使用 NADC24 與 M254 實現工業電子秤

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

應用簡述	此範例程式是使用 NADC24 和 M254 實現工業電子秤
BSP 版本	M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.005
開發平台	NADC24_M254_Weighing_Scale_V1.0

1. 概述

本範例程式碼使用 NADC24 類比數位轉換器和 M254 單晶片配合不同的稱重感測器，快速開發稱重秤、計數秤、台秤、冷媒秤等應用。

1.1 原理

稱重感測器是由應變片和電橋電路組成的感測器。當感測器受到拉力或壓力時，會產生與力成比例的電壓。使用者可透過 NADC24（24bit ADC）準確讀取電壓，轉換為數值，然後透過 SPI 介面傳輸到 M254 單晶片進行校正和計算，計算目前重量值並顯示在液晶螢幕上。

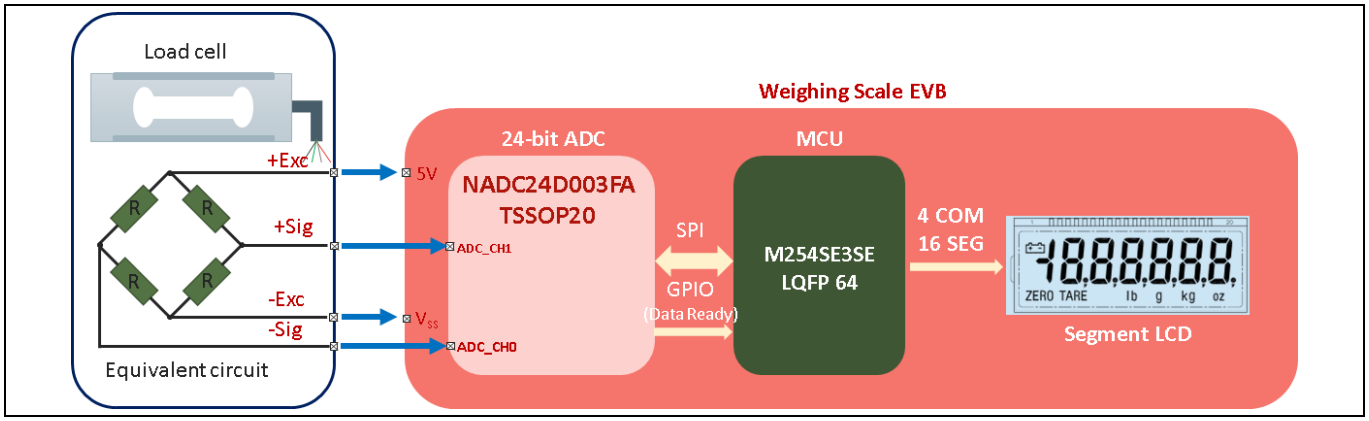


圖 1-1 工業電子秤方塊圖

1.2 NADC24 (24-Bit Delta-Sigma ADC) 概述

NADC24 是一款高精度 24 位元 ΔΣ 類比數位轉換器 (ADC)。它用於讀取稱重感測器的訊號。使用者可設定可程式增益放大器（PGA）、取樣速度、內部參考電壓、數位濾波器（FIR）等功能，以獲得穩定可靠的數據，並為 MCU 提供重量計算。

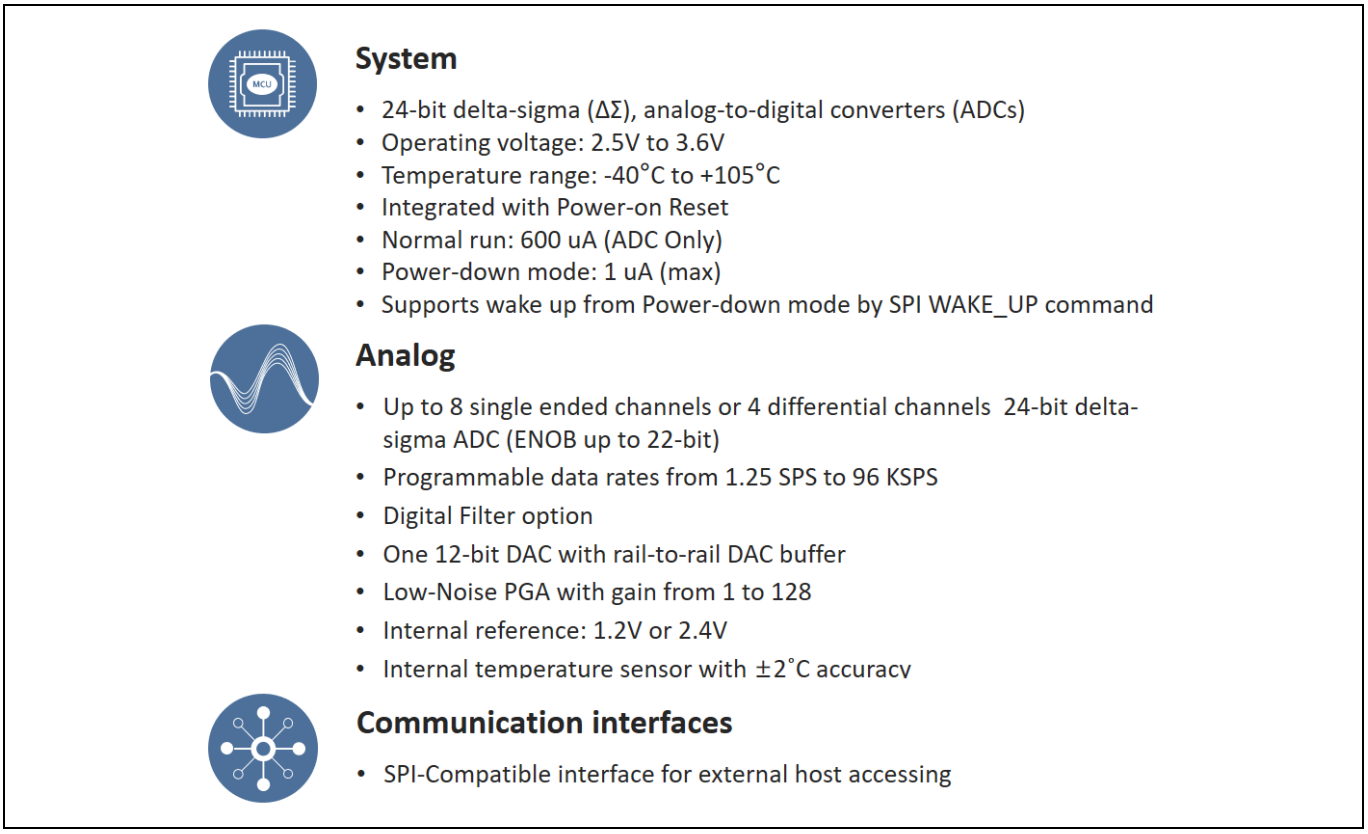
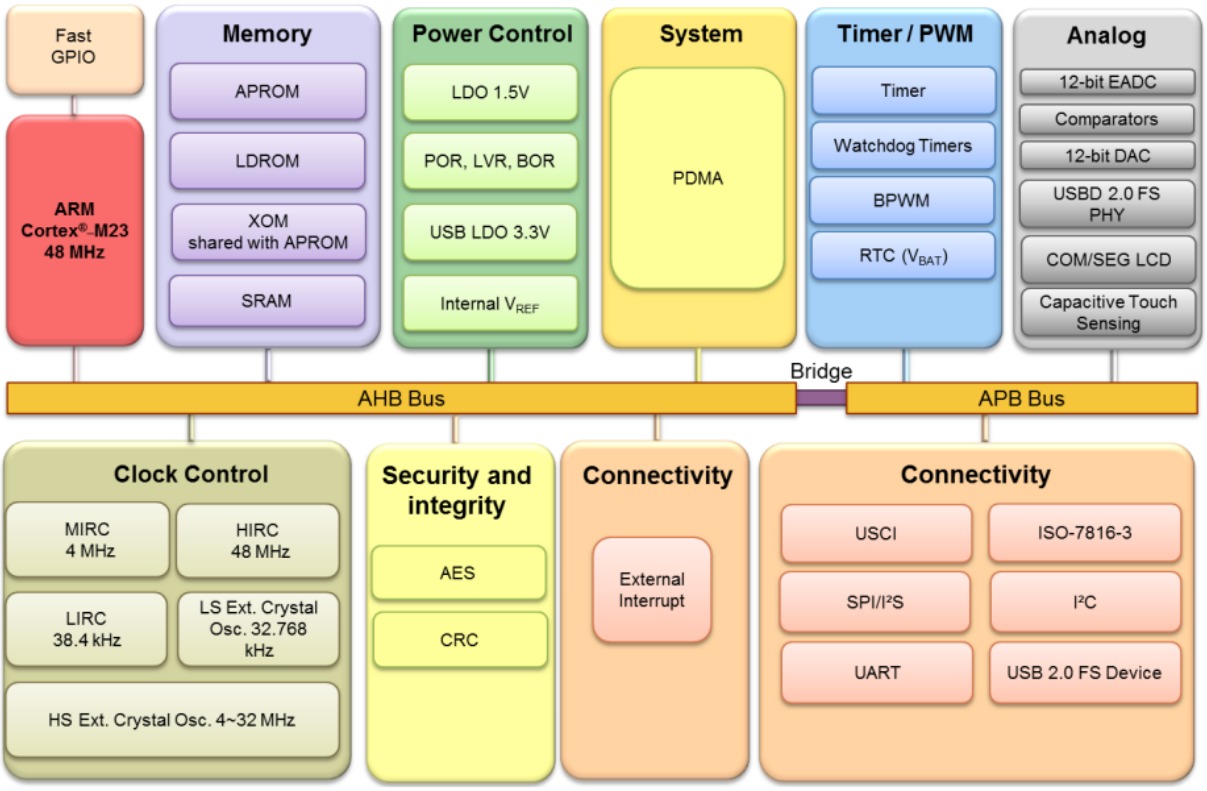


圖 1-2 NADC24 規格

1.3 M254 概述

M254 以 Arm Cortex-M23 為核心，工作頻率 48 MHz，Flash 128KB，SRAM 16KB，COM/SEG LCD Driver 以及 UART、I2C、SPI 等外圍介面。透過 SPI 介面讀取 NADC24 轉換後的數值，進行計算後，就可以透過按壓 **UNIT** 按鍵在段碼 LCD 上依序切換四種不同的重量單位，克（g）、公斤（kg）、磅（lb）和盎司（oz）。



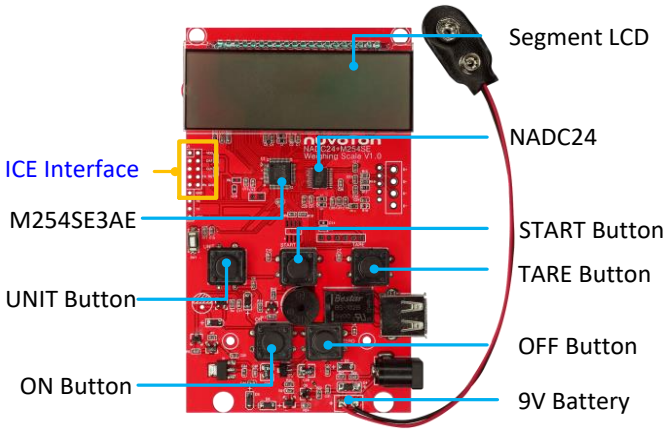
USB* : 只支援 M258 系列

Capacitive Touch Sensing* : 只支援M256/M258系列

圖 1-3 M254 方塊圖

1.4 Evaluation Board 描述

此開發板可以透過按鈕來開啟電源、關閉電源、扣重以及轉換單位。 板子連接 50 公斤重量感測器後，最大測量重量為 50 公斤，解析度為 1 克。 另外，程式會判斷 5 分鐘內重量值沒有變化，則自動關機，實現省電功能。



Key Description:

- ON Button: To turn on the power and start measurement
- OFF Button: To turn off the power
- START Button: To enter calibration mode
- TARE Button: To start calibration second point and reset the weight on the weighing pan to zero
- UNIT Button: To start calibration first point and switch unit of measure on the scale

Specifications:

- Maximum capacity: 50kg, resolution: 1g
- Operating temperature: 5~35 degrees Celsius
- Power supply: 9V alkaline battery
- Automatic shutdown: 5 minutes without weight change

圖 1-4 Evaluation Board 描述

2. 代碼介紹

初始化系統頻率、開機設定、蜂鳴器設定和 LCD 設定。

```
int main(void)
{
    uint32_t i;
    uint32_t u32DataCount;

    /* Unlock protected registers */
    SYS_UnlockReg();

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

    /* Power On */
    GPIO_SetMode(PC, BIT0, GPIO_MODE_OUTPUT);
    GPIO_SetMode(PD, BIT15, GPIO_MODE_OUTPUT);
    PWR_OFF_SIG = 1;
    PWR_SCAN = 1;

    /* Buzzer */
    GPIO_SetMode(PA, BIT8, GPIO_MODE_OUTPUT);
    Beep_20ms();

    /* Init LCD multi-function pins and settings */
    LCD_Init();
    LCD_Start();
    ……
}
```

初始化用於顯示的定時器和用於除錯的 UART。

```
    ……
    /* Set timer to 333 ms */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 3);

    /* Enable timer interrupt and run */
    TIMER_EnableInt(TIMER0);
    NVIC_EnableIRQ(TMR0_IRQn);
    TIMER_Start(TIMER0);

    /* Configure UART0: 115200, 8-bit word, no parity bit, 1 stop bit. */
    UART_Open(UART0, 115200);
    ……
```

初始化 SPI 並設定 NADC24。

```
……
/* Init SPI */
SPI_Init();

/* Initial NADC24 */
Reset_NADC24();
NADC24_Calibration_and_Initial(CALI_NADC24);
NADC24_Calibration_and_Initial(INIT_NADC24);

/* IO Interrupt */
GPIO_SetMode(PF, BIT15, GPIO_MODE_INPUT);
GPIO_EnableInt(PF, 15, GPIO_INT_FALLING);
NVIC_EnableIRQ(GPF_IRQn);
……
```

初始化按鈕並讀取校準資料。

```
……
/* GPIO */
GPIO_SetMode(PC, BIT1, GPIO_MODE_QUASI);    /* BUTTON_START   */
GPIO_SetMode(PC, BIT4, GPIO_MODE_QUASI);    /* BUTTON_UINT    */
GPIO_SetMode(PC, BIT3, GPIO_MODE_QUASI);    /* BUTTON_TARE    */
GPIO_SetMode(PC, BIT5, GPIO_MODE_INPUT);    /* BUTTON_ON      */
GPIO_SetMode(PC, BIT2, GPIO_MODE_QUASI);    /* BUTTON_OFF    */

/* Print */
printf("Nuvoton M254 + NADC24 for weighing scales.\n");
printf("CPU @ %d Hz\n\n", SystemCoreClock);

/* Read the calibration data */
Read_Data_from_APROM(Flash_Data);
Calibration_0g = Flash_Data[0];
printf("Calibration_0g   = %d\n", Calibration_0g);
Calibration_2000g = Flash_Data[1];
printf("Calibration_2000g = %d\n", Calibration_2000g);

/* Start conversion */
SPI_Send_ADC_Commnad(ADC_START_CONVERSION_CMD);

/* Button status */
Button_Start_Status = BUTTON_START;
Button_Uint_Status  = BUTTON_UINT;
Button_Tare_Status  = BUTTON_TARE;
Button_On_Status    = BUTTON_ON;
Button_Off_Status   = BUTTON_OFF;
……
```

偵測按鈕（下降沿）用於開啟電源、關閉電源、皮重、校準和單位轉換。

```
……
while (1)
{
     /* Power off */
    if (Button_Off_Status && !BUTTON_OFF)
    {
        printf("Power Off ...\n");
        PWR_OFF_SIG = 0;
        BUTTON_ON = 1;
        PWR_SCAN = 0;

        /* Buzzer */
        Beep_20ms();
    }

    Button_Off_Status = BUTTON_OFF;

    if (Button_Start_Status && !BUTTON_START)
    {
        Measurement_Flag = 0;
        printf("Enter calibration mode ...\n");
        LCDLIB_Printf(ZONE_SevenSeg_DIGIT, "0     ");
        LCDLIB_SetSymbol(SYMBOL_g,  0);
        Calibration_Mode_Flag = 1;

        /* Buzzer */
        Beep_20ms();
    }

    Button_Start_Status = BUTTON_START;

    /* Calibration mode */
    if (Calibration_Mode_Flag)
    {
        /* Calibration 1P */
        if (Button_Uint_Status && !BUTTON_UINT)
        {
            printf("Calibration 1P ...\n");
            LCDLIB_Printf(ZONE_SevenSeg_DIGIT, "1     ");
            LCDLIB_SetSymbol(SYMBOL_g,  0);
            Calibration_1P_Flag = 1;

            /* Buzzer */
            Beep_20ms();
        }

        Button_Uint_Status = BUTTON_UINT;
        /* Calibration 2P */
        if (Button_Tare_Status && !BUTTON_TARE)
        {
            printf("Calibration 2P ...\n");
            LCDLIB_Printf(ZONE_SevenSeg_DIGIT, "2     ");
            LCDLIB_SetSymbol(SYMBOL_g,  0);
```

```
                Calibration_2P_Flag = 1;

                /* Buzzer */
                Beep_20ms();
            }

            Button_Tare_Status = BUTTON_TARE;

        }
        else
        {
            /* Measurement */
            if (Button_On_Status && !BUTTON_ON)
            {
                printf("\nMeasurement ...\n");
                Measurement_Flag = 1;

                /* Buzzer */
                Beep_20ms();
            }

            Button_On_Status = BUTTON_ON;
            /* Unit translation */
            if (Button_Uint_Status && !BUTTON_UINT)
            {
                printf("Change unit ...\n");
                Change_Unit_Flag++;

                if (Change_Unit_Flag >= 4)
                    Change_Unit_Flag = 0;

                /* Buzzer */
                Beep_20ms();
            }

            Button_Uint_Status = BUTTON_UINT;
            /* Tare weight */
            if (Button_Tare_Status && !BUTTON_TARE)
            {
                printf("Tare weight ...\n");
                LCDLIB_SetSymbol(SYMBOL_TARE, 1);
                Tare_Weight_Flag = 1;

                /* Buzzer */
                Beep_20ms();
            }

            Button_Tare_Status = BUTTON_TARE;
        }
    }
    ……
```

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_V3.02.005
- IDE 版本
  - Keil uVersion 5.36

3.2 硬體需求

- 電路元件
  - NADC24_M254_Weighing_Scale_V1.0
- 線路示意圖
  - 將 UART0 TX (PB.13) pin 腳連接到 PC UART RX，以顯示範例代碼的執行結果。

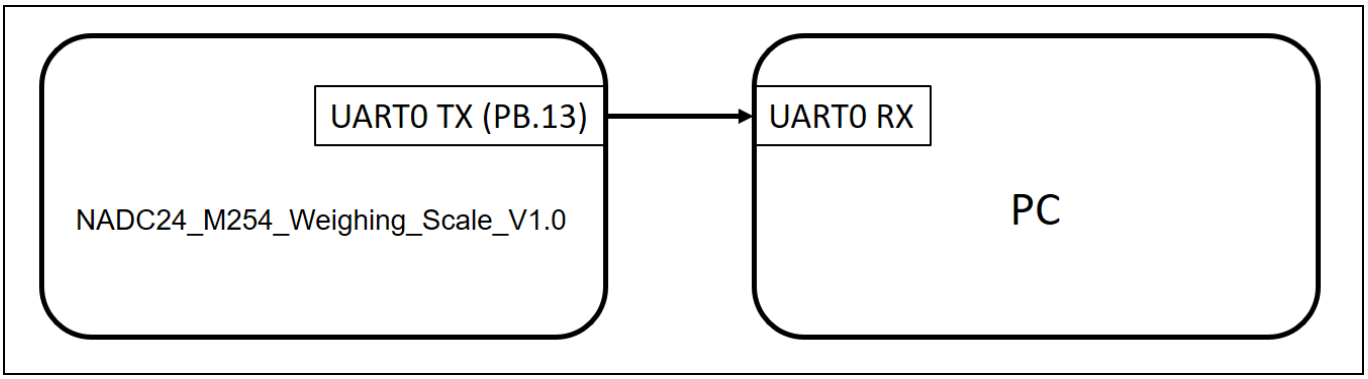


圖 3-1 線路示意圖

4. 目錄資訊

EC_NADC24_M254_Weighing_Scales_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
LCDLib	COM/SEG LCD source files
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	
Project	Source file of example code
Schematic	Schematic file of evaluation board

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 NADC24_M254_Weighing_Scales.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. 開啟超級終端機，其設定如圖 5-1 所示

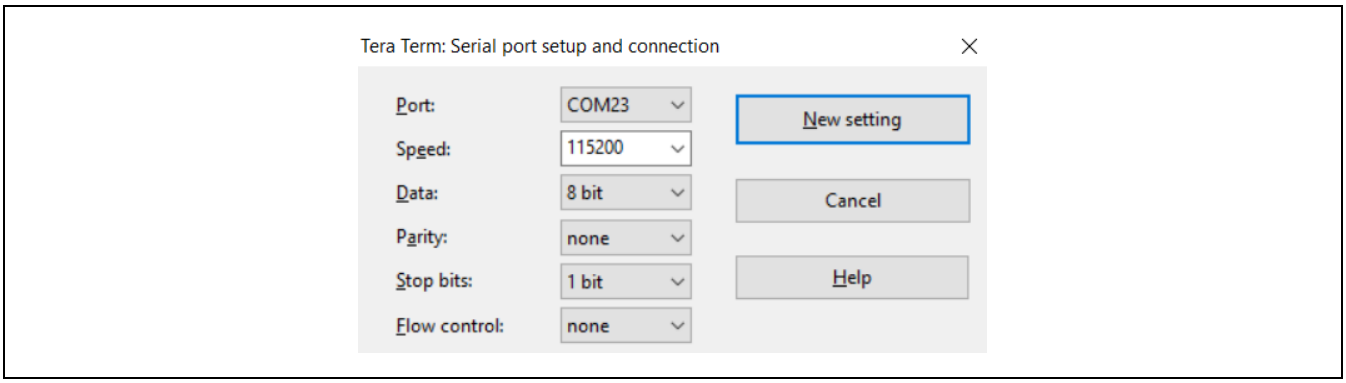


圖 5-1 超終端機設定

4. 進入除錯模式介面
   - 執行代碼

5. 超級終端機顯示系統資訊

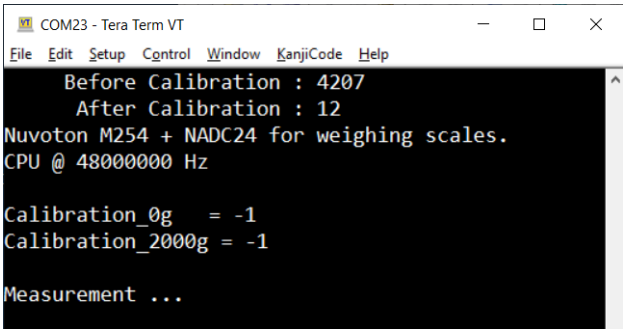


圖 5-2 系統資訊

6. 按下 START 進入校正模式

圖 5-3 校正模式

7. 按下 **UNIT** 開始第一點校正

圖 5-4 第一點校正

圖 5-5 第一點 ADC 數值

8. 放置 2Kg 砝碼到 LOADCELL 平台

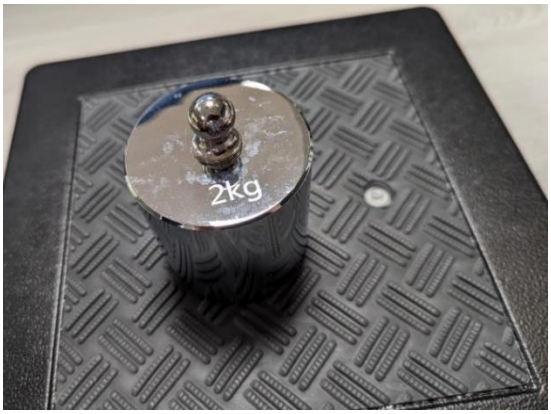


圖 5-6 2Kg 砝碼

9. 按下 **TARE** 開始第二點校正

圖 5-7 第二點校正

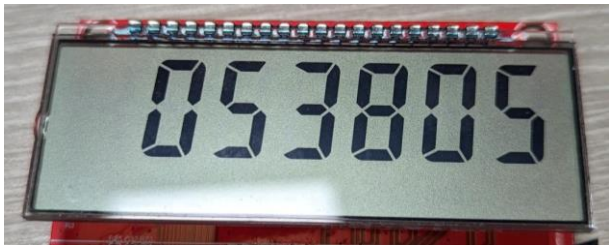


圖 5-8 第二點 ADC 數值

10. 按下 **ON** 開始量測重量

圖 5-9 量測結果

6. 修訂紀錄

Date	Revision	Description
2023.10.04	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.