

Find and Bind

1. 烧写 DimmerSwitch_JN5169_DR1199.bin 到扩展板 DR1199。
2. 烧写 DimmableLight_JN5169_DR1175_LED_EXP_MONO.bin 到扩展板 DR1175。
3. 上述二个板都加入到同一个网络。
4. 按下 DR1199 板的 SW3+DIO8 按钮（先按 SW3，再按 DIO8，释放 SW3，保持 DIO8 按下状态），进入到 Find and Bind 模式。
5. 间隔 1 秒钟按 DR1175 板的/REST 复位键三次，令 DimmableLight 进入 Identify 状态。此时板上的三个 LED 将会交替亮灭。
6. 按 DR1199 板的 SW1，完成 Bind 操作。

至此 Find and Bind 完成。按 SW1 发 On 命令，SW2 发 Off 命令，SW3 发 Move Up 命令，SW4 发 Move Down 命令。

Ch.	Stack	Layer	Packet Information	MAC Src.	MAC Dst.	NWK Src.	NWK Dst.
11	ZigBee	MAC	Acknowledgement				
11	ZigBee	ZDP	Management Permit Joining Request	0xF0CE	0xFFFF	0xF0CE	0xFFFC
11	ZigBee	ZDP	Management Permit Joining Request	0x0000	0xFFFF	0xF0CE	0xFFFC
11	ZigBee	MAC	Data Request	0x1072	0xF0CE		
11	ZigBee	MAC	Acknowledgement				
11	ZigBee	ZDP	Management Permit Joining Request	0xF0CE	0xFFFF	0xF0CE	0xFFFC
11	ZigBee	MAC	Data Request	0x1072	0xF0CE		
11	ZigBee	MAC	Acknowledgement				
11	ZigBee	ZCL	Identify: Identify Query	0x1072	0xF0CE	0x1072	0xFFFF
11	ZigBee	MAC	Acknowledgement				
11	ZigBee	MAC	Data Request	0x1072	0xF0CE		
11	ZigBee	MAC	Acknowledgement				
11	ZigBee	ZCL	Identify: Identify Query Response	0xF0CE	0x1072	0xF0CE	0x1072
11	ZigBee	MAC	Acknowledgement				
11	ZigBee	APS	Acknowledgement	0x1072	0xF0CE	0x1072	0xF0CE
11	ZigBee	MAC	Acknowledgement				
11	ZigBee	ZDP	Simple Descriptor Request	0x1072	0xF0CE	0x1072	0xF0CE

▷ Frame Information: (50 bytes)

▷ MAC Header: (9 bytes)

▲ MAC Payload: (39 bytes)

▷ NWK Header: 0xBA1EF0CE10720248

▷ NWK Aux Header: (14 bytes)

▲ NWK Payload: (13 bytes)

▷ APS Header: 0x5F01010400030140

▲ APS Payload: 0x00B1000119

▲ ZCL Header: 0x000119

▲ Frame Control: 0x19

.... 01 = Frame Type: [0x1] (

.... 00 = Manufacturer Specif

.... 1... = Direction: [0x1] Fr

...1 = Disable Default Res

000. = Reserved: 0x0

Transaction Sequence Number: 1

Command ID: [0x00] Identify Query

▲ ZCL Payload: 0x00B1

Timeout: 177

NWK MIC: 0x1A2D9D11

OS_TASK(APP_ZHA_Switch_Task)

→ vHandleAppEvent

```
PRIVATE void vHandleAppEvent( APP_tsEvent sAppEvent )
{
    switch(sAppEvent.eType)
    {
        case APP_E_EVENT_BUTTON_DOWN:
        case APP_E_EVENT_BUTTON_UP:
            vApp_ProcessKeyCombination(sAppEvent);
            break;
        .....
    }
}
```

vApp_ProcessKeyCombination

→ vSwitchStateMachine

→ vManageLightControlState

→ *peSwitchState = EZMODE_FIND_AND_BIND_MODE;

→ vManageEZModeCommissioning

→ eEZ_FindAndBind

→ eStartEZInitiator

→ eRestartEZInitiator

→ eDiscover

→eCLD_IdentifyCommandIdentifyQueryRequestSend (0xFFFE)

APP_ZCL_cbEndpointCallback

→vEZ_EPCallBackHandler(GENERAL_CLUSTER_ID_IDENTIFY)

→ vEZ_IEEEAddressReqSend

→ vEZ_SimpleDescriptorSend

OS_TASK(APP_ZHA_Switch_Task)

→vEZ_EZModeNWKFindAndBindHandler

→ vEZ_HandleZDPResponse

→ vEZ_SimpleDescriptorSend

→ vEZ_HandleSimpleDescriptorResponse

→ vBindCluster

Commissioning Groups

1. 烧写 DimmerSwitch_JN5169_DR1199.bin 到扩展板 DR1199。
2. 烧写 DimmableLight_JN5169_DR1175_LED_EXP_MONO.bin 到扩展板 DR1175。
3. 上述二个板都加入到同一个网络。
4. 按下 DR1199 板的 DIO8 按钮，保持 DIO8 按下状态。DR1199 将会广播“Match Descriptor Request”
5. 当 DR1199 板收到“Match Descriptor Response”后，发送 Identify 给相应的回应节点。
6. 当 DR1175 收到 Identify 请求后，将会闪烁三次。
7. 按 DR1199 板的 SW1，完成 AddGroup 操作。

至此 Commissioning Groups 完成。按 SW1 发 On 组播命令，SW2 发 Off 组播命令，SW3 发 Move Up 组播命令，SW4 发 Move Down 组播命令。

11	ZigBee	ZDP	Match Descriptor Request	0xF0CE	0xFFFF	0x0748	0xFFFF	▷ MAC Header: (9 bytes)
11	ZigBee	MAC	Data Request	0x0748	0x0000			▲ MAC Payload: (40 bytes)
11	ZigBee	MAC	Acknowledgement					▷ NWK Header: 0x5F1E00007480248
11	ZigBee	ZDP	Match Descriptor Response	0x0000	0x0748	0x0000	0x0748	▷ NWK Aux Header: (14 bytes)
11	ZigBee	MAC	Acknowledgement					▲ MAC Payload: (14 bytes)
11	ZigBee	ZDP	Match Descriptor Request	0xF0CE	0xFFFF	0x0748	0xFFFF	▷ APS Header: 0x0800000080060000
11	ZigBee	MAC	Data Request	0x0748	0x0000			▲ APS Payload: 0x010100000007
11	ZigBee	MAC	Acknowledgement					▲ Match Descriptor Response: 0x010100000007
11	ZigBee	MAC	Data Request	0x0748	0x0000			ZDP Transaction Sequence Number: 215
11	ZigBee	MAC	Acknowledgement					Status: [0x00] Success
11	ZigBee	MAC	Data Request	0x0748	0x0000			NWK Address of Interest: 0x0000
11	ZigBee	MAC	Acknowledgement					Matched Endpoints Count: 1
11	ZigBee	MAC	Data Request	0x0748	0x0000			▷ Matched Endpoints List: 0x01
11	ZigBee	MAC	Acknowledgement					
11	ZigBee	MAC	Data Request	0x0748	0x0000			▷ MAC Header: (9 bytes)
11	ZigBee	MAC	Acknowledgement					▲ MAC Payload: (39 bytes)
11	ZigBee	ZCL	Identify: Identify	0x0748	0x0000	0x0748	0xF0CE	▷ NWK Header: 0x551E0748F0CE0248
11	ZigBee	MAC	Acknowledgement					▷ NWK Aux Header: (14 bytes)
11	ZigBee	MAC	Data Request	0x0748	0x0000			▲ NWK Payload: (13 bytes)
11	ZigBee	MAC	Acknowledgement					▷ APS Header: 0x1701010400030100
11	ZigBee	ZCL	Identify: Identify	0x0000	0xF0CE	0x0748	0xF0CE	▲ APS Payload: 0x000500011
11	ZigBee	MAC	Acknowledgement					ZCL Header: 0x000011
11	ZigBee	MAC	Data Request	0x0748	0x0000			Frame Control: 0x11
11	ZigBee	MAC	Acknowledgement					Transaction Sequence Number: 0
11	ZigBee	MAC	Data Request	0x0748	0x0000			Command ID: [0x00] Identify
11	ZigBee	MAC	Acknowledgement					▲ ZCL Payload: 0x0005
11	ZigBee	MAC	Acknowledgement					Identify Time: [5] Seconds
11	ZigBee	MAC	Acknowledgement					
11	ZigBee	MAC	Acknowledgement					▷ MAC Header: (9 bytes)
11	ZigBee	ZCL	Groups: Add Group	0x0748	0x0000	0x0748	0xF0CE	▲ MAC Payload: (40 bytes)
11	ZigBee	MAC	Acknowledgement					▷ NWK Header: 0x111EF0CE07480248
11	ZigBee	ZCL	Groups: Add Group	0x0000	0xF0CE	0x0748	0xF0CE	▷ NWK Aux Header: (14 bytes)
11	ZigBee	MAC	Acknowledgement					▲ NWK Payload: (14 bytes)
11	ZigBee	ZCL	Groups: Add Group Respon...	0xF0CE	0x0000	0xF0CE	0x0748	▷ APS Header: 0xD801010400040140
11	ZigBee	MAC	Acknowledgement					▲ APS Payload: 0x074800000119
11	ZigBee	MAC	Data Request	0x0748	0x0000			ZCL Header: 0x000119
11	ZigBee	MAC	Acknowledgement					Frame Control: 0x19
11	ZigBee	ZCL	Groups: Add Group Respon...	0x0000	0x0748	0xF0CE	0x0748	Transaction Sequence Number: 1
11	ZigBee	MAC	Acknowledgement					Command ID: [0x00] Add Group Response
11	ZigBee	APS	Acknowledgement	0x0748	0x0000	0x0748	0xF0CE	▲ ZCL Payload: 0x074800
11	ZigBee	MAC	Acknowledgement					Status: [0x00] Success
11	ZigBee	MAC	Acknowledgement					Group ID: 0x0748
11	ZigBee	MAC	Acknowledgement					
11	ZigBee	MAC	Acknowledgement					▷ MAC Header: (9 bytes)
11	ZigBee	ZCL	On/Off: On	0x0748	0x0000	0x0748	0xFFFF	▲ MAC Payload: (38 bytes)
11	ZigBee	MAC	Acknowledgement					▷ NWK Header: 0x361E0748FFFD0248
11	ZigBee	ZCL	On/Off: On	0x0000	0xFFFF	0x0748	0xFFFF	▷ NWK Aux Header: (14 bytes)
11	ZigBee	ZCL	On/Off: On	0xF0CE	0xFFFF	0x0748	0xFFFF	▲ NWK Payload: (12 bytes)
11	ZigBee	ZCL	On/Off: On	0xF0CE	0xFFFF	0x0748	0xFFFF	▷ APS Header: (9 bytes)
11	ZigBee	ZCL	On/Off: On	0xF0CE	0xFFFF	0x0748	0xFFFF	Frame Control: 0x0C
11	ZigBee	MAC	Data Request	0x0748	0x0000			Group Address: 0x0748
11	ZigBee	MAC	Acknowledgement					Cluster ID: [0x0006] General: On/Off
11	ZigBee	ZCL	On/Off: Off	0x0748	0x0000	0x0748	0xFFFF	Profile ID: [0x0104] ZigBee Home Automation
11	ZigBee	MAC	Acknowledgement					Source Endpoint: 0x01
11	ZigBee	ZCL	On/Off: Off	0x0000	0xFFFF	0x0748	0xFFFF	APS Counter: 17
11	ZigBee	ZCL	On/Off: Off	0xF0CE	0xFFFF	0x0748	0xFFFF	▲ APS Payload: 0x010111
11	ZigBee	ZCL	On/Off: Off	0xF0CE	0xFFFF	0x0748	0xFFFF	ZCL Header: 0x010111
11	ZigBee	NWK	Link Status	0x0000	0xFFFF	0x0000	0xFFFF	Frame Control: 0x11
11	ZigBee	ZCL	On/Off: Off	0xF0CE	0xFFFF	0x0748	0xFFFF	Transaction Sequence Number: 1
11	ZigBee	ZCL	On/Off: Off	0xF0CE	0xFFFF	0x0748	0xFFFF	Command ID: [0x01] On

OS_TASK(APP_ZHA_Switch_Task)

→ vHandleAppEvent

PRIVATE void vHandleAppEvent(APP_tsEvent sAppEvent)

```
{
    switch(sAppEvent.eType)
    {
        case APP_E_EVENT_BUTTON_DOWN:
        case APP_E_EVENT_BUTTON_UP:
            vApp_ProcessKeyCombination(sAppEvent);
            break;
        .....
    }
}
```

vApp_ProcessKeyCombination

→vSwitchStateMachine

→vManageLightControlState

→*peSwitchState = IDENTIFY_MODE;

→ vAppDiscover

→ OS_eActivateTask(APP_CommissionTimerTask);

OS_TASK(APP_CommissionTimerTask)

→ vSendMatchDesc

→ vSetTheNextLightIdentify

→ vAppIdentify

→ eCLD_IdentifyCommandIdentifyRequestSend

OS_TASK(APP_ZHA_Switch_Task)

→ vHandleMatchResponses

→ sDeviceState.sMatchDev[...]

vApp_ProcessKeyCombination

→vSwitchStateMachine

→vManageIdentifyMode

→ vAppAddGroup

→ eCLD_GroupsCommandAddGroupRequestSend