

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 consome 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 otimização 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/1pJG66bL>

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:

```
(0) Default panic timeout
[ ] Configure standard kernel features (expert users) --->
[ ] Embedded system
  Kernel Performance Events And Counters --->
[*] Disable heap randomization
  Choose SLAB allocator (SLAB) --->
[*] Profiling support
<*> Profile system profiling
[ ] Kprobes
[ ] Optimize very unlikely/likely branches
  GCOV-based kernel profiling --->
```

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

```
Kernel Performance Events And Counters
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

[*] Kernel performance events and counters
[ ] Kernel performance counters (old config option)
[ ] Debug: use vmaalloc to back perf mmap() buffers
```

Destaques do Kernel

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

```
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

[*] Tickless System (Dynamic Ticks)
[*] High Resolution Timer Support
[*] Symmetric Multi-Processing
[*] Allow booting SMP kernel on uniprocessor systems (EXPERIMENTAL)
[*] Support cpu topology definition
[*] Multi-core scheduler support
[*] SMT scheduler support
[*] Architected timer support
[*] Timer counter delay
  Memory split (3G/1G user/kernel split) --->
```

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

```

(2) Maximum number of CPUs (2-32)
-* Support for hot-pluggable CPUs (EXPERIMENTAL)
[*] Use local timer interrupts
    Preemption Model (Preemptible Kernel (Low-Latency Desktop)) --->
[ ] Compile the kernel in Thumb-2 mode (EXPERIMENTAL)
[*] Use the ARM EABI to compile the kernel
[*] Allow old ABI binaries to run with this kernel (EXPERIMENTAL)
[*] High Memory Support
[*] Allocate 2nd-level pagetables from highmem
[*] Enable hardware performance counter support for perf events
    Memory model (Flat Memory) --->
[ ] Allow for memory compaction
[ ] Enable KSM for page merging
(4096) Low address space to protect from user allocation
[ ] Enable cleancache driver to cache clean pages if tmem is present
[ ] Use kernel mem(cpy,set)() for {copy_to,clear}_user() (EXPERIMENTAL)

```

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

```

[ ] ...
[*] Timer counter delay
    Memory split (3G/1G user/kernel split) --->
(2) Maximum number of CPUs (2-32)
-* Support for hot-pluggable CPUs (EXPERIMENTAL)
[*] Use local timer interrupts
    Preemption Model (Preemptible Kernel (Low-Latency Desktop)) --->
[ ] Compile the kernel in Thumb-2 mode (EXPERIMENTAL)
[*] Use the ARM EABI to compile the kernel
[*] Allow old ABI binaries to run with this kernel (EXPERIMENTAL)
[*] High Memory Support
[*] Allocate 2nd-level pagetables from highmem

```

CPU Power Management

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

```

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

CPU Frequency scaling
[*] CPU Frequency scaling
<*> CPU frequency translation statistics
[*] CPU frequency translation statistics details
    Default CPUFreq governor (performance) --->
    -* 'performance' governor
<*> 'powersave' governor
<*> 'userspace' governor for userspace frequency scaling
<*> 'ondemand' cpufreq policy governor
< > 'interactive' cpufreq policy governor

```

Kernel hacking

Entre em “**Kernel Features**” e seleccione “**Tracers**”, como na figura abaixo

```
< > CPU notifier error injection module
[ ] Fault-injection framework
[*] Debug page memory allocations
[*] Deprecated power event trace API, to be removed
[*] Tracers --->
[ ] Enable dynamic printk() support
[ ] Enable debugging of DMA-API usage
[ ] Perform an atomic64_t self-test at boot
[ ] Sample kernel code --->
```

Finalmente confirme se “**CONFIG_GENERIC_TRACER**” e “**CONFIG_TRACING**” estão seleccionados, como na figura abaixo :

```
Symbol: GENERIC_TRACER [=y]
Type : boolean
Selects: TRACING [=y]
Selected by: FUNCTION_TRACER [=y] && TRACING_SUPPORT [=y] && FTRACE [=y] && HAVE_FUNCTION_TRACER [=y] || I
```

```
Symbol: GENERIC_TRACER [=y]
Type : boolean
Selects: TRACING [=y]
Selected by: FUNCTION_TRACER [=y] && TRACING_SUPPORT [=y] && FTRACE [=y] && HAVE_FUNCTION_TRACER [=y] || I
```

Quando as opções acima forem seleccionadas, 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 driver [gator.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:

```
parker@parker:/work/jtag/gator-driver$ make -C android4.2_tablet_A20/lichee/linux-3.4 M='pwd' ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- modules
make: *** android4.2_tablet_A20/lichee/linux-3.4: No such file or directory. Stop.
parker@parker:/work/jtag/gator-driver$ make -C /work/android4.2_tablet_A20/lichee/linux-3.4 M='pwd' ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- modules
make: Entering directory `/work/android4.2_tablet_A20/lichee/linux-3.4'
  CC [M] /work/jtag/gator-driver/gator_main.o
  CC [M] /work/jtag/gator-driver/gator_events_block.o
  CC [M] /work/jtag/gator-driver/gator_events_irq.o
  CC [M] /work/jtag/gator-driver/gator_events_meminfo.o
  CC [M] /work/jtag/gator-driver/gator_events_mmapped.o
  CC [M] /work/jtag/gator-driver/gator_events_net.o
  CC [M] /work/jtag/gator-driver/gator_events_perf_pmu.o
  CC [M] /work/jtag/gator-driver/gator_events_sched.o
  CC [M] /work/jtag/gator-driver/gator_events_armv6.o
  CC [M] /work/jtag/gator-driver/gator_events_armv7.o
  CC [M] /work/jtag/gator-driver/gator_events_l2c-310.o
  CC [M] /work/jtag/gator-driver/gator_events_scorpion.o
  LD [M] /work/jtag/gator-driver/gator.o
Building modules, stage 2.
MODPOST 1 modules
  CC /work/jtag/gator-driver/gator.mod.o
  LD [M] /work/jtag/gator-driver/gator.ko
make: Leaving directory `/work/android4.2_tablet_A20/lichee/linux-3.4'
```

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

ARM registration **1** Name | 2 Details | 3 Activation | 4 Complete

Welcome to ARM

Use this form to register for a customer account with ARM. You do not need to re-register if you already have an account for the applications below. You may login in the upper right area now.

- silver.arm.com (for downloads, support cases)
- DesignStart for Downloads for Physical IP and Processor Design kits

ARM Connected Community access

Reading blogs, discussions, the partner directory, whitepapers, news, videos, technical documents and discussions doesn't require an account. To contribute, and take part in the discussions, please [register to the community](#)

Enter your name and email address * Required


Email Address:
 *

We recommend using your business email address to ensure you can access all of your relevant services.

First Name: * Last Name: *

Word Verification:

[Audio](#) [Try a new code](#)



Type the characters you see in the picture to the left:
 *

[Next](#)

Continue a preencher as informações:

ARM registration form

First Name: parker *	Last Name: parker1 *
Preferred Name: parker	Preferred Language: English
Company Name: FangTang *	Job Title:
Address: <div style="border: 1px solid #ccc; height: 100px;"></div> 500 Characters left	City: <div style="border: 1px solid #ccc; height: 20px;"></div>
	Country: China *
	State/Country: <div style="border: 1px solid #ccc; height: 20px;"></div>
	Zip/Postal Code: <div style="border: 1px solid #ccc; height: 20px;"></div>
Telephone: <div style="border: 1px solid #ccc; height: 20px;"></div>	Mobile: <div style="border: 1px solid #ccc; height: 20px;"></div>
Fax: <div style="border: 1px solid #ccc; height: 20px;"></div>	

Choose your password * Required

Choose a password: ***** *	Passwords must have: <ul style="list-style-type: none">• A minimum length of 8 characters• At least one uppercase and lowercase letter• At least one number or special character
Confirm password: ***** *	

From time to time, your details may be used to send information regarding ARM's products and services that we believe would be of interest to you. If you would like to receive this information please tick this box

[Back](#) [Clear](#) [Register](#)

ARM registration **1** Name | **2** Details | **3** Activation | **4** Complete

Thank you for your registration.

An email has been sent to **parker1@cubietech.com** which contains an activation link.

You must click on this link to activate your account before you will be able to access controlled sections of our website.

As informações de registro 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:

```
=====
Welcome to the Installer for ARM DS-5
=====
--- Host target check...[x86_64]
This installation has a post install step that requires root privileges
The post install stage performs the following functions:
- Set default toolkit selection
- Installation of USB drivers for RealView ICE and DSTREAM hardware units

(A) Abort the installation and try again as a user with root privileges (Recommended)
(B) Continue with the installation and skip stages that require root privileges
    (execute "run_post_install_for_ARM_DS-5.sh" afterwards as root to gain full functionality)

Please answer with one of: 'A/a' or 'B/b'
Enter option: [default: A] B
```

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

```
R. Clang is licensed to you under the University of Illinois/NSCA Open Source License.

S. The Python interpreter and standard libraries, version 2.7.4 found in your installation at <install_dir>\sw\python
2.7\
and licensed to you under the Python Software Foundation Licence for Python 2.7.4. This package is also subject to ot
her third party licenses.

T. libstdc++ is licensed to you under the GNU General Public License version 3 plus runtime exception.

U. Portions of the software and firmware of ARM's Target Connection Products contain:
(i) an ARM Embedded Linux operating system together with patches developed by ARM to the Linux kernel, both of which
are licensed under the GNU General Public License version 2;
(ii) the GNU C library (glibc), licensed to you under the GNU Lesser General Public License versions 2.0 and 2.1;
(iii) Busybox licensed to you under the GNU General Public License version 2;
(iv) Python 2.5.2 is licensed to you under the Python Software Foundation License version 2; and
(v) other embedded software or data in the hardware unit, other than files in embedded directory /real-ice and subdir
ectories, is licensed to you under the GNU General Public License version 2.

V. Jython is licensed to you under the Python Software Foundation License version 2 and portions of the code are also
subject to other terms and legal notices, including but not limited to the Apache Software License version 2.0, GNU
General Public License version 3, GNU Lesser General Public License version 3 and BSD.

W. The CoreSight Access Library is licensed to you under the Apache Licence version 2.0.

X. The Streamline Annotation Client is licensed to you under the terms of the BSD licence.

To the extent that ARM is obliged to do so, ARM hereby offers to supply the files which are subject to GNU licences (
identified above), in source code form, subject to the terms of the applicable GNU licence, upon request. This offer
is valid for three (3) years from the date of your acceptance of this licence.

ARM Development Studio 5 v5.20
/end

Please answer with one of: 'yes' or 'no/quit'
Do you agree to the above terms and conditions? yes
```



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 installation.

-----
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/jtaq/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 **gator /data** , carregue para a placa.

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

```
$ adb push gator /data
```

Entre no arquivo de sistema da placa

```
$ adb shell
```

Carregue o driver **gator.ko**

```
$ insmod gator.ko
```

Executando o **gator**

```
$ chmod 777 gator
```

```
$ ./gatord &
```

```
root@android:/data # insmod gator.ko
root@android:/data # lsmod
gator 57201 0 - Live 0x00000000 (0)
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_devlce 11512 0 - Live 0x00000000
mali 151201 31 - Live 0x00000000 (0)
hdmi 25437 0 - Live 0x00000000 (0)
lcd 5155 0 - Live 0x00000000
disp 288683 13 mali,hdmi,lcd, Live 0x00000000
nand 142727 8 - Live 0x00000000 (0)
root@android:/data # chmod +x gator
Bad mode
10|root@android:/data # chmod 777 gator
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 conectá-lo ao hardware, para

garantir que o PC possa identificar a CT.

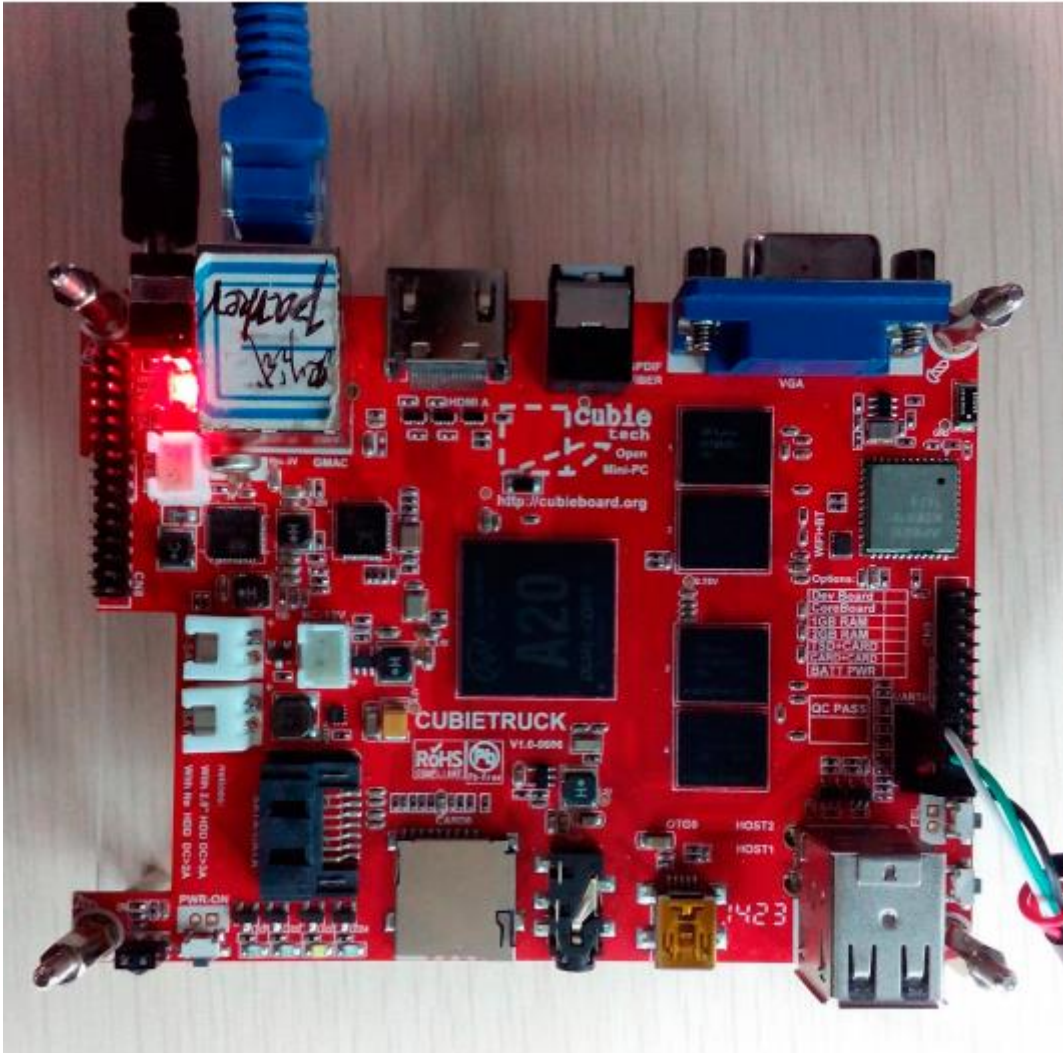
O método de conexão é muito simples, só precisa de energia e ser conectado ao mini-usb:



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 gator.d 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:



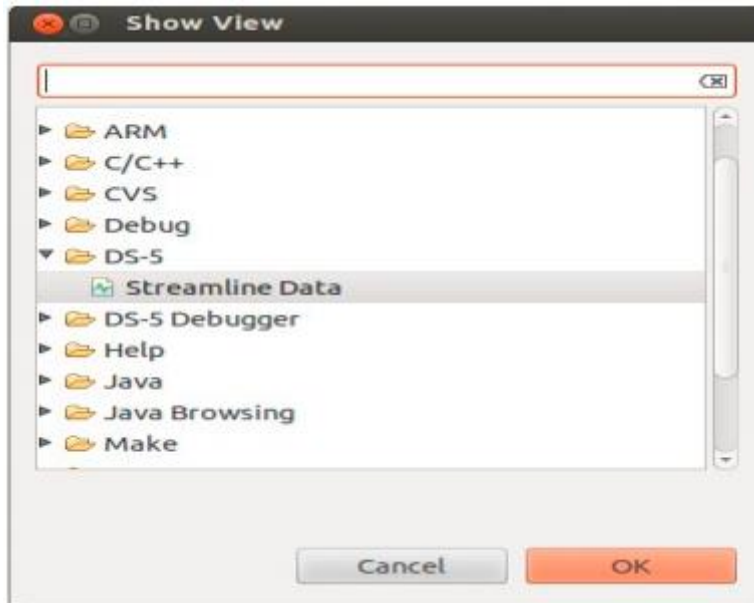
Use DS-5

Terminal input
\$ eclipse

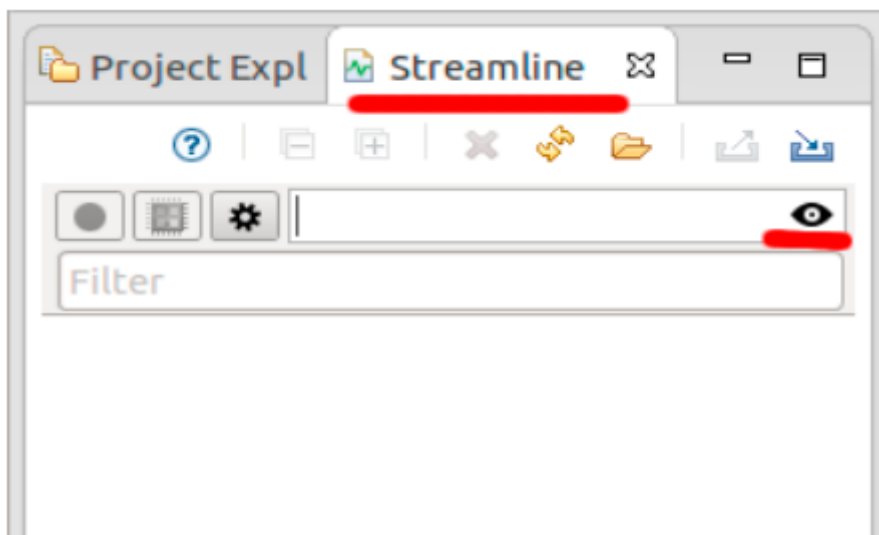
Crie o projeto da Streamline Data

Clique na barra de menu na Janela> Mostrar Visualização> Outro ..., selecione Dados Streamline, clique em

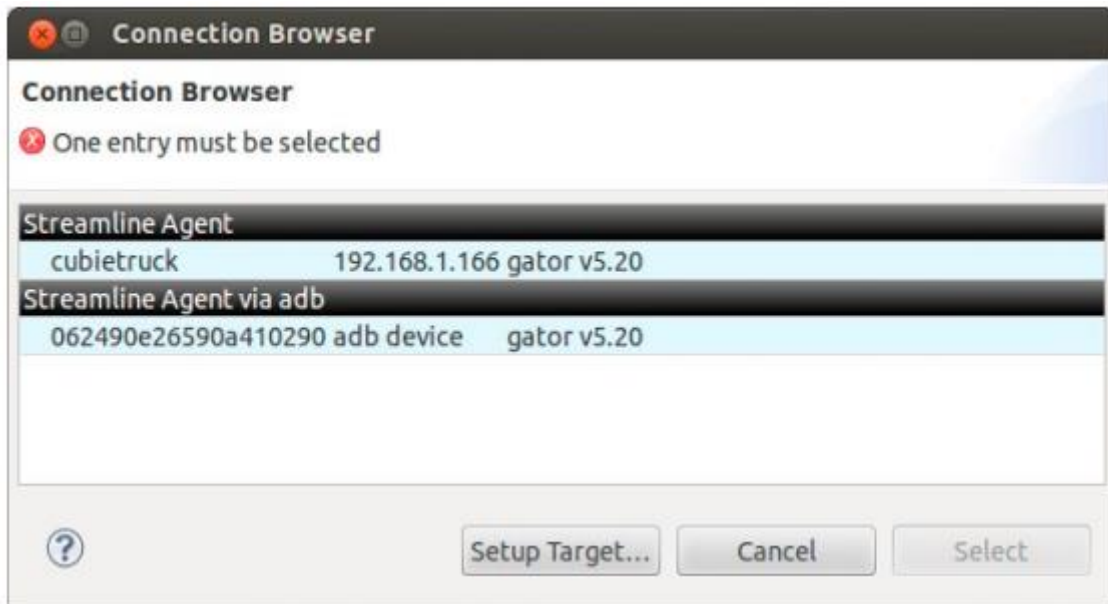
"OK", como mostrado abaixo:



Vamos configurar a placa, verifique se a USB-Line foi conectada 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:

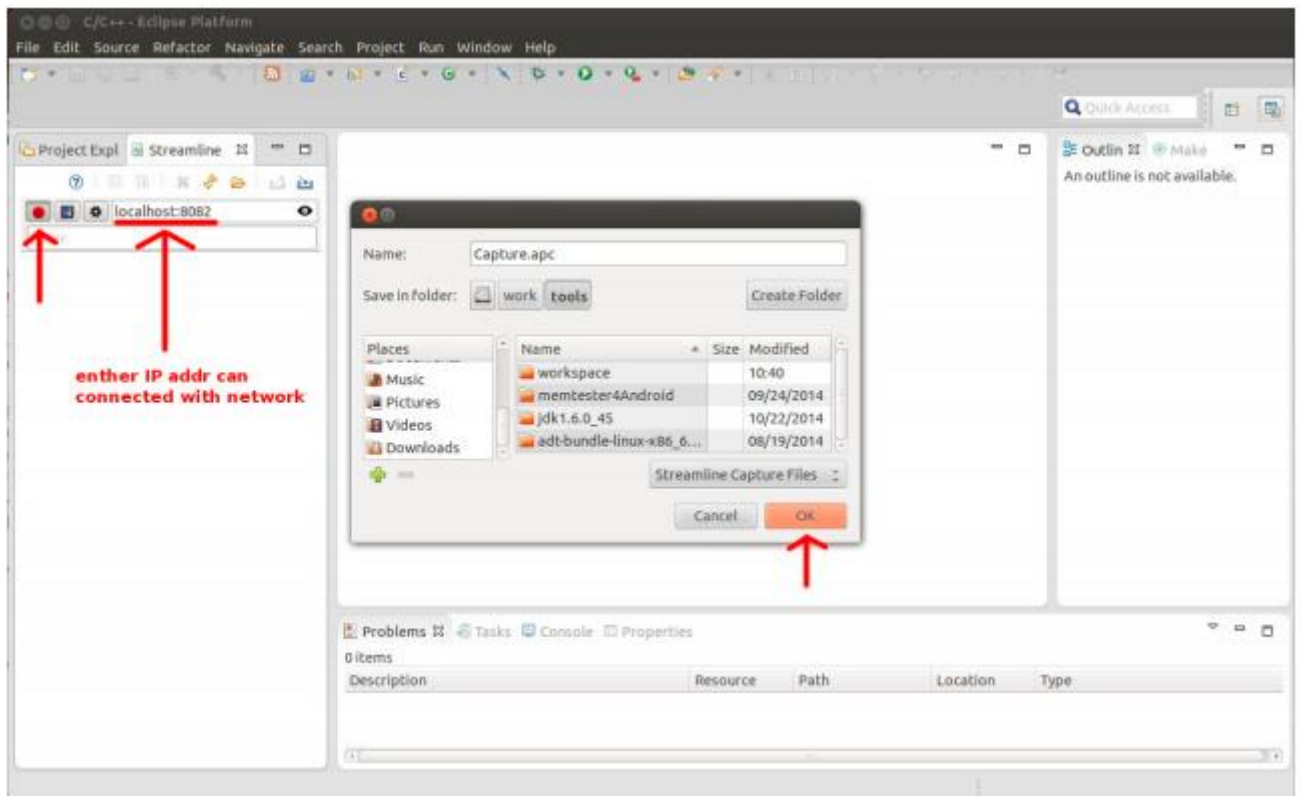
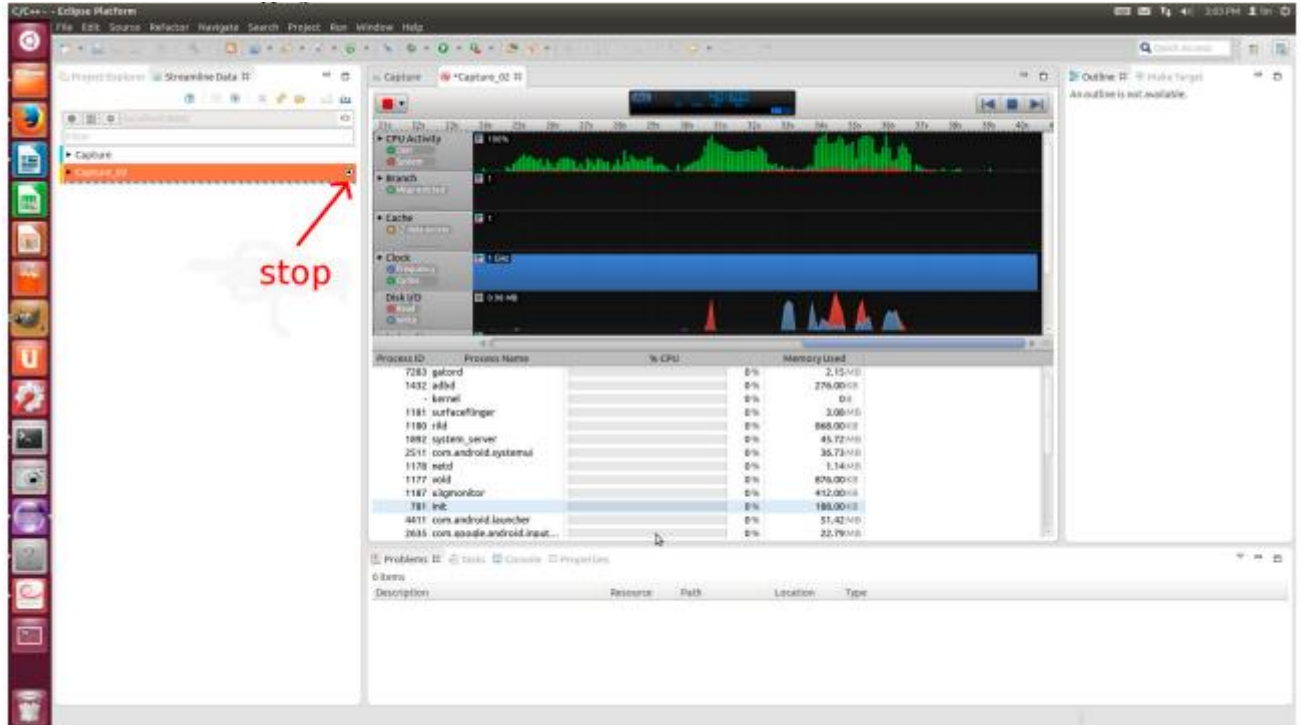
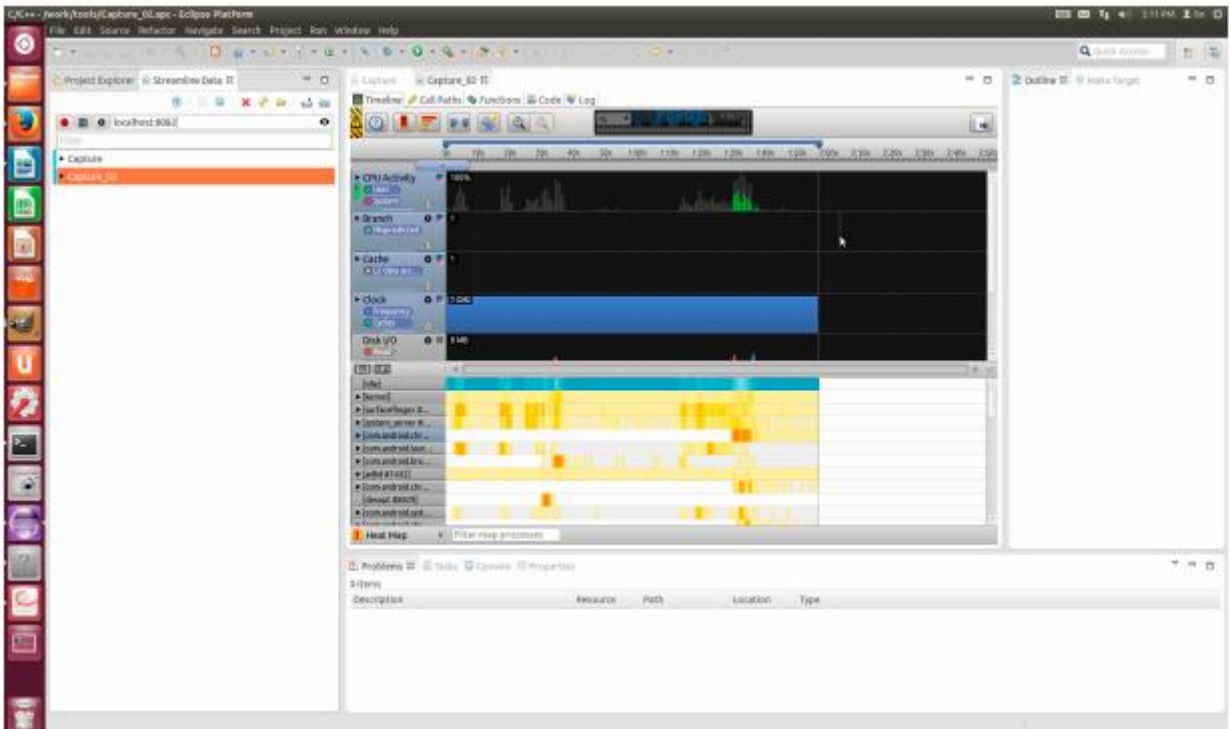


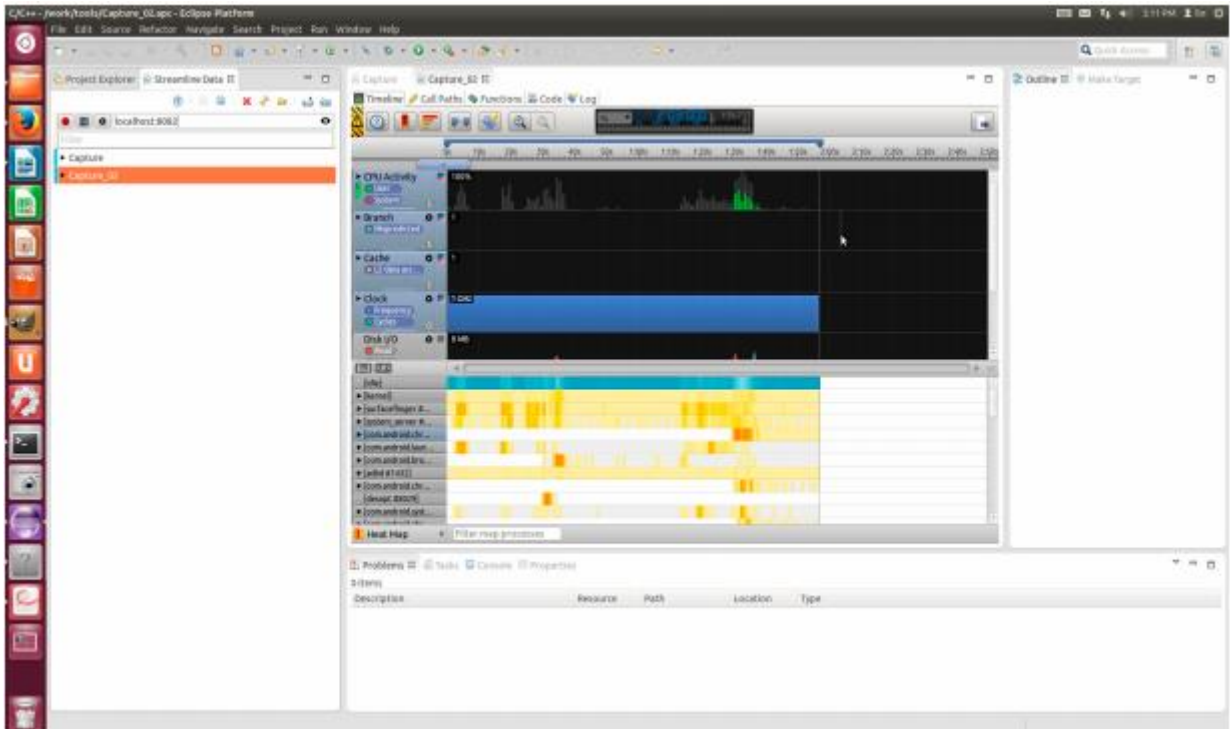
Imagem do DS-5 funcionando

DS-5 está monitorando o sistema:

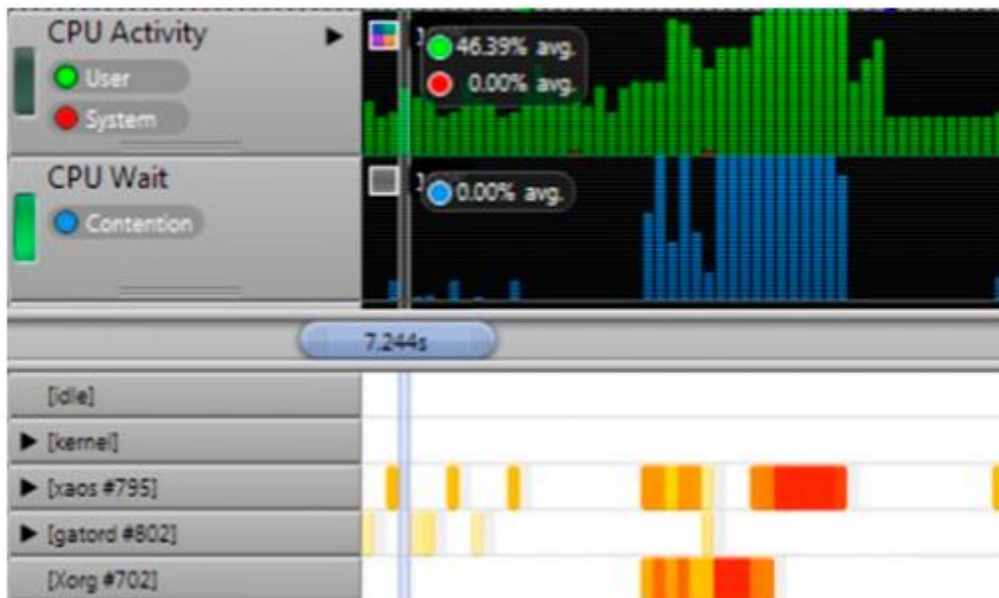


Na imagem abaixo o DS-5 pára de funcionar e apresenta os resultados analíticos obtidos:





Uma simples análise do Streamline





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