



STM32F429 開発ボード マニュアル

株式会社日昇テクノロジー

http://www.csun.co.jp

info@csun.co.jp

作成日 2014/9/22

copyright@2014



• 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2014/9/22

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。
 最新版は弊社ホームページからご参照ください。「http://www.csun.co.jp」
 ※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に
 禁じられています。



1.	概述	. 8
	1.1 STM32F429 チップの主な仕様	. 8
	1.2 開発ボードの主な仕様	. 9
2.	Hardware layout and configuration	10
	2.1 Power supply	11
	2.2 Boot option	11
	2.3 Clock source	11
	2.4 Reset source	11
	2.5 Audio	11
	2.6 EEPROM	12
	2.7 CAN	12
	2.8 RS-232	12
	2.9 RS-485	12
	2.10 SD card	13
	2.11 Potentiometer	13
	2.12 USB OTG FS	13
	2.13 Ethernet	13
	2.14 Camera	14
	2.15 SDRAM	15
	2.16 NAND Flash	15
	2.17 Debug support	15
	2.18 FSMC Port	15
	2. 19 TFT LCD	16
	2.20 Button	17
	2. 21 LED	18
3.	ソフトウェア説明	19
	3.1 Keil コンパイル環境	19
	3.1.1 コンパイル環境の構築	19
	3.1.2 コンパイル環境の設定	19
	3.2 DFU ソフトウェア	23
	3.2.1 DFU File Manager	23
	3 2.2 DfuSe Demonstration	25
4.	サンブルソースの説明	28
	4. 1 ¥Code¥STM32F4xx_DSP_StdPeriph_Lib_V1. 3. 0¥Project フォルダのプログラムの説明	28
	4.1.1 ¥DAC_SignalsGeneration¥MDK-ARM	28
	4. 1. 2 ¥DMA_FLASHTORAM¥MDK-ARM	30
	4.1.3 ¥EXTI_Example¥MDK-ARM	30
	4.1.4 ¥FLASH_Program¥MDK-ARM	30
	4.1.5 ¥12C_10Expander¥MDK-ARM	31
	4.1.6 ¥1WDG_Example¥MDK-AKM	31
	4.1.7 ¥NV1C_1KQMask¥MDK-AKM	31
	4.1.8 ¥NVIC_IKQPriority¥MDK-AKM	32
	4.1.9 ¥ NVIC_WFIMOde¥MUK-AKM	32
	4.1.10 ¥PWK_STANDBY¥MDK-AKM	32



	4.1.11	¥PWR_STOP¥MDK-ARM	33
	4.1.12	¥PWR_BOR¥MDK-ARM	33
	4.1.13	¥RNG_MultiRNG¥MDK-ARM	33
	4.1.14	<pre>¥SysTick_Example¥MDK-ARM</pre>	34
	4.1.15	¥TIM_TimeBase ¥MDK-ARM YMDK-ARM YmmeDase	34
	4.1.16	¥TIM_PWMOutput¥MDK-ARM	35
	4.1.17	¥TIM_CascadeSynchro¥MDK-ARM	38
	4.1.18	¥TIM_DMA¥MDK-ARM	39
	4.1.19	¥TIM_DMABurst¥MDK-ARM	41
	4.1.20	<pre>¥TIM_OCToggle¥MDK-ARM</pre>	42
	4.1.21	¥TIM_7PWMOutput¥MDK-ARM	43
	4.1.22	¥WWDG_Example¥MDK-ARM	45
	4.1.23	¥LTDC_Display_1Layer¥MDK-ARM	45
	4.1.24	¥LTDC_Display_2Layer¥MDK-ARM	45
	4.1.25	¥LTDC_ColorKeying¥MDK-ARM	46
	4.1.26	¥LTDC_PicturesFromSDCard¥MDK-ARM	47
	4.1.27	¥LTDC_AnimatedPictureFromSDCard¥MDK-ARM	48
	4.1.28	¥DMA2D_MemToMemWithLCD¥MDK-ARM	48
	4.1.29	¥DMA2D_RegToMemWithLCD¥MDK-ARM	49
	4.1.30	¥DMA2D_MemToMemWithPFC¥MDK-ARM	49
	4.1.31	¥DMA2D_MemToMemWithBlending¥MDK-ARM	50
4.2	¥Code¥	STemWin_Library_V1.1.1¥Project¥STM324 <mark>x</mark> 9I-EVAL フォルダのプログラム説明	51
	4.2.1	¥Standalone¥MDK-ARM	51
	4.2.2	¥RTOS¥MDK-ARM	54
4.3	¥Code¥	STM32F4x9_ETH_LwIP_V1.1.0¥Project フォルダのプログラム説明	58
	4.3.1	¥httpserver¥MDK-ARM	58
	4.3.2	¥tftpserver¥MDK-ARM	61
	4.3.3	<pre>¥tcp_echo_server¥MDK-ARM</pre>	63
	4.3.4	<pre>¥udp_echo_server¥MDK-ARM</pre>	65
	4.3.5	<pre>¥tcp_echo_client¥MDK-ARM</pre>	67
	4.3.6	¥udp_echo_client¥MDK-ARM	68
	4.3.7	Yhttpserver_netconnYMDK-ARM	69
	4.3.8	¥httpserver_socket¥MDK-ARM	71
	4.3.9	Yudptcp_echo_server_netconnYMDK-ARM	71
4.4	¥Code¥	STM324x9I-EVAL¥Project フォルダのプログラム説明	73
	4.4.1	¥Blinky	73
	4.4.2	¥Blinky_ADC	73
	4.4.3	¥RTX_Blinky	74
	4.4.4	¥SD_File	74
	4.4.5	¥FlashFS¥NAND_File	75
	4.4.6	¥USB_Device_Memory	77
	4.4.7	¥USB_Device_HID	77
	4.4.8	Example 2 SD_Memory	79
	4.4.9	¥USB_Device_NAND_Memory	80
	4.4.10	¥USB_Host_HID_COM	80



	4.4.11	<pre>¥USB_Host_MSD_File</pre>	81
	4.4.12	¥Http_demo	84
	4.4.13	¥Http_upload	91
	4.4.14	¥Telnet_demo	94
	4.4.15	¥DNS_demo	97
	4.4.16	¥FTP_demo	98
	4.4.17	¥SNMP_demo	. 100
	4.4.18	¥LEDSwitch	. 102
	4.4.19	¥BSD_server	. 102
	4.4.20	¥LEDClient	. 103
	4.4.21	¥BSD_client	. 105
4.5	¥Code ⁴	¥STM32F4x9_uCOSIII¥Project フォルダのプログラム説明	. 106
	4.5.1	¥LED_Blinky	. 106
	4.5.2	¥Binary_Semaphores	. 107
	4.5.3	¥Binary_Semaphores_Interrupt	. 107
	4.5.4	¥Mutex	. 107
	4.5.5	¥Mutex_LCD	. 108
4.6	¥Code ⁴	¥STM32Cube_FW_F4_V1.1.0¥Projects¥STM324x91_EVAL¥Examples フォルダのプログラム	ム説
明			. 109
	4.6.1	¥GPIO¥GPIO_IOToggle¥MDK-ARM	. 109
	4.6.2	¥GPIO¥GPIO_EXTI¥MDK-ARM	. 109
	4.6.3	¥ADC¥ADC_TriggerMode¥MDK-ARM	. 109
	4.6.4	¥ADC¥ADC_TripleModeInterleaved¥MDK-ARM	. 110
	4.6.5	¥ADC¥ADC_RegularConversion_Polling¥MDK-ARM	. 111
	4.6.6	¥ADC¥ADC_RegularConversion_Interrupt¥MDK-ARM	. 112
	4.6.7	¥ADC¥ADC_RegularConversion_DMA¥MDK-ARM	. 112
	4.6.8	¥ADC¥ADC_DualModeInterleaved¥MDK-ARM	. 112
	4.6.9	¥CAN¥CAN_LoopBack¥MDK-ARM	. 113
	4.6.10	YCAN¥CAN_Networking¥MDK-ARM	. 113
	4.6.11	<pre>¥Cortex¥CORTEXM_SysTick¥MDK-ARM</pre>	. 113
	4.6.12	¥Cortex¥CORTEXM_MPU¥MDK-ARM	. 114
	4.6.13	¥CRC¥CRC_Example¥MDK-ARM	. 114
	4.6.14	¥DAC¥DAC_SignalsGeneration¥MDK-ARM	. 114
	4. 6. 15	¥DMA¥DMA_FLASHToRAM¥MDK-ARM	. 115
	4.6.16	¥DMA¥DMA_FIFOMode¥MDK-ARM	. 115
	4.6.17	¥DMA2D¥DMA2D_MemoryToMemory¥MDK-ARM	. 116
	4.6.18	¥DMA2D¥DMA2D_MemToMemWithBlending¥MDK-ARM	. 116
	4.6.19	¥DMA2D¥DMA2D_MemToMemWithPFC¥MDK-ARM	. 116
	4.6.20	¥DMA2D¥DMA2D_MemToMemWithLCD¥MDK-ARM	. 117
	4.6.21	¥DMA2D¥DMA2D_RegToMemWithLCD¥MDK-ARM	. 118
	4.6.22	¥FLASH¥FLASH_EraseProgram¥MDK-ARM	. 118
	4.6.23	¥FLASH¥FLASH_WriteProtection¥MDK-ARM	. 118
	4.6.24	¥FMC¥FMC_SDRAM¥MDK-ARM	. 119
	4.6.25	¥IWDG¥IWDG_Example¥MDK-ARM	. 120
	4.6.26	<pre>¥LTDC¥LTDC_ColorKeying¥MDK-ARM</pre>	. 120



株式会社日昇テ	クノロジー
---------	-------

4.6.27 ¥ITDC¥ITDC Display 11 over¥MDK-APM	191
4. 6. 28 ¥I TDCYL TDC Display 21 avors¥MDK ARM	
4. 6. 20 YEAVER STANDRYYMDK-ARM	
4. 6. 20 YDWRYDWR STOPYMNK-ARM	
4. 0. 30 FIWATIWA_STOLEMDA AAM	122
4. 0. 51 FI WATI WA_CUITENTCONSUMPTIONTMOX AAM	122
4. 0. 52 FRUEFRUE_LIOCKUOIIIIgFMDR-ARM	122
4. 0. 35 +KIC+KIC_AIAIIIHMDK-ARM	123
4. 0. 54 FRICFRIC_TAMPETFMDK_ARM	123
4. 0. 35 FRICFRIC_CATENDATFMDK ADM	123
4. 0. 30 FRICFRIC_TIMEStampFMDR-ARM	124
4. 6. 37 ¥IIM¥IIM_IImeBase¥MDK-ARM	
4. 6. 38 ¥IIM¥IIM_6Steps¥MDK-ARM	124
4.6.39 ¥IIM¥IIM_7PWMOutput¥MDK-ARM	126
4.6.40 ¥TIM¥TIM_CascadeSynchro¥MDK-ARM	127
4. 6. 41 ¥TIM¥TIM_DMA¥MDK-ARM	129
4.6.42 ¥TIM¥TIM_DMABurst¥MDK-ARM	130
4.6.43 ¥TIM¥TIM_ExtTriggerSynchro¥MDK-ARM	131
4.6.44 ¥TIM¥TIM_InputCapture¥MDK-ARM	132
4.6.45 ¥TIM¥TIM_OCToggle¥MDK-ARM	133
4.6.46 ¥TIM¥TIM_OnePulse¥MDK-ARM	134
4.6.47 ¥TIM¥TIM_ParallelSynchro¥MDK-ARM	134
4.6.48 ¥TIM¥TIM_Prescaler_Selection¥MDK-ARM	135
4.6.49 ¥TIM¥TIM_PWMIntput¥MDK-ARM	136
4.6.50 ¥TIM¥TIM_PWMOutput¥MDK-ARM	137
4.6.51 ¥WWDG¥WWDG_Example¥MDK-ARM	138
4.7 ¥Code¥STM32Cube_FW_F4_V1.1.0¥Projects¥STM324x9I_EVAL フォルダのプログラム	、説明138
4.7.1 ¥Applications¥Display¥LTDC_AnimatedPictureFromSDCard¥MDK-ARM	138
4.7.2 ¥Applications¥Display¥LTDC_PicturesFromSDCard¥MDK-ARM	139
4.7.3 ¥Applications¥Display¥LTDC_Paint¥MDK-ARM	139
4.7.4 ¥Applications¥FatFs¥FatFs_RAMDisk¥MDK-ARM	141
4.7.5 \# Applications#FatFs#FatFs_RAMDisk_RTOS\MDK-ARM	141
4.7.6 ¥Applications¥FatFs¥FatFs_uSD¥MDK-ARM	142
4.7.7 ¥Applications¥FatFs¥FatFs_uSD_RTOS ¥MDK-ARM	142
4.7.8 ¥Applications¥FatFs¥FatFs_USBDisk¥MDK-ARM	142
4.7.9 ¥Applications¥FatFs¥FatFs_USBDisk_RTOS ¥MDK-ARM	143
4.7.10 ¥Applications¥FatFs¥FatFs_USBDisk_MultipleAccess_RTOS¥MDK-ARM	143
4.7. M ¥Applications¥FatFs¥FatFs_MultiDrives¥MDK-ARM	144
4.7.12 ¥Applications¥LwIP¥LwIP_HTTP_Server_Raw¥MDK-ARM	144
4.7.13 ¥Applications¥LwIP¥LwIP_HTTP_Server_Netconn_RTOS¥MDK-ARM	147
4.7.14 ¥Applications¥LwIP¥LwIP_HTTP_Server_Socket_RTOS¥MDK-ARM	149
4.7.15 ¥Applications¥LwIP¥LwIP_TCP_Echo_Server¥MDK-ARM	149
4.7.16 ¥Applications¥LwIP¥LwIP_UDP_Echo_Server¥MDK-ARM	151
4.7.17 ¥Applications¥LwIP¥LwIP_TCP_Echo_Client¥MDK-ARM	153
4.7.18 ¥Applications¥LwIP¥LwIP_UDP_Echo_Client¥MDK-ARM	155
4.7.19 ¥Applications¥LwIP¥LwIP_UDPTCP_Echo_Server_Netconn_RTOS¥MDK-ARM	156



4.7.20	¥Applications¥LwIP¥LwIP_TFTP_Server¥MDK-ARM 158
4.7.21	¥Applications¥LwIP¥LwIP_IAP¥MDK-ARM 159
4.7.22	¥Applications¥STemWin¥STemWin_HelloWorld¥MDK-ARM
4.7.23	¥Applications¥STemWin¥STemWin_SampleDemo¥MDK-ARM 164
4.7.24	¥Applications¥USB_Device¥HID_Standalone¥MDK-ARM165
4.7.25	¥Applications¥USB_Device¥CustomHID_Standalone¥MDK-ARM
4.7.26	¥Applications¥USB_Device¥CDC_Standalone¥MDK-ARM168
4.7.27	¥Applications¥USB_Device¥MSC_Standalone¥MDK-ARM170
4.7.28.	¥Applications¥USB_Device¥DFU_Standalone¥MDK-ARM 170
4.7.29	¥Applications¥USB_Host¥HID_Standalone¥MDK-ARM174
4.7.30	¥Applications¥USB_Host¥HID_RTOS¥MDK-ARM 177
4.7.31	¥Applications¥USB_Host¥MSC_Standalone¥MDK-ARM 177
4.7.32	¥Applications¥USB_Host¥MSC_RTOS¥MDK-ARM 179
4.7.33	¥Applications¥USB_Host¥AUDI0_Standalone¥MDK-ARM 180
4.7.34	¥Applications¥USB_Host¥DynamicSwitch_Standalone¥MDK-ARM. 181
4.7.35	Applications¥USB_Host¥FWupgrade_Standalone¥MDK-ARM 187
4.7.36	¥Applications¥FreeRTOS_ThreadCreation¥MDK-ARM.189
4.7.37	¥Applications¥FreeRTOS¥FreeRTOS_Timers¥MDK-ARM189
4.7.38	¥Applications¥FreeRTOS¥FreeRTOS_Semaphore¥MDK-ARM 189
4.7.39	¥Applications¥FreeRTOS¥FreeRTOS_SemaphoreFromISR¥MDK-ARM
4.7.40	¥Applications¥FreeRTOS¥FreeRTOS_Mutexes¥MDK-ARM190
4.7.41	¥Applications¥FreeRTOS¥FreeRTOS_Queues¥MDK-ARM 191
4.7.42	¥Applications¥FreeRTOS¥FreeRTOS_LowPower¥MDK-ARM 191
4.7.43	¥Applications¥Camera To_USBDisk¥MDK-ARM 191





1. 概述

1.1 STM32F429 チップの主な仕様

≻Core:

ARM 32-bit Cortex[™]-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator[™]) allowing 0-wait state execution from Flash memory, frequency up to 180 MHz, MPU, 225 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions

 \succ Memories

·Up to 2 MB of Flash memory organized into two banks allowing read-while-write

·Up to 256+4 KB of SRAM including 64-KB of CCM (core coupled memory) data RAM

·Flexible external memory controller with up to 32-bit data bus: SRAM, PSRAM, SDRAM, Compact Flash/NOR/NAND memories

►LCD parallel interface

 $\cdot 8080/6800$ modes

≻ LCD-TFT

controller up to SVGA resolution with dedicated Chrom-ART Accelerator™ for enhanced graphic content creation (DMA2D)

► Clock, reset and supply management

 $\cdot 1.8$ V to 3.6 V application supply and I/Os

·POR, PDR, PVD and BOR

·4-to-26 MHz crystal oscillator

·Internal 16 MHz factory-trimmed RC (1% accuracy)

- ·32 kHz oscillator for RTC with calibration
- ·Internal 32 kHz RC with calibration

·Sleep, Stop and Standby modes

·VBATsupply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM

 $> 3 \times 12$ -bit, 2.4 MSPS ADC:

up to 24 channels and 7.2 MSPS in triple interleaved mode

 $> 2 \times 12$ -bit D/A converters

≻ General-purpose DMA:

16-stream DMA controller with FIFOs and burst support

≻Up to 17 timers:

up to twelve 16-bit and two 32-bit timers up to 180 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input

≻Debug mode

·Serial wire debug (SWD)& JTAG interfaces

·Cortex-M4 Embedded Trace Macrocell™

> Up to 168 I/O ports with interrupt capability

•Up to 164 fast I/Os up to 90 MHz

·Up to 166 5 V-tolerant I/Os

> Up to 21 communication interfaces

·Up to 3 \times I2C interfaces (SMBus/PMBus)

·Up to 4 USARTs/4 UARTs (11.25 Mbit/s, ISO7816 interface, LIN, IrDA, modem control)



·Up to 6 SPIs (45 Mbits/s), 2 with muxed full-duplex I2S for audio class accuracy via internal audio PLL or external clock

·1 x SAI (serial audio interface)

 $\cdot 2$ $\,\times\,$ CAN (2.0B Active) and SDIO interface

➤ Advanced connectivity

·USB 2.0 full-speed device/host/OTG controller with on-chip PHY

 $\cdot \text{USB}$ 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI

 $\cdot 10/100$ Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII

≥ 8- to 14-bit parallel camera interface up to 54 MBs/s

≻ Cryptographic acceleration: hardware acceleration for AES 128, 192, 256, Triple DES, HASH (MD5, SHA-1, SHA-2), and HMAC(Only STM32F439)

≻ True random number generator

 \succ CRC calculation unit

≻96-bit unique ID

▶ RTC: subsecond accuracy, hardware calendar

1.2 開発ボードの主な仕様

▶産業用のデバイス部品を利用

▶128M Bit SDRAM(micron、データ幅は16ビット)

≻4G Bit NAND Flash(micron)

≻IIC EEPROM

▶20 ピン 2.54mm ピッチ JTAG インタフェーズ

>10/100M イーサネットインターフェイス (MII、RMII 二種類設定モード、物理層チップは DP83848)

▶二つ USB2.0 フルスピードホストとスレーブデバイスインターフェース

▶二つ USB2.0 フルスピードホストデバイスインターフェース

▶1 チャネル RS232 インターフェース

>1 チャネル RS485 インターフェース (DB9 ソケットの裏面に)

>2 チャネル CAN2.0 インターフェース (backup、USB と共用)

>SDメモリーカードスロット(4ビット、USBソケットの裏面に)

➤データ幅が8ビット及び12ビットのDCMI インターフェース(ピンヘッダー、IIC ピンが含まれる)、CMOS カメラモジュール(0V2640、200 万画素)と直結出来る

▶多機能オーディオインターフェース、一般的な RF モジュールを接続できる(ピンヘッダー、 SPI、SAI、 IIC のピンを含む)

▶16 ビットバス拡張インターフェース、6800/8080 の LCD を接続できる

▶24 ビット RGB LCD インタフェース、RGB888 (24-bits)利用の 4.3 インチ液晶 (TFT タッチパネル付き) 搭 載

▶制御可能 LED 一つ

▶4つのボタン、その中のに一つはウェークアップ機能付きボタン

▶調節可能抵抗一つ

▶RTCに給電する 0.33F コンデンサ(実装してない、スペース保留)



2. Hardware layout and configuration







2.1 Power supply

開発ボードは DC 5V で給電する。下記3つの給電方法がある:

- J10 の DC-2.1 ソケットを通じて外部 5V 電源で給電(内側が+、外側が-)
- J6 の Mini USB (フルスピード) インタフェースで給電
- J7 の Mini USB (フルスピード) インタフェースで給電

2.2 Boot option

三つの起動モードがあり:

- ユーザプログラムスペースから起動
- ISP ブートローダー (Boot loader) スペースから起動
- 内部 SRAM スペースから起動

```
上記の起動モードはボードのSW1、SW2により設定できる。
```

表1 BOOTの関連ジャンパー

BOOT O(SW1)	Boot1(SW2)	起動モード
0	無視	ユーザプログラムスペースから起動
1	1	内部 SRAM スペースから起動
1	0	ISP ブートローダースペースから起動

2.3 Clock source

四つのクロックソースがあり:

- Y1、32.768K STM32F4x9IG チップのRTC 水晶発振
- Y2、25MHz STM32F4x9IG チップのメイン周波数発振
- ●Y3、50MHz ネットワークPHYアクティブ発振、RMIIモードに使用される
- ●Y4、25MHz ネットワークPHY パッシブ発振、MII モードに使用される

2.4 Reset source

開発ボードのリセット信号はローレベル、リセット方法は下記種類があり:

- RST キー
- JTAG インタフェース Pin-15
- JP5 Pin-23
 JP2 Pin-36

2.5 Audio

開発ボードの JP4 ピン配列に SAI (Serial audio interface)、SPI、IIC 機能があるので、オーディオモ ジュール及び <u>NRF24L01 モジュール</u>を接続できる。

表2	JP4のピン定義

ピン位置	ピンの名前	オーディオモジュール	NRF24L01 モジュール
1	GND	GND	GND
2	+3.3V	+3. 3V	+3.3V
3	PI3	I2S_SD	RF_CE



4	PF6	SAI1_SDB	SPI5_NSS
5	PF7	SAI1_MCLKB	SPI5_SCK
6	PF9	SAI1_FSB	SPI5_MOSI
7	PF8	SAI1_SCKB	SPI5_MISO
8	PD6	I2S3_SD	RF_IRQ
9	PH5	I2C_SDA	I2C_SDA
10	PH4	I2C_SCL	I2C_SCL
11	PB3	I2S3_CK	
12	PD7		

2.6 EEPROM

開発ボードに一枚の I2C バズ EEPROM (I2C2 利用)を搭載。

2.7 CAN

開発ボードには2 チャネルの CAN2. 0A/B 標準の CAN バスがある。2 チャネル CAN バスのピンと2 チャネルの USB ピンが重用され、デフォルトの場合に、USB 機能を実現させる。CAN 機能を実現するために、R63、R64、R65、 R66 に抵抗を半田付ける必要。

表3 CAN 関連抵抗の設定

ピンの名前	CAN	USB
PA11	CAN1_RX、R63 に 1K 抵抗をハンダ付け、CAN	USB1_FS_DM
	機能を実現させる	
PA12	CAN1_TX、R64 に 1K 抵抗をハンダ付け、CAN	USB1_FS_DP
	機能を実現させる	
PB12	CAN2_RX、R65 に 1K 抵抗をハンダ付け、CAN	USB2_FS_ID
	機能を実現させる	
PB13	CAN2_TX、R66に IK 抵抗をハンダ付け、CAN	USB2_FS_VBUS
	機能を実 <mark>現</mark> させる	

CAN バスは固定高速モードに設定される (TJA1050の八番目のピンにより設定)

2.8 RS-232

開発ボードのRS232 インタフェースに使用された I/O が USB1 の ID、VBUS ピンと共用される。シリアルポートと USB1 機能が同時に使用するときに、USB1 の ID、VBUS ピンの設定を(ソフトウェアで)マスクする必要がある。

表4 UART1 と USB1 ピン定義

ピンの名前 UART		USB
PA9	UART1_TX	USB1_FS_VBUS
PA10	UART1_RX	USB1_FS_ID

2.9 RS-485

開発ボードの RS485 インタフェースに使用された I/O が DCMI ピンと共用される。RS485_A と RS485_B ピン は SMD パッドの形で D89 ソケットの裏面に置く。PD7 でデータの受送信を制御し、PD7 がローレベルの場合



に、RS485 チップが受信モードになる。PD7 がハイレベルの場合に、RS485 チップが送信モードになる。 *表5 UART6 と DCMI ピン定義*

ピンの名前	ピンの名前 RS485 カメラ	
PC6	UART6_TX	DCMI_DO
PC7	UART6_RX	DCMI_D1

2.10 SD card

開発ボードは 4bit の SDIO インタフェースで SD カードを接続し、I/O PI8 により SD カードが挿入されて いるかを検測する。SDIO に使用された I/O と DCMI インタフェースが共用するので、同時に両方の機能を実 現できない。

ピンの名前	SD カード	カメラ
PC8	SDIO_DO	DCMI_D2
PC9	SDIO_D1	DCMI_D3
PC10	SDIO_D2	DCMI_D8
PC11	SDIO_D3	DCMI_D4
PC12	SDIO_CK	DCMI_D9
PD2	SDIO_CMD	DCMI_D11

2.11 Potentiometer

開発ボードに10kの調節可能抵抗があり、ADC1/2の五番目のチャンネルと接続し、対応するデバイスは R58である。

2.12 USB OTG FS

開発ボードに四つのフルスピードUSBインタフェースがあり、その中に、Mini USB AB型のソケットが二つ、これらは通信接続機能があり、かつ開発ボードに 5V 500mA 電流を提供する。USB A 型のソケットが二つ、マウス、キーボード、USBメモリなどの USB 周辺デバイスを接続する。

Mini USB AB型のソケットと隣のUSB A型のソケットは物理的に接続しているので、同時に両方を使えない。

2.13 Ethernet

ネットワーク PHY チップは DP83848IVV で、統合されたフィルタ RJ45 のネットワークインターフェースを 使用し、10/100M アダプティブ方式のネットワーク接続を実現する。J1 キャップを設定することにより、 ネットワークが MII と RMII モードに設定される。MII モードにネットワーク PHY チップは、25M パッシブ発 振を使って多くの I/0を使う。RMII モードに PHY チップは、50MHz アクティブ発振を使って、比較的に少 ない I/0を占有する。デフォルトでは MII モードに設定される。

通信モード	J1	JP1
MII	••	$\bullet \bullet \bullet$
		$\bullet \bullet \bullet$
RMII		
		$\bullet \bullet \bullet$

表7 ネットワーク設定モード



MII モードに使われた I/ 0



PMII モードに使われた I/ 0





2.14 Camera

JP3 はカメラインタフェースで、8ビット及び12ビットデータ幅のモジュールを接続できる。例えば、 次の図に Pin1-Pin16の所に(赤いボックス)8ビットデータ幅のカメラモジュールが接続できる。それに 対して、12ビットデータ幅のカメラモジュールが接続するときに、JP3のすべてのピンが使用されている。 CMOS カメラモジュール(0V2640、200万画素)(8ビットデータ幅タイプ)と直結出来る。



表8 JP3 ビン定義					
ピンの仕里	ピンの名前	8ビットのカメラモジ	12 ビットのカメラモジ		
ビンの位置		ュール	ュール		
1	PH5	I2C_SDA	I2C_SDA		
2	+5V	+5V	+5V		
3	PB7	DCMI_VSYNC	DCMI_VSYNC		
4	PH4	I2C_SCL	I2C_SCL		
5	PB9	DCMI_D7	DCMI_D7		
6	PB8	DCMI_D6	DCMI_D6		
7	PD3	DCMI_D5	DCMI_D5		



8	PB8	DCMI_D6	DCMI_D6
9	PC11	DCMI_D4	DCMI_D4
10	PA6	DCMI_PCLK	DCMI_PCLK
11	PC9	DCMI_D3	DCMI_D3
12	PC6	DCMI_DO	DCMI_DO
13	GND	GND	GND
14	GND	GND	GND
15	PC7	DCMI_D1	DCMI_D1
16	PC8	DCMI_D2	DCMI_D2
17	PC10		DCMI_D8
18	PC12		DCMI_D9
19	PD6		DCMI_D10
20	PD2		DCMI_D11
21	PB0		DCMI_XSDN
22	PB1		DCMI_RST
23	GND		GND
24	PI8		DCMI_ PLUG

2.15 SDRAM

開発ボードに micron 社の MT48LC8M16A2P を搭載、容量は 128Mb、データ幅は 16 ビット。

2.16 NAND Flash

開発ボードに micron 社の MT29F4G08、4G Bit NAND Flash を搭載、チップセレクトは FSMC _NCE3 である。

2.17 Debug support

20 ピン 2.54 mm ピッチのシミュレーションソケット搭載。一般的な J-link と U-link2 を接続できる。通信モードは JTAG と SW 二種類をサポート。

2.18 FSMC Port

多機能バスインタフェース JP5 は、並列方式の AD チップ、8080/6800 並列モード LCD スクリーン、プログ ラマブルロジックデバイスなどを接続できる。

ラマブルロジック	ウデバイスなどを掛 宏差	送続できる。
<i>祝 9 月3 と</i> ピンの位置	と我 ピンの名前	機能説明
1	+3.3V	+3. 3V
2	GND	GND
3	PD14	FMC_D0
4	PD15	FMC_D1
5	PD0	FMC_D2
6	PD1	FMC_D3
7	PE7	FMC_D4
7	PE7	FMC_D4



HIC SUN)
	不可能への挑戦

8	PE8	FMC_D5		
9	PE9	FMC_D6		
10	PE10	FMC_D7		
11	PE11	FMC_D8		
12	PE12	FMC_D9		
13	PE13	FMC_D10		
14	PE14	FMC_D11		
15	PE15	FMC_D12		
16	PD8	FMC_D13		
17	PD9	FMC_D14		
18	PD10	FMC_D15		
19	PD7	FMC _NE1 (RS485 の制御ピンと共用)		
20	PF0	FMC_A0		
21	PD5	FSMC_NWE		
22	PD4	FSMC_NOE		
23	RESETn	RESETn		
24	PA8	MCO1		
25	PG10	NCE4_1 (R74 に 0Ω抵抗をハンダ付け、LCD_B2 と共 用)		
26	PH4	12C_SCL		
27	PG11	NCE4_2(R75 に 0Ω抵抗をハンダ付け、LCD_B3 と共 用)		
28	PH5	I2C_SDA		
29	PD3	予備 (DCMI_D5)		
30	GND	GND		
31	PD6	FMC_NWAIT (DCMI_D10 と共用)		
32	+3. 3V	+3. 3V		

2.19 TFT LCD

JP2 は RGB モードのLCD インタフェース、16 ビット・18 ビット・24 ビットの RGB スクリーンが接続できる。16 ビット及び 18 ビットスクリーンでは、高いから低いまで整列する。 *表 10 JP2 ピン定義*

A 10 012 20	No. 400			
ピンの位置	ピンの名前	RGB565	RGB666	RGB888
1	+5V			
2	+3.3V			
3	PH2			LCD_R0
4	PH3			LCD_R1
5	PH8		LCD_R0	LCD_R2
6	PH9	LCD_R0	LCD_R1	LCD_R3
7	PH10	LCD_R1	LCD_R2	LCD_R4
8	PH11	LCD_R2	LCD_R3	LCD_R5
9	PH12	LCD_R3	LCD_R4	LCD_R6
10	PG6	LCD_R4	LCD_R5	LCD_R7

低価格、高品質が不可能? 日昇テクノロジーなら可能にする

11	GND			
12	PE5			LCD_G0
13	PE6			LCD_G1
14	PH13	LCD_G0	LCD_G0	LCD_G2
15	PH14	LCD_G1	LCD_G1	LCD_G3
16	PH15	LCD_G2	LCD_G2	LCD_G4
17	PIO	LCD_G3	LCD_G3	LCD_G5
18	PI1	LCD_G4	LCD_G4	LCD_G6
19	PI2	LCD_G5	LCD_G5	LCD_G7
20	GND			
21	PE4			LCD_B0
22	PG12			LCD_B1
23	PG10		LCD_B0	LCD_B2
24	PG11	LCD_B0	LCD_B1	LCD_B3
25	PI4	LCD_B1	LCD_B2	LCD_B4
26	PI5	LCD_B2	LCD_B3	LCD_B5
27	PI6	LCD_B3	LCD_B4	LCD_B6
28	PI7	LCD_B4	LCD_B5	LCD_B7
29	GND			
30	PG7	LCD_CLK	LCD_CLK	LCD_CLK
31	PI10	LCD_HSYNC	LCD_HSYNC	LCD_HSYNC
32	PI9	LCD_VSYNC	LC <mark>D_</mark> VSYNC	LCD_VSYNC
33	PF10	LCD_DE	LCD_DE	LCD_DE
		MCO1(予備、	MCO1(予備、	MCO1(予備、
34	PA8	R53 をハンダ付	R53 をハンダ付	R53 をハンダ付
		け)	け)	け)
35	PCG	LCD_DISP(子	LCD_DISP(子	LCD_DISP(子
55	rco	備)	備)	備)
36	RESETn	RESETn	RESETn	RESETn
37	未使用			
90	DE 9	タッチスクリ	タッチスクリ	タッチスクリ
	FE2	ーン割り込み	ーン割り込み	ーン割り込み
39	PH5	I2C_SDA	I2C_SDA	I2C_SDA
40	PH4	I2C_SCL	I2C_SCL	I2C_SCL

2.20 Button

開発ボードには五つのボタンがあり、シルク RST のはリセットボタン、S1-S4 は普通 I/0 に接続される ボタン。

表 11 ボタンの機能

ボタン番号	機能説明
RST	開発ボードのリセットボタン
S1	PA0、押されるとハイレベル、通常がローレベル、ウェークアップ機能 付き



S2	PC13、押されるとローレベル、通常はハイレベル
S3	PE2、押されるとローレベル、通常はハイレベル
S4	PE3、押されるとローレベル、通常はハイレベル

2.21 LED

LED が二つあり、D1 は 3.3V 電源 LED、D2 は普通 I/O で制御する LED。

ホームページ: <u>http://www.csun.co.jp</u> メール: info@csun.co.jp



3. ソフトウェア説明

3.1 Keil コンパイル環境

3.1.1 コンパイル環境の構築

◆MDK473.exe をインストールする。このバージョン keil には rlarm がある。

◆keil コンパイラは体験版、一部分の機能が使われないので、License の購入を推奨する。(コンパ イラと RL-ARM 両方も License をインストールする必要)

◆コンパイラを C ディスクにインストールしなければならない。でなければ、一部分のプログラうが コンパイルができない。

3.1.2 コンパイル環境の設定

◆任意の例をオープンし、Options for Target をクリックする。 ← → | 陀 偽 偽 徴 | 徳 徳 北 版 | 🖄 - 🔊 🔒 🖶 Target Options... 📩 Blinky. (01 - / * * * * * * Configure target options ◆次に 🔣 Options for Target 'STM32F429_439xx' Device Target Output Listing User C/C++ Asm Linker Debug Utilities STMicroelectronics STM32F439II Code Generation Xtal (MHz): 8.0 Operating system: None Use Cross-Module Optimization Use MicroLIB 🔲 Big Endian System-Viewer File (.Sfr): C:\Keil\ARM\SFD\ST\STM32F4xx\STM 139x.SFR Floating Point Hardware: Use FPU • Use Custom SVD File Read/Only Memory Areas Read/Write Memory Areas default off-chip Start Size Startun default off-chip Start Size Nolnit ROM1: Г RAM1: Г C ROM2: Г Γ RAM2: Г ROM3: RAM3: Г on-chip IROM1: 0x800000 IRAM1: 0x20000000 0x30000 0x200000 Г $\mathbf{\overline{r}}$ Œ $\mathbf{\overline{\mathbf{v}}}$ IRAM2: 0x1000000 0x10000 IROM2: Г ОК Cancel Defaults Help

◆次のように、Utilitiesの「Use Target Driver for Flash Programming」にエミュレータを選択する。使用されたエミュレータがULINK2である場合に、「ULINK2/ME Cortex Debugger」を選択する、使用されたエミュレータがJLINK V8である場合に、「J-LINK/J-TRACE Cortex」を選択する。





◆Flash Download 選択バーの Programming Algorithm リストには、対応する開発ボードのデバイスが ない場合に、「add」をクリックし、新しいデバイスを追加する。





Download Function RAM	or Algorithm
Erase Full Chip 🔽 Program	
Erase Sectors	Nt 0x20000000 Size: 0x0800
Verify Sta	Size. Joxoboo
O Do not Erase	
- Programming Algorithm	
Description Device Type Device Size	Address Range
1	
St	art Size:
Add Remo	ve
Add Flash Programming Algorithm	
Description Elach Size Device Type Origin	
STM32E2vx Elash 1M On-chin Elash MDK Core	
STM32F2xx Flash Options 16B On-chip Flash MDK Core	
STM32F2xx Flash OTP 528B On-chip Flash MDK Core	
STM32F3xx Flash 256k On-chip Flash MDK Core	
STM32F4xx Flash 1M On-chip Flash MDK Core	
STM32F4xx 128kB Flash 128k On-chip Flash MDK Core	
STM32F4xx 2MB Flash 2M On-chip Flash MDK Core	
STM32F4xx 256kB Flash 256k On-chip Flash MDK Core	
STM32F4xx Flash OTP 528B On-chip Flash MDK Core	
STM32L15x Low Power Fl 128k On-chip Flash MDK Core	
STM32L15x Data EEPR0 4k On-chip Flash MDK Core	
STM32L1xx Med-density 128k On-chip Flash MDK Core	
STM32L1xx Mid-density 16B On-chip Flash MDK Core	
Selected Flash Algorithm File:	
C:\Keil\ARM\flash\STM32E4xx_2048ELM	
Add Cancel	
◆「Reset and Run」を選択したら、ULINK2「	エミュレータでアプリプログラムをダウンロードした後
に、プロガラトがオグに生存する	
$(c,) = j / \Delta M g \setminus (c \neq 1) g >_{0}$	
Debug Trace Flash Download	
Booldg Hate	
- Download Function	
LOAD C Erase Fun Chip Program	
Erase Sectors Verify	
🔿 Do not Erase 🔽 Reset and Run	
◆Debug 選択バーを設定する。プログラムが	ダウンロード失敗の時に、エミュレータのダウンロード
スビートを MT、IM以下の周波数に調節する。	





選択する。



Device Target Output Listing User	
Select Folder for <u>O</u> bjects	
Create Executable: .\Flash\Blinky	
✓ Debug Information	
Create HEX File	

✓ Browse Information

◆Browse Information も選択することを推奨する。そうすれば、コードをチェックすることが便利に なる。例えば、プログラムに init_display 関数を呼び出す時に、マウスのカーソルをこの関数の任意 フィールドに置き、右クリックする。ポップアップしたダイアログボックスに Go To Definition Of init_display を選択し、ソフトウェアは自動的にこの関数のエンティティにジャンプする。



3.2 DFU ソフトウェア

¥PC-Software ディレクトリの DfuSe_Demo_V3.0.3_Setup.exe をインストールする。

3.2.1 DFU File Manager

DFU File Manager はファイルフォーマット変換ソフトウェアである。

◆DFU File Manager.exe を実行し、I want to GENERATE a DFU file from S19, HEX or BIN files を選 択する。





◆S19 or Hexを選択する。

DFU File Manager (v3.0.2)	- Generation	
Device	Images	
Vendor ID 0x 0483		
Product ID 0x 0000	Injection	Multi BIN
Version 0x 0000	Target Name: ST	
	Deletion Delete se	elected Image
	<u>G</u> enerate	. <u>Cancel</u>
◆ ポップアップ	したダイアログボックスには変換する	HEXファイルを選択する。
😭 Open		
		_

		and the second s		
Look <u>i</u> n:	I STM324x	9I-EVAL_USBD-FS	← 🗈 💣 📰▼	
C.	Name	<u> </u>	Date mo	dified
Recent Places	STM324x	<pre>{9I-EVAL_USBD-FS.hex</pre>	2014/4/2	9:57
Desktop				
Libraries				
Computer				
Network				+
	File <u>n</u> ame:	STM324x9FEVAL_USBD-FS	•	<u>O</u> pen
	Files of type:	hex Files (*.hex)	-	Cancel
	\wedge	Open as <u>r</u> ead-only		

◆Generateを選び、DFU サフィックスファイルを生成する。



👔 DFU File Manager (v3.0.2) -	Generation	
	Images Image for Alternate Setting 00 (ST)	
Product ID 0x 0000 Version 0x 0000	Injection Signature Target ID: 0 Signature Signature Target Name: ST Signature Signature	<u>⊿ulti BIN</u>
	Deletion Delete select	red Image
◆ポップアップしたタ	イアログボックスには、DFU ファイルの名前と	呆存パスを新規作成する。
Save As	USBD-FS - + E 😤 🗐 -	



3.2.2 DfuSe Demonstration

DfuSe Demonstration は USB 通信方式でダウンロードされ、DFU フォーマットのファイルを FLASH にダウンロードできる。

◆SW1 ジャンパをシルク1側に接続し、usbケーブルを接続し、DfuSe Demonstration.exe を実行する。



🧼 DfuSe Demo (v3.0.3)	
Available DFU Devices STM Device in DFU Mode Supports Upload Manifestation tolerant Supports Download Accelerated Upload (Can Detach Leave DFU mode Enter DFU mode/HID detach Leave DFU mode Actions Select Target(s): Targe Name 00 Internal Flash	Application Mode: Vendor ID: Procuct ID: Version: Available Sectors (Double Click 24 sectors
Upload Action File: Choose Upload Transferred data size 0 KB(0 Bytes) of 0 KB(0 Bytes) Operation duration 00:00:00	or Verify Action D: Targets in file: D: Target
Abort	Elash な躍坦」 Ungueda an Vanifu Action / Chasses なんせいれた ガ
◆ Select Target(s)に Internal ウンロードプログラムを選択する。	Trash を速入し、 Upgrade of Verily Action (C Choose をクリックし、タ

DfuSe Demo (v3.0.3) Available DFU Devices STM Device in DFU Mode	Application Mode: DFU Mod	e: 0483
Supports Upload Manifestati Supports Download Accelerate Can Detach Enter DFU mode/HID detach Leaved Actions	Den tolerant ed Upload (ST) Procuct ID: Procuct II DFU mode Version: Version): DF11 : 0200
Select Target(s): Targe Name 00 Duernal Flash Upload Action File: <u>Qhoose</u> Upload Transferred data size 0 KB(0 Bytes) of 0 KB(0 Bytes) Operation duration	Available Sectors (Double Click) 24-sectors Upgrade or Verify Action File: Vendor ID: Procuct ID: Version: Version: Version: Optimize Upgrade duration (Remove some FFs)	
Abort		Quit



🧼 Open				-	-	X	
Look <u>i</u> n:	STM324x9I-EV	AL_USBD-FS	•	(† 🖻 🔿	* Ⅲ▼		
0	Name STM324x9I-EV	VAL_USBD-FS.dfu			Date mod 2014/4/2	lified 10:01	
Recent Places							
Desktop							
Libraries							
Computer							
Network	•	III				*	
	File <u>n</u> ame:	STM324x9I-EVAL_USE	BD-FS		•	<u>O</u> pen	
	Files of type:	Dfu Files (*.dfu)			•	Cancel	
		Open as <u>r</u> ead-only					

◆Upgrade をクリックして、プログラムを Flash にダウンロードする。ダウンロード終了、Leave DFU mode をクリックし、Flash にダウンロードされたアプリプログラムを実行する。電源を切るとき、SW1 ジャンパを シルク 0 側に接続しなければならない。

💭 DfuSe Demo (v3.0.3)			
Available DFU Devices			
STM Device in DFU Mode			
Supports Upload Supports Download Can Detach	Manifestation tolerant Accelerated Upload (ST)	-	
Enter <u>D</u> FU mode/HID detach	Leave DFU mode		



4. サンプルソースの説明

4.1 ¥Code¥STM32F4xx_DSP_StdPeriph_Lib_V1.3.0¥Project フォル

ダのプログラムの説明

このディレクトリのプログラムはST社からはドライブの更新は中止したので、新コードは ¥Code¥STM32Cube_FW_F4_V1.1.0¥Projects¥STM324x9I_EVAL¥Examples にご参照ください。

4.1.1 ¥DAC_SignalsGeneration¥MDK-ARM

このプログラムは DAC を使って、DMA を通じて、正弦波又はエスカレーター波形を生成する。USER キーを 押すと出力を切り替えが、オシロスコープで PA4、PA5 ピンの出力波形が見られる。





低価格、高品質が不可能? 日昇テクノロジーなら可能にする





低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.1.2 ¥DMA_FLASHToRAM¥MDK-ARM

このプログラムは DMA 方式で FLASH スペースのデータを RAM にコピーして比較する。正確な場合に、D2 LED が点灯している。

4.1.3 ¥EXTI_Example¥MDK-ARM

このプログラムが外部割り込みの機能を示した。プログラムは外部割り込みピンを posedge 又は negedge トリガーに設定し、対応するピンの割り込みサービスルーチンが LED に逆操作を行う。 割り込みピン機能の定義

ピン	対応キー	割り込みソース	割り込み方式	逆操作の LED
PAO	S1	EXTIO	posedge	D2
PC13	S2	EXTI13	negedge	D2

4.1.4 ¥FLASH_Program¥MDK-ARM

このプログラムは内部 FLASH のプログラミングを示す。

●プロセンサーがリセットされた後、FLASH スペースはロックされた。したがって、まず FLASH_Unlock を呼び出して、ロック解除する。FLASH プログラミングする前に、ページと単位にして FLASH_ErasePage 関数を呼び出して対応する FLASH スペースをイレースする。次に、FLASH_ProgramWord 関数を呼び出し、 バイト単位で flash スペースにプログラミングする。最後に、書き込まれたデータを読み出して比較し、 データが全部正確の場合に、S2 LED が点灯する。



4.1.5 ¥I2C_I0Expander¥MDK-ARM

このプログラムは I2C バスで、タッチスクリーン制御チップ STMPE811 が取得した座標情報を読み出す。 ◆プログラムを実行して、LED 画面には四つ異なる色のボックスが現れる。



4.1.6 ¥IWDG_Example¥MDK-ARM

プログラムは、ウォッチドッグが通常の状態で IWDG カウンタを更新する方法、及びドッグを食べさせる 時間を超えて発生された IWDG リセットを示す。

プログラムは WWDG タイムアウトを 250 ミリ秒に設定、240 ミリ秒ごとに犬を食べさせる。S2 キーを押す と、プログラムは無効なレジスタ書き込み動作を行う。これによって、プログラムが無限ループの空操作に 実行され、250 ミリ秒以内に犬を食べさせないと、IWDG リセットが発生する。プログラムが再実行して RCC_GetFlagStatus 関数を通じて、IWDG リセットが知られて、開発ボードの D2 LED が点灯している。

4.1.7 ¥NVIC_IRQMask¥MDK-ARM

このプログラムは、ネスト型ベクタ割り込みコントローラ(NVIC)の IRQ チャンネルの設定方法、及び別の IRQ をシールドする方法、有効にさせる方法を示す。



▶ TIM2の割り込みサービスルーチンにより、D2 LED が1秒ごとに1回点滅することを制御し; TIM3の割り込みサービスルーチンにより、PB0 が2秒ごとに否定されることを制御する; TIM4の割り込みサービスルーチンにより、PB1 が3秒ごとに否定されることを制御する。

>S1 キーを押すと、TIM2、TIM3、TIM4の割り込みが禁止、D2LEDの点滅を終了させる;S1 キーをもう一度押すと、TIM2、TIM3、TIM4の割り込みがオンにして、D2LEDが点滅する。

▶ S2 によりレジスタ BASEPRI を設定する。真ん中のキーを押すと、TIM3 と TIM4 の割り込みがマスク され、TIM2 の割り込みのみ動作するので、D2LED が点滅している。真ん中のキーをもう一度押すと、TIM3 と TIM4 の割り込みがオンにして、PB0 と PB1 のピンは、再びサイクルに変わる。

4.1.8 ¥NVIC_IRQPriority¥MDK-ARM

プログラムは、ネストされたベクタ割り込みコントローラの使用方法を示す。2つの外部割り込みと一つの SysTick 割り込みサービスルーチンを作成した。

▶ S1 と S2 キーは外部割り込みトリガモードに設定され、両方の先取り優先レベル PreemptionPriority も同じである。システムタイマーSysTick の割り込み優先度は外部割り込みより低 い。D2LED は、システムタイマーSytick が S1 キーに発生された外部割り込みを先に占有するかを示す。

▶ S1 キーを押して、EXTI の外部割込みが発生し、同時に対応する割り込みサービスルーチン内で、 システムタイマ SysTick をウェークアップするので、SysTick 割り込みが発生させた。SysTick の割り 込み優先度が低い、SysTick 割り込みが反応しなく、D2LED が続けて点灯している。

▶ S2 キーを押して、EXTI の外部割り込みが発生し、同時に、割り込みサービスルーチンに s1 キーに 発生された割り込みの優先レベルをレベル1に変更する。S1キーをもう一度押すと、SysTick 割り込 みが再びにウェークアップされる。この時、SysTick 優先レベルは、外部 S1 に発生された外部割り込み より高い、SysTick 割り込みが外部割り込みを占有し、かつ SysTick 割り込みサービス・ルーチンに PreemptionOccured を1に設定し、 D2 LED が点灯している。

▶ 上記の手順で USER と WAUP をトリガすると、D2 LED の点灯、点滅を実現させる。

4.1.9 ¥NVIC_WFIMode¥MDK-ARM

このプログラムは、システムがDMA 転送イネーブル WFI モードに入って、DMA 転送で割り込みを終了させ、システムをウェークアップする方法を示す。

DB9 インターフェイスにシリアルケーブルを接続し、ハイパーターミナルを開いてボーレートを 115200 に 設定し、ハードウェアフロー制御なし。プログラムが実行してから、D2LED が点滅している。S2 キーを押す と、プロセッサが WFI モードを入って、LED が消灯している。ハイパーターミナルに 0123456789 を入力し、 DMA 転送が終了、プロセッサがウェークアップされ、D2LED が再び点滅している。

4.1.10 ¥PWR_STANDBY¥MDK-ARM

このプログラムは、StandbyモードにRTCアラームによりプロセッサをウェークアップさせる方法を示す。 プログラムが実行された後に、D2LED が点滅する。S2 キーを押しと、D2 LED は消灯し、プロセッサが Standby モードに入る。電流計を通してボードに使用されている電流は小さく、5 秒後に RTC のアラームイベントが 発生し、プロセッサをウェークアップし、D2LED が続けて点滅する。





4.1.11 ¥PWR_STOP¥MDK-ARM

このプログラムは、stopモードに外部割り込み及びRTCアラームによりプロセッサをウェークアップさせる方法を示す。

プログラムは RTC アラームにより、プロセッサが 5 秒ごとに stop モードに入り、5 秒後にプロセッサが RTC アラームにウェークアップされる。stop モードには、s2 キーでプロセッサをウェークアップできる。そ の後、D2LED が点滅している。

4.1.12 ¥PWR_BOR¥MDK-ARM

プログラムはプロセッサ、低電圧閾値のプログラミング可能な機能を示す。

MAIN.Cのマクロ BOR_LEVEL は、低電圧の閾値を定義した。S2 キーを押し続け、開発ボードを再びリセットし、低電圧の閾値をチップの OPTCR レジスタに書き込める。

プログラムが実行された後に、D2LED が点滅している。開発ボードに調整可能な電源で給電し、プロセッサの動作電圧がプログラミングされた閾値電圧範囲内にさせ、プロセッサをリセットし、D2LED が点滅していない。

4.1.13 ¥RNG_MultiRNG¥MDK-ARM

プログラムは、複数の32ビット乱数を生成する機能を示す。

Main.h ファイルのマクロ PRINT_ON_USART または PRINT_ON_LCD は乱数をシリアルポートにプリントし(ボ ーレート:115200)、または液晶画面ディスプレイに表示する。プログラムが実行された後、S2 キーが 8 個 の 32 ビット乱数を受け取る。







4.1.14 ¥SysTick_Example¥MDK-ARM

プログラムは、システムタイマーの時間基準を1ミリ秒の遅延に設定されることを示す。クロックソースが HCLK で提供され、プログラムが実行されてから、D2LED が点滅している。

4.1.15 ¥TIM_TimeBase ¥MDK-ARM

プログラムは YIM3 CC1、TIM3 CC2、TIM3 CC3、TIM3 CC4 レジスタの数値を修正することにより、CC1、CC2、 CC3、CC4 の更新周波数をそれぞれに 73.24 Hz、109.8 Hz、219.7 Hz、439.4 Hz に設定される。割り込み処 理ルーチン TIM3_IRQHandler は開発ボードの四つ I/0 (PD13、PB0、PB1、PI8)を 73.24 Hz、109.8 Hz、219.7、 439.4 Hz ハイ・ローレベルに切り替え、オシロスコープで四つの周波数を検測する。





4.1.16 ¥TIM_PWMOutput¥MDK-ARM

プログラムは、汎用タイマ TIM3 を設定し、4 チャネル異なるデューティサイクルの PWM 信号を出力する方 法を示す。

◆汎用タイマ TIM3 の出力4 チャネルは: PC6、PC7、PC8、PC9。それぞれ出力された矩形信号のデュ ーティサイクルは: 50%、37.5%、25%、12.5%。試験方法: チャネル図をチェックし、逐次に4 チャ ネルピンをオシロスコープの入力チャンネルに入力し、出力波形を観察する。

◆以下はTIM タイマーPWM 出力設定の計算方法:

まず、予めに TIMCLK 周波数を SystemCoreClock / 2(HZ)に設定され、出力 PWM 周波数と TIM3 カウントク ロック周波数は TIM_Prescaler に基づき、ディバイドされ、TIM3 カウントクロックを取得する。もし、 TIM Prescaler = 0x03、それでは、

(1)TIM3 カウントクロック周波数= TIMCLK/(0x03+1));

(2) TIM_Period は波形周期を設定する。従って、その数値は TIM_Pulse より大きい。TIM_Period = 665、 つまり、TIM3 カウントクロックが 665 までにカウントすると、イベントジャンプが発生する。その時、出力 された PWM 波形周波数は TIMCLK/(665+1) /(0x03+1)) = 31.530 (KHZ)。

(3)出力 PWM デューディサイクルは CCR1_Val/(TIM_Period+1)。

◆各チャネルの出力 PWM 計算は

(1) TIM3 CCR1=333に設定し、TIM3 のチャンネル1の出力信号周波数は31.53kHz。

TIM3 Channel1 (PA6) PWMデューティサイクル = (TIM3_CCR1 / TIM3_ARR + 1)* 100 = 50% (2)TIM3 CCR2=249に設定し、TIM3 のチャンネル2の出力信号周波数は31.53kHz。

TIM3 Channel2 (PC7) PWMデューティサイクル = (TIM3_CCR2 / TIM3_ARR + 1)* 100 = 37.5%

(3) TIM3 CCR3=166に設定し、TIM3 のチャンネル3の出力信号周波数は31.53kHz。

TIM3 Channel3 (PC8) PWMデューティサイクル= (TIM3_CCR3 / TIM3_ARR + 1)* 100 = 25%

(4) TIM3 CCR4=83に設定し、TIM3 のチャンネル4の出力信号周波数は31.53kHz。

TIM3 Channel4 (PC9) PWMデューティサイクル = (TIM3_CCR4 / TIM3_ARR + 1)* 100 = 12.5% (5) オシロスコープで出力PWM波形を表示する。



出力結果は:

 \bullet Channel_1:PWM = 50%



 \bullet Channe1_3: PWM = 25%


低価格、高品質が不可能? 日昇テクノロジーなら可能にする









4.1.17 ¥TIM_CascadeSynchro¥MDK-ARM

プログラムは同期の TIM カスケードモードを示す。

◆プログラムは、TIM2、TIM3、TIM4 と3つのタイマを使用している。TIM2 をマスタタイマモードとして設定され、PWM 出力モードをオンにする。TIM2 更新イベントがトリガとして出力し、TIM2 出力周波数は 328.125 KHz、PWM デューティサイクルは 25%。

◆TIM3 は TIM2 のサブとして、TIM4 のホストとする。ITR1 は TIM2 の出力信号を取得する。TIM3 は PWM モードをオンにして TIM4 のトリガ出力とする。TIM3 の出力周波数は 82.02KHz、PWM のデューティ・ サイクルは 25%。

◆TIM4 は TIM3 のサブとする。ITR2 は TIM3 の出力信号を取得する。TIM4 も PWM モード出力信号をオ ンにして、TIM4 の出力周波数は 20.5 Hz、PWM のデューティ・サイクルが 25%。

◆TIM2のPWM 出力ピンはPA0:







低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.1.18 ¥TIM DMA¥MDK-ARM

このプログラムはタイマーDMA 転送モードを示す。プログラムは TIMI チャンネル 3 を設定し、17.57 KHz 周波数の相補 PWM 信号を生成する。

◆プログラムは 50%、37.5%、25%のデューティサイクルを定義した。これらの数値は aSRC_Buffer[] に保存され、TIMI リピートカウンタの数値によりデューティサイクルを選択する。

M 25.0µs

◆プログラムが実行された後、オシロスコープで PA10 ピンのデューティサイクル変化波形が見られ る。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする





低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.1.19 ¥TLM_DMABurst¥MDK-ARM

プログラムはDMAのバースト機能を示す。更新するたびに、 DMA がハーフワードデータを三回に TIM1 レジスタに書き込む。その中に、0x0FFF を ARR レジスタに書き込み、0x0000 を RCR レジスタに書き込み、0x0555 を CCR1 レジスタに書き込む。

◆プログラムが実行された後、オシロスコープで PA10 ピンの 5.8 KHz 出力周波数、33.33%デューティサイクルの波形が見られる。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.1.20 ¥TIM_OCToggle¥MDK-ARM

プログラムは、TIM3 周辺ディバイスを設定することにより、四つ異なる周波数の信号を生成する方法を示 す。TIM3 Channel 1 と PC6 ピンが対応して 256.35 Hz 矩形波を出力し、TIM3 Channel 2 と PC7 ピンが対応 して 512.7 Hz 矩形波を出力し、TIM3 Channel 3 と PC8 ピンが対応して 1025.4 Hz 矩形波を出力し、TIM3 Channel 4 と PC9 ピンが対応して 2050.78 Hz 矩形波を出力する。





低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.1.21 ¥TIM_7PWMOutput¥MDK-ARM

プログラムは、TIMIを設定することにより、七つのPWM信号と四つの異なるデューティサイクル(50%、37.5%、25%、12.5%)を生成する方法を示す。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする

◆プログラムが実行されたあと、PA8、PA10、PE11、PE14、PB13、PB14、PB15 などの七つピンが 17.57 KHz、異なるデューティサイクルの波形を出力する。

NO	ピン	チャンネル	デューティサイクル
1	PA8	TIM1Channel 1	50%
2	PB13	TIM1Channel 1N	50%
3	PE11	TIM1Channel 2	37.5%
4	PB14	TIM1Channel 2N	37.5%
5	PA10	TIM1Channel 3	25%
6	PB15	TIM1Channel 3N	25%
7	PE14	TIM1Channel 4	12.5%

◆四つデューティサイクルの波形:







4.1.22 ¥WWDG_Example¥MDK-ARM

プログラムは、システムウォッチドッグが通常の状態に WWDG カウンターを更新する方法、及びドッグを 食べさせる時間を超えて発生された WWDG リセットを示す。

◆プログラムは WWDG タイムアウトを 49.92 ミリ秒に設定され、40 ミリ秒ごとに大を食べさせる。S3 キーを押すと、プログラムは無効なレジスタ書き込み動作を行う。これによって、プログラムが無限ル ープの空操作に実行され、49.92 ミリ秒以内に大を食べさせない、リセットが発生する。プログラムが 再実行されて RCC_GetFlagStatus 関数を通じて、WWDG リセットが知られて、開発ボードの D2 LED が点 灯する。

プログラムを開発ボードにダウンロードされたあと、D2 LED が消灯している。S2 キーを押すと、プロセッサが IWDG リセットを生成し、D2LED が点灯している。

4.1.23 ¥LTDC_Display_1Layer¥MDK-ARM

プログラムは、LTDC 周辺デバイスの設定を示す。表示層により、LCD 画面に BMP 画像を表示する。

◆プログラムを開発ボードにダウンロードされたあとに、LCD 画面に以下のような写真を表示する。 写真の内容を配列型で RGB565_480x272.h に保存される。



4. 1.24 ¥LTDC_Display_2Layer¥MDK-ARM

◆プログラムは、二つの表示層が同時にそれぞれの写真を表示することを示す。写真の内容を配列型 で別々に RGB565_320x240.h と L8_320x240.h に保存される。プログラムを開発ボードにダウンロードされ たあとに、LCD 画面に以下のような写真を表示する。





4.1.25 ¥LTDC_ColorKeying¥MDK-ARM

プログラムはカラーキーイング機能を示す。

◆プログラムを開発ボードにダウンロードされたあとに、LCD 画面に以下のような写真を表示する。



◆S2 ★ を押すと、LCD 画面に以下のような写真を表示する。





4.1.26 ¥LTDC_PicturesFromSDCard¥MDK-ARM

プログラムは、SD カードの BMP 写真を LCD 画面に表示する。

◆テスト前に、¥Code¥STM32F4xx_DSP_StdPeriph_Lib_V1.3.0¥Project¥LTDC_PicturesFromSDCard ディレク トリの PICT フォルダを SD カードのルートディレクトリにコピーする。プログラムが実行されたあとに、S2 キーを押すと、次の写真をアクセスする。





4.1.27 ¥LTDC_AnimatedPictureFromSDCard¥MDK-ARM

プログラムは、SD カードの複数の写真を特定な時間順番に表示させ、動的表示を実現する。

◆テスト前に、予めに

¥Code¥STM32F4xx_DSP_StdPeriph_Lib_V1. 3. 0¥Project¥LTDC_AnimatedPictureFromSDCard ディレクトリの BACK と TOP フォルダを SD カードのルートディレクトリにコピーする。プログラムが実行されたあと、LCD 画 面の左上隅にはひらひら蝶がいる。



プログラムは、DMA2D 周辺デバイス機能を示す。プログラムは内蔵 flash に保存された写真が DMA 転送形式 で SRAM に出力し、かつ LCD で表示される。設定方法は DMA2D_Config 関数にご参考ください。





4.1.29 ¥DMA2D_RegToMemWithLCD¥MDK-ARM

プログラムは、DMA2D周辺デバイスがレジスタからSRAMまでにのDMA転送機能を示す。設定方法はDMA2D_Config 関数にご参考ください。



4. 1. 30 ¥DMA2D_MemToMemWithPFC¥MDK-ARM

プログラムはメモリ - メモリ DMA 転送およびフォーマット変換機能を示す。次の図には、上部カラーは ARGB8888フォーマットで、プログラムはこのカラーデータをDMA 転送モードに下部カラーの SRAM 領域に転送し、 かつフォーマットを ARGB4444 に変更する。





4.1.31 ¥DMA2D_MemToMemWithBlending¥MDK-ARM

プログラムは、DMAの混合転送モード機能を示す。次の写真は、前景と背景2つの画像から組み立てられた(赤・黄色のボックスは違う写真である)。





4.2 ¥Code¥STemWin_Library_V1.1.1¥Project¥STM324x9I-EVAL フォ

ルダのプログラム説明

4.2.1 ¥Standalone¥MDK-ARM

Segger 社は ST のプロセッサのために、emWin グラフィックライブラリをカスタマイズし、STemWin と名付ける。このプログラムは STemWin いくつかの機能を示す。そのプレゼンテーション機能は、ラジアルメニュー、透明ダイアログ、アイコンビュー、ツリービュー、リストビュー、仮想スクリーン、ズーム回転写真、描くグラフィック、ビットマップ、カーソル、カラーストリップがある。

◆Document ディレクトリにおける Getting started with STemWin Library.pdf ファイルには、プロジ ェクトファイルのCファイル機能、設定方法、および一部分のグラフィックスライブラリ関数機能につ いて説明した。

◆ラジアルメニュー





◆ツリービュー





(Halt)(Next

-80

25 50 75 100 125 150 175 200 225 250 275 300 325 350









4.2.2 ¥RTOS¥MDK-ARM

このプログラムは上記の例に基づいて、VNC サーバー機能を追加した。ネットワークプロトコルスタックは LwIP、操作システムは FreeRTOS である。

◆プロジェクトファイル Main.h はネットワークのマクロ定義を保存する:DHCP サービス、MAC アドレス、IP アドレス、サブネットマスク、ゲートウェイ、ネットワーク物理層モードの選択は、このファイル上に設定されている。プログラムがDHCP をオンにして自動的に IP アドレスを取得するとデフォルトする。

44	4 #define USE_DHCP /*	enable DHCP	
45	5		
46	6 /* Uncomment SERIAL DEBUG	to enables	
47	7 for debug purpose */		
48	8 //#define SERIAL DEBUG		
49	9		
50	0 /* MAC ADDRESS*/		
51	1 #define MAC_ADDR0 02		
52	2 #define MAC ADDR1 00		
53	3 #define MAC_ADDR2 00		
54	4 #define MAC_ADDR3 00		
55	5 #define MAC_ADDR4 00		
56	6 #define MAC_ADDR5 00		
57	7		
58	8 /*Static IP ADDRESS*/		
59	9 #define IP_ADDR0 192		
60	0 #define IP_ADDR1 168		
61	1 #define IP_ADDR2 1		
62	2 #define IP_ADDR3 108		

◆テストする前に、開発ボードをケーブルでDHCP機能が付きルータに接続する。プログラムを実行 すると、自動的にルータから IP アドレスを取得し、かつ、IP アドレスを LCD 画面に表示される:

VINC Server Use a VNC viewer to display and	
Control current Demo on PC DHCP Address assigned :	
192.168.1.101	Control Unc 12× Halt Next

◆上図の IP アドレスは 192.168.1.101、¥PC-Software¥VNC-4_1_3-x86_win32.exe ソフトウェアをイ ンストールし、Run Listening VNC Viewer ソフトウェアを実行し、かつ、[Server]フィールドに、開発 ボードの IP アドレスを入力し、[OK]ボタンをクリックして実行する。















ホームページ: <u>http://www.csun.co.jp</u> メール: info@csun.co.jp





◆VNC ソフトウェア機能:

PC ソフトウェアでディスプレイの実行状況を監視し、マウスでターゲットデバイスの実行状況を制御する (VNC ソフトウェアの表示ウィンドウにマウスを移動すると、ディスプレイにけるマウスも移動する。この 試験は、マウスで「Halt」、「Next」ボタンをクリックする)。

4.3 ¥Code¥STM32F4x9_ETH_LwIP_V1.1.0¥Project フォルダのプログ

ラム説明

4.3.1 ¥httpserver¥MDK-ARM

プログラムは 1wIP ネットワークプロトコルスタックを呼び出し、Web で開発ボードの LED と AD 値の読み 取りを制御する。keil のプロジェクトファイルは

¥Code¥STM32F4x9_ETH_LwIP_V1.1.0¥Project¥httpserver¥MDK-ARM ディレクトリにある。

◆Main.hはネットワークのマクロ定義を保存する。DHCP サービス、MAC アドレス、IP アドレス、サ ブネットマスク、ゲートウェイ、ネットワーク物理層モードの選択は、このファイル上に設定されてい る。

◆プログラムは、DHCP をオンにして自動的に IP アドレスを取得する機能とデフォルトする。開発ボ ードに電源を入れると、LCD 画面には割り当てられた IP アドレスを表示する。今回の開発ボードの IP は、192.168 1.105。



STM32F4x9 STM32F-4 Series Webserver Demo IP address assigned by a DHCP server: 192.168.1.105 ◆プログラムを開発ボードにダウンロードし、リセットキーを押して開発ボードをリセットする。 http://192.168.1.105/を入力すると、開発ボードにおける WEB が閲覧できる。 http://192.168.1.105/ P → C @ STM32F4x9 🕶 🛂 Search 🕶 More » 🖽 Google Sign In 🔌 🔹 🗶 🧭 Norton -网页安全 🔹 🖪 共享 y 🔲 访问保管库 STMicroelectronics 5/ STM32F4x9 Webserver Demo Based on the IwIP TCP/IP stack Home page Led control ADC status bar STM32 F-4 Series STM32 Releasing your creativity A new generation on STM32 with significant improvement in features / performance: More Memory Advanced features Maintain close pin-out compatibility Maintain close software compatibility Complement the existing family with more performance, memory and features The STM32F4x9 home page ◆Led control をクリックし、LED 制御インタフェースに入る。LED1 に選択するか、キャンセルする

か、Send キーを押して、LED1 は適切な変化を行う。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする

P → C 🥭 STM32F4x9LED \uparrow \star \ddot{a} 🔻 🛂 Search 🕌 More » 🖽 Sign In 🔍 - 🗶 🧭 Norton - 🔒 🧰 网页安全 - 🔤 共享 - 🔲 请问 Google STM32F4x9 Leds control Led control ADC status bar Home page This page allows you to control the four LEDs: LED1, LED2, LED3 and LED4 located in the STM324x9I-EVAL board. To put on/off a LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs configuration. Finally check in the STM324x9I-EVAL board that you get the desired LEDs are putted on/off. STM32 Webserver LEDs Control LED1 LED2 LED3 LED4 Send ◆ADC status bar をクリックすると、PA5の電圧値を取得できる。 P → C 🥭 STM32F4x9ADC × 🔹 🚼 Search 📲 More » 🕬 Google Sign In 🔌 🔹 🔤 网页安全 📲 共享 🛛 订门 x ton -STM32F4x9 ADC Conversion Home page Led control ADC status bai This page allows you to get continuously the ADC 1 Channel 5 analog input converted value. This ADC Channel is connected to the STM324x9I-EVAL board's potentiometer. The ADC value is updated, automatically, each 1s with the last converted ADC1 Channel 5 value. You could check this by changing the potentiometer position and check that the ADC value is updated, by an automatic refresh of this page, with the new converted value. ADC Converted Value 2880 mv

◆WEB内容をチップの内部FLASHに保存する。makefsdata.exeソフトウェアはWEBファイルをfsdata.c ファイルに変換できる。¥Code¥STM32F4x9_ETH_LwIP_V1.1.0¥Utilities¥Third_Party¥PC_Softwareディ レクトリのmakefsdata.exeソフトウェアを¥Code¥STM32F4x9_ETH_LwIP_V1.1.0¥Project¥httpserverに コピーし、コマンドウィンドウをオープンし、makefsdata.exeを実行すると、fsフォルダのWeb及び 写真ファイルをシングルなfsdata.cファイルに変換できる。fsdata.cファイルを ¥Code¥STM32F4x9_ETH_LwIP_V1.1.0¥Project ¥httpserver¥srcにコピーしてから再びコンパイルする。

Web ファイルが UltraEdit で編集できる。



C:\Windows\system32\cmd.exe				
processing subdirectory /STM32F processing /STM32F4x7_files/log processing /STM32F4x7_files/ST processing /STM32F4x7_files/St processing /index.html processing /index.html.bak failed to get file type for exp processing /STM32F4x7ADC.shtml processing /STM32F4x7ADC.shtml processing /STM32F4x7ADC.shtml failed to get file type for exp processing /STM32F4x7LED.html.pr failed to get file type for exp processing /STM32F4x7LED.html.pr failed to get file type for exp processing file type for exp processing file type for exp processing file type for exp failed to get file type for exp failed to get file type for exp Creating target file Processed 10 files - done. E:\Mywork\ARM\STM32F4x9\STM32F4 ct\httpserver>makefsdata.exe	⁷ 4x7_files/ go.jpg gif m32.jpg tension "bak", using de bak tension "bak", using de tension "bak", using de	fault. fault. fault. 2F4x9_ETH_LwIP_V1.	1.0\Proje	
EWARM	2013/12/26 22:13	File folder		
👢 fs	2014/1/6 18:17	File folder		
👃 inc	2013/12/26 22:13	File folder		
NDK-ARM	2014/1/6 17:50	File folder		
👢 RIDE	2013/12/26 22:13	File folder		
👢 src	2014/1/6 18:12	File folder		
👢 TrueSTUDIO	2013/12/26 22:13	File folder		
🔁 fsdata	2014/1/6 18:17	C Source File	362 KB	
💽 makefsdata	2011/11/14 19:05	Application	197 KB	
📄 readme	2013/11/13 23:41	Text Document	4 KB	
Release_Notes	2013/11/16 1:51	HTML Document	26 KB	
🗊 stm32f4xx_conf	2013/11/13 23:41	C/C++ Header File	5 KB	
🗊 system_stm32f4xx	2013/12/13 11:03	C Source File	39 KB	

4.3.2 ¥tftpserver¥MDK-ARM

TFTP の関連アプリプログラムである。TFTP ソフトウェア経由で、ファイルを開発ボードの SD カードにア ップロード又はダウンロードできる。

◆プログラムは、DHCP によって自動的に IP アドレスを取得することとデフォルトする。開発ボード に電源を入れると、LCD 画面には割り当てられた IP アドレスを表示する。今回の開発ボードの IP は、 192. 168. 1. 105。main. h には IP アドレス、サブネットマスク、ゲートウェイ、および DHCP サービスス イッチを変更できる。



STM32F4x9 STM32F-4 Series Tftpserver Demo

IP address assigned by a DHCP server 192.168.1.105

📩 main.c 🔄 🔜 main.h

064 #define NETMASK ADDR3

066 /*Gateway Address*/ 067 #define GW ADDR0

068 #define GW ADDR1

069 #define GW ADDR2

065

053					
054	/*Static IP ADDRESS: II	P_ADDR0.IP	ADDR1.IP	ADDR2.IP	ADDR3
055	#define IP_ADDR0 192				
056	#define IP_ADDR1 168				
057	#define IP_ADDR2 1				
058	#define IP_ADDR3 103				
059					
060	/*NETMASK*/				
061	#define NETMASK ADDR0	255			
062	#define NETMASK ADDR1	255			
063	#define NETMASK ADDR2	255			

0

192

168

1

070 #define GW_ADDR3 1
◆予めに、SD カードを挿入する。プログラムが開発ボードにダウンロードされ、リセットボタンを押して開発ボードをリセットし、コマンドラインウィンドウ(Cmd. exe)を開き、テキストファイルを作成する。コマンドラインウインドウで tftp -i 192.168.1.105 put test.txtを実行する。例えば、次の図に test.txt ファイルを SD カードスロットにアップロードする。(win7の操作グシステムには最新のオンライ、アップグレードパッチプログラムがないと、ネットワークカードのドライバの原因で、TFTP ノフトウェアが実行されなくなる)





◆D ドライブの test. txt ファイルをキャンセルする。コマンドラインに tftp -i 192.168.1.105 get test. txt を実行し、SD カードに保存された test. txt を D ドライブのルートディレクトリにダウンロー

ドする。

C. Windows (systems2	(cmd.exe
Transfers files t	o and from a remote computer running the TFTP service.
IFIP [-I] NOSE [G	
-i	Specifies binary image transfer mode (also called octet). In binary image mode the file is moved literally, byte by byte. Use this mode when transferring binary files.
host	Specifies the local or remote hos <mark>t.</mark>
GET	Transfers the file destination on the remote host to
	the file source on the local host.
PUT	Transfers the file source on the local host to
	the file destination on the remote host.
source	Specifies the file to transfer.
destination	Specifies where to transfer the file.
U:\>tftp -1 192.1	68.1.105 put lest.txt
Iranster successt	ul: 6 bytes in 1 second(s), 6 bytes/s
D:\>tftp -i 192.1	68.1.105 get test.txt
Successi	di. o byres in r second(s), o byres/s
D:\>_	-
このプログラム	は一部分の SD カードがサポートしていない。

4.3.3 ¥tcp_echo_server¥MDK-ARM

TCP 方式の echo サーバープログラムである。プログラムがクライアントからのデータを直接にクライアントに送信する。

◆DHCP を使用して IP アドレスを取得し、または固定 IP アドレスを使用する。DHCP 関数で IP アドレスを取得し、main.hに IP アドレスとゲートウェイを修正する。



main.c main.h
053
054 /*Static IP ADDRESS: IP ADDR0.IP ADDR1.IP ADDR2.IP ADDR3 */
056 #define IP_ADDR1 168
057 #define IP ADDR2 1
059 #define 1P_ADDR3 103
060 /*NETMASK*/
061 #define NETMASK_ADDR0 255
063 #define NETMASK ADDR1 255
064 #define NETMASK_ADDR3 0
065 066 /*Gateway Address*/
067 #define GW_ADDR0 192
068 #define GW ADDR1 168
069 #define GW_ADDR2 1 070 #define GW ADDR3 1
◆開発ボードにプログラムをダウンロードしてから、リセットキーを押して開発ボードをリセットす
る。¥PC-Software¥TCPUDPDbg ディレクトリのTCPUDPDbg.exeを実行する。「CreateConnn」をクリック
し、開発ボードに接続する準備を行う。
J TCP&UDP-Debug
Operate(O) View(V) Windows(W) Help(H) Language
Properties 4 ×
Client Mode
Server Mode
◆次の図にけ 開発ボードの TP が 192 168 1 105 に割り当てられる この TP アドレスを TP テキス
• 氏の因には、 備先が γ = 0 = 1 β = 100 = 100 に同う = C 540 β 。 この = 7 + ν λ ℓ = 1 γ γ γ
$[\pi, y \neq x] = [\pi, y \neq y \neq y \neq y \neq y = y = y = y = y = y =$
STM3274x2
STM32F-4 Series
TCP choserver Demo
IP address assigned
hy a DHCP cerver
192.100.1.105



Create Connection TCP • Type: DestIP: 192.168.1.105 Port: 7 Specia 7 LocalPort 🔘 Auto AutoConn: Eve σ Send When Conn: Eve ms Cancel Create ✗ TCP&UDP??工具 - [192.168.1.105:7] х Operate(O) View(V) Windows(W) Help(H) Language × 当 CreateConnn 🔇 CreateServer | 🐰 StartServer 迷 🕢 | 😪 Connect 😒 | 🛬 DisconnAll | 🎘 DeleteConn 🎇 0 8 Properties 4 Þ 🗙 Ψ × **192.168.1.105:7** 🖃 📃 Client Mode DestIP: Send 🗌 AtuoSend Eve 100 ms Stop 192.168.1.105:7 192.168.1.105 🔲 Send Hex 🔲 Send File 🗌 Send Received Server Mode Clear Option BroadOption DestPort: 7 abcdefghijklmnopqrstuvwxyz12345<mark>67</mark>890abcdefghijklmnopqrstuvwxyz1234567890abcdef 🔽 LocalPort ghijklmnopqrstuvwxyz1234567890a<mark>bc</mark>defghijklmnopqrstuvwxyz1234567890abcdefghijkl mnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqr stuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwx Type TCP • yz1234567890 AtuoConn Eve 0 AutoSend Eve 0 Connect Rec StopShow Clear Save Option ShowHex ount 📃 Save(In Time) o σ (lea Send Speed(B/S): 0 Receive Speed(B/S): 0

◆「Connect」をクリックし、開発ボードに接続する。送信領域にデータを入力し、「send」をクリ ックし、データを送信する。開発ボードが受信したデータを戻り、受信領域に表示される。

4.3.4 ¥udp_echo_server¥MDK-ARM

UDP 方式の echo サーバープログラムである。UDP 方式経由で、プログラムがクライアントからのデータを 直接にクライアントに送信する。



◆DHCP 関数により IP アドレスを取得し、開発ボードに電源を入れてから LCD 画面に割り当てられた IP アドレスを表示する。main.h が手動で開発ボードの IP アドレスを修正できる。

main.c main.h
053 054 /*Static IP ADDRESS: IP ADDR0.IP ADDR1.IP ADDR2.IP ADDR3 */ ⇒055 #define IP ADDR0 192 056 #define IP ADDR1 168 057 #define IP ADDR2 1 058 #define IP ADDR3 103 059
<pre>060 /*NETMASK*/ 061 #define NETMASK_ADDR0 255 062 #define NETMASK_ADDR1 255 063 #define NETMASK_ADDR2 255 064 #define NETMASK_ADDR3 0 065 066 /*Gateway Address*/</pre>
067 #define GW_ADDR0 192 068 #define GW_ADDR1 168 069 #define GW_ADDR2 1 070 #define GW_ADDR3 1
◆開発ホードにプロクラムをタワンロードしてから、リセットキーを押して開発ホードをリセットする。TCP&UDPDebug. exe を実行する。「CreateConnn」をクリックし、開発ボードに接続する準備を行う。
Market TCP&UDP-Debug
Operate(O) View(V) Windows(W) Help(H) Language
Properties Client Mode Server Mode
◆次の図には、開発ボードの IP が 192. 168. 1. 105 に割り当てられる。この IP アドレスを IP テキス トボックスに記入し、ポートが7、「Create」をクリックする。
STM32F4x9 STM32F-4 Series UDF Echoserver
IP address assigned by a DHCP server 192.168.1.105



Create Connection TCP • Type: DestIP: 192.168.1.105 Port: 7 ⊙ Specia 7 LocalPort 🔘 Auto AutoConn: Eve σ Send When Conn: Eve ms Create Cancel ◆次のインタフェースに「Connect」をクリックし、開発ボードに接続する。 X ✗ TCP&UDP??工具 - [192.168.1.105:7] Operate(O) View(V) Windows(W) Help(H) Language × 当 CreateConnn 🔕 CreateServer | 🐰 StartServer 迷 🕢 | 🗝 Connect 🐲 | 🛬 DisconnAll | 🎘 DeleteConn 🍇 0 8 Properties ąх 4 Þ 🗙 192.168.1.105:7 🖃 🖳 Client Mode DestIP: Send 🗌 AtuoSend Eve 100 Stop 9 192.168.1.105:7 192.168.1.105 🔲 Send Hex 🔲 Send File 🗌 Send Received Server Mode Option Clear BroadOption DestPort: 7 abcdefghi jklmnopqrstuvwxyz1234<mark>567</mark>890abcdefghi jklmnopqrstuvwxyz1234567890abcdef 🔽 LocalPort ghi jklmnopqrstuvwxyz1234567890abcdefghi jklmnopqrstuvwxyz1234567890abcdefghi jkl mnopqrstuww.xyz1234567890abcdefghijklmnopqrstuwwxyz1234567890abcdefghijklmnopqr stuwwxyz1234567890abcdefghijklmnopqrstuwwxyz1234567890abcdefghijklmnopqrstuwwx Type TCP • vz1234567890 T AtuoConn Eve 0 🔲 AutoSend Eve 0 Connect Rec StopShow Clear Save Option ShowHex ount 📃 Save(In Time) o σ (lea Send Speed(B/S): 0 Receive Speed(B/S): 0

◆送信エリアに送信文字を入力し、「Send」をクリックし、データを開発ボードに送信する。開発ボ ードが受信したデータをそのままに返信する。

4.3.5 ¥tcp_echo_client¥MDK-ARM

このプログラムは、1wIP ネットワークプロトコルスタックを呼び出して、TCP の echo クライアントを実現する。



◆プログラムは main.h にサーバとクライアントの IP アドレスを設定する必要がある。今回は PC が サーバ、IP アドレスが 192.168.1.100、開発ボードがクライアント、IP アドレスが自動的に DHCP 機能 を有するサーバによって割り当てられる(開発ボードの IP アドレスは 192.168.1.105 に割り当てられ、 LCD 画面に割り当てられた IP アドレスを表示する)、サーバーの IP アドレスは、実際のテスト PC によ り決められる。

044						
045	#define	DEST	IP ADD	R0	192	
046	#define	DEST	IP ADD	R1	168	
047	#define	DEST	IP ADD	R2	1	
048	#define	DEST	IP ADD	R3	100	
049						
050	#define	DEST	PORT		7	
051						
052	/* MAC /	ADDRE	SS: MAC	ADD	R0:MA	.C_ADDR1:MAC_ADDR2:MAC_ADDR3:MAC_
053	#define	MAC_	ADDR0	2		
054	#define	MAC	ADDR1	0		
055	#define	MAC	ADDR2	0		
056	#define	MAC_	ADDR3	0		
057	#define	MAC	ADDR4	0		
058	#define	MAC_	ADDR5	0		
059						
060	/*Statio	C IP	ADDRESS	: IP	ADDR	.0.IP_ADDR1.IP_ADDR2.IP_ADDR3 */
061	#define	IP_A	ADDR0	192		
062	#define	IP_A	ADDR1	168		
063	#define	IP_A	ADDR2	1		
064	#define	IP A	ADDR3	103		
005						

◆プログラムを開発ボードにラウンロードしてから、リセットキーを押して開発ボードのリセットを 行う。コマンドラインに echotool /p tcp /s を実行し、クライアントがポート 7 を介してサーバーへ 接続することを提示された。S2 キーを押して、開発ボードから送信したデータが echotool に表示され た。



◆Echotool. exe プログラムは¥PC-SoftWare に保存している。

4.3.6 ¥udp_echo_client¥MDK-ARM

このプログラムは、1wIP ネットワークプロトコルスタックを呼び出して、UDP の echo クライアントプロ グラムを実現する。



◆プログラムは main.h にサーバとクライアントの IP アドレスを設定する必要がある。、今回は PC がサーバ、IP アドレスが 192.168.1.100、開発ボードがクライアント、IP アドレスが自動的に DHCP 機 能を有するサーバによって割り当てられ、LCD 画面に表示する。サーバーの IP アドレスは、実際のテスト PC により決められる。

045	#define DEST IP ADDR0 192
046	#define DEST IP ADDR1 168
047	#define DEST IP ADDR2 1
048	#define DEST IP ADDR3 100
049	
050	<pre>#define UDP_SERVER_PORT 7 /* define the UDP local conne</pre>
051	#define UDP_CLIENT_PORT 7 /* define the UDP remote conr.
052	
053	/* MAC ADDRESS: MAC_ADDR0:MAC_ADDR1:MAC_ADDR2:MAC_ADDR3:MAC_
054	#define MAC_ADDR0 2
055	#define MAC_ADDR1 0
056	#define MAC_ADDR2 0
057	#define MAC ADDR3 0
058	#define MAC ADDR4 0
059	#define MAC_ADDR5 0
060	
061	/*Static IP ADDRESS: IP_ADDR0.IP_ADDR1.IP_ADDR2.IP_ADDR3 */
062	#define IP_ADDR0 192
063	#define IP_ADDR1 168
064	#define IP_ADDR2 1
065	#define IP_ADDR3 103

◆プログラムを開発ボードにラウンロードしてから、リセットキーを押して開発ボードのリセットを 行う。コマンドラインに echotool /p udp /s を実行し (echotool.exe プログラムは¥PC-SoftWare にあ る)、クライアントがポート7を介してサーバーへ接続することを提示された。S2キーを押して、echotool に受信した sending udp client message n を表示された。

C:\Windows\system32\cmd.exe - echotool /p udp /s	
	· · · · ·
E:\Mywork\ARM\STM32F4x9\STM32F4x9IG-DK-A-CD\PC-Software>echotool /p udp) /s
Waiting for UDP conncetion on port 7. Press any key to exit.	
22:11:45 from 192.168.1.105:49153 received [sending udp client message	0]
22:11:46 from 192.168.1.105:49154 received [sending udp client message	1]
22:11:46 from 192.168.1.105:49155 received [sending udp client message	2]
22:11:46 from 192.168.1.105:49156 received [sending udp client message	3]
22:11:47 from 192.168.1.105:49157 received [sending udp client message	4] =
22:11:47 from 192.168.1.105:49158 received [sending udp client message	5]
22:11:48 from 192.168.1.105:49159 received [sending udp client message	6]
22:11:48 from 192.168.1.105:49160 received [sending udp client message	7]
22:11:49 from 192.168.1.105:49161 received [sending udp client message	8]
22:11:49 from 192.168.1.105:49162 received [sending udp client message	9]
22:11:49 from 192.168.1.105:49163 received [sending udp client message	10]
22:11:50 from 192.168.1.105:49164 received [sending udp client message	11]
22:11:50 from 192.168.1.105:49165 received [sending udp client message	12]
22:11:51 from 192.168.1.105:49166 received [sending udp client message	13]
22:11:51 from 192.168.1.105:49167 received [sending udp client message	14]
22:11:51 from 192.168.1.105:49168 received [sending udp client message	15]
22:11:52 from 192.168.1.105:49169 received [sending udp client message	16]
22:11:52 from 192.168.1.105:49170 received [sending udp client message	17]
22:11:53 from 192.168.1.105:49171 received [sending udp client message	18]
22:11:55 from 192.168.1.105:49172 received [sending udp client message	19]
	~

4.3.7 ¥httpserver_netconn¥MDK-ARM

このプログラムは、1wIP ネットワークプロトコルスタックと FreeRTOS 操作システムを呼び出して、ネットワークの接続タイプは netconn、ウェブページを介してアクセスし、操作システムのスレッドをチェックることができる。

◆DHCP 関数により IP アドレスを取得する。IP アドレス及びゲートウェイが main.h に修正する。



📩 main.c 🔜 main.h

053					
054	/*Static IP ADDRESS: IP	ADDR0.IP	ADDR1.IP	ADDR2.IP	ADDR3 */
<₽>055	#define IP_ADDR0 192				
056	#define IP_ADDR1 168				
057	#define IP_ADDR2 1				
058	#define IP_ADDR3 103				
059					
060	/*NETMASK*/				
061	#define NETMASK_ADDR0	255			
062	#define NETMASK_ADDR1	255			
063	#define NETMASK_ADDR2	255			
064	#define NETMASK_ADDR3	0			
065					
066	/*Gateway Address*/				
067	#define GW_ADDR0 192				
068	#define GW_ADDR1 168				
069	#define GW_ADDR2 1				
070	#define GW_ADDR3 1				

◆プログラムを開発ボードにラウンロードしてから、リセットキーを押して開発ボードのリセットを 行う。LCD 画面で割り当てられた IP アドレスを取得する。今回が割り当てられた IP アドレスが 192.168.1.105。WEB に <u>http://192.168.1.105</u> を記入し、アクセスできる。





umber of page hits: 11 ame State Priority Stacl TTP R 5 296 DRR P 0 106	< Num	
ame State Priority Stac	c Num	
HTTP R 5 296		
DLE B 0 106	5	
P/TP B 6 894	1	
HCP B 4 178	6	
ED4 B 2 118	7	
th_if B 7 310	4	
_link B 3 224	2	
: Blocked, R : Ready, D : Deleted,	S : Suspended	

4.3.8 ¥httpserver_socket¥MDK-ARM

このプログラムは 1wIP ネットワークプロトコルスタックと FreeRTOS 操作システムを呼び出して、socket 通信方式で、ウェブページを介してアクセスし、操作システムのスレッドをチェックることができる。具体 的な方法は上例と同じである。

4.3.9 ¥udptcp_echo_server_netconn¥MDK-ARM

このプログラムは 1wIP ネットワークプロトコルスタックと FreeRTOS 操作システムを呼び出して、UDP 及び TCP 方式の echo サーバーを実現させる。netconn 通信方式を使用している。

◆DHCP 関数により、IP アドレスを取得する。IP アドレスとゲートウェイが main.h に修正する。

main.c main.h								
053								
054	/*Static IP ADDRESS: IF A	ADDR0.IP	ADDR1.IP	ADDR2.IP	ADDR3	*/		
<₽>055	#define IP_ADDR0 192							
056	#define IP ADDR1 📕 168							
057	#define_IP_ADDR2 1							
058	#define IP_ADDR3 103							
059								
060	/*NETMASK*/							
061	#define NETMASK_ADDR0 2	255						
062	#define NETMASK_ADDR1	255						
063	#define NETMASK_ADDR2	255						
064	#define NETMASK_ADDR3	0						
065								
066	/*Gateway Address*/							
067	#define GW_ADDR0 192							
068	#define GW_ADDR1 168							
069	#define GW_ADDR2 1							
070	#define GW_ADDR3 1							

◆開発ボードにプログラムをダウンロードしてから、リセットキーを押して開発ボードをリセットす



る。LCD 画面で割り当てられた IP アドレスを取得する。今回が割り当てられた IP アドレスが 192.168.1.105。¥PC-Software¥TCPUDPDbg ディレクトリの TCPUDPDbg.exe を実行する。「CreateConnn」 をクリックし、開発ボードに接続する準備を行う。




低価格、高品質が不可能? 日昇テクノロジーなら可能にする

: Operate(\underline{O}) View(\underline{V}) Windows(\underline{W}) Help(\underline{H}) Language X
🗄 🚰 CreateConnn 🚳 CreateServer 🐰 StartServer 🗏 🐼 😪 Connect 🧝 🌺 DisconnAll 💥 DeleteConn 🎇 🔯
Properties
Client Mode PestIP: Pe

◆「Connect」をクリックし、開発ボードへの接続を行う。送信エリアに送信データを入力し、「Send」 をクリックし開発ボードへのデータ送信を行う。開発ボードは受信されたデータを返信し、受信エリア に表示された。

◆同じ方法で、UDP データ転送方式を作成する。

4.4 ¥Code¥S1M324x91-EVAL¥Project フォルダのプログラム説明

4.4.1 ¥Blinky

コア基板における D2 LED 点滅を制御するプログラムである。LEDOn と LEDOff グローバル変数が SysTick_Handler に割り当てられる。

4.4.2 ¥Blinky_ADC

このプログラムは I/ 0 制御、タイマ、シリアルポートと LCD 画面表示を使用している。プログラムが 1 秒あたりに PA5 ところの AD 値をプリントし、同時に LCD 画面は進行状況バーと数値で AD 値を表示する。シ リアルは UART3、ボーレートが 115200、ハードウェアフロー制御なし。





4.4.3 ¥RTX_Blinky

プログラムは、RTX カーネルでステッピングモータの実行をシミュレーションし、実行モードが時計回りのハーフステップの駆動モードである。液晶画面には各実行されているスレッドが表示される。



RL Flash ファイルシステムを使用してシリアルポートを介して、コアボードに挿入された SD カードを制 御する。プログラムは FAT12、FAT16 および FAT32 ファイルのフォーマットをサポートしている。通常の SD カードと高速 SDHC カードと互換性があり、最大限 32GBytes をサポートする。 RL Flash ファイルシステム は、ファイル新規、ファイル修正、ファイル内容のチェック、ファイル名前の変更、ファイルのコピー、フ ァイルの削除、ルートディレクトリの情報のチェック、記憶媒体のフォーマット化などを行うことができる。

◆予めに、シリアルポート回線をDB9 ソケットに接続し、SD カードも挿入し、ハイパーターミナルを 実行し、ボーレートが 115200、ハードウェアフロー制御がない。開発ボードにプログラムをダウンロー ドして、リセットキーを押して、ボードのリセットを行う。ハイパーターミナルには、次の情報が出力 される。



■ USB-COM - 超级终端					
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(T)	帮助(<u>H</u>)				
De 93 DB 2					
SD/MMC Card	File Manipulation example				
+ command+	function				
CHP Thame L7HJ	[/A option appends data to a file]				
FILL "fname" [nnnn]	create a file filled with text				
TYPE "fname"	displays the content of a text file				
REN "fname1" "fname2"	renames a file 'fname1' to 'fname2'				
	['fin2' option merges 'fin' and 'fin2']				
DEL "fname"	deletes a file				
FORMAT [label [/FAT32]]	formats the device				
HELP or ?	l/FH132 option selects FH132 file system] displays this help				
· ····································	+				
Cmd> _					
连接的 0:03:01 自动检测 115200 8-N-1 50	TOLL CAPS NOM IN 1916				
◆上記の例にはcomm	mand 部分はコマンドフォーマット、function	部分はコマンドの機能と使用方法			
* 二日 ジリバーへ、 こう					
を紹介している。					
◆LCD 画面にはテスト	方法が表示される。				
SD File M	anipulation Example				
> Testing Method: 1: Insert SD Card to The	Card Slot.				
2: Connect the Serial Ca	ble to The DBS connector.				
4: BaudRate = 115200.					
5: Hardware flow control disabled.					

STM32F4x91 G Development Board

◆このプログラムは Canon の 16M SD カード、Kingston の 1G SD カード、Kingston の 8G SDHC をテストしたが、全ての SD カードをサポートする保証はできない。

4.4.5 ¥FlashFS¥NAND_File

RL Flash ファイルシステムを使用しシリアルポートを介して、コアボードにおける NAND Flash を制御する。プログラムは FAT12、FAT16 および FAT32 フォーマットをサポートしている。ファイルシステムは、ファイル新規、ファイル修正、ファイル内容のチェック、ファイル名前の変更、ファイルのコピー、ファイルの削除、ルートディレクトリの情報のチェック、記憶媒体のフォーマット化などを行うことができる。

◆予めに、シリアルポート回線をDB9 ソケットに接続し、ハイパーターミナルを実行し、ボーレートが 115200、ハードウェアフロー制御がない。開発ボードにプログラムをダウンロードして、リセットキ



ーを押して、ボードをのリセットを行う。NAND Flash がフォーマットされていない場合、ハイパーター ミナルは、次のメッセージをプリントし、書式設定するかを提示し、y の形式 NAND Flash を入力する。



◆上記の例には、command 部分はコマンドフォーマット、function 部分はコマンドの機能と使用方法 を紹介している。

◆LCD 画面にはテスト方法が表示される。





4.4.6 ¥USB_Device_Memory

USB マスストレージデバイスのプログラムで、高速およびフルスピードモード二種類がある。

◆プログラムは 30K 内蔵 RAM をストレージスペースとする。プログラムを開発ボードにダウンロード されたあと、Mini USB ケーブルを J6 コネクタに接続する。その時に、PC のモバイル機器のアイコンに は、新しいモバイルデバイスのアイコンが表示された。このデバイスをクリックし、テキストファイル が見られる。



USB HID 型のアプリプログラムである。PC に開発ボードを接続する時に、デバイスマネージャウィンド ウには、人体入力と出力のデバイスが接続されたと表示される。



文件(E) 操作(A) 查看(V) 帮助(H) ◆ ● ■ ■ ■ ・ 计算机 ● ● ■ ● ■ ● ■ ● ● ■ ● ■ ● ■ ● ■ ● ■ ● ■ ● ■ ● ■ ● ■ ● ■ ● ■ ● ■ ● ■ ● ■	🗿 设备管	理器		
 ● 注意 ● 目はetooth HID 设备 ● Bluetooth Remote Control ● HID-compliant device ● USB Human Interface Device ● USB Harman Interface Device ● USB Human Interface Device ● USB Human Interface Device ● USB Harman Interface Device ● USB Device 選択肢が見られる。 # HID Client ● Locie INputs (Buttons) ● Corrive Corries ● Device: ○ Dx00 ○ E 5 4 3 2 1 0 ● Dx00 ○ E 5 4 3 2 1 0 ● Dx00 ○ E 5 4 3 2 1 0 ● Dx00 ○ E 5 4 3 2 1 0 ● Dx00 ○ E 5 4 3 2 1 0 ● Dx00 ○ E 5 4 3 2 1 0 ● Dx00 ○ E 5 4 3 2 1 0 ● Dx00 ○ E 5 4 3 2 1 0 ● Dx00 ○ E 5 4 3 2 1 0 ● Dx01 ○ E 5 4 3 2 1 0 ● Dx01 	文件(<u>F</u>)	操作(<u>A</u>)	查看(⊻)	帮助(<u>H</u>)
 → 计算机 → 监视器 → 健盘 → 人体学输入设备 → Bluetooth HID 设备 → Bluetooth Remote Control → HID-compliant device → USB Human Interface Device → USB Human Interface Device → USB 抽入设备 → YPC-SoftWare¥HID_Client¥HID_Client → HID Client → HID Client → HID Client → Corridate Device → Device: Keil USB Device → Corridate Device → Device: Keil USB Devi		□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	12	
▲ 出版時 建盘 人体学输入设备 Bluetooth HID 设备 Bluetooth Remote Control HID-compliant device USB Human Interface Device USB Human Interface Device USB 物入设备 符合 HID 标准的用户控制设备 ◆ YPC-SoftWareYHID_ClientYI USB Device 選択肢が見られる。 ★ HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) 7 6 5 4 3 2 1 0 Dx00 7 6 5 4 3 2 1 0 Dx00 7 6 5 4 3 2 1 0 Dx00 0 corf バイスを選択し、0 corf HID Client Human Interface Device Device: Keil USB Device Device: Keil USB Device Inputs (Buttons) 0 x00 0		计算机		
 ▲ 人体学输入设备 Bluetooth HID 设备 Bluetooth Remote Control HID-compliant device USB Human Interface Device USB 抽入设备 ※ HID Client HUD Client Human Interface Device Device: Keil USB Device Inputs (Buttons) Qx00 7 6 5 4 3 2 1 0 Qx00 7 6 5 4 3 2 1 0 Qx00 7 6 5 4 3 2 1 0 Qx00 9 7 6 5 4 3 2 1 0 Qx00 9 6 5 4 3 2 1 0 Qx00 9 7 6 5 4 3 2 1 0 Qx00 9 6 5 4 3 2 1 0 Qx00 9 6 5 4 3 2 1 0 Qx00 9 6 5 4 3 2 1 0 Qx00 9 6 5 4 3 2 1 0 Qx00 		い 留盘 し		
■ Bluetooth Remote Control HID-compliant device USB Human Interface Device USB 轴入设备 WFPC-SoftWare¥HID_Client¥I USB Device 選択肢が見られる。 ● ¥PC-SoftWare¥HID_Client¥I USB Device 選択肢が見られる。 ● HID Client HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) 0x00 7 6 0x00 7 6 4 1 0 0x00 7 6 4 1 0 0x00 7 6 4 1 0 0x00 6 1 0 0x01 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 <	▲ · 🖓	人体学報 	俞入设备 etooth HII	D 设备
★ Lib Complete Letter USB Human Interface Device USB Human Interface Device USB 输入设备 符合 HID 标准的用户控制设备 ¥PC-SoftWare¥HID_Client¥ USB Device 選択肢が見られる。 USB Device 選択肢が見られる。 HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 0x00 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 0 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01		Blue	tooth Re	mote Control
◆ Cのデバイスを選択し、0ur HID Client ・ Corrice Covice ・ Covice Covice Covice ・ Covice Covice Covice ・ Covice Covice Covice ・ Covice Covice Covice Covice ・ Covice Covice Covice Covice Covice ・ Covice Covice Covice		USB	Human I	nterface Device
◆ ¥PC-SoftWare¥HID_Client¥H USB Device 選択肢が見られる。 ◆ HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0		USB USB USB	Human li 输入设备	nterface Device
◆ ¥PC-SoftWare ¥HID_Client ¥R USB Device 選択肢が見られる。 HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) ① 200 7 6 5 4 3 2 1 0 ② 200 7 6 5 4 3 2 1 0 ② 200 7 6 5 4 3 2 1 0 ② 200 7 6 5 4 3 2 1 0 ② 200 7 6 5 4 3 2 1 0 ② 200 7 6 5 4 3 2 1 0 ③ 200 7 6 5 4 3 2 1 0 ③ 200 7 6 5 4 3 2 1 0 ③ 200 7 6 5 4 3 2 1 0 ③ 200 7 6 5 4 3 2 1 0 ② 200 7 6 5 4 3 2 1 0 ③ 200 7 6 5 4 3 2 1 0 ③ 200 7 6 5 4 3 2 1 0 ③ 200 7 6 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6		崎 符合	HID 标准	的用户控制设备
■ HIC Softwarterning_offeneries USB Device 選択肢が見られる。 ■ HID Client ■ Human Interface Device ■ Device: Keil USB Device ■ Inputs (Buttons) 7 6 5 4 3 2 1 0 ■ Outputs (LEDs) ■ OUTPUTS (LEDs) ■ OUTPUTS (LEDs) ■ COデバイスを選択し、Outputs ■ HID Client ■ HID Client ■ HID Client ■ USB Device ■ USB		¥PC-S	oftWare	¥HID Client
HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) ① 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	USB	Devic	e 選択肢	友が見られる
HD Client Human Interface Device Device: Keil USB Device Inputs (Buttons) 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 Private Rest of the second se		Client		
Device: Keil USB Device Inputs (Buttons) 7 6 5 4 3 2 1 0 Outputs (LEDs) 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 Device: Keil USB Device Inputs (Buttons) 0x00 7 6 5 4 3 2 1 0 Outputs (LEDs) 0x00 7 6 5 4 3 2 1 0 Outputs (LEDs) 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0	-Huma	an Interfac	e Device –	
Inputs (Buttons) 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0v00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0	Dev	ice: Keil l	JSB Devic	:e
Outputs (LEDs) Outputs (LEDs) Outputs (LEDs) T 6 5 4 3 2 1 0 Ox00 T 6 5 4 3 2 1 0 Ox00 Ox0	Inpu	, uts (Button	s)	
Outputs (LEDs) 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		00 T	65	
◆このデバイスを選択し、0u ◆このデバイスを選択し、0u HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0		nute (LED	e)	
◆このデバイスを選択し、0u		7 puis (LED	s) 65	4 3 2 1
◆このデバイスを選択し、0ut HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) 0x00 7 6 5 4 3 2 1 0 Outputs (LEDs) 0x01 7 6 5 4 3 2 1 0		00		
◆このデバイスを選択し、Out HID Client Human Interface Device Device: Keil USB Device Inputs (Buttons) ①x00 7 6 5 4 3 2 1 0 ①x00 7 6 5 4 3 2 1 0 ①x01 7 6 5 4 3 2 1 0				
HID Client		▶このラ	デバイス	を選択し、
Human Interface Device Device: Keil USB Device Inputs (Buttons) 0x00 0x00 0x00 7 6 5 4 3 2 1 0 0utputs (LEDs) 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0	🗍 HID	Client		
Device: Keil USB Device	Hume	an Interfac	e Device –	\wedge
Inputs (Buttons) 0x00 7 6 5 4 3 2 1 0 0x00 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0 0x01 7 7 6 5 4 3 2 1 0 7 7 6 5 4 7 7 7 6 5 4 7 7 7 7 6 5 4 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	Dev	ice: Keil U	JSB Devic	e
0x00 7 6 5 4 3 2 1 0 Outputs (LEDs) 7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0	Inpu	uts (Button	s)	
Outputs (LEDs) 7 6 5 4 3 2 1 0 0x01 7 6 7 6 7 7 6 7 7 6 7 7 7 7 7 7 7 7 7	0×	00		
7 6 5 4 3 2 1 0 0x01 7 6 5 4 3 2 1 0	Out	puts (LED	s)	
	0x	01	65	4321

◆開発ボードの S1、S2、S3、S4 キーを押すと、Inputs (Buttons) も変化する。



👬 HID Client	HID Client
Human Interface Device	Human Interface Device
Device: Keil USB Device 💌	Device: Keil USB Device
Inputs (Buttons) 7 6 5 4 3 2 1 0 0x02	Inputs (Buttons) 7 6 5 4 3 2 1 0 0x01
Outputs (LEDs) 7 6 5 4 3 2 1 0 0x01 7 6 7 6 7 4 7 7 6 7 7 6 7 7 6 7 7 6 7 7 7 6 7	Outputs (LEDs) 7 6 5 4 3 2 1 0 0x01

4.4.8 ¥USB_Device_SD_Memory

このプログラムは USB デバイス MassStorage 型のアプリで、USB インタフェースで SD カードを読み書く。 開発ボードがカードリーダーとして使用できる。普通な SD カードと HCSD カードをサポートする。

◆テストする時に、USBを J6 に挿し込む。この時点で、PCが大容量デバイスのアクセスを検出する ことができる。暫く待つと、リムーバブルディスクのアイコンが見られる(SD カードの容量およびカー ドファイル数に応じて決める)。

▲ 有可移动存储的设备 (2) —



◆注意:このプログラムはCanonの16M SD カード、Kingstonの1G SD カード、Kingstonの8G SDHC をテストしたが、全てのSD カードをサポートする保証はできない。



4.4.9 ¥USB_Device_NAND_Memory

プログラムのテスト方法は上例(SD_Memory)と同じである。しかし、保存された記憶媒体はNAND Flash、 USB インタフェースは J6、LCD 画面には表示されない(表示ドライバを追加するが、プログラムが起動でき ない。原因は研究中)

▲ 有可移动存储的设备 (2)	
DVD RW 驱动器 (G:)	KEIL (H:) 486 MB 可用,共 486 MB
4.4.10 ¥USB_Host_HID_COM	
USB の HID クラスのマスタープログラム(フルス こに使用される。情報がハイパーターミナルに表示 ◆USB キーボードを J8 の USB-HOST インター ウンロードして、リセットキーを押して、ボー の情報が表示される。	スピード)である。USB キーボードのキー値を取得すると 示される。 - フェースに接続されたが、開発ボードにプログラムをダ ードをのリセットを行う。ハイパーターミナルは、次の図
■ USB-COM - 超级终端 文件任 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H) D ☞ @ ③ □ 户 函 ******* REMOTE MEASUREMENT RECORDER ****** This program is a simple Measurement Recorder.It is based on the SIM32F407 and records the state of the voltage on the analog input ADC1_12 . * command+ function	

◆USB キーボードにdを入力して、Enter キーを押して、ハイパーターミナルに電圧値が表示された。



 USB-COM - 超级终端 	
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(I) 帮助(H)	
0 quit recording \$ start recording	
Command: ****** REMOTE MEASUREMENT RECORDER ***** This program is a simple Measurement Recorder.It is based on the STM32F407 and records the state of the voltage on the analog input ADC1_12 . * command+ function+ R [n] read <n> records D display measurement T hh:mm:ss set time C clear records Q quit recording *</n>	
Display Measurements: (ESC aborts) Tim <u>e</u> : 0:00:53.207 AD value:2.22V	
连接的 0:52:34 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印	
◆USB キーボードに「ESC」キーを押して電圧表示を終了する。 ■ USB-COM - 超級終端	
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(I) 帮助(H) □ ☞ ③ ③ □ 凸 圖	
<pre>Command: ****** REMOTE MEASUREMENT RECORDER ***** I This program is a simple Measurement Recorder.It is based on the STM32F407 and records the state of the voltage on the analog input ADC1_12 . * command+ function</pre>	
	1

注意:このプログラムには LCD のドライバを追加しないが、 USB_Host_HID_LCD ファイルには LCD のドライバ を追加した。

4.4.11 ¥USB_Host_MSD_File

USBの Mass Storage 型マスタープログラム(フルスピード)である。ハイパーターミナルで USB メモリの 読み取り、書き込み、名前の変更、書式設定などの操作ができる。

予めに、シリアルポートを DB9 に接続し、USB メモリを J8 の USB-HOST インターフェイスに接続する。プロ グラムを開発ボードにダウンロードしてからリセットを行う。ハイパーターミナルが次の情報をプリントす





ハイパーターミナルに filltest. txt2 と入力し、Enter キーを押して、test. txt ファイルを作成する。このファイルには、ただ 2 行の内容がある。

Cmd> fill test.txt 2

File closed.

Cmd>

ハイパーターミナルに dir と入力し、Enter キーを押して、test. txt ファイルが見られる。



Cmd> fill test.txt 2 File closed. Cmd> dir File System Directory... MSWRITE.TXT 32 24.03.2011 10:32 <DIR> 26.03.2008 10:08 01.01.2010 12:00 FOUND.000 TEST.TXT 68 ハイパーターミナルに cap test. txt と入力し、Enter キーを押して、test. txt にデータを入力している。 Cmd> cap test.txt Capture data to file test.txt Press ESC to stop. ハイパーターミナルに Hello!と入力し、PC が Esc を押すと、終了。 Cmd> cap test.txt Capture data to file test.txt Press ESC to stop. Hello! File closed. Cmd> ハイパーターミナルに type test. txt と入力し、Enter キーを押して、入力された文字が見られる。 Cmd> type test.txt Read data from file test.txt Hello! File closed. Cmd> ハイパーターミナルに format udisk と入力し、Enter キーを押して、USB メモリを FAT32 フォーマットに 変更する。この時点で、スォーマットするかと提示される。 Cmd> format udisk Format Flash Mass Storage Device? [Y/N] yを押し、フォーマットを開始。 Cmd> format udisk I Format Flash Mass Storage Device? [Y/N] Mass Storage Device Formatted. Mass Storage Device Label is udisk Cmd> ハイパーターミナルに dir と入力し、USB メモリをチェックする。USB メモリの内容がクリアされた。 Cmd> dir

File System Directory... No files...

Cmd> _



注意:このプログラムは一部分のUSBメモリをサポートしていない。 LCD 画面にテスト方法を表示する。



4.4.12 ¥Http_demo

HTTP サーバのアプリケーションプログラムである。Web サーバーのCGI インターフェースを介して、開発ボードのデータを取得する上に、コマンドを送信し開発ボード資源を制御する。

◆ページファイルは、HTTP Filesのファイルグループに自動的に追加または削除することができ、 Web.inpがウェブ文書管理文書、プロジェクトファイルをコンパイルする過程に、自動的に C:¥Keil¥ARM¥bin¥fcarm @@Web¥Web.inp コマンドを実行し、WEB.cファイルを生成する。

🗄 🔄 HTTP_File	
🔜 index.htm	
🔜 network.cgi	
🔜 lcd.cgi	
📄 leds.cgi	
🔜 system.cgi	
🔜 tcp.cgi	
- 🔄 language.cgi	
🖳 🔜 buttons.cgi	
- g_footer.inc	
- 🔄 pg_header.ind	
🔜 buttons.cgx	Web.inp
🔜 ad.cgi	<pre>index.htm, keil.gif, st_logo.gif, llblue.jpg, pabb.gif, home.png,</pre>
🔜 ad.cgx	network.cgi, leds.cgi, tcp.cgi, system.cgi, lcd.cgi, language.cgi,
🖳 🛄 xml_http.js	ad.cgx. buttons.cgx to Web.c nopr root(Web)

◆予めいにデータ回線を開発ボードに接続する。ネットワークの設定問題を排除するために、DHCP機能がついているルータを推奨している(自動的に IP アドレスを割り当てる)。

◆HUB 使用又は直接に開発ボードに接続し、ニーズに応じて IP アドレス、サブネットマスク、ゲート ウェイを変更し、開発ボードの IP アドレスと PC の IP アドレスを一致にさせる。例えば、PC の IP アド レスは 192.168.1.100 の場合に、開発ボードの IP を 192.168.1.xxx に変更する。詳細は Net_Config.c 設定を参照してください。



Net_Config.c

Expand All Collapse All	
Option	Value
Ethernet Network Interface	~
Hac Address	
IP Address	
Address byte 1	192
Address byte 2	168
Address byte 3	1
Address byte 4	103
🕀 Subnet mask	
🖃 Default Gateway	
Address byte 1	192
Address byte 2	168
Address byte 3	1
, : : :	

Text Editor Configuration Wizard

◆CGI 関数に関連したアプリケーションは HTTP_CGI.c ファイルに配置、Webページの読み取りとプロ セッサリソース制御のブリッジである。CGI 関数を呼び出す方法は C:¥Keil¥ARM¥Hlp¥rlarm.chmの Script Language 及び CGI Functions 部分を参照してください。

◆プログラムをダウンロードし、リセットを行う。3秒後にコアボードのLED1が点滅している。ネットワークが問題ある場合に、10秒後に点滅する。LCD画面には割り当てられた IP アドレスを表示する。



◆ブラウザにhttp://stm324x9i-eval/又は割り当てられた IP を入力し、次のようなログインインタフェースが現れた。ユーザー名に admin を入力し、サーバーインタフェースに入る。









◆LED をクリックし、LED 制御インタフェースに入る。Browser を選択し、Submit をクリックして単 独に LED を制御できる。選択された LED が点灯している、選択されない LED が消灯している。第0桁が コアボードの LED1 を制御する。

Change Undo















4.4.13 ¥Http_upload

HTTP サーバーアプリケーション用のプログラムである。Web ブラウザを介してローカルファイルを開発ボードのSD カードに転送する。

◆予めに、開発ボードにネットワークケーブルを接続する。ネットワークの設定問題を排除するため に、DHCP機能が付いているルータを用意することを推奨している(自動的に IP アドレスを割り当てら れる)。

◆SD カードをスロットに挿入され、プログラムをダウンロードした後、リセットを行う。コア基板 D2LED が3秒後に点滅し始める。ネットワーク接続に問題がある場合に、約10秒後に点滅し始める。LCD 画面にが割り当てられた IP アドレスを表示している。



◆ブラウザに <u>http://stm324x9i-eval/</u>又は割り当てられた IP を入力し、次のようなログインインタフェースが現れた。ユーザー名に admin を入力し、パースワードなし、確認すると。



◆List Directory をクリックし、SD カードの内容をチェックする。









ウェブサイトでSDカードのテキストファイルが閲覧できる。



A R R DO MARK AND AND AND A				-	
← →	P - C (∂ stm324x9i-eval	×			h★‡
× Google	🕶 🐫 Search 🖬 More » 🖽	Sign In 🔌 🔹 🔉	< 🛧 🔤		= \$
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	llo!Hello	!	^
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	-
Hello!Hello!Hello!Hello!Hello!Hello!H	ello!Hello!Hello!Hello!H	ello!Hello!He	ello!Hello	!	

4.4.14 ¥Telnet_demo

このプログラムが Telnet サーバーの例で、IP に基づくコマンドラインのインタフェースを構築する方法 を紹介している。

◆上記のように、開発ボードがDHCPサーバーが付いているルータに接続している場合に、電源を入れると、自動的にIPアドレスを取得する。HUB又はPCを接続する場合に、Net_Config.cに開発ボードとPCを同じネットワークセグメントに設定してください。

◆プログラムをダウンロードした後、リセットを行う。コマンドウィンドウ(CMD. exe)をクリックし、 telnet stm324x9i-eval 又は telnet IPアドレスを実行し、ユーザー名 admin を記入し、パスワードが なし。







disconnect delete Character left recall Command History

◆機能は大体 Http_demo と同じが、コマンドライン方式が違う。

<ESC>,<^C> <BS>

<UP><DOWN>

Stm32>

_



Con Telnet stm324x9i-eval	
rinfo - display remote machine info lcdN text - write a text to LCD line N passw [new] - change system password passwd - display current password help, ? - display this help bye - disconnect	
<esc>,<^C> - disconnect <bs> - delete Character left <up><down> - recall Command History</down></up></bs></esc>	
Stm32> led 01> Running Lights OFF Stm32> led 00 Stm32> adin 5 ADIN 5 = 1893	
Stm32> lcd1 Hello! Stm32> lcd2 How are you? Stm32> meas 5 Nr. ADING ADIN1 ADIN2 ADIN3 ADIN4 ADIN5 ADIN6 ADIN7	
0 1872 1883 1880 1891 1881 1889 1890 1890 1 1886 1891 1885 1891 1890 1881 1879 1890 2 1891 1892 1888 1889 1898 1888 1889	
3 1887 1892 1889 1891 1884 1893 1881 1882 4 1891 1892 1889 1890 1880 1890 1881 1888 Stm32>	
> Testing Method: 1: Enable DHCP Function in The Router.	
2: Connect the Network cable to J2. 3: Run cmd.exe. 4: run teinet stm324x91-eval in the Command Line. 5: Authentication Username:admin.	
6: Authentication Password:Not Need.	

Hello! How are you?

STM32F4x9IG Development Board

◆Telnet_uif.cのアプリ関数は telnet ソフトウェア読み取りとプロセッサリソース制御のブリッジ とする。telnet 関連機能のオープン方法にご参照ください。

◆Win7 操作システムでは、telnet オフとデフォルトする。コントロールパネルで「プログラムと機能」をクリックして、「Windows の機能の有効化または無効化」をクリックする。

	ネル 🕨 プログラム 🕨 プログラムと機能
ファイル(<u>E</u>) 編集(<u>E</u>) 表示(⊻)	ツール(I) ヘルプ(<u>H</u>)
コントロール パネル ホーム	プログラムのアンインストールまたは変更
インストールされた更新プロ グラムを表示	プログラムをアンインストールするには、・ クします。
Windows の機能の有効化また は無効化	整理 ▼

◆「Telnet クライアント」と「TFTP クライアント」を選択する。「OK」を押すと、暫く待つと Telnet



がオンにする。



4.4.15 ¥DNS_demo

プログラムはワールド・ワイド・ウェブリモート DNS サーバに DNS 要求を送信し、サーバのネームにより サーバの IP アドレスを解析する機能を達成する、。

◆プログラムが自動的に IP アドレスが取得する DHCP サービスを使用しない、同じネットワークセグ メントに設定されるように、手動で開発ボードの IP アドレ、サブネットマスク、ゲートウェイなどを 設定する。5 秒ごとに次のサーバー、逐次に解析する:

<u>www.google.com</u> www.keil.com

www.microsoft.com

www. yahoo. com

www.notexisting.site

◆dns_cbfunc 関数が解析結果を判断し、シリアルポートで解析情報をプリントする。ポートのボーレ ートが 115200。



	III USB-COM - 超级终端	
	文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(I) 帮助(H)	
	Resolving host: www.notexisting.site Host name does not exist.	
1	Resolving host: www.google.com IP Address : 74.125.128.106	
	Resolving host: www.keil.com IP Address : 217.140.108.95	
	Resolving host: www.microsoft.com IP Address : 64.4.11.42	
	Resolving host: www.yahoo.com IP Address : 203.84.197.25	
	Resolving host: www.notexisting.site Host name does not exist.	
	Resolving host: www.google.com IP Address : 74.125.128.106	
	Resolving host: www.keil.com IP Address : 217.140.108.95 -	
	DNS request example > Testing Method: 1: Enable DHCP Function in The Router. 2: Connect the Network cable to J2. 3: Connect the Serial Cable to The DB9 Connector. 4: Run hypertrm.exe. 5: BaudRate = 115200. 6: Hardware Flow Control Disabled. P address: 192.168.1.104 STM32F4x91G Development Board	
	4.4.16 ¥FTP_demo FTP サーバプログラムである。PC の FTP クライアントにより SD た ◆プログラムをダウンロードした後、フォルダのアドレス欄 ユーザー名が admin、パスワードがなし。	ワードのファイルを管理する。 に <u>ftp://stm324x9i-eval/</u> を記入する。
	G G ftp://stm324x9i-eval/	







FTP Server demo example	
 > Testing Method: 1: Insert SD Card to The Card Slot. 2: Connect the Network cable to J2. 3: Enable DHCP Function in The Router. 4: Run ftp://stm324x9i-eval/ in the Address bar. 5: Authentication Username:admin. 6: Authentication Password:Not Need. 	
Paddress: 192,168.1.104 STM32F4x9IG Development Board	

4.4.17 ¥SNMP_demo

プロフラムは SNMP エージェントの演示例である。¥ PC-SOFTWARE ¥ SNMPTest¥SNMPTest.exe を実行し、 開発ボードの LED と読み取りキーのステータスを制御できる。ネットワークの設定問題を排除するために、 DHCP 機能が付いているルータを使用することを推奨している。

◆プログラムをダウンロードした後に、IPアドレスを自動的に取得し、かつLCD画面上に表示される。 今回、開発ボードの IP アドレスが 192.168.1.104 に割り当てられた。



◆SMPTest. exe を実行する。IP アドレス欄に DHCP サーバに自動的に割り当てられた IP アドレスを 入力する。Connect をクリックして、開発ボードに接続する。









4.4.18 ¥LEDSwitch

プログラムは UDP および TCP ポートを開く、¥ PC-SOFTWARE ¥ LEDSwitch¥LEDSwitch. exe から送信され たコマンドをリッスンする。LEDSwitch. exe は開発ボード LED のオンとオフを制御することができる。

◆プログラムが DHCP サービスを開かないので、手動で開発ボードの IP アドレスとゲートウェイを変 更する必要である。開発ボードと PC を同じ IP アドレスに設定され、Net_Config. c ファイルを変更する。 次の図では、IP アドレスが 192. 168. 1. 103 に設定され、ゲートウェイが 192. 168. 1. 1 に設定される。

LEDSwitch.c A Net_Config.c	
Expand All Collapse All	Hel
Option	Valu
Ethernet Network Interface	✓
H MAC Address	
□ IP Address	
Address byte 1	192
Address byte 2	168
Address byte 3	1
Address byte 4	103
⊞ Subnet mask	
🖻 Default Gateway	
Address byte 1	192
Address byte 2	168
Address byte 3	1
Address byte 4	1

◆プログラムをダウンロードしたあとに、LEDSwitch. exe を実行する。TCP/UDP Port を 1001 に設定 し、Board IP を 192.168.1.103 に設定し、Outputs[LEDs]の第 0 桁を選択するかがコアボードの LED1 の 点灯・消灯を制御している。TCP 又は UDP 転送方式が使用できる。

🛃 LEDSwitch	🛓 LEDSwitch
Protocol TCP/UDP Port	Protocol TCP/UDP Port
● TCP 1001	C TCP 1001
© UDP	● UDP
Board IP	Board IP
192 . 168 . 1 . 103	192 . 168 . 1 . 103
Outputs [LEDs] 7 6 5 4 3 2 1 0 0x01	Outputs [LEDs] 7 6 5 4 3 2 1 0 0x01

4.4.19 ¥BSD_server

プログラムは BSD socket サーバのデモで、TCP と UDP sockets を開き、PC クライアントと他のボードの クライアントから送信されたコマンドをリーソンする。このコマンドがコアボードの LED1 の on • off を制 御する。



◆テスト方法は前例と同じである。¥PC-SoftWare¥LEDSwitch¥ LEDSwitch. exe がクライアントとする。
 ◆プログラムをダウンロードしたあとに、LEDSwitch. exe を実行する。TCP/UDP Port を 1001 に設定し、Board IP を 192.168.1.103 に設定し、Outputs[LEDs]の第0桁を選択するかがコアボードの LED1の点灯・消灯を制御している。TCP 又は UDP 転送方式が使用できる。



4.4.20 ¥LEDClient

プログラムはLEDSwitchのクライアント機能を実現させる。PC にサーバーソフトウェアと合わせてテスト を行う。開発ボードのキーを押すと、サーバーソフトウェア側は開発ボードからのデータを受信する。

◆client.cにサーバ側の IP アドレスを設定し、Net_Config.cに開発ボードの IP アドレス、サブネ ットマスク、ゲートウェイを設定し、二つのデバイスが同じアドレスセクションである。

Client.c Net_Config.c		Client.c Net_Config.c	
Client.c Net_Config.c	xpand All Collapse All ption Value Remote IP Address IP1: Address byte 1 192 IP2: Address byte 2 168 IP3: Address byte 3 1 IP4: Address byte 4 100 Remote Port 1001 Communication Protocol UDP LED Blinking speed 2	Client.c Net_Config.c Expand All Collapse All Option Address byte 1 Address byte 2 Address byte 3 Address byte 4 Subnet mask Default Gateway Address byte 1 Address byte 4 Collapse All Collapse All Collapse All Collapse All Collapse All Collapse All Collapse All Address byte 1 Address byte 3 Collapse All Collapse All	Value 192 168 1 103 192 168
		Address byte 4	1
		\ Text Editor \ Configuration Wizard	

◆Client.cのマクロ定義を修正することにより、プログラムを TCP 又は UDP 通信方式に設定する。



Client.c Net_Config.c

058			
059	#define	BLINKLED	0x01
060	#define	SENDLEN	2
061	#define	TCP	0
062	#define	UDP	1
063			

◆¥PC-SoftWare の echotool. exe を C ドライブのルートディレクトリにコピーする。コマンドライン に echotool /p udp /s 1001 を実行し、UDP 方式でサーバを接続し、クライアントがポート 1001 を通過 するまで待つと提示される。



◆開発ボードに電源を入れると、サーバソフトウェアがクライアントからのデータを受信する。

C:\Windows\system32\cmd.exe - echotool.exe /p ud	p /s 1001
18:50:11 from 192.168.1.103:1001 received	[[○ ▶]
18:50:12 from 192.168.1.103:1001 received	[@+]
18:50:12 from 192.168.1.103:1001 received	[99]
18:50:12 from 192.168.1.103:1001 received	[99]
18:50:12 from 192.168.1.103:1001 received	[98]
18:50:14 from 192.168.1.103:1001 received	[③ ◆]
18:50:14 from 192.168.1.103:1001 received	[00]
18:50:14 from 192.168.1.103:1001 received	1 [Ø]
18:50:14 From 192.168.1.103:1001 received	[@▶]
18:50:14 from 192.168.1.103:1001 received]0
18:50:15 from 192.168.1.103:1001 received	[⊕◆]
18:50:15 from 192,168.1.103:1001 received	[88]
18:50:15 from 132 .168.1.103:1001 received	[88]
18:50:15 from 192.168.1.103:1001 received	[88]
18:50:15 from 192.168.1.103:1001 received	[⊕♦]
18:50:16 from 192.168.1.103:1001 received]0
18:50:16 from 192.168.1.103:1001 received	[⊕▶]
18:50:16 from 192.168.1.103:1001 received	[0]
18:50:16 from 192.168.1.103:1001 received	[00]
18:50:16 from 192.168.1.103:1001 received	[0?]
18:50:17 from 192.168.1.103:1001 received	[00]
18:50:17 from 192.168.1.103:1001 received	[0]
18:50:17 from 192.168.1.103:1001 received	[⊕▶]
18:50:17 from 192.168.1.103:1001 received]0
	· · · · · · · · · · · · · · · · · · ·



4.4.21 ¥BSD_client

このプログラムは BSD クライアントの演示例である。BSD サーバ又は BSD_server、LEDSwitchserver がインストールされた開発ボードを接続する。開発ボードのキーにより他のボードの led の点灯・消灯を制御している。

◆BSD_client.cにサーバのIPアドレスを設定する必要である。Net_Config.cに開発ボードのIPアドレス、サブネットマスク、ゲートウェイを設定し、二つのデバイスが同じアドレスセクションである。



◆¥PC-SoftWareのechotool.exeをCドライブのルートディレクトリにコピーする。コマンドライン にechotool/pudp/s1001を実行し、UDP方式でサーバを接続し、クライアントがポート1001を通過 するまで待つと提示される。





	o /s 1001
18:50:11 from 192.168.1.103:1001 received	I [@▶]
18:50:12 from 192.168.1.103:1001 received	[⊕ ♦]
18:50:12 from 192.168.1.103:1001 received	[@ @]
18:50:12 from 192.168.1.103:1001 received	[00]
18:50:12 from 192.168.1.103:1001 received	[99]
18:50:14 from 192.168.1.103:1001 received	[@◆]
18:50:14 from 192.168.1.103:1001 received	
18:50:14 from 192.168.1.103:1001 received	
18:50:14 from 192.168.1.103:1001 received	[(@▶]
18:50:14 from 192.168.1.103:1001 received] @
18:50:15 from 192.168.1.103:1001 received	
18:50:15 from 192.168.1.103:1001 received	
18:50:15 from 192.168.1.103:1001 received	[88]
18:50:15 from 192.168.1.103:1001 received	[99]
18:50:15 from 192.168.1.103:1001 received	[☺♠]
18:50:16 from 192.168.1.103:1001 received]⊕
18:50:16 from 192.168.1.103.1001 received	[☺▶]
18:50:16 from 192.168.1.103:1001 received	I [⊕]
18:50:16 from 192.168.1.103:1001 received	[86]
18:50:16 from 192.168.1.103:1001 received	[@?]
18:50:17 from 192.168.1.103:1001 received	
18:50:17 from 192.168.1.103:1001 received	
18:50:17 from 192.168.1.103:1001 received	[@▶]
18:50:17 from 192.168.1.103:1001 received	Θ

4.5 ¥Code¥STM32F4x9_uCOSIII¥Project フォルダのプログラム説明

4.5.1 ¥LED_Blinky

プログラムは uC/ OS-III 操作システムを使用して、メインスレッド Task_StartUp が Task_A と Task_B 二 つのスレッドを作成した。Task_StartUp は D2LED を各 100 クロックサイクルに一回に点滅させ、それ以外がしない。

◆Task_A、各100クロックサイクルにポートで「Task A is runing」をプリントする; Task_B、各



50 クロックサイクルにポートで「Task B is runing」をプリントする。ハイパーターミナルにはこの2 つのスレッドがプリントされた文字が見られる。

III USB-COM - 超级终端	_ D X	
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(I) 帮助(H)		
Task A is runing Task B is runing Task B is runing Task A is runing Task B is runing		
连接的 0:00:02 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印		

4.5.2 ¥Binary_Semaphores

プログラムは uC/OS-III にバイナリセマフォの使用方法を紹介する。Task_A は S2 ボタンを押るかリリースするかを検出する。Task_B は Task_A からの信号を待っている。S2 ボタンを押してリリースすると、Task_B は Task_A から送られた信号を受信し、D2LED のステータスを一度否定する。

4.5.3 ¥Binary_Semaphores_Interrupt

このプログラムは上例に基づいて外部割り込みアプリケーションを追加する。プログラムは、S1ボタンが 対応するピン PAO を posedge に設定し割り込みを生成する。割り込みルーチンは、主に Task_B に信号を送 って、Task_A が S2 ボタンの押し・リリースを検出する上に、Task_B に信号を送信する。

◆プログラムを実行すると、S1、S2を押したら、D2LEDのステータスを一度に否定される。しかし、 S1ボタンが押されると、D2LEDのステータスをすぐに否定される;S2ボタンを押してからリリースする と、D2_LEDのステータスが否定される。

4.5.4 ¥Mutex

プログラムは uC/ OS-III にミューテックスの使用方法を紹介する。スレッドは OSMutexPend を通して 他の周辺機器又は他のリソースの使用権を取得する。使用済、OSMutexPost を呼び出して使用権をリリース する。

◆スレッドAがポートで「Task A is runing」をプリントし、スレッドBがポートで「Task B is runing」 をプリントする。ボーレートが 115200、フロー制御なし。



■ USB-COM - 超级终端	x
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)	
Task B is runing Task A is runing Task B is runing Task A is runing Task B is runing	
Task B is runing Task B is runing Task B is runing Task B is runing -	
215200 000.22 BIOMAN 51-1-1-2-000 -1-1-2-000 CAPS 10000 BIOMAN BI	

4.5.5 ¥Mutex_LCD

プログラムは uC/OS-III にミューテックスの使用方法を紹介する。スレッドA とスレッドB は、LCD の 使用権のために競争している。スレッドが権利をもらって文字を LCD 画面に表示される。

◆スレッドAが実行するときに、LCD 画面には Task A is runing; Task B is waiting を表示され、

フォントは緑色である。スレッド B が実行するときに、LCD 画面には Task B is runing; Task A is waiting を表示され、フォントは赤である。






4.6 ¥Code¥STM32Cube_FW_F4_V1.1.0¥Projects¥STM324x9I_EVAL¥Examples フ

オルダのプログラム説明

STM32F4x9 プロセッサ用の周辺ドライバプログラムで、全部 STM32F4xx HAL API 関数を読み出す。

4.6.1 ¥GPIO¥GPIO_IOTogg1e¥MDK-ARM

簡単な I / 0 操作を実現。プログラムは 400ms ごとに D1LED と接続した I / 0 (PD. 13) を一回反転する。

4.6.2 ¥GPIO¥GPIO_EXTI¥MDK-ARM

プログラムは外部割り込み機能を紹介する。外部割り込みを posedge 又は negedge に設定し、対応する 割り込みが開発ボードの LED を否定する。

割り込みピン機能の定義

ピン	ピン 対応ボタン 割り込みソース		割り込み方式	否定された LED
PA0	S1	EXTIO	posedge	D2
PC13	S2	EXTI13	negedge	D2

4.6.3 ¥ADC¥ADC_TriggerMode¥MDK-ARM

プログラムは TIM8 により ADC1 チャネル 5 を連続出力モードに設定される。コールバック関数 HAL_ADC_ConvCpltCallback で HAL_ADC_GetValue を呼び出すことにより ADC 変換データを取得し、この値が シリアルポートを介してプリントアウトする。

◆シリアルポートのボーレートが115200、ハードウェアフロー制御なし。



■ USB-COM - 超级终端	
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)	
AD value = 0x082C	
$\begin{array}{c} \text{AD value} = 0 \times 0830 \\ \text{AD value} = 0 \times 0830 \\ \text{AD value} = 0 \times 0830 \\ \text{AD value} = 0 \times 0820 \\ \text{AD value} = $	
$\begin{array}{l} \text{HD value} = 0 \times 082B \\ \text{AD value} = 0 \times 0820 \\ \text{AD value} = $	
AD value = 0x0825 AD value = 0x082F	
AD value = 0x082D AD value = 0x082F	
AD value = 0x0823 AD value = 0x082D	
AD value = 0x082B AD value = 0x0828	
AD value = 0x0825 AD value = 0x0825	
$\begin{array}{l} \text{AD} \text{value} = 0 \times 082D \\ \text{AD} \text{value} = 0 \times 082D \\ \text{value} = 0 \times 082D \\$	
$\begin{array}{c c} AD & value = 0x002n \\ AD & value = 0x0022 \\ DD & ualue = 0x0020 \\ \end{array}$	
$\begin{array}{l} \text{AD value} = 0 \times 0050 \\ \text{AD value} = 0 \times 082 \\ \text{F} \\ \text{op value} = 0 \times 0000 \\ Op val$	
$\begin{array}{l} \text{HD value} = 0 \times 082 \text{H} \\ \text{AD value} = 0 \times 0827 \\ AD value$	
HD value = 0x082E	
◆プロセッサ速度を 144MHz に設定する。	

4.6.4 ¥ADC¥ADC_TripleModeInterleaved¥MDK-ARM

プログラムは三チャネル ADC トリプル代替モード機能を紹介する。STM32F4x9 プロセッサの PC2 ピン (ハ ードウェアでは DP83848 の MII_TXD2 ピンに接続された)は ADC1、ADC2、ADC3 チャンネル 12 の機能に達成 する。通常に、同じアプリケーションにはピンは唯一な機能を実現するが、これは、三チャネル交互 ADC 機 能を実現させる。

◆ADC変換データはaADCTripleConvertedValueに保存される。変換終了、割り込みを一回に生成する。 コールバック関数 HAL_ADC_ConvCpltCallback からの変換データはポートでプリントアウトする。

◆ポートのボーレートが 115200、ハードウェアフロー制御なし。PC2 ピンと DP83848 の MII_TXD2 が 接続しているので、ADC が取得されたデータはフローティングである。

USB-COM - 超级终端	
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(D) 帮助(H)	
AD1 value = 0x3830426 AD2 value = 0x4340341 AD3 value = 0x42039F AD1 value = 0x40039F AD1 value = 0x42039F AD1 value = 0x39803FA AD3 value = 0x39803FA AD3 value = 0x3980417 AD1 value = 0x3980416 AD3 value = 0x3980416 AD3 value = 0x3980416 AD3 value = 0x3980417 AD1 value = 0x4060379 AD2 value = 0x4060379 AD2 value = 0x408037F AD3 value = 0x308047 AD3 value = 0x308047 AD3 value = 0x308047 AD3 value = 0x37E041B AD1 value = 0x398047 AD2 value = 0x37E041B AD1 value = 0x430803F AD2 value = 0x430803PE AD3 value = 0x42003AD AD2 value = 0x42003AD AD2 value = 0x42003P7 AD2 value = 0x42003P7 AD2<	
连接的 3:38:27 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印	

◆三チャネル ADC と変換結果の関係図:



ADC1	
ADC2	
ADC3	
CH12data	
	Sampling time : 3 cycles
	Delay between sampling start of 2 ADC : 5 cycles
	Convertion phase : 12 cycles (resolution=12bit ==> Conversion time = 12 cycles)
	A Converted data availble

4.6.5 ¥ADC¥ADC_RegularConversion_Polling¥MDK-ARM

プログラムはポーリングモードによって ADC1 チャネル 5 の AD 値を取得する。500ms ごとにシリアルポートで ADC1 チャネル 5 現在の AD 値をプリントアウトする。抵抗 R58 の抵抗値を調整し、AD 値が変わる。ポートのボーレートが 115200、ハードウェアフロー制御なし。



◆ADC は対応するピンの AD 値を取得するために、5 つのステップがある。ステップ1、ADC のプロパ ティを設定して HAL_ADC_Init を呼び出し ADC の初期化を行う;ステップ2、ADC のチャンネルを設定し、 HAL_ADC_ConfigChannel を読み出す;ステップ3、HAL_ADC_Start を呼び出して、ADC 変換機能を開始; ステップ4、HAL_ADC_PollForConversion を呼び出し、ADC 変換終了まで待つ;ステップ5、 HAL_ADC_GetState を呼び出して、ADC 変換が成功するかを判断する。最後に、HAL_ADC_GetValue を呼び 出して、AD 値を取得する。次の ADC 変換はステップ3 から開始である。



4.6.6 ¥ADC¥ADC_RegularConversion_Interrupt¥MDK-ARM

プログラムは割り込みモードにより、ADC1 チャンネル5のAD 値を取得する。

◆コールバック関数 HAL_ADC_ConvCpltCallback に HAL_ADC_GetValue 関数を呼び出して、変換された AD 値を取得する。抵抗 R58 の抵抗値を調整し、AD 値が変わる。ポートのボーレートが 115200、ハード ウェアフロー制御なし。

■ USB-COM - 超级终端	
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(I) 帮助(H)	
AD value = 0x099D AD value = 0x099B AD value = 0x099F AD value = 0x099F AD value = 0x099C AD value = 0x099C AD value = 0x099C AD value = 0x099F AD <td></td>	

4.6.7 ¥ADC¥ADC_RegularConversion_DMA¥MDK-ARM

プログラムはDMA モードにより、ADC3 チャンネル8(対応ピンはPF10)のAD 値を取得する。

◆コールバック関数 HAL_ADC_ConvCpltCallback に HAL_ADC_GetValue 関数を呼び出して、変換された AD 値を取得する。AD 値がポートでプリントアウトできる。ポートのボーレートが 115200、ハードウェ アフロー制御なし。

4.6.8 ¥ADC¥ADC_DualModeInterleaved¥MDK-ARM

プログラムはDMAモード3に、ADC1とADC2のデュアルインターリーブ・モード変換の実現方法を紹介する。 変換周波数が5Msps、ADC1とADC2がPC2ピンを共有し、チャンネル12でデータ変換を行う。

◆DMA 要求の通りに、ADC2 変換結果が高8ビット、ADC1 変換結果が低8ビットとして ADC_CDR レジス タ(ADC_CDR[15:0] = (ADC2_DR[7:0] << 8) | ADC1_DR[7:0])に保存される。ADC 変換が終了、割り込み を生成し、ユーザーが割り込みコールバック関数 HAL_ADC_ ConvCpltCallback に uhADCDualConvertedValue 変数の値を読み取ることができる。

◆プログラムがコールバック関数にポートで変換の数値をプリントアウトする。ポートのボーレートが 115200、ハードウェアフロー制御なし。



📑 USB-COM - 超级终端	
文件(E) 编辑(E) 查看(V) 呼叫(C) 传送(I) 帮助(H)	
RD value = 0x343R RD value = 0x3734 RD value = 0x3734 RD value = 0x3633 RD value = 0x4335 RD value = 0x4355 RD value = 0x4343 RD value = 0x4542 RD value = 0x4542 RD value = 0x4542 RD value = 0x4542 RD value = 0x3241 RD value = 0x3240 RD value = 0x3240 RD value = 0x3333 RD value = 0x3333 RD value = 0x3637 A = 0x3637 A XEMEND 115200 8.N.1 CAPS NUM #E FTEE	

4.6.9 ¥CAN¥CAN_LoopBack¥MDK-ARM

プログラムはループバック・モードに CAN バスのデータ受送信機能を紹介する。一つの開発ボードのみで 完成できる。開発ボードにダウンロードして実行すると、受送信機能が正常の場合に、D2LED が点灯してい る。

4.6.10 ¥CAN¥CAN_Networking¥MDK-ARM

プログラムは正常なモードに CAN バスのデータ受送信機能を紹介する。

◆二つの開発ボードが必要。両方に同じプログラムをダウンロードする。I/0の多重化のため、R63 とR64に0Ω抵抗(J6コネクタにUSBケーブルが接続されていない)をハンダ付け必要がある。2本の ワイヤで、開発ボードAのCAN_H1、CAN_L1と開発ボードCAN_H1、CAN_L1と接続する。

◆プログラムが実行したあと、任意のボードの S2 ボタンを押すと、他のボードの D2 LED が点灯している。 いる。もう一回押すと、D2 LED が消灯している。

4.6.11 ¥Cortex¥CORTEXM_SysTick¥MDK-ARM

プログラムはタイマーを使用して 1mS 割り込みを生成する。割り込みサービスルーチン SysTick_Handler は HAL_IncTick 関数を呼び出して、変数 uwTick の数値インクリメントを割り当て、HAL_Delay 関数は uwTick 値の変化に従って正確な遅延を得る。STM32Cube の例は大体システムタイマーを使用している。プログラム が実行したあと、D2 LED が 100 ミリ秒ごとに 1 回にステータスを否定する。

4.6.12 ¥Cortex¥CORTEXM_MPU¥MDK-ARM

プログラムはプロセッサ MPU 機能を紹介する。プログラムは、読み取り専用のメモリ権限領域を備えており、いくつかの異なるモードに領域への読み出し動作と書き込み動作を行い、操作が正しい場合に、D1LED が点滅している。

◆実際は、プログラムが実行された後に、D1LED がいつも点滅している。プロセッサがエラー操作を 実行させるために、stm32_mpu.cファイルを開き、遮られた「PrivilegedReadOnlyArray[0] = 'e';」を オンにする。

4.6.13 ¥CRC¥CRC_Example¥MDK-ARM

プログラムは STM32Cube ソフトウェアの CRC API 関数機能を紹介する。まず、プログラムは HAL_CRC_Init 関数を呼び出し、CRC の初期化を行う。次に、HAL_CRC_Accumulate 関数を呼び出し、aDataBuffer データに CRC 計算を行う。計算結果を uwCRCValue に保存され、予想 CRC 値と比較し、数値が同じである場合に、D2 LED が点灯している。

4.6.14 ¥DAC¥DAC_SignalsGeneration¥MDK-ARM

プログラムはDMA コントローラで DAC 周辺機器を使用して複数の信号を生成する。

◆プログラムを実行した後、ピン PA4 が出力した三角波が見られる。S2 ボタンを押すと、PA4 ピンが エスカレーター波形に切り替える。





低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4. 6. 15 ¥DMA¥DMA_FLASHToRAM¥MDK-ARM

プログラムは DMA チャネル 0 を利用して、内部フラッシュメモリに保存されワード値をチップの内部 RAM 空間に転送する。 DMA 転送が正しい場合は、コールバック関数 TransferError が呼び出され、D2 LED が点 灯している。DMA 伝送エラーの場合は、コールバック関数 TransferComplete を実行している。

4.6.16 ¥DMA¥DMA_FIFOMode¥MDK-ARM

プログラムはDMAのFIF0モードを使用して、内部フラッシュメモリに保存されたワード値をチップの内



部 RAM 空間に転送する。 DMA 転送が正しい場合は、コールバック関数 TransferError が呼び出され、D2LED が点灯している。DMA 伝送エラーの場合、コールバック関数 TransferError を実行する。

4.6.17 ¥DMA2D¥DMA2D_MemoryToMemory¥MDK-ARM

プログラムは DMA2D 転送モードに aBufferInput [256]のデータを aBufferResult [256]に転送する (MemorytoMemory 転送モードを実現する)。転送が終了、プログラムは Buffercmp 関数を呼び出して、2つ の配列を比較すると、データが同じの場合に、D2 LED が点灯している。

4.6.18 ¥DMA2D¥DMA2D_MemToMemWithBlending¥MDK-ARM

プログラムは DMA2D 転送機能を実現する。転送モードは、混合メモリからメモリ転送モードにである。プ ログラムは LCD ディスプレイのフォアグラウンドデータバッファのデータをバックグラウンドのデータバッ ファに送信される。DMA2D 送信が正しい場合に、コールバック関数 TransferComplete が呼び出され、D2 LED が点灯している。MA2D 送信がエラーの場合に、コールバック関数 TransferError を実行する。



4. 6. 19 ¥DMA2D¥DMA2D_MemToMemWithPFC¥MDK-ARM

プログラムはメモリからメモリまでの DMA2D 転送、及びフォーマット変換機能を実現する。次の図のカ ラーチャートは ARGB8888 フォーマットで、プログラムは DMA2D 転送モードにこのカラーデータを SRAM 領域 に転送し、かつ ARGB4444 に変換する。





4.6.20 ¥DMA2D¥DMA2D_MemToMemWithLCD¥MDK-ARM

プログラムは DMA2D 周辺機能を紹介する。プログラムは、DMA2D 周辺機能で内部 FLASH に保存された画像 ファイルを SRAM に転送し、かつ LCD ディスプレイに表示される。設定方法は DMA2D_Config 関数を参照して ください。





4.6.21 ¥DMA2D¥DMA2D_RegToMemWithLCD¥MDK-ARM

プログラムは、DMA2D 周辺機器のデータをレジスタから SRAM の DMA2D に転送する機能を紹介する。設定 方法は DMA2D_Config 関数を参照してください。



4.6.22 ¥FLASH¥FLASH_EraseProgram¥MDK-ARM

プログラムが STM32Cube ソフトウェアの Flash イレース、プログラミングの API 関数呼び出し方法や機能を紹介している。

◆ステップ1、HAL_FLASH_Unlock を呼び出して、Flash コントローラにアクセスするために Flash の ロックを解除する。

◆ステップ2、GetSector 関数を呼び出し、イレースの開始と終了セクタを取得する。

◆ステップ3、HAL_FLASHEx_Erase 関数を呼び出して、プログラミングセクタをイレースする。

◆ステップ4、HAL_FLASH_Program を呼び出して、flash にデータを書き込み、今回のデータは 0x12345678。

◆ステップ 5、HAL FLASH_Lock を呼び出して、Flash の制御用レジスタを再びロックする。

◆最後に、flashに書き込まれたデータを読み出して比較する。データが正しい場合に、D2 LED が点 灯している。

4.6.23 ¥FLASH¥FLASH_WriteProtection¥MDK-ARM

プログラムは、内部 falsh の書き保護イネーブルと無効機能を紹介している。STM32Cube ソフトウェアの API 関数を呼び出した。

◆プログラムが実行された後に、S2ボタンを押すと、プログラムがFLASH_WRP_SECTORSの書き保護 ステータスをチェックする。チェックすると、FLASH_WRP_SECTORSの書き保護が無効の場合に、書き保 護ステータスをイネーブルに設定する;FLASH_WRP_SECTORSの書き保護がイネーブルの場合に、書き保 護ステータスを無効に設定する。





4.6.24 ¥FMC¥FMC_SDRAM¥MDK-ARM

プログラムが STM32Cube ソフトウェアの SDRAM 面の API 関数機能を紹介している。

◆まず、プログラムは HAL_SDRAM_Init 関数を呼び出して、SDRAM に占有された I / 0 への初期化を行



う。次に、SDRAM_Initialization_Sequence 関数を呼び出し、一定の順序に従って SDRAM の初期化を行う。初期化が終了、SDRAM の開始領域 0xD0000000 に 256 つの 0xA244250F を書き込む。書き込まれたデータを読み取り比較と、書き込みと読み取りのデータが同じの場合に、D2LED が点灯している。

4.6.25 ¥IWDG¥IWDG_Example¥MDK-ARM

プログラムは通常の状態でウォッチドッグが IWDG カウンタを更新する方法、犬を食べさせる時間以外に 生成した IWDG リセットを紹介する。

◆プログラムは WWDG タイムアウトを 250 ミリ秒に設定、240 ミリ秒ごとに犬を食べさせる。S2 キー を押すと、プログラムは無効な書き込みレジスタ動作を行う。これによって、プログラムが無限ループ の空操作に実行され、250 ミリ秒以内に犬を食べさせない、リセットが発生する。プログラムが再実行 して RCC_GetFlagStatus 関数を通じて、IWDG リセットが知られて、開発ボードの D2 LED が点灯する。

4.6.26 ¥LTDC¥LTDC_ColorKeying¥MDK-ARM

プログラムは液晶画面のカラーキーイング機能の起動方法と使用方法を紹介する。プログラムが実行された後、S2キーを押すと、カラーキーイング機能がオフになる。もう一度押し、オンにする。





4.6.27 ¥LTDC¥LTDC_Display_1Layer¥MDK-ARM

プログラムは、LTDC 周辺デバイスの設定方法を紹介する。表示層により、LCD 画面に BMP 画像を表示する。

◆プログラムを開発ボードにダウンロードされたあとに、写真の内容を配列型で RGB565_480x272.hに 保存される。



4.6.28 ¥LTDC¥LTDC_Display_2Layers¥MDK-ARM

プログラムは2つの表示層上に同時に各画像を表示する。画像は配列でそれぞれに RGB565_320x240.hと L8_320x240.hに保存される。実行されたあとに、液晶画面上に2つの画像が自分の斜め方向に移動する。



4.6.29 ¥PWR¥PWR_STANDBY¥MDK-ARM

このプログラムは、プロセッサが Standby モードに入り、RTC アラームによりプロセッサをウェークア ップさせる方法を紹介する。

◆プログラムが実行された後に、D2LED が点滅する。S2 キーを押すと、D2 LED は消灯し、プロセッサが Standby モードに入る。電流計で測るとボードに使用されている電流は小さく、5 秒後に RTC のアラ



ームイベントが発生し、プロセッサをウェークアップし、D2 の LED が継続に点滅する。

4.6.30 ¥PWR¥PWR_STOP¥MDK-ARM

このプログラムは、プロセッサが stop モードに入り、外部割り込み及び RTC アラームによりプロセッサ をウェークアップさせる方法を紹介する。

◆プログラムは RTC アラームにより、プロセッサが5秒ごとに stop モードに入り、5秒後にプロセッ サが RTC アラームにウェークアップされる。stop モードには、s2 キーでプロセッサをウェークアップ できる。その後、D2LED が点滅している。

4.6.31 ¥PWR¥PWR_CurrentConsumption¥MDK-ARM

プログラムは、低消費電力モードでプロセッサの消費電流を比較する。

◆ stm32f4xx_lp_modes.hファイルを開き、別々に次のマクロ定義で対応するテストを行う。

*/

*/

*/

*/

- /* #define SLEEP_MODE
- /* #define STOP_MODE
- /* #define STANDBY_MODE
- /* #define STANDBY_RTC_MODE

◆ S2 キーを押し、プロセッサが低電力モードに入らせる。電流計の電流変化を観察する。

注意:プログラムは実行されたあと、再びダウンロードすると次のエラーが起こりやすい。原因はプロセッサが低電力モードに入ったからである。コン基板の ISP を押し保持し、ボードをリセットし、プロセッサを ISP モードに入らせる。



4. 6. 32 ¥RCC¥RCC_ClockConfig¥MDK-ARM

プログラムは、動作中のプロセッサは、システムクロックソースを変更する方法を紹介する。

◆S2 キーを一度押すと、HSE と HIS スイッチングシステムクロックとの間で、システムクロックソースを切り替える。システムクロックを全部 180MHz に設定され、プログラムが実行された後、D2LED が点滅し、点滅の頻度から変化が見られない。

4.6.33 ¥RTC¥RTC_A1arm¥MDK-ARM

このプログラムは、RTC アラームの設定と生成方法を紹介する。

まず、プログラムは HAL_RTC_Init を呼び出して、RTC の関連レジスタを初期化する。次に、RTC_AlarmConfig 関数を呼び出し、RTC アラーム時刻が 02:20:30 に設定され、現在の実行時間は 02:20:00 に設定される。プ ログラムが実行され、D2LED が消灯している。30 秒後にアラーム時刻になって RTC アラーム割り込みを発生 させる。割り込みコールバック関数 HAL_RTC_ AlarmAEventCallback は D2LED が点灯させる。

4.6.34 ¥RTC¥RTC_Tamper¥MDK-ARM

プログラムは、RTC バックアップデータレジスタにデータ書き込みとデータ読み取りの方法を紹介している。

◆まず、プログラムはHAL_RTC_Init 関数を呼び出し、RTC プリスケーラと RTC データレジスタを設定 する。次に、RTC_TamperConfig 関数を読み出して、TAMPER1 pin (PC. 13)を nededge 割り込みとして設 定する。

◆20 つのバックアップデータレジスタに書き込んでから、aBKPDataReg[]配列に読み出される。S2 キ ーを押し、プログラムはデータバックアップレジスタに書き込まれたデータをクリアし、チェックする。 正確にクリアされたとチェックすると、D2LED が点滅している。

◆プログラムは、main.hファイルにLSEとLSIのクロック・ソースが選択でき、LSEを選択すること とデフォルトする。

4.6.35 ¥RTC¥RTC_Calendar¥MDK-ARM

プログラムは、RTC 日時の設定方法を紹介する。

◆まず、プログラムはHAL_RTC_Init 関数を呼び出し、RTC 周辺機器の関連レジスタを初期化する。次 に、HAL_RTCEx_BKUPRead を呼び出し、RTC のバックアップレジスタ0に 0x32F2 が書き込まれたかをチ ェックする。書き込まれない場合に、RTC_CalendarConfig 関数を呼び出して、RTC 日時を初期化する。 LCD 画面に現在時刻が表示すればいいである。





4.6.36 ¥RTC¥RTC_TimeStamp¥MDK-ARM

プログラムは、タイムスタンプの設定と使用方法を紹介する。

◆まず、プログラムは HAL_RTC_Init 関数を呼び出し、RTC プリスケーラと RTC データレジスタを設定 する。次に、RTC_TamperConfig 関数を呼び出して、TAMPER1 pin (PC.13)を nededge 割り込みとして、 RTC の日付と時刻、タイムスタンプを設定する。

◆プログラムが実行され、液晶画面に現在時刻が表示される。





4.6.37 ¥TIM¥TIM_TimeBase¥MDK-ARM

プログラムは TIM3 を設定することによって、1 秒タイマ割り込みを達成する。プログラムが実行され、D2 LED は1 秒に1回点滅する。

4.6.38 ¥TIM¥TIM_6Steps¥MDK-ARM

プログラムは、TIMを使用して6段階のPWM信号設定方法を紹介する。



◆STM32F4xxのTIM1周辺機器は次のPWM出力を設定する同時に、別のチャネル設定も変更できるが、 それらは、プロセッサのモータ制御などのアプリケーションイベントを処理する機能を実現することが できる。転流イベントは、ハードウェアともソフトウェアとも生成されることができる。この例は、シ ステムタイマによる生成する(SysTick_HandlerにHAL_TIM_GenerateEventを呼び出す)。TIM1がタイ マーモードに設定され、転流イベントが発生したあとに、事前にタイマーを設定しておく。

◆以下は TIM1 の 6 つのチャネルに出力された PWM 信号の関係図。





低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.6.39 ¥TIM¥TIM_7PWMOutput¥MDK-ARM

プログラムは YIM1 を四つの異なるデューティサイクル (50%、37.5%、25%、12.5%) の7 チャネル PWM 信号に設定される。

◆プログラムが実行され、PA8、PA10、PE11、PE14、PB13、PB14、PB15 など七つのピンが異なるデュ ーティサイクルの波形を出力する。

NO.	ピン	チャンネル	デューティサイクル
1	PA8	TIM1Channel 1	50%
2	PB13	TIM1Channel 1N	50%
3	PE11	TIM1Channel 2	37.5%
4	PB14	TIM1Channel 2N	37.5%



低価格、高品質が不可能? 日昇テクノロジーなら可能にする

5	PA10	TIM1Channel 3	25%
6	PB15	TIM1Channel 3N	25%
7	PE14	TIM1Channel 4	12.5%

4種類のデューティサイクルの波形:



4.6.40 ¥TIM¥TIM_CascadeSynchro¥MDK-ARM

プログラムは TIM 同期カスケード機能を紹介する。

◆プログラムは、TIM2、TIM3、TIM4 と3つのタイマを使用している。TIM2 をマスタタイマモードに 設定され、PWM 出力モードをオンにする。TIM2 更新イベントがトリガとして出力し、TIM2 出力周波数は 328.125 KHz、PWM デューティサイクルは 25%。

◆TIM3 は TIM2 のサブとして、TIM4 のホストとする。ITR1 は TIM2 の出力信号を取得する。TIM3 は



PWM モードをオープンして TIM4 のトリガ出力とする。TIM3 の出力周波数は 82.02KHz、PWM のデューティ・サイクルは 25%。

◆TIM4 は TIM3 のサブとする。ITR2 は TIM3 の出力信号を取得する。TIM4 も PWM モード出力信号をオ ープンにして、TIM4 の出力周波数は 20.5 Hz、PWM のデューティ・サイクルが 25%。





M 5.00µs

◆TIM3のPWM 出力端ピンはPB6

CH1== 1.00U

保存

CH1 /1.64V M Pos:0.00ps

🔁 = 82.0322kHz



低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.6.41 ¥TIM¥TIM_DMA¥MDK-ARM

プログラムは、タイマーの DMA 転送モードを紹介する。プログラムは TIM1 チャネル 3 を設定し、周波数 17.57 kHz の相補 PWM 信号を生成する。

◆プログラムは、50%、37.5%、25%三種類のデューティサイクルを定義した。これらの数値が aCCValue_Buffer[]配列に保存され、TIM1リピートカウンタ値によりデューティサイクルの振幅を確認 する。

◆プログラムを実行した後、オシロスロープを使用して PE. 13、PB. 15 ピンデューティ・サイクルでの 波形変化が見られる。





低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.6.42 ¥TIM¥TIM_DMABurst¥MDK-ARM

プログラムはDMA のバースト機能を紹介する。更新するたびに、DMA がハーフワードデータを三回目に TIM1 レジスタに書き込む。その中に、OxOFFF を ARR レジスタに書き込み、OxO000 を RCR レジスタに書き込 み、OxO555 を CCR1 レジスタに書き込む。

◆プログラムが実行された後、オシロスコープで PA10 ピンの出力波形が 5.8 KHz 周波数、33.33%デ ューティサイクルと見られる。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.6.43 ¥TIM¥TIM_ExtTriggerSynchro¥MDK-ARM

TIM 外部同期トリガカスケードモードの例である。プログラムは外部信号によってタイマーを起動および 停止する。posedge トリガ方式である。

◆プログラムは TIM1、TIM3、TIM4 タイマーを使用している。TIM1 がタイマ外部トリガモードに設定 され、外部トリガピンが PE.11(TIM1 TI2)で、ピンがゲートモードに設定される。posedge 信号で TIM1 を起動し、negedge 信号で TIM1を停止する。TIM1 CH1 (PA.08) ピンが信号の出力ポートである。

◆TIM3 は TIM1 スレーブであるだけでなく、TIM4 のホストとする。ゲートモードに設定される。TIM1 に出力された信号で TIM3 の起動・終了を制御する。TIM3 CH1 (PC.06) ピンが信号の出力ポートである。

◆TIM4はTIM3スレーブで、ゲートモードに設定される。TIM3(ITR2)に出力された信号でTIM4の起動・終了を制御する。TIM4 CH1 (PB.06)ピンが信号の出力ポートである。

◆次の図は PA. 08 ピンの出力信号。PE. 11 ピンに 1KHz の波形を追加すると、posedge 信号で波形の出力を起動し、negedge 信号で波形の出力を終了する。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.6.44 ¥TIM¥TIM_InputCapture¥MDK-ARM

プログラムは、TIMの周辺機器を使用して外部信号の周波数を測定する方法を紹介している。

◆TIM1 がインプットキャプチャモードに設定され、TIM1 チャネル2(PE.11)が外部信号として端子 に入力される。プログラムはTIM1 CC2 の割り込む回数によって、測定される信号の周波数を計算する。 エミュレータを介してプログラムを開発ボードにダウンロードし、 ◎をクリックし、シミュレーション ・インタフェースに入る。Main.cの180 行目にはブレークポイントを追加する。 ■をクリックして、フ ルスピードでプログラムを実行する。PE.11 ピンに 1kHz の信号を付加、この時点で、プログラムはブレ ークポイントに停止する。UwFrequency をマウスのカーソルでクリックし、取得した周波数値も見られ る。



	175	uwDiffCapture = 0;
	176 -	}
	177 🖕	/* Frequency computation: for this example TIMx (TIM1) is clocked by
	178 -	2xAPB2Clk */
	179	<pre>uwFrequency = (2*HAL RCC GetPCLK2Freq()) / uwDiffCapture;</pre>
∞	180	uhCaguwFrequency = 0x00000E60
	181	}

4.6.45 ¥TIM¥TIM_OCToggle¥MDK-ARM

プログラムは、TIM3 周辺機器を設定して4種類の異なる周波数の信号を生成する方法を紹介する。 TIM3 Channel 1とPC6 ピン対応して出力方形波 219.72Hz、TIM3 Channel 2とPC7 ピン対応して出力方形波 439.45Hz、、 TIM3 Channel 3とPC8 ピン対応して出力方形波 878.9Hz、TIM3 Channel 4とPC9 ピン対応して出力方形波 1757.81Hz である。





4.6.46 ¥TIM¥TIM_OnePulse¥MDK-ARM

タイマ入力端子にパルス信号を入力し、信号 posegde の時に単一パルス信号を生成する。

◆TIM4CH2 (PB.07) ピンをタイマ入力端子に設定され、TIM4_CH1 (PB.06) を単一のパルス出力端子 に設定される。次の図には、PB.07 に 1kHz の方形波信号を追加し、PB.06 の出力パルス信号は:



4.6.47 ¥TIM¥TIM_ParallelSynchro¥MDK-ARM

プログラムはパラレルモードに、TIM2、TIM3、TIM4の同期を実現する。

◆TIM2 はマスタタイマに設定され、PWM モードを使用して TIM2 更新イベントがトリガ出力とする。

◆TIM3、TIM4 がサブタイマに設定され、PWM モードを使用して、ホストの出力トリガ信号が TIM3、TIM4 タイマのカウンタを起動または停止する。



◆プログラムが実行された後に、TIM2 CH1 (PA.00)ピンが 328.125 KHz デューティサイクル 25%信号 を出力する。TIM3 CH1 (PC.06)ピンが 32.815 KHz デューティサイクル 30%信号を出力する。TIM4 CH1 (PB.06)ピンが 65.630 KHz デューティサイクル 60%信号を出力する。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.6.48 ¥TIM¥TIM_Prescaler_Selection¥MDK-ARM

プログラムは、クロックプリスケーラ選択機能が有効の場合に、TIM3のPWMモードを使用し、__ HAL_RCC_TIMCLKPRESCALER()関数が2倍出力周波数を出力することを紹介する。

◆プログラムを実行した後、PC.06 (TIM3_CH1) ピンが 20K Hz の精密出力波形を出力する。





4.6.49 ¥TIM¥TIM_PWMIntput¥MDK-ARM

プログラムはTIM4を使用して、外部入力信号の周波数とデューティサイクルを測定する。

◆TIM4CH2 (PB.07) ピンは外部入力信号を得るために使用される。TIM4 CC2 に生成された割り込みは コールバック関数 HAL_TIM_IC_CaptureCallback が呼び出される。コールバック関数で外部入力信号周 波数のデューティ・サイクルを計算することができ、周波数、とデューティサイクルとプロセッサ周波 数との関係は:

The "uwFrequency" variable contains the external signal frequency: TIM4 counter clock = SystemCoreClock / 2, Frequency = TIM4 counter clock / TIM4_CCR2 in Hz.

The "uwDutyCycle" variable contains the external signal duty cycle: DutyCycle = (TIM4_CCR1*100)/(TIM4_CCR2) in %.

The minimum frequency value to measure is (TIM4 counter clock / CCR MAX) = (180MHz/2) / 65535 = 1373 Hz

◆エミュレータを介してプログラムを開発ボードにダウンロードし、◎をクリックし、シミュレーション・インタフェースに入る。Main.cの188行目にはブレークポイントを追加する。■をクリックして、フルスピードでプログラムを実行する(誤動作を避ける)。PE.11 ピンに1kHzの信号を付加(テスト信号が1373 Hz以上)、この時点で、プログラムはブレークポイントに停止する。UwFrequencyをマウスのカーソルでクリックし、取得した周波数値も見られる。

📩 main.c	readme.txt startup_stm32f429xx.s
183	/* Duty cycle computation */
184	<pre>uwDutyCycle = ((HAL_TIM_ReadCapturedValue</pre>
185	uwDutyCycle = 0x00000124
186	/* uwFrequency computation

HI SUN	
	不可能への挑戦

	186]	/* uwFrequency computation		
	187		TIM4 counter clock = (RCC Clocks.HCLK Frequency)/2 */		
≯	188		<pre>uwFrequency = (HAL_RCC_GetHCLKFreq())/2 / uwIC2Value;</pre>		
	189	}	uwFrequency = 0x00000E60		

4.6.50 ¥TIM¥TIM_PWMOutput¥MDK-ARM

プログラムは STM32Cube パッケージの API 関数を呼び出すことによって、TIM3 が4 チャネルの異なるデュ ーティサイクル信号を生成させる。

◆TIM3 出力信号周波数は全部 10KHz、TIM3_CCRx の値を設定することによってデューティサイクルを 取得する。デューティサイクルの計算方法:(TIM3_CCRx/ TIM3_ARR + 1)* <u>100</u>。

◆プログラムが実行された後、オシロスコープによって PC. 06 (TIM3_CH1) PC. 07 (TIM3_CH2)、

PC.08 (TIM3_CH3)を測定する。PC.09 (TIM3_CH4)でデューティサイクルはそれぞれ 75%、50%、33.3%、25%。





低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.6.51 ¥WWDG¥WWDG_Example¥MDK-ARM

プログラムは、ウォッチドッグが通常の状態で WWDG カウンタを更新する方法、及びドッグを食べさせる時間以外に発生された WWDG リセットを紹介する。

◆プログラムは WWDG タイムアウトを 41.94 ミリ秒に設定、40 ミリ秒ごとに犬を食べさせる。S3 の真 ん中キーを押すと、プログラムは無効な書き込みレジスタ動作を行う。これによって、プログラムが無 限ループの空操作に実行され、41.94 ミリ秒以内に犬を食べさせない、リセットが発生する。プログラ ムが再実行して_HAL_RCC_GET_FLAG 関数を通じて、WWDG リセットが知られて、開発ボードの D2 LED が を点灯する。

◆プログラムを開発ボードにダウンロードされた時に、D2 LED が消灯している。S2 キーを押すと、 プロセッサが WWDG リセットを生成し、D2 LED が点灯する。

4.7 ¥Code¥STM32Cube_FW_F4_V1.1.0¥Projects¥STM324x9I_EVAL フ

オルダのプログラム説明

4.7.1 ¥Applications¥Display¥LTDC_AnimatedPictureFromSDCard¥MDK-ARM

プログラムは、SDカードの複数の写真を特定な時間順番に表示させ、動的表示を実現する。4.1.27と同じである。

◆テスト前に、予めに

¥Code¥STM32Cube_FW_F4_V1. 1. 0¥Projects¥STM324x9I_EVAL¥Applications¥Display¥LTDC_AnimatedPictureFr omSDCard¥Media ディレクトリの BACK と TOP ファイルを SD カードのルートディレクトリに保存する。プログラムが実行されたあと、LCD 画面の左上隅にはひらひら蝶がいる。



4.7.2 ¥Applications¥Display¥LTDC_PicturesFromSDCard¥MDK-ARM

プログラムは、SD カードの BMP 写真を LCD 画面に表示する。4.1.26 と同じ。

◆テスト前に、

¥Code¥STM32Cube_FW_F4_V1.1.0¥Projects¥STM324x9I_EVAL¥Applications¥Display¥LTDC_PicturesFrom SDCard ディレクトリの PICT ファイルを SD カードのルートディレクトリに保存する。プログラムが実行 されたあとに、S2 キーを押すと、次の写真を訪問する。



4.7.3 ¥Applications¥Display¥LTDC_Paint¥MDK-ARM

プログラムはスケッチブックの機能を実現する。プログラムは、タッチスクリーンでブラシのサイズと色を選択して、完成した写真を SD カードに保存する。

◆テストの前に、SD カードを挿し込む。プログラムが実行された後、LCD 画面上にタッチスクリーンの左上隅と右下隅のドットでスクリーンを補正すると提示されている。





◆タッチスクリーンがキャリブレーションされた後、次の画面が現れた。タッチスクリーン左側のラ インがブラシのサイズを選択し、タッチスクリーン下隅のドットがブラシのサイズを選択し、画面の右 下隅には選択されたカラーとサイズを表示する。







注意:全てのSDカードが利用できる保証はできない。本テストはKingston 8G SDカードで実行。

4.7.4 ¥Applications¥FatFs¥FatFs_RAMDisk¥MDK-ARM

プログラムは、FatFsシステムファイルを使用し、SDRAMがディスクとして、書式設定、新規ファイル、ファイル開き、ファイル記述し、ファイル読み取り、ファイル閉じるや他の操作などを実現する。

◆プログラムは上述した動作を実現し、書き込みと読み出しファイルの内容が同じの場合に、D2 LED が点灯している。

4.7.5 ¥Applications¥FatFs¥FatFs_RAMDisk_RTOS¥MDK-ARM

プログラムは、FreeRTOS 操作システムに FatFs ファイルシステムを使用して、SDRAM がディスクとして書 式設定、新規ファイル、ファイル開き、ファイル記述、ファイル読み取り、ファイル閉じや他の操作などを 実現する。

◆プログラムは実行された後、上述した動作を行うと、書き込みと読み取りファイルの内容が同じの 場合に、D2 LED が点灯している。





4.7.6 ¥Applications¥FatFs¥FatFs_uSD¥MDK-ARM

プログラムは、FreeRTOS 操作システムに FatFs ファイルシステムを使用して、SD カードがディスクとし て書式設定、新規ファイル、ファイル開き、ファイル記述、ファイル読み取り、ファイル閉じや他の操作な どを実現する。

◆試験の前に、SDカードを実装する。プログラムを実行したあと、上述の動作を行う。書き込みと読み取りファイルの内容が同じの場合に、D2 LED が点灯している。カードリーダーを介して SD カードに STM32. TXT という名前のテキストファイルが作成された。 SD カードがフォーマットされたので、テストの前に SD カードの内容をバックアップする方が良いである。

注意:全てのSDカードが利用できる保証はできない。本テストはKingston 8G SDカードご実行。

4.7.7 ¥Applications¥FatFs¥FatFs_uSD_RTOS ¥MDK-ARM

プログラムはFatFsファイルシステムを使用して、SDカードがディスクとして書式設定、新規ファイル、ファイル開き、ファイル記述、ファイル読み取り、ファイル閉じや他の操作などを実現する。

◆試験の前に、SD カードを実装する。プログラムを実行したあと、上述の動作を行う。書き込みと読み取りファイルの内容が同じの場合に、D2 LED が点灯している。カードリーダーを介して SD カードに STM32. TXT という名前のテキストファイルが作成された。 SD カードがフォーマットされたので、テスト前に SD カードの内容をバックアップする方が良いである。

注意:全てのSDカードが利用できる保証はできない。本テストはKingston 8G SDカードで実行。

4.7.8 ¥Applications¥FatFs¥FatFs_USBDisk¥MDK-ARM

プログラムは FatFs ファイルシステムを使用して、USB メモリがディスクとして書式設定、新規ファイル、ファイル開き、ファイル記述し、ファイル読み取り、ファイル閉じるや他の操作などを実現する。

◆テスト前に、USB メモリを J8 インタフェースに接続する。プログラムを実行したあと、上述の動 作を行う。書き込みと読み取りファイルの内容が同じの場合に、D2 LED が点灯している。PC を介して USB メモリに新規 STM32、TXT ファイルが作成された。



注意:

1. プログラムは全ての USB メモリが利用できる保証はできない。本テストは普通のカードリーダーで Kingston 8G SD カード挿入で実行。

2. 開発ボードはフルスピード USB インタフェースのみが提供するので、フルスピード USB プロジェクト (STM324x9I-EVAL_USBH-FS)を使用してください。





📀 🕮 🎬 🥪 🔜 🙀	STM324x9I-EVAL_U			
Project	STM324x9I-EVAL_USB			
	STM324x9I-EVAL_USBH STM324x9I-EVAL_USBH			
Application/MDK-ARM				

4.7.9 ¥Applications¥FatFs¥FatFs_USBDisk_RTOS ¥MDK-ARM

プログラムは FreeRTOS 操作システムに FatFs ファイルシステムを使用して、USB メモリがディスクとして 書式設定、新規ファイル、ファイル開き、ファイル記述、ファイル読み取り、ファイル閉じや他の操作など を実現する。

●試験の前に、USBメモリをJ8 インタフェースに接続する。プログラムを実行したあと、上述の動作を行う。書き込みと読み取りファイルの内容が同じの場合に、D2 LED が点灯している。PC を介して USB メモリに新規 STM32. TXT ファイルが作成された。



Project STM324x9I-EVAL_USBH STM324x9I-EVAL_USBH STM324x9I-EVAL_USBH STM324x9I-EVAL_USBH Application/MDK-ARM

4.7.10 ¥Applications¥FatFs¥FatFs_USBDisk_MultipleAccess_RTOS¥MDK-ARM

プログラムは FreeRTOS 操作システムに FatFs ファイルシステムを使用して、USB メモリがディスクとして それぞれ2スレッドで書式設定、新規ファイル、ファイル開き、ファイル記述、ファイル読み取り、ファイ ル閉じや他の操作などを実現する。

● 試験の前に、USB メモリを J8 インタフェースに接続する。プログラムを実行したあと、上述の動作を行う。書き込みと読み取りファイルの内容が同じの場合に、D2 LED が点灯している。PC を介して USB メモリに新規 STM32_1. TXT と STM32_2. TXT ファイルが作成された。



Removable Disk (H:)					
New folde	r				
	Name		Date modified	Туре	
	STM32_1			Text Document	
	STM32_2			Text Document	

注意:

1. プログラムは全ての USB メモリが利用できる保証はできない。本テストは普通のカードリーダーで Kingston 8G SD カード挿入で実行。

4. 開発ボードはフルスピード USB インタフェースのみが提供するので、フルスピード USB プロジェクト (STM324x9I-EVAL_USBH-FS)を使用してください。

🔗 🕮 🎬 🥪 🔜 🙀	STM324x9I-EVAL_U		
Project	STM324x9I-EVAL_USBF		
-	STM324x9I-EVAL_USBH		
STM324x9I-EVAL_USB			
Application/MDK-ARM			

4.7.11 ¥Applications¥FatFs¥FatFs_MultiDrives¥MDK-ARM

プログラムは FatFs ファイルシステムを使用して、SD カードと SDRAM で書式設定、新規ファイル、ファ イル開き、ファイル記述し、ファイル読み取り、ファイル閉じるや他の操作などを実現する。

◆試験の前に、SD カードを実装する。プログラムを実行したあと、SD カードと SDRAM がディスクと して、上述の動作を行う。書き込みと読み取りファイルの内容が同じの場合に、D2 LED が点灯している。 カードリーダーを介して SD カードに STM32. TXT という名前のテキストファイルが作成された。 SD カー ドがフォーマットされたので、テストの前に SD カードの内容をバックアップする方が良いである。

注意:プログラムは全ての SD カードが利用できる保証はできない。本テストは Kingston 8G SD カードで 実行。

4.7.12 ¥AppLications¥LwIP¥LwIP_HTTP_Server_Raw¥MDK-ARM

プログラムはLWIP ネットワークプロトコルスタックを呼び出して、ウェブで AD 値を読み取り、LED を制 御できる。

◆Main.h はネットワーク方面のマクロ定義を保存し、DHCP サービスのオープン、MAC アドレス、IP アドレス サブネットマスク、ゲートウェイ、ネットワーク物理層モードは、このファイル上に設定さ れている。

◆プログラムは DHCP をオンにして自動的に IP アドレスを取得することをディフォルトする。開発ボードに電源を入れ、LCD 画面には割り当てられた IP アドレスが表示される。今回の IP アドレスが 192.168.1.104 に割り当てられた。






低価格、高品質が不可能? 日昇テクノロジーなら可能にする

X Cocyle X <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>							
X Coogle Vig Search - More > em Sign In 4 - X Vintor A Image: A Mage: A Mage Image: A Mage: A	<pre>http://192.168.1.104/leds.cgr?led=1</pre>	STM32F4xxL	ED ×			<u> </u>	
STM32F4xx Leds control Home page Led control ADC status bar This page allows you to control the four LEDs: LED1, LED2, LED3 and LED4 located in the STM324x91-EVAL board. To put on/off a LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs configuration. Finally check in the STM324x91-EVAL board that you get the desired LEDs are putted on/off. STM32 Webserver LEDs Control	x Google	🔻 🚰 Search 📲 More » 🗵	🖽 🤄 Sign In 🔧 🔹 🗶	Vorton •	🕈 🔤 网页安全	• 🛃 共享 • 📮] 访问例:
Home page Led control ADC status bar This page allows you to control the four LEDs: LED1, LED2, LED3 and LED4 located in the STM324x91-EVAL board. To put on/off a LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs configuration. Finally check in the STM324x91-EVAL board that you get the desired LEDs are putted on/off. STM32 Webserver LEDs Control	STM22EAvy Leds control						1
Home page Led control ADC status bar This page allows you to control the four LEDs: LED1, LED2, LED3 and LED4 located in the STM324x91-EVAL board. To put on/off a LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs configuration. Finally check in the STM324x91-EVAL board that you get the desired LEDs are putted on/off. STM32 Webserver LEDs Control LED1 LED2 LED3 LED3 LED4 Send All rights reserved © 2013 STMicroelectronics							
This page allows you to control the four LEDs: LED1, LED2, LED3 and LED4 located in the STM324x91-EVAL board. To put on/off a LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs configuration. Finally check in the STM324x91-EVAL board that you get the desired LEDs are putted on/off. STM32 Webserver LEDs Control LED1 LED2 LED3 LED4 Send All rights reserved © 2013 STMicroelectronics	Home page	Led control	A	DC status bar			
his page allows you to control the four LEDs: LED1, LED2, LED3 and LED4 located in the STM324x9I-EVAL board. To put on/off LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs onfiguration. Finally check in the STM324x9I-EVAL board that you get the desired LEDs are putted on/off. TM32 Webserver LEDs Control LED1 LED2 LED3 LED4 Send All rights reserved © 2013 STMicroelectronics							
his page allows you to control the four LEDs: LED1, LED2, LED3 and LED4 located in the STM324x9I-EVAL board. To put on/off LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs onfiguration. Finally check in the STM324x9I-EVAL board that you get the desired LEDs are putted on/off. TM32 Webserver LEDs Control LED1 LED2 LED3 LED4 Send All rights reserved © 2013 STMicroelectronics							
LED you have to check/uncheck its corresponding checkbox. Then you have to click on "Send" button to submit the new LEDs onfiguration. Finally check in the STM324x9I-EVAL board that you get the desired LEDs are putted on/off. TM32 Webserver LEDs Control LED1 LED2 LED3 LED4 Send All rights reserved © 2013 STMicroelectronics	his page allows you to control the four LEDs: LE	D1. LED2. LED3 and LED4 located in	the STM324x9I-EVAL	poard. To put on	/off		
Onfiguration. Finally check in the STM324x91-EVAL board that you get the desired LEDs are putted on/off.	LED you have to check/uncheck its correspondi	ing checkbox. Then you have to click	on "Send" button to su	ibmit the new LE	Ds		
STM32 Webserver LEDs Control LED1 LED2 LED3 LED4 Send All rights reserved © 2013 STMicroelectronics	onfiguration. Finally check in the STM324x9I-EV	/AL board that you get the desired LEI	Os are putted on/off.				
STM32 Webserver LEDs Control							
STM32 Webserver LEDs Control							
LED1 LED2 LED3 LED4 Send All rights reserved © 2013 STMicroelectronics	STM32 Webserver LEDs Control						
□ LED1 □ LED2 □ LED3 □ LED4 Send All rights reserved © 2013 STMicroelectronics							
LED3 LED4 Send All rights reserved © 2013 STMicroelectronics							
LED3 LED4 Send All rights reserved © 2013 STMicroelectronics							
LED3 Send All rights reserved © 2013 STMicroelectronics							
Send All rights reserved © 2013 STMicroelectronics							
Send All rights reserved © 2013 STMicroelectronics							
All rights reserved © 2013 STMicroelectronics	Send						
All rights reserved © 2013 STMicroelectronics	00110						
All rights reserved © 2013 STMicroelectronics							
All rights reserved © 2013 STMicroelectronics							
All rights reserved © 2013 STMicroelectronics							
	All rights rese	rved © 2013 STMicroelectronics		_			
	All lights rese	Trea © 2015 STIMEIOElectronics		•			

A March 192 168 1 104/STM32E4vyADC shtml				
x Google	Search - More » 🖽	Sign In 🔧 - 🗶 🧭	Norton 🔹 🖻 🔤 网页安全	≥ • ■共享 • 🔲 访问係 🗘
STM32F4xx ADC Conversion			X	
Home page	Led control	ADC :	status bar	
This page allows you to get continuously the ADC 1 STM324x91-EVAL board's potentiometer. The ADC v You could check this by changing the potentiometer page, with the new converted value. ADC Converted Value 1678 mv	Channel 5 analog input converted value alue is updated, automatically, each 1s position and check that the ADC value i	This ADC Channel is with the last convert so updated, by an aut	s connected to the ed ADC 1 Channel 5 value tomatic refresh of this	3.

◆web コンテンツはチップの内蔵 Flash に保存される。makefsdata.exe ソフトウェアはページファイルを fsdata.c ファイルに変換し、¥Code¥STM32Cube_FW_F4_V1.1.0¥Utilities¥PC_Software ディレクトリの makefsdata.exe ファイルを

¥Code¥STM32Cube_FW_F4_V1.1.0¥Projects¥STM324x9I_EVAL¥Applications¥LwIP ¥LwIP_HTTP_Server_Raw ディレクトリにコピーする。コマンドラインを開き、makefsdata.exeファイルを実行すると、fsフォ ルダの Web と写真ファイルをシングルな fsdata.cファイルに変換される。それから、fsdata.cファイ ルを

¥Code¥STM32Cube_FW_F4_V1.1.0¥Projects¥STM324x9I_EVAL¥Applications¥LwIP¥LwIP_HTTP_Server_Raw ¥Src ディレクトリにコピーし、プロジェクトファイルを再コンパイルする。Web ファイルがUltraEdit で編集する。



C:\Windows\system32\cmd.exe				
processing subdirectory /STM32F ¹ processing /STM32F4xxADC.files/ processing /STM32F4xxADC.files/ processing /STM32F4xxADC.files/ processing /STM32F4xxADC.files/ failed to get file tupe for extr	HXXADC.files/ colorschememapping.xml ilelist.xml header.htm chemedata.thmx posion "thmx".usino default	Ē		
processing subdirectory /STM32F4 processing /STM32F4xx_files/logo processing /STM32F4xx_files/Stm3 processing /STM32F4xx_files/Stm3 processing /index.htm1 processing /STM32F4xxADC.shtm1 processing /STM32F4xxLED.htm1	Hxx_files/).jpg jf 2.jpg			
Creating target file				
Processed 11 files - done				
i occocca i i i i i i i i conce.				
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP HTTP Server F	132Cube_FW_F4_V1.1.0\Projects Naw>makefsdata.exe	<pre>\STM324x9I_EVAL\App1</pre>		-
E:\Mywork\ARM\\$TM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F	132Cube_FW_F4_U1.1.0\Projects Naw>makefsdata.exe 2014/2/277:41	-\\$TM324x9I_EUAL\App1 ↓		
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F L EWARM L Fs	132Cube_FW_F4_U1.1.0\Projects Raw>makefsdata.exe 2014/2/277:41 2014/3/319:45	File folder		
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F EWARM Fs Inc	132Cube_FW_F4_U1.1.0\Projects Raw>makefsdata.exe 2014/2/277:41 2014/3/319:45 2014/3/2616:33	File folder		
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F L EWARM Fs Inc MDK-ARM	132Cube_FW_F4_U1.1.0\Projects taw>makefsdata.exe 2014/2/27 7:41 2014/3/31 9:45 2014/3/26 16:33 2014/3/30 22:38	File folder File folder File folder File folder File folder File folder)
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F EWARM Fs Inc MDK-ARM Src	132Cube_FW_F4_U1.1.0\Projects Raw>makefsdata.exe 2014/2/27 7:41 2014/3/31 9:45 2014/3/26 16:33 2014/3/30 22:38 2014/3/26 16:31	File folder File folder File folder File folder File folder File folder File folder)
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F EWARM Fs Inc MDK-ARM Src TrueSTUDIO	132Cube_FW_F4_U1.1.0\Projects Raw>makefsdata.exe 2014/2/27 7:41 2014/3/31 9:45 2014/3/26 16:33 2014/3/26 16:31 2014/3/26 16:31 2014/2/27 7:41	File folder File folder File folder File folder File folder File folder File folder File folder		
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F EWARM Fs Inc MDK-ARM Src TrueSTUDIO	<pre>132Cube_FW_F4_U1.1.0\Projects 2014/2/27 7:41 2014/3/31 9:45 2014/3/26 16:33 2014/3/20 16:31 2014/3/26 16:31 2014/2/27 7:41 2014/2/27 7:41 2014/3/31 9:43</pre>	File folder File folder File folder File folder File folder File folder File folder File folder WinRAR 压缩文件	36 KB	
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F EWARM Fs Inc MDK-ARM Src TrueSTUDIO Fs fs	132Cube_FW_F4_U1.1.0\Projects 2014/2/27 7:41 2014/3/31 9:45 2014/3/26 16:33 2014/3/30 22:38 2014/3/26 16:31 2014/2/27 7:41 2014/3/31 9:43 2014/3/31 9:43	▲STM324×91_EUAL\App1 File folder File folder File folder File folder File folder File folder WinRAR 压缩文件 UltraEdit Docume	36 KB 441 KB	
E:\Mywork\ARM\STM32F4x9\Code\STM ications\LwIP\LwIP_HTTP_Server_F EWARM EWARM Fs Inc MDK-ARM Src TrueSTUDIO Fs fsdata imakefsdata	132Cube_FW_F4_U1.1.0\Projects Raw>makefsdata.exe 2014/2/27 7:41 2014/3/31 9:45 2014/3/26 16:33 2014/3/30 22:38 2014/3/26 16:31 2014/2/27 7:41 2014/2/27 7:41 2014/3/31 9:43 2014/3/31 9:48 2014/3/31 9:48	File folder File folder File folder File folder File folder File folder File folder File folder WinRAR 压缩文件 UltraEdit Docume Application	36 КВ 197 КВ	

4.7.13 ¥Applications¥LwIP_HTTP_Server_Netconn_RTOS¥MDK-ARM

プログラムは 1wIP ネットワークプロトコルスタックと FreeRTOS 操作システムを呼び出し、ネットワーク 接続タイプが netconn である。Web で開発ボードの Web サーバーを訪問し、操作システムのスレッドの呼び 出しをチェックする。

◆DHCP機能でIPアドレスを取得することとデフォルトする。IPアドレス、ゲートウェイがmain.h に変更される。





	main.h	thern	etif.c 👔	main.c		stm324x9i_	eval_io.c		stm324x	9i_eval_lcd.c
43	3 /*	Expo	rted	cons	sta	nts ·				
44	#de	efine	USE	LCD			/*	ena	ble	LCD
45	i #de	efine	USE	DHCI	2		/*	ena	ble	DHCP,
46	5									
47	/*:	Stati	c IP	ADDI	RES	S*/				
48	#de	efine	IP_A	ADDR()	192				
49) #de	efine	IP A	ADDR	1	168				
50) #de	efine	IP A	ADDR	2	1				
51	. #de	efine	IP A	ADDR	3	108				
52	2		_							
53	3 /*]	NETMA	SK*/							
54	#de	efine	NETN	1 ASK	AD	DR0	25	55		
55	5 #de	efine	NETN	1ASK	AD	DR1	25	55		
56	5 # de	efine	NETN	1 ASK	AD	DR2	25	55		
57	/ #de	efine	NETN	1ASK	AD	DR3	0			
58	3			_						
59) /*(Gatew	ay Ad	ddre	5s*	/				
60) #de	efine	GW A	ADDR()	192				
61	. #de	efine	GW A	ADDR	1	168				
62	2 #de	efine	GW A	ADDR	2	1				
63	8 #de	efine	GW A	ADDR	3	1				
	▲ 7°	ロガラ	ー 人を開え	公ボー	ドレ	・ラウン	/11-	K1	てから	11+2

◆プログラムを開発ボードにラウンロードしてから、リセットを行う。LCD 画面で割り当てられた IP アドレスを取得する。今回が割り当てられた IP アドレスが 192. 168. 1. 104。WEB に http://192. 168. 1. 104/を記入し、訪問できる。





低価格、高品質が不可能? 日昇テクノロジーなら可能にする



4.7.14 ¥Applications¥LwIP¥LwIP_HTTP_Server_Socket_RTOS¥MDK-ARM

プログラムは 1wIP ネットワークプロトコルスタックと FreeRTOS 操作システムを呼び出し、通信方式は socket である。Web で開発ボードの Web サーバーを訪問し、操作システムのスレッドの呼び出しをチェック する。4.7.13 と同じである。

4.7.15 ¥Applications¥LwIP¥LwIP_TCP_Echo_Server¥MDK-ARM

TCP 方式の echo サーバプログラムである。プログラムがクライアントから受信したデータをクライアント に返信する。

◆プログラムは特定な IP アドレスを使用している。pc と開発ボード通信のサブネットマスク、ゲートウェイと一致する。次の図の DEST_IP_ADDR はテスト用 PC の IP アドレスと一致しなければならない。



(今回のテスト用の IP アドレスが 192.168.1.103 である)、main.hに IP アドレスとゲートウェイを変 更する。

<u> </u>	ain.h 📩 main.c	thernetif.c		
43				
44	#define	DEST_IP_ADDR0	192	
45	#define	DEST_IP_ADDR1	168	
46	#define	DEST_IP_ADDR2	1	
47	#define	DEST_IP_ADDR3	103	
48				
49	#define	DEST_PORT	7	
50				
51	/*Statio	C IP ADDRESS: IP	ADDR0.IP	
52	#define	IP_ADDR0 192		
53	#define	IP_ADDR1 168		
54	#define	IP_ADDR2 1		
55	#define	IP_ADDR3 108		
56				
57	/*NETMAS	3K*/		
58	#define	NETMASK_ADDR0	255	
59	#define	NETMASK_ADDR1	255	
60	#define	NETMASK_ADDR2	255	
61	#define	NETMASK_ADDR3	0	
62	41.00.1			
63	/*Gatewa	ay Address*/		
64	#define	GW_ADDR0 192		
65	#define	GW_ADDRI 168		
	◆開発ボー	ードにプログラムをタ	でウンロード	してからリセットを行う。¥PC-Software¥TCPUDPDbg ディ!
ク	ィトリの TC	PUDPDbg.exe を実行し	, 「Create	Connn」をクリックして開発ボードに接続する。
🏂 Т(CP&UDP-De	bug	Y	
Or	perate(O)	View(V) Windows(W	/) Help(H)	Language
-	/			
	CreateConr	ın 💽 CreateServer 🐰	StartServe	r 28 🐼 🙀

Properties

Server Mode

◆次の図には、開発ボードの IP アドレスは 192.168.1.108 である。IP テキストボックスには割り当 てられた IP アドレスを入力し、ポート 7、「Create」をクリックして確認する。

ąх



Create Connection TCP • Type: DestIP: 192.168.1.108 Port: 7 Specia 7 LocalPort 🔘 Auto AutoConn: Eve σ Send When Conn: Eve ms Cancel Create X TCP&UDP??工具 - [192.168.1.108:7] Operate(\underline{O}) View(\underline{V}) Windows(\underline{W}) Help(\underline{H}) Language 실 CreateConnn 🔇 CreateServer | 🐰 StartServer 迷 🕢 | 👷 Connect 🐭 | 🛬 DisconnAll | 💥 DeleteConn 🗞 | 🧕 4 Þ 🗙 Properties Ψ× / 🎾 192.168.1.108:7 🖃 📕 Client Mode DestIP AtuoSend Eve 100 Send Stop 192.168.1.108:7 192.168.1.108 🔲 Send Hex 🥅 Send File 🔽 Send Received 📲 Server Mode Clear Option BroadOption DestPort: 7 abcdefghijklmnopgrstuvwxyz1234567890abcdefghijklmnopgrstuvwxyz1234567890abcdef 🔽 LocalPort ghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwxyz1234567890abcdefghijkl 17 mnopqrstuvwxyz1234567890abcdefg<mark>hi</mark>jklmnopqrstuvwxyz1234567890abcdefghijklmnopqr stuvwxyz1234567890abcdefghijklm<mark>no</mark>pqrstuvwxyz1234567890abcdefghijklmnopqrstuvwx Type TCP • yz1234567890 AtuoConn Eve 0 AutoSend Eve 0 Connect Rec StopShow Clear Save Option ShowHex Count 🔲 Save(In Time) Send R

◆ 「Connect」をクリックして開発ボードを接続する。送信エリアにデータを入力し、「Send」を押 して開発ボードにデータを送信する。ボードは受信したデータを戻り、受信エリアに表示された。

4.7.16 ¥Applications¥LwIP¥LwIP_UDP_Echo_Server¥MDK-ARM

UDP 方式の echo サーバプログラムである。プログラムはクライアントからの UDP 接続により送信されたデ ータを直接にクライアントに返信する。

◆プログラムは特定な IP アドレスを使用している。pc と開発ボード通信のサブネットマスク、ゲー



トウェイと一致する。次の図の DEST_IP_ADDR はテスト用 PC の IP アドレスと一致しなければならない。 (今回のテスト用の IP アドレスが 192.168.1.103 である)、main.hに IP アドレスとゲートウェイを変 更する。

	nain.h 🖹 main.c 🔝 ethernetif.c	
43		
44	#define DEST IP ADDR0	192
45	#define DEST_IP_ADDR1	168
46	#define DEST_IP_ADDR2	1
47	<pre>#define DEST_IP_ADDR3</pre>	103
48		
49	#define DEST_PORT	7
50		
51	/*Static IP ADDRESS:	IP_ADDR0.IP
52	#define IP_ADDR0 19	2
53	#define IP_ADDR1 16	8
54	<pre>#define IP_ADDR2 1</pre>	
55	#define IP_ADDR3 10	8
56		
57	/*NETMASK*/	
58	#define NETMASK_ADDR0	255
59	#define NETMASK_ADDR1	255
60	#define NETMASK_ADDR2	255
61	#define NETMASK_ADDR3	0
62		
63	/*Gateway Address*/	
64	#define GW_ADDR0 19	2
65	#define GW_ADDR1 16	8
	◆開発ボードにプログラムを	をダウンロート

「CreateConnn」をクリックして開発ボードに接続する。

✗ TCP&UDP-Debug			
Operate(<u>O)</u> View(<u>V</u>) Windows(<u>W</u>) Help	р(<u>Н</u>)	Langu	Jage
🔚 CreateConnn 🔕 CreateServer 🕺 Start:	Server	8	8 3
Properties 7 ×			
Client Mode			
🛄 🖪 Server Mode			

◆次の図には、開発ボードの IP アドレスは 192. 168. 1. 108 である。 IP テキストボックスには割り当 てられた IP アドレスを入力し、ポート7、「Create」をクリックして確認する。



Create Connection TCP • Type: DestIP: 192.168.1.108 Port: 7 ⊙ Specia 7 LocalPort C Auto AutoConn: Eve σ Send When Conn: Eve ms Cancel Create ✗ TCP&UDP??工具 - [192.168.1.108:7] Operate(<u>O</u>) View(<u>V</u>) Windows(<u>W</u>) $Help(\underline{H})$ Language × 🔚 CreateConnn 🔌 CreateServer | 🐰 StartServer 迷 🐼 | 🛫 Connect 🐲 | 🌺 DisconnÁll | 💥 DeleteConn 👹 🔯 | 😤 _ $\triangleleft \triangleright \mathbf{x}$ Properties **φ** × / 🎾 192.168.1.108:7 🖃 🔳 Client Mode DestIP Send AtuoSend Eve 100 Stop Send 192.168.1.108:7 192.168.1.108 Send Hex Send File Send Received 📕 Server Mode Clear Option BroadOption DestPort: 7 abcdefghijklmnopgrstuvwxyz1234567890abcdefghijklmnopgrstuvwxyz1234567890abcdef 🔽 LocalPort ghi jklmnopqrstuvwxyz<mark>12</mark>34567890abcdefghi jklmnopqrstuvwxyz1234567890abcdefghi jkl 17 mnopqrstuvwxyz1234567<mark>89</mark>0abcdefghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqr stuvwxyz1234567890abc<mark>de</mark>fghijklmnopqrstuvwxyz1234567890abcdefghijklmnopqrstuvwx Type TCP z1234567890 🗌 AtuoConn Eve 0 AutoSend Eve 0 Connec Rec StopShow Clear Save Option ShowHex ****** 🔲 Save(In Time) Send 0 Recv Clear

◆「Connect」をクリックし、開発ボードに接続する。送信エリアにデータを入力し、「Send」を押 して開発ボードにデータを送信する。ボードは受信したデータを戻り、受信エリアに表示された。無線 ネットワークで開発ボードを接続する通信速度試験である。

4.7.17 ¥Applications¥LwIP¥LwIP_TCP_Echo_Client¥MDK-ARM

プログラムは、1wIP ネットワークプロトコルを呼び出して TCP 方式の echo クライアントプログラムを実現する。

◆プログラムは特定な IP アドレスを使用している。pc と開発ボード通信のサブネットマスク、ゲー



トウェイと一致する。次の図の DEST_IP_ADDR はテスト用 PC の IP アドレスと一致しなければならない。 (今回のテスト用の IP アドレスが 192.168.1.103 である)、main.hに IP アドレスとゲートウェイを変 更する。

<u></u> m	ain.h 📩 main.c	ethernetif.c		
43				-
44	#define	DEST_IP_ADDR0	192	
45	#define	DEST_IP_ADDR1	168	
46	#define	DEST_IP_ADDR2	1	
47	#define	DEST_IP_ADDR3	103	
48				
49	#define	DEST_PORT	7	
50				
51	/*Statio	C IP ADDRESS: IP	ADDR0.IE	
52	#define	IP_ADDR0 192		
53	#define	IP_ADDR1 168		
54	#define	IP_ADDR2 1		
55	#define	IP_ADDR3 108		
56				
57	/*NETMAS	SK*/		
58	#define	NETMASK_ADDR0	255	
59	#define	NETMASK_ADDR1	255	
60	#define	NETMASK_ADDR2	255	
61	#define	NETMASK_ADDR3	0	
62	419.1			
63	/*Gatewa	ay Address*/		
64	#define	GW_ADDR0 192		
65	#define	GW_ADDRI 168		
	◆開発ボー	ードにプログラムをタ	ブウンロート	礼てからリセットを行う。コマンドラインに echotool /
te	cn /s を実	行し、ポート7を介す	してサーバー	ーレクライアント接続と提示される。S2 キーを押すと、

′p を実行し、ボート сp echotool に開発ボードから送信したデータを表示する。

cs. echotool.exe /p tcp /s	
E:\Mywork\ARM\STM32F4x9\STM32F4x9IG-DK-A-CD\PC-Software>echotool.exe	/p tcp /s 📋
Waiting for TCP connection on port 7. Press any key to exit.	
Client 192.168.1.108:49153 accepted at 12:03:26 12:03:26 received [sending tcp client message 0]	
Session closed by peer. Waiting for TCP connection on port 7. Press any key to exit.	
Client 192,168.1.108:49154 accepted at 12:03:26 12:03:26 received [sending tcp client message 1]	
Session closed by peer. Waiting for TCP connection on port 7. Press any key to exit.	
Client 192.168.1.108:49155 accepted at 12:03:27 12:03:27 received [sending tcp client message 2]	
Session closed by peer. Waiting for TCP connection on port 7. Press any key to exit.	
Client 192.168.1.108:49156 accepted at 12:03:28 12:03:28 received [sending tcp client message 3]	-

◆¥PC-SoftWare に Echotool. exe プログラムを保存している。



4.7.18 ¥Applications¥LwIP¥LwIP_UDP_Echo_Client¥MDK-ARM

プログラムは、1wIP ネットワークプロトコルを呼び出して UDP 方式の echo クライアントプログラムを実現する。

◆プログラムは特定な IP アドレスを使用している。pc と開発ボード通信のサブネットマスク、ゲートウェイと一致する。次の図の DEST_IP_ADDR はテスト用 PC の IP アドレスと一致しなければならない。 (今回のテスト用の IP アドレスが 192.168.1.103 である)、main.hに IP アドレスとゲートウェイを変更する。

```
💼 main.h 🔝 main.c 💼 ethernetif.c
43
   #define DEST IP ADDR0
                             192
44
45 #define DEST IP ADDR1
                             168
   #define DEST IP ADDR2
46
                             1
   #define DEST IP ADDR3
                             103
47
48
49
   #define DEST PORT
                             7
50
   /*Static IP ADDRESS: IP ADDR0.IP
51
52
   #define IP ADDR0
                       192
   #define IP ADDR1
53
                        168
   #define IP ADDR2
54
                        1
   #define IP ADDR3
55
                        108
56
57
   /*NETMASK*/
                             255
58
  #define NETMASK ADDR0
                             255
59
   #define NETMASK ADDR1
                             255
60
   #define NETMASK ADDR2
   #define NETMASK ADDR3
61
                             0
62
63
   /*Gateway Address*/
64 #define GW ADDR0
                        19
65 #define GW ADDR1
```

◆プログラムを開発ボードにダウンロードし、リセットを行う。コマンドラインに echotool /p udp /s を実行する(¥PC-SoftWare に chotool. exe プログラムを保存している)。ポート7を介してサーバーと クライアント接続と提示される。S2 キーを押すと、echotool に受信したデータ sending udp client message n を表示する。





🔤 C:\Windows\system32\cmd.exe - echotool.exe /p udp /s
E:\Mywork\ARM\STM32F4x9\STM32F4x9IG-DK-A-CD\PC-Software>echotool.exe /p udp /s
Waiting for UDP connection on port 7. Press any key to exit
12:09:10 from 192.168.1.108:49153 received [sending udp client message 0]
12:09:11 from 192.168.1.108:49153 received [sending udp client message 1]
12:09:11 from 192.168.1.108:49153 received [sending udp client message 1]
12:09:11 from 192.168.1.108:49153 received [sending udp client message 3]
12:09:11 from 192.168.1.108:49153 received [sending udp client message 4]
12:09:11 from 192.168.1.108:49153 received [sending udp client message 5]
12:09:12 from 192.168.1.108:49153 received [sending udp client message 6]
12:09:12 from 192.168.1.108:49153 received [sending udp client message 7]
12:09:12 from 192.168.1.108:49153 received [sending udp client message 8]
12:09:12 from 192.168.1.108:49153 received [sending udp client message 9]
12:09:12 from 192.168.1.108:49153 received [sending udp client message 10]
12:09:13 from 192.168.1.108:49153 received [sending udp client message 11]
12:09:13 from 192.168.1.108:49153 received [sending udp client message 12]
12:09:13 from 192.168.1.108:49153 received [sending udp client message 13]
12:09:13 from 192.168.1.108:49153 received [sending udp client message 14]
12:09:13 from 192.168.1.108:49153 received [sending udp client message 15]
12:09:13 from 192.168.1.108:49153 received [sending udp client message 16]
12:09:14 from 192.168.1.108:49153 received [sending udp client message 17]
12:09:15 from 192.168.1.108:49153 received [sending udp client message 18] 📃 📃

4.7.19 ¥Applications¥LwIP¥LwIP_UDPTCP_Echo_Server_Netconn_RTOS¥MDK-ARM

プログラムは 1wIP ネットワークプロトコル、FreeRTOS 操作システムを呼び出して、UDP 及び TCP 方式の echo サーバーを実現する。netconn 方式通信を使用している。

◆プログラムは特定な IP アドレスを使用している。pc と開発ボード通信のサブネットマスク、ゲートウェイと一致する。main.hに IP アドレスとゲートウェイを変更する。



◆開発ボードにプログラムをダウンロードしてからリセットを行う。¥PC-Software¥TCPUDPDbg ディレ クトリの TCPUDPDbg. exe を実行し、「CreateConnn」をクリックして開発ボードに接続する。



✗ TCP&UDP-Debug ○ Operate(Q) View(Y) Windows(W) Help(H) Language ○ CreateConnn CreateServer | StartServer Content Server CreateServer | StartServer Content Server CreateServer | StartServer Content Server Mode ○ Client Mode ○ Server Mode

◆IP テキストボックスに開発ボードの IP アドレス 192.168.1.108 を入力する。ポートが 7、「Create」 をクリックして確認する。

Create Connection		
Type: TCP	•	
DestIP: 192.168.1.108	Port: 7	
LocalPort C Auto	⊙ Specia	
🗖 AutoConn:	Eve 0	5
Send When Conn:	Eve	ms
Create	Cancel	
₩ TCP&UDP??T目 - [192.16	8.1.108:71	
Operate(Q) View(Y) W CreateConnn S Create Properties Client Mode 192.168.1.108;7 Server Mode	indows(₩) Help(H) Lang Server 30 Startserver 32 H × 22 192,168.1.108 UestIP: 192,168.1,108 DestPort: 7 VocalPort 7 Type TCP ▼ AtuoConn Eve 0 ms Connect Count Send 0 Recv 0 Clear	uage x Connect S DisconnAll DeleteConn DeleteConn DeleteConn Image 3:7 4 b x Send AtuoSend Eve 100 ms Send Stop Send AtuoSend Eve 100 ms Send Stop sbcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwxyz1234567890abcdefghijklmnopqrstuwx Rec StopShow Clear Save Option ShowHex Save(In Time)



◆「Connect」をクリックし、開発ボードに接続する。送信エリアにデータを入力し、「Send」を押 して開発ボードにデータを送信する。ボードは受信したデータを戻り、受信エリアに表示された。

◆同じ方法で、UDP データ転送方式を作成する。

4.7.20 ¥Applications¥LwIP¥LwIP_TFTP_Server¥MDK-ARM

TFTP の関連アプリプログラムである。TFTP ソフトウェア経由でファイルを SD カードにアップロードする、又は SD カードからファイルをダウンロードする。

◆プログラムは、DHCP が自動的に IP アドレスを取得するとデフォルトする。開発ボードに電源を入 れると、LCD 画面には割り当てられた IP アドレスが表示される。次の図に、ボードの IP アドレスが 192.168.1.10、main.hには IP アドレス、サブネットマスク、ゲートウェイ、および DHCP サービススイ ッチを変更できる。





◆テスト前に、SD カードを実装する。プログラムが開発ボードにダウンロードされてからリセットを 行う。コマンドラインウィンドウ(CMD. exe)を開いて、新規ファイルを作成する。コマンドラインに tftp -i 192.168.1.104 put test.txtを実行する。例えば、次の図には、Dディスクの test.txtを SD カードにアップロードされた。(win7 操作システムには最新のオンラインアップグレードパッチがなけ れば、TFTP ソフトウェアが実行されない)

C:\Windows	\system32\cm	d.exe		_ C X	
D:∖>dir Volume in Volume Ser	drive D i ial Numbe	s Local Disk er is 68FE-A158		*	
Directory	of D:\			=	
2013/06/17 2011/10/27 2013/03/20 2013/11/15 2013/10/17 2014/03/07 2014/02/10 2014/03/31 2014/02/19 2013/03/15	16:54 22:52 22:34 17:17 17:32 10:38 11:45 10:05 15:18 15:43 12:14 2 Fil 9 Dir	<pre><dir> <dir> <dir> <dir> 1,585,888,930 <dir> <dir <dir="" <dir<="" th=""><th>Cadence LPC2478_uClinux LPC2478_uClinux.rar microchip_solutions_u2013-02-15 NXP Program Files Program Files (x86) SiLabs Test.txt uCos Users 850 bytes 192 butes free</th><th></th><th></th></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></dir></pre>	Cadence LPC2478_uClinux LPC2478_uClinux.rar microchip_solutions_u2013-02-15 NXP Program Files Program Files (x86) SiLabs Test.txt uCos Users 850 bytes 192 butes free		
D:\>tftp -i Transfer su D:\> <u></u>	i 192.168. uccessful:	1.104 put test. 5873920 bytes :	txt in 74 second(s), 79377 bytes/s		

◆D ドライブの test. txt ファイルを削除する。コマンドラインに tftp -i 192. 168. 1. 104 get test. txt を実行し、SD カードに保存されている test. txt が D ドライブのルートディレクトリにダウンロードする。

C:\Windows\system32\cmd.exe	
Transfer successful: 5873920 butes in 74 second(s) 79377 butes/s	
D:\>dir	
Volume in drive D is Local Disk	
Volume Serial Wumber is Bore-H158	
Directory of D:\	
	-
2013/06/17 16:54 <dir> Cadence</dir>	
2011/10/27 22:52 <dir> LPC2478_uClinux</dir>	
2011/10/29 22:34 1,585,888,930 LPC2478_uClinux.rar	
2013/03/20 17:17 (DIR) microchip_solutions_v2013-02-15	
2013/11/15 17:32 (DIR) NXP	
2013/10/11 10:38 (DIR/ Program Files	
2014/03/01 11:43 (DIR) Program Files (x00)	
2014/02/19 15:43 (DIR) uCos	
2013/03/15 12:14 (DIR) Users	
1 File(s) 1,585,888,930 bytes	
9 Dir(s) 60,124,413,952 bytes free	
D:\>tftp -i 192.168.1.104 get test.txt	
Transfer successful: 5872384 bytes in 55 second(s), 106770 bytes/s	
D:\>_	~

注意:本テストは Canon の 16M SD カード、Kinston の 1G の SD カードで実行したが、全ての SD カードをさ ボートする保証はできない。

4.7.21 ¥Applications¥LwIP¥LwIP_IAP¥MDK-ARM

プログラムは IAP 機能を実現する。ネットワーク伝送モード(HTTP または TFTP)を介して、開発ボード



のアプリケーションを更新する。IAP プログラムが 0x800000-0x8010000flash 領域に保存され、アプリケー ションが 0x8010000 以降領域に保存される。プログラムが実行された後、すぐに S2 キーのステータスを確 認する。 S2 キーが押された場合に、IAP プログラムが実行中; S2 キーが押されていない場合、アプリケー ションが実行中。

◆IAP 機能のアプリケーションと一般アプリケーションの違いはプログラムの実行アドレスである ¥Applications¥ LwIP¥LwIP_HTTP_Server_Raw_0x08010000¥MDK-ARM ディレクトリのプロジェクトファイ ルにより生成されたコードは IAP プログラムを介してプロセッサにダウンロードされ、正しく実行する。 このプロジェクトファイルと一般的なプロジェクトファイルは、次の3つの場所が設定される必要があ る。

第一、プログラムの起動アドレスを 0x08010000 に設定し、残りのコードサイズを 0x1F0000 (0x200000-0x10000)に変更する。

Options fo	or Target	'STM324x9I_E	EVAL'	TA I			-	
Device Tar	get Outp	out Listing Us	ser C/C++Asn	n Linker				
STMicroele	ctronics S	TM32F429NI						
			<u>X</u> tal (MHz): 8.0					
Operating s	system:	None	-	•		\frown		
System-Vie	wer File (.	Sfr):						
C:\Keil\AR	M\SFD\S	T\STM32F4xx\	STM32F439x.SFR				*	
🗌 Use Cu	istom SVE) File				V		
Read/On	ly Memor	y Areas						
default	off-chip	Start	Size	Startup				
	ROM1:							
	ROM2:							
	ROM3:							
	on-chip							
~	IROM1:	0x8010000	0x1F0000	۲	V			
	IROM2:			0				

第二、bin フォーマットファイルを生成する。user 欄の Run User Programs After Build/Rebuild に Run#1 を選択し、かつテキストボックスに fromelf --bin .¥STM324x9I_EVAL¥STM324x9I_EVAL.axf -o .¥STM324x9I_EVAL¥STM324x9I_EVAL.bin に記入する。



Options for Targ	get 'STM324x9I_EVAL'	
Device Target 0	Dutput Listing User C/C++ Asm Linker Debug Utilities	
Run User Progra	ams Before Compilation of a C/C++ File	
🔲 Run #1:		
	Stop Build/Rebuild #1 on Exit Code: Not Specified	
🔲 Run #2:		
	Stop Build/Rebuild #2 on Exit Code: Not Specified	
Run User Progra	ams Before Build/Rebuild	
☐ Run #1:		
☐ Run #2:		
- Run User Progra	ams After Build/Rebuild	
⊡ Run #1:	fromelfbin .\STM324x9I_EVAL\STM324x9I_EVAL.axf -o .\STM324x9I_EVAL\STM324x9I_EV	
Run #2:		

第三、system_stm32f4xx.cを開く。VECT_TAB_OFFSETのマクロ定義数値を 0x10000 に変更する。

	main.c	🛓 system_s	tm32f4xx.c	
142	/* #def:	ine VECT	TAB SRAM	*/
143	#define	VECT TAE	3 OFFSET	0x10000

◆テスト用アプリプログラムにより生成された bin ファイルが¥Code¥STM32Cube_FW _F4_V1.1.0¥Projects¥STM324x9I_EVAL¥Applications¥LwIP¥LwIP_HTTP_Server_Raw_0x08010000¥MDK-AR M¥STM324x9I_EVAL ディレクトリに保存された。

◆まず、シミュレータにより IAP プログラムを開発ボードにダウンロードする。次に、IAP プログラ ムがネットワークによりアプリプログラムを開発ボードにダウンロードする。IAP プログラムは、DHCP サービスを開き、自動的にルータで IP アドレスを取得することをデフォルトする。main.h は、ボード の IP アドレス、サブネットマスク、ゲートウェイ、ネットワーク送信モードおよび DHCP サービスのオ ン/オフを開発することができる。

📩 ether	netif.c	main.c	🔜 <u>main.h</u>	📩 stm3	24x9i_eval.c	📩 startup	_stm32f429xx.s	
65	- *,	/						
66	#def	ine	USER_F	LASH_	FIRST	PAGE	ADDRES	5 0x08010000
67	#def	lne	USER_F	LASH	LAST	PAGE_	ADDRESS	0x081E0000
68	#def	ne	USER_F	LASH	_END_F	DDRES	S	0x081FFFFF
69								
70	/* Us	serI	D and	Passv	word d	lefini	tion ***	****
71	#def	ne	USERIC		"us	ser"		
72	#def:	ine	PASSWC	RD	"st	.m32"		
73	#def	ine	LOGIN_	SIZE	(15	5+ siz	eof (USEI	RID) + sizeo:
74								
75	/* St	tati	c IP A	ddres	ss def	initi	on ****	****
76	#def:	lne	IP_ADD	R0	192			
77	#def	lne	IP_ADD	R1	168			
78	#def:	lne	IP_ADD	R2	1			
79	#def:	lne	IP_ADD	R3	108			
80								
81	/* NH	ETMA	SK def	initi	ion **	*****	******	* * * * * * * * * * * * * * *
82	#def:	lne i	NETMAS	K_ADI	DR0	255		
83	#def	lne i	NETMAS	K_ADI	DR1	255		
84	#def:	lne i	NETMAS	K_ADI	DR2	255		
85	#def:	lne i	NETMAS	K_ADI	DR3	0		



◆IAP プログラムを開発ボードにダウンロードされ、S2 キーを押しながら、リセットを行う。この時、 プロセッサは、0x8000000-0x8010000Flash 域の IAP プログラムを実行している。次の図には、LCD 画面 に割り当てられた IP アドレス 192. 168. 1. 104 が表示されている。









4.7.22 ¥Applications¥STemWin¥STemWin_HelloWorld¥MDK-ARM

STemWin の Hello world テストプログラムである。プログラムが実行された後、画面の中央に Hello world が現れ、S2 LED が点滅している。

4.7.23 ¥Applications¥STemWin¥STemWin_SampleDemo¥MDK-ARM

4.2.1 と同じように、STemWin 一部分の機能を実証する。それらは、タッチスクリーンのキャリブレーション、ラジアルメニュー、透明ダイアログ、アイコンビュー、ツリービュー、リストビュー、仮想スクリーン、ズーム回転写真、グラフィックスの描画、グラフィック、ビットマップ、カーソル、カラーバーである。

◆プログラムが実行された後に、LCD 画面にタッチすると、タッチスクリーンを補正することと提示される。



◆タッチスクリーン右下隅ドットをタッチすることと提示される。





4.7.24 ¥Applications¥USB_Device¥HID_Standalone¥MDK-ARM

USB HID クラスのアプリケーションである。開発ボードを PC に接続する時に、デバイスマネージャウィン ドウには入出力デバイスが接続されたと表示する。PC が自動的にドライバをインストールした後に、開発ボ ードの S1 を押して、カーフルが上方向へ移動する;S2 を押して、カーソルが下方向へ移動する;S3 を押し て、カーノルが左方向へ移動する;S4 を押して、カーソルが右方向へ移動する。テスト用の USB インタフェ ースは J6 である。

注意:開発ボードはフルスピードUSB インタフェースを提供するので、フルスピードUSB プロジェクト (STM324x9I-EVAL_USBD-FS)を使用してください。

🔗 🕮 🎬 🥪 🔜 🙀	STM324x9I-EVAL_U
Project	STM324x9I-EVAL_USBH
	STM324x9I-EVAL_USBH
⊡🖻 STM324x9I-EVAL_U	STM324x9I-EVAL_USBH
	DK-ARM



4.7.25 ¥Applications¥USB_Device¥CustomHID_Standalone¥MDK-ARM

USB のカスタム HID のテストプログラムである。プログラムは PC ソフトウェアを介して LED の点滅を制 御したり、S2 キーのステータスを取得する。

◆予めに、¥PC-Software ディレクトリの HHIDDemo_V1.0.2_Setup. exe をインストールする。インスト ール済、PC には UsbHidDemonstrator. exe プログラムがある。J6 に USB ケーブルを接続し、プログラム を開発ボードにダウンロードする。次に、PC には新しいデバイスが接続されたと提示され、自動的に対 応するドライバを検索する(HIDDemo_V1.0.2_Setup. exe がインストールされたが前提とする)。

① Driver Software Installation	PC applet and o	X		•
Installing device driver software				
Custome HID in FS Mode	OSearching Windows Update		-	
Obtaining device driver software from Window Skip obtaining driver software from Windows	ws Update might take a while. <u>Update</u>			
		Close		
Driver Software Installation		×		
USB Input Device installed				
USB Input Device	✔ Ready to use			
		Close		

◆UsbHidDemonstrator.exeを実行する。USB HID Target に HID-compliant device を選び、左側の Device capabilities をクリックし、USB デバイスの基本的な記述子が見られる。

USB HID Demonstrator (v1.0.2)		×
Device capabilities	USB HID Target HID-compliant device	
Input/Output transfer	Vendor identifier 0x0483 Product identifier 0x5750 Version number 0x0200	
Graphic view Graphic customization	Usage Page: FF Input Report Byte Length: 2 Output Report Byte Length: 2 Feature Report Byte Length: 2 Number of Link Collection Nodes: 1 Number of Input Button Caps: 2 Number of Input Data Indices: 3 Number of Output Button Caps: 0 Number of Output Button Caps: 4 Number of Output Data Indices: 4 Number of Feature Button Caps: 2 Number of Feature Value Caps: 5 Number of Feature Data Indices: 7	

◆Input/Output transfer をクリックし、S2 キーを押して、Input Report byte lengt ウィンドウに 受信した 05 01 文字が見られる。S2 キーをリリースすると、05 00 文字を受信した。Output Report byte length ウィンドウに 01 01 を入力し、右下隅の Write を押して、D2 LED が点灯している。。Output Report byte length ウィンドウに 01 00 を入力し、右下隅の Write を押して、D2 LED が消灯している。



USB HID Demonstrator (v1.0.2)	
Device capabilities	USB HID Target HID-compliant device
Input/Output transfer	Input Report byte length (h) 0002 Bytes
Graphic view	
Graphic customization	• • • •
Output mode • Use SET_REPORT C Use SET_FEATURE	Output Report byte length (h) 0002 Bytes 0000 01 01 ··

◆左側の Graphic view をクリックし、S2 キーを押すと、Button1 所が点灯している;S2 キーをリリースすると、Button1 所が消灯している。Led 1 ReportID を選ぶ場合に、D2 LED が点灯している;選ばないい場合に、D2 LED が消灯している。

USB HID Demonstrator (v1.0.2)	per s a	×
Device capabilities	USB HID Target HID-compliant device	
Input/Output transfer	Buttons	Variable Inputs Var Input 1 Report ID (h) 07
Graphic view	Button 2 Report ID (h)	00h
Graphic customization		
- Output mode		Image: Led 1 Report ID (h) 01
Use SET_REPORT		Led 2 Report ID (h) 102
O Use SET_FEATURE		Led 3 Report ID (h)

◆Graphic customization をクリックし、Graphic viewのButtonsとLeds 数量をカスタマイズできる。次の設定に従って、Graphic viewの表示が変化する。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする

USB HID Demonstrator (v1.0.2)	Antimate P Q -	
	USB HID Target HID-compliant device	
	Buttons	Leds
Input/Output transfer	✓ Button 1: Report ID (h)	✓ Led 1: Report ID (h)
Graphic view	Button 2: Report ID (h)	
Graphic customization	Button 4: Report ID (h)	Led 4: Report ID (h) 04
	Button 5: Report ID (h)	Variable Inputs
Output mode	Button 5: Report ID (h) 00 Button 7: Report ID (h) 00	
Use SET_REPORT O Use SET_FEATURE	Variable Output	□ Variable Input 1: Report ID (h) 0'
	Variable Output: Benort ID (h) 00	
USB HID Demonstrator (v1.0.2)		
Device capabilities	USB HID Target HID-compliant device	
Input/Output transfer	Buttons	ble inputs
Graphic view	Button 1 Report ID (h)	
Graphic customization		
- Output mode		ed 1 Report ID (h) 01
Use SET_REPORT		
O Use SET_FEATORE		
注意:開発ホートかフル (STM324x91-EVAL USBD-F	スピード USB インタフェースを打 S)を使用してください。	是供するので、フルスピード USB フロシェクト
Project STN	1324x9I-EVAL_USBF	
STM STM324x9I-EVAL_USTM	1324x9I-EVAL_USBH 1324x9I-EVAL_USBH	

Application/MDK-ARM

4.7.26 ¥Applications¥USB_Device¥CDC_Standalone¥MDK-ARM

CDC クラスが PSTN サブプロトコルに従うテストプログラムである。STM32F4xxUSB On-The-Go ホストとデバイス側のライブラリを使用している。開発ボードが USB 転換シリアルポートのデバイスとして使用される。



◆予めに、¥PC-Software ディレクトリの VCP_V1.3.1_Setup. exe(PC のデバイスドライバプログラム) をインストールする。プログラムを開発ボードにダウンロードした後に、USBケーブルをJ6に接続する。 PC には新しいデバイスが接続されたと提示され、自動的にドライバプログラムをインストールする。

Driver Software Installation	×
STMicroelectronics Virtual COM Port (COM18) installed	
STMicroelectronics Virtual COM Port (COM18) 🖌 Ready to use	
	Close

◆インストール済、デバイスマネージャを開き、新しい仮想シリアルデバイスが見られる。例えば、 次の図には新しい仮想シリアルポートのシリアル番号が COM18 に割り当てられる。



◆開発ボードにケーブルで PC のシリアルポート (今回は USB Serial Port (COM6)を使用している)を 接続する。ハイパーターミナルを経由して COM6 と COM18 をオープンし、かつボーレートが 115200、フ ロー制御なし。COM6 と COM18 が互いにデータを送信することができる。



注意:開発ボードがフルスピードUSB インタフェースを提供するので、フルスピードUSB プロジェクト (STM324x9I-EVAL_USBD-FS)を使用してください。





🔗 🍱 🎬 🥪 🔜 💥	STM324x9I-EVAL_U
Project	STM324x9I-EVAL_USBF STM324x9I-EVAL_USBF
⊡ 🔁 STM324x9I-EVAL_U	STM324x9I-EVAL_USBH
Application/M	OK-ARM

4.7.27 ¥Applications¥USB_Device¥MSC_Standalone¥MDK-ARM

USB マスストレージデバイスのプログラムである。 SD カードに接続すると、USB リーダー機能が実現できる。STM32F4xxUSB On-The-Go ホストとデバイス側ライブラリを使用している。

◆予めに SD カードを実装する。USB ケーブルを J6 インタフェースに接続すると、リムーバブルハー ドディスクが現れる。これをクリックして、SD カードの内容が見られる。



注意:

◆テストに、8GのSDカードが使用されるが、PCはフォーマットする必要が求められる。フォーマットした後に3.27Gスペースしかない。具体的な理由はまだ調べている(おそらく4Gカード、大容量の再書き込みを行う)。4.4.4SD_FileプログラムでSDカードを再び8Gにフォーマットできる。
 ◆開発ボードがフルスピードUSBインタフェースを提供するので、フルスピードUSBプロジェクト

(STM324x91-EVAL_USBD-FS)を使用してください。

🧇 🎞 🕮 🥪 🔜 🚺	STM324x9I-EVAL_U
Project	STM324x9I-EVAL_USB
	STM324x9I-EVAL_USBH
🖃 📲 STM324x9I-EVAL_U	STM324x9I-EVAL_USBH
Application/MDK-ARM	

4.7.28.¥Applications¥USB_Device¥DFU_Standalone¥MDK-ARM

プログラムはDFU クラス、IAP 機能を実現する。USB 経由で開発ボードのアプリプログラムを更新する。 IAP プログラムは 0x8000000-0x0800C000 の flash 領域に保存され、アプリケーションが 0x0800C000 以降の flash 領域に保存される。プログラムが実行された後に、すぐに S2 キーのステータスを判断する。S2 キー を押すと、IAP が実行中; S2 キーが押さないと、アプリケーションが実行する。

◆IAP 機能のアプリケーションと一般アプリケーションの違いは、プログラムの実行アドレスである ¥Applications¥USB_Device¥HID_Standalone_0x0800C000¥MDK-ARM ディレクトリのプロジェクトファイ ルにより生成されたコードは IAP プログラムを介してプロセッサにダウンロードされ、正しく実行する。 このプロジェクトファイルと一般的なプロジェクトファイルは、次の3つの場所がリセットされる必要 がある。

第一、プログラムの起動アドレスを 0x0800C000 に設定し、残りのコードサイズを 0x1F4000 (0x200000-0xC000)に変更する。



V Options for Target 'STM324x9I-EVAL_USBD-FS'
Device Target Output Listing User C/C++ Asm Linker
STMicroelectronics STM32F429NI
∐tal (MHz): 8.0
Operating system: None
System-Viewer File (.Sfr):
C:\Keil\ARM\SFD\ST\STM32F4xx\STM32F439x.SFR
Use Custom SVD File
Read/Only Memory Areas default off-chip Start Size Startup
on-chip
IROM1: 0x0800C000 0x1F4000 €
第二、hex フォーマットのファイルを生成する。Output 欄の Create HEX File を選ぶ。
Contions for Target 'STM324x9I-EV/AL
Device larget Output Listing User C
Select Folder for Objects
Create Executable: STM324x9I-EV.
✓ Debug Information
Create HEX File
✓ Browse Information
C Create Library, STM324x9PEVAL_L
第三 avetem etm29f(ave を聞き VECT TAR OFFSET のマクロ定差粉を 0xC000 に設定する
新二、System_stimozran, chamber () に 定義数を 0xc000 (C設定) う。
system_stm32f4xx.c is stm32f4xx_it.c is startup_stm32f429xx.s is KBD.h
98
99 - K < Uncomment the following lin
100 Internal SRAM. */
101 /* #define VECT_TAB_SRAM */
102 #define VECT_TAB_OFFSET 0xC000

◆テスト用アプリプログラムにより生成された bin ファイルが¥Code¥STM32Cube_FW_F4_V1.1.0 ¥Projects¥STM324x9I_EVAL¥Applications¥USB_Device¥HID_Standalone_0x0800C000¥MDK-ARM¥STM324x9 I-EVAL_USBD-FS ディレクトリに保存された。

◆まず、シミュレータにより IAP プログラムを開発ボードにダウンロードする。次に、IAP プログラ



ムが DFU ソフトウェアによりアプリプログラムを開発ボードにダウンロードする。PC の DFU File Manager. exe プログラムで HEX フォーマットのファイルを DFU フォーマットに変換される。次に、変換 されたファイルが DfuSeDemo. exe で開発ボードにダウンロードする。詳細は 3.2 の部分にご参照ください。 ◆¥PC-Software ディレクトリの DfuSe_Demo_V3.0.3_Setup. exe をインストールする。IAP プログラム を開発ボードにダウンロードされ、S2 キーを押しながら、リセットを行う。この時、プロセッサは、 0x8000000-0x800C000F1ash 域の IAP プログラムを実行している。USB ケーブルを J6 インタフェースに 接続し、PC が新しいデバイスが接続されたと提示して、かつ自動的にドライバプログラムをインストー ルする (DfuSe_Demo_V3.0.3_Setup. exe がインストールされたが前提となる)

Driver Software Installation	
STM Device in DFU Mode installed	
STM Device in DFU Mode	
	Close
◆DfuSe Demonstration.exe を実行する。	
C DfuSe Demo (v3.0.3)	
Available DFU Devices STM Device in DFU Mode Image: Style of the style of t	DFU Mode: Vendor ID: 0483 Procuct ID: DF11 Version: 0200 k
Upload Action File: Choose Upload Transferred data size 0 KB(0 Bytes) of 0 KB(0 Bytes) Operation duration 00:00:00 Upgrade of Verify Action File: Vendor ID: Version: Version: Optimize Upgrade duration (Remov Chgose Upgrade	e: e some FFs)
	Quit

◆Select Target(s)に Intermal Flash を選び、Upgrade or Verify Action に Choose キーをクリックして ダウンロードプログラムを選ぶ。



C DfuSe Demo (v3.0.3	.3)	
Available DFU Device STM Device in DFU Supports Upload Supports Downlo Can Detach Enter DFU mode/F Actions Select Target(s):	Jes Application Mode: DFU Mode: Id Manifestation tolerant Vendor ID: Vendor ID: load Accelerated Upload (ST) Procuct ID: Procuct ID: HID detach Leave DFU mode Version: Version: Targe Name Available Sectors (Double Click) 00 Internal Flash 24 sectors	483 F11 200
Upload Action File: Choose Transferred data siz 0 KB(0 Bytes) of 0 K Operation duration - 00 Abort	Upgrade or Verify Action File: Vendor ID: Targets in file: Version: Version: Version: Version: Optimize Upgrade duration (Remove some FFs) Choose Upgrade	rify
Copen Look in:	STM324x9I-EVAL_USBD-FS	
Desktop Libraries Computer Network Fil	Image: STM324x9I-EVAL_USBD-FS Files of type: Dfu Files (*.dfu) Image: Open as read-only	Qpen Cancel

◆Upgrade をクリックし、プログラムを Flash にダウンロードする。その後、Leave DFU mode をクリック し、ダウンロードされたプログラムを実行する。





注意:内蔵 ROM には DFU プログラムがある。SW1 の短絡キャップをシルク1 側に接続する。そうすれば、USB ケーブルを接続したら、プロセッサが DFU プログラムを実行する。上述は FLASH に書き込まれた DFU プログラムを記述した。

4.7.29 ¥Applications¥USB_Host¥HID_Standalone¥MDK-ARM

STM32Cube 開発ソフトウェアパッケージの USB ホストライブラリを使用して、標準の USB HID クラスのホ スト・アプリケーションを実現した。それぞれ USB キーボードと USB マウスを接続できる。

◆予めに USB キーボードと USB マウスを準備する。プログラムが実行された後、LCD 画面に以下のように表示する。

USB OTG FS HID Host	-	
USB Host library started. Starting HID Demo		
Use Key [\$4 Left] to scroll up		
Jse Key [S1/S2 Up/Down] to scroll HID menu,S3 to sellect I - Start HID 2 - Re-Enumerate		

◆J8にUSBマウスを接続する。この時点で画面にUSBマウスが検出され、かつ基本的な情報が表示される。S4キーを押すと、画面の内容が見られる。

USB OTG 75 HID Host
Address (#1) assigned. Manufacturer : N/A Product : Lenovo Precision USB Mouse Serial Number : N/A Enumeration done. This device has only 1 configuration. Default configuration set. Switching to Interface (#0)
Class 3h SubClass 1h Protocol 2h Mouse device found! HID class started.
Use Key [S4 Left] to scroll up Use Key [S1/S2 Up/Down] to scroll HID menu,S3 to sellect 1 - Start HID 2 - Re-Enumerate





1 - Start ĤID 2 - Re-Enumerate

◆S1とS2キーを押すと、画面の現行メニューを上下移動できる(選択された項目が紫);S3キーを 押すと、現行選択された動作を実行する;S1キーを押し、現行メニューに1-Start HIDを選択させ;S3 キーを押して、選択された動作を行う。この時点では、画面は次のように表示する。



◆マウスを移動すると、白い長方形の中央にカーソルも一緒に移動する。 マウスの三つキーをクリ ックすると、白い長方形ボックスの下に対応する3つの小さなボックスが青に変わる。



◆S1 または S2 キーを押すと、現行メニューが 2-Return を選択させる。S3 キーを押すと、USB マウス



低価格、高品質が不可能? 日昇テクノロジーなら可能にする

テストが終了する。USBマウスを抜いて、J8インタフェースにUSBキーボードを接続する。、この時点でUSBキーボードの基本的な情報が画面上に表示される。

USB OTG FS HID Host	
Address (#1) assigned. Manufacturer : N/A Product : N/A Serial Number : N/A Enumeration done. This device has only 1 configuration. Default configuration set. Switching to Interface (#0) Class : 3h SubClass : 1h Protocol : 1h KeyBoard device found! HID class started.	
Use Key [S4 Left] to scroll up Use Key [S1/S2 Up/Down] to scroll HID menu,S3 to sellect 1 - Start HID 2 - Re-Enumerate	
◆S1キーを押し、現行メニューが1-Start HIDを選択させ;S3キーを押し、選択された操作 この時点で画面上に次の内容が表示される。	を行う。
USB OTG FS HID Host Use Keyboard to tape characters: Use Key [54 Left] to scroll up Use Key [54 Left] to scroll up Use Key [51/52 Up/Down] to scroll HID menu,53 to sellect I - Start Keyboard (Clear	
2 - Return ◆USB キーボードをクリックし、画面には対応する文字が表示される。	
Use Keyboard to tage characters: 0123456789 9876543210	
Use Key [S4 Left] to scroll up Use Key [S1/S2 Up/Down] to scroll HID menu,S3 to sellect 1 - Start Keyboard / Clear 2 - Beturn	

◆注意:開発ボードがフルスピード USB インタフェースを提供するので、フルスピード USB プロジェ



クト(STM324x91-EVAL_USBD-FS)を使用してください。

🔗 🖺 🕮 🥪 🔜 🙀	STM324x9I-EVAL_U
Project	STM324x9I-EVAL_USB
	STM324x9I-EVAL_USBH STM324x9I-EVAL_USBH
Application/MDK-ARM	

4.7.30 ¥Applications¥USB_Host¥HID_RTOS¥MDK-ARM

FreeRTOS に基づいて、STM32Cube 開発ソフトウェアパッケージの USB ホストライブラリを使用して、標準の USB HID クラスのホスト・アプリケーションを実現した。それぞれ USB キーボードと USB マウスを接続できる。

◆機能とテスト方法は上例と同じである。上例は検索方法でキー値を取得するが、本例は外部割り込みで取得する。main.cファイルにはEXTILine0_Config、EXTILine2_Config、EXTILine3_Config、EXTILine13_15_Config 四つのキーに対応するピンの外部割り込み関数を追加した。

HAL_NVIC_SetPriority 関数を呼び出す時、割り込みレベルが 5 以上にしなければならない(他の高い優 先レベルが FreeRTOS に占有された。FreeRTOSConfig.hの

configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY にご参照ください)

◆四つキーの割り込みプログラムが stm32f4xx_it.c ファイルに保存され、全部 HAL_GPI0_EXTI_Callback コールバック関数を呼び出す。menu.c ファイルの HAL_GPI0_EXTI_Callback コ ールバック関数は生成された割り込みのキーを判断して、かつ事情により該当な動作を行う。

◆注意:開発ボードがフルスピード USB インタフェースを提供するので、フルスピード USB プロジェ クト (STM324x91-EVAL_USBD-FS)を使用してください。

📀 🍱 🗃 🥪 🔜 🙀	STM324x9I-EVAL_U	
Project	STM324x9I-EVAL_USBF	
E STM324x9I-EVAL_U	STM324x9I-EVAL_USBF	
n Application/MDK-ARM		

4.7.31 ¥Applications¥USB_Host¥MSC_Standalone¥MDK-ARM

STM32Cube 開発ソフトウェアパッケージの USB ホストライブラリを使用して、標準の USB HID クラスのホ スト・アプリケーションを実現した。USB メモリが接続できる。

◆予めて、USB メモリに新規 USBHost.txt ファイルを作成し、かつ文字を入力する。プログラムが実行された後、LCD 画面に以下のように表示される:





◆S1とS2キーを押すと、画面の現行メニューを上下移動できる(選択された項目が紫)、S3キーを 押すと、現行選択された動作を実行する;S1キーを押し、現行メニューに1-File Operations を選択さ せ、S3キーを押して、選択された動作を行う。USBメモリのUSBHost.txtファイルに以下のような内容 を書き込む。



USB OTG FS MSC Host	
Inquiry Vendor : Generic- Inquiry Product : Multi-Card Inquiry Version : 1.00 Mass Storage Device ready	
Mass Storage Device Capacity : 5505 MB Block number : 15278079 Block Size : 512 INFO : FatFs Initialized INFO : 'USBHost.txt' opened for write INFO : Text written on the 'USBHost txt' file	
Read Text : USB Host Library : Mass Storage Example INFO : FatFs data compare SUCCES Use Key [S4 Left] to scroll up	
Use Key [S1/S2 Up/Down] to scroll HID menu,S3 to sellect 1 - File Operations 2 - Explorer Disk 3 - Re-Enumerate	

◆現行メニューが 2-Explorer Disk を選択せさ、S3 キーを押すと、選択された操作を行う。USB メモリのファイル情報を画面に表示して、S2 キーを押して、残りのファイル情報を表示する。



4.7.32 ¥Applications¥USB_Host¥MSC_RTOS¥MDK-ARM

FreeRTOS 操作システムに STM32Cube 開発ソフトウェアパッケージの USB ホストライブラリを使用して、 標準の USB HID クラスのホスト・アプリケーションを実現した。USB メモリが接続できる。機能とテスト方法 が上例と同じである。

注意:開発ボードがフルスピードUSBインタフェースを提供するので、フルスピードUSBプロジェクト (STM324x9I-EVAL_USBD-FS)を使用してください。





🔗 🏝 🖀 🥪 🚉 💥	STM324x9I-EVAL_U
Project	STM324x9I-EVAL_USBH
-	STM324x9I-EVAL_USBH
□ 🖃 🔁 STM324x9I-EVAL_U	STM324x9I-EVAL_USBH
Application/M	OK-ARM

4.7.33 ¥Applications¥USB_Host¥AUDIO_Standalone¥MDK-ARM

STM32Cube 開発ソフトウェアパッケージの USB ホストライブラリを使用して、標準の USB AUDIO クラスの ホスト・アプリケーションを実現した。それぞれ USB サウンドカードを接続できる。

◆予めに、WAV フォーマットのオーディオファイルを SD カードにコピーし、開発ボードに SD カード を挿し込み、J8 プラグを USB サウンドカードに挿し込み、プログラムをダウンロードした後、サウンド カードの基本情報が LCD 画面に表示される:




にヘッドセットを挿入すると、音楽が聞こえる。



🔗 🍱 🎬 🧼 🔣 🙀	STM324x9I-EVAL_U 🔽
Project	STM324x9I-EVAL_USBH
	STM324x9I-EVAL_USBH
⊡ 📲 STM324x9I-EVAL_U	STM324x9I-EVAL_USBH

- Application/MDK-ARM
 - 2. 一部分の SD カードと USB サウンドカードをサポートしていない。
- 3. ¥WavFile ディレクトリには WAV フォーマットのオーディオファイルがあり、プログラムは WAV フォーマットのオーディオファイルのみが再生できる。

4.7.34 ¥Applications¥USB_Host¥DynamicSwitch_Standalone¥MDK-ARM

STM32Cube 開発ソフトウェアパッケージの USB ホストライブラリを使用して、標準の USB ホスト MSC、HID 及び AUDIO クラスのアプリケーションを実現した。それぞれ USB メモリ、USB マウス、USB キーボード、USB サウンドカードを接続できる。

● 予めに、USB メモリ、USB マウス、USB キーボード、USB サウンドカード、SD カードを準備する。そ して、WAV フォーマットのオーディオファイルを SD カードにコピーし、プログラムを開発ボードにダウ ンロードして実行された後、画面には USB などのデバイスを挿入することと提示する。





● J8 の USB ソケットに USB メモリを挿入し、画面には USB メモリの基本情報が表示される。S2 キーを押すと、画面に表示された内容をチェックする;S3 キーを押すと、現行メニューを上に移動する;S4 キーを押すと、現行メニューを下に移動する;S1 キーを押すと、現行メニューの操作を行う。



●S3 キーを押すと、現行メニューが 1-File Operations を選択させる;S2 キーを押すと、選択された 操作を行って、USB メモリに USBHost.txt ファイルを書き込む。



USB FS DynamicSwitch	Host
Inquiry Vendor : USB2.0	
Inquiry Product : CARD-READER	
Mass Storage Device ready	
Mass Storage Device capacity : 3363 MB	
Block number : 15278079	
Block Size : 512	
INFO : MSBHost tyth opened for write	
INFO : Text written on the 'HSBHost tyt' file	
Read Text :	
USB Host Library : Mass Storage Example	
INFU : FatFs data compare SUCCES	
Use [51] to seriect [52] to scroll up	
1 - File Operations	
2 - Explorer Disk	
3 - Re-Enumerate	

●現行メニューを 2-Explorer Disk に選択させ、S1 キーを押すと、選択された操作を行う;プログラムは USB メモリのフィル情報を画面に表示させ、S2 キーを押すと、残りのファイル情報を表示する。



● USB メモリを抜いて、J8のUSB ソケットにはUSB キーボードを挿し込む。LCD 画面にはUSB キーボードの基本情報が表示される。



●S3キーを押すと、現行メニューを1-Start HID に選択させ、SI キーを押すと、選択された操作を



行う。その時、画面には以下のような内容が現れる。USB キーボードのボタンを押すと、画面には対応 する文字が表示される。



● S4 キーを押すと、現行メニューを 12-Return に選択させ、SL キーを押すと、終了する。USB キーボードを抜き出し、J8 の USB ソケットに USB マウスを挿入し、画面には USB マウスの基本情報が表示される。

USB FS DynamicSwitch Host Address (#1) assigned. Manufacturer : N/A Product : Lenovo Precision USB Mouse Serial Number : N/A Enumeration done. This device has only 1 configuration. Default configuration set. Switching to Interface (#0) Class : 3h SubClass : 1h Protocol : 2h Mouse device found! HID class started. Use [S1] to sellect [S2] to scroll up Use [S3/S4 Up/Down] to scroll MSC menu 1 - Start HID 2 - Re-Enumerate

◆S3又はS4キーを押すと、画面の現行メニューを上下移動できる(選択された項目が紫);S3キ ーを押し、現行メニューに1-Start HIDを選択させ;S1キーを押して、選択された動作を行う。画面に は以下のような内容が表示する。









USB	FS DynamicSwitch	1 Host
USB Host library star Starting DynamicSwitc Plug your device To C USB Device Attached PID: eh VID: d8ch Address (#1) assigned Manufacturer : N/A Product : Generic USB Serial Number : N/A Enumeration done. This device has only Default configuration Use [S1] to sellect [Use [S3/S4 Up/Down] t 1 - Explore aud 2 - Start audio 3 - Re-Enumerat	ted. h Demo ontinue Audio Device 1 configuration. set. S2] to scroll up o scroll MSC menu io file Player e	
◆S3 キーを押す S1 キーを押すと、注 イルを表示する。	と、現行メニューが 1-Expl 選択された操作を行う。プ	lore Audio fileを選択る ログアムは SD カードの



◆S4 キーを押すと、現行メニューを 2-Start audio player に移行させる。S1 キーを押すと、現行メ ニューの操作を行う。プログラムは、SD カード上のオーディオファイルを再生し、USB サウンドカード にヘッドセットを挿入すると、音楽が聞こえる。

USR FS DynamicSwitch	Host
Playing file (1/4): JINGGUO.WAV	
Sample rate : 44100 Hz Channels number : 2 File Size : 35 MB [03:29] [02:39]	
[S3]: Volume + [S4]: Volume - [S2]: Exit [S1]: Next [PLAY]	
Use [User Key] To Stop and return from player Use [S1] to sellect [S2] to scroll up Use [S3/S4 Up/Down] to scroll MSC menu 1 - Explore audio file 2 - Start audio Player 3 - Re-Enumerate	



◆S3キーを押すと、音量を上げる;S4キーを押すと、音量を下げる;S1キーを押すと、次の音楽 を再生する;S2キーを押すと、終了する。

注意:

1. 注意:開発ボードがフルスピード USB インタフェースを提供するので、フルスピード USB プロジェクト (STM324x9I-EVAL_USBD-FS)を使用してください。

🔗 🕮 🎬 🥪 🔜 🙀	STM324x9I-EVAL_U
Project	STM324x9I-EVAL_USBH
	STM324x9I-EVAL USBH
E 🔁 STM324x9I-EVAL_U	STM324x9I-EVAL_USBH

Application/MDK-ARM

2. 一部分の SD カード、USB メモリ、USB マウス、USB サウンドカードをサポートしていない。 3. ¥WavFile ディレクトリには WAV フォーマットのオーディオファイルがあり、プログラムは WAV フォ ーマットのオーディオファイルのみが再生できる。

4.7.35 Applications¥USB_Host¥FWupgrade_Standalone¥MDK-ARM

STM32Cube 開発ソフトウェアパッケージの USB ホストライブラリを使用して、標準の USB ホスト MSC クラ スのアプリケーションを実現した。USB メモリ経由でアプリプログラムを更新する。IAP プログラムは 0x800000-0x0800C000 の flash 領域に保存され、アプリケーションが 0x0800C000 以降の flash 領域に保存 される。プログラムが実行された後に、すぐに S2 キーのステータスを判断する。S2 キーを押すと、IAP が 実行中; S2 キーが押さないと、アプリケーションが実行する。

◆IAP 機能のアプリケーションと一般アプリケーションの違いは、プログラムの実行アドレスである ¥Applications¥USB_Host¥MSC_Standalone_0x0800C000¥MDK-ARM ディレクトリのプロジェクトファイル により生成されたコードは IAP プログラムを介してプロセッサにダウンロードされ、正しく実行する。 このプロジェクトファイルと一般的なプロジェクトファイルは、次の3つの場所が設定される必要があ る。

第一、プログラムの起動アドレスを0x0800C000に設定し、残りのコードサイズを0x1F4000 (0x200000-0xC000)に変更する。







Options for Target 'STM324x9I-EVAL_USBH-FS'				
Device Ta	arget Outp	ut Listing Us	er C/C++Asm	Linker
STMicroelectronics STM32F429NI				
<u>X</u> tal (MHz): 8.0				
Operating	system:	None		-
System-Viewer File (.Sfr):				
C:\Keil\Al	RM\SFD\ST	\STM32F4xx\S	TM32F439x.SFR	
Use Custom SVD File				
Read/0	nly Memory	Areas		
default	off-chip	Start	Size	Startup
	ROM1:			0
	ROM2:			0
	ROM3:			0
	on-chip			
	IROM1:	0x800C000	0x1F4000	۲
	IROM2:			0

第二、BIN フォーマットのファイルを生成する。User 欄の Run User Programs After Build/Rebuild 項目 の Run#1 を選択し、かつテキストボックスに fromelf → bin .¥STM324x9I-EVAL_USBH-FS¥STM324x9I-EVAL_USBH-FS.axf -o.¥STM324x9I-EVAL_USBH-FS¥STM324x9I-EVAL_USBH-FS.bin を記入し:

Coptions for Target 'STM324x9I-EVAL_USBH-FS'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Run User Programs Before Compilation of a C/C++ File
Run#1: DOS16
Stop Build/Rebuild #1 on Exit Code: Not Specified
DOS16
Stop Build/Rebuild #2 on Exit Code Not Specified
Run User Programs Before Build/Rebuild
🗌 🗌 Run#1:
Run #2:
Run User Programs After Build/Rebuild
Run #1: Itomelfbin \STM324x9I-EVAL_USBH-FS\STM324x9I-EVAL_USBH-FS.axf-o \STM324x9I-f DOS16
□ Run #2 □ DOS16
Eeep When Complete Start Debugging
OK Cancel Defaults Help

第三、system_stm32f4xx.cを開き、VECT_TAB_OFFSETマクロ定義数値を 0xC000 に変更する。

system_stm32f4xx.c is usbh_conf.c is stm32f4xx.ic is main.c is read 97 #endif /* DATA_IN_ExtSRAM && DATi 98 998/*!< Uncomment the following line 100 Internal SRAM. */ 101 /* #define VECT_TAB_SRAM */ 1028#define VECT_TAB_OFFSET_0xC000

◆テスト用アプリプログラムにより生成された bin ファイルが¥Code¥STM32Cube_FW



_F4_V1.1.0¥Projects¥STM324x9I_EVAL¥Applications¥USB_Host¥MSC_Standalone_0x0800C000¥MDK-ARM¥ STM324x9I-EVAL_USBH-FS ディレクトリに保存された。

◆まず、BIN ファイルを image. BIN と名前変更し、USB メモリにコピーする。次に、シミュレータに より IAP プログラムを開発ボードにダウンロードする。最後に、IAP プログラムが USB メモリのアプリ プログラムを開発ボードにダウンロードする。

◆予めに、image. BIN ファイルが付いている USB メモリを J8 インタフェースに接続し、IAP プログラ ムを開発ボードにダウンロードされ、S2 キーを押しながら、リセットを行う。この時、プロセッサは、 0x8000000-0x800C000Flash 域の IAP プログラムを実行している。IAP プログラムは USB メモリの BIN フ ァイルを取得し、かつ 0x800C000 アドレスの開始の FLASH 域に書き込む。FLASH を書きながら、D2 LED が次第に点灯から消灯になり、書き込まれた後に、D2 LED が点灯している。再び S2 キーを押し、IAP プログラムによりダウンロードされた FLASH のアプリプログラムが実行されている。

注意:開発ボードがフルスピード USB インタフェースを提供するので、フルスピード USB プロジェクト (STM324x91-EVAL_USBD-FS)を使用してください。

🧇 🎬 🎬 🧼 🔛 🎽	STM324x9I-EVAL_U	
Project	STM324x9I-EVAL_USBH	
	STM324x9I-EVAL_USBH	
🖃 🗟 STM324x9I-EVAL_U	STM324x9I-EVAL_USBH	
Application/MDK-ARM		

4.7.36 ¥Applications¥FreeRTOS¥FreeRTOS_ThreadCreation¥MDK-ARM

プログラムは FreeRTOS 操作ステムスレッドの作成機能を実現させる。プログラムは、二つの同じ優先度のスレッドが作成され、かつどちらでも実行サイクルが 15 秒であり、3 つの段階に分けられる。

最初の5秒間では、スレッド1が各200msにD2LEDの状態を切り替え、スレッド2が各500msにシリア ルポートに「Thread2 is running!」を送信する。次の5秒間では、スレッド1が自分自身を中断し、スレ ッド2がシリアルポートにデータを送信し続ける。最後の5秒間で、スレッド2がスレッド1をウェークア ップし、スレッド1が各400msにD2LEDの状態を切り替える。

4.7.37 ¥Applications¥FreeRTOS¥FreeRTOS_Timers¥MDK-ARM

プログラムは、FreeRTOS操作システムの内部システムタイマの使用方法を示す。プログラムは周期 200 ミリ秒のタイマを作成し、タイマーコールバック関数 osTimerCallback に D2 LED を一度否定させ、D2LED が各 200 ミリ秒に一回状態が変化する。

4.7.38 ¥Applications¥FreeRTOS¥FreeRTOS_Semaphore¥MDK-ARM

プログラムは FreeRTOS 操作システムのセマフォ使用方法を示す。プログラムは二つのスレッドを作成し、 スレッド1がスレッド2の優先レベルより高い。実行されると、スレッド1が優先にセマフォを取得し、D2 LED の状態が各 200ms に変化させ、かつシリアルポートに(ボーレートが 115200)「Thread1 is running!」 を送信する。5 秒後、スレット1がセマフォをリリースし、自分自身を一時停止する。この時、スレット2 は待っている過程に、スレット1がリリースしたセマフォを取得する。まず、スレット2はスレット1をウ ェークアップし(この時点でセマフォがスレット2に占有され、スレット1が待っている)、各 200ms にシ



リアルポートに「Thread2 is running!」を送信する。5 秒後、スレット2 がセマフォをリリースし、スレット1 が再びセマフォを取得する。

■ USB-COM - 超级终端	
文件(E) 編辑(E) 査看(V) 呼叫(C) 传送(I) 帮助(H)	
	1
Thread2 is running!	
Thread2 is running!	
Thread2 is running!	
Thread1 is running!	
连接的 1:01:04 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印	

4.7.39 ¥Applications¥FreeRTOS¥FreeRTOS_SemaphoreFromISR¥MDK-ARM

プログラムは、FreeRTOS 操作システムのセマフォ使用方法を示す。プロセッサは S2 キー対応する I/0 を外部割り込みに設定され、S2キーを押すと、外部割り込みを生成する。外部割り込みサービスルーチン (最終に HAL_GPI0_EXTI_Callback 関数を呼び出し)はセマフォをリリースし、テスト用スレッドがセマフ ォを取得して、D2 LED が一度否定される。

4.7.40 ¥Applications¥FreeRTOS¥FreeRTOS_Mutexes¥MDK-ARM

プログラムは FreeRTOS 操作システムのミューテックス使用方法を示し、優先度が高い、中、低レベルス レッドを作成する。

◆プログラムが実行された後、高いレベルのスレッドが優先にミューテックスを取得し、暫く待つと、 高いレベルのスレッドがミューテックスをリリースし、D2LED ステータスを否定させた後、高いレベル のスレッドが自分自身を一時中断する。

◆中レベルのスレッドがブロッキングモードでリリースされたミューテックスを待っている。ミュー テックスを取得してから、高いレベルのスレッドが中断されるかを確認する。高いレベルのスレッドが 中断されることを確認するまでに、中レベルのスレッドが「MutexMeduimPriorityThread is running!」 をプリントする。高いレベルのスレッドが中断されることを確認された後、中レベルのスレッドも自分 自身を一時中断される。この時、両方も中断された。

◆低レベルのスレッドもブロッキングモードでリリースされたミューテックスを待っている。ミュー テックスを取得してから、高い・中レベルのスレッドが中断されるかを確認する。高い・中レベルのス レッドが中断されることを確認するまでに、低レベルのスレッドが「LowPriorityThreadCycles++!」を プリントする。ハイパーターミナルには、次の情報が見られる(ボーレート 115200)。



■ USB-COM - 超级终端		
文件(F)编辑(E) 查看(V) 呼叫(C) 传送(D) 帮助(H)		
LowPriorityThreadCycles++!		
MutexMeduimPriorityThread is running!		
LowPriorityThreadCycles++!		
MutexMeduimPriorityThread is running!		
LowPriorityThreadCycles++!		
MutexMeduimPriorityThread is running!		
LowPriorityThreadCycles++!		
MutexMeduimPriorityThread is running!		
LowPriorityThreadCycles++!		
MutexMeduimPriorityThread is running!		
LowPriorityThreadCycles++!		
MutexMeduimPriorityThread is running!		
	-	
连接的 0:38:18 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印	1.	

4.7.41 ¥Applications¥FreeRTOS¥FreeRTOS_Queues¥MDK-ARM

プログラムは FreeRTOS 操作システムのメッセージ・キュー使用方法を示す。キューーつとスレッド二つを 作成し、スレッドが増分データを生成してからキューに送信する;他のスレッドがキューからデータを取得 し、かつ正確かを判断する。プログラムが実行された後に、D2 LED が各 250ms に状態を変更し、メッセージ ・キューには受送信問題が出る場合に、ポートには対応する情報が出力する。

4.7.42 ¥Applications¥FreeRTOS¥FreeRTOS_LowPower¥MDK-ARM

プログラムは FreeRTOS 操作システムで低電力モード機能が達成する方法を示す。キューーつとスレッド 二つを作成し、スレッドがメッセージ・キューにデータを送信するために使用され、別のスレッドがメッセ ージキューからデータを取得するために使用される。

◆まず、データ送信スレッドが 500ms スリープし、スリープにプロセッサは低電力モードに入る。ス リープ時間が終わると、スレッドがメッセージキューにデータを送信する。受信スレッドがずっとメッ セージキューが受信したデータを待っている。メッセージキューがデータを受信された後、D2 LED が 1 回点滅する。

4.7.43 ¥Applications¥Camera¥Camera_To_USBDisk¥MDK-ARM

プログラムは、カメラが採集した画像を USB メモリに保存され、デジタルカメラの機能を実現させる。

◆カメラの二つピンが RS485 の RXD 及び TXD と共用している。テスト前に、R56、R57 の 0 Ω 抵抗を除 去する(出荷時のデフォルト状態)。

◆予めに、USBメモリを J8 インタフェースに挿入する。カメラモジュールを JP3 Pin1-16 ピンに挿入 し、以下は挿入方法である。





◆実行された後、S2 キーを押すと、LCD 画面に表示された画像を USB メモリに保存され、画像の画素 は 480×272、下記は採集された画像である。

