

BeagleBone Black

Algumas especificações

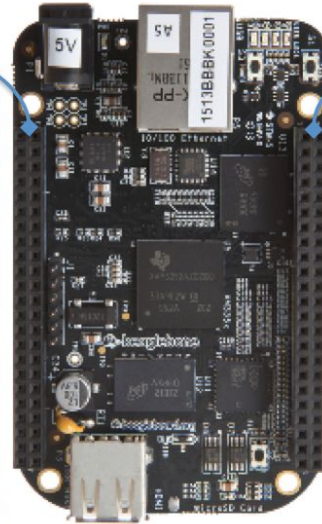
- 512MB DDR3 RAM
- Processor AM335x 1GHz ARM
- 4GB 8-bit eMMC on-board flash storage
- HDMI
- 3D graphics accelerator

Pinagem da BeagleBone Black

Cape Expansion Headers

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BTN	9	10	SYS_RESETN
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
SPI0_CS0	17	18	SPI0_D1
I2C2_SCL	19	20	I2C2_SDA
SPI0_DO	21	22	SPI0_SCLK
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	SPI1_CS0
SPI1_DO	29	30	GPIO_112
SPI1_SCLK	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWM0
DGND	43	44	DGND
DGND	45	46	DGND



P8

DGND	1	2	DGND
MMC1_DAT6	3	4	MMC1_DAT7
MMC1_DAT2	5	6	MMC1_DAT3
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	MMC1_CMD
MMC1_CLK	21	22	MMC1_DAT5
MMC1_DAT4	23	24	MMC1_DAT1
MMC1_DAT0	25	26	GPIO_61
LCD_VSYNC	27	28	LCD_PCLK
LCD_HSYNC	29	30	LCD_AC_BIAS
LCD_DATA14	31	32	LCD_DATA15
LCD_DATA13	33	34	LCD_DATA11
LCD_DATA12	35	36	LCD_DATA10
LCD_DATA8	37	38	LCD_DATA9
LCD_DATA6	39	40	LCD_DATA7
LCD_DATA4	41	42	LCD_DATA5
LCD_DATA2	43	44	LCD_DATA3
LCD_DATA0	45	46	LCD_DATA1

LEGEND	
POWER/GROUND/RESET	
AVAILABLE DIGITAL	
AVAILABLE PWM	
SHARED I2C BUS	
RECONFIGURABLE DIGITAL	
ANALOG INPUTS (1.8V)	

Instalação do SO

Como baixar e descompactar o debian

Passos (Comandos):

1 - sudo wget <https://rcn-ee.com/rootfs/2017-07-14/elinux/debian-9.0-console-armhf-2017-07-14.tar.xz>

2- tar xf debian-9.0-console-armhf-2017-07-14.tar.xz

3 - cd debian-9.0-console-armhf-2017-07-14

*Permanecer na pasta do sistema para realizar os próximos passos.

PS: A versão do linux, mostrada no pdf foi alterada.

PSS: O site <https://rcn-ee.com/rootfs> contém todas as versões disponíveis

Instalação do SO

Passando o sistema para o cartão SD.

Passos:

1 - Inserir o cartão SD no computador

2 - No terminal, listar os dispositivos de mídia e partições presentes no computador com o comando:

```
sudo ./setup_sdcard.sh --probe-mmc
```

2.1 Verificar o nome relativo ao cartão sd, na lista de nomes fornecida.

3 - Digitar o seguinte comando, substituindo 'mmcblk0' pelo nome do seu dispositivo:

```
sudo ./setup_sdcard.sh --mmc /dev/mmcblk0 --dtb beaglebone
```

OBS: Quando a operação de cópia acabar será exibida no terminal, a senha do sistema. No nosso caso: **temppwd**

4 - Após o término da cópia, acessar a pasta 'boot' do cartão sd, abrir o arquivo uEnv.txt e descomentar última linha.

```
Comando: sudo nano /media/nome_usuario/rootfs/boot/uEnv.txt
```

Exemplo de saída quando o comando do passo 2 é executado

rute@rute-pc: ~/debian-9.0-console-armhf-2017-07-14

lsblk:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sdb	8:16	1	7,3G	0	disk	
└─sdb1	8:17	1	7,3G	0	part	/media/rute/rootfs
sda	8:0	0	931,5G	0	disk	
├─sda4	8:4	0	750M	0	part	
├─sda2	8:2	0	40M	0	part	
├─sda9	8:9	0	7,9G	0	part	[SWAP]
├─sda7	8:7	0	8,6G	0	part	
├─sda5	8:5	0	840,1G	0	part	
├─sda3	8:3	0	128M	0	part	
├─sda1	8:1	0	500M	0	part	/boot/efi
├─sda8	8:8	0	73,2G	0	part	/
└─sda6	8:6	0	450M	0	part	

Instalação do SO

Como realizar a operação de flash?

Passos:

1 - Desconectar a beaglebone da alimentação (fonte ou usb)

2 - Inserir o cartão SD

3 - Segurar o botão de *boot*

4 - Ligar a beaglebone

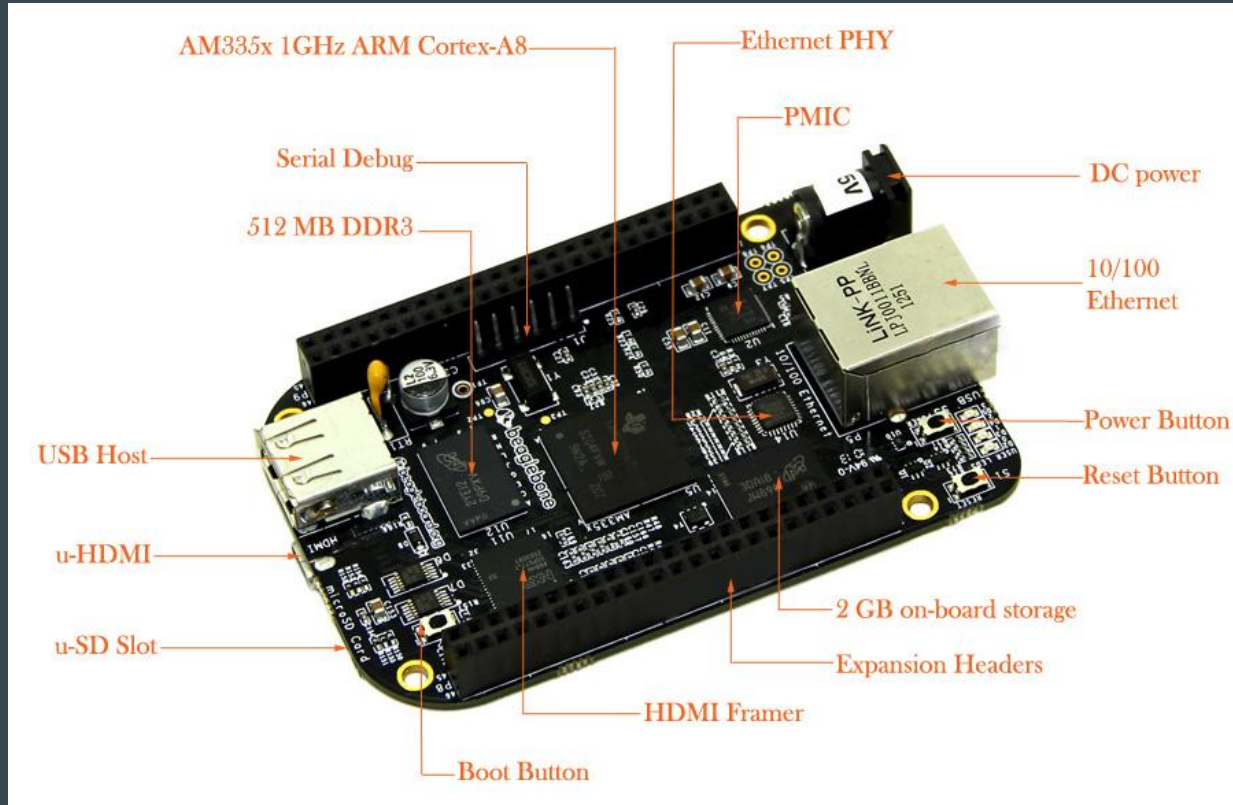
5 - Manter o botão de boot pressionado até que o seguinte padrão seja verificado:

- Os 4 leds acendem
- Os 4 leds apagam
- Os leds começam a piscar

Assim que o padrão for verificado, soltar o botão de boot.

Agora a beagle deverá produzir o seguinte padrão:

<https://youtu.be/o885-0BmEpo>



Instalação do SO

Como saber se acabou?

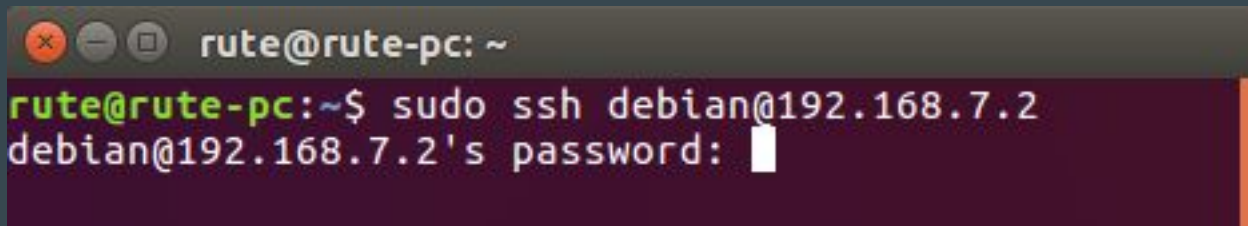
Ao final da operação de *flashing*, a placa irá desligar automaticamente. Você deve então, desconectar a alimentação e retirar o cartão SD. Após isso já pode ligar e usar a placa.

Acessando a beagle, utilizando ssh

No terminal digite:

```
sudo ssh debian@192.168.7.2
```

A senha do login ssh, será a senha disponibilizada ao final da cópia da imagem. Neste caso: 'temppwd'

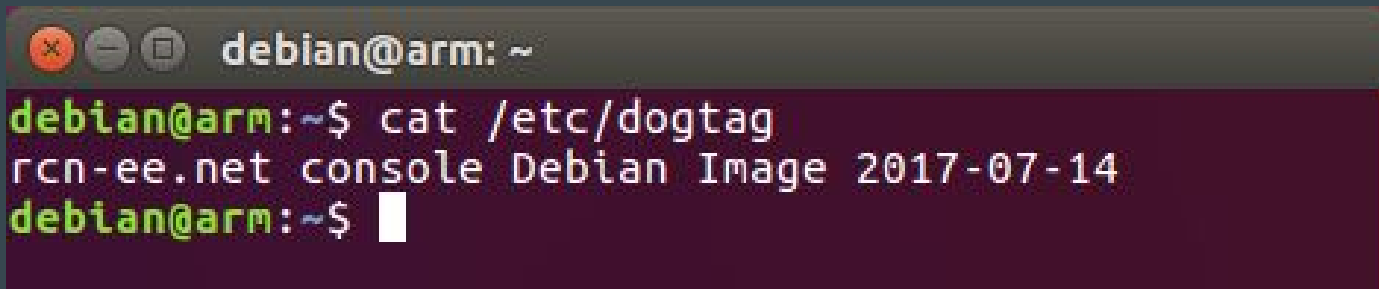
A screenshot of a terminal window with a dark background. The title bar at the top shows three window control icons (close, minimize, maximize) and the text 'rute@rute-pc: ~'. The terminal content shows a green prompt 'rute@rute-pc:~\$' followed by the command 'sudo ssh debian@192.168.7.2'. Below this, the text 'debian@192.168.7.2's password:' is displayed, followed by a white cursor block indicating where a password would be entered.

```
rute@rute-pc: ~  
rute@rute-pc:~$ sudo ssh debian@192.168.7.2  
debian@192.168.7.2's password: 
```

Verificando a instalação

Para verificar se sua instalação deu certo, verifique a data da imagem, utilizando o seguinte comando:

Se a data for a mesma data do *realease* da imagem escolhida, sua instalação terá sido feita corretamente.

A terminal window with a dark background and a title bar that reads "debian@arm: ~". The window contains the following text: the prompt "debian@arm:~\$" followed by the command "cat /etc/dogtag", the output "rcn-ee.net console Debian Image 2017-07-14", and the prompt "debian@arm:~\$" followed by a white cursor block.

```
debian@arm: ~  
debian@arm:~$ cat /etc/dogtag  
rcn-ee.net console Debian Image 2017-07-14  
debian@arm:~$
```

Conectando a placa à internet, via USB

Estando conectado na beagle via usb, digite os seguintes comandos:

```
debian@arm:~$ sudo ifconfig usb0 192.168.7.2
debian@arm:~$ sudo route add default gw 192.168.7.1
debian@arm:~$ sudo su
root@arm:/home/debian# echo "nameserver 8.8.8.8" >> /etc/resolv.conf
```

OBS: Esse comando precisa ser digitado sempre que a beagle for acessada, para que o compartilhamento de internet seja possível.

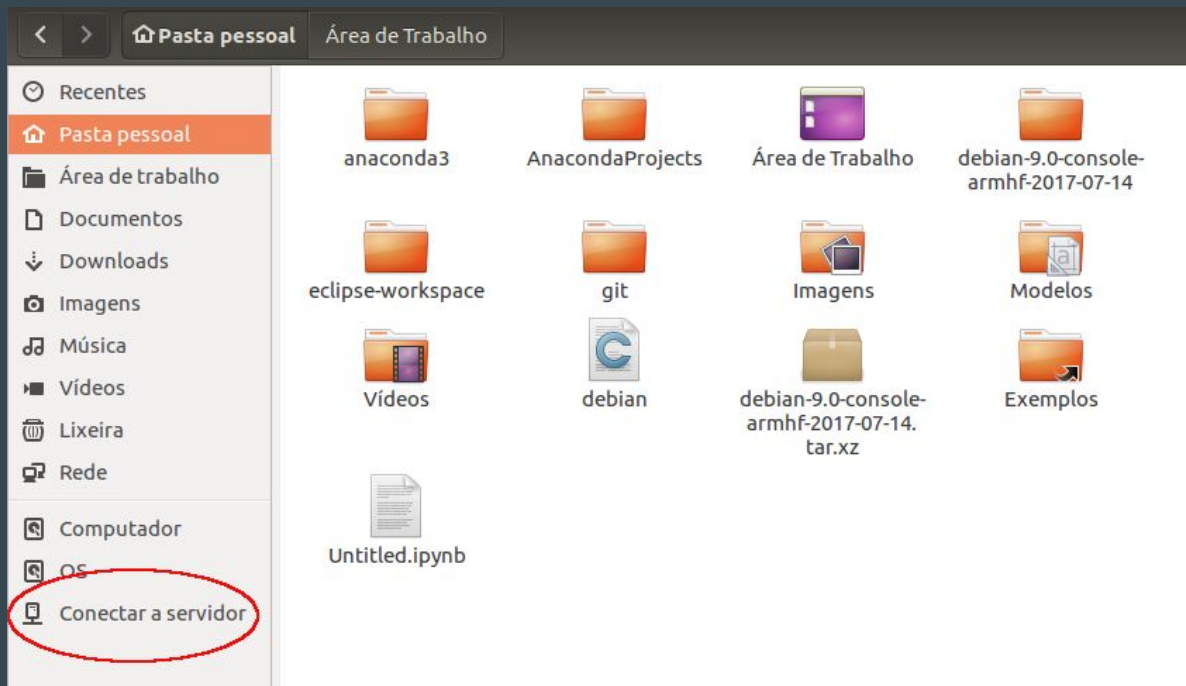
No seu computador, digite os seguintes comandos, substituindo wlan0 pelo nome da sua placa de rede e eth5 pelo nome da placa de rede da beagle:

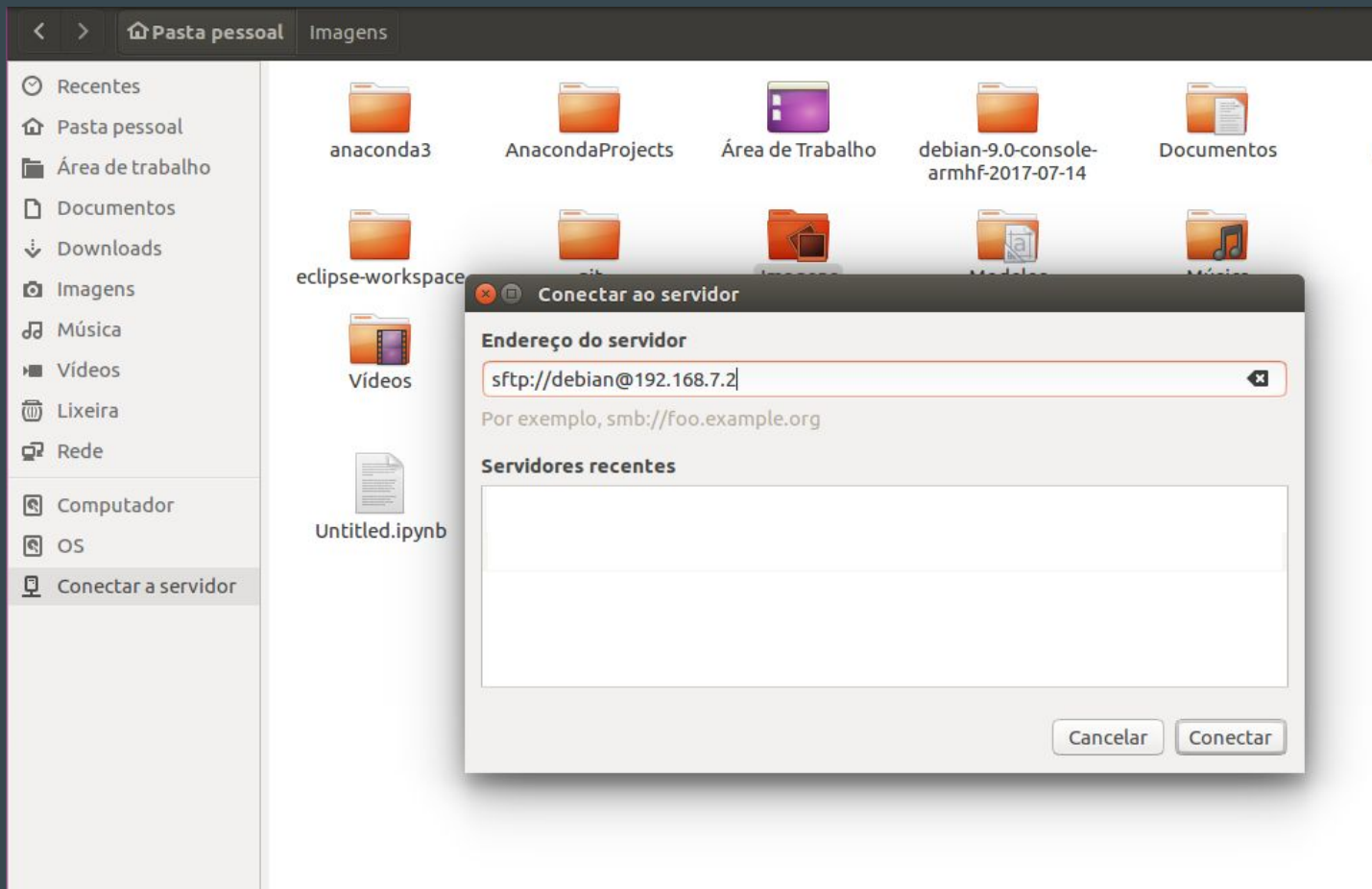
```
sudo su
#wlan0 is my internet facing interface, eth5 is the BeagleBone USB connection
ifconfig eth5 192.168.7.1
iptables --table nat --append POSTROUTING --out-interface wlan0 -j MASQUERADE
iptables --append FORWARD --in-interface eth5 -j ACCEPT
echo 1 > /proc/sys/net/ipv4/ip_forward
```

OBS: Veja no seu PC, os nomes da sua placa e da placa de rede da beagle, com o comando: ifconfig.

A placa da beagle que você deve escolher é a relativa ao ip : 192.168.7.1

Acessando o sistema de arquivos da beagle, utilizando o nautilus





Instalando a BlackLib

Para instalar a BlackLib na Beagle, você deve baixar a biblioteca, no endereço:

<https://github.com/tfmiranda/BlackLib-Modificada/archive/master.zip> e colocá-la dentro da Beaglebone Black.

Para fazer isso, siga os seguintes passos:

1 - Conecte-se via ssh a placa.

2 - Digite o seguinte comando:

```
sudo wget https://github.com/tfmiranda/BlackLib-Modificada/archive/master.zip
```

3 - Atualize os repositórios e instale o unzip:

```
sudo apt-get update
```

```
sudo apt-get install unzip
```

4 - Descompacte a biblioteca:

```
unzip master.zip
```

5 - Se quiser, remova o arquivo zip

```
sudo rm master.zip
```

Além das classes da BlackLib original a modificação tem uma classe a mais chamada ADC, ela é responsável por oferecer uma interface simples e prática para ler os valores analógicos. Entretanto, para utilizá-la é necessário executar os seguintes comandos na BeagleBone Black.

Para baixar esta classe , execute os seguintes comandos, no sistema:

```
git clone https://github.com/beagleboard/bb.org-overlays
```

```
cd ./bb.org-overlays
```

```
./dtc-overlay.sh
```

```
./install.sh
```

Antes de utilizar a BlackLib é necessário sobrescrever o kernel do Debian 8.3 (kernel 4.1.x) para um antigo que esteja apto com as antigas versões que eram compatíveis com a biblioteca. Para isto execute os seguinte comandos na BeagleBone Black:

```
sudo su
git clone https://github.com/tfmiranda/ARM-4.1.x-Overlay.git
cd ./ARM-4.1.x-Overlay
sh dtc-overlay.sh
sh install.sh
```

PS: Depois de executar esses passos, desligue e ligue novamente a placa.

Para verificar se a biblioteca foi instalada corretamente, execute os seguintes comandos:

```
cd BlackLib-Modificada-master
make
```

Após executar o comando make, o executável para o arquivo main, dentro da lib, é criado.

Para testar se deu tudo certo com a biblioteca, execute-o com o seguinte comando:

```
./main
```

O seguinte ‘cout’ é exibido: ‘Este é o seu main, implemente-o’

Você pode substituir este arquivo de main, ‘main.cpp’, pelo seu código ou escrever neste arquivo. Caso deseje adicionar mais arquivos cpp, além do main, crie o arquivo na pasta da biblioteca, e **adicione-o na lista de SOURCES dentro do arquivo makefile.**

CXX=g++

CPPFLAGS=-D__GXX_EXPERIMENTAL_CXX0X__ -D__GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 -
D__GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 -D__GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 -
D__GCC_HAVE_SYNC_COMPARE_AND_SWAP_8

CXXFLAGS=-std=c++0x -O0 -g3 -Wall -c -fmessage-length=0 -pthread \$(CPPFLAGS)

LDFLAGS=-lpthread

RM=rm -f

SOURCES=./ADC/Adc.cpp ./BlackGPIO/BlackGPIO.cpp ./BlackADC/BlackADC.cpp ./BlackDirectory/
BlackDirectory.cpp ./BlackI2C/BlackI2C.cpp BlackMutex/BlackMutex.cpp BlackPWM/BlackPWM.cpp
BlackSPI/BlackSPI.cpp BlackThread/BlackThread.cpp BlackTime/BlackTime.cpp BlackUART/
BlackUART.cpp ./BlackCore.cpp ./main.cpp

OBJECTS=\$(SOURCES:.cpp=.o)

EXECUTABLE=main

all: \$(SOURCES) \$(EXECUTABLE)

\$(EXECUTABLE): \$(OBJECTS)
\$(CXX) \$(OBJECTS) \$(LDFLAGS) -o \$@

.cpp.o:
\$(CXX) \$(CXXFLAGS) \$< -o \$@

clean:
\$(RM) \$(OBJECTS) \$(EXECUTABLE)

Programando pinos de I/O com BlackLib

```
#include <iostream>
#include "BlackGPIO/BlackGPIO.h"
|
using namespace BlackLib;

int main(int argc, char * argv[])
{
    std::cout << "Example 1 - GPIO (in/out)" << endl;
    BlackGPIO saida(GPIO_14,output);
    BlackGPIO entrada(GPIO_67,input);

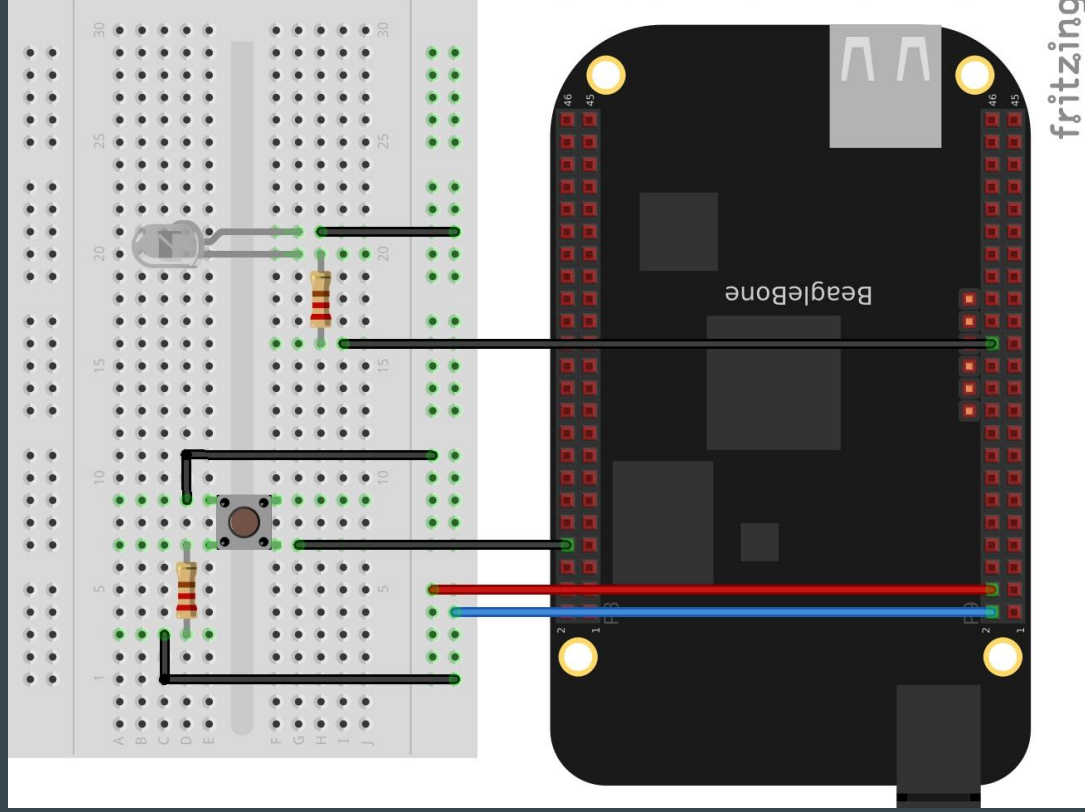
    while(true){
        std::string val = entrada.getValue();
        if(val=="1"){
            saida.setValue(high);
            std::cout << "i'm high"<<endl;
        }else{
            saida.setValue(low);
            std::cout << "i'm low"<<endl;
        }
    }
    return 0;
}
```

Como compilar e executar

Utilize o comando, 'make', para compilar seu código

Utilize o comando, 'make clean', para limpar todos os *.o criados

Execute o programa com o comando: ./main.



Páginas de suporte

<http://beagleboard.org/Support/bone101>

http://blacklib.yigityuce.com/v1_0/DOCUMENTS/EN/BlackLib-UserManual.pdf

OBS: Alguns pinos não estão mapeados na BlackLib-Modificada. Para ter certeza que seu pino está mapeado na lib, verifique no arquivo de classe da função que você deseja utilizar. Por exemplo, se precisar verificar um pino de GPIO, verifique o arquivo BlackGPIO.h dentro da pasta da biblioteca.