//The top of rmcu

#include "stm32f1xx\_hal.h"

#include "top.h"

#include "beep.h"

#include "commu.h"

int appLift;

int AppInit(){

appLift = 0;

TIM3\_Init();

U1\_Init();

U2\_Init();

I2C\_Init();

ADC\_Init();

return 0;

}

int AppMain(){

appLift++;

if((appLift & 0xfffff) > 0x80000)

LED\_103\_GPIO\_Port-> ODR |= LED\_103\_Pin;

else

LED\_103\_GPIO\_Port-> ODR &= (~LED\_103\_Pin);

IWDG->KR = 0xAAAA;

return 0;

}

int U1\_IRQ(){

if((USART1->SR & USART\_SR\_RXNE) == USART\_SR\_RXNE){

U1RecData();

USART1->SR = USART1->SR & (~USART\_SR\_RXNE);

}

return 0;

}

int U1\_IRQ\_END(){

USART1->CR1 = USART1->CR1 | USART\_CR1\_RXNEIE;

return 0;

}

int U2\_IRQ(){

if((USART2->SR & USART\_SR\_TC) == USART\_SR\_TC){

USART2->SR = USART2->SR & (~USART\_SR\_TC);

U2\_SendCon();

}

return 0;

}

int ADC\_Init(void){

ADC1->CR1 = ADC1->CR1 | ADC\_CR1\_JEOCIE;

ADC1->CR2 = ADC1->CR2 | ADC\_CR2\_JEXTTRIG;

ADC1->CR2 = ADC1->CR2 | ADC\_CR2\_ADON;

TIM4->CR1 = TIM4->CR1 | TIM\_CR1\_CEN;

return 0;

}

int ADC\_IRQ(void){

GetADCTemp();

U2\_Send();

return 0;

}

int ADC\_IRQ\_End(void){

ADC1->CR1 = ADC1->CR1 | ADC\_CR1\_JEOCIE;

ADC1->CR2 = ADC1->CR2 | ADC\_CR2\_JEXTTRIG;

ADC1->CR2 = ADC1->CR2 | ADC\_CR2\_ADON;

return 0;

}

//The commu of rmcu

//1-to low level mcu(STM32F407)

//2-to high level raspi

#include "stm32f1xx\_hal.h"

#include "string.h"

#include "types.h"

#include "commu.h"

#define LEN\_U1\_BUF 32

int recU1Number;

U8 recU1Buf[LEN\_U1\_BUF];

int recU1Index;

int sendU2Index;

U8 recI2C\_RecData;

U8 recI2C\_SendData;

float temp\_1,temp\_2,temp\_3,temp\_4;

uint16\_t t1;

int U1\_Init(){

recU1Number = 0;

recU1Index = 0;

memset(recU1Buf,0,LEN\_U1\_BUF\*sizeof(U8));

USART1->CR1 = USART1->CR1 | USART\_CR1\_RXNEIE;

return 0;

}

int U2\_Init(){

sendU2Index = 0;

USART2->CR1 = USART2->CR1 | USART\_CR1\_TCIE;

return 0;

}

int U1RecData(){

recU1Number ++;

U8 recU1Data = USART1->DR;

if((recU1Index == 0) && (recU1Data != 0x50))

return -1;

else if((recU1Index == 1) && (recU1Data != 0x54)){

recU1Index = 0;

return -1;

}

else{

if((recU1Index != 16) && (recU1Index != 17)) //16/17 is tem of F103

recU1Buf[recU1Index] = recU1Data;

recU1Index++;

if(recU1Index == LEN\_U1\_BUF) recU1Index = 0;

return 0;

}

}

int I2C\_Init(){

recI2C\_RecData = 0;

recI2C\_SendData = 0;

I2C1->CR2 = I2C1->CR2 | I2C\_CR2\_ITEVTEN;

I2C1->CR1 = I2C1->CR1 | I2C\_CR1\_ACK | I2C\_CR1\_NOSTRETCH;

return 0;

}

int I2C\_EVIRQ(){

if((I2C1->SR1 & I2C\_SR1\_RXNE) == I2C\_SR1\_RXNE){

recI2C\_RecData = I2C1->DR;

recI2C\_SendData = recU1Buf[recI2C\_RecData];

}

I2C1->DR = recI2C\_SendData;

return 0;

}

int I2C\_EVIRQ\_End(){

I2C1->CR2 = I2C1->CR2 | I2C\_CR2\_ITEVTEN;

I2C1->CR1 = I2C1->CR1 | I2C\_CR1\_ACK;

return 0;

}

int GetADCTemp(){

//temp = (V25 - Vsen ) / Avg\_slope + 25;

//V25 = 1.34; Avg\_slope = 0.0043;

temp\_1 = ADC1->JDR1 / 4096.0f;

temp\_2 = (temp\_1) \* 3.3f;

temp\_3 = 1.34 - temp\_2 ;

temp\_4 = temp\_3 / 0.0043f;

t1 = (U16)((temp\_4 + 25)\*16);

recU1Buf[16] = t1 & 0xff;

recU1Buf[17] = (t1 & 0xffff) >> 8;

return 0;

}

int U2\_Send(){

USART2->DR = recU1Buf[0];

sendU2Index = 1;

return 0;

}

int U2\_SendCon(){

if(sendU2Index < LEN\_U1\_BUF){

USART2->DR = recU1Buf[sendU2Index];

sendU2Index++;

}

return 0;

}

//The top of rmcu

#include "stm32f1xx\_hal.h"

#include "types.h"

#include "beep.h"

int flag;

int TIM3\_Init(){

flag = 0;

TIM3->DIER = TIM3->DIER | TIM\_DIER\_UIE;

TIM3->CR1 = TIM3->CR1 | TIM\_CR1\_CEN;

return 0;

}

//TIM3 IRQ is 4KHZ

int TIM3\_IRQ(){

flag ++;

if(flag %2 == 1)

BEEP\_GPIO\_Port->ODR = BEEP\_GPIO\_Port->ODR | BEEP\_Pin;

else

BEEP\_GPIO\_Port->ODR = BEEP\_GPIO\_Port->ODR & (~BEEP\_Pin);

return 0;

}

#include "stm32f0xx\_hal.h"

#include "base.h"

#include "types.h"

#include "action.h"

#include "led.h"

U8 D5\_Check;

U8 TL,TH;

U16 T\_last;

U16 T\_now;

int t;

int D5OutH(){

D5\_GPIO\_Port->MODER = D5\_GPIO\_Port->MODER | (0x01 << 10);

D5\_GPIO\_Port->ODR = D5\_GPIO\_Port->ODR | D5\_Pin;

return 0;

}

int D5OutL(){

D5\_GPIO\_Port->MODER = D5\_GPIO\_Port->MODER | (0x01 << 10);

D5\_GPIO\_Port->ODR = D5\_GPIO\_Port->ODR & (~D5\_Pin);

return 0;

}

int D5In(){

D5\_GPIO\_Port->MODER = D5\_GPIO\_Port->MODER & (~(0x3<<10));

return 0;

}

int D5Check(){

D5\_Check = D5\_GPIO\_Port->IDR;

if((D5\_Check & D5\_Pin) == D5\_Pin)

return 1;

else

return 0;

}

int D5ReadBit(){

//60us

D5OutL();

Delay1us();

D5In();

Delay10us();

int D5\_Data = D5\_GPIO\_Port->IDR;

DelayN10us(5);

if((D5\_Data & D5\_Pin) == D5\_Pin)

return 1;

else

return 0;

}

int D5WriteBit(int bit){

//60-120us

D5OutL();

Delay10us();

if(bit == 0)

DelayN10us(8);

else{

D5In();

DelayN10us(8);

}

D5In();

DelayN10us(4);

return 0;

}

int D5ReadByte(){

int j;

int q = 0;

for(int i=0;i<8;i++){

j = D5ReadBit();

j=j<<7;

q = (q>>1) | j;

}

return q;

}

int D5WriteByte(int byte){

int d;

int j = byte;

for(int i=0;i<8;i++){

d = (j & 0x1);

j = (j >> 1);

D5WriteBit(d);

}

return 0;

}

int T1\_Reset(){

D5OutL();

DelayN10us(75); //480-960us

D5In();

DelayN10us(4); //15-60us

int ret = D5Check();

DelayN10us(10); //60-240us

return ret;

}

int T1\_Init(){

TL = 0;

TH = 0;

T\_last = 0x1B0;

T\_now = 0x1B0;

T1\_Reset();

return 0;

}

int T1\_GetTemp(){

Led2Glint();

T1\_Reset();

D5WriteByte(0xCC);

D5WriteByte(0x44);

Delay1s();

Delay1s();

T1\_Reset();

D5WriteByte(0xCC);

D5WriteByte(0xBE);

TL = D5ReadByte();

TH = D5ReadByte();

if((TH & 0xff) == 0xff)

T\_now = T\_last;

else {

T\_now = (TH << 8 | TL);

T\_last = T\_now;

}

Led2Glint();

return T\_now;

}

//发现有时间会出现温度显示FFFF

#include "stm32f0xx\_hal.h"

#include "base.h"

int TickBegin;

int TickEnd;

int Tick;

int temp;

int BeginTick(){

TickBegin = SysTick->VAL;

return 0;

}

int EndTick(){

TickEnd = SysTick->VAL;

Tick = TickBegin - TickEnd;

return 0;

}

int Delay1ms(){

for(int i=8000;i>=5;i--)

temp++;

return 0;

}

int Delay5ms(){

for(int i=0;i<5;i++)

Delay1ms();

return 0;

}

int Delay10us(){

for(int i=0;i<66;i++)

temp++;

return 0;

}

int Delay1us(){

for(int i=0;i<6;i++)

temp++;

return 0;

}

int DelayN10us(int n){

for(int i=0;i<n;i++){

Delay10us();

}

return 0;

}

int Delay1s(){

for(int i=0;i<200;i++)

Delay5ms();

return 0;

}

#include "stm32f0xx\_hal.h"

#include "commu.h"

#include "types.h"

#include "string.h"

#define LEN\_BUF 32

U8 SendBuf[LEN\_BUF];

int lenSend;

int indexSend;

int U1\_Init(){

USART1->CR1 = USART1->CR1 | USART\_CR1\_TCIE;

memset(SendBuf,0,LEN\_BUF\*sizeof(U8));

lenSend = 0;

indexSend = 0;

SendBuf[0] = 0x54;

SendBuf[1] = 0x31;

// TP1();

return 0;

}

int U1\_IRQ(){

if((USART1->ISR & USART\_ISR\_TC\_Msk) == USART\_ISR\_TC\_Msk){

if(indexSend <lenSend){

USART1->TDR = SendBuf[indexSend];

indexSend++;

}

USART1->ICR = USART1->ICR | USART\_ICR\_TCCF;

}

return 0;

}

int U1\_Send(){

indexSend = 0;

USART1->TDR = SendBuf[indexSend];

indexSend++;

return 0;

}

int TP1(void){

lenSend = 6;

for(int i=0;i<LEN\_BUF;i++)

SendBuf[i] = 0x30+i;

return 0;

}

int PushBuf(int t){

U8 byte = (t & 0xffff)>>8;

SendBuf[2] = byte;

byte = t & 0xff;

SendBuf[3] = byte;

lenSend = 4;

return 0;

}

#include "stm32f0xx\_hal.h"

#include "led.h"

int stLed1;

int stLed2;

int LedInit(){

stLed1 = 0;

stLed2 = 1;

LED1\_GPIO\_Port->ODR = LED1\_GPIO\_Port->ODR | (LED1\_Pin);

LED2\_GPIO\_Port->ODR = LED2\_GPIO\_Port->ODR | (LED2\_Pin);

return 0;

}

int Led1Glint(){

if(stLed1 == 0){

stLed1 = 1;

LED1\_GPIO\_Port->ODR = LED1\_GPIO\_Port->ODR & (~LED1\_Pin);

}

else {

stLed1 = 0;

LED1\_GPIO\_Port->ODR = LED1\_GPIO\_Port->ODR | (LED1\_Pin);

}

return 0;

}

int Led2Glint(){

if(stLed2 == 0){

stLed2 = 1;

LED2\_GPIO\_Port->ODR = LED2\_GPIO\_Port->ODR & (~LED2\_Pin);

}

else {

stLed2 = 0;

LED2\_GPIO\_Port->ODR = LED2\_GPIO\_Port->ODR | (LED2\_Pin);

}

return 0;

}

//The top of STM32F030 project

#include "stm32f0xx\_hal.h"

#include "top.h"

#include "commu.h"

#include "led.h"

#include "base.h"

#include "action.h"

int T;

int AppInit(){

T = 0x550;

SysTick->LOAD =0xffffff;

U1\_Init();

LedInit();

T1\_Init();

return 0;

}

int AppMain2(){

BeginTick();

U1\_Send();

for(int i=0;i<200;i++)

Delay5ms();

Led1Glint();

EndTick();

IWDG->KR = 0xAAAA;

return 0;

}

int AppMain(){

T = T1\_GetTemp();

PushBuf(T);

U1\_Send();

Delay1s();

IWDG->KR = 0xAAAA;

return 0;

}