

Step 1:

```
FATFS mynewdiskFatFs; /* File system object for User logical drive */
FIL MyFile; /* File object */
```

Step 2:

```
void MX_USB_HOST_Init(void)
{
    /* Init Host Library,Add Supported Class and Start the library*/
    /*##-2- Init Host Library #####*/
    // HOST_FS = ? Confusion .In Demo it is '0' , In cube Generate it is '1' .
    USBH_Init(&hUsbHostFS, USBH_UserProcess, HOST_FS);
    /*##-3- Add Supported Class #####*/
    USBH_RegisterClass(&hUsbHostFS, USBH_MSC_CLASS);
    /*##-4- Start Host Process #####*/
    USBH_Start(&hUsbHostFS);
}
```

Step 3:

```
/*
 * Background task
 */ void MX_USB_HOST_Process(void)
{
    /* USB Host Background task */
    /*##-5- Run Application (Blocking mode) #####*/
    USBH_Process(&hUsbHostFS);
}
```

Step 4:

```
uint32_t wbytes; /* File write counts */
uint8_t wtext[] = "text to write logical disk"; /* File write buffer */
// USBH_Driver Copied from #include "usbh_diskio.h"
// USBH_Path Copied from #include "Fatfs.h"
if(FATFS_LinkDriver(&USBH_Driver, USBH_Path) == 0)
{
    MX_USB_HOST_Init(); // Calling Function
    MX_USB_HOST_Process(); // Calling Function
    if(f_mount(&mynewdiskFatFs, (TCHAR const*) USBH_Path, 0) == FR_OK)
    {
        // Not able to Create File
        if(f_open(&MyFile, "Try.TXT", FA_CREATE_ALWAYS |
        {
            if(f_write(&MyFile, wtext, sizeof(wtext), (void *)&wbytes) == FR_OK);
            {
                wbytes = 0;
                f_close(&MyFile);
            }
        }
    }
    else
        Error_Handler();
}
FATFS_UnLinkDriver(USBH_Path);
```