**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.

To do next:

-implement DIP\_SW work with UART

- add a pattern of operating on mode secure/non-secure

-combine all functions module (DIPSwitch,P2P,BTM\_Command)

// 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

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Calculate checksum by sending UART command packet with start Byte is (Payload Byte) to the end byte of the data packet

uint8\_t **calculateChecksum**(uint8\_t\* startByte, uint8\_t\* endByte)

{

uint8\_t checksum = 0;

**while**(startByte <= endByte)

{

checksum += \*startByte;

startByte++;

}

checksum = ~checksum + 1;

**return** checksum;

}

// sending UART command packet via HAL UART API

// sendData(Pointer to Byte 0 of UART Packet, sizeofPacket)

**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);

}