

One-to-one connection

This example shows how to create a one-to-one connection between the TIMER compare register and the SAADC start task.

The channel configuration is set up first. TIMER will publish its COMPARE0 event on channel 0, and SAADC will subscribe its START task to events on the same channel. When this is finished, the channel is enabled through the DPPIC.

```
NRF_TIMER20->PUBLISH_COMPARE[0] = (0 << TIMER_PUBLISH_COMPARE_CHIDX_Pos) |
    TIMER_PUBLISH_COMPARE_EN_Msk;
NRF_SAADC->SUBSCRIBE_START = (0 << SAADC_SUBSCRIBE_START_CHIDX_Pos) |
    SAADC_SUBSCRIBE_START_EN_Msk;
NRF_DPPIC20->CHENSET = DPPIC_CHENSET_CH0_Msk;
```

Many-to-many connection

The following example shows how to create a many-to-many connection, showcasing the DPPIC's channel group functionality.

A channel group that includes only channel 0 is set up first. Then the GPIOTE and TIMER configure their IN0 and COMPARE0 events respectively to be published on channel 0, while the SAADC configures its START task to subscribe to events on channel 0. Through DPPIC, the CHG0 DISABLE task is configured to subscribe to events on channel 0. After an event is received on channel 0 it will be disabled. Finally, channel 0 is enabled using the DPPIC task to enable a channel group.

```
NRF_DPPIC20->CHG[0] = (DPPIC_CHG_CH0_Included << DPPIC_CHG_CH0_Pos);
NRF_GPIOTE20->PUBLISH_IN[0] = (0 << GPIOTE_PUBLISH_IN_CHIDX_Pos) |
    GPIOTE_PUBLISH_IN_EN_Msk;
NRF_TIMER20->PUBLISH_COMPARE[0] = (0 << TIMER_PUBLISH_COMPARE_CHIDX_Pos) |
    TIMER_PUBLISH_COMPARE_EN_Msk;
NRF_SAADC->SUBSCRIBE_START = (0 << SAADC_SUBSCRIBE_START_CHIDX_Pos) |
    SAADC_SUBSCRIBE_START_EN_Msk;
NRF_DPPIC20->SUBSCRIBE_CHG[0].DIS = DPPIC_CHENSET_CH0_Msk | DPPIC_SUBSCRIBE_CHG_DIS_EN_Msk;
NRF_DPPIC20->TASK_CHG[0].EN = 1;
```

6.2.4 Special considerations for a system implementing TrustZone for Cortex-M processors

DPPI is implemented with split security in order to handle both secure and non-secure accesses. In a system implementing the TrustZone for Cortex-M technology, DPPI channels can be defined as secure or non-secure using the SPU.

A peripheral configured as non-secure can only subscribe to or publish on non-secure DPPI channels. A peripheral configured as secure can access all DPPI channels. DPPI handles both secure and non-secure accesses, but behaves differently depending on the access type.

- A non-secure peripheral access can only configure and control the DPPI channels defined as non-secure in the SPU.DPPI.PERM[n] register
- A secure peripheral access can control all the DPPI channels, independently of the SPU.DPPI.PERM[n] register