

# Acessando e controlando o target

Possuímos algumas formas de acessar e utilizar esse sistema:

1. Via terminal UART
2. Via terminal ssh
3. Se o linux possuir interface gráfica, podemos usar um teclado e mouse.
4. Interface gráfica via ssh

## UART

O linux no HPS é geralmente configurado (no bootloader) para possuir um terminal redirecionado para uma porta UART, essa porta UART é disponível via FTDI em um dos USBs da placa (o que estiver escrito UART) e então pode ser acessado como uma UART. Para isso há uma porta USB específica (UART).

Uma vez conectada a porta USB ao PC e energizada a placa, podemos verificar em qual *device* a porta serial foi mapeada (no `host`):

```
$ demsg | tail
[95158.497894] ftdi_sio 1-1.1:1.0: device disconnected
[95161.649187] usb 1-1.1: new full-speed USB device number 22
using xhci_hcd
[95161.748948] ftdi_sio 1-1.1:1.0: FTDI USB Serial Device
converter detected
[95161.749067] usb 1-1.1: Detected FT232RL
[95161.756092] usb 1-1.1: FTDI USB Serial Device converter now
attached to ttyUSB0
```

No log, verificamos que um dispositivo **FTDI USB Serial Device** foi conectado a **ttyUSB0**. E então podemos abrir o terminal por qualquer programa de porta serial (PUTTY, ...), nesse exemplo utilizaremos o programa **screen**.

```
$ screen /dev/ttyUSB0 115200,cs8
```

Para melhorarmos a interface com o screen crie um arquivo `~/screenrc` e adicione o seguinte conteúdo a ele

```
# Enable mouse scrolling and scroll bar history scrolling
termcapinfo xterm* ti@:te@
```

⚠ talvez seja necessário instalar o screen.

Feito isso, reinicie o `target` (religando a energia) para termos acesso a todo o log de inicialização.

notem que no meu caso o nome atribuído ao USB-UART foi o `/dev/ttyUSB0`, isso pode mudar no PC de vocês

Uma vez conectado, agora temos acesso a todas as funcionalidades do linux como um bash normal. Na verdade o terminal já é configurado para funcionar já no boot, ou seja, conseguimos ter acesso as informações do u-boot.

### Kernel e porta serial

Como o kernel sabe qual porta ele deve utilizar para imprimir o log e usar como terminal ? Esse parâmetro é passado pelo u-boot para o kernel via o **Device Tree**:

```
hps_0_uart0: serial@0xffc02000 {
    compatible = "snps,dw-apb-uart-1.0", "snps,dw-apb-uart";
    reg = < 0xFFC02000 0x00001000 >;
    interrupt-parent = < &hps_0_arm_gic_0 >;
    interrupts = < 0 162 4 >;
    reg-io-width = < 4 >;
    reg-shift = < 2 >;
    clock-frequency = < 100000000 >;
};
```

E então passado com o argumento para o kernel:

```
bootargs = "console=ttyS0,115200";
```

🚦 (esse exemplo não é da nossa FPGA)