

### Code Snippet Bluetooth Module Control via MCU Host

```
/******          BTM_Command.h          *****/

// Note: This is an example of some functions, not the complete version.  Need to
add/improve some functions later.

// BM83-MCU I/O
void BTM_Pin_MFB_SetLow();
void BTM_Pin_MFB_SetHigh();
void BTM_Pin_RESET_SetLow();
void BTM_Pin_RESET_SetHigh();

// UART Command Power State Control
void BTM_MFB_On_Pressed();
void BTM_MFB_On_Released();
void BTM_MFB_Off_Pressed();
void BTM_MFB_Off_Released();
void BTM_Power_ON();
void BTM_Power_OFF();
void BTM_Power_OFF_Soft();
void BTM_Power_RST();

// UART Command Operation State
void BTM_State_Standby_State();
void BTM_State_Enter_Pairing();
void BTM_State_Exit_Pairing();
void BTM_State_Disconnected_All_Profile();
void BTM_State_Mode_Inactive();
void BTM_State_Mode_Resume();
void BTM_State_Connect_A2DP();

// UART Command BTM Parameter Settings
void BTM_Set_UART_Buffersize();
void BTM_Set_Pairing_Timeout();
void BTM_Set_Device_Name();
void BTM_Set_Local_Name();

// Reset some EEPROM memory to Default value (Erase paired device information)
void BTM_Reset_Flash_Default();

// Profile Link-back
void BTM_Linkback_A2DP();
void BTM_Linkback_Lastdevice();

// Send UART Command function
uint8_t calculateChecksum(uint8_t *startByte, uint8_t *endByte);
void MCU_UART_sendData(uint8_t *Data, uint16_t dataSize);
```

```

/*****          BTM_Command.c          *****/

/*****
 *
 * BM83-MCU I/O functions
 * -MFB Pin
 * -Reset Pin
 *
 *****/

// set MFB pin to low
void BTM_Pin_MFB_SetLow()
{
    HAL_GPIO_WritePin(HCI_MFB_GPIO_Port, HCI_MFB_Pin, GPIO_PIN_RESET);
}

// set MFB pin to high
void BTM_Pin_MFB_SetHigh()
{
    HAL_GPIO_WritePin(HCI_MFB_GPIO_Port, HCI_MFB_Pin, GPIO_PIN_SET);
}

// set reset pin to low
void BTM_Pin_RESET_SetLow()
{
    HAL_GPIO_WritePin(HCI_RST_GPIO_Port, HCI_RST_Pin, GPIO_PIN_RESET);
}

// set reset pin to high
void BTM_Pin_RESET_SetHigh()
{
    HAL_GPIO_WritePin(HCI_RST_GPIO_Port, HCI_RST_Pin, GPIO_PIN_SET);
}

/*****
 * BM83 UART data packet protocol
 *
 * | Byte 0 | Byte 1-2 | Byte 3 | Byte 4 - XX | Byte (Length + 3) |
 * | Start | Length  | CMD ID | CMD Parameter| Checksum      |
 *
 * EX: uint8_t enter_pairing_cmd[7] = {0xAA,0x00,0x03,0x02,0x00,0x5D,0x9E};
 *
 * | Byte 0 | Byte 1-2 | Byte 3 | Byte 4 - XX | Byte (Length + 3) |
 * | 0xAA | 0x00 0x03 | 0x02 | 0x00 0x5D | 0x9E |
 * | Start | Length=3 | CMD=2 | Params=0,5D | CHKSUM=0x9E |
 *
 * -> Command ID 0x02 -> MMI_Action
 * -> Parameter 0x00 -> Database_index
 * -> Parameter 0x5D -> Fast enter pairing mode
 * -> Checksum 0x9E -> 1+ ~(length + Cmd + params)
 * -----> ( 03 + 02 + 5D ) = ( 62 )
 * -----> ~(0x62) = 9D
 * -----> 9D + 1 = 0x9E
 *
 *****/

```

```

uint8_t calculateChecksum(uint8_t* startByte, uint8_t* endByte)
{
    uint8_t checksum = 0;
    while(startByte <= endByte)
    {
        checksum += *startByte;
        startByte++;
    }
    checksum = ~checksum + 1;
    return checksum;
}

```

```

void MCU_UART_sendData(uint8_t *Data, uint16_t dataSize)
{
    for(uint16_t i = 0; i < dataSize; i++)
    {
        HAL_UART_Transmit(&huart1, &Data[i], 1, 50);
    }
}

```

```

// MMI Command Power on pressed
void BTM_MFB_On_Pressed()
{
    uint8_t command[7];
    command[0]=0xAA; // Header 0
    command[1]=0x00; // Header 1
    command[2]=0x03; // Payload length
    command[3]=0x02; // Command ID
    command[4]=0x00; // Database index
    command[5]=0x51; // Parameter (Action)
    command[6]=calculateChecksum(&command[2], &command[5]);
    MCU_UART_sendData(&command[0], 7);
}

```

```

// MMI Command Power on released
void BTM_MFB_On_Released()
{
    uint8_t command[7];
    command[0]=0xAA; // Header 0
    command[1]=0x00; // Header 1
    command[2]=0x03; // Payload length
    command[3]=0x02; // Command ID
    command[4]=0x00; // Database index
    command[5]=0x52; // Parameter (Action)
    command[6]=calculateChecksum(&command[2], &command[5]);
    MCU_UART_sendData(&command[0], 7);
}

```

```

// BTM Discoverable
void BTM_State_Enter_Pairing()
{

    uint8_t command[7];
    command[0]=0xAA; // Header 0
    command[1]=0x00; // Header 1
    command[2]=0x03; // Payload length
    command[3]=0x02; // Command ID
    command[4]=0x00; // Database index
    command[5]=0x5D; // Parameter (Action)
    command[6]=calculateChecksum(&command[2], &command[5]);
    MCU_UART_sendData(&command[0], 7);

}

// BTM Undiscoverable
void BTM_State_Exit_Pairing()
{

    uint8_t command[7];
    command[0]=0xAA; // Header 0
    command[1]=0x00; // Header 1
    command[2]=0x03; // Payload length
    command[3]=0x02; // Command ID
    command[4]=0x00; // Database index
    command[5]=0x6B; // Parameter (Action)
    command[6]=calculateChecksum(&command[2], &command[5]);
    MCU_UART_sendData(&command[0], 7);

}

// Set BTM Device name
void BTM_Set_Device_Name()
{

    uint8_t command[19];
    command[0]=0xAA; // Header 0
    command[1]=0x00; // Header 1
    command[2]=0x0E; // Payload length
    command[3]=0x05; // Command ID
    command[4]=0x43; // 'C'
    command[5]=0x6C; // 'l';
    command[6]=0x6F; // 'o'
    command[7]=0x75; // 'u'
    command[8]=0x64; // 'd'
    command[9]=0x20; // ' '
    command[10]=0x42; // 'B'
    command[11]=0x54; // 'T'
    command[12]=0x2D; // '-'
    command[13]=0x31; // '1'
    command[14]=0x20; // ' '
    command[15]=0x23; // '#'
    command[16]=0x31; // '1'
    command[17]=0x20; // ' ' // reserved space for 2 digits ID character

```

```

        command[18]=calculateChecksum(&command[2], &command[17]);
        MCU_UART_sendData(&command[0], 18);
    }

    // Reset some EEPROM memory to Default value (Erase paired device information)
    void BTM_Reset_Flash_Default()
    {
        uint8_t command[7];
        command[0]=0xAA; // Header 0
        command[1]=0x00; // Header 1
        command[2]=0x03; // Payload length
        command[3]=0x02; // Command ID
        command[4]=0x00; // Database index
        command[5]=0x56; // Parameter (Action)
        command[6]=calculateChecksum(&command[2], &command[5]);
        MCU_UART_sendData(&command[0], 7);
    }

```