in: Controller, Gamepad, USB Charge, and 7 more

Sony DualSense

Sony DualSense

Type	Gamepad
Brand	Sony
Vendor ID	`0x054C`
Product ID	`0x0CE6`

Input

- 13× Digital Buttons
  - Face: △, ○, ×, □
  - Shoulder: L1, R1
  - Other: "PS", CREATE, OPTIONS, touchpad click, L3, R3, Mute
- Digital Directional Pad
- Touch Pad
  - 2× Point
  - Capacitive
- 2× Analog Sticks
- 2× Analog Triggers
  - L2, R2
- Motion
  - 3 Axis Accelerometer
  - 3 Axis Gyroscope
- Mono? Body Mic with Back Mic sound canceling
- Mono Microphone via TRRS jack

Output

- Rumble Emulation via LRA
  - Right: Light (Simulated)
  - Left: Heavy (Simulated)
- Haptic Feedback via LRA
  - Right
  - Left

- Indicator Lights (5, white, capable of fade)
- Mute Indicator (pulsing or solid amber)
- Mono Speaker
- Stereo Audio via TRRS jack

Ports

- USB-C ⚡ (Charge) 💻 (Data)
- 3.5mm Phone TRRS 🎤 (1 Channel) 🎧 (2 Channels)
- Proprietary EXT Port
  - data (suspected)
  - charging

Wireless

- Bluetooth 5.1

Contents

HID Interface

Report Summaries

[Collapse]

USB

ReportID		Size	Type	Note
0x01	1	63	Input	Get Controller State

0x02	2	47	Output	Set Controller State
0x05	5	40	Feature	Get Calibration
0x08	8	47	Feature	Get bluetooth control (please document)
0x09	9	19	Feature	Get Controller and Host MAC
0x0A	10	26	Feature	Set bluetooth pairing (please document)
0x20	32	63	Feature	Get Controller Version/Date (Firmware Info)
0x21	33	4	Feature	Set audio control (please document)
0x22	34	63	Feature	Get Hardware Info
0x80	128	63	Feature	Set test command (please document)
0x81	129	63	Feature	Get test result (please document)
0x82	130	9	Feature	Set calibration command (please document)
0x83	131	63	Feature	Get calibration data (please document)
0x84	132	63	Feature	Set individual data (please document)
0x85	133	2	Feature	Get individual data result (please document)
0xA0	160	1	Feature	Set DFU enable (please document)
0xE0	224	63	Feature	Get system profile (please document)
0xF0	240	63	Feature	Flash command (please document)
0xF1	241	63	Feature	Get flash cmd status (please document)
0xF2	242	15	Feature	
0xF4	244	63	Feature	User update command (please document)
0xF5	245	3	Feature	User get update status (please document)

Bluetooth

ReportID		Size	Type	Note
0x01	1	62	Input	Get Controller State (simplified)
0x31	49	77	Input	Get Controller State
0x31	49	77	Output	Set Controller State or Audio (Audio theoretical)
0x32	50	141	Output	Set Controller State and/or Audio (unconfirmed)

0x33	51	205	Output	Set Controller State and/or Audio (unconfirmed)
0x34	52	269	Output	Set Controller State and/or Audio (unconfirmed)
0x35	53	333	Output	Set Controller State and/or Audio (unconfirmed)
0x36	54	397	Output	Set Controller State and/or Audio (unconfirmed)
0x37	55	461	Output	Set Controller State and/or Audio (unconfirmed)
0x38	56	525	Output	Set Controller State and/or Audio (unconfirmed)
0x39	57	546	Output	Set Controller State and/or Audio (unconfirmed)
0x05	5	40	Feature	Get Calibration
0x08	8	47	Feature	
0x09	9	19	Feature	Get Controller and Host MAC
0x20	32	63	Feature	Get Controller Version/Date (Firmware Info)
0x22	34	63	Feature	Get Hardware Info
0x80	128	63	Feature	
0x81	129	63	Feature	
0x82	130	9	Feature	
0x83	131	63	Feature	
0xF0	240	63	Feature	
0xF1	241	63	Feature	
0xF2	242	15	Feature	

Descriptor

[Collapse]

USB

```
0x05, 0x01,         // Usage Page (Generic Desktop
Ctrls)
0x09, 0x05,         // Usage (Game Pad)
0xA1, 0x01,         // Collection (Application)
0x85, 0x01,         //   Report ID (1)
0x09, 0x30,         //   Usage (X)
0x09, 0x31,         //   Usage (Y)
0x09, 0x32,         //   Usage (Z)
0x09, 0x35,         //   Usage (Rz)
0x09, 0x33,         //   Usage (Rx)
0x09, 0x34,         //   Usage (Ry)
0x15, 0x00,         //   Logical Minimum (0)
0x26, 0xFF, 0x00,   //   Logical Maximum (255)
0x75, 0x08,         //   Report Size (8)
0x95, 0x06,         //   Report Count (6)
0x81, 0x02,         //   Input (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position)
0x06, 0x00, 0xFF,   //   Usage Page (Vendor Defined
0xFF00)
0x09, 0x20,         //   Usage (0x20)
0x95, 0x01,         //   Report Count (1)
0x81, 0x02,         //   Input (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position)
0x05, 0x01,         //   Usage Page (Generic Desktop
Ctrls)
0x09, 0x39,         //   Usage (Hat switch)
0x15, 0x00,         //   Logical Minimum (0)
0x25, 0x07,         //   Logical Maximum (7)
0x35, 0x00,         //   Physical Minimum (0)
0x46, 0x3B, 0x01,   //   Physical Maximum (315)
0x65, 0x14,         //   Unit (System: English
Rotation, Length: Centimeter)
0x75, 0x04,         //   Report Size (4)
```

```
0x65, 0x00,        //   Unit (None)
0x05, 0x09,        //   Usage Page (Button)
0x19, 0x01,        //   Usage Minimum (0x01)
0x29, 0x0F,        //   Usage Maximum (0x0F)
0x15, 0x00,        //   Logical Minimum (0)
0x25, 0x01,        //   Logical Maximum (1)
0x75, 0x01,        //   Report Size (1)
0x95, 0x0F,        //   Report Count (15)
0x81, 0x02,        //   Input (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position)
0x06, 0x00, 0xFF,  //   Usage Page (Vendor Defined 0xFF00)
0x09, 0x21,        //   Usage (0x21)
0x95, 0x0D,        //   Report Count (13)
0x81, 0x02,        //   Input (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position)
0x06, 0x00, 0xFF,  //   Usage Page (Vendor Defined 0xFF00)
0x09, 0x22,        //   Usage (0x22)
0x15, 0x00,        //   Logical Minimum (0)
0x26, 0xFF, 0x00,  //   Logical Maximum (255)
0x75, 0x08,        //   Report Size (8)
0x95, 0x34,        //   Report Count (52)
0x81, 0x02,        //   Input (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position)
0x85, 0x02,        //   Report ID (2)
0x09, 0x23,        //   Usage (0x23)
0x95, 0x2F,        //   Report Count (47)
0x91, 0x02,        //   Output (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x05,        //   Report ID (5)
0x09, 0x33,        //   Usage (0x33)
0x95, 0x28,        //   Report Count (40)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x08,        //   Report ID (8)
0x09, 0x34,        //   Usage (0x34)
0x95, 0x2F,        //   Report Count (47)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x09,        //   Report ID (9)
0x09, 0x24,        //   Usage (0x24)
0x95, 0x13,        //   Report Count (19)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x0A,        //   Report ID (10)
0x09, 0x25,        //   Usage (0x25)
0x95, 0x1A,        //   Report Count (26)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x20,        //   Report ID (32)
0x09, 0x26,        //   Usage (0x26)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x21,        //   Report ID (33)
0x09, 0x27,        //   Usage (0x27)
0x95, 0x04,        //   Report Count (4)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x22,        //   Report ID (34)
0x09, 0x40,        //   Usage (0x40)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x80,        //   Report ID (128)
0x09, 0x28,        //   Usage (0x28)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x81,        //   Report ID (129)
0x09, 0x29,        //   Usage (0x29)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x82,        //   Report ID (130)
0x09, 0x2A,        //   Usage (0x2A)
0x95, 0x09,        //   Report Count (9)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x83,        //   Report ID (131)
0x09, 0x2B,        //   Usage (0x2B)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x84,        //   Report ID (132)
0x09, 0x2C,        //   Usage (0x2C)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0x85,        //   Report ID (133)
0x09, 0x2D,        //   Usage (0x2D)
0x95, 0x02,        //   Report Count (2)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
```

```
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0xE0,        //   Report ID (224)
0x09, 0x2F,        //   Usage (0x2F)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0xF0,        //   Report ID (240)
0x09, 0x30,        //   Usage (0x30)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0xF1,        //   Report ID (241)
0x09, 0x31,        //   Usage (0x31)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0xF2,        //   Report ID (242)
0x09, 0x32,        //   Usage (0x32)
0x95, 0x0F,        //   Report Count (15)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0xF4,        //   Report ID (244)
0x09, 0x35,        //   Usage (0x35)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0x85, 0xF5,        //   Report ID (245)
0x09, 0x36,        //   Usage (0x36)
0x95, 0x03,        //   Report Count (3)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
0xC0,              // End Collection

// 273 bytes
```

BT

```
0x05, 0x01,        // Usage Page (Generic Desktop Ctrls)
0x09, 0x05,        // Usage (Game Pad)
0xA1, 0x01,        // Collection (Application)
0x85, 0x01,        //   Report ID (1)
0x09, 0x30,        //   Usage (X)
0x09, 0x31,        //   Usage (Y)
0x09, 0x32,        //   Usage (Z)
0x09, 0x35,        //   Usage (Rz)
0x15, 0x00,        //   Logical Minimum (0)
0x26, 0xFF, 0x00,  //   Logical Maximum (255)
0x75, 0x08,        //   Report Size (8)
0x95, 0x04,        //   Report Count (4)
0x81, 0x02,        //   Input (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position)
0x09, 0x39,        //   Usage (Hat switch)
0x15, 0x00,        //   Logical Minimum (0)
0x25, 0x07,        //   Logical Maximum (7)
0x35, 0x00,        //   Physical Minimum (0)
0x46, 0x3B, 0x01,  //   Physical Maximum (315)
0x65, 0x14,        //   Unit (System: English Rotation, Length: Centimeter)
0x75, 0x04,        //   Report Size (4)
0x95, 0x01,        //   Report Count (1)
0x81, 0x42,        //   Input (Data,Var,Abs,No Wrap,Linear,Preferred State,Null State)
0x65, 0x00,        //   Unit (None)
0x05, 0x09,        //   Usage Page (Button)
0x19, 0x01,        //   Usage Minimum (0x01)
0x29, 0x0E,        //   Usage Maximum (0x0E)
0x15, 0x00,        //   Logical Minimum (0)
0x25, 0x01,        //   Logical Maximum (1)
0x75, 0x01,        //   Report Size (1)
0x95, 0x0E,        //   Report Count (14)
0x81, 0x02,        //   Input (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position)
0x75, 0x06,        //   Report Size (6)
0x95, 0x01,        //   Report Count (1)
0x81, 0x01,        //   Input (Const,Array,Abs,No Wrap,Linear,Preferred State,No Null Position)
0x05, 0x01,        //   Usage Page (Generic Desktop Ctrls)
0x09, 0x33,        //   Usage (Rx)
0x09, 0x34,        //   Usage (Ry)
0x15, 0x00,        //   Logical Minimum (0)
0x26, 0xFF, 0x00,  //   Logical Maximum (255)
0x75, 0x08,        //   Report Size (8)
0x95, 0x02,        //   Report Count (2)
0x81, 0x02,        //   Input (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position)
0x06, 0x00, 0xFF,  //   Usage Page (Vendor Defined 0xFF00)
0x15, 0x00,        //   Logical Minimum (0)
0x26, 0xFF, 0x00,  //   Logical Maximum (255)
0x75, 0x08,        //   Report Size (8)
0x95, 0x4D,        //   Report Count (77)
0x85, 0x31,        //   Report ID (49)
0x09, 0x31,        //   Usage (0x31)
0x91, 0x02,        //   Output (Data,Var,Abs,No Wrap,Linear,Preferred State,No Null Position,Non-volatile)
```

```
0x85, 0x32,        //   Report ID (50)
0x09, 0x32,        //   Usage (0x32)
0x95, 0x8D,        //   Report Count (141)
0x91, 0x02,        //   Output (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x33,        //   Report ID (51)
0x09, 0x33,        //   Usage (0x33)
0x95, 0xCD,        //   Report Count (205)
0x91, 0x02,        //   Output (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x34,        //   Report ID (52)
0x09, 0x34,        //   Usage (0x34)
0x96, 0x0D, 0x01,  //   Report Count (269)
0x91, 0x02,        //   Output (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x35,        //   Report ID (53)
0x09, 0x35,        //   Usage (0x35)
0x96, 0x4D, 0x01,  //   Report Count (333)
0x91, 0x02,        //   Output (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x36,        //   Report ID (54)
0x09, 0x36,        //   Usage (0x36)
0x96, 0x8D, 0x01,  //   Report Count (397)
0x91, 0x02,        //   Output (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x37,        //   Report ID (55)
0x09, 0x37,        //   Usage (0x37)
0x96, 0xCD, 0x01,  //   Report Count (461)
0x91, 0x02,        //   Output (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x38,        //   Report ID (56)
0x09, 0x38,        //   Usage (0x38)
0x96, 0x0D, 0x02,  //   Report Count (525)
0x91, 0x02,        //   Output (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x39,        //   Report ID (57)
0x09, 0x39,        //   Usage (0x39)
0x96, 0x22, 0x02,  //   Report Count (546)
0x91, 0x02,        //   Output (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x06, 0x80, 0xFF,  //   Usage Page (Vendor Defined
0xFF80)
0x85, 0x05,        //   Report ID (5)
0x09, 0x33,        //   Usage (0x33)
0x95, 0x28,        //   Report Count (40)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x08,        //   Report ID (8)
0x09, 0x34,        //   Usage (0x34)
0x95, 0x2F,        //   Report Count (47)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x09,        //   Report ID (9)
0x09, 0x24,        //   Usage (0x24)
0x95, 0x13,        //   Report Count (19)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x20,        //   Report ID (32)
0x09, 0x26,        //   Usage (0x26)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x22,        //   Report ID (34)
0x09, 0x40,        //   Usage (0x40)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x80,        //   Report ID (128)
0x09, 0x28,        //   Usage (0x28)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x81,        //   Report ID (129)
0x09, 0x29,        //   Usage (0x29)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x82,        //   Report ID (130)
0x09, 0x2A,        //   Usage (0x2A)
0x95, 0x09,        //   Report Count (9)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0x83,        //   Report ID (131)
0x09, 0x2B,        //   Usage (0x2B)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0xF1,        //   Report ID (241)
0x09, 0x31,        //   Usage (0x31)
0x95, 0x3F,        //   Report Count (63)
0xB1, 0x02,        //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
```

```
0x95, 0x0F,         //   Report Count (15)
0xB1, 0x02,         //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0x85, 0xF0,         //   Report ID (240)
0x09, 0x30,         //   Usage (0x30)
0x95, 0x3F,         //   Report Count (63)
0xB1, 0x02,         //   Feature (Data,Var,Abs,No
Wrap,Linear,Preferred State,No Null Position,Non-
volatile)
0xC0,               // End Collection

// 279 bytes
```

Data Structures

[Collapse]

Common Structures

```
// C++11 allows setting enum base type. If enum sizes
cannot be assured please use the indicated base types
instead of the enum types.
enum Direction : uint8_t {
    North = 0,
    NorthEast,
    East,
    SouthEast,
    South,
    SouthWest,
    West,
    NorthWest,
    None = 8
};

enum PowerState : uint8_t {
    Discharging         = 0x00, // Use PowerPercent
    Charging            = 0x01, // Use PowerPercent
    Complete            = 0x02, // PowerPercent not
valid? assume 100%?
    AbnormalVoltage     = 0x0A, // PowerPercent not
valid?
    AbnormalTemperature = 0x0B, // PowerPercent not
valid?
    ChargingError       = 0x0F  // PowerPercent not
valid?
};

enum MuteLight : uint8_t {
    Off = 0,
    On,
    Breathing,
    DoNothing, // literally nothing, this input is
ignored,
               // though it might be a faster blink
in other versions
    NoAction4,
    NoAction5,
    NoAction6,
    NoAction7= 7
};

enum LightBrightness : uint8_t {
    Bright = 0,
    Mid,
    Dim,
    NoAction3,
    NoAction4,
    NoAction5,
    NoAction6,
    NoAction7= 7
};

enum LightFadeAnimation : uint8_t {
    Nothing = 0,
    FadeIn, // from black to blue
    FadeOut // from blue to black
};

template<int N> struct BTCRC {
    uint8_t[N-4] Buff;
    uint32_t CRC;
};
```

FFB Trigger Effect Factories

Code hidden due to length: expand to view code in place. See:[Expand] Github Gist for latest version of this code (https://gist.github.com/Nielk1/6d54cc2c00d2201ccb8c2720ad7538db).

Input Reports

```
struct TouchFingerData { // 4
/*0.0*/ uint32_t Index : 7;
/*0.7*/ uint32_t NotTouching : 1;
/*1.0*/ uint32_t FingerX : 12;
/*2.4*/ uint32_t FingerY : 12;
};
```

```
/*8*/ uint8_t Timestamp;
};

struct BTSimpleGetStateData { // 9
/*0  */ uint8_t LeftStickX;
/*1  */ uint8_t LeftStickY;
/*2  */ uint8_t RightStickX;
/*3  */ uint8_t RightStickY;
/*4.0*/ Direction DPad : 4;
/*4.4*/ uint8_t ButtonSquare : 1;
/*4.5*/ uint8_t ButtonCross : 1;
/*4.6*/ uint8_t ButtonCircle : 1;
/*4.7*/ uint8_t ButtonTriangle : 1;
/*5.0*/ uint8_t ButtonL1 : 1;
/*5.1*/ uint8_t ButtonR1 : 1;
/*5.2*/ uint8_t ButtonL2 : 1;
/*5.3*/ uint8_t ButtonR2 : 1;
/*5.4*/ uint8_t ButtonShare : 1;
/*5.5*/ uint8_t ButtonOptions : 1;
/*5.6*/ uint8_t ButtonL3 : 1;
/*5.7*/ uint8_t ButtonR3 : 1;
/*6.1*/ uint8_t ButtonHome : 1;
/*6.2*/ uint8_t ButtonPad : 1;
/*6.3*/ uint8_t Counter : 6;
/*7  */ uint8_t TriggerLeft;
/*8  */ uint8_t TriggerRight;
// anything beyond this point, if set, is invalid
junk data that was not cleared
};

struct USBGetStateData { // 63
/* 0  */ uint8_t LeftStickX;
/* 1  */ uint8_t LeftStickY;
/* 2  */ uint8_t RightStickX;
/* 3  */ uint8_t RightStickY;
/* 4  */ uint8_t TriggerLeft;
/* 5  */ uint8_t TriggerRight;
/* 6  */ uint8_t SeqNo; // always 0x01 on BT
/* 7.0*/ Direction DPad : 4;
/* 7.4*/ uint8_t ButtonSquare : 1;
/* 7.5*/ uint8_t ButtonCross : 1;
/* 7.6*/ uint8_t ButtonCircle : 1;
/* 7.7*/ uint8_t ButtonTriangle : 1;
/* 8.0*/ uint8_t ButtonL1 : 1;
/* 8.1*/ uint8_t ButtonR1 : 1;
/* 8.2*/ uint8_t ButtonL2 : 1;
/* 8.3*/ uint8_t ButtonR2 : 1;
/* 8.4*/ uint8_t ButtonCreate : 1;
/* 8.5*/ uint8_t ButtonOptions : 1;
/* 8.6*/ uint8_t ButtonL3 : 1;
/* 8.7*/ uint8_t ButtonR3 : 1;
/* 9.0*/ uint8_t ButtonHome : 1;
/* 9.1*/ uint8_t ButtonPad : 1;
/* 9.2*/ uint8_t ButtonMute : 1;
/* 9.3*/ uint8_t ButtonLeftFunction : 1; // DualSense
Edge
/* 9.4*/ uint8_t ButtonRightFunction : 1; //
DualSense Edge
/* 9.5*/ uint8_t ButtonLeftPaddle : 1; // DualSense
Edge
/* 9.6*/ uint8_t ButtonRightPaddle : 1; // DualSense
Edge
/* 9.7*/ uint8_t UNK1 : 1; // appears unused
/*10  */ uint8_t UNK2; // appears unused
/*11  */ uint32_t UNK_COUNTER; // Linux driver calls
this reserved, tools leak calls the 2 high bytes
"random"
/*15  */ int16_t AngularVelocityX;
/*17  */ int16_t AngularVelocityZ;
/*19  */ int16_t AngularVelocityY;
/*21  */ int16_t AccelerometerX;
/*23  */ int16_t AccelerometerY;
/*25  */ int16_t AccelerometerZ;
/*27  */ uint32_t SensorTimestamp;
/*31  */ int8_t Temperature; // reserved2 in Linux
driver
/*32  */ TouchData TouchData;
/*41.0*/ uint8_t TriggerRightStopLocation: 4; //
trigger stop can be a range from 0 to 9 (F/9.0 for
Apple interface)
/*41.4*/ uint8_t TriggerRightStatus: 4;
/*42.0*/ uint8_t TriggerLeftStopLocation: 4;
/*42.4*/ uint8_t TriggerLeftStatus: 4; // 0
feedbackNoLoad
                                       // 1
feedbackLoadApplied
                                       // 0
weaponReady
                                       // 1
weaponFiring
                                       // 2
weaponFired
                                       // 0
vibrationNotVibrating
                                       // 1
vibrationIsVibrating
/*43  */ uint32_t HostTimestamp; // mirrors data from
report write
/*47.0*/ uint8_t TriggerRightEffect: 4; // Active
trigger effect, previously we thought this was status
max
/*47.4*/ uint8_t TriggerLeftEffect: 4;  // 0 for
reset and all other effects
                                        // 1 for
feedback effect
                                        // 2 for
weapon effect
                                        // 3 for
vibration
```

```
/*53.0*/ uint8_t PluggedHeadphones : 1;
/*53.1*/ uint8_t PluggedMic : 1;
/*53.2*/ uint8_t MicMuted: 1; // Mic muted by
powersave/mute command
/*53.3*/ uint8_t PluggedUsbData : 1;
/*53.4*/ uint8_t PluggedUsbPower : 1;
/*53.5*/ uint8_t PluggedUnk1 : 3;
/*54.0*/ uint8_t PluggedExternalMic : 1; // Is
external mic active (automatic in mic auto mode)
/*54.1*/ uint8_t HapticLowPassFilter : 1; // Is the
Haptic Low-Pass-Filter active?
/*54.2*/ uint8_t PluggedUnk3 : 6;
/*55  */ uint8_t[8] AesCmac;
};

struct BTGetStateData { // 77
/* 0*/ USBGetStateData StateData;
/*63*/ uint8_t UNK1; // Oscillates between 00101100
and 00101101 when rumbling
                     // Not affected by rumble volume
or enhanced vs normal rumble
                     // Audio rumble not yet tested
as this is only on BT
/*64*/ uint8_t BtCrcFailCount;
/*65*/ uint8_t[11] Pad;
};
```

HID Report 0x01 Input USB

```
struct ReportIn01USB {
    uint8_t ReportID; // 0x01
    USBGetStateData State;
};
```

HID Report 0x01 Input BT

```
struct ReportIn01BT {
    uint8_t ReportID; // 0x01
    BTSimpleGetStateData State;
};
```

HID Report 0x31 Input BT

```
struct ReportIn31 {
    union {
        BTCRC<78> CRC;
        struct {
            uint8_t ReportID; // 0x31
            uint8_t HasHID : 1; // Present for
packets with state data
            uint8_t HasMic : 1; // Looks mutually
exclusive, possible mic data
            uint8_t Unk1 : 2;
            uint8_t SeqNo : 4; // unclear progression
            BTGetStateData State;
        } Data;
    }
};
```

Output Reports

```
struct SetStateData { // 47
/*    */ // Report Set Flags
/*    */ // These flags are used to indicate what
contents from this report should be processed
/* 0.0*/ uint8_t EnableRumbleEmulation: 1; // Suggest
halving rumble strength
/* 0.1*/ uint8_t UseRumbleNotHaptics:   1; //
/*    */
/* 0.2*/ uint8_t AllowRightTriggerFFB: 1; // Enable
setting RightTriggerFFB
/* 0.3*/ uint8_t AllowLeftTriggerFFB: 1;  // Enable
setting LeftTriggerFFB
/*    */
/* 0.4*/ uint8_t AllowHeadphoneVolume: 1; // Enable
setting VolumeHeadphones
/* 0.5*/ uint8_t AllowSpeakerVolume: 1;   // Enable
setting VolumeSpeaker
/* 0.6*/ uint8_t AllowMicVolume: 1;       // Enable
setting VolumeMic
/*    */
/* 0.7*/ uint8_t AllowAudioControl: 1; // Enable
setting AudioControl section
/* 1.0*/ uint8_t AllowMuteLight: 1;    // Enable
setting MuteLightMode
/* 1.1*/ uint8_t AllowAudioMute: 1;    // Enable
setting MuteControl section
/*    */
/* 1.2*/ uint8_t AllowLedColor: 1; // Enable RGB LED
section
/*    */
/* 1.3*/ uint8_t ResetLights: 1; // Release the LEDs
from Wireless firmware control
/*    */                         // When in wireless
mode this must be signaled to control LEDs
/*    */                         // This cannot be
applied during the BT pair animation.
/*    */                         // SDL2 waits until
the SensorTimestamp value is >= 10200000
```

```
/* 1.4*/ uint8_t AllowPlayerIndicators: 1; // Enable
setting PlayerIndicators section
/* 1.5*/ uint8_t AllowHapticLowPassFilter: 1; //
Enable HapticLowPassFilter
/* 1.6*/ uint8_t AllowMotorPowerLevel: 1; //
MotorPowerLevel reductions for trigger/haptic
/* 1.7*/ uint8_t AllowAudioControl2: 1; // Enable
setting AudioControl2 section
/*     */
/* 2   */ uint8_t RumbleEmulationRight; // emulates
the light weight
/* 3   */ uint8_t RumbleEmulationLeft; // emulated the
heavy weight
/*     */
/* 4   */ uint8_t VolumeHeadphones; // max 0x7f
/* 5   */ uint8_t VolumeSpeaker; // PS5 appears to
only use the range 0x3d-0x64
/* 6   */ uint8_t VolumeMic; // not linier, seems to
max at 64, 0 is not fully muted
/*     */
/*     */ // AudioControl
/* 7.0*/ uint8_t MicSelect: 2; // 0 Auto
/*     */                       // 1 Internal Only
/*     */                       // 2 External Only
/*     */                       // 3 Unclear, sets
external mic flag but might use internal mic, do test
/* 7.2*/ uint8_t EchoCancelEnable: 1;
/* 7.3*/ uint8_t NoiseCancelEnable: 1;
/* 7.4*/ uint8_t OutputPathSelect: 2; // 0 L_R_X
/*     */                            // 1 L_L_X
/*     */                            // 2 L_L_R
/*     */                            // 3 X_X_R
/* 7.6*/ uint8_t InputPathSelect: 2;  // 0 CHAT_ASR
/*     */                            // 1 CHAT_CHAT
/*     */                            // 2 ASR_ASR
/*     */                            // 3 Does
Nothing, invalid
/*     */
/* 8   */ MuteLight MuteLightMode;
/*     */
/*     */ // MuteControl
/* 9.0*/ uint8_t TouchPowerSave: 1;
/* 9.1*/ uint8_t MotionPowerSave: 1;
/* 9.2*/ uint8_t HapticPowerSave: 1; // AKA
BulletPowerSave
/* 9.3*/ uint8_t AudioPowerSave: 1;
/* 9.4*/ uint8_t MicMute: 1;
/* 9.5*/ uint8_t SpeakerMute: 1;
/* 9.6*/ uint8_t HeadphoneMute: 1;
/* 9.7*/ uint8_t HapticMute: 1; // AKA BulletMute
/*     */
/*10   */ uint8_t[11] RightTriggerFFB;
/*21   */ uint8_t[11] LeftTriggerFFB;
/*32   */ uint32_t HostTimestamp; // mirrored into
report read
/*     */
/*     */ // MotorPowerLevel
/*36.0*/ uint8_t TriggerMotorPowerReduction : 4; //
0x0-0x7 (no 0x8?) Applied in 12.5% reductions
/*36.4*/ uint8_t RumbleMotorPowerReduction : 4;  //
0x0-0x7 (no 0x8?) Applied in 12.5% reductions
/*     */
/*     */ // AudioControl2
/*37.0*/ uint8_t SpeakerCompPreGain: 3; // additional
speaker volume boost
/*37.3*/ uint8_t BeamformingEnable: 1; // Probably
for MIC given there's 2, might be more bits, can't
find what it does
/*37.4*/ uint8_t UnkAudioControl2: 4; // some of
these bits might apply to the above
/*     */
/*38.0*/ uint8_t AllowLightBrightnessChange: 1; //
LED_BRIHTNESS_CONTROL
/*38.1*/ uint8_t AllowColorLightFadeAnimation: 1; //
LIGHTBAR_SETUP_CONTROL
/*38.2*/ uint8_t EnableImprovedRumbleEmulation: 1; //
Use instead of EnableRumbleEmulation
                                                    //
requires FW >= 0x0224
                                                    //
No need to halve rumble strength
/*38.3*/ uint8_t UNKBITC: 5; // unused
/*     */
/*39.0*/ uint8_t HapticLowPassFilter: 1;
/*39.1*/ uint8_t UNKBIT: 7;
/*     */
/*40   */ uint8_t UNKBYTE; // previous notes suggested
this was HLPF, was probably off by 1
/*     */
/*41   */ LightFadeAnimation LightFadeAnimation;
/*42   */ LightBrightness LightBrightness;
/*     */
/*     */ // PlayerIndicators
/*     */ // These bits control the white LEDs under
the touch pad.
/*     */ // Note the reduction in functionality for
later revisions.
/*     */ // Generation 0x03 - Full Functionality
/*     */ // Generation 0x04 - Mirrored Only
/*     */ // Suggested detection: (HardwareInfo &
0x00FFFF00) == 0X00000400
/*     */ //
/*     */ // Layout used by PS5:
/*     */ // 0x04 - -x- -  Player 1
/*     */ // 0x06 - x-x -  Player 2
/*     */ // 0x15 x -x- x  Player 3
/*     */ // 0x1B x x-x x  Player 4
/*     */ // 0x1F x xxx x  Player 5* (Unconfirmed)
/*     */ //
```

```
- x
/*43.1*/ uint8_t PlayerLight2 : 1; // - x-- - // - x-
x -
/*43.2*/ uint8_t PlayerLight3 : 1; // - -x- - // - -
x- -
/*43.3*/ uint8_t PlayerLight4 : 1; // - --x - // - x-
x -
/*43.4*/ uint8_t PlayerLight5 : 1; // - --- x // x --
- x
/*43.5*/ uint8_t PlayerLightFade: 1; // if low player
lights fade in, if high player lights instantly
change
/*43.6*/ uint8_t PlayerLightUNK : 2;
/*    */
/*    */ // RGB LED
/*44  */ uint8_t LedRed;
/*45  */ uint8_t LedGreen
/*46  */ uint8_t LedBlue;
// Structure ends here though on BT there is padding
and a CRC, see ReportOut31
};
```

HID Report 0x02 Output USB

```
struct ReportOut02 {
    uint8_t ReportID; // 0x02
    USBSetStateData State;
};
```

HID Report 0x31 Output BT

```
struct ReportOut31 {
    union {
        BTCRC<78> CRC;
        struct {
            uint8_t ReportID; // 0x31
            uint8_t UNK1 : 1;      // -+
            uint8_t EnableHID : 1; //  | - these 3
bits seem to act as an enum
            uint8_t UNK2 : 1;      // -+
            uint8_t UNK3 : 1;
            uint8_t SeqNo : 4; // increment for every
write // we have no proof of this, need to see some
PS5 captures
            SetStateData State;
        } Data;
    }
};
```

Feature Reports

Calibration

Reading calibration is required to switch BT input reports from the truncated 0x01 report to the expanded 0x31 report.

Linux hid-sony.c (does this apply to the DualSense?)

```
/* Set gyroscope calibration and normalization
parameters.
 * Data values will be normalized to
1/DS4_GYRO_RES_PER_DEG_S degree/s.
 */
speed_2x = (gyro_speed_plus + gyro_speed_minus);
sc->ds4_calib_data[0].abs_code = ABS_RX;
sc->ds4_calib_data[0].bias = gyro_pitch_bias;
sc->ds4_calib_data[0].sens_numer =
speed_2x*DS4_GYRO_RES_PER_DEG_S;
sc->ds4_calib_data[0].sens_denom = gyro_pitch_plus -
gyro_pitch_minus;

sc->ds4_calib_data[1].abs_code = ABS_RY;
sc->ds4_calib_data[1].bias = gyro_yaw_bias;
sc->ds4_calib_data[1].sens_numer =
speed_2x*DS4_GYRO_RES_PER_DEG_S;
sc->ds4_calib_data[1].sens_denom = gyro_yaw_plus -
gyro_yaw_minus;

sc->ds4_calib_data[2].abs_code = ABS_RZ;
sc->ds4_calib_data[2].bias = gyro_roll_bias;
sc->ds4_calib_data[2].sens_numer =
speed_2x*DS4_GYRO_RES_PER_DEG_S;
sc->ds4_calib_data[2].sens_denom = gyro_roll_plus -
gyro_roll_minus;

/* Set accelerometer calibration and normalization
parameters.
 * Data values will be normalized to
1/DS4_ACC_RES_PER_G G.
 */
range_2g = acc_x_plus - acc_x_minus;
sc->ds4_calib_data[3].abs_code = ABS_X;
sc->ds4_calib_data[3].bias = acc_x_plus - range_2g /
2;
sc->ds4_calib_data[3].sens_numer =
2*DS4_ACC_RES_PER_G;
sc->ds4_calib_data[3].sens_denom = range_2g;
```

```
sc->ds4_calib_data[4].bias = acc_y_plus - range_2g /
2;
sc->ds4_calib_data[4].sens_numer =
2*DS4_ACC_RES_PER_G;
sc->ds4_calib_data[4].sens_denom = range_2g;

range_2g = acc_z_plus - acc_z_minus;
sc->ds4_calib_data[5].abs_code = ABS_Z;
sc->ds4_calib_data[5].bias = acc_z_plus - range_2g /
2;
sc->ds4_calib_data[5].sens_numer =
2*DS4_ACC_RES_PER_G;
sc->ds4_calib_data[5].sens_denom = range_2g;
```

Bluetooth 0x05

```
struct ReportFeatureInCalibrateBT {
    union {
        BTCRC<41> CRC;
        struct {
            uint8_t ReportID; // 0x05 // does this
exist on USB? confirm
            int16_t GyroPitchBias;
            int16_t GyroYawBias;
            int16_t GyroRollBias;
            int16_t GyroPitchPlus;
            int16_t GyroPitchMinus;
            int16_t GyroYawPlus;
            int16_t GyroYawMinus;
            int16_t GyroRollPlus;
            int16_t GyroRollMinus;
            int16_t GyroSpeedPlus;
            int16_t GyroSpeedMinus;
            int16_t AccelXPlus;
            int16_t AccelXMinus;
            int16_t AccelYPlus;
            int16_t AccelYMinus;
            int16_t AccelZPlus;
            int16_t AccelZMinus;
            int16_t Unknown;
        } Data;
    }
};
```

MAC

Need to confirm these on BT

Get All MAC USB

```
struct ReportFeatureInMacAll {
    uint8_t ReportID; // 0x09
    uint8_t[6] ClientMac; // Right to Left
    uint8_t Hard08;
    uint8_t Hard25;
    uint8_t Hard00;
    uint8_t[6] HostMac; // Right to Left
    uint8_t[3] Pad; // Size according to Linux driver
};
```

Date and Version

Date/Version 0x20 USB/BT

```
struct ReportFeatureInVersion {
    union {
        BTCRC<64> CRC;
        struct {
            uint8_t ReportID; // 0x20
            char[11] BuildDate; // string
            char[8] BuildTime; // string
            uint16_t FwType;
            uint16_t SwSeries;
            uint32_t HardwareInfo; // 0x00FF0000 -
Variation
                                   // 0x0000FF00 -
Generation
                                   // 0x0000003F -
Trial?
                                   // ^ Values tied
to enumerations
            uint32_t FirmwareVersion; // 0xAABBCCCC
AA.BB.CCCC
            char[12] DeviceInfo;
            ////
            uint16_t UpdateVersion;
            char UpdateImageInfo;
            char UpdateUnk;
            ////
            uint32_t FwVersion1; // AKA SblFwVersion
                                 // 0xAABBCCCC
AA.BB.CCCC
                                 // Ignored for
FwType 0
                                 // HardwareVersion
used for FwType 1
                                 // Unknown behavior
if HardwareVersion < 0.1.38 for FwType 2 & 3
                                 // If
HardwareVersion >= 0.1.38 for FwType 2 & 3
            uint32_t FwVersion2; // AKA
SpiderDspFwVersion AKA BettyFwVer
                                    // May be Memory
Control Unit for Non Volatile Storage
          }
     }
};
```

Official Interface

The official interface for the DualSense controller is not public and likely confidential. This static library is likely a descendant of the original libPad library used for the DualShock 4 controller.

Limitations

- Three trigger effects (and reset)
- USB only, no Bluetooth support

Likely Interface

Apple's `GCDualSenseAdaptiveTrigger` interface reveals some aspects of Sony's API. We can assume the following:

- The controller is handled through an instance, be it class or indexed static, which stores at least some of the state data written to the controller. This can be determined because while there is no way to read the full parameters of the current trigger effect back from the controller the API somehow knows not to send the same parameters to the controller again when provided with the same input.
- The names of effects can be determined as follows:
  - Valid effects with their custom construct enumeration values that match `Trigger*Effect` from the input:
    - off = 0
    - feedback = 1
    - weapon = 2
    - vibration = 3
  - Valid trigger statuses are derived from the combination of `Trigger*Status` and `Trigger*Status` that do not match `Trigger*Status` from the input report:
    - unknown = -1, adaptive trigger status cannot be determined.
    - feedbackNoLoad = 0, adaptive trigger is in feedback mode, and a resistive load has not been applied yet.
    - feedbackLoadApplied = 1, adaptive trigger is in feedback mode, and a resistive load is applied.
    - weaponReady = 2, adaptive trigger is in weapon mode, the trigger is ready to fire, and a resistive load has not been applied yet.
    - weaponFiring = 3, adaptive trigger is in weapon mode, the trigger is firing, and a resistive load is currently being applied.
    - weaponFired = 4, adaptive trigger is in weapon mode, the trigger has fired, and a resistive load is no longer being applied.
    - vibrationNotVibrating = 5, adaptive trigger is in vibration mode, and the trigger is not vibrating.
    - vibrationIsVibrating = 6, adaptive trigger is in vibration

FANDOM
FAN CENTRAL
BETA
GAMES
ANIME
MOVIES
TV
VIDEO
WIKIS
START A WIKI

Hardware Revisions

There are at this point at least 3 hardware revisions that have been seen in the wild. The first is the original hardware, or Mass Production model, and the second and third are Revision 1 models. When processing the `ReportFeatureInVersion` structure, note the HardwareInfo's Generation and Trial values.

Generation 0x03, Trial 0x13

This hardware revision operates as expected.

Generation 0x04, Trial 0x13/0x14

This hardware revision has made a 'breaking change' to the player LED handling. Now the two pairs PlayerLight1+PlayerLight5 and PlayerLight2+PlayerLight4 are "wired" together. This change makes it impossible to separately control these LEDs but does not disrupt the intended player light configurations. This change was likely made to allow the use of 3 larger LEDs with light guides which may even have a lower part cost instead of 5 small ones. If interfacing with a controller of this hardware revision, note that only symmetrical player LED configurations are possible.

Categories

No Comments Yet

Be the first to comment on Sony DualSense!

Fan Feed

More Game Controller Collective Wiki

1 Sony DualSense

2 Wireless Joystick for P4 Model No 8951

3 Sony DualShock 4

Wii
Nintendo

AfterGlow Pro Controller
Nintendo

FANDOM

FAN CENTRAL
BETA

GAMES

ANIME

MOVIES

TV

VIDEO

WIKIS

START A WIKI

Lab

Amiibo
Nintendo

The Guy!
Super Meat Boy Wiki

Joy-Con Controller
Nintendo

Xbox 360 controller
Xbox Wiki

Xbox Wiki
Xbox Wiki

Steam
SteamWiki

Fandom

EXPLORE PROPERTIES

Fandom

Fanatical

Muthead

FOLLOW US

OVERVIEW

What is Fandom?

Terms of Use

About

Privacy Policy

Careers

Global Sitemap

Press

Local Sitemap

Contact

COMMUNITY

Community Central

Help

Support

ADVERTISE

Media Kit

Contact

FANDOM APPS

Take your favorite fandoms with you and never miss a beat.

FANDOM

FAN CENTRAL
BETA

GAMES

ANIME

MOVIES

TV

VIDEO

WIKIS

START A WIKI

VIEW MOBILE SITE