

HT82A525R DMA Interface 在指纹识别中的运用

文件编码: HA0251S

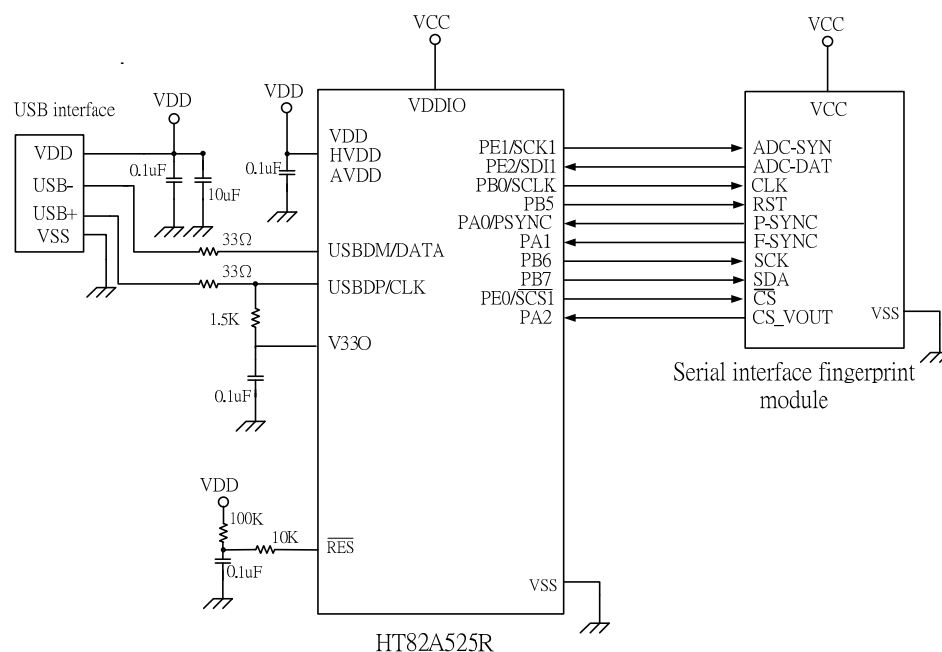
简介

本文介绍如何运用 HT82A525R 控制指纹识别模块及如何使用 HT82A525R DMA 接口快速的接收指纹识别模块产生的大量数据, HT82A525R 包含了 SPI 串行 DMA 接口和 8 bits 并行 DMA 接口。本文将分别举例串行 DMA 接口指纹识别模块和并行 DMA 接口指纹识别模块的运用, 着重考虑如何控制 DMA 接口并且利用 USB FIFO 传送至 USB 接口。

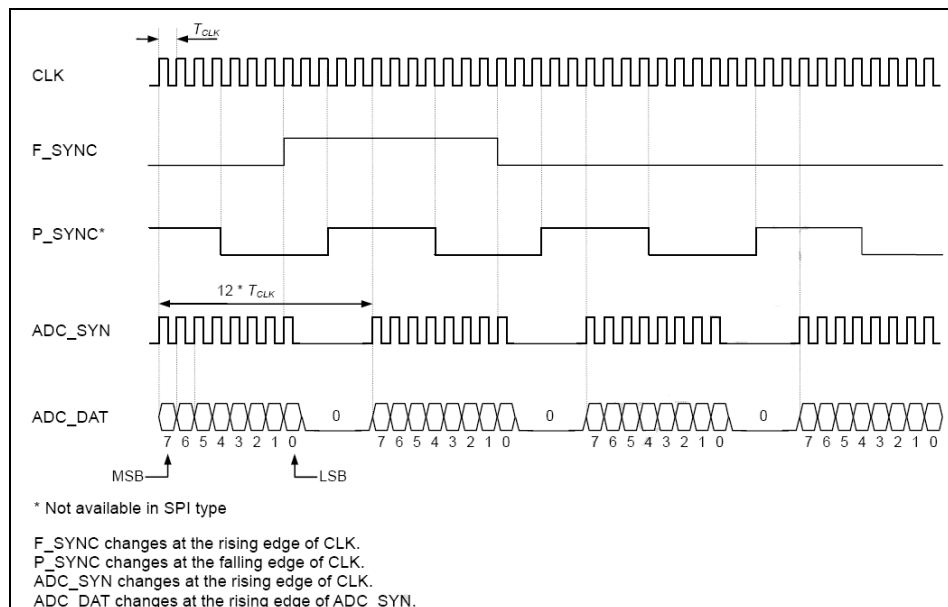
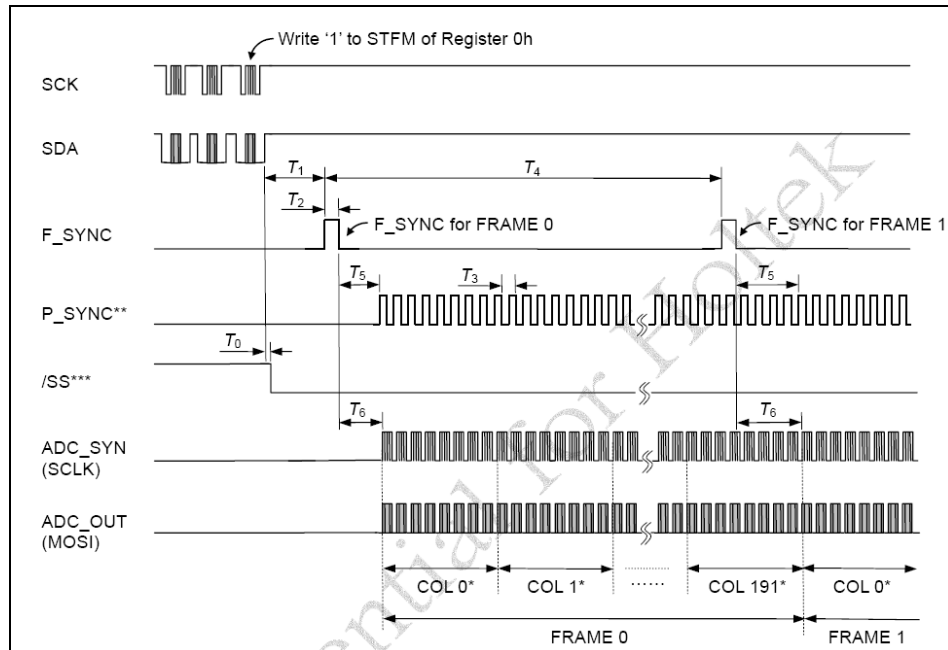
串行DMA接口指纹识别模块应用

在串行 DMA 接口指纹识别模块应用中, 使用祥群科技 LTT-SUPER S500 模块做为应用范例。

应用电路



应用时序图



范例程序

```

=====
// File name: main.asm
// [option]
// PB0/SCK      =      SCLK
// EP1          =      Enable
// EP2          =      Enable
// EP3          =      Enable
// SCLK         =      12Mhz
// SPI1_EN     =      Enable
// SPI1_Mode   =      mode=1
// WDT         =      Disable
// PA0/PSYNC,VSYNC = PSYNC or VSYNC
// PA5/PCLK    =      PCLK
// PA4/HSYNC   =      HSYNC
// 作者: Yen
// 说明: 该程序是主程序,主要为做一些初始设定。
//       该程序使用 LIT-SS500A sensor
=====
// 定义一些控制引脚
=====
#define _CLK      [39h]          ; CLK Register
#define _SCLKEN  _CLK.0 ; PDMA 0: disable; 1: enable
#define _CLKAUTOB _CLK.1      ; SCLK halt 0: when FIFO3 is full or SCLKEN is 0
                                ; 1: when SCLKEN is 0

#define SDMAEN   MISC.3          ; SBDMA DMA control bit 0: disable; 1: enable
#define SDMASEL MISC.4          ; Serial DMA interface selection bit
                                ; 0: Serial interface 1 (pin-shared with port E)
                                ; 1: Serial interface 2 (pin-shared with port C)

#define SCLK_OUT PBC.0
#define SW_RST_CTL PBC.5

#define CHIP_EN PE.3
#define CHIP_EN_CTL PEC.3

#define I2C_SD_CTL PBC.7
#define I2C_SC_CTL PBC.6
#define I2C_SD PB.7
#define I2C_SC PB.6

=====
// 设定 USB Status
=====
Set on_usb_clk          ; enable usb clock
Set set_v33O            ; Turn on V33O
clr ucc.6               ; 1=6M, 0=12M
clr USB_EP2_IO          ; Set EP2 OUTPUT FIFO
set USB_EP3_IO          ; Set EP3 INPUT FIFO

set SDMAEN

=====
// 初始化控制信号
=====

clr SCLK_OUT

clr SW_RST_CTL
    
```

```

clr  I2C_SD_CTL
clr  I2C_SC_CTL
SET  CS_VOUT_CTL

clr  SW_RST
clr  I2C_SD
clr  I2C_SC
call wait_28us
set  SW_RST

SET  SBCR.@M0      ;M0
SET  SBCR.@M1      ;M1;slave
SET  SBCR.@MLS     ;set, msb
clr  SBCR.@SBEN

;=====
;//  设定 MCU 数据抓取方式
;=====

CLR  _CLKAUTOB
CLR  SDMAEN
CLR  SDMASEL
set  _SCLKEN

;=====
;//  设定 LTT-SS500
;=====
Call  LTT-SS500_Init

;=====
;//  interrupt enable
;=====
Set  intc0.0      ; emi
Set  intc1.0      ; usb

MainLoop:
sz   suspend_line ; USC.0=1, USB enter suspend mode
jmp  Tosuspend    ; 处理 suspend
jmp  MainLoop

;=====

;=====
;//  File name: usb_int.asm
;//  说明: 本 asm 文件为处理 USB interrupt 数据
;=====
;=====
;//  处理 endpoint 1 数据
;=====

AccessFIFO1:
Sz   nCmdIndex    ; Reset to 0
Jmp  ExitExt1     ;

CALLFIFO1_Rd_Check ; FIFO2 ready?
Clr  MISC.@USB_Req ;
SZ   Z            ;
jmp  ExitExt1     ;
out_fifo1_data:
CALLFIFO1_Rd_Check ; FIFO2 ready?
CALLReadFIFO1     ;
;=====
    
```

```

// Enable Serial DMA
//=====
set   SDMAEN           ;start SDMA
set   SBCR.@SBEN      ;start SPI
ExitExt1:
Clr   USR.@int1       ;
jmp   Exit_of_Ext_start ;

//=====
// 处理 endpoint 2 数据
//=====
AccessFIFO2:
ExitExt2:
clr   USR.@int2       ;
jmp   Exit_of_Ext_start ;
//=====
// 处理 endpoint 3 数据
//=====
AccessFIFO3:
ExitExt3:
Clr   USR.@int3       ;
Jmp   Exit_of_Ext_start ;
    
```

程序说明

本范例主要是示范串行 DMA 的使用；使用 LTT-SS500，配合 HT82A525R USB MCU，完成取像并作指纹识别的一个应用。

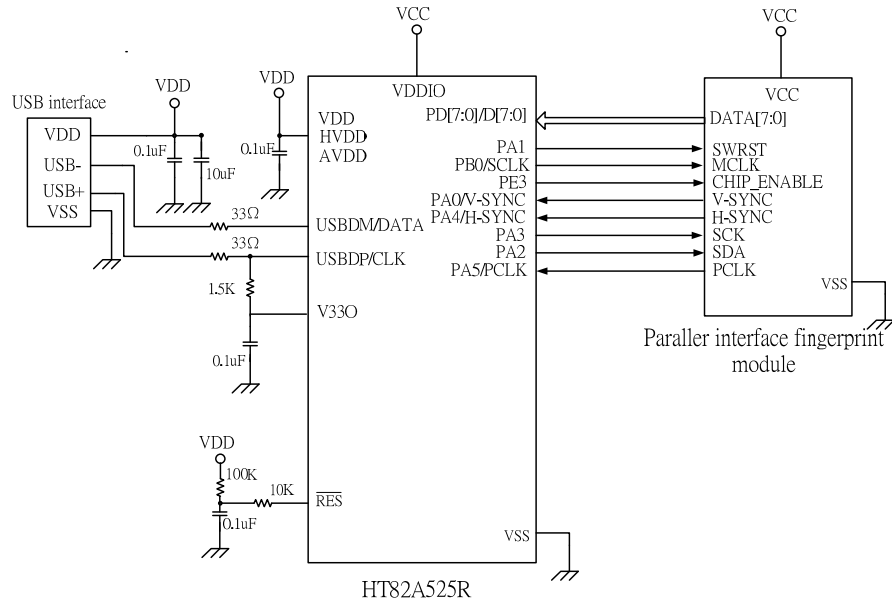
MCU 负责控制 Sensor，并利用第 1 组 SPI (PE0~PE4) 从 LTT-SS500 接收影像数据，通过 DMA 的方式，由 USB ENDPOINT 3 以 BULK 传输方式送到 PC 端，PC 端需自行开发 Vendor Class Driver，采用 BULK 传输方式接收数据。

程序一开始先设定相关寄存器，含 SPI 控制，因为 LTT-SS500 的 SPI 是固定为 Master 端，所以必须将 SPI 设定成 Slave，之后再初始化 Sensor，以 I²C 方式设定 Sensor，等待数据开始传送。在取影像前，本程序通过 PC 端对 ENDPOINT 1 下指令，先将 SDMA 使能及将 SCS 使能，打开串行 DMA 功能，开始接收传送数据，此时 ENDPOINT 3 为 SDMA 模式，所以不用再手动处理数据，数据会以 DMA 的方式传送给 PC 端。

并行DMA接口指纹识别模块应用电路

在并行 DMA 接口指纹识别模块应用中，使用 Hynix YACBAA0SDDAS 模块做为应用范例。

应用电路



应用时序图



范例程序

```

//=====
// File name: main.asm
// PB0/SCK      = SCLK
// EP1          = Enable
// EP2          = Enable
// EP3          = Enable
// SCLK         = 12Mhz
// SPI1_EN     = Enable
// SPI1_Mode   = mode=1
// WDT         = Disable
// PA0/PSYNC,VSYNC = PSYNC or VSYNC
// PA5/PCLK    = PCLK
// PCLK        = Falling Edge
// PA4/HSYNC   = HSYNC

// 作者: Yen
// 说明: 该程序是主程序,主要为做一些初始设定。
// 该程序使用 Hynix 702 cmos sensor
    
```

```

//=====
// 定义一些控制引脚
//=====
#define _CLK [39h] ;CLK Register
#define _SCLKEN_CLK.0 ;PDMA 0: disable; 1: enable
#define _CLKAUTOB_CLK.1 ;SCLK halt 0: when FIFO3 is full or SCLKEN is 0
; 1: when SCLKEN is 0

#define _PDMA_CLK.2 ; PDMA 0: disable; 1: enable
#define _PDATA_SEL_CLK.3; PDATA_SEL 0: odd byte 1: got all
#define _PDMA_MOD_CLK.4 ; PDMA_MOD 0: finger-print 1: normal pdma
#define _FIG_PIX_CLK.5 ; FIG_PIX 0: QVGA(320x240) 1: CIF(352X288)

#define SCLK_OUT PBC.0
#define I2C_SD_CTL PAC.2
#define I2C_SC_CTL PAC.3
#define I2C_SD PA.2
#define I2C_SC PA.3

#define SW_RST PA.1
#define CHIP_EN PE.3
#define SW_RST_CTL PAC.1
#define CHIP_EN_CTL PEC.3

//=====
// 设定 USB Status
//=====
set on_usb_clk ; enable usb clock
set set_v330 ; Turn on V330
clr ucc.6 ; 1=6M, 0=12M
clr USB_EP1_IO ; Set EP1 OUTPUT FIFO
set USB_EP2_IO ; Set EP2 INPUT FIFO
set USB_EP3_IO ; Set EP3 INPUT FIFO

//=====
// 初始化控制信号
//=====
clr SCLK_OUT
cClr CHIP_EN_CTL
clr SW_RST_CTL
clr I2C_SD_CTL
clr I2C_SC_CTL
clr I2C_SD
set I2C_SC
clr CHIP_EN_CTL
clr SW_RST_CTL

//=====
// chip enable
//=====
set CHIP_EN
call Delay4uSec

//=====
// enable MCLK
//=====
set _SCLKEN ; SCLKEN 0: SCLK disable 1: SCLK out enable
set _CLKAUTOB ; CLKAUTOB 0: stop by FIFO FULL 1: stop by SCLKEN
call Delay300mSec
    
```

```

//=====
//  RESET Signal is High
//=====
set  SW_RST
call Delay4uSec
//=====
//  设定 MCU 数据抓取方式
//=====
clr  _PDATA_SEL      ; 只取奇数笔数据
clr  _PDMA_MOD       ; 并列 DMA:指纹识别模式
clr  _FIG_PIX        ; QVGA(320x240)模式

//=====
//  设定 cmos sensor 值,影像大小,格式,时序等
//=====
call Sensor_Init

//=====
//  interrupt          enable
//=====
set  intc0.0         ; emi
set  intc1.0         ; usb

MainLoop:
sz   suspend_line    ; USC.0=1, USB enter suspend mode
jmp  Tosuspend       ; 处理 suspend
jmp  MainLoop

//=====

//=====
//  File name: usb_int.asm
//  说明: 本 asm 文件为处理 USB interrupt 数据
//=====

//=====
//  处理 endpoint 1 数据
//=====

AccessFIFO1:
Sz   nCmdIndex       ; Reset to 0
Jmp  ExitExt1        ;

CALLFIFO1_Rd_Check ; FIFO2 ready?
Clr  MISC.@USB_Req   ;
SZ   Z               ;
Jmp  ExitExt1        ;
out_fifo1_data:
CALLFIFO1_Rd_Check ; FIFO2 ready?
CALLReadFIFO1       ;
//=====
//  Enable Parallel DMA
//=====
Set  _PDMA
ExitExt1:
Clr  USR.@int1       ;
Jmp  Exit_of_Ext_start ;
    
```

```

//=====
// 处理 endpoint 2 数据
//=====
AccessFIFO2:
ExitExt2:
    Clr    USR.@int2        ;
    Jmp    Exit_of_Ext_start ;
//=====
// 处理 endpoint 3 数据
//=====
AccessFIFO3:
ExitExt3:
    clrSR.@int3        ;
    jmp    xit_of_Ext_start ;
    
```

程序说明

本范例主要是示范并列 DMA 的使用；使用 Hynix CMOS Sensor (HI-702)，配合 HT82A525R USB MCU，完成取像并作指纹识别的一个应用。

MCU 负责控制 Sensor，并利用 PD0~PD7 接收影像数据，通过 DMA 的方式，由 USB ENDPOINT 2 以 BULK 传输方式送到 PC 端，PC 端需自行开发 Vendor Class Driver，采用 BULK 传输方式接收数据。

程序一开始初始化 Sensor，由 I²C 方式控制 Sensor 寄存器设定 Sensor 的模式，如影像数据格式、大小及频率等。因为指纹识别不需使用到彩色的数据，Hi-702 数据格式有 RGB 及 YUV 两种格式，一个像素二个字节，若只接收 Y 或三原色中的一种原色，也可以顺利还原成单色影像，所以为了 MCU 可将 PDATA_SEL 设定为 1，允许只接收奇数笔的数据，加快速度。

另外，为配合市面上常见的影像格式 QVGA 及 CIF，可藉由 FIG_PIX 的设定，设定 MCU 目前所要处理影像为 QVGA 或者 CIF 的格式。若将 PDMA_MOD 设定为 0，则表示为指纹识别模式，若设定为 1，则表示为正常模式。在指纹模式下，MCU 内部会根据 VSYNC 及 HSYNC 的信号并控制 MCLK，自动判断每张影像的开始，如此一来，可确保 MCU 所抓取的数据，每次都从影像的第一笔数据开始抓取，避免发生影像偏差的情形，若为正常模式，则 MCU 会一直收取数据，不会自动判断起始点，适合一般性的数据。

在取影像前，本程序通过 PC 端对 ENDPOINT 1 下指令，先将 PDMA 使能，打开并列 DMA 功能，开始接收传送数据，此时 ENDPOINT 2 为 DMA 模式，所以不用再手动处理数据，数据会以 DMA 的方式传送给 PC 端。

HT82A525R 可藉由 PDATA_SEL、PDMA_MOD、FIG_PIX 设定，适用于大部份的并列 DMA 应用。

版本记录

版本: V1.10

修改人员: 谢琳

修改日期: 2011 年 12 月 16 日

修改内容:

- 应用电路图中 USBD-引脚与 USBDM/DATA 引脚间、USB D+引脚与 USB DP/CLK 引脚间多加了一个 33Ω 电阻, $\overline{\text{RES}}$ 引脚与 0.1μF 电容间多加了一个 10KΩ 电阻
- 电路图中 USB D- / USB D+ 改为 USB DM / USB DP, RES 改为 $\overline{\text{RES}}$