

SC310 Extra Software Programming Guide

Custom Property

- 1. **KSPROPERTY_CUSTOM_XET_GPIO_DIRECTION (940)**
- 1. **KSPROPERTY_CUSTOM_XET_GPIO_DATA (941)**
- 1. **KSPROPERTY_CUSTOM_XET_GPIO_SUPPORT (942) (READ ONLY)**

The property allows you to access CX25820/1's GPIO interface. The property KSPROPERTY_CUSTOM_XET_GPIO_DIRECTION allows you to control its direction. Here, writing 1 to bit enables this pin as output pin. Usually, the GPIO is controlled by the first chipset in one board.

SUPPORT VALUE: 0 ~ 1 - INPUT ~ OUTPUT

The property KSPROPERTY_CUSTOM_XET_GPIO_DATA allows you to access GPIO's data.

SUPPORT VALUE: 0 ~ 1 - LOW ~ HIGH

The property KSPROPERTY_CUSTOM_XET_GPIO_SUPPORT allows you to obtain GPIO's information (pin size) on hardware board. Developer can use it to check if the device can support GPIO access.

SUPPORT VALUE: 0 IS NON-SUPPORT

EXAMPLE#01: TO DEFINE GPIO AS 16 OUTPUT PINS [0:15] AND 16 INPUT PINS [16:31].
`AMESDK_SET_CUSTOM_PROPERTY(hDev, 940, 0x0000FFFF);`

EXAMPLE#02: TO DEFINE GPIO AS 32 OUTPUT PINS [0:31] THEN PULL HIGH FOR ALL.
`AMESDK_SET_CUSTOM_PROPERTY(hDev, 940, 0xFFFFFFFF);`
`AMESDK_SET_CUSTOM_PROPERTY(hDev, 941, 0xFFFFFFFF);`

EXAMPLE#03: TO DEFINE GPIO AS 32 INPUT PINS [0:31] THEN READ DATA FROM IT.
`AMESDK_SET_CUSTOM_PROPERTY(hDev, 940, 0x00000000);`
`AMESDK_GET_CUSTOM_PROPERTY(hDev, 941, &GPIO);`

2. Application Note for AMESDK_GET_LOCK()

Customer to use AMESDK_GET_LOCK, please notes it. CX28820/1 is one 8CH integrated SOC. In order to reducing your software loading, we can group 8 channels' status into 8bits return value. You can call AMESDK_GET_LOCK to obtain 8CHs' status at the same time.

EXAMPLE#01: GET SC310N8 SIGNAL STATUS.

```
AMESDK_GET_LOCK( hDev[ 0 ], &status ); // GET CH01 ~ CH08 STATUS
ULONG status_ch01 = (status >> 0) & 0x01;
ULONG status_ch02 = (status >> 1) & 0x01;
...
ULONG status_ch08 = (status >> 7) & 0x01;
```

EXAMPLE#02: GET SC310N16 SIGNAL STATUS.

```
AMESDK_GET_LOCK( hDev[ 0 ], &status ); // GET CH01 ~ CH08 STATUS
ULONG status_ch01 = (status >> 0) & 0x01;
ULONG status_ch02 = (status >> 1) & 0x01;
...
ULONG status_ch08 = (status >> 7) & 0x01;
```

```
AMESDK_GET_LOCK( hDev[ 8 ], &status ); // GET CH09 ~ CH16 STATUS
ULONG status_ch09 = (status >> 0) & 0x01;
ULONG status_ch10 = (status >> 1) & 0x01;
...
ULONG status_ch16 = (status >> 7) & 0x01;
```

3. Access Custom Property for DirectShow Developer

Customer uses DirectShow to develop software can bypass our SDK to access CX25820/1 directly. All custom properties are implemented by IKsPropertySet interface. The interface can be queried from our capture source filter.

EXAMPLE#01: GET GPIO DATA.

```
static const GUID GUID_KPS_CX2581 = { 0xD1E5209F, 0x68FD, 0x4529, 0xBE, 0xE0, 0x5E, 0x7A, 0x1F, 0x47, 0x92, 0x17 };
ULONG gpio = 0;
m_pKsPropertySet->Set( GUID_KPS_CX2581, 941, NULL, 0, &gpio, sizeof(ULONG) );
```

4. Application Note for DirectShow Developer

The developer who uses DirectShow to access our capture source filter need check the frame size in the callback function of your SampleGrabber class. If the frame size is 0 bytes, it means the frame is one bad frame. You should drop it. More detail, please check with our engineer team directly.