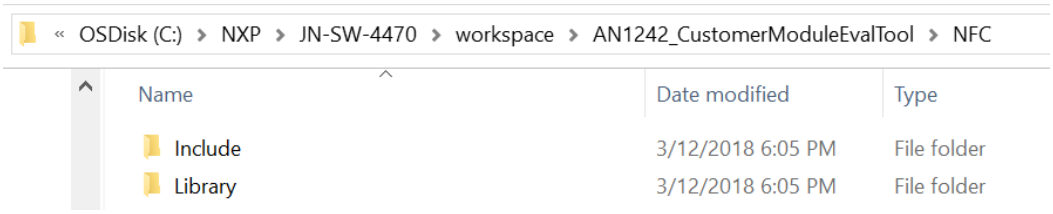
如何集成 NTAG 功能到 CMET

(shaozhong.liang@nxp.com)

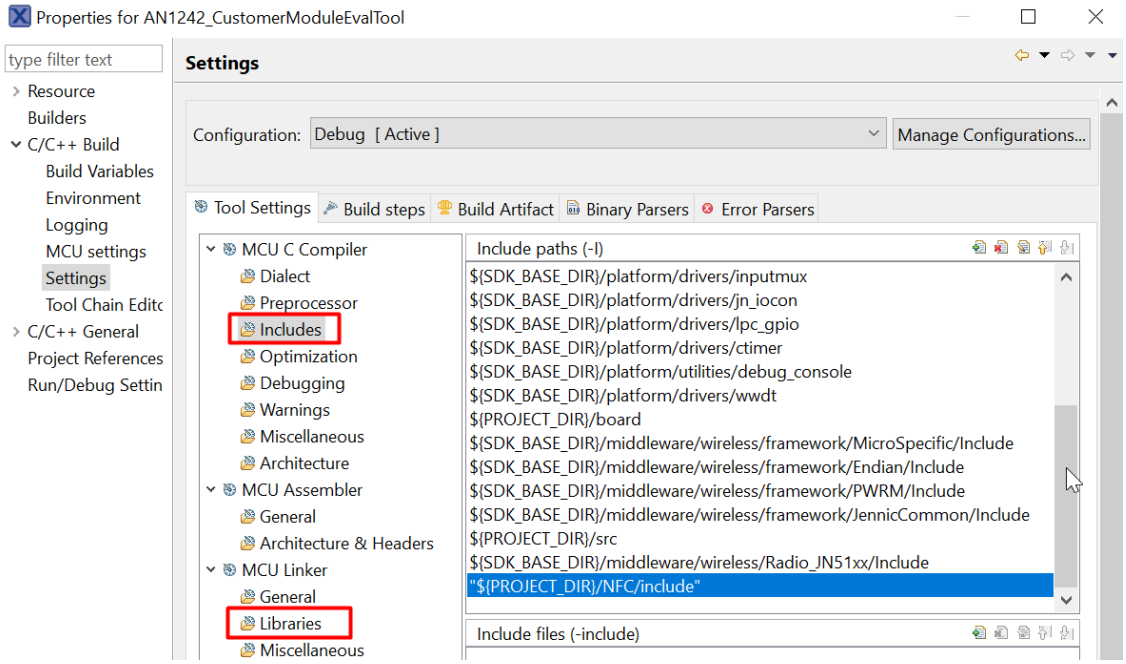
JN-5189 内置 NTAG32211 标签，完全符合 NFC Forum 类型 2 标签和 ISO/IEC14443 类型 A 规范。可以实现 ZigBee 简单入网配对、设备识别等功能。

为了将 NFC 驱动集成到 JN-AN1242-CMET 中，参考下面的步骤修改 JN-AN1242-CMET 工程代码。

1. 从 JN-AN-1244 中复制 NFC 目录到 JN-AN1242-CMET



2． 修改 JN-AN1242-CMET 工程设置，增加 NFC 的头目录好库文件目录。



3．修改 pin_mux.c 源代码文件，增加 I2C0 的管脚映射配置。

```
void BOARD_InitPins(void)
{
    /* Enable IOCON clock */
    CLOCK_EnableClock(kCLOCK_Iocon);
    CLOCK_EnableClock(kCLOCK_InputMux);

    /* USART0 RX/TX pin */
    IOCON_PinMuxSet(IOCON, 0, 8, IOCON_MODE_INACT | IOCON_FUNC2 | IOCON_DIGITAL_EN);
    IOCON_PinMuxSet(IOCON, 0, 9, IOCON_MODE_INACT | IOCON_FUNC2 | IOCON_DIGITAL_EN);

    /* Debugger signals */
    IOCON_PinMuxSet(IOCON, 0, 12, IOCON_FUNC2 | IOCON_MODE_INACT | IOCON_DIGITAL_EN);
    IOCON_PinMuxSet(IOCON, 0, 13, IOCON_FUNC2 | IOCON_MODE_INACT | IOCON_DIGITAL_EN);

    /* I2C0  */
    IOCON_PinMuxSet(IOCON, 0, 10, IOCON_FUNC5 | IOCON_DIGITAL_EN | IOCON_STDI2C_EN);  /* I2C0_SCL */
    IOCON_PinMuxSet(IOCON, 0, 11, IOCON_FUNC5 | IOCON_DIGITAL_EN | IOCON_STDI2C_EN);  /* I2C0_SDA */

    /* Bridge SSEL requires detection to set state correctly */
    //ConfigureBridgeSSEL();

    /* IOCON clock left on, this is needed if CLKIN is used. */
    /* Initialize GPIO */
    CLOCK_EnableClock(kCLOCK_Gpio0);
    RESET_PeripheralReset(kGPIO0_RST_SHIFT_RSTn);

    CLOCK_EnableClock(kCLOCK_I2c0) ;
    CLOCK_AttachClk(kOSC32M_to_I2C_CLK);
}
```

4．增加 NFC 驱动代码。下面是 NFC 功能主要接口 API 函数。

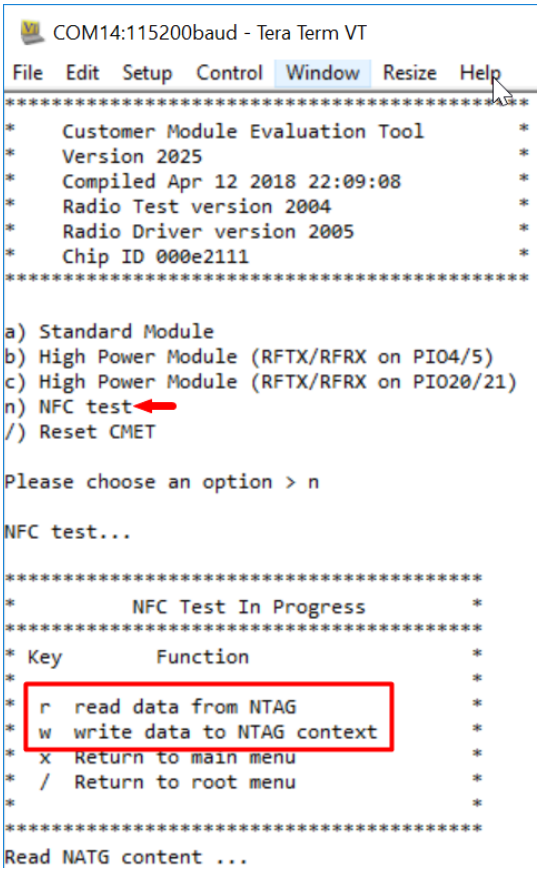
```
PUBLIC void NTAG_vInitialise(
       uint8    u8Address,     /*!< Reader I2C address (0xFF for automatic detection) */
       uint8    u8I2cScl,      /*!< 6x: 16   = DIO16
                                *       others = DIO14
                                *   7x: 4      = DIO4
                                *       others = DIO3
                                *   8x: 6      = I2C1 DIO6
                                *       12     = I2C1 DIO12
                                *       15     = I2C0 DIO15
                                *       0xff   = I2C2 (internal NTAG)
                                *       others = I2C0 DIO10 */
       uint32  u32FrequencyHz, /*!< Frequency in Hz */
       uint8    u8InputFd      /*!< Input DIO for field detect
                                *   0xff to ignore FD line */
       );

PUBLIC  bool_t NTAG_bRead(
        uint32    u32ReadAddress, /*!< Byte address of data to read */
        uint32    u32ReadLength,  /*!< Number of bytes to read */
        uint8    *pu8ReadData     /*!< Buffer to read data into */
        );

PUBLIC  bool_t NTAG_bWrite(
        uint32    u32WriteAddress,  /*!< Byte address of write */
        uint32    u32WriteLength,   /*!< Number of bytes to write */
        uint8    *pu8WriteData      /*!< Buffer to write data from */
        );

PUBLIC void NTAG_vRegCbEvent(
       tprNtagCbEvent prRegCbEvent /*!< Pointer to event callback function */
       );

PUBLIC void NTAG_vTick(
       uint32  u32TickMs /*!< Number of ms since previous call */
       );
```

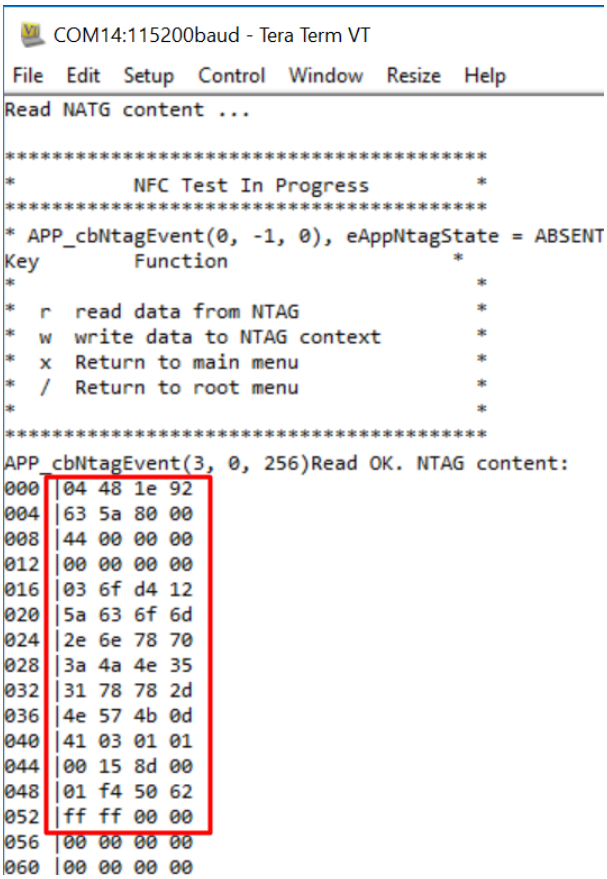
5．由于通过 I2C 方式进行 NTAG 读、写速度比较慢，必须通过异步方式进行读写。因此需要启动一个 5ms 的定时器，驱动 NTAG 的读、写状态机。通过 NTAG_vRegCbEvent 注册的事件回调函数获得 NTAG 读、写执行结果。

```
PUBLIC  void        APP_cbNtagEvent(    /* Called when an event takes place */
        teNtagEvent eNtagEvent,         /* Event raised */
        uint32      u32Address,
        uint32      u32Length,
        uint8       *pu8Data)           /* Event data (NULL if no data) */
{
    /* Which event ? */
    switch (eNtagEvent)
    {
        /* Absent ? */
        case E_NTAG_EVENT_ABSENT:
        {
            /* Add customer’s code here */
        }
        break;

        /* Present ? */
        case E_NTAG_EVENT_PRESENT:
        {
            /* Add customer’s code here */
        }
        break;

        /*!< Read request failed */
       case E_NTAG_EVENT_READ_FAIL:
        {
            /* Add customer’s code here */

        }
        break;

        /*!< Read request succeeded */
```

```
        case E_NTAG_EVENT_READ_OK:
        {
            /* Add customer's code here */
        }
        break;

        /*!< Write request failed */
        case E_NTAG_EVENT_WRITE_FAIL:
        {
            /* Add customer's code here */
        }
        break;

        /*!< Write request succeeded */
        case E_NTAG_EVENT_WRITE_OK:
        {
            /* Add customer's code here */
        }
        break;

        /* Others ? */
        default:
        {
            /* Do nothing */
        }
        break;
    }
}
```

6． 完成上述步骤后，在 CMET 中集成了 NFC 功能，可以读、写 NTAG 内部的 EEPROM 数据。





下图是在支持 NFC 功能的手机上运行 NFC TagInfo 应用的截图，通过 CMET 读取的数据内容与通过手机 NFC 读取的数据一致。