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-bottons* e LEDs indicadores (Itens 7 e 8 da figura 8), 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 no 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 básicas, 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 o 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 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, como 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 conhecido como 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 \cite{derekbbb}. 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 \emph{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. E ela é composta obrigatoriamente de 5 das 7 camadas do padrão OSI (Figura 1). As outras duas camadas, dependendo da aplicação, podem existir, mas são opcionais.

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[[1]](#footnote-1) (*voice over* IP).

A camada de rede é responsável por movimentar os pacotes de dados de uma máquina para outra. Esta camada tem dois componentes principais, o primeiro deles é o protocolo IP, que define o famoso conjunto de 4 números separados por pontos, os endereços de IP. O protocolo de IP é único e é ele quem caracteriza os sistemas que utilizam a pilha de dados TCP/IP. O outro componente da camada de rede são os protocolos de roteamento. Estes protocolos só fazem sentido em redes maiores, como as redes locais ou a rede mundial de computadores. Como, na nossa aplicação, estamos interessados apenas na comunicação ponto a ponto entre o computador e o BBB, utilizando a USB, esta segunda parte da camada de rede é irrelevante. Neste caso, o endereço de IP padrão do BeagleBone é . O componente roteamento só se torna relevante se o BBB for conectado à rede local pela porta Ethernet (Item 10 da figura X), através de um roteador.

A camada de enlance de dados, que, por sua vez, significa, ligação ou *link* de dados, define as especificações de transmissão, recepção, controle de fluxo, opcionalmente, a correção de erros que venham a ocorrer na camada física. As especificações desta camada estão intimamente ligadas ao meio em que se propagam. A família de protocolo DSL, por exemplo, é muito comum nas bandas largas comerciais, pois foi projetado para utilizar as linhas telefônicas como meio de transmissão. Já o protocolo Wi-Fi foi projetado para ser utilizado em redes locais cujo o meio de transmissão é o ar, assim é tem suporte a senhas de acesso, correção de erro e tem um limite de usuários simultâneos menor do que protocolos para redes sem fio maiores, como o LTE, padrão utilizado na *internet* móvel dos celulares. Nas redes locais com fio é comum utilizar o protocolo Ethernet na camada de enlace.

A camada física é o próprio meio de transmissão, ou seja, o ar, cabo coaxial, linhas telefônicas. Como dito anteriormente, é comum que os padrões da camada de enlance seja implementado em conjunto com a camada física, por isso, os circuitos integrados de rede geralmente já implementam as três camadas mais básicas (Rede, enlance e físico) via *hardware*. Por exemplo, as placas de rede dos computadores pessoais especificam que deve ser utilizado como camada física cabos UTP com conectores RJ45 em conjunto com o protocolo Ethernet na camada de enlance e o protocolo IP na camada de rede. O CI SMSC do BeagleBone (Item 6 da figura X) segue esta mesma especificação e é utilizado em conjunto com o conector Ethernet 10/100 (Item 10 da figura X). Neste caso o BeagleBone deve ser conectado a um roteador como mostra na figura Y. Como não estamos utilizando a porta USB, o endereço de IP da placa de desenvolvimento será diferente de .

Um outro cenário possível é a utilização do protocolo Ethernet sobre a porta USB (Ethernet over USB). A porta USB é bastante utilizada para conectar periféricos a computadores e muitas vezes estes periféricos deve utilizar protocolos definidos para funcionar exclusivamente sobre IP, por isso, é comum que impressoras, celulares, *smartphones* e dispositivos embarcados que utilizam a porta USB para se comunicar com o computador, implementem as camadas de enlance e rede via *software* para que seja possível utilizar os serviços disponibilizados por IP.

No caso de dispositivos embarcados com Linux isso é quase uma obrigação, visto que o próprio Linux já vem com driver USB-eth, que implementa os protocolos Ethernet e IP sobre a interface física da USB, criando uma rede local entre cliente e hospedeiro, no nosso caso computador e BeagleBone. A partir daí quase todos os serviços e protocolos disponibilizados por IP estão possíveis, incluindo aí a comunicação remota via SSH. No Windows é importante que os *drivers* disponibilizados em (<http://beagleboard.org/getting-started>) estejam atualizados, pois ambos os lados (Computador e Beaglebone Black) devem estar preparados para este tipo de conexão.

Pelo fato do protocolo Ethernet não ser um padrão definido na comunicação USB, e ainda ser implementado via software, é normal que o desempenho teórico fique abaixo dos teórico da USB 2.0. Segundo o usuário *jons34yp* do site de perguntas e respostas, superuser.com \cite{usbethernetspeed}, foi possível atingir a média de em uma comunicação entre um *smartphone* com Android e um computador. Para este trabalho taxas de transferências próximas a são mais do que o suficiente.

Os protocolos da *internet* são muito susceptíveis a ataques de maliciosos, geralmente por vírus. Para evitar possíveis ataques os *firewalls* bloqueiam a maior parte das portas de conexões TCP e UDP, podendo tornar a comunicação SSH indisponível em alguns casos. Em outros casos pode ser que os *drivers* não funcione corretamente e a conexão por IP entre o BBB e computador não seja estabelecida. Para estas situações é possível utilizar a comunicação serial, ou UART. O acrônimo UART significa Universal asynchronous receiver/transmitter, é um tipo de conexão ponto a ponto assíncrona, ou seja, que não precisa do clock para sincronizar os dados. Para que a conexão seja estabelecida são necessários pelo menos dois fios o RX responsável por receber os dados e o TX responsável por enviar os dados. Fios adicionais podem ser úteis, como um fio para igualar o referencial de terra (GND) entre os dois dispositivos de comunicação (FIGURA CONEXAO SERIAL).

O protocolo UART tem pacotes pequenos que podem ser configurados de acordo com a aplicação (Figura PACOTES SERIAL). Para que a conexão funcione ambos os dispositivos devem ser configurados com os mesmos parâmetros. Estes parâmetros inclui as especificações dos pacotes e a quantidade de pacotes por segundo (*baud rate*) e estão listados na tabela X.

|  |  |  |  |
| --- | --- | --- | --- |
| Parâmetro | Valores permitidos | UART0 | Função |
| Data bits | 5 a 9 | 8 | Determina a quantidade de bit de dados enviado em um pacote |
| Stop bit | 1 a 2 | 1 | Quantidade de bits de para determinar o fim do pacote |
| Parity | N/O/M/S | N | Bit de paridade (Correção de erro) |
| Speed (Baud rate) | Vários | 115200 | Velocidade de transmissão dada em pacotes por segundo |
| Flow Control | None | None | O controle de fluxo não é suportado |

O BeagleBone tem 5 portas UART, sendo que a UART0 já vem pré-configurada pelo sistema operacional e tem os parâmetros definidos na tabela X. Seu uso exclusivo para comunicação serial com proposito de *debug* e está disponível pelos pinos Serial TTL3V3 Header (Item 13 na figura X). O protocolo USB especifica a conexão serial através da classe CDC. Existem cabos ou conversores que já fazem a conversão da UART para a conexão serial especificada pelo protocolo USB, estes cabos são conhecidos como USB Serial TTL. Eles têm de um lado um conector USB macho e do outro lado alguns fios coloridos que devem ser conectados diretamente aos pinos Serial TTL3V3 Header do Beaglebone Black. A coloração dos fios varia de fabricante para fabricante, por isso, antes de fazer as ligações é importante saber é a função de cada fio. Na Wiki do E-Linux, disponível em (CITE USBTTL ELINUX), têm as principais colorações utilizadas.

Verificada a função de cada fio do conversor TTL, conecte os fios aos pinos do Serial TTL3V3 Header como mostra a figura X. A ordem de conexão depende da marca do conversor, mas de uma maneira geral deve-se conectar o GND do cabo no pino 1, RX no pino 4 e o TX no pino 5. Depois de conectado clique com o botão direito no menu iniciar e selecione gerenciador de dispositivos e procure por Portas (COM e LPT) e estará listado o cabo TTL, anote o número da porta COM. Com o número em mãos abra o PuTTY selecione Serial e preencha os campos de acordo com a figura Z e clique em Open. A partir daí uma janela de terminal se abrirá e será possível controlar o BeagleBone remotamente semelhante ao protocolo SSH.

1. VoIP: Tecnologia de comunicação que utiliza o *Internet Protocol* para transferir chamadas telefônicas e serviços de voz de uma maneira geral. [↑](#footnote-ref-1)