使用 NADC24 与 M258 实现功率量测

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	此范例程序代码是使用 NADC24 和 M258 实现功率量测。
BSP 版本	M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.00
开发平台	NADC24+PSU_Evaluation_Kit_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. 概述

本范例程序使用 NADC24 模拟数字转换器和 M258 微控制器，在加上电压、电流感测电路，就可以快速开发电力监控相关产品，实时显示电压/电流/功率等电力信息。

1.1 原理

当 PSU 电源讯号经过分压电阻和电流互感器处理后，分别产生微小的电压讯号，透过差动的方式输入至 NADC24。 NADC24 配备内部高分辨率的 Delta-Sigma 转换器，可将这些模拟电压讯号转换为数字讯号。 M258 单芯片用于设定 NADC24，并透过 SPI 通讯接口读取数字讯号，接着进行计算以显示线路频率、电压有效值、电流有效值、视在功率、有功功率、无功功率、以及功率因子等相关参数。

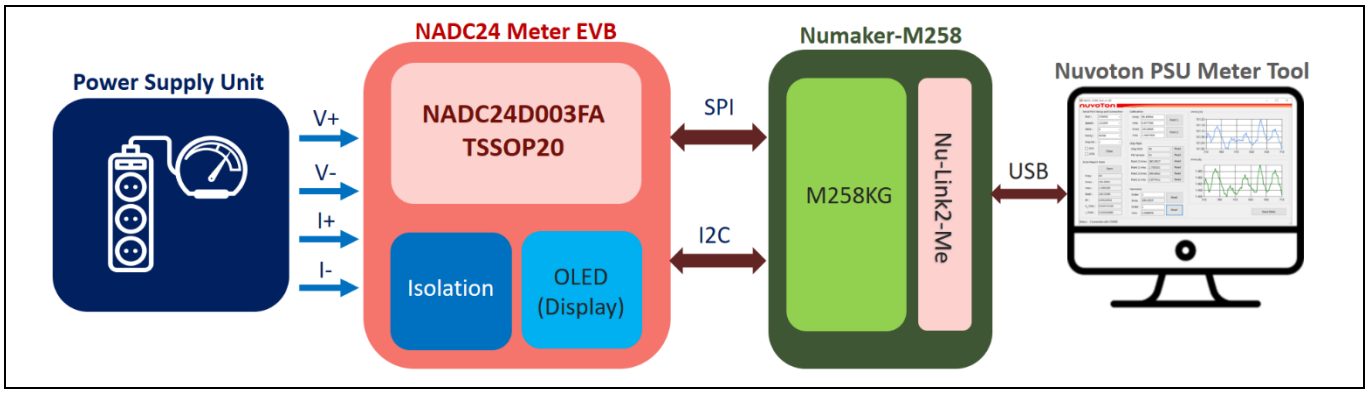

图 1-1 功率量测方块图

1.2 NADC24 概述

NADC24 是一个高精度的 24 位 ΔΣADC（模拟数字转换器），其主要功能是读取电源的电压和电流信号。我们可以通过设定资料输出率为 96K 和内部参考电压为 2.4V 等功能，来获得稳定且可靠的数据，以供微控制器（MCU）进行电力计算。

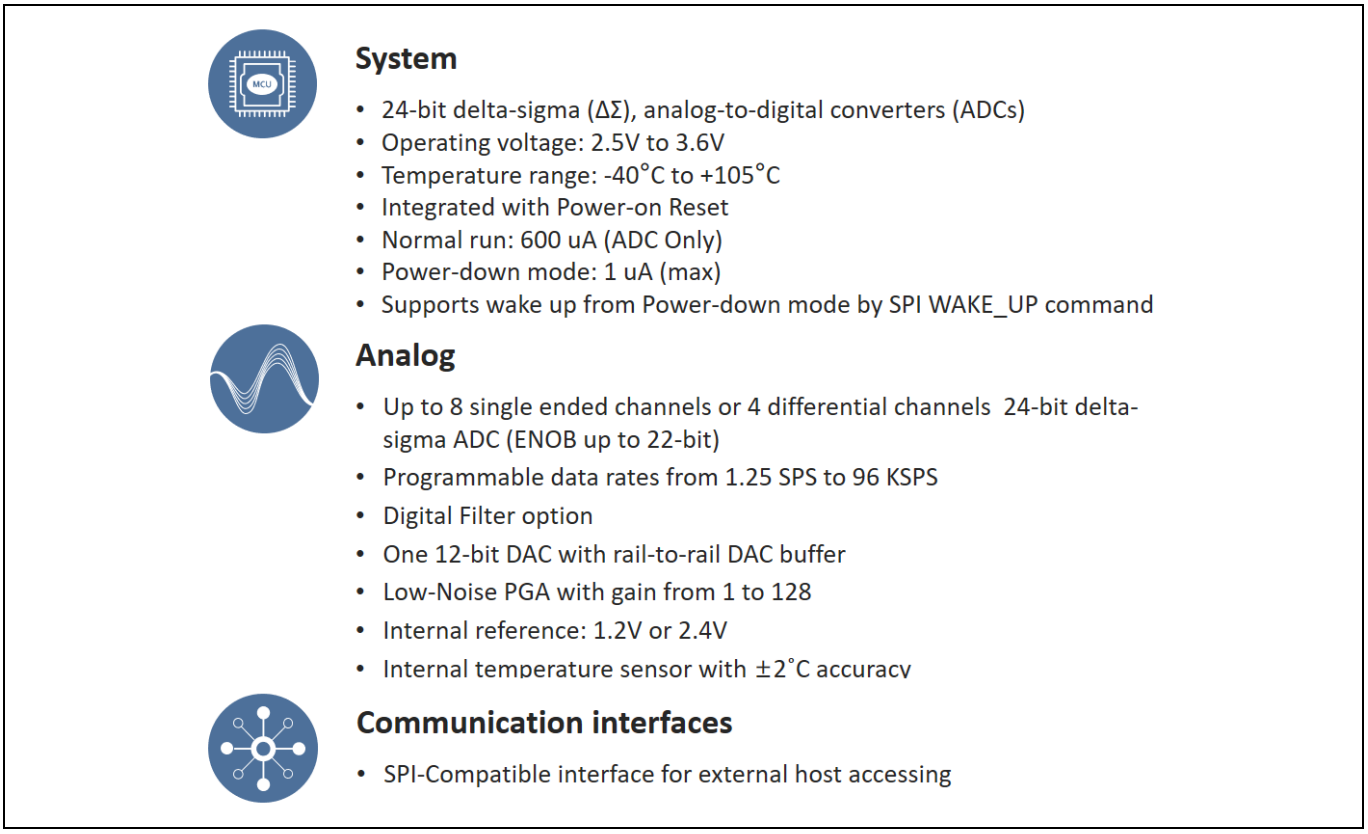

图 1-2 NADC24 规格

1.3 M258 概述

M258 是 Arm Cortex-M23 的核，运行频率 48MHz，Flash 256KB，SRAM 32KB，具有 COM/SEG LCD Driver 与 UART，I2C，SPI 等周边界面。M258 透过 SPI 接口读取 NADC24 转换的数值后，再经过计算，可以显示电源频率，电压有效值，电流有效值，视在功率，实功率，虚功率，功率因子等的资料。

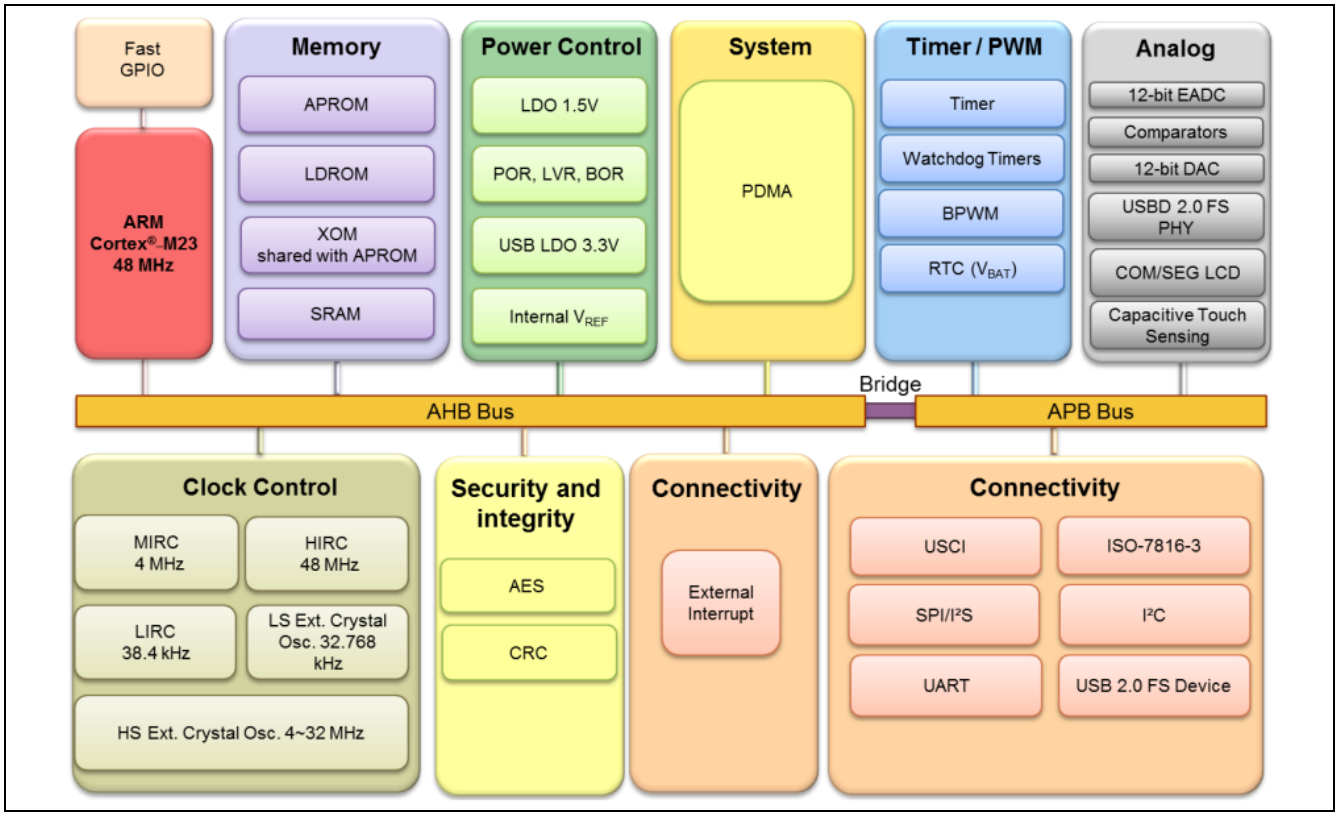

图 1-3 M258 方块图

1.4 Evaluation Board 描述

首先，将 NADC24+PSU_Evaluation_Kit (EVB) 连接到 110V 市电和家用电器。接着，透过 USB 将 NuMaker-M258KG 连接到计算机。在 EVB 上的 OLED 屏幕可用于实时显示家电的功耗，或者透过计算机上的工具，实时呈现数值和曲线。

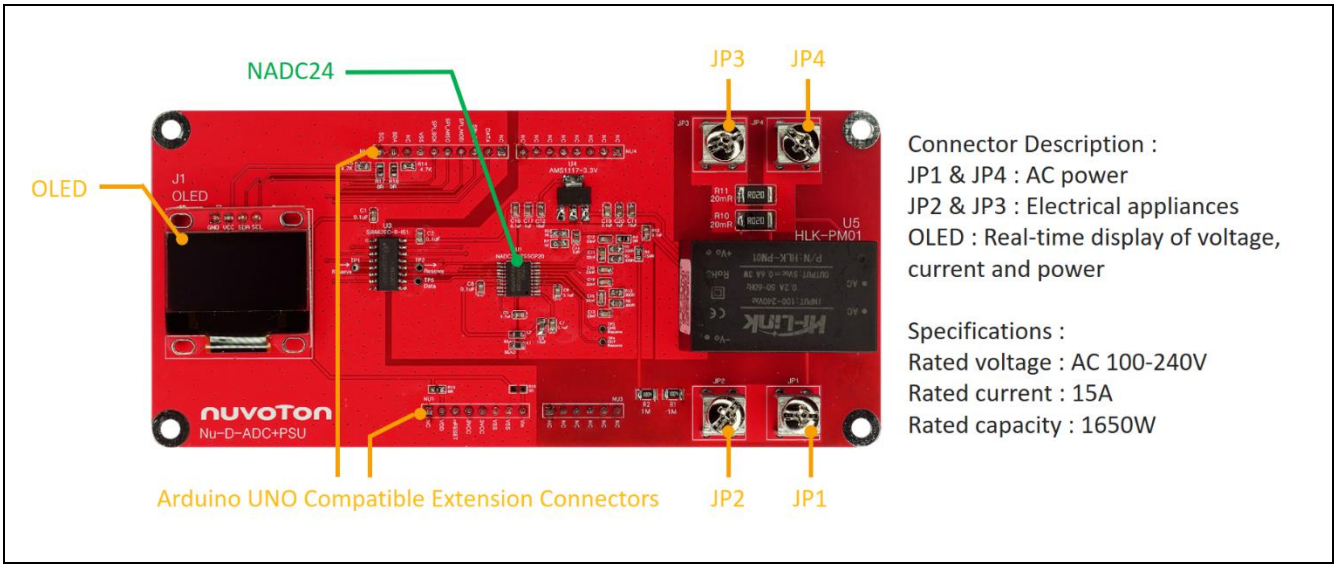

图 1-4 EVB 描述

2. 代码介绍

初始化系统频率、用于 LCD 的 I2C0 和用于除错的 UART。

```
int main(void)
{
    uint32_t Data_Addr;
    int32_t U32_Data_1, U32_Data_2, U32_Data_3, U32_Data_4, U32_Data_5, U32_Data_6;

    /* Unlock protected registers */
    SYS_UnlockReg();

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

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

    /* Init I2C0 */
    I2C0_Init();
    LCD_Disp_Info();
```

初始化 NADC24 并设定 PA4 为 Data ready。

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

    /* Enable UART RDA and THRE interrupt */
    NVIC_EnableIRQ(UART0_IRQn);
    UART_EnableInt(UART0, UART_INTEN_RDAIEN_Msk);
    UART0->FIFO = ((UART0->FIFO & (~UART_FIFO_RFITL_Msk)) | UART_FIFO_RFITL_1BYTE);

    /* IO Interrupt */
    GPIO_SetMode(PA, BIT4, GPIO_MODE_INPUT);
    GPIO_EnableInt(PA, 4, GPIO_INT_FALLING);
    NVIC_EnableIRQ(GPA_IRQn);

    /* Interrupt Priority */
    NVIC_SetPriority(SPI0_IRQn, 0) ;
    NVIC_SetPriority(GPA_IRQn, 1) ;
    NVIC_SetPriority(UART0_IRQn, 2) ;
    NVIC_SetPriority(TMR0_IRQn, 3) ;
    NVIC_SetPriority(I2C0_IRQn, 3) ;
```

初始化 FFT 并读取校准数据。

```
    /* FFT initial */
    arm_rfft_fast_instance_f32 S;
    arm_rfft_fast_init_f32(&S, SAMPLE_CYCLE_512);

    /* Read Calibration value to Variable */
    Read_Data_from_APROM(&Read_Cali_Value[0]);
    V1_Rms_Calibration = Read_Cali_Value[0];
    I1_Rms_Calibration = Read_Cali_Value[1];
    V2_Rms_Calibration = Read_Cali_Value[2];
    I2_Rms_Calibration = Read_Cali_Value[3];
    V1_DC_Calibration  = Read_Cali_Value[4];
    I1_DC_Calibration  = Read_Cali_Value[5];
    V2_DC_Calibration  = Read_Cali_Value[6];
    I2_DC_Calibration  = Read_Cali_Value[7];
    V_RMS_OFFSET_1     = Read_Cali_Value[8];
    I_RMS_OFFSET_1     = Read_Cali_Value[9];
    V_RMS_OFFSET_2     = Read_Cali_Value[10];
    I_RMS_OFFSET_2     = Read_Cali_Value[11];
    V_RMS_GAIN         = Read_Cali_Value[12];
    I_RMS_GAIN         = Read_Cali_Value[13];
    V_DC_OFFSET_1      = Read_Cali_Value[14];
    I_DC_OFFSET_1      = Read_Cali_Value[15];
    V_DC_OFFSET_2      = Read_Cali_Value[16];
    I_DC_OFFSET_2      = Read_Cali_Value[17];
    V_DC_GAIN          = Read_Cali_Value[18];
    I_DC_GAIN          = Read_Cali_Value[19];
```

如果 512 笔 ADC 数据已经准备完成后，进行运算。

```
while(1)
{
    if(Read_512_Start_Flag==0)
    {
        /* Calibration */
        if((VI_Calibration_Falg)||(DC_Calibration_Falg))
        {
            ……
        }

        if(Freq_Calc_Falg)
        {
            ……
        }

        /* Re-start saving */
        Read_512_Start_Flag = 1;
    }


    if(Power_Measurement_Falg)
    {
```

```
        /* 50Hz signal */
        if(Read_50Hz_Done_Flag && Enble_50Hz_Cycle_Flag)
        {
            ……
        }

        /* 60Hz & DC signal */
        else if(Read_60Hz_Done_Flag && (Enble_60Hz_Cycle_Flag || Enble_DC_Cycle_Flag))
        {
            ……
        }
    }

    /* Command analysis */
    if(Input_Command_Flag)
    {
        ……
    }
    else
    {
        ……
    }

    /* Finish */
    Input_Command_Flag = 0;
}
```

3. 软件与硬件需求

3.1 软件需求

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

3.2 硬件需求

- 电路组件
  - NADC24+PSU_Evaluation_Kit_V1.0
  - NuMaker -M258KG
- 线路示意图

  - 两块板 NADC24+PSU_EVB 和 NK-M258KG 透过 Arduino 接口连接。

  - 将 UART0 TX (PB.13) 接脚连接到 PC UART RX 以显示此范例程序代码的执行结果。

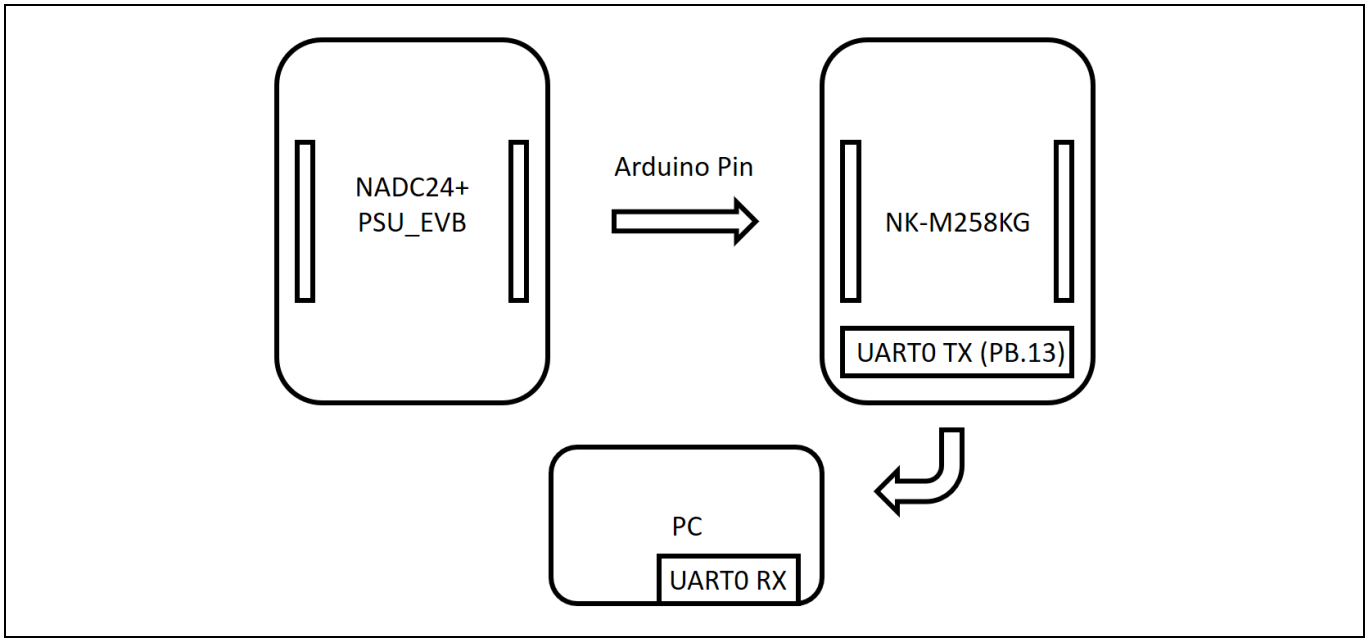

图 3-1 线路示意图

4. 目录信息

EC_NADC24_M258_Power_Measurement_V1.00	
Tool	Tool of power measurement
Schematic	Schematic of power measurement
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

图 4-1 目录信息

5. 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 NADC24_M258_Power_Measurement.uvprojx。
2. 进入编译模式接口
   - 编译
   - 下载代码至内存
   - 进入 / 离开除错模式
3. 开启 NADC_COM_Tool_v1_00.exe，点选「开启」连接串行埠

注意：你的计算机需要先安装 Microsoft .NET Framework 4.7.2

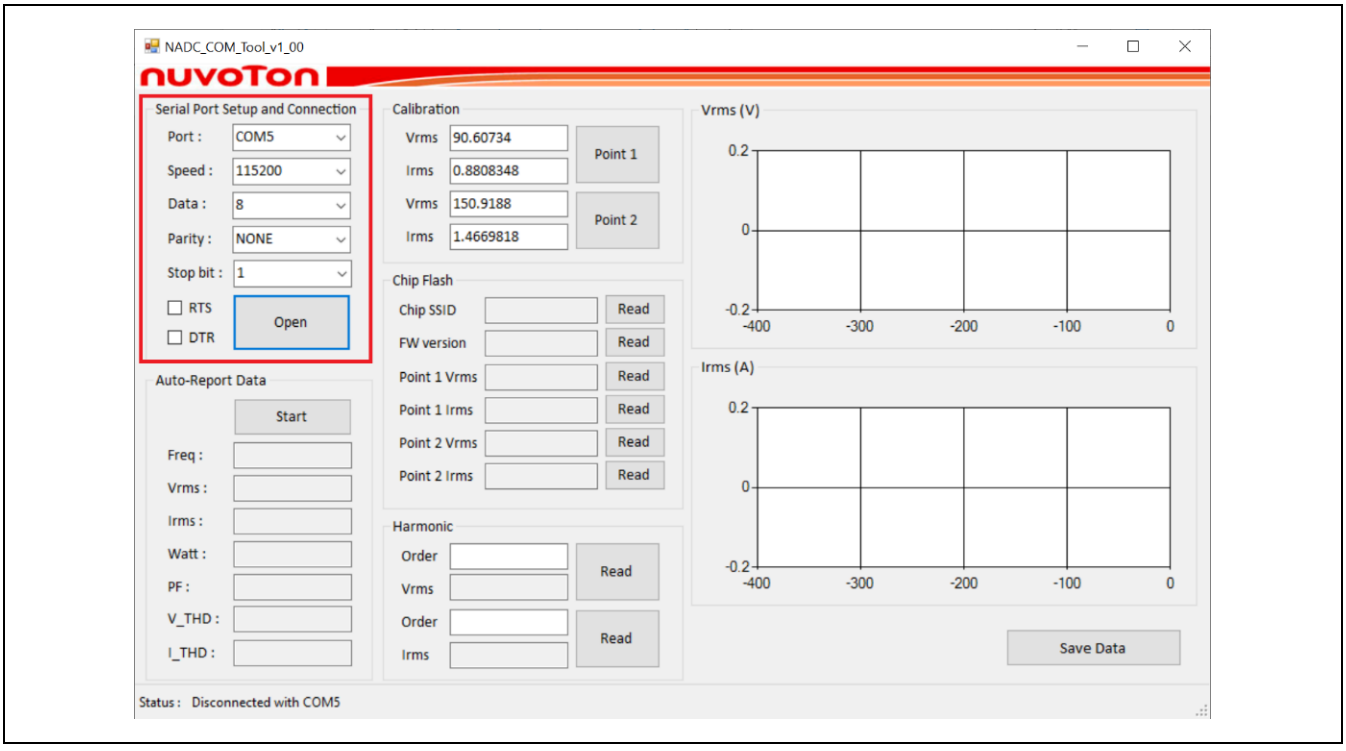

图 5-1 串行埠连接

4. 两个点校准。

5. 根据应用选择两个点进行校准。 下面以 90V 和 150V 为例。 连接 100 奥姆、高功率后，设定 90V，按 Point 1，等待几秒钟，验证值将显示在 Point 1 Vrms 和 Point 1 Irms 上。 设定 150V 后，按第 2 点并等待几秒钟。 验证值将显示在 Point 2 Vrms 和 Point 2 Irms 上以完成校准。

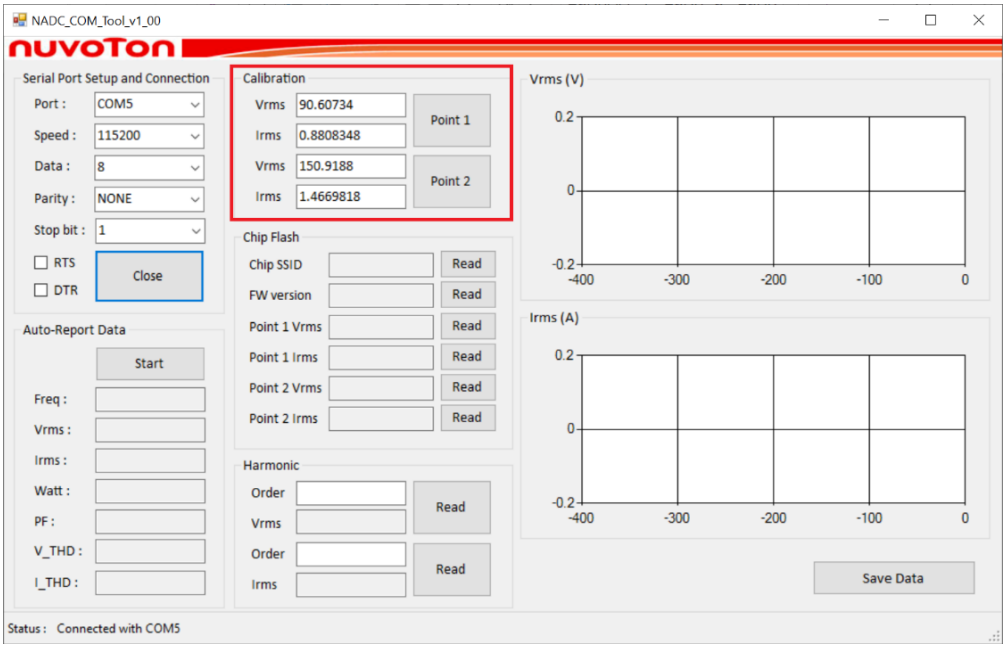

图 5-2 校准模式

6. 点选「开始」后，开始测量

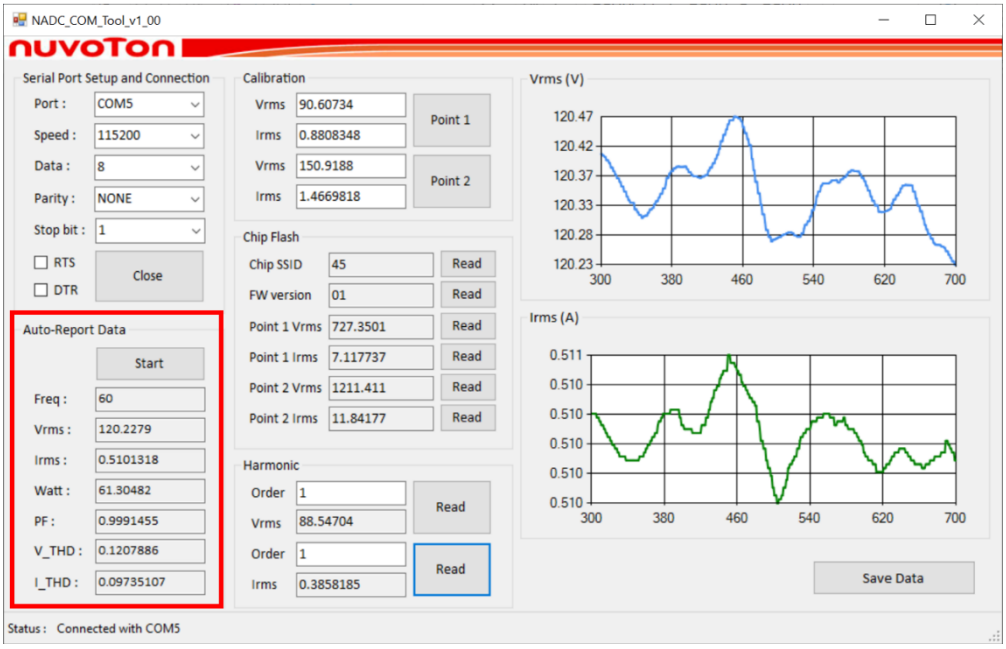

图 5-3 自动回报模式

7. 透过曲线查看 Vrms 和 Irms 数值变动并储存

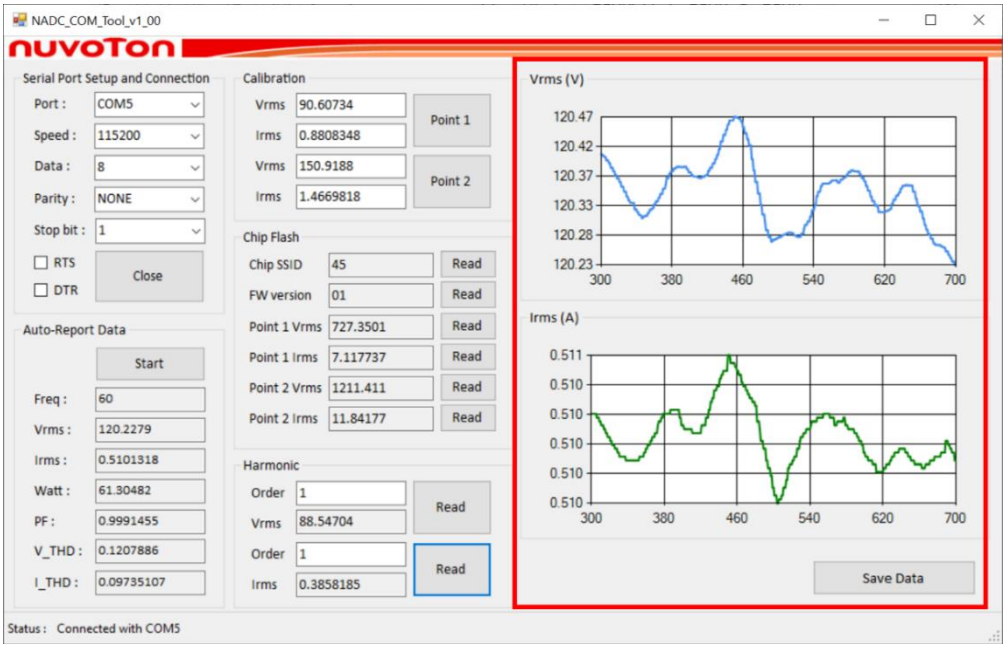

图 5-4 Vrms和Irms数值曲线

6. 修订纪录

Date	Revision	Description
2023.11.06	1.00	初始发布。

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.