



**UNIVERSIDADE FEDERAL DA BAHIA  
ESCOLA POLITÉCNICA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



LUCAS LIMA RODRIGUES

**USO DO BEAGLEBONE BLACK COMO UM SISTEMA DE  
AQUISIÇÃO DE DADOS**

SALVADOR  
2016



**UNIVERSIDADE FEDERAL DA BAHIA**  
**ESCOLA POLITÉCNICA**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

LUCAS LIMA RODRIGUES

**USO DO BEAGLEBONE BLACK COMO UM SISTEMA DE  
AQUISIÇÃO DE DADOS**

Trabalho apresentado ao Curso de Graduação em  
Engenharia Elétrica da Universidade Federal da  
Bahia como parte dos requisitos para a obtenção  
do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Antônio Cezar de Castro Lima

SALVADOR  
2016



## **TERMO DE APROVAÇÃO**

LUCAS LIMA RODRIGUES

# **USO DO BEAGLEBONE BLACK COMO UM SISTEMA DE AQUISIÇÃO DE DADOS**

Este Trabalho de Graduação foi julgado adequado para a obtenção do grau de Engenheiro Eletricista e aprovado em sua forma final pela Comissão Examinadora e pelo Colegiado do Curso de Graduação em Engenharia Elétrica da Universidade Federal da Bahia.

---

Prof. Dr. Antônio Cezar de Castro Lima  
Orientador

---

Prof. Dra. Ana Isabela Araújo Cunha  
Examinador

---

Prof. Me. Leroy Umasi Ramos  
Examinador

SALVADOR  
2016



*Este trabalho é dedicado a minha família, minha vó minha tia e todos os amigos e professores que acreditaram em mim.*





*"O único lugar aonde o sucesso vem antes do trabalho é no dicionário."  
(Albert Einstein)*



# Agradecimentos

Aos meus pais e meu irmão, aos meus tios, João e Simone, e à Aline, minha “prima-irmã” e afilhada. Enfim, a toda minha primeira e grande família: ensinamentos e valores que transcendem em muito a importância de títulos acadêmicos.

À minha segunda família: mãe Zélia, irmãos Rodrigo, Juliana, Bruno, Lara, Ariel, Karen e tantos outros, que foram se agregando, compartilhando e tornando a jornada mais fácil. Ao engrandecimento que me trouxeram.

À Petúnia, pelo amor, incentivo e uma carinhosa paciência, difíceis de traduzir em palavras. Também aos seus pais, Wilson e Derci, pelo acolhimento e exemplo.

Aos amigos da UFBA, melhor legado que levo desse curso: Simon, Jivago, Adinaílson, Ana Rita, Túlio, Fred, Roberto e Tássio. Poderia discorrer muito sobre a contribuição que cada um deu nestes longos anos de graduação, mas a beleza está na cumplicidade da amizade.

Aos professores - concordando ou não com suas metodologias, todo aprendizado adquirido foi válido.

Aos orientadores Edson e Ana Isabela, meu mais profundo apreço pelo auxílio, desde 2011, na iniciação científica, até agora, no presente trabalho.



# Resumo

Este é o resumo em Português

**Palavras-chaves:** sistema de aquisição de dados, linux embarcado, beaglebone black, interface USB.



# Abstract

English abstract

**Key-words:** data acquisition systems, embedded linux, beaglebone black, USB interface.





# Sumário

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>1</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>7</b>
<b>2.1</b>	<b>Componentes do Beaglebone Black . . . . .</b>	<b>7</b>
<b>2.2</b>	<b>Beaglebone Black e o Linux Embarcado . . . . .</b>	<b>7</b>
2.2.1	Instalando uma nova distribuição no BBB . . . . .	9
2.2.2	Comunicando com o BeagleBone Black . . . . .	11
2.2.3	Terminal no BeagleBone Black . . . . .	14
	<b>REFERÊNCIAS . . . . .</b>	<b>15</b>



# Lista de ilustrações

Figura 1 – Placas PCI e PCI-Express da National Instruments. (NATIONAL INSTRUMENTS, 2016) .	1
Figura 2 – Exemplo de anúncio de venda casada do Kit DI-149. (DATAQ INSTRUMENTS INC., ) . .	2
Figura 3 – Esquemático de um DAQ conectado a um computador (Próprio Autor) . . . . .	3
Figura 4 – O Beaglebone Black e seus componentes.(MOLLOY, 2014) . . . . .	8
Figura 5 – Diagrama do Satira AM3358A.(COLEY, 2013) . . . . .	8
Figura 6 – Beaglebone controlando uma CNC e rodando uma aplicação para CNC simultaneamente.(PRACTICAL MICRO, 2014) . . . . .	9
Figura 7 – Gravando a imagem do Debian 7.5 de 14/05/2014 no cartão SD (Próprio Autor) . . . . .	10
Figura 8 – Verificando a atualização do Debian 7.5 foi instalada com sucesso. (Próprio Autor) . . . .	11
Figura 9 – Comparação entre o modelo OSI e a pilha de protocolos da <i>internet</i> (Próprio Autor). . . .	13
Figura 10 – Configurações para conectar o BBB com o computador através do PuTTY (Próprio Autor)	14



# Lista de tabelas

Tabela 1 – Comparação entre os <i>E-linux board open source</i> mais usados no mercado (DICOLA, 2014).	4
Tabela 2 – Relação dos componentes do BBB de acordo com a figura 4 (MOLLOY, 2014). . . . .	7



# Lista de abreviaturas e siglas

ADC	<i>Analog-to-Digital Converter</i>
ARM	<i>Advanced RISC Machine</i>
CNC	<i>Computer Numeric Control</i> (Máquina de usinagem de peças metálicas)
CPU	<i>Central Processing Unit</i>
BBB	<i>Beaglebone Black</i>
DAQ	<i>Data Acquisition Systems</i>
DC	Tensão Contínua ( <i>Direct Current</i> )
DDR	<i>Double Data Rate</i> (Tecnologia de memória RAM)
DSI	<i>Display Serial Interface</i>
DSL	<i>Digital Subscriber Line</i>
DOS	<i>Disk Operating System</i>
E-Linux	<i>Embedded Linux</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
Freq.	Frequência
FAT	<i>File Allocation Table</i> (Sistema de arquivo criado pela Microsoft para gerenciar acesso de arquivos em partições de disco)
GUI	<i>Graphical User Interface</i>
GPU	<i>Graphics Processing Unit</i>
GTK+	<i>Framework</i> multiplataforma para criação de programas com interface gráfica
HDMI	<i>High-Definition Multimedia Interface</i>
I2C	<i>Inter-Integrated Circuit</i>
I/O	<i>Input &amp; Output pin</i>
JTAG	<i>Joint Test Action Group</i> (Conector utilizado para teste de circuitos eletrônicos).
LAN	<i>Local Area Network</i>
LCD	<i>Liquid Crystal Display</i>
LDO	<i>Low-dropout regulator</i> (Regulador de tensão capaz de regular a voltagem de saída quando a tensão de entrada está próxima da tensão de saída)
MIPS	<i>Microprocessor without Interlocked Pipeline Stage</i>

NI	<i>National Instruments</i>
OSI	<i>Open Systems Interconnection</i>
PC	<i>Personal Computer</i>
PCI	Padrão de barramento interno para computadores
PCI-E	<i>PCI Express</i>
PDA	<i>Personal Digital Assistant</i>
PRU	Programmable Real-Time Unit
PWM	Pulse Width Modulation
Qt	Outra <i>framework</i> multiplataforma para criação de programas com interface gráfica
RAM	<i>Random Access Memory</i>
RCA	Conector criado pela <i>Radio Corporation of America</i>
rev.	<i>revision</i>
RJ45	Conector utilizado no padrão Ethernet
RS232	Padrão de comunicação serial
RS485	Outro padrão de comunicação serial
SO	Sistema operacional
SOC	<i>System-on-chip</i>
SPI	<i>Serial Peripheral Interface</i>
SSH	<i>Secure Shell</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>
Wi-Fi	Padrão de comunicação sem fio IEEE 802.11



# 1 Introdução

Um sistema de aquisição de dados, ou DAQ do inglês *Data Acquisition Systems*, é um dispositivo, ou um conjunto deles, capaz de coletar, armazenar e distribuir uma determinada informação de tal forma que esta, posteriormente, possa ser manipulada ou utilizada para entender melhor o fenômeno. Na prática esses sistemas são utilizados para capturar dados de uma determinada variável física de um processo, geralmente vinda de um sensor.

Inicialmente os sistemas de aquisição de dados eram dispositivos eletromecânicos que mostravam uma determinada grandeza física em um visor analógico e registravam as mesmas em papéis ou fita magnéticas, estes primeiros aparelhos que registrava dados de maneira independente ficaram conhecido como Data Loggers. Atualmente o uso de Data Loggers ainda é bastante comum, por serem sistemas simples e robustos, entretanto, com o avanço da eletrônica e com o advento dos computadores surgiram novas classes de aparelhos que capturam dados, dentre eles os que podem ser conectados a PCs e trabalham em conjunto com software como MATLAB e LabVIEW, esta classe de dispositivo é conhecida como *PC-based data acquisition equipment*, ou em português, equipamentos de aquisição de dados para computadores (MEASUREMENT COMPUTING CORPORATION, 2012).

Os primeiros DAQs que se comunicava com computadores surgiram na década de 60 inicialmente utilizado pelas grandes indústrias e centros de pesquisa, entretanto, a partir da década de 90, com a popularização dos computadores pessoais surgiram os primeiros DAQs na forma de placas de extensão para computadores utilizando os slots de extensão PCI, as chamadas *plugin-in cards*. Hoje em dia os *slots* PCs evoluíram para o padrão *PCI-express*, um barramento com elevadíssima taxa de transmissão de dados, podendo chegar a até 32GB/s para o padrão *PCI-express 4.0*, ou PCI-E (PEREIRA, 2012), as placas PCI ainda existem, mas estão entrando em desuso pois todas as placas-mãe atuais vêm de fábrica com o novo padrão. Na figura 1 tem-se dois modelos de DAQs na forma de placa de extensão de computadores com o padrão PCI (Esquerda) e o padrão PCI-Express (Direita).

Os modelos de DAQs na forma de extensão de placas para PCs foram bastante comuns na década de 90, porém esta classe de dispositivos podem sofrer interferência eletromagnética e eletrostática devido as máquinas rotativas dos computadores, como os *coolers* e a própria estrutura de barramentos do computador, necessitando de um sistema de proteção contra este tipo de interferência. Atualmente os fabricantes vem disponibilizando sistema de aquisição de dados em uma caixa separada do computador garantindo um isolamento

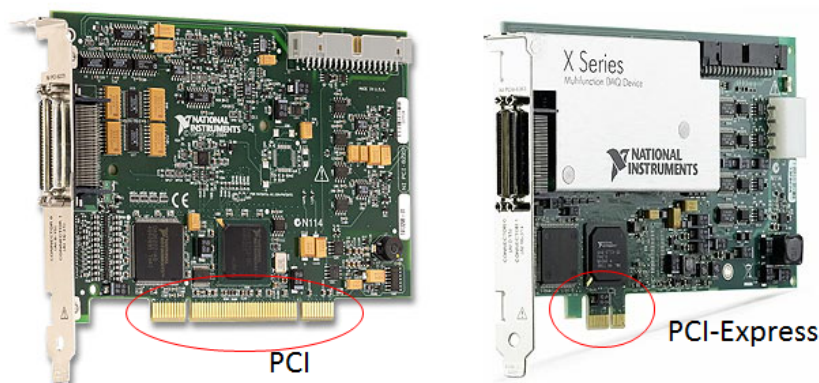


Figura 1 – Placas PCI e PCI-Express da National Instruments. (NATIONAL INSTRUMENTS, 2016)

## DI-149 Bundle

Bundle the DI-149 and save \$109

The DI-149 Bundle includes the following products:

- ✓ DI-149 USB data acquisition starter kit (a \$59.00 value)
- ✓ WinDaq/XL software — A Real time link to Microsoft Excel (2003 version or better) requiring no programming (a \$99.00 value)
- ✓ An Unlock Code for high speed data acquisition with WinDaq software (a \$195.00 value)

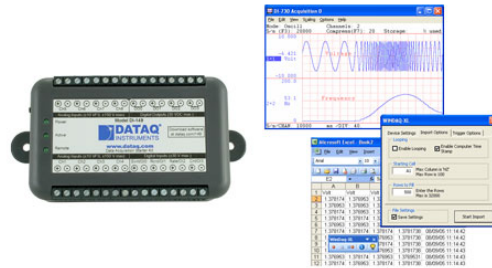


Figura 2 – Exemplo de anúncio de venda casada do Kit DI-149. (DATAQ INSTRUMENTS INC., )

melhor em relação as interferências do computador. O fato de está externo ao computador, permite, também, uma maior mobilidade e menos restrição de espaço podendo ser utilizado em conjunto com *notebooks*, não se limitando apenas a uma máquina, e ainda, em alguns casos, permitindo a comunicação entre DAQs a metros ou quilômetros de distância e o computador.

Estes dispositivos isolados do computador só foram possíveis porque a comunicação entre o PC e seus periféricos externos evoluiu bastante. Os primeiros dispositivos isolados, ou *stand-alone*, utilizavam principalmente a comunicação serial, RS232 e suas variantes como o RS485, para o caso do meio industrial. Estes padrões ainda são bastante comum no meio industrial e entre dispositivos embarcados, porém, não permite altas taxas de transferência, daí a preferência por placas PCI no passado. Com o surgimento de protocolos de comunicação mais recentes, como a USB, à partir da versão 2.0 e a *Ethernet*, incluindo, mais recentemente, as redes sem fio, os DAQs *stand-alone* com vários canais de alta velocidade, no qual necessita de um tráfego intenso de dados, se tornaram viáveis.

Historicamente os sistemas de aquisição de dados foram criados para atender a necessidade de grandes indústrias e centros de pesquisa. Estes tipos de equipamentos, por sua vez, nunca foram baratos, devido a necessidade de equipamentos de alta qualidade, principalmente, devido ao grande poder de compra dos clientes. Por isso, os maiores fabricantes dessa classe de dispositivos possuem soluções com preços pouco convidativos, e quase sempre atrelados a *software* e *hardware* proprietário. Para exemplificar, a *National Instruments (NI)*, uma grande fabricante no ramo, vendem DAQs que variam de R\$ 2.087,88 a R\$ 63,540, segundo a loja oficial no Brasil (NATIONAL INSTRUMENTS, 2016). O uso de *software* e *hardware* proprietários torna comum a prática de venda casada, pois para o funcionamento completo do DAQ é necessário adquirir mais de um produto, um anúncio da DAQ Instruments de um pacote incluindo o DAQ e os softwares necessários para o seu funcionamento exemplifica esta prática. Sozinho o aparelho custa \$59,00, porém, para ter a melhor experiência o fabricante sugere a compra de um pacote de \$244,00 (Figura 2).

Na topologia padrão de um DAQ conectado a um computador é necessário converter o sinal analógico, geralmente vindo de um sensor, para dados digitais, de tal forma que possa ser reconhecido pelo computador. Esses dados posteriormente serão enviados através de uma interface de comunicação entre o computador e o DAQ. Para que tudo isso seja possível existe uma CPU capaz de gerenciar os dados de entrada conversor analógico digital, ou ADC (*Analog-to-digital converter*), e a interface de comunicação. Um esquemático dessa topologia pode ser visto na figura 3.

Na verdade, a topologia do DAQ representado na figura 3 é algo bastante comum na eletrônica embarcada, por isso, já vem embutida em alguns microcontroladores. Sistemas embarcados, são dispositivos feitos para um propósito específico, desde ao controle de um elevador ao gerenciamento de um instrumento de aquisição de dados. Nestes sistemas é comum o uso de microcontroladores, que são computadores dedicados àquela tarefa exclusiva. Muitos microcontroladores, além da CPU e memória, vem com diversos protocolos

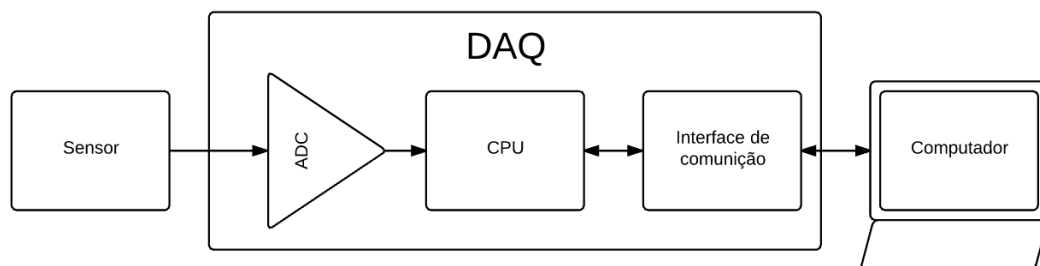


Figura 3 – Esquemático de um DAQ conectado a um computador (Próprio Autor)

de comunicação, ADCs, DACs (Digital Analog Converters) embutidos em um único encapsulamento. Essa integração de diversos periféricos em um único *chip* deixa o projeto mais simples e barato, por isso é cada vez mais comum os fabricantes disponibilizarem sistemas cada vez mais completos em um único encapsulamento. Esse conceito é conhecido como *System on Chip*, os famosos SOCs, que estão embarcados nos mais diversos dispositivos, até mesmo em celulares.

Os primeiros microcontroladores surgiram no final do século passado. Inicialmente eles não eram muito acessíveis, tanto na questão de preço quanto na complexidade de programação e desenvolvimento. Muito deles tinham que ser programados em *Assembly*, ou linguagem de máquina, por causa da baixa memória disponível. Entretanto, com o tempo os novos modelos vieram com mais memória, mais poder de processamento e periféricos mais avançado, junto a isso, como tudo na tecnologia, surgiu novas ferramentas e compiladores, facilitando bastante o desenvolvimento de dispositivos microcontrolados. Uma grande evolução nesse sentido aconteceu em 2005, quando um grupo de pesquisadores da Itália propuseram uma plataforma de prototipagem de baixo custo, fácil utilização e de código aberto, conhecida como Arduino. O *hardware* do Arduino é uma placa de desenvolvimento completa, ou seja, além do microcontrolador, ela vem com todos os circuitos externos para o seu funcionamento, incluindo fonte de alimentação, regulador de tensão, oscilador externo, entrada USB e pinos de extensão (TED CONFERENCES, LLC, 2012). Além do *hardware* pronto para o uso o Arduino dispõe uma linguagem de programação que abstrai os elementos de *hardware* do microcontrolador, portanto é possível programar no Arduino sem precisar setar nenhum valor em registradores do microcontrolador, as bibliotecas e funções disponível na linguagem já faz todas as configurações necessárias para o usuário leigo. O fato de ser código aberto e a facilidade de utilização atraiu diversos entusiastas, hobbistas e profissionais para a plataforma, se tornando uma das maiores comunidades, se não a maior, de sistemas embarcados atualmente.

De uma forma geral, o Arduino, além de aumentar o número de pessoas interessadas em eletrônica, aumentou a quantidade de projetos e velocidade com que eles eram feitos, possibilitando a criação de projetos como tênis que amarra sozinho e impressoras 3D (BLAKEBEVIN, 2010). Hoje existem diversos fabricantes que fabricam módulos de expansão para o Arduino conhecidos como Shields. Existem módulos para tudo, controle de motores, relés, conexão *ethernet* e Wi-Fi, *bluetooth*, dentre outros. Com os módulos o usuário só precisa comprar conectores e programar, não havendo a necessidade de criar circuitos complexos na *protoboard* ou fazer placas de circuito impresso.

Depois que o Arduino se tornou popular diversas organizações e empresas resolveram criar suas placas de desenvolvimento *open source*. Na época em que o Arduino foi criado surgiram os primeiros *smartphones*, junto com eles, os SOCs com processadores ARM e GPUs de baixo consumo que permitiu aliar o baixo consumo, dissipação de calor, tamanho portátil e alto desempenho em um único *chip*. Em 2008 a Texas Instruments, uma empresa privada que desenvolve diversos produtos semicondutores como microcontroladores e SOCs com arquitetura ARM para celulares, resolveu criar a Beagleboard em parceria com a Digikey e a Element14, duas empresas do ramo de varejo eletrônico nos EUA. A ideia era demonstrar o poder do SOC OMAP3530 em uma

	Arduino Yun	Beaglebone Black	Raspberry Pi
CPU	MIPS32	ARM Cortex-A8	ARM1176
CPU Freq.	400Mhz	1Ghz	700Mhz
Microcontrolador	ATmega32U4	PRU	Não tem
$\mu$ controlador Freq.	16Mhz	200Mhz	-
RAM	64Mb	512Mb	512Mb
GPU	Não tem	PowerVR SGX530	Broadcom VideoCore IV
Memória interna	16Mb	4GB	Não tem
Memória externa	Micro SD	Micro SD	SD
I/O	20	65	17
Ethernet	10/100 Mbit	10/100 Mbit	10/100 Mbit
ADC	12x 10bits, 0-5V	7x 12bits 0-1.8V	Não tem
PWM	7x	8x	1x
UART	2x	4x	1x
SPI	1x	2x	2x
I2C	1x	2x	1x
USB <i>Host</i>	1x	1x	2x
USB <i>Client</i>	1x	1x	Não tem
Video	Não tem	Micro HDMI	HDMI, RCA, DSI
Audio	Não tem	Micro HDMI	HDMI, P2
Preço	\$75	\$55	\$35

Tabela 1 – Comparação entre os *E-linux board open source* mais usados no mercado (DICOLA, 2014).

placa de desenvolvimento do tamanho de cartão de crédito, capaz de rodar uma distribuição Linux portada para ARM incluindo os aplicativos compilado este sistema operacional. A ideia foi promissora, pois foi uma das pioneiras na área de computadores embarcados, mas o preço de \$125 não fez a placa decolar. O Linux embarcado, ou E-Linux, só se tornou popular quando a fundação Raspberry Pi resolveu criar um produto de mesmo nome por apenas 35 dólares, em fevereiro de 2012. A ideia da fundação era criar uma plataforma de programação de baixíssimo custo (MACHADO, 2012). Hoje Raspeberry Pi é de longe a *E-Linux board* mais vendida e com a maior comunidade. *E-Linux boards* é o termo utilizado para computadores que rodam linux embarcado.

Junto ao sucesso do Raspberry Pi, surgiram diversas outras opções de empresas e fundações que queria por sua plataforma *open source* com Linux embarcado no mercado, ou atualizá-las como foi o caso da empresa criadora do Beagleboard. Assim aproveitando a onda dos computadores *single board* a Texas Instruments criou o BeagleBone Black. O grande diferencial deste computador em relação ao Raspberry Pi à Beagleboard está nos periféricos. O BBB (Abreviação de Beaglebone Black) tem 65 pinos de extensão, um microcontrolador auxiliar para programação em tempo real, o PRU, e muito mais periféricos que o Raspberry Pi. Para efeito de comparação na tabela 1 tem-se o comparativo entre o Beaglebone Black, Arduino Yun, uma das versões do arduino com Linux embarcado, e o Raspeberry Pi. Pela tabela percebe-se que o BBB é o único concorrente que alia o alto poder de processamento com a grande quantidade de periféricos e I/O, termo inglês para representar os pinos de extensão de entrada, *input*, e saída, *output*. A quantidade elevada de I/Os permitiu a fabricante do Beaglebone criar o conceito de placas de expansão para a *E-Linux board* conhecida como Capes, semelhante aos Shields disponíveis para Arduino.

O Beaglebone Black quando foi lançado fez um relativo sucesso, mais que o seu antecessor, principalmente pela nova política de preço, seu microcontrolador integrado e a quantidade generosa de I/Os e periféricos. Entretanto isso não foi o suficiente para desbancar o Raspberry Pi. Parte disso é devido a forma de como as *E-Linux boards* são utilizadas. A maioria das aplicações desses *single board computers* está na área de computação, como na criação de pequenos servidores, centrais de emulação, *media centers*, câmeras de vigilância, dentre outros. Para isso, as poucas portas e periféricos do Raspberry Pi é o suficiente. Quando há

a necessidade de alguma aplicação em tempo real geralmente utiliza-se o Raspberry Pi em conjunto com o Arduino. O outro motivo para o Beaglebone não ter desbancado o Raspberry Pi, foi o fato de este último ter sido lançado muito tarde. Em 2013, quando foi lançado, a comunidade do Pi já estava grande e já existiam outros concorrentes, como a Cubie Board. Além disso, as Capes que poderiam ser um grande diferencial são muito caras e não há muita variedade. Na prática o BBB é uma dos *single board computers* mais utilizados, mas está longe de ser a mais popular e o conceito de Capes não engrenou como a fabricante previa.

Embora Beaglebone Black tenha todos esses contras, ele ainda é a melhor opção de *E-Linux computer* para a construção de um DAQ. Pois para a criação de um DAQ é necessário um ADC de boa qualidade, uma CPU de alto desempenho para processar todos os dados capturados e interfaces de comunicação de alta velocidade, como Ethernet e USB 2.0. Por isso este trabalho pretende utilizar o Beaglebone Black para criar um sistema de aquisição de dados de baixo custo conectado ao computador pela porta USB. Sistema no qual será utilizado posteriormente para a captura de dados de um acelerômetro analógico industrial, em trabalhos futuros.



## 2 Fundamentação Teórica

### 2.1 Componentes do Beaglebone Black

No capítulo 1 foi introduzido o Beaglebone Black como um mini computador de arquitetura ARM. Esta seção irá mostrar os componentes da placa e os integrados ao SOC Sitara AM335x. Na figura 4 tem-se uma foto de frente e verso da placa de circuito impresso do BBB, identificando os componentes, e na tabela 2 identifica a função de cada componente na placa. Para complementar na figura 5 mostra os componentes integrados ao SOC Sitara AM3358A, este último disponível na rev.3 deste produto.

No.	Componente	Função
1	Sitara AM335x	SOC do BBB contendo diversos componentes integrados incluindo a CPU, GPU e PRU.
2	HDMI Framer	Converte o controlador de LCD do AM335x.
3	Memória RAM	512MB DDR3.
4	eMMC	4GB de memória de armazenamento interna.
5	TPS65217C	Regulador de potência sofisticado com 4 reguladores de tensão LDO controlado por I2C
6	Ethernet PHY	Conecta o processador ARM a conexão física RJ45 com a velocidade de 100Mbit para enviar e 10Mbit para receber.
7	7x LEDs	<i>Power</i> LED (azul), 4 <i>user</i> LEDs e 2 LEDs (amarelo = dados enviados/ verde = dados recebidos).
8	<i>Push Buttons</i>	Liga/Desliga ( <i>Power</i> ), <i>Reset Button</i> e <i>Boot Switch</i> . Este último seleciona se o deve ser feito <i>boot</i> do eMMC ou do cartão micro SD.
9	Micro HDMI	Para conectar em monitores de até 1280x1024@60Hz ou 1920x1080@24Hz.
10	Ethernet RJ45	Conector RJ45.
11	5V DC	<i>Jack</i> de 5mm para usar o BBB sem precisar alimentar com o cabo USB.
12	<i>Slot</i> micro SD	<i>Slot</i> para cartão micro SD;
13	<i>Serial Debug</i>	Conector de 6 pinos para acessar o terminal através da conexão serial (UART0).
14	USB 2.0 <i>Client</i>	Conector mini-USB 2.0 utilizado para conectar o BBB ao computador.
15	USB 2.0 <i>Host</i>	Conector USB-A 2.0 para conectar os periféricos do BBB, como <i>webcams</i> , <i>mouse</i> , teclado e outros. Pode-se utilizar um <i>hub</i> USB para conectar mais de um periférico.
16&17	Expansões P8 e P9	Soquete de pinos de extensão P8 e P9. Cada soquete tem 2x23 pinos totalizando 92 pinos no total.
18	JTAG	Espaço para conector JTAG, bastante utilizado em testes de placas de circuito impresso. Porém, necessita de <i>software</i> e <i>hardware</i> adicional.
19	Conector de bateria	É possível soldar estes 4 pinos para adicionar um conector de bateria a placa.

Tabela 2 – Relação dos componentes do BBB de acordo com a figura 4 (MOLLOY, 2014).

### 2.2 Beaglebone Black e o Linux Embarcado

Como foi dito nos capítulos anteriores, o Beaglebone Black é um minicomputador completo que é capaz de rodar diversas distribuições de *E-Linux*, Android e outros sistemas operacionais portados para arquitetura ARM. Atualmente a comunidade do Beaglebone portou apenas Android e algumas distribuições Linux, como Debian, Ubuntu, Ångström e Arch Linux. As primeiras versões do Beaglebone Black, mais especificamente as revisões 1 e 2, vinham com Ångström instalado por padrão, uma distribuição linux criada exclusivamente para

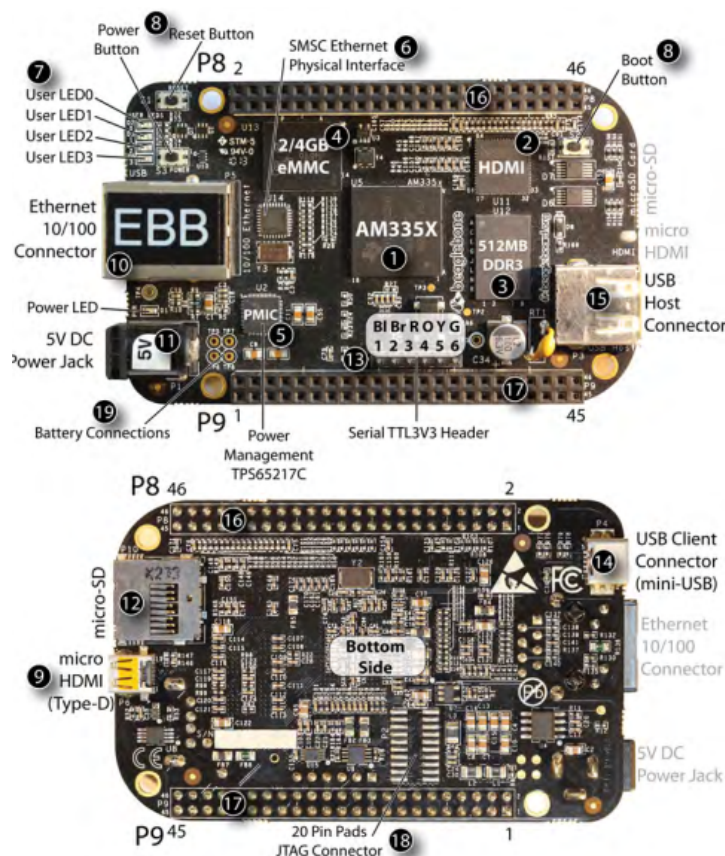


Figura 4 – O Beaglebone Black e seus componentes.(MOLLOY, 2014)

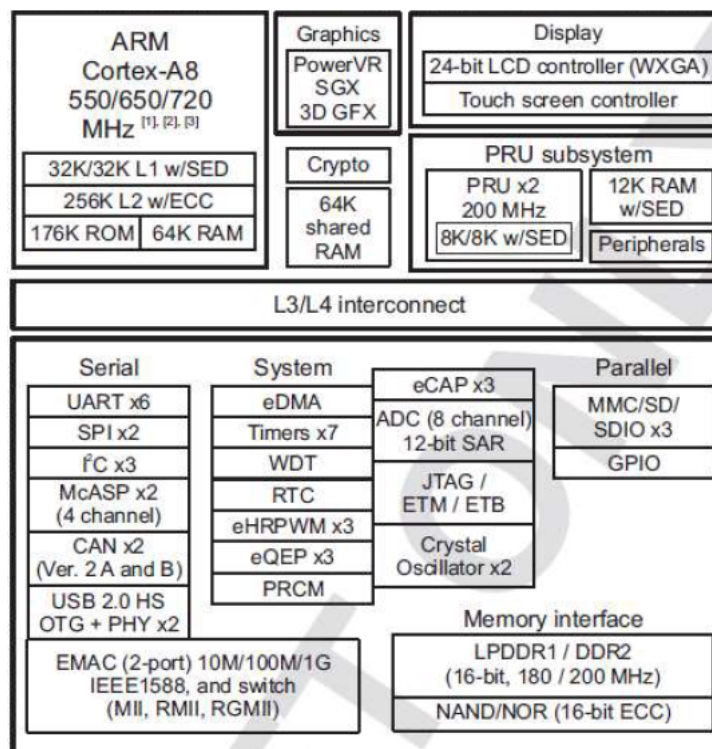


Figura 5 – Diagrama do Satira AM3358A.(COLEY, 2013)



sistemas embarcados, *tablets*, PDAs, *set top boxes*, roteadores e outros dispositivos baseados na arquitetura ARM (KOEN, 2015).

Entretanto, uma parte dos usuários preferiam instalar outra distribuição, como o Ubuntu, pois esta parcela de usuários na maioria das vezes não tinha muita convivência com o Linux e o fato de trabalhar numa distribuição que não tem suas raízes comuns as distribuições populares para *desktop* tornava o sistema menos amigável. Percebendo a popularidade de tutoriais ensinando como trocar de distribuição, a Texas passou a incluir a distribuição Debian a partir da revisão 3 do Beaglebone Black, lançada em 2014, e revisão mais recente desta placa, além disso, na revisão 3 houve um aumento na memória interna de 2GB para 4GB. Essas adições tornaram o BBB mais atraente para ser usado como computador, pois o espaço extra permitia a instalação de aplicativos para Linux como o *Open Office*, reprodutores de vídeo e editores de imagens.

A nova distribuição facilitou, também, na execução de aplicativos com interface gráfica (GUI) baseados nas *frameworks* GTK+ e Qt, que são bastante populares no Linux *desktop* baseados no Debian e seus derivados, como Ubuntu e Linux Mint. O uso de aplicativos gráficos é bastante comum nas *E-Linux Board* para servir como interface entre o usuário e a máquina. Na figura 6 mostra o exemplo de uma máquina CNC controlada por um Beaglebone e a *cape K9 CNC I/O*. Para controlar a CNC a fabricante da *cape* criou um aplicativo com interface gráfica para Linux onde o usuário é capaz de visualizar o processo e interagir com a CNC.

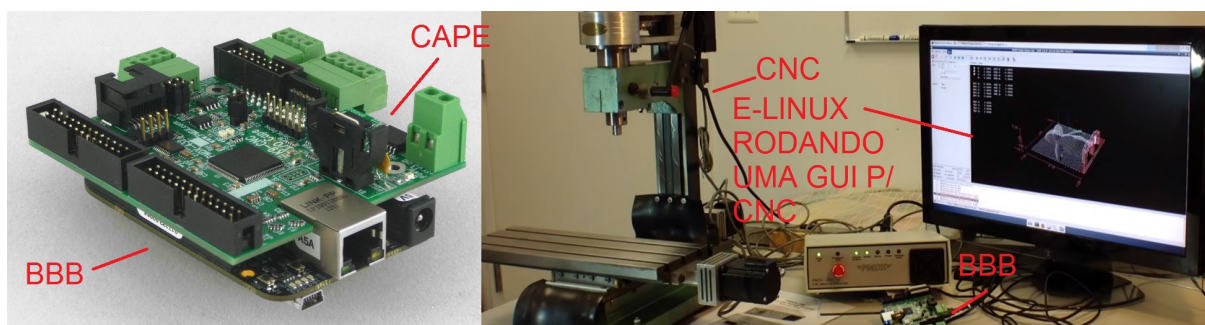


Figura 6 – Beaglebone controlando uma CNC e rodando uma aplicação para CNC simultaneamente.(PRACTICAL MICRO, 2014)

As distribuições baseadas no Debian já vem com as interfaces gráficas e pacotes relacionados instalados por padrão, incluindo aqueles necessários para executar aplicativos criados em Qt ou GTK+. Além disso, essas distribuições vem com diversos outros pacotes que são padrão para a maioria dos usuários Linux, um grande exemplo é o gerenciador de pacotes *apt* que está disponível no Debian, mas não no Ångström.

### 2.2.1 Instalando uma nova distribuição no BBB

Na seção 2.2 falo-se da importância de utilizar a distribuição Debian, portanto, se o Beaglebone Black adquirido for anterior a rev.3 deve-se instalar o Debian 7.5 Wheezy, versão de 14/05/2014. Caso o Beaglebone venha uma versão do Debian superior a esta é recomendado fazer o *downgrade* para a versão 7.5, principalmente, se for a versão 8 Jessie ou superior. A versão 7.5 foi projetada pensando, também, nas revisões anteriores a rev. 3, por isso ocupa um pouco menos de 2 Gb, sobrando aproximadamente 2 Gb dos 4Gb da eMMC do BBB, caso este tenha sido adquirido depois de 2014. Esse espaço extra será o suficiente para instalar novos programas, módulos e gerar grandes arquivos de dados. Um outro motivo fazer o *downgrade* é que neste trabalho e na maioria das bibliografias encontrada na literatura atualmente faz o uso desta versão do Debian, portanto pode haver procedimentos que não sejam os mesmos em versões diferentes, dificultando a reprodução dos experimentos feitos nesta monografia.

A primeira coisa que deve ser feita para instalar uma nova distribuição é fazer o *download* do

sistema operacional, que pode ser baixado na página que contém as últimas imagens pra Beaglebone Black (<https://beagleboard.org/latest-images>). Caso o leitor desta monografia esteja lendo-a muito tempo depois da sua publicação existe uma alternativa no domínio oficial do Debian (<https://debian.beagleboard.org/images/>), ou ainda, no site do E-Linux ([http://elinux.org/Beagleboard:BeagleBoneBlack\\_Debian](http://elinux.org/Beagleboard:BeagleBoneBlack_Debian)). No site das últimas imagens para BBB existem duas versões do Debian 7.5 de 14/05/2014. Uma delas é para utilizar o sistema operacional pelo cartão SD (*without flashing the eMMC*), enquanto a outra grava o sistema operacional na eMMC (*eMMC flasher*). É preferível que o sistema operacional seja instalado na memória interna do Beaglebone, pois, além desta ter uma taxa de leitura e escrita maior que o cartão SD, ficamos com o *slot* SD livre para ser utilizado em outras ocasiões.

O formato do arquivo baixado é *.img.xz* que é um formato de compactação bastante comum no Linux. Caso o usuário esteja usando o Windows talvez seja necessário fazer o *download* de alguma ferramenta capaz de descompactar este formato de arquivo, visto que alguns aplicativos de descompactação, como WinRar, não trabalham com este tipo de arquivo. Uma sugestão de descompactador capaz de trabalhar com este formato de compactação é o 7-Zip (Disponível em: <http://www.7-zip.org/>), gratuito, *open source* e sem propagandas.

Enquanto a imagem do sistema operacional está sendo baixada, deve-se fazer o *download* de outro programa para gravar a imagem no cartão de memória. Uma sugestão dada pela Wiki do E-Linux é o Win32 Disk Imager (<https://sourceforge.net/projects/win32diskimager/>) (EMBEDDED LINUX WIKI, 2016). Quando ambos os arquivos estiverem baixados, descompacte a imagem do Debian que está no formato *.img.xz* e irá obter um arquivo *.img*. Insira o cartão SD no leitor de cartão do seu computador. Abra o Disk Imager selecione a letra correspondente a partição do cartão de memória, selecione a imagem do Debian e clique em gravar como mostra na figura 7. Irá aparecer uma caixa de mensagem explicando que esta operação pode danificar os dados no cartão, clique em *yes* espere o processo ser finalizado.

## ② Imagem do Debian

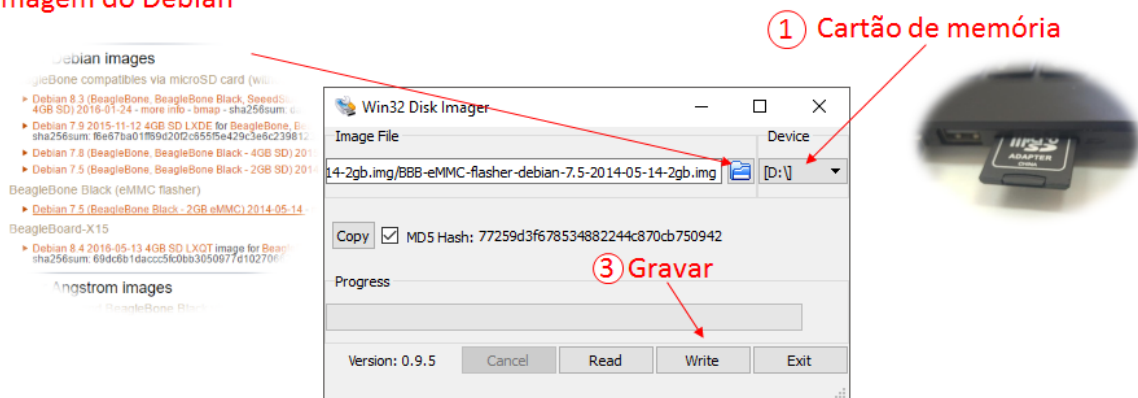


Figura 7 – Gravando a imagem do Debian 7.5 de 14/05/2014 no cartão SD (Próprio Autor)

Depois do processo de gravação finalizar com sucesso retire o cartão SD do leitor e com o BBB ainda desligado, insira o microSD no *slot* apropriado (Item 12 na figura 4). Antes de ligar a placa, mantenha pressionado o *Boot Switch* (Item 8 na figura 4), e ainda com o botão pressionado, conecte a porta USB *Client* do Beaglebone (Item 14 na figura 4) e ao computador ou alguma fonte de alimentação. Depois dos *User LEDs* (Item 7 na figura 4) começarem a piscar solte o *Boot Switch*. A partir daí, o sistema operacional será instalado na eMMC o processo dura cerca de 30 a 40 minutos, durante este tempo não desligue o Beaglebone. Quando a instalação estiver concluída os 4 *User LEDs* irão se apagar. Neste momento desconecte o cabo USB da fonte de alimentação retire o cartão SD e conecte novamente o BBB ao computador. Será montada uma partição FAT com o nome *BeagleBone Getting Started* clique nela e abra o arquivo *ID.txt* com o bloco de notas. Se aparecer *BeagleBoard.org BeagleBone Debian Image 2014-04-23* a atualização da Distro foi realizada com

sucesso (Figura 8).

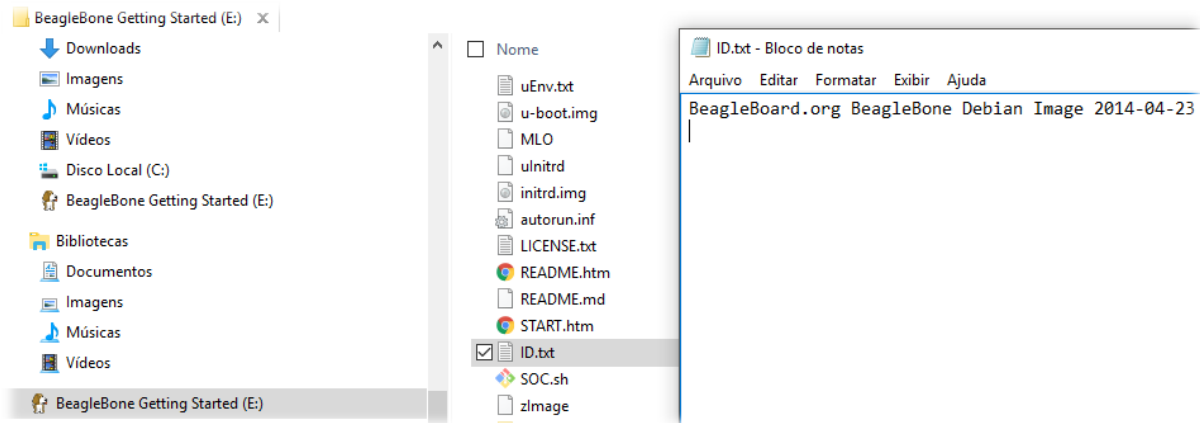


Figura 8 – Verificando a atualização do Debian 7.5 foi instalada com sucesso. (Próprio Autor)

### 2.2.2 Comunicando com o BeagleBone Black

O Beaglebone Black *vanilla*, ou seja, da forma como veio de fábrica, não tem nenhuma interface de interação com o usuário além de alguns *push-buttons* e LEDs indicadores (Itens 7 e 8 da figura 4), e isso não é o suficiente para programar ou adicionar alguma nova função a placa de desenvolvimento, a menos que o usuário use o BeagleBone ligado a um monitor, mouse e teclado. Para interagir com o *board computer* sem a necessidade desses aparelhos é necessário se comunicar com o BBB através de um computador hospedeiro. Existem diversas formas de fazer esta comunicação, mas esta seção irá focar na protocolo SSH e no *serial debug*, primeiro iremos falar do SSH.

Nos primórdios da informática não existia mouse, nem sistema operacional com interface gráfica, janelas e menus. Portanto, a interação homem-máquina era feita por comandos que executavam aplicativos e faziam operações, como entrar em diretórios, copiar arquivos, escrever documentos e até mesmo programar. Este tipo de interação homem-máquina ficou conhecido como linha de comando e era bastante popular em sistemas operacionais antigos, como DOS. Com o surgimento do Windows, a Microsoft passou a desestimular o uso da linha de comando para o uso em detrimento à interface de janelas, em conjunto com o *mouse*. Foi a partir daí que a informática começou se popularizar. A interface de janelas era mais intuitiva e, por isso, atraiu a atenção das massas.

Com o tempo as pessoas se acostumaram com o ambiente gráfico do Windows e a linha de comando ficou em abandono, não apenas pelos seus usuários, mas também pela própria Microsoft, cujo o principal objetivo era deixar o ambiente de janelas cada vez mais rico e moderno. Por motivos de legado, a empresa de Bill Gates disponibilizou o programa *cmd.exe* ou Prompt de Comando. Ele é um emulador do DOS onde, é possível executar boa parte das operações do antigo sistema operacional. Contudo, como não houve atualizações do DOS durante décadas, o Prompt de Comando é ultrapassado, pois as novas tecnologias foram implementadas apenas na interface gráfica, tornando-o bastante pobre em relação ao ambiente de janelas. Entretanto, isto só é verdade no Windows, seus grandes concorrentes, Linux e Mac, não abandonaram a linha de comando.

O Linux e Mac são baseados em um ancestral comum, o Unix. Por isso, sua base de arquivos e a forma de como são organizados são parecidas. Uma das principais semelhanças desses sistemas operacionais é o uso da mesma linguagem de linha de comando, o Shell. Esta linguagem, ao contrário do Prompt Comando, é bastante completa sendo capaz de fazer quase todas do sistema operacional, as vezes até mais operações que a própria interface gráfica. O programa que executa os comandos Shell é chamado de terminal.

Por muito tempo, o Linux só permitia a instalação de programas e *drivers* através da linha de comando, uma tarefa um tanto complicada para usuários comuns. Isso, teoricamente, afastou as pessoas comuns deste sistema operacional, chegando ao ponto que muitos atribuírem a baixa taxa de adoção do SO por causa da ausência de uma interface gráfica tão completa quanto a do Windows. Com o tempo isso mudou e hoje o Linux permite fazer quase tudo, incluindo a instalação de *drivers* e aplicativos, pela interface gráfica. Porém, é possível fazer o mesmo pela linha de comando e isso é uma vantagem enorme do Linux e Mac em relação ao Windows, o freguês pode escolher a maneira de interagir com o sistema.

Existem algumas tarefas que são muito mais rápidas e práticas serem feitas através do terminal, embora não sejam tão intuitivas quanto. Um grande exemplo disso é a automação de tarefas, com o Shell é possível criar macros ou pequenos *scripts* para automatizar tarefas chatas ou repetitivas de maneira rápida. Além disso, por utilizar apenas textos para enviar e mostrar informações, esta é uma forma de interação que consome poucos recursos da máquina. E ainda, o uso exclusivo de texto facilita o compartilhamento de artigos e tutoriais relacionados a interações com o sistema, principalmente se eles forem mais complexos, como alterar parâmetros de configurações, pois basta que o usuário abra o terminal copie os comandos do tutorial, cole e aperte *Enter*. O mesmo tutorial utilizando interface gráfica necessitaria de alguns textos explicativos e capturas de tela, por isso que a maioria dos tutoriais para Linux disponíveis na internet são feitos para a linha de comando, embora seja perfeitamente possível fazer o mesmo pela interface gráfica. O lado ruim dessa prática é que o usuário leigo muitas vezes não tem noção dos processos necessários para a execução de tal tarefa e, por isso, não aprende o porquê ou como reproduzir o mesmo sem copiar e colar a sequência de comandos. Com esta prática, muitos usuários recém-chegados no Linux pensam que está é a única forma de fazer determinada tarefa, e acabam abandonando a plataforma por pensar que o Linux não foi feito para "seres humanos".

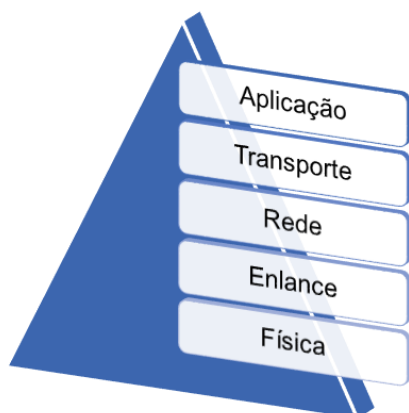
A sigla SSH significa Secure Shell, ou seja, uma forma de enviar comandos com criptografia, ou de forma segura, de uma máquina para outra. Este protocolo de comunicação foi criado, inicialmente, para facilitar o acesso remoto de máquinas dentro dos servidores (MOLLOY, 2014). Para exemplificar, imagine que um engenheiro necessita instalar um novo aplicativo em um servidor a quilômetros de distância. Com o SSH o engenheiro pode iniciar uma conexão SSH entre o seu computador e uma das máquinas do servidor e escrever os comandos para a instalação do aplicativo, tudo através do terminal da sua máquina. Claro que operar a linha de comando não é uma tarefa para qualquer um, entretanto o fato do SSH ser baseado somente em texto, torna-o uma forma de comunicação que consome pouca largura de banda e fácil de ser implementada, além de trazer todas as outras vantagens que o Shell oferece em relação ao ambiente de janelas. Isso é evidente quando se compara o esforço necessário para fazer uma comunicação remota através de interface gráfica. Para que isso seja possível é necessário, no mínimo, fazer a captura da tela, mouse e teclado do computador remoto, enquanto, com SSH, só é necessário que haja a captura do teclado e o envio de alguns dados no formato de texto.

O protocolo SSH é orientado por IP (Internet Protocol), ou seja, faz parte do mesmo padrão de comunicação utilizado pela internet, também conhecido como pilha de protocolos TCP/IP. Este modelo surgiu como um projeto de comunicação do exército americano que com o tempo projeto se expandiu e tornou-se a *internet* como conhecemos hoje.

Atualmente o TCP/IP é baseado num modelo simplificado do padrão de referência da ISO (International Organization for Standardization) para sistemas de comunicação, o OSI (Open Systems Interconnectons). O modelo da ISO é baseado em 7 camadas hierárquicas. Cada camada é responsável por executar determinada função e deve ser especificada através de protocolos de comunicação. Estes, por sua vez, podem, ou não, ser compatíveis com os protocolos de outras camadas. Quando tomado em conjunto os protocolos das diferentes camadas, denominamos pilhas de protocolos. A pilha de protocolos da internet é chamada de TCP/IP. Ela é composta obrigatoriamente de 5 das 7 camadas do padrão OSI (Figura 9). As outras duas camadas,

dependendo da aplicação, podem existir, mas são opcionais.

### Pilha de protocolos da internet



### Modelo OSI



Figura 9 – Comparação entre o modelo OSI e a pilha de protocolos da *internet* (Próprio Autor).

A camada com maior grau de abstração do TCP/IP é a aplicação. Nela contém os protocolos de nível de abstração mais elevados, responsáveis por oferecer os serviços à máquina ou usuário final. O SSH é um exemplo de protocolo desta camada que oferece o serviço de acesso remoto aos computadores. Outros serviços dessa camada são enviar arquivos (Protocolo FTP, BitTorrent), navegar na web (Protocolo HTTP) e comunicar-se através de chat (Protocolo IRC). Estes protocolos geralmente são implementados via software junto com a camada de transporte.

A camada de transporte é responsável por transportar as mensagens da camada de aplicação. Existem dois protocolos nesta camada, o TCP, que veio primeiro e também foi o responsável pelo acrônimo TCP/IP, e o UDP, que veio depois devido a necessidade de aumentar a velocidade de transmissão em aplicações em tempo real. É importante ressaltar que alguns protocolos da camada de aplicação foram feitos para ser usado sob o protocolo TCP ou protocolo UDP, ou ambos. A maioria dos protocolos de aplicação são definidos sob o protocolo TCP, como a navegação web (HTTP) e transferência de arquivos (FTP), outros são definidos apenas sob o protocolo UDP, como os protocolos de jogos online, vídeo conferência e VoIP <sup>1</sup> (voice over IP).

Voltando a falar sobre SSH, o Linux, Mac e sistemas operacionais baseados no Unix, já vem com suporte nativo a SSH, que pode ser acessado através do terminal. Entretanto, usuários do Windows devem baixar um cliente SSH como o PuTTY (Disponível em: <http://www.putty.org/>). Antes de iniciar o processo de comunicação é importante atualizar os drivers disponível em: <http://beagleboard.org/static/beaglebone/latest/README.htm>. Depois da atualização feita e o PuTTY baixado abra este aplicativo e no campo *Host Name (or IP address)* escreva o IP do BeagleBone que por padrão é 192.168.7.2. Deixe as outras opções padrões como mostra a figura 10. Por fim, clique em Open para iniciar uma conexão SSH entre o computador e o Beaglebone Black. Uma janela de terminal se abrirá, inicialmente, pedindo para que o usuário entre com um *login* e senha. Mais detalhes de como operar o terminal do Beaglebone Black pode ser visto na seção 2.2.3.

Existe a possibilidade de criar configurações predefinidas neste aplicativo. Para isso, preencha os campos e selecione as opções de acordo com o desejado. Depois, no campo *Saved Sections* escreva o nome desta configuração e clique em Save. Caso o usuário, por algum motivo, feche o PuTTY e abra-o novamente os campos do aplicativo serão restaurados para os seus valores padrões. Se houver alguma configuração pre-definida é possível carregá-la rapidamente apertando o botão *Load*.

<sup>1</sup> Tecnologia de comunicação que utiliza o Internet Protocol para transferir chamadas telefônicas e serviços de voz de uma maneira geral.

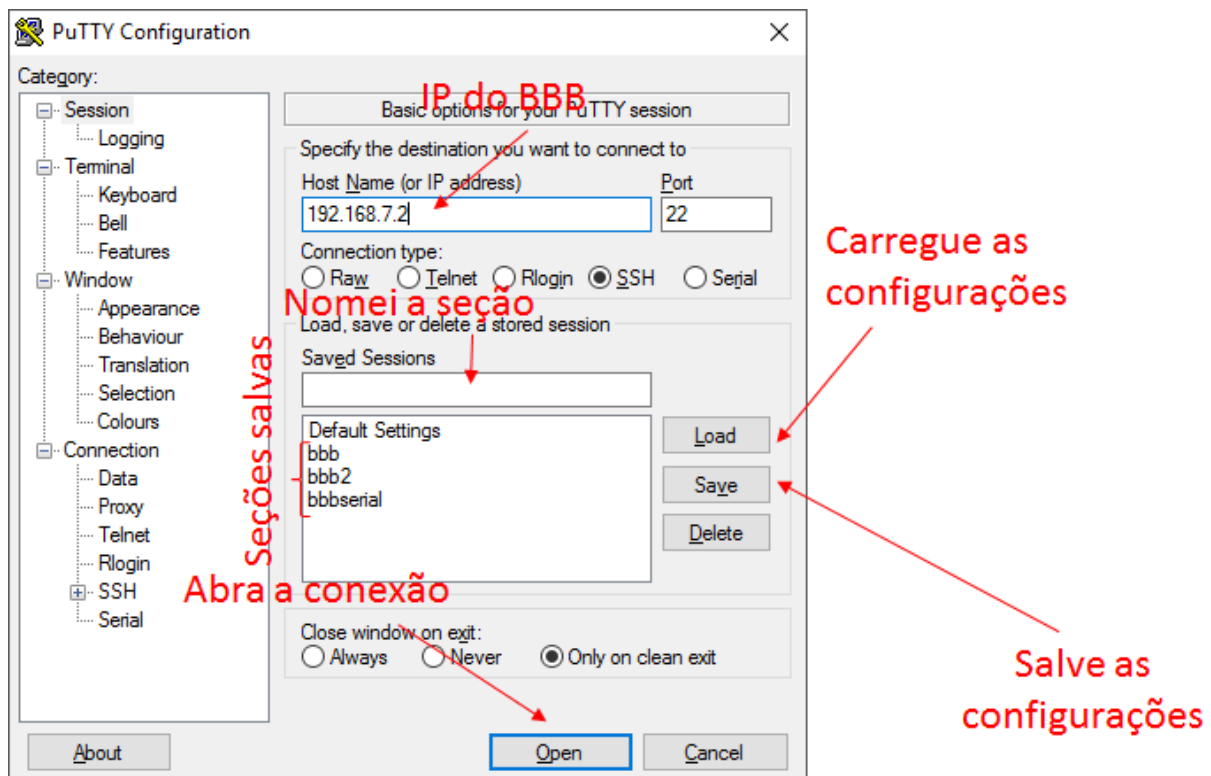


Figura 10 – Configurações para conectar o BBB com o computador através do PuTTY (Próprio Autor)

### 2.2.3 Terminal no BeagleBone Black

Na seção

## Referências

- BLAKEBEVIN. *TPower Laces- the Auto lacing shoe*. 2010. Disponível em: <<http://www.instructables.com/id/Power-Laces-the-Auto-lacing-shoe/>>.
- COLEY, G. Beaglebone black system reference manual. In: TEXAS INSTRUMENTS. A5.2. Dallas, 2013. Disponível em: <[https://cdn-shop.adafruit.com/datasheets/BBB\\_SRM.pdf](https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf)>.
- DATAQ INSTRUMENTS INC. *DI-149 USB Data Acquisition Bundle*. Disponível em: <<http://www.dataq.com/products/di-149/bundle.html>>. Acesso em: 26 de setembro de 2016.
- DICOLA, T. *Embedded Linux Board Comparison*. 2014. Disponível em: <<https://cdn-learn.adafruit.com/downloads/pdf/embedded-linux-board-comparison.pdf>>.
- EMBEDDED LINUX WIKI. *Beagleboard:Updating The Software*. 2016. Disponível em: <[http://elinux.org/Beagleboard:Updating\\_The\\_Software](http://elinux.org/Beagleboard:Updating_The_Software)>.
- FOROUZAN, B.; MOSHARRAF, F. *Redes de Computadores: Uma Abordagem Top-Down*. [S.l.]: AMGH Editora, 2013. ISBN 9788580551693.
- KOEN. *The Ångström Distribution*. 2015. Disponível em: <<http://wp.angstrom-distribution.org/introduction/>>. Acesso em: 05 de outubro de 2016.
- MACHADO, J. *Raspberry Pi: computador de 50 reais pode revolucionar a informática*. 2012. Disponível em: <<http://www.tecmundo.com.br/hardware/23175-raspberry-pi-como-um-computador-de-50-reais-pode-revolucionar-a-informatica.htm>>.
- MEASUREMENT COMPUTING CORPORATION. In: \_\_\_\_\_. *Data Aquisition Handbook*. 3. ed. Norton, 2012. Disponível em: <<http://www.mccdaq.com/pdfs/anpdf/Data-Acquisition-Handbook.pdf>>. Acesso em: 26 de setembro de 2016.
- MOLLOY, D. *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*. Wiley, 2014. ISBN 1118935128. Disponível em: <<http://www.exploringbeaglebone.com/>>.
- NATIONAL INSTRUMENTS. *Aquisição de dados*. 2016. Disponível em: <<http://www.ni.com/data-acquisition/pt/>>. Acesso em: 25 de setembro de 2016.
- PEREIRA, A. L. *O que é PCI-Express?* 2012. Disponível em: <<http://www.tecmundo.com.br/hardware/1130-o-que-e-pci-express-.htm>>. Acesso em: 26 de setembro de 2016.
- PRACTICAL MICRO. *K9 CNC I/O Cape for the BeagleBone Black*. Lorton, 2014. Disponível em: <<http://www.pmdx.com/PMDX-432>>.
- STACK OVERFLOW. *The best way to do TCP/IP over USB on Linux*. 2013. Disponível em: <<http://superuser.com/questions/593757/the-best-way-to-do-tcp-ip-over-usb-on-linux>>.
- TED CONFERENCES, LLC. *Massimo Banz: How Arduino is open-sourcing imagination*. 2012. Disponível em: <[https://www.ted.com/talks/massimo\\_banzi\\_how\\_arduino\\_is\\_open\\_sourcing\\_imagination](https://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination)>.