使用 M258 驅動 STK3420 感測器

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例代碼使用 M258 透過 I²C 驅動 STK3420 感測器來演示手勢的感測功能。
BSP 版本	M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.00.005
開發平台	NuMaker-M258KE V1.2

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. 概述

隨著非接觸控制裝置、互動式科技的普及，光線感測、距離感測和手勢感測技術在微控制器應用領域逐漸扮演重要角色。光感測能夠偵測光線強度，用於環境監測、照明系統等；距離感測技術可精確測量物體與感測器之間的距離，能廣泛運用於自動化、機器人技術領域；手勢感測則讓裝置能夠識別人類手勢，可應用於互動式科技、智慧家庭裝置等。

本範例程式使用 Sensortek 昇佳電子的環境光與近接感測晶片(Proximity with Ambient Light Sensor) - STK3420，說明如何使用 I²C 取得此感測器的手勢數值。

1.1 原理

STK3420 感測器內部整合光電二極體、ADC，內建 940nm 紅外線 LED 並使用噪音消除解決方案，避免環境中的紅外線干擾，STK3420 內部方塊圖可參考圖 1-1。其手勢識別的工作原理為由內部的光電二極管區分物體的移動，配合軟體上的演算法計算，得到正確的 3D 手勢(上/下/左/右/遠/近)。此範例程式僅判斷手勢的上/下/左/右。

對外則以 I²C 介面連接，STK3420 的位址設定為 7-bit 值為 0x58，其餘所有操作皆透過指令控制/存取暫存器，請參考圖 1-2 暫存器總表，此範例程式設置之暫存器與數值如表 1-1 描述，其他詳細設置請參考 STK3420 Datasheet。

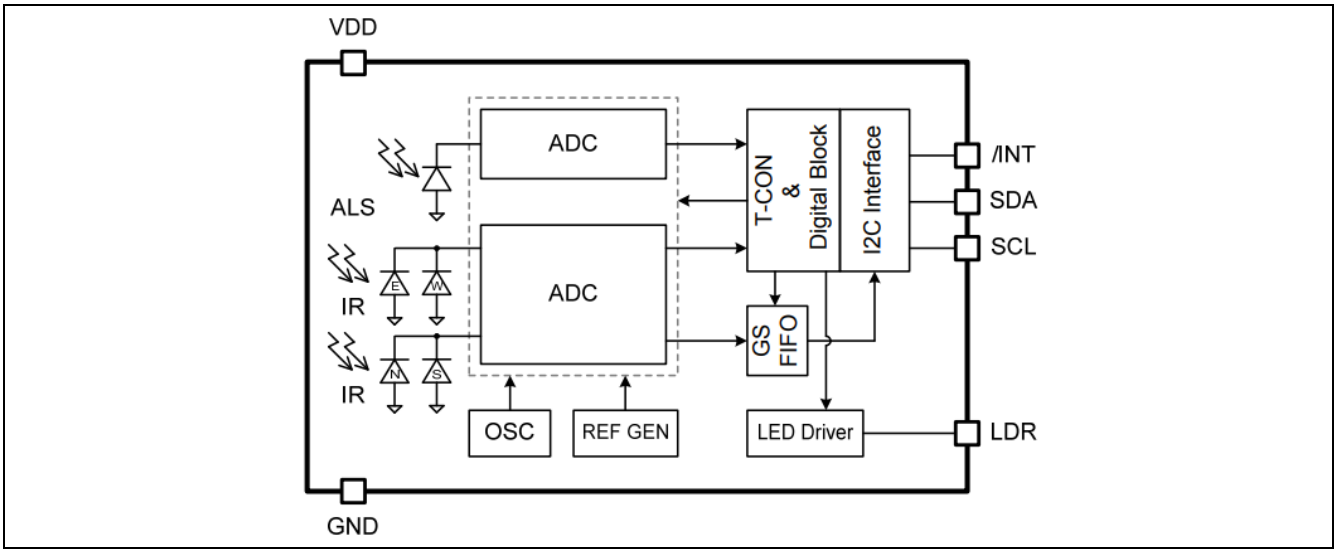

圖 1-1 Block Diagram of STK3420

ADDR	REG NAME	BIT								Default
		7	6	5	4	3	2	1	0	
0x00	STATE	EN_IRS		EN_ALS_RO	EN_PS_RO	EN_WAIT_ALS	EN_WAIT_PSGS	EN_ALS	EN_PS	0x00
0x01	PS/GSCTRL1	PRST_PS[1:0]				IT_PSGS[3:0]				0x01
0x02	ALSCTRL1	PRST_ALS[1:0]		GAIN_ALS[1:0]		IT_ALS[3:0]				0x34
0x03	LEDCTRL	IRDR_LED[1:0]								0x40
0x04	INT	INT_CTRL			EN_GS_INT	EN_ALS_INT		PS_MODE	EN_PS_INT	0x00
0x05	WAIT1_PSGS	WAIT1_PSGS[7:0]								0x00
0x06	THDH1_PS	THDH_PS[15:8]								0xFF
0x07	THDH2_PS	THDH_PS[7:0]								0xFF
0x08	THDL1_PS	THDL_PS[15:8]								0x00
0x09	THDL2_PS	THDL_PS[7:0]								0x00
0x0A	THDH1_ALS	THDH_ALS[15:8]								0xFF
0x0B	THDH2_ALS	THDH_ALS[7:0]								0xFF
0x0C	THDL1_ALS	THDL_ALS[15:8]								0x00
0x0D	THDL2_ALS	THDL_ALS[7:0]								0x00
0x10	FLAG	FLG_ALS_DR	FLG_PS_DR	FLG_ALS_INT	FLG_PS_INT	FLG_GS_INT		FLG_IRS_DR	FLG_NF	0x01
0x11	DATA1_PS	DATA_PS[15:8]								
0x12	DATA2_PS	DATA_PS[7:0]								
0x13	DATA1_ALS	DATA_ALS[15:8]								
0x14	DATA2_ALS	DATA_ALS[7:0]								
0x17	DATA1_IRS	DATA_IRS[15:8]								
0x18	DATA2_IRS	DATA_ IRS[7:0]								
0x19	ALSCTRL2	ITER_IRS[1:0]		GAIN_IRS[1:0]		IT_IRS[3:0]				0x74
0x1B	WAIT_ALS	WAIT_ALS[7:0]								0x00
0x1C	WAIT2_PS		WAIT2_PS[6:0]							0x00
0x1D	PS/GSCTRL2		EN_INTELLIGENT_PRST					EN_GSNS	EN_GSEW	0xF0
0x1E	GSFLAG	FLG_GS_FIFO_OV			GS_FIFO_LEN[4:0]					
0x1F	GSFIFOCTRL					GS_FIFO_THD[3:0]				0x08
0x20	DATA1_GSE	DATA_GSE[15:8]								
0x21	DATA2_GSE	DATA_GSE[7:0]								
0x22	DATA1_GSW	DATA_GSW[15:8]								
0x23	DATA2_GSW	DATA_GSW[7:0]								
0x24	DATA1_GSN	DATA_GSN[15:8]								
0x25	DATA2_GSN	DATA_GSN[7:0]								
0x26	DATA1_GSS	DATA_GSS[15:8]								
0x27	DATA2_GSS	DATA_GSS[7:0]								
0x3E	PDT_ID	PDT_ID[7:0]								0x50
0x3F	Reserved	Reserved								
0x80	SOFT_RESET	Write any to soft reset								
0x94	MISC1						LED_DIV2			0xC3

圖 1-2 STK3420 暫存器總表

Reg Addr	Reg Name	Value	Description
0x1d	PS/GSCTRL2	0xf3	Enable EWNS gesture detection and PS intelligent persistence.
0x5	WAIT1_PSGS	0x6	Set waiting state period. (6+1)*780 us = 5.46 ms
0x1	PS/GSCTRL1	0x2	Set gesture refreshing time to 390 us, and PS persistence times to 1.
0x95	Analog Gain	0x90	To perform gesture well, the analog gain should be set to 2 times, and the register value is 0x90. Note: This register is not mentioned in Datasheet.
0x3	LEDCTRL	0xc0	Set LED current sink to 150 mA.
0x94	MISC1	0x35	Set LED current control (LED_DIV2) to 1.
0x2	ALSCTRL1	0x2	Set ALS persistence times to 1, ALS gain to 1, and ALS refreshing time to 50 ms.
0x0	STATE	0x4	Enable PS/GS waiting state.

表 1-1 暫存器設置與描述

STK3420 以一組 128 bytes FIFO 存放 16 組手勢數值。每組手勢數據是在每一個 ADC 轉換週期中由 E/W/N/S 的輸出資料組成。如圖 1-3 所示，此 FIFO 數據可透過讀取 DATA_GSE/W/N/S 暫存器取得，取得後的數值經由 Sensortek 提供的 library 計算，即可得到正確的手勢判斷。

ADDR	REG NAME	BIT								Default
		7	6	5	4	3	2	1	0	
0x20	DATA1_GSE	DATA_GSE[15:8]								
0x21	DATA2_GSE	DATA_GSE[7:0]								
0x22	DATA1_GSW	DATA_GSW[15:8]								
0x23	DATA2_GSW	DATA_GSW[7:0]								
0x24	DATA1_GSN	DATA_GSN[15:8]								
0x25	DATA2_GSN	DATA_GSN[7:0]								
0x26	DATA1_GSS	DATA_GSS[15:8]								
0x27	DATA2_GSS	DATA_GSS[7:0]								

圖 1-3 手勢數值暫存器

注：此文件圖片來源取自STK3420 Datasheet。

1.2 執行結果

本範例代碼執行編譯和燒錄的結果，輸出後如圖 1-4 所示。

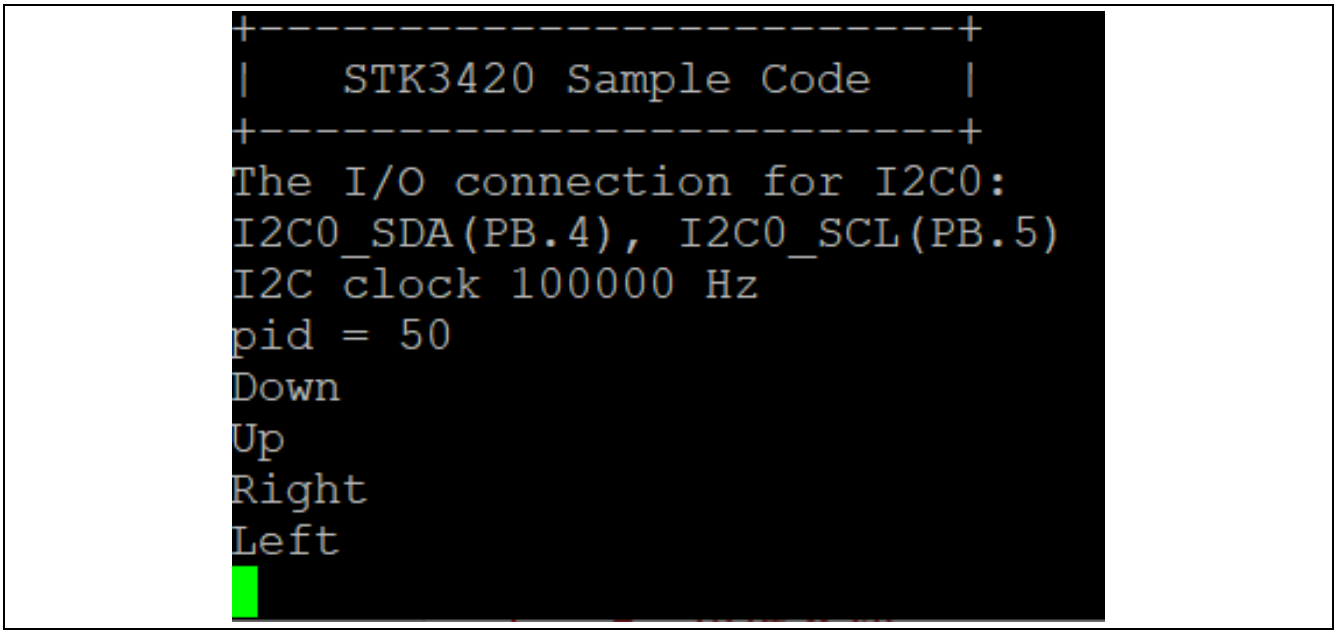

圖 1-4 執行編譯和燒錄結果

2. 代碼介紹

主程式由 *main.c* 進入，包括以下 4 部分：

- 系統時鐘與引腳初始化：SYS_Init
- 介面初始化與開啟：UART0_Init, I2C0_Init
- STK3420 暫存器設置：STK3420_start
- STK3420 手勢運算：讀取 DATA_GSE/W/N/S 暫存器、STK_Gesture.lib

STK3420 暫存器設置與描述說明，請參考表 1-1 與 STK3420 Datasheet 設置。

```
void STK3420_start(void)
{
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x80, 0x00);
    CLK_SysTickDelay(20000);   // wait STK3420 IC Reset

    I2C_ReadByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x3E, &PID);
    printf("pid = %x\n", PID);
    //enable EWNS gesture detection and PS intelligent persistence
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x1d, 0xf3);
    //set wait state period (6+1)*780 us = 5.46 ms
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x5,  0x6);
    //set gesture refresh time to 390 us, PS persistence setting to 1 times
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x1,  0x2);
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x95, 0x90);//set analog gain
    //set LED current sink to 150mA
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x3,  0xc0);
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x94, 0x35);//set LED_DIV2 to 1
    /*ALS setting: set ALS persistence to 1 times, ALS gain to 1,
    and ALS refresh time to 50 ms*/
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x2,  0x2);
    I2C_WriteByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x0,  0x4);//enable PS/GS wait state
    memset(&xtGes, 0, sizeof(tfGesture));
    Gesture_Init(&xtGes);
}
```

經過系統初始化與感測器開啟設定後，進入迴圈讀取存放在 FIFO 中的 E/W/N/S 數值，再透過 *Stk_Gesture.lib* 內的演算法計算後，得到正確的手勢方向。

```
int32_t main(void)
{
    int i = 0;
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Init System, IP clock and multi-function I/O */
    SYS_Init();
```

```
    /* Init UART0 for printf */
    UART0_Init();

    /* Lock protected registers */
    SYS_LockReg();

    printf("+-------------------------+\n");
    printf("|   STK3420 Sample Code   |\n");
    printf("+-------------------------+\n");

    printf("The I/O connection for I2C0:\n");
    printf("I2C0_SDA(PB.4), I2C0_SCL(PB.5)\n");

    /* Init I2C0 */
    I2C0_Init();

    STK3420_start();

    while(1) {
        uint8_t fifo_cnt = 0;
        uint8_t ges_data[8];
        uint16_t gse = 0;
        uint16_t gsw = 0;
        uint16_t gsn = 0;
        uint16_t gss = 0;

        // Read FIFO Frame Cnt
        I2C_ReadByteOneReg(I2C0, STK342X_SLAVE_ADDR, 0x1E, &fifo_cnt);

        if(fifo_cnt) {
            for(i = 0; i < (fifo_cnt & 0x1F); i++) {
                // Read FIFO Data
                I2C_ReadMultiBytesOneReg(I2C0, STK342X_SLAVE_ADDR, 0x20, ges_data, 8);
                gse = (ges_data[0] << 8 | ges_data[1]);
                gsw = (ges_data[2] << 8 | ges_data[3]);
                gsn = (ges_data[4] << 8 | ges_data[5]);
                gss = (ges_data[6] << 8 | ges_data[7]);
                int act = iGesture_Do(&xtGes, gse, gsw, gss, gsn);
                switch(act) {
                    case GESTURE_EVENT_UP:
                        printf("Up\n");
                        break;
                    case GESTURE_EVENT_DOWN:
                        printf("Down\n");
                        break;
                    case GESTURE_EVENT_LEFT:
                        printf("Left\n");
                        break;
                    case GESTURE_EVENT_RIGHT:
                        printf("Right\n");
                        break;
                    default:
                        break;
                }
            }
        }
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.00.005
- IDE 版本
  - Keil uVersion 5.38

3.2 硬體需求

- 電路元件
  - NuMaker-M258KE V1.2
- 線路示意圖
  - 請參考圖 3-1 與表 3-1 的說明與線路圖

STK3420	Description	M258KE3AE
VDD	Power supply for 2.4V to 3.6V.	VDD
SDA	Data line for I²C interface.	I²C0_SDA Pin 2 (PB.4)
SCL	Serial clock for I²C interface.	I²C0_SCL Pin 1 (PB.5)
LEDA	The anode of the embedded IR LED, connect to power.	VDD
GND	Ground.	VSS

表 3-1 腳位說明與 MCU 腳位對照表

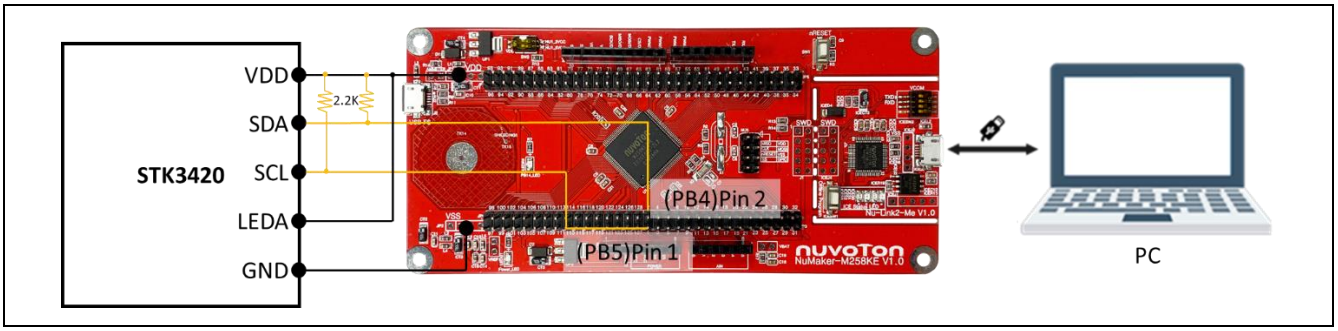

圖 3-1 線路連接示意圖

4. 目錄資訊

EC_M258_Gesture_Sensor_STK3420_V1.00	
Library	Sample code header and source files
CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
Device	CMSIS compliant device header file
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	
Project	Source file of example code

圖 4-1 目錄資訊

5. 範例程式執行

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 *M258_Gesture_Sensor_STK3420.uvprojx*。
2. 進入編譯模式介面
   - 編譯
   - 下載代碼至記憶體
   - 進入 / 離開除錯模式
3. 進入除錯模式介面
   - 執行代碼

6. 修訂紀錄

Date	Revision	Description
2023.12.01	1.00	初始發布。

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.