

**CELSO MACHADO MAIA PADILHA**

**SISTEMA PARA AQUISIÇÃO DE SINAIS DE TENSÃO E CORRENTE  
UTILIZANDO A PLATAFORMA BEAGLEBONE BLACK**

Dissertação de mestrado apresentado ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador:

Prof. Dr. Fabiano Salvadori

Co-orientadora:

Profa. Dra. Camila Seibel Gehrke

**JOÃO PESSOA**

**2015**

**CELSO MACHADO MAIA PADILHA**

**SISTEMA PARA AQUISIÇÃO DE SINAIS DE TENSÃO E CORRENTE  
UTILIZANDO A PLATAFORMA BEAGLEBONE BLACK**

Dissertação de mestrado apresentado ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Fabiano Salvadori  
Co-orientadora:  
Profa. Dra. Camila Seibel Gehrke

**JOÃO PESSOA**

**2015**

## FICHA CATALOGRÁFICA

P123s Padilha, Celso Machado Maia.  
Sistema para aquisição de sinais de tensão e corrente utilizando a  
plataforma Beaglebone Black / Celso Machado Maia Padilha. – João  
Pessoa, 2015.

92f. : il.

Orientador: Fabiano Salvadori

Coorientadora: Camila Seibel Gherke

Dissertação (Mestrado) – UFPB/CEAR

1. Engenharia Elétrica. 2. Plataformas de desenvolvimento. 3.  
Beaglebone Black. 4. Raspberry. 5. Redes inteligentes.

UFPB/BC  
(043)

CDU: 621.3

**UNIVERSIDADE FEDERAL DA PARAÍBA – UFPB**  
**CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS – CEAR**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGE**

A Comissão Examinadora, abaixo assinada, aprova a Dissertação de Mestrado

**SISTEMA PARA AQUISIÇÃO DE SINAIS DE TENSÃO UTILIZANDO A  
PLATAFORMA BEAGLEBONE BLACK**

Elaborado por

**CELSONO MACHADO MAIA PADILHA**

como requisito parcial para obtenção do grau de  
**Mestre em Engenharia Elétrica.**

**COMISSÃO EXAMINADORA**

**Prof. Dr. Fabiano Salvadori – Orientador**

**Profa. Dra. Camila Seibel Gherke – Coorientadora**

**Prof. Dr. Alexandre Cunha Oliveira – Examinador Externo (DEE/UFCG)**

**Prof. Dr. Isaac Soares de Freitas – Examinador Interno (DEE/UFPB)**

João Pessoa/PB, 27 de novembro de 2015

A minha família, meus professores, amigos e principalmente a filha Gabriela

Dedico

## **AGRADECIMENTOS**

Primeiramente a Deus por me providenciar a oportunidade de viver esse e todos os outros momentos, agradeço especialmente a minha esposa Taisa e minha filha Gabriela, também agradeço a minha atual família pelo suporte e motivação, Marcus, Rosangela, Mayara e Maya.

Em especial agradeço aos professores Dr. Fabiano Salvadori e Dra. Camila Seibel que me deram suporte durante a jornada do mestrado, não só me proporcionando conhecimentos científicos, mas também proporcionando ensinamentos para a vida.

Aos meus colegas de pós-graduação que tornaram um período de longa dedicação em algo divertido.

“Pouco conhecimento faz com que as pessoas se sintam orgulhosas. Muito conhecimento, que se sintam humildes. É assim que as espigas sem grãos erguem desdenhosamente a cabeça para o Céu, enquanto que as cheias as baixam para a terra, sua mãe.”

Leonardo da Vinci.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>17</b>
1.1	OBJETIVO.....	21
1.2	ORGANIZAÇÃO DO TEXTO.....	22
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA.....</b>	<b>25</b>
2.1	PLATAFORMAS DE DESENVOLVIMENTO .....	25
2.1.1	Arduino Yun® .....	26
2.1.2	Raspberry Pi® .....	28
2.1.3	Intel Galileo® .....	29
2.1.4	Beaglebone Black® .....	30
2.2	LEVANTAMENTO DAS ESPECIFICAÇÕES ENTRE AS PLATAFORMAS .....	31
2.2.1	Especificações Gerais .....	32
2.2.2	Especificações de entradas e saídas .....	33
2.3	TESTE DE BENCHMARK ENTRE AS PLATAFORMAS.....	34
2.4	CONSUMO DOS MÓDULOS.....	36
2.5	DESENVOLVIMENTO DE SOFTWARE DAS PLATAFORMAS .....	37
2.5.1	Arduino Yun® .....	37
2.5.1.1	Sistema Operacional.....	37
2.5.1.2	Arquitetura de comunicação.....	37
2.5.1.3	Desenvolvimento de Software.....	38
2.5.2	Raspberry Pi® .....	39
2.5.2.1	Sistema Operacional.....	39
2.5.2.2	Arquitetura da plataforma Raspberry Pi® .....	39
2.5.2.3	Desenvolvimento.....	40
2.5.3	Intel Galileo® .....	41
2.5.3.1	Arquitetura da plataforma Intel Galileo® .....	41
2.5.3.2	Desenvolvimento.....	42
2.5.4	Beaglebone Black® .....	43
2.5.4.1	Sistema Operacional.....	43
2.5.4.2	Arquitetura da Beaglebone Black® .....	43
2.5.4.3	Desenvolvimento de Software.....	44



---

<b>2.5.5 Conclusão sobre as plataformas de desenvolvimento.....</b>	<b>45</b>
2.6 BEAGLEBONE BLACK / PRU.....	46
<b>3 SISTEMA PROPOSTO .....</b>	<b>57</b>
3.2 SOFTWARE E HARDWARE .....	60
<b>4 RESULTADOS.....</b>	<b>65</b>
4.1 EQUIPAMENTOS UTILIZADOS PARA ESTUDO COMPARATIVO .....	65
4.2 EXPERIMENTO .....	65
4.3 TESTE COM SINAIS DE TENSÃO E CORRENTE DA REDE ELÉTRICA .....	68
<b>4.3.1 Sinais de Tensão .....</b>	<b>69</b>
<b>4.3.2 Sinais de Corrente.....</b>	<b>71</b>
4.4 EXPERIMENTO COMPARATIVO.....	74
<b>5 CONCLUSÕES .....</b>	<b>80</b>
5.1 SUGESTÕES PARA TRABALHOS FUTUROS .....	81
<b>REFERÊNCIAS.....</b>	<b>82</b>

## LISTA DE ILUSTRAÇÕES

<b>Figura 1.1 Smart Grid Proposal [5] .....</b>	<b>19</b>
<b>Figura 2.1 - Arduino<sup>®</sup> Yun [14].....</b>	<b>27</b>
<b>Figura 2.2 - Raspberry Pi<sup>®</sup> model A, B [34] .....</b>	<b>29</b>
<b>Figura 2.3 - Intel Galileo<sup>®</sup> [30].....</b>	<b>30</b>
<b>Figura 2.4 - Beaglebone Black<sup>®</sup> [19] .....</b>	<b>31</b>
<b>Figura 2.5 – NBENCH test graph[14] .....</b>	<b>35</b>
<b>Figura 2.6 - NBENCH testing consumption[14] .....</b>	<b>36</b>
<b>Figura 2.7 - Arduino Yun<sup>®</sup> Architecture [14], [40] .....</b>	<b>38</b>
<b>Figura 2.8 - Raspberry Pi<sup>®</sup> Platform Architecture [48].....</b>	<b>40</b>
<b>Figura 2.9 - Intel Galileo<sup>®</sup> platform architecture[50] .....</b>	<b>42</b>
<b>Figura 2.10 - Beaglebone Black<sup>®</sup> ARM Processor Architecture [43].....</b>	<b>44</b>
<b>Figura 2.11 – SoC ARM Architecture [55, 56].....</b>	<b>47</b>
<b>Figura 2.12 – ARM Architecture with PRU [55,56] .....</b>	<b>48</b>
<b>Figura 2.13 – PRU block diagram [56] .....</b>	<b>49</b>
<b>Figura 2.14 - PRU I/O Interface [56].....</b>	<b>50</b>
<b>Figura 2.15 - ARM X PRU [56].....</b>	<b>50</b>
<b>Figura 3.1 – Sistema proposto completo .....</b>	<b>58</b>
<b>Figura 3.2 - Proposta de Leitura.....</b>	<b>59</b>
<b>Figura 3.3 - SoC Software Architecture .....</b>	<b>60</b>
<b>Figura 3.4 - Programação em nível de Software e Hardware.....</b>	<b>61</b>
<b>Figura 3.5 - Diagrama do Software a ser desenvolvido .....</b>	<b>62</b>
<b>Figura 3.6 - Diagrama básico do módulo de aquisição de tensão .....</b>	<b>63</b>
<b>Figura 3.7 - Módulo de Aquisição e condicionamento de sinais .....</b>	<b>63</b>
<b>Figura 4.1 - Testes Beaglebone Black<sup>®</sup> .....</b>	<b>66</b>
<b>Figura 4.2 - Testes da Tabela 4 .....</b>	<b>67</b>
<b>Figura 4.3 - Ciclos de Estabilização.....</b>	<b>67</b>
<b>Figura 4.4 - Formas de onda resultante dos sinais fornecidos pela plataforma Beaglebone Black<sup>®</sup> .....</b>	<b>68</b>
<b>Figura 4.5 - Tensão de saída fornecida pelo módulo de aquisição e condicionamento de sinais .....</b>	<b>70</b>
<b>Figura 4.6 – Tensão de Entrada do ADC da plataforma.....</b>	<b>70</b>
<b>Figura 4.7 – Sinal de tensão fornecido pela Beaglebone Black<sup>®</sup> .....</b>	<b>71</b>
<b>Figura 4.8 - Tensão e corrente de saída fornecida pelo módulo de aquisição e condicionamento de sinais .....</b>	<b>72</b>
<b>Figura 4.9 - Forma de onda gerada pelo sinal de Corrente armazenado no ARM.....</b>	<b>73</b>
<b>Figura 4.10 - Curva de Tensão e Corrente.....</b>	<b>73</b>
<b>Figura 4.11 - Análise do experimento com o multímetro e alicate amperímetro .....</b>	<b>74</b>

---

<b>Figura 4.12 - Experimento.....</b>	<b>75</b>
<b>Figura 4.13 - Tensão de Saída BBB / Multímetro Hikari / Multímetro Keysight..</b>	<b>77</b>
<b>Figura 4.14 - Comparativo entre o sinal da Beaglebone e os equipamentos de análise com relação a corrente.....</b>	<b>78</b>

---

## LISTA DE TABELAS

<b>Tabela 1 – Especificações gerais [13], [14], [20], [21], [24] - [36] .....</b>	<b>32</b>
<b>Tabela 2 – Especificações de entradas e saídas [13], [14], [20], [21], [24] - [36] .....</b>	<b>33</b>
<b>Tabela 3 - Resultados do teste NBENCH [14] .....</b>	<b>35</b>
<b>Tabela 4 - Resumo do teste com 5 canais analógicos .....</b>	<b>66</b>
<b>Tabela 5 - Comparativo da Beaglebone Black<sup>®</sup> com equipamentos de referência .....</b>	<b>75</b>

---

## LISTA DE SIGLAS

ADC – *Analog to Digital Converter*  
ADC – Analógico Digital  
ARM - *Advanced RISC Machine*  
CPU – *Central Processing Unit*  
DSP – *Digital Signals Processor*  
SOC – *System on Chip*  
E/S – Entradas/Saídas  
FPGA – *Field Programmable Gate Array*  
FPU – *Float Point Unit*  
GCC – *GNU Compiler Collection*  
GPU – *Graphic Processing Unit*  
GPL – *General Public License*  
IEEE – *Institute of Electrical And Electronics Engineers*  
IDE – *Integrated Development Environment*  
PC – *Personal Computer*  
PRU – *Programmable Realtime Unit*  
USB – *Universal serial bus*  
SD – *Secure Digital*  
RISC – *Reduced Instruction Set Computer*  
REI – REIs  
SSH – *Secure Shell*  
eMMC – *Embedded Multi-Media Controller*

## RESUMO

No atual cenário mundial, a crise do setor elétrico está cada vez mais evidente, e para se manter um nível de desenvolvimento econômico compatível com este crescimento, muitos países estão investindo no aumento da produção de energia elétrica a partir de diversas fontes, sejam estas renováveis e limpas ou não, para suprir a crescente demanda. A integração dessas novas fontes de energia demanda coordenação em tempo real. Desta forma, unidades confiáveis de monitoramento, processamento e transmissão de dados são necessárias. Neste trabalho propõe-se a implementação de um sistema para aquisição de dados para sinais de tensão e corrente, por meio de uma plataforma de desenvolvimento embarcado de baixo custo associado com foco na programação de hardware, aliado a um módulo de aquisição e condicionamento de sinais para aplicação do conceito de Redes Elétricas Inteligentes - REI (do inglês *Smart Grid*). O sistema é responsável por realizar a leitura da tensão e da corrente fornecidos pela rede elétrica e transformá-los para informações analisáveis pela plataforma de desenvolvimento, possibilitando assim, a manipulação por sistemas supervisórios. Em posse destes dados, os sistemas supervisórios podem, além de analisar, tomar decisões baseadas nas informações fornecidas pela plataforma e utilizá-las para aplicar em diversos conceitos presentes nas REIs, tais como controle do fluxo de energia, minimização dos efeitos da falta da energia elétrica, concepção de geração distribuída a partir de fontes renováveis, identificação de furtos de energia, redução de perdas técnicas, monitoramento da qualidade de energia e outros. Para o desenvolvimento deste sistema foi utilizado a plataforma de desenvolvimento Beaglebone Black<sup>®</sup>, associada a uma unidade de condicionamento de sinais de tensão e corrente desenvolvido pelo Programa de Pós-Graduação em Engenharia Elétrica no Laboratório de Otimização de Sistemas de Energia (LOSE) do Departamento de Engenharia Elétrica (DEE), da Universidade Federal da Paraíba (UFPB). Este módulo é responsável por condicionar os sinais de tensão e corrente fornecidos pela rede elétrica adequando-os às especificações necessárias para leitura e conversão na plataforma de desenvolvimento Beaglebone Black<sup>®</sup>.

**Descritores:** Plataformas de desenvolvimento, Beaglebone Black, Raspberry Pi, Intel Galileo, Arduino, NBENCH, redes inteligentes.

## ABSTRACT

In the current global scenario, the power sector crisis is evident, and to maintain an economic development compatible with this growth, many countries are investing in energy production from many different sources, renewable and clean or not, to supply the growing demand. The integration of these new energy sources demands realtime coordination. Thus, monitoring units need to be reliable besides a good processing and data transmission are needed. In this paper it's proposed to implement a data acquisition system for current and voltage signals with a low cost embedded platform, focused in hardware programation coupled with the acquisition and conditioning module to apply a Smart Grid concept. This system are responsible for performing analysis of voltage and current provided from electrical grid and transform these informations to readable informations for embedded platform, making possible manipulate this informations from surpevisory systems. With these data, the supervisory systems can analyze and make decisions based on information provided by the platform and use to apply in different concepts present in Smart Grid, such as energy flow control, minimizing the effects of electricity lack, design of distributed generation from renewable sources, identification of energy theft, reduction of technical losses, power monitoring and others. To develop this system was used a Beaglebone Black<sup>®</sup> development platform, associated with a module for conditioning unit of voltage and current signals, that module was developed by the Optimization Lab Power Systems program (LOSE) of the Electrical Engineering Department (DEE) from Federal University of Paraíba (UFPB). This module is responsible for conditioning the signal voltage and current supplied by the power grid adapting them to the specifications required for reading and conversion in Beaglebone Black<sup>®</sup> development platform.

Key words: Development platforms, Beaglebone Black, Raspberry Pi, Intel Galileo, Arduino, NBENCH, Smart Grids.

## **1 INTRODUÇÃO**



## 1 INTRODUÇÃO

O crescimento do consumo energético ao longo do século XXI está gerando uma série de problemas, a exemplo da insuficiência da geração de energia elétrica, que não está acompanhando o crescimento da demanda energética além de outros problemas. Muitos desses problemas podem ser atenuados ou até mesmo resolvidos através de uma melhor coordenação da geração, distribuição e consumo de energia elétrica.

Redes Elétricas Inteligentes (REI) é um conceito moderno que atribui a atual rede elétrica uma inteligência, proporcionando benefícios como a melhoria na eficiência, segurança e confiabilidade, através do controle automatizado, aliado com tecnologia de distribuição x fluxo de informações, segundo Gungor [1] e Palensky [2].

O termo rede inteligente ou *Smart Grid*, foi utilizado pela primeira vez em 2005, num artigo publicado por S. Massoud Amim e Bruce F. Wollenberg [3], intitulado de *Toward a Smart Grid*, onde os autores se propõem a resolver problemas tais como o *blackout*, aplicando o conceito de REI e auto recuperação (*Self-Healing*). No entanto, não existe um conceito formalizado e único para definir a REI, o que se constata de acordo com o Ministério de Minas e Energia [4], é que as várias proposições para o conceito de REI em sua grande maioria convergem para um fator comum, que a rede convencional deve estar associada ao uso de elementos digitais e de comunicação nas redes que transportam a energia elétrica.

Além disso, dentro do conceito de REI, foi alterado o paradigma do fluxo unidirecional de potência para um novo paradigma, onde temos um fluxo bidirecional (as unidades consumidores podem passar a ser também unidade geradoras).

Nessa ótica os incentivos em tecnologia para as redes elétricas estão crescendo, visto seu grande potencial econômico e ambiental. Alguns estudos evidenciam que o investimento em REI tem retorno de 1 a 7 vezes maior que o investimento em redes elétricas convencionais [5], [6]. Vale ressaltar que essa conclusão está diretamente relacionada aos parâmetros onde o conceito de REI será aplicado, tais como o custo e vida útil dos equipamentos assim como outras premissas.

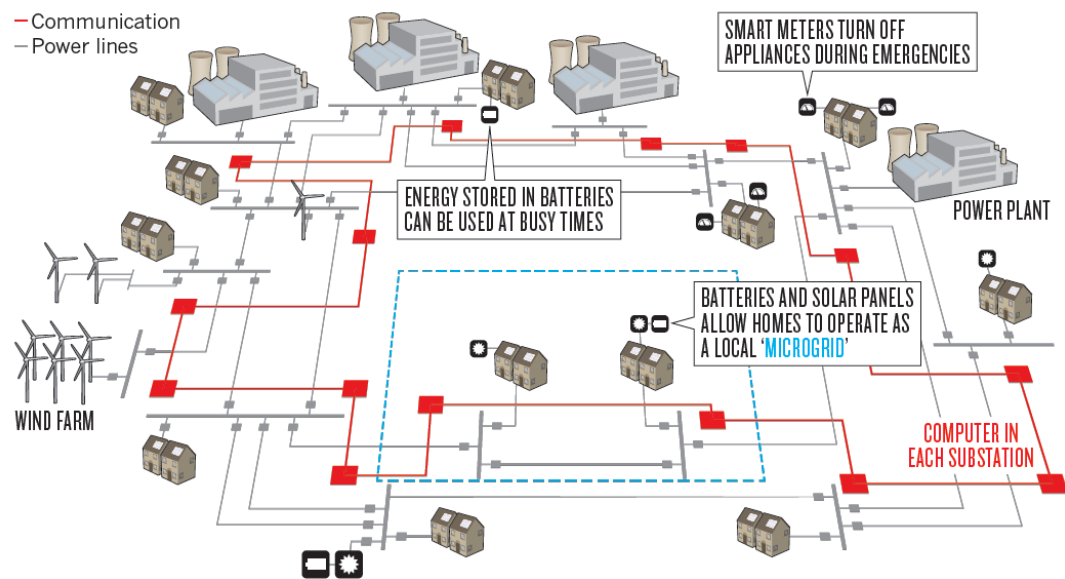
Dentro do domínio das redes elétricas, projeta-se que, mais e mais fontes de energias renováveis estarão presentes e serão utilizadas por clientes residenciais, motivadas por iniciativas dos governos, preocupações ambientais, retorno financeiro, qualidade de vida e controle de energia. Em face desse contexto um dos maiores incentivos é que a geração de energia em ambientes residenciais está próxima dos locais onde ela será consumida, isso implica que, os níveis de perda nas linhas de transmissão são reduzidos significativamente além de aliviar o congestionamento da rede elétrica externa, implicando assim na não necessidade da modernização de todas as linhas de transmissão da cidade ou de um país em uma única etapa, o que amenizaria os custos iniciais da modernização das redes elétricas.

Além destes fatores, a necessidade do gerenciamento do fluxo de potência é evidente, fornecida pela concessionária e gerada pelos consumidores (aqui entendidos como microgeradores), é evidente, e é um problema a ser estudado e resolvido com a aplicação do conceito da REI.

No que se refere a situação energética brasileira, a aplicabilidade dos conceitos de REI é uma necessidade, frente a um consumo cada vez maior e a escassez de recursos financeiros para a ampliação do sistema elétrico. Devido a instabilidade no sistema elétrico, corremos o risco de ver ocorrer uma série de problemas como os apagões e racionamentos, como aconteceu nos anos 2000 e 2001 devido a diminuição da geração de energia das usinas hidroelétricas [8].

O Brasil está investindo e atuando em novos projetos de REI em parceria com empresas de automação e distribuição de energia para a criação de uma nova REI que tem como um dos principais objetivos a auto recuperação em casos de queda, este paradigma é conhecido pelo termo *self-healing*. Com sua utilização é possível obter um serviço de melhor qualidade e confiabilidade para os clientes [9]. Segundo Massoud [5], para a implantação das REI é de fundamental importância melhorar a rede elétrica através da instalação de dispositivos e sistemas acoplados com sensores com alto poder de processamento em toda rede elétrica, o autor ainda explica que a rede de comunicação deve estar interligada como está apresentado na Figura 1.1 [5] .

Figura 1.1 Smart Grid Proposal [5]



Massoud [5] defende claramente que os dados devem ser interligados entre si através de uma rede, e este trabalho tem como foco desenvolver uma proposta de aquisição de dados capaz de realizar a captura dos sinais de tensão e corrente e também se conectar a outros dispositivos externos por meio de uma comunicação, seja esta com fio ou sem fio, fazendo com que o sistema possa ser aplicável no paradigma de REI proposto por Massoud [5].

De acordo com os vários conceitos sobre REI, a utilização de equipamentos capazes monitorar e analisar a energia dentro da rede elétrica é evidente, e de acordo com Bright [51] e Salvadori [52] sistemas embarcados podem ser utilizados e configurados para desempenhar várias tarefas ou papéis, que são de fundamental importância nas REI, tais como:

- Monitoramento do sistema;
- Proteção do sistema;
- Desempenho de atividades de controle central;
- Gerenciamento dos custos de energia;
- Manutenção do nível de tensão;
- Controle do fator de potência;
- Controle dos níveis de geração de energia;
- Controle do carregamento do sistema e outros.

Atualmente no mercado, quando se trata de soluções integradas para monitoramento de tensão, corrente, potência e outros, estes sistemas são em sua grande maioria desenvolvidos por grandes empresas, e são fortemente acoplados em um único produto, o que dificulta na difusão de novas tecnologias a cerca das REI. Outra característica é que quase todos os sistemas são conectados a um computador que se comunica com os dispositivos através da rede local LAN (*Local Area Network*), que em sua grande maioria utilizam cabos de par trançado como aborda Moslehi e Kumar [54], aumentando mais ainda o custo e a complexidade na modernização da rede elétrica.

Este trabalho é uma base inicial para aplicação de plataformas de desenvolvimento embarcadas no conceito das REI. Visto que no atual contexto, as ferramentas e tecnologias disponíveis são de alto valor, tornando assim um problema para aplicação do conceito de REI nas redes convencionais, gerando um impacto negativo para pequenas empresas ou pesquisadores autônomos que não detêm um elevado poder financeiro e tem interesse em investir na modernização das redes elétricas.

Dentro deste contexto, é necessário realizar o desenvolvimento de soluções e ferramentas de baixo custo associado para integração de novas tecnologias nas redes convencionais. Este trabalho está voltado para a utilização de uma plataforma de desenvolvimento embarcado que atende as exigências da ANEEL (Agência Nacional de Energia Elétrica) [18], de tal forma que o sistema possa desempenhar o monitoramento do sistema elétrico, permitindo que diversos sistemas de controle possam acessar a plataforma e consumir as informações para manipular seus respectivos sistemas.

Para isso é necessário que a plataforma apresente uma boa resolução nos sinais de saída, de tal forma que, os valores adquiridos possam ser utilizados para diversas aplicações nas redes elétricas, tais como: gerenciamento do fluxo de potência na rede elétrica e cálculo de tensão e corrente eficaz. A plataforma Beaglebone Black® possui duas unidades programáveis em tempo real (PRU, do inglês *Programmable Realtime Unit*), o que indica que a plataforma possui uma alta capacidade de processamento de dados pois tem dois processadores dedicados apenas para cálculos em tempo real, possibilitando o uso desta plataforma para a aquisição de sinais de tensão/corrente em tempo real.

Por outro lado, a plataforma ainda pode-se considerar recente, por isso, e ainda apresenta um certo grau de dificuldade com relação ao desenvolvimento e aplicação quanto ao uso da PRU, visto que o uso das unidades de processamento em tempo real ainda não são amplamente entendidas e bem definidas em meios acadêmicos e científicos e pelas comunidades de desenvolvimento.

## 1.1 OBJETIVO

Este trabalho tem como proposta desenvolver um sistema de aquisição de sinais de tensão e corrente, utilizando a plataforma de desenvolvimento embarcada Beaglebone Black<sup>®</sup>, por meio do uso da unidade lógica programável em tempo real (PRU). Para realizar a leitura, os sinais fornecidos pela rede são conectados as entradas analógicas do módulo de aquisição e condicionamento de sinais, responsável por emitir os sinais adequados para as entradas analógicas da plataforma. O sistema possibilita a aplicação dos conceitos das REIs e outros paradigmas e funcionalidades, tais como, o gerenciamento do fluxo de potência da rede elétrica. A Beaglebone Black<sup>®</sup> é uma plataforma com sistema operacional baseado em Linux, o que torna este tipo de plataforma interessante para projetos com finalidades como a aqui apresentada. Isto deve-se ao fato que a plataforma possui um sistema operacional embarcado que pode executar diversas tarefas que um computador pessoal faria com mesmo desempenho com custo e tamanho reduzidos. Desta maneira a plataforma poderá atuar como um sistema de controle supervisor, aumentando ainda mais a abrangência de uso da plataforma. Uma vantagem da plataforma é que não é necessário se conectar a um computador externo, transferir os dados, e depois realizar a tomada de decisão. A plataforma pode processar e tomar as decisões baseada nas instruções apresentadas pelo sistema operacional linux, isto faz com que não exista a obrigatoriedade de uma comunicação externa para as tomadas de decisão de acordo com as informações obtidas baseadas na rede, fornecendo autonomia para o sistema.

Este trabalho tem os seguintes objetivos:

1. Levantar as especificações e realizar um estudo comparativo entre as 4 plataformas de desenvolvimento Linux para realizar a aquisição de dados.

As plataformas selecionadas para análise foram: Arduino Yun<sup>®</sup> , Raspberry Pi<sup>®</sup>, Intel Galileo<sup>®</sup> e Beaglebone Black<sup>®</sup>.

2. Analisar características relevantes tais como a arquitetura da plataforma e forma de programação com objetivo de analisar a viabilidade do projeto e constatar que a plataforma é capaz de se adequar aos requisitos do sistema proposto.
3. Propor o desenvolvimento de um sistema de aquisição utilizando a plataforma Beaglebone Black<sup>®</sup> utilizando a biblioteca de acesso a PRU juntamente com o módulo de sinais de tensão e corrente.
4. Analisar os resultados obtidos pelo sistema proposto, afim de comprovar a eficiência do mesmo.

## 1.2 ORGANIZAÇÃO DO TEXTO

De modo a proporcionar uma compreensão facilitada com relação ao assunto tratado, o texto está organizado em 6 capítulos:

1. Introdução;
2. Revisão Bibliográfica;
3. Proposta do sistema;
4. Resultados experimentais;
5. Conclusões;
6. Referências.

No segundo capítulo são apresentadas as especificações a respeito das plataformas de desenvolvimento embarcadas, o capítulo explicita uma bateria de testes comparativos entre as plataformas, com objetivo de analisar e justificar o uso da plataforma escolhida.

O terceiro capítulo visa demonstrar a proposta do trabalho que é aliar a plataforma Beaglebone Black<sup>®</sup> ao módulo de aquisição e condicionamento de sinais.

O quarto capítulo relata os resultados experimentais, obtidos a partir da plataforma de desenvolvimento em três etapas:

- 1 – Aquisição da plataforma de desenvolvimento selecionada utilizando a biblioteca de acesso a PRU sem o uso do

condicionamento de sinais, utilizando apenas equipamentos específicos;

- 2 – Aquisição da plataforma selecionada utilizando a biblioteca de acesso a PRU com uso do módulo de aquisição e condicionamento de sinais, e para finalizar, é apresentado uma comparação dos resultados obtidos com os resultados fornecidos por instrumentos de referência.

O quinto capítulo é reponsável por apresentar as conclusões e trabalhos futuros a cerca deste trabalho.

O sexto capítulo descreve todas as referências bibliográficas utilizadas que permitiram o desenvolvimento deste trabalho.

## **2 REVISÃO BIBLIOGRÁFICA**



## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 PLATAFORMAS DE DESENVOLVIMENTO

Atualmente desenvolvedores e distribuidores de tecnologia estão oferecendo novos produtos, além dos computadores pessoais, atualmente as plataformas de desenvolvimento estão em ampla ascendência e apresentam vantagens que muitos computadores pessoais não têm, o baixo custo. Estas plataformas são unidades microcontroladas que executam um sistema operacional e algumas até mesmo executam a Interface Homem Máquina (IHM). Autores referem-se a essas plataformas como pseudo-computadores, pois podem ser utilizadas como um computador normal.

Empresas estão adotando diversos tipos de plataformas de desenvolvimento para otimizar a produção de seus produtos e serviços nas mais diversas áreas, a exemplo da indústria que está utilizando FPGAs, DSPs e Microprocessadores para controle de robôs, medição de temperatura e outros. As plataformas embarcadas vem se tornando cada vez mais acessíveis e fáceis de manipular, devido a fatores como alto desempenho de processamento, segurança, baixo custo e possibilidade de utilizar ferramentas de *software* já difundidas, a exemplo de ferramentas de manipulação de imagens, que em sua grande maioria é feita para computadores potentes. Possibilitando assim, por exemplo, utilizar estas ferramentas para edição de imagem e vídeo dentro de uma plataforma de baixo custo com suporte a gráficos 3D [7], [15].

Ainda segundo Kickham [7], é possível visualizar que o mercado empresarial está focando no uso dessas plataformas de desenvolvimento em diversas áreas, a exemplo dos sistemas de controle para fins industriais, que estão trocando os computadores por plataformas de desenvolvimento. Ainda segundo Kickham [7], na área da educação, as plataformas são um forte meio de aprendizado, pois é possível executar nas plataformas diversas ferramentas de aprendizado, a exemplo do *software* Scratch que é uma ferramenta que utiliza blocos lógicos, e itens de sons e imagem para desenvolver histórias interativas, estimulando a criatividade e possibilitando o aprendizado de diversos assuntos, dentre estes, desenvolvimento de *software* e lógica de programação.

Visto isso, a difusão de conhecimento a cerca das plataformas de desenvolvimento está evoluindo rapidamente enquanto o custo de mercado das plataformas estão ficando cada vez menores e mais estáveis, tornando o uso de plataformas de desenvolvimento em uma nova tendência.

Foram selecionadas, de acordo com foco na proposta de trabalho, as plataformas com maior ordem de prioridade: rapidez de processamento, capacidade de armazenamento, custo financeiro, portabilidade e linguagem de programação, as escolhidas foram: Arduino Yun<sup>®</sup>, Raspberry Pi<sup>®</sup>, Intel Galileo<sup>®</sup> e Beaglebone Black<sup>®</sup>.

Com objetivo de mensurar o potencial de cada plataforma, é necessário analisar os requisitos e realizar testes comparativos, cuja intenção é analisar e validar o desempenho no processamento de informações de cada uma das plataformas, antes de iniciar o desenvolvimento do sistema.

É de fundamental importância levantar o maior número de características de cada plataforma, para escolher a plataforma de desenvolvimento que mais se adequa a nossa proposta. Em face desta necessidade, Dicola [13] realizou diversos testes comparativos entre as plataformas.

### **2.1.1 Arduino Yun<sup>®</sup>**

Segundo Banzi [10], a Arduino<sup>®</sup> é uma plataforma de desenvolvimento embarcado de código fonte aberto baseado em uma simples placa de entrada e saída, possibilitando a criação de objetos capazes de realizar interações com ambientes e usuários.

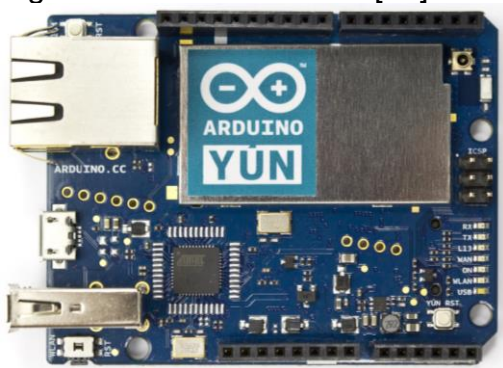
O projeto Arduino existe desde 2005 e possui uma grande variedade de plataformas de desenvolvimento, componentes e sensores disponíveis no mercado que permitem por exemplo o controle de luzes, motores e outros. A linguagem de programação é baseada na linguagem *Processing*, que possui similaridades com as linguagens C/C++, se diferenciando basicamente em métodos e funções já desenvolvidas que auxiliam no desenvolvimento, aumentando o nível de abstração de código da plataforma, tornando a linguagem mais simples e fácil de desenvolver [10], [11].

Algumas vantagens da plataforma Arduino<sup>®</sup> são: baixo custo, o fato de ser multiplataforma, possuir código aberto de *software*, o *hardware* possui

documentação detalhada para livre reprodução, possuir um fórum que é moderado pela equipe oficial do Arduino® dedicado a projetos criados pelos próprios usuários [12].

A plataforma Arduino® selecionada foi o modelo Yun (Figura 2.1) que é baseado no microcontrolador ATmega32u4™. Em termos de *hardware* e capacidade de processamento, a plataforma possui um processador Atheros AR9331 com frequência de 400 MHz que suporta uma distribuição Linux, a distribuição mais recomendada é a OpenWrt também conhecida como OpenWrt-Yun, que é uma adaptação específica para a plataforma [13], [14].

Figura 2.1 - Arduino® Yun [14]



O Arduino Yun® tem a grande vantagem sobre outras plataformas quanto a facilidade de desenvolvimento de código, devido ao uso da linguagem *Processing* que já trás uma série de configurações e parametrizações para o desenvolvedor [13]. Uma das dificuldades da utilização do Arduino Yun® é que a distribuição OpenWrt-Yun depois de instalada, o espaço remanescente utilizável é de apenas 16MB. Por tanto existe um número limitado de ferramentas e utilitários que podem ser utilizados nessa distribuição [13], [14]. Porém para nossa proposta, o trabalho tem a necessidade de análise da rede elétrica em tempo real, quando comparado com outras plataformas a frequência de clock do processador apresenta uma forte desvantagem.

### 2.1.2 Raspberry Pi®

A Raspberry Pi® pode ser considerada como um computador do tamanho de um cartão de crédito, de baixo valor, que pode se conectar a um monitor. A plataforma possui um processador baseado na arquitetura ARM com frequência de 700 MHz podendo chegar até 1 GHz por meio de configuração. Além do disso a Raspberry Pi® permite a utilização de linguagens de fácil entendimento como a linguagem *Python*. A Raspberry Pi® é capaz de fazer quase tudo que se esperaria de um computador pessoal, desde navegar na internet à assistir vídeos em alta definição [16].

A Raspberry Pi® pode operar com diversas distribuições do Linux, sendo a mais conhecida e recomendada, a distribuição Raspbian, que é uma derivação da distribuição Debian específica para a plataforma. A plataforma possui suporte a compiladores de linguagens de programação como o GCC, *Python*, *Ruby* e *Node.js*, porém, não fornece suporte para aplicações com execução de tempo real.

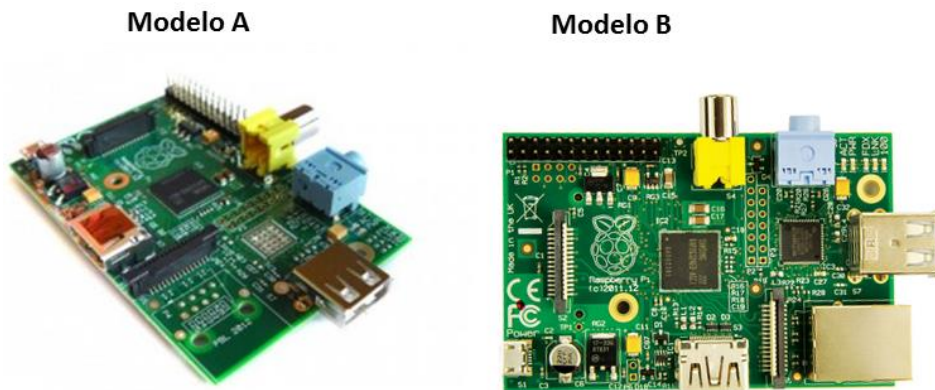
Por outro lado, a Raspberry® possui uma vantagem com relação as outras plataformas de desenvolvimento aqui estudadas, ela possui o maior grupo de desenvolvedores dentre todas as plataformas selecionadas o que facilita no desenvolvimento de soluções de maneira mais ágil [13], [15]. As comunidades são de extrema importância quando se está iniciando os estudos em plataformas de desenvolvimento mais complexas como a Raspberry Pi®, as comunidades são responsáveis por ensinar, resolver problemas, se comunicar diretamente com o fornecedor, aumentando o nível de abstração para o desenvolvimento de aplicações, porém, quando se trata de aplicações mais complexas, as comunidades não são muito relevantes, pois grande parte dela é destinada a usuários iniciantes.

Na comunidade da Raspberry Pi® é possível encontrar diversos tutoriais, canais de tv, canais online, módulos específicos para Android e iOS, maratonas de desenvolvimento com a plataforma e muito mais.

Para Oliveira [17] a Raspberry® é classificada como um SBC – *Single-Board Computer*, ou seja, é um computador embarcado em uma placa, pois possui em um único chip: processador, memória, entradas e saídas, portas USB, vídeo, internet e outros. Atualmente a Raspberry® possui diversos modelos no mercado, porém para objeto de estudo apenas os modelos A e o B presentes na Figura 2.2 serão

testados. Os preços variam entre US\$ 25.00 e US\$ 35.00, a diferença basicamente é que a memória que no modelo B possui o dobro da capacidade de memória do modelo A, e a interface *ethernet* que está disponível apenas no modelo B [15], [17].

Figura 2.2 - Raspberry Pi<sup>®</sup> model A, B [34]

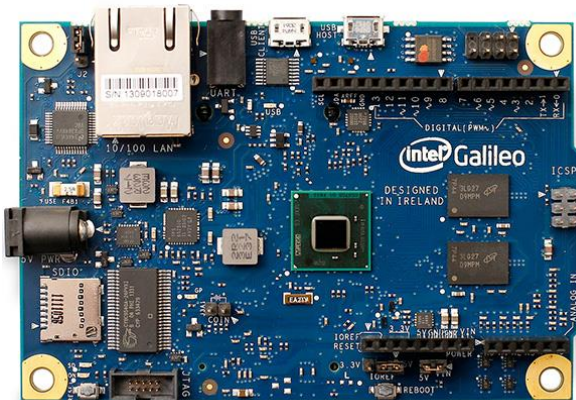


A Raspberry Pi<sup>®</sup> possui uma grande vantagem sobre a plataforma Arduino<sup>®</sup> em termos como: armazenamento interno, interface gráfica e clock. Entretanto o que dificulta o desenvolvimento da nossa proposta, é o fato da plataforma não possuir suporte a aplicações em tempo real e não possuir um ADC interno, implicando na necessidade de utilizar um conversor externo, o que pode diminuir a qualidade e eficiência das aquisições realizadas pela plataforma frente a presença de fatores externos, além do custo extra do conversor [18].

### 2.1.3 Intel Galileo<sup>®</sup>

A plataforma de desenvolvimento Intel Galileo<sup>®</sup>, presente na Figura 2.3 é baseada no sistema Intel<sup>®</sup> SoC Quark X1000 com um processador da classe Pentium<sup>®</sup> de 32 bits. A Intel<sup>®</sup> Galileo foi a primeira plataforma de desenvolvimento embarcada baseada na arquitetura Intel<sup>®</sup> projetada para funcionar como um microcontrolador com entradas e saídas. Inovando, a Intel<sup>®</sup> ofereceu suporte aos *Shields* da plataforma Arduino<sup>®</sup>, o que incentivou os usuários da plataforma Arduino<sup>®</sup> a experimentarem a plataforma com processador de arquitetura Intel<sup>®</sup>, inclusive a plataforma está presente na lista de produtos oficiais do programa *Arduino Certified* [13], [21], [22].

Figura 2.3 - Intel Galileo® [30]



Devido ao fato da plataforma poder operar com os *shields* da plataforma Arduino, o sistema operacional oficial da plataforma é a mesmo utilizado pelo Arduino Yocto.

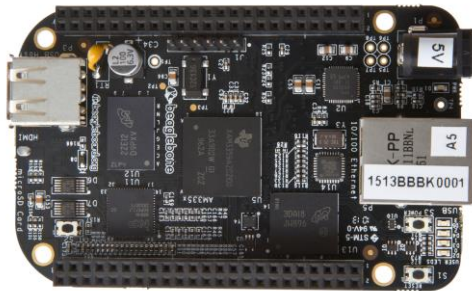
Porém um dos pontos negativos que pode se destacar é que além de ser uma plataforma recente, também é a mais cara, e possui uma pequena comunidade de desenvolvedores [23].

#### 2.1.4 Beaglebone Black®

Assim como as plataformas Arduino®, Intel® e Raspberry® a Beaglebone Black® também é uma plataforma de desenvolvimento de baixo custo, de tamanho reduzido e alto poder de processamento, além de ter uma razoável comunidade de desenvolvedores para suporte [19].

A Beaglebone Black® (Figura 2.4) assim como a Raspberry Pi® também pode ser considerada como um SBC, sendo uma plataforma livre e de código fonte aberto. Em Beagleboard [19] todas as distribuições estão disponíveis para *download* e além disso na própria página tem um manual de instalação e configuração, algumas distribuições mesmo não sendo Linux, como o Android, podem ser instaladas, porém, não são oficiais e não tem o suporte oficial da plataforma. A distribuição oficial mais usada pela comunidade é a Angstrom que já vem pré-configurada para maioria dos recursos disponíveis na plataforma [13], [20].

Figura 2.4 - Beaglebone Black® [19]



A Beaglebone Black® possui grande suporte com relação aos sistemas operacionais que podem ser utilizados, entre eles estão as distribuições Debian, Ubuntu e Angstrom, assim como outros sistemas operacionais embarcados. A plataforma conta com o processador Sitara XAM3359AZCZ100 Cortex A8 de baixo custo com arquitetura ARM de 1.0 GHz fabricado pela empresa *Texas Instruments* [20]. Assim como a Raspberry Pi®, a Beaglebone Black® oferece suporte à instalação de pacotes de compiladores e linguagens de programação em seu sistema operacional como o GCC, *Python*, *Ruby*, *Node.js* e outros [15].

A Beaglebone Black® possui suporte ao desenvolvimento de aplicações em tempo real através da PRU e uma grande quantidade de memória interna, além de uma grande capacidade de armazenamento se comparado as plataformas Arduino® e Intel Galileo®.

## 2.2 LEVANTAMENTO DAS ESPECIFICAÇÕES ENTRE AS PLATAFORMAS

Focado em analisar qual plataforma utilizar, foi necessário realizar um estudo aprofundado sobre as especificações técnicas a respeito de cada plataforma para realizar a tomada de decisão apropriada para o trabalho. Com base na proposta este tópico destina-se a levantar informações relevantes para o domínio das especificações técnicas a respeito de informações como arquitetura do processador, memória, SoC e outros.

### 2.2.1 Especificações Gerais

É necessário descrever e analisar as principais características e especificações de cada plataforma objetivando embasar a escolha de uma plataforma para desenvolver o sistema de aquisição proposto. Sendo assim, este tópico aborda as especificações gerais de cada plataforma tais como SoC, CPU, Arquitetura/Velocidade do processador, Memória, GPU, Armazenamento, Rede, Potência, Dimensões e Peso.

Tabela 1 – Especificações gerais [13], [14], [20], [21], [24] - [36]

	<b>Beaglebone Black<sup>®</sup></b>	<b>Raspberry Pi<sup>®</sup></b>	<b>Arduino Yun<sup>®</sup></b>	<b>Intel Galileo<sup>®</sup></b>
<b>SoC</b>	Texas Instruments AM3358	Broadcom BCM2835	Atheros AR9331	Intel Quark X1000
<b>CPU</b>	ARM Cortex-A8	ARM1176	MIPS32 24K / ATmega32U4	Intel X1000
<b>Arquitetura CPU</b>	ARMv7	ARMv6	MIPS e AVR	i586
<b>Velocidade CPU</b>	1ghz	700mhz	400mhz(AR9331) 16mhz(ATmega)	400mhz
<b>Memória</b>	512MB	256MB (A) 512MB (B)	64MB (AR9331) 2.5KB(ATmega)	256MB
<b>Armazenamento Interno</b>	2GB (ver B) or 4GB (ver C)	Não Possui	16MB(AR9331) 32KB(ATmega)	8MB
<b>Armazenamento Externo</b>	MicroSD	SD card	MicroSD(AR9331)	MicroSD
<b>Rede</b>	10/100Mbit Ethernet	Apenas Modelo B 10/100 Mbit Ethernet	10/100Mbit Ethernet e 802.11b/g/n WiFi	10/100Mbit Ethernet
<b>GPU</b>	PowerVR SGX530	Broadcom VideoCore IV	Não Possui	Não Possui
<b>Tensão</b>	5V do USB mini B, ou Pelo jack 2.1mm ou pino header.	5V do conector USB micro B, ou pino header.	5V do USB micro B ou pelo pino Header	5V do conector jack 2.1mm , ou pino header
<b>Tamanho</b>	86.4 mm x 53.3 mm	85.6mm x 56mm	68.6 mm x 53.3 mm	106.7mm x 71.1mm
<b>Peso</b>	40 gramas	45 gramas	41 gramas	50 gramas
<b>Preço</b>	\$55 (Mod. C) \$45 (Mod. B)	\$25 (Mod. A) \$35 (Mob. B)	\$75	\$80



### 2.2.2 Especificações de entradas e saídas

Neste tópico serão tratadas as especificações com respeito aos pinos de entrada e saída, operação dos pinos, tensão limite e outras especificações a respeito das entradas e saídas mais relevantes das plataformas.

Tabela 2 – Especificações de entradas e saídas [13], [14], [20], [21], [24] - [36]

	<b>Beaglebone Black®</b>	<b>Raspberry Pi®</b>	<b>Arduino Yun®</b>	<b>Intel Galileo®</b>
<b>Pinos Digitais de E/S</b>	65	17	20	14
<b>Entradas Analógicas</b>	7 com ADC de 12 Bits de 0 – 1.8 V	Não Possui	12 com ADC de 10 Bits de 0 – 5 V	7 com ADC de 12 Bits de 0 – 5 V
<b>Tensão de operação dos Pinos Digitais</b>	3V	3,3V	5V	3,3V ou 5V (Alternado com Jumper)
<b>Saídas PWM</b>	8	1	7	6 de velocidade Limitada
<b>Tensão de operação</b>	3,3 V até 250mA, 5 V até 1A	3,3 V até 50 mA, 5 V até 300-500mA	3,3 V até 50mA, 5 V	3,3 V até 800mA, 5 V até 800mA
<b>UART</b>	4	1	2 (1 ligado no AR9331)	2 (1 conectdo ao jack 3.5mm)
<b>SPI</b>	2	2	1	1
<b>I2C</b>	2	1	1	1
<b>USB Host</b>	1 Conector Tipo A	1(Mob. B) 2(Mod. A) Tipo A	1 Conector Tipo A	1 Conector Tipo AB
<b>USB Client</b>	1 Mini B	Não Possui	1 Micro B	1 Micro B
<b>Saída de Vídeo</b>	Micro HDMI	HDMI	Não Possui	Não Possui
<b>Entrada de Vídeo</b>	Não Possui	CSI (Câmera)	Não Possui	Não Possui

<b>Saída de Áudio</b>	Micro HDMI	HDMI ,conector 3,5 mm	Não Possui	Não Possui
-----------------------	------------	-----------------------	------------	------------

De acordo com as necessidades definidas para o desenvolvimento do projeto, a Beaglebone Black<sup>®</sup> apresenta o melhor custo benefício dentre todas as plataformas analisadas, pois possui processador robusto, grande quantidade de GPIOs, boa quantidade de memória além de apresentar um valor de compra inferior ao da Intel Galileo<sup>®</sup>.

Visando a necessidade de comprovar os recursos e especificações de cada plataforma de modo a analisar o desempenho destas em diversos testes, Dicola [13] realizou testes de *benchmark* com objetivo pontuar e avaliar os recursos e capacidades de cada plataforma

### 2.3 TESTE DE BENCHMARK ENTRE AS PLATAFORMAS

Para mensurar e quantificar o desempenho de cada plataforma foi necessário realizar uma bateria de testes utilizando o *software* NBENCH [13]. O NBENCH é um teste de *benchmark*, desenvolvido na década de 90 pela revista BYTE cuja proposta é medir o desempenho de CPU, FPU e memória, atribuindo uma pontuação para cada um destes testes, medindo em pontos a capacidade de cada dispositivo testado [38]. O NBENCH leva cerca 10 minutos para ser executado na maioria das máquinas e compara o sistema que está sendo testado com dois sistemas de referência, o Dell Pentium 90 com 256 KB executando o MSDOS e o outro sistema é um Linux AMD K6/233 com 512 KB de memória cache.

O teste é realizado em 2 etapas. Na primeira etapa é realizada a calibração do circuito para determinar o tamanho do teste. Na segunda etapa é executado o teste selecionado repetidas vezes para obter um resultado estatisticamente significativo, o resultado do teste então é comparado com os sistemas de referência para construir uma pontuação do sistema selecionado [13], [39].

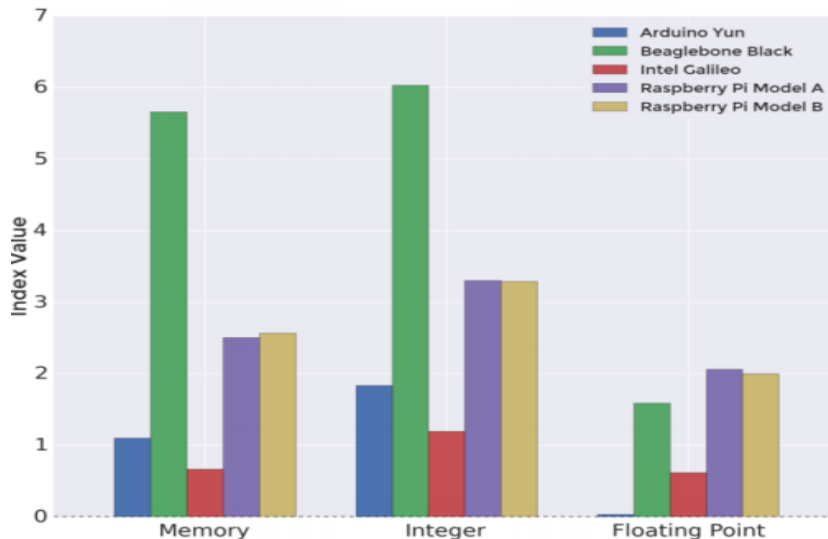
Dicola [13] realizou diversos testes focando em mensurar os índices de inteiro, de pontos flutuantes e de memória. As quatro plataformas foram testadas e geraram os seguintes resultados apresentados na Tabela 3.

Tabela 3 - Resultados do teste NBENCH [14]

	Arduino Yun <sup>®</sup>	Beaglebone Black <sup>®</sup>	Intel Galileo <sup>®</sup>	Raspberry Pi <sup>®</sup> (mod A)	Raspberry Pi <sup>®</sup> (mod B)
Índice de memória	1.104	5.661	0.669	2.509	2.570
Índice de Inteiro	1.840	6.032	1.198	3.305	3.291
Índice de pontos Flutuantes	0.038	1.591	0.621	2.064	2.002

A partir dos dados constantes na Tabela 3, foi gerada a Figura 2.5 para melhor visualização dos resultados.

Figura 2.5 – NBENCH test graph[14]



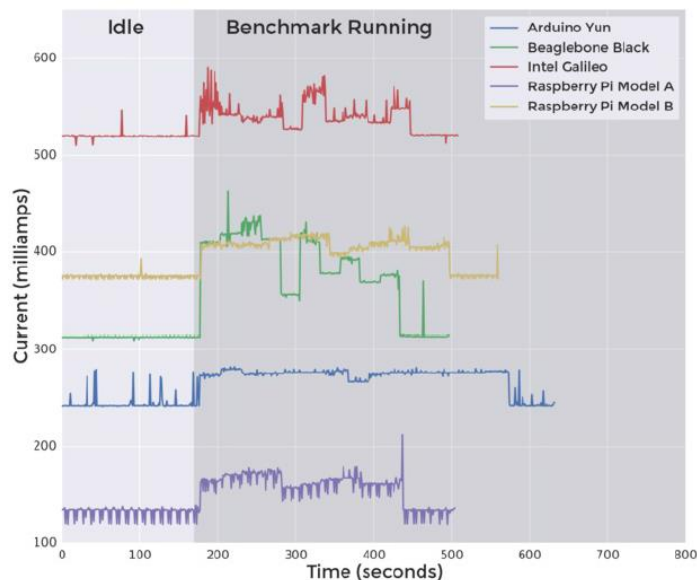
A partir da análise dos dados constante no gráfico, é possível deduzir que a Beaglebone<sup>®</sup> apresenta os melhores índices nos testes de memória e inteiro. Tais resultados são devido ao processador de alto desempenho. Porém nos testes de ponto flutuante, o índice da Beaglebone<sup>®</sup> teve desempenho inferior ao da Raspberry<sup>®</sup> (Modelo A/B). A Raspberry Pi<sup>®</sup> possui um processador com arquitetura ARM Cortex-A8 assim como a Beaglebone Black<sup>®</sup>, entretanto o cálculo de pontos flutuantes na plataforma Beaglebone Black<sup>®</sup> é feito pelo co-processador de ponto flutuante (VFPLite) que não é tão rápido quanto as FPUs do próprio processador ARM, usados pela Raspberry Pi<sup>®</sup> [13], [39].

Outro aspecto relevante é que apesar do Arduino Yun<sup>®</sup> e da Intel Galileo<sup>®</sup> trabalharem com a mesma frequência de clock de 400 MHz, o Arduino Yun<sup>®</sup> possui uma vantagem devido a arquitetura MIPS comparativamente com a arquitetura i586 da Intel<sup>®</sup>. Quanto ao desempenho de ponto flutuante do Arduino Yun<sup>®</sup> este é quase nulo, pois o processador da plataforma não possui uma FPU, e para calcular os valores de ponto flutuante só é possível através de *software*. As Raspberry Pi<sup>®</sup> (A/B) apresentam índices muito próximos em todos os testes devido ao processador ser o mesmo, o que é muito bom quando se trata de valores pois o modelo A custa em média \$25.00, porém o modelo A não possui uma série de periféricos, tais como conexão ethernet, o que impossibilitaria comunicação com fio.

## 2.4 CONSUMO DOS MÓDULOS

O consumo de energia dos módulos foi analisado durante os testes do NBENCH através da medição da corrente elétrica consumida. Para medir a corrente, foi utilizado um sensor de corrente modelo INA219 conectado a um Arduino<sup>®</sup> Uno, cujo resultado está presente na Figura 2.6.

Figura 2.6 - NBENCH testing consumption[14]



Cada plataforma foi ligada ao mínimo de periféricos possível. O Arduino Yun<sup>®</sup> estava com o WiFi ativado, a Beaglebone Black<sup>®</sup>, Raspberry Pi<sup>®</sup> (Modelo B) e a Intel Galileo<sup>®</sup> estavam com as suas conexões ethernet ativas e a Raspberry Pi<sup>®</sup>

(Modelo A) estava conectada a porta USB e a um monitor HDMI. O motivo da utilização dos periféricos foi para compensar os recursos adicionais das plataformas mais potentes. De acordo com o teste, a plataforma que possui o maior consumo é a Intel Galileo<sup>®</sup>. Já a Beaglebone Black<sup>®</sup>, se manteve estável no início do teste como é possível visualizar na Figura 2.6, entretanto em seguida o consumo aumentou consideravelmente chegando a cerca de 450 mA.

## 2.5 DESENVOLVIMENTO DE SOFTWARE DAS PLATAFORMAS

Com intenção de analisar os impactos no desenvolvimento da proposta em cada plataforma, este tópico destina-se a levantar os pontos fortes e fracos com relação ao desenvolvimento de Software de cada plataforma.

### 2.5.1 Arduino Yun<sup>®</sup>

#### 2.5.1.1 Sistema Operacional

O Arduino Yun<sup>®</sup> devido ao seu poder de processamento, é possível executar a distribuição Linux OpenWrt-Yun, que deve ser configurada a partir da linha de comando, o que implica numa maior dificuldade para iniciantes em plataformas embarcadas. É possível a instalação de pacotes de desenvolvimento de software por meio do gerenciador de pacotes do Linux, o opkg, porém a plataforma não permite a instalação de muitos programas/pacotes de desenvolvimento, devido ao seu limitado espaço de armazenamento [13], [14], [40].

O sistema operacional é livre e de código fonte aberto e possui licença GPL, o que facilita o acesso de novos usuários em comunidades de desenvolvimento [13], [14], [40].

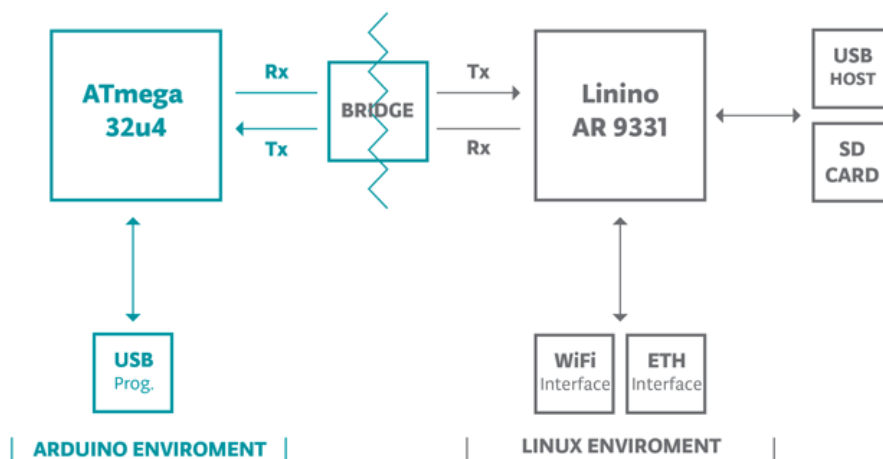
#### 2.5.1.2 Arquitetura de comunicação

O Arduino Yun<sup>®</sup> possui dois processadores o ATmega32u4 e o AR9331, entretanto estes não se comunicam de maneira contínua. A arquitetura possui uma comunicação USB embarcada no ATmega32u4, e caso haja necessidade de

acessar o outro processador, basta utilizar as bibliotecas de ponte, conhecidas como *Bridge Library*, disponibilizada pelo próprio Arduino®.

A comunicação entre o ambiente Linux e processador ATmega é realizada através de uma conexão serial, a biblioteca de ponte facilita a comunicação entre os dois processadores, como ilustra a Figura 2.7, esta configuração de comunicação na arquitetura possibilita o uso de ferramentas essenciais como *shell scripts*, interfaces de rede e recebimento de informações do processador AR9331. Os periféricos de entrada USB, interface de rede e cartão SD estão conectados ao processador AR9331, então para programar o Arduino através da porta USB do ATmega32u4 e utilizar tais recursos é necessário o uso da biblioteca de ponte[14], [40].

Figura 2.7 - Arduino Yun® Architecture [14], [40]



### 2.5.1.3 Desenvolvimento de Software

No Arduino Yun® é possível programar em linguagens como *Python*, que já vem instalada no sistema operacional OpenWrt-Yun, permitindo ao usuário escrever aplicativos ou *scripts* sem necessidade de instalação de nenhum pacote de desenvolvimento ou compilador adicional. Como o sistema operacional oferece suporte para ferramentas de desenvolvimento através do gerenciador de pacotes, é possível programar em outras linguagens como *PHP* ou *Perl*, o que do ponto de vista técnico é bom, entretanto o limitado espaço restringe a quantidade de aplicações e ferramentas instaladas na plataforma [13], [14].

## 2.5.2 Raspberry Pi®

### 2.5.2.1 Sistema Operacional

Assim como todas as plataformas abordadas pelo trabalho, a plataforma Raspberry Pi® possui suporte a sistemas operacionais de alto nível (HLOS), a distribuição mais recomendada pela comunidade é a Raspbian que é uma derivação da distribuição Debian, muito utilizada em *desktops* e servidores [13], [46].

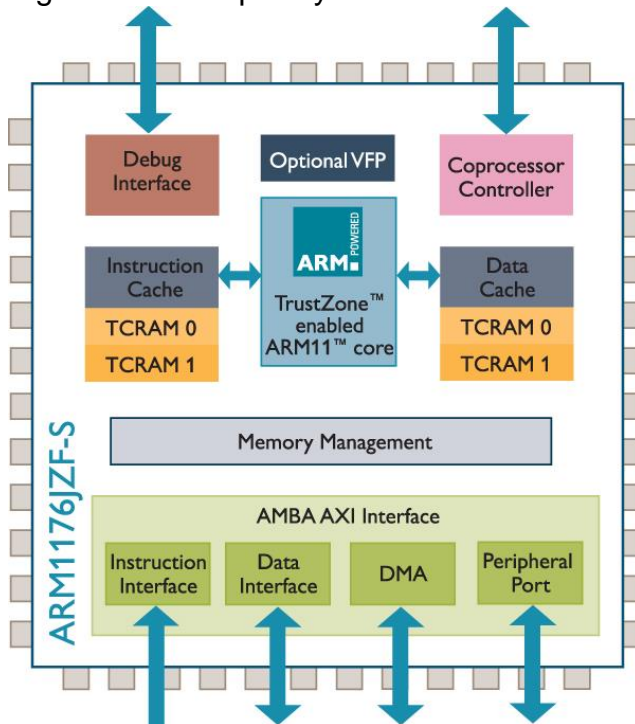
Como o objetivo da Raspberry Pi® foi criar uma plataforma de desenvolvimento educacional, visando facilitar o seu uso, a empresa fornece um cartão SD que permite ao usuário selecionar o sistema operacional que deseja instalar, facilitando ainda mais o processo de imersão na plataforma [47]. A Raspberry Pi® também oferece suporte a diversos sistemas operacionais além do Raspbian, tais como [47]:

- Pidora baseado no Fedora Remix;
- OPNELEC baseado no An XBMC Media Centre;
- RASPBMC baseado no An XBMC Media Centre;
- Risc OS - sistema operacional não baseado em Linux;
- Arch Linux – distribuição baseada em Linux, porém com recursos essenciais, visando performance.

### 2.5.2.2 Arquitetura da plataforma Raspberry Pi®

A plataforma Raspberry Pi® utiliza um processador ARM1176 que tem sua arquitetura baseada no ARMV6 (Figura 2.8). Este modelo de processador é indicado para aplicações de dispositivos embarcados tais como *smartphones*, TV Digital e *eReaders*. O processador possui um único núcleo RISC de 32 bits com 700 MHz de frequência, com configuração de *overclock* através da interface, permite que o usuário configure facilmente o *clock*, podendo chegar até 1 GHz [34], [48].

Figura 2.8 - Raspberry Pi® Platform Architecture [48]



### 2.5.2.3 Desenvolvimento

A Raspberry Pi® fornece suporte a instalação de pacotes de desenvolvimento e compiladores através do sistema operacional, assim como as outras. O foco da plataforma está voltado para a educação, logo, é mais indicada para usuários iniciantes em plataformas de desenvolvimento. A linguagem mais utilizada é a *Python*, por ser uma linguagem de fácil entendimento e implementação [13], [34].

A arquitetura ARMV6 proporciona suporte a diversas linguagens de programação, dentre elas estão: C, C++, *Java*, *Scratch* e *Ruby*. O que chama atenção e agrega valor a plataforma, é que ela possui a maior comunidade de usuários entre todas as plataformas [13], [34].



### 2.5.3 Intel Galileo®

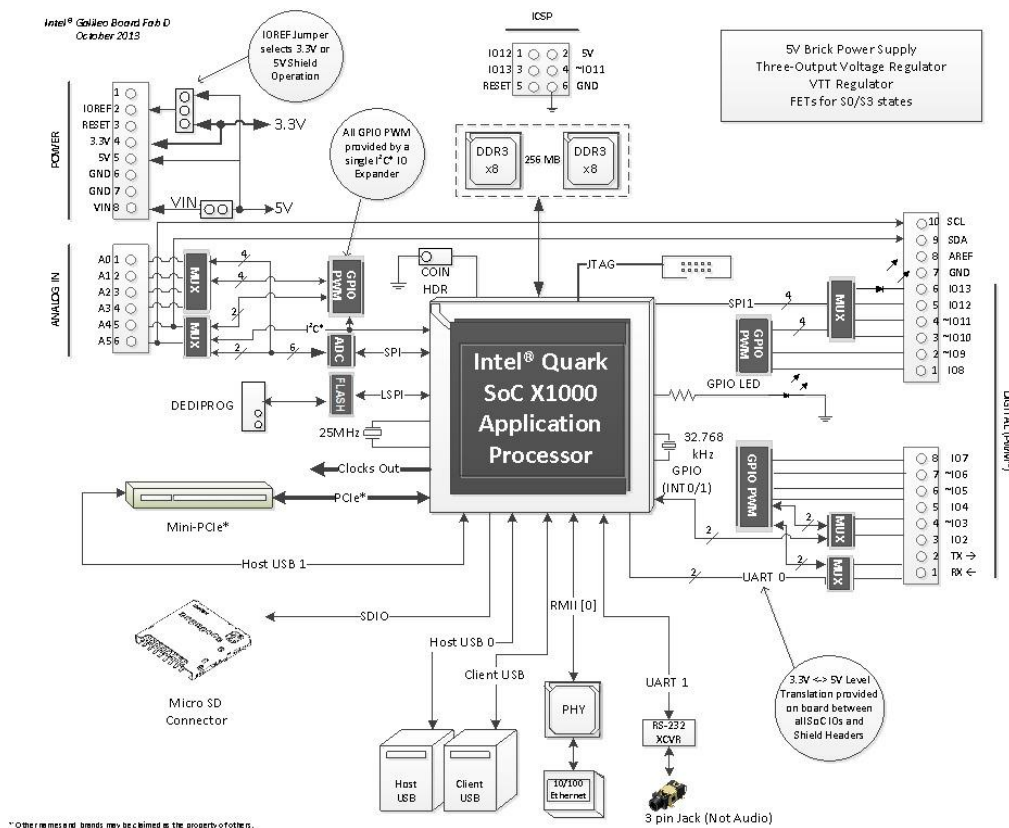
Um dos focos da Intel® foi fazer com que a plataforma trabalhasse em conjunto com a plataforma Arduino®, a distribuição utilizada na plataforma é o Linux Yocto que também é baseada no Linux para dispositivos embarcados.

Uma das vantagens da distribuição Yocto é que, ele já possui uma comunidade bem consolidada de desenvolvedores, devido ao fato da distribuição já ser utilizada em outros dispositivos. O kernel do Yocto pode ser executado nas arquiteturas ARM, PPC, MIPS, x86, e x86-64 [13], [44].

#### 2.5.3.1 Arquitetura da plataforma Intel Galileo®

A Intel Galileo® é baseada no processador Intel® SoC Quark X1000 e é a primeira plataforma de desenvolvimento com E/S embarcado baseada na arquitetura Intel® i586 (Figura 2.9) para ser compatível com a plataforma Arduino®. A plataforma Arduino® trabalha com uma tensão de operação de 5 V e a Intel Galileo® trabalha com uma tensão de 3,3 V, para resolver este problema, a arquitetura da plataforma disponibiliza a configuração de tensão através de um *jumper* que altera a tensão nos pinos de E/S de 5 V para 3,3 V e vice-versa [45]. Com essas configurações é possível trabalhar com uma programação de *Hardware* idêntica a da plataforma Arduino®.

Figura 2.9 - Intel Galileo<sup>®</sup> platform architecture[50]



### 2.5.3.2 Desenvolvimento

Uma das vantagens da Intel Galileo<sup>®</sup> é que ela pode ser programada da mesma maneira que o Arduino<sup>®</sup>, inclusive no mesmo ambiente de programação, facilitando ainda mais a migração de usuários da plataforma Arduino<sup>®</sup> para a Intel Galileo<sup>®</sup> [45]. Entretanto, para um desenvolvimento mais aprofundado com a Intel Galileo<sup>®</sup>, é necessária a instalação de pacotes de compiladores como o GCC e isso para alguns usuários da plataforma pode ser difícil, visto seu que a comunidade de suporte para a plataforma ainda está crescendo. Na plataforma também é possível programar em outras linguagens como *Python* e *Processing* [13], [45].

## 2.5.4 Beaglebone Black<sup>®</sup>

### 2.5.4.1 Sistema Operacional

A plataforma Beaglebone Black<sup>®</sup> possui uma boa vantagem com relação a algumas plataformas devido a possibilidade de utilizar sistemas operacionais mais robustos e conhecidos, como a distribuição Ubuntu baseada em Linux. A distribuição Linux oficializada pela Beagleboard<sup>®</sup>, é a distribuição Angstrom. Já a distribuição Ubuntu apesar de não ter sido oficializada pela *Beagleboard*, possui uma das maiores comunidades de usuários devido ao seu grande número de usuários que já utilizam o sistema operacional em computadores pessoais [13], [41].

O Angstrom é uma versão simples do Linux projetado especificamente para a Beaglebone Black<sup>®</sup> e pode ser adaptado para dispositivos embarcados, o Angstrom é baseado na distribuição Debian muito utilizada em *desktops* e servidores.

Assim como a distribuição Angstrom, uma distribuição Ubuntu foi especificamente projetada também para a Beaglebone Black<sup>®</sup>. Além das distribuições Angstrom e Ubuntu a plataforma também permite a instalação de outras distribuições como: Debian, Android, ArchLinux, Gentoo, Sabayon, Buildroot, Erlang, Fedora [13], [41], [42].

### 2.5.4.2 Arquitetura da Beaglebone Black<sup>®</sup>

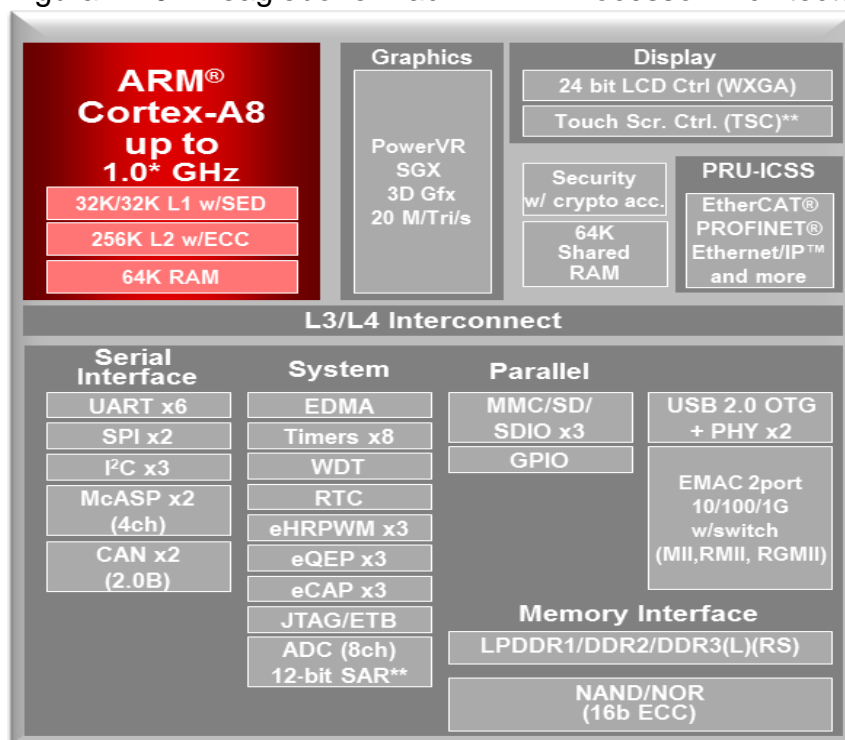
Os microprocessadores AM335x são baseados no processador Cortex-A8 da ARM, que estão focados no processamento de imagens, gráficos 3D, periféricos e protocolos de comunicação industriais tais como EtherCAT e PROFIBUS. A arquitetura ARM (Figura 2.10) fornece suporte a sistemas operacionais de alto nível (HLOS) possibilitando a execução de sistemas operacionais como o Ubuntu ou o Android. O microprocessador está associado a um subsistema de aceleração gráfica, o PowerVR SGX, que fornece o suporte para aplicações em 3D como edições de imagem e jogos [42], [43].

Uma das características mais interessantes a respeito da plataforma da Beaglebone Black<sup>®</sup> é a PRU, frente ao seu potencial de processamento em tempo real, entretanto para utilizar a unidade é necessário realizar uma série de

configurações. A PRU é constituída por dois núcleos RISC de 32 Bits com 256 KB de memória, controle de interrupções e periféricos internos que permitem controle de interfaces e protocolos de comunicação. Entre as interfaces suportadas pela PRU se destacam os protocolos de comunicação industriais de execução em tempo real que utilizam a topologia mestre-escravo tais como:

- EtherCat.
- PROFINET.
- EtherNet / IPTM.
- PROFIBUS.
- POWERLINK.
- SERCOS III

Figura 2.10 - Beaglebone Black<sup>®</sup> ARM Processor Architecture [43]



#### 2.5.4.3 Desenvolvimento de Software

Devido a sua arquitetura e grande espaço interno é possível instalar diversos pacotes de ferramentas de desenvolvimento no sistema, possibilitando a utilização

de compiladores e pacotes de desenvolvimento como o G++, GCC, *Python*, *Ruby*, *Node.js*, *Perl* e outras. O sistema operacional Angstrom fornece um ambiente de desenvolvimento virtual através do navegador, chamado de Cloud9 IDE, permitindo ao usuário iniciante dar seus primeiros passos com a Beaglebone Black<sup>®</sup> sem que seja necessário um amplo conhecimento sobre a plataforma. A linguagem usada para programar no Cloud9 é a *Bonescript* que é uma variação da linguagem Node.js baseada em *JavaScript* [13], [42].

Para utilização dos núcleos da PRU é necessário configurar os registradores da unidade e desenvolver o *Software* de programação. Outro problema no desenvolvimento é a manipulação das E/S utilizando a PRU da plataforma, implicando em um elevado conhecimento sobre a plataforma [13].

### 2.5.5 Conclusão sobre as plataformas de desenvolvimento

Devido aos recentes avanços da tecnologia móvel e microcontroladores, podemos levantar algumas considerações tais como os sistemas embarcados estão cada vez mais baratos e potentes com o passar dos anos.

As plataformas Arduino<sup>®</sup> e Intel Galileo<sup>®</sup> não se mostraram interessantes devido a baixa capacidade de processamento. Além da baixa capacidade, a Intel Galileo<sup>®</sup> não possui um suporte tão favorável quanto as outras plataformas por meio das comunidades de desenvolvimento. Mesmo a Intel Galileo<sup>®</sup> possuindo uma grande vantagem no quesito acadêmico, visto que ela é uma plataforma recente. Devido aos fatores de desempenho e desenvolvimento ela não foi selecionada, além da capacidade de processamento, a quantidade de periféricos é inferior se comparada com as plataformas Raspberry Pi<sup>®</sup> e a Beaglebone Black<sup>®</sup>, o que dificulta numa possível expansão do projeto mais a frente. Durante os testes foi a plataforma que apresentou a maior temperatura de operação de todas, o que pode dificultar a aplicação da nossa proposta com a Intel Galileo<sup>®</sup> em localizações com alta temperatura, podendo danificar a plataforma [13].

Assim como a Beaglebone Black<sup>®</sup>, o processador da Raspberry Pi<sup>®</sup> também pode chegar a 1 GHz, sendo assim, ambas possuem o mesmo espaço de armazenamento interno e com arquitetura ARM, então subentende-se que ambas funcionariam da mesma maneira, diferenciando-se apenas pelo preço, o que torna a

Raspberry Pi<sup>®</sup> mais atrativa. Entretanto a plataforma Raspberry Pi<sup>®</sup> não possui um ADC interno o que implica na necessidade de integrar um conversor externo aumentando a possibilidade de falhas ocasionadas por fatores externos, além do acréscimo monetário do ADC, que no final, ficaria na mesma faixa de preço da Beaglebone Black<sup>®</sup>. Além do mais, a Raspberry Pi<sup>®</sup> não possui uma unidade específica para aplicações em tempo real o que aumenta mais ainda a vantagem da Beaglebone Black<sup>®</sup> sobre a Raspberry Pi<sup>®</sup>.

Visto isso, a estes fatores, a plataforma mais indicada foi a Beaglebone Black<sup>®</sup>, além do mais, tem uma característica única entre as plataformas de desenvolvimento, que é a PRU. Com seu alto poder de processamento além de poder utilizar os processadores dedicados para aplicações em tempo real, a plataforma permite a instalação de diversos compiladores e pacotes que auxiliam o desenvolvimento devido ao seu grande espaço de armazenamento, o que a torna ideal para o projeto.

## 2.6 BEAGLEBONE BLACK / PRU

A Beaglebone Black<sup>®</sup> é uma plataforma potente do tamanho de um cartão de crédito baseado no SoC da *Texas Instruments Sitara AM335x ARM Cortex-A8* conforme visto neste capítulo. O subsistema da PRU é composto de 2 processadores com frequência de clock de 200 MHz que operam independentemente do processador principal, o ARM. Um ou ambos processadores da PRU podem enviar ou receber sinais de temporização o que evita problemas de *software*. Os dois processadores da PRU, tem uma frequência máxima de 200 MHz, e 26 portas de E/S que podem ser utilizadas para controle, cada pino pode ser alternado em um ciclo de clock, fornecendo a Beaglebone Black<sup>®</sup> uma resolução de tempo máxima de 5 ns a cada pulso.

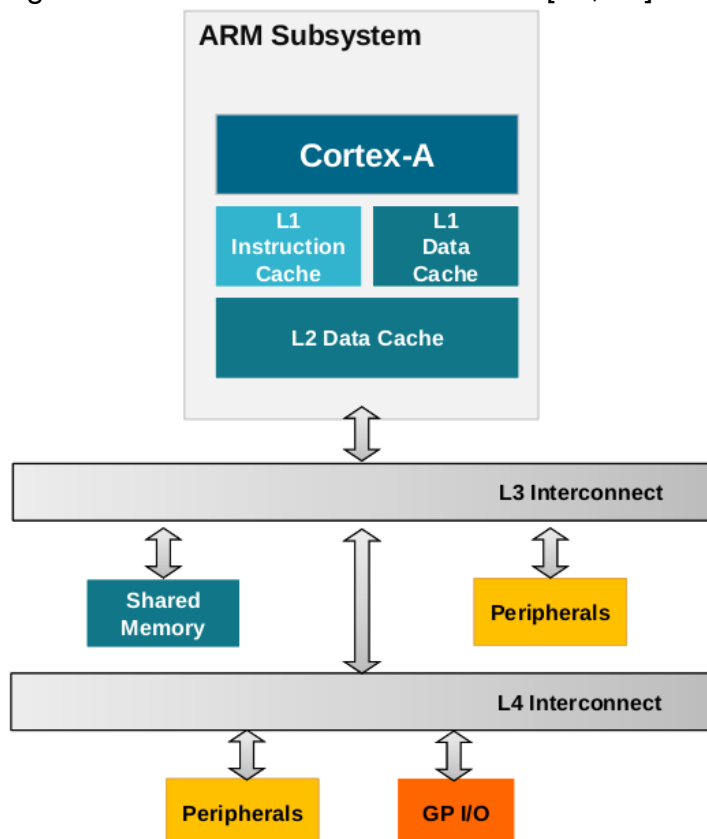
O subsistema pode ser desenvolvido em linguagens como C ou *Assembly*, ou utilizar algum pacote de bibliotecas já disponíveis na comunidade da plataforma, uma vez que as bibliotecas abstram a maior parte das configurações dos registradores, o que facilita e agiliza no uso da unidade. Para utilizar os ADCs da plataforma é necessário utilizar o módulo *Touchscreen* da plataforma, conhecido

como *Touchscreen Controller and Analog-to-digital Converter Subsystem* (TSC\_ADC\_SS), o módulo é constituído por 8 canais analógicos e pode ser configurado de diversas maneiras, dentre essas podemos citar [55] :

- 8 canais ADC de uso geral;
- TSC de 4 canais com 4 canais ADC de uso geral;
- TSC de 5 canais com 3 canais ADC de uso geral;
- Apenas Touchscreen em todos os canais.

Para entender melhor a comunicação e como desenvolver o *Software*, a necessidade de analisar a arquitetura da comunicação entre a PRU e o processador ARM é imprescindível, visto que, antes de analisar a arquitetura da PRU é necessário entender como a arquitetura ARM que está presente na Figura 2.11 trabalha.

Figura 2.11 – SoC ARM Architecture [55, 56]

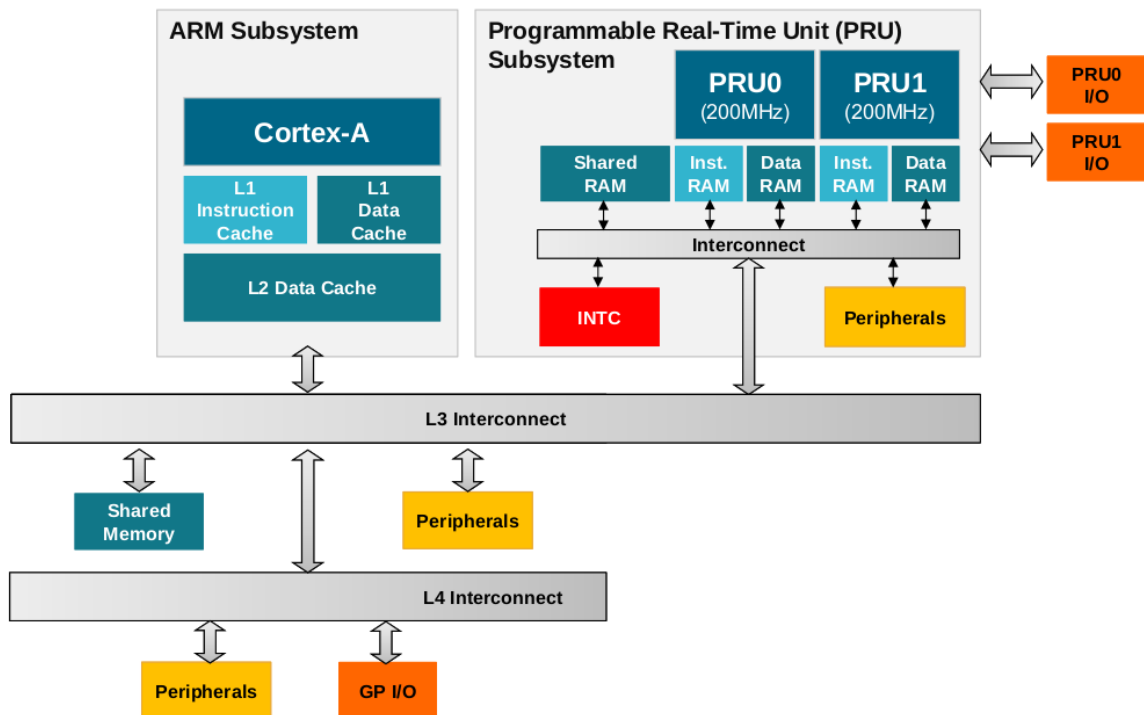


As interconexões L3 e L4 são canais de comunicação de alta frequência de transmissão, a interconexão L3 é baseado em um *Network-on-Chip* (NOC) que usa

um protocolo baseado em pacotes. Este protocolo baseado é em pacotes de transição para frente e para trás. Na transição para frente, o protocolo realiza a seguinte sequência: lê o comando, e posteriormente escreve o comando que será carregado. Já na transição para trás, o protocolo realiza a sequência: lê a resposta com os dados carregados, e posteriormente escreve a resposta que confirma a leitura. A interconexão L3 foi construída visando desempenho, por isso sua topologia foi dividida em 2 domínios de *clocks* diferentes, L3F que corresponde ao L3 *Fast Clock* para transições mais rápidas e prioritárias, já o L3S que corresponde ao L3 *Slow Clock* para transições menos prioritárias.

Já a interconexão L4 é uma interconexão de baixa latência e pode lidar com tráfego de entrada de até 4 iniciadores, além disso pode distribuir as mensagens para até 63 alvos diferentes, a interconexão L4 possui 3 interfaces de comunicação de alta velocidade com a interconexão L3, sendo assim possível se comunicar com diversos periféricos da plataforma. É possível visualizar mais claramente a arquitetura de comunicação entre a PRU e o ARM na Figura 2.12.

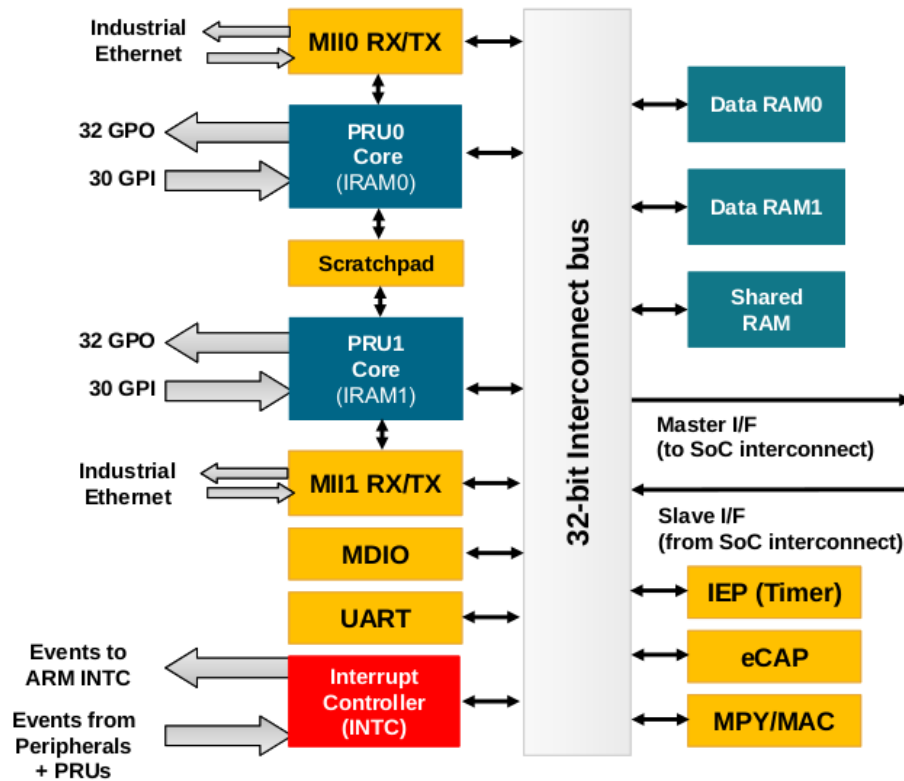
Figura 2.12 – ARM Architecture with PRU [55,56]





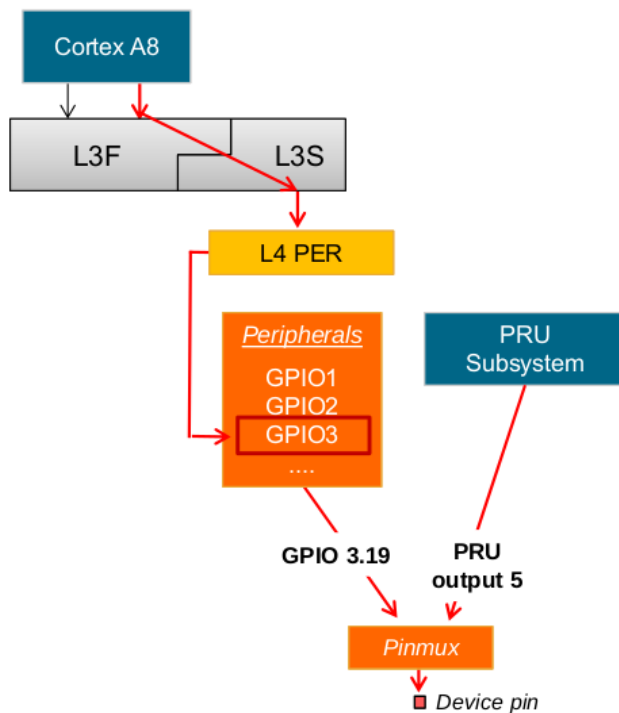
A PRU possui uma interconexão interna que permite a comunicação com a arquitetura ARM do processador principal da Beaglebone Black®. O diagrama de blocos da PRU pode ser visto na Figura 2.13, onde é possível visualizar como a comunicação é feita entre a interconexão e os periféricos, memória e outros.

Figura 2.13 – PRU block diagram [56]



Uma das razões da comunicação da PRU ser muito mais rápida é porque além de ter memórias posicionadas mais próximas ao processador, a unidade antecipa algumas etapas que são necessárias na rotina de processamento das informações que são feitas na arquitetura ARM como é possível visualizar na Figura 2.14.

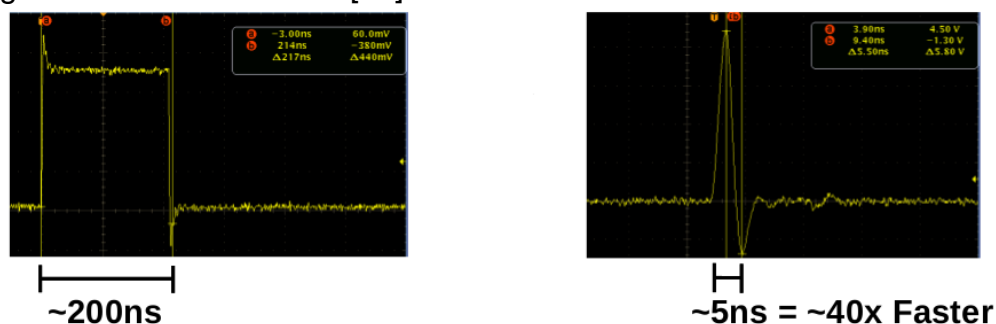
Figura 2.14 - PRU I/O Interface [56]



Diante dos fatos é possível comprovar que a PRU tem desempenho superior em aplicações se comparadas com o processador ARM principal e com todas as outras plataformas aqui estudadas. Sendo assim, na nossa proposta, a maior vantagem de utilizar a unidade está voltada para a baixa latência de acesso aos pinos por meio do acesso direto, fornecendo uma alta frequência de aquisição de informações nas portas analógicas.

Devido a sua arquitetura desenvolvida para aplicações em tempo real, o processo executado pela unidade é 40 vezes mais rápida que o mesmo processo executado pelo processador ARM principal como ilustra a Figura 2.15 [56].

Figura 2.15 - ARM X PRU [56]



Para acessar a PRU é necessário realizar as configurações nas device trees para que estas permitam ativar e acessar o módulo TSC\_ADC\_SS e acessar as portas analógicas, sendo assim, foi feita uma configuração que ativa a o módulo faz a chamada das portas analógicas e transmite para o ARM.

Para ativar a PRU via device tree é necessário transformar o arquivo via *device tree compiler* (DTC) am335x-boneblack.dtb que está em formato binário, para o formato .dts que apresenta as configurações em forma de string para realizar as alterações necessárias. Depois de convertido o arquivo para .dts, é necessário localizar o fragmento pruss@4a300000 e alterar o seu status para “okay”, por padrão, esta configuração vem com “disabled”. Feito isto, é necessário criar os arquivos que serão responsáveis pela aquisição na PRU e o arquivo que é responsável por capturar os dados fornecidos pela PRU e transferir para o ARM, via memória compartilhada. O processo para acessar a PRU e ler as informações dos canais analógicos é realizado em duas etapas:

1. Iniciar o ADC
  - a. Ativa o Stepconfig de 1 a 7 (onde 1 representa o canal 0 e assim por diante).
2. Ler o ADC
  - a. Verifica se tem informação na memória;
  - b. Inicia contador;
  - c. Escreve os valores na memória compartilhada da PRU;
  - d. Incrementa contador;
  - e. Informa ao host de destino a flag de que existe informação;
  - f. Espera o host acessar para leitura;
  - g. Desativa o ADC
  - h. Reinicia o *loop*.

Já o arquivo que é responsável por receber as informações da memória compartilhada da PRU e alocar no ARM a indicação do arquivo criado para acessar a PRU, onde ele aguarda a flag que informa que existe informação, feito isto, é criado um buffer para armazenar os valores que foram encontrados na memória compartilhada, após a transferência completada, o buffer armazena no arquivo

desejado e é zerado em seguida, aguardando a próxima flag indicando que existem mais valores a serem lidos.

Além desta maneira de configuração e acesso a PRU, existem bibliotecas de acesso rápido a PRU que facilitam na programação e o uso, as bibliotecas já vem com uma série de configurações prontas para uso, sem que haja a necessidade de uma configuração para utilizar a PRU. Foi testado o acesso com a biblioteca indicada pela própria Beaglebone [57] que é a LibPruIO.

### 2.6.1 LibPruIO

A LibPruIO é uma biblioteca de fácil acesso a PRU, visto que, desenvolver aplicações baseadas em entradas e saídas na plataforma não é algo tão simples, ainda mais quando se é necessário a amostragem em tempo real. A biblioteca é recomendada pelo próprio fabricante [57]. Quando se trabalha com entradas e saídas na Beaglebone seja na versão Black ou White é possível verificar questões que podem impactar no projeto como:

- Sobreposição de *Device Trees* (criando ou adaptando novas);
- Diversas versões do compilador dos *device trees* (DTC);
- Documentação espalhada pelo código (dificulta o entendimento).

Quando se trata das aplicações para portas analógicas é ainda pior, pois algumas versões do kernel para o ADC vêm com defeito que não possibilita controlar o ADC e as amostras são lentas.

Em contrapartida a LibPruIO veio para facilitar o acesso a PRU e solucionar boa parte ou até todos esses problemas ocasionados pelo uso das portas analógicas da plataforma. A LibPruIO é um driver da família do processador AM33xx que é possibilita controlar diversos subsistemas com ele, tais como:

- Módulo de controle (Pinmux)
- TSC\_ADC\_SS (Touch Screen Controller and Analog to Digital Converter SubSystem)
- PWM\_SS (Pulse With Modulation SubSystem)

O Subsistema do TSC\_ADC\_SS é o responsável pelo conversor analógico digital (ADC) da plataforma, o que possibilita controlar as amostras que serão enviadas do módulo de aquisição e condicionamento de sinais, ou seja só é possível ler as portas analógicas se ativar o subsistema nas *device trees*. Uma vantagem com relação ao uso da biblioteca, é que ela pode ser utilizada por qualquer pessoa ou empresa, pois está sob a licença Lesser GNU Public Licence version 2 (LGPLv2). A LibPrulo é totalmente configurável e possui características como:

- Pode ser executada em ambas as PRUs (PRU-0 e PRU-1);
- Subsistemas de controle em tempo de execução (pode ativar ou desativar via configuração);
- Diferentes modos de execução (IO, RB e MM);
- Acesso direto aos GPIOs;
- Definir saída dos GPIOs;
- Obter valores das entradas analógicas (ADC);
- Realizar medições de alta velocidade;
- Permite a utilização de gatilhos para linhas digitais ou analógicas.

No Modo IO, a PRU é executada em um loop infinito e lida com linhas de entrada e saída para todos os subsistemas acessados pela biblioteca com a mesma prioridade. Dependendo do número de subsistemas que são acessados, a configuração pode apresentar atrasos de até 50 ns antes do ADC iniciar uma nova série de amostras.

O Modo RB é executado em loop infinito também e manipula as reinicializações do subsistema de acordo com a prioridade. Entradas e saídas são manipuladas apenas quando a PRU está esperando a próxima reinicialização. Isso também pode causar um delay de até 50 ns.

No modo MM o PRU espera por um comando de partida e executa uma única medição. Ele lida apenas com amostras analógicas, em compensação tem um delay de espera mais baixo. Para escolher o modo de Entrada e Saída, basta configurar o parâmetro Samp na chamada da função Prulo::config(), alterando apenas o valor do parâmetro que é o Samp que define o modo de operação se será IO, RB ou MM.

Vale lembrar que para quebrar o loop dos modos RB ou IO, basta chamar a função Prulo::config() novamente, ou destruir a chamada da função utilizando a função Prulo::~~Prulo.

Todos os Subsistemas são acessados da mesma maneira e são executados em 3 passos por amostra, estes passos são:

- 1 – Criar uma estrutura do tipo PrulO e lê as configurações iniciais (constructor `Prulo::Prulo()` );
- 2 – Carregar a configuração personalizada para os registradores do subsistema e inicia a operação (`Prulo::config()` );
- 3 – Depois de concluído, o registrador original é restaurado e em seguida a estrutura da PRU que foi carregada na primeira etapa é destruída (destructor `Prulo::~~Prulo()`).

Para configuração do ADC, as alterações devem ser feitas antes da segunda etapa, que é responsável por carregar as customizações de código.

A PRU pode acessar o conversor analógico em todos os modos (IO, RB e MM), o sinal de entrada pode ser analisado por meio de funções como a `AdcUdt`, que é responsável por controlar o subsistema Touch Screen Controller (TSC\_ADC\_SS) da plataforma. É possível analisar as amostras analógicas por meio de 8 canais (AIN-0 até AIN-7). É importante observar que o canal AIN-7 está conectado a um divisor de tensão para alimentação de 3,3V e não de 1,8V.

O Subsistema ADC pode usar uma configuração de até 16 passos para realizar uma medição, a configuração padrão da `libpruio` é entre 1 e 8 passos para realizar a aquisição dos canais entre AIN-0 e AIN-7.

Funções:

- `Prulo::config()` para configurar o modo de execução, tempo de medição, passos e o bit encoding.
- `AdcUdt::setStep()` para personalizar a configuração dos passos da aquisição (Steps)
- `AdcUdt::Value`, responsável por retornar os valores amostrados, que neste caso depende do modo de execução da biblioteca.

Durante um breve teste a biblioteca `LibPrulO` apresentou resultado semelhante ao programa desenvolvido para aquisição das portas analógicas, cuja intenção foi analisar os sinais de tensão e corrente fornecidos pelo módulo de aquisição e condicionamento de sinais.

Visto as informações necessárias a respeito da plataforma Beaglebone Black<sup>®</sup> com a PRU, foi possível iniciar o desenvolvimento do sistema de tal forma que seja possível operar a Beaglebone Black<sup>®</sup> em conjunto com a plataforma de

aquisição e condicionamento de sinais para análise das informações de tensão e corrente fornecidos pela rede elétrica.

### **3      SISTEMA PROPOSTO**



### 3 SISTEMA PROPOSTO

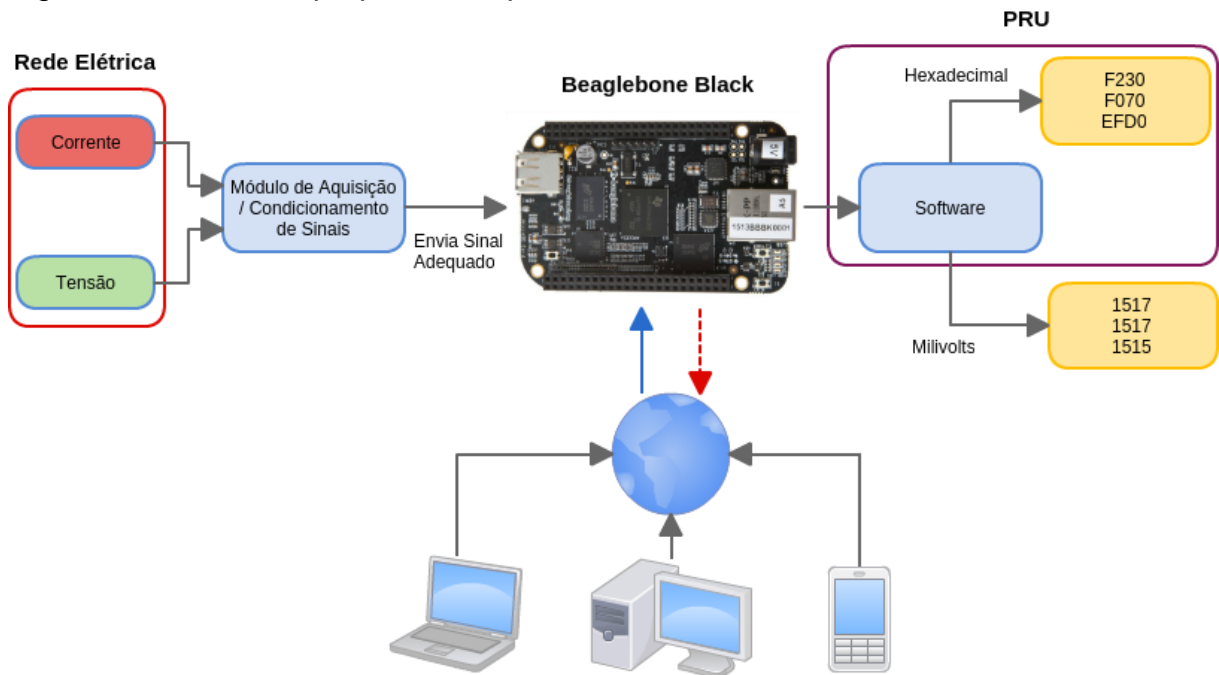
Este trabalho propõe a implementação de um sistema de aquisição de sinais de tensão e corrente da rede elétrica, utilizando a plataforma Beaglebone Black<sup>®</sup>. A escolha desta plataforma ocorreu baseada no estudo comparativo realizado no Capítulo 2. Incorporada a plataforma há um módulo de aquisição e condicionamento de sinais, cujo objetivo é adequar os sinais da rede elétrica para a entrada as portas analógicas da plataforma.

Para escolha da plataforma de desenvolvimento, foram levados em consideração os seguintes pontos:

- Possuir custo menor ou igual a U\$ 80,00;
- Atender as exigências da ANEEL [18];
- Capacidade de processamento compatível com as necessidades do projeto;
- Possibilidade de integrar a plataforma de desenvolvimento ao módulo de aquisição e condicionamento de sinais;
- Plataforma com sistema operacional livre de licença.

A Figura 3.1 ilustra a proposta do sistema completo, contemplando os módulo de aquisição e conversão dos sinais de tensão e corrente da rede elétrica, a plataforma Beaglebone Black<sup>®</sup> e a IHM.

Figura 3.1 – Sistema proposto completo



A ANEEL na resolução normativa nº 641/2014 - 6a revisão, define no módulo 8 que trata da Qualidade de Energia, as especificações mínimas a serem atingidas pelos equipamentos desenvolvidos para medição de qualidade de energia, que são [18]:

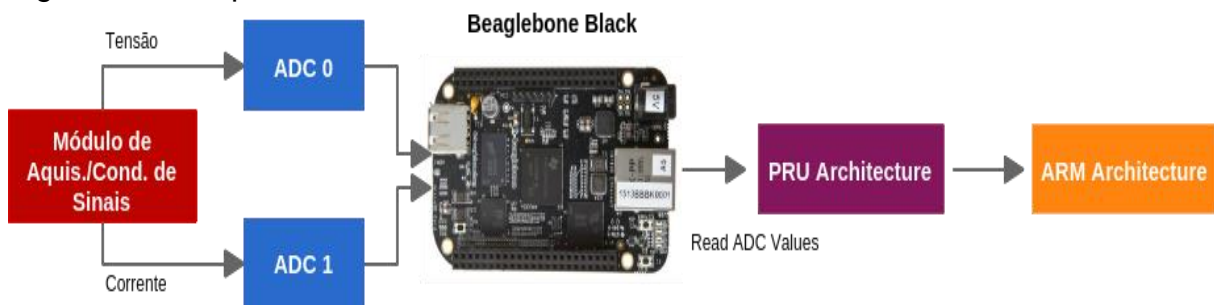
- 2.6.1.1 As leituras devem ser obtidas por meio de equipamentos que operem segundo o princípio da amostragem digital.
- 2.6.1.2 Os equipamentos de medição devem atender os seguintes requisitos mínimos:
  - Taxa amostral: 16 amostras/ciclo;
  - ADC com palavras de: 12 bits;
  - Precisão: até 1% da leitura.

As vantagens na utilização da Beaglebone Black<sup>®</sup> são: (a) além realizar as aquisições ela pode desempenhar as funções de controlador supervisorio; (b) a plataforma traz uma série de configurações e drivers importantes para o desenvolvimento de sistemas. Cita-se como exemplo os drivers de conversão, que convertem automaticamente os valores adquiridos pelas portas analógicas em

milivolts, facilitando no desenvolvimento do *Software*. Entretanto, estes drivers só estão disponíveis para acesso apenas na arquitetura ARM, ou seja, com a PRU não é possível utilizar estes drivers.

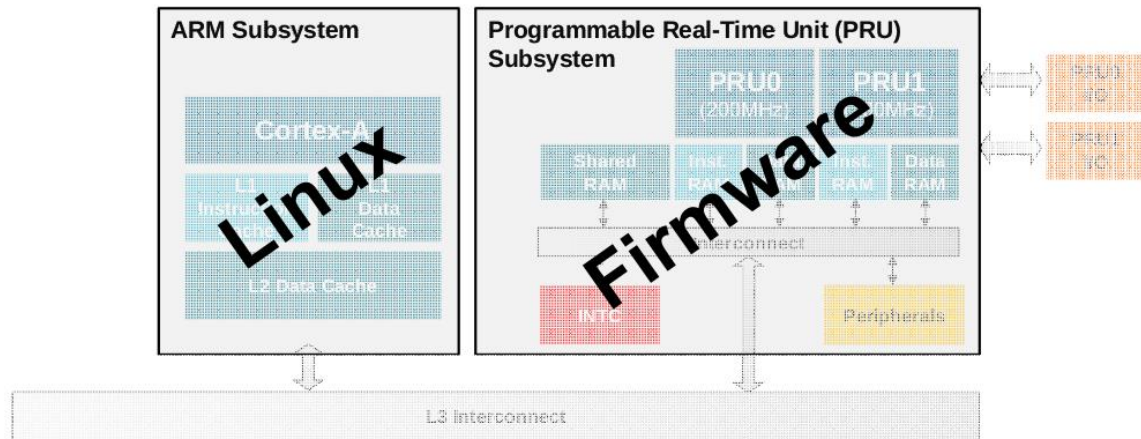
De acordo com as especificações apresentadas pela ANEEL, os sinais adquiridos pela plataforma são convertido para valores que vão de 0, menor valor, a 4095, maior valor. Após realizada a conversão do valor pelo ADC, este é mantido na plataforma do sistema operacional no ARM. Desta forma é possível convertê-lo para qualquer outro valor, alterando apenas a rotina do *software*. Visando aumentar o desempenho da plataforma, os valores são adquiridos pela PRU e em seguida transmitidos para o ARM, visto que apenas o ARM permite o acesso de uma comunicação externa como ilustra a Figura 3.2.

Figura 3.2 - Proposta de Leitura



Após os valores de tensão e corrente serem adquiridos e convertidos pela PRU, estes são enviados através da interconexão L3 cuja capacidade de transmissão é limitada apenas pela capacidade de memória da PRU. A interconexão L3 é responsável pela comunicação entre o Linux (ARM) e o Firmware (PRU) (Figura 2.12) para o ARM, que é onde o sistema operacional é executado. Possibilitando assim, a comunicação com outros sistemas e plataformas para análise ou captura das informações adquiridas pela PRU (Figura 3.3) [55].

Figura 3.3 - SoC Software Architecture



A necessidade de utilizar o módulo de aquisição e condicionamento de sinais deve-se ao fato que para a Beaglebone Black<sup>®</sup> realizar a leitura de maneira segura e correta, é necessário que o valor máximo de entrada na porta analógica seja de no máximo 1,8 V conforme as especificações técnicas da Tabela 2. Logo o módulo de aquisição e condicionamento de sinais é responsável por transformar uma tensão eficaz de entrada de 220 V em um valor de saída de 1,8 V.

Após realizada a aquisição e conversão dos sinais de tensão e corrente pela plataforma, então o valor é armazenado no sistema para que possa ser utilizado ou acessado.

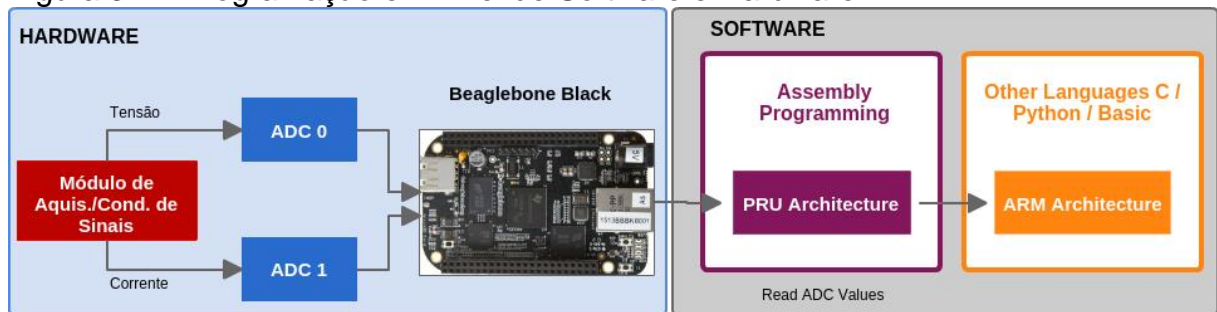
### 3.2 SOFTWARE E HARDWARE

De acordo com as informações reunidas e levantadas sobre as especificações de desenvolvimento de *Software* e *Hardware* de cada plataforma, este tópico destina-se a detalhar o desenvolvimento do sistema responsável pela aquisição de sinais de tensão e corrente pela plataforma. Para desenvolver a proposta é necessário realizar a comunicação entre o *Hardware* e o *Firmware*, conforme visto na Figura 3.3, implicando em duas etapas de desenvolvimento onde cada uma destas etapas será desenvolvida utilizando uma linguagem de programação específica.

Primeiramente é necessário adquirir os sinais de tensão e corrente da plataforma pelos ADCs da plataforma através PRU, em seguida, estes valores serão transferidos para o ARM. Para isso seja possível, é necessário desenvolver um

programa que a partir do ARM seja feita uma solicitação de informações pela memória compartilhada que é fornecida pela PRU, em seguida o ARM é responsável por armazenar essas informações, estas informações estão armazenadas em um arquivo de texto contendo os valores de tensão e corrente processados pela PRU, possibilitando a utilização e tratamento das informações da maneira que for cabível, como ilustra a Figura 3.4.

Figura 3.4 - Programação em nível de Software e Hardware

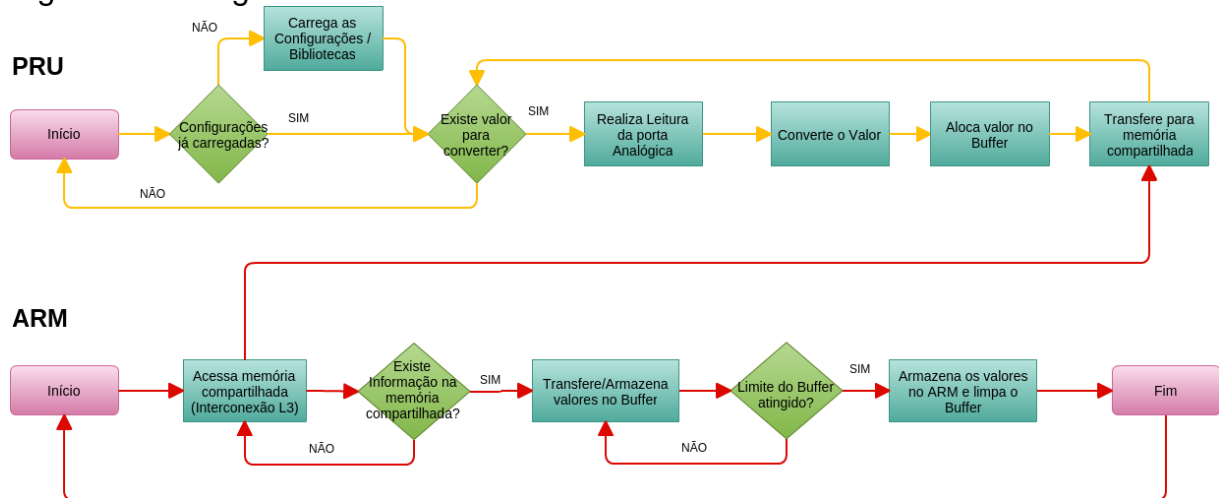


A plataforma traz consigo uma linguagem de programação nativa que é a *Bonescript*, entretanto o uso da PRU exige configuração e um nível de conhecimento de arquitetura e programação em sistemas operacionais embarcados relativamente elevado. Isto implica que, diferente da linguagem *Bonescript* que possui uma IDE de programação que facilita o desenvolvimento, para programar o *Software* da PRU, é necessário configurar a unidade com a linguagem nativa de programação da PRU (*Assembly*), realizando as configurações para que seja possível a utilização e configuração dos canais analógicos.

Para acessar a memória compartilhada da PRU, é necessário o desenvolvimento de outro programa que faça esta conexão de ponte entre o ARM (Linux) e a PRU (*Firmware*), no presente caso, foi utilizado um programa na linguagem *Basic*.

Para a conexão com a PRU, o ARM precisa acessar a interconexão L3 e verificar se existem informações na memória compartilhada, caso existam informações o *Software* é responsável por processar estas informações e armazenar no ARM para que possam ser tratadas como ilustra a Figura 3.5.

Figura 3.5 - Diagrama do Software a ser desenvolvido



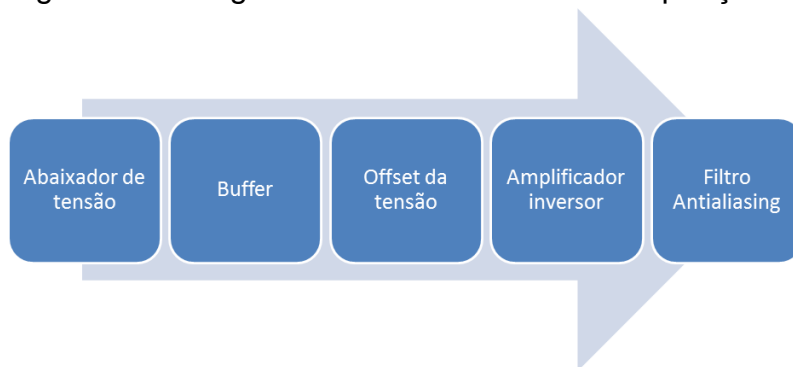
Conforme a Figura 3.5 para o desenvolvimento do *Software*, é necessário realizar a comunicação entre o ARM e a PRU e a configuração deste processo de comunicação é executada na primeira etapa da programação onde ocorre o carregamento das configurações e das bibliotecas necessárias para inicializar o módulo *TouchScreen* da plataforma e consequentemente, o ADC da Beaglebone Black<sup>®</sup> [55].

Existem diversas formas para configurar a PRU e uma delas pode ser por meio de bibliotecas que possuem uma série de configurações padrão preparadas para alguns objetivos específicos, o que facilita na utilização da PRU. Para configurar a PRU manualmente é necessário configurar ou criar as *Devices Trees*, cuja responsabilidade é descrever todas as configurações de *hardware* da plataforma.

Em se tratando do *Hardware*, o sistema é composto pelo módulo de aquisição e condicionamento de sinais que adequa os valores para os níveis de tensão de operação das entradas analógicas da Beaglebone Black<sup>®</sup>.

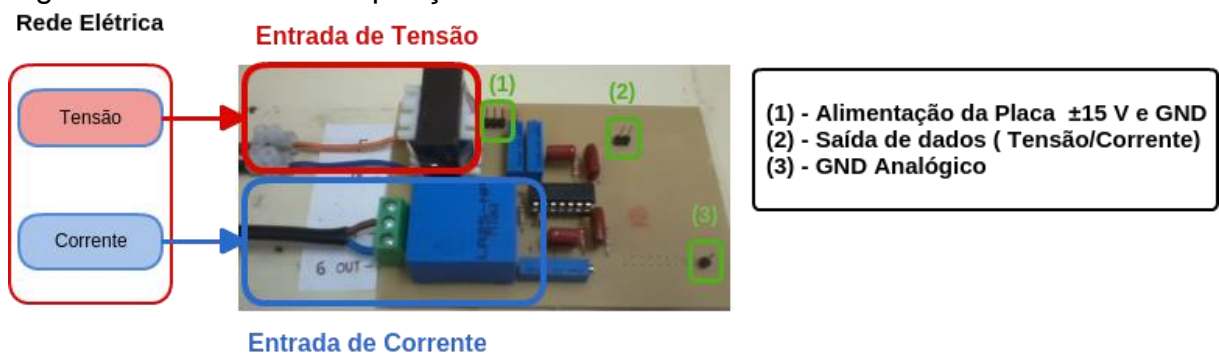
O sinal de tensão na rede elétrica tem forma senoidal com valor eficaz de 220 V com uma frequência de 60Hz. De acordo com essas especificações, não é possível adquirir a tensão e a corrente diretamente pela plataforma, por isso, é necessário a utilização do módulo de aquisição e condicionamento de sinais convertendo a tensão primária para uma tensão máxima de 1,8V que é a tensão limite suportada pela plataforma (ver Figura 3.6).

Figura 3.6 - Diagrama básico do módulo de aquisição de tensão



O módulo de aquisição de tensão apresentado na Figura 3.7 possui duas configurações básicas, estas configurações são responsáveis para que a tensão e a corrente de saída possam ser ajustadas de acordo com as especificações da plataforma.

Figura 3.7 - Módulo de Aquisição e condicionamento de sinais



Sendo assim além do módulo adequar os sinais para a plataforma Beaglebone Black<sup>®</sup>, é possível ajustar o módulo para trabalhar em outras plataformas de desenvolvimento.

## **4      RESULTADOS**



## 4 RESULTADOS

Este capítulo é responsável por descrever os resultados obtidos. Os valores obtidos pela plataforma nos experimentos serão comparados com os sinais de três equipamentos de referência: um multímetro, um alicate amperímetro e um analisador de qualidade de energia.

### 4.1 EQUIPAMENTOS UTILIZADOS PARA ESTUDO COMPARATIVO

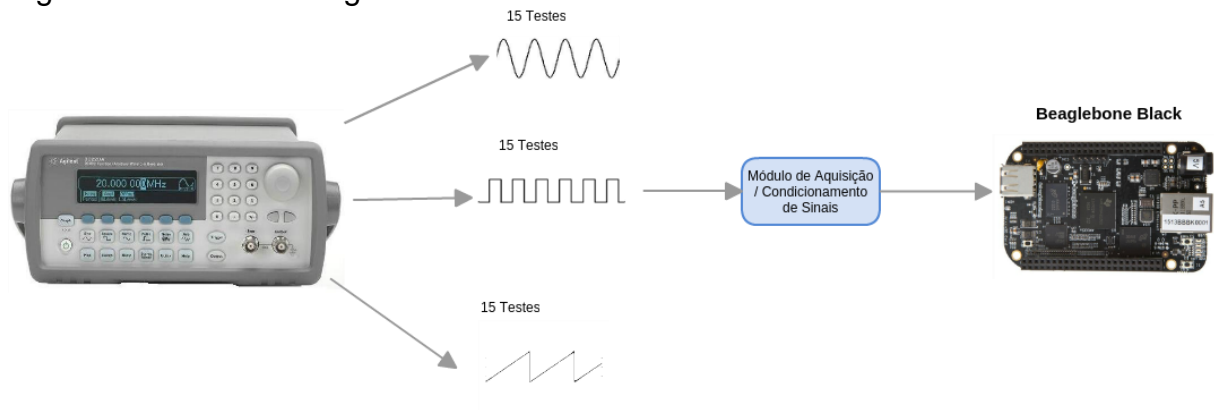
- Alicate Amperímetro Digital Profissional Et-3200 – Minipa. Padronização CAT. II padrão IEC 1010. Taxa de Amostragem: 3 Amostras/s. Corrente AC: com Faixa de 20 A, precisão de  $\pm 3\%$  e resolução de 10mA;
- Multímetro Digital HM-2030 – HIKARI. Padronização CAT IV 600V. Taxa de Amostragem: 3 Amostras/s. Tensão AC: Faixa de 750V, precisão de  $\pm 1.2\%$ , resolução 1V, impedância de entrada 10M $\Omega$ .
- Multímetro de precisão Keysight 34465<sup>a</sup>, Resoluções até 6½ dígitos, até 50.000 leituras/segundo, Faixas de tensão de 100 mV a 1.000 V, faixas de corrente de 1  $\mu$ A a 10 A.
- Osciloscópio DSO-X-2012A - Agilent InfiniiVision 2000 série-X.

### 4.2 EXPERIMENTO

De início a plataforma foi testada sem o módulo de aquisição e condicionamento de sinais. Antes de iniciar o desenvolvimento do *Software* com a PRU, a plataforma realizou entre 20 e 4400 leituras por segundo utilizando apenas o ADC do ARM, visto que não é possível controlar a frequência do ADC de maneira fixa no ARM, o que dificulta o uso para este tipo de aplicação.

Já com a PRU o aumento na quantidade de amostras foi significativamente maior como ilustra a Tabela 4, onde foi realizado 15 vezes a mesma leitura de três formas de onda (senóide, quadrada e rampa) fornecido pelo gerador de funções da Agilent modelo 33220A presente na Figura 4.1. Estes testes foram feitos para analisar e avaliar a capacidade da plataforma de realizar a aquisição de amostras de sinais de tensão ou corrente.

Figura 4.1 - Testes Beaglebone Black<sup>®</sup>



A plataforma possui 7 portas analógicas que podem ser utilizados pela PRU, porém existe apenas um ADC, o que implica que quanto maior o número de portas a serem utilizadas, menor será a quantidade de aquisições por ciclo, a Tabela 4 mostra a média da contagem de aquisições por ciclo, realizada 15 vezes em cada formato de onda.

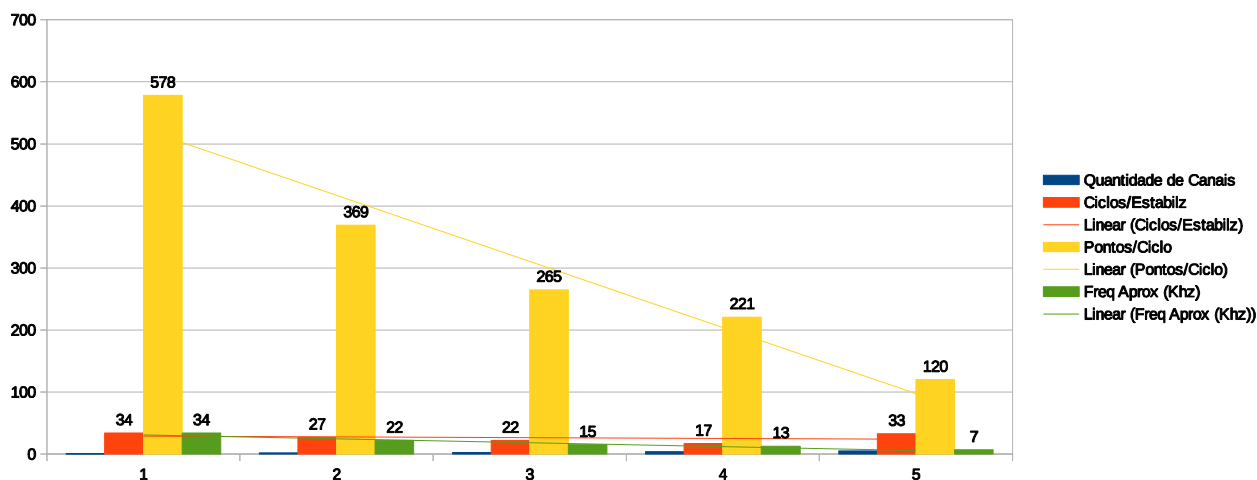
A plataforma Beaglebone Black<sup>®</sup> possui um recurso que visa proteger os componentes da plataforma, uma vez que se iniciado o módulo *Touchscreen*, a plataforma leva um tempo até que seja atingida a maior frequência do módulo, este tempo é chamado de ciclos de estabilização, que nada mais é a quantidade de ciclos com aquisição em baixa frequência.

Tabela 4 - Resumo do teste com 5 canais analógicos

Quantidade de Canais	Ciclos para Estabilização	Pontos por Ciclo	Frequência (KHz)
1	34	578	34
2	27	369	22
3	22	265	15
4	17	221	13
5	33	120	7

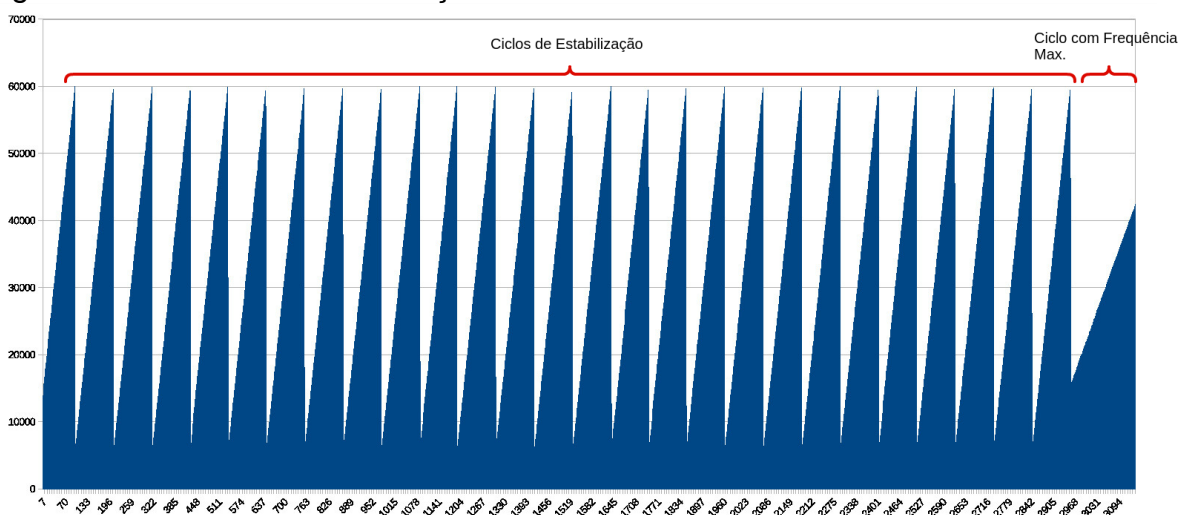
Na Figura 4.2 visualiza-se a quantidade de pontos por ciclo, aonde se constata que apesar de ser um valor crescente o número de aquisições por ciclo não é linear.

Figura 4.2 - Testes da Tabela 4



A plataforma por padrão já vem configurada para que a aquisição não se inicie de imediato com frequência máxima, mas aumentando a frequência depois de um certo número de ciclos, esse recurso visa não comprometer unidade da plataforma devido ao repentino consumo da unidade ao iniciar as aquisições. Este é um recurso obrigatório da PRU e pode atingir no máximo 35 ciclos, antes de iniciar a aquisição com a frequência máxima como é possível visualizar na Figura 4.3.

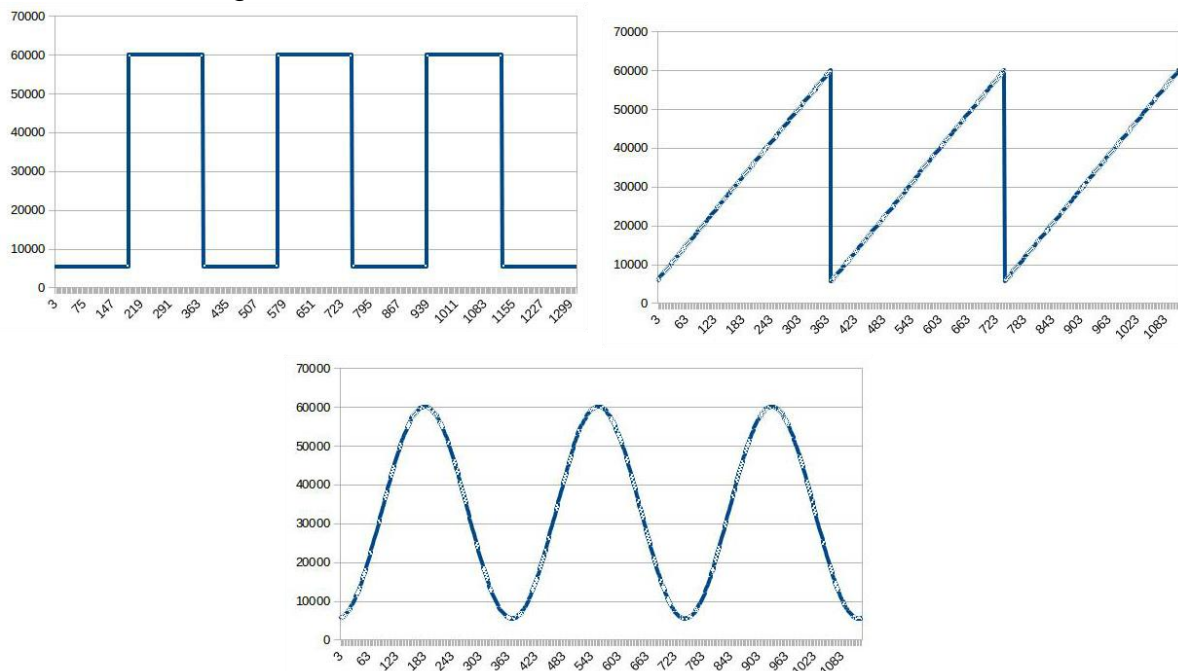
Figura 4.3 - Ciclos de Estabilização



Vale ressaltar que os ciclos de estabilização só irão acontecer enquanto as configurações da PRU estiverem sendo carregadas pela primeira vez, uma vez feita a estabilização, as aquisições feitas na sequência pela plataforma serão realizadas com a maior frequência possível para a quantidade de portas selecionadas. Também

foram realizados testes com 2 canais, um para tensão e um para corrente com um *offset* de 0,9 V, que gerou formas de ondas semelhantes as geradas pelo osciloscópio como ilustra a Figura 4.4.

Figura 4.4 - Formas de onda resultante dos sinais fornecidos pela plataforma Beaglebone Black<sup>®</sup>



Ambos os testes tiveram a mesma configuração, pois o objetivo era analisar a forma de onda e a quantidade de pontos por ciclo, visto a necessidade de comprovar a frequência de amostragem, frente à necessidade de adequar à plataforma as exigências da ANEEL.

#### 4.3 TESTE COM SINAIS DE TENSÃO E CORRENTE DA REDE ELÉTRICA

Antes de conectar o módulo de aquisição e condicionamento de sinais, foi utilizado o gerador de funções para emitir uma senóide na entrada do módulo e constatado que o sinal está de acordo com as expectativas para aquisição. Então, foram feitos os testes utilizando a rede elétrica como sinal de entrada do módulo de aquisição e condicionamento de sinais. Para certificar que os valores de saída correspondem aos da entrada, o sistema desenvolvido foi comparado com os equipamentos de referência apresentados no Tópico 4.1.

### 4.3.1 Sinais de Tensão

Antes de iniciar os testes com a rede elétrica, foi necessário testar a plataforma Beaglebone Black<sup>®</sup> com o módulo de aquisição e condicionamento de sinais utilizando o gerador de funções, em vista de analisar o sinal de saída apresentado pelo módulo de aquisição e condicionamento. Também foi necessário analisar qual o grau de desvio de fase na saída do módulo de aquisição e condicionamento de sinais para verificar se o sinal de saída estava com uma boa representação do sinal original. Foi constatado que o desvio foi de 2,2 graus no sinal de saída do módulo, implicando que o sinal possui uma boa representação do sinal original presente na entrada do módulo.

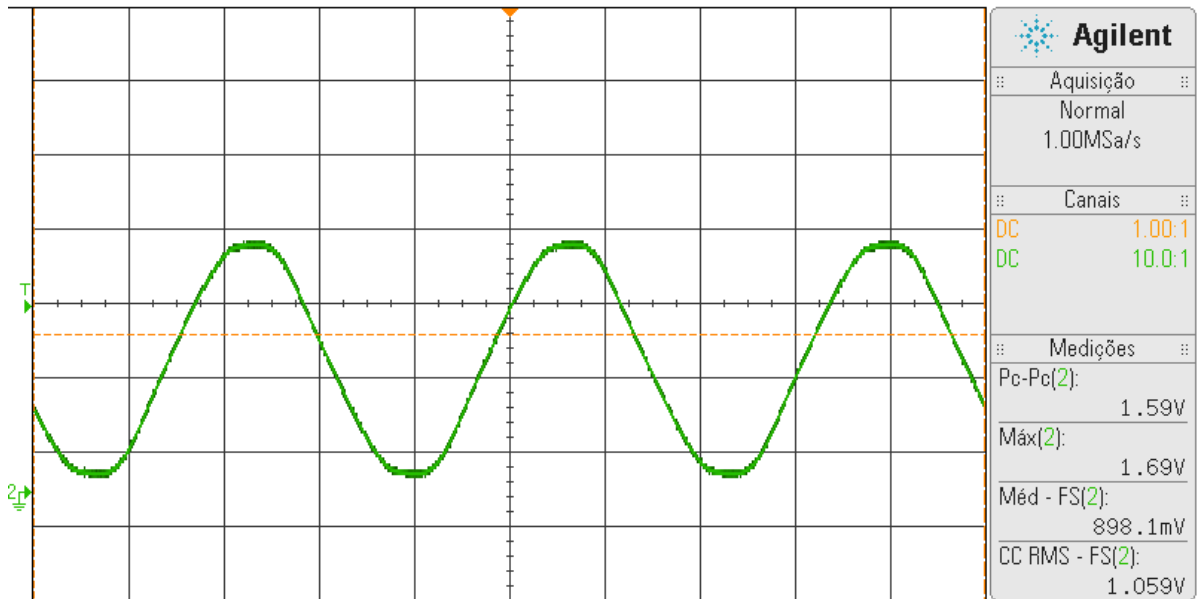
Conforme explicado, o objetivo da utilização do módulo de aquisição e condicionamento de sinais, é que o sinal da rede convencional possa ser adequado para a leitura pelas portas analógicas da plataforma, por isso, durante as medições dos sinais de tensão, todos os valores foram monitorados por meio do osciloscópio afim de acompanhar e comprovar a eficácia do sistema desenvolvido.

Então foi necessário ajustar o módulo de aquisição e condicionamento de sinais para que na sua saída fosse apresentado um sinal adequado para a plataforma Beaglebone Black<sup>®</sup>, para isso foi necessário ter que ajustar o módulo de tensão para as seguintes características:

- Amplitude de Pico a Pico de 1,7 V
- Tensão de *offset* de 0,9V

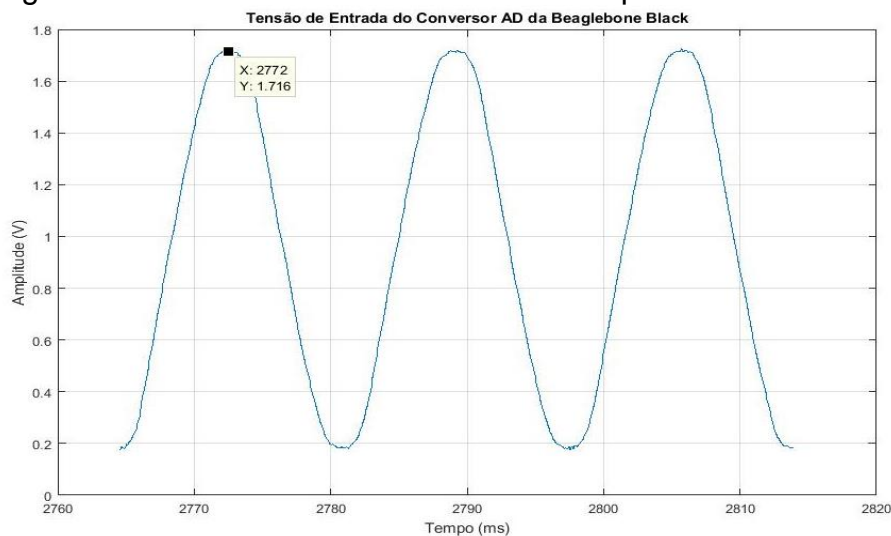
O valor da amplitude de pico a pico foi fixado em 1,7 V, devido a ausência de um circuito que proteja a plataforma caso, seja excedido a tensão limite de 1,8 V. Isto se deveu ao fato que o módulo demonstrou uma leve instabilidade no sinal de saída que variou entre 0 e 0,7 mV, erro apresentado pelo osciloscópio. Existindo assim a necessidade de verificar os sinais de saída do módulo antes que sejam conectados na Beaglebone Black<sup>®</sup>. A validação das informações foi feita com auxílio do osciloscópio que está presente na Figura 4.5. Foi constatado que o módulo apresentava uma pequena variação de tensão entre 1,69 V e 1,76 V, o que afirma que o sinal está correspondente aos padrões de aquisição da Beaglebone Black<sup>®</sup>.

Figura 4.5 - Tensão de saída fornecida pelo módulo de aquisição e condicionamento de sinais



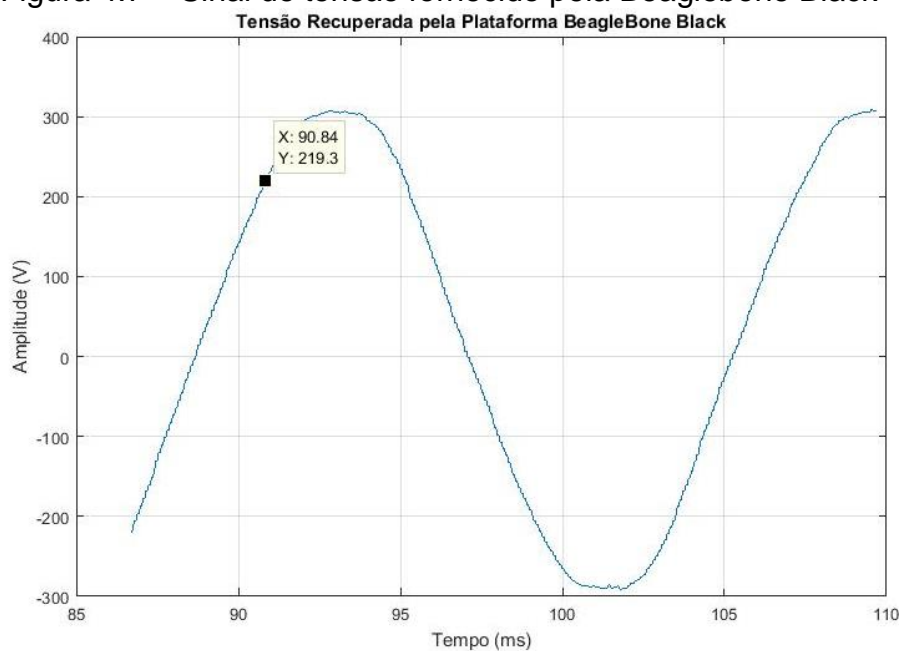
De acordo com a Figura 4.5 todos os requisitos de tensão de entrada são suportados pela Beaglebone Black<sup>®</sup>, então logo em sequência o módulo foi conectado a plataforma e o programa foi iniciado. Para que fosse feito o tratamento dos valores armazenados no ARM, estes foram tratados pelo *Software* MATLAB cujo sinal recuperado está na Figura 4.6, onde é possível visualizar o valor na amplitude de pico de 1,7 V confirmando a leitura do osciloscópio.

Figura 4.6 – Tensão de Entrada do ADC da plataforma



Depois de constatado que sistema se comportava da maneira esperada, ilustrando as formas de onda com as mesmas configurações apresentadas pelo osciloscópio, foi realizado o experimento com a carga resistiva (Figura 4.11), onde os novos valores que foram transportados da PRU para o ARM e em seguida transportados para o *Software* MATLAB, que por final possibilitou a visualização da tensão eficaz convertida a partir dos valores fornecidos pela plataforma, gerando a Figura 4.7.

Figura 4.7 – Sinal de tensão fornecido pela Beaglebone Black®



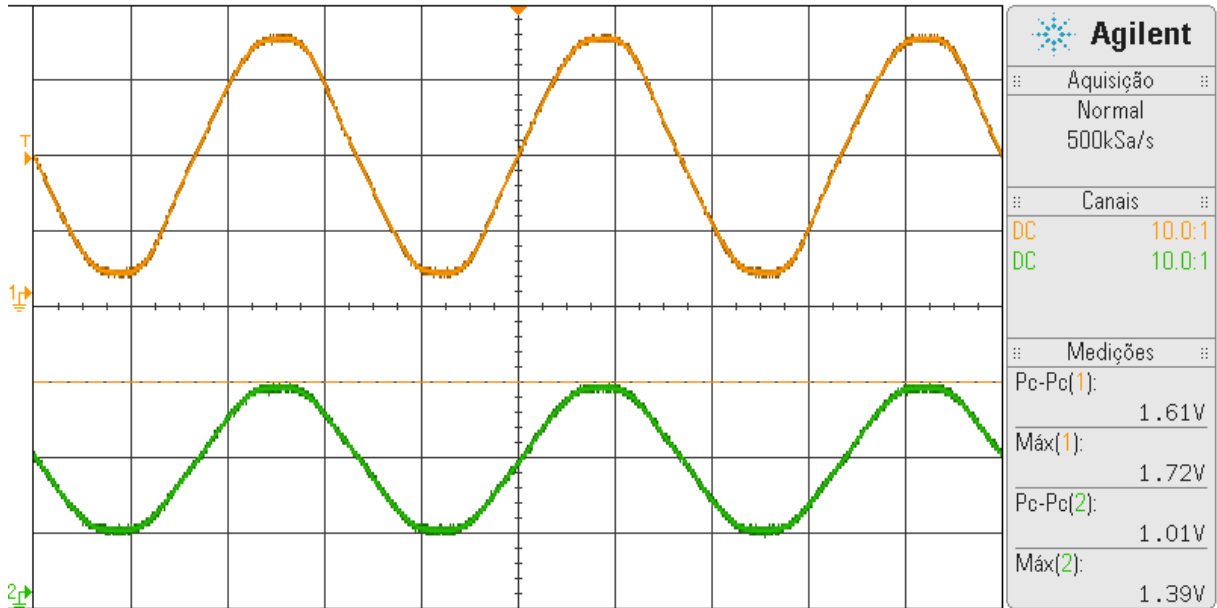
#### 4.3.2 Sinais de Corrente

Após testado a plataforma com sinais de tensão, a plataforma foi desconectada do módulo para analisar o sinal de corrente apresentado na saída do módulo. É importante ressaltar que o sinal de corrente ao ser condicionado pelo módulo é convertido em um sinal de tensão proporcional a esta corrente. Conforme especificações da BBB os ajustes para corrente foram realizados no módulo de aquisição e condicionamento de sinais, ajustando as configurações do sensor de corrente para as seguintes especificações:

- Amplitude de Pico a Pico de 1,4 V
- Tensão de *offset* de 0,4V

As configurações feitas no módulo de aquisição e condicionamento de sinais foram analisadas pelo osciloscópio (Figura 4.8), com o sinal de tensão (laranja) e corrente (verde).

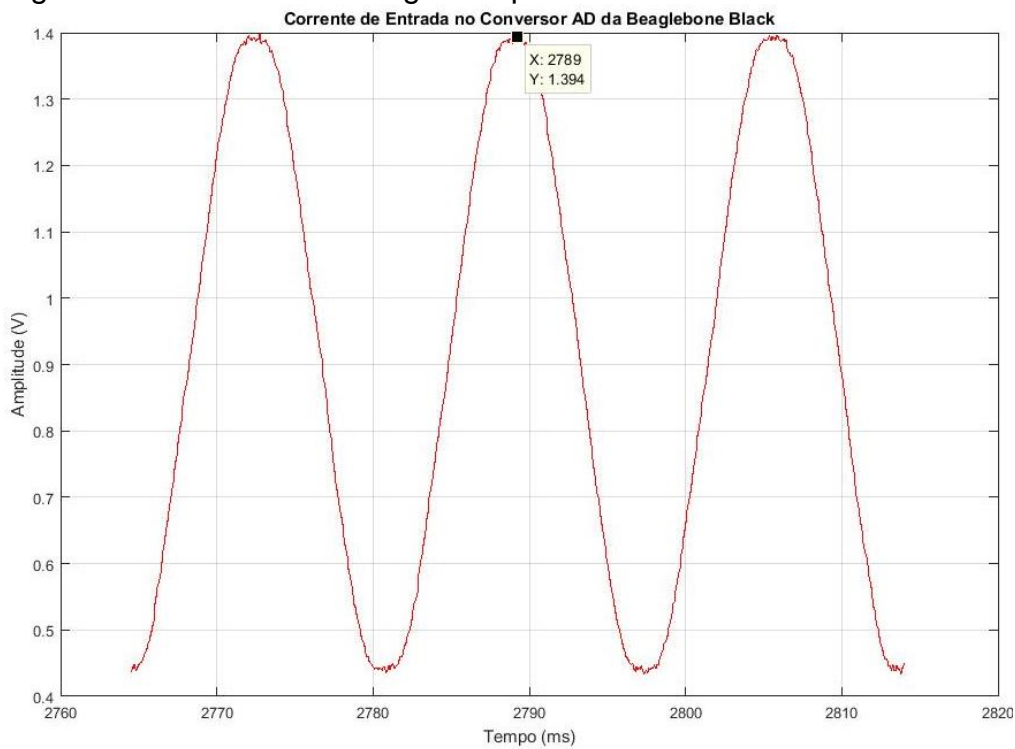
Figura 4.8 - Tensão e corrente de saída fornecida pelo módulo de aquisição e condicionamento de sinais



Após constatado a partir do osciloscópio que os valores de corrente e tensão estão compatíveis com as especificações da Beaglebone Black®, os valores estão aptos a serem adquiridos pela plataforma. É possível visualizar na Figura 4.9 o sinal de corrente recuperado que apresenta a amplitude máxima de 1,39 V, o mesmo sinal apresentado pelo osciloscópio.

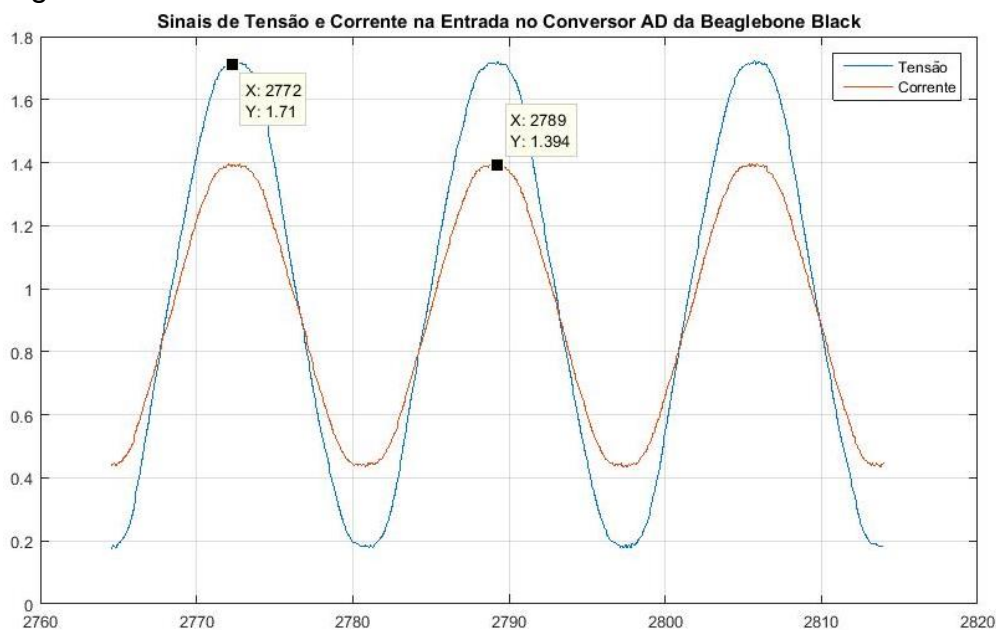


Figura 4.9 - Forma de onda gerada pelo sinal de Corrente armazenado no ARM



Pode-se observar na Figura 4.10, os sinais de tensão e corrente fornecidos pela plataforma, estes que são correspondentes aos da Figura 4.8, confirmando que o módulo e a plataforma estão passando uma boa representação dos sinais de entrada.

Figura 4.10 - Curva de Tensão e Corrente



#### 4.4 EXPERIMENTO COMPARATIVO

Para os testes foi utilizado uma carga resistiva com potência de 340W formada pela associação de 6 lâmpadas incandescentes como ilustra a Figura 4.11 e Figura 4.12.

Figura 4.11 - Análise do experimento com o multímetro e alicate amperímetro

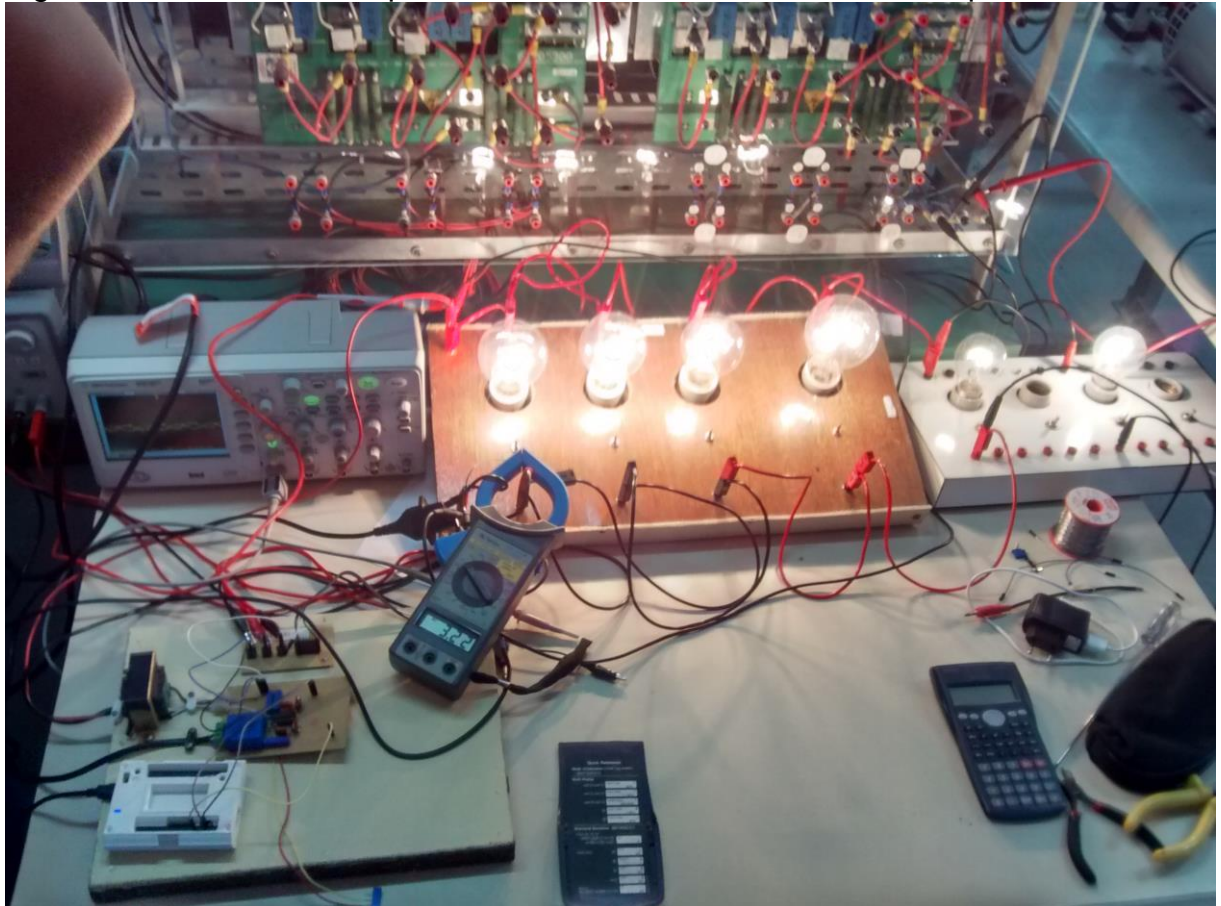
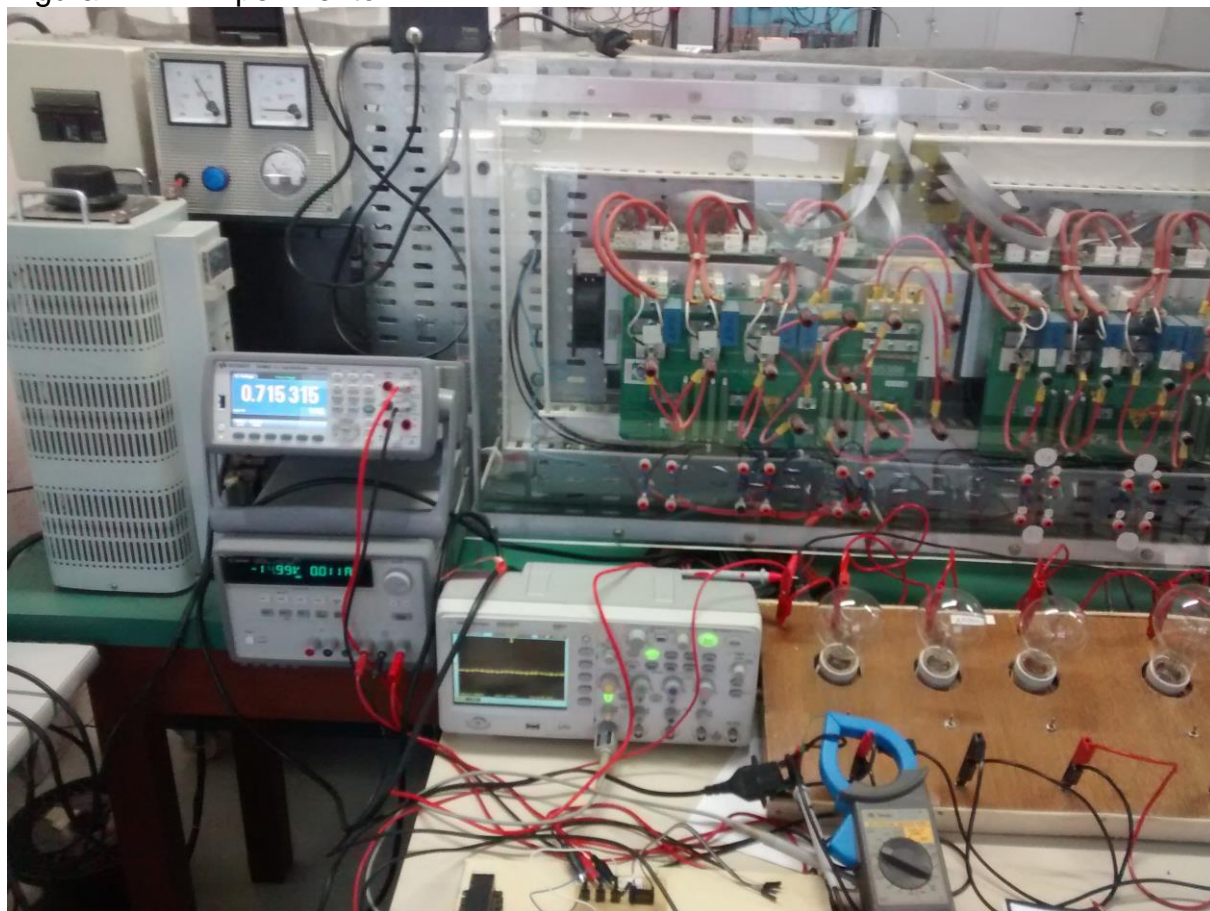


Figura 4.12 - Experimento



Com o propósito de analisar e comparar o sistema desenvolvido na Beaglebone Black<sup>®</sup> com algum equipamento de referência, para isso foi utilizado um alicate amperímetro minipa ET3200, o multímetro de bancada Keysight e o multímetro digital Hikari HK-M204 e a plataforma Beaglebone Black, o experimento foi feito realizando uma análise de medição a cada 10 volts na faixa de 0 V a 200 V e mais uma medição a cada 5 V na faixa de 200 V a 240 V, com o auxílio de um regulador de tensão, cujo valores estão presentes na Tabela 5.

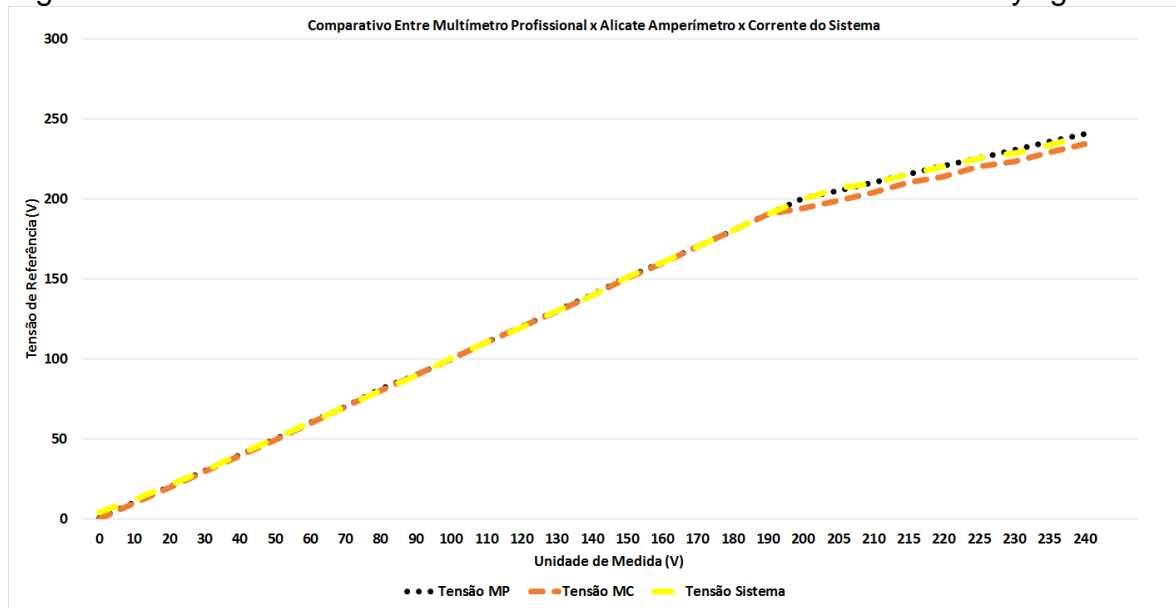
Tabela 5 - Comparativo da Beaglebone Black<sup>®</sup> com equipamentos de referência

Multímetro Keysight	Mult. Hikari	Tensão Eficaz (Beaglebone Black)	Corrente Eficaz (Beaglebone Black)	Alicate Amp.	Multímetro Keysight
0,32	0	3,8	0,09	0,002	0
10,22	9,6	11,9	0,9	0,835	0,92
20,16	19,6	21,24	1,13	1,15	1,18
30,3	29,7	30,01	1,35	1,24	1,40
40,02	39,5	40,61	1,53	1,40	1,59

50,01	49,5	50,32	1,72	1,56	1,78
60,12	59,6	60,32	1,86	1,71	1,95
70,4	69,9	70,26	2,05	1,85	2,12
80,85	80	80,01	2,18	1,95	2,28
90,1	89,7	89,38	2,37	2,11	2,42
100,07	99,7	100,3	2,48	2,32	2,57
110,5	110,2	110,39	2,64	2,35	2,72
120,2	119,9	119,5	2,77	2,46	2,84
130,08	129,7	129,93	2,89	2,56	2,97
140,1	139,3	139,37	2,99	2,67	3,09
150,6	150,5	150,62	3,11	2,78	3,22
160,2	159,9	160,12	3,22	2,87	3,32
170,3	170,2	170,21	3,3	2,97	3,44
180	179,8	180,07	3,43	3,06	3,55
190,3	190,3	190,26	3,58	3,15	3,66
200,1	194	199,91	3,63	3,24	3,76
205	199	206,4	3,71	3,28	3,81
210,4	204	209,85	3,79	3,33	3,86
215,5	210	215,41	3,86	3,38	3,91
220,4	214	219,96	3,92	3,41	3,96
225,5	220	225,28	3,98	3,46	4
230,3	223	228,3	4,01	3,49	4,05
235,5	229	233,65	4,05	3,54	4,1
240,2	234	239,27	4,1	3,58	4,18

De acordo com a tabela podemos visualizar na Figura 4.13, o estudo comparativo entre a tensão eficaz de saída e os equipamentos de referência e na Figura 4.12 o estudo comparativo entre os valores de corrente.

Figura 4.13 - Tensão de Saída BBB / Multímetro Hikari / Multímetro Keysight

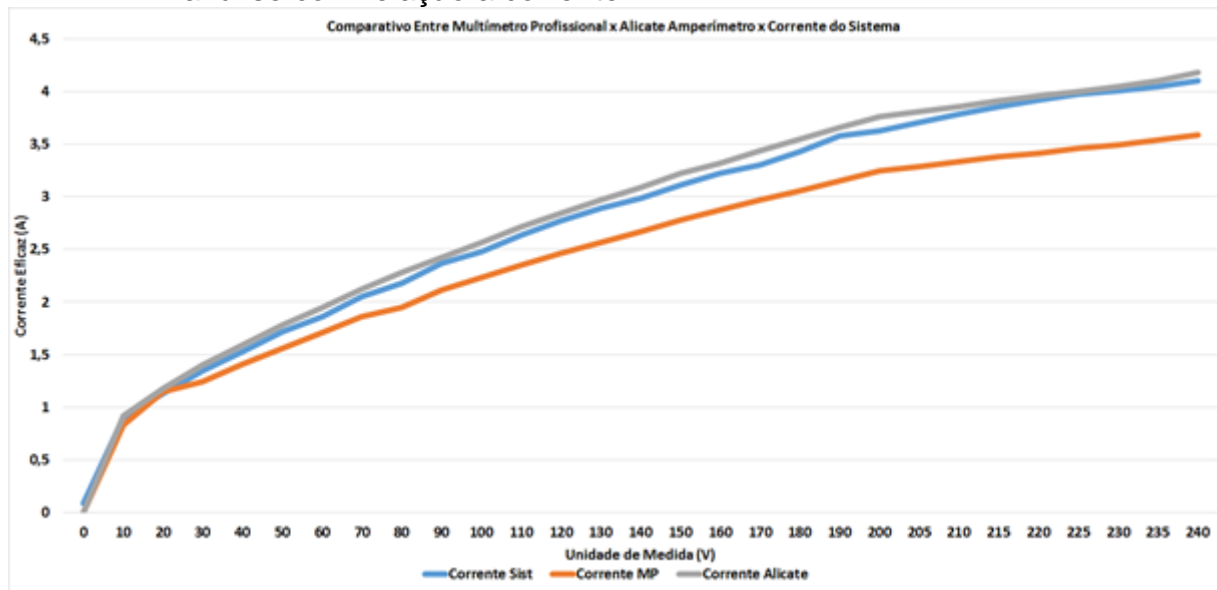


De acordo com o gráfico é visível que as aquisições realizadas pela plataforma estão quase em plena concordância com os equipamentos de referência, salve exceção, das análises dos valores de tensão entre 0 V e 10 V em que no sistema apresentou uma tensão eficaz de 4 V. Entretanto a proposta deste trabalho visa analisar os valores de tensão e corrente da rede elétrica convencional, tornando este problema, de fato, pouco relevante.

O mesmo processo foi realizado para as leituras de corrente, realizou-se uma leitura de cada um dos sistemas, analisando primeiramente o sinal da corrente eficaz de saída da plataforma, depois o alicete e depois o analisador de qualidade de energia.



Figura 4.14 - Comparativo entre o sinal da Beaglebone e os equipamentos de análise com relação a corrente



Então, é possível deduzir, que de acordo com a linearidade apresentada pelo sistema da Beaglebone Black<sup>®</sup>, a plataforma possui um forte indicativo com relação a sua funcionalidade, comprovando a sua eficiência frente a equipamentos já comercializados em grande escala no país.

## **5 CONCLUSÕES**

## 5 CONCLUSÕES

De acordo com o estudo comparativo feito, a plataforma Beaglebone Black<sup>®</sup> foi a que apresentou um conjunto maior de características desejáveis para o desenvolvimento deste projeto. Esta plataforma possui uma PRU que possibilitou um maior número de aquisições por ciclo com o uso do ADC, uma vez que as outras plataformas possuem desempenho inferior ou não tem o ADC interno. O uso da PRU foi de fundamental importância, visto que, sem a utilização da unidade não é possível garantir uma alta e constante frequência de aquisições, já com a PRU a frequência é definida pela quantidade de portas utilizadas pelo ADC, implicando em uma maior resolução e robustez do sistema. Além disso, também foi levado em consideração, fatores como: capacidade de memória, capacidade de armazenamento, capacidade de processamento e custo.

Diante deste cenário o projeto desenvolvido demonstrou-se como uma possível ferramenta de análise de qualidade de energia, pois nos testes a Beaglebone Black<sup>®</sup> se comportou de maneira bastante semelhante aos equipamentos já presentes no mercado utilizados para fazer a comparação, que foram: o Alicata Amperímetro Et-3200 Minipa, Multímetro Digital HM-2030 HIKARI, Osciloscópio DSO-X-2012A e o Analisador de Qualidades de Energia Trifásica - Fluke 434/435.

Uma vantagem é que a plataforma Beaglebone Black<sup>®</sup> pode ser explorada como um sistema de controle supervisor, tornando possível que a plataforma controle outros sistemas. Esta vantagem se deve ao fato de que existe um sistema operacional Linux embarcado, possibilitando a execução de programas e outros cálculos enquanto a PRU faz as transferências para o ARM. Este processo de comunicação entre PRU e o ARM não ocupa uma parcela considerável do processamento de 1 GHz do ARM, deixando assim uma grande parte do sistema ainda ocioso, podendo ser utilizado para diversas aplicações, a exemplo do sistema de controle supervisor. Plataformas mais baratas como a *Texas Tivaware*, não tem capacidade suficiente para executar um sistema operacional embarcado na plataforma, o que reforça ainda mais a possibilidade da utilização da plataforma Beaglebone Black<sup>®</sup> como um produto de grande potencial e alcance de mercado.



O projeto teve como intenção atender as exigências normativas da ANEEL, que descreve em normas os limites de adequação para um sistema de análise de qualidade de energia, possibilitando assim que a Beaglebone Black<sup>®</sup> possa ser desenvolvida como um produto destinado ao mercado com intenção de comercialização.

Por fim, a plataforma atendeu as necessidades do projeto, e se apresenta com uma ferramenta funcional que poderá ser tomada como ponto de partida em outros trabalhos.

## 5.1 SUGESTÕES PARA TRABALHOS FUTUROS

É sugerido por este trabalho, analisar o comportamento da plataforma durante as aquisições em diferentes ambientes de trabalho, tais como, temperatura, humidade, sol e outros, visto que para tornar este sistema em um produto, é necessário que ele apresente bom desempenho em diferentes cenários.

O sistema também poderá ser expandido para uma versão trifásica, se fazendo necessária a utilização de 3 módulos de aquisição e condicionamento de sinais aliados a Beaglebone Black<sup>®</sup>, com intenção de avaliar o comportamento da plataforma.

Um ponto importante que deve ser ressaltado é que a PRU pode ser configurada com maior exatidão por meio das configurações nas *device trees* que aumentaria ainda mais o desempenho das aquisições dos sinais de tensão e corrente pelo ADC, porém essas configurações ainda exigem elevado nível de conhecimento da arquitetura ARM e profundo conhecimento sobre o subsistema da PRU.

Para aprimorar o sistema como um todo, é sugerido realizar a comunicação de informações entre a Beaglebone Black<sup>®</sup> e sistemas de controle supervisorio por meio de comunicação sem fio de longo alcance, possibilitando assim realizar o monitoramento distribuído das redes elétricas de maneira mais eficiente em locais mais distantes, aplicando assim o conceito das REIs.

## REFERÊNCIAS

- [1] GUNGOR, Vehbi C. et al. **Smart grid technologies: communication technologies and standards**. Industrial informatics, IEEE transactions on, v. 7, n. 4, p. 529-539, 2011.
- [2] PALENSKY, Peter; DIETRICH, Dietmar. **Demand side management: Demand response, intelligent energy systems, and smart loads**. Industrial Informatics, IEEE Transactions on, v. 7, n. 3, p. 381-388, 2011.
- [3] AMIN, S. Massoud; WOLLENBERG, Bruce F. **Toward a smart grid: power delivery for the 21st century**. Power and Energy Magazine, IEEE, v. 3, n. 5, p. 34-41, 2005.
- [4] MINISTÉRIO DE MINAS E ENERGIA. **Relatório Smart Grid**. Disponível em: <[http://www.mme.gov.br/mme/galerias/arquivos/acoes/Energia/Relatxrio\\_GT\\_Smart\\_Grid\\_Portaria\\_440-2010.pdf](http://www.mme.gov.br/mme/galerias/arquivos/acoes/Energia/Relatxrio_GT_Smart_Grid_Portaria_440-2010.pdf)> 20-36 p. 2010. Acesso em 11 de Setembro de 2014
- [5] AMIN, S. Massoud. **The smart-grid solution: Massoud Amin outlines how the United States should make its electricity infrastructure self-healing to avoid massive power failures**. ISSN [0028-0836] , 2013 vol:499 iss:7457 pg:145.
- [6] US Department of Energy. **Economic Impact of Recovery Act Investments in the Smart Grid (DOE, 2013)**. US. Department of Energy.
- [7] KICKHAM. F. Victoria. **Speed Time-To-Market With New Design Tools**. Global Purchasing. Junho 2014. Electronic design. vol:62 iss:6 pg:47. ISSN [0013-4872]
- [8] MANDELMAN, Marcio. **Análise Crítica da Matriz Energética Brasileira e a Implementação de "Smart Grid"**. Universidade de Taubaté. Dissertação de Mestrado, 2011
- [9] CPFL, Energia. **Seminário Internacional Portugal - Brasil: Smart Cities & Smart Grids no Setor Elétrico**. Rio de Janeiro, 25 de outubro de 2013
- [10] BANZI, Massimo. **Primeiros Passos com Arduino** . ISBN: 978-85-7522-290-4. Novatec Editora, São Paulo, 2011.
- [11] ARDUINO. **Arduino Homepage**. Arduino, 2014. Disponível em: <<http://www.Arduino.cc>> Acesso em: 12 de Setembro de 2014.
- [12] MCROBETS, Michael. **Arduino Básico**. Novatec Editora, São Paulo, 2011. ISBN: 978-85-7522-274-4
- [13] DICOLA, Tony. **Embedded Linux Board Comparison**. Adafruit Industries. 16p. Last updated on 2014-05-06.

- [14] ARDUINO. **Arduino Yun**. Arduino, 2014. Disponível em: <<http://arduino.cc/en/Main/ArduinoBoardYun?from=Products.ArduinoYUN>> Acesso em: 12 de setembro de 2014
- [15] MALINOWSKI, Aleksander; YU, Hao. **Comparison of embedded system design for industrial applications**. Industrial Informatics, IEEE Transactions on, v. 7, n. 2, p. 244-254, 2011.
- [16] RASPBERRY. **What is a Raspberry Pi?**. Disponível em: <<http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>> Acesso em: 13 de setembro de 2014
- [17] OLIVEIRA, Ricardo Mercês de. **Raspberry Pi - Conceito & Prática**. ISBN: 978-85-399-0476-1. Rio de Janeiro: Editora Ciência Moderna Ltda., 2013.
- [18] ANEEL. **Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST Módulo 8**. Disponível em: <[http://www.aneel.gov.br/arquivos/PDF/Modulo8\\_F.pdf](http://www.aneel.gov.br/arquivos/PDF/Modulo8_F.pdf)> Acessado em 20 de setembro de 2014.
- [19] BEAGLEBOARD. **Beaglebone Black**. Disponível em: <<http://beagleboard.org/black>>. Acessado em 21 de setembro de 2014.
- [19] RICHARDSON, Matt. **Getting Started With Beaglebone**. ISBN: 978-1-449-34537-2. October 2013: First Edition. Sebastopol, 2013.
- [20] ELINUX. **Beaglebone Black Features**. Disponível em: <[http://www.elinux.org/Beagleboard:BeagleboneBlack#Beaglebone\\_Black\\_Features](http://www.elinux.org/Beagleboard:BeagleboneBlack#Beaglebone_Black_Features)>. Acessado em 21 de setembro de 2014.
- [21] INTEL. **Intel Galileo Board User Guide**. Disponível em: <<http://www.Intel.com/content/dam/www/public/us/en/documents/guides/galileo-user-guide.pdf>>. Março de 2014. Acessado em: 22 de setembro de 2014.
- [22] Arduino. **Arduino Certified**. Disponível em: <<http://Arduino.cc/en/ArduinoCertified/Products>> Acessado em: 22 de setembro de 2014.
- [23] INTEL . **Intel Quark SoC X1000 Board Support Package (BSP) Build Guide**. Disponível em: <<http://www.Intel.com/content/dam/www/public/us/en/documents/guides/galileo-quark-x1000-bsp-guide.pdf>> pg: 5-7. Acessado em: 22 de setembro de 2014.
- [24] ATHEROS. **AR9331 Highly-Integrated and Cost Effective IEEE 802.11n 1x1 2.4 GHz SoC for AP and Router Platforms**. Disponível em: <<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Dev/Arduino/Boards/AR9331.pdf>>. 2010. Acessado em: 23 de setembro de 2014.
- [25] ATMEL. **8-bit Microcontroller with 16/32K Bytes of ISP Flash and USB Controller ATmega16U4 & ATmega32U4**. Disponível em:

<<http://dlmh9ip6v2uc.cloudfront.net/datasheets/Dev/Arduino/Boards/ATMega32U4.pdf>>. 2010. Acessado em: 23 de setembro de 2014.

[26] ADAFRUIT. **Arduino Yun**. Disponível em: <<http://www.adafruit.com/products/1498>>. 2014. Acessado em: 23 de setembro de 2014.

[27] SPARKFUN. **Arduino Yun**. Disponível em: <<https://www.sparkfun.com/products/12053>>. 2014. Acessado em: 23 de setembro e 2014. (Preço)

[28] COLEY, Gerald. **Beaglebone Black System Reference Manual**. Disponível em: < [http://www.adafruit.com/datasheets/BBB\\_SRM.pdf](http://www.adafruit.com/datasheets/BBB_SRM.pdf)>. 2013. Acessado em: 23 de setembro de 2014.

[29] ARCHLINUX, Arm. **Beaglebone Black**. Disponível em: < <http://archlinuxarm.org/platforms/armv7/ti/Beaglebone-black>>. 2014. Acessado em: 23 de setembro de 2014.

[30] INTEL. **Product Brief Intel Galileo Board**. Document: 329680-003US, 2013. Disponível em: <<http://www.intel.com.br/content/www/br/pt/embedded/products/galileo/galileo-g1-product-brief.html>>. Acessado em 24 de setembro de 2014.

[31] INTEL. **Intel Galileo Board**. Document: 329685-004US, 2013. Disponível em: <[http://www.silica.com/fileadmin/02\\_Products/Productdetails/Intel/SILICA\\_Intel\\_Galileo-gs.pdf](http://www.silica.com/fileadmin/02_Products/Productdetails/Intel/SILICA_Intel_Galileo-gs.pdf)>. Acessado em 24 de setembro de 2014.

[32] VAILLIET, Nicolas. **Intel Do-It-Yourself Challenge Lab 2: Intel Galileo's Linux side. Intel Software Academic Program**. Disponível em: < [http://www.Intel-software-academic-program.com/courses/diy/Intel\\_Academic\\_-\\_DIY\\_-\\_InternetOfThings/IntelAcademic\\_IoT\\_Lab\\_02\\_Linux\\_side.pdf](http://www.Intel-software-academic-program.com/courses/diy/Intel_Academic_-_DIY_-_InternetOfThings/IntelAcademic_IoT_Lab_02_Linux_side.pdf)>. 2014. Acessado em 24 de setembro de 2014.

[33] ADAFRUIT, Products. **Intel Galileo Development Board - Arduino Certified**. Disponível em: <<http://www.adafruit.com/products/1637>> Acessado em: 24 de setembro de 2014.

[34] RASPBERRY PI. **Faqs**. Disponível em: < <http://www.raspberrypi.org/help/faqs/>>. Acessado em: 24 de setembro de 2014.

[35] CHACOS, Brad. **The battle of the pint-size PCs: they're incredibly small and surprisingly powerful. Here's what these tiny devices can do, and how they might meet your needs (Forward)**. PC World. ISSN [0737-8939]. 2013.

[36] MYRASPBERRYPI. **Raspberry-pi technical specifications**. Disponível em: <<http://www.myraspberry-pi.org/>>. Acessado em 24 de setembro de 2014

[37] THO, P. Bá. **Intel Galileo board: Quick comparison with Raspberry Pi an Arduino Yun.** 2014. Disponível em: <<http://blog.siliconstrait.com/Intel-galileo-review/>>. 2014. Acessado em 25 de setembro de 2014.

[38] MAYER, F. Uwe. **Nbench.** Disponível em: <<http://www.novellshareware.com/info/nbench.html>>. 2014. Acessado em: 25 de setembro de 2014.

[39] ARM. **Cortex-A8 Technical Reference Manual Revision: r3p2.** Disponível em: <<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0344k/ch16s07s01.html>>. 2014. Acessado em 03 de outubro de 2014.

[40] ARDUINO. **Guide to the Arduino Yún.** Disponível em: <<http://Arduino.cc/en/Guide/ArduinoYun#toc3>>. Acessado em: 03 de outubro de 2014.

[41] COOPER, Justin. **Beaglebone Black: Installing Operating Systems.** 2014. Disponível em: <<http://learn.adafruit.com/downloads/pdf/Beaglebone-black-installing-operating-systems.pdf>>. Acessado em: 04 de outubro de 2014.

[42] BEAGLEBOARD. **Getting Started with Beaglebone & Beaglebone Black.** Disponível em: <<http://beagleboard.org/getting-started>>. Acessado em: 04 de outubro de 2014.

[43] TEXAS INSTRUMENTS. **AM335x Sitara™ Processors.** Disponível em: <<http://www.ti.com/lit/ds/symlink/am3356.pdf>>. Acessado em: 04 de outubro de 2014.

[44] LINUX FOUNDATION. **Yocto Project.** Disponível em: <<https://www.yoctoproject.org/about>>. Acessado em: 04 de outubro de 2014.

[45] INTEL. **Intel Galileo Datasheet.** Disponível em: <[https://communities.intel.com/servlet/JiveServlet/downloadBody/21835-102-525148/Galileo\\_Datasheet\\_329681\\_003.pdf](https://communities.intel.com/servlet/JiveServlet/downloadBody/21835-102-525148/Galileo_Datasheet_329681_003.pdf)>. Acessado em: 04 de outubro de 2014.

[46] RASPBERRY PI. **TECHRADAR COMPARES FIVE RASPBERRY PI OPERATING SYSTEMS.** Disponível em: < <http://www.raspberrypi.org/techradar-compares-five-raspberry-pi-operating-systems/> > Acessado em: 05 de outubro de 2014.

[47] RASPBERRY PI. **Operating Systems Download.** Disponível em: <<http://www.raspberrypi.org/downloads/>>. Acessado em: 05 de outubro de 2014.

[48] ARM. **ARM1176 Processor.** Disponível em: <<http://www.arm.com/products/processors/classic/arm11/arm1176.php>>. Acessado em 05 de outubro de 2014.

[49] BEAGLEBOARD. **Bonescript Library.** Disponível em: <<http://beagleboard.org/Support/BoneScript>>. Acessado em: 06 de outubro de 2014.

---

[50] INTEL. **Galileo Datasheet**. Order Number: 329681-003US. Disponível em: <[https://communities.intel.com/servlet/JiveServlet/downloadBody/21835-102-5-25148/Galileo\\_Datasheet\\_329681\\_003.pdf](https://communities.intel.com/servlet/JiveServlet/downloadBody/21835-102-5-25148/Galileo_Datasheet_329681_003.pdf)> 2013. Acessado em 23 de setembro de 2014.

[51] J. A. Bright and Wei-Jen Lee, "**Integrated Monitoring, Protection, and Control Systems for Industrial and Commercial Power Systems** [Artigo]," IEEE Transactions on Industry Application., vol. Vol. 36, Jan/Feb 2000.

[52] SALVADORI. F., et al., "**Monitoring in Industrial Systems Using Wireless Sensor Network With Dynamic Power Management.**" IEEE Transactions on Instrumentation and Measurement., vol. 58, September 2009.

[53] LINS. Alisson. **Desenvolvimento de um sistema para aquisição e processamento de sinais utilizando a plataforma TIVAWARE para aplicações em Smart Grid.** [Dissertação de Mestrado]. 2015

[54] Khosrow Moslehi and Ranjit Kumar, "A Reliability Perspective of the Smart Grid.," IEEE Transaction on Smart Grid, vol. 1, no. 1, pp. 57-64, June 2010.

[55] AM335x Technical Reference manual. Texas Instruments. 2015.

[56] BIRKETT, R. Enhancing Real-time Capabilities with the PRU. 2014.

[57] BEAGLEBOARD, Foundation. Disponível em: <<http://beagleboard.org/project/libpruio/>>. Acessado em 30 Dezembro de 2015

Arquivo responsável pela PRU

.origin 0 // offset of the start of the code in PRU memory  
 .entrypoint START // program entry point, used by debugger only

#include "ADCCollector.hp"

.macro RESETADC 1

//Enable ADC STEPCONFIG 1-7

MOV r2, 0x44E0D054

MOV r3, 0x000000FE

SBBO r3, r2, 0, 4

MOV r5, 8 //Offset to pass in num\_samples,num\_ads,...

MOV r8, 0 //current count

.endm

.macro READADC 2

READ:

//Data avail?

MOV r2, 0x44E0D0E4

LBBO r3, r2, 0, 4

QBEQ READ, r3, 0

//Read ADC from FIFO

MOV r2, 0x44E0D100

LBBO r3, r2, 0, 4

//Add address counting

ADD r5, r5, 4

//Write ADC to PRU Shared RAM

SBCO r3, CONST\_PRUSHAREDGRAM, r5, 4

ADD r8,r8,1 //current count

QBNE READ, r8, r6

ADD r9,r9,1

QBEQ END, r9, r7

//Tell host to read

MOV r3, 1

SBCO r3, CONST\_PRUSHAREDGRAM, 8, 4

//Wait for host to read

WAIT:

LBCO r3, CONST\_PRUSHAREDGRAM, 8, 4

QBNE WAIT, r3, 0

RESETADC

## QBA READ

END:

//Tell host to read

MOV r3, 1

SBCO r3, CONST\_PRUSHAREDDRAM, 8, 4

//Disable TSC\_ADC\_SS

//Set Writable

MOV r2, 0x44E0D040

MOV r3, 0x00000004

SBBO r3, r2, 0, 4

//Send event to host program

MOV r31.b0, PRU0\_ARM\_INTERRUPT+16

HALT

.endm

START:

// Enable OCP master port

LBCO r0, CONST\_PRUCFG, 4, 4

CLR r0, r0, 4

SBCO r0, CONST\_PRUCFG, 4, 4

//C28 will point to 0x00012000 (PRU shared RAM)

MOV r0, 0x00000120

MOV r1, CTPPR\_0

ST32 r0, r1

//Reset SYSConfig Register to 0

MOV r2, 0x44E0D010 // load register address

MOV r3, 0x00000000

SBBO r3, r2, 0, 4 // set register

//Wait for ADC to be idle

AdcIdle:

MOV r2, 0x44E0D044 // load register address

QBBS AdcIdle, r2, 5

//Disable TSC\_ADC\_SS

//Set Writable

MOV r2, 0x44E0D040

MOV r3, 0x00000004

SBBO r3, r2, 0, 4

//Enable ADC STEPCONFIG 1-7

MOV r2, 0x44E0D054

MOV r3, 0x000000FE

SBBO r3, r2, 0, 4



```
//Init ADC STEPCONFIG 1
MOV r2, 0x44E0D064
MOV r3, 0x00000010 //AIN0
SBBO r3, r2, 0, 4
```

```
//Init ADC STEPCONFIG 2
MOV r2, 0x44E0D06C
MOV r3, 0x00080010 //AIN1
SBBO r3, r2, 0, 4
```

```
//Init ADC STEPCONFIG 3
MOV r2, 0x44E0D074
MOV r3, 0x00100010 //AIN2
SBBO r3, r2, 0, 4
```

```
//Init ADC STEPCONFIG 4
MOV r2, 0x44E0D07C
MOV r3, 0x00180010 //AIN3
SBBO r3, r2, 0, 4
```

```
//Init ADC STEPCONFIG 5
MOV r2, 0x44E0D084
MOV r3, 0x00200010 //AIN4
SBBO r3, r2, 0, 4
```

```
//Init ADC STEPCONFIG 6
MOV r2, 0x44E0D08C
MOV r3, 0x00280010 //AIN5
SBBO r3, r2, 0, 4
```

```
//Init ADC STEPCONFIG 6
MOV r2, 0x44E0D094
MOV r3, 0x00300010 //AIN6
SBBO r3, r2, 0, 4
```

```
//Enable TSC_ADC_SS
//Set writable
MOV r2, 0x44E0D040
MOV r3, 0x00000005
SBBO r3, r2, 0, 4
```

```
MOV r5, 8 //Offset to pass in num_samples,num_ads,...
MOV r8, 0 //current count
MOV r9, 0 //total count
LBCO r6, CONST_PRUSHAREDDRAM, 0, 4 //num_ads
LBCO r7, CONST_PRUSHAREDDRAM, 4, 4 //num_samples
```

```
//Read ADC and FIFOCOUNT
READADC
```

Arquivo responsável por receber as informações da memória compartilhada PRU e armazenar no arm.

```
#define PRUSS0_SHARED_DATARAM 4
#define NUM_ADS 7 //Num of A/Ds being used

/*****
 * Functions declarations
 *****/
static unsigned int ProcessingADC1(unsigned int value);

/*****
 * Global variable Declarations
 *****/
static void *sharedMem;
static unsigned int *sharedMem_int;

/*****
 * Main
 *****/
int main (int argc, char* argv[])
{
    unsigned int ret;
    tpruss_intc_initdata pruss_intc_initdata = PRUSS_INTC_INITDATA;
    int i = 0;
    int num_samples = 0;

    if(argc != 2){
        printf("\tERROR: Sampling period is required by second\n");
        printf("\t    %s [num samples]\n", argv[0]);
        return 0;
    }
    num_samples = atoi(argv[1]);

    /* Initializing PRU */
    prussdrv_init();
    ret = prussdrv_open(PRU_EVTOUT_0);
    if (ret){
```

---

```

    printf("\tERROR: prussdrv_open open failed\n");
    return (ret);
}

prussdrv_pruinit(&pruss_intc_initdata);
printf("\tINFO: Initializing.\r\n");
prussdrv_map_prumem(PRUSS0_SHARED_DATARAM, &sharedMem);

sharedMem_int = (unsigned int*) sharedMem;
sharedMem_int[OFFSET_SHAREDDRAM + 0] = NUM_ADS;
sharedMem_int[OFFSET_SHAREDDRAM + 1] = num_samples;
sharedMem_int[OFFSET_SHAREDDRAM + 2] = 0;

/* Executing PRU. */
//prussdrv_exec_program (PRU_NUM, "ADCCollector_bin.h");
prussdrv_pru_write_memory(PRUSS0_PRU0_IRAM, 0, PRUcode,
sizeof(PRUcode));
prussdrv_pru_enable(0);

printf("\tINFO: Sampling is started for %d samples\n", num_samples);
printf("\tINFO: Collecting");

/* Read ADC */

printf("\n\n AIN0 AIN1 AIN2 AIN3 AIN4 AIN5 AIN6 \n");
printf(" ---- ---- ---- ---- ---- ---- ----\n\n");
for(i = 0; i < num_samples*NUM_ADS; i+=NUM_ADS){

    if( (i/NUM_ADS) >= num_samples) break;

    while(!sharedMem_int[OFFSET_SHAREDDRAM + 2]){
        sleep(1);
    }

    printf(" %4d", ProcessingADC1(sharedMem_int[OFFSET_SHAREDDRAM + 3]));
    printf(" %4d", ProcessingADC1(sharedMem_int[OFFSET_SHAREDDRAM + 4]));
    printf(" %4d", ProcessingADC1(sharedMem_int[OFFSET_SHAREDDRAM + 5]));
    printf(" %4d", ProcessingADC1(sharedMem_int[OFFSET_SHAREDDRAM + 6]));
    printf(" %4d", ProcessingADC1(sharedMem_int[OFFSET_SHAREDDRAM + 7]));
    printf(" %4d", ProcessingADC1(sharedMem_int[OFFSET_SHAREDDRAM + 8]));
    printf(" %4d", ProcessingADC1(sharedMem_int[OFFSET_SHAREDDRAM + 9]));
    printf("\n");

    sharedMem_int[OFFSET_SHAREDDRAM+2] = 0;

}

    printf("\n\n\tINFO: Sampling completed ...\n");
printf("\tINFO: PRU completed transfer.\r\n");

```

---

```
prussdrv_pru_clear_event (PRU_EVTOUT_0, PRU0_ARM_INTERRUPT);

/* Disable PRU*/
prussdrv_pru_disable(PRU_NUM);
prussdrv_exit();

return(0);
}

static unsigned int ProcessingADC1(unsigned int value){
//Clear upper 20 bits
return value & 0x00000FFF;
```