NUC123 as HID RFMouse through RF24L01

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The sample code is NUC123 applied to wireless mouse through RF24L01
BSP Version	NUC123Series_BSP_CMSIS_v3.01.001
Hardware	NuTiny-EVB-123-LQFP64 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 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.

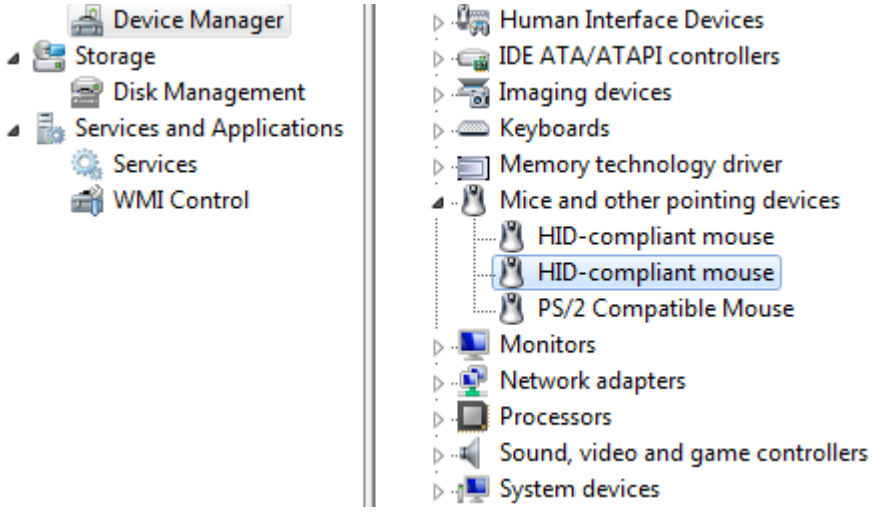
www.nuvoton.com

1 Function Description

1.1 Introduction

This sample code uses SPI0 interface to initialize RF24L01 and make it as the receiving mode to receive the data. After receiving the RF data, the corresponding data will be converted into the mouse movement direction value after processing, and the data will be uploaded to PC through USB to control the mouse action.
Use a timer to record, if 50ms does not receive data, frequency hopping once.
The IRQ (external GPIO) interrupt is used to detect whether data is received. If data is received, an IRQ interrupt will be generated. In the IRQ interrupt, if the received data flag is detected, the data will be read. If the first data is 0xAA, the second byte data will be judged and stored in the RF key variable.

1.2 Demo Result



2 Code Description

Initialize RF24L01 and set to receive mode：

```
    void ifnnrf_rx_mode(void)
    {
       power_off();
     CE=0;
            SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, TX_ADDRESS, TX_ADR_WIDTH); // Use the same address on the RX device as the TX device


            SPI_RW_Reg(WRITE_REG + EN_AA, 0x01);      // Enable Auto.Ack:Pipe0
            SPI_RW_Reg(WRITE_REG + EN_RXADDR, 0x01);  // Enable Pipe0
            SPI_RW_Reg(WRITE_REG + RF_CH, iCurCH);        // Select RF channel 40
            SPI_RW_Reg(WRITE_REG + RX_PW_P0, TX_PLOAD_WIDTH); // Select same RX payload width as TX Payload width
            SPI_RW_Reg(WRITE_REG + RF_SETUP, 0x27);   // TX_PWR:0dBm, Datarate:2Mbps, LNA:HCURR
            SPI_RW_Reg(WRITE_REG + CONFIG, 0x0f);     // Set PWR_UP bit, enable CRC(2 bytes) & Prim:RX. RX_DR enabled..


            CE = 1; // Set CE pin high to enable RX device


      //  This device is now ready to receive one packet of 16 bytes payload from a TX device sending to address
      //  '3443101001', with auto acknowledgment, retransmit count of 10, RF channel 40 and datarate = 2Mbps.


    }
```

Receive RF data：

```
void EINT0_IRQHandler(void)
{
      uint8_t u8_Status=0;
   /* For PB14, clear the INT flag */

   GPIO_CLR_INT_FLAG(PB, BIT14);
      u8_Status=SPI_Read(RF_STATUS);

       if(u8_Status&STA_MARK_TX)
       {

       }
       else if((u8_Status&STA_MARK_RX))
       {
```

```
            SPI_Read_Buf(RD_RX_PLOAD,rx_buf,TX_PLOAD_WIDTH);
          if (rx_buf[0] == 0xAA) {
            uTestRevKey=rx_buf[1];
          } else if (rx_buf[0] == 0x55)  {
            uTestRevCount1++;
          }
          rx_buf[0] = 0;
          uRevFlag = 1;
        }
        SPI_RW_Reg(WRITE_REG+RF_STATUS,0xff);

}
```

Upload data to PC via USB：

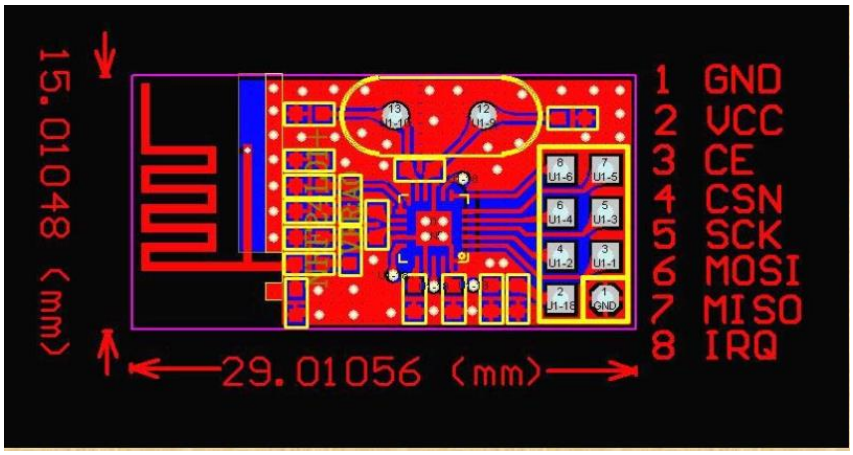
```
if (g_u8EP2Ready)
   {
      buf = (int8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP2));
      mouse_mode ^= 1;

      if (mouse_mode)
      {
         if (move_len > 14)
         {
            /* Update new report data */
            /* To control Y axis */
            if ((u32Reg & 1))
               buf[2] = -16;
            else if ((u32Reg & 2))
               buf[2] = 16;

            /* To control X axis */
            if ((u32Reg & 0x4))
               buf[1] = -16;
            else if ((u32Reg & 0x8))
               buf[1] = 16;

            /* Mouse key */
            u32MouseKey = 0;
```

```
            if ((u32Reg & 0x10))
                u32MouseKey |= 1; /* Left key */

            if ((u32Reg & 0x20))
                u32MouseKey |= 2; /* Right key */

            /* Update new report data */
            buf[0] = u32MouseKey;
            buf[3] = 0x00;
            move_len = 0;
        }
    }
    else
    {
        buf[0] = buf[1] = buf[2] = buf[3] = 0;
    }

    move_len++;
    g_u8EP2Ready = 0;
    /* Set transfer length and trigger IN transfer */
    USBD_SET_PAYLOAD_LEN(EP2, 4);
}
```

3 Software and Hardware Environment

- **Software Environment**
  - BSP Version
    - NUC123Series_BSP_CMSIS_v3.01.001
  - IDE Version
    - Keil uVersion 4.6
- **Hardware Environment**
  - Circuit Components
    - NuTiny-EVB-123-LQFP64 V1.0
    - RF24L01
  - Diagram

Schematic diagram of RF24L01 module



Nutiny-EVB-123-LQFP64 SPI0 and RF24L01 module connection diagram

VCC	↔	VCC（PIN2）
GPIO (PB.8)	↔	CE（PIN3）
SPI_SS0 (PC.0)	↔	CSN（PIN4）
SPI_CLK0 (PC.1)	→	SCK（PIN5）
SPI_MOSI_0(PC.3)	↔	MOSI（PIN6）
SPI_MISO_0(PC.2)	↔	MISO(PIN7)
INT(PB.14)	←	IRQ(PIN8)
GND	↔	GND(PIN1)
NUC123		RF24L01

4 Directory Information

- EC_NUC123_HID_RFMouse_With_RF24L01_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 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click NUC123_HID_RFMouse_With_RF24L01.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Oct.18, 2019	1.00	1. Initially issued.

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.