

CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA



Universidade Federal
de Campina Grande

ALEX CARLOS RODRIGUES ALVES

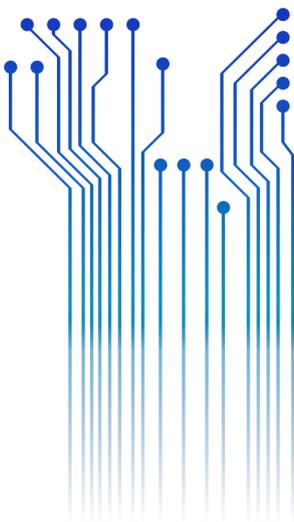


Centro de Engenharia
Elétrica e Informática

TRABALHO DE CONCLUSÃO DE CURSO
SISTEMA DE AQUISIÇÃO DE DADOS BASEADO NA BEAGLEBONE BLACK



Departamento de
Engenharia Elétrica



Campina Grande
Junho de 2016

ALEX CARLOS RODRIGUES ALVES

SISTEMA DE AQUISIÇÃO DE DADOS BASEADO NA BEAGLEBONE BLACK

*Trabalho de Conclusão de Curso submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.*

Área de Concentração: Sistemas Elétricos

Orientador:

Professor George Rossany Soares de Lira, D. Sc.

Campina Grande
Junho de 2016

ALEX CARLOS RODRIGUES ALVES

SISTEMA DE AQUISIÇÃO DE DADOS BASEADO NA BEAGLEBONE BLACK

*Trabalho de Conclusão de Curso submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.*

Área de Concentração: Sistemas Elétricos

Aprovado em / /

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professor George Rossany Soares de Lira, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho ao meu pai, que sempre lutou para que seus filhos tivessem oportunidade de alcançar seus objetivos, e à minha mãe, Maria Miriam (*in memoriam*), sempre em meu coração, a luz que ilumina o meu caminhar.

AGRADECIMENTOS

Agradeço ao meu pai, Paulo Elesbão Alves, e ao meu irmão, Paulo Roberto Rodrigues Alves, por todo o apoio, confiança e companheirismo em mim depositados, e por toda sabedoria e ensinamentos compartilhados e que fazem de mim a pessoa que sou.

Aos amigos de graduação, Armando Gomes Ferreira e José Lucas Vieira, pelos momentos vivenciados durante todo o período de graduação e por todas as lembranças, que fizeram e fazem esta caminhada ainda mais significativa.

Ao amigo Raphael Ribeiro da Silva por todos os anos de convivência, aprendizado, amizade e apoio nos momentos bons e ruins, desde o início da graduação.

Aos amigos do Laboratório de Eletrônica Aplicada a Sistemas Elétricos (LEASE), por toda a ajuda e pelos momentos de descontração durante a execução deste trabalho.

Aos companheiros de projeto, Eddy Lopes Medeiros e Rafael Mateus Carvalho de Paiva, por todo o empenho, ajuda e orientação na realização deste projeto.

Aos meus amigos por fazerem parte dos diversos momentos da minha vida.

Por fim, mas não menos importante, ao professor George Rossany Soares de Lira pela atenção, oportunidade e orientação no decorrer do trabalho.

“Life comes from within your heart and desire.”

Pearl Jam.

RESUMO

O crescente desenvolvimento de Computadores de Placa Única têm tornado possível o acesso a diversos dispositivos capazes de serem aplicados às mais diversas áreas da Engenharia Elétrica. A *BeagleBone Black* é um exemplo desse tipo de computador e é utilizada, no presente trabalho, como base para o desenvolvimento de um sistema de aquisição e armazenamento de dados (tensão e corrente) para aplicações genéricas em Sistemas Elétricos. Para o auxílio na aquisição de dados, fez-se o uso de um conversor analógico-digital. O sistema permite a coleta de dados de até oito sinais de tensão ou corrente. Alguns testes básicos são efetuados com o intuito de validar o sistema. Por fim, são apresentados os resultados dos testes por meio de figuras geradas nos *software* MATLAB.

Palavras-chave: Computadores de Placa Única, *BeagleBone Black*, Sistema de Aquisição de Dados, Conversor Analógico-Digital.

ABSTRACT

The growing development of Single Board Computers has made possible the access to a wide variety of devices capable of being applied to various fields of Electrical Engineering. The BeagleBone Black is an example of this type of computer and it is used in this study as a basis for the development of an acquisition and storage data system (voltage and current) for generic applications in Electrical Systems. For assistance in data acquisition, an analog-digital converter is used. The system can collect data from up to eight voltage or current signals. Some basic tests are performed in order to validate the system. Finally, the results of the tests are presented by means of plots generated in MATLAB.

Keywords: Single Board Computer, BeagleBone Black, Data Acquisition System, Analog-to-Digital Converter.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação de um sinal analógico (a) em tempo contínuo e (b) em tempo discreto.	18
Figura 2 – Variação de um sinal digital binário ao longo do tempo.	18
Figura 3 – Conversor A/D com saída paralela.	19
Figura 4 – Diagrama de blocos de um sistema de instrumentação básico.	20
Figura 5 – Representação de um amplificador operacional.	21
Figura 6 – Interface SPI básica.	23
Figura 7 – Visões do topo e da base da BeagleBone Black.	25
Figura 8 – Conjuntos de pinos P8/P9.	27
Figura 9 – Etapas para implantação de um programa no PRU-ICSS.	28
Figura 10 – Pinos do MCP3008.	29
Figura 11 – (a) Aplicação típica do ACS712. (b) Tensão de saída x corrente medida.	30
Figura 12 – Diagrama de blocos do sistema de aquisição de dados.	31
Figura 13 – Circuito para condicionamento do sinal de tensão.	32
Figura 14 – Circuito para condicionamento do sinal de corrente.	32
Figura 15 – Diagrama de fluxo (programa da PRU0).	35
Figura 16 – Comunicação entre a BBB e o MCP3008.	37
Figura 17 – Circuito para medição de tensão e corrente.	40
Figura 18 – Circuito com quatro lâmpadas incandescentes.	40
Figura 19 – Curvas de tensão: (a) sistema de aquisição de dados desenvolvido e (b) osciloscópio.	41
Figura 20 – Curvas de corrente: (a) sistema de aquisição de dados desenvolvido e (b) osciloscópio.	42
Figura 21 – Curvas de Tensão e corrente obtidas pelo sistema de aquisição desenvolvido.	42
Figura 22 – Tensões de fase ($100 V_{RMS}$) obtidas com o sistema desenvolvido.	43
Figura 23 – Correntes de fase com $100 V_{RMS}$ obtidas com o sistema desenvolvido.	44

LISTA DE TABELAS

Tabela 1 – Configuração do pino GPIO.	26
Tabela 2 – Pinos utilizados com o MCP3008.....	33

LISTA DE ABREVIATURAS E SIGLAS

A/D	Analógico-Digital
ADC	<i>Analog to Digital Converter</i>
AGND	<i>Analog Ground</i>
AMD	<i>Advanced Micro Devices</i>
ASICs	<i>Application Specific Integrated Circuits</i>
BBB	<i>BeagleBone Black</i>
CAN	<i>Controller Area Network</i>
CCSv6	<i>Code Composer Studio v6</i>
CGT	<i>Code Generation Tools</i>
CH0-CH7	Canais de Entrada Analógica 0-7
CLK	<i>Serial Clock</i>
CS	<i>Chip Select</i>
\overline{CS} /SHDN	<i>Chip Select/Shutdown</i>
DDR3	<i>Double Data Rate 3</i>
DGND	<i>Digital Ground</i>
D _{IN}	<i>Serial Data Input</i>
D _{OUT}	<i>Serial Data Output</i>
DTO	<i>Device Tree Overlay</i>
eMMC	<i>embedded MultiMediaCard</i>
FPGA	<i>Field Programmable Gate Array</i>
GPIO	<i>General Purpose Input-Output</i>
GPMC	<i>General Purpose Memory Controller</i>
HDMI	<i>Hign Definition Multimedia Interface</i>
PC	<i>Inter-Integrated Circuit</i>
LCD	<i>Liquid Crystal Display</i>
McASP	<i>Multi-Channel Audio Serial Port</i>
MISO	<i>Master In-Slave Out</i>
MMC	<i>MultiMediaCard</i>

MOSI	<i>Master Out-Slave In</i>
PRU	<i>Programmable Real-Time Unit</i>
PRU-ICSS	<i>Programmable Real-Time Unit and Industrial Communication Subsystem</i>
PWM	<i>Pulse Width Modulation</i>
RISC	<i>Reduced Instruction Set Computer</i>
SBC	<i>Single Board Computer</i>
SPI	<i>Serial Peripheral Interface</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>
V _{DD}	Alimentação de 2,7V a 5,5V
V _{REF}	Entrada de Tensão de Referência

SUMÁRIO

1	Introdução.....	14
2	Fundamentação Teórica.....	16
2.1	Computadores de Placa Única (SBCs).....	16
2.2	Conversão Analógico/Digital.....	17
2.3	Condicionamento de Sinais.....	19
2.3.1	Amplificador Operacional	20
2.3.2	Filtros.....	21
2.4	Comunicação SPI.....	22
3	Principais Componentes do Sistema de Aquisição de Dados.....	24
3.1	<i>BeagleBone Black</i> (BBB).....	24
3.1.1	Visão Geral.....	24
3.1.2	<i>Device Tree Overlay</i> (DTO)	26
3.1.3	<i>Programmable Real-Time Unit and Industrial Communication Subsystem</i> (PRU-ICSS).....	27
3.2	Conversor A/D MCP3008.....	29
3.3	Sensor de Corrente ACS712	30
4	Sistema de Aquisição de Dados.....	31
4.1	Circuito de Condicionamento	31
4.2	Aquisição de Dados Utilizando a <i>BeagleBone Black</i>	32
4.2.1	Configuração do <i>Device Tree Overlay</i>	32
4.2.2	Configuração da PRU0	33
4.2.3	Implementação do Código em C	36
5	Resultados e Discussão.....	37
5.1	Comunicação Entre a BBB e o MCP3008	37
5.2	Calibração para o Condicionamento dos Sinais de tensão e Corrente	38
5.3	Circuito com Lâmpadas Incandescentes	39
5.4	Sistema Trifásico com Lâmpadas Incandescentes	43
6	Conclusão	45
	Referências	46

1 INTRODUÇÃO

Nos últimos anos, o surgimento e desenvolvimento de diversos computadores em miniatura, chamados de Computadores de Placa Única (SBC, do inglês, *Single Board Computer*), têm tornado viável a expansão da criatividade tecnológica ao oferecer plataformas de baixo custo, tamanho compacto, baixo consumo de energia e com acesso a projetos de código aberto (*open source*).

Apesar de serem minúsculos (alguns são do tamanho de um cartão de crédito), os SBCs possuem microprocessador, memória, interfaces de entrada/saída e alguns são até mesmo capazes de rodar um sistema operacional (como Linux, Android, entre outros). Estas características, entre outras, fazem com que estes minicomputadores sirvam como ferramentas educacionais para o ensino e aprendizado de eletrônica e programação. Além disso, oferecem suporte computacional suficiente para que usuários mais avançados criem projetos mais complexos e ousados.

Como exemplos de SBCs disponíveis no mercado, podem ser citadas a Raspberry PI, a Intel Galileu e a *BeagleBone Black* (BBB). Esta difere das outras, principalmente, por ter sido construída para ser interfaceada. Por exemplo, o microprocessador da BBB possui dois microcontroladores adicionais que podem ser usados para interface em tempo real. Outra característica de destaque da BBB é o fato dela ser uma plataforma completamente *open source* (é possível ter acesso aos esquemas, *layout* de *hardware*, materiais e manuais de referência técnica, o que faz com que o usuário seja capaz de modificar o *design* da *BeagleBone* e integrar a um produto próprio).

Além das características descritas anteriormente, a BBB: contém um processador capaz de executar mais de dois bilhões de instruções por segundo; pode ser expandida por meio do uso de dispositivos USB e placas de expansão (*capex*); recebe suporte de uma enorme comunidade de inovadores e entusiastas; possui baixo custo; apresenta baixo consumo de energia; e suporta ferramentas e aplicações de *software* livre.

Diante do que foi exposto anteriormente, a BBB se apresenta como uma alternativa para o desenvolvimento de projetos de custo relativamente baixo e que necessitam tratar dados de forma rápida e eficaz.

Deste modo, o objetivo geral deste trabalho é desenvolver um sistema de aquisição de dados (tensão e corrente) tendo como base a *BeagleBone Black*. Os dados devem ser adquiridos com o auxílio de um conversor analógico-digital e armazenados na memória da BBB.

Além disso, este trabalho tem como objetivos específicos: a elaboração de uma estratégia de configuração e programação que possibilite a BBB trabalhar como sistema de controle da aquisição e armazenamento de dados; e a determinação do conversor A/D com melhor custo benefício para realizar a aquisição de sinais de corrente e tensão para aplicações genéricas em sistemas de potência.

Este documento está dividido do seguinte modo: na seção 2 são apresentadas as bases teóricas para a compreensão e elaboração do trabalho; na seção 3 são descritos os principais componentes do sistema de aquisição de dados, entre eles, a BBB e o conversor A/D utilizado; na seção 4, encontram-se o circuito de condicionamento de sinal e uma descrição da configuração e programação da BBB para que sejam efetuados o controle e o armazenamento dos dados; os resultados e discussões dos testes realizados para validar o sistema estão dispostos na seção 5; por fim, na seção 6, são apresentadas as conclusões deste trabalho, seguidas da bibliografia.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentadas as bases teóricas para execução do trabalho.

2.1 COMPUTADORES DE PLACA ÚNICA (SBCs)

O primeiro SBC surgiu em meados dos anos 70 com o nome de *Dyna-micro* e sua concepção foi possível graças ao advento dos circuitos integrados, microprocessadores, e conjuntos miniaturizados que apresentam múltiplos componentes eletrônicos em um único pacote (ROSCH, 2003).

Segundo (BRAUN, MOSS e STIGALL, 1986), “um computador de placa única é melhor definido como uma placa de circuito impresso contendo os recursos necessários para criar um microcomputador funcional”.

Algumas razões contribuíram para que ocorresse a redução de um computador a uma única placa de circuito. Entre elas, tem-se o custo. Uma quantidade menor de placas reduz os custos de fabricação e de materiais. Além disso, o *design* de um SBC elimina os conectores de barramento, que representam a parte mais sujeita a falhas em sistemas computacionais. Por outro lado, a placa única não pode ser adaptada a novas tecnologias, pois tem suas capacidades de conexão fixadas a partir do momento em que é soldada, durante o processo de fabricação (ROSCH, 2003).

Nos dias atuais, os computadores de placa única estão agrupados em duas categorias principais. A primeira delas diz respeito aos SBCs que são, geralmente, projetos industrializados que passaram pelo mesmo tipo de teste exigido por um produto final e são, frequentemente, integrados a produtos finais (produtos prontos para uso). Eles são ditos “de proprietário”. A segunda categoria é composta pelos SBCs ditos de código aberto, os quais oferecem aos usuários acesso tanto ao projeto do *hardware* da placa quanto ao código utilizado na mesma. Isto permite que os usuários entendam facilmente como o *software* e o *hardware* funcionam, o que faz com que sejam capazes de adotar o modelo mais adequado às necessidades de seus projetos (ORTMEYER, 2014).

Os SBCs atuais são disponibilizados com uma vasta variedade de processadores, muitos destes com unidade de processamento gráfico *on-board*. Esses processadores variam desde processadores baseados em x86 (AMD e Intel) até processadores ARM. No que se refere ao software utilizado nos computadores de placa única, o Linux é o mais comum (ORTMEYER, 2014).

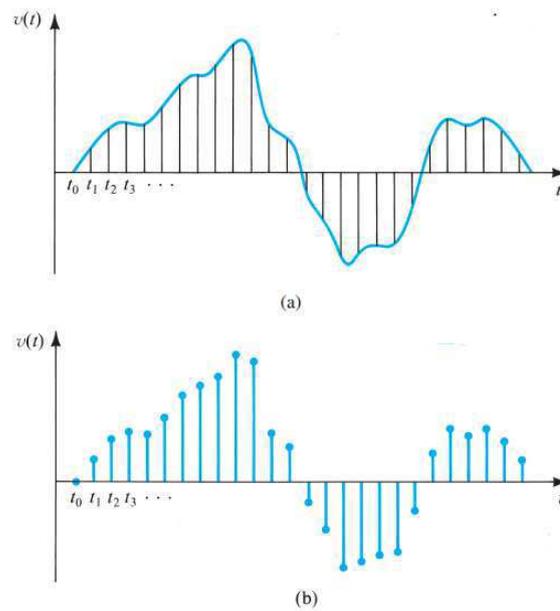
2.2 CONVERSÃO ANALÓGICO/DIGITAL

A amplitude de um sinal analógico pode assumir qualquer valor em um intervalo contínuo. Isto significa que ela apresenta uma variação contínua ao longo de sua faixa de atividade. A grande maioria dos sinais a nossa volta são analógicos e os circuitos eletrônicos encarregados de processá-los são chamados de circuitos analógicos (SEDRA e SMITH, 2014).

Um sinal também pode ser representado como uma sequência de números, cada número correspondendo à magnitude do sinal em um instante de tempo. Como resultado, tem-se um sinal digital. Na Figura 1, podem ser observadas curvas de tensão, as quais são utilizadas pra ilustrar como sinais são convertidos da forma analógica para digital. Inicialmente, são marcados intervalos de tempo igualmente espaçados ao longo do eixo x (t_0 , t_1 , t_2 , e assim por diante). Para cada um desses intervalos, a magnitude é medida por meio de um processo conhecido como amostragem. O sinal resultante desse processo é um sinal em tempo discreto (não mais uma função de tempo contínuo). Contudo, uma vez que sua magnitude pode assumir qualquer valor no intervalo contínuo, ele ainda é um sinal analógico (SEDRA e SMITH, 2014).

A representação da amplitude de cada amostra do sinal na Figura 1(b) por um valor contendo um número finito de dígitos faz com que a amplitude do sinal deixe de ser contínua. Ela passa, então, a ser chamada de quantizada, discretizada, ou digitalizada. Deste modo, obtém-se como resultado um sinal digital que é simplesmente uma sequência de números que representam a magnitude de sucessivas amostras do sinal (SEDRA e SMITH, 2014).

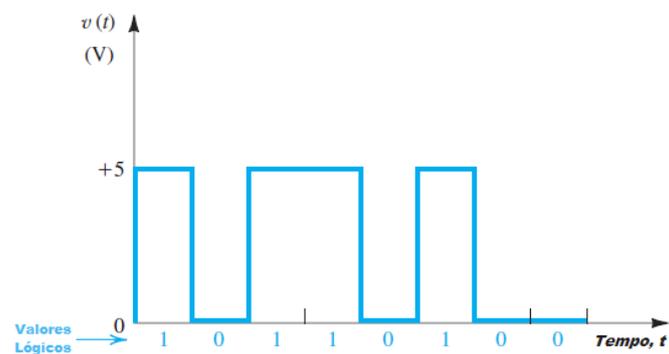
Figura 1 – Representação de um sinal analógico (a) em tempo contínuo e (b) em tempo discreto.



Fonte: (SEDRA e SMITH, 2014).

A amplitude de sinais digitais pode, então, assumir apenas valores dentro de um conjunto finito. Quando apenas dois valores são possíveis (um e zero), os sinais digitais são ditos binários e a informação é codificada em conjuntos de uns e zeros (PALLÁS-ARENY e WEBSTER, 1999). A Figura 2 apresenta a variação de um sinal digital ao longo do tempo. Neste caso, o nível alto (lógica 1) é representado por +5V e o nível baixo (lógica 0) é representado por 0V.

Figura 2 – Variação de um sinal digital binário ao longo do tempo.



Fonte: Adaptado de (SEDRA e SMITH, 2014).

Um sistema digital não pode receber dados analógicos diretamente em sua entrada. Sendo assim, esses dados devem ser convertidos para forma digital. Inicialmente, em intervalos de tempo discreto, são tomadas amostras do sinal analógico

que, em seguida, são convertidas para forma digital. Essa, por sua vez, corresponde à representação binária da tensão de entrada no instante de amostragem. Muitos conversores analógico-digitais (ADCs, do inglês, Analog to Digital Converters) produzem uma “palavra” (sequência de bits de tamanho fixo) que possui tamanho entre 8 e 16 bits (WINDER, 2002).

O ADC recebe em seus terminais de entrada amostras de um sinal analógico e fornece em seus N terminais de saída, para cada uma das amostras, sua representação binária de N-bits (Figura 3) (SEDRA e SMITH, 2014). Neste caso, os bits são apresentados, de forma paralela, na saída do conversor. Porém, há também conversores que transmitem o resultado da conversão de forma serial (comunicação serial).

Figura 3 – Conversor A/D com saída paralela.



Fonte: Adaptado de (SEDRA e SMITH, 2014).

Para que os dados convertidos não sejam corrompidos, a frequência de amostragem deve ser o dobro da maior frequência do sinal amostrado (teorema de Nyquist). Sendo assim, por exemplo, um sinal que tem frequência mais alta igual a 3.4 kHz deve ser amostrado em 8 kHz (WINDER, 2002).

2.3 CONDICIONAMENTO DE SINAIS

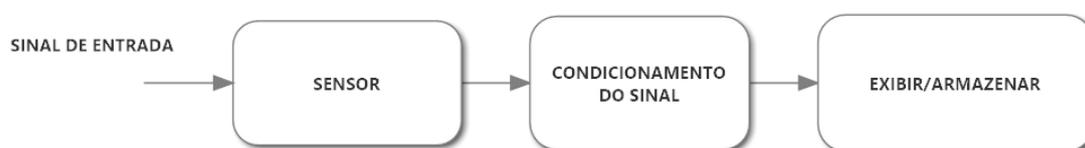
Para descrever a manipulação de um sinal analógico com o objetivo de torná-lo apropriado para utilização em outro estágio de processamento, utiliza-se o termo condicionamento de sinal (MOLLOY, 2015).

Deste modo, os condicionadores de sinais medem elementos do sistema que correspondem à saída de um sensor elétrico e, então, produzem um sinal adequado para transmissão, exibição, ou registro (armazenamento). Esses condicionadores, normalmente, consistem de circuitos eletrônicos que desempenham qualquer uma das

seguintes funções: amplificação, deslocamento de nível, filtragem, casamento de impedância, modulação e demodulação (PALLÁS-ARENY e WEBSTER, 2000).

A necessidade de se condicionar um sinal, tipicamente, surge de um sistema como o apresentado na Figura 4. Neste caso, se o elemento responsável pela exibição/armazenamento do sinal opera com um uma faixa de valores diferente daquela disponível na saída do sensor, faz-se necessário o uso de algum circuito capaz de transformar essa informação de saída em um sinal que esteja dentro da faixa de operação do sistema de aquisição/exibição.

Figura 4 – Diagrama de blocos de um sistema de instrumentação básico.



Fonte: o próprio autor.

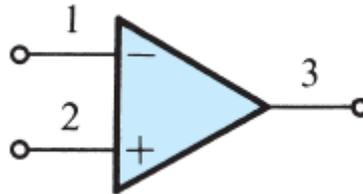
Geralmente, um dos estágios do sistema de medição é digital, enquanto a saída do sensor é analógica. Os ADCs são capazes de fornecer um código digital a partir de um sinal analógico. Porém, eles requerem que o sinal de entrada esteja em uma margem de amplitude específica, geralmente menor que $\pm 10V$. Portanto, como as saídas dos sensores podem ter uma amplitude na ordem de milivolts, elas precisam ser condicionadas antes de serem aplicadas ao ADC (PALLÁS-ARENY e WEBSTER, 2000). Há também casos em que a informação que se deseja converter apresenta valor maior do que a entrada do ADC pode suportar. Nestes casos, o condicionamento também se faz necessário.

2.3.1 AMPLIFICADOR OPERACIONAL

O amplificador operacional é um dispositivo eletrônico composto por dois terminais de entrada e um terminal de saída (Figura 5). A entrada positiva (+) é conhecida como não-inversora e a entrada negativa (-) é conhecida como inversora (MORRIS, 2001). Algumas das características do amplificador operacional ideal incluem (MOLLOY, 2015):

- Um ganho infinito em malha aberta;
- Uma impedância de entrada infinita;
- Uma impedância de saída zero.

Figura 5 – Representação de um amplificador operacional.



Fonte: (SEDRA e SMITH, 2014).

Esse dispositivo é projetado para detectar a diferença entre os sinais de tensão aplicados aos seus terminais de entrada (isto é, $v_2 - v_1$), multiplicar por um número G , e produzir uma tensão de saída igual a $G(v_2 - v_1)$ (SEDRA e SMITH, 2014). Portanto, o amplificador operacional pode ser utilizado quando, por exemplo, deseja-se amplificar o sinal de saída de um sensor.

2.3.2 FILTROS

A detecção e análise de sinais podem ser impossíveis, caso ruídos e sinais indesejados estejam presentes no objeto de estudo. Os filtros eletrônicos são então utilizados, pois permitem a passagem de informações desejadas e impede a passagem daquelas que não são consideradas de interesse para análise. Em outras palavras, os filtros permitem que algumas frequências do sinal aplicado a seus terminais de entrada passem até seus terminais de saída com pouca ou nenhuma redução no nível deste sinal (WINDER, 2002).

Deste modo, filtros podem ser utilizados, por exemplo, para eliminar ou reduzir componentes com determinada frequência em um sinal de saída de um sensor.

2.4 COMUNICAÇÃO SPI

A transferência de dados entre duas unidades pode ocorrer de forma serial ou paralela. Na transmissão paralela, toda mensagem é transmitida de uma única vez e cada bit da mensagem segue seu próprio caminho. Ou seja, uma mensagem de n -bits é transmitida em paralelo por meio de n condutores separados. Enquanto isso, na transmissão serial, cada bit da mensagem é enviado em sequência e esse tipo de transmissão requer o uso de uma ou duas linhas de sinal. Sendo assim, para curtas distâncias e quando a velocidade é importante, utiliza-se a transmissão paralela, uma vez que as múltiplas linhas de sinais operando em paralelo faz com que ela seja mais rápida. Por outro lado, apesar de ser mais lenta, a transmissão serial é mais barata, pois requer o uso de uma quantidade menor de condutores (MANO, KIME e MARTIN, 2015).

Atualmente, entre os tipos de comunicação serial utilizados, tem-se a *Serial Peripheral Interface* (também conhecida como SPI), a qual foi desenvolvida pela Motorola com o objetivo de fornecer uma interface simples e de baixo custo entre microcontroladores e chips periféricos (CATSOULIS, 2005).

Todas as transmissões no protocolo síncrono SPI são referenciadas a um *clock* comum, o qual é gerado pelo mestre (processador). O periférico receptor, chamado de escravo, utiliza o *clock* para sincronizar a sua aquisição do fluxo de bits em série. Vários escravos podem ser conectados a um mesmo mestre. Neste caso, o mestre utiliza a entrada *Chip Select* (\overline{CS}) do escravo para selecioná-lo como receptor. Deste modo, os periféricos que não estão selecionados não participarão da transmissão SPI (CATSOULIS, 2005).

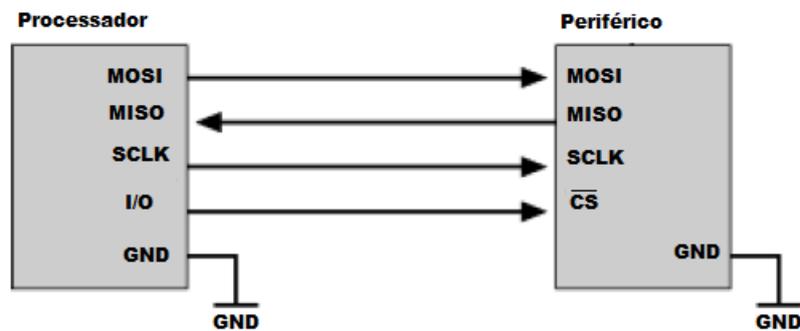
Na Figura 6, uma interface SPI básica pode ser observada. A comunicação entre mestre e escravo, ilustrada na figura, segue os seguintes passos (MOLLOY, 2015):

- A frequência de *clock* em que os canais de comunicação de dados são sincronizados é definida pelo SPI mestre.
- O SPI mestre coloca a linha CS em nível baixo para que o dispositivo cliente (escravo) seja ativado (nível baixo ativo).
- Após um pequeno atraso, ciclos de *clock* são emitidos pelo mestre, o qual envia dados por meio da linha MOSI (do inglês, *master out-slave*

in) e recebe dados por meio da linha MISO (do inglês, *master in-slave out*). Por outro lado, o dispositivo escravo lê dados da linha MOSI e envia dados por meio da linha MISO. A cada ciclo de *clock*, um bit é enviado e outro é recebido.

- Quando a transmissão se completa, o mestre para de enviar o sinal de *clock* e coloca a linha CS em nível alto. Conseqüentemente, o dispositivo escravo é desativado.

Figura 6 – Interface SPI básica.



Fonte: Adaptado de (CATSOULIS, 2005).

3 PRINCIPAIS COMPONENTES DO SISTEMA DE AQUISIÇÃO DE DADOS

O sistema de aquisição de dados desenvolvido é composto por elementos principais como a *BeagleBone Black*, o conversor A/D MCP3008 e o sensor de corrente ACS712, os quais são descritos nesta seção. Outros componentes que podem ser incluídos no sistema dependerão da aplicação que se deseja realizar. Entre esses componentes, podem ser citados: resistores, capacitores, reguladores de tensão, amplificadores operacionais e diodos. Além disso, fontes de tensão devem ser utilizadas para alimentação dos circuitos projetados.

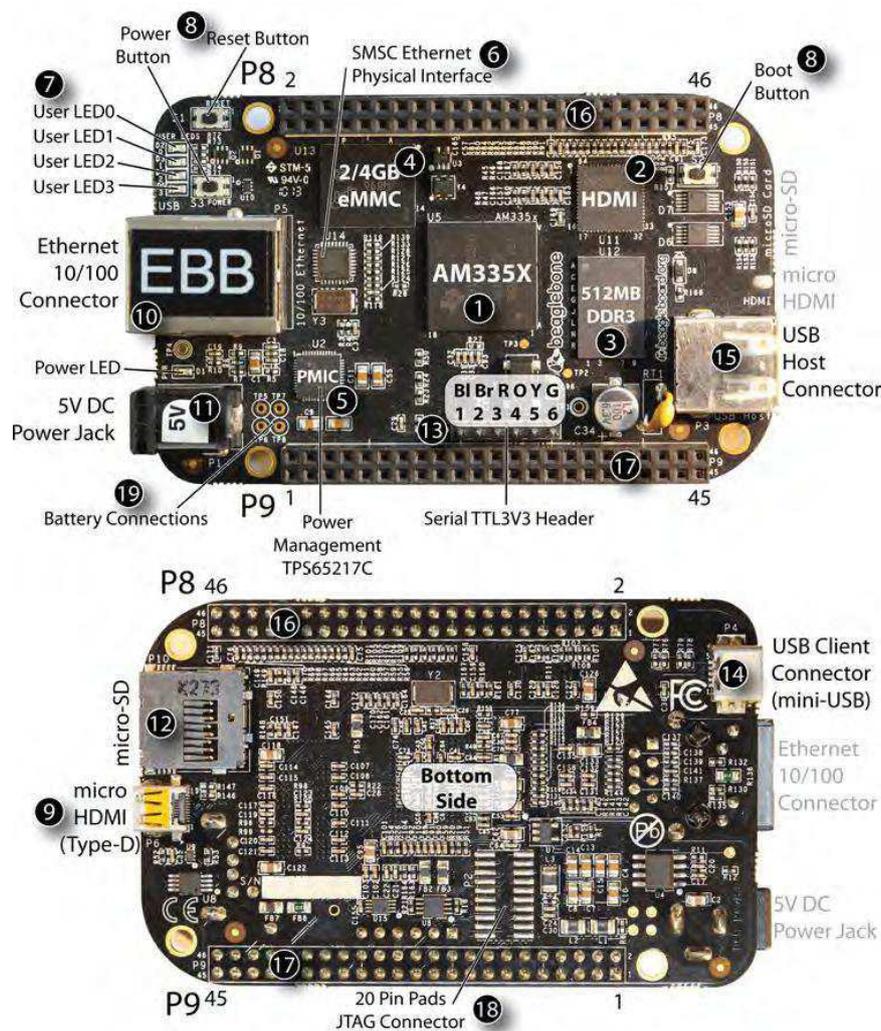
3.1 *BEAGLEBONE BLACK* (BBB)

3.1.1 VISÃO GERAL

A *BeagleBone Black* é uma plataforma para desenvolvedores e hobbystas, a qual possui um processador de 1 GHz (Texas Instruments Sitara AM335x Cortex A8 ARM capaz de executar 2 bilhões de instruções por segundo), memória DDR3 de 512MB, Ethernet, armazenamento eMMC e suporte de vídeo HDMI. Além disso, possui dois microcontroladores que permitem interface em tempo real (unidades programáveis em tempo real).

Na Figura 7, podem ser observados os sistemas centrais da BBB e os diversos componentes da plataforma.

Figura 7 – Visões do topo e da base da BeagleBone Black.



Fonte: (MOLLOY, 2015).

A BBB possui dois conjuntos de pinos P8 e P9 que são compostos por um número máximo de 65 GPIOs (General Purpose Input/Outputs), os quais suportam até 3,3 V (uma tensão de 5 V, por exemplo, pode vir a queimar a placa). Possui ainda 8 saídas analógicas (saídas PWM), que permitem a emissão de um sinal de saída analógica variável (0 a 3,3 V), e 7 entradas analógicas de 12 bits que suportam até 1,8 V. No que se refere ao fornecimento de energia, podem ser encontrados pinos que fornecem 5 V e 3,3 V e uma alimentação de referência de 1,8 V (para as entradas analógicas). Outras funcionalidades disponíveis são: 8 contadores; 2 barramentos I²C; 4 UARTs usados para comunicação serial entre dois dispositivos; 2 Controller Area Networks (CAN); 2 SPIs; General-Purpose Memory Controller (GPMC) utilizada para conexão de dispositivos de memória externos, como FPGAs ou ASICs; 2 MMCs; LCD; 2 McASP (MOLLOY, 2015).

3.1.2 DEVICE TREE OVERLAY (DTO)

Os pinos da *BeagleBone Black* (BBB) podem ser configurados em oito diferentes modos. Para configurar o pino de acordo com o modo desejado, utiliza-se um número binário de três dígitos, como apresentado na primeira coluna da Tabela 1 (baseada na tabela 9-60 do *AM335x Technical Reference Manual*). Estes três bits são utilizados, em conjunto com outros quatro bits (formando um número binário de sete bits), para configurar o comportamento exato de um pino GPIO usando *Device Tree Overlays* (MOLLOY, 2015).

Nas tabelas 12 e 13 do *BeagleBone Black System Reference Manual*, podem ser encontrados os diferentes modos em que os pinos da BBB podem ser configurados.

Tabela 1 – Configuração do pino GPIO.

BIT	Campo do AM335x	Descrição
0,1,2	mmode	Modo Multiplexador: os bits menos significativos podem ser usados para selecionar entre 0 (000 ₂) e 7 (111 ₂). Deste modo, cada pino pode ter até 8 modos diferentes.
3	puden	0 ativa o resistor de <i>pull-up/pull-down</i> e 1 desativa.
4	putypesel	0 seleciona o resistor de <i>pull-down</i> . 1 seleciona o resistor de <i>pull-up</i> .
5	rxactive	Quando este bit está selecionado para nível alto, o pino se comporta como uma entrada. Quando colocado em nível baixo, o pino se comporta como uma saída.
6	slewctrl	Fornecer controle ao tempo subida/descida de uma saída. Rápido = 0, Devagar = 1.

Fonte: Adaptado de (MOLLOY, 2015).

Device Tree Overlay (DTO) é uma maneira de descrever *hardware* em um sistema. O influxo de sistemas ARM nos últimos anos fez com que ocorressem muita

confusão e conflitos no *kernel* do Linux envolvendo componentes ARM. Deste modo, foi decidido implementar *Device Tree* nas últimas versões do *kernel* instaladas com a *BeagleBone Black* (COOPER, 2014).

Em (COOPER, 2014), podem ser encontrados detalhes sobre como criar, habilitar, desabilitar e compilar um DTO.

3.1.3 PROGRAMMABLE REAL-TIME UNIT AND INDUSTRIAL COMMUNICATION SUBSYSTEM (PRU-ICSS)

O processador AM335x da *BeagleBone Black* possui duas Unidades Programáveis em Tempo Real (PRU0 e PRU1) – *Programmable Real-Time Units* (PRUs). A PRU é um processador de 32 bits e 200 MHz (arquitetura RISC) com acesso a um determinado número de pinos e acesso completo a memória interna e periféricos do processador AM335x na *BeagleBone*. Estas unidades fazem parte do *Programmable Real-time Unit Industrial Control Subsystem* (PRU-ICSS) e podem ser usadas para interfaces que necessitam de operações em tempo real.

As PRUs são capazes de ativar interrupções e compartilhar memória com o servidor Linux do dispositivo. Além disso, podem também usar os conjuntos de pinos P9/P8 da BBB (MOLLOY, 2015). A Figura 8 representa estes conjuntos de pinos e os GPIOs específicos do subsistema da PRU.

Figura 8 – Conjuntos de pinos P8/P9.

P9				P8			
DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	GPIO_38	3	4	GPIO_39
VDD_5V	5	6	VDD_5V	GPIO_34	5	6	GPIO_35
SYS_5V	7	8	SYS_5V	GPIO_66	7	8	GPIO_67
PWR_BTN	9	10	SYS_RESETN	GPIO_69	9	10	GPIO_68
GPIO_30	11	12	GPIO_60	PRU0_15 OUT	11	12	PRU0_14 OUT
GPIO_31	13	14	GPIO_50	GPIO_23	13	14	GPIO_26
GPIO_48	15	16	GPIO_51	GPIO_47	15	16	GPIO_46
GPIO_5	17	18	GPIO_4	GPIO_27	17	18	GPIO_65
I2C2_SCL	19	20	I2C2_SDA	GPIO_22	19	20	PRU1_13
GPIO_3	21	22	GPIO_2	PRU1_12	21	22	GPIO_37
GPIO_49	23	24	GPIO_15	GPIO_36	23	24	GPIO_33
PRU0_7	25	26	PRU1_16 IN	GPIO_32	25	26	GPIO_61
PRU0_5	27	28	PRU0_3	PRU1_8	27	28	PRU1_10
PRU0_1	29	30	PRU0_2	PRU1_9	29	30	PRU1_11
PRU0_0	31	32	VDD_ADC	GPIO_10	31	32	GPIO_11
AIN4	33	34	GNDA_ADC	GPIO_9	33	34	GPIO_81
AIN6	35	36	AIN5	GPIO_8	35	36	GPIO_80
AIN2	37	38	AIN3	GPIO_78	37	38	GPIO_79
AIN0	39	40	AIN1	PRU1_6	39	40	PRU1_7
PRU0_6	41	42	PRU0_4	PRU1_4	41	42	PRU1_5
DGND	43	44	DGND	PRU1_2	43	44	PRU1_3
DGND	45	46	DGND	PRU1_0	45	46	PRU1_1

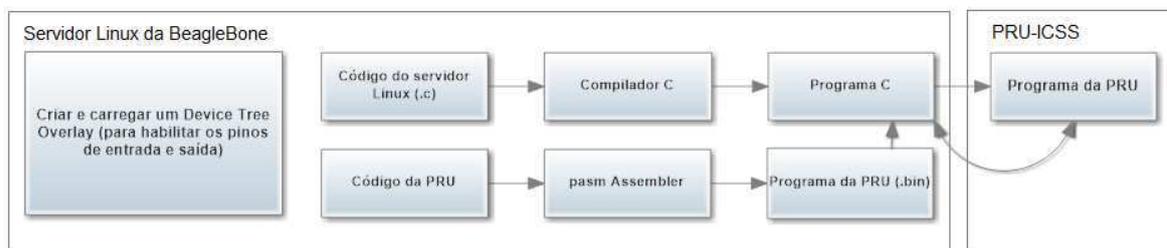
Fonte: Adaptado de (BEAGLEBOARD.ORG, 2016).

Observa-se que 10 pinos estão disponíveis na PRU0 e 15 pinos estão disponíveis na PRU1. Estes pinos precisam ser configurados utilizando um *Device Tree Overlay* e alguns deles já se encontram alocados e não devem ser usados a menos que a funcionalidade a que se encontram alocados seja desabilitada (por exemplo, a saída do HDMI ou o eMMC).

Cada PRU possui uma memória de dados com 8 kB e uma memória de programa com 8 kB. A primeira é usada para armazenar dados que são manipulados pelas instruções do programa e a segunda armazena as instruções a serem executadas pela PRU. Cada PRU é capaz de acessar a memória de dados da outra e, além disso, há uma memória compartilhada de uso geral com 12 kB (MOLLOY, 2015).

Para implantar um programa no PRU-ICSS, são necessárias várias etapas. Estas etapas estão ilustradas na Figura 9 e dizem respeito a: criação de um *Device Tree Overlay*; criação de um programa do servidor Linux (.c); criação de um programa da PRU (.p); compilação do programa em c (.c); compilação do programa em assembly (.p); execução do programa do servidor Linux.

Figura 9 – Etapas para implantação de um programa no PRU-ICSS.



Fonte: Adaptado de (MOLLOY, 2015).

Os programas para PRU são escritos em *assembly* e precisam ser compilados para arquivos binários para serem executados nas PRUs. A arquitetura da PRU-ICSS (RISC) é relativamente pequena e possui aproximadamente 45 instruções que podem ser categorizadas como operações aritméticas, operações lógicas, carregamento e armazenamento do registrador, e controle de fluxo de programa. Cada instrução pode ser vista em detalhes no *AM335x PRU-ICSS Reference Guide* (MOLLOY, 2015).

Há ainda uma alternativa para escrever programas para PRU na linguagem C, ao invés de *assembly*. A *Texas Instruments* lançou o *PRU Code Generation Tools* (CGT) como parte do *Code Composer Studio v6* (CCSv6), o qual pode ser utilizado para escrever programas para PRU em C.

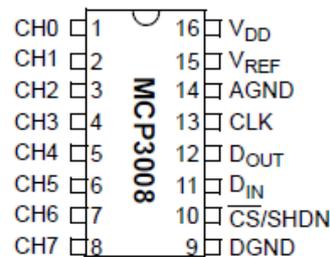
3.2 CONVERSOR A/D MCP3008

O MCP3008 é um conversor A/D de 8 canais, 10 bits de resolução, com interface serial SPI. Ele opera com tensões na faixa de 2,7 V até 5 V. Caso seja alimentado com uma tensão de 2,7 V, sua taxa de amostragem máxima é igual a 75 ksp/s. Quando sua tensão de alimentação é igual a 5 V, a taxa de amostragem máxima que pode ser atingida é 200 ksp/s. Entre as aplicações deste sensor, tem-se:

- Interface de sensores;
- Controle de processos;
- Aquisição de dados;
- Sistemas operados a bateria.

Os pinos do MCP3008 são apresentados na Figura 10. Nesta, podem ser observados os 8 canais de entrada analógica (CH0-CH7), o pino de alimentação (V_{DD}), o pino para tensão de referência (V_{REF}), o terra analógico (AGND), o terra digital (DGND) e os pinos para comunicação SPI (CLK, D_{OUT} , D_{IN} , \overline{CS} /SHDN).

Figura 10 – Pinos do MCP3008.



Fonte: (Microchip Technology Inc., 2008).

Em (Microchip Technology Inc., 2008), podem ser encontradas todas as especificações deste dispositivo. Além disso, podem ser observados os diagramas de tempo correspondentes à comunicação SPI.

Esse dispositivo apresenta custo em torno de 3,75 dólares e pode ser encontrado em sites brasileiros com valor em torno de 20 reais. Sendo assim, por possuir um custo relativamente baixo e ser capaz de operar com taxa de amostragem de até 200 ksp/s (200.000 amostras a cada segundo), o MCP3008 pode ser considerado como uma boa alternativa de custo-benefício.

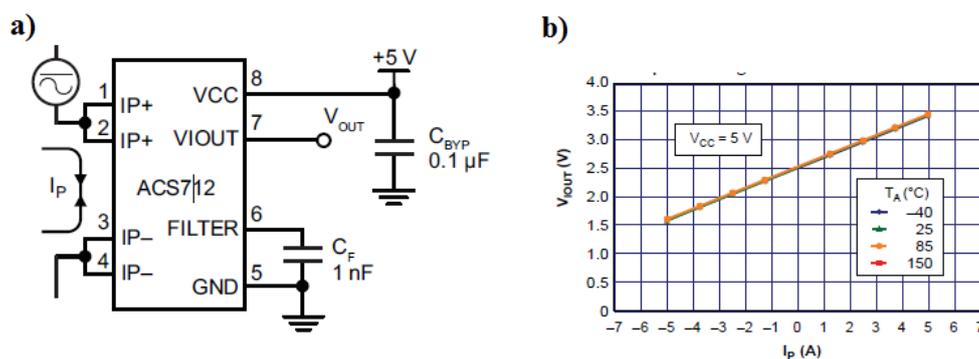
3.3 SENSOR DE CORRENTE ACS712

O sensor de corrente ACS712 da Allegro™ fornece soluções precisas e econômicas para medição de corrente em sistemas industriais, comerciais e de comunicação.

O dispositivo é um sensor de efeito Hall com um caminho condutor de cobre próximo a superfície do molde. Quando um fluxo de corrente é aplicado a este caminho condutor, um campo magnético é gerado e convertido, para um valor proporcional de tensão, por meio do efeito Hall.

Na Figura 11, tem-se: (a) uma aplicação típica para o ACS712 e (b) um gráfico com a tensão de saída em função da corrente medida para o sensor com tolerância de ± 5 A.

Figura 11 – (a) Aplicação típica do ACS712. (b) Tensão de saída x corrente medida.



Fonte: Adaptado de (Allegro MicroSystems, 2012).

A partir da Figura 11, observa-se que o ACS712 apresenta uma tensão de saída igual a, aproximadamente, 1,5 V para um valor de corrente igual a -5 A. Para uma corrente igual a 5 A, a saída de tensão possui valor próximo a 3,5 V. Sendo assim, uma onda de corrente senoidal, quando convertida pelo ACS712, deve apresentar valores entre 1,5 V e 3,5 V, com centro em 2,5 V.

4 SISTEMA DE AQUISIÇÃO DE DADOS

O sistema de aquisição de dados é composto pelo circuito de condicionamento, o qual condiciona os sinais de corrente (provenientes do ACS712) e tensão que são fornecidos aos canais de entrada analógica do MCP3008. Esse, por sua vez, converte os sinais analógicos e os transmite para a BBB por meio da comunicação serial SPI. Na Figura 12, o diagrama de blocos do sistema de aquisição de dados pode ser observado.

Figura 12 – Diagrama de blocos do sistema de aquisição de dados.

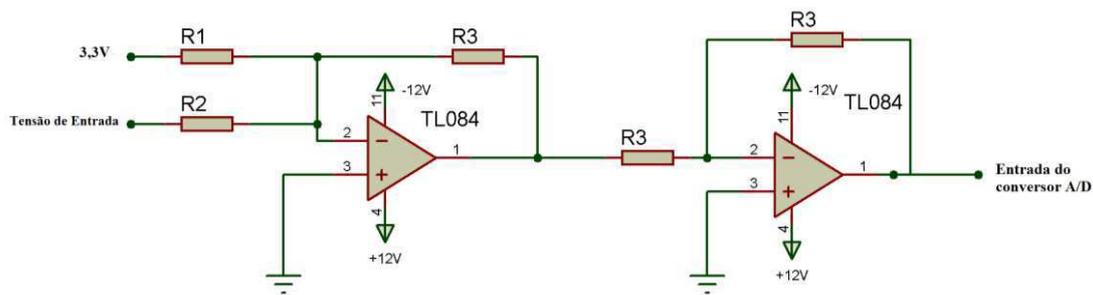


Fonte: o próprio autor.

4.1 CIRCUITO DE CONDICIONAMENTO

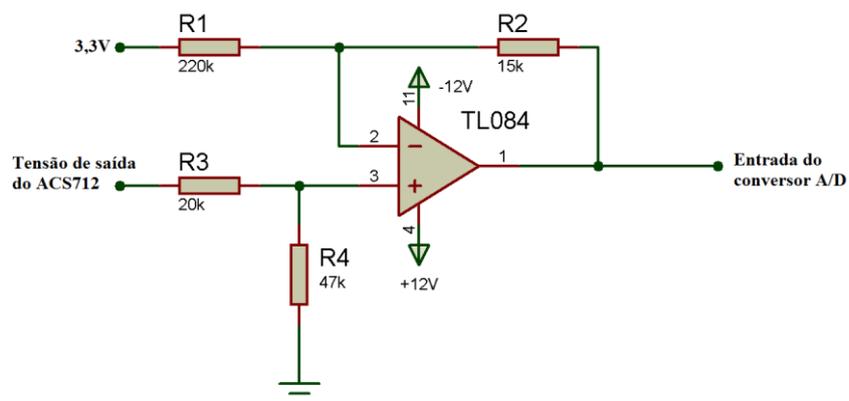
O circuito de condicionamento foi construído com amplificadores operacionais, resistores e capacitores. O sinal de tensão medido foi condicionado a partir de amplificadores operacionais configurados no modo somador e no modo inversor (Figura 13). Uma vez que a saída do circuito somador é negativa, foi utilizado um circuito inversor para que ela fosse mantida em valores positivos (o conversor não recebe tensões negativas). Para o condicionamento do sinal de corrente, foram construídos circuitos subtratores de modo que a variação de corrente foi considerada como sendo ± 5 A (Figura 14). Os capacitores foram usados para filtragem de ruídos de alta frequência.

Figura 13 – Circuito para condicionamento do sinal de tensão.



Fonte: o próprio autor.

Figura 14 – Circuito para condicionamento do sinal de corrente.



Fonte: o próprio autor.

4.2 AQUISIÇÃO DE DADOS UTILIZANDO A *BEAGLEBONE BLACK*

A PRU0 da *BeagleBone Black* foi utilizada para efetuar a comunicação com o MCP3008 e o armazenamento dos dados da conversão. O programa escrito em *assembly* oferece suporte à aquisição de dados dos 8 canais do ADC. É importante destacar que as instruções desse programa são executadas a cada 5 ns (1/200 Mhz). A seguir, são descritas as etapas de configuração da BBB para a aquisição dos dados.

4.2.1 CONFIGURAÇÃO DO *DEVICE TREE OVERLAY*

A criação e o carregamento de um *Device Tree Overlay* dizem respeito à primeira etapa na execução de um programa nas PRUs. É necessário configurar os pinos de acordo com o uso a que se destinam. Deste modo, para algumas aplicações, a saída HDMI deve ser desativada para que não haja conflitos no momento em que se faz a

aquisição de dados, uma vez que alguns dos pinos alocados à saída HDMI pode vir a ser usados como entradas de dados.

Na Tabela 2 são apresentados os pinos da PRU0 que necessitaram ser configurados para o controle do conversor MCP3008 e para a aquisição dos dados enviados pelo mesmo. A primeira coluna da tabela diz respeito aos pinos da BBB que foram utilizados na interação com o conversor A/D (ver Figura 8). A coluna endereço especifica o endereço de memória em que o pino é mapeado. Na coluna Entrada/Saída, pode ser observado se o pino da BBB foi configurado como entrada ou como saída. A última coluna se refere aos pinos do MCP3008 (dispositivo SPI escravo) que deveriam ser conectados aos respectivos pinos da BBB (dispositivo SPI mestre). O pino P9.31 foi utilizado para gerar o *clock* em que os canais de comunicação são sincronizados. A entrada P9.29 foi conectada ao pino D_{OUT} do conversor para receber os dados da conversão. A configuração do MCP3008 (especificação modo e do canal de conversão, segundo *datasheet*) foi feita por meio do pino P9.30. Além disso, a conversão foi controlada pelo pino P9.28.

Tabela 2 – Pinos utilizados com o MCP3008.

Pino – Beaglebone	Endereço	Entrada/Saída	Pino – MCP3008
PRU0_0 (P9.31)	0x190	Saída	CLK
PRU0_1 (P9.29)	0x194	Entrada	D _{OUT}
PRU0_2 (P9.30)	0x198	Saída	D _{IN}
PRU0_3 (P9.28)	0x19c	Saída	$\overline{\text{CS}}/\text{SHDN}$

Fonte: o próprio autor.

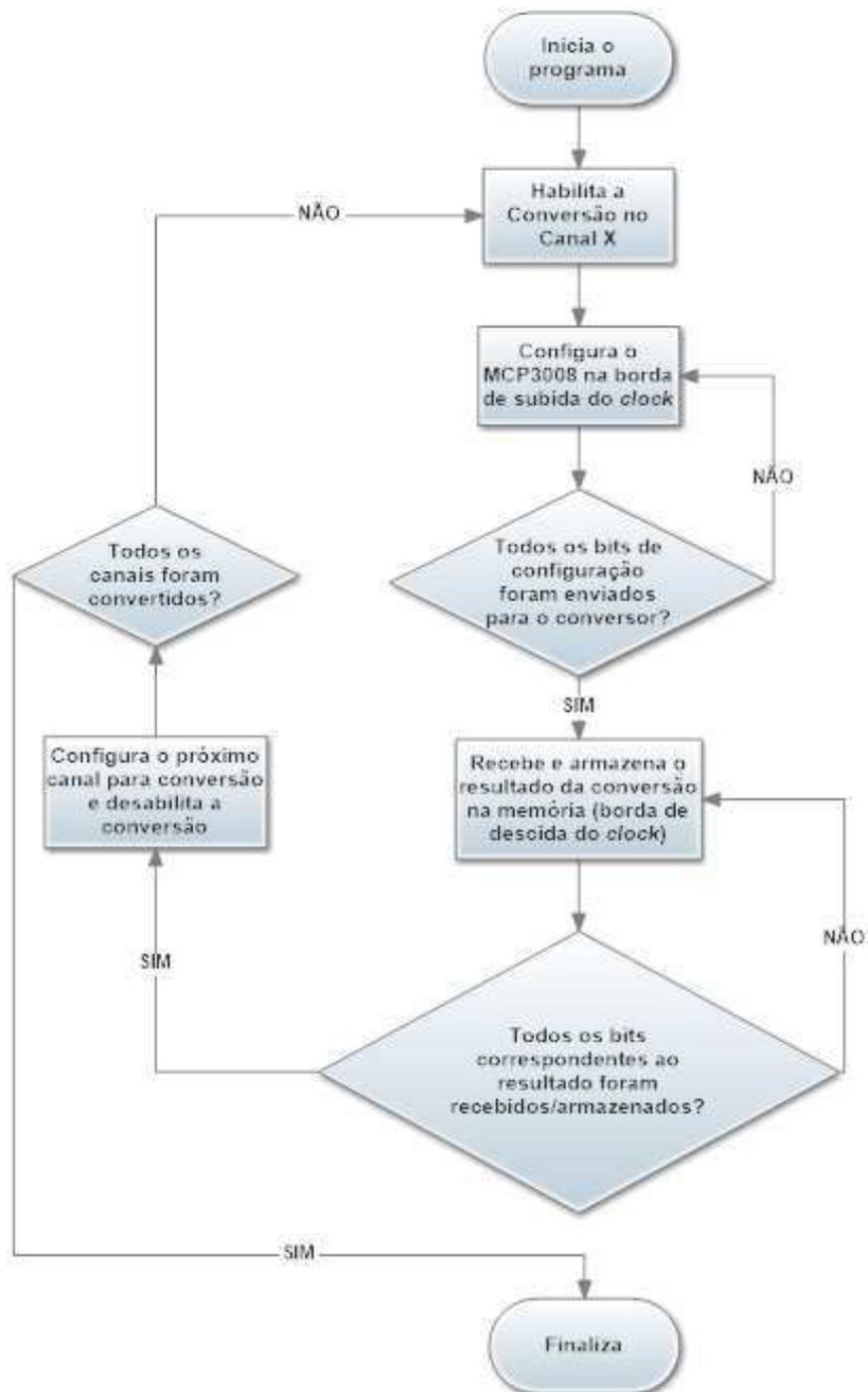
4.2.2 CONFIGURAÇÃO DA PRU0

O programa em *assembly* encarregado de estabelecer a comunicação entre a BBB e o MCP3008 foi configurado de modo que a conversão pudesse ser feita em todos os canais de entrada analógica do conversor A/D.

Na Figura 15, apresenta-se, de um modo mais geral, o diagrama de fluxo do programa em *assembly* executado na PRU0. O programa inicia configurando alguns registradores e, logo em seguida, habilita a conversão no canal X (inicialmente, CH0). No próximo passo, o MCP3008 é configurado por meio do pino D_{IN}. Para isso, a BBB envia a informação (bit a bit), na borda de subida do *clock*, referente ao início da conversão, ao modo de operação (*single-ended* ou *differential*, de acordo com o

datasheet) e ao canal de entrada analógica em que o sinal deve ser convertido. Quando todos os bits de configuração são emitidos, a BBB começa a receber do conversor o resultado da conversão (um bit por vez a cada borda de descida do *clock*). Após receber todos os bits correspondentes ao resultado da conversão, o programa salva o resultado em um espaço de memória compartilhada (inicialmente 0x00000000) entre o servidor Linux e a PRU. Em seguida, o pino \overline{CS} é colocado em nível alto (desabilita a conversão) e um registrador recebe o valor para configuração do próximo canal a ser convertido. Caso a conversão tenha ocorrido em todos os 8 canais, o programa finaliza. Se ainda houverem canais em que a conversão precisa ser realizada, o processo se repete a partir da habilitação da conversão (pino \overline{CS} em nível baixo).

Figura 15 – Diagrama de fluxo (programa da PRU0).



Fonte: o próprio autor.

Embora, na Figura 15, seja apresentada apenas a execução do programa da PRU0, esse programa também se encarrega de enviar um evento de interrupção para o programa em C do servidor Linux (descrito no próximo tópico). Esta interrupção ocorre depois que certo número de amostras é salvo na memória da PRU0.

4.2.3 IMPLEMENTAÇÃO DO CÓDIGO EM C

O programa em C é utilizado para que o arquivo binário (.bin) – gerado quando o programa em *assembly* (.p) é compilado – seja transferido para o PRU-ICSS. Além disso, ele atua como ponte de comunicação entre a PRU e o Linux (MOLLOY, 2015).

Existem determinadas funções que são utilizadas para toda interação entre o programa do servidor Linux (.c) e o programa da PRU. Muitas destas funções estão presentes na biblioteca PRUSSDRV e são usadas, entre outras coisas, para: controle básico da PRU (habilitar/desabilitar/resetar); auxiliar (carregar e executar código na PRU); fazer mapeamento da memória; e, fazer o gerenciamento de eventos (por exemplo, gerar interrupções, esperar pela ocorrência de algum evento). Uma descrição detalhada das funções presentes na biblioteca citada pode ser conferida em (Texas Instruments, 2015).

Além das funções descritas anteriormente, o código em C faz uso de *POSIX threads* (*Pthreads*), o qual corresponde a um conjunto de instruções, tipos e constantes que tornam capaz a implementação de *threads* nas aplicações em C da BBB (MOLLOY, 2015). Deste modo, uma parte do código é executada, aparentemente, em paralelo.

No sistema de aquisição de dados, o código na função *thread* cria um arquivo (.txt) e espera um evento de interrupção gerado pela PRU0. Quando este evento ocorre, o arquivo é aberto e os dados armazenados na memória da PRU0 são salvos no mesmo. Esses dados são acessados através de uma função de mapeamento da memória. Por fim, o arquivo é fechado e o evento de interrupção é “limpo”.

O número de dados salvos no arquivo (.txt) depende da quantidade de amostras que se deseja analisar.

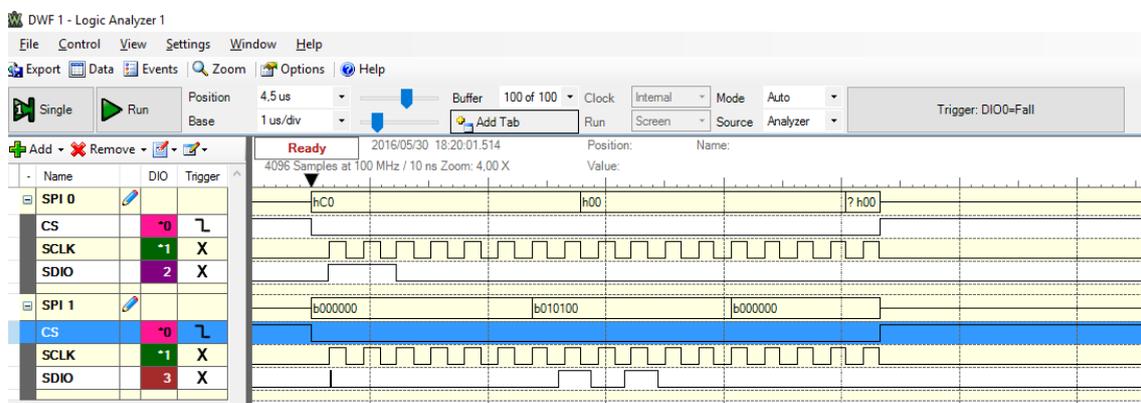
5 RESULTADOS E DISCUSSÃO

Esta seção apresenta os resultados obtidos durante os testes do sistema de aquisição de dados. Os testes aqui apresentados fizeram uso de todos os canais do ADC.

5.1 COMUNICAÇÃO ENTRE A BBB E O MCP3008

A comunicação entre a BBB e o conversor A/D foi verificada por meio da utilização do analisador lógico do dispositivo *Analog Discovery* da *Digilent*. Para este teste, o conversor foi alimentado com 3,3 V (V_{DD}) da *BeagleBone Black* e com uma tensão de referência igual a 3,3 V (V_{REF}). No canal CH0, foi aplicada uma tensão igual a 2 V e a conversão foi feita utilizando um programa na PRU semelhante àquele descrito na seção 4.2.2 desse trabalho. O resultado do teste pode ser verificado na Figura 16. Nesta, podem ser observados os sinais de *clock* e *chip select*, bem como, a informação de configuração do conversor enviada pela BBB ($0xC0 = 0b11000000$) e o resultado da conversão no pino D_{OUT} ($0b101000000$). Os 10 bits de resultado apresentam um valor em decimal igual a 640, o que corresponde a uma tensão de aproximadamente 2,064 V ($640 * 3,3 / 1023$). Sendo assim, o resultado obtido apresentou um erro de 3,2%. Levando-se em consideração que a tensão da fonte não foi verificada como sendo exatamente 2 V, este erro pode ter sido ainda menor. Além disso, verificou-se que a BBB foi capaz de coletar todos os bits enviados pelo conversor A/D.

Figura 16 – Comunicação entre a BBB e o MCP3008.



Fonte: o próprio autor.

5.2 CALIBRAÇÃO PARA O CONDICIONAMENTO DOS SINAIS DE TENSÃO E CORRENTE

Antes de serem efetuados os testes com valores de tensão na ordem de 220 V_{rms} e correntes acima de 1 A, o sistema de medição (circuitos de condicionamento de tensão e corrente) foi calibrado a fim de serem obtidos resultados mais precisos. Sendo assim, a tensão de entrada do canal CH0 foi calibrada do seguinte modo:

- Aplicou-se uma tensão de 0 V na entrada do circuito de condicionamento;
- Um programa da PRU foi utilizado para obtenção de 160 amostras do sinal na entrada CH0;
- O programa em C encarregado da execução do programa da PRU fez o cálculo da média dos valores das 160 amostras;
- O resultado (decimal) foi salvo em um arquivo (.txt);
- Verificou-se o arquivo (.txt) e o valor foi anotado.
- Os procedimentos descritos anteriormente foram repetidos para um valor de tensão igual a 30 V.
- Para obtenção da constante de calibração do sistema de medição de tensão, foi feito o seguinte cálculo:

$$\alpha_V = \frac{30}{(v_D - v_0) * \left(\frac{V_{m\acute{a}x}}{v_{0c}}\right)}. \quad (1)$$

Na equação (1), v_D é o valor medido para 30 V, em decimal; v_0 é o valor medido para 0 V, em decimal; $V_{m\acute{a}x}$ é o valor máximo de tensão em volts; v_{0c} é o valor em decimal calculado para 0 V, ou seja, corresponde ao valor esperado caso não necessitasse ser feita uma calibração.

O valor da constante de calibração foi utilizado para o cálculo das tensões no circuito apresentado na próxima seção. A equação utilizada foi a seguinte:

$$V_m = \alpha_V * (v_D - v_0) * \left(\frac{V_{m\acute{a}x}}{v_{0c}}\right) \quad (2)$$

onde V_m corresponde ao valor da tensão medida pelo sistema de aquisição desenvolvido, em volts.

Para a calibração do sistema de medição de corrente, foi efetuado um método análogo àquele apresentado para a medição de tensão. Porém, foi utilizado um fio de comprimento considerável e aplicada uma tensão através do mesmo de modo a se obter uma corrente de 0 A e, em seguida, uma corrente de 1 A. A constante de calibração foi então obtida pela fórmula:

$$\alpha_I = \frac{1}{(i_D - i_0) * \left(\frac{I_{máx}}{i_{0c}}\right)}. \quad (3)$$

Na equação (3), i_D é o valor medido para 1 A, em decimal; i_0 é o valor medido para 0 A, em decimal; $I_{máx}$ é o valor máximo de corrente em ampéres; i_{0c} é o valor em decimal calculado para 0 A, ou seja, corresponde ao valor esperado caso não necessitasse ser feita uma calibração.

Na próxima seção, será apresentado o circuito montado para a aquisição de dados. A constante de calibração do sistema de medição de corrente foi utilizada para que pudessem ser calculados os valores de corrente neste circuito. Os cálculos foram efetuados por meio da equação:

$$I_m = \alpha_I * (i_D - i_0) * \left(\frac{I_{máx}}{i_{0c}}\right) \quad (4)$$

onde I_m corresponde ao valor de corrente medido pelo sistema de aquisição desenvolvido, em ampéres.

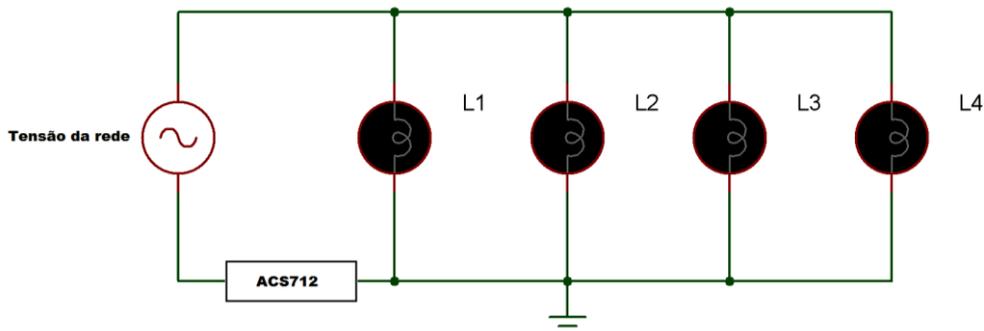
As equações (2) e (4) foram inseridas no programa em C encarregado da execução do programa da PRU. Como o código em C também salva os valores dos dados da aquisição em um arquivo de texto, as equações foram utilizadas para que os resultados presentes no arquivo estivessem nas unidades de tensão (V) e corrente (A).

5.3 CIRCUITO COM LÂMPADAS INCANDESCENTES

O circuito da Figura 17 foi montado a fim de ser efetuada a aquisição de dados de tensão e corrente (taxa de amostragem de aproximadamente 52,8 ksp/s). O circuito foi alimentado pela tensão de uma fonte programável (220 V_{RMS}), a qual foi conectada ao circuito de condicionamento de tensão para que pudesse ser convertida no canal CH0 do

MCP3008. Neste circuito, foram utilizadas quatro lâmpadas incandescentes (Figura 18) que totalizaram uma potência de 240 W. O sensor de corrente ACS712 foi conectado em série com o circuito e seu sinal de saída foi conectado ao circuito de condicionamento de corrente para, então, ser convertido no canal CH1 do conversor. Esse circuito foi considerado como sendo puramente resistivo.

Figura 17 – Circuito para medição de tensão e corrente.



Fonte: o próprio autor.

Figura 18 – Circuito com quatro lâmpadas incandescentes.

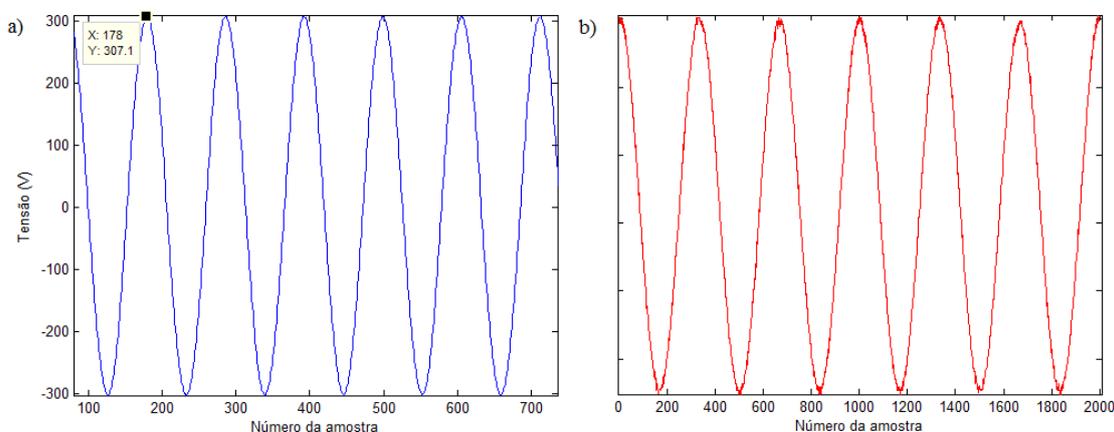


Fonte: o próprio autor.

O resultado da conversão foi salvo em um arquivo (.txt) e transferido da BBB para um computador, onde os dados puderam ser manipulados. O *software* MATLAB foi utilizado para auxiliar na plotagem das curvas de tensão e corrente a partir dos valores obtidos pelo sistema de aquisição de dados desenvolvido (sem tratamento filtragem matemática) e pelo osciloscópio.

Na Figura 19 são apresentadas as curvas de tensão obtidas com o sistema de aquisição de dados desenvolvido (Fig. 19a) e com o osciloscópio (Fig. 19b). Como podem ser observadas, as formas de onda apresentaram comportamentos semelhantes e o valor da tensão de pico se aproximou de 311 volts ($220 V_{rms}$). Também pode ser verificada, a presença de ruído nos sinais amostrados. Uma vez que toda montagem foi feita utilizando *protoboard* e considerando que as tensões de alimentação proveniente das fontes de tensão (alimentação dos amplificadores e tensão de referência 3,3 V) não foram totalmente filtradas, o ruído apresentado no sinal se torna justificável.

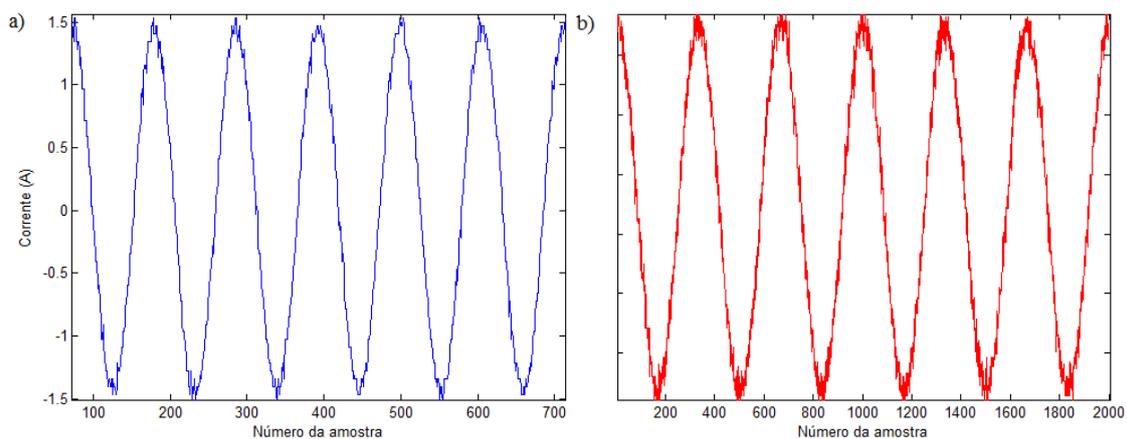
Figura 19 – Curvas de tensão: (a) sistema de aquisição de dados desenvolvido e (b) osciloscópio.



Fonte: o próprio autor.

As curvas de corrente construídas a partir dos dados coletados pelo sistema de aquisição de dados desenvolvido (Figura 20a) e pelo osciloscópio (Figura 20b) apresentaram formas de onda com comportamento semelhante. Porém, a presença de sinais ruidosos ficou bem mais evidente (se comparados com as curvas de tensão). Embora o sistema de medição de corrente tenha sido calibrado, o sensor de corrente ACS712, algumas vezes, apresenta dificuldades em sua utilização. A medição de corrente, assim como no caso da tensão, foi feita com o circuito montado em *protoboard* e as tensões de alimentação não foram totalmente filtradas. Esses fatores podem ter contribuído para presença de ruídos nos sinais medidos.

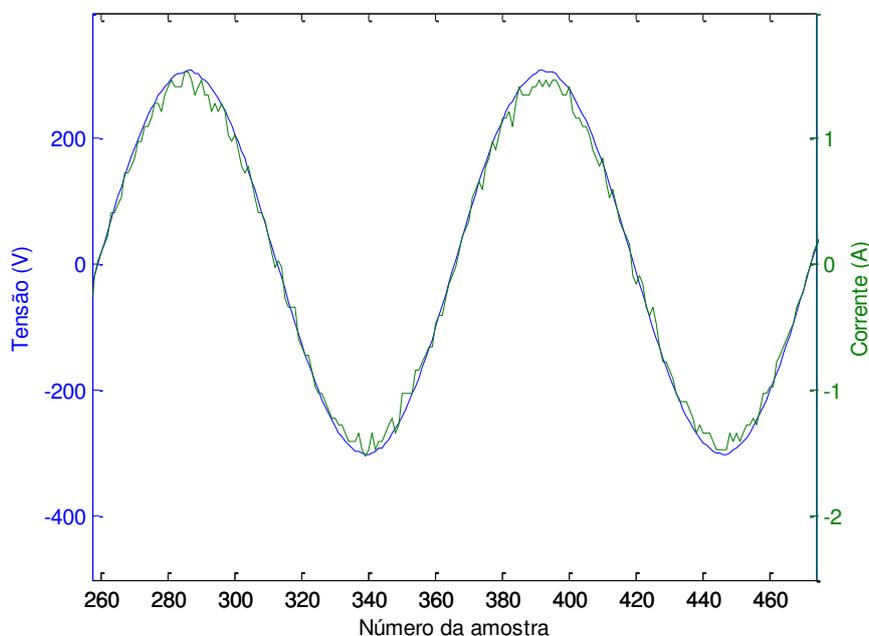
Figura 20 – Curvas de corrente: (a) sistema de aquisição de dados desenvolvido e (b) osciloscópio.



Fonte: o próprio autor.

Na Figura 21, as curvas de tensão e corrente, obtidas a partir dos dados colhidos pelo sistema de aquisição desenvolvido, podem ser observadas. Uma vez que a carga utilizada para o teste foi resistiva, as curvas mostram sinais de tensão e corrente com diferença de fase muito pequena (desconsiderando os ruídos no sinal de corrente). Em circuitos com carga puramente resistiva, os sinais de tensão e corrente apresentam mesma fase, o que faz com que não haja diferença de fase.

Figura 21 – Curvas de Tensão e corrente obtidas pelo sistema de aquisição desenvolvido.



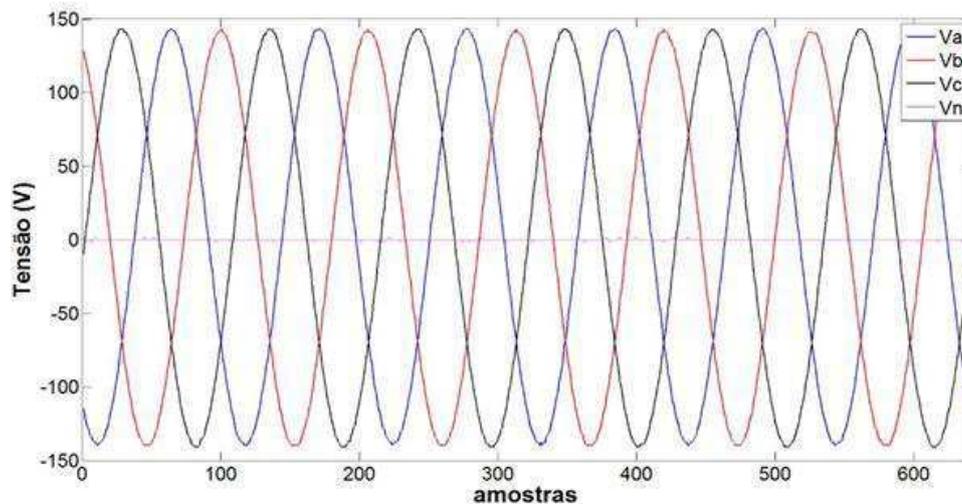
Fonte: o próprio autor.

5.4 SISTEMA TRIFÁSICO COM LÂMPADAS INCANDESCENTES

Com o intuito de testar a aquisição em todos os canais do conversor A/D, uma fonte programável foi utilizada para que fossem gerados três sinais de tensão ($100 V_{RMS}$) defasados de 120° . Cada um dos sinais de tensão foi aplicado a um circuito semelhante àquele apresentado na Figura 16. Um dos circuitos (correspondente à fase C) apresentou duas lâmpadas queimadas e por este motivo a amplitude do sinal de corrente foi menor do que a das outras duas fases. Além disso, foi realizada a aquisição dos sinais de tensão de neutro e corrente de neutro.

Na Figura 22 podem ser verificadas as curvas correspondentes às tensões de fase e de neutro obtidas pelo sistema desenvolvido. Observa-se que as formas de onda estão defasadas de 120 graus e que a tensão de neutro apresentou um valor muito pequeno.

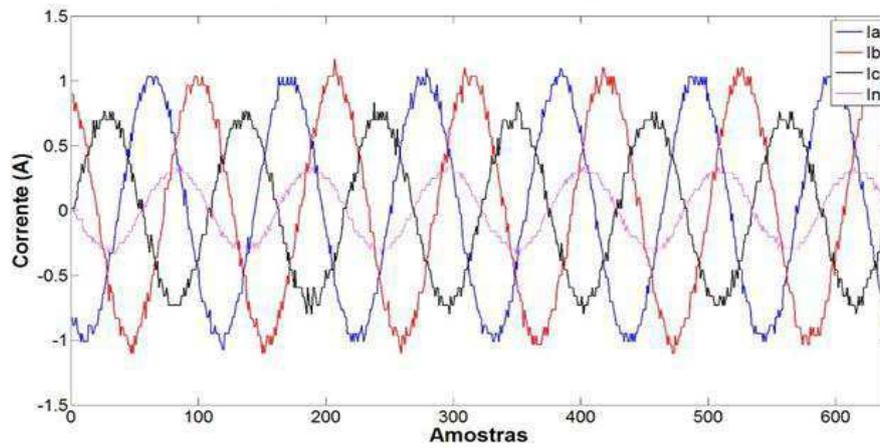
Figura 22 – Tensões de fase ($100 V_{RMS}$) obtidas com o sistema desenvolvido.



Fonte: o próprio autor.

As curvas das correntes de fase e de neutro obtidas por meio do sistema de aquisição desenvolvido são apresentadas na Figura 23. Podem ser observadas a presença de ruído em todos os sinais e uma amplitude menor no sinal da fase C (como discutido anteriormente).

Figura 23 – Correntes de fase com 100 V_{RMS} obtidas com o sistema desenvolvido.



Fonte: o próprio autor.

Além dos testes apresentados nessa seção, foram realizados outros testes menores a fim de validar o uso da BBB em conjunto com o ADC.

6 CONCLUSÃO

Neste trabalho, um sistema de aquisição de dados, baseado na *BeagleBone Black*, foi desenvolvido para aplicações genéricas em sistemas de potência. Nos testes realizados, constatou-se que o sistema foi capaz de fazer a aquisição e o armazenamento dos dados provenientes de canais do conversor A/D.

O programa em *assembly* executado na PRU0 da BBB mostrou-se eficaz no controle do ADC e no armazenamento dos resultados na memória compartilhada com o servidor Linux. Além disso, o programa em C gerenciou a PRU e o arquivo (.txt) com sucesso. O uso da PRU se mostrou necessário, uma vez que a versão do Linux instalada na BBB não permite aplicações em tempo real.

Ficou evidenciado, pela análise das curvas do último teste, que o sistema desenvolvido se comportou de maneira satisfatória, pois, quando comparados os dados colhidos pelo sistema e pelo osciloscópio, observou-se um comportamento similar.

No que se refere ao ADC, o MCP3008 apresentou-se como uma boa alternativa de custo-benefício. Esse ADC se mostrou suficiente para a aplicação, porém um conversor com maior resolução seria capaz de atender mais aplicações em Sistemas Elétricos, como por exemplo, para medição de parâmetros de qualidade de energia, os quais exigem um conversor com resolução maior ou igual a 12 bits.

Como trabalhos futuros, sugere-se: avaliar a onda quadrada com tempo de subida na ordem de nanosegundos; avaliar o coeficiente de determinação (R^2) entre o osciloscópio e o sistema; comparar o ACS712 com resistor *shunt* ou pinça amperimétrica.

REFERÊNCIAS

ALLEGRO MicroSystems. **Datasheet: ACS712**, 2012.

ATWELL, C. **The Biggest-Little Revolution: 10 Single-Board Computers for Under \$100**, 2013. Disponível em: <http://www.eetimes.com/document.asp?doc_id=1319262>. Acesso em: 22 Maio 2016.

BEAGLEBOARD.ORG. **BeagleBone: open-hardware expandable computer**, 2016. Disponível em: <<http://beagleboard.org/support/bone101>>. Acesso em: 22 Maio 2016.

BRAUN, P. A.; MOSS, R. H.; STIGALL, P. D. The Single Board Computer: A Practical Alternative to Custom Designed Microsystem. **Computers and Electrical Engineering**, 12, 1986. 109-118.

CATSOULIS, J. **Designing Embedded Hardware**. 2ª. ed. [S.l.]: O'Reilly Media, 2005. ISBN 978-0-59600-755-3.

COOPER, J. **Introduction to the BeagleBone Black Device Tree**, 2014. Disponível em: <<https://cdn-learn.adafruit.com/downloads/pdf/introduction-to-the-beaglebone-black-device-tree.pdf>>. Acesso em: 30 maio 2016.

MANO, M. M.; KIME, C. R.; MARTIN, T. **Logic and Computer Design Fundamentals**. 5ª. ed. [S.l.]: Pearson, 2015. ISBN 978-0-133-76063-7.

MICROCHIP Technology Inc. **Datasheet MCP3004/3008**, 2008.

MOLLOY, D. **Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux**. [S.l.]: Wiley, 2015. ISBN 978-1-118-93512-5.

MORRIS, A. S. **Measurement & Instrumentation Principles**. 3ª. ed. [S.l.]: Butterworth-Heinemann, 2001. ISBN 978-0-750-65081-6.

ORTMEYER, C. Then and Now: A Brief Story of Single Board Computers. **Electronic Design Uncovered**, n. 6, 2014.

PALLÁS-ARENY, R.; WEBSTER, J. G. **Analog Signal Processing**. [S.l.]: Wiley-Interscience, 1999. ISBN 978-0-471-12528-0.

PALLÁS-ARENY, R.; WEBSTER, J. G. **Sensor and Signal Conditioning**. 2ª. ed. [S.l.]: Wiley, 2000. ISBN 978-0-471-33232-9.

ROSCH, W. L. **The Winn L. Rosch Hardware Bible**. 6ª. ed. [S.l.]: Que Publishing, 2003. ISBN 978-0-789-72859-3.

SANTOS, R.; PERESTRELO, L. M. C. **BeagleBone for Dummies**. [S.l.]: For Dummies, 2015. ISBN 978-1-118-99291-3.

SEDRA, A. S.; SMITH, K. C. **Microelectronic Circuits**. 7ª. ed. New York: Oxford University Press, 2014. ISBN 978-0-199-33913-6.

TEXAS Instruments. **PRU Linux Application Loader API Guide**, 2015. Disponível em: <http://processors.wiki.ti.com/index.php/PRU_Linux_Application_Loader_API_Guide>. Acesso em: 22 Maio 2016.

WINDER, S. **Analog and Digital Filter Design**. 2ª. ed. [S.l.]: Newnes, 2002.