```
/*
 *     SocialLedge.com - Copyright (C) 2013
 *
 *     This file is part of free software framework for embedded processors.
 *     You can use it and/or distribute it as long as this copyright header
 *     remains unmodified.  The code is free for personal use and requires
 *     permission to use in a commercial product.
 *
 *      THIS SOFTWARE IS PROVIDED "AS IS".  NO WARRANTIES, WHETHER EXPRESS,
  IMPLIED
 *      OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF
 *      MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS
  SOFTWARE.
 *      I SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL,
  OR
 *      CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
 *
 *     You can reach the author of this software at :
 *          p r e e t . w i k i @ g m a i l . c o m
 */

/**
 * @file
 * @brief I2C2 Interrupt driven IO driver
 *
 * 20131010 : Removed read vs. write mode variable
 */
#ifndef i2c2hpp_
#define i2c2hpp_

#include "singleton_template.hpp"
#include "i2c_base.hpp"



/**
 * I2C2 Singleton Driver
 * This is a thin wrapper around I2C_Base class and merely gives the base
 * address of the I2C2 memory map.
 *
 * @ingroup Drivers
 */
class I2C2 : public I2C_Base, public SingletonTemplate<I2C2>
{
    public:
        /// Initializes I2C2 at the given @param speedInKhz
        bool init(unsigned int speedInKhz);
        /// Initializes I2C2 at the given @param speedInKhz but as a Slave
        bool initSlave(const uint8_t slaveAddr, volatile uint8_t * buffer,
         uint32_t buffer_size);
    private:
        I2C2(); ///< Private constructor for this singleton class
        friend class SingletonTemplate<I2C2>;  ///< Friend class used for
         Singleton Template
};


/**
 * I2C Addresses for on-board devices attached to I2C Bus
 */
enum Board_I2C_Device_Addresses{
    I2CAddr_AccelerationSensor = 0x38,
    I2CAddr_TemperatureSensor  = 0x90, /* AD0 pin tied to Gnd */
    I2CAddr_LED_Display        = 0x40,
};



#endif /* i2c2.hpp_ */
```