

# BeagleBone Black

---

Utilizando as funções ADC e PWM

Rute Abreu

# Função ADC

---

# Que classe deve ser utilizada?

Classe ADC, localizada dentro da BlackLib modificada.

Não confundir com BlackADC.

Voltagem máxima: 1.8V

Pino de alimentação e terra: VDD\_ADC (P9\_32), GND\_ADC (P9\_34)

Os pinos analógicos devem ser referenciados no construtor utilizando a numeração de 1 a 6.

## Pinos disponíveis

# 65 possible digital I/Os

- AIN0
- AIN1
- AIN2
- AIN3
- AIN4
- AIN5
- AIN6

P9				P8			
DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	GPIO_38	3	4	GPIO_39
VDD_5V	5	6	VDD_5V	GPIO_34	5	6	GPIO_35
SYS_5V	7	8	SYS_5V	GPIO_66	7	8	GPIO_67
PWR_BTN	9	10	SYS_RESETN	GPIO_69	9	10	GPIO_68
GPIO_30	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
GPIO_31	13	14	GPIO_50	GPIO_23	13	14	GPIO_26
GPIO_48	15	16	GPIO_51	GPIO_47	15	16	GPIO_46
GPIO_5	17	18	GPIO_4	GPIO_27	17	18	GPIO_65
I2C2_SCL	19	20	I2C2_SDA	GPIO_22	19	20	GPIO_63
GPIO_3	21	22	GPIO_2	GPIO_62	21	22	GPIO_37
GPIO_49	23	24	GPIO_15	GPIO_36	23	24	GPIO_33
GPIO_117	25	26	GPIO_14	GPIO_32	25	26	GPIO_61
GPIO_115	27	28	GPIO_113	GPIO_86	27	28	GPIO_88
GPIO_111	29	30	GPIO_112	GPIO_87	29	30	GPIO_89
GPIO_110	31	32	VDD_ADC	GPIO_10	31	32	GPIO_11
AIN4	33	34	GNDA_ADC	GPIO_9	33	34	GPIO_81
AIN6	35	36	AIN5	GPIO_8	35	36	GPIO_80
AIN2	37	38	AIN3	GPIO_78	37	38	GPIO_79
AIN0	39	40	AIN1	GPIO_76	39	40	GPIO_77
GPIO_20	41	42	GPIO_7	GPIO_74	41	42	GPIO_75
DGND	43	44	DGND	GPIO_72	43	44	GPIO_73
DGND	45	46	DGND	GPIO_70	45	46	GPIO_71

# Funções disponíveis na classe

`getIntValue() : int`

`getFloatValue() : float`

`getPercentValue() : float`

PWM

---

# Que classe utilizar

Classe PWM, localizada dentro da BlackLib modificada, atualizada.

Não confundir com BlackPWM.

Métodos:

`setPeriod(int period): void`

`getPeriod() : int`

`setDutyCycle(int dutyCycle) : void`

`getDutyCycle(): int`

`setState(statePwm state): void (run, stop)`

`getState(): statePwm`

# Pinos Mapeados

EHRPWM2B = P8\_13

EHRPWM2A = P8\_19

EHRPWM1A = P9\_14

EHRPWM1B = P9\_16

EHRPWM0B = P9\_21

EHRPWM0A = P9\_22

## 8 PWMs and 4 timers

P9				P8			
DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	GPIO_38	3	4	GPIO_39
VDD_5V	5	6	VDD_5V	GPIO_34	5	6	GPIO_35
SYS_5V	7	8	SYS_5V	TIMER4	7	8	TIMER7
PWR_BTN	9	10	SYS_RESETN	TIMER5	9	10	TIMER6
GPIO_30	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
GPIO_31	13	14	EHRPWM1A	EHRPWM2B	13	14	GPIO_26
GPIO_48	15	16	EHRPWM1B	GPIO_47	15	16	GPIO_46
GPIO_5	17	18	GPIO_4	GPIO_27	17	18	GPIO_65
I2C2_SCL	19	20	I2C2_SDA	EHRPWM2A	19	20	GPIO_63
EHRPWM0B	21	22	EHRPWM0A	GPIO_62	21	22	GPIO_37
GPIO_49	23	24	GPIO_15	GPIO_36	23	24	GPIO_33
GPIO_117	25	26	GPIO_14	GPIO_32	25	26	GPIO_61
GPIO_115	27	28	ECAPPWM2	GPIO_86	27	28	GPIO_88
EHRPWM0B	29	30	GPIO_112	GPIO_87	29	30	GPIO_89
EHRPWM0A	31	32	VDD_ADC	GPIO_10	31	32	GPIO_11
AIN4	33	34	GND_ADC	GPIO_9	33	34	EHRPWM1B
AIN6	35	36	AIN5	GPIO_8	35	36	EHRPWM1A
AIN2	37	38	AIN3	GPIO_78	37	38	GPIO_79
AIN0	39	40	AIN1	GPIO_76	39	40	GPIO_77
GPIO_20	41	42	ECAPPWM0	GPIO_74	41	42	GPIO_75
DGND	43	44	DGND	GPIO_72	43	44	GPIO_73
DGND	45	46	DGND	EHRPWM2A	45	46	EHRPWM2B



# Exemplo de utilização

```
#include <iostream>
#include "PWM/PWM.h"
#include "ADC/Adc.h"
#include "unistd.h"

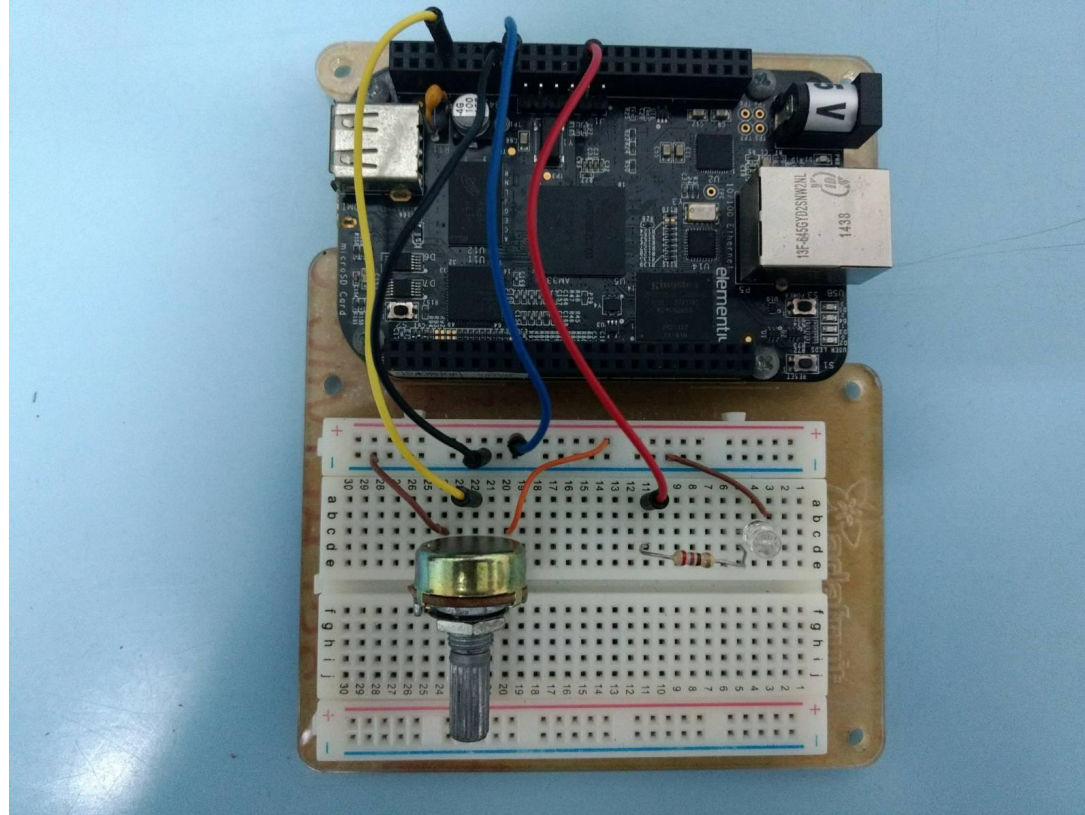
int main(int argc, char * argv[]){
    ADC adc(AINx::AIN0);
    PWM pwm(P9_22);
    pwm.setState(run);
    int period 0;

    while(true){
        float dutyCycle = adc.getPercentValue();
        cout << "Duty Value " << dutyCycle << endl;
        period = 1000000000;
        pwm.setPeriod(period);
        pwm.setDutyCycle(dutyCycle*period/100.0);

    }

    return 0;
}
```

# Circuito PWM + ADC



# Informações úteis

Desligar placa pela linha de comando : `sudo shutdown -h now`

Desligar a placa manualmente: Apertar o botão de power.

**Não** utilizar os pinos de voltagem 5V.

Utilizar pinos de voltagem 3.3V (Pinos P9\_3 e P9\_4)

Quando utilizar o ADC, utilizar pinos de voltagem e terra próprios (1,8V) :

VDD\_ADC: pino P9\_32.

GND\_ADC: pino P9\_34.

# Páginas de soporte

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

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