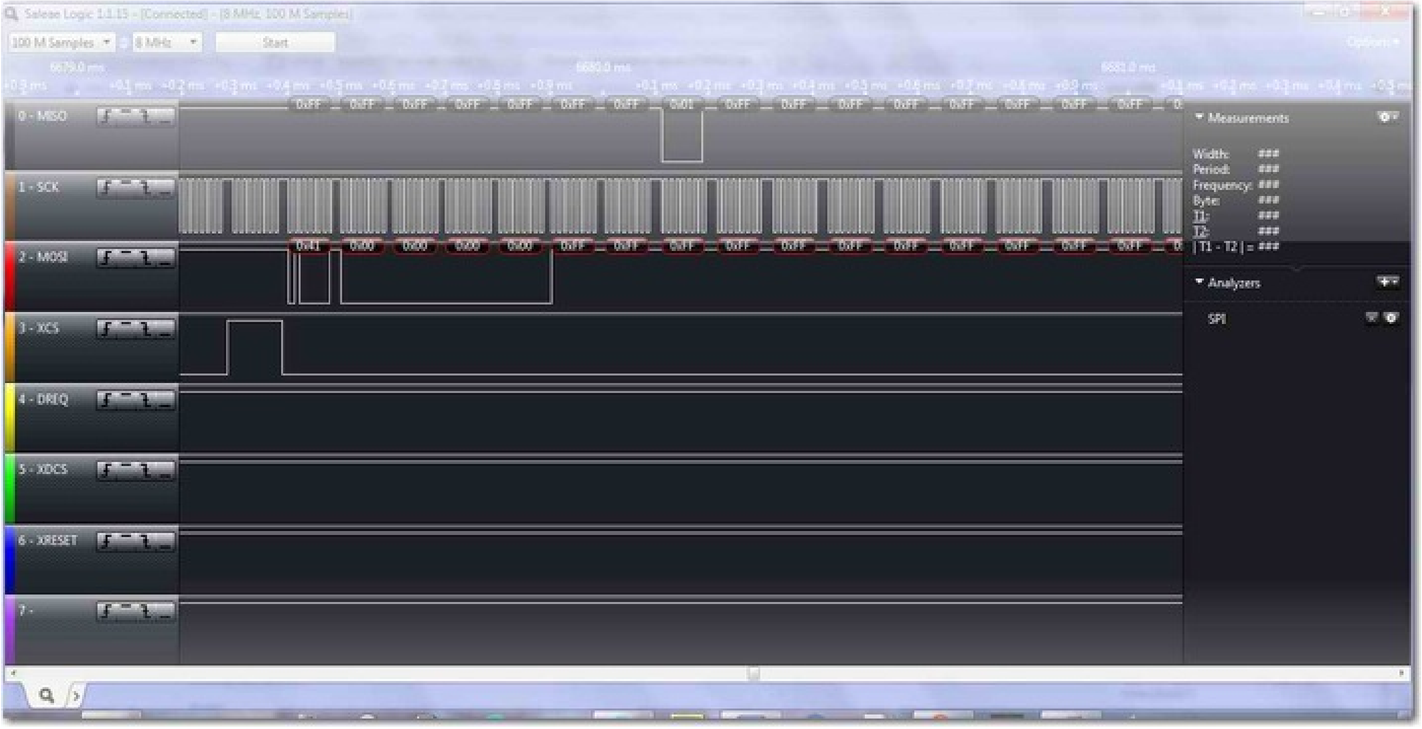
Guys,

I tried to compile the application example from STM32Cube FW F1 V1 FATFs,

I can see a signal on logic analyzer, but it hasn't written STM32.TXT yet...

What else I can check ?

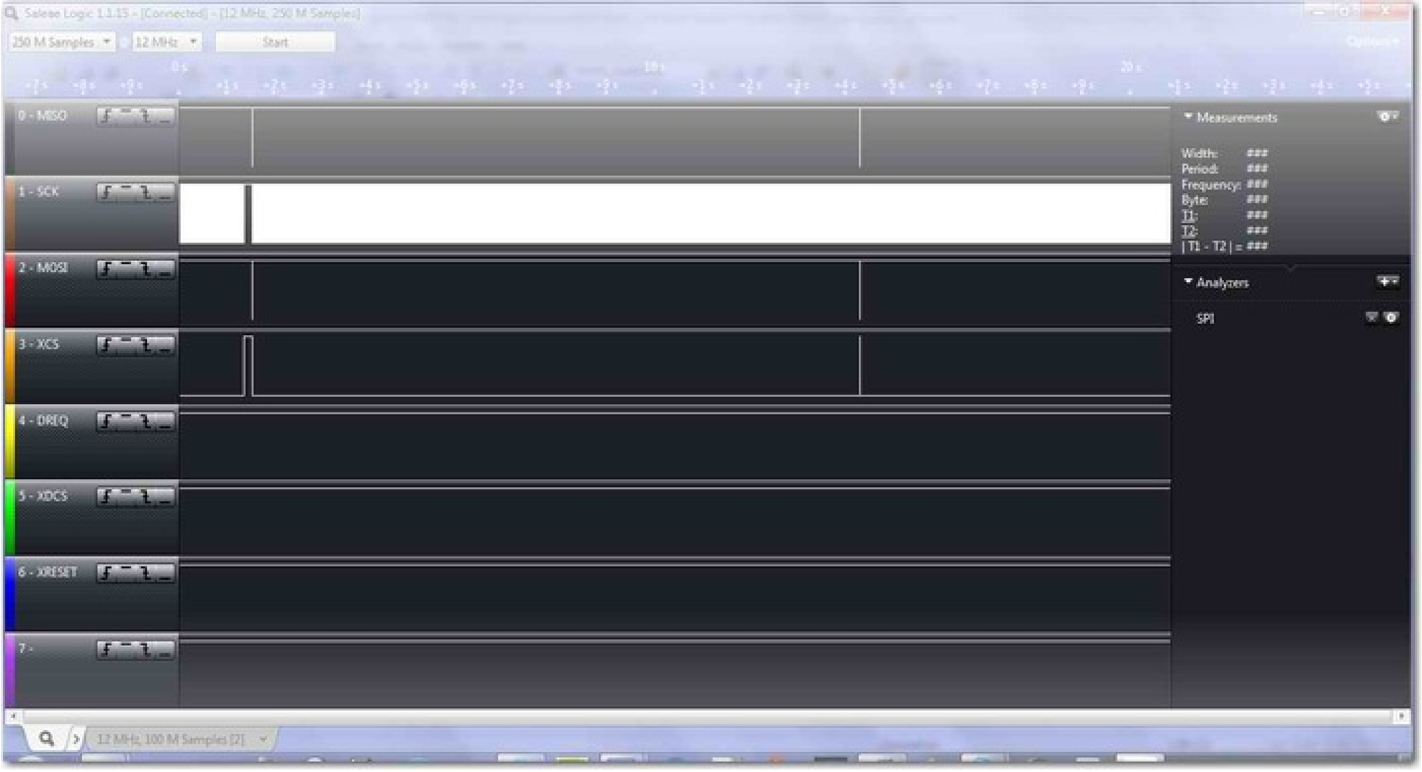
I'm using 8GB Micro SD...



The code :

```
int main(void)
{
  FRESULT res;                                          /* FatFs function common result code */
  uint32_t byteswritten, bytesread;                     /* File write/read counts */
  uint8_t wtext[] = "This is STM32 working with FatFs"; /* File write buffer */
  uint8_t rtext[100];                                   /* File read buffer */
     printf("** Test finished successfully. ** \n\r");

  /* STM32F107xC HAL library initialization:
       - Configure the Flash prefetch
       - Systick timer is configured by default as source of time base, but user
         can eventually implement his proper time base source (a general purpose
         timer for example or other time source), keeping in mind that Time base
         duration should be kept 1ms since PPP_TIMEOUT_VALUEs are defined and
         handled in milliseconds basis.
       - Set NVIC Group Priority to 4
       - Low Level Initialization
     */
  HAL_Init();

  /* Configure the system clock to 72 MHz */
  SystemClock_Config();

  /* Configure LED_GREEN and LED_RED */
  BSP_LED_Init(LED_GREEN);
  BSP_LED_Init(LED_RED);
```

```
/*##-1- Link the micro SD disk I/O driver ##################################*/
if(FATFS_LinkDriver(&SD_Driver, SDPath) == 0)
{
  /*##-2- Register the file system object to the FatFs module ##############*/

  if(f_mount(&SDFatFs, (TCHAR const*)SDPath, 0) != FR_OK)
  {
    /* FatFs Initialization Error */
    Error_Handler();
  }
  else
  {
    /*##-3- Create a FAT file system (format) on the logical drive #########*/
    /* WARNING: Formatting the uSD card will delete all content on the device */
    if(f_mkfs((TCHAR const*)SDPath, 0, 0) != FR_OK)
    {
      /* FatFs Format Error */
      Error_Handler();

    }
    else
    {
      /*##-4- Create and Open a new text file object with write access #####*/

      if(f_open(&MyFile, "STM32.TXT", FA_CREATE_ALWAYS | FA_WRITE) != FR_OK)
      {
        /* 'STM32.TXT' file Open for write Error */
        Error_Handler();

      }
      else
      {
        /*##-5- Write data to the text file ################################*/

        res = f_write(&MyFile, wtext, sizeof(wtext), (void *)&byteswritten);

        /*##-6- Close the open text file #################################*/

        if (f_close(&MyFile) != FR_OK )
        {
          Error_Handler();
        }

        if((byteswritten == 0) || (res != FR_OK))
        {
          /* 'STM32.TXT' file Write or EOF Error */
          Error_Handler();
        }
        else
        {
          /*##-7- Open the text file object with read access ###############*/
          if(f_open(&MyFile, "STM32.TXT", FA_READ) != FR_OK)
          {
            /* 'STM32.TXT' file Open for read Error */
            Error_Handler();
          }
          else
          {
            /*##-8- Read data from the text file ###########################*/
            res = f_read(&MyFile, rtext, sizeof(rtext), (UINT*)&bytesread);

            if((bytesread == 0) || (res != FR_OK))
            {
              /* 'STM32.TXT' file Read or EOF Error */
              Error_Handler();
            }
            else
            {
              /*##-9- Close the open text file #############################*/
              f_close(&MyFile);

              /*##-10- Compare read data with the expected data ############*/
              if((bytesread != byteswritten))
              {
                /* Read data is different from the expected data */
                Error_Handler();
              }
              else
              {
                /* Success of the demo: no error occurrence */
                BSP_LED_On(LED_GREEN);

              }
            }
          }
        }
      }
    }
  }
}

/*##-11- Unlink the RAM disk I/O driver ####################################*/
FATFS_UnLinkDriver(SDPath);
```



Thanks