

PHẦN LÝ THUYẾT

===//===

BÀI 2- NGÔN NGỮ C VÀ THƯ VIỆN CHUẨN CHO STM8

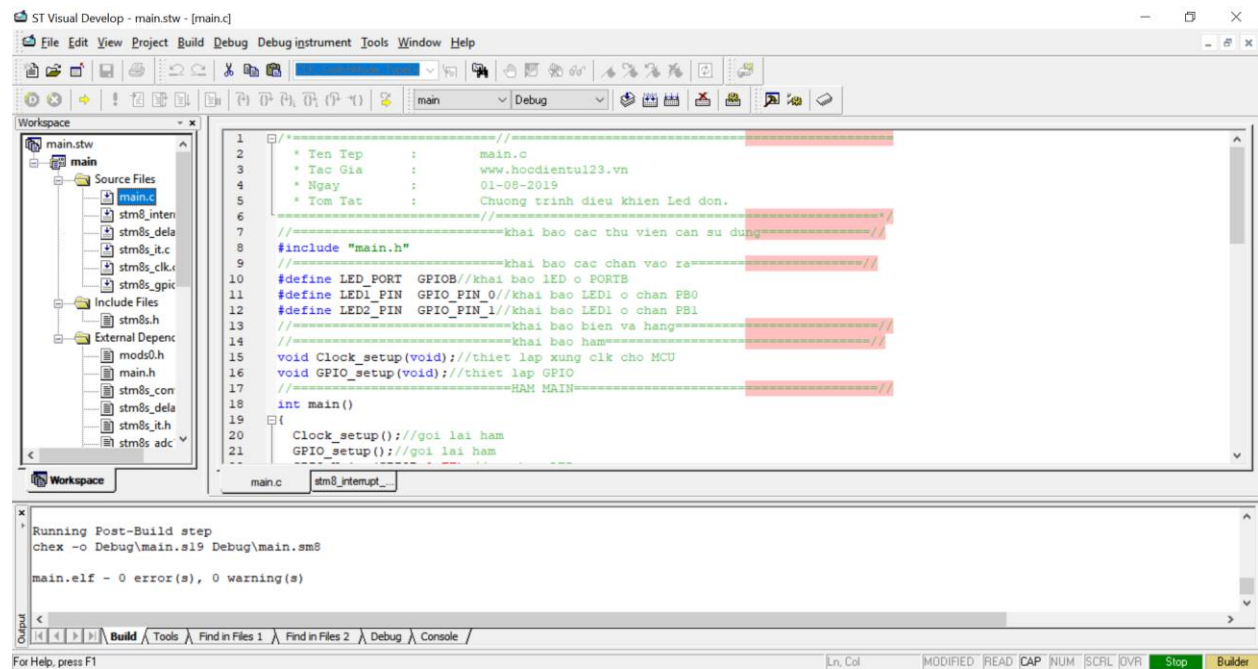
NỘI DUNG:

- 1- Phần mềm viết code và nạp code.
- 2- Ngôn ngữ C cho STM8.
- 3- Cấu trúc chương trình C trong ST Visual Develop
- 4- Giới thiệu thư viện cho STM8.
- 5- Lưu đồ giải thuật.

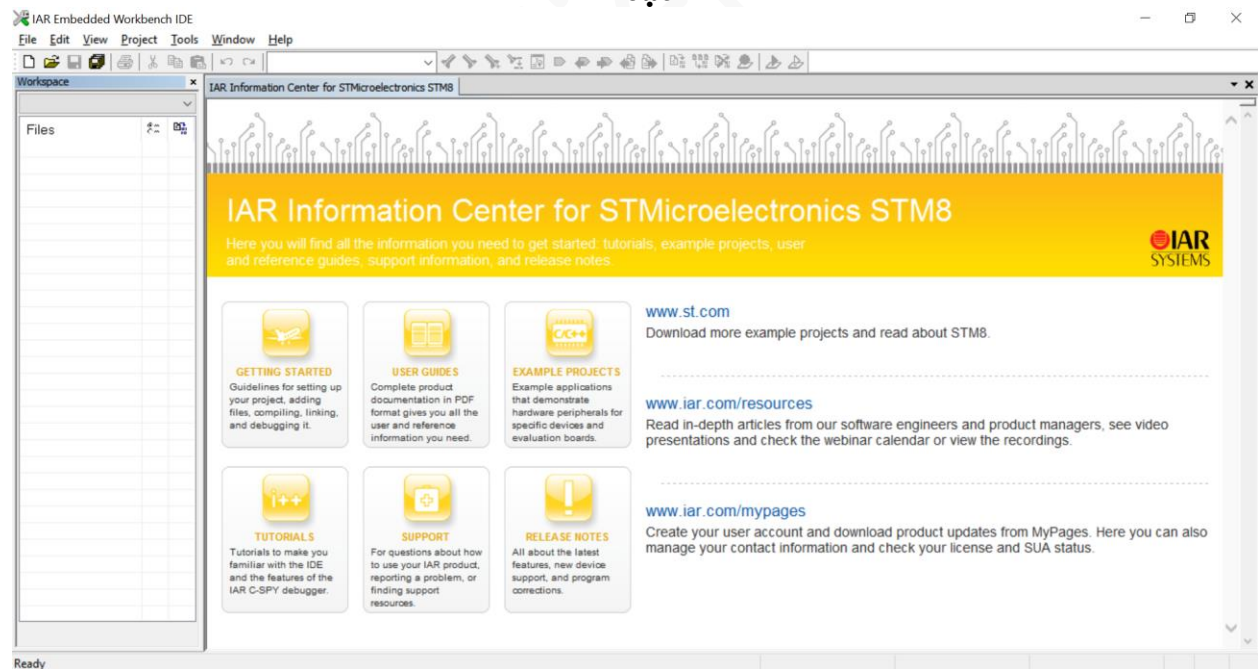
Chú ý: các bạn phải có kiến thức cơ bản về lập trình C/C++, các bạn có thể xem các bài học về C/C++ ở Phần Lập Trình C cơ bản (các bài học hoàn toàn miễn phí).

1- Phần mềm viết code và nạp code

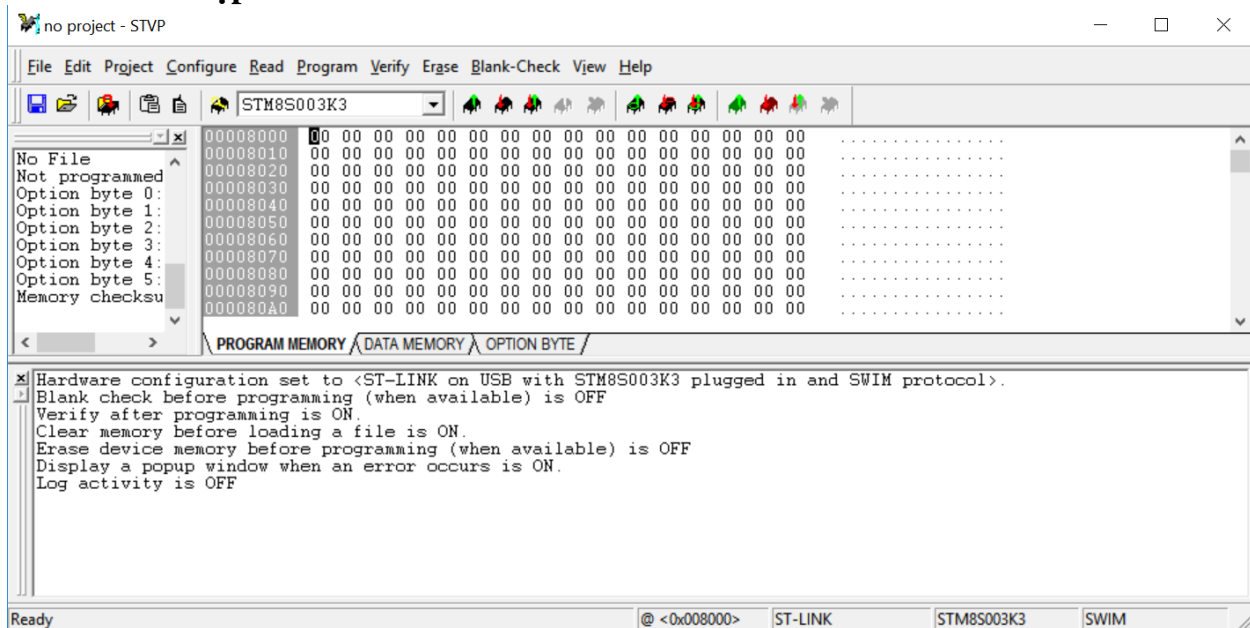
- Phần mềm viết code:



Hoặc



- Phần mềm nạp code:



2. Ngôn ngữ C cho STM8

Sử dụng ngôn ngữ C/C++ để lập trình cho STM8 trong phần mềm ST Visual Develop như:

- Khai báo các biến/hằng: **Ví dụ:** unsigned char dem1=0;

Tên kiểu dữ liệu (Data type)	Số byte	Khoảng dữ liệu (Range)
char	1	-127 to 127 or 0 to 255
unsigned char	1	0 to 255
signed char	1	-127 to 127
int	2	-32,767 to 32,767
unsigned int	2	0 to 65,535
signed int	2	Như kiểu int
short int	2	Như kiểu int
unsigned short int	2	0 to 65,535
signed short int	2	Như kiểu short int
long int	4	-2,147,483,647 to 2,147,483,647
signed long int	4	Như kiểu long int
unsigned long int	4	0 to 4,294,967,295

- Sử dụng các hệ số đếm:

- + Hệ nhị phân (là biểu diễn bằng 0 hoặc 1)
- + Hệ thập phân (là biểu diễn giống số thực hay thường gọi là hệ cơ số 10)
- + Hệ thập lục phân (là biểu diễn theo số đếm từ 1 đến 16)

Bảng biểu diễn 3 hệ số đếm trên:

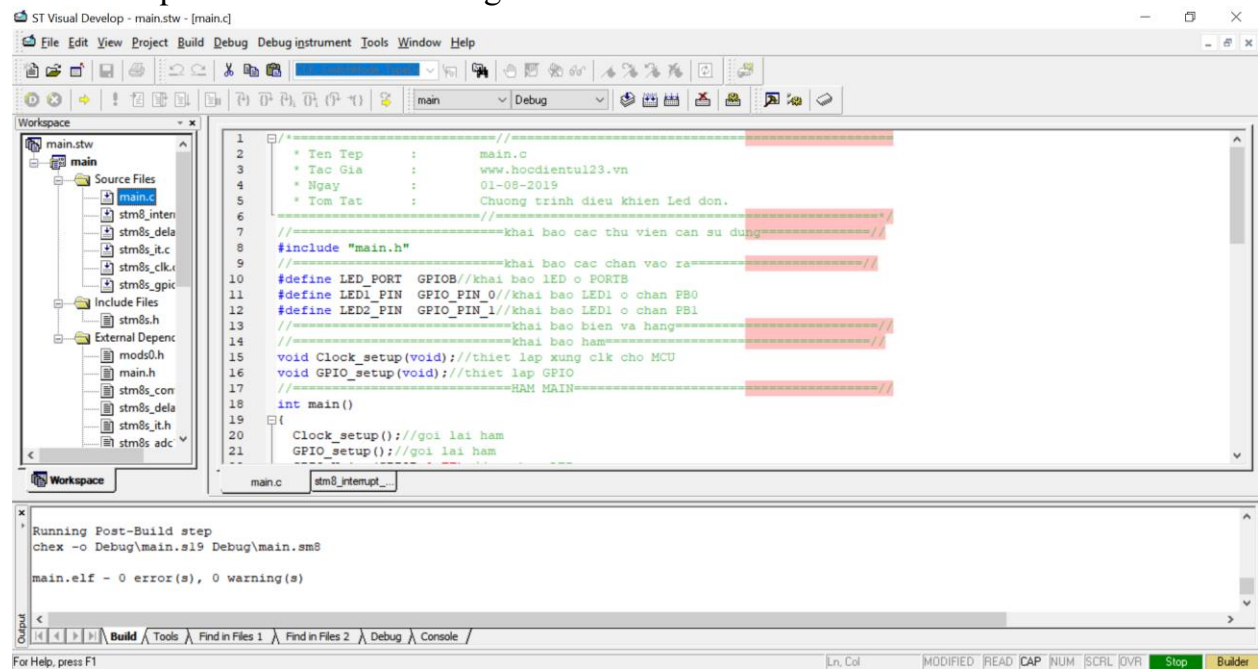
Số	Hệ nhị phân	Hệ thập phân	Hệ thập lục phân
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	8
9	1001	9	9
10	1010	10	A
11	1011	11	B
12	1100	12	C
13	1101	13	D
14	1110	14	E
15	1111	15	F

- Sử dụng các phép toán số học và các phép toán Logic.
- Sử dụng các cấu trúc điều kiện: **if-else; switch-case**
- Sử dụng vòng lặp : **while(); while-do; for()**
- Các thao tác xử lý mảng 1 chiều; 2 chiều.
- Cách khai báo và viết hàm.
- Các biến con trỏ....

=> Các bạn xem lại các Clip về học lập trình C cơ bản (các Clip hoàn toàn miễn phí tại khóa Lập Trình C Cơ bản) để hiểu rõ thêm!!!

3. Cấu trúc chương trình C trong ST Visual Develop

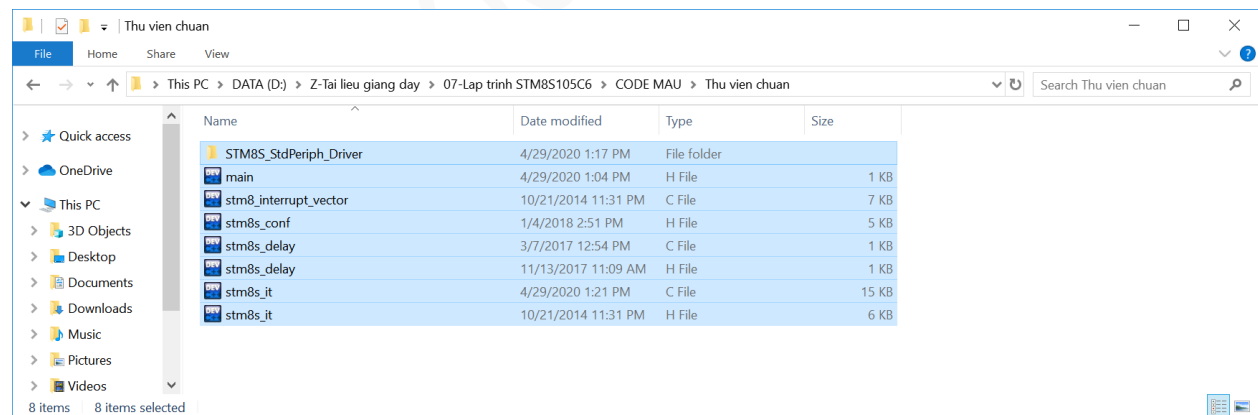
Các thành phần và cấu trúc chương trình C:



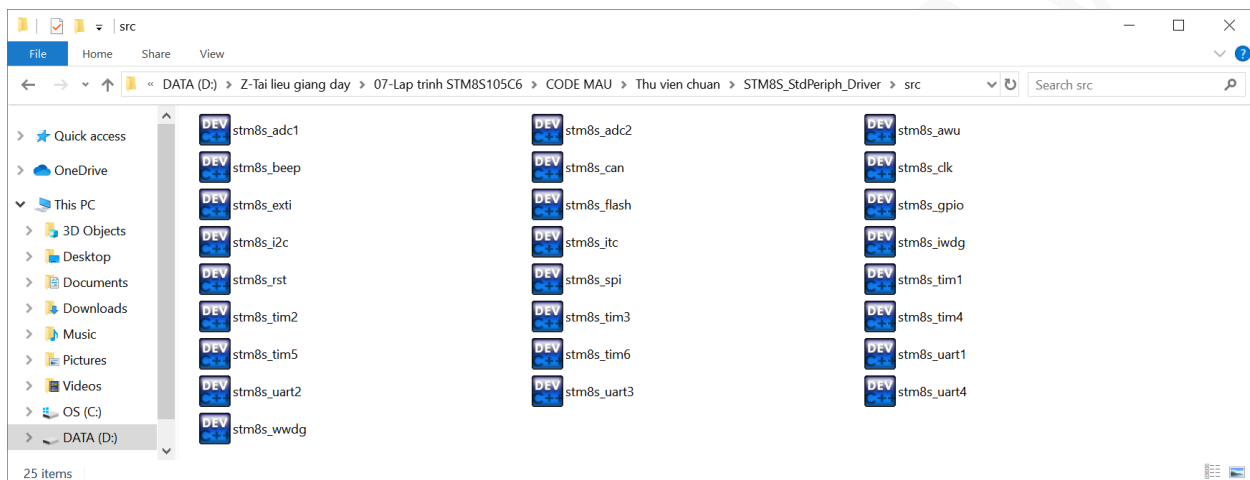
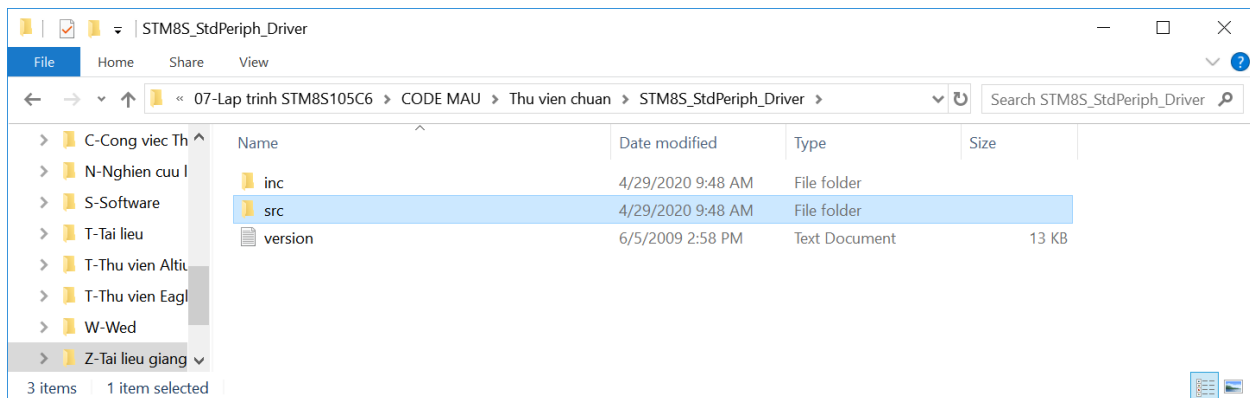
S

4. Giới thiệu thư viện cho STM8

Khi bạn tạo Project xong bạn phải copy các file trong Thư viện chuẩn vào trong cùng thư mục với Project mà bạn đã tạo:



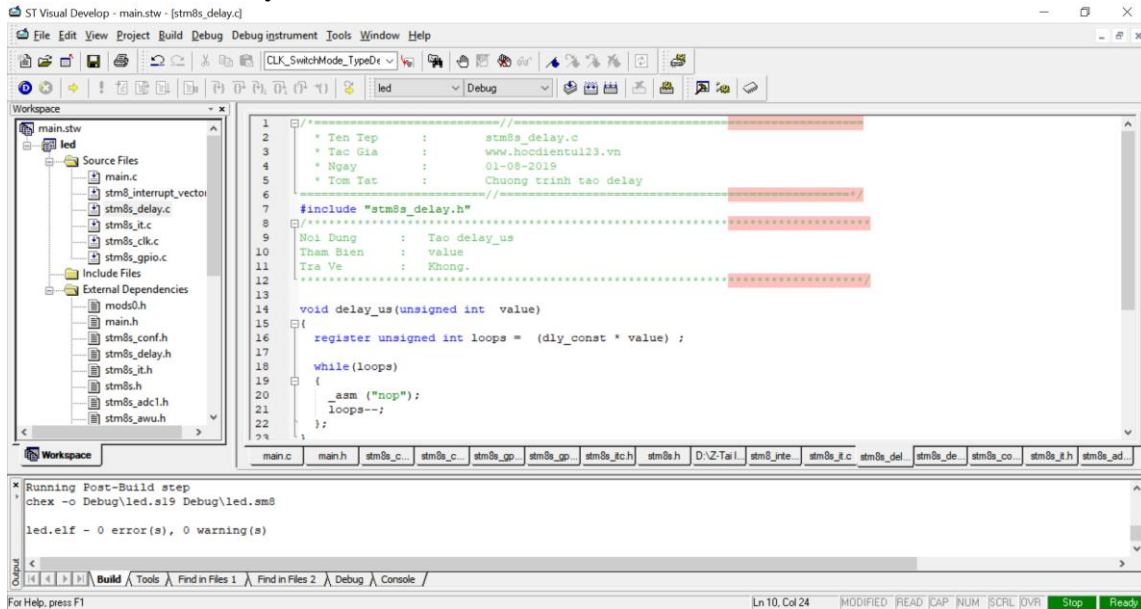
- Mục đích của việc này là để ta sử dụng các hàm, các chương trình, các biến, các định nghĩa có sẵn của nhà sản xuất.
- Trong thư viện **STM8S_StdPeriph_Driver** có hai thư mục **inc** và **src**; **inc** là chứa thư viện **.h**; còn **src** là thư viện **.c**;



- Thư viện này rất nhiều, nên chúng ta cần thư viện nào liên quan đến STM8 mà chúng ta đang lập trình thì chúng ta lấy ra sử dụng.

- Thư viện stm8s_delay

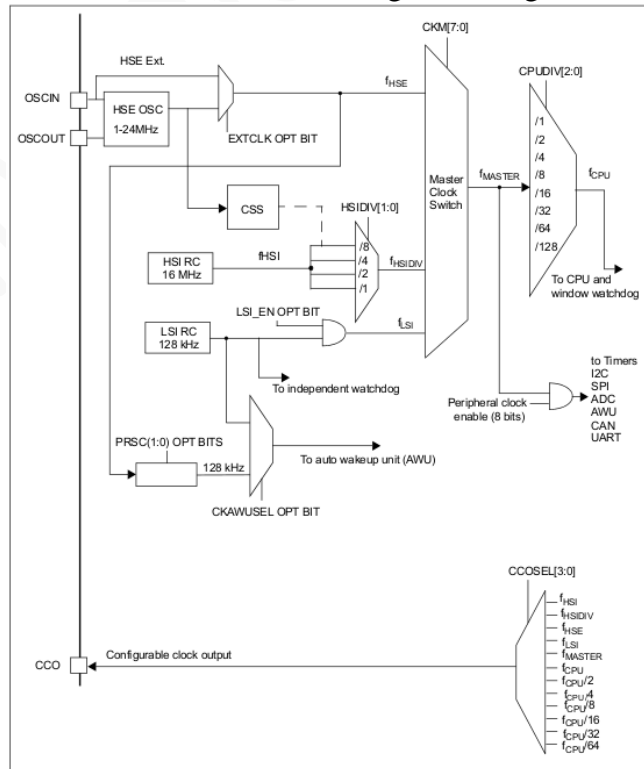
Tạo khoản delay us và ms:



- Thư viện stm8s_clk

- Trong STM8 có 3 nguồn xung clk:

- + High Speed Internal Clock (HSI) => Dùng dao động nội
- + High Speed External Clock (HSE) => Dùng dao động ngoại.
- + Low Speed Internal Clock (LSI) => Dùng dao động nội.



- Thư viện stm8s_clk.c

The screenshot displays the STM8S/A Standard Peripherals Firmware Library website. The left sidebar shows a search bar and a list of contents, including 'stm8s_clk.c'. The main content area is titled 'CLK_Public_Functions' and lists various functions with their descriptions and parameters. The functions listed are:

- CLK_AdjustHSICalibrationValue**: Adjusts the Internal High Speed oscillator (HSI) calibration value.
- CLK_CCOCmd**: Enables or disables the Configurable Clock Output (CCO).
- CLK_CCOConfig**: Output the selected clock on a dedicated I/O pin.
- CLK_ClearITPendingBit**: Clears the CLK interrupt pending bits.
- CLK_ClockSecuritySystemEnable**: Enables the Clock Security System.
- CLK_ClockSwitchCmd**: Starts or Stops manually the clock switch execution.
- CLK_ClockSwitchConfig**: Configures the Switch from one clock to another.
- CLK_DeInit**: Deinitializes the CLK peripheral registers to their default reset values.
- CLK_FastHaltWakeUpCmd**: Configures the High Speed Internal oscillator (HSI).
- CLK_GetClockFreq**: This function returns the frequencies of different on chip clocks.
- CLK_GetFlagStatus**: Checks whether the specified CLK flag is set or not.
- CLK_GetITStatus**: Checks whether the specified CLK interrupt has been enabled or not.
- CLK_GetSYSCLKSource**: Returns the clock source used as system clock.
- CLK_HSECmd**: Enable or Disable the External High Speed oscillator (HSE).
- CLK_HSICmd**: Enable or Disable the Internal High Speed oscillator (HSI).
- CLK_ITConfig**: Enables or disables the specified CLK interrupts.
- CLK_LSIcmd**: Enables or disables the Internal Low Speed oscillator (LSI).
- CLK_PeripheralClockConfig**: Enables or disables the specified peripheral CLK.
- CLK_SlowActiveHaltWakeUpCmd**: Configures the slow active halt wake up.
- CLK_SWIMConfig**: Configures the SWIM clock frequency on the fly.
- CLK_SYSCLKConfig**: Configures the HSI and CPU clock dividers.
- CLK_SYSClkEmergencyClear**: Reset the SWBSY flag (SWICR Register).

+ Cấu hình xung dùng dao động nội:

```
CLK_DeInit(); //Reset xung
CLK_HSICmd(ENABLE); //hai báo dùng dao động nội
CLK_ClockSwitchCmd(ENABLE); //cho hệ thống xung hoạt động
CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV8); //Chọn hệ số chia HSI
CLK_SYSCLKConfig(CLK_PRESCALER_CPUDIV2); //Chọn hệ số chia CPU
CLK_ClockSwitchConfig(CLK_SWITCHMODE_AUTO, CLK_SOURCE_HSE, DISABLE,
CLK_CURRENTCLOCKSTATE_DISABLE); //Cấu hình một số xung khác
```

+ Cấu hình xung dùng thạch anh ngoại:

```
CLK_DeInit();
CLK_HSECmd(ENABLE);
CLK_ClockSwitchCmd(ENABLE); //hai báo dùng dao động ngoại
CLK_SYSCLKConfig(CLK_PRESCALER_CPUDIV2);
CLK_ClockSwitchConfig(CLK_SWITCHMODE_AUTO, CLK_SOURCE_HSE, DISABLE,
CLK_CURRENTCLOCKSTATE_DISABLE);
```

+ Cấp xung cho các bộ I2C, SPI, UART1, AWU, ADC, TIMER...

```
CLK_PeripheralClockConfig(CLK_PERIPHERAL_I2C, DISABLE);
CLK_PeripheralClockConfig(CLK_PERIPHERAL_SPI, DISABLE);
CLK_PeripheralClockConfig(CLK_PERIPHERAL_UART1, DISABLE);
CLK_PeripheralClockConfig(CLK_PERIPHERAL_AWU, DISABLE);
CLK_PeripheralClockConfig(CLK_PERIPHERAL_ADC, DISABLE);
```



```
CLK_PeripheralClockConfig(CLK_PERIPHERAL_TIMER1, DISABLE);
CLK_PeripheralClockConfig(CLK_PERIPHERAL_TIMER2, DISABLE);
CLK_PeripheralClockConfig(CLK_PERIPHERAL_TIMER4, DISABLE);
```

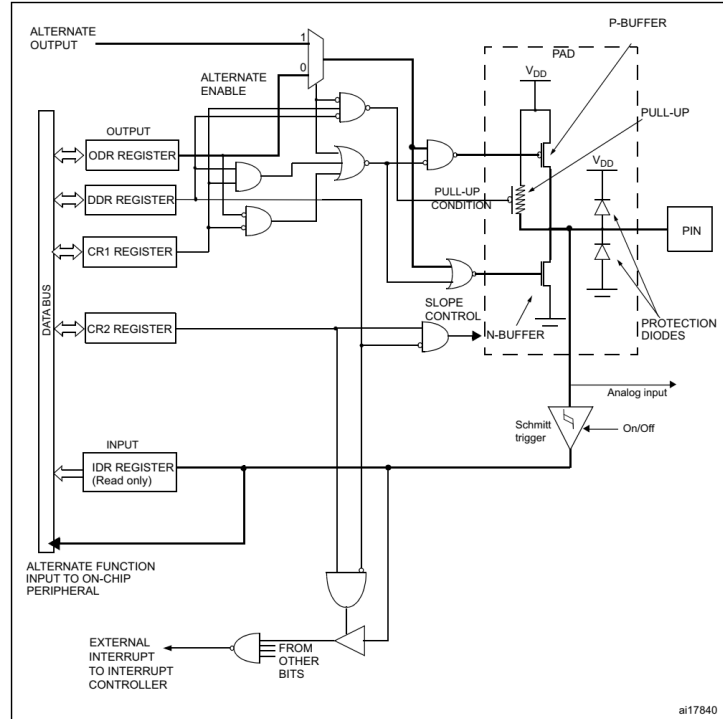
+ Xuất xung ra chân CCO với tần số được tính từ tần số HSE hoặc HSI

```
CLK_CCOCmd(ENABLE);
CLK_CCOCmd(ENABLE);
```

- Thư viện `stm8s_gpio`

+ Cấu trúc IO của một GPIO

Figure 24. GPIO block diagram



Chú ý:

- Khi lập trình I2C thì hai chân giao tiếp I2C(SDA và SCL) phải có trở kéo lên nguồn.
- Các chân GPIO đều có chức năng ngắt ngoài.

STM8S/A Standard Peripherals Firmware Library

Contents Index Search Fav

Type in the keyword to find:

GPIO_Public_Functions

GPIO_Pin
GPIO_PIN_0
GPIO_PIN_1
GPIO_PIN_2
GPIO_PIN_3
GPIO_PIN_4
GPIO_PIN_5
GPIO_PIN_6
GPIO_PIN_7
GPIO_PIN_ALL
GPIO_PIN_HNIB
GPIO_PIN_LNIB
GPIO_Private_Macros
IS_GPIO_MODE_OK
IS_GPIO_PIN_OK
GPIO_Public_Functions
GPIO_DeInit
GPIO_ExternalPullUpConfig
GPIO_Init
GPIO_ReadInputData
GPIO_ReadInputPin
GPIO_ReadOutputData
GPIO_Write
GPIO_WriteHigh
GPIO_WriteLow
GPIO_WriteReverse
GPIO_ReadInputData
GPIO_Exported_Functions
GPIO_Public_Functions
stm8s_gpio.c
stm8s_gpio.h
GPIO_ReadInputPin
GPIO_Exported_Functions
GPIO_Public_Functions
stm8s_gpio.c

Display

STM8S/A Standard Peripherals Firmware Library

Main Page Related Pages Modules Data Structures Files Directories Functions

GPIO_Public_Functions

STM8S_StdPeriph_Driver

Functions

void **GPIO_DeInit** (GPIO_TypeDef *GPIOx)
Deinitializes the GPIOx peripheral registers to their default reset values.

void **GPIO_ExternalPullUpConfig** (GPIO_TypeDef *GPIOx, GPIO_Pin_TypeDef GPIO_Pin, FunctionalState NewState)
Configures the external pull-up on GPIOx pins.

void **GPIO_Init** (GPIO_TypeDef *GPIOx, GPIO_Pin_TypeDef GPIO_Pin, GPIO_Mode_TypeDef GPIO_Mode)
Initializes the GPIOx according to the specified parameters.

uint8_t **GPIO_ReadInputData** (GPIO_TypeDef *GPIOx)
Reads the specified GPIO input data port.

BitStatus **GPIO_ReadInputPin** (GPIO_TypeDef *GPIOx, GPIO_Pin_TypeDef GPIO_Pin)
Reads the specified GPIO input data pin.

uint8_t **GPIO_ReadOutputData** (GPIO_TypeDef *GPIOx)
Reads the specified GPIO output data port.

void **GPIO_Write** (GPIO_TypeDef *GPIOx, uint8_t PortVal)
Writes data to the specified GPIO data port.

void **GPIO_WriteHigh** (GPIO_TypeDef *GPIOx, GPIO_Pin_TypeDef PortPins)
Writes high level to the specified GPIO pins.

void **GPIO_WriteLow** (GPIO_TypeDef *GPIOx, GPIO_Pin_TypeDef PortPins)
Writes low level to the specified GPIO pins.

void **GPIO_WriteReverse** (GPIO_TypeDef *GPIOx, GPIO_Pin_TypeDef PortPins)
Writes reverse level to the specified GPIO pins.

Function Documentation

void **GPIO_DeInit** (GPIO_TypeDef * GPIOx)

Deinitializes the GPIOx peripheral registers to their default reset values.

main.c - Dev-C++

11:05 AM
4/29/2020

+ Các lệnh thiết lập các GPIO:

```
//khai báo LED kết nối với GPIOB
#define LED_PORT    GPIOB//khai báo LED ở PORTB
#define LED1_PIN    GPIO_PIN_0//khai báo LED1 ở chân PB0
#define LED2_PIN    GPIO_PIN_1//khai báo LED2 ở chân PB1

//thiết lập ngõ vào ra
//Reset GPIOB
GPIO_DeInit(GPIOB);
//Thiết lập chân PB0 hay LED1 là ngõ ra
GPIO_Init(LED_PORT,LED1_PIN,GPIO_MODE_OUT_PP_LOW_FAST);
//Thiết lập chân PB0,PB1 hay LED1, LED2 là ngõ ra
GPIO_Init(LED_PORT,LED1_PIN|LED2_PIN,GPIO_MODE_OUT_PP_LOW_FAST);
//Reset GPIOD
GPIO_DeInit(GPIOD);
//Thiết lập chân PD0 là ngõ vào không ngắt
GPIO_Init(GPIOD, GPIO_PIN_0,GPIO_MODE_IN_FL_NO_IT);
//Thiết lập chân PD0 vào có ngắt
GPIO_Init(GPIOD, GPIO_PIN_0,GPIO_MODE_IN_FL_IT);
```

+ Các lệnh xử lý GPIO:

```
GPIO_WriteLow(LED_PORT,LED1_PIN);//xuất ra mức thấp(mức 0).
GPIO_WriteHigh(LED_PORT,LED1_PIN);//xuất ra mức cao(mức 1).

GPIO_WriteLow(GPIOB,GPIO_PIN_7);//xuất ra mức thấp(mức 0).
GPIO_WriteHigh(GPIOB,GPIO_PIN_7);//xuất ra mức cao(mức 1).
GPIO_WriteReverse(GPIOB,GPIO_PIN_7);//đảo trạng thái của chân xuất ra.
```

```
GPIO_Write(GPIOB,0x00); //xuất dữ liệu 8 bit ra GPIOB
```

```
GPIO_ReadInputPin(GPIOD, GPIO_PIN_0); //đọc 1 bit vào ở chân D0
```

```
GPIO_ReadInputData(GPIOD, GPIO_PIN_0); //đọc 8 bit vào ở GPIOD
```


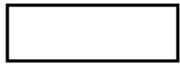
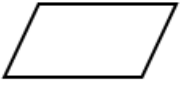
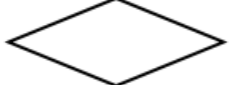

Một số thư viện khác như: Ngắt ngoài, ADC, Timer, UART, I2C, SPI, Flash, BEEP...



5. Lưu đồ giải thuật

- Thể hiện ý tưởng lập trình.
- Thể hiện nội dung và các bước lập trình.
- Kiểm soát tốt lỗi và làm cho chương trình nhỏ hơn.

Các ký hiệu:

Ký hiệu	Giải thích
	Bắt đầu hoặc kết thúc
	Tính toán
	Nhập hoặc xuất
	Rẽ nhánh
	Luồng xử lý

Ví dụ: Viết chương trình chớp tắt led.
Ta có lưu đồ giải thuật:

