```cpp
/* ****************************************************************** */
/* File name:        DroneTimer.cpp                                   */
/* File description: Timer interface implementation file              */
/* Author name:      Giacomo Dollevedo, Gustavo Fernandes             */
/* Creation date:    18nov2020                                        */
/* Revision date:    18dec2020                                        */
/* ****************************************************************** */

#include "DroneTimer.h"

/* ************************************************************************************ */
/* Method's name:          initTimer                                                     */
/* Description:            Initialize hardware timer interruptions                       */
/*                                                                                        */
/* Entry parameters:       int frequency -> timer interrupt frequency                    */
/*                         void (*fn)(void) -> function pointer that will be exectued    */
/*                                                                                        */
/* Return parameters:      n/a                                                            */
/* ************************************************************************************ */
void DroneTimer::initTimer(int frequency, void (*fn)(void))
{
  timerFreq = frequency;
  /*timerBegin(NumeroDoTimer, PreScaler, UpCount/DownCount*/
  /*80 -> clock do timer eh 80 MHz. Entao 80.000.000/80 = 1.000.000s = 1 microssegundo*/
  timer0 = timerBegin(0, 80, true);

  /*timerAttachInterrupt(ObjetoTimer, FuncaoDisparada, TipoDaInterrupcao)*/
    timerAttachInterrupt(timer0, fn, true);
  /*timerAlarmWrite(ObjetoTimer, periodoInterrupcao(microssegundos) , RepetirContagem)*/
    timerAlarmWrite(timer0, 1000000/frequency, true);
    Serial.println("Timer Interrupt Attached!");
}

/* ************************************************************************************ */
/* Method's name:          enableTimer                                                    */
/* Description:            Enable timer interruptions.                          */
/*                                                                                        */
/* Entry parameters:       n/a                                */
/*                                                                                        */
/* Return parameters:      n/a                                                     */
/* ************************************************************************************ */
void DroneTimer::enableTimer()
{
  timerAlarmEnable(timer0);
}

/* ************************************************************************************ */
/* Method's name:          disableTimer                                            */
/* Description:            Disable timer interruptions.                          */
/*                                                                                 */
/* Entry parameters:       n/a                          */
/*                                                                                 */
/* Return parameters:      n/a                                                     */
/* ************************************************************************************ */
void DroneTimer::disableTimer()
{
  timerAlarmDisable(timer0);
}




/*NAO USAMOS PRA NADA*/
/* ************************************************************************************ */
/* Method's name:          setFrequency                                            */
/* Description:            Set the timer frequency                             */
/*                                                                                 */
/* Entry parameters:       int frequency -> frequency to be set             */
/*                                                                                 */
/* Return parameters:      n/a                                                     */
/* ************************************************************************************ */
void DroneTimer::setFrequency(int frequency)
{
  timerFreq = frequency;
}

/*NAO USAMOS PRA NADA*/
/* ************************************************************************************
```

```
*/
/* Method's name:          getFrequency
  */
/* Description:            Get the timer frequency                            */
/*
   */
/* Entry parameters:       n/a                                */
/*
   */
/* Return parameters:      int -> Timer frequency
  */
/*
*************************************************************************************
*/
int DroneTimer::getFrequency()
{
  return timerFreq;
}

```