电子学基础II实验

--单片机第3次课

詹洪陈

2022-11-4

目录/Contents

PWM作业

OLED

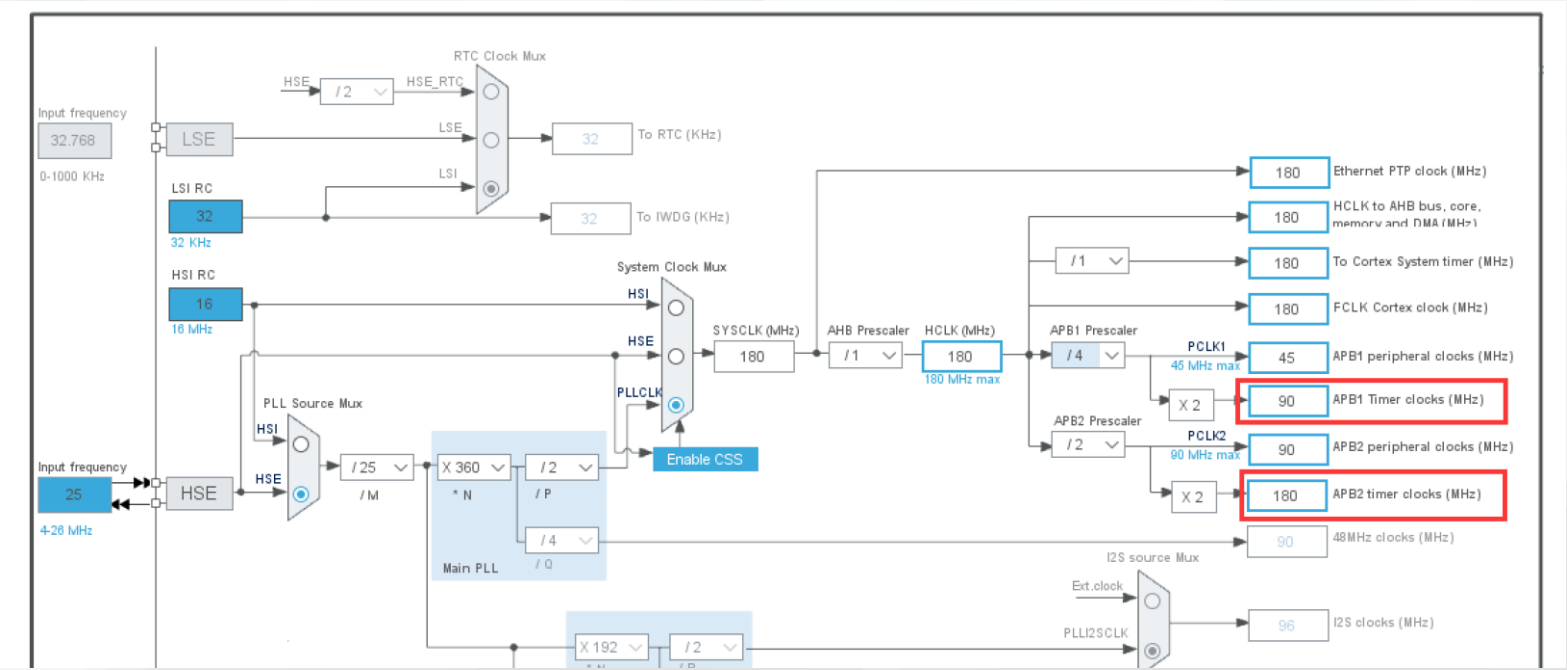
课堂作业

PWM作业讲解

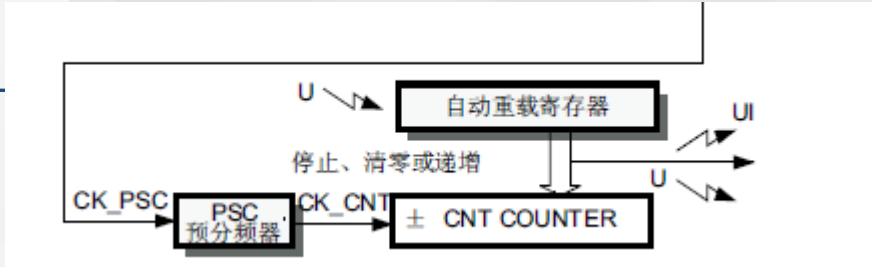
时钟源

通用定时器Timer2~Timer5、通用定时器12~Timer14、基本定时器Timer6，Timer7的时钟来源是**APB1总线**

高级定时器Timer1、Timer8以及通用定时器9~Timer11的时钟来源是**APB2总线**



时基单元



(1)16位的预分频器TIMx_PSC对内部时钟CK_INT进行分频之后，得到计数器时钟CK_CNT=CK_PSC/PSC+1

(2)计数器CNT在计数器时钟的驱动下开始计数，计数一次的时间为1/CK_CNT

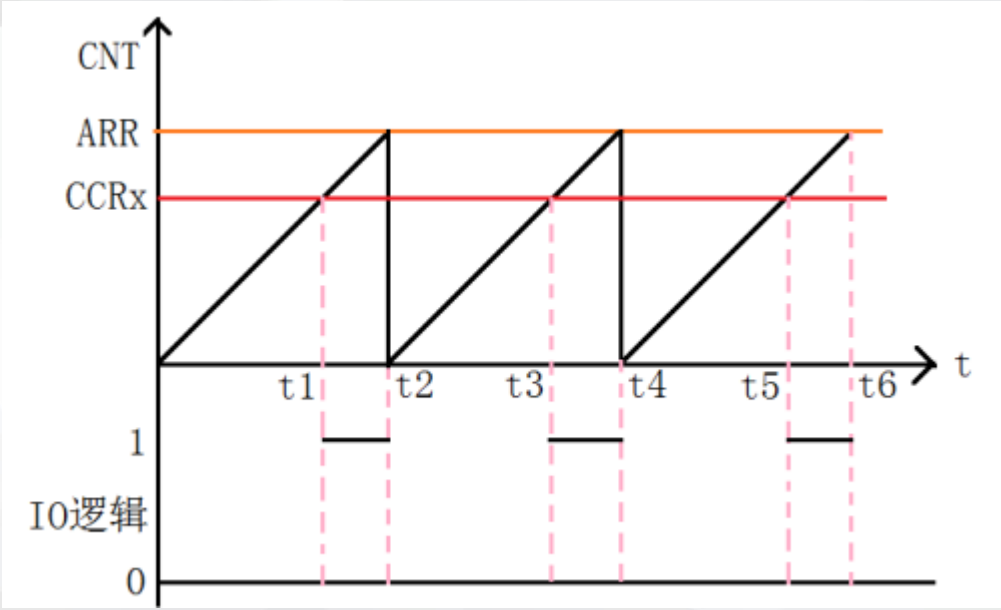
(3)定时器使能(CEN 置 1)后，计数器 CNT在CK_CNT 驱动下向上计数，当TIMx_CNT 值与 TIMx_ARR 的设定值相等时就自动生成事件并 TIMx_CNT 自动清零，然后自动重新开始计数，如此重复以上过程。

$$Tout= (ARR+1)(PSC+1)/CK\_PSC$$

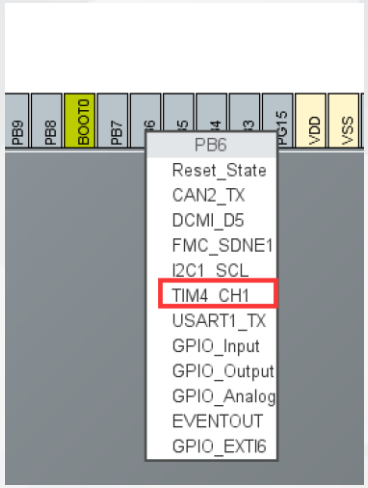
PWM 的工作原理

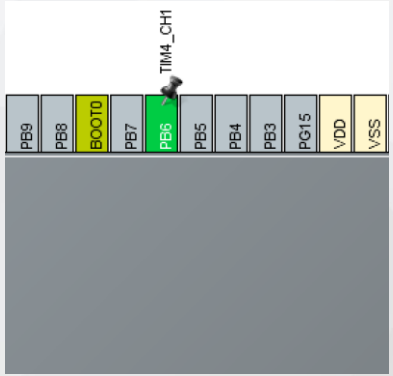
假定定时器工作在向上计数PWM模式，且当CNT<CCRx时，输出0，当CNT>=CCRx时输出1。

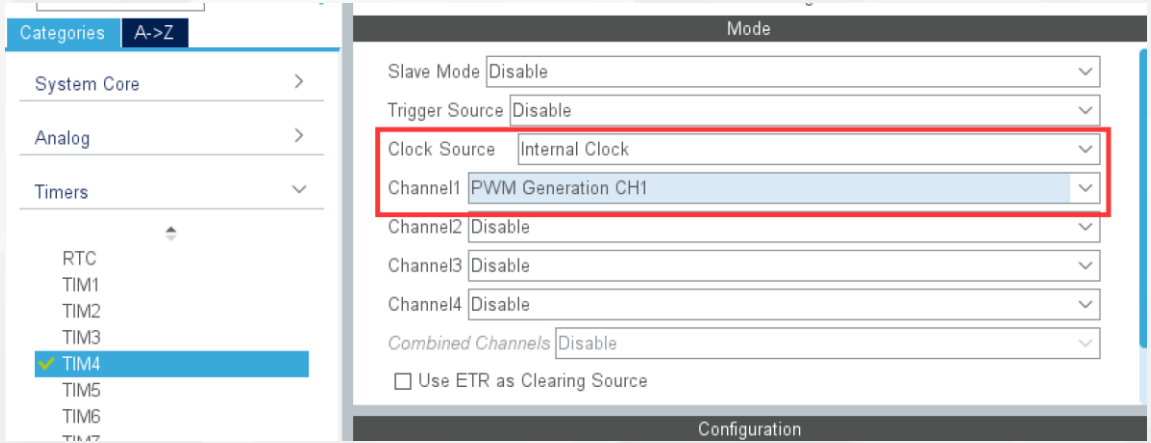
则PWM示意如图：当CNT值小于CCRx的时候，IO输出低电平(0)，当CNT值大于等于CCRx的时候，IO输出高电平(1)，当CNT达到ARR值的时候，重新归零，然后重新向上计数，依次循环。改变CCRx的值，就可以改变PWM输出的占空比，改变ARR的值，就可以改变PWM输出的频率，这就是PWM输出的原理。



引脚选择


PB9
PB8
BOOT0
PB7
PB6
PB5
PB4
PB3
PG15
VDD
VSS
PB6
Reset_State
CAN2_TX
DCMI_D5
FMC_SDNE1
I2C1_SCL
TIM4_CH1
USART1_TX
GPIO_Input
GPIO_Output
GPIO_Analog
EVENTOUT
GPIO_EXTI6


TIM4_CH1
PB9
PB8
BOOT0
PB7
PB6
PB5
PB4
PB3
PG15
VDD
VSS


Categories
A->Z
System Core
Analog
Timers
RTC
TIM1
TIM2
TIM3
TIM4
TIM5
TIM6
Mode
Slave Mode
Disable
Trigger Source
Disable
Clock Source
Internal Clock
Channel1
PWM Generation CH1
Channel2
Disable
Channel3
Disable
Channel4
Disable
Combined Channels
Disable
Use ETR as Clearing Source
Configuration

参数设置


Parameter Settings
User Constants
Configure the below parameters :
Search (Crtl+F)
Counter Settings
Prescaler (PSC - 16 bits value) 90-1
Counter Mode Up
Counter Period (AutoReload Regist... 10000-1
Internal Clock Division (CKD) No Division
auto-reload preload Enable
Trigger Output (TRGO) Parameters
Master/Slave Mode (MSM bit) Disable (Trigger input effect not delayed)
Trigger Event Selection Reset (UG bit from TIMx_EGR)
PWM Generation Channel 1
Mode PWM mode 1
Pulse (16 bits value) 5000
Output compare preload Enable
Fast Mode Disable
CH Polarity High
预分频值PSC
计数模式，向上计数
自动重装载值ARR
PWM模式1
比较寄存器值CCR
输出比较极性为高

添加功能代码

1、开启定时器对应的通道的PWM输出

HAL_StatusTypeDef HAL_TIM_PWM_Start(TIM_HandleTypeDef *htim, uint32_t Channel)

2、改变CCR的值（改变占空比）

一种是调用hal库里面的__HAL_TIM_SetCompare( )函数

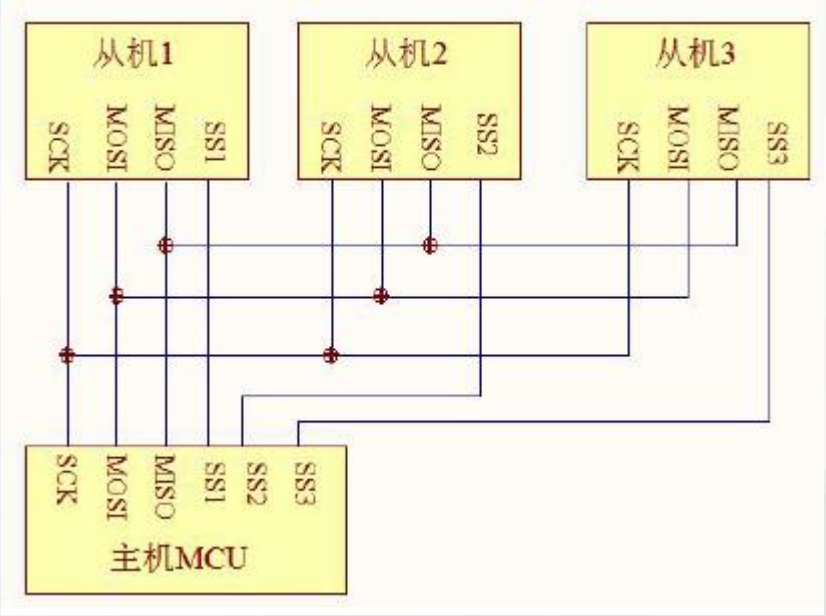
一种就是直接操作寄存器TIMx->CCRy=compare;

OLED简介

参考：5-Arm Cortex-M4系统设计-STM32F429开发板教材
实验四 LCD 显示—硬件 SPI

SPI简介

SPI，是一种高速的，全双工，同步的通信总线，并且在芯片的管脚上只占用四根线，节约了芯片的管脚，同时为PCB的布局上节省空间，提供方便。



(1)SS(Slave Select)：从设备选择信号线，常称为片选信号线，也称为NSS、CS

(2) SCK (Serial Clock)：时钟信号线，用于通讯数据同步

(3) MOSI (Master Output， Slave Input)：主设备输出/从设备输入引脚

(4) MISO(Master Input,，Slave Output)：主设备输入/从设备输出引脚。

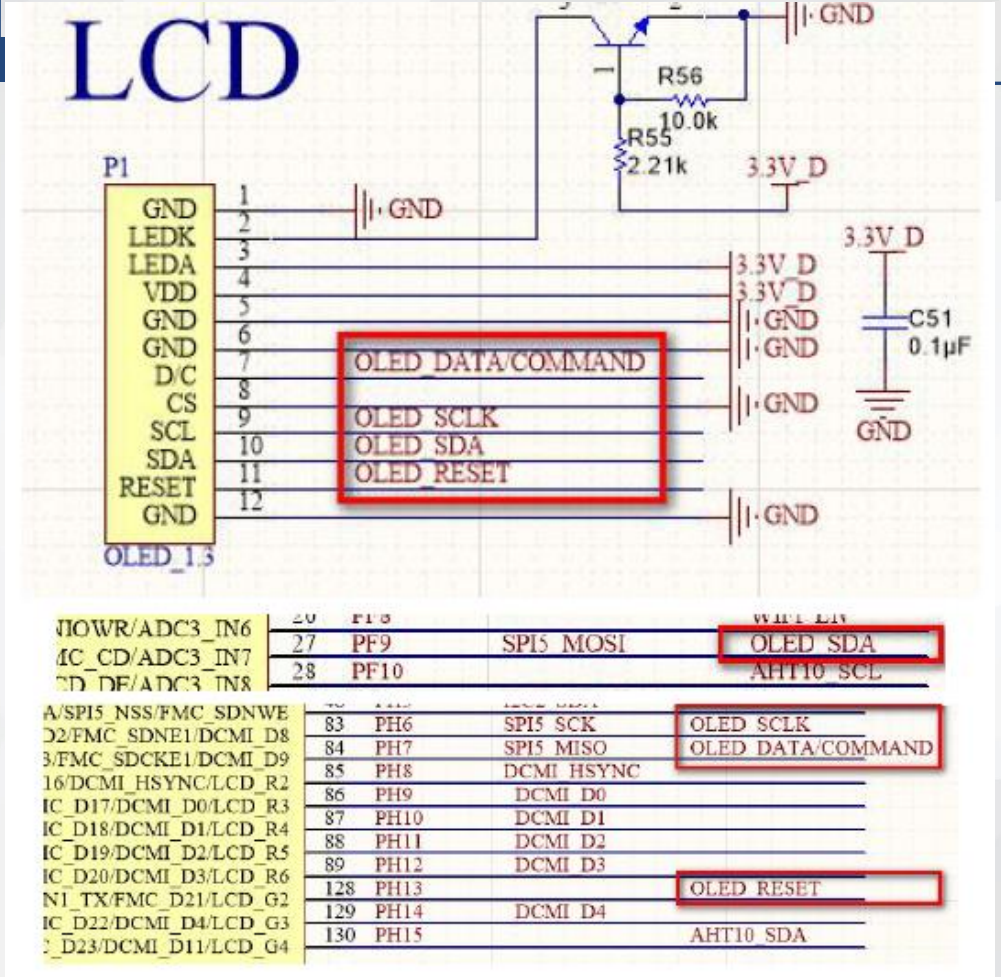
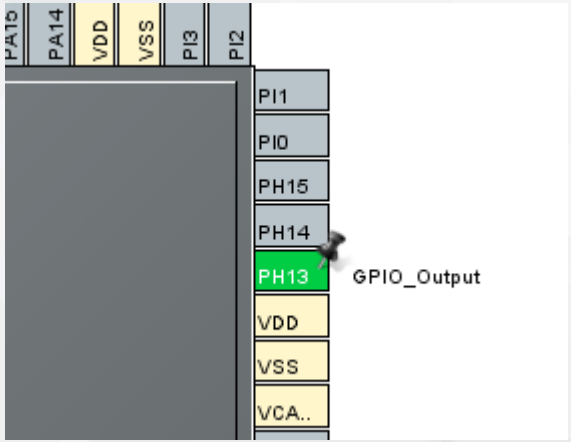
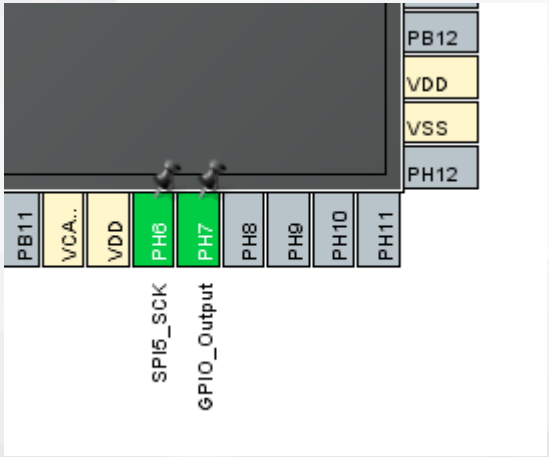
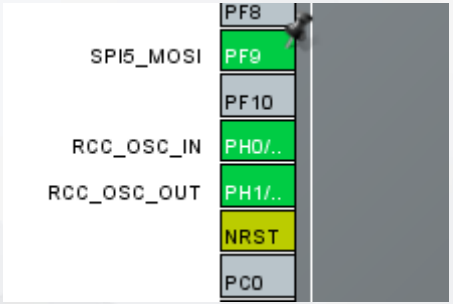


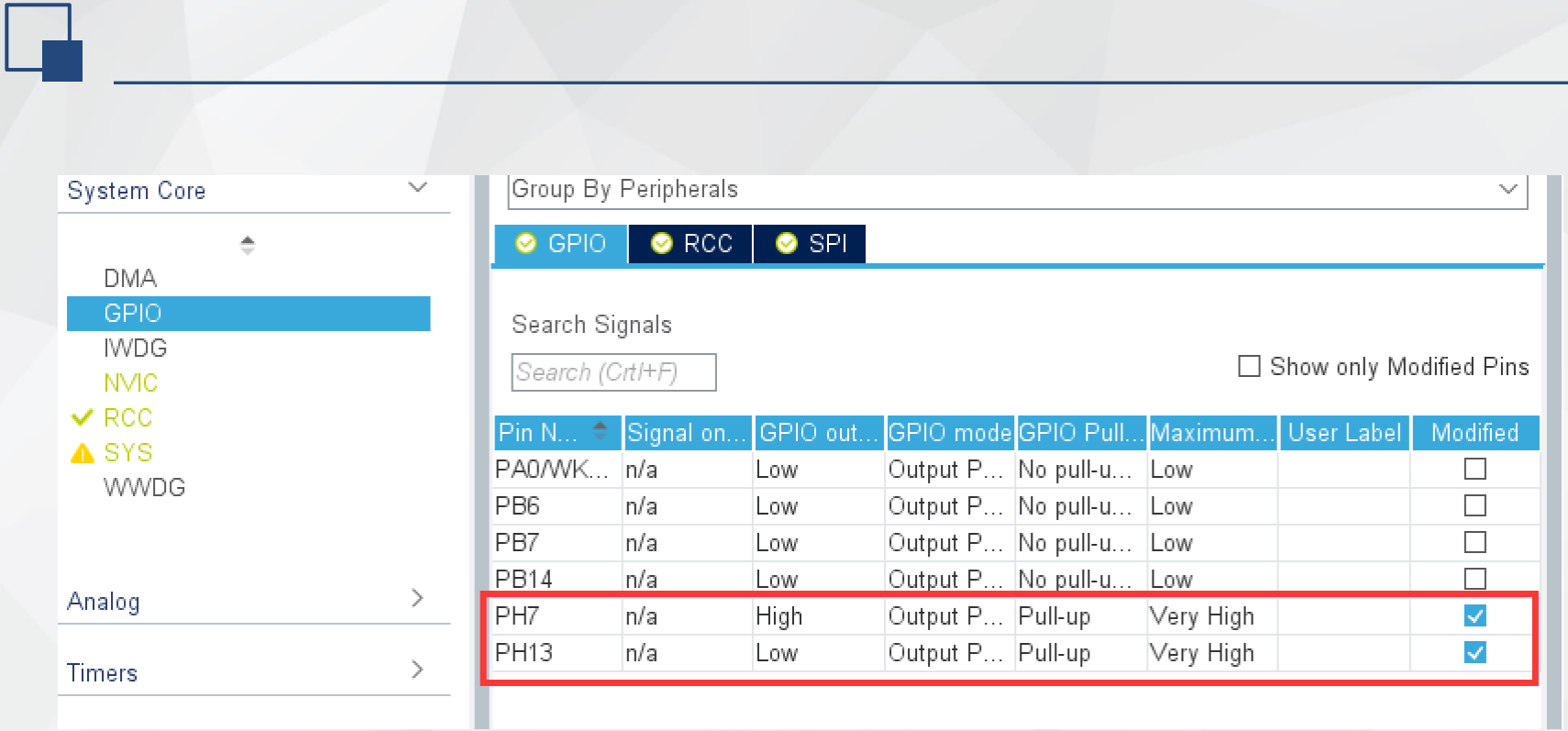
表 4.1　LCD 引脚说明

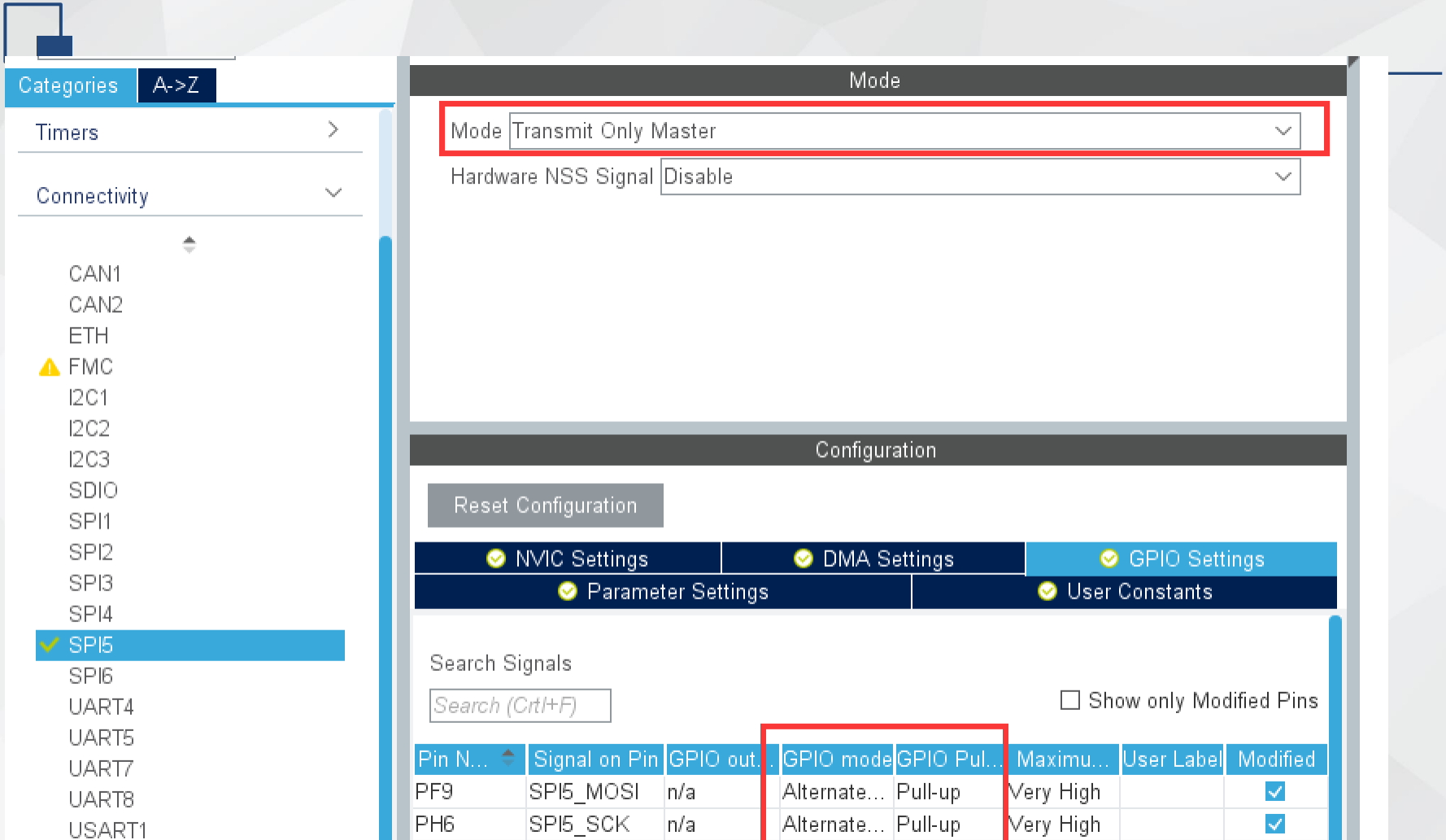
设备名	引脚号
OLED_DATA/COMMAND	PH7
OLED_SCLK	PH6
OLED_SDA	PF9
OLED_RESET	PH13

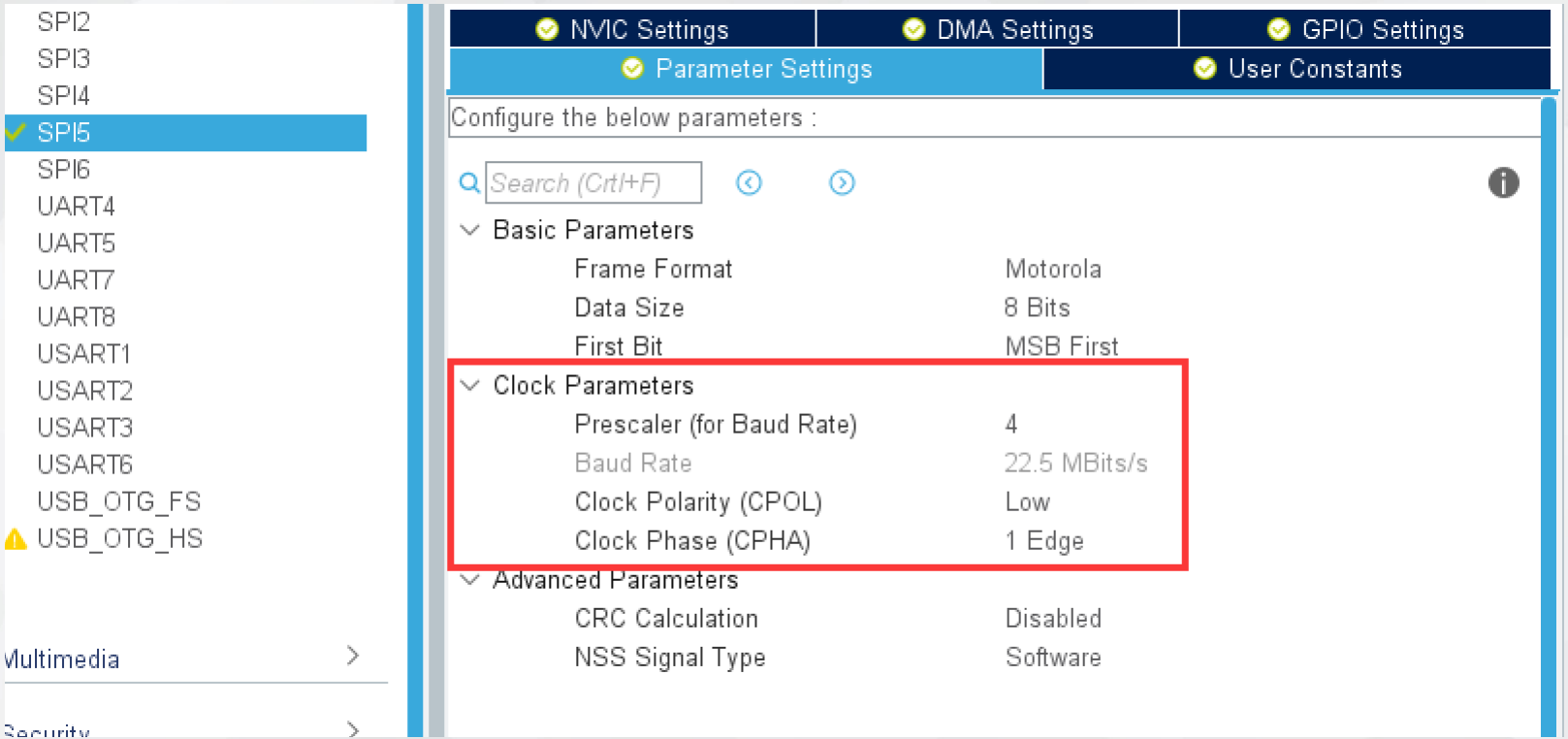

PA14
VDD
VSS
PI3
PI2
PI1
PI0
PH15
PH14
PH13
GPIO_Output
VDD
VSS
VCA..

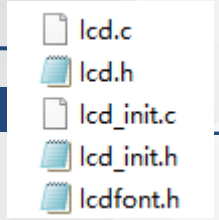

PB12
VDD
VSS
PH12
PB11
VCA..
VDD
PH6
PH7
PH8
PH9
PH10
PH11
SPI5_SCK
GPIO_Output


PF8
SPI5_MOSI
PF9
PF10
RCC_OSC_IN
PH0/..
RCC_OSC_OUT
PH1/..
NRST
PC0


System Core
DMA
GPIO
IWDG
NVIC
RCC
SYS
WWDG
Analog
Timers
Group By Peripherals
GPIO
RCC
SPI
Search Signals
Search (Crtl+F)
Show only Modified Pins
Pin N...
Signal on...
GPIO out...
GPIO mode
GPIO Pull...
Maximum...
User Label
Modified
PA0/WK... n/a Low Output P... No pull-u... Low
PB6 n/a Low Output P... No pull-u... Low
PB7 n/a Low Output P... No pull-u... Low
PB14 n/a Low Output P... No pull-u... Low
PH7 n/a High Output P... Pull-up Very High
PH13 n/a Low Output P... Pull-up Very High


Categories
A->Z
Timers
Connectivity
CAN1
CAN2
ETH
FMC
I2C1
I2C2
I2C3
SDIO
SPI1
SPI2
SPI3
SPI4
SPI5
SPI6
UART4
UART5
UART7
UART8
USART1
Mode
Mode
Transmit Only Master
Hardware NSS Signal
Disable
Configuration
Reset Configuration
NVIC Settings
DMA Settings
GPIO Settings
Parameter Settings
User Constants
Search Signals
Search (Crtl+F)
Show only Modified Pins
Pin N...
Signal on Pin
GPIO out...
GPIO mode
GPIO Pul...
Maximu...
User Label
Modified
PF9
SPI5_MOSI
n/a
Alternate...
Pull-up
Very High
PH6
SPI5_SCK
n/a
Alternate...
Pull-up
Very High


SPI2
SPI3
SPI4
SPI5
SPI6
UART4
UART5
UART7
UART8
USART1
USART2
USART3
USART6
USB_OTG_FS
USB_OTG_HS
Multimedia
Security
NVIC Settings
DMA Settings
GPIO Settings
Parameter Settings
User Constants
Configure the below parameters :
Search (Crtl+F)
Basic Parameters
Frame Format
Motorola
Data Size
8 Bits
First Bit
MSB First
Clock Parameters
Prescaler (for Baud Rate)
4
Baud Rate
22.5 MBits/s
Clock Polarity (CPOL)
Low
Clock Phase (CPHA)
1 Edge
Advanced Parameters
CRC Calculation
Disabled
NSS Signal Type
Software




```
void LCD_GPIO_Init(void);//初始化GPIO
void LCD_Writ_Bus(uint8_t dat);//模拟SPI时序
void LCD_WR_DATA8(uint8_t dat);//写入一个字节
void LCD_WR_DATA(uint16_t dat);//写入两个字节
void LCD_WR_REG(uint8_t dat);//写入一个指令
void LCD_Address_Set(uint16_t x1,uint16_t y1,uint16_t x2,uint16_t y2);//设置坐标函数
void LCD_Init(void);//LCD初始化
#endif
```

```
void LCD_Fill(uint16_t xsta,uint16_t ysta,uint16_t xend,uint16_t yend,uint16_t color);//指定区域填充颜色
void LCD_DrawPoint(uint16_t x,uint16_t y,uint16_t color);//在指定位置画一个点
void LCD_DrawLine(uint16_t x1,uint16_t y1,uint16_t x2,uint16_t y2,uint16_t color);//在指定位置画一条线
void LCD_DrawRectangle(uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2,uint16_t color);//在指定位置画一个矩形
void Draw_Circle(uint16_t x0,uint16_t y0,uint8_t r,uint16_t color);//在指定位置画一个圆

void LCD_ShowChinese(uint16_t x,uint16_t y,uint8_t *s,uint16_t fc,uint16_t bc,uint8_t sizey,uint8_t mode);//显示汉字串
void LCD_ShowChinese16x16(uint16_t x,uint16_t y,uint8_t *s,uint16_t fc,uint16_t bc,uint8_t sizey,uint8_t mode);//显示单个16x16汉字
void LCD_ShowChinese24x24(uint16_t x,uint16_t y,uint8_t *s,uint16_t fc,uint16_t bc,uint8_t sizey,uint8_t mode);//显示单个24x24汉字
void LCD_ShowChinese32x32(uint16_t x,uint16_t y,uint8_t *s,uint16_t fc,uint16_t bc,uint8_t sizey,uint8_t mode);//显示单个32x32汉字

void LCD_ShowChar(uint16_t x,uint16_t y,uint8_t num,uint16_t fc,uint16_t bc,uint8_t sizey,uint8_t mode);//显示一个字符
void LCD_ShowString(uint16_t x,uint16_t y,const uint8_t *p,uint16_t fc,uint16_t bc,uint8_t sizey,uint8_t mode);//显示字符串
uint32_t mypow(uint8_t m,uint8_t n);//求幂
void LCD_ShowIntNum(uint16_t x,uint16_t y,uint16_t num,uint8_t len,uint16_t fc,uint16_t bc,uint8_t sizey);//显示整数变量
void LCD_ShowFloatNum1(uint16_t x,uint16_t y,float num,uint8_t len,uint16_t fc,uint16_t bc,uint8_t sizey);//显示两位小数变量

void LCD_ShowPicture(uint16_t x,uint16_t y,uint16_t length,uint16_t width,const uint8_t pic[]);//显示图片
```

课堂作业

课堂作业

单片机题目：

通过AD采集一组信号（如：信号源产生的正弦波信号），求出最大值、最小值、平均值在OLED 屏幕上显示

注意： ADC 的输入电压范围为：0~3.3V ！！！