MEDIDOR DE CONSUMO DE ENERGIA ELÉTRICA UTILIZANDO REDE EM MALHA SEM FIO

Felippe Kern Noel

Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Carlos José Ribas d'Avila

Rio de Janeiro

Abril de 2013

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

MEDIDOR DE CONSUMO DE ENERGIA ELÉTRICA UTILIZANDO REDE EM MALHA SEM FIO

Autor:

Felippe Kern Noel

Orientador:

Prof. Carlos José Ribas d'Avila, M. Sc.

Examinador:

Prof Gelson Vieira Mendonça, D. Sc.

Examinador:

Prof. Mauros Campello Queiroz, M. Sc.

DEL

Abril de 2013

Dedicatória

Aos meus pais, irmãs e aos meus avôs.

Agradecimentos

Aos meus pais, irmãs e avós por permitirem a conclusão dessa etapa da minha vida.

Ao professor Carlos José Ribas d'Avila por orientar esse projeto e aos professores Gelson Vieira Mendonça e Mauros Campello Queiroz por aceitarem participar da banca de avaliação.

Ao professor Joarez Bastos Monteiro por me instruir no conceito inicial da medição de energia.

Aos amigos que muito me ajudaram. Como Tiago Bitarelli Gomes, que me ajudou em diversas partes do projeto e na confecção da placa de circuito impresso. E Oliver Von Behr Kuster, que me ajudou no uso e aprendizado rápido da linguagem de programação PHP.

Aos meus amigos Maurício Féo Pereira Rivello de Carvalho e Paulo José Gonçalves Ferreira que sempre me apoiaram na conclusão deste projeto.

Ao Rafael Rodrigues Martinho e Caio Christóvão da Silva Porto por tornarem menos cansativo e muitas vezes divertido os laboratórios de eletrônica.

As pessoas que frequentaram o GECOM com quem pude passar bons momentos durante os intervalos das aulas.

Ao Fabio Nascimento de Carvalho e membros do LIOc que forneceram parte dos conhecimentos necessários para o desenvolvimento do projeto.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para obtenção do grau de Engenheiro Eletrônico e de Computação.

Medidor de consumo de energia elétrica utilizando rede em malha sem fio

Felippe Kern Noel

Abril/2013

Orientador: Carlos José Ribas d'Avila

Curso: Engenharia Eletrônica e de Computação

O controle do consumo de energia de um aparelho elétrico é importante para o crescimento de uma economia sustentável.

Este projeto tem como objetivo a criação de um sistema capaz de realizar a medição do consumo de energia elétrica de qualquer aparelho eletrônico, transmitir as medidas através de uma rede em malha e tornar disponíveis os dados, de forma simplificada. Além disso, também, possibilita o controle sobre o ponto elétrico onde está instalado.

Neste documento está descrito o funcionamento dos circuitos utilizados, o desenvolvimento do projeto e o código fonte.

Palavras-chave: Consumo de Energia, Rede em Malha.

Abstract of Undergraduate Project presented to POLI/UFRJ as partial fulfillment of the requirements for the degree of Engineer.

Energy consumption meter using wireless mesh network

Felippe Kern Noel

April/2013

Advisor: Carlos José Ribas d'Avila

Course: Electronic and Computer Engineering

The control of energy consumption of an electronic device is important for the growth of a sustainable economy.

This project has the goal of creating a system capable of performing the measurement of energy consumption of any electronic device, make their transmission through a mesh network, make available their data in a simplified way and also allow control of the point electricity where it is installed.

In this document is described the operation of the integrated circuits used, the development of electronic parts and source code for their use.

Keywords: Electric Energy Consumption, Mesh Network

Sumário

Lista de figuras

Lista de tabelas

Lista de abreviaturas

ASCII - American Standard Code for Information Interchange

CAD - Conversor analógico digital

UFRJ - Universidade Federal do Rio de Janeiro

CBPF - Centro Brasileiro de Pesquisas Físicas

IME - Instituto Militar de Engenharia

CI - Circuito integrado

CC - Corrente continua

CA - Corrente alternada

SPI - Serial Peripherical Interface Bus

I^2C - Inter-Integrated Circuit

RMS - Root mean square

Capítulo 1 - Introdução

1.1 - Conteúdos dos capítulos

O capitulo 1 é referente ao estudo do problema a ser solucionado por este projeto. Já o capitulo 2, é focado no estudo da topologia de rede e,também, no circuito integrado (CI) que será usado. No capitulo 3 são discutidos os CI's medidores de consumo de energia.

O capitulo 4 descreve o projeto e as soluções adotadas. Já no capitulo 5 inicia-se a descrição de um dos dois módulos do projeto, o módulo coordenador, ou como definido no capitulo, circuito coordenador. Após ele, no capitulo 6, o segundo módulo, chamado de circuito do sensor, têm descritas suas funções e singularidades.

O capitulo 7 destina-se a descrever sugestões de projetos futuros com o proposito de melhorá-lo no atendimento de seus objetivos.

Finalizando o texto, o capitulo 8 é responsável por mostrar os resultados e conclusões do projeto.

1.2 - Tema

Este trabalho tem como tema a construção de uma rede de sensores capazes de realizar a medição e o controle do consumo de energia em aparelhos elétricos.

1.3 - Objetivos

O objetivo principal é o desenvolvimento de um circuito eletrônico capaz de realizar medições de consumo de energia e controle. Além disso, o desenvolvimento de um sistema de sensores com uma rede em malha e o desenvolvimento de uma interface de monitoramento e controle que possa ser acessada de forma simples por um usuário. Também se procura uma solução simples e economicamente viável para que possa ser demonstrada a capacidade de sua utilização no âmbito comercial.

1.4 - Justificativa

Sensores que medem consumo de energia elétrica e circuitos que utilizam topologia de rede em malha não são novidades. Porém, a utilização dos dois em simultaneidade, junto com um sistema para controle e exibição, é uma possibilidade pouco explorada. Principalmente, por permitir a visualização e o controle dos dados, podendo até mesmo ser usado para fins comerciais.

1.5 - Delimitação

Temos como delimitação para o projeto o uso de tecnologias já existentes e o não desenvolvimento de partes como o protocolo de comunicação e o roteamento da rede em malha. Assim como não se pretende desenvolver um equipamento a ser homologado para a medição do consumo de energia elétrica.

1.6 - Descrição

O projeto da rede é constituído de forma que existam dois ou mais dispositivos, onde apenas um é responsável pelo gerenciamento (circuito coordenador) e os outros (circuitos de sensores) ficam responsáveis pelo monitoramento e controle direto dos pontos de consumo elétrico.

O circuito coordenador é responsável pelo gerenciamento dos dados, controle e tomadas de decisão as tarefas que os sensores irão executar.

O circuito coordenador possui uma unidade de processamento e um comunicador. A unidade de processamento é formada por um microcomputador com capacidade reduzida, cujo o objetivo é gerenciar o acesso do usuário às tarefas e ao cadastramento e condição de uso dos sensores. Já o comunicador é responsável pelo envio e recebimento dos dados transmitidos entre os pontos finais e a unidade de processamento. Uma explicação mais detalhada sobre o acesso dos usuários ao sistema e as configurações necessárias para a unidade de processamento serão feitos no capitulo 5 e o transmissor poderá ser encontrado, em detalhes, no capitulo 2.

Os circuitos de sensores deverão ter pelo menos uma unidade de medição de consumo, um micro controlador para realizar as configurações iniciais e um transmissor. Seu funcionamento será explicado no capitulo 6

Capítulo 2 - Comunicação em malha

2.1 - Topologia de rede em malha

Existem diversos tipos de topologias de redes de comunicação. Elas possuem vantagens e desvantagens. Neste capítulo iremos abordar os tipos mais comuns, explicar suas vantagens e desvantagens e os motivos da escolha pela topologia em malha.

Os tipos mais comuns são a topologia anel, estrela e em malha.

A topologia em anel como o próprio nome deixa entender, são dispositivos conectados em série e com formato de um anel, conforme mostrado na figura 1. Um bom exemplo desse tipo de topologia é parte da estrutura da Rede Rio, que fornece internet para várias instituições no Rio de Janeiro, como a UFRJ, CBPF e IME. Sua vantagem está principalmente na fácil instalação e em um desempenho uniforme. Porém quando seu número de pontos cresce, perde-se confiabilidade e o atraso nas retransmissões começa a ser prejudicial, além da possibilidade de uma falha parar todo o funcionamento do sistema.

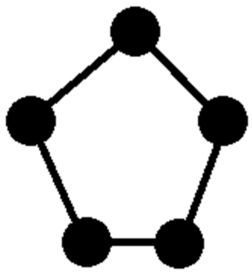

Figura 1 - Exemplo da topologia em anel

Já topologia em estrela, mostrada na figura 2, possui um ponto central que se encarrega de retransmitir para os pontos corretos todas as informações que chegam até ele. Um exemplo seria um “switch” que pode ser encontrado em residências, conectado a um roteador. A vantagem dessa topologia é a robustez, pois caso tenha alguma falha, apenas uma parte da rede irá ficar inoperante. A desvantagem é que quanto mais pontos conectados ao concentrador, maior a capacidade exigida para realizar o gerenciamento do fluxo, além do custo de instalação mais elevado devido à utilização de mais cabeamento.

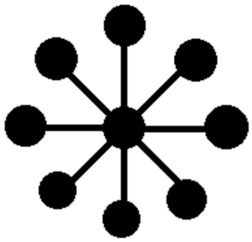

Figura 2 - Exemplo da topologia em estrela

A topologia em malha é uma abordagem próxima da topologia em estrela. Porém, ela pode ser do tipo totalmente conectada quando qualquer ponto pode se comunicar com todos os outros pontos sem necessidade de um concentrador repetir a informação, como visto na figura 3. Ou, ainda, do tipo parcialmente conectado, como pode ser visto na figura 4, quando somente alguns pontos podem se comunicar com os outros, e alguns pontos precisam de uma repetição para chegar ao seu destino. Um exemplo dessa malha é a Internet. As suas vantagens principais são conseguir conectar grandes áreas de forma simples e possuir redundância que gera maior confiabilidade do sistema. A principal desvantagem é a elevada complexidade da rede.

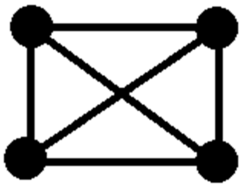

Figura 3 - Exemplo da topologia em malha totalmente conectada

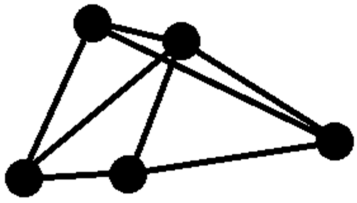

Figura 4 - Exemplo da topologia em malha parcialmente conectada

2.2 - Zigbee e concorrentes - Justificativa da escolha

Atualmente vários protocolos implementam a topologia em malha, dentre os principais estão o Zigbee, Insteon, Z-Wave e WirelessHART.

Apesar de todos utilizarem de uma topologia em malha para se comunicar, cada um possui características que os tornam únicos.

A opção por utilizar o Zigbee deve-se ao fato de que ele é facilmente encontrado no mercado na forma de XBee, uma versão do fabricante Digi International que possui um transceptor e um micro controlador para implementar o protocolo próprio da Digi International [1].

Esse protocolo é amplamente divulgado e facilmente se encontra material sobre ele. Uma dessas referências é o livro *Building Wireless Sensor Networks* [2] o qual possui explicações das diversas funcionalidades que o protocolo do XBee pode realizar.

2.3 - Funcionamento da comunicação com XBee

O XBee possui duas formas de operação., Uma é chamada de *AT*, um modo de operação considerado como uma forma simples de comunicação, onde o usuário informa ao sistema a configuração mandando uma sequência de bytes. E o outro é o modo API, onde é passada uma sequência de bytes que o XBee interpreta e dá o destino correto.

2.3.1 - Explicação simples do funcionamento do modo *AT*

O modo AT possui duas formas de comunicação.Uma é o modo "transparente", apenas para transmitir os dados, abstraindo a camada do XBee, ou seja, se for enviado um byte '0x10', o dispositivo irá transmitir o byte.

Já o modo de "comando", serve para configurar os parâmetros do XBee e será visto na seção 2.3.2.

2.3.2 - Configuração através do modo AT

Para configurar é necessário que se envie “+++”, isto é, o hexadecimal referente ao “+” na tabela ASCII e esperar o recebimento de “OK”, que significa que o XBee está pronto para ter suas funções alteradas. Após isso, para configurar o identificador, deve-se enviar “ATIDnnnn”, onde “nnnn” é o numero do identificador da rede. Como exemplo, foi usado o “2011”.

2.3.3 - Funcionamento através da API

O modo AT tem vantagens, pois é amigável ao usuário, porém para uma rede onde se precisam enviar várias informações seguidas, ele é limitado. Para isso se usa a API do XBee. A tabela 1 serve como uma referência rápida para as explicações abaixo.

- Byte de início

A API funciona de forma simplificada, pois para se realizar alguma ação no XBee, basta enviar uma sequência de bytes e esperar a ação acontecer. Essa sequência é feita através de um formato chave onde o primeiro byte é o 0x7E, e em seguida dois bytes que definem o tamanho da sequência.

- Tamanho da sequência

O tamanho da sequência é quantos bytes existem entre ele até o byte de *checksum*.

É composto por dois byes, o primeiro chamado de *MSB (most significant byte)* e o segundo *LSB (least significant byte).*

Tabela 1 - Modelo de envio dos dados para o XBee em modo API

0x7E	0x00	0x10	0x00...0x00	0xFF
Byte de inicio	MSB do tamanho	LSB do tamanho	Bytes do quadro de dados	Checksum

- Bytes do quadro de dados

O quadro de dados a ser enviado é subdividido em tipo de quadro, identificador do quadro, endereço do XBee, configuração e mensagem.

  - Tipo de quadro

O tipo de quadro possui diversas opções. Na tabela 2 definimos os que foram usados nesse projeto.

Tabela 2 - Tipos de quadros utilizados

Byte	Descrição
0x10	Transmissão de mensagem
0x17	Comando remoto
0x90	Recebimento de mensagem

  - Identificador de quadro

Somente é importante se estiver enviando um dado muito grande a ponto de não ficar dentro do tamanho permitido pela API, que é de 65025 bytes. No projeto todos os identificadores foram postos como 0x00, já que nesse projeto em nenhum momento precisaremos passar uma informação tão grande.

  - Endereço do XBee

Nesse ponto é enviado o endereço de 64 bits do XBee, normalmente encontrado na face inferior do mesmo, como mostrado na figura 5. Após isso temos a opção de enviar um endereço que é dinâmico de 16 bits. Esse endereço é atribuído ao XBee quando ele se conecta a um coordenador. Mas, se não houver essa preocupação, basta enviar os dois bytes,0xFF e 0xFE, em sequência.

Figura 5 - Parte inferior do XBee

- Mensagem e opções

Existem dois casos de passagem de bytes: mensagem e configuração.

Para a Mensagem é necessário enviar o máximo de saltos, ou seja, por quantos outros XBee a mensagem poderá passar antes de chegar ao seu destino. No caso de não se saber a localização e a rota, pode-se escolher a opção de *broadcast*, ou seja, a mensagem ira passar por quantos saltos forem necessários.

Em seguida, é enviado o byte para dizer qual tipo de mensagem: se ela possui criptografia ou se é apenas uma mensagem simples.

Finalmente, é transmitida a mensagem em bytes com o tamanho desejado. Para realizar uma mudança de configuração, é enviado o byte 0x02 para que a mudança seja realizada no XBee de destino. Para o comando AT, por exemplo, o ID seria 0x49 e 0x44. Por fim, é enviado o parâmetro do comando AT desejado.

A tabela 3 mostra um exemplo do envio de um comando que faz o XBee de destino modificar o pino D05 para nível lógico alto. A tabela 4 mostra o pacote de dados para enviar uma mensagem “teste” para o destinatário.

- *Checksum*

  Por último é enviado o *checksum*. O cálculo do *checksum* utiliza a soma de todos os bytes depois dos bytes de tamanho até o byte anterior a ele. É realizada uma operação lógica “E” com o byte 0xFF e em seguida é subtraído o byte 0xFF.

Tabela 3 - Exemplo de envio para modificação de nível lógico do pino D05 do XBee

Byte (em hexadecimal)	Descrição
7E	Byte de inicio
00	MSB do tamanho do quadro
10	LSB do tamanho do quadro
17	Tipo de quadro
00	Identificador de quadro
00 13 A2 00 40 8A 20 C9	Endereço 64-bit
FF FE	Endereço 16-bit
02	Opção do comando AT
44 35	Comando AT
05	Parâmetro do comando AT
03	Checksum

Tabela 4 - Exemplo de envio de mensagem com texto "teste"

Byte	Descrição
7E	Byte de inicio
00	MSB do tamanho do quadro
13	LSB do tamanho do quadro
10	Tipo de quadro
00	Identificador de quadro
00 13 A2 00 40 8A 20 C9	Endereço 64-bit
FF FE	Endereço 16-bit
00	Alcance do broadcast

00	Opção de envio
54 65 73 54 65	Mensagem enviada
03	Checksum

Com as sequências de bytes preparadas, basta enviar os dados através da porta serial para qualquer XBee configurado para utilizar a API e verificar a saída do XBee endereçado.

- Pacote de recebimento

Quando um pacote de dados é recebido, a estrutura é ligeiramente diferente. O campo do byte “tipo de quadro” possui o valor 0x90. Seguido do byte de identificação de quadro e dos endereços de 64 bits e 16 bits do XBee que enviou a mensagem. Os últimos bytes devem ser a mensagem enviada pela origem, sendo o último o *checksum*.

‘

Capítulo 3: Medidor de consumo

3.1 - CS5480 e concorrentes

O mercado atual, motivado pela iniciativa ambientalista e pela busca de maior eficiência tornou o conceito de *Smart Meter* muito comum e, com isso, teve inicio uma concorrência na indústria pelo melhor CI para a realização de medição do consumo de energia. Com essa concorrência, os CI's da atualidade trazem uma gama grande de recursos. Dentre os recursos mais comuns, podemos listar o cálculo automático do fator de potência, potência real, potência reativa e potência aparente, entre outros. Além dos cálculos automáticos, eles também possuem suporte a redes bifásicas e trifásicas.

Dentre os circuitos encontrados, podemos citar ADE7753 (figura 7), ADE7763, ADE7757 do fabricante *Analog Devices*, os AFE253 que vem junto com o micro controlador MSP430, da *Texas Instrument*, e os poucos conhecidos CS5490, CS5480 (figura 6) e CS5484 do fabricante *Cirrus Logic*.

Grande parte dos últimos modelos trazem melhorias que facilitam o uso, como o processamento de informações internamente e vários tipos de comunicação para atender os diversos *designs* de sistemas. Atualmente, o grande diferencial é o isolamento interno evitando, assim, a adição de novos CI's para isolar a fonte de alimentação do CI da rede elétrica. Nesses quesitos, o CS5480 foi escolhido devido à comunicação *UART* e por utilizar uma alimentação de 3,3V, que facilitou o *design*, evitando colocar outro CI para fazer a conversão entre ele e o XBee.

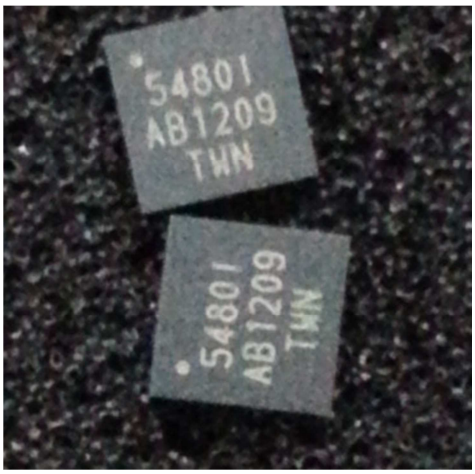

Figura 6 - CS5480

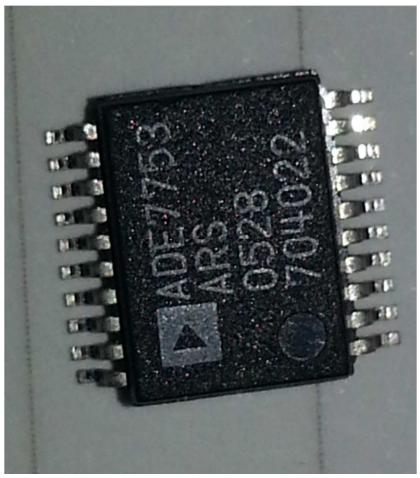

Figura 7 - ADE7753

O CS5480 possui a capacidade de medir o consumo de energia em uma rede trifásica. Apesar do CI possuir atributos e capacidade de operação em várias configurações, a utilização no projeto ficou limitada aos circuitos com 1 fase com 1 neutro.

Como podemos observar na figura 8, o circuito adquire o sinal elétrico através das entradas IN1, IN2 e VIN, aplica um ganho e utiliza os CAD's para a digitalização. Em seguida passa por 2 filtros, configuráveis. Por fim, o sinal é registrado para permitir o cálculo das variáveis derivadas da aquisição.

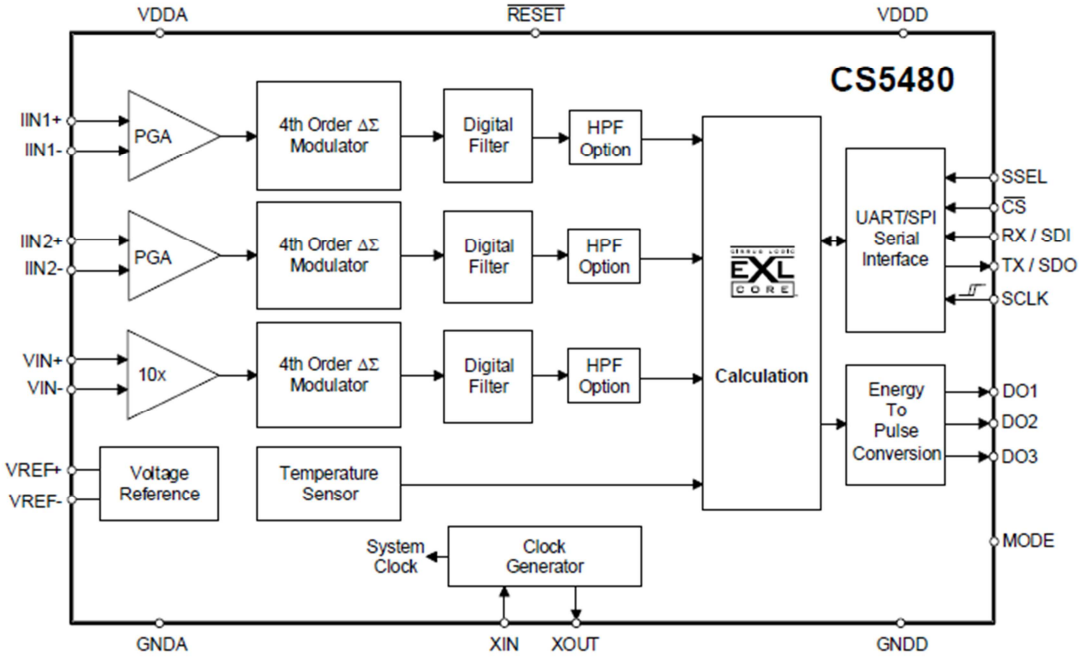

Figura 8 - Arquitetura do CS5480 [3]

3.2 - Circuitos para medir consumo

Optou-se pelo *design* com o uso de um transformador de corrente devido à sugestão do fabricante e, também, ao preço menor considerando a opção de uma bobina de Rogowski. Na figura 9 pode-se observar o circuito utilizado para medição.

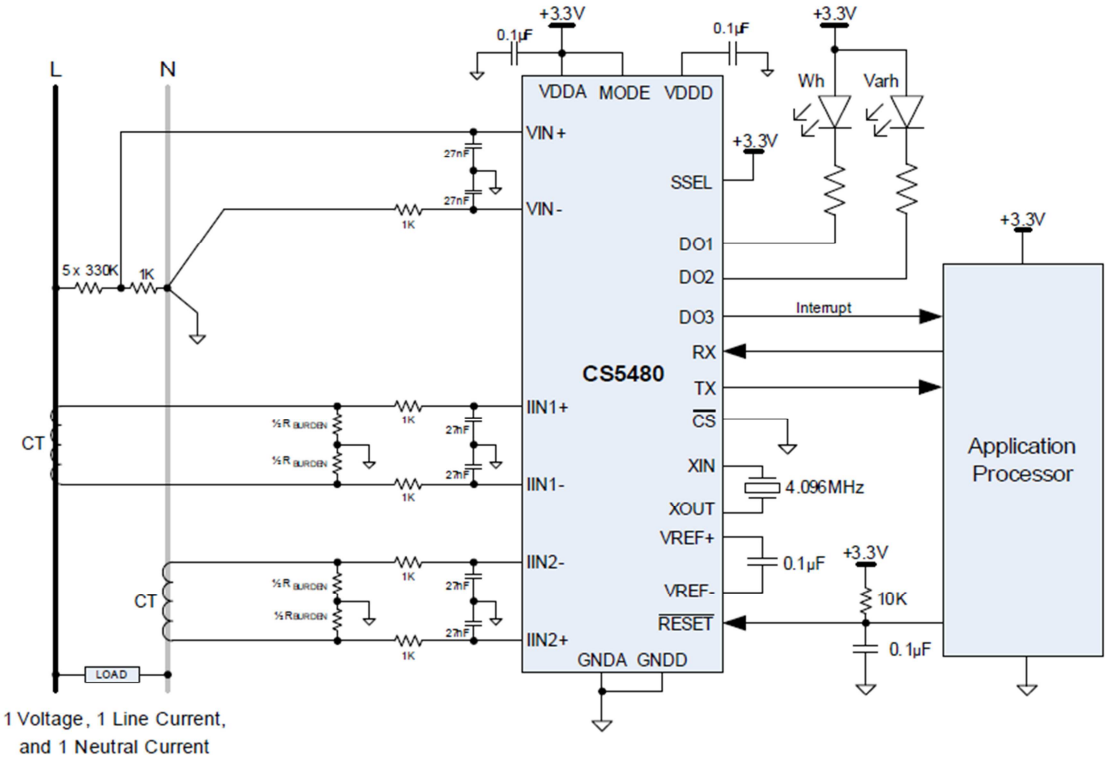

Figura 9 - Circuito da medição do consumo

3.3 - Funcionamento do CS5480

O CS5480 possui alguns detalhes importantes. A velocidade de transferência padrão é de 600 bps (bits por segundo) e, como não há memória, é necessário configurar o CI na inicialização, para uso em outra velocidade.

O CI possui quatro “páginas” de registradores, ou seja, existe um multiplexador que dá acesso aos registradores com mesmo endereço. Essas páginas são 0, 16, 17 e 18. Para acessá-las é necessário enviar seu valor binário precedido dos bits 10. Com isso, o CI começa a responder a qualquer pedido de leitura ou escrita para os registradores dentro dessas páginas.

Para realizar uma leitura em um registrador, é enviado o endereço do registrador precedido dos bits 00. Um detalhe importante é que o CI retorna o valor do registrador começando pelo byte LSB e terminando pelo byte MSB. Na tabela 5, existe a referência aos endereços dos registradores do CS5480 pertinentes ao projeto, sua descrição e valor padrão.

Tabela 5 - Registradores interessantes do CS5480 [3]

Página	Endereço em bits	Nome	Descrição	Valor padrão
00	0b00 0111	SerialCtrl	Controle UART	0x02004D
16	0b00 0100	P1	Potência instantânea 1	0x000000
16	0b00 0110	$I1_{RMS}$	Corrente RMS em I1	0x000000
16	0b00 0111	$V1_{RMS}$	Tensão RMS em V1	0x000000
16	0b10 0000	$I1_{DCOFF}$	Valor offset CC em I1	0x000000
16	0b10 0001	$I1_{Gain}$	Ganho de I1	0x400000
16	0b10 0010	$V1_{DCOFF}$	Valor offset CC em V1	0x000000
16	0b10 0011	$V1_{Gain}$	Ganho de V1	0x400000

O funcionamento do mecanismo para escrever em um registrador é simples. Toda mensagem após o byte início informa a função a ser realizada e o endereço do registrador, começando com o LSB e terminando com o MSB. Ou seja, para escrever em um registrador, são enviados os bits 01 seguidos do endereço do registrador. Após isso são enviados os 3 bytes para o preenchimento do registrador, seguindo o protocolo esperado de começar pelo LSB e terminar no MSB. A tabela 6 pode ser utilizada para uma rápida referência para entender o formato de envio de funções ao CS5480.

Tabela 6 - Formato de envio de funções ao CS5480 [3]

Função	Valor binário
Ler registrador	00 A5 A4 A3 A2 A1 A0
Escrever no registrador	01 A5 A4 A3 A2 A1 A0
Selecionar página	10 P5 P4 P3 P2 P1 P0
Instrução	00 C5 C4 C3 C2 C1 C0

3.4 - Calibração do CS5480

O CS5480 permite uma calibração para compensar a tolerância dos componentes, *offset* CC e CA nos CADs e ruídos.

O erro por offset CC acontece quando existe um valor na amostra que não pertence a ela. Esse valor aparece como um nível CC no resultado RMS da onda, pois adiciona a cada parcela da integração um valor.

Para compensar o *offset* CC, é necessário que o CI esteja sem tensão e corrente na entrada. No caso do nível CC o CI verifica o nível CC da saída e, quando comandado, modifica o valor do registrador responsável por adicionar um valor negativo, compensando o nível CC. Para realizar a calibração são enviados os comandos binários descritos na tabela 7. O CS5480 lê o valor existente na saída e preenche o registrador para compensar.

Quando acontece uma falta de sincronia entre o período medido e o período real da onda, surge o que se chama resíduo RMS. Com a integração do sinal para o cálculo do valor RMS, a falta de sincronia aparece e é chamada de *offset* CA.

Assim como a compensação do *offset* CC, para compensar o *offset* CA, é necessário que o circuito não possua nenhuma carga ou tensão na entrada e que seja enviado o comando para realizar a calibração, conforme descrito na tabela 7.

Tabela 7 - Comando para calibração

Hexadecimal	Tipo de calibração
E1	CC *Offset* do I1
F1	CA *Offset* do I1
F9	Ganho do I1
E3	CC *Offset* do I2
F3	CA *Offset* do I2
FB	Ganho do I2
E2	CC *Offset* do V1
F2	CA *Offset* do V1
FA	Ganho do V1

Na calibração de ganho, no caso da entrada $V_{1,}$ há a modificação do valor do registrador $V_{1\ Gain}$ de forma que o valor do registrador $V_{1\ RMS}$ seja igual a 0,6. E no caso de qualquer entrada de corrente, I_1 ou I_2, o registrador $I_{1,2\ Gain}$ deve possuir o mesmo valor do registrador *Scale*.

A figura 10 mostra como os valores nos registradores de calibração são utilizados no cálculo dos valores finais fornecidos pelo sistema. Esses valores são retirados da amostra antes de serem armazenados no registrador para utilização futura.

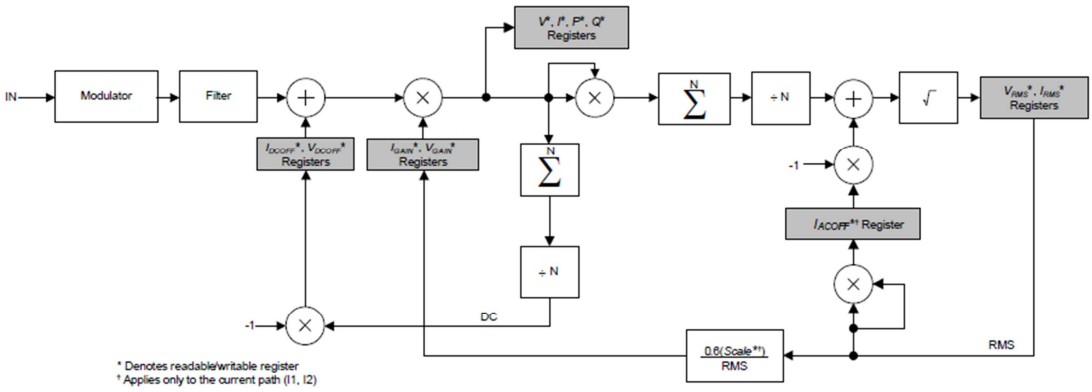

Figura 10 - Funcionamento do sistema com calibração [3]

Capítulo 4 - Detalhamento do projeto

O desenvolvimento do projeto procurou seguir as restrições descritas na seção 1.5, acrescentando a busca pelo menor preço de custo.

O material utilizado no projeto pode ser encontrado no apêndice10.

4.1 - Diagrama de blocos

O diagrama contido na figura 11 representa o funcionamento do sistema. O funcionamento do sistema é cíclico, o algoritmo implementado no arquivo controller.php, é encontrado no apêndice 1. Existe a etapa responsável por buscar novas tarefas, executar estas tarefas, guardar o resultado e atualizar os horários para as tarefas serem realizadas novamente, caso seja desejado.

A parte responsável por adicionar, editar e remover algo no sistema pode ser encontrada nos apêndices 4,5 e 6, que descrevem, respectivamente, as funcionalidades para grupos, sensores e tarefas.

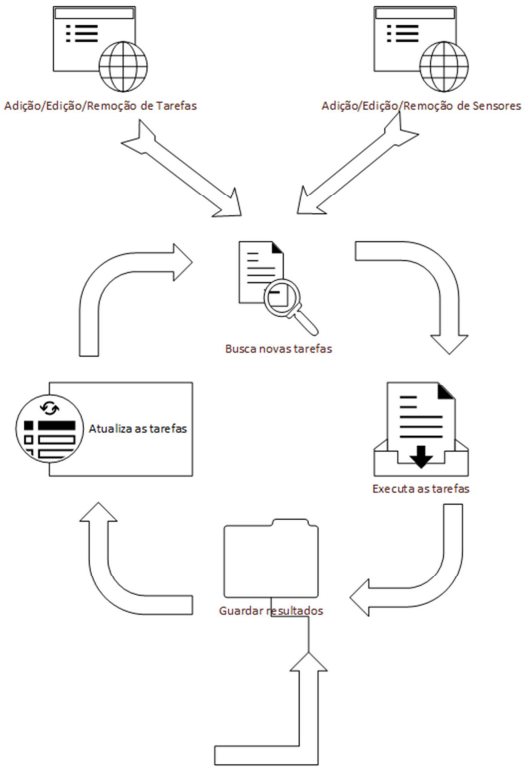

Figura 11 - Diagrama de funcionamento do sistema

No diagrama da figura 12 observa-se que o sistema consegue se comunicar com qualquer sensor que esteja mais longe que o alcance do XBee conectado no circuito de coordenação, usando outros sensores ou, apenas, outro XBee posicionado com o objetivo de aumentar o alcance da rede.

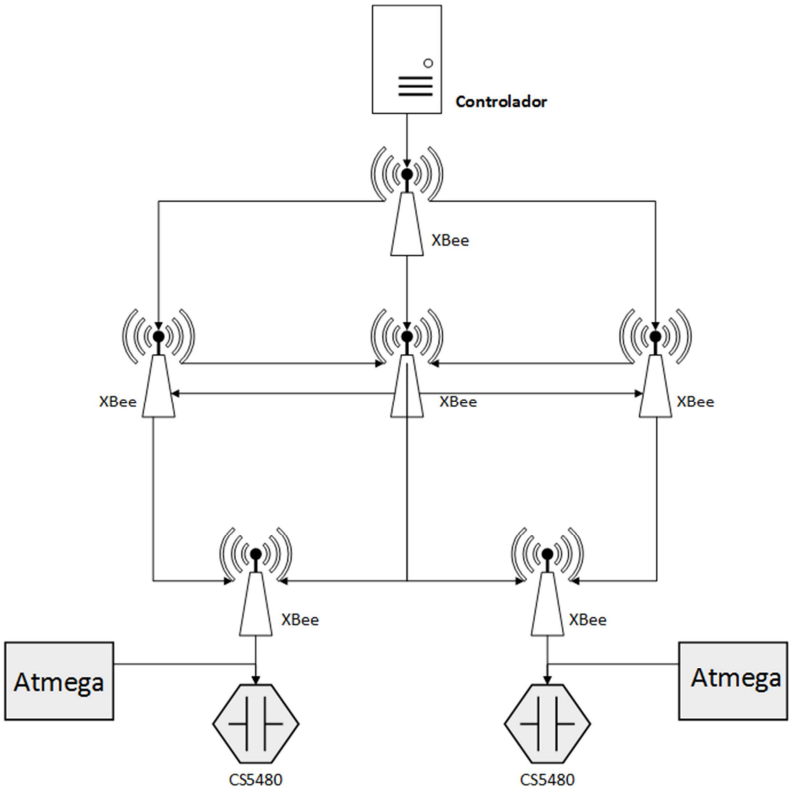

Figura 12 - Diagrama de comunicação do sistema

4.2 - Comunicações usadas internamente pelo sistema

Uma preocupação deste projeto é a utilização de tecnologias já conhecidas e desenvolvidas com o foco de reduzir os custos de implementação.

O circuito UART possibilita uma comunicação assíncrona para transferir os dados de um circuito para outro utilizando apenas um caminho. Esse tipo de comunicação é conhecido como comunicação serial e normalmente é realizada apenas entre dois pontos. Em alguns casos podem ser utilizados diversos pontos num mesmo

barramento na forma de *Master* e *Slave*, mas que permitem apenas uma comunicação por vez.

Os circuitos como o XBee, CS5480, Atmega328 eo microcomputador Raspberry Pi possuem um circuito UART, o que tornou possível a utilização em conjunto dos mesmo.

Capítulo 5 - Circuito coordenador

Definimos o circuito coordenador como o circuito que contém toda a parte de interface com o usuário e envio dos comandos para os sensores. Ou seja, nesse projeto será o microcomputador conectado ao dispositivo de transmissão e recepção.

Já o microcomputador utiliza o sistema operacional Linux e possui um servidor WEB configurado de forma a permitir o uso da linguagem e programação PHP e o banco de dados SQLite. O transceptor é um XBee Series 2 configurado de forma a utilizar a API e operando no modo “coordenador”.

5.1 - Transceptor e microcomputador

A comunicação dos dois, XBee e microcomputador, é realizada através da interface serial UART. A tabela 8 e as figuras 13 e 14 indicam a numeração dos CI’s e as ligações para a comunicação entre eles.

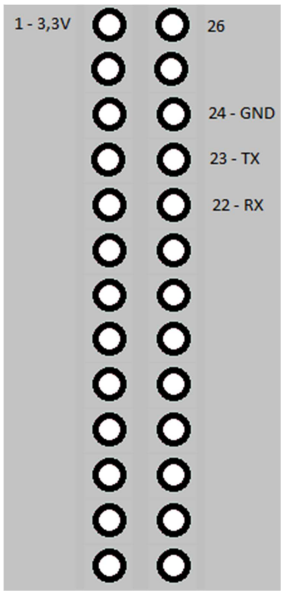

Figura 13 - Pinagem do Raspberry PI

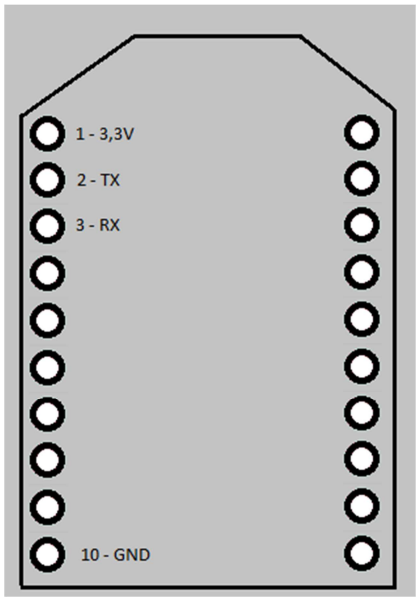

Figura 14 - Pinagem do XBee

Tabela 8 - Referência entre pinagem do Raspberry Pi e XBee

Raspberry Pi	XBee
Pino 1	Pino 1
Pino 22	Pino 2
Pino 23	Pino 3
Pino 24	Pino 10

5.2 - Configuração do XBee como coordenador

A configuração do XBee na forma de coordenador é feita utilizando o programa do fabricante X-CTU [7], como visto na figura15.

Para o funcionamento correto, é preciso que o PAN ID seja o mesmo dos outros configurados na rede.

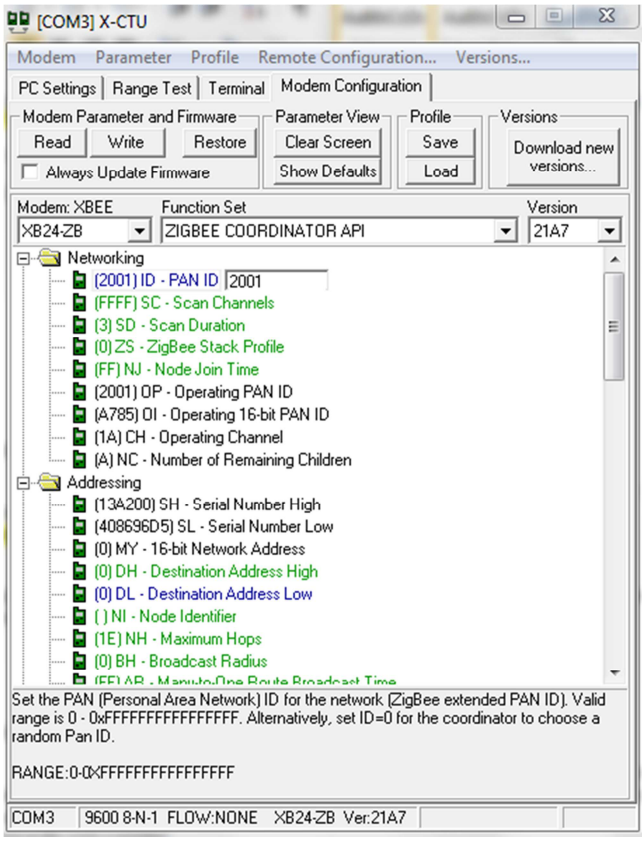

Figura 15 - Configuração do XBee como coordenador no modo API

5.3 - Interface do usuário

A interface do usuário é a forma que o usuário tem para enviar comandos ao sistema. Foi desenvolvida de forma modular, ou seja, ela pode ser substituída por outra sem afetar o funcionamento do sistema, desde que respeitada às formas de comunicação com o núcleo.

5.3.1 - Escolha do uso do PHP

Por ser uma linguagem interpretada e livre, o PHP possui uma comunidade grande e bastante suporte.

A escolha da linguagem de programação PHP foi devido a sua portabilidade pelos diversos sistemas operacionais e arquiteturas de computador. Isso torna possível que o circuito opere em quase todas as plataformas existentes, desde que respeitados os pré-requisitos da linguagem.

5.3.2 - Escolha do banco de dados SQLite e modelo usado para o sistema

Atualmente no mercado existem diversos tipos de banco de dados. A escolha do SQLite é devido a sua mobilidade, uso em sistemas embarcados e suporte à linguagem SQL.

A modelagem do banco de dados é apresentada na figura 16. O banco de dados do projeto possui seis tabelas.

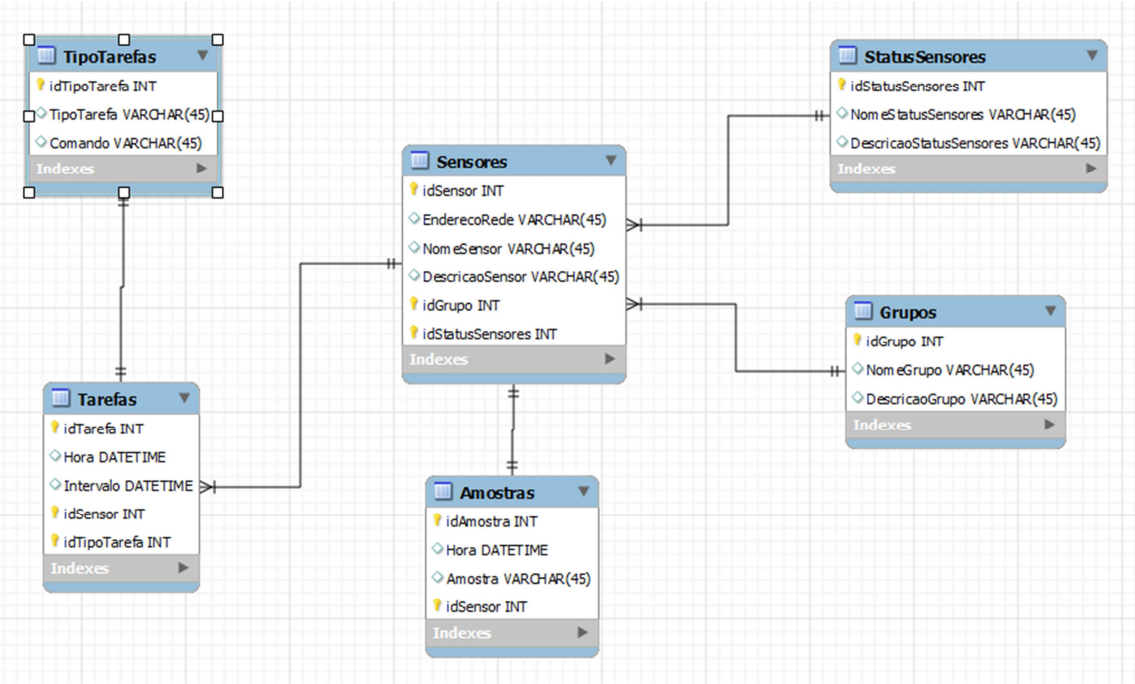

Figura 16 - Modelo do banco de dados

5.3.3 - Uso da Interface WEB

O desenvolvimento da interface foi feito para ser simples e de fácil entendimento.

Na figura 17 podemos ver a forma escolhida para trabalhar o encadeamento das necessidades do sistema. A página principal dá acesso a duas partes do sistema: a de controle de tarefas e de controle de grupo dos sensores.

Caso o usuário escolha o controle de grupo dos sensores, ele encontrará os grupos disponíveis para seleção, assim como, poderá modificar parâmetros e remover grupos. Dentro desses grupos ele tem acesso aos sensores e, assim como nos grupos, pode realizar modificações nos parâmetros e remover o sensor. O detalhe mais importante foi o uso da API do Google, *Google Chart Tools* [8], que foi utilizada para quando o usuário selecionar a exibição de amostras do sensor e quiser observar em gráfico o consumo daquele sensor, como demonstrado na figura 18.

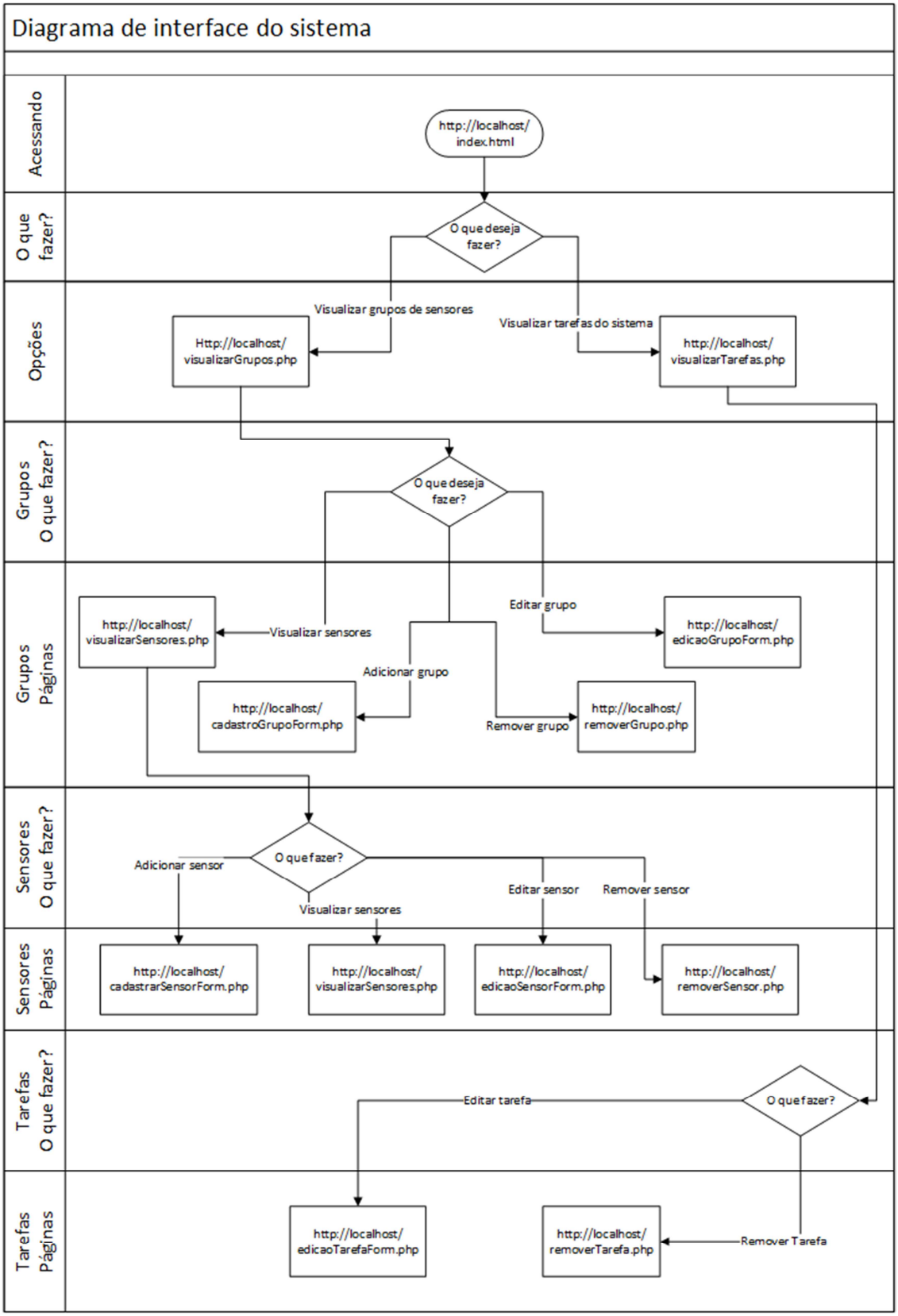

Figura 17 - Diagrama de interface do usuário

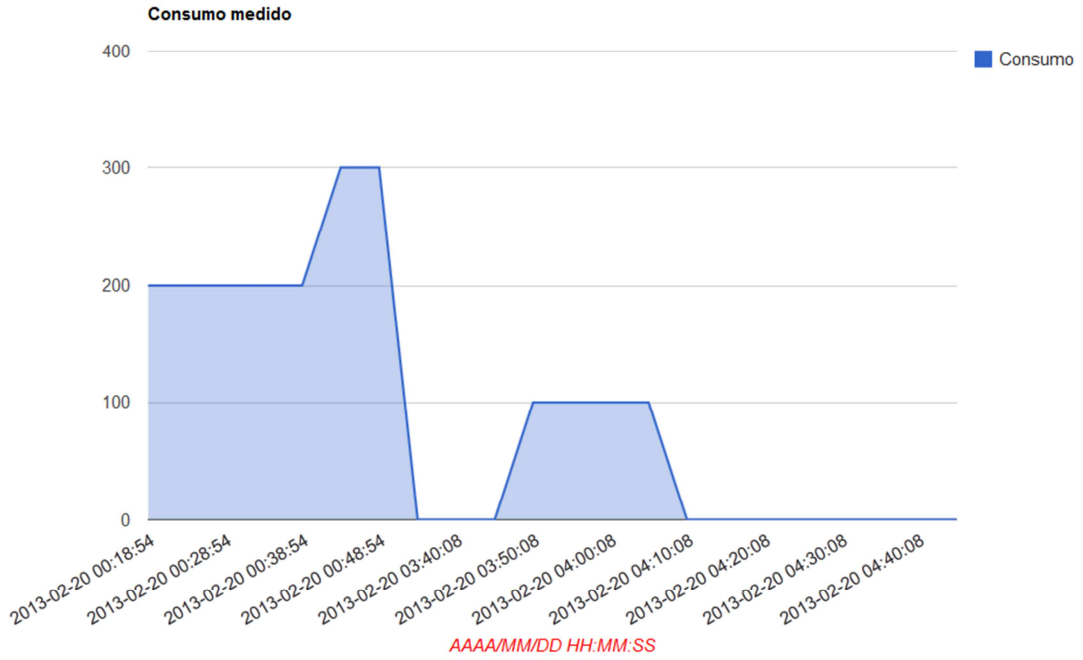

Figura 18 - Consumo mostrado em gráfico

Porém, caso o usuário escolha a visualização de tarefas, ele tem a oportunidade de verificar as atuais tarefas do sistema e tem a possibilidade de realizar alterações, ou mesmo adicionar uma nova tarefa, como mostrado na figura 19.

ID da Tarefa	Hora de inicio	Intervalo	Sensor	Tipo da Tarefa	Editar	Apagar
71	2013-02-20 22:13:47	+1 minutes	Sensor1	LigarRele	editar	apagar
72	2013-02-20 22:18:47	+2 minutes	Sensor1	DesligarRele	editar	apagar
73	2013-02-20 22:23:47	+3 minutes	Sensor1	Calibrar	editar	apagar
74	2013-02-20 22:28:47	+2 minutes	Sensor1	Amostrar	editar	apagar
75	2013-02-20 22:33:47	+5 minutes	Sensor1	Amostrar	editar	apagar

Cadastrar Tarefa

Figura 19 - Resultado do arquivo visualizarTarefas.php

5.3.4 - O controle

O controle é uma função, escrita em PHP cujo objetivo é ser executado a cada 1 minuto e buscar as tarefas daquele minuto para serem realizadas. O valor de 1

minuto foi escolhido devido ao limite mínimo de tempo imposto pelo sistema de agendamento, que é o responsável por realizar a chamada da função.

Além de buscar as tarefas, o controle também as executas., Porém, antes, é necessário montar a mensagem a ser enviada pela porta serial para o XBee. Para isso o controle chama a função *sendMessage*, que é a responsável por montar e enviar a mensagem.

Após isso, dependendo da mensagem, pode haver um retorno, como o caso da requisição de uma amostra. Nestes casos é chamada da função *readMessage*, para que ela leia e processe os dados que chegaram à porta serial armazenando-os para consultas futuras.

O sistema também foi desenvolvido para ser capaz de permitir ou não o uso de um ponto de medição. Para isso foi utilizado um relé que controla a possibilidade de conectar a rede elétrica na carga. Os detalhes desse funcionamento serão abordados no capitulo 6.

Capítulo 6 - Circuito do sensor

O circuito do sensor contém o CI medidor de consumo elétrico CS5480, o micro controlador Atmega328 e o transceptor micro controlado XBee. A forma de interação deles é descrita nas próximas subseções. A figura 20 mostra de forma simplificada a comunicação e a organização do circuito do sensor.

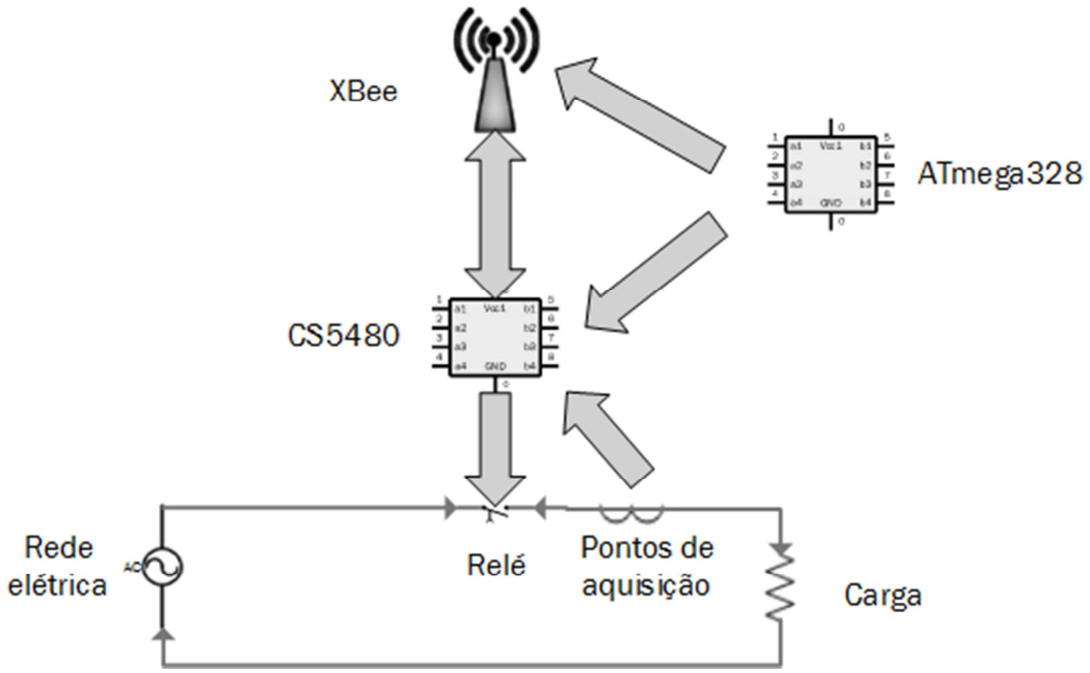

Figura 20 - Diagrama do funcionamento do circuito do sensor

6.1 - Transceptor, micro controlador e medidor de consumo.

Os dois CI's, o CS5480 e o XBee possuem um circuito UART que permite a comunicação serial. Porém, devido ao fato do CS5480 não possuir memória e, também, ter a taxa de transmissão padrão em 600 bps, que é a metade do menor valor alcançado pelo XBee, tornou-se necessário o uso de um micro controlador.

O micro controlador Atmega328, cuja pinagem pode ser encontrada na figura 21, atua no momento inicial do circuito do sensor e também quando recebe uma sinalização vinda do pino 17 do XBee. No caso de um valor lógico alto e depois zero, o micro controlador atua enviando os comandos de configuração para o CS5480.

Os comandos de configuração que são enviados pelo Atmega328 são: inicializar as conversões e as operações internas para disponibilizar os dados e configurar o CS5480 para operar com taxa de transmissão de 9600 bps. A taxa padrão do XBee, porém, pode ser escolhida dentro da faixa entre 1200 bps e 230400 bps, pois o XBee em sua concepção não foi projetado para trabalhar com 600 bps.

Um detalhe importante do projeto é o fato do XBee e o Atmega328 compartilharem a mesma entrada do CS5480. Para solucionar o problema ocasionado por dois circuitos enviando dados na mesma trilha, foi utilizada uma seleção por *Chip Select*, onde se deve informar o CI que pode enviar dados enquanto os outros ficam apenas aguardando a vez.

Foi proposta uma solução simples. Quando o sistema é ligado, a primeira atitude do Atmega328 é colocar valor lógico baixo no pino 5 do XBee. Esse pino é responsável pelo Reset do XBee e o valor lógico baixo significa que ele está desligado e, portanto, não atuante na linha serial.

Após colocar o valor lógico baixo, o Atmega328 abre comunicação serial com o CS5480 na taxa de transferência de 600 bps e envia os comandos para configurar alguns parâmetros como taxa de amostragem, ganho e para modificar a taxa de transmissão do CS5480 para 9600 bps.

Por fim, após desligar sua comunicação serial, seu ultimo ato é colocar nível lógico alto no pino 5 do XBee, permitindo assim que ele converse com o CS5480. Caso seja necessária alguma reconfiguração remota, o sistema prevê que o XBee receba informações para modificar os valor de seu pino 17 e assim disparar a tarefa do Atmega328.

Para realizar o controle sobre a permissão do consumo de energia no circuito do sensor, o circuito comunicador envia comandos para o XBee que utilizando o transistor, aciona o relé.

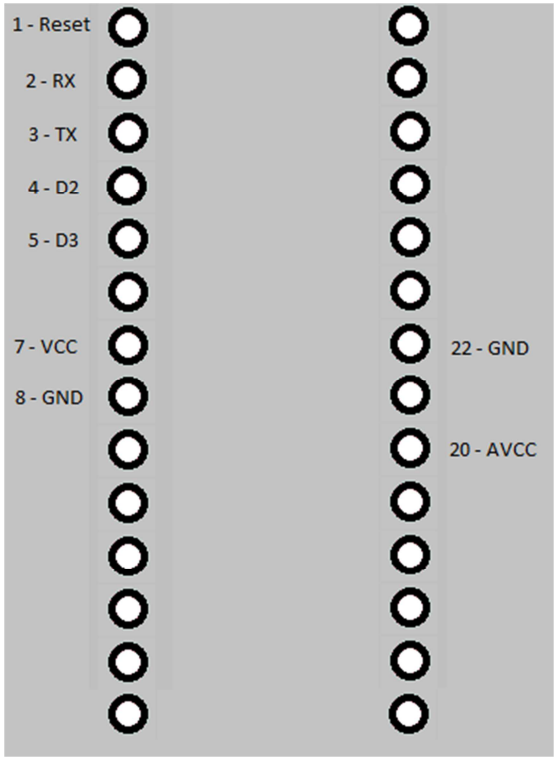

Figura 21 - Pinagem do Atmega328

6.2 - Configuração do XBee como roteador no modo AT

Utilizando o mesmo programa do capitulo 5.2, podemos configurar o XBee para operar como roteador e assim permitir o seu uso para a rede em malha. É importante a operação como roteador, pois caso seja configurado como sensor, o mesmo não repetirá o sinal que chega, ou seja, ficar como ponto final da rede.

Da mesma forma como a configuração do coordenador, mostrada na figura 22, a parte importante é a configuração do PAN ID, colocando o mesmo do coordenador, no caso, 2001. Também deixar o endereço de destino como 0, tanto campo DH quanto no DL, dessa forma todos os dados irão ser encaminhados ao coordenador.

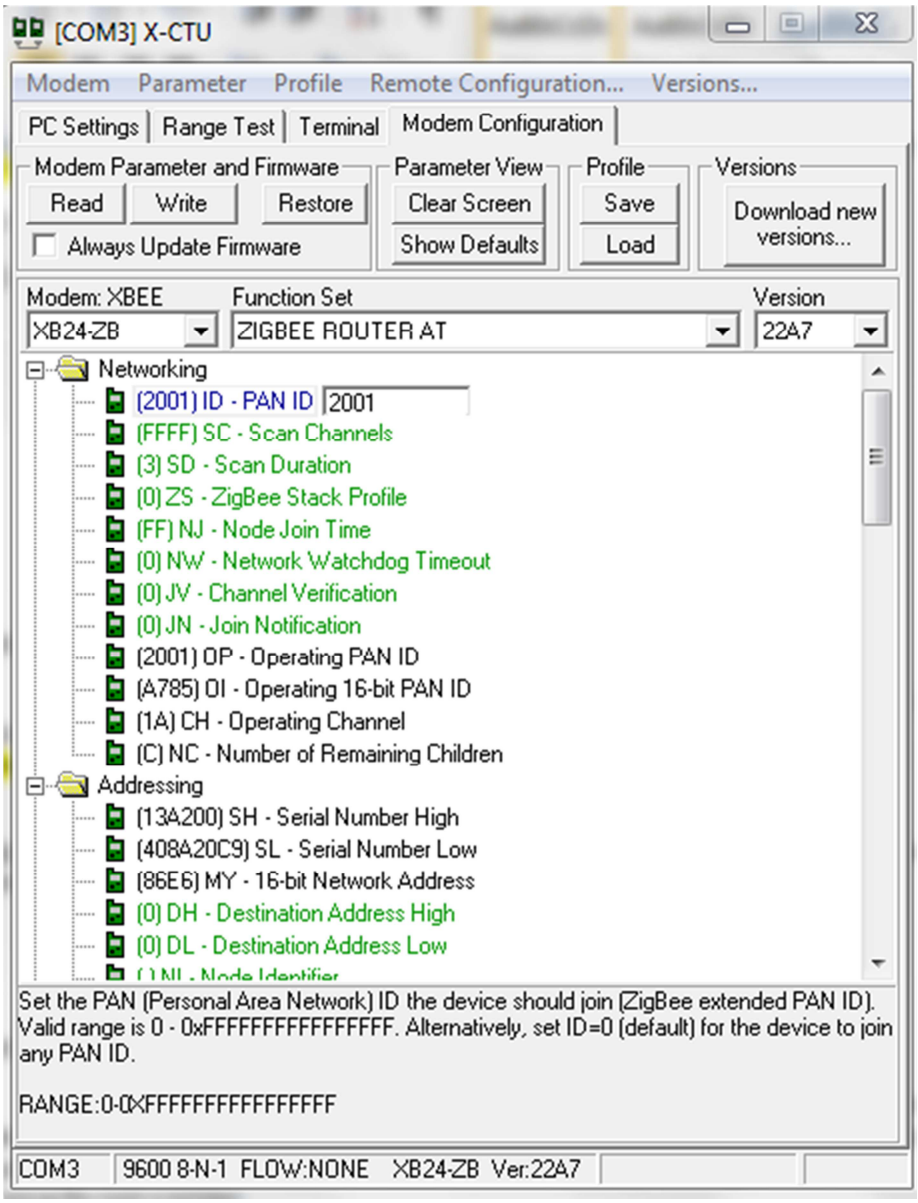

Figura 22 - Configuração do XBee como roteador no modo AT

6.2 - Configuração e calibração do sensor de consumo

Para realizar a configuração do CS5480 para trabalhar com o XBee, o Atmega328 envia a sequencia de bytes 0x47, 0xD0, 0x04 e 0x02. Isso faz com que o CS5480 comece a responder com taxa de transmissão de 9600 bps. Mas, antes de fazer a modificação de taxa de transmissão optou-se por também configurar o ganho da entrada do medidor de corrente para 50 vezes. Assim são enviados os bytes 0x80, 0x40, 0xA0, 0x20, 0xC0. Em seguida, modificamos quantas amostras o sistema faz antes de realizar os cálculos, sendo agora utilizados 16.000., Para a configuração basta

enviar 0x90, 0x73, 0x80, 0x3E, 0x00. E, finalmente, fazemos a última modificação na configuração padrão, para que o sistema faça as contas considerando que a entrada de corrente é retirada de um transformador de corrente. Nesse caso, é necessário realizar a integração dos dados e, para isto, envia-se 0x90, 0x40, 0x18, 0x02, 0x00, 0xA5.

6.3 – Calibração

Seguindo o processo de calibração descrito na seção 3.4 e o projeto do circuito, basta “desligar” a parte responsável pela alimentação da carga para realizar a calibração.

Após realizar o procedimento de calibração com as configurações informadas na seção 6.2, foi possível verificar que os valores que apareceram nos registradores para compensar os erros, estiveram sempre próximos a 1. Considerando uma faixa de valores de 0 até 16.777.215, a calibração do sistema não foi considerada mandatória.

Capítulo 7 – Resultados

Foram realizados três testes, com três cargas diferentes e utilizadas cinco amostras em cada teste.

Os valores obtidos diretamente do CS5480 foram valores convertidos pelo CAD de 24 bits, dentro de uma faixa entre 0 e 16.777.215.

O primeiro teste foi realizado com uma lâmpada de 60W, incandescente Os resultados obtidos em cada amostra foram: 64256, 65004, 64873, 65398 e 65494. A média aritmética forneceu o valor de 65005.

No segundo teste, foi utilizada uma lâmpada de 20W, fluorescente. Os resultados foram: 29659, 29884, 29885, 29590, 29551, com uma média aritmética de 29713,8.

O terceiro teste consistiu em observar o possível erro de *offset* na saída e as medidas foram: -1, -3, 1, -1, -3, com uma média aritmética de -1,4.

Verificando as proporções referentes a cada carga, 1485.69 e 1083,41, para os valores de lâmpada 20W e 60W, observa-se uma oscilação grande no valor. Como não houve tempo suficiente para testar com diversas cargas e, então, estimar a função que rege essas proporções, foi considerada uma relação linear e a média dos valores das duas cargas utilizadas. Assim, sempre que receber um valor da medição, este será dividido por 1284, 55 e guardar o resultado.

Capítulo 8 - Conclusões e trabalhos futuros

Apesar de ter sido construído com cuidado, o transformador de corrente do sensor, apresenta medições com grande variância, devido ao fio não estar passando precisamente no meio do transformador. Esse efeito pode ser minimizado, utilizando um transformado projetado industrialmente para essa finalidade.

Outro aspecto a ser destacado é o requisito de atender a um baixo custo. A utilização do transceptor XBee correspondeu a 40% do valor do circuito do sensor, impactando o custo final. Uma sugestão para projetos futuros seria o desenvolvimento de um transceptor XBee que atendesse as necessidades do projeto e que apresentasse um menor custo.

Referências bibliográficas

[1] Digi International – XBee/XBee-PRO ZB RF modules datasheet. Disponível em < http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>. Acesso em: Dezembro 2012.

[2] Faludi, R. , *Building Wireless Sensor Networks*. 1 ed. United States of America, O'Reilly Media, 2011.

[3] Cirrus Logic – ***CS5480 – Three Channel Energy Measurement IC***. Disponível em < http://www.cirrus.com/en/pubs/proDatasheet/CS5480_F2.pdf>. Acesso em: Setembro 2012.

[4] Bergen, R. van, ***Zigbee Mesh Networking***. Disponível em <http://www.scribd.com/doc/84626423/18411202-Zigbee-Mesh-Networking>. Acesso em: Outubro 2012.

[5] Cirrus Logic – ***CS5460A – Single Phase, Bi-directional Power/Energy IC***. Disponível em <http://www.cirrus.com/en/pubs/proDatasheet/CS5460A_F5.pdf>. Acesso em: Novembro 2012.

[6] Cirrus Logic – ***Application Note AN- 366***. Disponível em < http://www.cirrus.com/en/pubs/appNote/AN366REV2.pdf>. Acesso em: Novembro 2012.

[7] Digi International – ***X-CTU***. Disponível em <http://www.digi.com/support/productdetail?pid=3352&type=utilities>. Acesso em: Outubro 2012.

Apêndice 1 - Código da Interface WEB – Controle

Arquivo controller.php

```
<?php
	include "Envio.php";
	include "Recebimento.php";
	function pegarTarefas () {
		$port = fopen("/dev/ttyUSB0", "r+");
		while (!$port){
			$port = fopen("/dev/ttyUSB0", "r+");
			echo "!!!!!!!!!!!!!!!!!!!! Erro ao abrir dispositivo\n";
		}
		$intervaloExecucao = '+1 minutes';
		$dbh = new PDO('sqlite:/var/www/database.db');
		recuperarAtrasados($dbh);
		$queryTeste = "select Tarefas.idTarefa, Tarefas.Hora,
Tarefas.Intervalo, Tarefas.idSensor, Sensores.EnderecoRede,
TipoTarefas.TipoTarefa, TipoTarefas.Comando from Tarefas, Sensores, TipoTarefas
where Tarefas.idSensor = Sensores.idSensor and Tarefas.idTipoTarefa =
TipoTarefas.idTipoTarefa and (Hora between datetime('now') and datetime('now',
'$intervaloExecucao'));";
		echo "query: $queryTeste </br>\n";
		$tarefas = $dbh->query($queryTeste);
	    foreach($tarefas as $row) {
			$idTarefa = $row['idTarefa'];
```

```
                $Hora = $row['Hora'];
                $Intervalo = $row['Intervalo'];
                $idSensor = $row['idSensor'];
                $EnderecoRede = $row['EnderecoRede'];
                $TipoTarefa = $row['TipoTarefa'];
                $Comando = $row['Comando'];
                if ($Comando !="")
                        echo "Comando de dados!!!\n";
                atualizarHorario($dbh, $idTarefa, $Hora, $Intervalo);
                if ($Comando !="")
                        echo "Atualizou!!!\n";
                sendMessage($port, $EnderecoRede, $TipoTarefa, $Comando);
                $controle = 0;
                if ($Comando != "") {
                        do {
                                $saida = readMessage($port);
                                if (($saida[0] == 0) AND (bin2hex($saida[3]) ==
"90")){
                                        armazenarAmostra($dbh, $idSensor,
$Hora, bin2hex($saida[2]));
                                }
                                $controle++;
                        } while (($controle      != 2) AND ($saida[0] != 0));
```

```
                $controle = 0;

            }

            if ($TipoTarefa == "DesligarCS5480") {

                $queryTeste = "update Sensores set idStatusSensores =
2 where idSensor = $idSensor";

                $dbh->query($queryTeste);

            }

            if ($TipoTarefa == "LigarCS5480") {

                $queryTeste = "update Sensores set idStatusSensores =
1 where idSensor = $idSensor";

                $dbh->query($queryTeste);

            }

            if ($TipoTarefa == "DesligarRele") {

                $queryTeste = "update Sensores set idStatusRele = 1
where idSensor = $idSensor";

                $dbh->query($queryTeste);

            }

            if ($TipoTarefa == "LigarRele") {

                $queryTeste = "update Sensores set idStatusRele = 2
where idSensor = $idSensor";

                $dbh->query($queryTeste);

            }

        echo "FINALIZOU 1 TAREFA!!!!\n";

        }
```

```
		fclose($port);
	}

	function atualizarHorario($db, $idTarefa, $Hora, $Intervalo) {
		$query = "update Tarefas set Hora = datetime('$Hora','$Intervalo') where idTarefa=$idTarefa;";
		echo "Atualizando Tarefa: $query</br>\n";
		$db->query($query);
	}

	function armazenarAmostra($db, $idSensor, $Hora, $Amostra) {
		if ($Amostra[0] != 'f') {
		$valorTemp = sumStringHex($Amostra);
		}
		else {
			$valorTemp = 0;
		}
		echo "valorTemp: ".$valorTemp."\n";
		$valorGuardado = $valorTemp/1162;
		$query = "insert into Amostras (idAmostra, Hora, Amostra, idSensor) values (NULL, '$Hora', '$valorGuardado', $idSensor);";
		echo "Guardando amostra: $query</br>\n";
		$db->query($query);
```

```
}

function recuperarAtrasados($dbh) {
	while(1) {
		$query = "select count(Tarefas.idTarefa) from Tarefas where (Hora between datetime('0') and datetime('now')) AND (Tarefas.Intervalo <> '0');";
		$contagem = $dbh->query($query);
		if ($contagem->fetchColumn(0) == 0) {
			break;
		}
		$query = "select Tarefas.idTarefa, Tarefas.Hora, Tarefas.Intervalo from Tarefas where (Hora between datetime('0') and datetime('now')) AND (Tarefas.Intervalo <> '0');";
		$tarefas = $dbh->query($query);
		foreach($tarefas as $row) {
			$idTarefa = $row['idTarefa'];
			$Hora = $row['Hora'];
			$Intervalo = $row['Intervalo'];
			atualizarHorario($dbh, $idTarefa, $Hora, $Intervalo);
			fazerLog ($dbh, $idTarefa, 'N Realizada');
		}
	};
	echo "TEVE MENSAGEM ATRASADAA!!!!!\n";
}
```

```
        echo "\n\n##################INICIANDO####################\n\n";
        pegarTarefas();
?>
```

Apêndice 2 - Código da Interface WEB - Recebimento

Arquivo recebimento.php

```
<?php
	function tratarMensagem($msg, $size) {
		for($i=0, $j=$size; $i<=($size/2); $i++) {
			$controle = 0;
			$msgAlterada[$i] = substr($msg, $j, 2);
			$j = $j-2;
		}
		$saida = implode($msgAlterada);
		return $saida;
	}

	function readBytes($fh, $len){
		stream_set_timeout($fh,1);
		$saida = fread($fh, $len);
		return $saida;
	}

	function readMessage($port) {
	$saida; // 0->chksum OK  1->address1   2->msg    3->frameType      4->
rcv_opts
```

```
$status = 0;

$lsb = 0;

$stop = 1;

$unknownByte = 100;

echo "+Tentando ler\n";

do {

        $byte = readBytes($port, 1);

} while (ord($byte[0]) != 0x7e);

if ((ord($byte[0]) == 0x7e)){

        $stop = 0;

        $status = 1;

        $msb = ord(readBytes($port, 1));

        $lsb = ord(readBytes($port, 1));

        $size = $msb+$lsb+1;

        $offset = 0;

        $chksm = 0;

        $payload = readBytes($port,$size);

        $frametype = $payload[$offset];

        $chksm += ord($frametype);

        $offset++;

                if (ord($frametype) != 0x90){

                echo "ERRO: frametype expected 0x90, was
".bin2hex($frametype)."!\n";
```

```
}else{
        $saida[3] = $frametype;
}
$lsb--;
$addr1 = substr($payload,$offset,8);
$offset+=8;
for ($i=0;$i<8;$i++) $chksm += ord($addr1[$i]);
$saida[1] = $addr1;
$lsb-=8;
$addr2 = substr($payload,$offset,2);
$offset+=2;
for ($i=0;$i<2;$i++) $chksm += ord($addr2[$i]);
$lsb-=2;
$rcv_opts = substr($payload,$offset,1);
$offset++;
$chksm += ord($rcv_opts);
$saida[4] = $rcv_opts;
$lsb--;
$size --; //vai entender...
$msg = substr($payload,$offset,$size-$offset);
for ($i=0;$i<$size-$offset;$i++) $chksm += ord($msg[$i]);
$msgTratada = tratarMensagem (bin2hex($msg), (($size-$offset)*2));
$saida[2] = hex2bin($msgTratada);
```

```
                $offset = $size;
                $checksum = substr($payload,$offset,1);
                $offset++;
                $chksm = $chksm & 0xFF ;
                $chksm = 0xFF - $chksm;
                if($chksm !== ord($checksum)){
                        echo "ERROR: checksum differs! Got: $checksum, calculated:
$chksm\n";
                        $saida[0] = 1;
                }
                else {
                        $saida[0] = 0;
                }
        }
        $unknownByte --;
        echo "RECEBEU!!!  Saida[0]: ".$saida[0]."  Saida[1]: ".bin2hex($saida[1])."
Saida[2]: ".bin2hex($saida[2])."  Saida[3]: ".bin2hex($saida[3])."  Saida[4]:
".bin2hex($saida[4])."\n";
        return $saida;
        }
?>
```

Apêndice 3 - Código da Interface WEB – Envio

Arquivo envio.php

```
<?php
	function stringHex2ord ($texto) {
		for ($i=0, $k=0, $j=0; $i<strlen($texto); $i++) {
			$arrayNumero[$i] = ord($texto[$i]) - 48;
			if ($arrayNumero[$i] >= 10) {
				$arrayNumero[$i] -= 7;
			}
			if ($k == 0) {
				if($arrayNumero[$i] != 0)
					$arrayNumero[$i] *= 16;
				$k++;
			}
			else {
				$arrayNumero[($i-1)] += $arrayNumero[$i];
				$arrayCorreto[$j] = $arrayNumero[$i-1];
				$k--;
				$j++;
			}
		}
		return $arrayCorreto;
	}
```

```
function sumStringHex ($texto) {
	$soma = 0;
	$textoOrd = stringHex2ord($texto);
	for ($i=0; $i< strlen($texto)/2; $i+=1) {
		$soma += $textoOrd[$i];
	}
	return $soma;
}

function makeChecksum ($message) {
	$checksum = sumStringHex($message);
	$checksum = $checksum & 0xFF ;
	$checksum = 0xFF - $checksum;
	return dechex($checksum);
}

function makelength($numero) {
	$msb = 0;
	$msbM = 0;
	$msbm = 0;
	$lsb = 0;
	$lsbM = 0;
```

```
$lsbm = 0;
while ($numero > 255) {
        $msb += 1;
        $numero -255;
}
$lsb = $numero;
while (($lsb/16) >= 1) {
        $lsb = $lsb % 16;
        $lsbM ++;
}
if (($lsb/16) >= 1) {
        $lsbm = $lsb % 16;
}
else {
        $lsbm = $lsb;
}
if ($lsbm >= 10)
        $lsbm += 7;
if ($lsbM >= 10)
        $lsbM += 7;
$lsbm += 48;
$lsbM += 48;
$msbM = 0x30;
```

```
        $msbm = 0x30;


        $texto = chr($msbM).chr($msbm).chr($lsbM).chr($lsbm);
        return $texto;
}


//typeMessage: 1 - turnOn a pin-11
//                      2 - turnOff pin-11
//                      3 - turnOn a pin-17
//                      4 -  turnOff a pin-17
//                      5 - turnOn a pin-7
//                  6 - turnOff a pin-7
//                      7 - send message
function writeMessage($address, $typeMessage, $message){
        $startByte = "7E"; //Para iniciar
        $address = strtoupper($address);
        $address1 = $address;//Para fazer o endereço 1
        $address2 = "FFFE"; //Para fazer o endereço 2
        if ($typeMessage == 'D04ON') { //turnOn a pin-D4
                $frameType = "17";
                $frameId = "00";
                $remoteCommand = "02";
                $nameCommand = "4434";
```

```
                $commandParameter = "05";

                $length =
makelength((strlen($frameType)+strlen($frameId)+strlen($address1)+strlen($address
2)+strlen($remoteCommand)+strlen($nameCommand)+strlen($commandParameter))/
2);

                $payload =
sprintf("%02s%02s%02s%02s%02s%02s%02s",$frameType,$frameId,$address1,$ad
dress2,$remoteCommand,$nameCommand,$commandParameter);

                $checksum = makeChecksum($payload);

                $toSend =
sprintf("%02s%02s%02s%02s",$startByte,$length,$payload,$checksum);

        }

        if ($typeMessage == 'D04OFF') { //turnOff a pin-D4

                $frameType = "17";

                $frameId = "00";

                $remoteCommand = "02";

                $nameCommand = "4434";

                $commandParameter = "04";

                $length =
makelength((strlen($frameType)+strlen($frameId)+strlen($address1)+strlen($address
2)+strlen($remoteCommand)+strlen($nameCommand)+strlen($commandParameter))/
2);

                $payload =
sprintf("%02s%02s%02s%02s%02s%02s%02s",$frameType,$frameId,$address1,$ad
dress2,$remoteCommand,$nameCommand,$commandParameter);

                $checksum = makeChecksum($payload);
```

```
                $toSend =
sprintf("%02s%02s%02s%02s",$startByte,$length,$payload,$checksum);

        }

        if ($typeMessage == 'D03ON') { //turnOn a pin-D3

                $frameType = "17";

                $frameId = "00";

                $remoteCommand = "02";

                $nameCommand = "4433";

                $commandParameter = "05";

                $length =
makelength((strlen($frameType)+strlen($frameId)+strlen($address1)+strlen($address
2)+strlen($remoteCommand)+strlen($nameCommand)+strlen($commandParameter))/
2);

                $payload =
sprintf("%02s%02s%02s%02s%02s%02s%02s",$frameType,$frameId,$address1,$ad
dress2,$remoteCommand,$nameCommand,$commandParameter);

                $checksum = makeChecksum($payload);

                $toSend =
sprintf("%02s%02s%02s%02s",$startByte,$length,$payload,$checksum);

        }

        if ($typeMessage == 'D03OFF') { //turnOff a pin-D3

                $frameType = "17";

                $frameId = "00";

                $remoteCommand = "02";

                $nameCommand = "4432";
```

```
                $commandParameter = "04";

                $length =
makelength((strlen($frameType)+strlen($frameId)+strlen($address1)+strlen($address
2)+strlen($remoteCommand)+strlen($nameCommand)+strlen($commandParameter))/
2);

                $payload =
sprintf("%02s%02s%02s%02s%02s%02s%02s",$frameType,$frameId,$address1,$ad
dress2,$remoteCommand,$nameCommand,$commandParameter);

                $checksum = makeChecksum($payload);

                $toSend =
sprintf("%02s%02s%02s%02s",$startByte,$length,$payload,$checksum);

        }

        if ($typeMessage == 'D11ON') { //turnOn a pin-D11

                $frameType = "17";

                $frameId = "00";

                $remoteCommand = "02";

                $nameCommand = "5031";

                $commandParameter = "05";

                $length =
makelength((strlen($frameType)+strlen($frameId)+strlen($address1)+strlen($address
2)+strlen($remoteCommand)+strlen($nameCommand)+strlen($commandParameter))/
2);

                $payload =
sprintf("%02s%02s%02s%02s%02s%02s%02s",$frameType,$frameId,$address1,$ad
dress2,$remoteCommand,$nameCommand,$commandParameter);

                $checksum = makeChecksum($payload);
```

```
                $toSend =
sprintf("%02s%02s%02s%02s",$startByte,$length,$payload,$checksum);
            }
            if ($typeMessage == 'D11OFF') { //turnOff a pin-D11
                $frameType = "17";
                $frameId = "00";
                $remoteCommand = "02";
                $nameCommand = "5031";
                $commandParameter = "04";
                $length =
makelength((strlen($frameType)+strlen($frameId)+strlen($address1)+strlen($address
2)+strlen($remoteCommand)+strlen($nameCommand)+strlen($commandParameter))/
2);
                $payload =
sprintf("%02s%02s%02s%02s%02s%02s%02s",$frameType,$frameId,$address1,$ad
dress2,$remoteCommand,$nameCommand,$commandParameter);
                $checksum = makeChecksum($payload);
                $toSend =
sprintf("%02s%02s%02s%02s",$startByte,$length,$payload,$checksum);
            }
            if ($typeMessage == 'MESSAGE') { //send message
                $frameType = "10";
                $frameId = "00";
                $broadcast = "00";
                $option = "00";
```

```
                $message;

                $length =
makelength((strlen($frameType)+strlen($frameId)+strlen($address1)+strlen($address
2)+strlen($broadcast)+strlen($option)+strlen($message))/2);

                $payload =
sprintf("%02s%02s%02s%02s%02s%02s%02s",$frameType,$frameId,$address1,$ad
dress2,$broadcast,$option,$message);

                $checksum = makeChecksum($payload);

                $toSend =
sprintf("%02s%02s%02s%02s",$startByte,$length,$payload,$checksum);

        }

        return $toSend;

    }


    function sendMessage($port, $address, $typeMessage, $message) {

        if ($typeMessage == 'LigarCS5480') {

            $package = writeMessage($address,'D04ON',$message);

            $package = writeMessage($address,'D03ON',$message);

            usleep (300000);

            $package = writeMessage($address,'D03OFF',$message);

        }

        if ($typeMessage == 'DesligarCS5480') {

            $package = writeMessage($address,'D04OFF',$message);

        }
```

```
		if ($typeMessage == 'LigarRele') {
			$package = writeMessage($address,'D11ON',$message);
		}
		if ($typeMessage == 'DesligarRele') {
			$package = writeMessage($address,'D11OFF',$message);
		}
		if ($typeMessage == 'Amostrar') {
			$package = writeMessage($address,'MESSAGE',$message);
		}
		echo "ENVIANDO!!! package: $package</br>\n";
		fwrite($port, pack("H*", $package), strlen($package)/2);
	}
?>
```

Apêndice 4 - Código da Interface WEB – Grupos

Arquivo cadastroGrupoForm.php

```
 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Cadastro de Grupos</title>

</head>

<body>

<h1>Cadastro de Grupos</h1>

<form action="cadastroGrupoAcao.php" method="POST">

 Nome do Grupo <input type="text" name="nomeGrupo" />

 Descricao do Grupo <input type="text" name="descricaoGrupo" />

 <input type="submit">

</form>

</body>

</html>
```

Arquivo cadastroGrupoAcao.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Cadastro de Grupos</title>
</head>
<body>
<h1>Cadastro de Grupos</h1>
<?php
        $nomeGrupo = $_POST['nomeGrupo'];
        $descricaoGrupo = $_POST['descricaoGrupo'];

        $dbh = new PDO('sqlite:/var/www/database.db');
        $queryTeste = "insert into Grupos (NomeGrupo, DescricaoGrupo) values
('$nomeGrupo', '$descricaoGrupo');";
        echo "Query: $queryTeste\n";
        $dbh->query($queryTeste);
?>
</body>
</html>
```

Arquivo edicaoGrupoForm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Edicao de Grupo</title>
</head>
<body>
<h1>Edicao de Grupo</h1>
<?php
	$idGrupo = $_GET['idGrupo'];
	$dbh = new PDO('sqlite:/var/www/database.db');
	$queryTeste = "select * from Grupos where idGrupo = $idGrupo;";
	echo "Query: $queryTeste\n";
	$grupo = $dbh->query($queryTeste);
	foreach($grupo as $row) {
		$NomeGrupo = $row['NomeGrupo'];
		$DescricaoGrupo = $row['DescricaoGrupo'];
	}
	echo "nome: $NomeGrupo    descr: $DescricaoGrupo</br>\n";
	echo "<form action='edicaoGrupoAcao.php' method='POST'>\n";
	echo "<input type='hidden' name='idGrupo' value='$idGrupo'/></br>";
	echo "Nome do grupo <input type='text' name=nomeGrupo
value='$NomeGrupo' />\n";
	echo "Descricao: <input type='text' name=descricaoGrupo
value='$DescricaoGrupo' />\n";
```

```
        echo "<input type='submit'>\n";
        echo "</form>\n";
?>
</body>
</html>
```

Arquivo edicaoGrupoAcao.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Cadastro de sensor</title>
</head>
<body>
<h1>Cadastro de sensor</h1>
<?php
        $idGrupo = $_POST['idGrupo'];
        $nomeGrupo = $_POST['nomeGrupo'];
        $descricaoGrupo = $_POST['descricaoGrupo'];
        $dbh = new PDO('sqlite:/var/www/database.db');
        $queryTeste = "update Grupos set NomeGrupo = '$nomeGrupo',
DescricaoGrupo = '$descricaoGrupo' where idGrupo = $idGrupo;";
```

```
        echo "Query: $queryTeste\n";
        $dbh->query($queryTeste);
        echo "Grupo atualizado!\n</br>"
?>
</body>
</html>
```

Arquivo removerGrupo.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Remover grupo</title>
</head>
<body>
<h1>Remover grupo</h1>
<?php
        include "remover.php";
        $idGrupo = $_GET['idGrupo'];
        removerGrupo($idGrupo);
        echo "Grupo removido!";
?>
```

```
</body>

</html>
```

Arquivo visualizarGrupos.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Grupos</title>

</head>

<body>

<h1>Grupos</h1>

<?php

        $dbh = new PDO('sqlite:/var/www/database.db');

        $queryTeste = "select * from Grupos;";

        echo "Query: $queryTeste</br>\n";

        $result = $dbh->query($queryTeste);

        echo "<table border='1'>";

        echo "<tr>";

        echo "<th>ID do Grupo</th>";

        echo "<th>Nome do Grupo</th>";

        echo "<th>Descricao do Grupo</th>";
```

```
        echo "<th>Link de acesso</th>";
        echo "<th>Editar</th>";
        echo "<th>Remover</th>";
        echo "</tr>";
    foreach($result as $row) {
                $idGrupo = $row['idGrupo'];
                echo "<tr>";
                echo "<td>".$row['idGrupo']."</td>\n";
                echo "<td>".$row['NomeGrupo']."</td>\n";
                echo "<td>".$row['DescricaoGrupo']."</td>\n";
                echo "<td> <a
href='./visualizarSensores.php?idGrupo=$idGrupo'>acessar</a> </td>\n";
                echo "<td> <a
href='./edicaoGrupoForm.php?idGrupo=$idGrupo'>Editar</a> </td>\n";
                echo "<td> <a
href='./removerGrupo.php?idGrupo=$idGrupo'>Remover</a> </td>\n";
                echo "</tr>\n";
        }
        echo "</table>\n";
        echo "<a href='./cadastroGrupoForm.php'>Cadastrar Grupo</a></br>\n";
        echo "<a href='./cadastroSensorForm.php'>Cadastrar Sensor</a></br>\n";
        echo "<a href='./cadastroTarefaForm.php'>Cadastrar Tarefa</a></br>\n";
?>
</body>
```

```
</html>
```

Apêndice 5 - Código da Interface WEB – Sensores

Arquivo cadastroSensorForm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Cadastro de Sensores</title>

</head>

<body>

<h1>Cadastro de Sensores</h1>

<form action="cadastroSensorAcao.php" method="POST">

 Nome do sensor <input type="text" name="nomeSensor" />

 Endereco <input type='text' name="enderecoSensor" /></br>

 Descricao: <input type="text" name="descricaoSensor" />

 <?php>

        $dbh = new PDO('sqlite:/var/www/database.db');

        echo "Grupo do sensor: <select name='grupoSensor'>";

        $queryTeste = "select * from Grupos;";

        $grupoSensor = $dbh->query($queryTeste);

        foreach ($grupoSensor as $row1) {

                $grupoSensorNome = $row1['NomeGrupo'];
```

```
                    echo "<option
value='$grupoSensorNome'>$grupoSensorNome</option>";
            }
            echo "</select>";
 ?>
 <input type="submit">
</form>
</body>
</html>
```

Arquivo cadastroSensorAcao.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Cadastro de sensor</title>
</head>
<body>
<h1>Cadastro de sensor</h1>
<?php
            $nomeSensor = $_POST['nomeSensor'];
            $enderecoSensor = $_POST['enderecoSensor'];
```

```
	$descricaoSensor = $_POST['descricaoSensor'];

	$grupoSensor = $_POST['grupoSensor'];


	$dbh = new PDO('sqlite:/var/www/database.db');

	$queryTeste = "select idGrupo from Grupos where
NomeGrupo='$grupoSensor';";

	$grupoSensor = $dbh->query($queryTeste);

	foreach ($grupoSensor as $row1) {

		$idGrupo = $row1['idGrupo'];

	}


	$dbh = new PDO('sqlite:/var/www/database.db');

	$queryTeste = "insert into Sensores (NomeSensor, EnderecoRede,
DescricaoSensor, idGrupo, idStatusSensores) values ('$nomeSensor',
'$enderecoSensor', '$descricaoSensor', $idGrupo, 1);";

	echo "Query: $queryTeste\n</br>";

	$dbh->query($queryTeste);

	echo "Cadastrado!\n</br>";

?>

</body>

</html>
```

Arquivo edicaoSensorForm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Edicao de sensor</title>

</head>

<body>

<h1>Edicao de sensor</h1>

<?php

        $idSensor = $_GET['idSensor'];

        $dbh = new PDO('sqlite:/var/www/database.db');

        $queryTeste = "select * from Sensores where idSensor = $idSensor;";

        echo "Query: $queryTeste\n";

        $sensor = $dbh->query($queryTeste);

        foreach($sensor as $row) {

                $NomeSensor = $row['NomeSensor'];

                $EnderecoRede = $row['EnderecoRede'];

                $DescricaoSensor = $row['DescricaoSensor'];

                $idGrupo = $row['idGrupo'];

                echo "<form action='edicaoSensorAcao.php' method='POST'>";

                echo "<input type='hidden' name='idSensor'
value='$idSensor'/></br>";
```

```
                echo "Nome Sensor <input type='text' name='nomeSensor'
value='$NomeSensor'/></br>";

                echo "Endereco <input type='text' name='enderecoSensor'
value='$EnderecoRede'/></br>";

                echo "Descricao: <input type='text' name='descricaoSensor'
value='$DescricaoSensor'/></br>";

                echo "Grupo do sensor: <select name='grupoSensorNome'>";

                $queryTeste = "select * from Grupos;";

                echo "Query: $queryTeste\n";

                $grupoSensor = $dbh->query($queryTeste);

                foreach ($grupoSensor as $row1) {

                        $grupoSensorNome = $row1['NomeGrupo'];

                        echo "<option value='$grupoSensorNome' ";

                        if ($row1['idGrupo'] == $idGrupo)

                                echo "selected ";

                        echo ">$grupoSensorNome</option>";

                }

                echo "</select>";

                echo "<input type='submit'>";

                echo "</form>";

        }

?>

</body>

</html>
```

Arquivo edicaoSensorAcao.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Editor de sensor</title>

</head>

<body>

<h1>Editor de sensor</h1>

<?php

        $idSensor = $_POST['idSensor'];

        $nomeSensor = $_POST['nomeSensor'];

        $enderecoSensor = $_POST['enderecoSensor'];

        $descricaoSensor = $_POST['descricaoSensor'];

        $grupoSensor = $_POST['grupoSensorNome'];

        $dbh = new PDO('sqlite:/var/www/database.db');

        $queryTeste = "select idGrupo from Grupos where NomeGrupo =
'$grupoSensor';";

        echo "Query: $queryTeste\n</br>";

        $grupoSensorRetorno = $dbh->query($queryTeste);

        foreach ($grupoSensorRetorno as $row) {
```

```
                $idGrupo = $row['idGrupo'];
        }
        $dbh = new PDO('sqlite:/var/www/database.db');
        $queryTeste = "update Sensores set NomeSensor='$nomeSensor',
EnderecoRede='$enderecoSensor', DescricaoSensor='$descricaoSensor',
idGrupo=$idGrupo where idSensor = $idSensor;";
        echo "Query: $queryTeste\n</br>";
        $dbh->query($queryTeste);
        echo "Alterado!\n</br>";
?>
</body>
</html>
```

Arquivo removerSensor.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Remover sensor</title>
</head>
<body>
<h1>Remover sensor</h1>
<?php
```

```
        include "remover.php";
        $idSensor = $_GET['idSensor'];
        removerSensor($idSensor);
        echo "Sensor removido!";
?>
</body>
</html>
```

Arquivo visualizarSensores.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Sensores por grupo</title>
</head>
<body>
<h1>Sensores por grupo</h1>
<?php
        include "Envio.php";
        $idGrupo = $_GET["idGrupo"];
        $dbh = new PDO('sqlite:/var/www/database.db');
```

```
        $queryTeste = "select Sensores.idSensor, Sensores.NomeSensor,
Sensores.EnderecoRede, Sensores.DescricaoSensor,
StatusSensores.NomeStatusSensores from Sensores, StatusSensores where
Sensores.idGrupo = '$idGrupo' AND Sensores.idStatusSensores =
StatusSensores.idStatusSensores;";

        echo "Query: $queryTeste</br>\n";

        $result = $dbh->query($queryTeste);

        echo "<table border='1'>";

        echo "<tr>";

        echo "<th>ID do Sensor</th>";

        echo "<th>Nome do Sensor</th>";

        echo "<th>Endereco do Sensor</th>";

        echo "<th>Descricao do Sensor</th>";

        echo "<th>Status do Sensor</th>";

        echo "<th>Link de acesso</th>";

        echo "<th>Editar</th>";

        echo "<th>Remover</th>";

        echo "</tr>";
    foreach($result as $row) {
                echo "<tr>";

                $idSensor = $row['idSensor'];

                echo "<td>".$row['idSensor']."</td>\n";

                echo "<td>".$row['NomeSensor']."</td>\n";

                echo "<td>".$row['EnderecoRede']."</td>\n";
```

```
                echo "<td>".$row['DescricaoSensor']."</td>\n";

                echo "<td>".$row['NomeStatusSensores']."</td>\n";

                echo "<td><a
href='./selecionarAmostraForm.php?idSensor=$idSensor'>acessar</a></td>\n";

                echo "<td><a
href='./edicaoSensorForm.php?idSensor=$idSensor'>Editar</a></td>\n";

                echo "<td><a
href='./removerSensor.php?idSensor=$idSensor'>Remover</a></td>\n";

                echo "</tr>\n";

        }

        echo "</table>\n";

        echo "<a href='./cadastroGrupoForm.php'>Cadastrar Grupo</a></br>\n";

        echo "<a href='./cadastroSensorForm.php'>Cadastrar Sensor</a></br>\n";

        echo "<a href='./cadastroTarefaForm.php'>Cadastrar Tarefa</a></br>\n";

?>

</body>

</html>
```

Apêndice 6 - Código da Interface WEB – Tarefas

Arquivo cadastroTarefaForm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Cadastro de Tarefas</title>

</head>

<body>

<h1>Cadastro de Tarefas</h1>

<?php

        echo "<form action='cadastroTarefaAcao.php' method='POST'>";

        echo "Hora de inicio <input type='text' name='horaTarefa' value='AAAA-
MM-DD HH:MM:SS'/></br>";

        echo "Intervalo de realizacao da Tarefa <input type='text'
name='intervaloTarefa' value='em segundos'/></br>";

        $dbh = new PDO('sqlite:/var/www/database.db');

        echo "Sensor: <select name='idSensor'>";

        $queryTeste = "select idSensor, NomeSensor from Sensores;";

        $grupoSensor = $dbh->query($queryTeste);

        foreach ($grupoSensor as $row) {

                $idSensor = $row['idSensor'];

                $NomeSensor = $row['NomeSensor'];
```

```
            echo "<option value='$idSensor'>$NomeSensor</option>";
        }
        echo "</select></br>";
        echo "Tipo da Tarefa: <select name='idTipoTarefa'>";
        $queryTeste = "select idTipoTarefa, TipoTarefa from TipoTarefas;";
        $grupoSensor = $dbh->query($queryTeste);
        foreach ($grupoSensor as $row) {
            $idTipoTarefa = $row['idTipoTarefa'];
            $TipoTarefa = $row['TipoTarefa'];
            echo "<option value='$idTipoTarefa'>$TipoTarefa</option>";
        }
        echo "</select></br>";
        echo "<input type='submit'>";
        echo "</form>";
?>
</body>
</html>
```

Arquivo cadastroTarefaAcao.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Cadastro de Tarefas</title>
</head>
<body>
<h1>Cadastro de Tarefas</h1>
<?php
        $horaTarefa = $_POST['horaTarefa'];
        $intervaloTarefa = $_POST['intervaloTarefa'];
        $idSensor = $_POST['idSensor'];
        $idTipoTarefa = $_POST['idTipoTarefa'];

        $dbh = new PDO('sqlite:/var/www/database.db');
        $queryTeste = "insert into Tarefas (Hora, Intervalo, idSensor, idTipoTarefa)
values ('$horaTarefa', '$intervaloTarefa', $idSensor, $idTipoTarefa);";
        echo "Query: $queryTeste\n";
        $dbh->query($queryTeste);
?>
</body>
</html>
```

Arquivo edicaoTarefaForm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Edicao Tarefa</title>
</head>
<body>
<h1>Edicao Tarefa</h1>
<?php
	$idTarefa = 1;
	$dbh = new PDO('sqlite:/var/www/database.db');
	$queryTeste = "select Tarefas.Hora, Tarefas.Intervalo, Sensores.idSensor, Sensores.NomeSensor, TipoTarefas.idTipoTarefa, TipoTarefas.TipoTarefa from Sensores, Tarefas, TipoTarefas where Tarefas.idTarefa = $idTarefa AND Tarefas.idSensor = Sensores.idSensor AND Tarefas.idTipoTarefa = TipoTarefas.idTipoTarefa;";
	echo "query: $queryTeste</br>\n";
	$DadosTarefa = $dbh->query($queryTeste);
	foreach ($DadosTarefa as $row) {
		$Hora = $row['Hora'];
		$Intervalo = $row['Intervalo'];
		$idSensorSelecionada = $row['idSensor'];
		$NomeSensorSelecionada = $row['NomeSensor'];
		$idTipoTarefaSelecionada = $row['idTipoTarefa'];
		$TipoTarefaSelecionada = $row['TipoTarefa'];
```

```
        }
        echo "<form action='edicaoTarefaAcao.php' method='POST'>\n";
        echo "<input type='hidden' name='idTarefa' value='$idTarefa' />\n";
        echo "Hora de inicio <input type='text' name='horaTarefa'
value='$Hora'/></br>\n";
        echo "Intervalo de realizacao da Tarefa <input type='text'
name='intervaloTarefa' value='$Intervalo'/></br>\n";
        echo "Sensor: <select name='idSensor'>\n";
        $queryTeste = "select idSensor, NomeSensor from Sensores;";
        $sensor = $dbh->query($queryTeste);
        foreach ($sensor as $row) {
                $idSensor = $row['idSensor'];
                $NomeSensor = $row['NomeSensor'];
                echo "<option value='$idSensor' ";
                if ($row['idSensor'] == $idSensorSelecionada) {
                        echo "selected ";
                }
                echo ">$NomeSensor</option>\n";
        }
        echo "</select></br>\n";
        echo "Tipo da Tarefa: <select name='idTipoTarefa'>\n";
        $queryTeste = "select idTipoTarefa, TipoTarefa from TipoTarefas;";
        $tarefa = $dbh->query($queryTeste);
```

```
        foreach ($tarefa as $row) {
                $idTipoTarefa = $row['idTipoTarefa'];
                $TipoTarefa = $row['TipoTarefa'];
                echo "<option value='$idTipoTarefa' ";
                if ($row['idTipoTarefa'] == $idTipoTarefaSelecionada) {
                        echo "selected ";
                }
                echo ">$TipoTarefa</option>\n";
        }
        echo "</select></br>\n";
        echo "<input type='submit'>\n";
        echo "</form>\n";
?>
</body>
</html>
```

Arquivo edicaoTarefaAcao.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Edicao Tarefa</title>
```

```
</head>

<body>

<h1>Edicao Tarefa</h1>

<?php

        $idTarefa = $_POST['idTarefa'];

        $Hora = $_POST['horaTarefa'];

        $Intervalo = $_POST['intervaloTarefa'];

        $idSensor = $_POST['idSensor'];

        $idTipoTarefa = $_POST['idTipoTarefa'];

        $dbh = new PDO('sqlite:/var/www/database.db');

        $queryTeste = "update Tarefas set Hora='$Hora', Intervalo='$Intervalo',
idSensor=$idSensor, idTipoTarefa=$idTipoTarefa where idTarefa = $idTarefa;";

        echo "Query: $queryTeste</br>\n";

        $dbh->query($queryTeste);

        echo "Tarefa Atualizada!</br>\n"

?>

</body>

</html>
```

Arquivo removerTarefa.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Remover tarefa</title>
</head>
<body>
<h1>Remover tarefa</h1>
<?php
        include "remover.php";
        $idTarefa = $_GET['idTarefa'];
        removerTarefaIdTarefa($idTarefa);
        echo "Tarefa removida!";
?>
</body>
</html>
```

Arquivo visualizarTarefas.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Tarefas</title>
</head>
```

```
<body>

<h1>Tarefas</h1>

<?php

        $dbh = new PDO('sqlite:/var/www/database.db');

        $queryTeste = "select Tarefas.idTarefa, Tarefas.Hora, Tarefas.Intervalo,
Sensores.NomeSensor, TipoTarefas.TipoTarefa from Tarefas, Sensores, TipoTarefas
where Tarefas.idSensor = Sensores.idSensor AND Tarefas.idTipoTarefa =
TipoTarefas.idTipoTarefa;";

        echo "Query: $queryTeste</br>\n";

        $result = $dbh->query($queryTeste);

        echo "<table border='1'>";

        echo "<tr>";

        echo "<th>ID da Tarefa</th>";

        echo "<th>Hora de inicio</th>";

        echo "<th>Intervalo</th>";

        echo "<th>Sensor</th>";

        echo "<th>Tipo da Tarefa</th>";

        echo "<th>Editar</th>";

        echo "<th>Apagar</th>";

        echo "</tr>";

   foreach($result as $row) {

                $idTarefa = $row['idTarefa'];

                echo "<tr>";

                echo "<td>".$row['idTarefa']."</td>\n";
```

```
                echo "<td>".$row['Hora']."</td>\n";

                echo "<td>".$row['Intervalo']."</td>\n";

                echo "<td>".$row['NomeSensor']."</td>\n";

                echo "<td>".$row['TipoTarefa']."</td>\n";

                echo "<td> <a
href='./edicaoTarefaForm.php?idTarefa=$idTarefa'>editar</a> </td>\n";

                echo "<td> <a
href='./removerTarefa.php?idTareda=$idTarefa'>apagar</a> </td>\n";

                echo "</tr>\n";

        }

        echo "</table>\n";

        echo "<a href='./cadastroGrupoForm.php'>Cadastrar Grupo</a></br>\n";

        echo "<a href='./cadastroSensorForm.php'>Cadastrar Sensor</a></br>\n";

        echo "<a href='./cadastroTarefaForm.php'>Cadastrar Tarefa</a></br>\n";

?>

</body>

</html>
```

Apêndice 7 - Código da Interface WEB – Amostras

Arquivo selecionarAmostraForm.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Selecionar intervalo</title>

</head>

<body>

<h1>Selecionar intervalo</h1>

<?php

        $idSensor = $_GET["idSensor"];

        echo "<form action='selecionarAmostraAcao.php' method='POST'>";

        echo "<input type='hidden' name='idSensor' value='$idSensor'/></br>";

        echo "Hora inicio <input type='text' name='horarioInicio' value='AAAA-MM-
DD HH:MM:SS'/></br>";

        echo "Hora final: <input type='text' name='horarioFinal' value='AAAA-MM-
DD HH:MM:SS'/></br>";

        echo "<input type='submit'>";

        echo "</form>";

?>

</body>

</html>
```

Arquivo selecionarAmostraAcao.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Selecionar dados</title>

   <script type="text/javascript" src="https://www.google.com/jsapi"></script>

   <script type="text/javascript">

     google.load("visualization", "1", {packages:["corechart"]});

     google.setOnLoadCallback(drawChart);

     function drawChart() {

       var data = google.visualization.arrayToDataTable([

       ['Dia e Hora', 'Consumo'],

                  <?php

                        $idSensor = $_POST['idSensor'];

                        $horarioInicio = $_POST['horarioInicio'];

                        $horarioFinal = $_POST['horarioFinal'];


                        $dbh = new PDO('sqlite:/var/www/database.db');

                        $queryTeste = "select * from Amostras where Hora between
'$horarioInicio' and '$horarioFinal' order by Hora ASC;";
```

```
                $result = $dbh->query($queryTeste);


                foreach($result as $row) {
                        $Hora = $row['Hora'];
                        $Amostra = $row['Amostra'];
                        echo "['$Hora',  $Amostra],\n\t\t";
                }
            ?>
        ]);
        var options = {
          title: 'Consumo medido',
          hAxis: {title: 'AAAA/MM/DD HH:MM:SS',  titleTextStyle: {color: 'red'}}
        };
        var chart = new
google.visualization.AreaChart(document.getElementById('chart_div'));
        chart.draw(data, options);
      }
    </script>
</head>
<body>
<h1>Selecionar dados</h1>
          <?php echo "$queryTeste\n </br>\n"; ?>
<div id="chart_div" style="width: 1100px; height: 650px;"></div>
```

```
</br></br>



<?php

        $idSensor = $_POST['idSensor'];

        $horarioInicio = $_POST['horarioInicio'];

        $horarioFinal = $_POST['horarioFinal'];


        $dbh = new PDO('sqlite:/var/www/database.db');

        $queryTeste = "select * from Amostras where Hora between '$horarioInicio'
and '$horarioFinal';";

        echo "Query: $queryTeste</br>\n";

        $result = $dbh->query($queryTeste);


    foreach($result as $row) {

                echo "idAmostra: " . $row['idAmostra'] . "</br>\n";

                echo "Hora: " . $row['Hora'] . "</br>\n";

                echo "Amostra: " . $row['Amostra'] . "</br>\n";

                echo "idSensor: " . $row['idSensor'] . "</br>\n";

                echo "\n";

        }

?>
```

```
</body>

</html>
```

Apêndice 8 – Script para criação do banco de dados

Arquivo scriptCriacao.sql

```
-- Creator:       MySQL Workbench 5.2.40/ExportSQLite plugin 2009.12.02
-- Author:        Felippe Kern Noel
-- Caption:       New Model
-- Project:       Name of the project
-- Changed:       2013-02-12 14:24
-- Created:       2013-02-03 00:31
PRAGMA foreign_keys = OFF;


-- Schema: mydb
--BEGIN;
CREATE TABLE "Grupos"(
  "idGrupo" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
  "NomeGrupo" VARCHAR(45),
  "DescricaoGrupo" VARCHAR(45)
);
CREATE TABLE "TipoTarefas"(
  "idTipoTarefa" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
  "TipoTarefa" VARCHAR(45),
  "Comando" VARCHAR(45)
);
CREATE TABLE "StatusSensores"(
```

```
  "idStatusSensores" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
  "NomeStatusSensores" VARCHAR(45),
  "DescricaoStatusSensores" VARCHAR(45)
);
CREATE TABLE "Sensores"(
  "idSensor" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
  "EnderecoRede" VARCHAR(45),
  "NomeSensor" VARCHAR(45),
  "DescricaoSensor" VARCHAR(45),
  "idGrupo" INTEGER NOT NULL,
  "idStatusSensores" INTEGER NOT NULL,
  CONSTRAINT "fk_Sensores_Grupos1"
    FOREIGN KEY("idGrupo")
    REFERENCES "Grupos"("idGrupo"),
  CONSTRAINT "fk_Sensores_StatusSensores1"
    FOREIGN KEY("idStatusSensores")
    REFERENCES "StatusSensores"("idStatusSensores")
);
CREATE INDEX "Sensores.fk_Sensores_Grupos1_idx" ON "Sensores"("idGrupo");
CREATE INDEX "Sensores.fk_Sensores_StatusSensores1" ON
"Sensores"("idStatusSensores");
CREATE TABLE "Tarefas"(
  "idTarefa" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
```

```
  "Hora" DATETIME,
  "Intervalo" VARCHAR(45),
  "idSensor" INTEGER NOT NULL,
  "idTipoTarefa" INTEGER NOT NULL,
  CONSTRAINT "fk_Tarefas_Sensores1"
    FOREIGN KEY("idSensor")
    REFERENCES "Sensores"("idSensor"),
  CONSTRAINT "fk_Tarefas_TipoTarefas1"
    FOREIGN KEY("idTipoTarefa")
    REFERENCES "TipoTarefas"("idTipoTarefa")
);
CREATE INDEX "Tarefas.fk_Tarefas_Sensores1_idx" ON "Tarefas"("idSensor");
CREATE INDEX "Tarefas.fk_Tarefas_TipoTarefas1_idx" ON
"Tarefas"("idTipoTarefa");
CREATE TABLE "Amostras"(
  "idAmostra" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
  "Hora" DATETIME,
  "Amostra" VARCHAR(45),
  "idSensor" INTEGER NOT NULL,
  CONSTRAINT "fk_Amostras_Sensores1"
    FOREIGN KEY("idSensor")
    REFERENCES "Sensores"("idSensor")
);
```

```
CREATE INDEX "Amostras.fk_Amostras_Sensores1_idx" ON
"Amostras"("idSensor");


INSERT INTO Grupos VALUES (NULL,'default', 'Default');

INSERT INTO StatusSensores VALUES (NULL,'ON', 'Ligado');

INSERT INTO StatusSensores VALUES (NULL,'OFF', 'Desligado');

INSERT INTO TipoTarefas VALUES (NULL,'LigarCS5480','LigarCS5480');

INSERT INTO TipoTarefas VALUES (NULL,'DesligarCS5480','DesligarCS5480');

INSERT INTO TipoTarefas VALUES (NULL,'LigarRele','LigarRele');

INSERT INTO TipoTarefas VALUES (NULL,'DesligarRele','DesligarRele');

INSERT INTO TipoTarefas VALUES (NULL,'Calibrar','');

INSERT INTO TipoTarefas VALUES (NULL,'Amostrar','9004');

INSERT INTO Sensores VALUES (NULL,'0013A200408A20C9', 'Sensor1',
'Sensor1', 1, 1);

INSERT INTO Sensores VALUES (NULL,'0013A200408A20C9', 'Sensor2',
'Sensor2', 1, 1);
```

Apêndice 9 – Layout do circuito do sensor

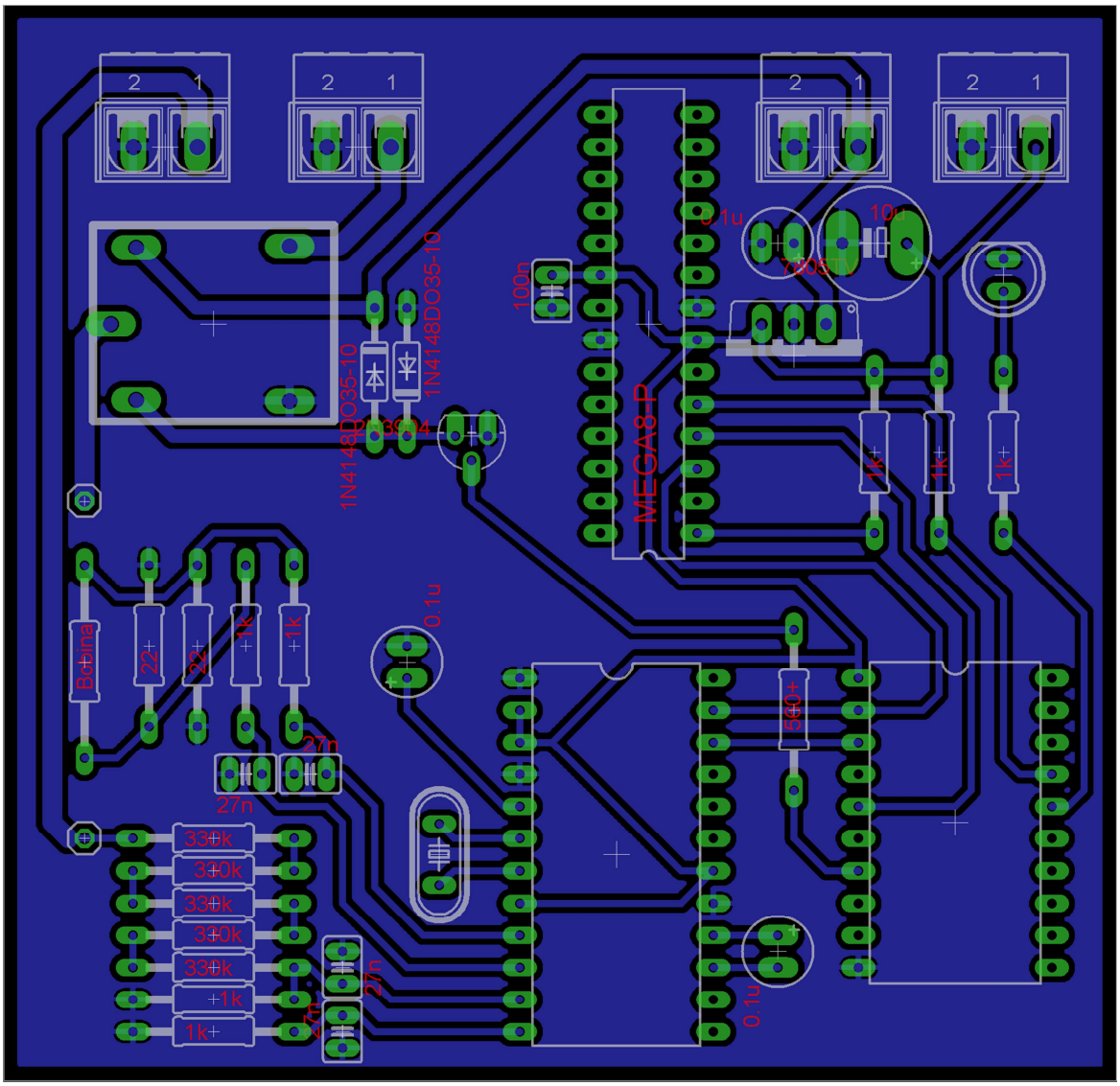

Figura 23 - Layout do circuito do sensor

Apêndice 10 – Material utilizado no circuito do sensor

- XBee Serie 2 – Firmware 21A7
- CS5480
- Atmega328
- Raspberry Pi
- Transformador de corrente com 18 voltas
- 5 Resistores de 330 kΩ
- 7 Resistores de 1 kΩ
- 2 Resistores de 22Ω
- 2 Capacitores de 0,1 µF
- 1 Capacitor de 10uF
- 1 Capacitor de 100nF
- 1 Capacitor de 1uF
- 4 Capacitores de 27nF
- 1 Regulador de tensão LM7805
- 1 Cristal de 4.096 MHz
- 1 Diodo 1N4004
- 1 Diodo 1N4148
- 1 Led
- 1 Transistor 2N3904
- 1 Relé HK3FF