

NADC24 and M254 Industrial Weighing Scales

Example Code Introduction for 32-bit NuMicro® Family

Document Information

Application	This example code uses NADC24 and M254 for industrial weighing scales.
BSP Version	M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.005
Hardware	NADC24_M254_Weighing_Scale_V1.0

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. Overview

This example code uses the NADC24 series analog-to-digital converter and M254 series microcontroller (MCU) with different weighing sensors to quickly develop applications such as weighing scales, counting scales, bench scales, and refrigerant scales.

1.1 Principle

Load Cell is a sensor that is composed of a strain gauge and a bridge circuit. When the sensor is subjected to tension or pressure, a voltage proportional to the force is generated. User can accurately read the voltage through NADC24 (24-bit ADC), convert it into a numerical value, and then transmit it to the M254 series through the SPI interface for correction and calculation. The current weight value is calculated and displayed on the LCD screen.

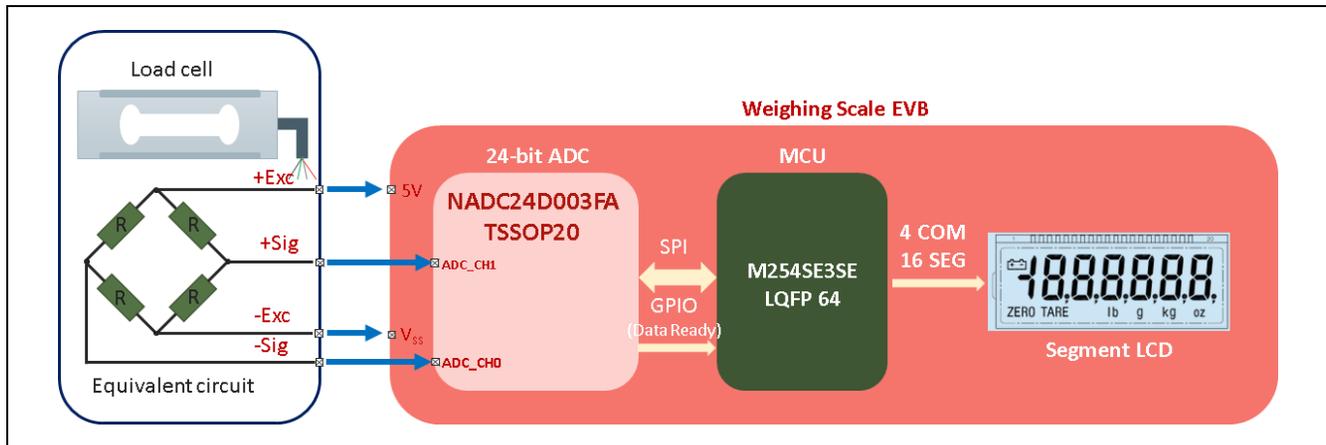


Figure 1-1 Block Diagram of Weighing Scales

1.2 NADC24 Overview

The NADC24 is a high-precision 24-bit $\Delta\Sigma$ analog-to-digital converter (ADC). It is used to read the signal of the load cell. User can set the programmable gain amplifier (PGA), sample rate, internal reference voltage, digital filter (FIR) and other functions to obtain stable and reliable data and provide the MCU for weight calculation.

	<p>System</p>	<ul style="list-style-type: none"> • 24-bit delta-sigma ($\Delta\Sigma$), analog-to-digital converters (ADCs) • Operating voltage: 2.5V to 3.6V • Temperature range: -40°C to +105°C • Integrated with Power-on Reset • Normal run: 600 uA (ADC Only) • Power-down mode: 1 uA (max) • Supports wake up from Power-down mode by SPI WAKE_UP command
	<p>Analog</p>	<ul style="list-style-type: none"> • Up to 8 single ended channels or 4 differential channels 24-bit delta-sigma ADC (ENOB up to 22-bit) • Programmable data rates from 1.25 SPS to 96 KSPS • Digital Filter option • One 12-bit DAC with rail-to-rail DAC buffer • Low-Noise PGA with gain from 1 to 128 • Internal reference: 1.2V or 2.4V • Internal temperature sensor with $\pm 2^\circ\text{C}$ accuracy
	<p>Communication interfaces</p>	<ul style="list-style-type: none"> • SPI-Compatible interface for external host accessing

Figure 1-2 NADC24 Specification

1.3 M254 Overview

The M254 series includes the Arm Cortex-M23 core, operates at 48 MHz with Flash of 128 KB, SRAM of 16 KB, and supports interfaces for COM/SEG LCD Driver and UART, I²C, SPI and other peripheral interfaces.

After reading the NADC24 converted value through the SPI interface, and then calculate it, the M254 can display four different weight units, including grams (g), kilograms (kg), pounds (lb), and ounces (oz) on the segmented LCD. User can press **UNIT** button to change units in sequence.

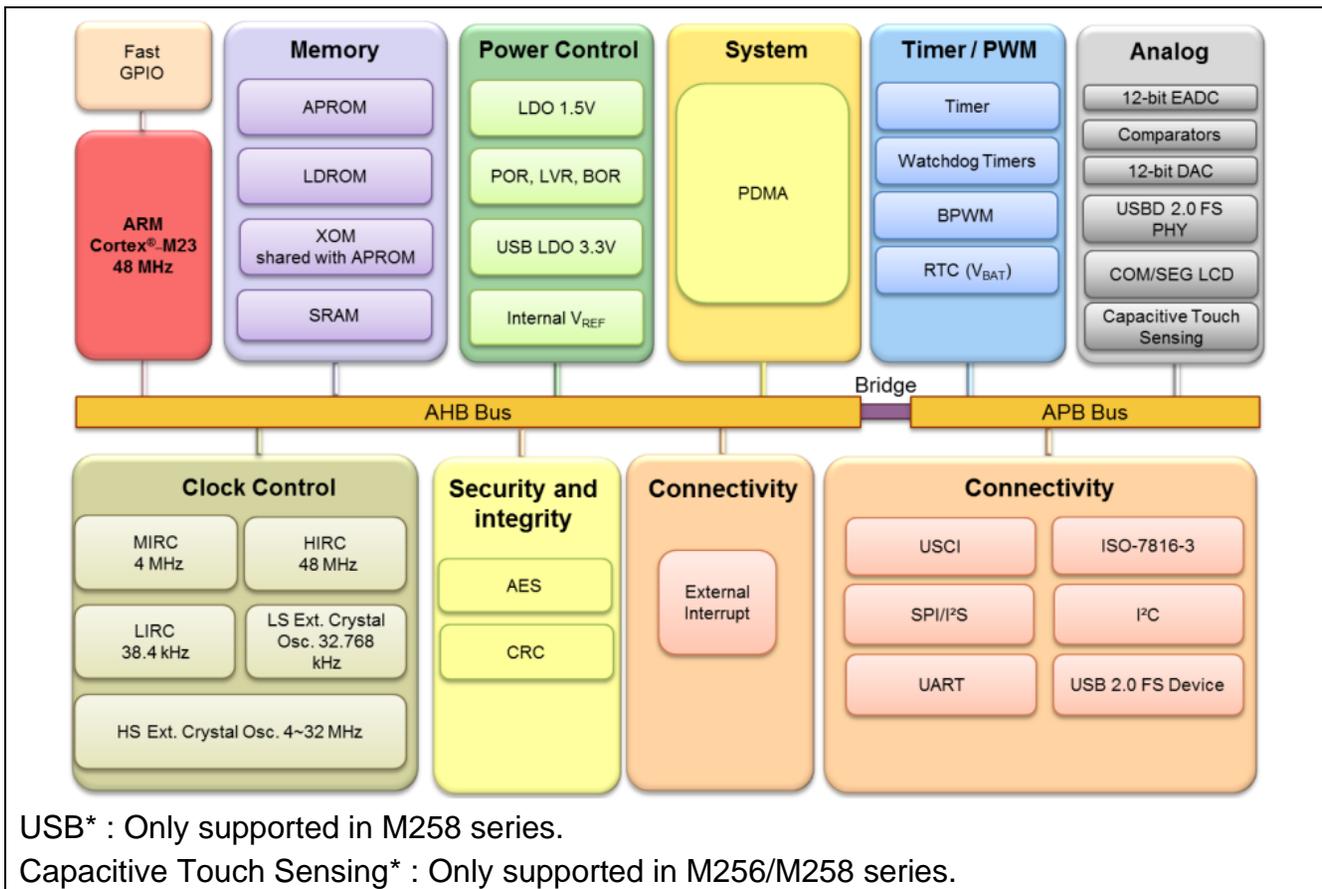
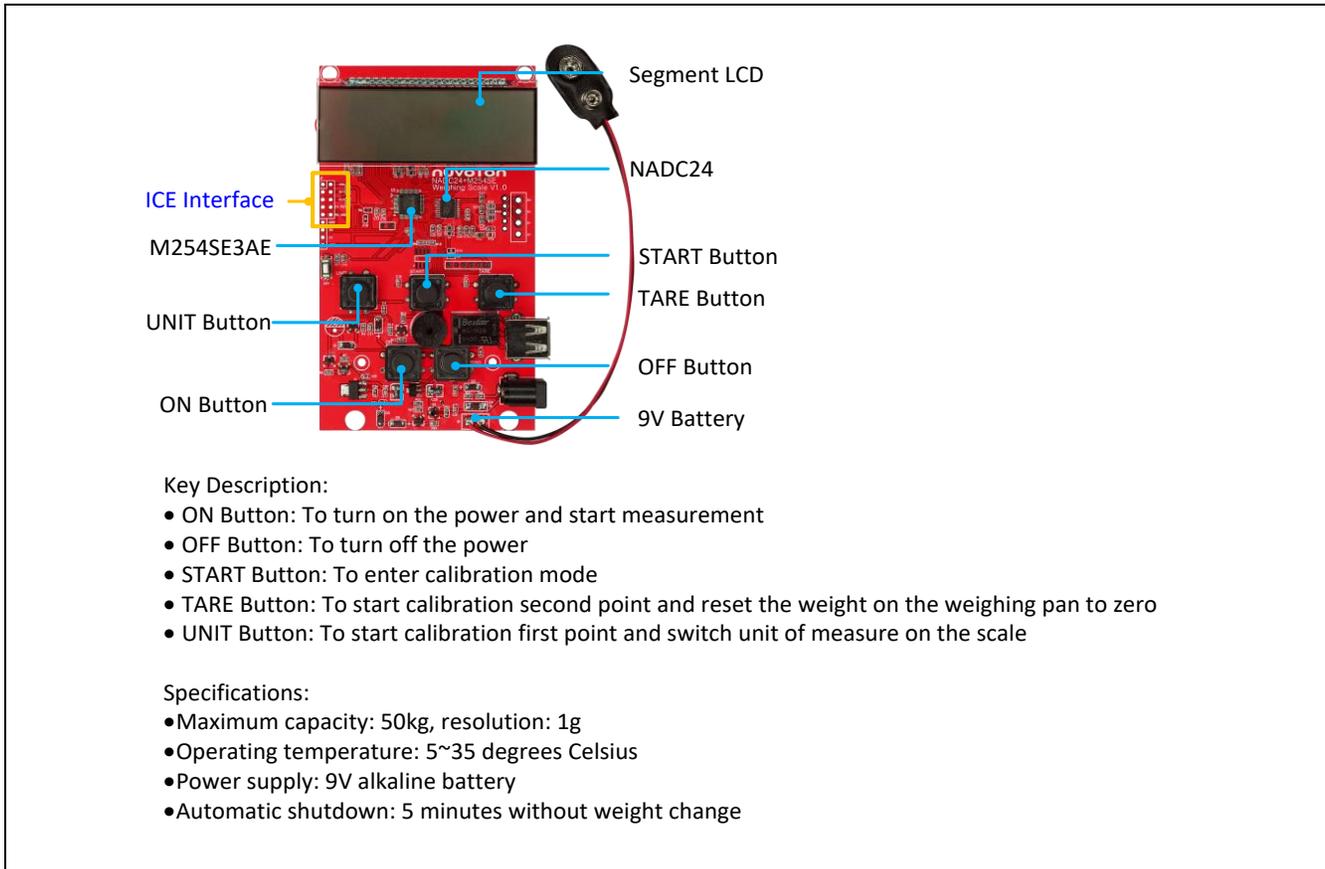


Figure 1-3 M254 Series Block Diagram

1.4 Evaluation Board Description

User can turn on the power, turn off the power, tare and convert units through buttons on the evaluation board. After connecting the board to a 50 kg weight sensor, the maximum measured weight is 50 kg and the resolution is 1 gram. In addition, the example code will determine if there is no change in weight value within 5 minutes and it will automatically shut down to achieve power saving function.



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

Figure 1-4 Evaluation Board Description

2. Code Description

Initialize system frequency, power on settings, buzzer settings and LCD settings.

```
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();
    .....
}
```

Initialize timer for display and UART for debugging.

```
.....
/* 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);
.....
```

Initialize SPI and set NADC24 for weighing scales.

```

.....
/* 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);
.....

```

Initialize buttons and read calibration data.

```

.....
/* 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_Command(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;
.....

```

Detect the button (Falling edge) for turning on the power, turning off the power, tare weight, calibration, and unit conversion.

```

.....
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. Software and Hardware Requirements

3.1 Software Requirements

- BSP version
 - M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.005
- IDE version
 - Keil uVersion 5.36

3.2 Hardware Requirements

- Circuit components
 - NADC24_M254_Weighing_Scale_V1.0
- Pin Connect
 - Connect the UART0 TX (PB.13) pin to the PC UART RX to show the execution results of this example code.

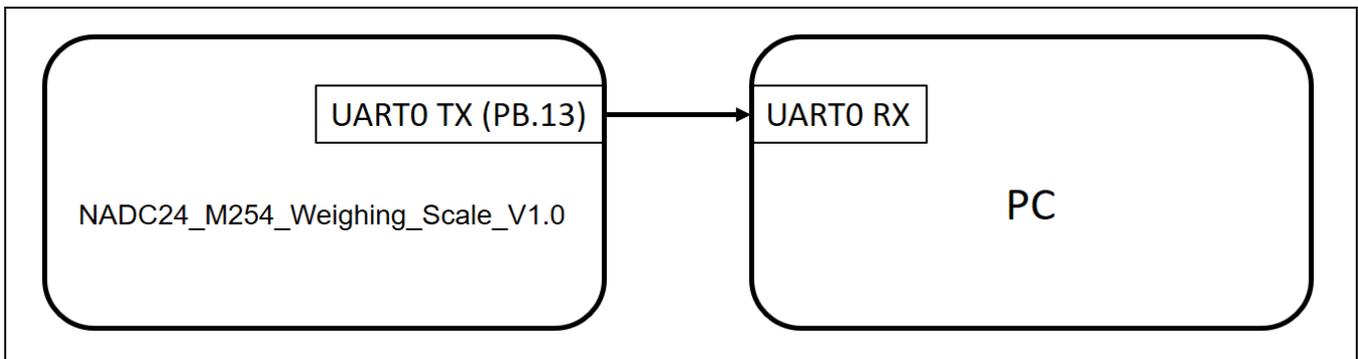


Figure 3-1 Pin Connect

4. Directory Information

The directory structure is shown below.

 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

Figure 4-1 Directory Structure

5. Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *NADC24_M254_Weighing_Scales.uvprojx*.
2. Enter Keil compile mode.
 - Build
 - Download
 - Start/Stop debug session
3. Open HyperTerminal. The settings are shown in Figure 5-1.

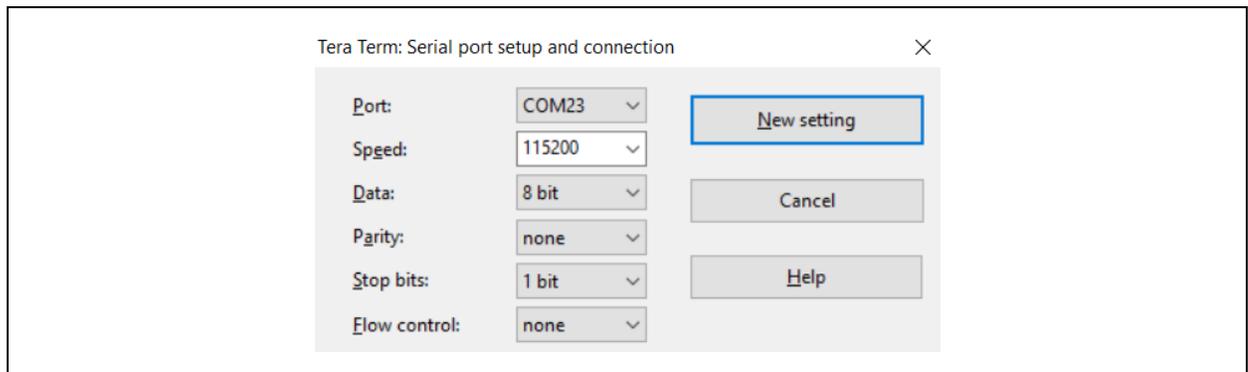


Figure 5-1 HyperTerminal Settings

4. Enter debug mode.
 - Run

- HyperTerminal will display as Figure 5-2.

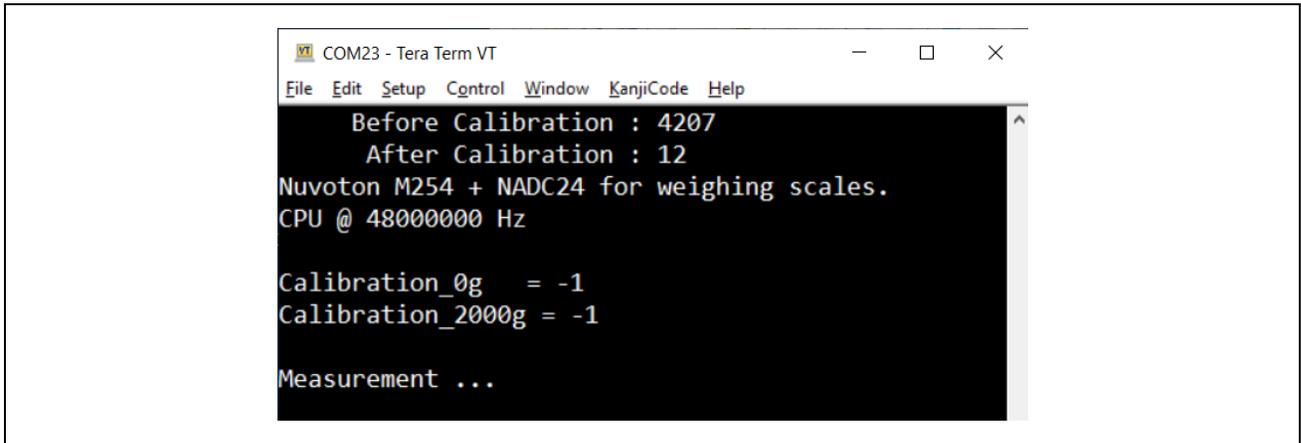


Figure 5-2 System Information

- Press **START** button to enter calibration mode.



Figure 5-3 Calibration Mode

- Press **UNIT** button to start calibrating the first point.



Figure 5-4 First Point Calibration



Figure 5-5 First Point ADC Value

8. Place 2 kilograms of weight on the LOADCELL platform.



Figure 5-6 2kg of Weight

9. Press **TARE** button to start calibrating the second point.



Figure 5-7 Second Point Calibration



Figure 5-8 Second Point ADC Value

10. Press **ON** button to start measurement.



Figure 5-9 Measurement

6. Revision History

Date	Revision	Description
2023.10.04	1.00	Initial version.

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.*