

## 1. 系统主频 80M Hz, 实现 1000ms 的计时器。

SYSTick.c

SYSTick.h

计时器

```
SYSTick.c 子程序.
#define NVIC_ST_CTRL_R (* (volatile unsigned long *) 0xE000E010)
#define NVIC_ST_RELOAD_R (* (volatile unsigned long *) 0xE000E014)
#define NVIC_ST_CURRENT_R (* (volatile unsigned long *) 0xE000E018)

void SYSTick_Init(void) {
    NVIC_ST_CTRL_R = 0;
    NVIC_ST_CTRL_R = 0x0000005;
}

void SYSTick_Wait(unsigned long delay) {
    NVIC_ST_RELOAD_R = delay - 1;
    NVIC_ST_CURRENT_R = 0;
    while ((NVIC_ST_CTRL_R & 0x00000005) == 0) {}
}

void SYSTick_Wait10ms(unsigned long delay) {
    unsigned long i;
    for(i=0; i<delay; i++) {
        SYSTick_Wait(800000); // 10ms.
    }
}

void SYSTick_Wait100ms(unsigned long delay) {
    unsigned long i;
    for(i=0; i<delay; i++) {
        SYSTick_Wait10ms(100); // 100x10ms = 1000ms.
    }
}
```

SYSTick.h 头文件.

```
void SYSTick_Init(void);
void SYSTick_Wait(unsigned long delay);
void SYSTick_Wait10ms(unsigned long delay);
void SYSTick_Wait100ms(unsigned long delay);
```

PLL.c

PLL.h

main.c

初始化 PLL 电路

子程序.

```

PLL.c.
#include "PLL.h"
Void PLL_Init(Void) {
    SYSTCL_RCC2_R |= SYSTCL_RCC2_UserRCC2;
    SYSTCL_RCC2_R |= SYSTCL_RCC2_BYPASS2;
    SYSTCL_RCC_R &= ~SYSTCL_RCC_XTAL_M;
    SYSTCL_RCC_R += SYSTCL_RCC_XTAL_16M;
    SYSTCL_RCC2_R &= ~SYSTCL_RCC2_OSCSRC2_M;
    SYSTCL_RCC2_R += SYSTCL_RCC2_OSCSRC2_M0;
    SYSTCL_RCC2_R &= ~SYSTCL_RCC2_PWRDN2;
    SYSTCL_RCC2_R |= SYSTCL_RCC2_DIV400;
    SYSTCL_RCC2_R = (SYSTCL_RCC2_R & ~0X1FC0000) + (SYSDIV2 << 22);
    While ((SYSTCL_RIS_R & SYSTCL_RIS_PLLRIS) == 0) {}
    SYSTCL_RCC2_R &= ~SYSTCL_RCC2_BYPASS2;
}

```

PLL.h. 头文件.

```

#define SYSDIV2 4
Void PLL_Init(Void);

```

主程序 main.c.

```

#include "PLL.h"
#include "SYSTick.h". 其中
#include "tm4c123gh6pm.h". # 声明宏定义头文件.

```

```

int main() {
    PLL_Init(); // 设置系统时钟 80MHz.
    SYSTick_Init(); // 初始化 SYSTick.
    SysTick_Wait1000ms(2); // 等待 2x1000ms.
}

```

## 2. CAN, I2C, SSI 三种总线协议的讲解、分析

### CAN 控制器局域网

具有 CAN-High 和 CAN-Low 一对差分总线, 以差分信号的形式进行通信。

高速 CAN 传输速率最大为 1M bps, 支持最长距离 40m。

低速 CAN 最高速率只有 125 kbps, 且可以在 CAN-High 或 Low 任意一端断掉时保持正常通信。

是异步通信方式, 使用位同步来抗干扰和吸收时序误差。

数据传输由四种帧类型组成: 数据帧, 远程帧, 错误帧, 过载帧。

不分优先, 但可以设置优先级来满足不同的实时要求。

### I2C 集成电路总线

由 SDA (串行数据线) 和 SCL (串行时钟线) 以及上拉电阻组成。

具有三种传输模式: 标准模式 100 kbps, 快速模式 400 kbps, 高速 3.4 M bps。

多主和使用时, 使用仲裁方式占用总线; 每个设备有独立的地址, 主机利用地址进行

该协议定义了通信的起始, 停止信号, 数据有效性, 仲裁, 时钟同步和地址广播等环节。

使用 SDA 传输数据, SCL 控制同步, SDA 在 SCL 的每个时钟周期传输一位数据。

### SSI 同步串行接口

选用主机电动式读出方法, 在时钟信号控制下, 从 MSB 开始同步传输数据。

由发送电路, 接收电路, 串行时钟和帧同步发生电路组成。

有三线制的 SPI 和四线制的 Microwire。

提供串行时钟的设备为主机, 可与其它设备进行双向通信。可增加设备选择线选择多个从设备。

收发独立, 可同步进行。