O que é ARM Streamline?

ARM Streamline é uma parte da série de ferramentas da ARM DS-5. A ARM Streamline permitiu que desenvolvedores de software fizessem pleno uso dos recursos disponíveis ajudando-os a criarem produtos que exigem um alto desempenho e eficiência.

Ele tem uma interface gráfica que exibem informações da CPU e os contadores de desempenho do GPU para os códigos fontes hotspots e exibem o consumo real de energia. Desta forma, os desenvolvedores melhoram o paralelismo do código, estendem a vida útil da bateria e melhoram a experiência do usuário.

Ele oferece um poderoso ambiente de análise de sistema funcional utilizado para a otimização de software.

Por que usar uma Streamline?

Melhora o Rate do Código

- **Encontrando a posição do CPU que consume tempo**
- **Melhorando a plataforma de código multi-core**
- **Ajustando o código para alcançar o uso ideal do cache e vetor.**

Reduz o consumo de energia

- **Usando o detector de energia ARM para monitorar o consumo real de energia, Corrente e Voltagem**
- **Descubra a chance de melhores soluções de gerenciamento de energia**
- **A optimização da tarefa de cálculo é para atingir o melhor eficiência energética**

Use os recursos do sistema de forma eficaz

- **Analisando e otimizando o Mali GPU e o código do CPU.**
- **Monitorando o CPU e o Cache do GPU Mali e memória de sistema**
- **Verificando a distribuição da carga entre vários núcleos.**

Personalize o sistema

- **Verifique seus dados para conectar Dinamize vista de análise**
- **Expandir drivers de código aberto para as variáveis de relógios e componentes**
- **Testar o código como printf que envia mensagem para agilizar**

Antes do desenvolvimento, o que preciso ter:

1) Ubuntu12.04 como Sistema Operacional do computador
2) Uma Cubietruck
3) Cabo de dados USB-miniUSB para a interação de dados do PC e da Placa (O sistema Linux não tem ferramentas ADB, interação dados através da rede)
4) Faça o download do código fonte do Android cubieboard ou linux
5) Faça o download do pacote de código fonte DS-5

Faça o download do código fonte e ferramentas:

Código fonte do Linux

```
$ mkdir linux-sdk-card $ cd linux-sdk-card 1)
```

Código do Kernel

```
$ git clone https://github.com/cubieboard/linux-sdk-kernel-source.git $ mv linux-sdk-kernel-source linux-sunxi
```

Ferramentas

```
$ git clone https://github.com/cubieboard/linux-sdk-card-tools.git

$ mv linux-sdk-card-tools tools
```

Produtos

```
$ git clone https://github.com/cubieboard/linux-sdk-card-products.git

$ mv linux-sdk-card-products products
```

rootfs&u-boot:

```
$ git clone https://github.com/cubieboard/linux-sdk-binaries.git $ mv linux-sdk-binaries binaries

$ mv linux-sdk-binaries binaries
```

Baixe o arquivo do endereço:

http://dl.cubieboard.org/model/commom/linux-sdk-binaries

Código Fonte do Android 4.2

https://bitbucket.org/cubietech/a20-android4.2_lichee.git

Ferramenta DS-5 para o pacote de código-fonte

http://pan.baidu.com/s/1pJG66b L

Adicionar opções de Kernel

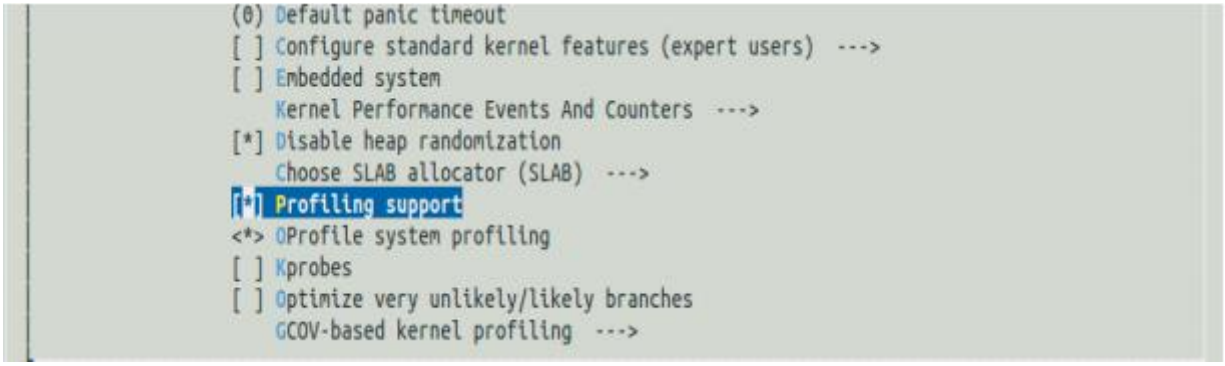
O ARM Streamline precisa recompilar o kernel, a versão pública do SDK.
O Diretório do Android kernel é:

```
lichee/linux-3.4
```

O diretório kernel do linux é linux-sunxi .

Configurações Gerais

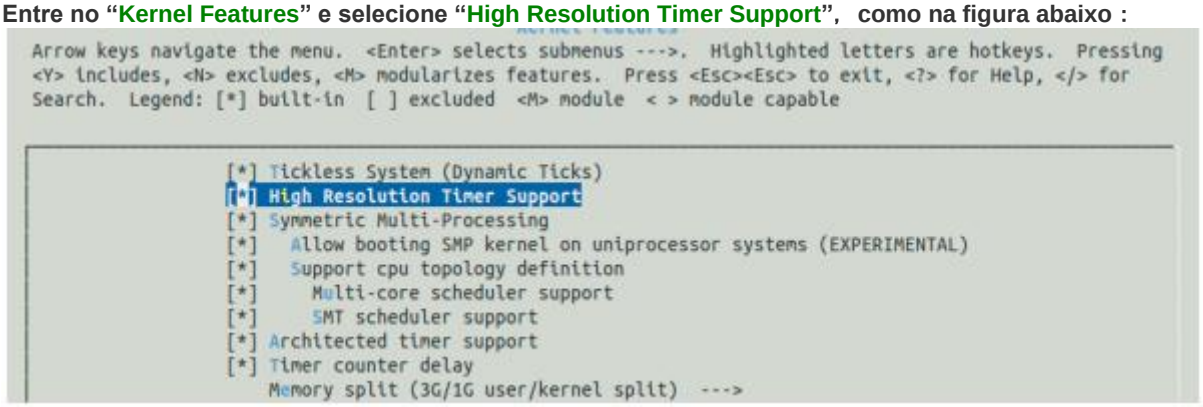
Em "Configurações Gerais" marque o "Profiling Support" como na figura abaixo:



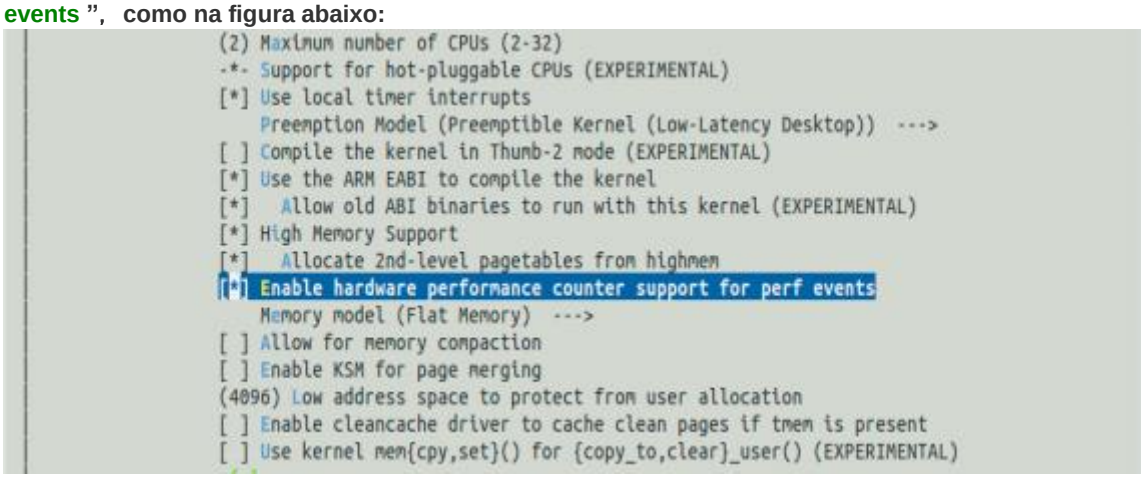
Entre em "Configurações Gerais" e marque "Kernel Performance Events And Counters" como na figura abaixo:

Destaques do Kernel

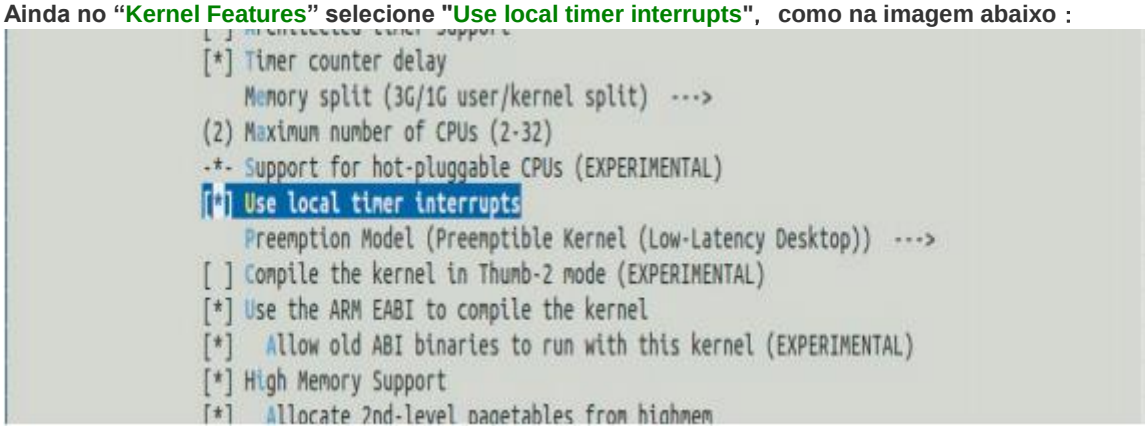
Entre no "Kernel Features" e selecione "High Resolution Timer Support", como na figura abaixo :



Entre no “Kernel Features” e selecione “Enable hardware performance counter support for perf events ”, como na figura abaixo:

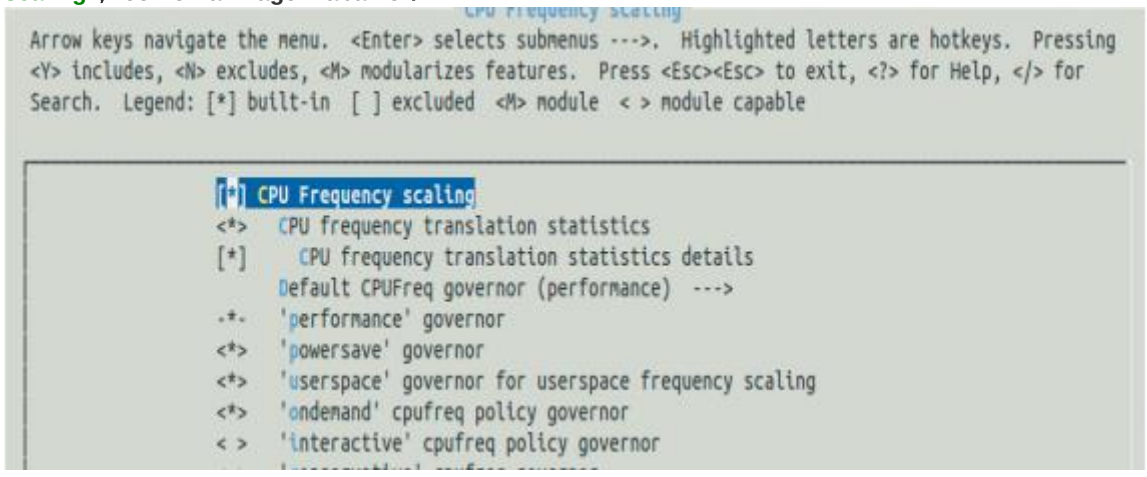


Ainda no “Kernel Features” selecione "Use local timer interrupts", como na imagem abaixo :



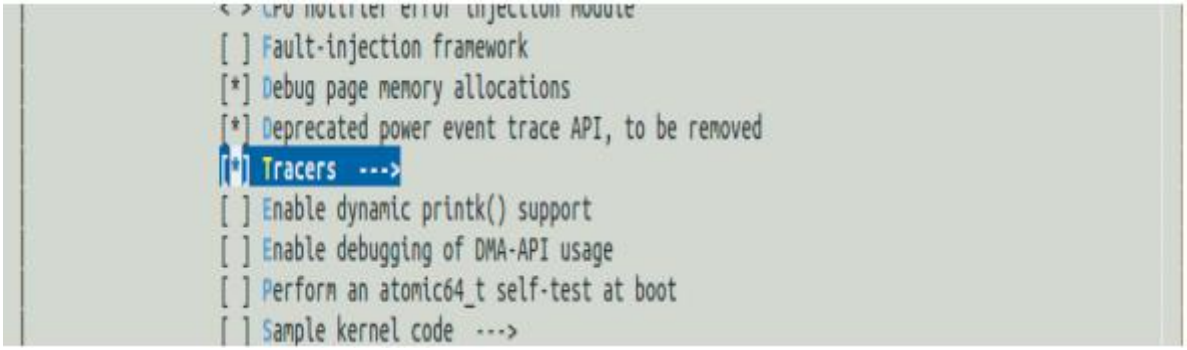
CPU Power Management

Entre no CPU Power Management -> CPU Frequency scaling e selecione “CPU Frequency scaling”, como na imagem abaixo :

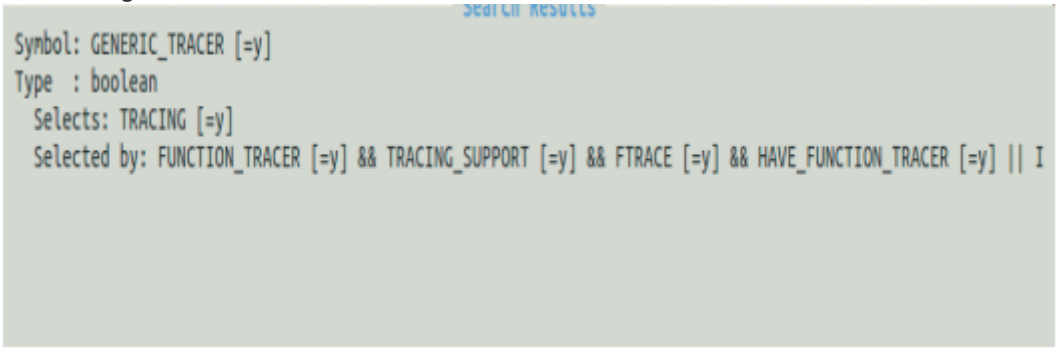


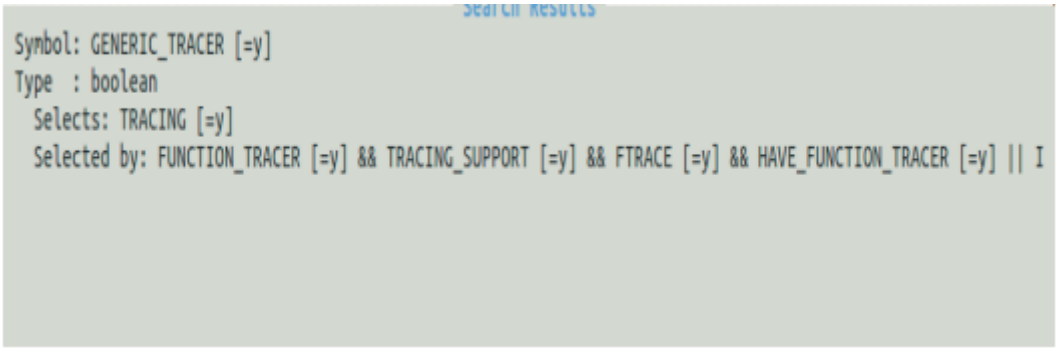
Kernel hacking

Entre em "Kernel Features" e selecione "Tracers", como na figura abaixo



Finalmente confirme se "CONFIG_GENERIC_TRACER" e "CONFIG_TRACING" estão selecionados, como na figura abaixo :





Quando as opções acima forem selecionadas, compile o kernel, fazendo firmware, vamos usar o firmware mais tarde.

Compile o Gator Driver

Unzip o pacote Código Fonte do primeiro passo e obtenha o driver gator-driver . Em seguida, comece a compilar o drivergator.ko.

Entre no diretório gator-driver

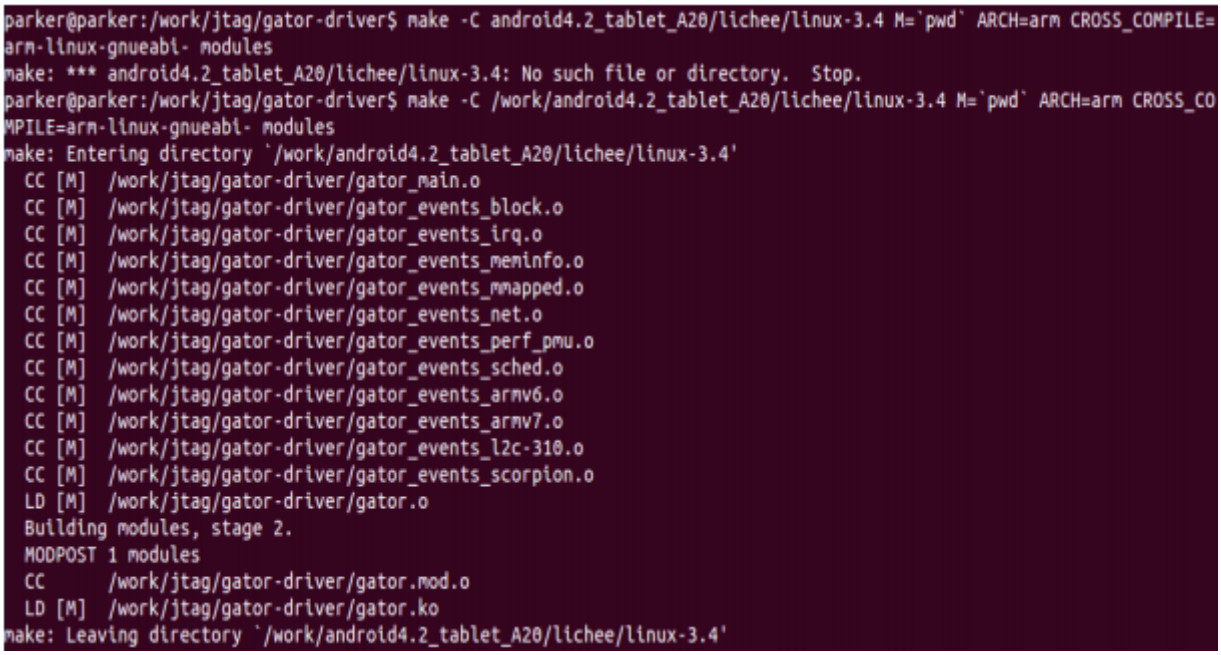
```
$ cd gator-driver
```

Substitua o path correspondente ao kernel. Garanta que tenha instalado a ferramenta de compilação no ambiente de desenvolvimento do seu computador

```
$ make -C /work/android4.2_tablet_A20/lichee/linux-3.4 M='pwd' ARCH=arm

CROSS_COMPILE=arm-linux-gnueabi- modules
```

Será gerado um arquivo gator.ko no diretório atual depois de compilado com sucesso, como mostra a figura abaixo:

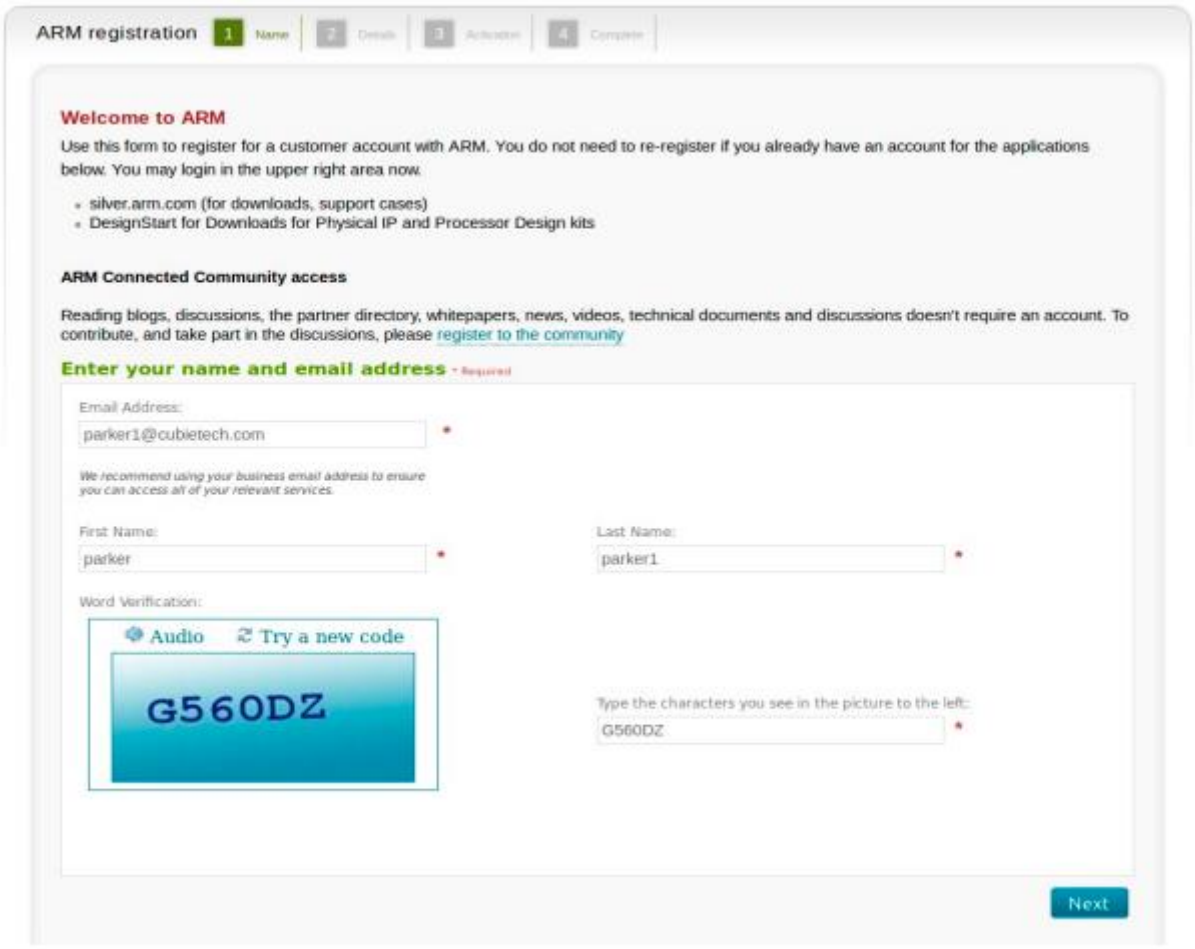


Registre uma Conta ARM

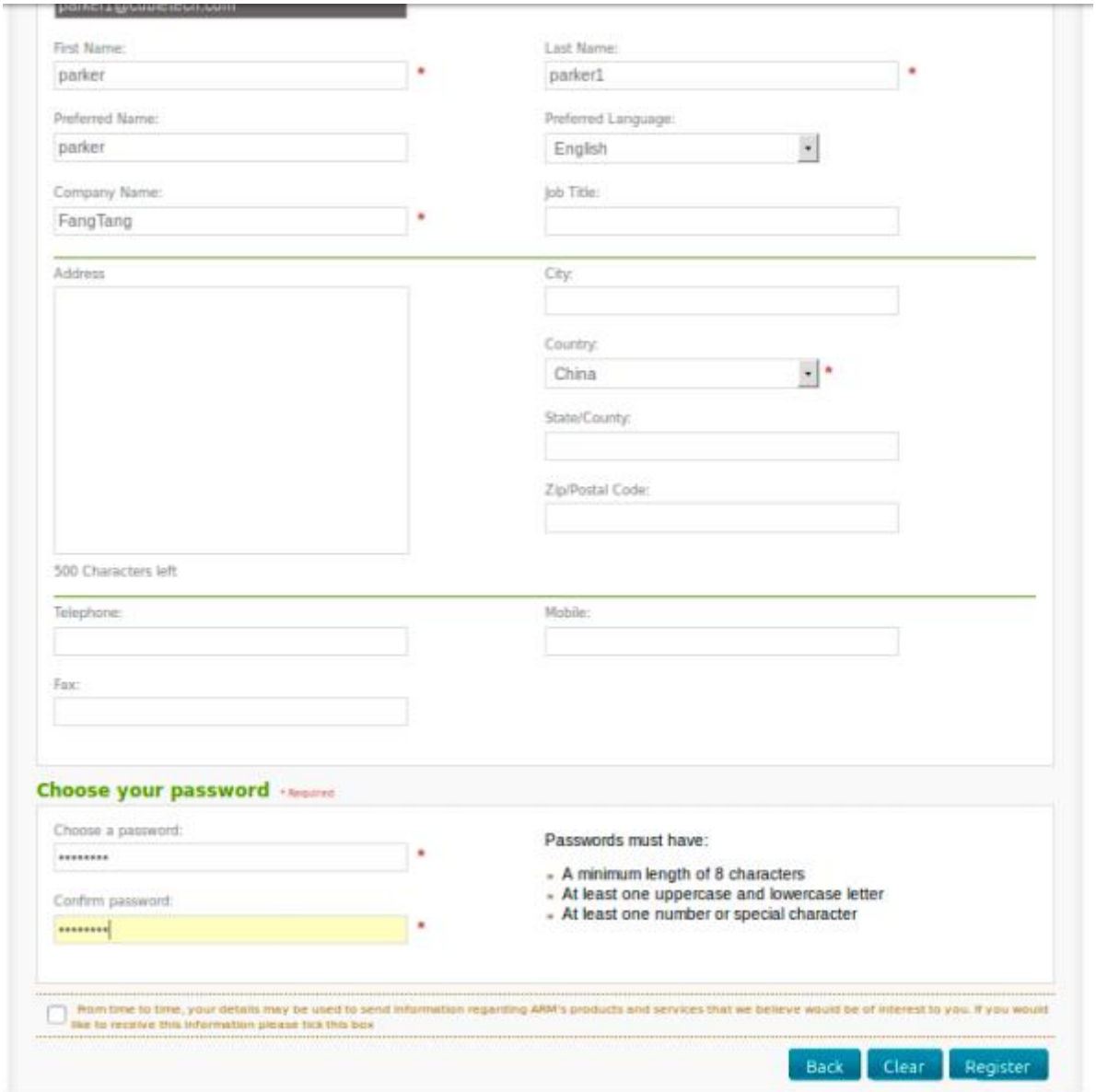
Acesse a página de registro da ARM diretamente pelo endereço abaixo:

https://login.arm.com/register.php

Registre-se em 4 passos



Continue a preencher as informações:



As informações de regitro serão enviados para o seu email. Você pode completar o registro clicando no link que será enviado no seu email.

Instalando DS-5 no seu computador

Entre no diretório do código fonte do DS500-BN-00019-r5p0-20rel1

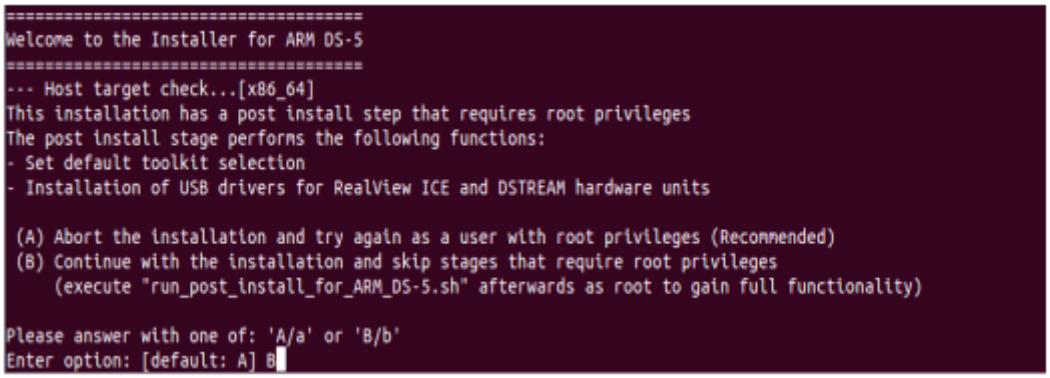
Adicione as permissões executáveis do script:

```
$ chmod +x install.sh
```

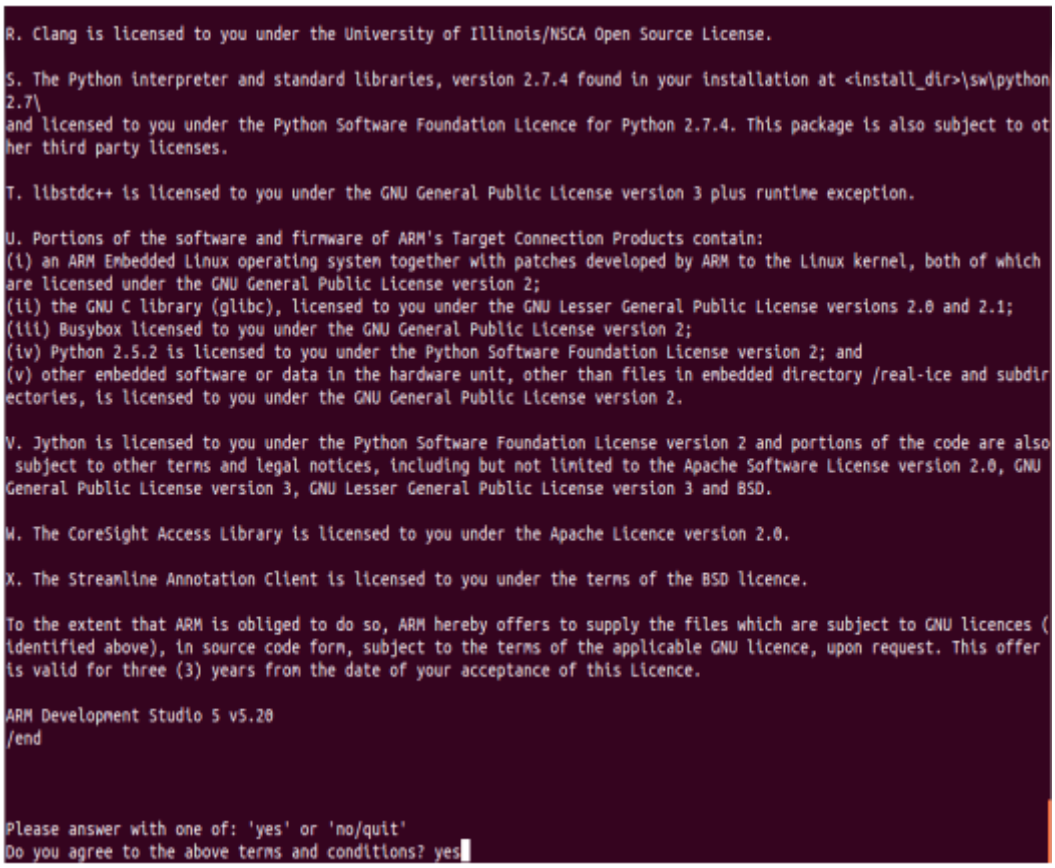
Execute o script:

```
$ ./install.sh
```

Digite B para entrar no próximo passo, como mostra a figura abaixo:



Pressione ENTER para ler as informações e digite "yes", como mostra a figura abaixo:



Digite "yes", selecione o diretório de instalação DS-5, como mostrado na figura abaixo:

```
Please answer with one of: 'yes/y' or 'no/n'
Run installation platform requirement checks? [default: yes] yes

--- Running installation platform requirement checks

Running dependency check [succeeded]

Where would you like to install to? [default: /home/parker/DS-5]
```

Espere pela instalação, digite "yes" depois de concluído, como mostra a figura abaixo:

```
Please answer with one of: 'yes/y' or 'no/n'
Install desktop menu item additions? [default: yes] yes

--- Installing menu entries

--- Skipping post install setup scripts
    You can run these later by executing ./run_post_install_for_ARM_DS-5.sh with root privileges from inside the inst
allation.

-----------------------------------
Installation completed successfully
-----------------------------------

To start using ARM DS-5 either:
- Add /home/parker/DS-5/bin to your PATH
- Create a suite sub-shell using /home/parker/DS-5/bin/suite_exec <shell>
- Launch GUI tools via their desktop menu entries

The Release notes for the product can be found here: file:///home/parker/DS-5/sw/ARM_DS-5/readme.html

===================================
parker@parker:/work/jtag/DS-5/DS500-BN-00019-r5p0-20rel1$
```

Adicione a DS-5 environment variable, o bin folde abaixo do diretório de instalação DS-5 r. Precisamos pegar esta variável, como mostra na figura abaixo:

```
$ vim ~/.bashrc
```

```
JAVA_HOME=/work/tools/jdk1.6.0_45
export ANDROID_SDK_PATH=/work/tools/adt-bundle-linux-x86_64-20140321/sdk
export JRE_HOME=/work/tools/jdk1.6.0_45/jre
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$ANDROID_SDK_PATH/platform-tools:/home/parker/DS-5/bin:$PATH
```

```
$ source ~/.bashrc
```

Abra as ferramentas de Debug no CT

Carregue o drive e shell

O gator.ko compilado anteriormente e código fonte no diretório gatord /data , carregue para a placa.

```
$ adb push gator.ko /data
$ adb push gatord /data
```

Entre no arquivo de sistema da placa

```
$ adb shell
```

Carregue o driver gator.ko

```
$ insmod gator.ko
```

Executando o gatord

```
$ chmod 777 gatord
$ ./gatord &
```

```
root@android:/data # insmod gator.ko
root@android:/data # lsmod
gator 57201 0 - Live 0x00000000 (O)
cdc_ether 3163 0 - Live 0x00000000
rtl8150 9023 0 - Live 0x00000000
mcs7830 5644 0 - Live 0x00000000
qf9700 5884 0 - Live 0x00000000
asix 13586 0 - Live 0x00000000
usbnet 13741 4 cdc_ether,mcs7830,qf9700,asix, Live 0x00000000
sunxi_csi0 30818 0 - Live 0x00000000
gc2035 13734 0 - Live 0x00000000
gc0308 11800 0 - Live 0x00000000
camera 36086 1 sunxi_csi0, Live 0x00000000
videobuf_dma_contig 4157 1 sunxi_csi0, Live 0x00000000
videobuf_core 16284 2 sunxi_csi0,videobuf_dma_contig, Live 0x00000000
sun7i_ir 5797 0 - Live 0x00000000
security_system 1067129 0 - Live 0x00000000
sw_device 11512 0 - Live 0x00000000
mali 151201 31 - Live 0x00000000 (O)
hdmi 25437 0 - Live 0x00000000 (O)
lcd 5155 0 - Live 0x00000000
disp 288683 13 mali,hdmi,lcd, Live 0x00000000
nand 142727 8 - Live 0x00000000 (O)
root@android:/data # chmod +x gatord
Bad mode
10|root@android:/data # chmod 777 gatord
root@android:/data # ./gatord &
[1] 3646
```

Use o ADB para dados interativos

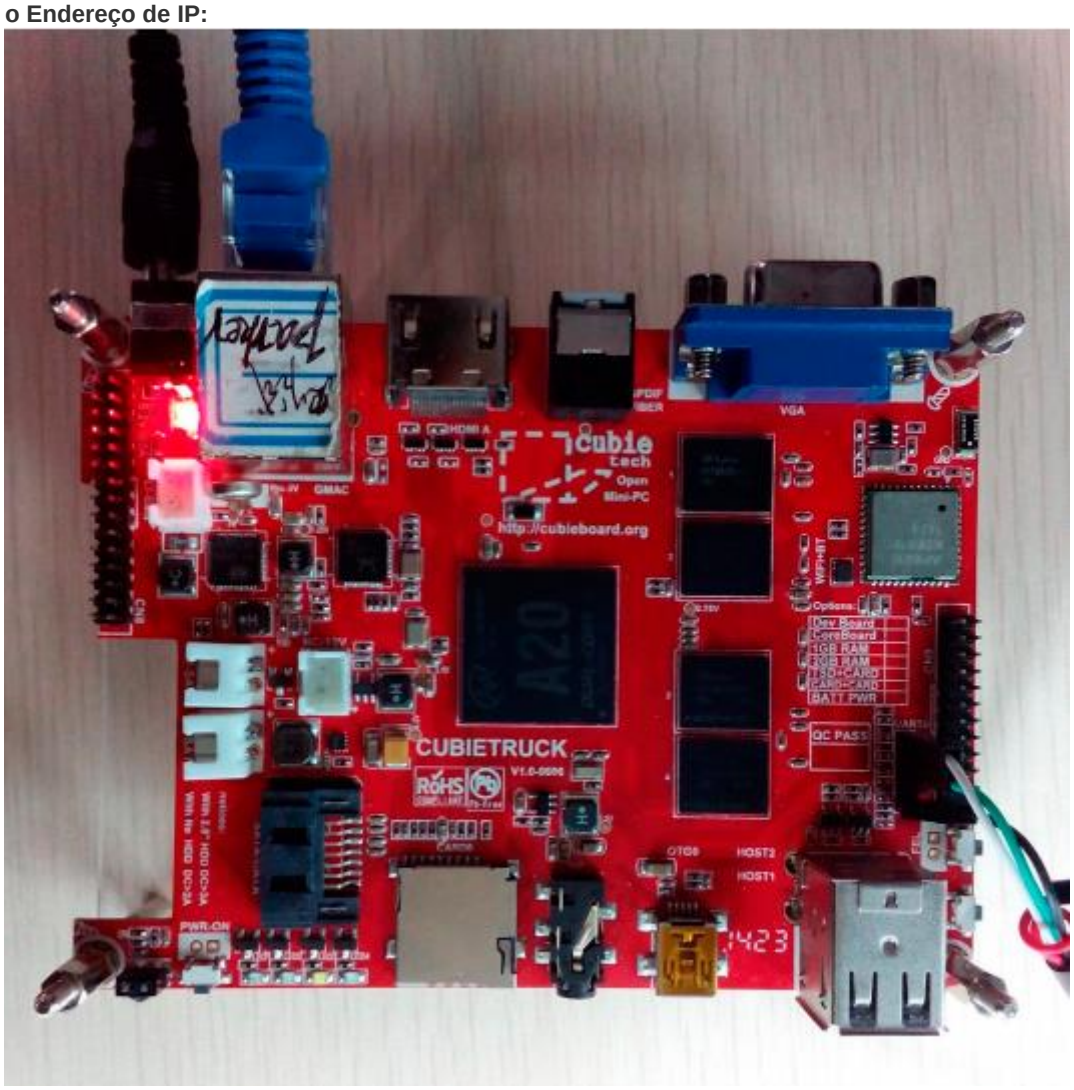
Android usa mini-USB como a mídia de transmissão de dados do CT e PC. O desenvolvimento do ADB
e suas opções de depuração foram ativadas por padrão, só precisamos connectá-lo ao hardware, para
garantir que o PC possa identificar a CT.

O método de conexão é muito simples, só precisa de energia e ser conectado ao mini-usb:

A utilização de dados de rede interativo

Usar o mini-USB é apenas uma maneira, você pode ser conectado ao PC e CT pela rede e ferramentas de depuração de série, caso não haja ferramentas mini-USB e ADB,
mas gator.ko e gatord
vão usar o stick USB ou cópia do cartão do TF para a placa interna.

Se você estiver usando Wi-Fi, você não precisa levar o cabo de rede, apenas certifique-se de obter o Endereço de IP:



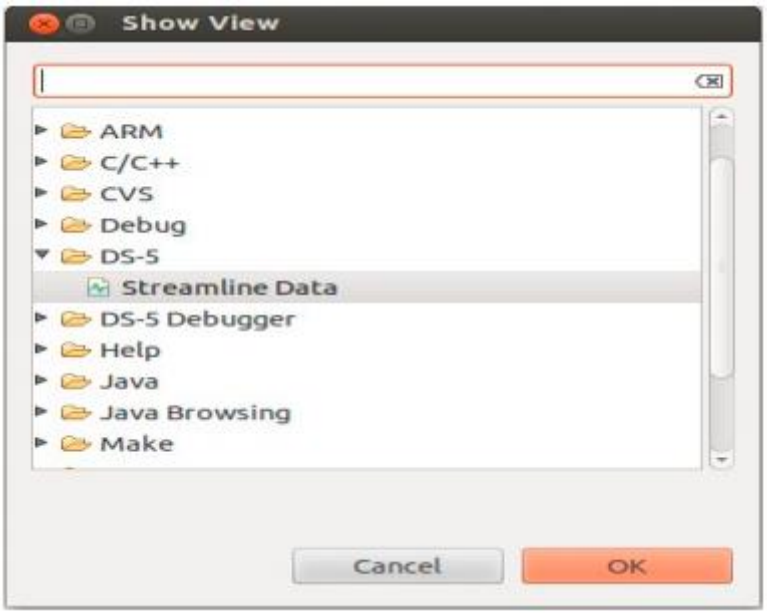
Use DS-5

Terminal input

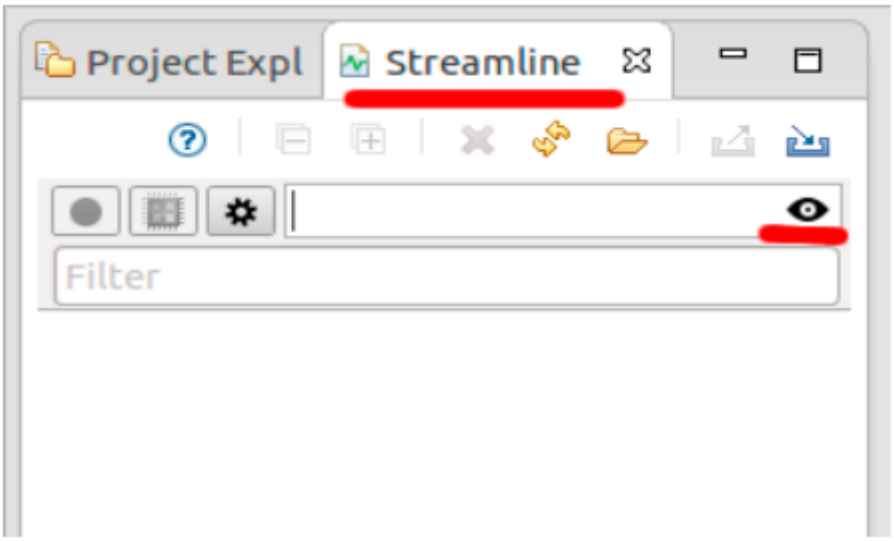
```
$ eclipse
```

Crie o projeto da Streamline Data

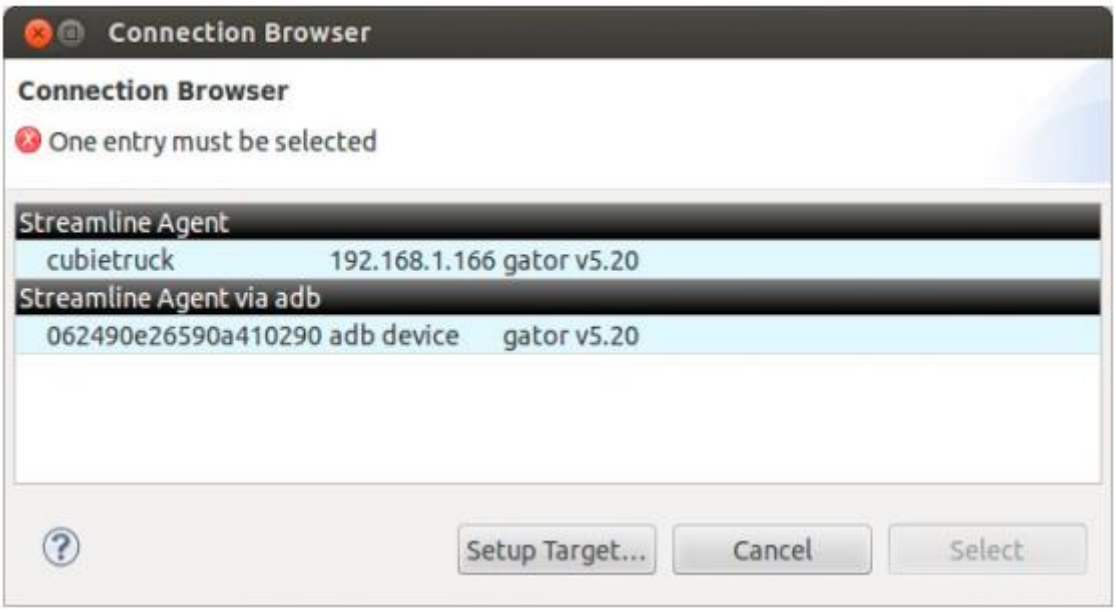
Clique na barra de menu na Janela> Mostrar Visualização> Outro ..., selecione Dados Streamline, clique em
"OK", como mostrado abaixo:



Vamos configurar a placa, verifique se a USB-Line foi conectado ao computador e cartão, selecione o projeto Streamline e clique no ícone com o desenho de um olho, como mostrado abaixo:



Em seguida, selecione a opção "Streamline Agent via ADB", como mostrado na figura abaixo:



Por fim, clique no ícone vermelho, selecione o caminho para salvar, você pode iniciar a depuração, como mostrado abaixo:

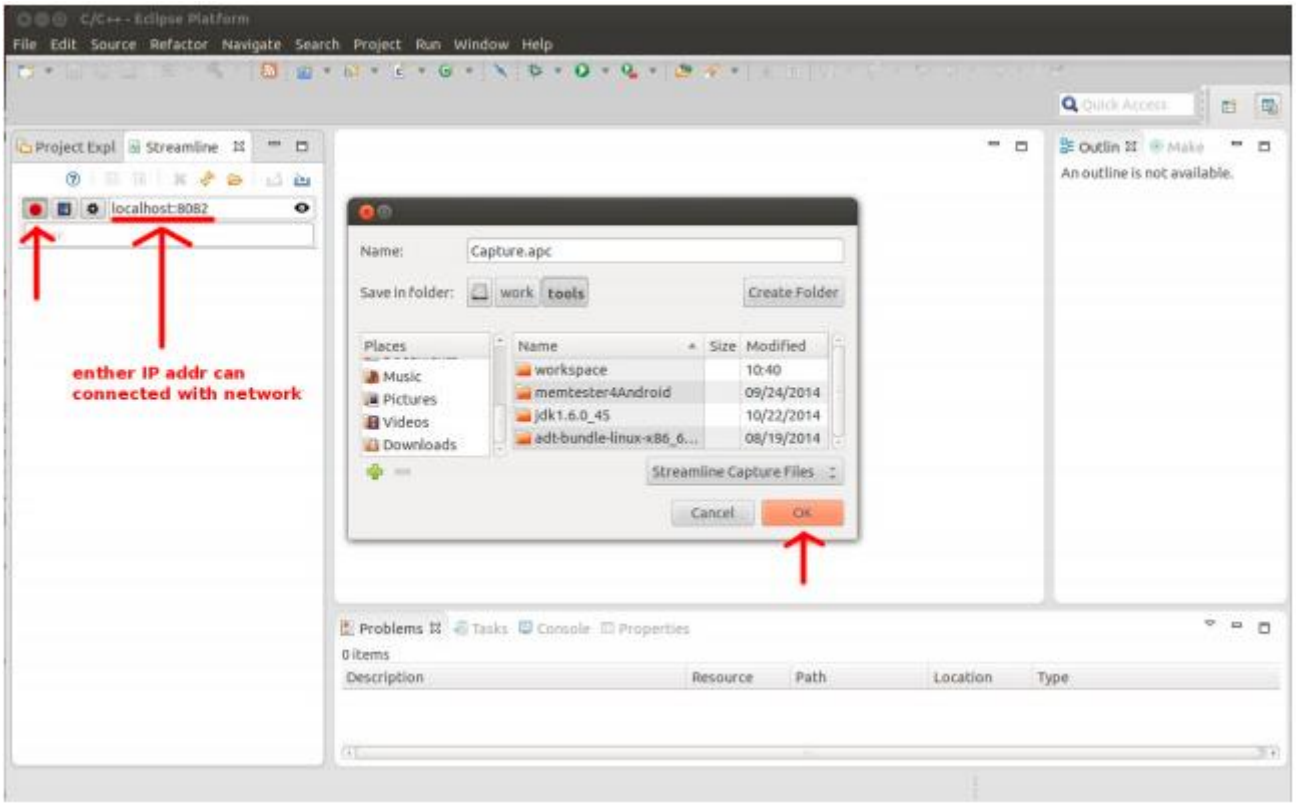
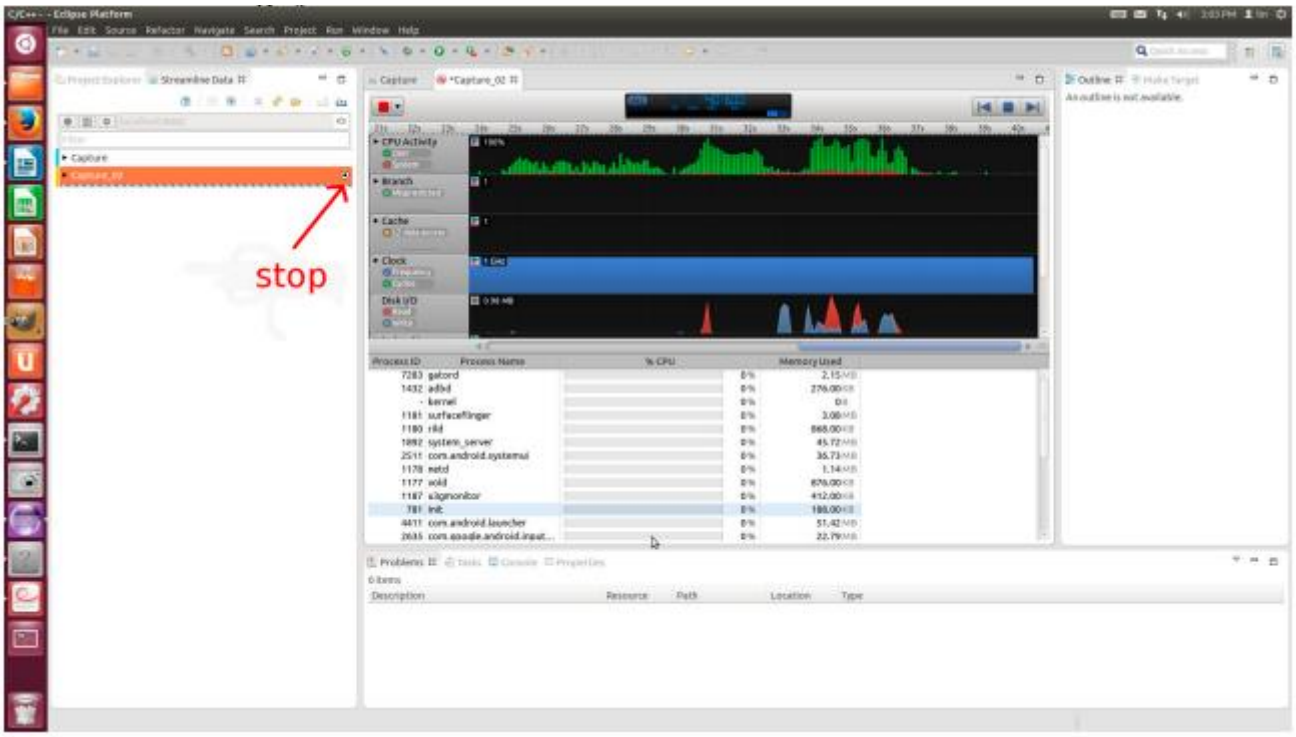
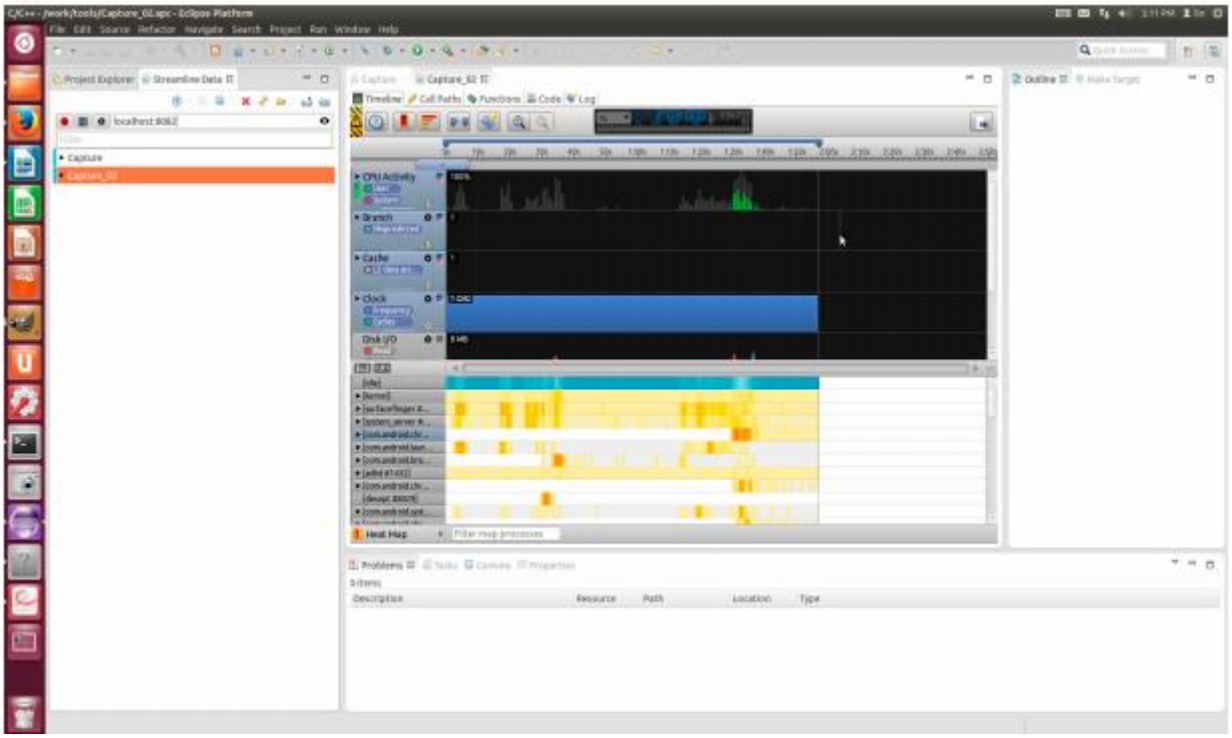


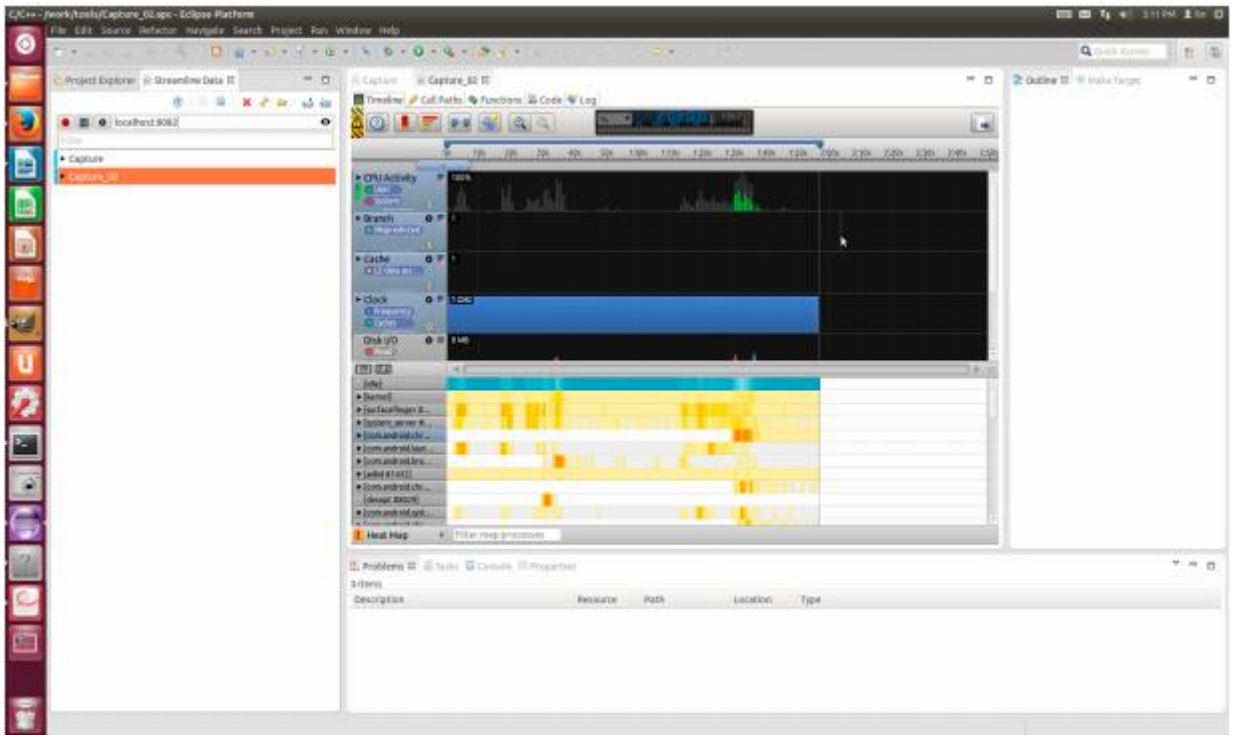
Imagem do DS-5 funcionando

DS-5 está monitorando o sistema:

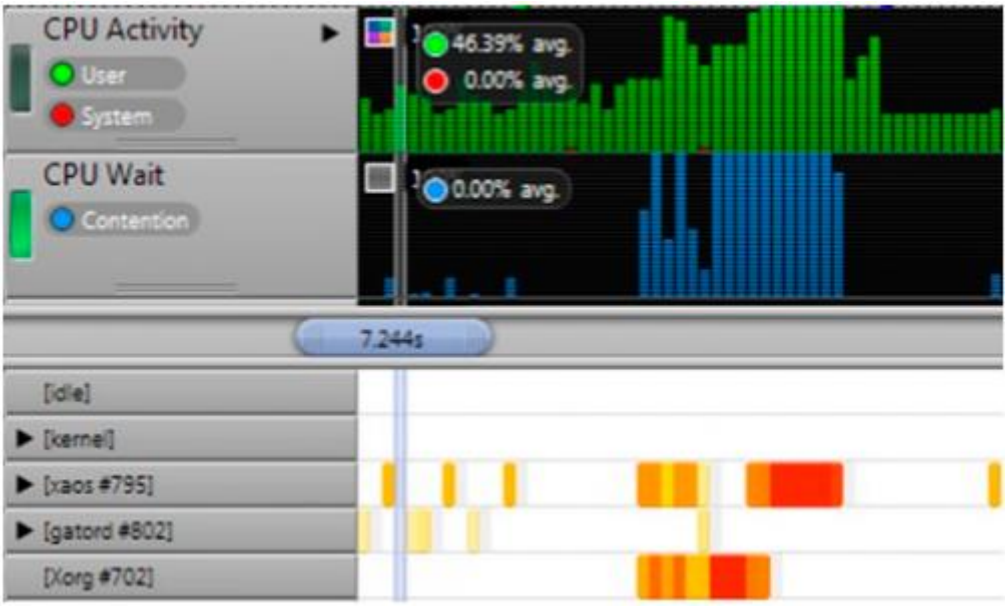


Na imagem abaixo o DS-5 pára de funcionar e apresenta os resultados analíticos obtidos:





Uma simples análise do Streamline



Fonte: http://dl.cubieboard.org/developers/debug-tools/ARM-DS-5/docs/Using%20ARM%20Streamline%20base%20on%20Cubieboard.pdf