4線電阻式觸控螢幕控制

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	本範例代碼基於NUC126平台，實現4線電阻式觸控螢幕控制
BSP 版本	NUC126Series_BSP_CMSIS_v3.00.003
開發平台	NuTiny-NUC126

1 功能介紹

1.1 簡介

目前市面上的電阻式觸控螢幕，多分成四線、五線、七線或八線等架構，本範例程式以最常見的四線式ADC架構作說明與實現。
在觸控螢幕上，螢幕左邊至最右邊(X方向)可視為一個長條電阻，螢幕上面至最下面(Y方向)可視為另一個長條電阻，當按壓觸控螢幕的時候，可由兩組ADC分別採樣X方向與Y方向得到兩個電壓值，並依照比例計算目前的觸控點。
舉例來說，觸控螢幕解析度為320x240，由於NUC126的ADC有12-bit的解析度，因此5V可以被切成4096個等分。但電阻式螢幕並不是理想的，在螢幕最左邊X座標應為0的位置，X座標的ADC讀數可能會是500多而不是0；螢幕最右邊X座標應為320的位置，X座標讀數可能僅有3200而不是4095，因此必須將範圍做正規化處理。
舉個例子，假如觸控後，ADC採樣回來的數值為(2048, 1024)，請問對應320*240屏幕上的哪個座標點？

1. 先將 ADC 範圍做正規化處理：
   設ADC讀數500時為X座標0點，ADC讀數為3200時為X座標320點，因此能夠將320個座標點切成3200-500 = 2700個等分。
2. 計算 X 的座標位置
   320: 2700 = X 座標 : 2048 => X 座標 = 177.124
3. Y 座標位置同 X 座標的計算方式

1.2 執行結果


串口配置
端口: COM69
波特率: 115200
數據位: 8
停止位: 1
校驗: NONE
關閉串口
線路控制
DTR
BREAK
RTS
線路狀態（只讀）
CTS
DSR
RING
RLSD
接收區：已接收3449位元組，速度0位元組/秒，接收狀態[允許]，輸出文本狀態[已停止]
(0, 16)
(1, 17)
(6, 20)
(13, 26)
(19, 32)
(25, 39)
(33, 48)
(40, 57)
(47, 64)
(53, 72)
(60, 79)
(67, 88)
(74, 95)
(80, 103)
(84, 109)
(90, 114)
(98, 121)
(105, 128)
(112, 134)
(118, 142)
(124, 147)
(127, 153)

2 代碼介紹

2.1 Main 函數

在Main函數中，將會做系統初始化、UART初始化以及觸控螢幕的初始化。然後在while迴圈中每秒回報一次觸控點。

```
/*---------------------------------------------------------------------------*/
/*  Main Function                                                            */
/*---------------------------------------------------------------------------*/
int32_t main(void)
{
    uint32_t u32TouchX, u32TouchY;
    SYS_Init();
    UART0_Init();

    // Init Touch Panel HW
    Drv_NuTFT_HWInit();

    // Touch Test
    while (1)
    {
        if (App_GetTouchPoint(&u32TouchX, &u32TouchY) == Get_Success)
            printf("(%d, %d) \n", u32TouchX, u32TouchY);
        CLK_SysTickDelay(1000000);
    }
}
```

2.2 獲得 X 軸方向的觸控點

由於這是電阻式的觸控螢幕，當感測X軸方向的觸控點時，需要將Y方向的兩條線設置為輸入模式，並將X方向的兩條線(XR, XL)設置為輸出模式。令XR輸出高邏輯、XL輸出低邏輯，並使用ADC採樣YU線上的電壓，即可獲得X方向上的觸控座標。

```
static uint16_t _Api_Get_TP_X(void)
{
    uint32_t u32_ADC_Convert_Result_X_Axis;
    uint16_t u16_nNormalizationFactor;

    /* Init ADC for TP */
```

```
    /* Set input mode as single-end and enable the A/D converter */
    ADC_Open(ADC, ADC_ADCR_DIFFEN_SINGLE_END, ADC_ADCR_ADMD_SINGLE, BIT6);

    /* Get X from ADC input */
    GPIO_SetMode(PE, BIT2, GPIO_MODE_OUTPUT);   // XR
    GPIO_SetMode(PB, BIT11, GPIO_MODE_INPUT);   // YD
    GPIO_SetMode(PB, BIT10, GPIO_MODE_OUTPUT);  // XL
    PE2 = 1; //XR High
    PB10 = 0; //XL Low

    /* Configure the GPB9 ADC analog input pins.  */
    SYS->GPE_MFPL &= ~(SYS_GPE_MFPL_PE2MFP_Msk);    // Disable ADC CH9
    SYS->GPB_MFPH &= ~(SYS_GPB_MFPH_PB9MFP_Msk);    // Enable ADC CH6
    SYS->GPB_MFPH |= SYS_GPB_MFPH_PB9MFP_ADC0_CH6;  //YU sample

    /* Disable the GPB8 digital input path to avoid the leakage current. */
    GPIO_DISABLE_DIGITAL_PATH(PB, BIT9);            //YU

    /* Power on ADC module */
    ADC_POWER_ON(ADC);

    /* Enable the sample module 1 interrupt.  */
    ADC_EnableInt(ADC, ADC_ADF_INT); //Enable sample module A/D ADINT1 interrupt.
    NVIC_EnableIRQ(ADC_IRQn);
    /* Clear the A/D ADINT1 interrupt flag for safe */
    ADC_CLR_INT_FLAG(ADC, ADC_ADF_INT);

    /* Reset the ADC interrupt indicator and trigger sample module 1 to start A/D
conversion */
    g_u32AdcIntFlag_TP = 0;
    ADC_START_CONV(ADC);

    /* Wait ADC interrupt (g_u32AdcIntFlag_TP will be set at IRQ_Handler function) */
    while (g_u32AdcIntFlag_TP == 0);

    /* Get x-axis ADC convert result */
    u32_ADC_Convert_Result_X_Axis = ADC_GET_CONVERSION_DATA(ADC, 6);

    /* Close ADC module */
    ADC_Close(ADC);

    //Normalization
```

```
    if (u32_ADC_Convert_Result_X_Axis <= ReportThreshold)
    {
        u32_ADC_Convert_Result_X_Axis -= Init_MinBorderCaliValue_X;
        u16_nNormalizationFactor = (Init_MaxBorderCaliValue_X - Init_MinBorderCaliValue_X)
* 1000 / LCD_Resolution_X;
        u32_ADC_Convert_Result_X_Axis *= 1000;
        u32_ADC_Convert_Result_X_Axis /= u16_nNormalizationFactor;
    }
    return u32_ADC_Convert_Result_X_Axis;
}
```

3 軟體與硬體環境

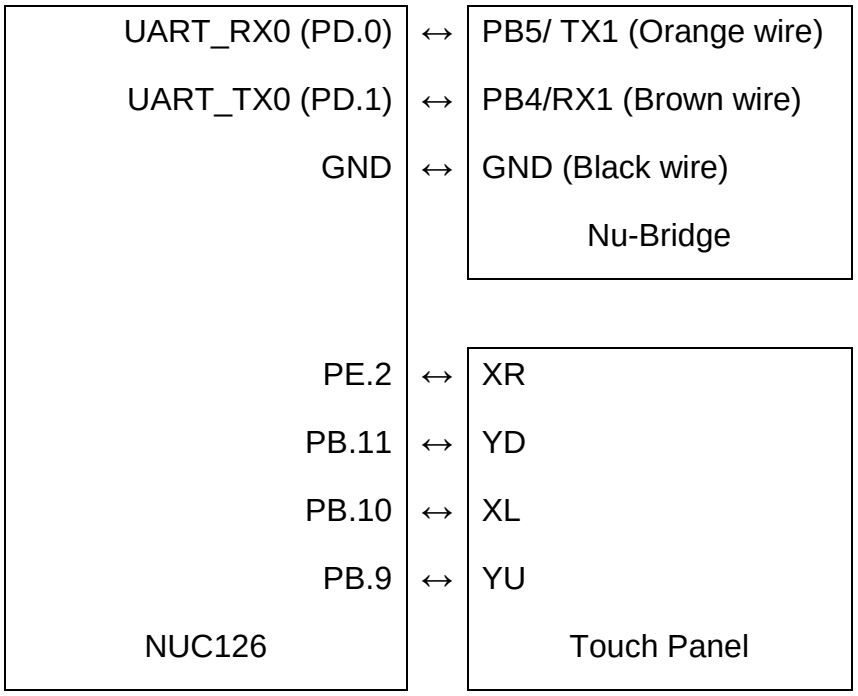
● 軟體環境

- ■ BSP 版本
  - ◆ NUC126Series_BSP_CMSIS_v3.00.003
- ■ IDE 版本
  - ◆ Keil uVersion 5.26

● 硬體環境

- ■ 電路元件
  - ◆ NuTiny-NUC126
  - ◆ Nu-Bridge
  - ◆ NuTFT v1.3 Touch Panel
- ■ 接線圖

使用者可以將 NUC126 上的 UART_RX0(PD.0), UART_TX0(PD.1)接到 Nu-Bridge 上打印 UART 的除錯訊息。使用者需要先打開 PC 上的裝置管理員，查看“NuBridge Virtual Com Port (COMX)”，並打開終端機，將 Baudrate 設定為 115200 接收對應的 Com Port 訊息。相關的觸控螢幕、Nu-Bridge 與 NUC126 的連接方式請參考下圖。


UART_RX0 (PD.0)
UART_TX0 (PD.1)
GND
NUC126
↔
↔
↔
PB5/ TX1 (Orange wire)
PB4/RX1 (Brown wire)
GND (Black wire)
Nu-Bridge
PE.2
PB.11
PB.10
PB.9
↔
↔
↔
↔
XR
YD
XL
YU
Touch Panel

4 目錄資訊

- EC_NUC126_4Wired_Resistive_Touch_Panel_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 — Source file of example code

5 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode/4Wired_Resistive_Touch_Panel 路徑中的 KEIL 資料夾，雙擊 4Wired_Resistive_Touch_Panel.uvproj。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
Jun. 24, 2019	1.00	1. 初版

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.