

NS-RX231 Shield Mode Guide

Contents

1 Shield Mode.....	2
1.1 Summary.....	2
1.2 Protocol.....	3
1.3 Command.....	3
1.4 Function setting	4
2 Function	6
2.1 Default mode	6
2.2 Infrared Transmission (IR TX)	7
2.3 Infrared Receive (IR RX).....	8
2.4 Speaker (SPK)	9
2.5 Bluetooth (BLT).....	10
2.6 MPU-6050(MPU).....	12

1 Shield Mode

1.1 Summary

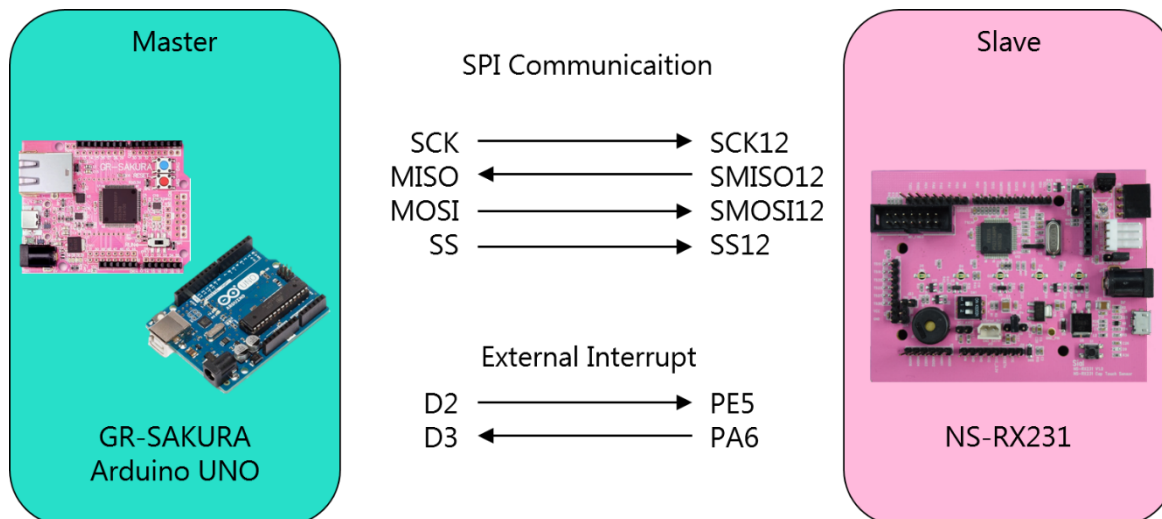


Figure 1-1 Communication pin connection

The NS-RX231 shield mode uses the E2 studio to send commands from the master board by attaching a board (Arduino Uno, GR-SAKURA), which is compatible with Arduino pin placement, without changing the NS-RX231's firmware. (Infrared Transmitter, Infrared Receiver / Speaker / Bluetooth / MPU-6050).

Commands and data are sent and received using SPI communication. Four pins are used for SPI communication, and two pins for interrupt to indicate data receiving. Total of 6 pins are used for shield mode.

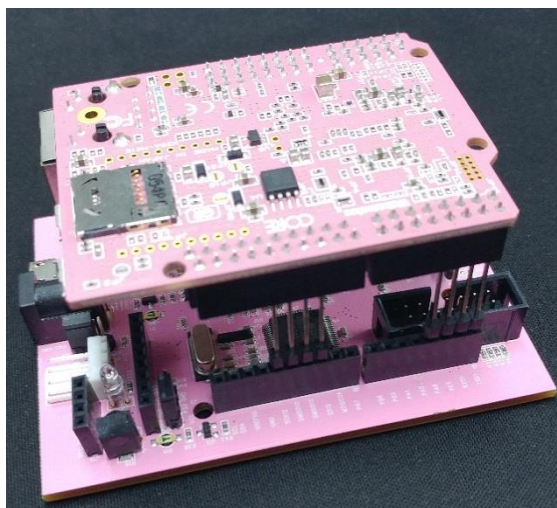


Figure 1-2 Linkage example

1.2 Protocol

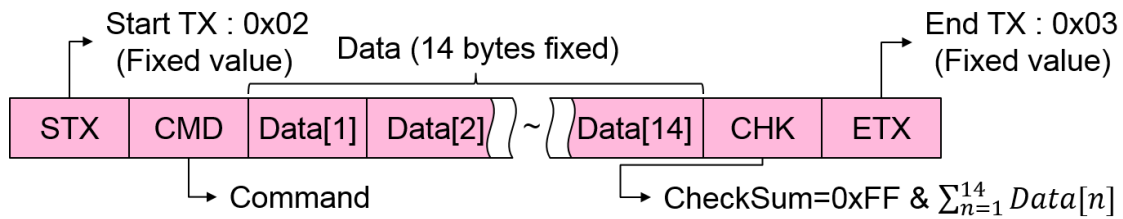


Figure 1-3 Protocol

1.3 Command

Name	Defined	Value
Infrared Transmission(TX)	CMD_IRTX	0x01
Infrared Receive(RX)	CMD_IRRX	0x03
Speaker	CMD_SPK	0x04
Bluetooth	CMD_BLT	0x10
MPU-6050	CMD_MPU	0x20
LED, Touch	CMD_LED	0x40
Function Setting	CMD_SET	0x80

The command is used to set the function or to distinguish which function the received data is from.

In coding, the library header file is defined in "CMD_[function]" format for the convenience of the programmer.

※ Function Setting cannot set multiple function

- Infrared transmission, reception, speaker ... Only one function of these can be performed.

※ The function settings are set and executed only once when the power is turned on and the operation starts. You cannot change to another function in processing.

1.4 Function setting

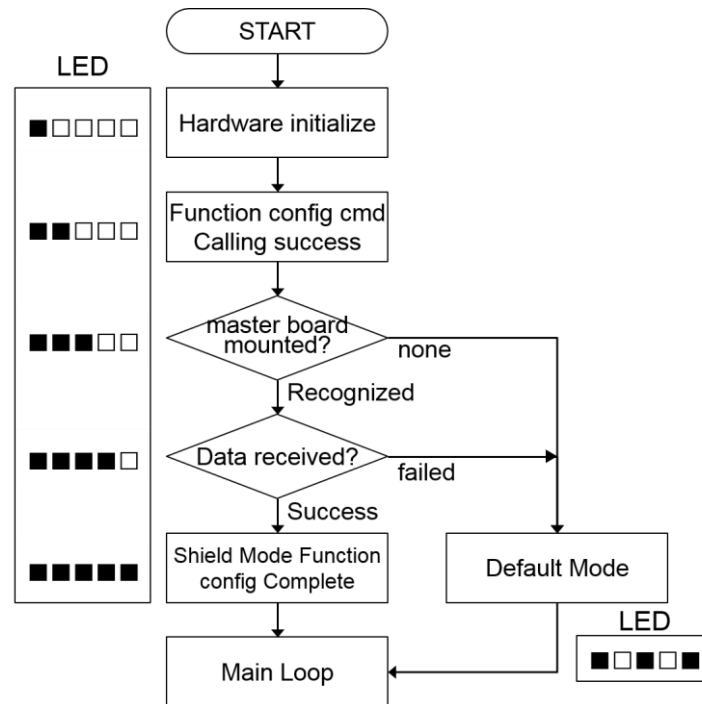


Figure 1-4 Flowchart

When using the firmware in NS-RX231 Shield Mode, it is necessary to set the functions (IR RX, Speaker, Bluetooth...) to be used by communicating with the additional master board before entering the main loop. Once each step is completed, the LEDs will light one by one to indicate the current step.

- i. When the NS-RX231 is powered on and starts operating, it will proceed port and hardware initialization.
- ii. Call the function that handles the dedicated protocol for setting the function.
- iii. Confirm that the master board is connected.
 [Arduino] D2 → [NS-RX231] (interrupt)
 (interrupt) [Arduino] ← PA6 [NS-RX231]
 If the master board is not connected or does not recognize correctly, it will enter the default mode.
- iv. When the master board is connected, it receives data from the master board. This data is important for function setting and operation. If the data is not received correctly, it will enter the default mode.
- v. Function setting is completed.

When the function setting is completed, the LED corresponding to the set function flashes for about 2 seconds to show which function is set. (Following figure 1-5)

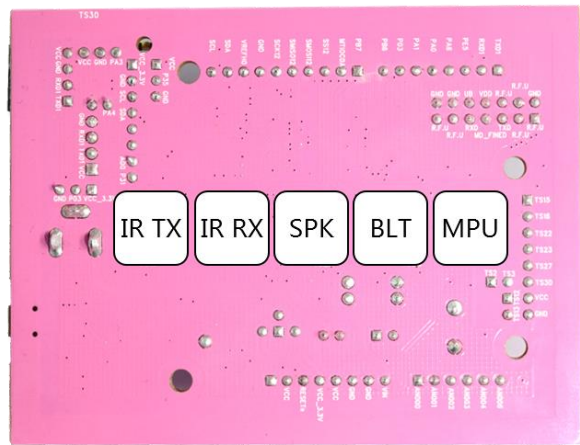


Figure 1-5 LED for function

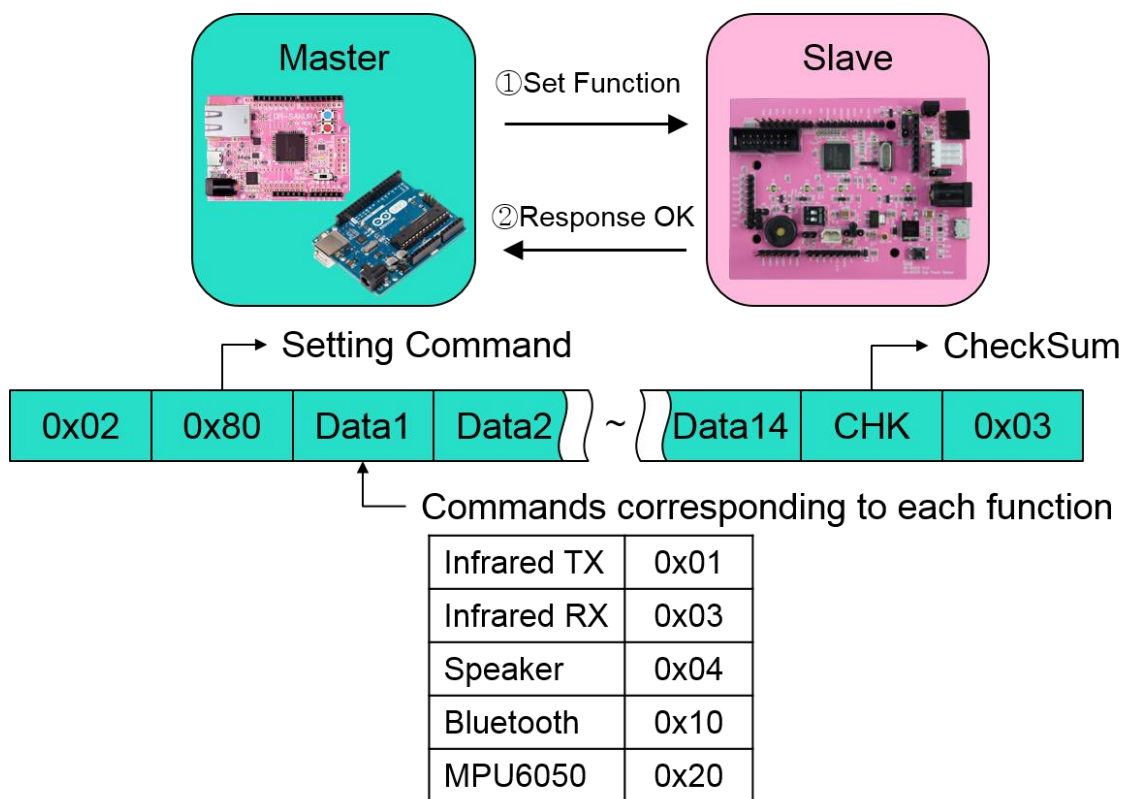


Figure 1-6 Function setting

If you want to use the Speaker function, put the command code corresponding to each function in "CMD" of the protocol. And, transmit the command by using the following protocol.

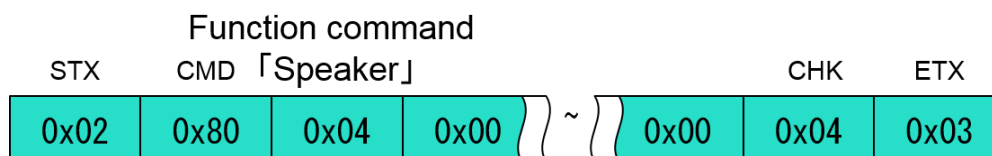


Figure 1-7 SPK function setting example

2 Function

All modes of BLT, IR RX, and SPK except for the default mode transmit ASCII values "1" ~ "6" to the master board with the command CMD_LED in touch.

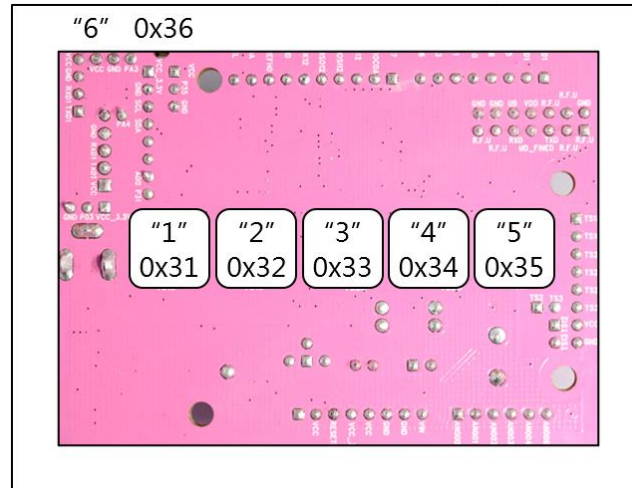


Figure 2-1 Output data

2.1 Default mode

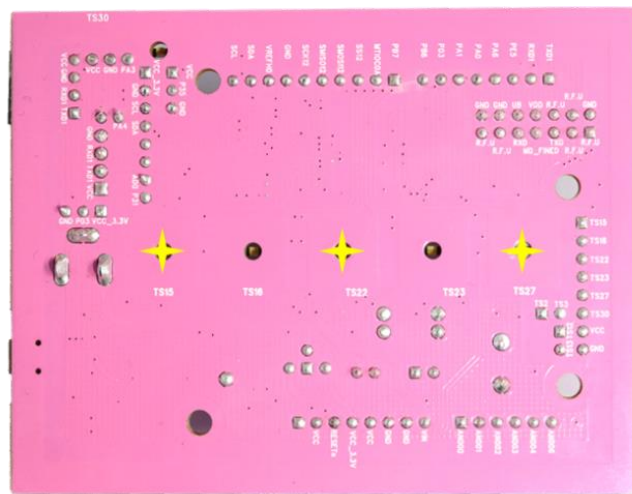


Figure 2-2 Entering the default mode

The default mode is set when the master board is not connected or the master board is not recognized due to a communication error. When entering the default mode, LED 1, 3, 5 flashes to inform the user. When touching in the default mode, ASCII values "1" ~ "6" are output by **UART communication** through TXD1, RXD1 digital pin or J19, J5 connector.

2.2 Infrared Transmission (IR TX)

To IRTX, put command 0x01 and the example source performs the following actions.

`requestIR(uint8_t customData, uint8_t data)`



Figure 2-3 Description of operation



Figure 2-4 Example of transferring custom code 0x08 and data code 0x04

When a touch event occurs, the NS-RX231 transmits the touch data to the master board. The master board receives the touch data and transmits the custom code and data code to the NS-RX231 using the requestIR function. The NS-RX231 uses the IR data received from the master board to output signals from the IR LED elements.

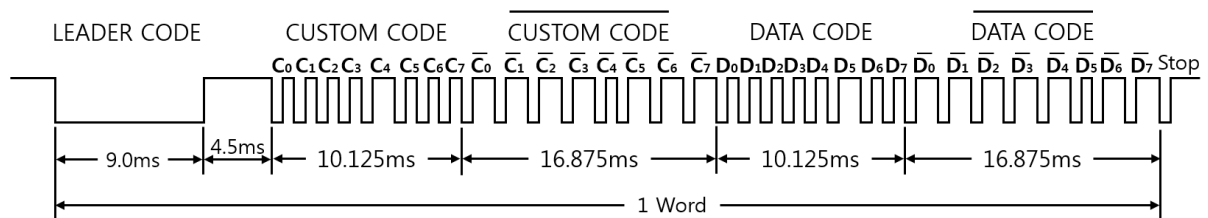


Figure 2-5 Waveform example

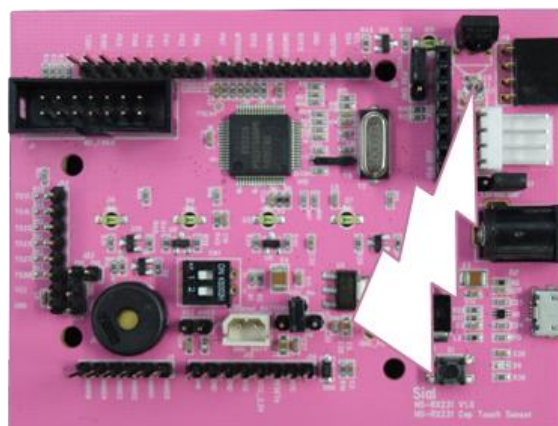


Figure 2-6 Infrared transmitter

2.3 Infrared Receive (IR RX)

To IRRX, put command 0x03, and the example source performs the following actions.

Output to serial monitor  IR data transfer

Figure 2-7 Description of operation

When the NS-RX231 receives the appropriate infrared data, it transmits infrared data to the master board and the master board outputs it to the serial monitor

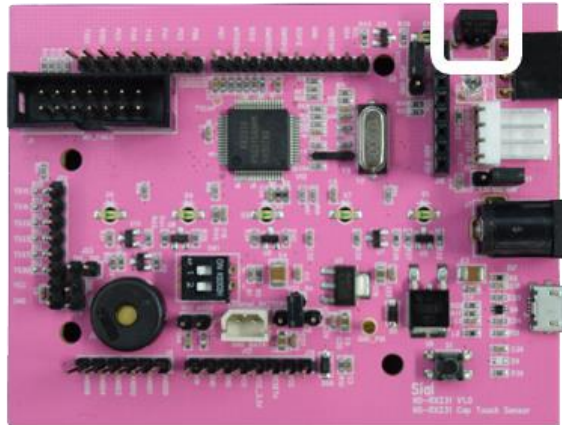


Figure 2-8 Infrared receiver

STX	CMD	Leader Code_H	Leader Code_L	Custom Code	Custom Code	Data Code	Data Code	CHK	ETX
0x02	0x03	0x05	0x46	0x08	0xF7	0x04	0x48	0x00	0x49 0x03

Figure 2-9 Example of receiving Leader code 0x0546, Custom code 0x08, Data code 0x04
(See below waveform for your reference)

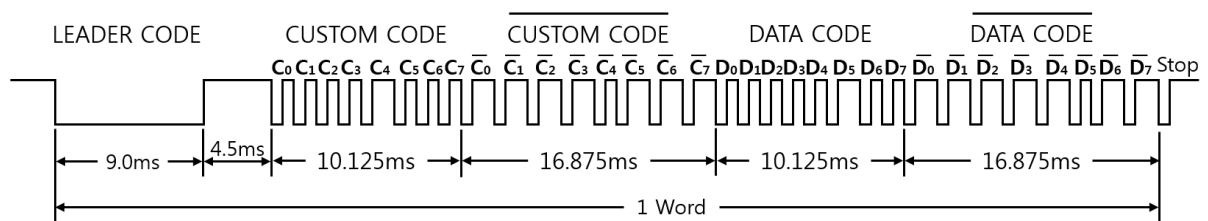


Figure 2-10 Waveform example

The leader code $9\text{ms} + 4.5\text{ms} = 13.5\text{ms} = 13500\mu\text{s}$,
 $13500\mu\text{s} / 10\mu\text{s} = 1350\text{count}(0x0546)$. (1 count is $10\mu\text{s}$)

The custom code is a value 8 (0x08) with C_0 being 0, C_1 being 0 ... 00001000, and the inverse data of custom code $\overline{C_0}$ being 1, $\overline{C_1}$ being 1 ... 11110111 and is a value of 247 (0xF7).

2.4 Speaker (SPK)

In the example program, the following operations are performed with SPK command 0x04.

requestSPK(uint16_t hz, uint16_t ms)



Figure 2-11 Description of operation

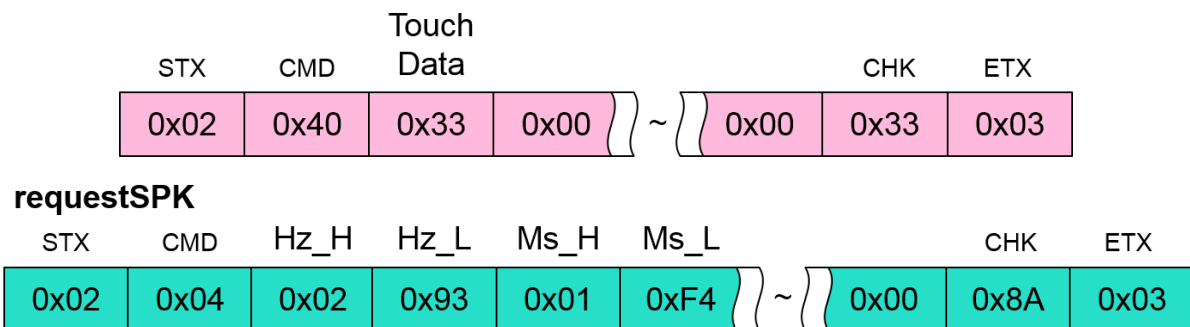


Figure 2-12 Example of 659Hz and 500ms protocol when touching TS22

When a touch event occurs, the NS-RX231 transmits the touch data to the master board, and the master board receives the touch data and transmits the frequency (Hz) and play time (Ms) to the NS-RX231 using the requestSPK function. The NS-RX231 uses the data received from the master board to output sound from the speaker device.

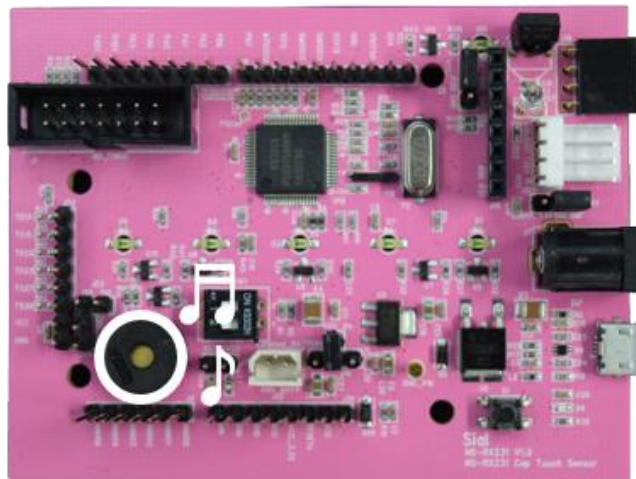


Figure 2-13 Speaker

2.5 Bluetooth (BLT)

The Bluetooth function is a function that can be used by purchasing an additional Bluetooth module.

To BLT, put command 0x10, and the example source performs the following actions.

`requestBLT(uint8_t data)`



Figure 2-14 Description of operation

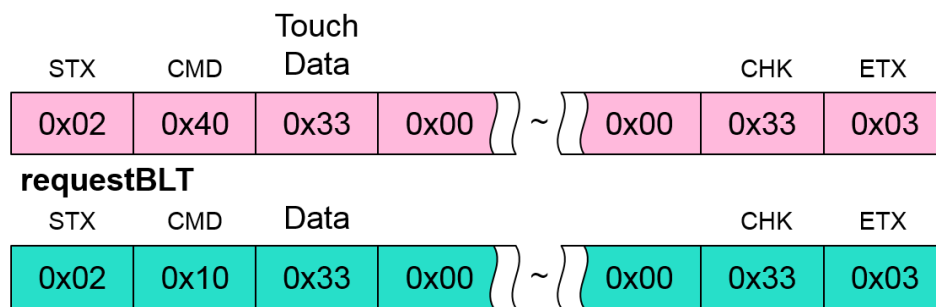


Figure 2-15 Bluetooth data informing that TS 22 has been touched

When a touch event occurs, the NS-RX231 transmits the touch data to the master board, and the master board receives the touch data and transmits the data to the NS-RX231 using the requestBLT function. The NS-RX231 uses the data received from the master board to output signals from the Bluetooth module.



Figure 2-16 Description of operation

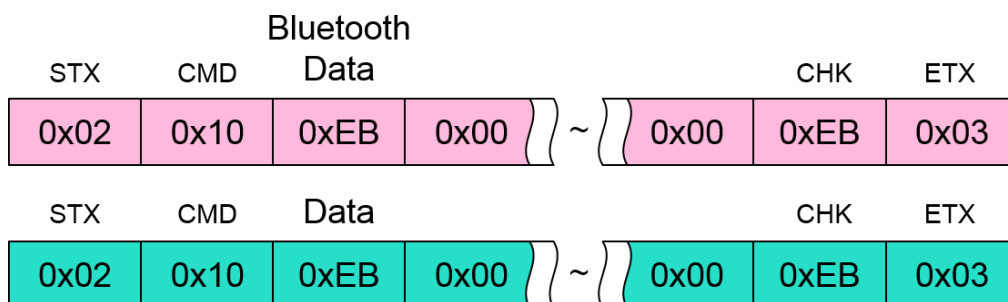


Figure 2-17 When receiving data 0xEB from the Bluetooth module installed in NS-RX231

When the Bluetooth module attached to the NS-RX231 receives data, it transmits the data to the master board. The master board receives the data and sends a code to control the LED of NS-RX231, and LEDs lights up.

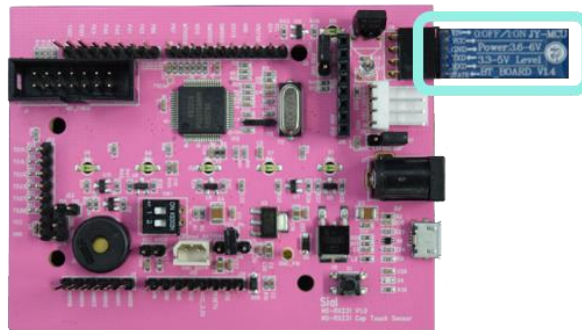


Figure 2-18 Bluetooth module

Touch TS15 to output 0x31, touch TS16 to output 0x32, ... Touch TS30 to output 0x36

2.6 MPU-6050(MPU)

The MPU function is a function that can be used by purchasing an additional MPU-6050 module.

To MPU, put command 0x20, and the example source performs the following actions.

requestMPU(void)

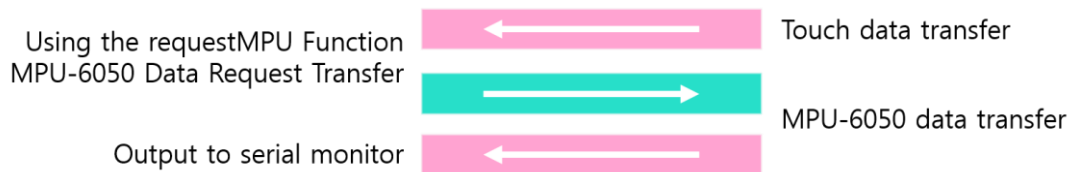


Figure 2-19 Description of operation

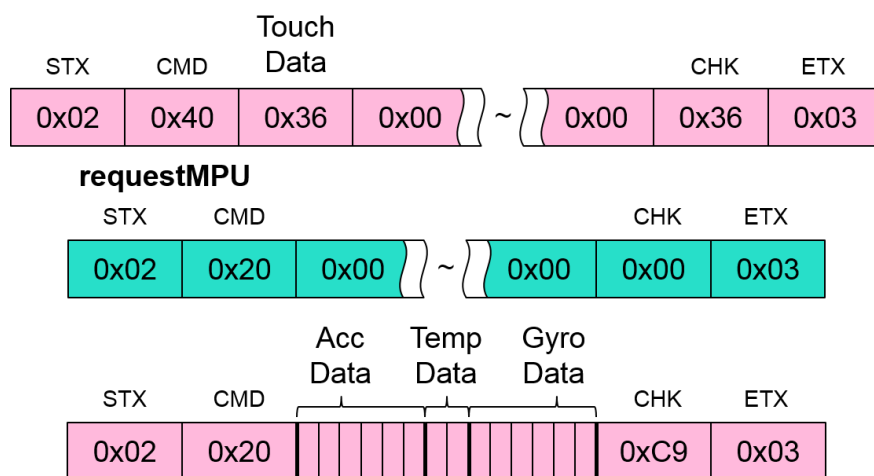


Figure 2-20 Protocol example of MPU function

When a touch event occurs, the NS-RX231 transmits touch data to the master board. If the proximity sensor TS30 of the perimeter of the NS-RX231 board is recognized, the master board requests the data of the MPU-6050 module to the NS-RX231 using the requestMPU function. The NS-RX231 transmits the MPU-6050 sensor data with the total of 14 bytes in the order of 6-byte Acceleration Value / 2-byte Temperature Sensor Value / 6-byte Gyro Sensor Value, and the master board outputs the data to the serial monitor.

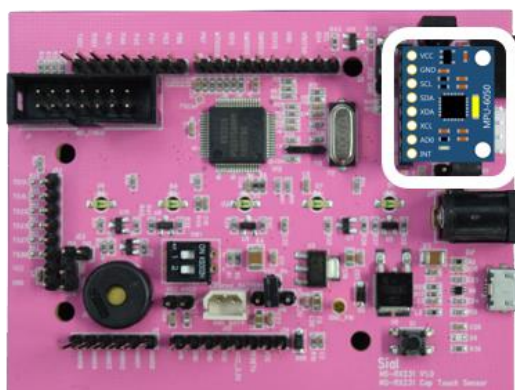


Figure 2-21 MPU-6050