

```
import beaglebone_pru_adc as adc #Importa a biblioteca como o objeto adc
numsamples = 1000000 #
capture = adc.Capture() #cria o objeto Capture, filho de ADC

#capture.oscilloscope_init(No do registrador do ADC, No de amostras)
capture.oscilloscope_init(adc.OFF_VALUES, numsamples) #Método de capture para configurar o
modo osciloscópio
#adc.OFF_VALUES <- Atributo de adc que representa o endereço de memória do ADC0
capture.start() #Método de capture que inicia a captura
while True:
    if capture.oscilloscope_is_complete(): #capture.oscilloscope_is_complete() <- Método
    para verificar se o todas as amostras foram capturadas
        break
capture.stop() #Método para parar a captura de dados e finalizar o arquivo do PRU
capture.wait() #Método que espera o programa do PRU ser finalizado
print capture.oscilloscope_data(numsamples) #Método que retorna uma tupla com as amostras
salvas na memória RAM
Capture.close() #Método para liberar os dados da RAM para o sistema operacional

#Outros atributos de capture
Capture.ema_pow #Se for diferente de 0, aplica um filtro EMA (Exponential moving average),
cuja fórmula é  $ema += (value - ema / 2^{ema\_pow})$ , por padrão é desativado
Capture.cap_delay #Atributo que define a taxa de amostragem. Valor padrão é 0, ou seja. É
aplicado um loop no código assemble do PRU
```