

/***** (C) COPYLEFT 2010 Leafgrass *****/

This project runs on uC/OS-II V2.52 and is based on STM32F103x(ARM Cortex-M3 Core)

Take care of the configuration of project. There're some Include Paths while linking and some Defines which cannot be

Procedure of driving a device(IMPORTANT!!!):

1. Connect hardware correctly.
2. Prepare DEVICE_NAME.h and DEVICE_NAME.c.
3. Enable clock of related components.
4. Configure its related registers.
5. Enable related interrupt while configure NVIC.
6. Configure related IO pins.
7. Finish test code and begin testing.

2010-06-10(original):

Brief: Sucessfully port uCOS-II V2.52.

2010-06-11:

Add code "SysTick_CounterCmd(SysTick_Counter_Enable);" in SysTick_Configuration(), solve the problem that t
SysTickHandler(), as below:

```
/* *****  
 * Function Name : SysTick_Config  
 * Description : Configures SysTick  
 * Input : None  
 * Output : None  
 * Return : None  
 * ***** */
```

```
void SysTick_Configuration(void)
```

```
{  
    NVIC_SystemHandlerPriorityConfig(SystemHandler_SysTick, 0, 1);  
    /* Configure HCLK clock as SysTick clock source */  
    SysTick_CLKSourceConfig(SysTick_CLKSource_HCLK_Div8);  
    /* SysTick interrupt each 1000 Hz with HCLK equal to 72MHz */  
    SysTick_SetReload(9000);  
    /* Enable the SysTick Interrupt */  
    SysTick_ITConfig(ENABLE);  
    SysTick_CounterCmd(SysTick_Counter_Enable); //Important!! Solve "the program can't get into SysTickHandle  
}
```

2010-06-12(update to V1.0):

1. Add EXTI8, EXTI9 on PC8, PC9, only for EXTI test.
2. Modify some comments in LCD5110_lib.
3. (in main.c) Modify tasks' priority LED_TASK_Prio 1 -> 5 START_TASK_Prio 2 -> 10
4. (in main.c) Modify code "OSTaskDel(START_TASK_Prio);" to "OSTaskSuspend(START_TASK_Prio);"

2010-07-13

1. Drive LTM022A69B LCD ok. Just use I/O to imitate SPI. Still cannot use internal SPI to drive it.
2. Move LCD library of LTM022A69B into "LCD_lib.c".

2010-07-15:

Add color bar display in LCD_test().

2010-07-16(update to V1.1):

1. Solve problem about position error and color error, when display a set of 16-bit color data. Mainly resulted from big endian or little endian. STM32F103RBT6 is little endian while in default set.
2. Add Draw_img();
3. Add colorful characters and strings, add parameters about colors in display functions.(colors are defined in LTM

2010-07-17:

- 1.Add comments in LCD_Init().
- 2.Add parameter "color" to function Draw_dot().
- 3.Add function SetCursor().
- 4.Unify LCD related functions' name. Add prefix "LCD_".

2010-07-19:

Modify data type in LCD_Lib. normal --> const. At one time, modify the pointers type to const.

2010-07-20:

- 1.Correct the error that OSTimeDlyHMSM() can't delay an accurate time, by modifying SysTick_Configuration() to clock frequency.

2010-07-31:

Add STM32_Init.c but is not referenced, for future use of Configuration Wizard.

2010-08-01:

Configure SPI ok, and do some test(SPI2 tx -> SPI1 rx, display on LCD).

2010-08-03:

- 1.Add SPI test code in main.c, SPI2 send data, SPI1 receive data , display info on LCD, use soft SPI.
- 2.Add exported macro "USESPI" to choose if use SPI to drive LCD or not.
- 3.After several days researching about hard SPI to drive LCD, failed. So switch to use a faster GPIO method, as b
Replace "#define LCD_RST_H() GPIO_SetBits(LCD_CTRL_PORT, LCD_RST)" in ST library
with "#define LCD_RST_H() (LCD_CTRL_PORT->BSRR = LCD_RST)" for a faster LCD refresh speed.
- 4.Modify name "LCD_SCI" to "LCD_SI", 'cause it means Serial Interface.
- 5.Modify function name "LCD_test()" to "LCD_Test()".

2010-08-06:

Comment off "typedef enum {FALSE = 0, TRUE = !FALSE} bool;" in stm32f10x.type.h in order to avoid warning.

2010-08-09:

Prepare to update to v2.0.

2010-08-10(update to v2.0):

Brief: Use ST's FWLib3.1.2.

1. Set project(Keil RealView MDK).
 - (1) RO memory area : IROM1 -- start at 0x08000000, size 0x20000.
 - (2) RW memory area : IRAM1 -- start at 0x20000000, size 0x5000.
 - (3) Output setup.
 - (4) Listing setup, no assembler list.
 - (5) C/C++ setup.
Preprocessor Symbols : Define -- USE_STDPERIPH_DRIVER, STM32F10X_MD.
Include Paths : ..\;
 ..\..\Libraries\CMSIS\Core\CMS3;
 ..\..\Libraries\STM32F10x_StdPeriph_Driver\inc;
 ..\..\App;
 ..\..\Driver;
 ..\..\OS;
 ..\..\OS\port
- (6) Linker, Use memory layout from target dialog.
- (7) Debug and Utilities, Cortex-M/R J-Link/J-Trace. Disable Trace.
2. Modify code in "includes.h".
3. Modify startup code, did not use the st original one. Port those used in FWLib2.0 instead.
Of course, some code are revised for FWLib3.1.2(StdPeriph_Driver and CMSIS).
Mainly rewrite the names of the ISR.

The most important two are : "PendSV_Handler" and "SysTick_Handler".

---> "OSPendSV" and "SysTick_Handler".

(in "os_cpu_a.asm") (in "stm32f10x_it.c")

4. Modify initial code for SysTick in "SysTick_Configuration()" in "CPU_Init.c" because of the changing of library.

Note : A general problem is that program always go into "B OSStartHang" in "OSStartHighRdy" in "os_cpu_a.asm" and will die in there... That's caused by program hasn't step into the correct ISR yet.

Enable the SysTick interrupt, check if the name of Interrupt Service Routines are right, then all will be OK.

5. Still some problem with retargeting "printf()". If use "printf()" in program, system wouldn't run.

6. Change Project name to "RTOS_STM32".

2010-08-11(update to v2.1):

Brief: Use ST's FWLib3.3.0.

1. Modify Include Path in Target Options.

2. Comment "typedef enum {FALSE = 0, TRUE = !FALSE} bool;" off in order to avoid error:

"..\stm32f10x.h(454): error: #40: expected an identifier".

2010-08-12(update to v2.2):

1. Remove the incorrect use of SPI in LCD display, add a new schedule(maily used SPI busy flag).

2. Commen off code "assert_param(IS_SPI_ALL_PERIPH(SPIx));" in "SPI_I2S_SendData()",

"SPI_I2S_ReceiveData()",

and "SPI_I2S_GetFlagStatus" in "stm32f10x_spi.h",

in order to improve LCD refresh frequency.

3. Finish to retarget C library printf() to UART.

As a foundation, change the setup of KEIL in "Target Options" --> "Code Generation" --> tick "Use MicroLIB".

There are still some problem with building, modify a few lines in "stm32f10x_startup.s" :

=====

IF :DEF: __MICROLIB

EXPORT __initial_sp

EXPORT __heap_base

EXPORT __heap_limit

ELSE

IMPORT __use_two_region_memory

EXPORT __user_initial_stackheap

=====

Then rebuild.

Add retarget code in main.c :

=====

#ifdef __GNUC__

// With GCC/RAISONANCE, small printf

// (option LD Linker->Libraries->Small printf set to 'Yes')

// calls __io_putchar()

#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)

#else

#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)

#endif // __GNUC__

PUTCHAR_PROTOTYPE

{

// Place your implementation of fputc here

// e.g. write a character to the USART

USART_SendData(USARTx, (u8)ch);

// Loop until the end of transmission

while (USART_GetFlagStatus(USART1, USART_FLAG_TC) == RESET)

{}

=====

1. Rewrite the different part of code into its C file.
2. Drive TIM3 output PWM wave successfully.
3. Add "servo" file, for servo control.

1. Add "gps" file.
2. Add code in "USART3_IRQHandler()", mainly to receive data from GPS. But there's still something confusing, that the pair "USART_ITConfig(USART3, USART_IT_RXNE, DISABLE);" and "USART_ITConfig(USART3, USART_IT_RXNE, ENABLE);" cannot work well. Because once use them, some data will be lost.

Refresh firmware of GPS, modify to Baudrate 9600, China timezone, WGS-84 coordinate.

Replace those two old startup files "cortexm3_macro.s" and "stm32f10x_startup.s" with the one "startup_stm32f10x" which is created by MCD Application Team in FWLib3.3.0.

1. Add `general_task.h` and `.c` for miscellaneous experimental purpose.
2. Add `uart` output information in booting stage.
3. Add macro `"LED_TOGGLE()"` use `"(GPIO_ReadOutputDataBit(GPIOA, GPIO_Pin_1))"`
4. Create Logo:

```

      _/ /
  _/ / _/ / _/ / _/ / _/ / _/ / _/ / _/ / _/ / _/ /
  | | / \ _/ / _/ / _/ / _/ / _/ / _/ / _/ / _/ /
  | | _/ / ( ( | | | | ( ( | | | | ( ( | | \ \ \ \
  | | _/ / \ \ \ \ | | | | \ \ \ \ / / | | \ \ \ \ / / / /
  | | _/ / \ \ \ \ / / \ \ \ \

```

setup procedure: GPIO IN FLOATING -> NVIC IRQ -> EXTI Config -> _it.c

1. Modify GPIOC Pin6 mode to "GPIO_Mode_IPU"(input with pull-up) for EXTI Line6 test.
2. Add LED4 control macro at PD2.
3. ADC1 with DMA test OK. Related code are placed in file "general_task".(Be careful of input: VREF- <= VIN <= V)
4. The same time internal temperature sensor test OK.

fatfs 0.07e test ok for itself, but some error occurred during linking in this project if add app code(SD_Test()) in "ge

1. Find the reason that some error occurred during linking is about RAM overflow.
A low class mistake caused by the "wbuff2[512*26]" being so large and stored as ZI data in RAM, therefore....
Just decrease the size, modify it to "512*10".
2. Another error occurred, while running fatfs test, it always goes into "HardFault_Handler()", currently do not know why.
If comment off those w/r related wbuff2 which cause Hard Fault, fatfs works well. It is able to recognize SD card.
Almost test fatfs at ucos-ii ok, but sometimes Hard Fault stills occurred.

Brief: FatFs v0.07e along with uCOS-II v2.52.

1. FatFs test ok in uCOS-II. Fix the Hard Fault bug while w/r files. Just increase the stack size of this task in "task_GENERAL_STK_SIZE 64 -> 1024
(OH MY! What a shame! It takes me large amount of time to find it out.... - -!)
2. Add LED3 blink into SD operation in "__inline u8 _spi_read_write(u8 data)" of SD_Driver.c.
3. Move "ADC_SoftwareStartConvCmd(ADC1, ENABLE);" from general_task to CPU_Init(ADC_Configuration()).

2010-09-25

Test Software Reset by write to SCB_AIRCR(System Control Block_Application Interrupt and Reset Control Register)
(added to test in "EXTI9_5_IRQHandler(void)")

2010-11-02

Add another task "TaskLed3" just for fun.

2010-11-09

1. Create another target. Add device STM32F103VC to project. Now there are two targets:
STM32_DQ_128K_64P, STM32_CORE_512K_100P
2. Add platform_conf.h.

2010-11-10(update to v3.0.0)

1. Add if "#define CHECK_INSERT == TRUE" in "SD_Driver.h" for SD(TRUE) or TF card.
On STM32_MAIN board, there's no insertion checking.
2. Begin to use two bits of decimal for version control.

2011-11-12(v3.0.1)

1. Add function "Serial_Echo()" in "general_task.c".

/***** (C) COPYLEFT 2010 Leafgrass *****/