

前言

本文档的目的是介绍 BlueNRG-MS Bluetooth® 低功耗（LE）栈应用指令接口（ACI）的设计。本文档指定了 BlueNRG-MS ACI 支持的指令列表。

目录

1	概述	15
1.1	BlueNRG-MS ACI 架构	16
2	ACI 指令数据格式	18
2.1	概述	18
2.2	HCI 指令数据包	18
2.3	HCI 事件数据包	20
2.4	HCI ACL 数据包	21
3	标准 HCI 指令	22
3.1	支持的 HCI 指令	22
3.2	支持的 HCI 事件	23
3.3	正确使用 HCI 指令	24
4	厂商特定指令	25
4.1	VS 指令和 VS 事件格式	25
4.2	L2CAP VS 指令	27
4.2.1	Aci_L2CAP_Connection_Parameter_Update_Request	27
4.2.2	Aci_L2CAP_Connection_Parameter_Update_Response	28
4.3	L2CAP VS 事件	29
4.3.1	Evt_Blue_L2CAP_Conn_Upd_Resp	30
4.3.2	Evt_Blue_L2CAP_Procedure_Timeout	30
4.3.3	Evt_Blue_L2CAP_Conn_Upd_Req	30
4.4	GAP VS 指令	32
4.4.1	概述	32
4.4.2	GAP VS 指令	33
4.4.3	Aci_Gap_Set_Non_Discoverable	34
4.4.4	Aci_Gap_Set_Limited_Discoverable	34
4.4.5	Aci_Gap_Set_Discoverable	37
4.4.6	Aci_Gap_Set_Direct_Connectable	39
4.4.7	Aci_Gap_Set_IO_Capability	40
4.4.8	Aci_Gap_Set_Auth_Requirement	41
4.4.9	Aci_Gap_Set_Author_Requirement	42

4.4.10	Aci_Gap_Pass_Key_Response	43
4.4.11	Aci_Gap_Authorization_Response	43
4.4.12	Aci_Gap_Init	44
4.4.13	Aci_Gap_Set_Non_Connectable	45
4.4.14	Aci_Gap_Set_Undirected_Connectable	46
4.4.15	Aci_Gap_Slave_Security_Request	48
4.4.16	Aci_Gap_Update_Adv_Data	49
4.4.17	Aci_Gap_Delete_AD_Type	49
4.4.18	Aci_Gap_Get_Security_Level	50
4.4.19	Aci_Gap_Set_Event_Mask	51
4.4.20	Aci_Gap_Configure_WhiteList	52
4.4.21	Aci_Gap_Terminate	52
4.4.22	Aci_Gap_Clear_Security_Database	53
4.4.23	Aci_Gap_Allow_Rebond	54
4.4.24	Aci_Gap_Start_Limited_Discovery_Proc	54
4.4.25	Aci_Gap_Start_General_Discovery_Proc	56
4.4.26	Aci_Gap_Start_Name_Discovery_Proc	57
4.4.27	Aci_Gap_Start_Auto_Conn_Establishment	59
4.4.28	Aci_Gap_Start_General_Conn_Establishment	61
4.4.29	Aci_Gap_Start_Selective_Conn_Establishment	62
4.4.30	Aci_Gap_Create_Connection	64
4.4.31	Aci_Gap_Terminate_Gap_Procedure	66
4.4.32	Aci_Gap_Start_Connection_Update	67
4.4.33	Aci_Gap_Send_Pairing_Request	68
4.4.34	Aci_Gap_Resolve_Private_Address	69
4.4.35	Aci_Gap_Get_Bonded_Devices	70
4.4.36	Aci_Gap_Set_Broadcast_Mode	71
4.4.37	Aci_Gap_Start_Observation_Proc	72
4.4.38	Aci_Gap_Is_Device_Bonded	73
4.5	GAP VS 事件	74
4.5.1	Evt_Blue_Gap_Limited_Discoverable	75
4.5.2	Evt_Blue_Gap_Pairing_Cmplt	75
4.5.3	Evt_Blue_Gap_Pass_Key_Request	75
4.5.4	Evt_Blue_Gap_Authorization_Request	76
4.5.5	Evt_Blue_Gap_Slave_Security_Initiated	76
4.5.6	Evt_Blue_Gap_Bond_Lost	76
4.5.7	Evt_Blue_Gap_Procedure_Complete	76

4.5.8	Evt_Blue_Gap_Addr_Not_Resolved	77
4.6	GATT VS 指令	77
4.6.1	Aci_Gatt_Init	79
4.6.2	Aci_Gatt_Add_Serv	79
4.6.3	Aci_Gatt_Include_Service	80
4.6.4	Aci_Gatt_Add_Char	81
4.6.5	Aci_Gatt_Add_Char_Desc	83
4.6.6	Aci_Gatt_Update_Char_Value	85
4.6.7	Aci_Gatt_Del_Char	86
4.6.8	Aci_Gatt_Del_Service	87
4.6.9	Aci_Gatt_Del_Include_Service	88
4.6.10	Aci_Gatt_Set_Event_Mask	88
4.6.11	Aci_Gatt_Exchange_Configuration	89
4.6.12	Aci_Att_Find_Information_Req	90
4.6.13	Att_Find_By_Type_Value_Req	91
4.6.14	Aci_Att_Read_By_Type_Req	92
4.6.15	Aci_Att_Read_By_Group_Type_Req	93
4.6.16	Aci_Att_Prepare_Write_Req	94
4.6.17	Aci_Att_Execute_Write_Req	95
4.6.18	Aci_Gatt_Disc_All_Prim_Services	96
4.6.19	Aci_Gatt_Disc_Prim_Service_By_UUID	97
4.6.20	Aci_Gatt_Find_Included_Services	98
4.6.21	Aci_Gatt_Disc_All_Charac_Of_Serv	99
4.6.22	Aci_Gatt_Disc_Charac_By_UUID	100
4.6.23	Aci_Gatt_Disc_All_Charac_Descriptors	101
4.6.24	Aci_Gatt_Read_Charac_Val	102
4.6.25	Aci_Gatt_Read_Charac_Using_UUID	103
4.6.26	Aci_Gatt_Read_Long_Charac_Val	104
4.6.27	Aci_Gatt_Read_Multiple_Charac_Val	105
4.6.28	Aci_Gatt_Write_Charac_Value	106
4.6.29	Aci_Gatt_Write_Long_Charac_Val	107
4.6.30	Aci_Gatt_Write_Charac_Reliable	108
4.6.31	Aci_Gatt_Write_Long_Charac_Desc	109
4.6.32	Aci_Gatt_Read_Long_Charac_Desc	110
4.6.33	Aci_Gatt_Write_Charac_Descriptor	111
4.6.34	Aci_Gatt_Read_Charac_Desc	112
4.6.35	Aci_Gatt_Write_Without_Response	112

4.6.36	Aci_Gatt_Signed_Write_Without_Resp	113
4.6.37	Aci_Gatt_Confirm_Indication	114
4.6.38	Aci_Gatt_Write_Response	115
4.6.39	Aci_Gatt_Allow_Read	116
4.6.40	Aci_Gatt_Set_Security_Permission	117
4.6.41	Aci_Gatt_Set_Desc_Value	118
4.6.42	Aci_Gatt_Read_Handle_Value	119
4.6.43	Aci_Gatt_Read_Handle_Value_Offset	120
4.6.44	Aci_Update_Char_Value_Ext	121
4.7	GATT VS 事件	123
4.7.1	Evt_Blue_Gatt_Attribute_modified	123
4.7.2	Evt_Blue_Gatt_Procedure_Timeout	124
4.7.3	Evt_Blue_Gatt_Procedure_Complete	124
4.7.4	Evt_Blue_Gatt_Disc_Read_Charac_By_UUID_Resp	124
4.7.5	Evt_Blue_Gatt_Write_Permit_req	125
4.7.6	Evt_Blue_Gatt_Read_Permit_Req	125
4.7.7	Evt_Blue_Gatt_Read_Multi_Permit_Req	126
4.7.8	Evt_Blue_Gatt_Tx_Pool_Available	126
4.7.9	Evt_Blue_Att_Exchange_MTU_Resp	127
4.7.10	Evt_Blue_Att_Find_Information_Resp	127
4.7.11	Evt_Blue_Att_Find_By_Type_Value_Resp	127
4.7.12	Evt_Blue_Att_Read_By_Type_Resp	128
4.7.13	Evt_Blue_Att_Read_Resp	128
4.7.14	Evt_Blue_Att_Read_Blob_Resp	128
4.7.15	Evt_Blue_Att_Read_Multiple_Resp	129
4.7.16	Evt_Blue_Att_Read_By_Group_Type_Resp	129
4.7.17	Evt_Blue_Att_Prepare_Write_Resp	129
4.7.18	Evt_Blue_Att_Exec_Write_Resp	130
4.7.19	Evt_Blue_Gatt_Indication	130
4.7.20	Evt_Blue_Gatt_notification	130
4.7.21	Evt_Blue_Gatt_Error_Resp	131
4.7.22	Evt_Gatt_Server_Confirmation	131
4.8	HCI 厂商特定指令	132
4.8.1	Aci_Hal_Write_Config_Data	132
4.8.2	Aci_Hal_Read_Config_Data	134
4.8.3	Aci_Hal_Set_Tx_Power_Level	134
4.8.4	Aci_Hal_Device_Standby	136

4.8.5	Aci_Hal_LE_Tx_Test_Packet_Number	136
4.8.6	Aci_Hal_Tone_Start	137
4.8.7	Aci_Hal_Tone_Stop	138
4.8.8	Aci_Hal_Get_Link_Status	138
4.8.9	Aci_Hal_Get_Fw_Build_Number	139
4.8.10	Aci_Hal_Get_Anchor_Period	139
4.9	HCI VS 事件	140
4.9.1	Evt_Blue_Initialized event	140
4.9.2	Evt_Blue_Lost_Events	141
4.9.3	故障数据事件	142
4.9.4	硬件错误事件代码	143
5	SPI 接口	144
5.1	硬件 SPI 接口	144
5.2	SPI 通信协议	146
5.2.1	数据头	146
5.2.2	向 BlueNRG-MS 写入数据	147
5.2.3	从 BlueNRG-MS 读取数据	147
5.2.4	BlueNRG-MS 睡眠模式下的 SPI 操作	148
6	版本历史	149

表格索引

表 1.	OGF 值	19
表 2.	HCI 指令	22
表 3.	HCI 事件	23
表 4.	CGID 组	25
表 5.	EGID 组	26
表 6.	L2CAP VS 指令	27
表 7.	Aci_L2CAP_Connection_Parameter_Update_Request	27
表 8.	Aci_L2CAP_Connection_Parameter_Update_Requests 指令参数	27
表 9.	Aci_L2CAP_Connection_Parameter_Update_Requests 返回参数	28
表 10.	Aci_L2CAP_Connection_Parameter_Update_Response	28
表 11.	Aci_L2CAP_Connection_Parameter_Update_Response 指令参数	28
表 12.	Aci_L2CAP_Connection_Parameter_Update_Response 返回参数	29
表 13.	L2CAP VS 事件	29
表 14.	Evt_Blue_L2CAP_Conn_Upd_Resp	30
表 15.	Evt_Blue_L2CAP_Procedure_Timeout	30
表 16.	Evt_Blue_L2CAP_Conn_Upd_Req	31
表 17.	GAP VS 指令	33
表 18.	Aci_gap_set_non_discoverable	34
表 19.	Aci_gap_set_non_discoverable 返回参数	34
表 20.	Aci_Gap_Set_Limited_Discoverable	34
表 21.	Aci_Gap_Set_Limited_Discoverable 指令参数	35
表 22.	Aci_Gap_Set_Limited_Discoverable 返回参数	36
表 23.	Aci_Gap_Set_Discoverable	37
表 24.	Aci_Gap_Set_Discoverable 指令参数	37
表 25.	Aci_Gap_Set_Discoverable 返回参数	38
表 26.	Aci_Gap_Set_Direct_Connectable	39
表 27.	Aci_Gap_Set_Direct_Connectable 指令参数	39
表 28.	Aci_Gap_Set_Direct_Connectable 返回参数	40
表 29.	Aci_Gap_Set_IO_Capability	40
表 30.	Aci_Gap_Set_IO_Capability 指令参数	40
表 31.	Aci_Gap_Set_IO_Capability 返回参数	40
表 32.	Aci_Gap_Set_Auth_Requirement	41
表 33.	Aci_Gap_Set_Auth_Requirement 指令参数	41
表 34.	Aci_Gap_Set_Auth_Requirement 返回参数	42
表 35.	Aci_Gap_Set_Author_Requirement	42
表 36.	Aci_Gap_Set_Author_Requirement 指令参数	42
表 37.	Aci_Gap_Set_Author_Requirement 返回参数	42
表 38.	Aci_Gap_Pass_Key_Response	43
表 39.	Aci_Gap_Pass_Key_Response 指令参数	43
表 40.	Aci_Gap_Pass_Key_Response 返回参数	43
表 41.	Aci_Gap_Authorization_Response	43
表 42.	Aci_Gap_Authorization_Response 指令参数	44
表 43.	Aci_Gap_Authorization_Response 返回参数	44
表 44.	Aci_Gap_Init	44
表 45.	Aci_Gap_Init 指令参数	44
表 46.	Aci_Gap_Init 返回参数	45
表 47.	Aci_Gap_Set_Non_Connectable	45
表 48.	Aci_Gap_Set_Non_Connectable 指令参数	45

表 49.	Aci_Gap_Set_Non_Connectable 返回参数	46
表 50.	Aci_Gap_Set_Undirected_Connectable	46
表 51.	Aci_Gap_Set_Undirected_Connectable 指令参数	46
表 52.	Aci_Gap_Set_Undirected_Connectable 返回参数	46
表 53.	Aci_Gap_Slave_Security_Request	48
表 54.	Aci_Gap_Slave_Security_Request 指令参数	48
表 55.	Aci_Gap_Slave_Security_Request 返回参数	48
表 56.	Aci_Gap_Update_Adv_Data	49
表 57.	Aci_Gap_Update_Adv_Data 指令参数	49
表 58.	Aci_Gap_Update_Adv_Data 返回参数	49
表 59.	Aci_Gap_Delete_AD_Type	49
表 60.	Aci_Gap_Delete_AD_Type 指令参数	50
表 61.	Aci_Gap_Delete_AD_Type 返回参数	50
表 62.	Aci_Gap_Get_Security_Level	50
表 63.	Aci_Gap_Get_Security_Level 返回参数	51
表 64.	Aci_Gap_Set_Event_Mask	51
表 65.	Aci_Gap_Set_Event_Mask 指令参数	51
表 66.	Aci_Gap_Set_Event_Mask 返回参数	51
表 67.	Aci_Gap_Configure_WhiteList	52
表 68.	Aci_Gap_Configure_WhiteList 返回参数	52
表 69.	Aci_Gap_Terminate	52
表 70.	Aci_Gap_Terminate 指令参数	52
表 71.	Aci_Gap_Terminate 返回参数	53
表 72.	Aci_Gap_Clear_Security_Database	53
表 73.	Aci_Gap_Clear_Security_Database 返回参数	53
表 74.	Aci_Gap_Allow_Rebond	54
表 75.	Aci_Gap_Allow_Rebond 指令参数	54
表 76.	Aci_Gap_Allow_Rebond 返回参数	54
表 77.	Aci_Gap_Start_Limited_Discovery_Proc	54
表 78.	Aci_Gap_Start_Limited_Discovery_Proc 指令参数	55
表 79.	Aci_Gap_Start_Limited_Discovery_Proc 返回参数	55
表 80.	Aci_Gap_Start_General_Discovery_Proc	56
表 81.	Aci_Gap_Start_General_Discovery_Proc 指令参数	56
表 82.	Aci_Gap_Start_General_Discovery_Proc 返回参数	56
表 83.	Aci_Gap_Start_Name_Discovery_Proc	57
表 84.	Aci_Gap_Start_Name_Discovery_Proc 指令参数	57
表 85.	Aci_Gap_Start_Name_Discovery_Proc 返回参数	58
表 86.	Aci_Gap_Start_Auto_Conn_Establishment	59
表 87.	Aci_Gap_Start_Auto_Conn_Establishment 指令参数	59
表 88.	Aci_Gap_Start_Auto_Conn_Establishment 返回参数	60
表 89.	Aci_Gap_Start_General_Conn_Establishment	61
表 90.	Aci_Gap_Start_General_Conn_Establishment 指令参数	61
表 91.	Aci_Gap_Start_General_Conn_Establishment 返回参数	62
表 92.	Aci_Gap_Start_Selective_Conn_Establishment	62
表 93.	Aci_Gap_Start_Selective_Conn_Establishment 指令参数	62
表 94.	Aci_Gap_Start_General_Conn_Establishment 返回参数	63
表 95.	Aci_Gap_Create_Connection	64
表 96.	Aci_Gap_Create_Connection 指令参数	64
表 97.	Aci_Gap_Create_Connection 返回参数	65
表 98.	Aci_Gap_Terminate_Gap_Procedure	66
表 99.	Aci_Gap_Terminate_Gap_Procedure 指令参数	66
表 100.	Aci_Gap_Terminate_Gap_Procedure 返回参数	66

表 101.	Aci_Gap_Start_Connection_Update	67
表 102.	Aci_Gap_Start_Connection_Update 指令参数	67
表 103.	Aci_Gap_Start_Connection_Update 返回参数	68
表 104.	Aci_Gap_Send_Pairing_Request	68
表 105.	Aci_Gap_Send_Pairing_Request 指令参数	69
表 106.	Aci_Gap_Send_Pairing_Request 返回参数	69
表 107.	Aci_Gap_Resolve_Private_Address	69
表 108.	Aci_Gap_Resolve_Private_Address 指令参数	70
表 109.	Aci_Gap_Resolve_Private_Address 返回参数	70
表 110.	Aci_Gap_Get_Bonded_Devices	70
表 111.	Aci_Gap_Get_Bonded_Devices 返回参数	70
表 112.	Aci_Gap_Set_Broadcast_Mode	71
表 113.	Aci_Gap_Set_Broadcast_Mode 指令参数	71
表 114.	Aci_Gap_Set_Broadcast_Mode 返回参数	72
表 115.	Aci_Gap_Start_Observation_Proc	72
表 116.	Aci_Gap_Start_Observation_Proc 指令参数	73
表 117.	Aci_Gap_Start_Observation_Proc 返回参数	73
表 118.	Aci_Gap_Is_Device_Bonded	73
表 119.	Aci_Gap_Is_Device_Bonded 指令参数	74
表 120.	Aci_Gap_Is_Device_Bonded 返回参数	74
表 121.	GAP VS 事件	74
表 122.	Evt_Blue_Gap_Pairing_Cmplt	75
表 123.	Evt_Blue_Gap_Pass_Key_Request	75
表 124.	Evt_Blue_Gap_Authorization_Request	76
表 125.	Evt_Blue_Gap_Procedure_Complete	76
表 126.	Evt_Blue_Gap_Addr_Not_Resolved	77
表 127.	GATT VS 指令	77
表 128.	Aci_Gatt_Init	79
表 129.	Aci_Gatt_Init 返回参数	79
表 130.	Aci_Gatt_Add_Serv	79
表 131.	Aci_Gatt_Add_Serv 指令参数	79
表 132.	Aci_Gatt_Add_Serv 返回参数	80
表 133.	Aci_Gatt_Include_Service	80
表 134.	Aci_Gatt_Include_Service 指令参数	81
表 135.	Aci_Gatt_Include_Service 返回参数	81
表 136.	Aci_Gatt_Add_Char	81
表 137.	Aci_Gatt_Add_Char 指令参数	82
表 138.	Aci_Gatt_Add_Char 返回参数	83
表 139.	Aci_Gatt_Add_Char_Desc	83
表 140.	Aci_Gatt_Add_Char_Desc 指令参数	84
表 141.	Aci_Gatt_Add_Char_Desc 返回参数	85
表 142.	Aci_Gatt_Update_Char_Value	85
表 143.	Aci_Gatt_Update_Char_Value 指令参数	85
表 144.	Aci_Gatt_Update_Char_Value 返回参数	86
表 145.	Aci_Gatt_Del_Char	86
表 146.	Aci_Gatt_Del_Char 指令参数	86
表 147.	Aci_Gatt_Del_Char 返回参数	87
表 148.	Aci_Gatt_Del_Service	87
表 149.	Aci_Gatt_Del_Service 指令参数	87
表 150.	Aci_Gatt_Del_Service 返回参数	87
表 151.	Aci_Gatt_Del_Include_Service	88
表 152.	Aci_Gatt_Del_Include_Service 指令参数	88

表 153.	Aci_Gatt_Del_Include_Service 返回参数	88
表 154.	Aci_Gatt_Set_Event_Mask	88
表 155.	Aci_Gatt_Set_Event_Mask 指令参数	89
表 156.	Aci_Gatt_Set_Event_Mask 返回参数	89
表 157.	Aci_Gatt_Exchange_Configuration	89
表 158.	Aci_Gatt_Exchange_Configuration 指令参数	90
表 159.	Aci_Gatt_Exchange_Configuration 返回参数	90
表 160.	Aci_Att_Find_Information_Req	90
表 161.	Aci_Att_Find_Information_Req 指令参数	90
表 162.	Aci_Att_Find_Information_Req 返回参数	91
表 163.	Att_Find_By_Type_Value_Req	91
表 164.	Aci_Att_Find_Attr_By_Typ_And_Value_Req 指令参数	91
表 165.	Aci_Att_Find_Attr_By_Typ_And_Value_Req 返回参数	92
表 166.	Aci_Att_Read_By_Type_Req	92
表 167.	Aci_Att_Read_By_Type_Req 指令参数	93
表 168.	Aci_Att_Read_By_Type_Req 返回参数	93
表 169.	Aci_Att_Read_By_Group_Type_Req	93
表 170.	Aci_Att_Read_By_Group_Type_Req 指令参数	94
表 171.	Aci_Att_Read_By_Group_Type_Req 返回参数	94
表 172.	Aci_Att_Prepere_Write_Req	94
表 173.	Aci_Att_Prepere_Write_Req 指令参数	95
表 174.	Aci_Att_Prepere_Write_Req 返回参数	95
表 175.	Aci_Gatt_Execute_Write_Req	95
表 176.	Aci_Att_Execute_Write_Req 指令参数	96
表 177.	Aci_Att_Execute_Write_Req 返回参数	96
表 178.	Aci_Gatt_Disc_All_Prim_Services	96
表 179.	Aci_Gatt_Disc_All_Prim_Services 指令参数	96
表 180.	Aci_Gatt_Disc_All_Prim_Services 返回参数	97
表 181.	Aci_Gatt_Disc_Prim_Service_By_UUID	97
表 182.	Aci_Gatt_Disc_Prim_Service_By_UUID 指令参数	97
表 183.	Aci_Gatt_Disc_Prim_Service_By_UUID 返回参数	98
表 184.	Aci_Gatt_Find_Included_Services	98
表 185.	Aci_Gatt_Find_Included_Services 指令参数	98
表 186.	Aci_Gatt_Find_Included_Services 返回参数	99
表 187.	Aci_Gatt_Disc_All_Charac_Of_Serv	99
表 188.	Aci_Gatt_Disc_All_Charac_Of_Serv 指令参数	99
表 189.	Aci_Gatt_Disc_All_Charac_Of_Serv 返回参数	100
表 190.	Aci_Gatt_Disc_Charac_By_UUID	100
表 191.	Aci_Gatt_Disc_Charac_By_UUID 指令参数	100
表 192.	Aci_Gatt_Disc_Charac_By_UUID 返回参数	101
表 193.	Aci_Gatt_Disc_All_Charac_Descriptors	101
表 194.	Aci_Gatt_Disc_All_Charac_Descriptors 指令参数	101
表 195.	Aci_Gatt_Disc_All_Charac_Descriptors 返回参数	102
表 196.	Aci_Gatt_Read_Charac_Val	102
表 197.	Aci_Gatt_Read_Charac_Val 指令参数	102
表 198.	Aci_Gatt_Read_Charac_Val 返回参数	102
表 199.	Aci_Gatt_Read_Charac_Using_UUID	103
表 200.	Aci_Gatt_Read_Charac_Using_UUID 指令参数	103
表 201.	Aci_Gatt_Read_Charac_Using_UUID 返回参数	103
表 202.	Aci_Gatt_Read_Long_Charac_Val	104
表 203.	Aci_Gatt_Read_Long_Charac_Val 指令参数	104
表 204.	Aci_Gatt_Read_Long_Charac_Val 指令返回参数	104

表 205.	Aci_Gatt_Read_Multiple_Charac_Val	105
表 206.	Aci_Gatt_Read_Multiple_Charac_Val 指令参数	105
表 207.	Aci_Gatt_Read_Multiple_Charac_Val 返回参数	105
表 208.	Aci_Gatt_Write_Charac_Value	106
表 209.	Aci_Gatt_Write_Charac_Value 指令参数	106
表 210.	Aci_Gatt_Write_Charac_Value 返回参数	106
表 211.	Aci_Gatt_Write_Long_Charac_Val	107
表 212.	Aci_Gatt_Write_Long_Charac_Val 指令参数	107
表 213.	Aci_Gatt_Write_Long_Charac_Val 返回参数	107
表 214.	Aci_Gatt_Write_Charac_Reliable	108
表 215.	Aci_Gatt_Write_Charac_Reliable 指令参数	108
表 216.	Aci_Gatt_Write_Charac_Reliable 返回参数	108
表 217.	Aci_Gatt_Write_Long_Charac_Desc	109
表 218.	Aci_Gatt_Write_Long_Charac_Desc 指令参数	109
表 219.	Aci_Gatt_Write_Long_Charac_Desc 返回参数	109
表 220.	Aci_Gatt_Read_Long_Charac_Desc	110
表 221.	Aci_Gatt_Read_Long_Charac_Desc 指令参数	110
表 222.	Aci_Gatt_Read_Long_Charac_Desc 返回参数	110
表 223.	Aci_Gatt_Write_Charac_Descriptor	111
表 224.	Aci_Gatt_Write_Charac_Descriptor 指令参数	111
表 225.	Aci_Gatt_Write_Charac_Descriptor 返回参数	111
表 226.	Aci_Gatt_Read_Charac_Desc	112
表 227.	Aci_Gatt_Read_Charac_Desc 指令参数	112
表 228.	Aci_Gatt_Read_Charac_Desc 返回参数	112
表 229.	Aci_Gatt_Write_Without_Response	112
表 230.	Aci_Gatt_Write_Without_Response 指令参数	113
表 231.	Aci_Gatt_Write_Without_Response 返回参数	113
表 232.	Aci_Gatt_Signed_Write_Without_Resp	113
表 233.	Aci_Gatt_Signed_Write_Without_Resp 指令参数	114
表 234.	Aci_Gatt_Signed_Write_Without_Resp 返回参数	114
表 235.	Aci_Gatt_Confirm_Indication	114
表 236.	Aci_Gatt_Confirm_Indication 指令参数	114
表 237.	Aci_Gatt_Confirm_Indication 返回参数	115
表 238.	Aci_Gatt_Write_Response	115
表 239.	Aci_Gatt_Write_Response 指令参数	115
表 240.	Aci_Gatt_Write_Response 返回参数	116
表 241.	Aci_Gatt_Allow_Read	116
表 242.	Aci_Gatt_Allow_Read 指令参数	116
表 243.	Aci_Gatt_Allow_Read 返回参数	117
表 244.	Aci_Gatt_Set_Security_Permission	117
表 245.	Aci_Gatt_Set_Security_Permission 指令参数	117
表 246.	Aci_Gatt_Set_Security_Permission 返回参数	118
表 247.	Aci_Gatt_Set_Desc_Value	118
表 248.	Aci_Gatt_Set_Desc_Value 指令参数	118
表 249.	Aci_Gatt_Set_Desc_Value 返回参数	119
表 250.	Aci_Gatt_Read_Handle_Value	119
表 251.	Aci_Gatt_Read_Handle_Value 指令参数	119
表 252.	Aci_Gatt_Read_Handle_Value 返回参数	119
表 253.	Aci_Gatt_Read_Handle_Value_Offset	120
表 254.	Aci_Gatt_Read_Handle_Value_Offset 指令参数	120
表 255.	Aci_Gatt_Read_Handle_Value_Offset 返回参数	120
表 256.	Aci_Gatt_Update_Char_Value_Ext	121

表 257.	Aci_Gatt_Update_Char_Value_Ext 指令参数	121
表 258.	Aci_Gatt_Update_Char_Value_Ext 返回参数	122
表 259.	GATT VS 事件	123
表 260.	Evt_Blue_Gatt_Attribute_modified	124
表 261.	Evt_Blue_Gatt_Procedure_Timeout	124
表 262.	Evt_Blue_Gatt_Procedure_Complete	124
表 263.	Evt_Blue_Gatt_Disc_Read_Charac_By_UUID_Resp	125
表 264.	Evt_Blue_Gatt_Write_Permit_req	125
表 265.	Evt_Blue_Gatt_Read_Permit_Req	126
表 266.	Evt_Blue_Gatt_Read_Multi_Permit_Req	126
表 267.	Evt_Blue_Gatt_Tx_Pool_Available	126
表 268.	Evt_Blue_Att_Exchange_MTU_Resp	127
表 269.	Evt_Blue_Att_Find_Information_Resp	127
表 270.	Evt_Blue_Att_Find_By_Type_Value_Resp	127
表 271.	Evt_Blue_Att_Read_By_Type_Resp	128
表 272.	Evt_Blue_Att_Read_Resp	128
表 273.	Evt_Blue_Att_Read_Blob_Resp	128
表 274.	Evt_Blue_Att_Read_Multiple_Resp	129
表 275.	Evt_Blue_Att_Read_By_Group_Type_Resp	129
表 276.	Evt_Blue_Att_Prepare_Write_Resp	129
表 277.	Evt_Blue_Att_Exec_Write_Resp	130
表 278.	Evt_Blue_Gatt_Indication	130
表 279.	Evt_Blue_Gatt_notification	130
表 280.	Evt_Blue_Gatt_Error_Resp	131
表 281.	Evt_Gatt_Server_Confirmation	131
表 282.	HCI VS 指令	132
表 283.	Aci_Hal_Write_Config_Data	132
表 284.	Aci_Hal_Write_Config_Data 指令参数	133
表 285.	Aci_Hal_Write_Config_Data 成员	133
表 286.	Aci_Hal_Write_Config_Data 返回参数	133
表 287.	Aci_Hal_Read_Config_Data	134
表 288.	Aci_Hal_Read_Config_Data 指令参数	134
表 289.	Aci_Hal_Read_Config_Data 返回参数	134
表 290.	Aci_Hal_Set_Tx_Power_Level	134
表 291.	Aci_Hal_Set_Tx_Power_Level 指令参数	135
表 292.	Tx_power_level 指令参数组合	135
表 293.	Aci_Hal_Set_Tx_Power_Level 返回参数	135
表 294.	Aci_Hal_Device_Standby	136
表 295.	Aci_Hal_Device_Standby 返回参数	136
表 296.	Aci_Hal_LE_Tx_Test_Packet_Number	136
表 297.	Aci_Hal_LE_Tx_Test_Packet_Number 返回参数	137
表 298.	Aci_Hal_Tone_Start	137
表 299.	Aci_Hal_Tone_Start 指令参数	137
表 300.	Aci_Hal_Tone_Start 返回参数	137
表 301.	Aci_Hal_Tone_Stop	138
表 302.	Aci_Hal_Tone_Stop 返回参数	138
表 303.	Aci_Hal_Get_Link_Status	138
表 304.	Aci_Hal_Get_Link_Status 返回参数	138
表 305.	Aci_Hal_Get_Fw_Build_Number	139
表 306.	Aci_Hal_Get_Fw_Build_Number 返回参数	139
表 307.	Aci_Hal_Get_Anchor_Period	139
表 308.	Aci_Hal_Get_Anchor_Period 返回参数	140

表 309.	Evt_Blue_Initialized 事件	140
表 310.	Evt_Blue_Lost_Events	141
表 311.	Lost_Events_Map (8 字节)	141
表 312.	故障数据事件	142
表 313.	SPI 引脚	144
表 314.	文档版本历史	149
表 315.	中文文档版本历史	149

图片索引

图 1. 具有单独主机和控制器的蓝牙 LE 系统 15

图 2. 具有组合主机和控制器的蓝牙 LE 系统 16

图 3. BlueNRG-MS ACI 架构 17

图 4. HCI 指令数据包 18

图 5. HCI 事件数据包 20

图 6. HCI ACL 数据包 21

图 7. VS 指令的操作码格式 25

图 8. VS 事件的事件参数 0 格式 26

图 9. SPI 连线 145

图 10. SPI 数据头格式 146



1 概述

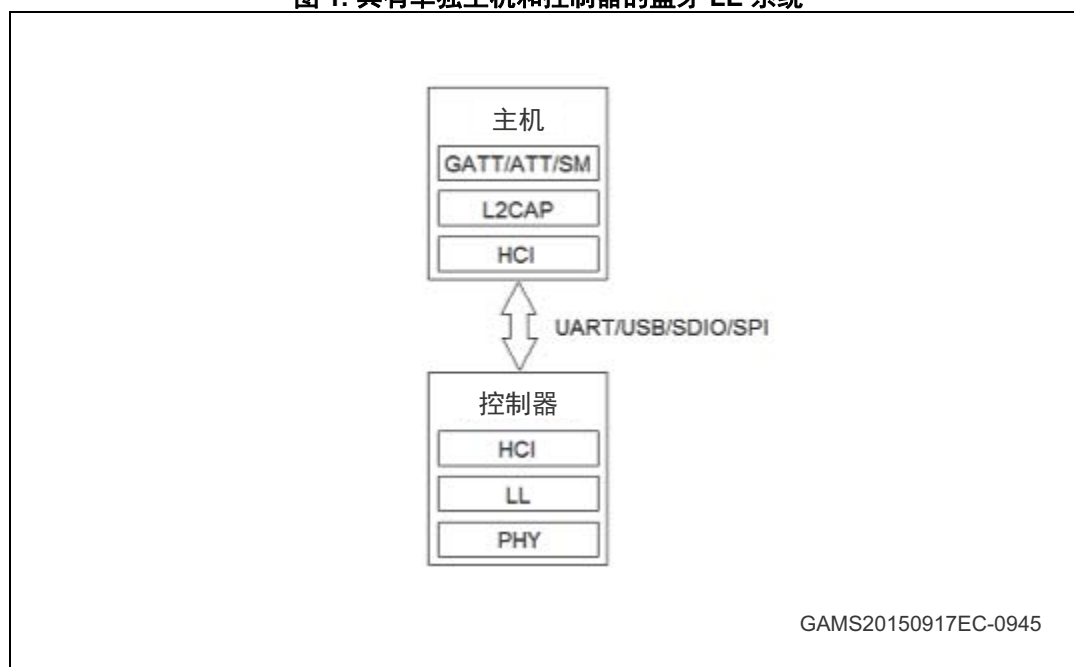
蓝牙 SIG 从蓝牙核心规范 v4.1 采纳蓝牙 LE 技术。相比于传统蓝牙或高速蓝牙系统，低功耗蓝牙技术专为要求低功耗、低复杂度和低成本的产品而设计。

典型 BLE 系统包含 LE 控制器和主机。LE 控制器由物理层（PHY）组成，包括无线设备、链路层（LL）和标准主机控制器接口（HCI）。主机包括 HCI 和其他更高协议层，例如 L2CAP、SM、ATT/GATT、GAP 等。

在许多设计中，LE 控制器和主机使用两个独立的硅芯片，由 2 个不同的微控制器控制。二者之间的通信通过硬件连接（例如 UART、SPI、USB 等）进行。主机可以发送 HCI 指令以控制 LE 控制器。蓝牙核心规范对 HCI 接口和 HCI 指令进行了标准化。欲了解详细信息，请参考官方文档。

ACI 接口具有显著优势。任何蓝牙测试仪都可以轻松地通过硬件连接（例如 SPI）连接控制器，以通过发送 HCI 指令测试控制器。因此，如果只需要测试控制器，无需主机参与。

图 1. 具有单独主机和控制器的蓝牙 LE 系统



LE 控制器和主机有时可以组合成单个芯片组解决方案。这将降低硬件成本，首先是因为硅芯片的数量从两个减少到一个，其次是主机与控制器之间不再需要硬件连接。

这种情况下，主机可以直接控制 LL/PHY。这样一来，无需生成标准 HCI 指令，将指令发送到 LL，然后处理指令。因此，节约了执行时间，并减少了 CPU 的计算工作量。但是，如果完全移除 HCI，使用外部蓝牙测试仪测试控制器将变得更加困难。

图 2. 具有组合主机和控制器的蓝牙 LE 系统



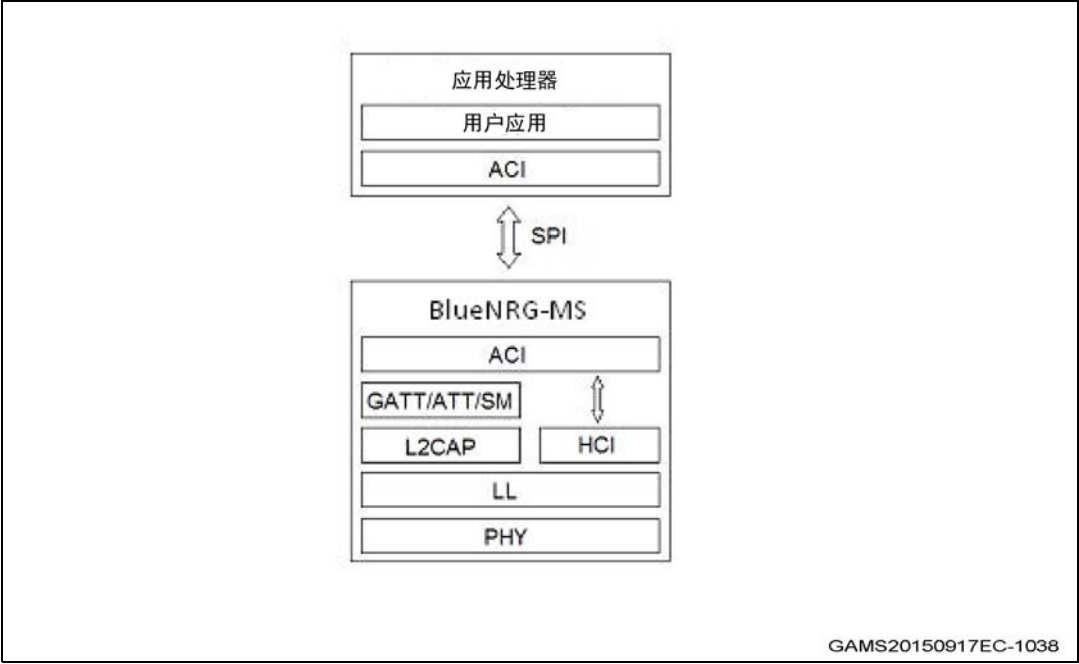
1.1 BlueNRG-MS ACI 架构

BlueNRG-MS 是使用组合主机和控制器解决方案实现的。应用指令接口（ACI）提供用于访问 BlueNRG-MS 主机和控制器。

用户应用，即在另一硅芯片上运行的程序，可以发送 ACI 指令控制 BlueNRG-MS。ACI 指令应通过 SPI 连接发送。后面一节将介绍 SPI 协议。

ACI 接口还支持 HCI 指令。在接收到指令时，ACI 将检查指令适用于主机还是控制器。如果是 HCI 指令，即控制器的指令，ACI 将指令直接转发至控制器，绕过主机。这种实现方式有两个优势：(1) 主机通常可以控制 LL/PHY，无需使用 HCI 指令，因此提高了性能；(2) 用户应用仍然可以独立测试控制器，或通过 HCI 指令设置一些底层硬件参数，无需通过主机。

图 3. BlueNRG-MS ACI 架构



2 ACI 指令数据格式

在向 BlueNRG-MS 发送 ACI 指令时，指令必须采用本章所描述的格式。

2.1 概述

BlueNRG-MS ACI 指令利用并扩展标准 HCI 数据格式。标准 HCI 数据格式是蓝牙核心规范的一部分。为避免冗长和不一致，没有将文本复制到本文档中。欲了解详细信息，请参考官方蓝牙规范第 2 卷 E 部分第 5 章。

根据蓝牙规范，标准 HCI 数据包可以是：

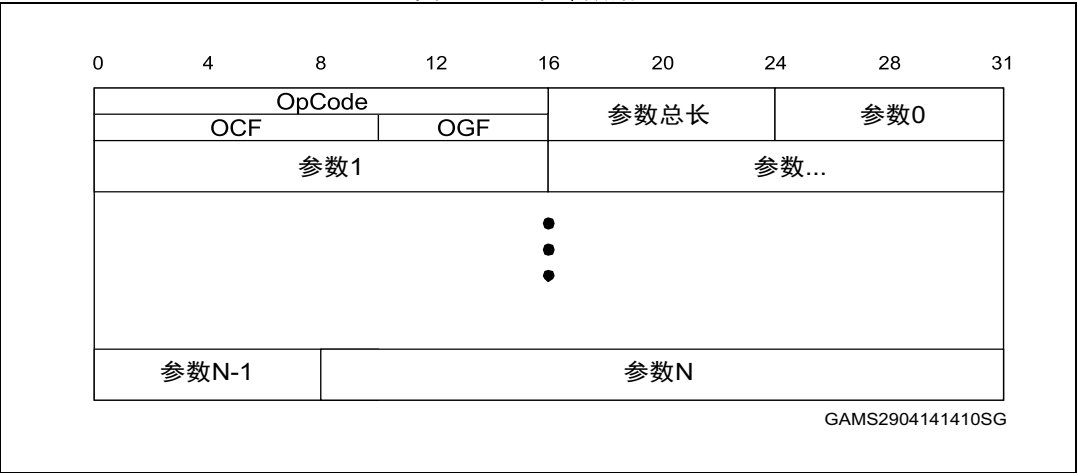
- 1. HCI 指令数据包
- 2. HCI ACL 数据包
- 3. HCI 同步数据包
- 4. HCI 事件数据包

在 BlueNRG-MS 中，不支持 HCI 同步数据包，但是支持其他三种数据包。

2.2 HCI 指令数据包

当外部设备向 BlueNRG-MS 发出指令时，指令必须采用 HCI 指令数据包的格式。BlueNRG-MS 仅接收 HCI 指令数据包，但是不发送。

图 4. HCI 指令数据包



每个 HCI 指令使用 2 字节操作码唯一标识不同类型的指令。操作码分为两个字段：操作码组字段（OGF）和操作码指令字段（OCF）。OGF 是较高的 6 位，剩余较低的 10 位由 OCF 使用。

全部 HCI 指令按蓝牙规范分成不同的逻辑组，每组分配有唯一 OGF 值。

表 1. OGF 值

组名称	OGF 值
链路控制指令	0x01
链路策略指令	0x02
控制器和基带指令	0x03
信息参数	0x04
状态参数	0x05
测试指令	0x06
LE 控制器指令	0x08
厂商特定指令	0x3F

根据 OGF，指令可分为 3 组：

1. 标准 HCI 指令：OGF 值不是 0x3F 的指令。这些指令是为控制器设计的。蓝牙规范中定义了所有标准 HCI 指令，在此不予赘叙。
2. 厂商特定（VS）HCI 指令：为控制器设计的 OGF 值为 0x3F 的指令。每个厂商可以根据硬件实现定义自己的 VS 指令。本文档中描述了为 BlueNRG-MS 定义的 VS 指令。
3. 厂商特定（VS）ACI 指令：为控制 / 访问主机设计的 OGF 值为 0x3F 的指令。蓝牙规范没有为主机定义任何指令，因此所有 ACI 指令自然而然都是厂商特定指令。本文档中描述了 BlueNRG-MS 的 ACI 指令。

具体来说，为控制控制器设计的指令被称为 HCI 指令。此名称由蓝牙规范定义，为了符合规范，我们维持这一用法。为控制主机设计的指令被称为 ACI 指令。使用不同名称的目的是为了表明该指令是分配给主机，即整个 BLE 系统，而不只是底层控制器。

不过，由于 ACI 和 HCI 指令都是通过 ACI 接口从外部设备接收的指令，并且二者使用相同数据格式，因此二者的区别有时并不十分重要。因此在本文档中，当在一般意义上提及指令时，我们使用的是术语“ACI 指令”。

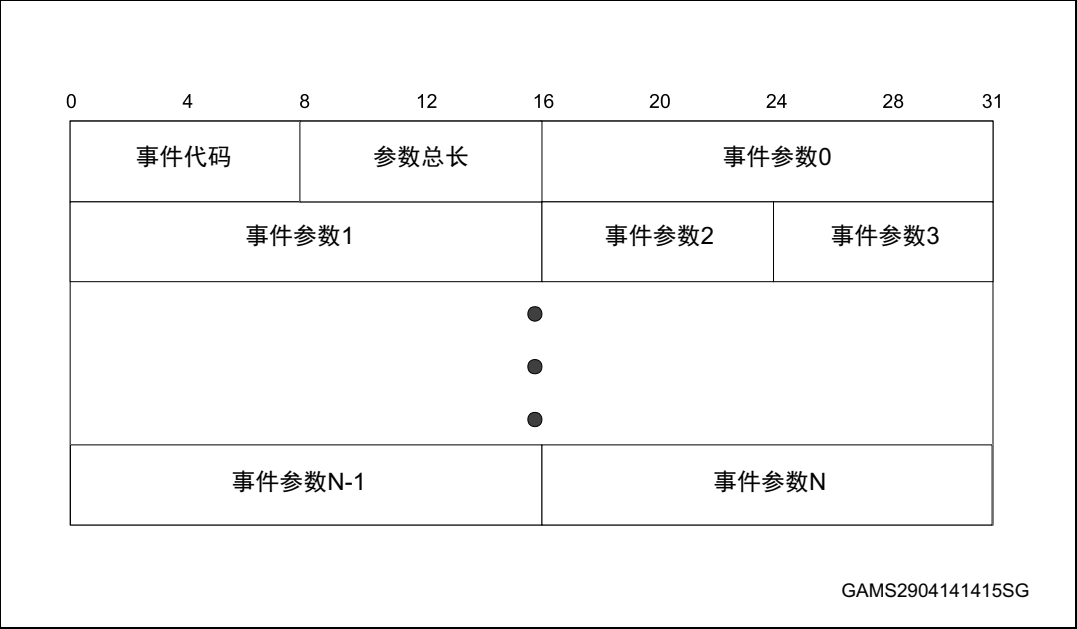
实际上，操作码的值更为重要。每一条指令，无论是 HCI 还是 ACI 指令，都只匹配一个操作码值。用户应用应保证操作码值的正确使用。

请注意，蓝牙规范对操作码使用按位小端序格式。因此在图 4 中，OGF 字段位于右侧。但是在本文档中，当以 0xFFFF 的格式写入操作码时，最高有效位位于左侧。例如，如果操作码为 0xFE81，则 6 位 OGF 为“111111”，即 0x3F，因此是厂商特定指令。其 OCF 为 10 位 0x281。再举一个例子，如果操作码为 0x0406，则 6 位 OGF 为 0x01，OCF 为 0x06。因此，这是标准 HCI 指令 HCI_Disconnect。

2.3 HCI 事件数据包

BlueNRG-MS 使用事件数据包确认指令或通知用户应用其状态已更新。BlueNRG-MS 仅发送 HCI 事件数据包，但是不接收。

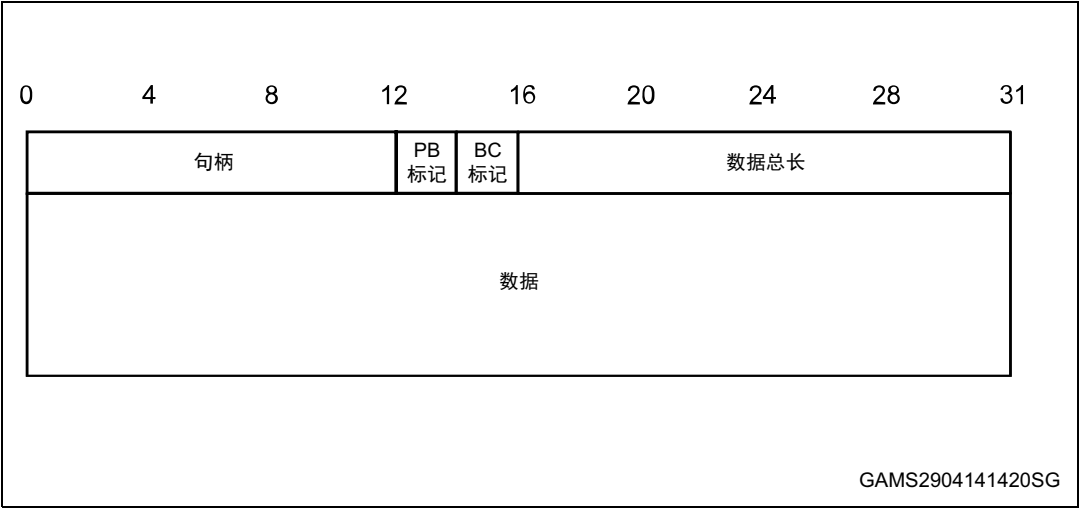
图 5. HCI 事件数据包



2.4 HCI ACL 数据包

ACL 数据包用于交换数据。

图 6. HCI ACL 数据包



3 标准 HCI 指令

3.1 支持的 HCI 指令

下表列出了 BlueNRG-MS 支持的标准 HCI 指令。欲了解每个指令的详细信息，例如指令描述、参数等，请参考蓝牙核心规范第 2 卷 E 部分第 7 章。

表 2. HCI 指令

项目	命令	OGF	OCF	操作码
1	HCI_Disconnect	0x01	0x06	0x0406
2	HCI_Read_Remote_Version_Information	0x01	0x1D	0x041D
3	HCI_Set_Event_Mask	0x03	0x01	0x0C01
4	HCI_Reset	0x03	0x03	0x0C03
5	HCI_Read_Transmit_Power_Level	0x03	0x2D	0x0C2D
6	HCI_Read_Local_Version_Information	0x04	0x01	0x1001
7	HCI_Read_Local_Supported_Commands	0x04	0x02	0x1002
8	HCI_Read_Local_Supported_Features	0x04	0x03	0x1003
9	HCI_Read_BD_ADDR	0x04	0x09	0x1009
10	HCI_Read_RSSI	0x05	0x05	0x1405
11	HCI_LE_Set_Event_Mask	0x08	0x01	0x2001
12	HCI_LE_Read_Buffer_Size	0x08	0x02	0x2002
13	HCI_LE_Read_Local_Supported_Feature	0x08	0x03	0x2003
14	HCI_LE_Set_Random_Address	0x08	0x05	0x2005
15	HCI_LE_Set_Advertizing_Parameters	0x08	0x06	0x2006
16	HCI_LE_Read_Advertizing_Channel_Tx_Power	0x08	0x07	0x2007
17	HCI_LE_Set_Advertizing_Data	0x08	0x08	0x2008
18	Hci_Le_Set_Scan_Resp_Data	0x08	0x09	0x2009
19	HCI_LE_Set_Advertize_Enable	0x08	0x0A	0x200A
20	HCI_LE_Set_Scan_Parameters	0x08	0x0B	0x200B
21	HCI_LE_Set_Scan_Enable	0x08	0x0C	0x200C
22	HCI_LE_Create_Connection	0x08	0x0D	0x200D
23	HCI_LE_Create_Connection_Cancel	0x08	0x0E	0x200E
24	HCI_LE_Read_White_List_Size	0x08	0x0F	0x200F
25	HCI_LE_Clear_While_List	0x08	0x10	0x2010
26	HCI_LE_Add_Device_To_While_List	0x08	0x11	0x2011
27	HCI_LE_Remove_Device_From_While_List	0x08	0x12	0x2012
28	HCI_LE_Connection_Update	0x08	0x13	0x2013
29	HCI_LE_Set_Host_Channel_Classification	0x08	0x14	0x2014

表 2. HCI 指令 (续)

项目	命令	OGF	OCF	操作码
30	HCI_LE_Read_Channel_Map	0x08	0x15	0x2015
31	HCI_LE_Read_Remote_Used_Features	0x08	0x16	0x2016
32	HCI_LE_Encrypt	0x08	0x17	0x2017
33	HCI_LE_Rand	0x08	0x18	0x2018
34	HCI_LE_Start_Encryption	0x08	0x19	0x2019
35	HCI_LE_Long_Term_Key_Request_Reply	0x08	0x1A	0x201A
36	HCI_LE_Long_Term_Key_Requested_Negative_Reply	0x08	0x1B	0x201B
37	HCI_LE_Read_Supported_States	0x08	0x1C	0x201C
38	HCI_LE_Receiver_Test	0x08	0x1D	0x201D
39	HCI_LE_Transmitter_Test	0x08	0x1E	0x201E
40	HCI_LE_Test_End	0x08	0x1F	0x201F

3.2 支持的 HCI 事件

下表列出了 BlueNRG-MS 支持的 HCI 事件。

表 3. HCI 事件

项目	事件	事件代码	子事件代码
1	Evt_Disconn_Complete	0x05	-
2	Evt_Encrypt_Change	0x08	-
3	Evt_Read_Remote_Version_Complete	0x0C	-
4	Evt_Cmd_Complete	0x0E	-
5	Evt_Cmd_Status	0x0F	-
6	Evt_Hardware_Error	0x10	-
7	Evt_Num_Comp_Pkts	0x13	-
8	Evt_Data_Buffer_Overflow	0x1A	-
9	Evt_Encryption_Key_Refresh_Complete	0x30	-
10	Evt_LE_Conn_Complete	0x3E	0x01
11	Evt_LE_Advertising_Report	0x3E	0x02
12	Evt_LE_Conn_Update_Complete	0x3E	0x03
13	Evt_LE_Read_Remote_Used_Features_Complete	0x3E	0x04
14	Evt_LE_LTK_Request	0x3E	0x05

3.3 正确使用 HCI 指令

使用 HCI 指令进行广告时，需按照特定顺序才能确保正确结果。

下面是开始广告时的指令顺序。

在 HCI 层面使用指令时，必须按照以下正确顺序：

1. HCI_LE_Set_Advertising_Parameters - 用于设置广告参数。
2. HCI_LE_Set_Advertising_Data - 用于设置广告数据包中使用的有数据字段的数据。
3. HCI_LE_Set_Scan_Resp_Data - 用于设置扫描数据包中使用的有数据字段的数据。
4. HCI_LE_Set_Advertise_Enable - 启用广告。

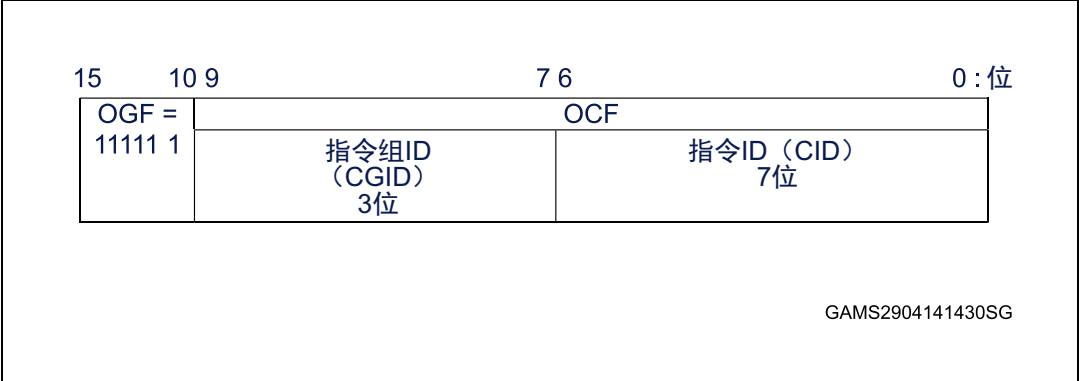
4 厂商特定指令

VS 指令可以是用于访问 BlueNRG-MS 主机的 ACI VS 指令，或用于访问 LE 控制器的 HCI VS 指令。两种类型的指令使用的 OGF 值都是 0x3F。

4.1 VS 指令和 VS 事件格式

VS 指令操作码的 OCF 字段进一步划分为两个字段：指令组 ID 和指令 ID。

图 7. VS 指令的操作码格式



上面的图 7 给出了 16 位操作码格式（另请参见第 2.2 节）。VS 指令的 OGF 字段总是为 0x3F。10 位 OCF 字段分为两部分：3 位指令组 ID（CGID）和 7 位指令 ID（CID）。CGID 由 BlueNRG-MS ACI 接口用于将指令路由到不同逻辑层，例如 L2CAP、GAP、GATT 等。它还有助于以更清晰的结构分类 VS 指令。CID 决定了每个指令的 ID。每个 CGID 组可以有最多 128 个 VS 指令。

表 4. CGID 组

指令组	说明	CGID
HCI	HCI 扩展指令	0x0
GAP	通用访问配置文件指令	0x1
GATT	通用属性配置文件指令	0x2
L2CAP	L2CAP 指令	0x3
Reserved		0x4 – 0x7

VS 事件的格式与标准 HCI 事件相比也稍有不同（另请参见第 2.3 节）。(1) VS 事件的 8 位事件代码的值总是为 0xFF。(2) 16 位事件参数 0 具有不同格式。

图 8. VS 事件的事件参数 0 格式



事件参数 0 是 HCI 事件数据包中的第一个返回参数，位于参数长度字段之后。这 2 个字节一起被定义为 BlueNRG-MS 事件代码（ECODE）。ECODE 被进一步划分为 2 个字段：事件组 ID（EGID）和事件 ID（EID）。BlueNRG-MS 使用 EGID 将事件组合成逻辑组。使用 EID 指定组中的事件。EGID 占据 ECODE 字段的 6 位，而 EID 占据剩余 10 位。

表 5. EGID 组

事件组	说明	EGID
HCI	HCI 扩展事件	0x0
GAP	通用访问配置文件指令	0x1
L2CAP	L2CAP 事件	0x2
GATT	通用属性配置文件事件	0x3



4.2 L2CAP VS 指令

表 6. L2CAP VS 指令

项目	命令	CGID	CID	操作码
1	Aci_L2CAP_Connection_Parameter_Update_Request	0x03	0x01	0xFD81
2	Aci_L2CAP_Connection_Parameter_Update_Response	0x03	0x02	0xFD82

4.2.1 Aci_L2CAP_Connection_Parameter_Update_Request

表 7. Aci_L2CAP_Connection_Parameter_Update_Request

指令名	参数	返回
Aci_L2CAP_Connection_Parameter_Update_Request (0xFD81)	Connection_handle Interval_min Interval_max Slave_latency Timeout_multiplier	状态

说明：

从从设备向主设备发送 L2CAP 连接参数更新请求。

表 8. Aci_L2CAP_Connection_Parameter_Update_Requests 指令参数

参数	大小	说明
Connection_handle	2 字节	必须发送连接参数更新请求的链路的连接句柄
Interval_min	2 字节	用以下方式连接事件间隔定义最小值： $connIntervalMin = Interval\ Min * 1.25\ ms$
Interval_max	2 字节	用以下方式连接事件间隔定义最大值： $connIntervalMax = Interval\ Max * 1.25\ ms$
Slave_latency	2 字节	定义从设备延迟参数（LL 连接事件的数量）
Timeout_multiplier	2 字节	用以下方式定义连接超时参数：超时乘数 * 10 ms

表 9. Aci_L2CAP_Connection_Parameter_Update_Requests 返回参数

参数	大小	说明
Status	1 字节	0x0: 连接参数已接受 0x1: 连接参数已拒绝

生成事件:

指令接收时的指令状态事件和主设备响应请求（接受或拒绝）时的
Evt_Blue_L2CAP_Conn_Upd_Resp 事件。

4.2.2 Aci_L2CAP_Connection_Parameter_Update_Response

表 10. Aci_L2CAP_Connection_Parameter_Update_Response

指令名	参数	返回
Aci_L2CAP_Connection_Parameter_Update_Response (0xFD82)	Conn_Handle Conn_Interval_Min Conn_Interval_Max Conn_Latency Timeout_Multiplier Min_CE_length Max_CE_length 标识符 接受	状态

说明:

控制器在响应 Evt_Blue_L2CAP_Conn_Upd_Req 事件时应发送该指令。如果事件中给出的
连接参数可接受，必须将接受参数设置为 1。

表 11. Aci_L2CAP_Connection_Parameter_Update_Response 指令参数

参数	大小	说明
Conn_Handle	2 字节	在 Evt_Blue_L2CAP_Conn_Upd_Req 事件中接收的句柄。
Min connection Interval	2 字节	在 L2CAP 连接更新请求事件中接收的连接间隔参数
Max connection Interval	2 字节	在 L2CAP 连接更新请求事件中接收的最大连接间隔参数。
latency	2 字节	在 L2CAP 连接更新请求事件中接收的从设备延迟参数。
Timeout_Multiplier	2 字节	在 L2CAP 连接更新请求事件中接收的监控连接超时参数。
Min_CE length	2 字节	LE 连接所需的最小连接事件长度。 范围: 0x0000 – 0xFFFF 时间 = N * 0.625 ms。
Max_CE_length	2 字节	LE 连接所需的最大连接事件长度。 范围: 0x0000 – 0xFFFF 时间 = N * 0.625 ms。



表 11. Aci_L2CAP_Connection_Parameter_Update_Response 指令参数（续）

参数	大小	说明
Identifier	1 字节	在 Evt_Blue_L2CAP_Conn_Upd_Req 事件中接收的标识符。
Accept	1 字节	0x00: 连接更新参数不可接受。 0x01: 连接更新参数可接受。

表 12. Aci_L2CAP_Connection_Parameter_Update_Response 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件：

生成指令完成事件。如果状态为成功，则以在连接更新完成事件中接收的新设定值更新链路参数。

4.3 L2CAP VS 事件

表 13. L2CAP VS 事件

项目	事件	EGID	EID	ECODE
1	Evt_Blue_L2CAP_Conn_Upd_Resp	0x02	0x00	0x0800
2	Evt_Blue_L2CAP_Procedure_Timeout	0x02	0x01	0x0801
3	Evt_Blue_L2CAP_Conn_Upd_Req	0x02	0x02	0x0802

4.3.1 Evt_Blue_L2CAP_Conn_Upd_Resp

当主设备以连接更新响应数据包或指令拒绝数据包响应连接请求数据包时，生成该事件。

表 14. Evt_Blue_L2CAP_Conn_Upd_Resp

参数	大小	说明
Event code	2 字节	连接更新响应事件的事件代码
Conn_handle	2 字节	与事件相关的连接句柄
Event_data_length	1 字节	下列数据的长度
Code	1 字节	0x13 （有效的 L2CAP 连接参数更新响应数据包）。 0x01 （指令拒绝）。
Identifier	1 字节	响应的标识符。它等于请求。
L2cap_length	2 字节	下列数据的长度。应总是为 2。
Result	2 字节	对于连接参数更新响应（代码 =0x13），为结果代码（参数被接受或被拒绝）；对于指令拒绝（代码 =0x01），为拒绝的原因代码。

4.3.2 Evt_Blue_L2CAP_Procedure_Timeout

当主设备在 30 秒内没有以连接更新响应数据包或指令拒绝数据包响应连接请求数据包时，生成该事件。

表 15. Evt_Blue_L2CAP_Procedure_Timeout

参数	大小	说明
Event code	2 字节	L2CAP 流程超时事件的事件代码
conn_handle	2 字节	发出的指令所针对的连接句柄
event_data_length	1 字节	总是为 0

4.3.3 Evt_Blue_L2CAP_Conn_Upd_Req

当接收到来自从设备的连接更新请求时，L2CAP 层给出事件。接收到该事件的上层必须以向 BlueNRG-MS 发送 Aci_L2CAP_Connection_Parameter_Update_Response 指令作为响应。



表 16. Evt_Blue_L2CAP_Conn_Upd_Req

参数	大小	说明
Event code	2 字节	连接更新请求事件的事件代码。
Conn_Handle	2 字节	已接收到连接更新请求的连接的句柄。在以指令 Aci_L2CAP_Conn_Upd_Resp 响应事件的同时，必须返回相同句柄。
event_data_length	1 字节	后续数据的长度。数据将为接收的 L2CAP 连接更新请求。
Identifier	1 字节	这是将请求关联到响应的标识符。上层必须以指令 Aci_L2CAP_Conn_Upd_Resp 返回相同标识符。
l2cap_length	2 字节	L2CAP 连接更新请求的长度
Interval_Min	2 字节	值如蓝牙规范 v4.1 第 3 卷 A 部分第 4.20 节所定义
Interval_Max	2 字节	值如蓝牙规范 v4.1 第 3 卷 A 部分第 4.20 节所定义
Slave_Latency	2 字节	值如蓝牙规范 v4.1 第 3 卷 A 部分第 4.20 节所定义
timeout_mult	2 字节	值如蓝牙规范 v4.1 第 3 卷 A 部分第 4.20 节所定义

4.4 GAP VS 指令

4.4.1 概述

为 LE 设备定义的 GAP 角色有 4 个：

- 广播方：广播方是发送广告事件的设备。广播方应具有发送器且可选择配备接收器。
- 监听方：处于监听方角色的设备接收广告事件。监听方应具有接收器且可选择配备发送器。
- 外设：接受 LE 物理链路建立请求的任何设备都被称为外设。在 LL 连接状态下，作为外设工作的设备将充当从设备。外设应同时配备发送器和接收器。
- 中央设备：发起 LE 物理链路建立请求的设备被称为中央设备。在 LL 连接状态下，作为中央设备工作的设备将充当主设备。中央设备应同时配备发送器和接收器。

对于 GAP 配置文件，BlueNRG-MS 可以是外设或中央设备，原因在于：

- BlueNRG-MS 控制器能够在连接中充当从设备，即它可以接受来自中央设备的 LL 连接请求，并能支持 GAP 外设角色的所有强制要求，如蓝牙核心规范 4.1 通用访问配置文件 C 部分表 2.1 所述。
- BlueNRG-MS 控制器能够在连接中充当主设备，即它可以向外设发出 LL 连接请求，并能支持 GAP 中央设备角色的所有强制要求，如蓝牙核心规范 4.1 通用访问配置文件 C 部分表 2.1 所述。

4.4.2 GAP VS 指令

表 17. GAP VS 指令

项目	命令	CGID	CID	操作码
1	Aci_Gap_Set_Non_Discoverable	0x01	0x01	0xFC81
2	Aci_Gap_Set_Limited_Discoverable	0x01	0x02	0xFC82
3	Aci_Gap_Set_Discoverable	0x01	0x03	0xFC83
4	Aci_Gap_Set_Direct_Connectable	0x01	0x04	0xFC84
5	Aci_Gap_Set_IO_Capability	0x01	0x05	0xFC85
6	Aci_Gap_Set_Auth_Requirement	0x01	0x06	0xFC86
7	Aci_Gap_Set_Author_Requirement	0x01	0x07	0xFC87
8	Aci_Gap_Pass_Key_Response	0x01	0x08	0xFC88
9	Aci_Gap_Authorization_Response	0x01	0x09	0xFC89
10	Aci_Gap_Init	0x01	0x0A	0xFC8A
11	Aci_Gap_Set_Non_Connectable	0x01	0x0B	0xFC8B
12	Aci_Gap_Set_Undirected_Connectable	0x01	0x0C	0xFC8C
13	Aci_Gap_Slave_Security_request	0x01	0x0D	0xFC8D
14	Aci_Gap_Update_Adv_Data	0x01	0x0E	0xFC8E
15	Aci_Gap_Delete_AD_Type	0x01	0x0F	0xFC8F
16	Aci_Gap_Get_Security_Level	0x01	0x10	0xFC90
17	Aci_Gap_Set_Event_Mask	0x01	0x11	0xFC91
18	Aci_Gap_Configure_WhiteList	0x01	0x12	0xFC92
19	Aci_Gap_Terminate	0x01	0x13	0xFC93
20	Aci_Gap_Clear_Security_Database	0x01	0x14	0xFC94
21	Aci_Gap_Allow_Rebond	0x01	0x15	0xFC95
22	Aci_Gap_Start_Limited_Discovery_Proc	0x01	0x16	0xFC96
23	Aci_Gap_Start_General_Discovery_Proc	0x01	0x17	0xFC97
24	Aci_Gap_Start_Name_Discovery_Proc	0x01	0x18	0xFC98
25	Aci_Gap_Start_Auto_Conn_Establishment	0x01	0x19	0xFC99
26	Aci_Gap_Start_General_Conn_Establishment	0x01	0x1A	0xFC9A
27	Aci_Gap_Start_Selective_Conn_Establishment	0x01	0x1B	0xFC9B
28	Aci_Gap_Create_Connection	0x01	0x1C	0xFC9C
29	Aci_Gap_Terminate_Gap_Procedure	0x01	0x1D	0xFC9D
30	Aci_Gap_Start_Connection_Update	0x01	0x1E	0xFC9E
31	Aci_Gap_Send_Pairing_Request	0x01	0x1F	0xFC9F
32	Aci_Gap_Resolve_Private_Address	0x01	0x20	0xFCA0
33	Aci_Gap_Get_Bonded_Devices	0x01	0x23	0xFCA3

表 17. GAP VS 指令（续）

项目	命令	CGID	CID	操作码
34	Aci_Gap_Set_Broadcast_Mode	0x01	0x21	0xFCA1
35	Aci_Gap_Start_Observation_Proc	0x01	0x22	0xFCA2
36	Aci_Gap_Is_Device_Bonded	0x01	0x24	0xFCA4

4.4.3 Aci_Gap_Set_Non_Discoverable

表 18. Aci_gap_set_non_discoverable

指令名	参数	返回
Aci_Gap_Set_Non_Discoverable (0xFC81)		状态

说明：
将设备设置为不可发现模式。该指令将禁用 LL 广告并使设备进入待机状态。

表 19. Aci_gap_set_non_discoverable 返回参数

参数	大小	说明
Status	1 字节	0x0: 状态成功 0xC: 指令不被允许

生成事件：
当 Aci_Gap_Set_Non_Discoverable 指令完成时，控制器将生成指令完成事件。

4.4.4 Aci_Gap_Set_Limited_Discoverable

表 20. Aci_Gap_Set_Limited_Discoverable

指令名	参数	返回
Aci_Gap_Set_Limited_Discoverable (0xFC82)	Advertising_Event_Type Adv_Interval_Min Adv_Interval_Max Address_Type Adv_Filter_Policy Local_Name_Length Local_Name Service_Uuid_Length Service_Uuid_List Slave_Conn_Interval_Min Slave_Conn_Interval_Max	状态

说明：

将设备设置为有限可发现模式（如 GAP 规范第 3 卷第 9.2.3 节所定义）。设备处于可发现模式的时间为 TGAP 的最长周期 (lim_adv_timeout) = 180 秒（来自勘误表）。随时可通过发出 Aci_Gap_Set_Non_Discoverable 指令禁用广告。

Adv_Interval_Min 和 Adv_Interval_Max 参数为可选参数。如果将二者均设置为 0，GAP 将对有限可发现模式下的广告间隔使用默认值。

为了允许快速连接，主机可以设置 Local_Name、Service_Uuid_List、Slave_Conn_Interval_Min 和 Slave_Conn_Interval_Max。在该指令中，这些参数为可选参数。可以在广告的数据中单独使用 GAP_Update_Adv_Data 指令设置这些值。

表 21. Aci_Gap_Set_Limited_Discoverable 指令参数

参数	大小	说明
Advertising_Event_Type	1 字节	0x00: 可连接非定向广告（默认） 0x02: 可扫描非定向广告 0x03: 不可连接非定向广告
Adv_Interval_Min	2 字节	非定向广告的最小广告间隔。 范围: 0x0020 至 0x4000 默认值: N = 0x0800 (1.28 s) 时间 = N * 0.625 ms 时间范围: 20 ms 至 10.24 s
Adv_Interval_Max	2 字节	非定向广告的最大广告间隔。 范围: 0x0020 至 0x4000 默认值: N = 0x0800 (1.28 s) 时间 = N * 0.625 ms 时间范围: 20 ms 至 10.24 s
Address_Type	1 字节	0x00: 公共设备地址（默认） 0x01: 随机设备地址
Adv_Filter_Policy	1 字节	0x00: 允许任何来源的扫描请求，允许任何来源的连接请求（默认）。 0x01: 仅允许来自白名单设备的扫描请求，允许任何来源的连接请求。 0x02: 允许任何来源的扫描请求，仅允许来自白名单设备的连接请求。 0x03: 仅允许来自白名单设备的扫描请求，仅允许来自白名单设备的连接请求。
Local_Name_Length	1 字节	本地名称字符串的八位组长度。 如果长度设置为 0x00，不应使用 Local_Name 参数。
Local_Name	0-N 字节	设备的本地名称。这是一个 ASCII 字符串，无 NULL 字符。 第一个字节为 AD 类型： AD_TYPE_SHORTENED_LOCAL_NAME 或 AD_TYPE_COMPLETE_LOCAL_NAME。
Service_UUID_Length	1 字节	服务 UUID 列表的八位组长度。如果没有要广告的服务，将该字段设置为 0x00。
Service_UUID_List	0-N 字节	这是 UUID 的 AD 类型列表，GAP 规范第 3 卷第 11.1.1 节对其进行了定义。

表 21. Aci_Gap_Set_Limited_Discoverable 指令参数 (续)

参数	大小	说明
Slave_Conn_Interval_Min	2 字节	从设备连接间隔最小值。 用以下方式定义连接间隔： $\text{connIntervalmin} = \text{Slave_Conn_Interval_Min} * 1.25 \text{ ms}$ Slave_Conn_Interval_Min 范围：0x0006 至 0x0C80 值 0xFFFF 表示无特定最小值。
Slave_Conn_Interval_Max	2 字节	从设备连接间隔最大值。 $\text{ConnIntervalmax} = \text{Slave_Conn_Interval_Max} * 1.25 \text{ ms}$ Slave_Conn_Interval_Max 范围：0x0006 至 0x0C80 Slave_Conn_Interval_Max 应大于等于 Slave_Conn_Interval_Min。 值 0xFFFF 表示无特定最大值。

表 22. Aci_Gap_Set_Limited_Discoverable 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x0C：指令不被允许 0x11：不支持的特性

生成事件：

当控制器接收到指令时，将生成指令状态事件和返回参数。在这之后，控制器开始广告，当发生广告超时（即有限发现周期已结束）时，控制器生成 Evt_Blue_Gap_Limited_Discoverable_Complete 事件。

4.4.5 Aci_Gap_Set_Discoverable

表 23. Aci_Gap_Set_Discoverable

指令名	参数	返回
Aci_Gap_Set_Discoverable (0xFC83)	Advertising_Event_Type Adv_Interval_Min Adv_Interval_Max Address_Type Adv_Filter_Policy Local_Name_Length Local_Name Service_Uuid_Length Service_Uuid_List Slave_Conn_Interval_Min Slave_Conn_Interval_Max	状态

说明：

将设备设置为一般可发现模式（如 GAP 规范第 3 卷第 9.2.4 节所定义）。设备将处于可发现模式，直至主机发出 Aci_Gap_Set_Non_Discoverable 指令。Adv_Interval_Min 和 Adv_Interval_Max 参数为可选参数。如果将二者均设置为 0，GAP 将对一般可发现模式下的广告间隔使用默认值。

表 24. Aci_Gap_Set_Discoverable 指令参数

参数	大小	说明
Advertising_Event_Type	1 字节	0x00：可连接非定向广告（默认） 0x02：可扫描非定向广告 0x03：不可连接非定向广告
Adv_Interval_Min	2 字节	非定向广告的最小广告间隔。 范围：0x0020 至 0x4000 默认值：N = 0x0800（1.28 s）时间 = N * 0.625 ms 时间范围：20 ms 至 10.24 s
Adv_Interval_Max	2 字节	非定向广告的最大广告间隔。 范围：0x0020 至 0x4000 默认值：N = 0x0800（1.28 s）时间 = N * 0.625 ms 时间范围：20 ms 至 10.24 s
Address_Type	1 字节	0x00：公共设备地址（默认） 0x01：随机设备地址
Adv_Filter_Policy	1 字节	0x00：允许任何来源的扫描请求，允许任何来源的连接请求（默认）。 0x01：仅允许来自白名单设备的扫描请求，允许任何来源的连接请求。 0x02：允许任何来源的扫描请求，仅允许来自白名单设备的连接请求。 0x03：仅允许来自白名单设备的扫描请求，仅允许来自白名单设备的连接请求。
Local_Name_Length	1 字节	本地名称字符串的八位组长度。 如果长度设置为 0x00，不应使用 Local_Name 参数。

表 24. Aci_Gap_Set_Discoverable 指令参数（续）

参数	大小	说明
Local_Name	0-N 字节	设备的本地名称。这是一个 ASCII 字符串，无 NULL 字符。 第一个字节为 AD 类型： AD_TYPE_SHORTENED_LOCAL_NAME 或 AD_TYPE_COMPLETE_LOCAL_NAME。
Service_UUID_Length	1 字节	服务 UUID 列表的八位组长度。如果没有要广告的服务，将该字段设置为 0x00。
Service_UUID_List	0-N 字节	这是 UUID 的 AD 类型列表， GAP 规范第 3 卷第 11.1.1 节对其进行了定义。
Slave_Conn_Interval_Min	2 字节	从设备连接间隔最小值。 用以下方式定义连接间隔： $connIntervalmin = Slave_Conn_Interval_Min * 1.25ms$ Slave_Conn_Interval_Min 范围：0x0006 至 0x0C80 值 0xFFFF 表示无特定最小值。
Slave_Conn_Interval_Max	2 字节	从设备连接间隔最大值。 $ConnIntervalmax = Slave_Conn_Interval_Max * 1.25ms$ Slave_Conn_Interval_Max 范围：0x0006 至 0x0C80 Slave_Conn_Interval_Max 应大于等于 Slave_Conn_Interval_Min。 值 0xFFFF 表示无特定最大值。

表 25. Aci_Gap_Set_Discoverable 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x0C：指令不被允许 0x11：不支持的特性

生成事件：
当 Aci_Gap_Set_Discoverable 指令完成时，控制器将生成指令完成事件。

4.4.6 Aci_Gap_Set_Direct_Connectable

表 26. Aci_Gap_Set_Direct_Connectable

指令名	参数	返回
Aci_Gap_Set_Direct_Connectable (0xFC84)	Own_Address_Type 广告类型 Initiator_Address_Type Initiator_Direct_Address	状态

说明：

将设备设置为直接可连接模式（如 GAP 规范第 3 卷第 9.3.3 节所定义）。设备在直接可连接模式下，使用高占空比广告事件或低占空比广告事件和在自有地址类型参数中指定的地址进行广告。指令中的广告类型参数指定了使用的广告类型。

设备的定向可连接模式仅持续 1.28 秒。如果在此期间没有建立连接，设备将进入不可发现模式，并且必须明确地重新启用广告功能。

表 27. Aci_Gap_Set_Direct_Connectable 指令参数

参数	大小	说明
Own_Address_Type	1 字节	0x00：公共设备地址（默认） 0x01：随机设备地址
Advertising Type	1 字节	广告类型 0x01：高占空比定向广告 0x04：低占空比定向广告
Initiator_Address_Type	1 字节	0x00：公共设备地址（默认） 0x01：随机设备地址
Initiator_Direct_Address	6 字节	发起方蓝牙地址
Adv_Interval_min	2 字节	低占空比定向广告的最小广告间隔。范围：0x0020 至 0x4000。时间 = $N * 0.625 \text{ ms}$ 。时间范围：20 ms 至 10.24 s。
Adv_Interval_max	2 字节	低占空比定向广告的最大广告间隔。范围：0x0020 至 0x4000。时间 = $N * 0.625 \text{ ms}$ 。时间范围：20 ms 至 10.24 s。

表 28. Aci_Gap_Set_Direct_Connectable 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x0C: 指令不被允许 0x11: 不支持的特性

生成事件:

当指令完成时，控制器将生成指令完成事件。控制器还生成 LE_Connection_Complete 事件，并且如果没有建立连接，状态设置为 directed_advertising_timeout，如果连接成功建立，则为 0x00。

4.4.7 Aci_Gap_Set_IO_Capability

表 29. Aci_Gap_Set_IO_Capability

指令名	参数	返回
Aci_Gap_Set_IO_Capability (0xFC85)	IO_Capability	状态

说明:

设置设备的 IO 能力。只能在设备未处于连接状态时发出该指令。

表 30. Aci_Gap_Set_IO_Capability 指令参数

参数	大小	说明
IO_Capability	1 字节	0x00: 仅显示 0x01: 显示是 / 否 0x02: 仅键盘 0x03: 无输入，无输出 0x04: 键盘，显示

表 31. Aci_Gap_Set_IO_Capability 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x0C: 指令不被允许

生成事件:

当指令完成时，控制器将生成指令完成事件。



4.4.8 Aci_Gap_Set_Auth_Requirement

表 32. Aci_Gap_Set_Auth_Requirement

指令名	参数	返回
Aci_Gap_Set_Auth_Requirement (0xFC86)	MIMT_mode OOB_enable OOB_data Min_encryption_key_size Max_encryption_key_size Use_fixed_pin Fixed_pin Bonding_mode	状态

说明：

为设备设置验证要求。如果 OOB_Enable 设置为 0，下列 16 个八位组的 OOB_Data 将在接收时被忽略。只能在设备未处于连接状态时发出该指令。

表 33. Aci_Gap_Set_Auth_Requirement 指令参数

参数	大小	说明
MITM_mode	1 字节	0x00：不要求 MITM 保护 0x01：要求 MITM 保护
OOB_enable	1 字节	0x00：带外验证不启用 0x01：带外验证启用
OOB_data	16 字节	OOB 数据
Min_encryption_key_size	1 字节	配对过程中要使用的加密密钥的最小大小
Max_encryption_key_size	1 字节	配对过程中要使用的加密密钥的最大大小
Use_fixed_pin	1 字节	0x00：配对过程中使用 PIN 码。 0x01：配对过程中不使用 PIN 码。 这种情况下，GAP 将向主机生成 Evt_Blue_Pin_Code_Request 事件。
Fixed_pin	4 字节	MIMT 保护启用时配对过程中要使用的固定引脚。0 至 999999 之间的任何随机数
Bonding_mode	1 字节	0x00：不要求绑定 0x01：要求绑定

表 34. Aci_Gap_Set_Auth_Requirement 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x0C: 指令不被允许 0x11: 不支持的特性

生成事件:

当指令完成时, 控制器将生成指令完成事件。

4.4.9 Aci_Gap_Set_Author_Requirement

表 35. Aci_Gap_Set_Author_Requirement

指令名	参数	返回
Aci_Gap_Set_Author_Requirement (0xFC87)	Connection_handle Authorization_enable	状态

说明:

设置设备的授权要求。在连接到设备时, 如果需要授权才能访问要求授权的服务, 必须发出该指令。

表 36. Aci_Gap_Set_Author_Requirement 指令参数

参数	大小	说明
Connection_handle	2 字节	连接的句柄。
Authorization_enable	1 字节	0x00: 不要求授权 (默认) 0x01: 要求授权。在设备中启用授权, 在远程设备尝试读取 / 写入具有授权要求的特征时, 栈将发回包含“权限不足”错误代码的错误响应。在配对完成后, 将向主机发送 Evt_Blue_Gap_Authorization_Request 事件。

表 37. Aci_Gap_Set_Author_Requirement 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x0C: 指令不被允许

生成事件:

当指令完成时, 控制器将生成指令完成事件。

4.4.10 Aci_Gap_Pass_Key_Response

表 38. Aci_Gap_Pass_Key_Response

指令名	参数	返回
Aci_Gap_Pass_Key_Response (0xFC88)	Conn_handle Pass_Key	状态

说明：

该指令应由主机发送以响应 Evt_Blue_Gap_Pass_Key_Request 事件。指令参数包含配对过程中将使用的输入密钥。

表 39. Aci_Gap_Pass_Key_Response 指令参数

参数	大小	说明
Conn_handle	2 字节	Connection_handle
Pass_Key	4 字节	输入密钥，值范围：0 - 999999（十进制）

表 40. Aci_Gap_Pass_Key_Response 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x0C：指令不被允许

生成事件：

当控制器接收到该指令时，将生成指令状态事件和返回参数。当配对过程完成时，将生成 Evt_Blue_Gap_Pairing_Cmplt 事件。

4.4.11 Aci_Gap_Authorization_Response

表 41. Aci_Gap_Authorization_Response

指令名	参数	返回
Aci_Gap_Authorization_Response (0xFC89)	Connection_handle 授权	状态

说明：

该指令应由主机发送以响应 Evt_Blue_Gap_Authorization_Request 事件。

表 42. Aci_Gap_Authorization_Response 指令参数

参数	大小	说明
Connection_handle	2 字节	Connection_handle
Authorize	1 字节	0x01: 授权 (接受连接) 0x02: 拒绝 (拒绝连接)

表 43. Aci_Gap_Authorization_Response 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x0C: 指令不被允许

生成事件：
当指令完成时，控制器将生成指令完成事件。

4.4.12 Aci_Gap_Init

表 44. Aci_Gap_Init

指令名	参数	返回
Aci_Gap_Init (0xFC8A)	角色 enable_Privacy device_name_char_len	Service_handle Dev_name_char_handle Appearance_Char_handle

说明：
注册具有 GATT 的 GAP 服务。默认添加设备名称特征和外观特征，并在事件数据中返回这些特征的句柄。角色参数可以是下述任意值的按位或。

表 45. Aci_Gap_Init 指令参数

参数	大小	说明
Role	1 字节	0x01: 外设 0x02: 广播方 0x04: 中央设备 0x08: 监听方
enable_Privacy	1 字节	0x00: 私有禁用 0x01: 私有启用
device_name_char_len	1 字节	设备名称特性的长度



表 46. Aci_Gap_Init 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误。没有足够内存用于添加需要的 GATT 特征 0x12: HCI_Invalid 参数
Service_handle	2 字节	GAP 服务的句柄
Dev_name_char_handle	2 字节	添加到 GAP 服务的设备名称特征的句柄
Appearance_Char_handle	2 字节	添加到 GAP 服务的外观特征的句柄

生成事件:

在指令完成时，生成指令完成事件。状态指示成功或失败。如果 GAP 服务的注册成功，事件数据包含注册了 GATT 的 GAP 服务的句柄以及服务名称和外观特征的属性句柄。

4.4.13 Aci_Gap_Set_Non_Connectable

表 47. Aci_Gap_Set_Non_Connectable

指令名	参数	返回
Aci_Gap_Set_Non_Connectable (0xFC8B)	Advertising_Event_Type own_address_Type	状态

说明:

使设备进入不可连接模式。该模式不支持连接。gap_init 指令中完成的私有设置在决定该指令的有效参数方面起到重要作用。

表 48. Aci_Gap_Set_Non_Connectable 指令参数

参数	大小	说明
Advertising_Event_Type	1 字节	0x02: 可扫描非定向广告 0x03: 不可连接非定向广告
own_address_type	1 字节	如果私有禁用: 0x00: 公共设备地址类型 0x01: 静态随机地址类型 如果私有启用: 0x02: 私有可解析地址类型 0x03: 私有不可解析地址类型

表 49. Aci_Gap_Set_Non_Connectable 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x41: BLE_Status_Failed 0x42: BLE_Status_Invalid 参数 0x0C: 指令不被允许 0x12: 无效 HCI 参数

生成事件:

在指令完成时生成指令完成事件。

4.4.14 Aci_Gap_Set_Undirected_Connectable

表 50. Aci_Gap_Set_Undirected_Connectable

指令名	参数	返回
Aci_Gap_Set_Undirected_Connectable (0xFC8C)	Advertising_filtering_policy Own_address_type	状态

说明:

使设备进入非定向可连接模式。gap_init 指令中完成的私有设置在决定该指令的有效参数方面起到重要作用。

表 51. Aci_Gap_Set_Undirected_Connectable 指令参数

参数	大小	说明
Advertising_filtering_policy	1 字节	0x00: 允许任何来源的扫描请求, 允许任何来源的连接请求 (默认)。 0x03: 仅允许来自白名单设备的扫描请求, 仅允许来自白名单设备的连接请求。
Own_Address_Type	1 字节	如果私有禁用 0x00: 公共设备地址 (默认) 0x01: 随机设备地址 如果私有启用 0x02: 私有可解析地址 0x03: 不可解析地址

表 52. Aci_Gap_Set_Undirected_Connectable 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x0C: 指令不被允许 0x11: 不支持的特性

生成事件:

在指令完成时生成指令完成事件。

4.4.15 Aci_Gap_Slave_Security_Request

表 53. Aci_Gap_Slave_Security_Request

指令名	参数	返回
Aci_Gap_Slave_Security_Request (0xFC8D)	Connection_handle 接合 MITM_Protection	状态

说明：
必须发出该指令以将从设备的安全要求通知主设备。

表 54. Aci_Gap_Slave_Security_Request 指令参数

参数	大小	说明
Connection_handle	2 字节	将发送从设备安全请求的连接的句柄（在仅从设备模式下被忽略）
Bonding	1 字节	0x00：不绑定 0x01：要求绑定
MITM_protection	1 字节	0x00：不要求 MITM 保护 0x01：要求 MITM 保护

表 55. Aci_Gap_Slave_Security_Request 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x1F：内存不足 0x0C：指令不被允许

生成事件：
当接收到有效指令时，将生成指令状态事件。当指令完成时，即当安全请求成功发送至主设备时，将生成 Evt_Blue_Gap_Slave_Security_Initiated 厂商特定事件。

4.4.16 Aci_Gap_Update_Adv_Data

表 56. Aci_Gap_Update_Adv_Data

指令名	参数	返回
Aci_Gap_Update_Adv_Data (0xFC8E)	Adv_Data_Length Adv_Data	状态

说明：

该指令可用于更新特定 AD 类型的广告数据。如果指定的 AD 类型不存在，则为广告数据添加 AD 类型。如果更新后的广告数据总长度超过 31 个八位组，则指令将被拒绝并保留原数据。

表 57. Aci_Gap_Update_Adv_Data 指令参数

参数	大小	说明
Adv_Data_Length	1 字节	advData 参数中后续的数据字节数
Adv_Data	0-N 字节	advData 的格式（包括数据长度）必须符合蓝牙规范（第 3 卷 C 部分第 11 节）

表 58. Aci_Gap_Update_Adv_Data 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x41：失败

生成事件：

当指令完成时，将生成指令完成事件且状态指示成功或失败。

4.4.17 Aci_Gap_Delete_AD_Type

表 59. Aci_Gap_Delete_AD_Type

指令名	参数	返回
Aci_Gap_Delete_AD_Type (0xFC8F)	AD_type	状态

说明：

该指令可用于从广告数据中删除指定的 AD 类型（如果有）。

表 60. Aci_Gap_Delete_AD_Type 指令参数

参数	大小	说明
AD_type	1 字节	0x01: AD_TYPE_FLAGS 0x02: AD_TYPE_16_BIT_SERV_UUID 0x03: AD_TYPE_16_BIT_SERV_UUID_CMPLT_LIST 0x04: AD_TYPE_32_BIT_SERV_UUID 0x05: AD_TYPE_32_BIT_SERV_UUID_CMPLT_LIST 0x06: AD_TYPE_128_BIT_SERV_UUID 0x07: AD_TYPE_128_BIT_SERV_UUID_CMPLT_LIST 0x08: AD_TYPE_SHORTENED_LOCAL_NAME 0x09: AD_TYPE_COMPLETE_LOCAL_NAME 0x0A: AD_TYPE_TX_POWER_LEVEL 0x10: AD_TYPE_SEC_MGR_TK_VALUE 0x11: AD_TYPE_SEC_MGR_OOB_FLAGS 0x12: AD_TYPE_SLAVE_CONN_INTERVAL 0x14: AD_TYPE_SERV_SOLICIT_16_BIT_UUID_LIST 0x15: AD_TYPE_SERV_SOLICIT_32_BIT_UUID_LIST 0x16: AD_TYPE_SERVICE_DATA 0xFF: AD_TYPE_MANUFACTURER_SPECIFIC_DATA

表 61. Aci_Gap_Delete_AD_Type 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x1F: 未找到 AD 类型

生成事件：
将生成指令完成事件且状态指示成功或失败。

4.4.18 Aci_Gap_Get_Security_Level

表 62. Aci_Gap_Get_Security_Level

指令名	参数	返回
Aci_Gap_Get_Security_Level (0xFC90)		状态 MIMT_protection_required Bonding_required OOB_data_present Passkey_required

说明：
该指令可用于获取设备的当前安全设置。



表 63. Aci_Gap_Get_Security_Level 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功
MIMT_protection_required	1 字节	0x00: 不要求 0x01: 要求
Bonding_required	1 字节	0x00: 不要求 0x01: 要求
OOB_data_present	1 字节	0x00: 不存在 0x01: 存在
Passkey_required	1 字节	0x00: 不要求 0x01: 固定引脚存在且正在使用 0x2: 配对需要输入密钥。需要时将生成事件。

生成事件:

将在指令完成事件中返回安全参数。第一个字节指示安全管理器的状态（通道打开或关闭）。

4.4.19 Aci_Gap_Set_Event_Mask

表 64. Aci_Gap_Set_Event_Mask

指令名	参数	返回
Aci_Gap_Set_Event_Mask (0xFC91)	Event_mask	Event_mask 状态

说明:

它允许来自 GAP 的屏蔽事件。默认配置为屏蔽所有事件。

表 65. Aci_Gap_Set_Event_Mask 指令参数

参数	大小	说明
Event_mask	2 字节	0x0001: Evt_Blue_Gap_Limited_Discoverable 0x0002: Evt_Blue_Gap_Pairing_Cmplt 0x0004: Evt_Blue_Gap_Pass_Key_Request 0x0008: Evt_Blue_Gap_Authorization_Request 0x0010: Evt_Blue_Gap_Slave_Security_Initiated 0x0020: Evt_Blue_Gap_Bond_Lost

表 66. Aci_Gap_Set_Event_Mask 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x12: 无效 HCI 指令参数

生成事件：
在指令完成时生成指令完成事件。

4.4.20 **Aci_Gap_Configure_WhiteList**

表 67. Aci_Gap_Configure_WhiteList

指令名	参数	返回
Aci_Gap_Configure_WhiteList (0xFC92)		状态

说明：
使用出现在安全数据库中的设备配置控制器的白名单。

表 68. Aci_Gap_Configure_WhiteList 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x12：无效 HCI 指令参数 0x47：错误

生成事件：
当指令完成时，控制器将生成指令完成事件且状态指示是否配置了白名单。如果数据库中没有设备或设备不能添加到白名单，指令将返回错误。

4.4.21 **Aci_Gap_Terminate**

表 69. Aci_Gap_Terminate

指令名	参数	返回
Aci_Gap_Terminate (0xFC93)	Conn_handle 原因	状态

说明：
命令控制器终止连接。

表 70. Aci_Gap_Terminate 指令参数

参数	大小	说明
Conn_handle	2 字节	要终止的连接的句柄
Reason	1 字节	请求断开的原因。错误代码可以是 HCI 规范中为已断开指令指定的任意代码

表 71. Aci_Gap_Terminate 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x0C: 指令不被允许

生成事件:

当接收到指令时, 控制器将生成指令状态事件, 当链路断开时, 将生成断开完成事件。

4.4.22 Aci_Gap_Clear_Security_Database

表 72. Aci_Gap_Clear_Security_Database

指令名	参数	返回
Aci_Gap_Clear_Security_Database (0xFC94)		状态

说明:

清空安全数据库。安全数据库中的所有设备都将被删除。

表 73. Aci_Gap_Clear_Security_Database 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x4B: Flash 擦除失败

生成事件:

当指令完成时, 控制器将生成指令完成事件。

4.4.23 Aci_Gap_Allow_Rebond

表 74. Aci_Gap_Allow_Rebond

指令名	参数	返回
Aci_Gap_Allow_Rebond (0xFC95)	Connection_handle	状态

表 75. Aci_Gap_Allow_Rebond 指令参数

Connection_Handle	2 字节	需要与之重新绑定的连接的句柄

说明：
如果希望成功实现重新绑定，应用应在接收到 EVT_BLUE_BOND_LOST 事件时发出该指令。如果在接收事件时没有发出该指令，绑定流程将超时。

表 76. Aci_Gap_Allow_Rebond 返回参数

参数	大小	说明
Status	1 字节	0x00：成功

生成事件：
当指令完成时，控制器将生成指令完成事件。如果在指令无效时发出了指令，将返回成功状态，但实质上没有作用。

4.4.24 Aci_Gap_Start_Limited_Discovery_Proc

表 77. Aci_Gap_Start_Limited_Discovery_Proc

指令名	参数	返回
Aci_Gap_Start_Limited_Discovery_Proc (0xFC96)	Scan_Interval Scan_Window Own_Address_Type filterDuplicates	状态

说明：
启动有限发现流程。命令控制器开始主动扫描。当该流程启动时，仅处于有限可发现模式的设备返回上层。

表 78. Aci_Gap_Start_Limited_Discovery_Proc 指令参数

参数	大小	说明
Scan_Interval	2 字节	从控制器启动其上一次 LE 扫描到它开始下一次 LE 扫描之间的时间间隔。扫描间隔应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Window	2 字节	LE 扫描持续的时间长度。Scan_Window 应小于等于 Scan_Interval。扫描窗口应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Own_Address_Type	1 字节	0x00: 公共设备地址（默认） 0x01: 随机设备地址
filterDuplicates	1 字节	0x00: 不过滤重复项（默认） 0x01: 过滤重复项

表 79. Aci_Gap_Start_Limited_Discovery_Proc 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件：

在发出指令时立即生成指令状态事件。如果返回了 BLE_STATUS_SUCCESS，流程将在上层发出终止流程的指令（发出指令 Aci_Gap_Terminate_Gap_Procedure 且流程代码设置为 0x01）时或发生超时时终止。当流程因上述任何原因终止时，返回 Evt_Blue_Gap_Procedure_Complete 事件且流程代码设置为 0x01。在流程正在进行时发现的设备通过 LE_Advertising_Report 事件返回上层。

4.4.25 Aci_Gap_Start_General_Discovery_Proc

表 80. Aci_Gap_Start_General_Discovery_Proc

指令名	参数	返回
Aci_Gap_Start_General_Discovery_Proc (0xFC97)	Scan_Interval Scan_Window Own_Address_Type filterDuplicates	状态

说明：
启动一般发现流程。命令控制器开始主动扫描。

表 81. Aci_Gap_Start_General_Discovery_Proc 指令参数

参数	大小	说明
Scan_Interval	2 字节	从控制器启动其上一次 LE 扫描到它开始下一次 LE 扫描之间的时间间隔。扫描间隔应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Window	2 字节	LE 扫描持续的时间长度。Scan_Window 应小于等于 Scan_Interval。扫描窗口应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Own_Address_Type	1 字节	0x00：公共设备地址（默认） 0x01：随机设备地址
filterDuplicates	1 字节	0x00：不过滤重复项（默认） 0x01：过滤重复项

表 82. Aci_Gap_Start_General_Discovery_Proc 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件：
在发出指令时立即生成指令状态事件。如果返回了 BLE_STATUS_SUCCESS，流程将在上层发出终止流程的指令（发出指令 Aci_Gap_Terminate_Gap_Procedure 且流程代码设置为 0x02）时或发生超时时终止。当流程因上述任何原因终止时，返回 Evt_Blue_Gap_Procedure_Complete 事件且流程代码设置为 0x02。在流程正在进行时发现的设备通过 LE_Advertising_Report 事件返回上层。



4.4.26 Aci_Gap_Start_Name_Discovery_Proc

表 83. Aci_Gap_Start_Name_Discovery_Proc

指令名	参数	返回
Aci_Gap_Start_Name_Discovery_Proc (0xFC98)	Scan_Interval Scan_Window Peer_Address_Type Peer_Address Own_Address_Type Conn_Interval_Min Conn_Interval_Max Conn_Latency Supervision_Timeout Conn_Len_Min Conn_Len_Max	状态

说明：

启动名称发现流程。在发起方过滤策略被设置为“只对特定设备忽略白名单并处理可连接广告数据包”的情况下，GAP 从控制器调用 LE_Create_Connection。连接建立后，GATT 流程启动以读取设备名称特征。当读取完成时（成功或不成功），向上层给出 Evt_Blue_Gap_Procedure_Complete 事件。如果成功读取了设备名称，事件还包含设备名称。

表 84. Aci_Gap_Start_Name_Discovery_Proc 指令参数

参数	大小	说明
Scan_Interval	2 字节	从控制器启动其上一次 LE 扫描到它开始下一次 LE 扫描之间的时间间隔。扫描间隔应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Window	2 字节	LE 扫描持续的时间长度。Scan_Window 应小于等于 Scan_Interval。扫描窗口应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Peer_Address_Type	1 字节	0x00：公共设备地址（默认） 0x01：随机设备地址
Peer_Address	6 字节	必须与之建立连接的对端设备的地址。
Own_Address_Type	1 字节	0x00：公共设备地址（默认） 0x01：随机设备地址
Conn_Interval_Min	2 字节	连接事件间隔的最小值。它应小于等于 Conn_Interval_Max。 范围：0x0006 至 0x0C80 时间 = N * 1.25 ms 时间范围：7.5 ms 至 4 s

表 84. Aci_Gap_Start_Name_Discovery_Proc 指令参数 (续)

参数	大小	说明
Conn_Interval_Max	2 字节	连接事件间隔的最大值。它应大于等于 Conn_Interval_Min。 范围：0x0006 至 0x0C80 时间 = $N * 1.25 \text{ ms}$ 时间范围：7.5 ms 至 4 s
Conn_Latency	2 字节	以连接事件数量表示的连接从设备延迟。 范围：0x0000 至 0x01F4
Supervision_Timeout	2 字节	LE 链路的监控超时。 范围：0x000A 至 0x0C80 时间 = $N * 10 \text{ ms}$ 时间范围：100 ms 至 32 s
Conn_Len_Min	2 字节	LE 连接所需的最小连接事件长度。 范围：0x0000 – 0xFFFF 时间 = $N * 0.625 \text{ ms}$ 。
Conn_Len_Max	2 字节	LE 连接所需的最大连接事件长度。 范围：0x0000 – 0xFFFF 时间 = $N * 0.625 \text{ ms}$ 。

表 85. Aci_Gap_Start_Name_Discovery_Proc 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件：

在发出指令时立即生成指令状态事件。如果返回了 BLE_STATUS_SUCCESS，当流程完成时，返回 Evt_Blue_Gap_Procedure_Complete 事件且流程代码设置为 0x04。

4.4.27 Aci_Gap_Start_Auto_Conn_Establishment

表 86. Aci_Gap_Start_Auto_Conn_Establishment

指令名	参数	返回
Aci_Gap_Start_Auto_Conn_Establishment (0xFC99)	Scan_Interval Scan_Window Own_Address_Type Conn_Interval_Min Conn_Interval_Max Conn_Latency Supervision_Timeout Conn_Len_Min Conn_Len_Max Num_WhiteList_Entries Addr_Array	状态

说明:

启动自动连接建立流程。将指定设备添加到控制器的白名单，在发起方过滤策略被设置为“使用白名单确定要连接的广告方”的情况下，GAP 从控制器调用 LE_Create_Connection。当上层发出指令以终止流程时，GAP 从控制器调用 LE_Create_Connection_Cancel。

表 87. Aci_Gap_Start_Auto_Conn_Establishment 指令参数

参数	大小	说明
Scan_Interval	2 字节	从控制器启动其上一次 LE 扫描到它开始下一次 LE 扫描之间的时间间隔。扫描间隔应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Window	2 字节	LE 扫描持续的时间长度。Scan_Window 应小于等于 Scan_Interval。扫描窗口应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Own_Address_Type	1 字节	0x00: 公共设备地址（默认） 0x01: 随机设备地址
Conn_Interval_Min	2 字节	连接事件间隔的最小值。它应小于等于 Conn_Interval_Max。 范围: 0x0006 至 0x0C80 时间 = N * 1.25 ms 时间范围: 7.5 ms 至 4 s
Conn_Interval_Max	2 字节	连接事件间隔的最大值。它应大于等于 Conn_Interval_Min。 范围: 0x0006 至 0x0C80 时间 = N * 1.25 ms 时间范围: 7.5 ms 至 4 s
Conn_Latency	2 字节	以连接事件数量表示的连接从设备延迟。 范围: 0x0000 至 0x01F4

表 87. Aci_Gap_Start_Auto_Conn_Establishment 指令参数（续）（续）

参数	大小	说明
Supervision_Timeout	2 字节	LE 链路的监控超时。 范围：0x000A 至 0x0C80 时间 = N * 10 ms 时间范围：100 ms 至 32 s
Conn_Len_Min	2 字节	LE 连接所需的最小连接事件长度。 范围：0x0000 – 0xFFFF 时间 = N * 0.625 ms。
Conn_Len_Max	2 字节	LE 连接所需的最大连接事件长度。 范围：0x0000 – 0xFFFF 时间 = N * 0.625 ms。
Num_WhiteList_Entries	1 字节	必须添加到白名单的设备数量。
Addr_Array	N * 7 字节	addr_array 将包含必须添加到白名单的地址及其地址类型的 Num_White_List_Entries 数。addr_array 的格式应是地址类型在先，地址在后。

表 88. Aci_Gap_Start_Auto_Conn_Establishment 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件：

在发出指令时立即生成指令状态事件。如果返回了 BLE_STATUS_SUCCESS，流程将在与白名单中的指定设备之一成功建立连接时或在流程代码设置为 0x08 的情况下通过发出指令 Aci_Gap_Terminate_Gap_Procedure 明确终止流程时终止。返回 Evt_Blue_Gap_Procedure_Complete 事件且流程代码设置为 0x08。

4.4.28 Aci_Gap_Start_General_Conn_Establishment

表 89. Aci_Gap_Start_General_Conn_Establishment

指令名	参数	返回
Aci_Gap_Start_General_Conn_Establishment (0xFC9A)	Scan_Type Scan_Interval Scan_Window Own_Address_Type filterDuplicates	状态

启动通用连接建立流程。在扫描仪过滤策略设置为“接受所有广告数据包”的情况下，主机在控制器中启用扫描，并使用事件 LE_Advertising_Report 将扫描结果中的所有设备发送至上层。然后，上层必须选择一个它要连接（通过发出指令 Aci_Gap_Create_Connection）的设备。如果私有启用，则根据指令中指定的地址类型将扫描仪地址设置为私有可解析地址或不可解析地址，但如果通用连接建立流程处于激活状态，GAP 总是使用私有可解析地址创建连接。

表 90. Aci_Gap_Start_General_Conn_Establishment 指令参数

参数	大小	说明
Scan_Type	1 字节	0x00: 被动扫描 0x01: 主动扫描
Scan_Interval	2 字节	从控制器启动其上一次 LE 扫描到它开始下一次 LE 扫描之间的时间间隔。扫描间隔应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Window	2 字节	LE 扫描持续的时间长度。Scan_Window 应小于等于 Scan_Interval。扫描窗口应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Own_Address_Type	1 字节	如果私有禁用，则扫描仪地址应设置为 0x00: 公共设备地址（默认） 0x01: 随机设备地址 如果私有启用，则扫描仪地址应设置为 0x02: 私有可解析地址 0x03: 不可解析地址
filterDuplicates	1 字节	0x00: 不过滤重复项（默认） 0x01: 过滤重复项

表 91. Aci_Gap_Start_General_Conn_Establishment 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件:

在发出指令时生成指令状态事件。如果返回了状态 BLE_STATUS_SUCCESS，当流程完成时，将生成 Evt_Blue_Gap_Procedure_Completed 事件且流程代码设置为 0x10。流程将在连接建立时或上层在流程代码设置为 0x10 的情况下通过发出指令 Aci_Gap_Terminate_Gap_Procedure 终止流程时终止。

4.4.29 Aci_Gap_Start_Selective_Conn_Establishment

表 92. Aci_Gap_Start_Selective_Conn_Establishment

指令名	参数	返回
Aci_Gap_Start_Selective_Conn_Establishment (0xFC9B)	Scan_Type Scan_Interval Scan_Window Own_Address_Type filterDuplicates Num_WhiteList_Entries Addr_Array	状态

说明:

启动选择性连接建立流程。GAP 将指定设备地址添加到白名单中，并在扫描仪过滤策略设置为“仅接受来自白名单中的设备的数据包”的控制器中启用扫描。通过事件 LE_Advertising_Report 将找到的所有设备发送至上层。然后，上层必须选择一个它要连接（通过发出指令 Aci_Gap_Create_Connection）的设备。

表 93. Aci_Gap_Start_Selective_Conn_Establishment 指令参数

参数	大小	说明
Scan_Type	1 字节	0x00: 被动扫描 0x01: 主动扫描
Scan_Interval	2 字节	从控制器启动其上一次 LE 扫描到它开始下一次 LE 扫描之间的时间间隔。扫描间隔应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Window	2 字节	LE 扫描持续的时间长度。Scan_Window 应小于等于 Scan_Interval。扫描窗口应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Own_Address_Type	1 字节	0x00: 公共设备地址（默认） 0x01: 随机设备地址

表 93. Aci_Gap_Start_Selective_Conn_Establishment 指令参数（续）

参数	大小	说明
filterDuplicates	1 字节	0x00: 不过滤重复项（默认） 0x01: 过滤重复项
Num_WhiteList_Entries	1 字节	必须添加到白名单的设备数量。
Addr_Array	7 字节	addr_array 将包含必须添加到白名单的地址及其地址类型的 Num_White_List_Entries 数。addr_array 的格式应是地址类型在先，地址在后。

表 94. Aci_Gap_Start_General_Conn_Establishment 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件：

在发出指令时生成指令状态事件。如果返回了状态 BLE_STATUS_SUCCESS，当流程完成时，将生成 Evt_Blue_Gap_Procedure_Completed 事件且流程代码设置为 0x20。流程将在连接建立时或上层在流程代码设置为 0x20 的情况下通过发出指令 Aci_Gap_Terminate_Gap_Procedure 终止流程时终止。

4.4.30 Aci_Gap_Create_Connection

表 95. Aci_Gap_Create_Connection

指令名	参数	返回
Aci_Gap_Create_Connection (0xFC9C)	Scan_Interval Scan_Window Peer_Address_Type Peer_Address Own_Address_Type Conn_Interval_Min Conn_Interval_Max Conn_Latency Supervision_Timeout Conn_Len_Min Conn_Len_Max	状态

说明:

启动直接连接建立流程。在发起方过滤策略被设置为“只对特定设备忽略白名单并处理可连接广告数据包”的情况下，GAP 从控制器调用 LE_Create_Connection。上层可通过发出指令 Aci_Gap_Terminate_Gap_Procedure 明确终止流程。当上层发出指令以终止流程时，GAP 从控制器调用 LE_Create_Connection_Cancel。

表 96. Aci_Gap_Create_Connection 指令参数

参数	大小	说明
Scan_Interval	2 字节	从控制器启动其上一次 LE 扫描到它开始下一次 LE 扫描之间的时间间隔。扫描间隔应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Window	2 字节	LE 扫描持续的时间长度。Scan_Window 应小于等于 Scan_Interval。扫描窗口应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Peer_Address_Type	1 字节	0x00: 公共设备地址（默认） 0x01: 随机设备地址
Peer_Address	6 字节	必须与之建立连接的对端设备的地址。
Own_Address_Type	1 字节	0x00: 公共设备地址（默认） 0x01: 随机设备地址
Conn_Interval_Min	2 字节	连接事件间隔的最小值。它应小于等于 Conn_Interval_Max。 范围: 0x0006 至 0x0C80 时间 = N * 1.25 ms 时间范围: 7.5 ms 至 4 s

表 96. Aci_Gap_Create_Connection 指令参数 (续)

参数	大小	说明
Conn_Interval_Max	2 字节	连接事件间隔的最大值。它应大于等于 Conn_Interval_Min。 范围：0x0006 至 0x0C80 时间 = $N * 1.25 \text{ ms}$ 时间范围：7.5 ms 至 4 s
Conn_Latency	2 字节	以连接事件数量表示的连接从设备延迟。 范围：0x0000 至 0x01F4
Supervision_Timeout	2 字节	LE 链路的监控超时。 范围：0x000A 至 0x0C80 时间 = $N * 10 \text{ ms}$ 时间范围：100 ms 至 32 s
Conn_Len_Min	2 字节	LE 连接所需的最小连接事件长度。 范围：0x0000 – 0xFFFF 时间 = $N * 0.625 \text{ ms}$ 。
Conn_Len_Max	2 字节	LE 连接所需的最大连接事件长度。 范围：0x0000 – 0xFFFF 时间 = $N * 0.625 \text{ ms}$ 。

表 97. Aci_Gap_Create_Connection 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件：

在发出指令时立即生成指令状态事件。如果返回了 BLE_STATUS_SUCCESS，当流程终止时，将返回 LE_Connection_Complete 事件。上层可通过在 procedure_code 设置为 0x40 时发出指令 Aci_Gap_Terminate_Gap_Procedure 明确终止流程。

4.4.31 Aci_Gap_Terminate_Gap_Procedure

表 98. Aci_Gap_Terminate_Gap_Procedure

指令名	参数	返回
Aci_Gap_Terminate_Gap_Procedure (0xFC9D)	Procedure_Code	状态

说明：
终止指定的 GAP 流程。

表 99. Aci_Gap_Terminate_Gap_Procedure 指令参数

参数	大小	说明
Procedure_Code	1 字节	0x01: LIMITED_DISCOVERY_PROC 0x02: GENERAL_DISCOVERY_PROC 0x04: NAME_DISCOVERY_PROC 0x08: AUTO_CONNECTION_ESTABLISHMENT_PROC 0x10: GENERAL_CONNECTION_ESTABLISHMENT_PROC 0x20: SELECTIVE_CONNECTION_ESTABLISHMENT_PROC 0x40: DIRECT_CONNECTION_ESTABLISHMENT_PROC 0x80: OBSERVATION_PROC

表 100. Aci_Gap_Terminate_Gap_Procedure 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER

生成事件：
 为该指令生成指令完成事件。如果成功处理了指令，状态字段将返回值
 BLE_STATUS_SUCCESS 和 Evt_Blue_Gap_Procedure_Completed 事件且流程代码设置为
 对应流程。

4.4.32 Aci_Gap_Start_Connection_Update

表 101. Aci_Gap_Start_Connection_Update

指令名	参数	返回
Aci_Gap_Start_Connection_Update (0xFC9E)	Conn_Handle Conn_Interval_Min Conn_Interval_Max Conn_Latency Supervision_Timeout Conn_Len_Min Conn_Len_Max	状态

说明：
 启动连接更新流程。 GAP 从控制器调用 LE_Connection_Update

表 102. Aci_Gap_Start_Connection_Update 指令参数

参数	大小	说明
Conn_Handle	2 字节	必须启动更新流程的连接句柄。
Conn_Interval_Min	2 字节	连接事件间隔的最小值。它应小于等于 Conn_Interval_Max。 范围：0x0006 至 0x0C80 时间 = N * 1.25 ms 时间范围：7.5 ms 至 4 s
Conn_Interval_Max	2 字节	连接事件间隔的最大值。它应大于等于 Conn_Interval_Min。 范围：0x0006 至 0x0C80 时间 = N * 1.25 ms 时间范围：7.5 ms 至 4 s
Conn_Latency	2 字节	以连接事件数量表示的连接从设备延迟。 范围：0x0000 至 0x01F4
Supervision_Timeout	2 字节	LE 链路的监控超时。 范围：0x000A 至 0x0C80 时间 = N * 10 ms 时间范围：100 ms 至 32 s

表 102. Aci_Gap_Start_Connection_Update 指令参数（续）

参数	大小	说明
Conn_Len_Min	2 字节	LE 连接所需的最小连接事件长度。 范围：0x0000 – 0xFFFF 时间 = N * 0.625 ms。
Conn_Len_Max	2 字节	LE 连接所需的最大连接事件长度。 范围：0x0000 – 0xFFFF 时间 = N * 0.625 ms。

表 103. Aci_Gap_Start_Connection_Update 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件：

在发出指令时立即生成指令状态事件。如果返回了 BLE_STATUS_SUCCESS，当连接更新完成时，向上层返回 LE_Connection_Update_Complete 事件。

4.4.33 Aci_Gap_Send_Pairing_Request

表 104. Aci_Gap_Send_Pairing_Request

指令名	参数	返回
Aci_Gap_Send_Pairing_Request (0xFC9F)	Conn_Handle Force_Rebond	状态

说明：

发送启动配对过程的 SM 配对请求。在使用 Aci_Set_IO_Capabilities 和 Aci_Set_Authentication_Requirements 发出该指令之前，应设置验证要求和 IO 能力。

表 105. Aci_Gap_Send_Pairing_Request 指令参数

参数	大小	说明
Conn_Handle	2 字节	必须发送配对请求的连接句柄。
Force_Rebond	1 字节	第 0 位决定了在设备之前已绑定或没有绑定时是否必须发送配对请求。 0: 仅在设备之前没有绑定时发送配对请求 1: 即使设备之前已绑定, 也将发送配对请求。 第 1 位决定了交换密钥后是否必须重新加密链路。 0: 在配对结束时, 将不使用 LTK 重新加密链路 1: 在配对结束时, 将使用 LTK 重新加密链路

表 106. Aci_Gap_Send_Pairing_Request 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

生成事件:

在接收指令时生成指令状态事件。如果在指令状态事件中返回了 BLE_STATUS_SUCCESS, 将在配对过程完成后返回 Evt_Blue_Pairing_Complete 事件。

4.4.34 Aci_Gap_Resolve_Private_Address

表 107. Aci_Gap_Resolve_Private_Address

指令名	参数	返回
Aci_Gap_Resolve_Private_Address (0xFCA0)	Peer_address	状态 地址

说明:

该指令尝试使用其数据库中的 IRK 解析地址。如果已使用数据库中的任何一个 IRK 成功解析地址, 将返回成功事件和使用该 IRK 存储在数据库中的对应公共 / 静态随机地址。

表 108. Aci_Gap_Resolve_Private_Address 指令参数

参数	大小	说明
Peer_address	6 字节	要解析的地址。

表 109. Aci_Gap_Resolve_Private_Address 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x48: 地址未解析
Address	6 字节	对端设备的公共或静态随机地址，在配对阶段分配。

生成事件：

生成指令完成事件。如果在状态字节中返回了 BLE_STATUS_SUCCESS，则还将在事件中返回地址。

4.4.35 Aci_Gap_Get_Bonded_Devices

表 110. Aci_Gap_Get_Bonded_Devices

指令名	参数	返回
Aci_Gap_Get_Bonded_Devices(0xFCA3)	-	状态 Number_of_addresses_to_follow List_of_addr_type-address

描述：该指令获取已绑定设备的列表。它返回地址数量和对应的地址类型及值。

表 111. Aci_Gap_Get_Bonded_Devices 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED
Number_of_addresses_to_follow	1 字节	后续地址类型 - 地址对的数量
List_of_addr_type-address	7 字节	第一个字节指示地址类型： 0x00- 公共地址 0x01 随机 剩余 6 个字节将包含已绑定设备的地址

生成事件：
 返回指令完成事件，其中的状态指示指令是否成功。如果成功，后面是 Address_Type-Address 对列表（如上文所述）。

4.4.36 Aci_Gap_Set_Broadcast_Mode

表 112. Aci_Gap_Set_Broadcast_Mode

指令名	参数	返回
Aci_Gap_Set_Broadcast_Mode (0xFCA1)	Advertising_Interval_Min Advertising_Interval_Max Advertising_Type Own_Address_Type adv_data_len adv_data Num_WhiteList_Entries 白名单	状态

说明：
 该指令使设备进入广播模式。启用了私有的设备按照指令的指定 Own_Addr_Type 参数使用可解析私有地址或不可解析私有地址。

表 113. Aci_Gap_Set_Broadcast_Mode 指令参数

参数	大小	说明
Advertising_Interval_Min	2 字节	最小广告间隔。 范围：0x00A0 至 0x4000 时间 = N * 0.625 ms 时间范围：100 ms 至 10.24 s
Advertising_Interval_Max	2 字节	最大广告间隔。 范围：0x00A0 至 0x4000 时间 = N * 0.625 ms 时间范围：100 ms 至 10.24 s
Advertising_Type	1 字节	0x02：可扫描非定向广告 0x03：不可连接非定向广告
Own_Address_Type	1 字节	如果私有禁用： 0x00：公共地址 0x01：静态随机地址 如果私有启用： 0x02：可解析私有地址 0x03：不可解析私有地址
adv_data_len	1 字节	广告数据包中广告数据的长度
adv_data	adv_data_len bytes	广告时设备使用的广告数据

表 113. Aci_Gap_Set_Broadcast_Mode 指令参数

参数	大小	说明
Num_WhiteList_Entries	1 字节	要添加到白名单的设备数量
whitelist	Num_WhiteList_Entries * 7 字节	addr_array 将包含必须添加到白名单的地址及其地址类型的 Num_White_List_Entries 数。addr_array 的格式应是地址类型在先，地址在后。

表 114. Aci_Gap_Set_Broadcast_Mode 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: HCI_INVALID_PARAMETER 0x0C: 指令不被允许

生成事件：
返回指令完成事件，其中的状态指示指令是否成功。

4.4.37 Aci_Gap_Start_Observation_Proc

表 115. Aci_Gap_Start_Observation_Proc

指令名	参数	返回
Aci_Gap_Start_Observation_Proc (0xFCA2)	Scan_Type Scan_Interval Scan_Window Own_Address_Type filterDuplicates	状态



说明:

当设备处于监听方角色时，启动监听流程。主机在控制器中启用扫描。使用标准 LE 广告报告事件将广告报告发送至上层。参见蓝牙核心规范 v4.1 第 2 卷 E 部分第 7.7.65.2 节“LE 广告报告事件”。

指令参数:

表 116. Aci_Gap_Start_Observation_Proc 指令参数

参数	大小	说明
Scan_Interval	2 字节	从控制器启动其上一次 LE 扫描到它开始下一次 LE 扫描之间的时间间隔。扫描间隔应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Window	2 字节	LE 扫描持续的时间长度。Scan_Window 应小于等于 Scan_Interval。扫描窗口应为范围 0x0004 至 0x4000 之间的数字。这对应于时间范围 2.5 ms 至 10240 ms。对于数字 N，时间 = N * 0.625 ms。
Scan_Type	1 字节	0x00: 被动扫描 0x01: 主动扫描
Own_Address_Type	1 字节	如果私有禁用，则扫描仪地址应设置为 0x00: 公共设备地址（默认） 0x01: 随机设备地址 如果私有启用，则扫描仪地址应设置为 0x02: 私有可解析地址 0x03: 不可解析地址
filterDuplicates	1 字节	0x00: 不过滤重复项（默认） 0x01: 过滤重复项

表 117. Aci_Gap_Start_Observation_Proc 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: BLE_STATUS_INVALID_PARAMETER 0x0C: ERR_COMMAND_DISALLOWED

4.4.38 Aci_Gap_Is_Device_Bonded

表 118. Aci_Gap_Is_Device_Bonded

指令名	参数	返回
Aci_Gap_Is_Device_Bonded (0xFCA4)	Peer_address_type Peer_address	状态

说明：
指令检查指令中指定地址所属的设备是否已绑定。如果设备使用可解析私有地址并已绑定，则指令将返回 BLE_STATUS_SUCCESS。

表 119. Aci_Gap_Is_Device_Bonded 指令参数

参数	大小	说明
Peer_address_type	1 字节	对端设备的地址类型
Peer_address	6 字节	广告时对端设备使用的地址。

表 120. Aci_Gap_Is_Device_Bonded 返回参数

参数	大小	说明
Status	1 字节	0x00 : BLE_STATUS_SUCCESS 0x12 : HCI_INVALID_PARAMETER 0x5C: DEV_NOT_FOUND_IN_DATABASE 0x49 : FLASH_READ_FAILED

生成事件：
生成指令完成事件。如果在事件的状态字节中返回了 BLE_STATUS_SUCCESS，则设备已绑定。

4.5 GAP VS 事件

表 121. GAP VS 事件

项目	命令	EGID	EID	ECODE
1	Evt_Blue_Gap_Limited_Discoverable	0x01	0x00	0x0400
2	Evt_Blue_Gap_Pairing_Cmplt	0x01	0x01	0x0401
3	Evt_Blue_Gap_Pass_Key_Request	0x01	0x02	0x0402
4	Evt_Blue_Gap_Authorization_Request	0x01	0x03	0x0403
5	Evt_Blue_Gap_Slave_Security_Initiated	x01	0x04	0x0404
6	Evt_Blue_Gap_Bond_Lost	0x01	0x05	0x0405
7	Evt_Blue_Gap_Procedure_Complete	0x01	0x07	0x0407
8	Evt_Blue_Gap_Addr_Not_Resolved	0x01	0x08	0x0408

4.5.1 Evt_Blue_Gap_Limited_Discoverable

当有限可发现模式因超时而结束时，控制器生成该事件。超时为 180 秒。

4.5.2 Evt_Blue_Gap_Pairing_Cmplt

当配对过程成功完成或发生配对流程超时或配对失败时，生成该事件。其目的是通知应用，已与远程设备配对因此可以执行进一步操作，或者发生了超时因此上层可以决定断开链路。

表 122. Evt_Blue_Gap_Pairing_Cmplt

参数	大小	说明
Event Code	2 字节	0x0401
Connection_handle	2 字节	配对流程已完成的连接句柄
Status	1 字节	0x00：配对成功 与远程设备的配对成功 0x01：配对超时 SMP 超时已结束且在重新连接前不会处理更多 SMP 指令 0x02：配对失败 与远程设备的配对失败

4.5.3 Evt_Blue_Gap_Pass_Key_Request

当需要输入密钥用于配对时，由安全管理器向应用生成该事件。在接收到该事件时，应用必须以 Aci_Gap_Pass_Key_Response 指令响应。

表 123. Evt_Blue_Gap_Pass_Key_Request

参数	大小	说明
Connection_handle	2 字节	请求了输入密钥的连接句柄

4.5.4 Evt_Blue_Gap_Authorization_Request

当应用设置了需要授权才能读取 / 写入属性时，由安全管理器向应用生成该事件。将在配对完成时立即生成该事件。当接收到该事件时，应用应使用 Aci_Gap_Authorization_Response 指令响应。

表 124. Evt_Blue_Gap_Authorization_Request

参数	大小	说明
Event Code	2 字节	0x0403
Connection_handle	2 字节	正在请求授权的连接句柄

4.5.5 Evt_Blue_Gap_Slave_Security_Initiated

当从设备安全请求成功发送至主设备时，生成该事件。

4.5.6 Evt_Blue_Gap_Bond_Lost

在配对请求发出时生成该事件，以响应来自主设备（之前已与从设备绑定）的从设备安全请求。在接收到该事件时，上层必须发出指令 Aci_Gap_Allow_Rebond，以允许从设备继续完成与主设备的配对过程。

4.5.7 Evt_Blue_Gap_Procedure_Complete

当之前启动的流程已被上层终止或因任何其他原因已完成时，GAP 向上层发送该事件。

表 125. Evt_Blue_Gap_Procedure_Complete

参数	大小	说明
Event Code	2 字节	0x0407
Procedure_Code	1 字节	0x01: LIMITED_DISCOVERY_PROC 0x02: GENERAL_DISCOVERY_PROC 0x04: NAME_DISCOVERY_PROC 0x08: AUTO_CONNECTION_ESTABLISHMENT_PROC 0x10: GENERAL_CONNECTION_ESTABLISHMENT_PROC 0x20: SELECTIVE_CONNECTION_ESTABLISHMENT_PROC 0x40: DIRECT_CONNECTION_ESTABLISHMENT_PROC
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x41: BLE_STATUS_FAILED 0x05: ERR_AUTH_FAILURE, 验证要求导致流程失败
Procedure_Specific_Data	1 个或多个字节	流程 Aci_Gap_Name_Discovery_Proc: 如果流程成功完成，为对端设备的名称 流程 Aci_Gap_General_Conn_Establishment_Proc: 如果为外设启用了私有，为写入外设的重新连接地址

4.5.8 Evt_Blue_Gap_Addr_Not_Resolved

该事件仅由启用了私有的外设发送。当外设在连接对端设备后没有成功解析对端设备的可解析地址时，向上层发送该事件。

表 126. Evt_Blue_Gap_Addr_Not_Resolved

参数	大小	说明
Event Code	2 字节	0x0408
Connection_handle	2 字节	其私有地址不能使用存储的任何 IRK 进行解析的连接句柄。

4.6 GATT VS 指令

表 127. GATT VS 指令

项目	命令	CGID	CID	操作码
1	Aci_Gatt_Init	0x02	0x01	0xFD01
2	Aci_Gatt_Add_Serv	0x02	0x02	0xFD02
3	Aci_Gatt_Include_Service	0x02	0x03	0xFD03
4	Aci_Gatt_Add_Char	0x02	0x04	0xFD04
5	Aci_Gatt_Add_Char_Desc	0x02	0x05	0xFD05
6	Aci_Gatt_Update_Char_Value	0x02	0x06	0xFD06
7	Aci_Gatt_Del_Char	0x02	0x07	0xFD07
8	Aci_Gatt_Del_Service	0x02	0x08	0xFD08
9	Aci_Gatt_Del_Include_Service	0x02	0x09	0xFD09
10	Aci_Gatt_Set_Event_Mask	0x02	0x0A	0xFD0A
11	Aci_Gatt_Exchange_configuration	0x02	0x0B	0xFD0B
12	Aci_Att_Find_Information_Req	0x02	0x0C	0xFD0C
13	Aci_Att_Find_By_Type_Value_Req	0x02	0x0D	0xFD0D
14	Aci_Att_Read_By_Type_Req	0x02	0x0E	0xFD0E
15	Aci_Att_Read_By_Group_Type_Req	0x02	0x0F	0xFD0F
16	Aci_Att_Prepare_Write_Req	0x02	0x10	0xFD10
17	Aci_Att_Execute_Write_Req	0x02	0x11	0xFD11
18	Aci_Gatt_Disc_All_Prim_Services	0x02	0x12	0xFD12
19	Aci_Gatt_Disc_Prim_Service_By_UUID	0x02	0x13	0xFD13
20	Aci_Gatt_Find_Included_Services	0x02	0x14	0xFD14
21	Aci_Gatt_Disc_All_Charac_Of_Serv	0x02	0x15	0xFD15
22	Aci_Gatt_Disc_Charac_By_UUID	0x02	0x16	0xFD16
23	Aci_Gatt_Disc_All_Charac_Descriptors	0x02	0x17	0xFD17

表 127. GATT VS 指令 (续)

项目	命令	CGID	CID	操作码
24	Aci_Gatt_Read_Charac_Val	0x02	0x18	0xFD18
25	Aci_Gatt_Read_Charac_Using_UUID	0x02	0x19	0xFD19
26	Aci_Gatt_Read_Long_Charac_Val	0x02	0x1A	0xFD1A
27	Aci_Gatt_Read_Multiple_Charac_Val	0x02	0x1B	0xFD1B
28	Aci_Gatt_Write_Charac_Value	0x02	0x1C	0xFD1C
29	Aci_Gatt_Write_Long_Charac_Val	0x02	0x1D	0xFD1D
30	Aci_Gatt_Write_Charac_Reliable	0x02	0x1E	0xFD1E
31	Aci_Gatt_Write_Long_Charac_Desc	0x02	0x1F	0xFD1F
32	Aci_Gatt_Read_Long_Charac_Desc	0x02	0x20	0xFD20
33	Aci_Gatt_Write_Charac_Descriptor	0x02	0x21	0xFD21
34	Aci_Gatt_Read_Charac_Desc	0x02	0x22	0xFD22
35	Aci_Gatt_Write_Without_Response	0x02	0x23	0xFD23
36	Aci_Gatt_Signed_Write_Without_Resp	0x02	0x24	0xFD24
37	Aci_Gatt_Confirm_Indication	0x02	0x25	0xFD25
38	Aci_Gatt_Write_Response	0x02	0x26	0xFD26
39	Aci_Gatt_Allow_Read	0x02	0x27	0xFD27
40	Aci_Gatt_Set_Security_Permission	0x02	0x28	0xFD28
41	Aci_Gatt_Set_Desc_Value	0x02	0x29	0xFD29
42	Aci_Gatt_Read_Handle_Value	0x02	0x2A	0xFD2A
43	Aci_Gatt_Read_Handle_Value_Offset	0x02	0x2B	0xFD2B
44	Aci_Gatt_Update_Char_Value_Ext	0x02	2C	0xFD2C

4.6.1 Aci_Gatt_Init

表 128. Aci_Gatt_Init

指令名	参数	返回
Aci_Gatt_Init (0xFD01)		状态

说明：
 在从设备上初始化 GATT 服务器。初始化所有缓存和活动节点。另外，它还添加具有服务更改特征的 GATT 服务。在发出该指令前，GATT 通道将不处理任何指令，即使连接打开。在使用任何 GAP 特性前，必须发出该指令。

表 129. Aci_Gatt_Init 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误

4.6.2 Aci_Gatt_Add_Serv

表 130. Aci_Gatt_Add_Serv

指令名	参数	返回
Aci_Gatt_Add_Serv (0xFD02)	Service_UUID_Type Service_UUID Service_Type Max_Attribute_Records	状态 Service_handle

说明：
 为 GATT 服务器添加服务。在服务器中创建服务时，主机需使用 Max_Attribute_Records 参数为该服务预留句柄范围。该参数指定了可以添加到该服务的属性记录的最大数量（包括服务属性、包含属性、特征属性、特征值属性和特征描述符属性）。在指令完成事件中返回所创建服务的句柄。

表 131. Aci_Gatt_Add_Serv 指令参数

参数	大小	说明
Service_UUID_Type	1 字节	0x01: 16 位 UUID 0x02: 128 位 UUID
Service_UUID	2 字节或 16 字节	16 位或 128 位 UUID（取决于 UUID 类型字段）
Service_Type	1 字节	0x01: 主要服务 0x02: 次要服务
Max_Attribute_Records	1 字节	可添加到该服务的属性记录的最大数量

表 132. Aci_Gatt_Add_Serv 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误 0x1F: 内存不足 0x61: 无效参数 0x62: 缺少句柄 0x64: 资源不足
Service_handle	2 字节	服务的句柄 在将该服务添加到服务时，服务器为该服务分配句柄。此外，服务器还为该服务分配从 Service_Handle 到 (Service_Handle + Max_Attribute_Records) 的句柄范围。

生成事件：
当 Aci_Gatt_Add_Serv 指令完成时，控制器将生成指令完成事件并以服务的状态和句柄作为参数。

4.6.3 Aci_Gatt_Include_Service

表 133. Aci_Gatt_Include_Service

指令名	参数	返回
Aci_Gatt_Include_Service (0xFD03)	Service_Handle Include_Start_Handle Include_End_Handle Included_Uuid_Type Included_Uuid	状态 Included_handle

说明：
将由 Service_Include_Handle 给出的服务包含到由 Service_Handle 给出的另一个服务。属性服务器创建 INCLUDE 定义属性，并在指令完成事件中以 Included_Handle 的形式返回该定义属性的句柄。

表 134. Aci_Gatt_Include_Service 指令参数

参数	大小	说明
Service_handle	2 字节	必须包含另一个服务的服务的句柄
Include_Start_Handle	2 字节	服务中必须包括的服务的开始句柄
Include_End_Handle	2 字节	服务中必须包括的服务的结束句柄
Include_UUID_Type	1 字节	0x01: 16 位 UUID 0x02: 128 位 UUID
Include_UUID	2 字节或 16 字节	服务的 16 位或 128 位 UUID

表 135. Aci_Gatt_Include_Service 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误 0x1F: 内存不足 0x60: 无效句柄 0x61: 无效参数 0x62: 缺少句柄 0x63: 无效操作 0x64: 资源不足
Included_handle	2 字节	包含声明的句柄

生成事件:

当 Aci_Gatt_Add_Serv 指令完成时, 控制器将生成指令完成事件并以服务的状态和句柄作为参数。

4.6.4 Aci_Gatt_Add_Char

表 136. Aci_Gatt_Add_Char

指令名	参数	返回
Aci_Gatt_Add_Char (0xFD04)	Service_Handle Char_UUID_Type Char_UUID Char_Value_Length Char_Properties Security_Permissions Gatt_Evt_Mask Encryption_Key_Size isVariable	Char_handle

说明：

为服务添加特征。

表 137. Aci_Gatt_Add_Char 指令参数

参数	大小	说明
Service_Handle	2 字节	必须添加特征的服务的句柄
Char_UUID_Type	1 字节	0x01: 16 位 UUID 0x02: 128 位 UUID
Char_UUID	2 字节或 16 字节	16 位或 128 位 UUID
Char_Value_Length	1 字节	特征值的最大长度
Char_Properties	1 字节	特征属性的按位或（如蓝牙规范 4.1 第 3 卷第 3.3.3.1 节所定义）
Security_Permissions	1 字节	0x00: ATTR_PERMISSION_NONE 0x01: 需要验证才能读取 0x02: 需要授权才能读取 0x04: 应加密链路才能读取 0x08: 需要验证才能写入 0x10: 需要授权才能写入 0x20: 应加密链路才能写入
Gatt_Evt_Mask	1 字节	0x01: GATT_SERVER_ATTR_WRITE 当客户端写入此属性时，将通知应用 0x02: GATT_INTIMATE_AND_WAIT_FOR_APPL_AUTH 当服务器接收到该属性的写入请求 / 写入指令 / 签名写入指令时，将通知应用。 0x04: GATT_INTIMATE_APPL_WHEN_READ_N_WAIT 当获取了该属性的任何类型的读取请求时，将通知应用
Encryption_Key_Size	1 字节	该属性的加密密钥最小长度要求。有效范围：7 至 16
isVariable	1 字节	0x00: 属性具有长度固定的值字段 0x01: 属性具有长度可变的值字段

表 138. Aci_Gatt_Add_Char 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误 0x1F: 内存不足 0x60: 无效句柄 0x61: 无效参数 0x62: 缺少句柄 0x64: 资源不足 0x66: 字符已存在
Char_Handle	2 字节	特征的句柄

生成事件：
当指令完成时，控制器将生成指令完成事件，该事件以参数的形式承载着指令状态和特征句柄。

4.6.5 Aci_Gatt_Add_Char_Desc

表 139. Aci_Gatt_Add_Char_Desc

指令名	参数	返回
Aci_Gatt_Add_Char_Desc (0xFD05)	Service_Handle Char_Handle Char_Desc_UUID_Type Char_Desc_UUID Char_Desc_Value_Max_Length Char_Desc_Value_Length Char_Desc_Value Security_Permissions Access_Permissions Gatt_Evt_Mask Encryption_Key_Size isVariable	状态 Char_desc_handle

说明：
为服务添加特征描述符。

表 140. Aci_Gatt_Add_Char_Desc 指令参数

参数	大小	说明
Service_Handle	2 字节	特征所属服务的句柄
Char_Handle	2 字节	要添加描述的特征的句柄。
Char_Desc_UUID 类型	1 字节	0x01: 16 位 UUID (默认) 0x02: 128 位 UUID
Char_Desc_UUID	2 字节或 16 字节	0x2900: 特征扩展属性描述符 0x2901: 特征用户描述符 0x2902: 客户端配置描述符 0x2903: 服务器配置描述符 0x2904: 特征表达格式 0x2905: 特征聚合格式 0XXXXX: 16/128 位用户特定的描述符
Char_Desc_Value_Max_Length	1 字节	描述符值的最大长度
Char_Desc_Value_Length	1 字节	特性描述符值的当前长度
Char_Desc_Value	0-N 字节	特征描述的值
Security_Permissions	1 字节	0x00: 无安全许可 0x01: 要求验证 0x02: 要求授权 0x04: 要求加密
Access_Permissions	1 字节	0x00: 不能访问 0x01: 可读 0x02: 可写 0x03: 读 / 写
Gatt_Event_Mask	1 字节	0x01: GATT_SERVER_ATTR_WRITE 当客户端写入此属性时, 将通知应用 0x02: GATT_INTIMATE_AND_WAIT_FOR_APPL_AUTH 当服务器接收到该属性的写入请求 / 写入指令 / 签名 写入指令时, 将通知应用。 0x04: GATT_INTIMATE_APPL_WHEN_READ_N_WAIT 当获取了该属性的任何类型的读取请求时, 将通知应用
Encryption_Key_Size	1 字节	该属性的加密密钥最小长度要求。有效范围: 7 至 16
isVariable	1 字节	0x00: 属性具有长度固定的值字段 0x01: 属性具有长度可变的值字段

表 141. Aci_Gatt_Add_Char_Desc 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误 0x1F: 内存不足 0x60: 无效句柄 0x61: 无效参数 0x62: 缺少句柄 0x63: 无效操作 0x64: 资源不足
Char_Desc_Handle	2 字节	特征描述符的句柄

生成事件:

当该指令完成时, 控制器将生成指令完成事件, 该事件承载着指令状态和特征描述符句柄。

4.6.6 Aci_Gatt_Update_Char_Value

表 142. Aci_Gatt_Update_Char_Value

指令名	参数	返回
Aci_Gatt_Update_Char_Value (0xFD06)	Serv_Handle Char_Handle Val_Offset Char_Value_Length Char_Value	状态

说明:

更新服务中的特征值。

表 143. Aci_Gatt_Update_Char_Value 指令参数

参数	大小	说明
Serv_Handle	2 字节	特征所属服务的句柄
Char_Handle	2 字节	特征的句柄
Val_Offset	1 字节	必须更新属性值的偏移。如果将其设置为 0 且属性值的长度可变, 则将属性的长度设置为 Char_Value_Length。如果将 Val_Offset 设置为大于 0 的值, 则将属性长度设置为在添加特征时为该属性指定的最大长度。
Char_Value_Length	1 字节	特征值的八位组长度
Char_Value	0-N 字节	特征值

表 144. Aci_Gatt_Update_Char_Value 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误 0x60: 无效句柄 0x61: 无效参数 0x64: 资源不足

生成事件：
当指令完成时，控制器将生成指令完成事件。

4.6.7 Aci_Gatt_Del_Char

表 145. Aci_Gatt_Del_Char

指令名	参数	返回
Aci_Gatt_Del_Char(0xFD07)	Serv_Handle Char_Handle	状态

说明：
从服务中删除指定特征。

表 146. Aci_Gatt_Del_Char 指令参数

参数	大小	说明
Serv_Handle	2 字节	特征所属服务的句柄
Char_Handle	2 字节	要删除描述的特征的句柄



表 147. Aci_Gatt_Del_Char 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误 0x60: 无效句柄 0x61: 无效参数 0x64: 资源不足

生成事件:

当 Aci_Gatt_Del_Char 指令完成时, 控制器将生成指令完成事件。

4.6.8 Aci_Gatt_Del_Service

表 148. Aci_Gatt_Del_Service

指令名	参数	返回
Aci_Gatt_Del_Service (0xFD08)	Service_handle	状态

说明:

从 GATT 服务器数据库中删除指定服务。

表 149. Aci_Gatt_Del_Service 指令参数

参数	大小	说明
Service_Handle	2 字节	要删除的服务的句柄

表 150. Aci_Gatt_Del_Service 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误 0x60: 无效句柄 0x61: 无效参数 0x64: 资源不足

生成事件:

当指令完成时, 控制器将生成指令完成事件。

4.6.9 Aci_Gatt_Del_Include_Service

表 151. Aci_Gatt_Del_Include_Service

指令名	参数	返回
Aci_Gatt_Del_Include_Service	Serv_Handle Include_Handle	状态

说明：
从服务中删除包含定义。

表 152. Aci_Gatt_Del_Include_Service 指令参数

参数	大小	说明
Serv_Handle	2 字节	包含定义所属服务的句柄
Include_Handle	2 字节	要删除的包含定义的句柄

表 153. Aci_Gatt_Del_Include_Service 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x47：错误 0x60：无效句柄 0x61：无效参数 0x64：资源不足

生成事件：
当指令完成时，控制器将生成指令完成事件。

4.6.10 Aci_Gatt_Set_Event_Mask

表 154. Aci_Gatt_Set_Event_Mask

指令名	参数	返回
Aci_Gatt_Set_Event_Mask (0xFD0A)	事件屏蔽	状态

说明：
它允许来自 GATT 的屏蔽事件。默认配置为屏蔽所有事件。

表 155. Aci_Gatt_Set_Event_Mask 指令参数

参数	大小	说明
Event mask	4 字节	0x00000001: Evt_Blue_Gatt_Attribute_modified 0x00000002: Evt_Blue_Gatt_Procedure_Timeout 0x00000004: Evt_Blue_Att_Exchange_MTU_Resp 0x00000008: Evt_Blue_Att_Find_Information_Resp 0x00000010: Evt_Blue_Att_Find_By_Type_Value_Resp 0x00000020: Evt_Blue_Att_Read_By_Type_Resp 0x00000040: Evt_Blue_Att_Read_Resp 0x00000080: Evt_Blue_Att_Read_Blob_Resp 0x00000100: Evt_Blue_Att_Read_Multiple_Resp 0x00000200: Evt_Blue_Att_Read_By_Group_Resp 0x00000800: Evt_Blue_Att_Prepare_Write_Resp 0x00001000: Evt_Blue_Att_Exec_Write_Resp 0x00002000: Evt_Blue_Gatt_Indication 0x00004000: Evt_Blue_Gatt_notification 0x00008000: Evt_Blue_Gatt_Error_Resp 0x00010000: Evt_Blue_Gatt_Procedure_Complete 0x00020000: Evt_Blue_Gatt_Disc_Read_Charac_By_UUID_Resp 0x00040000: Evt_Blue_Gatt_Tx_Pool_Available

表 156. Aci_Gatt_Set_Event_Mask 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x12: 无效 HCI 指令参数

生成事件：
在指令完成时生成指令完成事件。

4.6.11 Aci_Gatt_Exchange_Configuration

表 157. Aci_Gatt_Exchange_Configuration

指令名	参数	返回
Aci_Gatt_Exchange_Configuration (0xFD0B)	Connection_handle	状态

说明：
执行 ATT MTU 交换。

表 158. Aci_Gatt_Exchange_Configuration 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄

表 159. Aci_Gatt_Exchange_Configuration 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当 ATT MTU 交换流程完成时，生成 Evt_Blue_Att_Exchange_MTU_Resp 事件。此外，还生成流程完成事件以指示流程结束。

4.6.12 Aci_Att_Find_Information_Req

表 160. Aci_Att_Find_Information_Req

指令名	参数	返回
Aci_Att_Find_Information_Req (0xFD0C)	Connection_handle Start_handle End_handle	状态

说明：
提出查找信息请求。

表 161. Aci_Att_Find_Information_Req 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Start_handle	2 字节	服务器上待发现属性的范围的开始句柄
End_handle	2 字节	服务器上待发现属性的范围的结束句柄



表 162. Aci_Att_Find_Information_Req 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件:

在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Find_Information_Resp 事件提供流程响应。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。

4.6.13 Att_Find_By_Type_Value_Req

表 163. Att_Find_By_Type_Value_Req

指令名	参数	返回
Att_Find_By_Type_Value_Req (0xFD0D)	Connection_handle Start_handle End_handle UUID AttrValLen AttrVal	状态

说明:

提出按类型值查找请求。

表 164. Aci_Att_Find_Attr_By_Typ_And_Value_Req 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Start_handle	2 字节	服务器上待发现属性的范围的开始句柄
End_handle	2 字节	服务器上待发现属性的范围的结束句柄
UUID	2 字节	16 位 UUID

表 164. Aci_Att_Find_Attr_By_Typ_And_Value_Req 指令参数（续）

参数	大小	说明
AttrValLen	1 字节	后续属性值的长度。 注意：尽管规范允许的最大属性值长度为 512 个八位组，但由于传输层的限制（指令数据包最大长度为 255 个字节），值的长度仅限于 255 个字节。
AttrVal	0-N 字节	要匹配的属性值

表 165. Aci_Att_Find_Attr_By_Typ_And_Value_Req 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：

在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Find_By_Type_Value_Resp 事件提供流程响应。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。

4.6.14 Aci_Att_Read_By_Type_Req

表 166. Aci_Att_Read_By_Type_Req

指令名	参数	返回
Aci_Att_Read_By_Type_Req (0xFD0E)	Connection_handle Start_handle End_handle UUID_type UUID	状态

说明：

发送按类型读取请求。

表 167. Aci_Att_Read_By_Type_Req 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Start_handle	2 字节	服务器上待读取值的范围的开始句柄
End_handle	2 字节	服务器上待读取值的范围的结束句柄
UUID_type	1 字节	0x01: 16 位 UUID 0x02: 128 位 UUID
UUID	2 字节或 16 字节	UUID

表 168. Aci_Att_Read_By_Type_Req 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件:

在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Read_By_Type_Resp 事件提供流程响应。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。

4.6.15 Aci_Att_Read_By_Group_Type_Req

表 169. Aci_Att_Read_By_Group_Type_Req

指令名	参数	返回
Aci_Att_Read_By_Group_Type_Req (0xFD0F)	Connection_handle Start_handle End_handle UUID_type UUID	状态

说明:

发送按组类型读取请求。

表 170. Aci_Att_Read_By_Group_Type_Req 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Start_handle	2 字节	服务器上待读取值的范围的开始句柄
End_handle	2 字节	服务器上待读取值的范围的结束句柄
UUID_type	1 字节	0x01: 16 位 UUID 0x02: 128 位 UUID
UUID	2 字节或 16 字节	UUID

表 171. Aci_Att_Read_By_Group_Type_Req 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件:

在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Read_By_Group_Resp 事件提供流程响应。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。按组类型读取请求用于获取属性类型已知但句柄未知的分组属性的值。分组属性在 GATT 层定义。分组属性类型为: 主要服务、次要服务和特征。

4.6.16 Aci_Att_Prepate_Write_Req

表 172. Aci_Att_Prepate_Write_Req

指令名	参数	返回
Aci_Att_Prepate_Write_Req (0xFD10)	Connection_handle AttrHandle valOffset AttrValLen AttrVal	状态

说明：

发送准备写入请求。

表 173. Aci_Att_Prepate_Write_Req 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
AttrHandle	2 字节	必须写入值的属性的句柄
valOffset	2 字节	必须写入值的偏移
AttrValLen	1 字节	属性值长度（最大值为 ATT_MTU - 5）。
AttrVal	0-N 字节	要写入的属性值

表 174. Aci_Att_Prepate_Write_Req 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：

在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Prepate_Write_Resp 事件提供流程响应。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。

4.6.17 Aci_Att_Execute_Write_Req

表 175. Aci_Gatt_Execute_Write_Req

指令名	参数	返回
Aci_Att_Execute_Write_Req (0xFD11)	Connection_handle 执行	状态

说明：

发送执行写入请求。

表 176. Aci_Att_Execute_Write_Req 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Execute	1 字节	0x00 – 取消所有已准备写入 0x01 – 立即写入所有挂起的准备值。

表 177. Aci_Att_Execute_Write_Req 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - ATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：

在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Exec_Write_Resp 事件提供流程结果。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。

4.6.18 Aci_Gatt_Disc_All_Prim_Services

表 178. Aci_Gatt_Disc_All_Prim_Services

指令名	参数	返回
Aci_Gatt_Disc_All_Prim_Services (0xFD12)	Connection_handle	状态

说明：

该指令将启动 GATT 客户端流程以发现服务器上的所有主要服务。

表 179. Aci_Gatt_Disc_All_Prim_Services 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄

表 180. Aci_Gatt_Disc_All_Prim_Services 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件:

在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Read_By_Group_Resp 事件提供流程响应。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。

4.6.19 Aci_Gatt_Disc_Prim_Service_By_UUID

表 181. Aci_Gatt_Disc_Prim_Service_By_UUID

指令名	参数	返回
Aci_Gatt_Disc_Prim_Service_By_UUID (0xFD13)	Connection_handle UUID_type UUID	状态

说明:

该指令将启动流程以发现服务器上指定 UUID 的主要服务。

表 182. Aci_Gatt_Disc_Prim_Service_By_UUID 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
UUID_type	1 字节	0x01: 16 位 UUID (默认) 0x02: 128 位 UUID
UUID	2 字节或 16 字节	UUID

表 183. Aci_Gatt_Disc_Prim_Service_By_UUID 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Find_By_Type_Value_Resp 事件提供流程响应。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。

4.6.20 Aci_Gatt_Find_Included_Services

表 184. Aci_Gatt_Find_Included_Services

指令名	参数	返回
Aci_Gatt_Find_Included_Services (0xFD14)	Connection_handle Start_handle End_handle	状态

说明：
启动寻找所有已包含服务的流程。

表 185. Aci_Gatt_Find_Included_Services 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Start_handle	2 字节	服务的开始句柄
End_handle	2 字节	服务的结束句柄



表 186. Aci_Gatt_Find_Included_Services 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件:

在接收到指令时生成指令状态事件。通过 Evt_Blue_Att_Read_By_Type_Resp 事件提供流程响应。通过 Evt_Blue_Gatt_Procedure_Complete 事件指示流程结束。

4.6.21 Aci_Gatt_Disc_All_Charac_Of_Serv

表 187. Aci_Gatt_Disc_All_Charac_Of_Serv

指令名	参数	返回
Aci_Gatt_Disc_All_Charac_Of_Serv (0xFD15)	Connection_handle Start_Attr_handle End_Attr_handle	状态

说明:

启动发现给定服务的所有特征的流程。

表 188. Aci_Gatt_Disc_All_Charac_Of_Serv 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Start_Attr_handle	2 字节	服务的开始属性句柄
End_Attr_handle	2 字节	服务的结束属性句柄

表 189. Aci_Gatt_Disc_All_Charac_Of_Serv 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件:

在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Read_By_Type_Resp 事件提供响应数据包。

4.6.22 Aci_Gatt_Disc_Charac_By_UUID

表 190. Aci_Gatt_Disc_Charac_By_UUID

指令名	参数	返回
Aci_Gatt_Disc_Charac_By_UUID (0xFD16)	Connection_handle Start_Attr_handle End_Attr_handle UUID_type UUID	状态

说明:

启动发现 UUID 指定的所有特征的流程。

表 191. Aci_Gatt_Disc_Charac_By_UUID 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Start_Attr_handle	2 字节	服务的开始属性句柄
End_Attr_handle	2 字节	服务的结束属性句柄
UUID_type	1 字节	0x01: 16 位 UUID (默认) 0x02: 128 位 UUID
UUID	2 字节或 16 字节	UUID

表 192. Aci_Gatt_Disc_Charac_By_UUID 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件:

在接收到指令时生成指令状态事件。当流程完成时, 生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前, 通过 Evt_Blue_Gatt_Disc_Read_Charac_By_UUID_Resp 事件提供响应数据包。

4.6.23 Aci_Gatt_Disc_All_Charac_Descriptors

表 193. Aci_Gatt_Disc_All_Charac_Descriptors

指令名	参数	返回
Aci_Gatt_Disc_All_Charac_Descriptors (0xFD17)	Connection_handle charValHandle EndHandle	状态

说明:

启动发现服务器上所有特征描述符的流程。

表 194. Aci_Gatt_Disc_All_Charac_Descriptors 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
charValHandle	2 字节	特征值的句柄
EndHandle	2 字节	特征的结束句柄

表 195. Aci_Gatt_Disc_All_Charac_Descriptors 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件:

在接收到指令时生成指令状态事件。当流程完成时, 生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前, 通过 Evt_Blue_Att_Find_Info_Resp 事件提供响应数据包。

4.6.24 Aci_Gatt_Read_Charac_Val

表 196. Aci_Gatt_Read_Charac_Val

指令名	参数	返回
Aci_Gatt_Read_Charac_Val (0xFD18)	Connection_handle attrHandle	状态

说明:

启动读取属性值的流程。

表 197. Aci_Gatt_Read_Charac_Val 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
attrHandle	2 字节	要读取的特征的句柄

表 198. Aci_Gatt_Read_Charac_Val 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：

在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Read_Resp 事件提供响应数据包。

4.6.25 Aci_Gatt_Read_Charac_Using_UUID

表 199. Aci_Gatt_Read_Charac_Using_UUID

指令名	参数	返回
Aci_Gatt_Read_Charac_Using_UUID (0xFD19)	Connection_handle Start_handle End_handle UUID_type UUID	状态

说明：

启动读取 UUID 指定的所有特征的流程。

表 200. Aci_Gatt_Read_Charac_Using_UUID 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Start_handle	2 字节	要搜索的范围的开始句柄
End_handle	2 字节	要搜索的范围的结束句柄
UUID_type	1 字节	0x01: 16 位 UUID (默认) 0x02: 128 位 UUID
UUID	2 字节或 16 字节	UUID

表 201. Aci_Gatt_Read_Charac_Using_UUID 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Gatt_Disc_Read_Charac_By_UUID_Resp 事件提供响应数据包。

4.6.26 Aci_Gatt_Read_Long_Charac_Val

表 202. Aci_Gatt_Read_Long_Charac_Val

指令名	参数	返回
Aci_Gatt_Read_Long_Charac_Val (0xFD1A)	Connection_handle attrHandle valOffset	状态

说明：
启动读取长特征值的流程。

表 203. Aci_Gatt_Read_Long_Charac_Val 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
attrHandle	1 字节	要读取的特征的句柄
valOffset	2 字节	需要读取值的偏移

表 204. Aci_Gatt_Read_Long_Charac_Val 指令返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Read_Blob_Resp 事件提供响应数据包。



4.6.27 Aci_Gatt_Read_Multiple_Charac_Val

表 205. Aci_Gatt_Read_Multiple_Charac_Val

指令名	参数	返回
Aci_Gatt_Read_Multiple_Charac_Val (0xFD1B)	Connection_handle numHandles setOfHandles	状态

说明：

启动从服务器读取多个特征值的流程。

当客户端已知特征值句柄时，该子流程用于从服务器读取多个特征值。

表 206. Aci_Gatt_Read_Multiple_Charac_Val 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
numHandles	1 字节	必须读取值的句柄的数量
setOfHandles	numHandle * 2 字节	必须读取的属性值的句柄

表 207. Aci_Gatt_Read_Multiple_Charac_Val 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：

在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Read_Multiple_Resp 事件提供响应数据包。

4.6.28 Aci_Gatt_Write_Charac_Value

表 208. Aci_Gatt_Write_Charac_Value

指令名	参数	返回
Aci_Gatt_Write_Charac_Value (0xFD1C)	Connection_handle attrHandle ValLen AttrVal	状态

说明：
启动写入特征值的流程。

表 209. Aci_Gatt_Write_Charac_Value 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
attrHandle	2 字节	要写入的特征的句柄
ValLen	1 字节	要写入的值的长度
AttrVal	0-N 字节	要写入的值

表 210. Aci_Gatt_Write_Charac_Value 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。



4.6.29 Aci_Gatt_Write_Long_Charac_Val

表 211. Aci_Gatt_Write_Long_Charac_Val

指令名	参数	返回
Aci_Gatt_Write_Long_Charac_Val (0xFD1D)	Connection_handle AttrHandle ValOffset ValLen AttrVal	状态

说明：
启动写入长特征值的流程。

表 212. Aci_Gatt_Write_Long_Charac_Val 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
AttrHandle	2 字节	要写入的属性的句柄
ValOffset	2 字节	必须写入属性的偏移
ValLen	1 字节	要写入的值的长度
AttrVal	0-N 字节	要写入的值

表 213. Aci_Gatt_Write_Long_Charac_Val 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Prepate_Write_Resp 和 Evt_Blue_Att_Exec_Write_Resp 事件提供响应数据包。

4.6.30 Aci_Gatt_Write_Charac_Reliable

表 214. Aci_Gatt_Write_Charac_Reliable

指令名	参数	返回
Aci_Gatt_Write_Charac_Reliable (0xFD1E)	Connection_handle AttrHandle ValOffset ValLen AttrVal	状态

说明：
启动安全写入特征的流程。

表 215. Aci_Gatt_Write_Charac_Reliable 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
AttrHandle	2 字节	要写入的属性的句柄
ValOffset	2 字节	必须写入属性的偏移
ValLen	1 字节	要写入的值的长度
AttrVal	0-N 字节	要写入的值

表 216. Aci_Gatt_Write_Charac_Reliable 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Prepate_Write_Resp 和 Evt_Blue_Att_Exec_Write_Resp 事件提供响应数据包。



4.6.31 Aci_Gatt_Write_Long_Charac_Desc

表 217. Aci_Gatt_Write_Long_Charac_Desc

指令名	参数	返回
Aci_Gatt_Write_Long_Charac_Desc (0xFD1F)	Connection_handle attrHandle ValOffset ValLen AttrVal	状态

说明：
启动写入长特征描述符的流程。

表 218. Aci_Gatt_Write_Long_Charac_Desc 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
attrHandle	2 字节	要写入的属性的句柄
ValOffset	2 字节	必须写入属性的偏移
ValLen	1 字节	要写入的值的长度
AttrVal	0-N 字节	要写入的值

表 219. Aci_Gatt_Write_Long_Charac_Desc 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Prepate_Write_Resp 和 Evt_Blue_Att_Exec_Write_Resp 事件提供响应数据包。

4.6.32 Aci_Gatt_Read_Long_Charac_Desc

表 220. Aci_Gatt_Read_Long_Charac_Desc

指令名	参数	返回
Aci_Gatt_Read_Long_Charac_Desc (0xFD20)	Connection_handle attrHandle valOffset	状态

说明：
启动读取长特征值的流程。

表 221. Aci_Gatt_Read_Long_Charac_Desc 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
attrHandle	2 字节	特征描述符的句柄
ValOffset	2 字节	需要读取值的偏移

表 222. Aci_Gatt_Read_Long_Charac_Desc 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Read_Blob_Resp 事件提供响应数据包。



4.6.33 Aci_Gatt_Write_Charac_Descriptor

表 223. Aci_Gatt_Write_Charac_Descriptor

指令名	参数	返回
Aci_Gatt_Write_Charac_Descriptor (0xFD21)	Connection_handle AttrHandle ValLen AttrVal	状态

说明：
启动写入特征值的流程。

表 224. Aci_Gatt_Write_Charac_Descriptor 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
AttrHandle	2 字节	要写入的属性的句柄
ValLen	1 字节	要写入的值的长度
AttrVal	0-N 字节	要写入的值

表 225. Aci_Gatt_Write_Charac_Descriptor 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmpl 事件。

4.6.34 **Aci_Gatt_Read_Charac_Desc**

表 226. Aci_Gatt_Read_Charac_Desc

指令名	参数	返回
Aci_Gatt_Read_Charac_Desc (0xFD22)	Connection_handle AttrHandle	状态

说明：
启动读取指定描述符的流程。

表 227. Aci_Gatt_Read_Charac_Desc 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
AttrHandle	2 字节	要读取的描述符的句柄

表 228. Aci_Gatt_Read_Charac_Desc 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 交换是否已经发生 - GATT 是否正在等待对之前请求的响应 - 请求已在队列中等待发送 - 通道未打开 - 已启动了一个 GATT 流程

生成事件：
在接收到指令时生成指令状态事件。当流程完成时，生成 Evt_Blue_Gatt_Procedure_Cmplt 事件。在流程完成前，通过 Evt_Blue_Att_Read_Resp 事件提供响应数据包。

4.6.35 **Aci_Gatt_Write_Without_Response**

表 229. Aci_Gatt_Write_Without_Response

指令名	参数	返回
Aci_Gatt_Write_Without_Response (0xFD23)	Connection_handle AttrHandle ValLen AttrVal	状态

说明：
启动写入特征值的流程，不等待服务器的任何响应。

表 230. Aci_Gatt_Write_Without_Response 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
AttrHandle	2 字节	要写入的属性的句柄
ValLen	1 字节	要写入的值的长度
AttrVal	0-N 字节	要写入的值

表 231. Aci_Gatt_Write_Without_Response 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x42：无效参数 0x46：不允许 - 通道未打开 - 之前的 Aci_Gatt_Write_Without_Response 仍在进行（仅限于 BlueNRG-MS FW 栈 7.1、7.1.a） 0x64：资源不足（BlueNRG-MS FW 栈 7.1.b 或更高版本） 才有 0x64 返回值

生成事件：
在该指令处理完毕时生成指令完成事件。

4.6.36 Aci_Gatt_Signed_Write_Without_Resp

表 232. Aci_Gatt_Signed_Write_Without_Resp

指令名	参数	返回
Aci_Gatt_Signed_Write_Without_Resp (0xFD24)	Connection_handle AttrHandle ValLen AttrVal	状态

说明：
启动使用验证签名写入特征值的流程，不等待服务器的任何响应。在链路已加密时不能使用。

表 233. Aci_Gatt_Signed_Write_Without_Resp 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
AttrHandle	2 字节	要写入的属性的句柄
ValLen	1 字节	要写入的值的长度
AttrVal	0-N 字节	要写入的值

表 234. Aci_Gatt_Signed_Write_Without_Resp 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x42: 无效参数 0x46: 不允许 - 通道未打开 - 之前的 Aci_Gatt_Write_Without_Response 仍在进行（仅限于 BlueNRG-MS FW 栈 7.1、7.1.a） 0x64: 资源不足（BlueNRG-MS FW 栈 7.1.b 或更高版本）

生成事件：
在该指令处理完毕时生成指令完成事件。

4.6.37 Aci_Gatt_Confirm_Indication

表 235. Aci_Gatt_Confirm_Indication

指令名	参数	返回
Aci_Gatt_Confirm_Indication (0xFD25)	Connection_handle	状态

说明：
允许应用确认指示。当应用接收到事件 Evt_Blue_Gatt_Indication 时，必须发送该指令。

表 236. Aci_Gatt_Confirm_Indication 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄



表 237. Aci_Gatt_Confirm_Indication 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x46: 不允许 - 交换是否已经发生 - 通道未打开

生成事件：
在该指令处理完毕时生成指令完成事件。

4.6.38 Aci_Gatt_Write_Response

表 238. Aci_Gatt_Write_Response

指令名	参数	返回
Aci_Gatt_Write_Response (0xFD26)	Connection_handle Attribute_handle Write_status Error_Code Att_Val_Len Att_Val	状态

说明：
该指令允许或拒绝来自客户端的写入请求。当应用接收到 Evt_Blue_Gatt_Write_Permit_req 时，必须发送该指令。如果可以允许写入，则必须将状态和错误代码设置为 0。如果不能允许写入，则必须将状态设置为 1，并将错误代码设置为必须传递至客户端的错误代码。

表 239. Aci_Gatt_Write_Response 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄
Attribute_handle	2 字节	事件 Evt_Blue_Gatt_Write_Permit_req 中传递的属性的句柄
Write_status	1 字节	0x00: 值可以写入句柄指定的属性 0x01: 值不应写入句柄指定的属性
Error_Code	1 字节	必须拒绝写入时必须传递至客户端的错误代码
Att_Val_Len	1 字节	事件 Evt_Blue_Gatt_Write_Permit_req 中传递的要写入的值的长度
Att_Val	0-N 字节	事件 Evt_Blue_Gatt_Write_Permit_req 中传递的值

表 240. Aci_Gatt_Write_Response 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x46: 不允许 指令在栈的当前状态下不被允许

生成事件：
在该指令处理完毕时生成指令完成事件。

4.6.39 Aci_Gatt-Allow_Read

表 241. Aci_Gatt-Allow_Read

指令名	参数	返回
Aci_Gatt-Allow_Read (0xFD27)	Connection_handle	状态

说明：
该指令允许 GATT 服务器对来自客户端的读取请求发送响应。
当应用接收到 Evt_Blue_Gatt_Read_Permit_Req 或 Evt_Blue_Gatt_Read_Multi_Permit_Req 时，必须发送该指令。该指令向栈表明，可以向客户端发送响应。因此，如果应用想要在客户端读取属性前更新任意属性，必须使用 Aci_Gatt_Update_Charac_Value 更新特征值，然后给出该指令。应用应在 30 秒内执行所需操作。否则，GATT 流程将超时。

表 242. Aci_Gatt-Allow_Read 指令参数

参数	大小	说明
Connection_handle	2 字节	发出的指令所针对的连接句柄



表 243. Aci_Gatt_Allow_Read 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x46: 不允许 指令在栈的当前状态下不被允许

生成事件：
在该指令处理完毕时生成指令完成事件。

4.6.40 Aci_Gatt_Set_Security_Permission

表 244. Aci_Gatt_Set_Security_Permission

指令名	参数	返回
Aci_Gatt_Set_Security_Permission (0xFD28)	Service_handle Attr_handle Security_permission	状态

说明：
该指令为指定的属性句柄设置安全许可。目前，仅允许为客户端配置描述符设置安全许可。

表 245. Aci_Gatt_Set_Security_Permission 指令参数

参数	大小	说明
Service_handle	2 字节	包含必须修改安全许可的属性的服务句柄。
Attr_handle	2 字节	必须修改安全许可的属性的句柄。
Security_permission	1 字节	0x00: ATTR_PERMISSION_NONE 0x01: ATTR_PERMISSION_AUTHEN_READ 0x02: ATTR_PERMISSION_AUTHOR_READ 0x04: ATTR_PERMISSION_ENCRY_READ 0x08: ATTR_PERMISSION_AUTHEN_WRITE 0x10: ATTR_PERMISSION_AUTHOR_WRITE 0x20: ATTR_PERMISSION_ENCRY_WRITE

表 246. Aci_Gatt_Set_Security_Permission 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x12: 无效参数 0x41: 失败 无法找到指定句柄或句柄不是客户端配置描述符的句柄

生成事件：
在该指令处理完毕时生成指令完成事件。

4.6.41 Aci_Gatt_Set_Desc_Value

表 247. Aci_Gatt_Set_Desc_Value

指令名	参数	返回
Aci_Gatt_Set_Desc_Value (0xFD29)	Service_Handle Char_Handle Desc_Handle val_offset desc_val_len desc_val	状态

说明：
该指令设置通过 Desc_handle 指定的描述符的值。

表 248. Aci_Gatt_Set_Desc_Value 指令参数

参数	大小	说明
Service_handle	2 字节	包含描述符的服务的句柄。
Char handle	2 字节	包含描述符的特征的句柄。
Desc handle	2 字节	必须设置值的描述符的句柄。
val_offset	2 字节	必须更新描述符值的偏移。
desc_val_len	2 字节	描述符值的长度
desc_val	0-N 字节	描述符值



表 249. Aci_Gatt_Set_Desc_Value 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误 0x60: 无效句柄 0x61: 无效参数

生成事件：
在该指令处理完毕时生成指令完成事件。

4.6.42 Aci_Gatt_Read_Handle_Value

表 250. Aci_Gatt_Read_Handle_Value

指令名	参数	返回
Aci_Gatt_Read_Handle_Value (0xFD2A)	Attribute_handle	状态 长度 值

说明：
从本地 GATT 数据库读取指定属性句柄的值。

表 251. Aci_Gatt_Read_Handle_Value 指令参数

参数	大小	说明
Attribute handle	2 字节	要读取的属性的句柄

表 252. Aci_Gatt_Read_Handle_Value 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x41: BLE_STATUS_FAILED 0x12: ERR_INVALID_HCI_CMD_PARAMS
Length	2 字节	值的长度
Value	0-N 字节	值读取。长度字段中指定的长度。

生成事件：
在接收到该指令时返回指令完成事件。事件将包含指定属性句柄的长度和值。

4.6.43 **Aci_Gatt_Read_Handle_Value_Offset**

表 253. Aci_Gatt_Read_Handle_Value_Offset

指令名	参数	返回
Aci_Gatt_Read_Handle_Value_Offset (0xFD2B)	Attribute_Handle 偏移	状态 Attribute_value_Length Attribute_value

说明：

指令返回来自指定偏移的属性句柄的值。如果要返回的长度大于 128，则仅返回 128 个字节。应用应以递增的偏移发送该指令，直至获得具有指定偏移的错误或返回的属性值的字节数小于 128。

表 254. Aci_Gatt_Read_Handle_Value_Offset 指令参数

参数	大小	说明
Attribute_handle	2 字节	要读取的属性句柄
Offset	1 字节	偏移指定了需要读取值的位置。

表 255. Aci_Gatt_Read_Handle_Value_Offset 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x63: 状态无效操作 0x12: HCI_Invalid 参数 0x41: BLE_Status_Failed
Attribute_value_Length	2 字节	后续属性值的长度
Attribute_value	数据字节长度	读取属性值

生成事件：

在指令完成时，生成指令完成事件。状态指示成功或失败。如果状态为成功，则事件数据包包含属性值的数据长度和紧接其后的属性值。

4.6.44 **Aci_Update_Char_Value_Ext**

表 256. Aci_Gatt_Update_Char_Value_Ext

指令名	参数	返回
Aci_Gatt_Update_Char_Value_Ext (0xFD2C)	Serv_Handle Char_Handle Update_Type Val_Total_Length Val_Update_Offset Val_Update_Length Char_Update_Value	状态

说明：

更新属于指定服务的特征的属性值。
该指令是 Aci_Gatt_Update_Char_Value 指令（操作码 =0xFD06）的扩展，支持长属性值（最长 512 字节）的更新。
此外，还提供特定参数，用于选择是否可以发送指示和 / 或通知（如果已在相关客户端特征配置描述符中启用）。

表 257. Aci_Gatt_Update_Char_Value_Ext 指令参数

参数	大小	说明
Serv_Handle	2 字节	特征所属服务的句柄。
Char_Handle	2 字节	特征的句柄。
Update_Type	1 字节	控制通过属性值更新生成的通知 / 指示： 0x00：既不能发送通知也不能发送指示（没有与对端客户端共享更新） 0x01：只能发送通知 0x02：只能发送指示 0x03：能发送通知和指示。
Val_Total_Length	2 字节	更新后属性值的总长度。 如果特征值长度可变，该字段指定更新后特征值的新长度；如果特征值长度固定，则忽略此字段。
Val_Update_Offset	2 字节	必须更新属性值的偏移。
Val_Update_Length	1 字节	后续 Char_Update_Value 缓冲区的长度。
Char_Update_Value	Val_Update_Length 字节	特征的更新值。

表 258. Aci_Gatt_Update_Char_Value_Ext 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功 0x47: 错误, GATT 没有初始化 0x60: 无效句柄 0x61: 无效参数 0x64: 资源不足



4.7 GATT VS 事件

表 259. GATT VS 事件

项目	事件	EGID	EID	ECODE
1	Evt_Blue_Gatt_Attribute_modified	0x03	0x01	0x0C01
2	Evt_Blue_Gatt_Procedure_Timeout	0x03	0x02	0x0C02
3	Evt_Blue_Att_Exchange_MTU_Resp	0x03	0x03	0x0C03
4	Evt_Blue_Att_Find_Information_Resp	0x03	0x04	0x0C04
5	Evt_Blue_Att_Find_By_Type_Value_Resp	0x03	0x05	0x0C05
6	Evt_Blue_Att_Read_By_Type_Resp	0x03	0x06	0x0C06
7	Evt_Blue_Att_Read_Resp	0x03	0x07	0x0C07
8	Evt_Blue_Att_Read_Blob_Resp	0x03	0x08	0x0C08
9	Evt_Blue_Att_Read_Multiple_Resp	0x03	0x09	0x0C09
10	Evt_Blue_Att_Read_By_Group_Type_Resp	0x03	0x0A	0x0C0A
11	Evt_Blue_Att_Prepare_Write_Resp	0x03	0x0C	0x0C0C
12	Evt_Blue_Att_Exec_Write_Resp	0x03	0x0D	0x0C0D
13	Evt_Blue_Gatt_Indication	0x03	0x0E	0x0C0E
14	Evt_Blue_Gatt_notification	0x03	0x0F	0x0C0F
15	Evt_Blue_Gatt_Procedure_Complete	0x03	0x10	0x0C10
16	Evt_Blue_Gatt_Error_Resp	0x03	0x11	0x0C11
17	Evt_Blue_Gatt_Disc_Read_Charac_By_UUID_Resp	0x03	0x12	0x0C12
18	Evt_Blue_Gatt_Write_Permit_req	0x03	0x13	0x0C13
19	Evt_Blue_Gatt_Read_Permit_Req	0x03	0x14	0x0C14
20	Evt_Blue_Gatt_Read_Multi_Permit_Req	0x03	0x15	0x0C15
21	Evt_Blue_Gatt_Tx_Pool_Available	0x03	0x16	0x0C16
22	Evt_Blue_Gatt_Server_Confirmation	0x03	0x16	0x0C17

4.7.1 Evt_Blue_Gatt_Attribute_modified

作为下列 GATT 流程的结果之一，GATT 服务器在客户端修改服务器上的任何属性时向应用生成该事件：

- 写入，无响应
- 签名写入，无响应
- 写入特征值
- 写入长特征值
- 安全写入

表 260. Evt_Blue_Gatt_Attribute_modified

参数	大小	说明
Event code	2 字节	厂商特定事件的事件代码
Connection_handle	2 字节	修改了属性的连接句柄
Attribute_handle	2 字节	被修改的属性的句柄
data_length	1 字节	Attribute_value 字段的长度。
Datalen	2 字节	第 0-30 位：属性中已报告值的偏移。 第 31 位：如果一个 Evt_Blue_Gatt_Attribute_modified 事件不能容纳属性的全部值，将该位设置为 1 以通知将由后续的其他 Evt_Blue_Gatt_Attribute_modified 事件报告剩余值。
Attribute_value	0-N 字节	新属性值，从给定偏移开始。

4.7.2 Evt_Blue_Gatt_Procedure_Timeout

由客户端 / 服务器在 GATT 超时（30 秒）时向应用生成该事件。

表 261. Evt_Blue_Gatt_Procedure_Timeout

参数	大小	说明
Event code	2 字节	厂商特定事件的事件代码
Connection_handle	2 字节	GATT 流程超时的连接句柄
data_len	1 字节	后续数据（总是为 0）

4.7.3 Evt_Blue_Gatt_Procedure_Complete

当 GATT 客户端流程完成但出错或成功完成时，生成该事件。

表 262. Evt_Blue_Gatt_Procedure_Complete

参数	大小	说明
Event code	2 字节	厂商特定事件的事件代码
Connection_handle	2 字节	GATT 流程已完成的连接句柄
data_len	1 字节	总是为 1 字节。
error_code	0-N 字节	指示流程完成但出错（BLE_STATUS_FAILED）或成功完成（BLE_STATUS_SUCCESS）。

4.7.4 Evt_Blue_Gatt_Disc_Read_Charac_By_UUID_Resp

在“按 UUID 发现特征”流程或“使用特征 UUID 读取”流程期间，可能生成该事件。

按照蓝牙核心规范 v4.1（第 3 卷 G 部分第 3.3.1 节）的定义，如果启用了“按 UUID 发现特征”，属性值将为服务声明。如果执行了“使用特征 UUID 读取”，将为特征的值。

表 263. Evt_Blue_Gatt_Disc_Read_Charac_By_UUID_Resp

参数	大小	说明
Event code	2 字节	厂商特定事件的事件代码
Connection_Handle	2 字节	流程启动的连接句柄。
Data_len	1 字节	下列数据的长度
Attribute_Handle	2 字节	已发现属性的句柄
Attribute_Value	(Data_len - 2) 字节	属性值

4.7.5 Evt_Blue_Gatt_Write_Permit_req

当服务器接收到来自客户端的写入请求、写入指令或签名写入指令时，向应用给出该事件。仅当在添加特征时为该事件的生成设置了事件位时，才向应用给出该事件。

在接收到该事件时，应用必须检查正在请求写入的值是否允许写入，并以指令 Aci_Gatt_Write_Response 进行响应。可以找到指令参数的详细信息。栈将根据来自应用的响应修改属性值。如果应用拒绝写入，将不修改属性值。对于写入请求，将向客户端发送错误响应，错误代码由应用指定。对于写入 / 签名写入指令，不向客户端发送响应，但不修改属性。

表 264. Evt_Blue_Gatt_Write_Permit_req

参数	大小	说明
Event code	2 字节	厂商特定事件的事件代码
Connection_handle	2 字节	有写入属性请求的连接的句柄
Attribute_handle	2 字节	客户端发出了写入请求的属性的句柄
data_len	1 字节	后续数据的长度
data_buffer	0-N 字节	客户端请求写入的数据

4.7.6 Evt_Blue_Gatt_Read_Permit_Req

当服务器接收到来自客户端的读取请求或读取 BLOB 请求时，向应用给出该事件。仅当在添加特征时为该事件的生成设置了事件位时，才向应用给出该事件。

在接收该事件时，应用可以视需要更新句柄的值，完成后，它必须发送 Aci_Gatt_Allow_Read 指令以指示栈它可以向客户端发送响应。

表 265. Evt_Blue_Gatt_Read_Permit_Req

参数	大小	说明
Event code	2 字节	厂商特定事件的事件代码
Connection_handle	2 字节	有读取属性请求的连接的句柄
Attribute_handle	2 字节	客户端请求读取的属性的句柄

表 265. Evt_Blue_Gatt_Read_Permit_Req (续)

参数	大小	说明
Datalen	1 字节	后续数据的长度
Value	0-N 字节	包含已请求读取的偏移

4.7.7 Evt_Blue_Gatt_Read_Multi_Permit_Req

当服务器接收到来自客户端的读取多个请求或按类型读取请求时，向应用给出该事件。仅当在添加特征时为该事件的生成设置了事件位时，才向应用给出该事件。

在接收该事件时，应用可以视需要更新句柄的值，完成后，它必须发送 Aci_Gatt_Allow_Read 指令以指示栈它可以向客户端发送响应。

表 266. Evt_Blue_Gatt_Read_Multi_Permit_Req

参数	大小	说明
Event code	2 字节	厂商特定事件的事件代码
Connection_handle	2 字节	请求读取属性的连接的句柄
data_len	1 字节	后续数据的长度
data_buffer	0-N 字节	客户端已请求读取的属性的句柄

4.7.8 Evt_Blue_Gatt_Tx_Pool_Available

每次 BlueNRG-MS FW 栈生成错误代码 BLE_STATUS_INSUFFICIENT_RESOURCES (0x64) 时，只要有至少两个缓冲区可用于通知或写入指令，即生成 Evt_Blue_Gatt_Tx_Pool_Available 事件。

表 267. Evt_Blue_Gatt_Tx_Pool_Available

参数	大小	说明
Event code	2 字节	该厂商特定事件的事件代码 (0x0C16)
Connection_handle	2 字节	GATT 流程正在运行的连接句柄
Available_Buffers	2 字节	指示 attrTxPool 列表中可用元素的数量。

注：从 BlueNRG-MS FW 栈版本 7.1.b 开始支持该事件。



4.7.9 Evt_Blue_Att_Exchange_MTU_Resp

在响应交换 MTU 请求时生成该事件。请参见 [Aci_Gatt_Exchange_Configuration](#)。

表 268. Evt_Blue_Att_Exchange_MTU_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	后续数据的长度（总是为 1）。
server_rx_mtu	2 字节	属性服务器接收的 MTU 长度

4.7.10 Evt_Blue_Att_Find_Information_Resp

在响应 *查找信息请求* 时生成该事件。参见 [Aci_Att_Find_Information_Req](#) 和蓝牙核心规范 v4.0 中的“Find Information Response”。

表 269. Evt_Blue_Att_Find_Information_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	下列数据的长度
format	1 字节	handle_uuid_pair 的格式。 16 位 UUID 128 位 UUID
handle_uuid_pair	0-N 字节	handle-uuid 对的顺序。 如果格式 =1，每对为： [2 个八位组的句柄，2 个八位组的 UUID] 如果格式 =2，每对为： [2 个八位组的句柄，16 个八位组的 UUID]

4.7.11 Evt_Blue_Att_Find_By_Type_Value_Resp

在响应按 *类型 值 查找 请求* 时生成该事件。

表 270. Evt_Blue_Att_Find_By_Type_Value_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	下列数据的长度
handles_info_list	0-N 字节	句柄信息列表如蓝牙核心规范 v4.1 所定义。句柄对的顺序： [2 个八位组的找到属性句柄，2 个八位组的结束组句柄]

4.7.12 Evt_Blue_Att_Read_By_Type_Resp

在响应按类型读取请求时生成该事件。见 [Aci_Gatt_Find_Included_Services](#) 和 [Aci_Gatt_Disc_All_Charac_Of_Serv](#)。

表 271. Evt_Blue_Att_Read_By_Type_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	下列数据的长度
handle_value_pair_length	1 字节	每个属性 handle-value 对的长度
handle_value_pair	0-N 字节	属性数据列表如蓝牙核心规范 v4.1 所定义。handle-value 对的顺序： [2 个八位组的属性句柄, (handle_value_pair_length - 2 个八位组) 的属性值]

4.7.13 Evt_Blue_Att_Read_Resp

在响应读取请求时生成该事件。请参见 [Aci_Gatt_Read_Charac_Val](#)。

表 272. Evt_Blue_Att_Read_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	下列数据的长度
attribute_value	0-N 字节	属性的值。

4.7.14 Evt_Blue_Att_Read_Blob_Resp

在响应读取请求时生成该事件。请参见 [Aci_Gatt_Read_Charac_Val](#)。

表 273. Evt_Blue_Att_Read_Blob_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	下列数据的长度
part_attribute_value	0-N 字节	属性值的组成部分。

4.7.15 Evt_Blue_Att_Read_Multiple_Resp

在响应读取多个请求时生成该事件。

表 274. Evt_Blue_Att_Read_Multiple_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	下列数据的长度
set_of_values	0-N 字节	一组两个或更多值。

4.7.16 Evt_Blue_Att_Read_By_Group_Type_Resp

在响应按组类型读取请求时生成该事件。请参见 [Aci_Gatt_Disc_All_Prim_Services](#)。

表 275. Evt_Blue_Att_Read_By_Group_Type_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	下列数据的长度
attribute_data_length	1 字节	每个属性数据的长度。
attribute_data_list	0-N 字节	属性数据列表，其中的属性数据包含： 2 个八位组的属性句柄 2 个八位组的结束组句柄 (attribute_data_length - 4) 八位组的属性值

4.7.17 Evt_Blue_Att_Prepare_Write_Resp

在响应准备写入请求时生成该事件。

表 276. Evt_Blue_Att_Prepare_Write_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	下列数据的长度
attribute_handle	2 字节	要写入的属性的句柄。
Offset	1 字节	要写入的第一个八位组的偏移。
part_attr_value	0-N 字节	要写入的属性的值。

4.7.18 Evt_Blue_Att_Exec_Write_Resp

在响应执行写入请求时生成该事件。

表 277. Evt_Blue_Att_Exec_Write_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与响应相关的连接句柄
event_data_length	1 字节	总是为 0。

4.7.19 Evt_Blue_Gatt_Indication

在接收到来自服务器的指示时生成该事件。

表 278. Evt_Blue_Gatt_Indication

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与事件相关的连接句柄
event_data_length	1 字节	下列数据的长度
attribute_handle	2 字节	属性的句柄
attr_value	0-N 字节	属性的当前值

4.7.20 Evt_Blue_Gatt_notification

在接收到来自服务器的通知时生成该事件。

表 279. Evt_Blue_Gatt_notification

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	与事件相关的连接句柄
event_data_length	1 字节	下列数据的长度
attribute_handle	2 字节	属性的句柄
attr_value	0-N 字节	属性的当前值

4.7.21 Evt_Blue_Gatt_Error_Resp

在接收到来自服务器的错误响应时生成该事件。当一个 GATT 发现流程结束时，服务器可能给出错误响应。这并不意味着流程结束并出错，而是该错误事件是流程本身的一部分。

表 280. Evt_Blue_Gatt_Error_Resp

参数	大小	说明
event_code	2 字节	厂商特定事件的事件代码
connection_handle	2 字节	GATT 流程正在运行的连接句柄
data_len	1 字节	后续数据的长度
req_opcode	1 字节	生成该错误响应的请求
attr_handle	2 字节	生成该错误响应的属性句柄
error_code	1 字节	请求生成错误响应的原因

4.7.22 Evt_Gatt_Server_Confirmation

表 281. Evt_Gatt_Server_Confirmation

参数	大小	说明
Connection_handle	2 字节	有指示的连接的句柄

4.8 HCI 厂商特定指令

表 282. HCI VS 指令

项目	事件	CGID	CID	操作码
1	Aci_Hal_Write_Config_Data	0x00	0x0C	0xFC0C
2	Aci_Hal_Read_Config_Data	0x00	0x0D	0xFC0D
3	Aci_Hal_Set_Tx_Power_Level	0x00	0x0F	0xFC0F
4	Aci_Hal_Device_Standby	0x00	0x13	0xFC13
5	Aci_Hal_LE_Tx_Test_Packet_Number	0x00	0x14	0xFC14
6	Aci_Hal_Tone_Start	0x00	0x15	0xFC15
7	Aci_Hal_Tone_Stop	0x00	0x16	0xFC16
8	Aci_Hal_Get_Link_Status	0x00	0x17	0xFC17
9	Aci_Hal_Get_Fw_Build_Number	0x00	0x00	0xFC00
10	Aci_Hal_Get_Anchor_Period	0x00	0xF8	0xFCF8
11	Aci_Updater_Start	0x00	0x20	0xFC20
12	Aci_Updater_Reboot	0x00	0x21	0xFC21
13	Aci_Get_Updater_Version	0x00	0x22	0xFC22
14	Aci_Get_Updater_Buffer_Size	0x00	0x23	0xFC23
15	Aci_Erase_Blue_Flag	0x00	0x24	0xFC24
16	Aci_Reset_Blue_Flag	0x00	0x25	0xFC25
17	Aci_Updater_Erase_Sector	0x00	0x26	0xFC26
18	Aci_Updater_Program_Data_Block	0x00	0x27	0xFC27
19	Aci_Updater_Read_Data_Block	0x00	0x28	0xFC28
20	Aci_Updater_Calc_CRC	0x00	0x29	0xFC29
21	Aci_Updater_HW_Version	0x00	0x2A	0xFC2A

1. 对于更新器指令，即操作码为 0xFC2x 的指令，请参考独立文档 BlueNRGMS Updater Specification。

4.8.1 Aci_Hal_Write_Config_Data

表 283. Aci_Hal_Write_Config_Data

指令名	参数	返回
Aci_Hal_Write_Config_Data (0xFC0C)	偏移 长度 值	状态

说明：

该指令将值写入底层配置数据结构。它用于在运行时为系统直接设置一些底层参数。

表 284. Aci_Hal_Write_Config_Data 指令参数

参数	大小	说明
Offset	1 字节	数据结构中的偏移。数据结构中起始成员的偏移为 0。
Length	1 字节	要写入的数据的长度
Value	0-N 字节	要写入的数据

表 285. Aci_Hal_Write_Config_Data 成员

数据成员	大小	偏移	说明
Public address	6 字节	0x00	蓝牙公共地址
DIV	2 字节	0x06	用于得出 CSRK 的 DIV
ER	16 字节	0x08	用于得出 LTK 和 CSRK 的加密根密钥
IR	16 字节	0x18	用于得出 LTK 和 CSRK 的身份根密钥
LLWithoutHost	1 字节	0x2C	打开 / 关闭仅链路层模式
Role	1 字节	0x2D	选择 BlueNRG-MS 角色和模式配置。 1. 主设备和从设备 仅一个连接 6 KB RAM 内容保留 2. 主设备和从设备 仅一个连接 12 KB RAM 内容保留 3. 主设备和从设备 最多 8 个连接 12 KB RAM 内容保留 4. 主设备和从设备 同时广告和扫描 最多 4 个连接 从 BlueNRG-MS FW 栈版本 7.1.b 开始支持该模式。

表 286. Aci_Hal_Write_Config_Data 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: ERR_INVALID_HCI_CMD_PARAMS

生成事件：

控制器将生成指令完成事件。

4.8.2 Aci_Hal_Read_Config_Data

表 287. Aci_Hal_Read_Config_Data

指令名	参数	返回
Aci_Hal_Read_Config_Data (0xFC0D)	偏移	状态 值

说明：
该指令请求底层配置数据结构中的值。欲了解更多信息，请参见指令 Aci_Hal_Write_Config_Data。不同偏移的返回值字节数各不相同。

表 288. Aci_Hal_Read_Config_Data 指令参数

参数	大小	说明
Offset	1 字节	数据结构中的偏移。数据结构中起始成员的偏移为 0。

表 289. Aci_Hal_Read_Config_Data 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: ERR_INVALID_HCI_CMD_PARAMS
Value	0-N 字节	指定偏移处读取的值

生成事件：
控制器将生成指令完成事件。

4.8.3 Aci_Hal_Set_Tx_Power_Level

表 290. Aci_Hal_Set_Tx_Power_Level

指令名	参数	返回
Aci_Hal_Set_Tx_Power_Level (0xFC0F)	EN_HIGH_POWER PA_LEVEL	状态

说明：
该指令通过控制 EN_HIGH_POWER 和 PA_LEVEL（二者共同决定了输出功率水平（dBm））设置 BlueNRG-MS 的发送功率水平。参见下表。
当系统启动或重启时，将使用默认的发送功率水平，其最大值为 8 dBm。一旦发出该指令，将立即更改输出功率，无论是否正在进行蓝牙通信。例如，为了进行调试，可将 BlueNRG-MS 设置为一直广告，并使用该指令监听信号强度变化。
系统将保留从指令接收的最新发送功率水平，即第二个指令覆盖之前的发送功率水平。在收到另一条设置发送功率的指令或系统重启前，将维持新的发送功率水平。



表 291. Aci_Hal_Set_Tx_Power_Level 指令参数

参数	大小	说明
EN_HIGH_POWER	1 字节	只能是 0 或 1。将高功率位设置为 ON 或 OFF。
PA_LEVEL	1 字节	可以从 0 到 7。设置 PA 水平值。

表 292. Tx_power_level 指令参数组合

EN_HIGH_POWER	PA_LEVEL	发送功率水平 (dBm)
0	0	-18
0	1	-14.7
0	2	-11.4
0	3	-8.1
0	4	-4.9
0	5	-1.6
0	6	1.7
0	7	5.0
1	0	-15
1	1	-11.7
1	2	-8.4
1	3	-5.1
1	4	-2.1
1	5	1.4
1	6	4.7
1	7	8.0 (默认)

表 293. Aci_Hal_Set_Tx_Power_Level 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x12: ERR_INVALID_HCI_CMD_PARAMS

生成事件:

控制器将生成指令完成事件。

4.8.4 Aci_Hal_Device_Standby

表 294. Aci_Hal_Device_Standby

指令名	参数	返回
Aci_Hal_Device_Standby(0xFC13)		状态

说明：

BlueNRG-MS 通常会自动进入睡眠模式以节约电力。该 Aci_Hal_Device_Standby 指令进一步使设备进入待机模式而不是睡眠模式。二者的区别在于，在睡眠模式下，系统仍可通过内部定时器唤醒自己。而在待机模式下，该定时器也被禁用。因此，只能通过外部信号（例如通过 SPI 总线发送的 HCI 指令）唤醒设备。

根据测量，睡眠模式下的电流消耗为 ~2 uA，而待机模式下为 ~1.5 uA。

仅当没有任何其他蓝牙活动时，才会接受该指令。否则，会返回错误代码“指令不被允许”。

表 295. Aci_Hal_Device_Standby 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS 0x0C: ERR_COMMAND_DISALLOWED 当蓝牙活动正在进行时，该指令不被允许。

生成事件：

控制器将生成指令完成事件。

4.8.5 Aci_Hal_LE_Tx_Test_Packet_Number

表 296. Aci_Hal_LE_Tx_Test_Packet_Number

指令名	参数	返回
Aci_Hal_LE_Tx_Test_Packet_Number (0xFC14)		状态 数据包计数器

说明：

在直接测试模式下，在执行直接测试结束指令时，不返回发送测试中发送的数据包数量。此命令实现此特性。

如果启动了直接发送测试，将使用 32 位计数器对已发送数据包的数量进行计数。在直接测试结束后，可以使用该指令检查直接发送测试期间发送的数据包数量。

计数器从 0 开始向上计数。如果全部 32 位均已使用，计数器将复位并重新从 0 开始计数。在下一次直接发送测试开始前，计数器不会清零。

表 297. Aci_Hal_LE_Tx_Test_Packet_Number 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS
Packet counter	4 字节	上一次直接发送测试中发送的数据包数量

生成事件：
控制器将生成指令完成事件。

4.8.6 Aci_Hal_Tone_Start

表 298. Aci_Hal_Tone_Start

指令名	参数	返回
Aci_Hal_Tone_Start (0xFC15)	通道 ID	状态

说明：
该指令在特定通道上启动载频。特定通道上的频率正弦波只能用于调试用途。对于 40 个 BLE 通道而言，通道 ID 是范围为 0x00 至 0x27 的参数，例如 0x00 表示 2.402 GHz，0x01 表示 2.404 GHz 等等。
蓝牙正常工作时，不应使用该指令。
应通过 Aci_Hal_Tone_Stop 指令停止。

表 299. Aci_Hal_Tone_Start 指令参数

参数	大小	说明
Channel ID	1 字节	BLE 通道 ID，从 0x00 至 0x27，表示 $(2.402 + 2 \times 0xXX)$ GHz

表 300. Aci_Hal_Tone_Start 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS

生成事件：
控制器将生成指令完成事件。

4.8.7 Aci_Hal_Tone_Stop

表 301. Aci_Hal_Tone_Stop

指令名	参数	返回
Aci_Hal_Tone_Stop (0xFC16)		状态

说明:

该指令用于停止之前启动的 Aci_Hal_Tone_Start 指令。

表 302. Aci_Hal_Tone_Stop 返回参数

参数	大小	说明
Status	1 字节	0x00: BLE_STATUS_SUCCESS

生成事件: 控制器将生成指令完成事件。

4.8.8 Aci_Hal_Get_Link_Status

表 303. Aci_Hal_Get_Link_Status

指令名	参数	返回
Aci_Hal_Get_Link_Status (0xFC17)		状态 Link_Status Connection_Handle

说明:

该指令用于返回链路层状态和连接句柄。

表 304. Aci_Hal_Get_Link_Status 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功

表 304. Aci_Hal_Get_Link_Status 返回参数

参数	大小	说明
Link_Status	8 字节	每个客户端的链路状态，其中字节 0 是客户端 0 的链路状态，字节 1 是客户端 1 的链路状态，以此类推。 0: 空闲 1: 广告 2: 以从设备角色连接 3: 扫描 4: 保留 5: 以主设备角色连接 6: 发送测试 7: 接收测试
Connection_Handle	16 字节	每个客户端的连接句柄，其中字节 [0,1] 表示客户端 0 的连接句柄，字节 [2,3] 表示客户端 1 的连接句柄，以此类推。

4.8.9 Aci_Hal_Get_Fw_Build_Number

表 305. Aci_Hal_Get_Fw_Build_Number

指令名	参数	返回
Aci_Hal_Get_Fw_Build_Number (0xFC00)		状态 Revision_Number

说明:

该指令用于检索固件版本号。

表 306. Aci_Hal_Get_Fw_Build_Number 返回参数

参数	大小	说明
Status	1 字节	0x00: 成功
Revision_Number	2 字节	固件版本号

4.8.10 Aci_Hal_Get_Anchor_Period

表 307. Aci_Hal_Get_Anchor_Period

指令名	参数	返回
Aci_Hal_Get_Anchor_Period (0xFCF8)		状态 Anchor_Interval

说明：
该指令用于检索关于当前锚定间隔和可分配时隙的信息。

表 308. Aci_Hal_Get_Anchor_Period 返回参数

参数	大小	说明
Status	1 字节	0x00：成功 0x43：忙。在连接更新被挂起时返回该状态。
Anchor_Interval	4 字节	当前锚定间隔为 0.625 ms 的倍数。
Max slot	4 字节	可以分配给新连接时隙的最大可用长度（0.625 ms 的倍数）。

4.9 HCI VS 事件

4.9.1 Evt_Blue_Initialized 事件

当 BlueNRG-MS 固件正常启动时，它会给用户一个 Evt_Blue_Initialized 事件，表明系统已经启动（原因代码 0x01）。

Evt_Blue_Initialized 事件是与其他事件的格式相同的 ACI 事件。[表 309](#) 中描述了所有字段。

表 309. Evt_Blue_Initialized 事件

参数	大小	说明
event_code	1 字节	0x0001 - Evt_Blue_Initialized 事件的事件代码
Reason code	1 字节	0x00 – 保留 0x01 – 固件正常启动 0x02 – Aci_Updater_Start 指令导致进入更新器模式 0x03 - 不良 BLUE 标记导致进入更新器模式 0x04 - 通过 IRQ 引脚进入更新器模式 0x05 - 看门狗导致复位 0x06 - 死锁导致复位 0x07 - 欠压复位 0x08 - 冲突导致复位（NMI 或硬性故障） 0x09 - ECC 错误导致复位



4.9.2 Evt_Blue_Lost_Events

表 310. Evt_Blue_Lost_Events

参数	大小	说明
Event_Code	2 字节	0x0002: Evt_Blue_Lost_Events 事件的事件代码
Lost_Events_Map	8 字节	丢失事件位图

丢失事件映射的每一位对应于特定事件：

表 311. Lost_Events_Map (8 字节)

Lost_Event_Map 位	说明
0	EVT_DISCONN_COMPLETE
1	EVT_ENCRYPT_CHANGE
2	EVT_READ_REMOTE_VERSION_COMPLETE
3	EVT_CMD_COMPLETE
4	EVT_CMD_STATUS
5	EVT_HARDWARE_ERROR
6	EVT_NUM_COMP_PKTS
7	EVT_ENCRYPTION_KEY_REFRESH
8	EVT_BLUE_INITIALIZED
9	EVT_BLUE_GAP_SET_LIMITED_DISCOVERABLE
10	EVT_BLUE_GAP_PAIRING_CMPLT
11	EVT_BLUE_GAP_PASS_KEY_REQUEST
12	EVT_BLUE_GAP_AUTHORIZATION_REQUEST
13	EVT_BLUE_GAP_SECURITY_REQ_INITIATED
14	EVT_BLUE_GAP_BOND_LOST
15	EVT_BLUE_GAP_PROCEDURE_COMPLETE
16	EVT_BLUE_GAP_ADDR_NOT_RESOLVED
17	EVT_BLUE_L2CAP_CONN_UPDATE_RESP
18	EVT_BLUE_L2CAP_PROCEDURE_TIMEOUT
19	EVT_BLUE_L2CAP_CONN_UPDATE_REQ
20	EVT_BLUE_GATT_ATTRIBUTE_MODIFIED
21	EVT_BLUE_GATT_PROCEDURE_TIMEOUT
22	EVT_BLUE_EXCHANGE_MTU_RESP
23	EVT_BLUE_ATT_FIND_INFORMATION_RESP
24	EVT_BLUE_ATT_FIND_BY_TYPE_VAL_RESP
25	EVT_BLUE_ATT_READ_BY_TYPE_RESP

表 311. Lost_Events_Map (8 字节)

Lost_Event_Map 位	说明
26	EVT_BLUE_ATT_READ_RESP
27	EVT_BLUE_ATT_READ_BLOB_RESP
28	EVT_BLUE_ATT_READ_MULTIPLE_RESP
29	EVT_BLUE_ATT_READ_BY_GROUP_RESP
30	EVT_BLUE_ATT_WRITE_RESP
31	EVT_BLUE_ATT_PREPARE_WRITE_RESP
32	EVT_BLUE_ATT_EXEC_WRITE_RESP
33	EVT_BLUE_GATT_INDICATION
34	EVT_BLUE_GATT_NOTIFICATION
35	EVT_BLUE_GATT_PROCEDURE_COMPLETE
36	EVT_BLUE_GATT_ERROR_RESP
37	EVT_BLUE_GATT_DISC_READ_CHARAC_BY_UUID_RESP
38	EVT_BLUE_GATT_WRITE_PERMIT_REQ
39	EVT_BLUE_GATT_READ_PERMIT_REQ
40	EVT_BLUE_GATT_READ_MULTI_PERMIT_REQ
41	EVT_BLUE_GATT_TX_POOL_AVAILABLE
42	EVT_BLUE_GATT_SERVER_RX_CONFIRMATION
43	EVT_BLUE_GATT_PREPARE_WRITE_PERMIT_REQ
44	EVT_LL_CONNECTION_COMPLETE
45	EVT_LL_ADVERTISING_REPORT
46	EVT_LL_CONNECTION_UPDATE_COMPLETE
47	EVT_LL_READ_REMOTE_USED_FEATURES
48	EVT_LL_LTK_REQUEST

4.9.3 故障数据事件

在发生 NMI 或硬性故障时，在 Evt_Blue_Initialized 事件后自动发送故障数据事件。

表 312. 故障数据事件

参数	大小	说明
Event_Code	2 字节	0x0003: Evt_Fault_Data 事件的事件代码
Fault reason	1 字节	6: NMI 故障 7: 硬性故障
SP	4 字节	MCU SP 寄存器



表 312. 故障数据事件

参数	大小	说明
R0	4 字节	MCU R0 寄存器
R1	4 字节	MCU R1 寄存器
R2	4 字节	MCU R2 寄存器
R3	4 字节	MCU R3 寄存器
R12	4 字节	MCU R12 寄存器
LR	4 字节	MCU LR 寄存器
PC	4 字节	MCU PC 寄存器
xPSR	4 字节	MCU xPSR 寄存器
Fault data length	1 字节	N: 额外故障数据部分的长度
Fault data	N 字节	额外的故障转储数据

4.9.4 硬件错误事件代码

HCI Hardware_Error_Event 可报告下列错误代码：

- 0: SPI 帧错误
- 1: 无线状态错误。当启动指令已发送时，如果无线 FSM 没有达到激活 2 状态，将生成该错误代码。这可能是晶振启动缓慢造成的。
- 2: 定时器上溢错误。

如果没有在预期时间发出无线设备启动指令，将生成该错误代码。这可能是晶振启动缓慢造成的。

5 SPI 接口

BlueNRG-MS 设备提供 SPI 接口，可用于连接外部设备。外部设备通常为应用处理器。它通过发送 ACI 指令控制 BlueNRG-MS 设备。BlueNRG-MS 生成 ACI 事件以向外部应用处理器报告。ACI 指令和事件均通过 SPI 总线发送。

本章将介绍 BlueNRG-MS 硬件 SPI 接口。此外，还将介绍 BlueNRG-MS 和外部设备之间 SPI 总线上的通信协议。

5.1 硬件 SPI 接口

BlueNRG-MS SPI 接口符合 Motorola 协议，有 5 条线：CLK（时钟）、nCS（芯片选择）、MOSI（主设备输出从设备输入）、MISO（主设备输入从设备输出）和 DataRdy_IRQ（发送中断请求）。

在 SPI 总线上，BlueNRG-MS 总是充当从设备。外部设备必须为 SPI 主设备。

下表显示了 BlueNRG-MS 设备上的 5 个 SPI 引脚，包括硅芯片的引脚数量和从 BlueNRG-MS 侧来看的引脚方向。

表 313. SPI 引脚

引脚名称	引脚编号	方向
SPI_CLK	2	输入
SPI_NCS	31	输入
SPI_MOSI	1	输入
SPI_MISO	32	输出
SPI_IRQ	3	输出

- 下面是对 SPI 参数的总结：
- 1. CS 线为低电平有效。
 - 2. 时钟信号的频率可高至 8 MHz 或低至 100 kHz。
 - 3. 时钟为低电平有效，即时钟极性 CPOL = 0。
 - 4. 数据适用于时钟前沿，即时钟相位 CPHA = 0。
 - 5. 数据传输总是基于 8 位字节。
 - 6. 对于每个字节而言，先发送最高有效位。


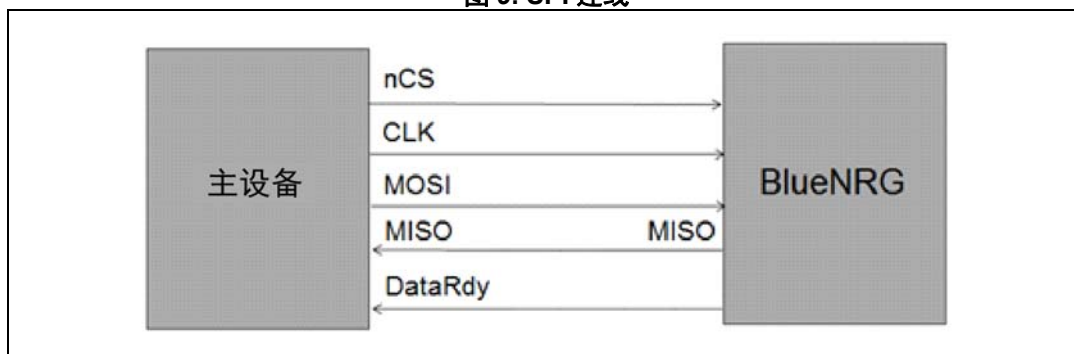
 9 显示了如何将 BlueNRG-MS SPI 总线连接到外部 SPI 主设备。



图 9. SPI 连线



SPI 主设备（即应用处理器）始终控制着启动 SPI 事务的时间。启动可以在任何时间发生。通过拉低 nCS 线激活 BlueNRG-MS 设备来启动。然后，主设备应提供 SPI 时钟。除了时钟，主设备还应通过 MOSI 线输出 SPI 总线上的数据（空数据或有效数据）。与此同时，从设备返回来自 MISO 线的数据，因为 SPI 原则上是移位寄存器。

当 SPI 主设备要向 BlueNRG-MS 发送数据（即 SPI 写入）时，它将 nCS 设置为低电平，然后发送 SPI 时钟和空数据以从 BlueNRG-MS 读取 SPI 数据头。SPI 数据头格式将在下面一节进行介绍。如果 SPI 数据头指示 BlueNRG-MS 的缓冲区有足够空间，则 SPI 主设备可以通过 SPI 总线发送实际数据。SPI 主设备不应覆盖 BlueNRG-MS SPI 缓冲区，否则通信将失败。

当 BlueNRG-MS 要向 SPI 主设备发送数据（即 SPI 读取）时，它必须使用 IRQ 引脚，因为 SPI 从设备不能控制 nCS 线。BlueNRG-MS 会将 IRQ 引脚设置为高电平，从而通知 SPI 主设备有要读取的数据。然后，SPI 主设备应启动 SPI 事务，读取 SPI 数据头，并从中得知它应从 BlueNRG-MS 回读的字节数量。SPI 主设备不应读取 SPI 数据头指示的字节以外的字节，否则通信将失败。

BlueNRG-MS 的 DataRdy（IRQ）引脚应按以下方式工作：(1) 当 BlueNRG-MS 有数据要发送给 SPI 主设备时，它必须为高电平；(2) 当 BlueNRG-MS 没有数据要发送给 SPI 主设备时，它必须具有高阻抗。在 SPI 主设备侧，该引脚必须配置为输入引脚。SPI 主设备可以轮询该引脚，或者将它用作中断源，用于检测何时需要 SPI 读取。另外，还应有位于 SPI 主设备 IRQ 引脚上或外接的下拉电阻。该下拉电阻确保 DataRdy 恢复为 0，以免误导 SPI 主设备读错数据。请注意，IRQ 引脚在没有数据发送时必须为 High-Z 状态的原因是，我们还将该引脚用作 BlueNRG-MS 启动时的输入。IRQ 引脚值将决定是否更新器模式。在 BlueNRG-MS 进入正常工作状态后，将 IRQ 引脚配置为 SPI 接口的输出。

还需注意，当 BlueNRG-MS 不发送数据时，MISO 也进入 High-Z 状态。因此，同一总线上可以有多个 SPI 从设备。当 BlueNRG-MS 不访问总线时，其他 SPI 从设备可以访问。

5.2 SPI 通信协议

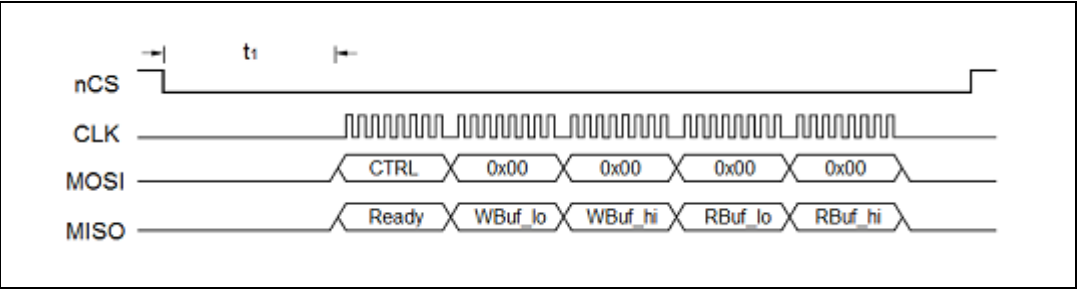
为了与 BlueNRG-MS 通信，SPI 总线上的数据格式必须符合本节中的描述。

SPI 事务的定义是从主设备将 nCS 线设置为低电平这一时刻开始，直至 nCS 线恢复高电平这一时刻为止。

每个 SPI 事务应包含 1 个数据帧。每个数据帧应包含至少 5 字节的数据头，并可能有 0-N 字节的数据。

5.2.1 数据头

图 10. SPI 数据头格式



上面的图 10 所示为一个 SPI 帧，具有至少 5 字节的 SPI 数据头和 0 字节的 SPI 数据。

主设备的数据头通过 MOSI 线传输，是 1 个控制字节和 4 字节 0x00。控制字节的值只能是 0x0A（SPI 写入）或 0x0B（SPI 读取）。

BlueNRG-MS 同时通过 MISO 线返回从设备数据头。第 1 个字节为 SPI READY 指示。它必须为 0x02，以指示从设备 SPI 接口已准备就绪。如果是 0x02 以外的任何值，主设备必须忽略后面的 4 个字节并中止 SPI 事务。如果 BlueNRG-MS SPI 准备就绪（Ready 字节 = 0x02），后面的 4 个字节给出用于写入和读取的 2 个缓冲区大小，每个缓冲区 2 个字节。对于 2 字节缓冲区，先是低字节，然后是高字节。写入缓冲区大小表示主设备可以写入 BlueNRG-MS 的字节数。读取缓冲区大小表示 BlueNRG-MS 中等待主设备读取的字节数。

例如，如果从设备数据头为 0x0200010500，则写入缓冲区大小为 0x0100，读取缓冲区大小为 0x0005。因此，主设备不能向 BlueNRG-MS 写入超过 256 字节的数据，并应尽快从 BlueNRG-MS 回读 5 字节。

SPI 主设备可通过发送空 SPI 数据头轮询 BlueNRG-MS，以便检查 BlueNRG-MS SPI 数据头中的值。例如，SPI 主设备可以发送只有 5 字节数据头的 SPI 帧：CTRL 字节可以是写入或读取，后面是 4 字节的 0x00 空字节。在数据头之后，SPI 主设备停止帧，不发送任何实际数据。因此，实际上不会发生写入 / 读取操作。



5.2.2 向 BlueNRG-MS 写入数据

为了向 BlueNRG-MS 写入数据，SPI 主设备必须启动 SPI 帧以检查 BlueNRG-MS SPI 数据头，并且必须满足下列 3 个条件：

1. 主设备以 0x0A 的形式发送 CTRL 字节。
2. 从设备以 0x02 的形式回复 SPI READY 字节。
3. 写入缓冲区大小大于 0。

如果是这样，主设备可以在同一事务中将数据置于 MOSI 线上的 5 个头字节之后。

如果主设备足够快，它可以在满足上述 3 个条件时在同一 SPI 事务中立即开始发送数据字节。或者，主设备可以选择暂停 SPI 事务，继续准备数据，并在后面的 SPI 事务中发送数据。

SPI 主设备允许写入少于 / 禁止写入大于写入缓冲区大小的字节数。如果 ACI 指令数据包大于 128 字节，主设备必须在 2 个 SPI 写入事务中发送它。主设备首先发送 128 字节，等到写入缓冲区再次可用时写入剩余字节。

BlueNRG-MS 将对在 MOSI 线上实际接收的字节数进行计数。因此，主设备无需指示它要写入的字节数。

写入缓冲区大小值是载荷的数据字节数。它不包括 5 个头字节。

当主设备通过 MOSI 线发送实际数据字节时，BlueNRG-MS 将从自己的移位寄存器返回相同数量的字节。主设备应忽略这些字节。

5.2.3 从 BlueNRG-MS 中读取数据

BlueNRG-MS 使用 IRQ 引脚通知 SPI 主设备有要读取的数据。当 SPI 主设备检测到 IRQ 引脚为高电平时，应从 BlueNRG-MS 读取数据。

该协议类似于 SPI 写入。SPI 主设备启动事务以检查 SPI 数据头。主设备将 0xB 放入 CTRL 字节，并再次检查从设备是否准备就绪（0x02），以及读取缓冲区大小是否非零。然后，主设备通过 MOSI 线发送特定数量的空字节（例如，0x00 或 0xFF），同时从 MISO 线回读从设备数据。读取过程可以发生在 1 个或多个 SPI 事务中。

只要 IRQ 引脚为高电平，SPI 主设备就应从 BlueNRG-MS 读取。

5.2.4 BlueNRG-MS 睡眠模式下的 SPI 操作

BlueNRG-MS 设备设计有深度睡眠模式，在这种模式下，它将关闭大部分内部功能以降低功率。当没有正在进行的蓝牙活动时，BlueNRG-MS 自动进入深度睡眠模式。因此，当主设备启动 SPI 事务时，BlueNRG-MS 很可能处于睡眠状态。

BlueNRG-MS 在 nCS 线被设置为低电平时立即唤醒。但是，随后需要初始化并准备 SPI 缓冲区。这需要一些时间。因此，SPI 总线上可能按顺序发生如下活动：

1. 主设备将 nCS 线设置为低电平，BlueNRG-MS 唤醒。BlueNRG-MS 等待指令，在 2 ms 后进入睡眠状态。
2. 主设备发送数据头，但 BlueNRG-MS 回复的 READY 字节为 0x00 或 0xFF。这意味着 BlueNRG-MS SPI 接口仍未初始化。
3. 主设备持续轮询 BlueNRG-MS（释放并重新断言 CS）。为了再次读取数据头，SPI 主设备必须在上一次 CS 断言后的 2ms 内释放并断言 CS 线。一段时间后，READY 字节变为 0x02，但两个缓冲区大小均为 0x00。这表明 BlueNRG-MS SPI 已初始化，但缓冲区尚未准备就绪。
4. 一旦缓冲区大小变为非零，主设备即可向 BlueNRG-MS 写入数据。

经测量，为了让 BlueNRG-MS 完成对 SPI 通信的准备活动，主设备应预留约 0.5 ms 的时间。请注意，0.5 ms 是 BlueNRG-MS 唤醒时必须初始化其非保留 RAM 的时间（即设备处于模式 1 - 仅从设备模式）。如果无需初始化非保留 RAM，SPI 可以更快地完成准备活动，这一时间可以缩短为 0.3 ms。

仅当 SPI 总线处于空闲状态一段时间后，才需要此等待操作。如果主设备从第 2 个写入事务开始连续写入，则无需等待，因为 BlueNRG-MS 在仍有等待处理的 SPI 数据时不会进入睡眠状态。此外，在从 BlueNRG-MS 读取时无需等待，因为它在仍然保存着主设备需要的数据时不能进入睡眠状态。

6 版本历史

表 314. 文档版本历史

日期	版本	变更
2015 年 3 月 4 日	1	初始版本。
2015 年 7 月 21 日	2	更新了图 3、表 38、表 39、表 123、表 155、表 256 和表 292。 增加了第 4.6.8 节。
2015 年 9 月 30 日	3	更新了 GATT 事件部分中支持的全部 GATT 事件。 更新了一些 ACI 参数名称和字段。 对整个文档的少量文字修订。
2015 年 12 月 2 日	4	修正了一些 ACI 和事件名称。对整个文档的少量文字修订。
2016 年 8 月 25 日	5	更新了 ACI 指令和事件。

表 315. 中文文档版本历史

日期	版本	变更
2017 年 1 月 24 日	1	中文初始版本。

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2017 STMicroelectronics - 保留所有权利