## 先进处理器报告

#### 项目完成情况与整体思路

对于bootloader来说，首先需要设置两个中断，一个是计数器中断，用来检测modbus协议中数据帧超时的情况，另一个是EUSART的接收中断，可以在bootloader接收串口发送的协议的时候直接写入用户程序。因为rs485的特性，需要首先需要将RC7引脚置零，然后当接收到一帧数据的时候将RC7置高。CRC校验使用的是标准的CRC-16-MODBUS协议用来检测串口发送的HEX文件是否正确，单片机接收到HEX文件命令后，利用memory写入特定地址的flash，还可以通过特定命令檫除相关地址的程序，最后实现完整的bootloader功能。

#### 设计目的

可以通过串口线直接下载程序，而不需要在利用下载器，可以有效的满足大楼里面空调、楼梯等相关需求

#### 关键技术方案

##### Modbus协议与CRC校验

主从机的modbus协议，利用0041写入hex到flash里面，CRC校验利用的是标准的CRC-16-MODBUS协议。

##### HEX文件与C语言的串口发送

将编译器编译好的hex文件重新排列，前四位是操作字节，然后是写入的字节长度以及需要写入的字符数据，最后两位是CRC校验。

CRC是多项式的计算方法，校验多项式的初始值是8005，CRC校验的基本流程：

1. 原始数据与初始值进行异或运算
2. REFIN为TRUE，需要先对原始数据进行翻转
3. 原始数据左移8位，即后面补8个0
4. 处理之后的数据和多项式进行模2除法，求得余数
5. 与XOROUT进行异或
6. 因为REFOUT为TRUE，对结果进行翻转得到最终的CRC-8值
7. 数据+CRC：0011 0100 1101 1111 = 34DF，相当于原始数据左移8位+余数

#### 关键流程

首先下载bootloader的代码，然后将解析好的hex文件通过串口发送到单片机，最后发送1F1F进行中断跳转，跳转到用户程序。

#### 关键实验代码

##### 主函数代码：

void TMR0\_Int(void)

{

uint8\_t num = 0; //define the data length in byte

num = frame[4]/2;

uint16\_t addr;

i = 0;

uint16\_t array[8] ;

uint16\_t CRC; //CRC check

uint16\_t crc\_word = 0; // CRC check contains two bytes.

uint8\_t \*crc\_byte = (uint8\_t \*)&crc\_word;

TMR0\_Reload(0xE9);

TMR0\_StopTimer();

if(frame[0] == 0x00 )

{

if(frame[1] == 0x41)

{

CRC = ((frame[6+num\*2]|0x0000)<<8)| frame[5+num\*2];

crc\_word = crc\_test(frame, 2\*num + 5);

if( crc\_word == CRC )

{

addr = ((frame[2]|0x0000)<<8)|frame[3];

for(int j = 0; j< num; j++)

{

array[j] = ((frame[6+2\*j]|0x0000)<<8)| frame[5+2\*j];

}

for(int j = 0; j<num ; j++ )

{

FLASH\_Write\_a\_word(addr++, array[j]); //write data to designated address

}

}

}

else if( frame[1] == 0x52) //0052, restart.

{

crc\_word = ((frame[3]|0x0000)<<8)| frame[2];

if( (crc\_word) == crc\_test(frame, 2))

{

WDTCON0bits.SEN = 1; //enable watchdog

INTERRUPT\_GlobalInterruptDisable();

INTERRUPT\_PeripheralInterruptDisable();

while(1);

}

}

else if(frame[1] == 0x54) //0054, erase.

{

addr = ((frame[3]|0x0000)<<8)|frame[2];

if(addr >= 0x1000)

{

flag\_erase = 1;

FLASH\_EraseBlock(addr);

}

}

}

if(frame[0] == 0x1F && frame[1] ==0x1F) //1F1F, goto 0x1000

{

crc\_word = ((frame[3]|0x0000)<<8)| frame[2];

if( crc\_word == crc\_test(frame, 2))

{

if(FLASH\_ReadWord(0x1000) != 0x3FFF)

{

asm("MOVLP 0x10");

asm("GOTO 0x00");

}

}

}

}

##### CRC校验代码

uint16\_t crc\_test(uint8\_t \*addr, int num)

{

uint8\_t data;

uint16\_t Data = 0x0000;

uint16\_t crc\_reg = 0xFFFF;

for (; num > 0; num--)

{

data = \*addr++;

data = reverse\_8(data); //

Data = 0x0000;

Data = (Data | data) <<8;

crc\_reg = crc\_reg ^ Data; //

for (int i = 0; i < 8; i++) //cylce

{

if (crc\_reg & 0x8000) //if the highest bit is 1, then left shift and xor with poly

{

crc\_reg = (crc\_reg << 1) ^ polynomial;

}

else //left shift directly

{

crc\_reg <<= 1;

}

}

}

crc\_reg = reverse\_16(crc\_reg);

crc\_reg = crc\_reg ^ 0x0000; //xor with 0x0000

return crc\_reg;

}

#### 实验结果

蜂鸣器以100ms打拍子，同时灯光以100ms的周期闪烁。（课上已经展示）

#### 学到的知识总结

在整个先进处理器课程中，安装软件之后的第一次作业，我学到了PIC单片机计数器的使用，以及如何用计数器控制PWM进行特定占空比的电平输出。之后第二节课学到了memory的使用，了解了如何擦除和写入相关数据。第三节学到了串口的使用。第四节课学习了modbus协议，以及CRC校验等待，最后学到了如何解析hex文件，并通过上位机将HEX文件写入PIC单片机。最后的bootloader的变量和中断也是提高了对于单片机和C语言的许多认知。