Nano130 Communication With BT Module

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The sample code shows how to use Nano130 to communicate with Bluetooth module VICTOR-BT4030 through UART1
BSP Version	Nano100B_Series_BSP_CMSIS_v3.03.000
Hardware	NuEdu-EVB-Nano130+NuEdu-Basic01

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

This example describes how to communicate with the Bluetooth module to achieve wireless communication. The Bluetooth module is Wicket VICTOR-BT4030, which can support the Dual-Mode Bluetooth module of Bluetooth 4.0 standard agreement. The control chip used is also from the Nuvoton M051 series.

In sample code, we use UART1 on NuEdu-EVB-Nano130 board to connect Bluetooth module. Users can install the attached file BT_Control.apk and give instructions to NuEdu board through App. When the NuEdu board receives the command from the user, it will control the corresponding components on the Basic01 board, such as: LED, 7-segment digital tube, Buzzer.

The command format is as follows:

Data format:

Byte1	Byte2	Byte3	Byte4
LED	7-segment digital tube (left)	7-segment digital tube (right)	Buzzer

Four byte commands are transmitted at a time,So the settings are set:
UART0->TLCTL |= UART_TLCTL_RFITL_4BYTES ;

when FIFO receives 4 bytes will it trigger the interrupt and enter the interrupt processing to receive the control commands of these 4 bytes.

2 Code Description

First, initialize UART1 and set the baud rate to 9600 to receive data from Bluetooth module：

```
    void UART1_Init(void)
    {
        /* Enable IP clock */
        CLK_EnableModuleClock(UART1_MODULE);

        /* Select IP clock source */
        CLK_SetModuleClock(UART1_MODULE,CLK_CLKSEL1_UART_S_HIRC,CLK_UART_CLK_DIVIDER(1));

       /* Set PA multi-function pins for UART1 RXD and TXD */
        SYS->PC_H_MFP &= ~( SYS_PC_H_MFP_PC10_MFP_Msk | SYS_PC_H_MFP_PC11_MFP_Msk);
        SYS->PC_H_MFP |= (SYS_PC_H_MFP_PC10_MFP_UART1_RX |
SYS_PC_H_MFP_PC11_MFP_UART1_TX );

         SYS_ResetModule(UART1_RST);

        /* Configure UART1 and set UART1 Baudrate */
        UART_Open(UART1, 9600);
         /* Set RX FIFO Trigger Level to 4 bytes */
         UART0->TLCTL |= UART_TLCTL_RFITL_4BYTES ;

    }
```

The 4-byte data received is stored in the Command array through the interrupt service function of UART1.

```
void UART1_IRQHandler(void)
{
    /* Get all the input characters */
    while(UART_IS_RX_READY(UART1))
    {
                /* Get the character from UART Buffer */
        UART_Read(UART1, Command, 4);
    }
}
```

Parse the data of each byte and control the LED, 7-segment digital tube and buzzer on the NuEdu-Basic01 board.

```
void LED_on(unsigned int temp)
{
    if((temp&1)!=1)
        LED1_OFF;
    else
        LED1_ON;

    temp=temp>>1;

    if((temp&1)!=1)
        LED2_OFF;
    else
        LED2_ON;

    temp=temp>>1;
    if((temp&1)!=1)
        LED3_OFF;
    else
        LED3_ON;

    temp=temp>>1;
    if((temp&1)!=1)
        LED4_OFF;
    else
        LED4_ON;

    temp=temp>>1;
    if((temp&1)!=1)
        LED5_OFF;
    else
        LED5_ON;

    temp=temp>>1;
    if((temp&1)!=1)
        LED6_OFF;
    else
        LED6_ON;
```

```
    temp=temp>>1;
    if((temp&1)!=1)
        LED7_OFF;
    else
        LED7_ON;

    temp=temp>>1;
    if((temp&1)!=1)
        LED8_OFF;
    else
        LED8_ON;

}
void Show_Seven_Segment(unsigned char no, unsigned char number)
{
    SEG_A_OFF;
    SEG_B_OFF;
    SEG_C_OFF;
    SEG_D_OFF;
    SEG_E_OFF;
    SEG_F_OFF;
    SEG_G_OFF;
    SEG_H_OFF;
    SEG_CONTROL1_OFF;
    SEG_CONTROL2_OFF;

    switch(number)
    {
    case 1:
        SEG_CONTROL1_ON;
        break;

    case 2:
        SEG_CONTROL2_ON;
        break;
    }

    switch(no)
    {
    //show 0
    case 0:
```

```
        SEG_A_ON;
        SEG_B_ON;
        SEG_C_ON;
        SEG_D_ON;
        SEG_E_ON;
        SEG_F_ON;
        break;

    //show 1
    case 1:
        SEG_B_ON;
        SEG_C_ON;
        break;

    //show 2
    case 2:
        SEG_A_ON;
        SEG_B_ON;
        SEG_G_ON;
        SEG_E_ON;
        SEG_D_ON;
        break;

    //show 3
    case 3:
        SEG_A_ON;
        SEG_B_ON;
        SEG_G_ON;
        SEG_C_ON;
        SEG_D_ON;
        break;

    //show 4
    case 4:
        SEG_F_ON;
        SEG_B_ON;
        SEG_G_ON;
        SEG_C_ON;
        break;

    //show 5
```

```
    case 5:
        SEG_A_ON;
        SEG_F_ON;
        SEG_G_ON;
        SEG_C_ON;
        SEG_D_ON;
        break;

    //show 6
    case 6:
        SEG_A_ON;
        SEG_F_ON;
        SEG_E_ON;
        SEG_G_ON;
        SEG_C_ON;
        SEG_D_ON;
        break;

    //show 7
    case 7:
        SEG_A_ON;
        SEG_B_ON;
        SEG_C_ON;
        SEG_F_ON;
        break;

    //show 8
    case 8:
        SEG_A_ON;
        SEG_B_ON;
        SEG_C_ON;
        SEG_D_ON;
        SEG_E_ON;
        SEG_F_ON;
        SEG_G_ON;
        break;

    //show 9
    case 9:
        SEG_A_ON;
        SEG_B_ON;
```

```
        SEG_C_ON;
        SEG_F_ON;
        SEG_G_ON;
        break;
    }
}
void Write_Buzzer(unsigned int frequence, unsigned int duty)
{

    // Config PWM1 channel 1
    PWM_ConfigOutputChannel(PWM1, 1, frequence, duty);
    // Enable PWM1 channel 1 to drive buzzer module
    PWM_EnableOutput(PWM1, 0x02);
}
```

3 Software and Hardware Environment

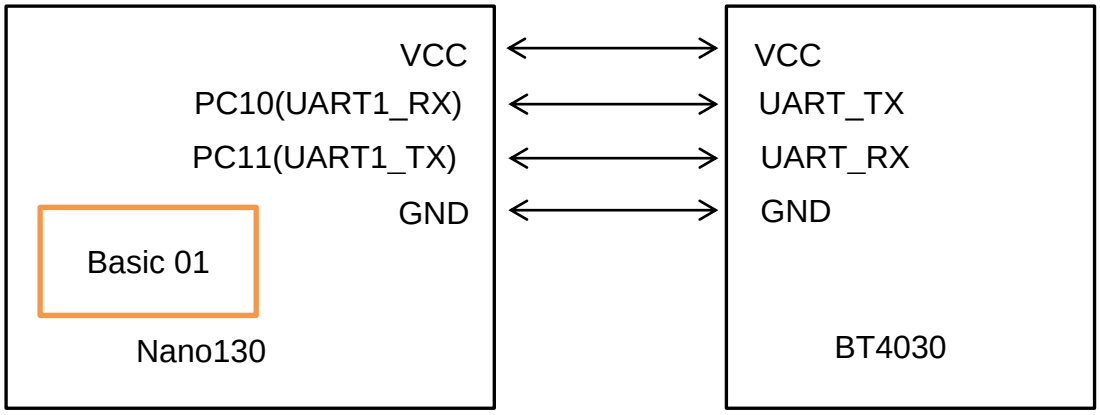
Software Environment

- BSP Version
  - Nano100B_Series_BSP_CMSIS_v3.03.000
- IDE Version
  - Keil uVersion 4.6

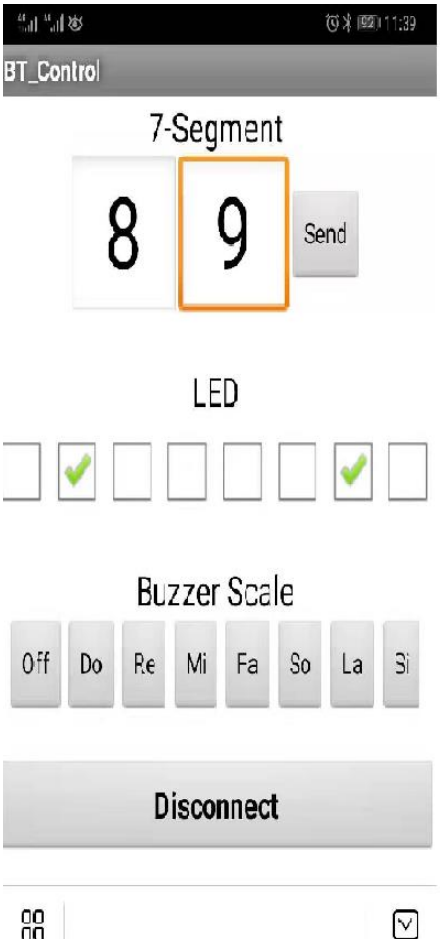
Hardware Environment

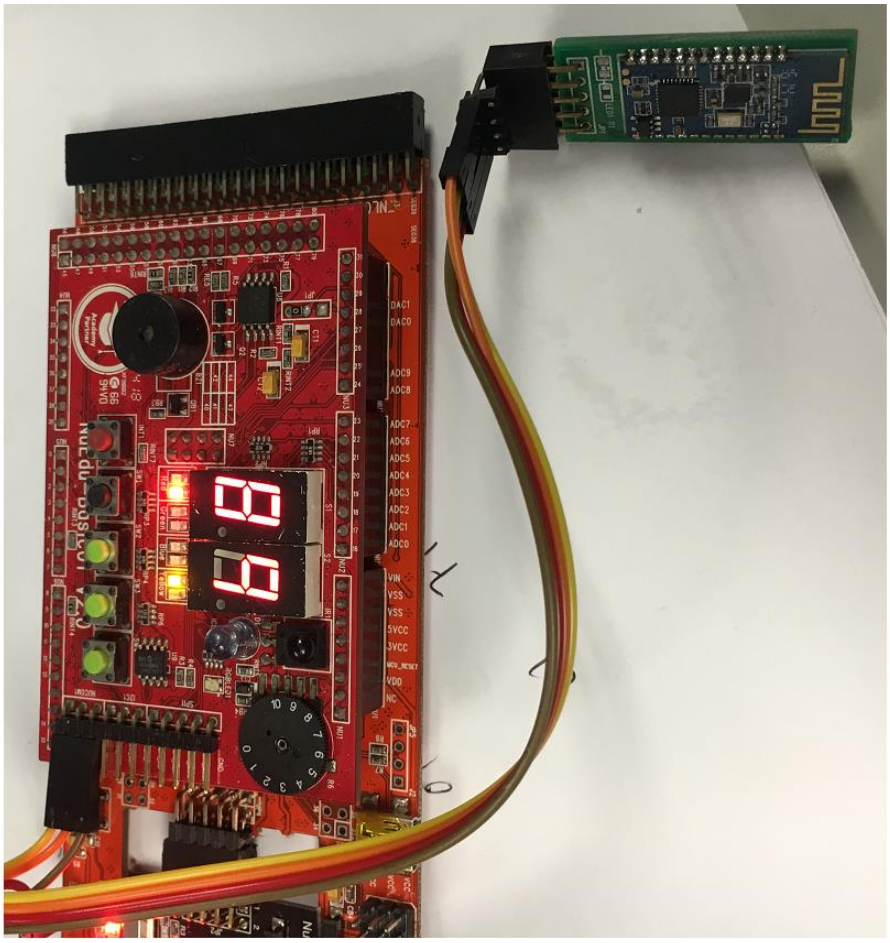
- Circuit Components
  - NuEdu-EVB-Nano130
  - NuEdu-Basic01
  - VICTOR-BT4030
- Diagram

  Nano130 receive the data of BT4030 through UART1, then analyzes the data and controls the LED light on the basic1 board.



Results view





4 Directory Information

- EC_Nano130_Wireless_Communication_With_Bluetooth_Module_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
    - NuEdu — Library for NuEdu board
    - 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 Nano130_Communication_With_Bluetooth.uvproj
2. Enter the compilation mode interface
   a. Compilation
   b. Download code to Flash
   c. Enter/leave debugging mode
3. Enter the debugging mode interface
   a. Execution code

6 Revision History

Date	Revision	Description
Sep.25, 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.