```
// ************************************** clock.c ************************************
// Project: Keil Labs and Project
// File: clock.c
// Class: ENEL 351 Lab Works
// Programmer: Amandip Padda
// SID: 200455829
// Modified from : Dave Duguid (https://uregina.ca/~duguidda/)
// Description: The project is based on the STM32F103RB that is being used in ENEL 351 Labs.
// It will also be used in the Project related to input and output of various sensors.

// ===========================================================================
// ***************************************IMPORTANT*********************************
//                          Copyright (c) 2023. All rights reserved.
//                          See the bottom of this file for the license terms.
// ===========================================================================

// ************************************ Header Files **********************************
#include "stm32f10x.h"
#include "header.h"

// ********************************** Clock Configuration *****************************
void clockInit(void)
{
// ************************* Enable HSI and wait for it to be ready ***************************
        RCC->CR |= RCC_CR_HSION;
  while (((RCC->CR) & (RCC_CR_HSION | RCC_CR_HSIRDY)) == 0);

// ******************* Enable HSE with Bypass and wait for it to be ready **********************
        RCC->CR |= RCC_CR_HSEON | RCC_CR_HSEBYP;
```

```
  while (((RCC->CR) & (RCC_CR_HSEON | RCC_CR_HSEBYP | RCC_CR_HSERDY)) == 0);

// ******************* SET HSI as SYSCLK and wait for it to be recognized ***********************
        // RCC->CFGR = RCC_CFGR_SW_HSI; //
        // while (((RCC->CFGR) & (RCC_CFGR_SW_HSI | RCC_CFGR_SWS_HSI))!=0); //

// ******************** SET HSE as SYSCLK and wait for it to be recognized **********************
        RCC->CFGR = RCC_CFGR_SW_HSE;
  while (((RCC->CFGR) & (RCC_CFGR_SW_HSE | RCC_CFGR_SWS_HSE)) == 0);

// ********************************** To Use PLL as SYSCLK ******************************
        RCC->CR &= ~RCC_CR_PLLON;

// ************************* Change PLL source and set the PLL multiplier **********************

// These are the SYSCLK values when using the PLL with HSI/2 as the input. The max value is 64 MHz
        // RCC->CFGR = 0x00000000;// 8 MHz
        // RCC->CFGR = 0x00040000;// 12 MHz
        // RCC->CFGR = 0x00080000;// 16 MHz
        // RCC->CFGR = 0x000c0000;// 20 MHz
        // RCC->CFGR = 0x00100000;// 24 MHz
        // RCC->CFGR = 0x00140000;// 28 MHz
        // RCC->CFGR = 0x00180000;// 32 MHz
        // RCC->CFGR = 0x001C0000;// 36 MHz
        // RCC->CFGR = 0x00200000;// 40 MHz
        // RCC->CFGR = 0x00280000;// 48 MHz
        // RCC->CFGR = 0x002C0000;// 52 MHz
        // RCC->CFGR = 0x003C0000;// 64 MHz
```

```
// These are the SYSCLK values when using the PLL with HSE/Bypass as the input. The max value is 72
// MHz
        // RCC->CFGR = 0x00010000;// 16 MHz
        // RCC->CFGR = 0x00050000;// 24 MHz
        // RCC->CFGR = 0x00090000;// 32 MHz
        // RCC->CFGR = 0x000d0000;// 40 MHz
        // RCC->CFGR = 0x00110000;// 48 MHz
        // RCC->CFGR = 0x00150000;// 56 MHz
        // RCC->CFGR = 0x00190000;// 64 MHz
         RCC->CFGR = 0x001d0000;// 72 MHz

// ****************************** ENABLE PLL and wait for it to be ready ********************
        RCC->CR |= RCC_CR_PLLON;
        while(((RCC->CR) & (RCC_CR_PLLON | RCC_CR_PLLRDY)) == 0);

// ***************************** Set PLL as SYSCLK and wait for it to be ready ******************
        RCC->CFGR &= ~(RCC_CFGR_SW);
        RCC->CFGR |= RCC_CFGR_SW_PLL; // 0x00000002;
        while ((RCC->CFGR & RCC_CFGR_SWS) != RCC_CFGR_SWS_PLL) {}

// Enable clock visibity on PA8 and select desired source (By default, the output will be the PLL frequency
// divided by 2)
        // RCC->CFGR |= RCC_CFGR_MCO_HSI;
        // RCC->CFGR |= RCC_CFGR_MCO_HSE;
        // RCC->CFGR |= RCC_CFGR_MCO_SYSCLK;
        RCC->CFGR |= RCC_CFGR_MCO_PLL;
        RCC->CFGR &= ~(RCC_CFGR_PLLSRC | RCC_CFGR_PLLMULL);
  RCC->CFGR |= (RCC_CFGR_PLLSRC_HSE | RCC_CFGR_PLLMULL9);
        RCC->CR |= RCC_CR_HSEON;
```

```
  while (!(RCC->CR & RCC_CR_HSERDY)) {}
}

// ================================================================================
// ************************************IMPORTANT*********************************
//                                Copyright (c) 2016. All rights reserved.
// This library is free software; you can redistribute it and/or modify it under the terms of the
// GNU Lesser General Public License as published by the Free Software Foundation; either version
// 2.1 of the License, or (at your option) any later version. This library is distributed in the
// hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU Lesser General Public
// License
// for more details. You should have received a copy of the GNU Lesser General Public License along
// with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street,
// Fifth Floor, Boston, MA  02110-1301  USA
// ================================================================================
```