How to use HID

HID Report Descriptor for MCU:

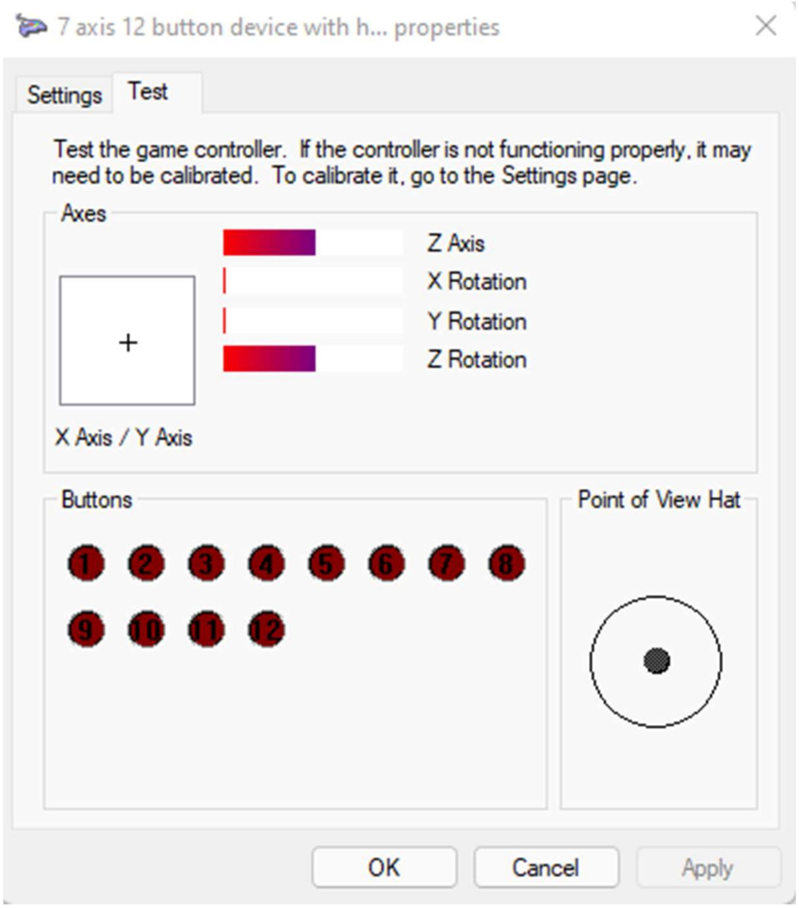
HID defined for gamepad ==================================

```
REPORT_ID(1),      GAMEPAD_ID,
// ------------------------------------------------ Dpad and Buttons (1 to 16)
USAGE(1),0x39, //     USAGE_PAGE (Dpad)
LOGICAL_MINIMUM(1),  0x01, //     LOGICAL_MINIMUM (0)
LOGICAL_MAXIMUM(1),  0x08, //     LOGICAL_MAXIMUM (1)
REPORT_SIZE(1),      0x04, //     REPORT_SIZE (4)
REPORT_COUNT(1),     0x01, //     REPORT_COUNT (1)
HIDINPUT(1),         0x42, //     INPUT (Data, Variable, Absolute)  1 Dpad

USAGE_PAGE(1),       0x09, //     USAGE_PAGE (Button)   12 Buttons
USAGE_MINIMUM(1),    0x01, //     USAGE_MINIMUM (Button 1)
USAGE_MAXIMUM(1),    0x0C, //     USAGE_MAXIMUM (Button 12)
LOGICAL_MINIMUM(1),  0x00, //     LOGICAL_MINIMUM (0)
LOGICAL_MAXIMUM(1),  0x01, //     LOGICAL_MAXIMUM (1)
REPORT_SIZE(1),      0x01, //     REPORT_SIZE (1)
REPORT_COUNT(1),     0x0C, //     REPORT_COUNT (12)
HIDINPUT(1),         0x02, //     INPUT (Data, Variable, Absolute) ;12 button bits

// ------------------------------------------------ X/Y/Z position
USAGE_PAGE(1),             0x01,         //            USAGE_PAGE (Generic Desktop)
USAGE(1),                  0x30,         //     USAGE (X)
USAGE(1),                  0x31,         //     USAGE (Y)
USAGE(1),                  0x32,         //     USAGE (Z)
0x15,                      0x00,         //     LOGICAL_MINIMUM (0)
```

```
0x26,                    0xFF, 0x00,   //    LOGICAL_MAXIMUM (255)
REPORT_SIZE(1),    0x08,        //              REPORT_SIZE (8)
REPORT_COUNT(1),   0x03,        //              REPORT_COUNT (3)
HIDINPUT(1),       0x02,        //    INPUT (Data,Var,Abs)
// ------------------------------------------------ rX/rY/rZ position
USAGE(1),0x33,  //    USAGE (rX) Left Trigger
USAGE(1),0x34,  //    USAGE (rY) Right Trigger
USAGE(1),0x35,  //    USAGE (rZ)
0x15, 0x00,    //          Logical Minimum (0)
0x26, 0xFF, 0x00,       //         Logical Maximum (255)
REPORT_SIZE(1),     0x08,  //    REPORT_SIZE (8)
REPORT_COUNT(1),    0x03,  //    REPORT_COUNT (3)
HIDINPUT(1),        0x02,  //    INPUT (Data, Variable, Absolute) ;3 bytes (rX,rY,rZ)
```



7 axis 12 button device with h... properties

Test the game controller. If the controller is not functioning properly, it may need to be calibrated. To calibrate it, go to the Settings page.

Vendor defined input report ================================

```
// ------------------------------------------------ Vendor defined
0x09,  0x03,    // USAGE ID - Vendor defined
0x15,  0x00,    // LOGICAL_MINIMUM (0)
0x26,  0xff, 0x00,         // LOGICAL_MAXIMUM (255)
REPORT_SIZE(1),0x08,       // REPORT_SIZE (8)
REPORT_COUNT(1), GP_VDATAIN_LEN,        // REPORT_COUNT (10)
//You can change the data length
0x81, 0x02,     // INPUT (Data,Var,Abs)
```

Vendor defined output report ===============================

```
// The Output report
0x09,   0x03,   //     USAGE ID - Vendor defined
0x15,   0x00,   //     LOGICAL_MINIMUM (0)
0x26,   0xff, 0x00,        //     LOGICAL_MAXIMUM (1)
REPORT_SIZE(1), 0x08,      //     REPORT_SIZE (1)
REPORT_COUNT(1), GP_VDATAOUT_LEN,        //     REPORT_COUNT (10)
//You can change the data length
0x91, 0x02, // OUTPUT (Data,Var,Abs)
```

Input Report Length:

Report ID: 1 byte

Dpad and Buttons: 4bits+12bits = 16bits = 2 bytes

X/Y/Z: 3 bytes

rX/rY/rZ: 3 bytes

Vendor: 10 bytes

Total: 19 bytes

Vendor ID in MCU

```
BleGamepadInstance->hid->pnp(0x02, 0x045e, 0x028e, 0x0002);
BleGamepadInstance->hid->hidInfo(0x00, 0x02);
```

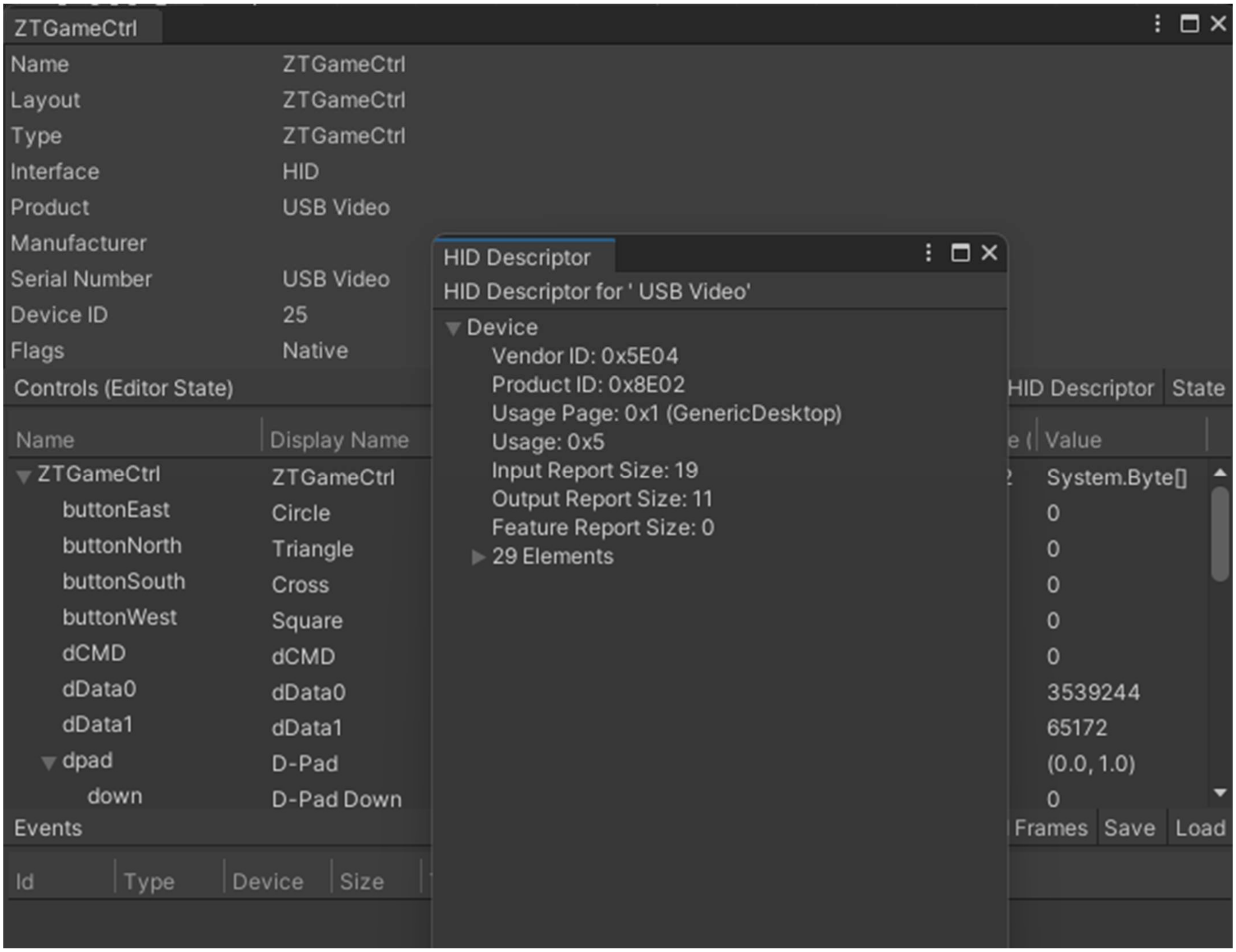
C# code Struct Layout in Unity

```
[StructLayout(LayoutKind.Explicit, Size = 19)]
[FieldOffset(0)] public byte reportId;
[FieldOffset(1)] public byte buttons1;
[FieldOffset(2)] public byte buttons2;
[FieldOffset(3)] public byte leftStickX;
[FieldOffset(4)] public byte leftStickY;
[FieldOffset(5)] public byte rightStickX;
[FieldOffset(6)] public byte rightStickY;
[FieldOffset(7)] public byte leftTrigger;
[FieldOffset(8)] public byte rightTrigger;

//Vendor defined data,length:1+1+4+4 = 10
[FieldOffset(9)] public byte Data0;
[FieldOffset(10)] public byte Data1;
[FieldOffset(11)] public UInt32 Data2;
[FieldOffset(15)] public UInt32 Data3;
//You can add your own data here
```

Input Device Matcher in Unity

```
    InputSystem.RegisterLayout<ZTGameCtrl>(
        matches: new InputDeviceMatcher()
        .WithInterface("HID")
        .WithCapability("vendorId", 0x5E04)
        .WithCapability("productId", 0x8E02));
```



Name	ZTGameCtrl
Layout	ZTGameCtrl
Type	ZTGameCtrl
Interface	HID
Product	USB Video
Manufacturer	
Serial Number	USB Video
Device ID	25
Flags	Native

Controls (Editor State)

Name	Display Name	Value
ZTGameCtrl	ZTGameCtrl	System.Byte[]
buttonEast	Circle	0
buttonNorth	Triangle	0
buttonSouth	Cross	0
buttonWest	Square	0
dCMD	dCMD	0
dData0	dData0	3539244
dData1	dData1	65172
dpad	D-Pad	(0.0, 1.0)
down	D-Pad Down	0

Events

Id	Type	Device	Size

HID Descriptor

HID Descriptor for ' USB Video'

- Device
  - Vendor ID: 0x5E04
  - Product ID: 0x8E02
  - Usage Page: 0x1 (GenericDesktop)
  - Usage: 0x5
  - Input Report Size: 19
  - Output Report Size: 11
  - Feature Report Size: 0
  - 29 Elements

Step by Step

1. Upload the code for MCU, the BLE device will work.
2. Open the Bluetooth on PC
3. Add the BLE device “BLE_Unity”
4. Open Unity IDE, 2019 or higher
5. Install the package “Input System”, restart IDE
6. Put the “ZTDevice” and “MyControlCommand” Script files into your Unity assets folder, anywhere is OK.
7. Unity: Window->Analysis->Input Debugger, find the “ZTGameCtrl”

How to add your own data

Input Data:

```
#define GP_VDATAIN_LEN      10
```

Change the input report length for MCU

```
[StructLayout(LayoutKind.Explicit, Size = 19)]
```

Change the input report length in Unity

```
[InputControl(name = "dCMD", layout = "Integer", format = "BYTE")]
[FieldOffset(9)] public byte Data0;
```

Add your control data name(“dCMD”), layout(“Integer”), format (“BYTE” or “UINT”). FieldOffset is the position of the data.

```
public IntegerControl dCMD { get; private set; }
dCMD = GetChildControl<IntegerControl>("dCMD");
```

Input Data USE:

```
int data = bleGamepad.dCMD.ReadValue();
```

Output Data:

```
#define  GP_VDATAOUT_LEN   10
```

Change the output report length for MCU

Code in Unity, MyControlCommand.cpp

```
internal const int id = 0x01;
internal const int kSize =
InputDeviceCommand.BaseCommandSize + 11;
```

Id(0x01) is the “#define GAMEPAD_ID 0x01” in the code for MCU

Command Size+11, the output report data length(include id)

```
[FieldOffset(InputDeviceCommand.BaseCommandSize+1)]
public byte dataID;
```

Add your data, fieldOffset is the position for the data.

```
public static MyControlCommand Create(
        byte indataID, byte indataFun, UInt32 indata1, UInt32 indata2)
```

Change the command create function input parameter

Output Data USE:

```
var command = MyControlCommand.Create(p0,p1,p2,p3);
bleGamepad.device.ExecuteCommand(ref command);
```