

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

SISTEMAS DE TEMPO REAL – DCA0125

Luiz Affonso H. Guedes

Tiago Fernandes de Miranda

Roteiro de uso da BlackLib-Modificada

A seguir será demonstrado o procedimento básico para gerenciar os pinos digitais da Beagle Bone Black com o auxílio da biblioteca BlackLib-Modificada. Antes de tudo, supõe-se que os seguintes requisitos foram satisfeitos:

- O cartão SD foi formatado e preenchido com o SO atualizado;
- O acesso via SSH foi realizado com sucesso utilizando o USB;
- O roteiro de configuração de ambiente de desenvolvimento foi seguido e o Eclipse Cpp está instalado e acessando a beagle;

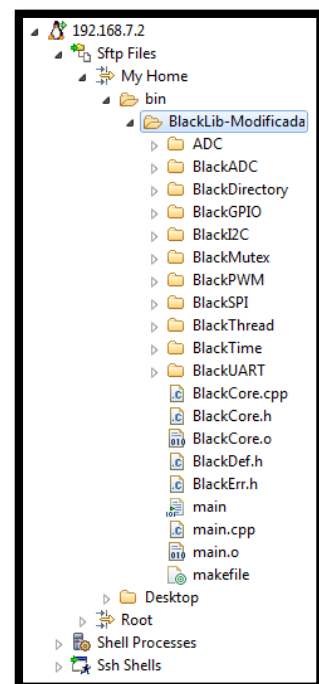
A versão da BlackLib-Modificada pode ser obtida através do link a seguir, baixem e copiem o arquivo descompactado para a beagle.

<https://github.com/tfmiranda/BlackLib-Modificada>

Feito isto, já é possível fazer um teste inicial. Através do prompt de comando acesse o diretório e execute o arquivo *make*.

```
root@beaglebone:/home/debian/bin/BlackLib-Modificada# make
```

Este arquivo irá compilar cada módulo (pasta) da biblioteca e por último irá compilar o código *main.cpp*. Ao finalizar a compilação será criado o arquivo *main*, basta executá-lo *./main*. Se ao executar exibir a mensagem: *Este e o seu main, implemente-o*, significa que tudo ocorreu como esperado e basta fazer a devida implementação no código.



Utilizar os pinos digitais com auxílio da biblioteca se torna um processo simples, a seguir um código demonstrando o seu uso:

```
#include <iostream>
#include <pthread.h>
#include <math.h>
#include <unistd.h>
#include "BlackGPIO/BlackGPIO.h"
#include "ADC/Adc.h"

#define UNIT_MS 1000;
#define UNIT_SEC 1000000;

using namespace std;

int main(int argc, char * argv[])
{
    cout << "Example - GPIO" << endl;
    BlackLib::BlackGPIO led(BlackLib::GPIO_51,BlackLib::output, BlackLib::SecureMode);

    while(true){
        long tempo = 3*UNIT_SEC;
        led.setValue(BlackLib::high);
        usleep(tempo);
        led.setValue(BlackLib::low);
        usleep(tempo);
    }
    return 0;
}
```

Neste trecho, uma instancia do pino 51 é iniciada como saída e então é feito um *loop* infinito onde é feita a escrita de nível HIGH e LOW periodicamente. Façam o teste, lembrem de fazer o circuito com LED e resistor no pino 51 (ver mapa de pinos) e compilar como *superuse*.

No exemplo a seguir, é demonstrado como é feita a leitura digital.

```
#include <iostream>
#include <pthread.h>
#include <math.h>
#include <unistd.h>
#include "BlackGPIO/BlackGPIO.h"
#include "ADC/Adc.h"

#define UNIT_MS 1000;
#define UNIT_SEC 1000000;

using namespace std;

int main(int argc, char * argv[])
{
    cout << "Example - GPIO" << endl;
    long tempo = 1*UNIT_SEC;
    BlackLib::BlackGPIO led(BlackLib::GPIO_51,BlackLib::input, BlackLib::SecureMode);

    while(true){
        cout << "valor: " << led.getValue() << endl;
        usleep(tempo);
    }
    return 0;
}
```