

M031 Controls AS7221 by AT Command

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This example code uses AT command to communicate with the AS7221 module through the USCI_UART interface.
BSP Version	M031 BSP CMSIS V3.03.000
Hardware	NuMaker-M031KI V1.0, AS7221 Module

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

In this example code, the M031 can send AT command by the UART interface to communicate with the AS7221 embedded module to control the lamp behavior, including brightness adjustment, color temperature control, gradient time tuning and so on.

1.1 Principle

The AS7221 is an intelligent lighting management chip, which adopts embedded three-color CIE XYZ color sensor technology. It can realize the precise induction of illumination and color temperature, and then control its own PWM output to achieve the precise control of brightness and color temperature. By combining with the original Digital Addressable Lighting Interface (DALI) application solution, accurate control of lamps can be realized with DALI instructions.

The DALI host sends dimming and color instruction through the DALI bus. The microcontroller receives and parses the instruction as the slave, and then sends the corresponding AT control command to the AS7221 through the UART interface. Excluding receiving instruction from DALI bus, this sample code implements some features to control the AS7221 module by AT commands, including the dimming level, illumination bright control, color temperature and light ON/OFF control.

The AT command format of AS7221 consists of Set and Query commands as described below.

- Set command
This command includes the command string with parameters, such as “ATcmd=<value>”. After receiving AT Set command, the AS7221 will return the string “OK” to indicate Success or “ERROR” for Failure.
- Query command
This command is only with the command string, such as: “ATcmd”. If the AS7221 accepts the query, the string “<value> OK” will be returned. Otherwise, the AS7221 returns the string “ERROR” to indicate Failure.

Specifically, the ATcmd indicates the related AT command defined by AS7221, where <value> is the value to be set in Set command or the value to be found in Query command. The detailed command set has been introduced in AS7221 Datasheet (sourced from <http://www.ams.com>).

1.2 Operation Process

First, the related hardware connection must be ready. Then, power on the AS7221 module before M031 NuMaker Board. After around one second, the AS7221 can be ready to receive the AT command from M031. When this sample code begins to execute on NuMaker Board, the console can display the message as Figure 1-1.

```

=====
Bright Control by
  a. Bright Control is determined by Dimming or LUX value.
  b. Dimming value (0-100 percents; must disable LUX mode)
  c. LUX value (0-64000; must enable LUX mode)

Color Temperature (CT) Control by
  a. Must enable CCT mode
  b. CT value (Kelvin range: 400-1500)

=====
Press key to do following selection
=====
0: increase Dimming value
1: decrease Dimming value
2: increase LUX value (must LUX mode enabled?)
3: decrease LUX value (must LUX mode enabled?)
4: increase CT value (must CCT mode enabled)
5: decrease CT value (must CCT mode enabled)
a: Light ON
b: Light OFF
h: enable CCT mode
i: disable CCT mode
j: enable LUX mode
k: disable LUX mode

=====
Current Settings:
  CCT mode: [Enabled; CT Control by CT value]
  LUX mode: [Enabled; Bright Control by LUX value]
  CT value = 2600
  LUX value = 1000
  Dimming value = 50
=====

```

Figure 1-1 Console Message

In this sample code, the M031 provides some features of the AS7221 module, including brightness control, color temperature control and light ON/OFF. Specifically, only the Correlated Color Temperature (CCT) mode is enabled, the color temperature control can be effective. Setting both the dimming and LUX values can control the brightness individually. In the AS7221 module, the LUX mode enabled or disabled determines that the brightness control is based on the LUX value or dimming value, respectively.

1.3 Demo Result

✚ Set the color temperature 2600 and dimming value 50% as Figure 1-2.

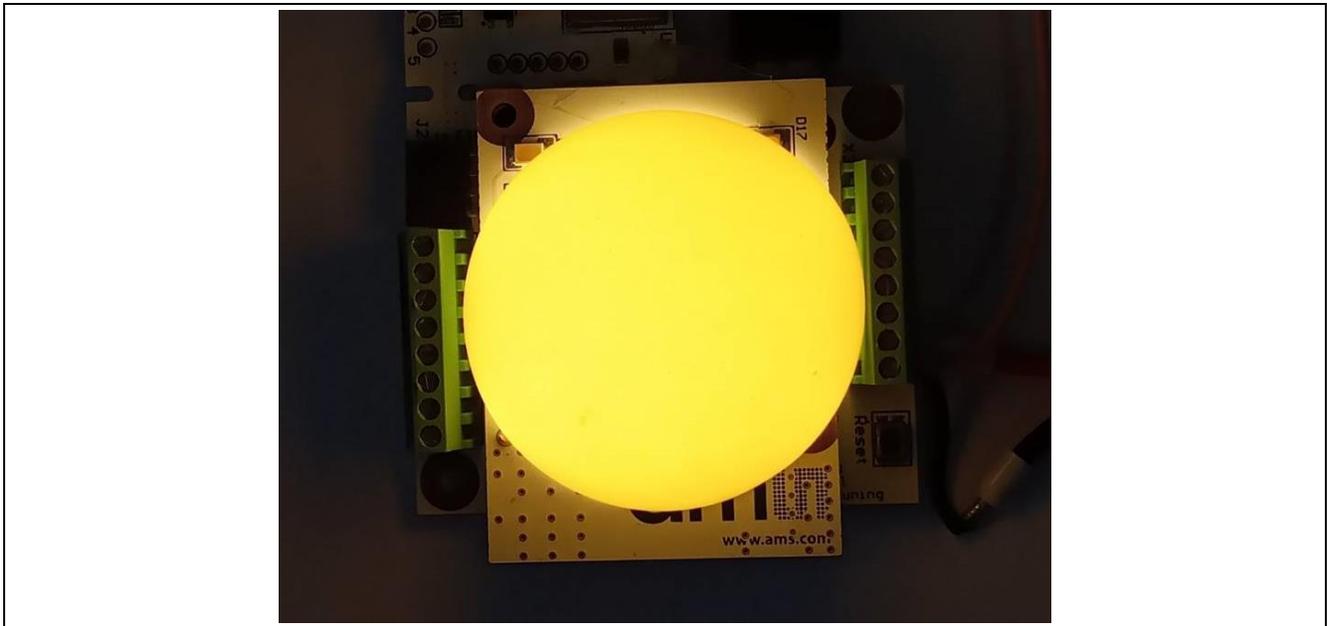


Figure 1-2 CT 2600 and Dimming Value 50%

✚ Set the color temperature 4500 and dimming value 50% as Figure 1-3.

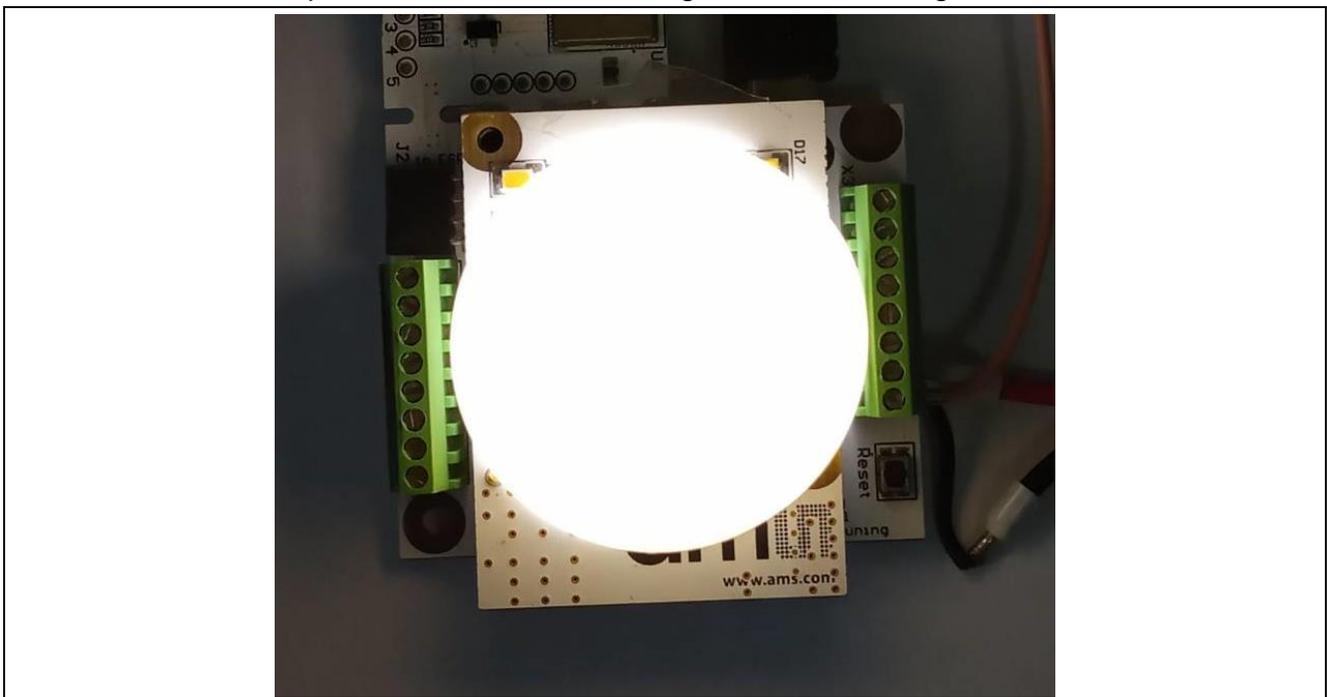


Figure 1-3 CT 4500 and Dimming Value 50%

2 Code Description

The main command flow between the M031 and AS7221 is based on AS7221_Send_SetCmd() function. From this function, both AT Set and Query commands for AS7221 can be implemented well. First, the command string is packed to charbuf[] and sent from M031 by USCI_UART port. After receiving the command, the AS7221 will return the response string to g_u8RecData[] buffer. From decoding the buffer, the user can know whether the previous AT command to be accepted or not.

```
void AS7221_Send_SetCmd(uint8_t cmd, uint32_t value)
{
    ...
    switch (cmd)
    {
        case eAT_LIGHT:
            sprintf(charbuf, "ATLIGHT=%d\r\n", numb);
            UART_Write(UUART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
            break;
        case eAT_ATDIM:
            lightnumb = value * 100;
            lightnumb = lightnumb >> 8;
            sprintf(charbuf, "ATDIM=%d\r\n", lightnumb);
            UART_Write(UUART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
            break;
        case eAT_ATLED23M:
            lightnumb = value;
            sprintf(charbuf, "ATLED23M=%d\r\n", lightnumb);
            UART_Write(UUART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
            break;
        case eAT_ATCCTT:
            lightnumb = value;
            sprintf(charbuf, "ATCCTT=%d\r\n", lightnumb);
            UART_Write(UUART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
            break;
        case eAT_ATCT:
            sprintf(charbuf, "ATCT=%d\r\n", value);
            UART_Write(UUART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
            break;
        case eAT_GET_ATCT:
            sprintf(charbuf, "ATCT\r\n");
            UART_Write(UUART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
    }
}
```

```
        break;

    case eAT_ATDL:
        sprintf(charbuf, "ATDL=%d\r\n", value);
        UART_Write(UART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
        break;
    case eAT_ATLUXT:
        sprintf(charbuf, "ATLUXT=%d\r\n", value);
        UART_Write(UART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
        break;
    case eAT_GET_ATLUXT:
        sprintf(charbuf, "ATLUXT\r\n");
        UART_Write(UART0, (uint8_t *)charbuf, ASS7221_Charbuf_Size);
        g_GetRecFalg = 1;
        break;
    default:
        break;
}
}
```

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
 - ◆ M031 BSP CMSIS V3.03.000
- IDE version
 - ◆ Keil uVersion 5.27

3.2 Hardware Requirements

- NuMaker-M031KI V1.0 Board
- AS7221 Module
 - ◆ a0061i_CSS SLIK AS7221
 - ◆ a0062b_CSS SLIK AS3834 LED

- Pins Connection

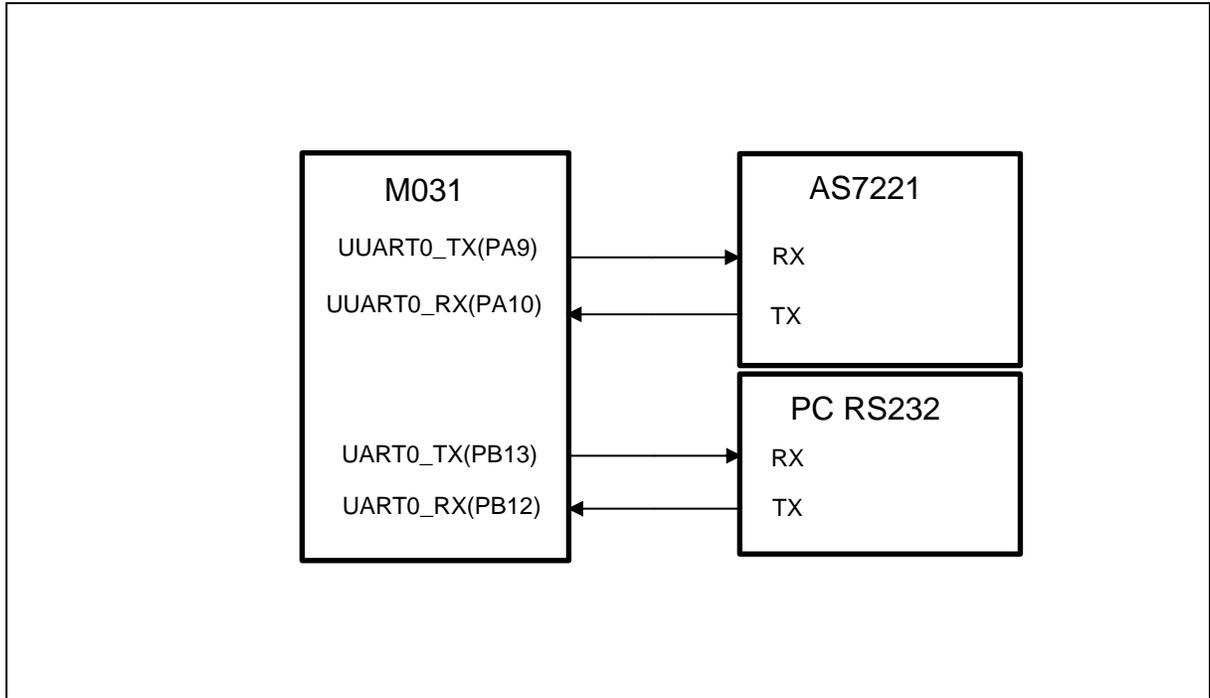


Figure 3-1 Hardware Environment

4 Directory Information

The directory structure is shown below.

 EC_M031_AS7221_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 Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *M031_AS7221.uvproj*.
2. Enter Keil compile mode.
 - Build
 - Download
 - Start/Stop debug session
3. Enter debug mode.
 - Run

6 Revision History

Date	Revision	Description
2021.07.16	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.*