem se integrar à rede antes de permitir que outros roteadores e dispositivos finais se integrem a ele. Sua utilização dentro de uma rede é opcional.

Dispositivo final: é um nó que deve sempre interagir com um roteador ou um coordenador (nó pai) na rede para receber ou transmitir dados. Os dispositivos

finais, também conhecidos como ZEB (ZigBee End Device), não possuem capacidade para redirecionar pacotes de dados.

B3 - Endereçamento

Quase todas as pessoas tem um endereço onde ela pode ser encontrada. Muitas pessoas têm mais de um endereço, endereços de email, números de telefone, números de passaporte, e lista continua. Cada tipo de endereço ou identificador serve a um propósito diferente. É o mesmo com os rádios *XBee, se* deseja-se enviar uma mensagem de ZigBee, precisa-se saber o endereço do rádio de destino.

Cada rádio *XBee* tem um endereço de 64 bits número de série impresso na parte de trás (Figura 117). A parte de cima do endereço será 0013A200, que é uma Digi internacional designação. A parte abaixo do endereço será diferente para cada rádio. Ele será parecido com este: 403B9E21.

Figura 104 - Endereço único *Xbee*

B4 – Forma de operação dos módulos

Modo Transparente

Os dados recebidos da UART pelo pino DI(RX) são colocados na fila para transmissão via RF. Já os dados recebidos do canal de RF, são transmitidos através do pino DO(TX).

No modo transparente os dados são transmitidos e recebidos da mesma forma que uma comunicação Serial RS232 padrão. Os módulos dispõem de buffers de transmissão e recepção para um melhor desempenho na comunicação serial. (ROGECOM,2011)

Modo API (Application Programming Interface)

Esse modo de operação é uma alternativa ao modo de operação Transparente padrão. O modo API é baseado em frame e assim estende o nível para o qual uma aplicação de Host pode interagir com as capacidades de Rede do módulo.

No modo API os dados transmitidos e recebidos estão contidos em frames, que definem operações ou eventos dentro do módulo. Através desse modo de operação é possível um determinado módulo enviar endereço fonte, endereço destino, nome de um determinado nó, sinal RSSI, seu estado entre outros dados. (ROGECOM,2011)

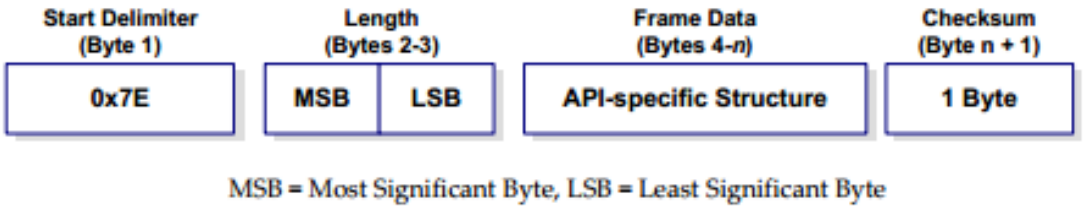


Figura 105 -Formato Frame modo API

Fonte: Digi International

Os frames indicam parâmetros importantes para o funcionamento da comunicação, sendo eles:

a) *Start Delimiter:* Indica o início de um pacote (0x7E) ;
b) *Length:* Comprimento da mensagem;
c) *Frame* Data: dados a serem incluídos na mensagem, é a parte do pacote que teremos que definir e implementar;
d) *Checksum:* responsável por verificar a integridade da mensagem.

B5 – Configurando **os Módulos**

Os módulos podem ser programados, mediante uma sequencia de comandos em modo texto (também chamados de comandos AT) em qualquer programa de tipo terminal ou na interface gráfica como a ferramenta disponibilizada pela *Digi International* chamada de *X-CTU* ela possui 71 parâmetros de configuração¸ que estão agrupados em sete categorias: rede e segurança, interface de RF, baixo consumo, interface serial, configuração de I/O, diagnóstico e opções, algumas destas configurações podem ser vistas na tabela 11.

Tabela 9- Alguns comandos AT da configuração *Xbee*

Comando	Função	Parâmetros
AP	O comando AP é usado para ativar o módulo de RF para operar utilizando uma API	0 - Desabilita modo API 1 - Habilita API
BD	Configura a taxa de dados da interface serial.	0- 1200 1- 2400 2- 4800 3- 19200 4- 38400 5- 57600 6- 115200
CH	O comando CH é usado para definir / ler o canal de operação em que Conexões	

	RF são feitas entre os módulos de RF	
CN	O comando CN é usado para explicitamente sair do módulo de RF do AT Modo de comando	
D0 – D4	Os comandos D0, D1, D2, D3 e D4 são usados para selecionar / ler o comportamento dos seus respectivos AD / DIO linhas (pinos 20, 19, 18, 17 e 11, respectivamente).	0 – Disable 1 –N/A 2 – ADC 3 –DI 4 –DO low 5 –DO high
D5	O comando é usado para D5 selecione / ler o comportamento da linha DIØ5 (pino 15). As opções incluem: a) Indicador de Associação (LED pisca quando o b) módulo é associado) c) Conversor analógico-digital 1 Indicador Associado d) Entrada Digital 2 ADC e) Saída Digital	0 – disable 1 – indicador de associação 2 –ADC 3 – DI 4 – DO low 5 – DO high
D6	O comando D6 é usado para selecionar / ler o comportamento da linha DIO6 (pino 16). As opções incluem: • controle de fluxo RTS • Conversor analógico-digital • Entrada Digital • Saída Digital	0-disable 1-Controle de Fluxo 2-N/A 3-DI 4-DO low 5-DO high
D7	O comando D7 é usado para selecione / ler o comportamento da linha DIO7 (pino 12). a) controle de fluxo CTS b) Conversor analógico para digital c) Entrada Digital Saída Digital RS485 TX Habilita (esta saída é 3V nível CMOS, e é útil em uma conversão de circuito CMOS para RS485 3V)	0-disable 1-Controle de Fluxo 2-N/A 3-DI 4-DO low 5-DO high 6-RS485 TX Enable low 7-RS485 TX Enable High
D8	O comando é usado para D8 selecione / ler o comportamento da linha DI8 (pino 9). Este comando permite configurar o pino para funcionar como uma entrada digital. Esta linha também é usado com Sleep Pin.	0 – disable 3 – DI

DL	O comando DL é usado para definir e ler os 32 bits menos significativos do endereço de 64 bits do módulo RF de destino	
DH	O comando DH é usado para definir e ler os 32 bits superiores do endereço de 64 bits do módulo RF de destino	
MY	O comando MY é usado para definir e ler o endereço de origem de 16 bits do módulo de RF	
ND	O comando ND é usado para descobrir e relatar todos os módulos em seu canal de operação atual (parâmetro CH) e PAN ID (parâmetro ID)	

A tabela 9 possui os principais comandos de configuração dos módulos *Xbee*. Para testar alguns destes comandos primeiramente precisa-se acessar o programa *X-CTU* disponibilizado gratuitamente pela *DIGI international* , nas Figura 106 e 120 pode-se visualizar a tela inicial do *X-CTU* acessando um módulo *Xbee* de uma das placas de rele , pode-se observar no teste de comunicação que tem-se a informação de que tipo de módulo é utilizado e o numero serial contendo o endereço único deste módulo.

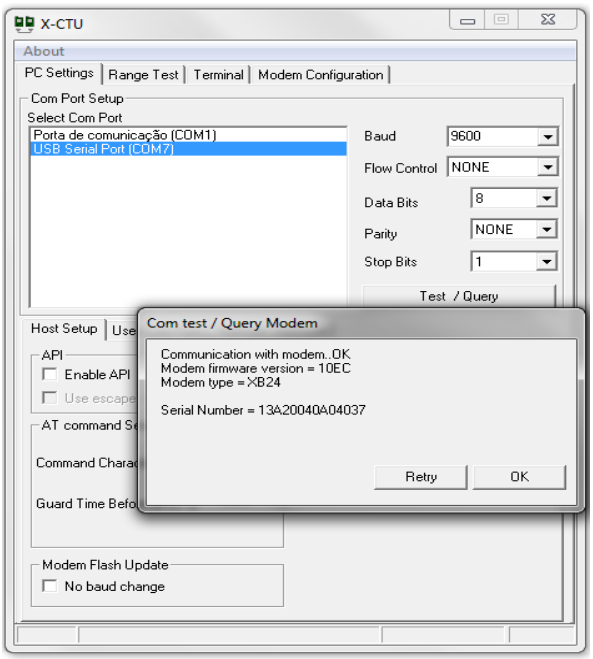


Figura 106 - Tela inicial X-CTU

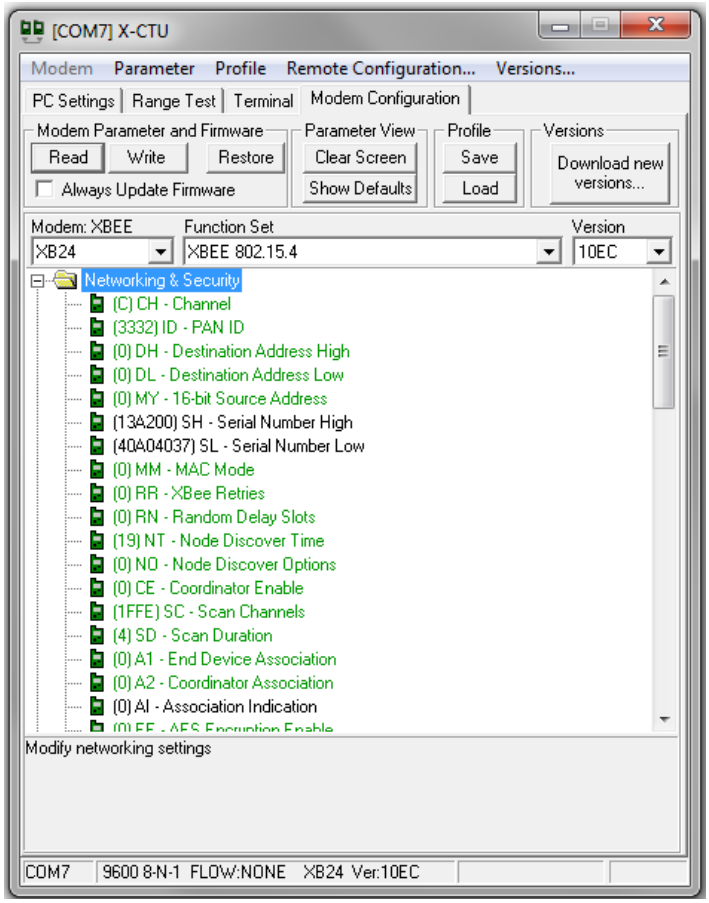


Figura 107 - Tela com parâmetros do módulo

Na tabela 9 tem os comandos para fazermos as seguintes modificações em no módulo XBee utilizado:

a) Modificar o endereço de destino
b) Fazer a leitura deste endereço
c) Modificar o endereço do módulo
d) Ler o endereço do módulo
e) Fazer a gravação das modificações

Tabela 10– Comandos AT utilizados no exemplo teste

Comando	Resposta do *Xbee*	Descrição
+++	OK	Coloca o módulo no modo de comando
ATDL2014<ENTER>	OK	Modifica o endereço destino (do módulo local para 2014
ATDL<ENTER>	2014	Faz a leitura do endereço de destino (DL)
ATMY1234	OK	Muda o endereço fonte (MY) do módulo .
ATMY	1234	Lê o endereço fonte (MY)
ATWR	OK	Grava as modificações efetuadas na memoria flash

Na Figura 108 pode-se ver os resultados mostrados no terminal incluso no programa *X-CTU*, todos os comandos utilizados podem ser configurados usando qualquer terminal de acesso serial.

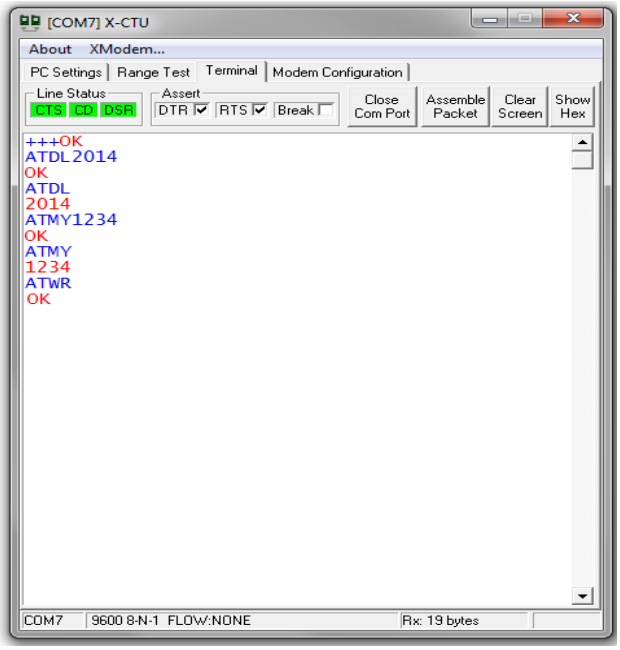


Figura 108 -Tela com resultado do módulo

Utilizando o comando **ND** que tem a função de fazer uma varredura dos módulos disponíveis na rede, observa-se na Figura 109 que ele encontrou as 02 placas de rele que está sendo controlado com a plataforma desenvolvida.

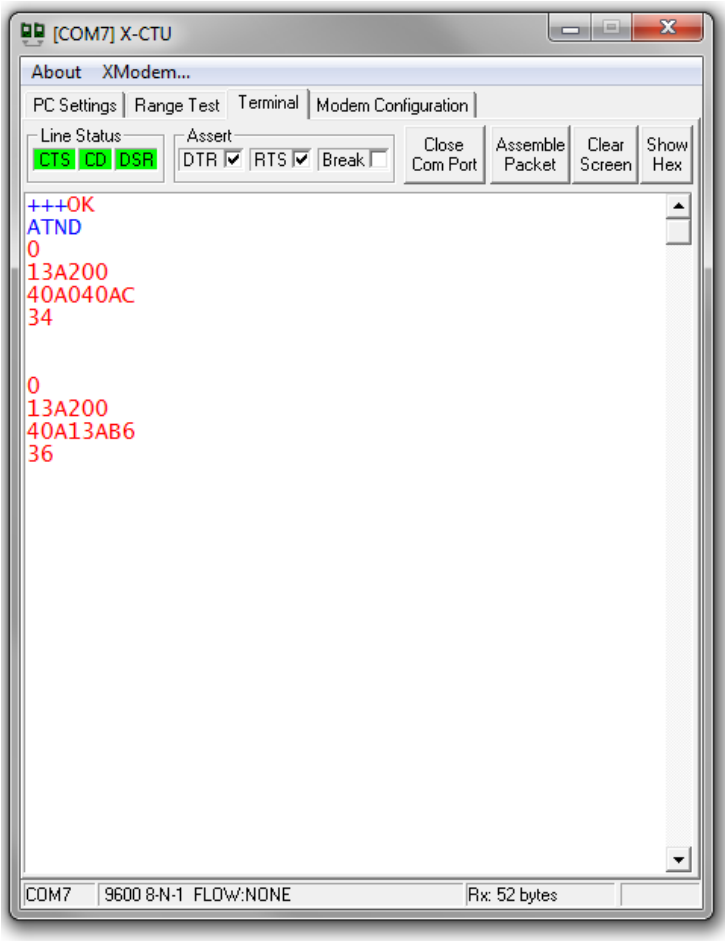


Figura 109 - Resultado comando ND

Conclusão do Apêndice **B**

Neste breve apêndice procurou-se demonstrar de uma forma simples as condições necessárias para que se configure os módulos *Xbee*. Foram utilizados os módulos *Xbee* série 1 da *Digi International,* mas é recomendado que os estudos e testes atentem para os novos módulos da Serie 2 e também aos módulos *WIFI* que usa uma conexão *Ethernet* com endereçamento *TCPIP* e com isso ter mais informações para escolher o melhor modulo *Xbee* que se adeque a aplicação que esta sendo desenvolvida.

É importante observar que caso não seja feito nenhuma configuração inicial nos módulos novos adquiridos, os mesmos já possuem uma configuração inicial que permite através da utilização de uma configuração em modo API, que as placas de reles possam ser controladas usando-se os endereços únicos constantes na parte de traz de cada modulo Xbee.

O entendimento do modo API e as funcionalidades que podem ser enviadas pelo frame configurado, são essenciais para o controle de varias placas de reles, pois cada modulo Xbee possui um endereço único, os exemplos constantes no Apêndice A não tem funcionalidade sem o entendimento deste modo de programação, pois é necessário a mudança do endereçamento bem como recalcular o *Checksum* de cada final de frame.

APÊNDICE C – APRESENTAÇÃO DA PLACA DE RELES

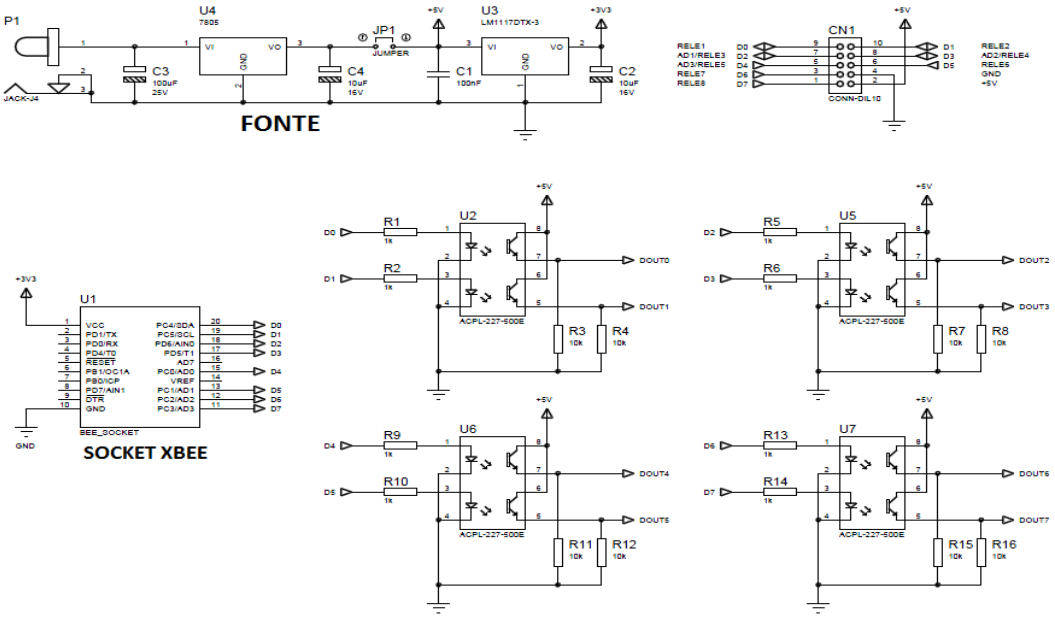
C1 - Visão **Geral**

De forma a permitir maior escalabilidade de utilização de uma aplicação de um CLP na plataforma *Graco 1.4*, houve a necessidade da criação de uma placa de controle de reles que tivessem as seguintes características :

a) Possuir Fonte de alimentação própria ;
b) Ter uma porta de comunicação por cabo com o *Plataforma Graco 1.4*;
c) 8 Reles NA/NF opto-isolados ;
d) 8 leds de identificação do sinal de cada Rele;
e) Um barramento de controle para instalação de um modulo *Xbee*;

C2 – Esquemático **da placa**

O circuito desenvolvido é demostrado na figura 123



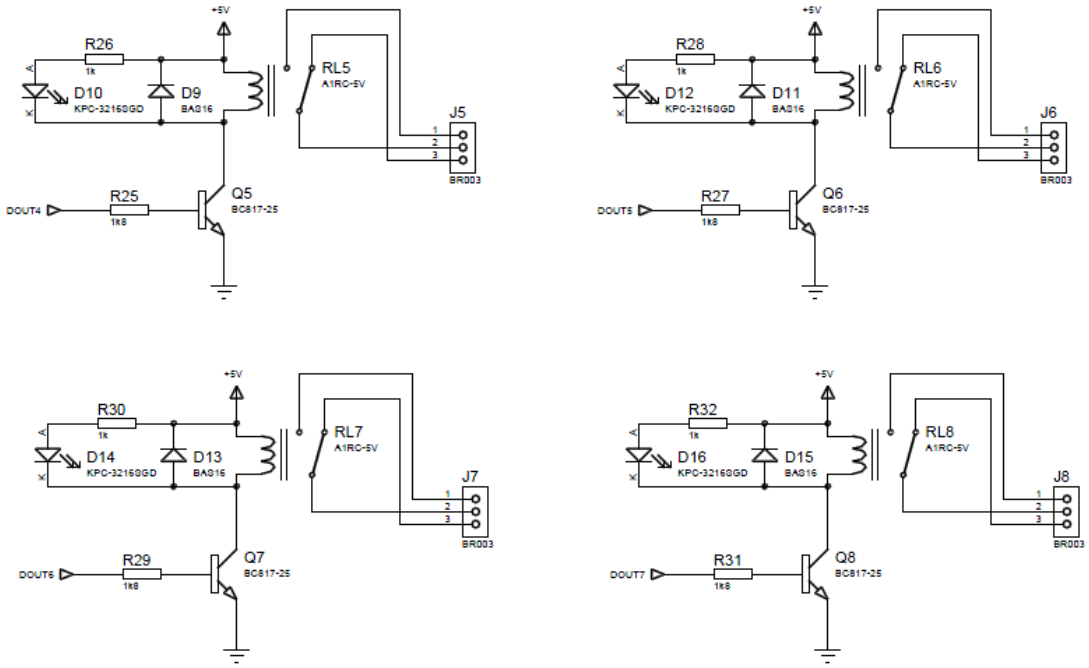


Figura 110 - Esquemático da placa de Rele com *Xbee*

C3 – Layout da placa

A seguir é apresentado a disposição dos circuitos e componentes da placa nas figuras 124,125 e 126.

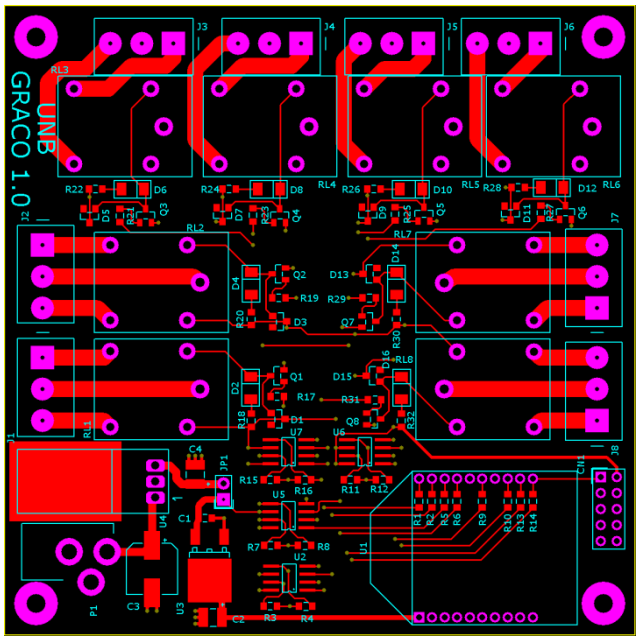


Figura 111 - *Layout* superior da Placa

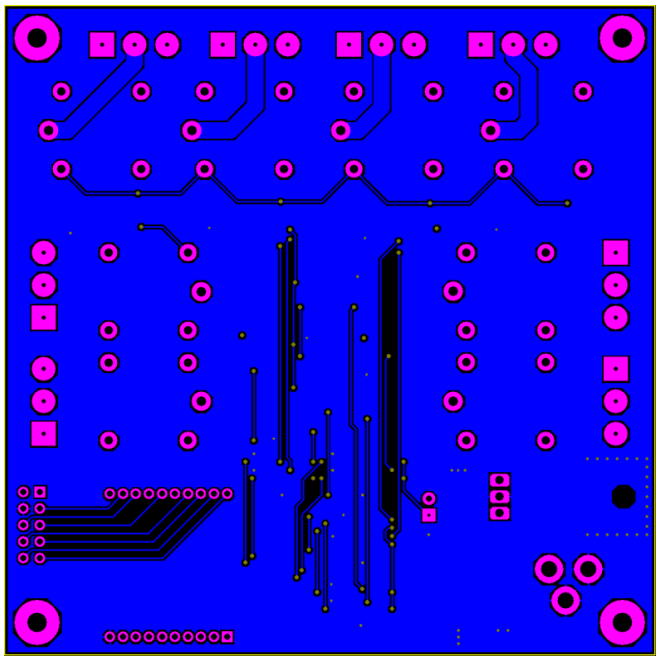


Figura 112 - *Layout* inferior da Placa

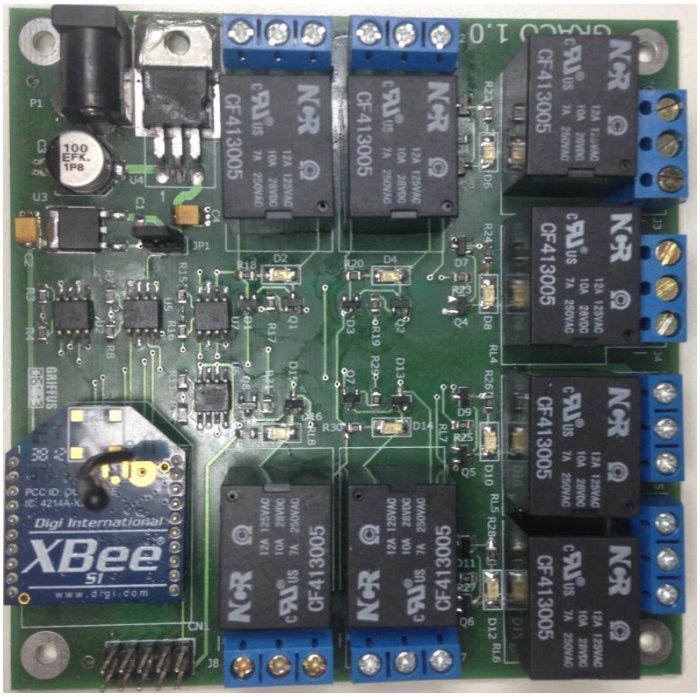


Figura 113 - Placa com componentes

C4 Circuitos da placa

Na Figura 114 é demostrado o circuito de controle do modulo *Xbee* que controla cada um dos 8 reles existentes na placa. A comunicação remota desta placa com plataforma principal é feita atraves de uma comunicação serial *Xbee* que é melhor demonstrado no Apendice B .

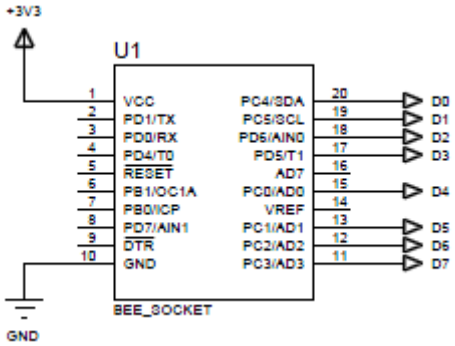


Figura 114 - Circuito *Xbee*

Na Figura 115 é demonstrado o circuito de conexão via cabo com a plataforma principal, para seu uso é necessário a devida configuração dos jumpers 3,4,6,7 e 8

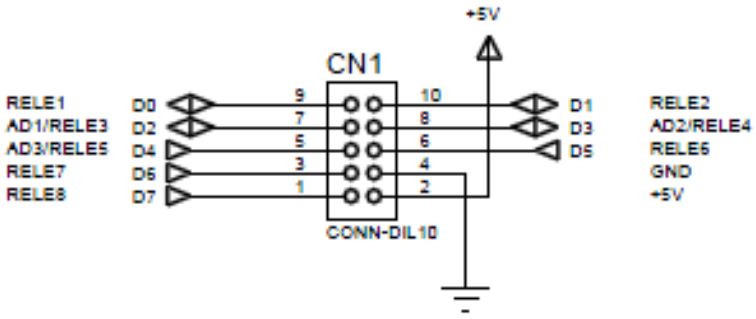


Figura 115 - Circuito

Pode-se ver na Figura 116 a ligação física da placa de reles com a plataforma principal.

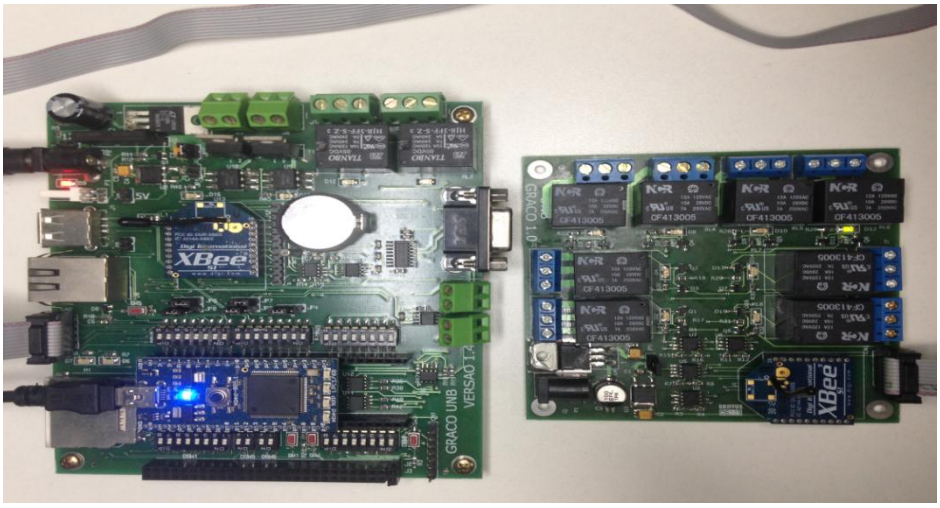


Figura 116 - Ligação da placa com cabo

C5 Teste da placa

No exemplo a seguir é demonstrada uma simulação do controle de acionamento de cargas através dos reles da placa de rele externa usando cabo de comunicação.

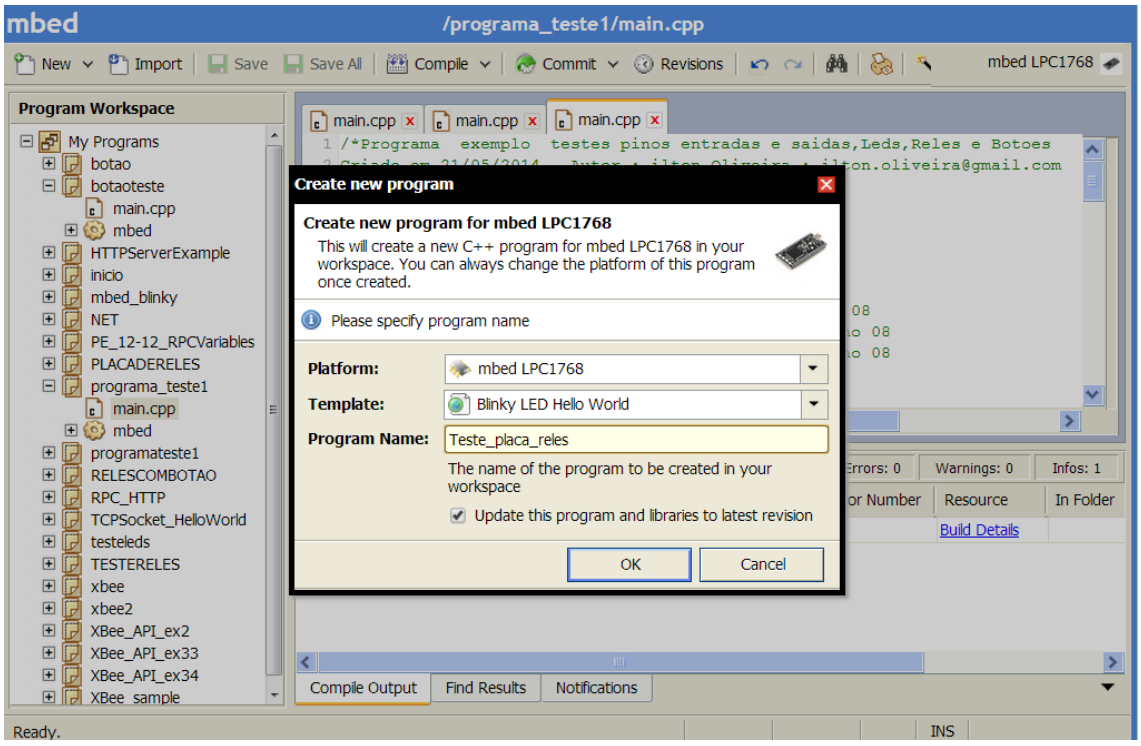


Figura 117 - Tela inicio da IDE

O Código do programa é mostrado abaixo :

```
        /* BRASILIA 09-02-2014 16H10
AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
CURSO : MESTRADO EM SISTEMAS MECATRONICOS
COMPILADOR : https://MBED.org/compiler
PROGRAMA DA PLACA  GRACO V. 1.4 - TESTE RELES 1 A 8 E BOTOES sw1,sw2 e sw3
OBS: O PROGRAMA FOI BAIXADO NO MEBED E TESTADO NA PLACA GRACO 1.4

#include "MBED.h"
DigitalOut myled(LED1); // led da placa MBED
DigitalOut rele1(p28); // rele1 ligado ao pino 22
DigitalOut rele2(p27); // rele2 ligado ao pino 22
DigitalOut rele3(p15); // rele3 ligado ao pino 22
DigitalOut rele4(p16); // rele4 ligado ao pino 22
DigitalOut rele5(p17); // rele5 ligado ao pino 22
DigitalOut rele6(p26); // rele6 ligado ao pino 21
DigitalOut rele7(p25); // rele7 ligado ao pino 22
DigitalOut rele8(p24); // rele8 ligado ao pino 22
DigitalIn botao(p8); // SW1 - botao ligado ao pino 08
DigitalIn botao2(p20); // SW2 - botao ligado ao pino 08
DigitalIn botao3(p19); // SW3 - botao ligado ao pino 08
int main() {
while(1) {
if (botao==1) { //verifica se o botao foi ativado
rele1 = 1; //rele1 é ligado
rele2 = 1; //rele1 é ligado
rele3 = 1; //rele1 é ligado
myled = 1; //liga o led da placa MBED
wait(0.2); // espera ,2 segundos
}
else { //caso o botao SW1 não esteja precionado
rele1 = 0; //rele1 é ligado
rele2 = 0; //rele1 é ligado
rele3 = 0; //rele1 é ligado
myled = 0; //liga o led da placa MBED
wait(0.2); // espera ,2 segundos

if (botao2==1) { //verifica se o botao foi ativado
rele4 = 1; //rele1 é ligado
rele5 = 1; //rele1 é ligado
rele6 = 1; //rele1 é ligado
myled = 1; //liga o led da placa MBED
wait(0.2); // espera ,2 segundos
}
else { //caso o botao SW1 não esteja precionado
rele4 = 0; //rele1 é ligado
rele5 = 0; //rele1 é ligado
rele6 = 0; //rele1 é ligado
myled = 0; //liga o led da placa MBED
wait(0.2); // espera ,2 segundos

if (botao3==1) { //verifica se o botao foi ativado
rele7 = 1; //rele1 é ligado
rele8 = 1; //rele1 é ligado
myled = 1; //liga o led da placa MBED
wait(0.2); // espera ,2 segundos
}
else { //caso o botao SW1 não esteja precionado
rele7 = 0; //rele1 é ligado
rele8 = 0; //rele1 é ligado
myled = 0; //liga o led da placa MBED
wait(0.2); // espera ,2 segundos
     }
     }
}
}
}
```

```
60
```

Listagem 9: Código programa teste placa Rele conexão com cabo

Após fazer o *download* do programa para o microcontrolador da placa, pode-se ver abaixo que foram acionados os reles 7 e 8 da placa de reles que pode ser visualizado através dos Leds dos reles ligados.

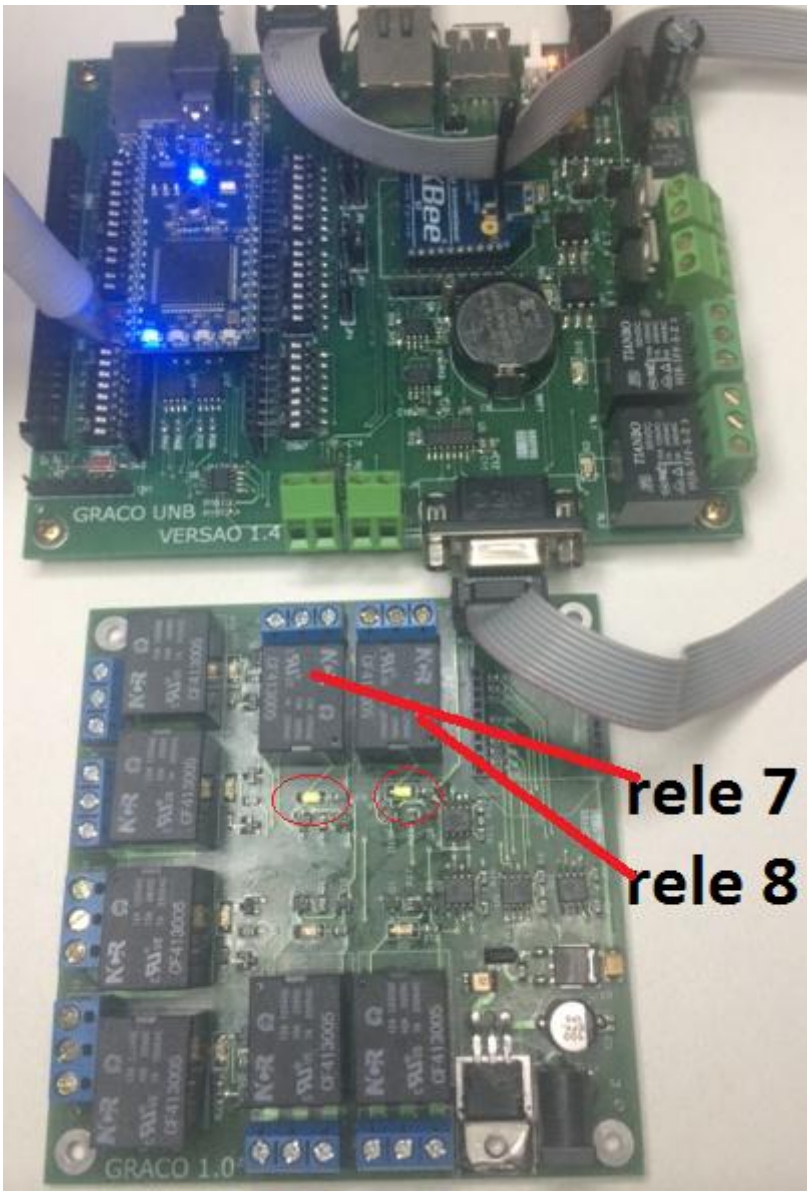


Figura 118 - Placa de rele acionada

Na listagem 10 abaixo tem-se um exemplo que demonstra o funcionamento da plataforma criada e a placa de reles com acesso remoto através de uma conexão utilizando os módulos *Xbees* instalados nas placas.

Na listagem 10 linhas 30 a 43 é enviado via serial pelos pinos 13 e 14 para o modulo *XBEE* um Frame com endereço único linhas 36 à 42 e nos bytes das linhas 71 à 73 informa que pino será usado e que estado ele deve ter ao apertar o botão SW1.

```
        /* BRASILIA 19-03-2014 13H00
 9 AUTOR : ILTON P. OLIVEIRA - ilton.oliveiradf@gmail.com
10 CURSO : MESTRADO EM SISTEMAS MECATRONICOS
11 COMPILADOR : https://MBED.org/compiler
12 PROGRAMA DA PLACA  GRACO V. 1.4 E PLACA DE RELE 1.0 - TESTE XBEE
13 */
14
15 #include "MBED.h"
16 Serial Xbee1(p13,p14); //TX E RX do MBED
17 DigitalOut myled(LED3);//Cria uma variavel para Led 3
18 DigitalIn botao(p8); // SW1 - botao ligado ao pino 08
19 DigitalIn botao2(p20); // SW2 - botao ligado ao pino 20
20
21 int main()
22 {
23     myled = 0;
24
25     while (1) {
26     if (botao==1) { //verifica se o botao foi ativado
27         myled = 1; // Led 3 Off
28         // Inicio do FRAME XBEE 1 PRIMEIRA PLACA DE RELAY
29         Xbee1.putc(0x7E);  //inicia byte API frame
30         Xbee1.putc(0x00);
31         Xbee1.putc(0x10);
32         Xbee1.putc(0x17);
33         Xbee1.putc(0x05);
34         Xbee1.putc(0x00);
35         Xbee1.putc(0x13); //XBEE Endereço
36         Xbee1.putc(0xA2); //XBEE Endereço
37         Xbee1.putc(0x00); //XBEE Endereço
38         Xbee1.putc(0x40); //XBEE Endereço
39         Xbee1.putc(0xA1); //XBEE Endereço
40         Xbee1.putc(0x3A); //XBEE Endereço
41         Xbee1.putc(0xB6); //XBEE Endereço
42         Xbee1.putc(0xFF);
43         Xbee1.putc(0xFE);
44         Xbee1.putc(0x02);
45         Xbee1.putc(0x44);     // commando ASCII caracter "D" PINO NO XBEE
46         Xbee1.putc(0x34);     // commando ASCII caracter "4" PINO NO XBEE
47         Xbee1.putc(0x05);     // D 4 HIGH ATIVA O PINO D4 EM HIGH LIGA RELAY
48         Xbee1.putc(0xE1);     //CHECKSUM
49         wait(1);
50         myled = 0; //Desliga Led 3
51         wait(1);
52         }
```

```
53          else {
54          Xbee1.putc(0x7E);         //inicia byte API frame
55          Xbee1.putc(0x00);
56          Xbee1.putc(0x10);
57          Xbee1.putc(0x17);
58          Xbee1.putc(0x05);
59          Xbee1.putc(0x00);
60          Xbee1.putc(0x13); //XBEE Endereço
61          Xbee1.putc(0xA2); //XBEE Endereço
62          Xbee1.putc(0x00); //XBEE Endereço
63          Xbee1.putc(0x40); //XBEE Endereço
64          Xbee1.putc(0xA1); //XBEE Endereço
65          Xbee1.putc(0x3A); //XBEE Endereço
66          Xbee1.putc(0xB6); //XBEE Endereço
67          Xbee1.putc(0xFF);
68          Xbee1.putc(0xFE);
69          Xbee1.putc(0x02);
70          Xbee1.putc(0x44);    // commando ASCII caracter "D" PINO NO XBEE
71          Xbee1.putc(0x34);    // commando ASCII caracter "4" PINO NO XBEE
72          Xbee1.putc(0x04);    // D 4 LOW DESATIVA O PINO D4 EM LOW DESLIGA RELAY
73          Xbee1.putc(0xE2);//CHECKSUM
74          wait(1);
75           }
76          if (botao2==1) { //verifica se o botao foi ativado
77                // Inicio do FRAME XBEE 2 SEGUNDA PLACA DE RELAY
78          Xbee1.putc(0x7E);
79          Xbee1.putc(0x00);
80          Xbee1.putc(0x10);
81          Xbee1.putc(0x17);
82          Xbee1.putc(0x05);
83          Xbee1.putc(0x00);
84          Xbee1.putc(0x13);
85          Xbee1.putc(0xA2);
86          Xbee1.putc(0x00);
87          Xbee1.putc(0x40);
88          Xbee1.putc(0xA0);
89          Xbee1.putc(0x40);
90          Xbee1.putc(0x37);
91          Xbee1.putc(0xFF);
92          Xbee1.putc(0xFE);
93          Xbee1.putc(0x02);
94          Xbee1.putc(0x44);
95          Xbee1.putc(0x34);
96          Xbee1.putc(0x05);
97          Xbee1.putc(0x5B);
98          wait(1);
99          }
100         else {
101         Xbee1.putc(0x7E);
102         Xbee1.putc(0x00);
103         Xbee1.putc(0x10);
104         Xbee1.putc(0x17);
105         Xbee1.putc(0x05);
106         Xbee1.putc(0x00);
107         Xbee1.putc(0x13);
108         Xbee1.putc(0xA2);
109         Xbee1.putc(0x00);
110         Xbee1.putc(0x40);
111         Xbee1.putc(0xA0);
112         Xbee1.putc(0x40);
113         Xbee1.putc(0x37);
```

```
114          Xbee1.putc(0xFF);
115          Xbee1.putc(0xFE);
116          Xbee1.putc(0x02);
117          Xbee1.putc(0x44);
118          Xbee1.putc(0x34);
119          Xbee1.putc(0x04);
             Xbee1.putc(0x5C);

   wait(1);
            }
       }
       }
```

Listagem 10: Código programa teste placa Rele *XBEE*

Na Figura 119 é demonstrado o teste na placa base e placa de rele usando os módulos *Xbee*, pode-se observar que o Led do Rele 8 esta ligado determinando que o sinal foi enviado com sucesso via serial para o *Xbee* da placa base que replicou para as placas dos reles. No exemplo apenas liga-se um rele de cada placa mas todos os 8 reles podem ser controlados apenas modificando as linhas de 71 à 73 não esquecendo de recalcular o Checksum da linha 74.

Como os módulos *Xbee* podem trabalhar com até 65 mil endereços, em tese tem-se uma grande faixa de outras placas que pode-se controlar , sendo que sempre é necessário mudar o endereço único constante em cada modulo *XBEE*.

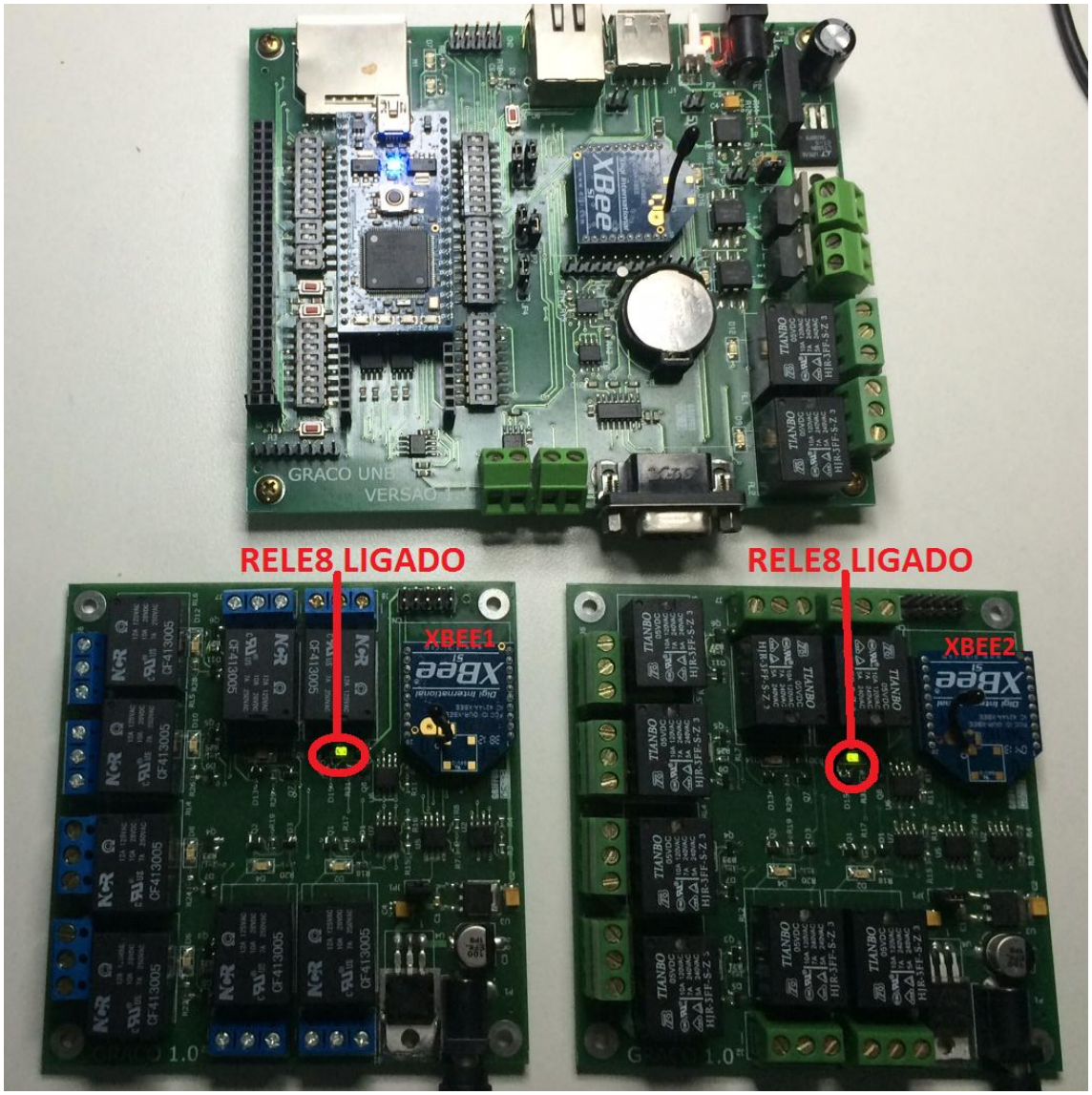


Figura 119 - Resultado Teste Placas Reles *XBEE*

Conclusão sobre a apresentação da placa de reles

Esta apresentação teve o objetivo de apresentar fisicamente as funcionalidades da placa criada de forma que o usuário possa se localizar facilmente servindo de orientação para criação de novas aplicações

É importante salientar que para o uso dos módulos *Xbee* é necessário o entendimento do apêndice B que trata especificamente deste tema.