ARM Cortex-M3 STM32F407

Sample source manual

Ltd. NichiNoboru Technology

http://www.csun.co.jp

info@csun.co.jp

2015/12/22

- **Correction history**

NO	version	Fixes	Modification date
One	Ver1.0	Create New	2015/12/22

table of contents

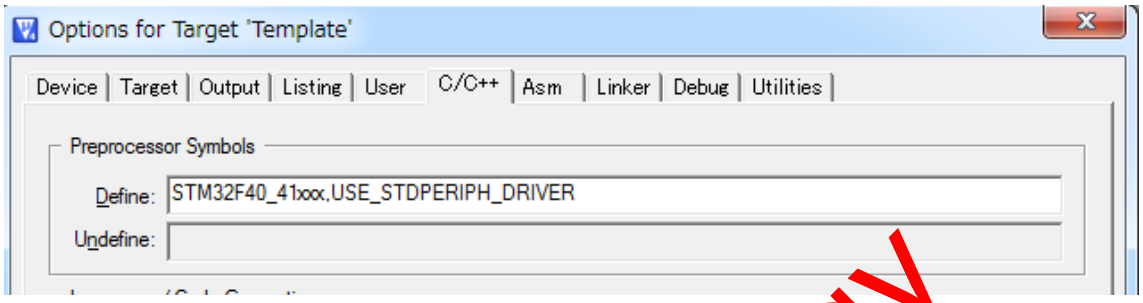
(1) Template

This sample, please reference when creating a new project. Based V1.3.0 of the

project of the firmware of STM32F4.

Note: when you create a new project, all of the macro definition of Define the place is STM32F40_41xxx of C / C ++ tag of

Option for target xxx, you need to USE_STDPERIPH_DRIVER.



(2) WaterLED

This sample is to control two LED of STM32F407 development board (DS0 and DS1), alternate to blink.

(3) KEY

This sample through STM32F407 2 one of the buttons of the development board (KEY0, KEY1), to control

the two LED (DS0 and DS1) of the board. KEY1 button controls the DS1, turned on by pressing once, off the

press again. KEY0 button controls the DS0 turned on by pressing once, off the press again.

(4) UART

The serial port 1 is the sample continues to send the message on your PC, receives the data from the simultaneously

serial port, and transmits the received data to the PC. Note: To set the Poreto of serial port to 115200bps.

Hardware: STM32F407

development board

RS232C-TTL level converter board USB RS232

converter cable (D Sabuosu) 4 Pin array conversion

cable

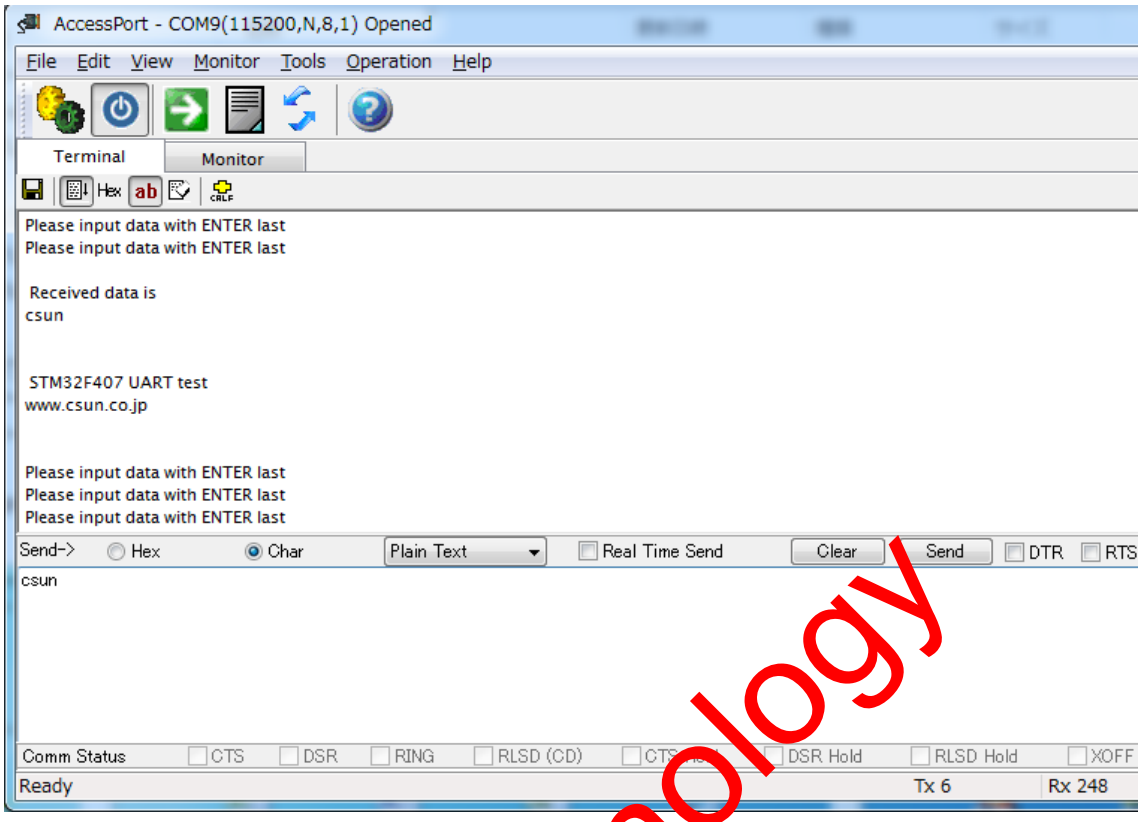
Connection method: STM32F407 connect the RX of the RX and RS232C-TTL level converter board serial port 1 of the development board, connect the

TX and a converter board serial port 1 TX. VCC and GND and 5V power supply and GND

Connecting.

Hardware connection Image:

Serial port output image:



(5) INT

This sample through the two buttons of STM32F407 development board with an external interrupt (KEY0, KEY1), to control the two LED (DS0 and DS1) of the board. KEY1 button controls the DS1, turned on by pressing once, off the press again. KEY0 button controls the DS0 and DS1 at the same time, turned on by pressing once, off the press again.

(6) Watchdog

If this sample is necessary to reset the watchdog (watchdog), DS0 is much lights. When you press the button WK_UP, to feed. If continue to press the WK_UP button, watchdog is much without resetting, DS0 also much lights. Once the watchdog installation time if you do not press the WK_UP button beyond the (one second), the program will restart, DS0 will be turned off once.

(7) WinWatchdog

This sample through DS0, indicating STM32 is reset. If you 300ms lighting the DS0 When you are reset.

DS1 shows the interrupt watchdog. The time DS1 to interrupt generation to rotate once. STM32 is to be reset, DS0 is much turned off.

(8) Timer

This sample shows a project running in DS0, the period is 400ms. DS1 shows the execution of the timer interrupt. Rotating at interrupt. Period is 1000ms. Phenomenon of post-execution, DS0 flashes fast, DS1 flashes on and off a little slowly.

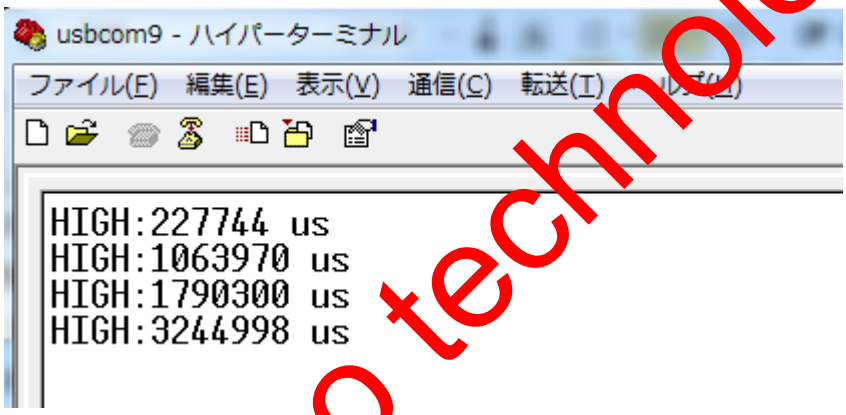
(9) PWM

This sample creates a PWM using TIM14_CH1, to control the DS0 of brightness. Phenomenon after execution is a dark → bright → dark → bright → of circulation.

(10) Input

This sample is to get a high level of PA0 using TIM15_CH1. To create a high-level press the WK_UP. And it outputs a pulse width of the high level from the serial port. It has left even the treatment of the same PWM in the previous sectio

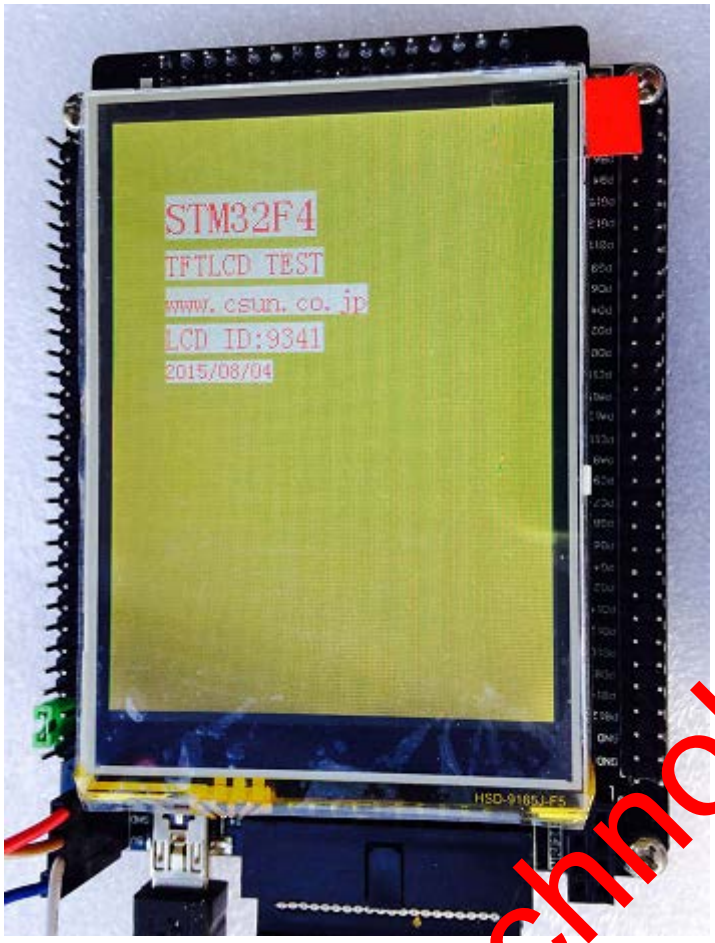
Output example of the serial port (time pressed the WK_UP key is output):



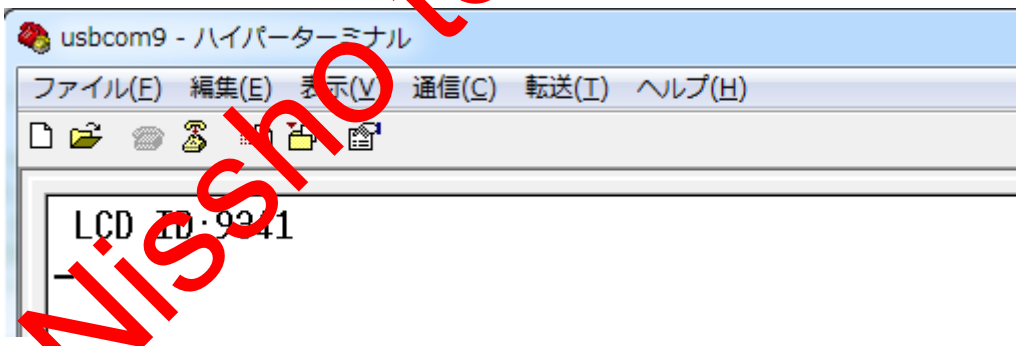
(11) TFT LCD

This sample is to realize a display of TFTLCD. After the execution, to display the text information to the LCD, automatically switch the background color. And also outputs the ID of the LCD driver every time you reset from the serial port.

Executable image:



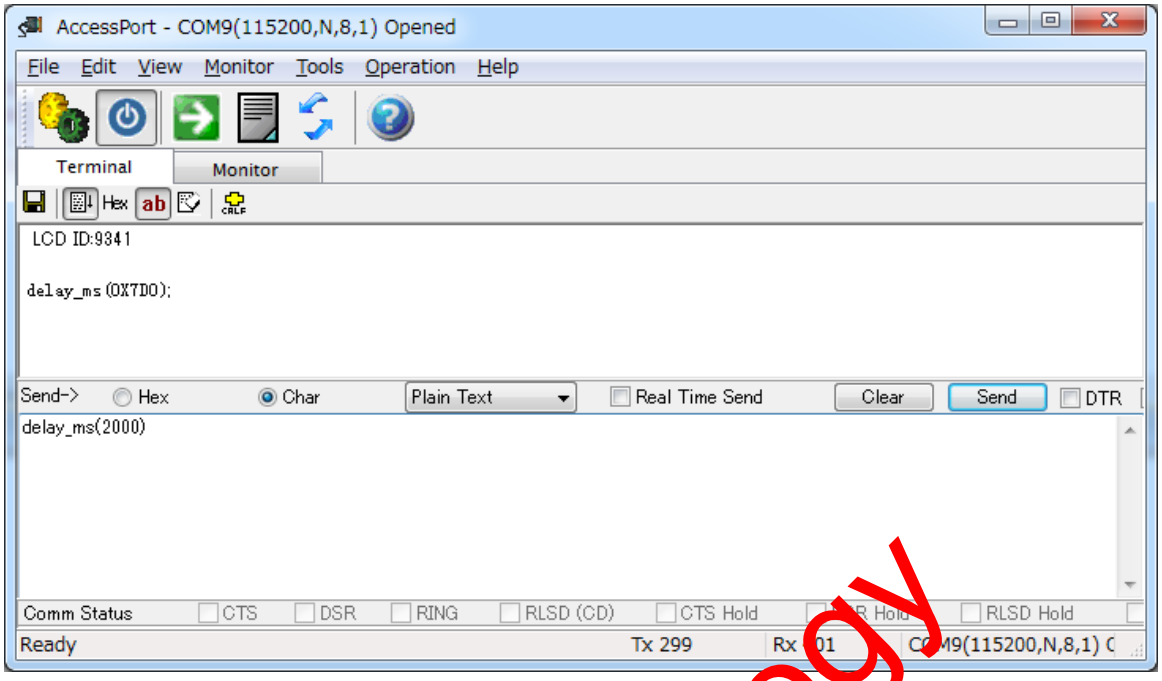
Serial port output:



(12) USMART

This sample is to control the display and delay of LCD and LED by calling the built-in MCU function using usmart.

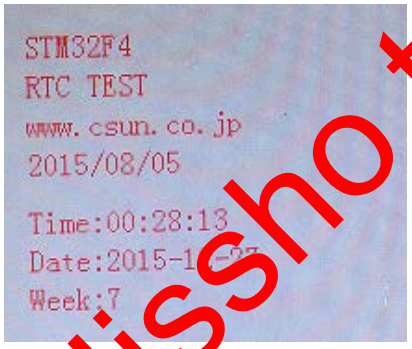
Execution example: When you dispatch from the serial port delay_ms the (2000), DS0 of the state is extended. Serial port output state:



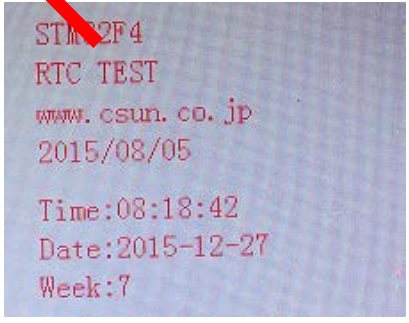
(13) RTC

This sample displays the RTC time in TFTLCD. Also wear set the RTC time using usmart. Execution example:

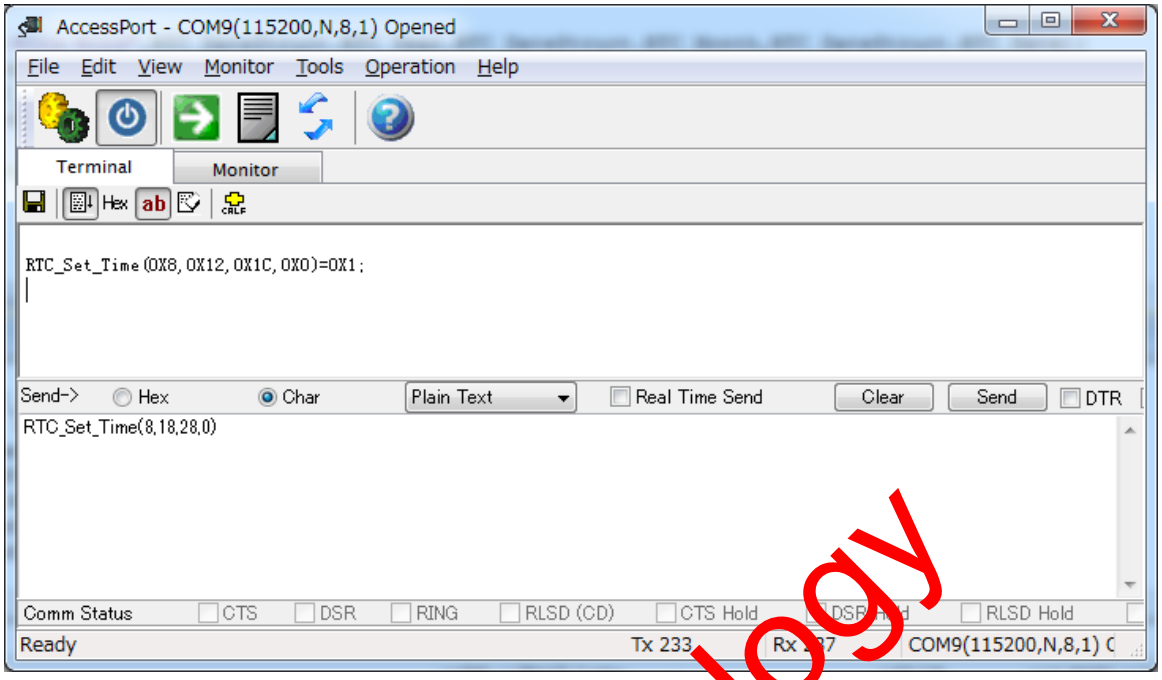
Time setting before the display:



Display after time setting from the serial port:



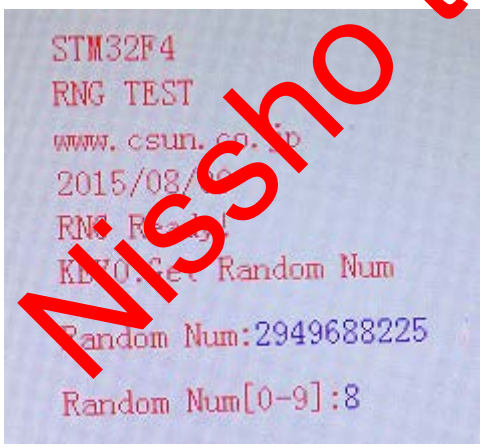
Serial port information:



(14) RandomGen

This sample generates an random number in STM32F4 built-in hardware random number generator of the (RNG), it is displayed on the LCD. To get a random number by pressing the KEY0 key. Also to obtain the random number in the range of 0-9, and displays it on the LCD. D0 shows the program execution state.

Executable image:

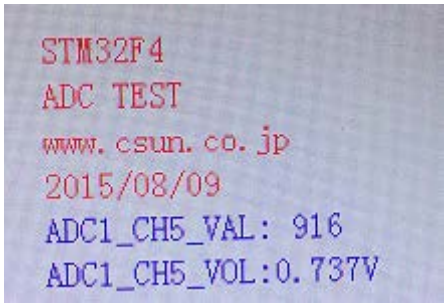


(15) ADC

This sample is to get the voltage of the channel 5 (PA5) in ADC1, to display the voltage value after the ADC conversion value and conversion on the LCD.

Note: The reference voltage of this test is 3.3V. If you want to use the other reference voltage, it can be set in P7 port of STM32F4 board. After you set the other reference voltage, the input voltage is necessary to be careful so as not to exceed the maximum value of the reference voltage.

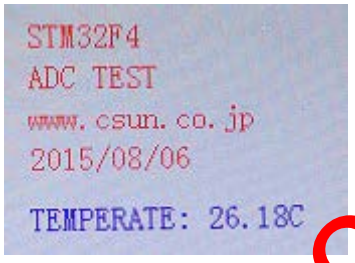
Executable image:



(16) Temprature

This sample is converted to a temperature to obtain the voltage value of STM32F4 internal temperature sensor in the channel 16 of ADC1, is displayed on the LCD.

Executable image:



(17) DAC

This sample is to control Lumia the output voltage of the channel 1 of STM32F4 built-in DAC with a key or USMART. The DAC output voltage in the channel 5 in ADC1 taken is displayed on the LCD is. Also you can set the output voltage of the DAC by calling the Dac_Set_Vol function in Usmart.

Note: need to short-circuit the PA4 and PA5 pin of the board.

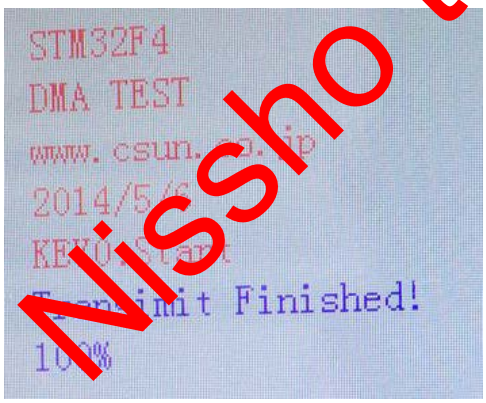
(18) PWM DAC

This sample is to control the PWM output of TIM9_CH2 of STM32F4 in key or USMART. It is converted to the DAC output after RC filter to collect the output voltage of the PWM DAC channel 5 of ADC1 displayed on the LCD to.

Note: need to short-circuit the PA3 and PA5 pin of the board.

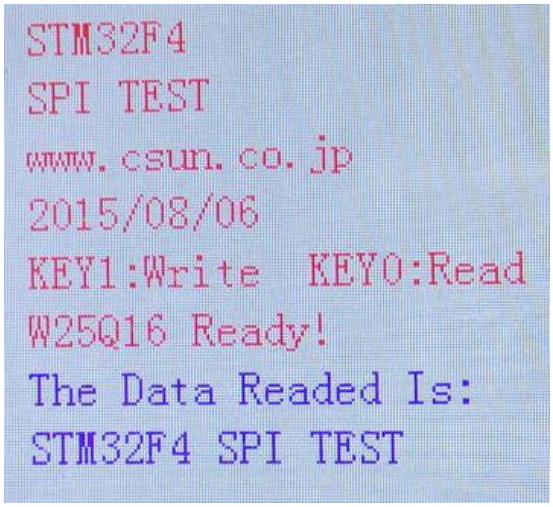
(19) DMA

This sample is to control the data transmission of the DMA serial port, in KEY0 key. When you press the KEY0, DMA transfer starts, displays the transfer progress at the same time on the LCD. You can receive the contents of the DMA transfer in the serial debugging tools. Note 1: serial Baurette 115200.



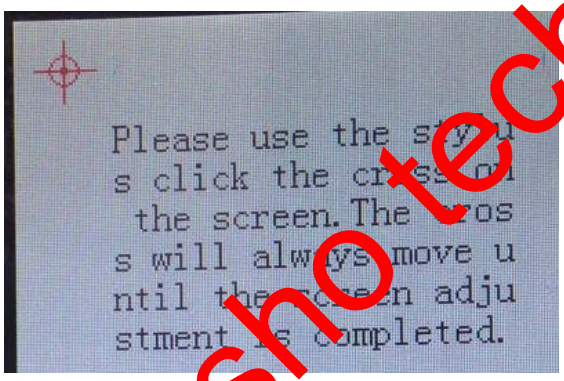
(20) SPI

This sample is to control the written to the W25Q16 in KEY1 key. Control the reading of W25Q16 or et al in KEY0 key. At the same time to display the information on the LCD.



(21) TouchPanel

This sample is static electricity or the touch panel by the first LCD ID, carry out the relevant instruction to check whether resistive touch panel. The default is resistive touch panel. To check whether calibrated, to perform the calibration if there is. If the calibration enters the handwriting program. There are chestnut Aeria (RST) on the screen, are all Click here to clear. In addition to running the calibration KEY0.



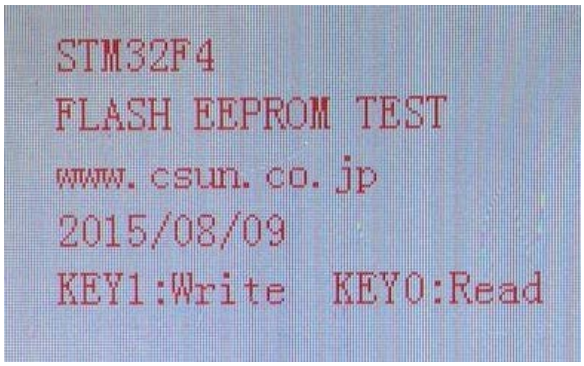
(22) NRF24L01

This sample is when you start first of all, to make sure whether there is NRF24L01 module. NRF24L01 after gage modules, check the module's mode of operation depending on the installation of KEY0 and KEY1, after setting the mode of operation accurately, it is possible to transmit / receive data continuously, running in DS0 at the same time It shows that you are. Caution:

This test is a development board plus two NRF24L01 wireless module of the two sets, it is a test child successfully. It can not be tested in one of the development board and one of the modules.

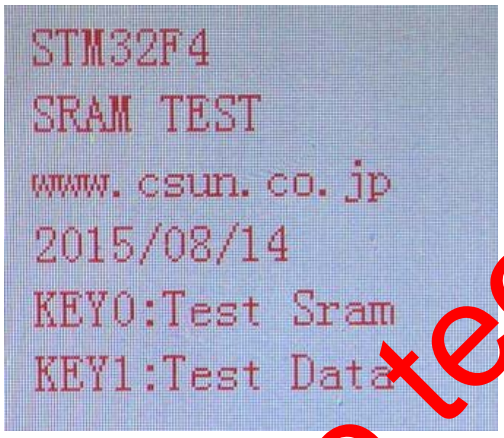
(23) FLASH2EEPROM

This sample is first of all, to display the information presented on the screen when you start, and then to measure the two buttons in the main loop, one of the buttons (KEY1) executes a written of FLASH. Another button (KEY0) performs a read. To view the related information on the TFTLCD. It shows that it is running in a DS0.



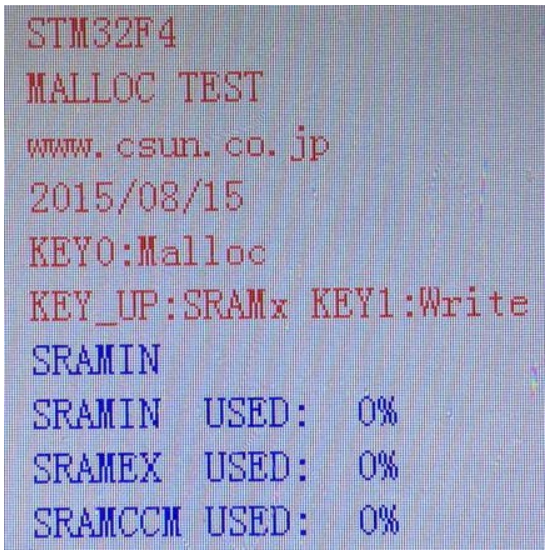
(24) SRAM

After this sample was started, to display the information presented on the screen, by pressing KEY0 key, to measure the size of the external SRAM capacity, to display on the LCD. When you press the KEY1 key, to display the data in advance saved external SRAM. It shows that it is running in a DS0.



(25) Memory

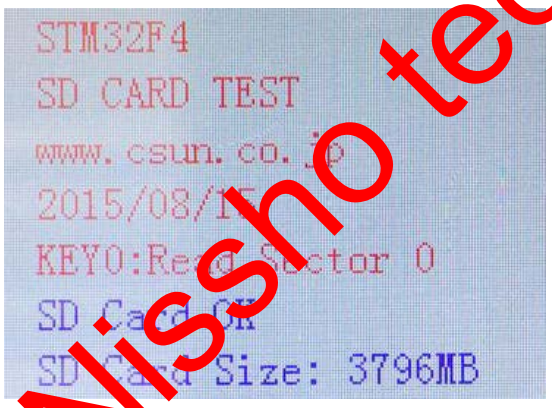
The sample after the start, and display the information presented on the screen, waiting for the external input. KEY0 apply for memory. To apply for a memory of 2K bytes each time. Function of KEY1 write the data into the memory of the application. KEY_UP to realize that switching the operation memory area (internal SRAM memory / external SRAM memory / internal CCM memory). At the same time indicating the is running in a DS0.



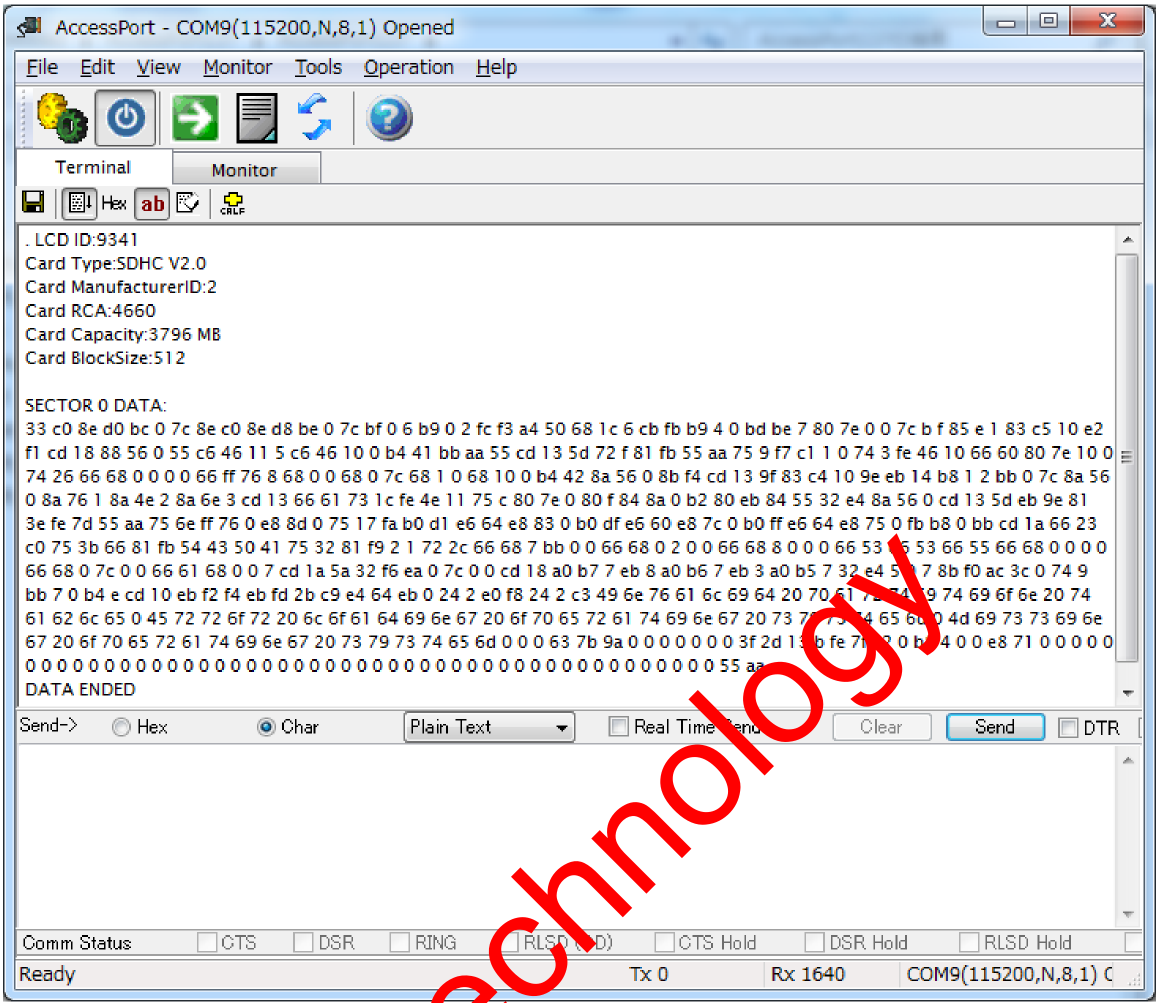
(26) SD

This sample is to initialize the SD card earlier when you start. If successful, prompted that the LCD to initialize, press the KEY0, it reads the data of the sector 0 of the SD card, shipped to PC with a serial port. If you can not be initialized, to display a failed information on the LCD. It shows that it is running in a MCS0.

LCD display content:



(If you press the KEY0) serial port output image:

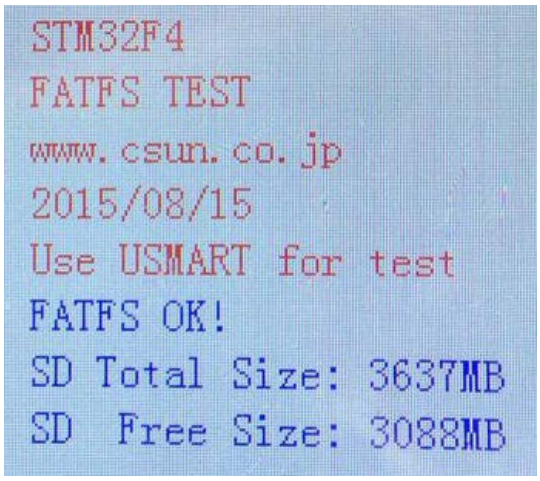


(27) FATFS

This sample is to initialize the SD card earlier when you start. If successful, to register the two work areas (one for SD card, because one is for SPIFLASH, to measure the capacity of the SD card and a spare space, and display on the LCD, the command from the last USMART in performing the test. It shows that it is running in a DS0. Caution:

1, Please prepare one of the SD card.

Through 2, USMART, to test by calling the various fatfs.



(29) IMAGE

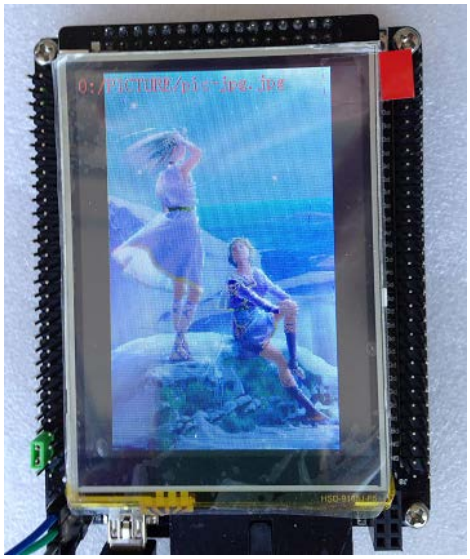
This sample is to determine whether the presence of the SD card when you want to start. If present, the search for the PICTURE folder under the SD card root directory. Find and in the file of the image of the bottom of this folder (bmp, jpg, to support the jpeg or gif) and displayed in a loop, KEY0 and KEY1, can browse the PICTURE. WK_UP key is a function of the pause / play, DS1 is to indicate whether suspend the time being of the state. If you can not find the file in the PICTURE folder / image, to view the presentation of the error. This test shows that it is running in a DS0. Caution:

1, this test, please provide one of the SD card. And root directory in the SD card

Li in the PICTURE folder making, please put some of the image (BMP / JPG / JPEG / GIF).

2, Once if it read the part jpg / jpeg, Kudazai to open and save in a paint tool in Windows XP.

3, JPEG / JPG / BMP can be automatically zoomed by the LCD resolution. GIF is not read at night in the following LCD resolution.



(30) AVPlay

This sample source to achieve the following functions: After launching, first to initialize the peripheral device. If there is no yellow tail problem, begin to play the video (avi format) in the VIDEO files in the TF card.

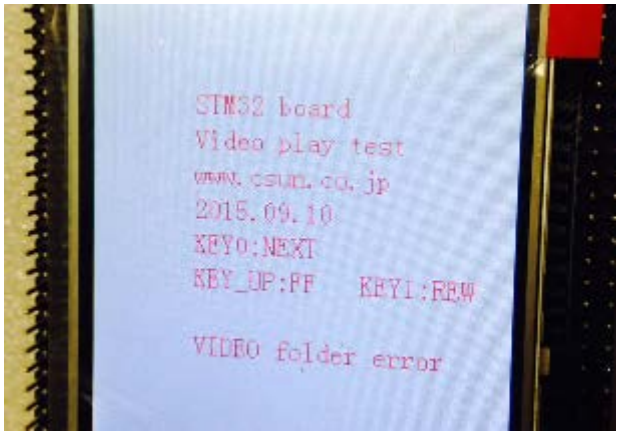
When you play the video, showing the name of this video on the TFT LCD, number, the total number of video, sound tracks the number, audio sampling rate, frame rate, also information such as the playback time and total time. Press the KEY0 key to play the next video. WK_UP key is fast forward, KEY1 key can Mr. Hayamodo. Caution:

1, need to prepare a single TF card.

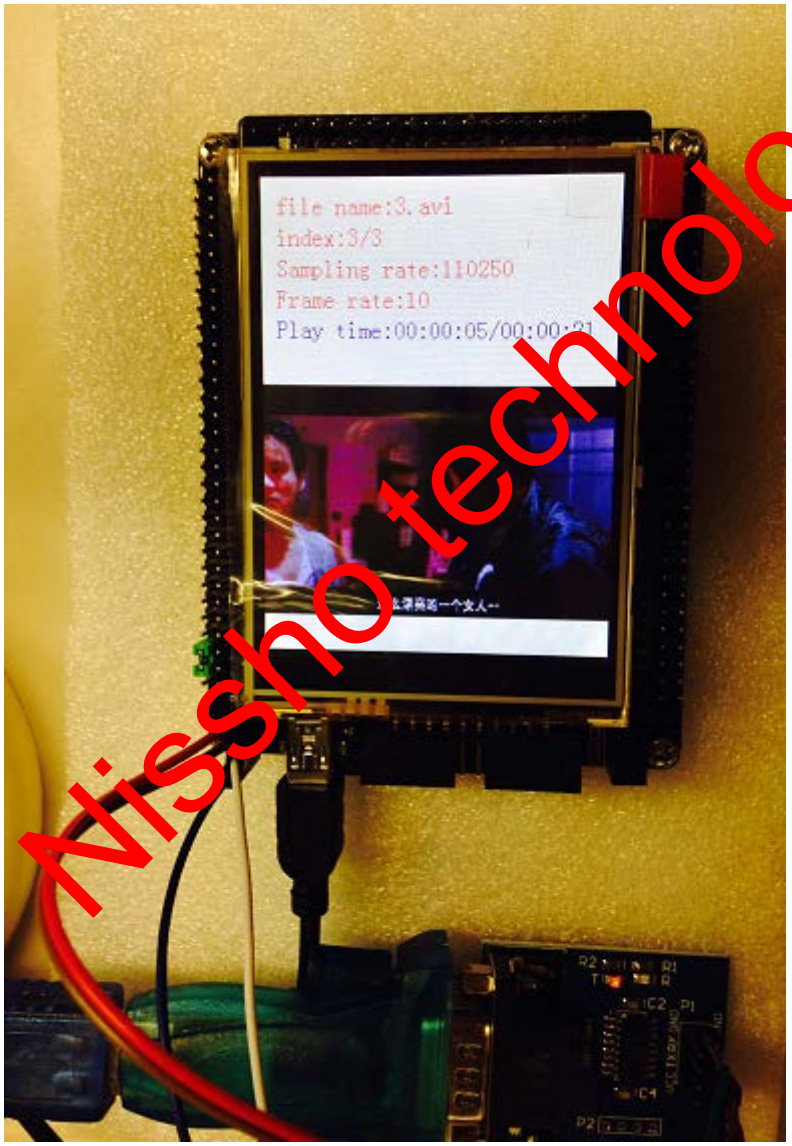
2, to create a VIDEO folder in the root catalog of the TF card, AVI video file (video only supports MJPG, audio should be PCM. And, the resolution of the video is not the less than or the same LCD resolution It shall) put. 3, in the present board, because the audio decoding is not mounted, voice can not be confirmed.

Im[illegible] the time of execution

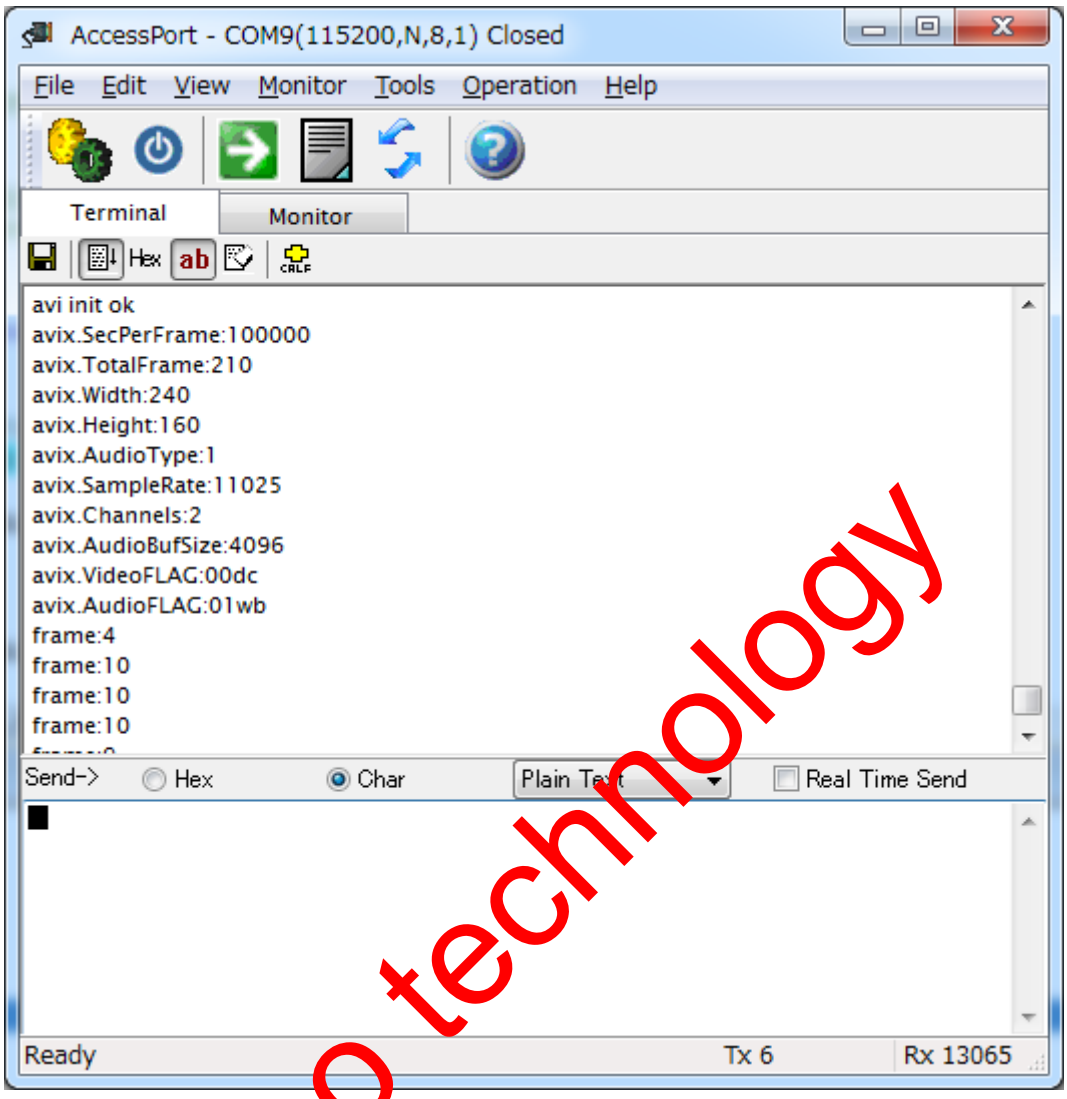
1, If you have not inserted TF card:



2, avi files saved If you insert TF card are:



Information output from the serial port:



(31) FPU (Julia)

This sample source to achieve the following functions: After launched, to create a color table than the number of iterations (RGB565), and is displayed on the LCD to calculate the Julia minute form. Further, for ease of observation compared, in order to statistically one frame time, the program starts the timer 3, after finishing to view the Julia fraction form drawing of one frame, the execution time of the program, FPU use It shows information such as the zoom magnification and whether it has been. It is possible to adjust the zoom magnification KEY0 / KEY1. To set the automatic zoom and manual zoom in WK_UP. DS0 is showing the execution state of the program. Caution:

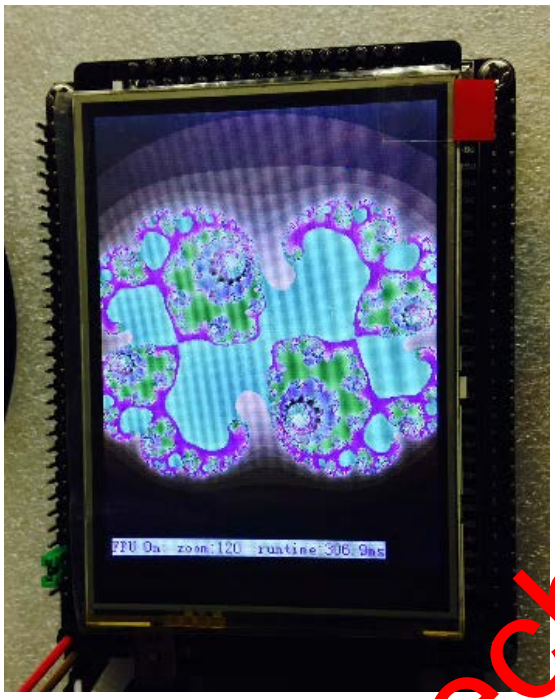
1, "46_1_FPU (Julia) _openHardwareFPU" and of "46_2_FPU (Julia) _closeHardwareFPU" Soviet Sukodo is exactly the same, only the hardware FPU only turned on and off.

2, when you want to test, first download one, to record the corresponding time and parameters. And download the other one, the same parameters, in particular, look at the time, wear in the comparison.

46_1_FPU (Julia) _openHardwareFPU during the run:

46_2_FPU (Julia) _closeHardwareFPU

During the run:



(32) DSP

For 47_1_DSP-BasicMath

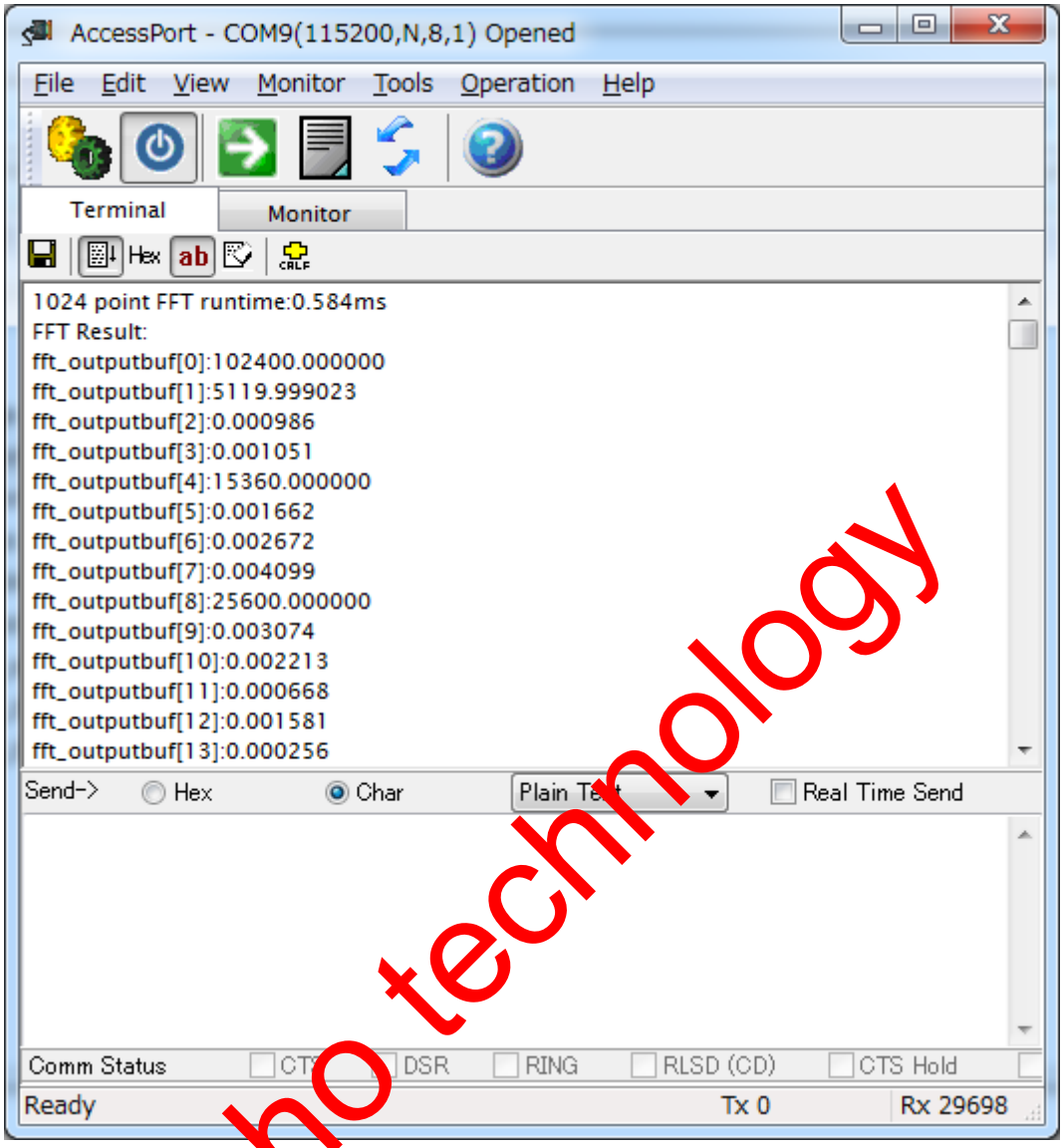
STM32F4 basic math functions to the DSP library of: arm_cos_f32 and arm_sin_f32, and the standard library of basic mathematical function: Check the speed difference between the cosf and sinf, to display the time it takes the two of calculation on the LCD. DS0 instructs that the program is in a running state

Executable image:

For 47_2_DSP-FFT

STM32F4 to test the FFT function of the DSP library. After the program execution, 1024-point inspection order is created automatically. Then, each time you press the KEY0, performing an FFT calculation by calling FFT calculation method of the DSP library (based on 4 method). LCD to display the computation time, also outputs the results of the FFT from the serial port at the same time. DS0 instructs that the program is in a running state. Executable image:

Serial port output:



(33) HandWriter

This example source to achieve the following functions: launch, enter the calibration mode of the touch panel. Follow the on-screen prompts, click four times the cross mark. If you do not enter 10 seconds ending with the automatic. And, in a state to wait for input. To enter a number or Kayaraku data on the LCD in the handwriting area. Each time you have finished the input, identify entered into automatically identify state. In addition to display the results of identification in LCD (at the same time also be output to the serial port). When you press the KEY0, it is possible to switch the mode (there are four kinds of mode), and press the KEY1, entering the calibration mode of the touch panel. DS0 instructs a state in which the program is running.

Note: Because the capacitive touch screen does not need to be adjusted, when using the capacitive screen, not out reaction also press the KEY1.

Executable image:

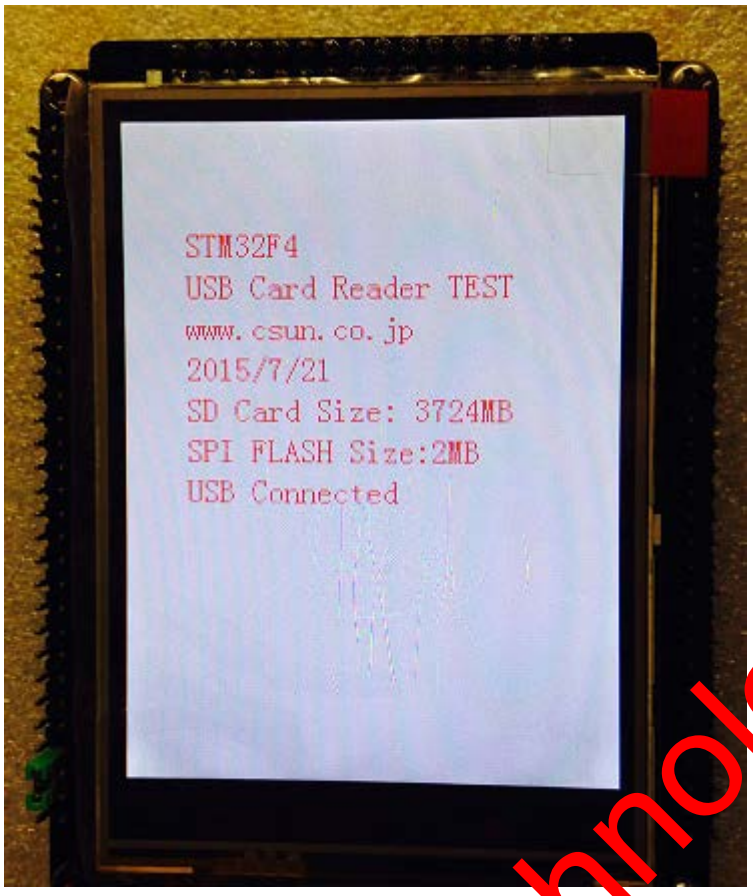
(35) USBCardReader (Slave)

This sample source to achieve the following functions: when to launch, when the SD card and the SPI FLASH to test whether the presence, if present, to display its capacity on the LCD. (If it does not exist, it informs the error). Then, begin to set the USB. After the setting is successful, two removable disk find on the PC. It indicates that the USB in DS1 is accessing, also displayed on the LCD. DS0 shows a state in which the program is running.

Caution:

Please prepare a single 1, SD card. (If not, you do not use only SPI FLASH disk mounted on the board).

2, need to connect the board to the PC with a USB cable.



(37) UCOSII1-1-EventCall

This sample is to realize the following functions. To create a three tasks in UCOSII: Start task, LED0 tasks and LED1 task. Start task is to hang after creating the other tasks (LED0 tasks and LED1 task). LED0 task is to control the DS0 LED, 80ms lights up for one second. LED1 task is to control the DS1 LED, to 300ms off to 300ms lighting. ucosii version used in this test is V2.91.

(38) UCOSII1-2-EventOther

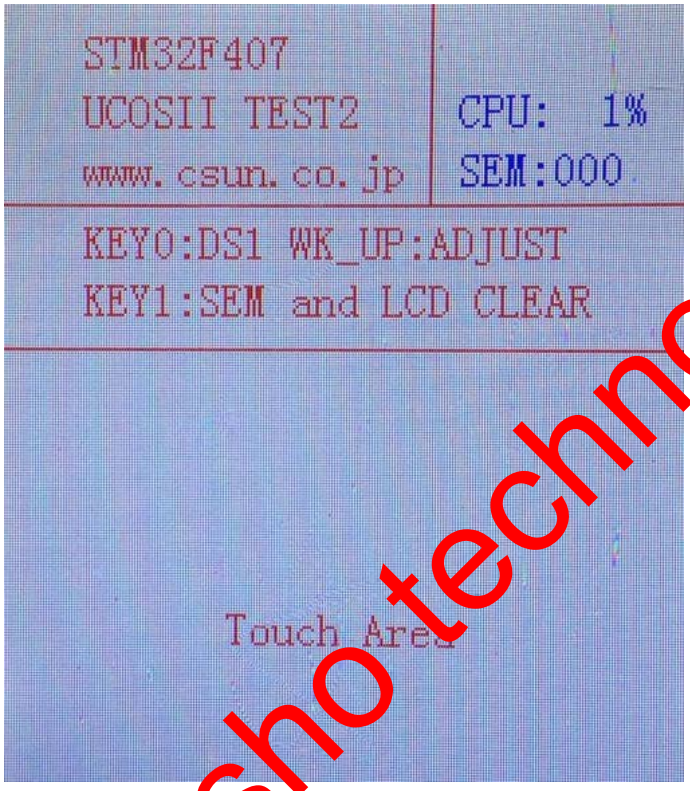
This sample is to realize the following functions. To create a three tasks in UCOSII: Start task, LED task and the KEY task. Start task is to hang after creating the other tasks (LED task and the KEY task). LED task is to control the DS0 / DS1 LED. Press KEY0 turned off to hang the LED task. Press KEY1 lights and restart the LED task. ucosii version used in this test is V2.91.

(39) UCOSII-2-SemaphoreMailbox

This sample is to realize the following functions. To create a six tasks in UCOSII: Start Task,

LED task, touch panel task, BEEP task, the main task and the KEY task. Start task is to hang after creating the other tasks. LED task is to control the DS0 LED. BEEP task is to apply for a semaphore. Touch panel task is to test the usage of Handwritting and the CPU. KEY task is to scan a key, highest priority, after scanning the key, to ship in the Mailbox. The main task will control a variety of tasks to find the key in the Mailbox.

To control the blinking of DS1 in KEY0. To apply for a semaphore in KEY1, to display the current value in the LCD, to clear the display of Handwritting area at the same time. Calibrate the touch panel at WK_UP.



ucosii version used in this test is V2.91.