Tasks and Events, PPI, GPIOTE, and EasyDMA

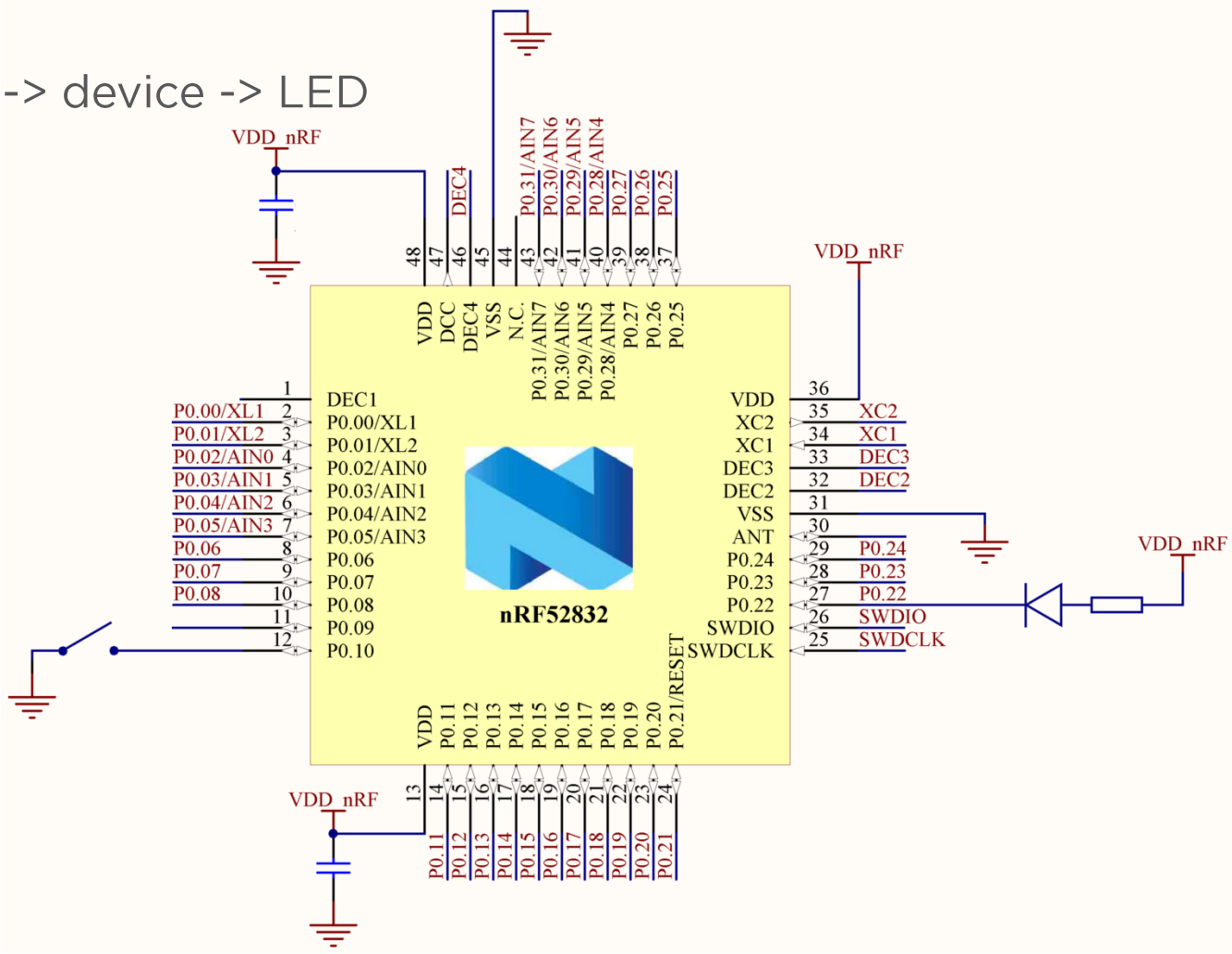
nRF52 Global Tech Tour

Outline

- **Tasks and events**
- PPI
- GPIO and GPIOTE
- EasyDMA
- Demo

What to solve

- Button -> device -> LED



What to solve, SW solution

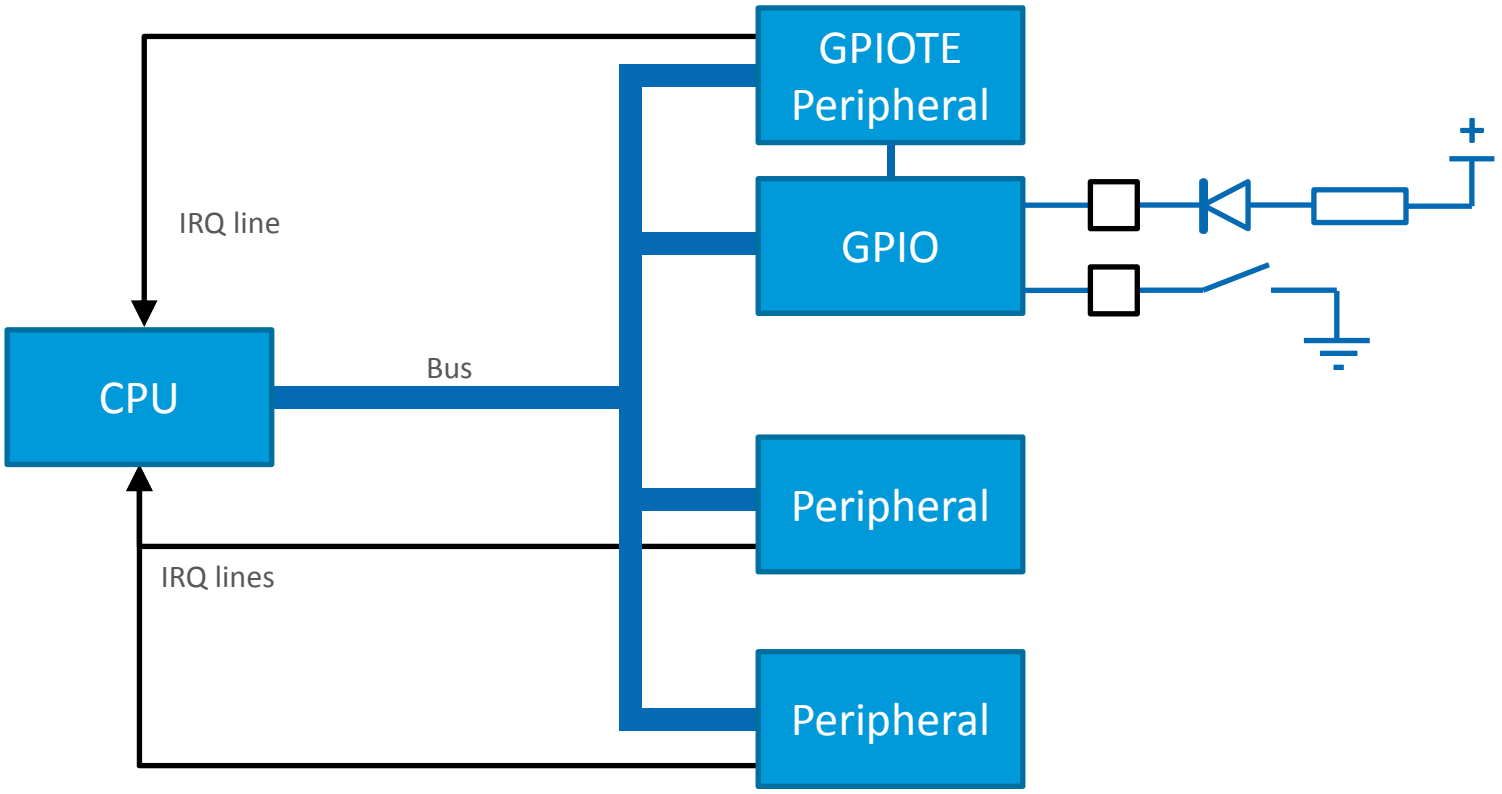
- Loop mode

```
While(1){
  if(pin_high)
    output = true;
  else
    output = false;
....
}
```

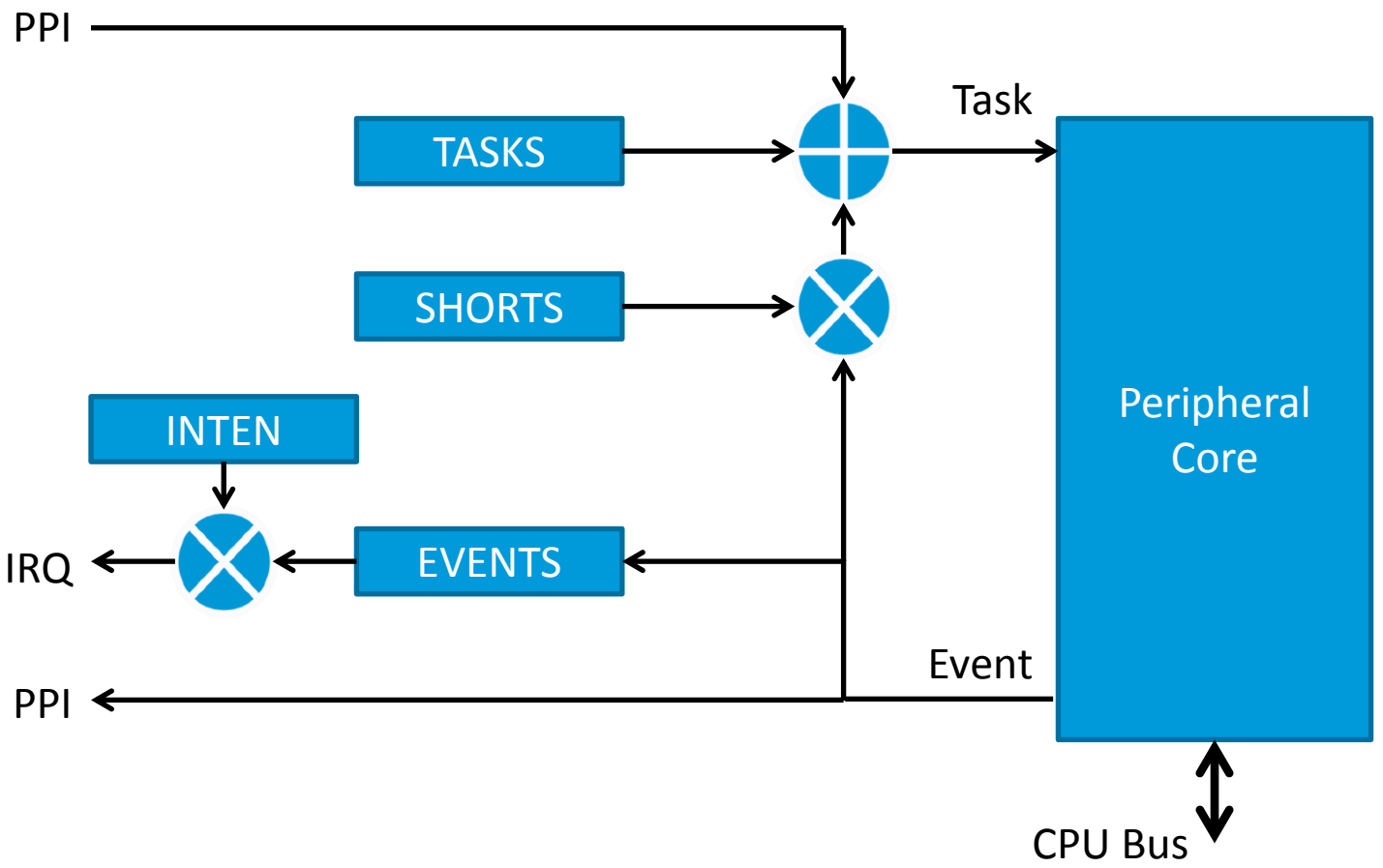
- Interrupt mode

```
Enable external interrupt
Enable global interrupts
ISR extint(void){
  if(pin_high)
    output = true;
  else
    output = false;
}
```

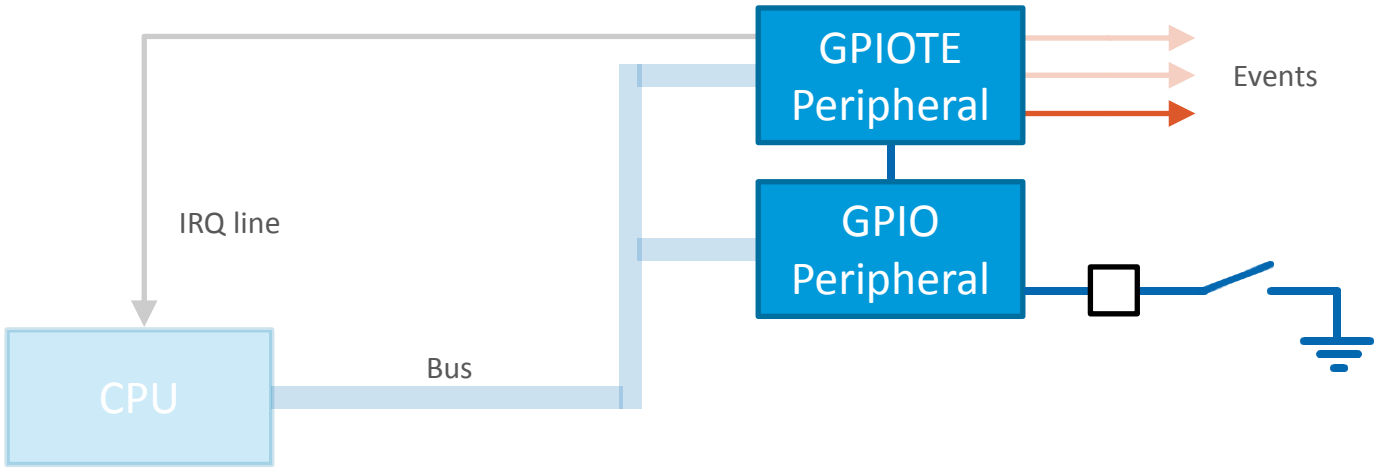
Interrupts



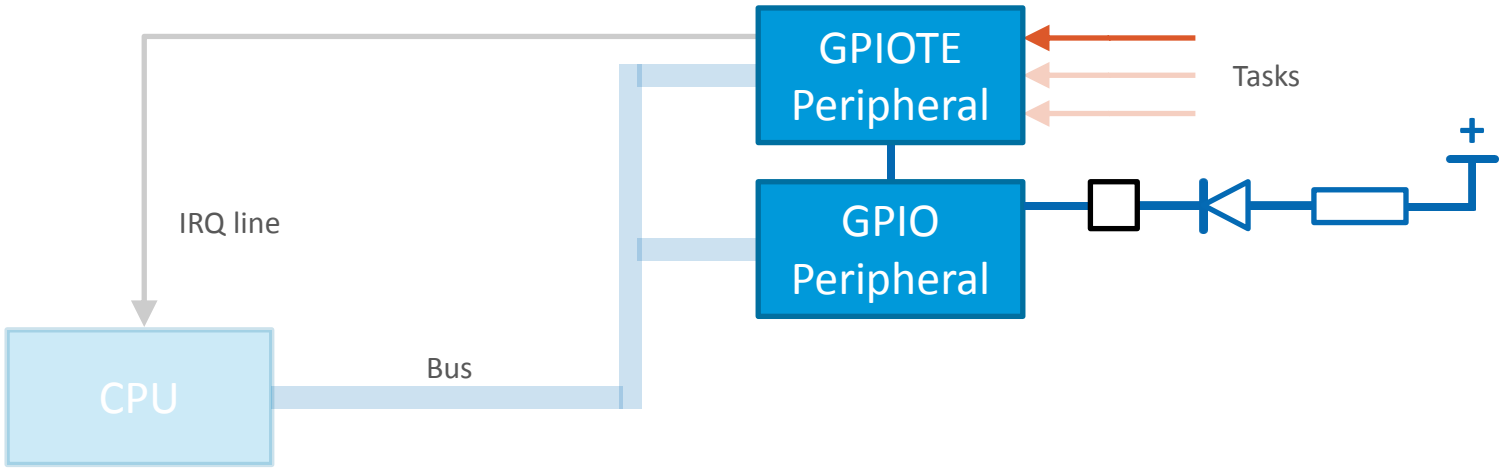
Tasks and events



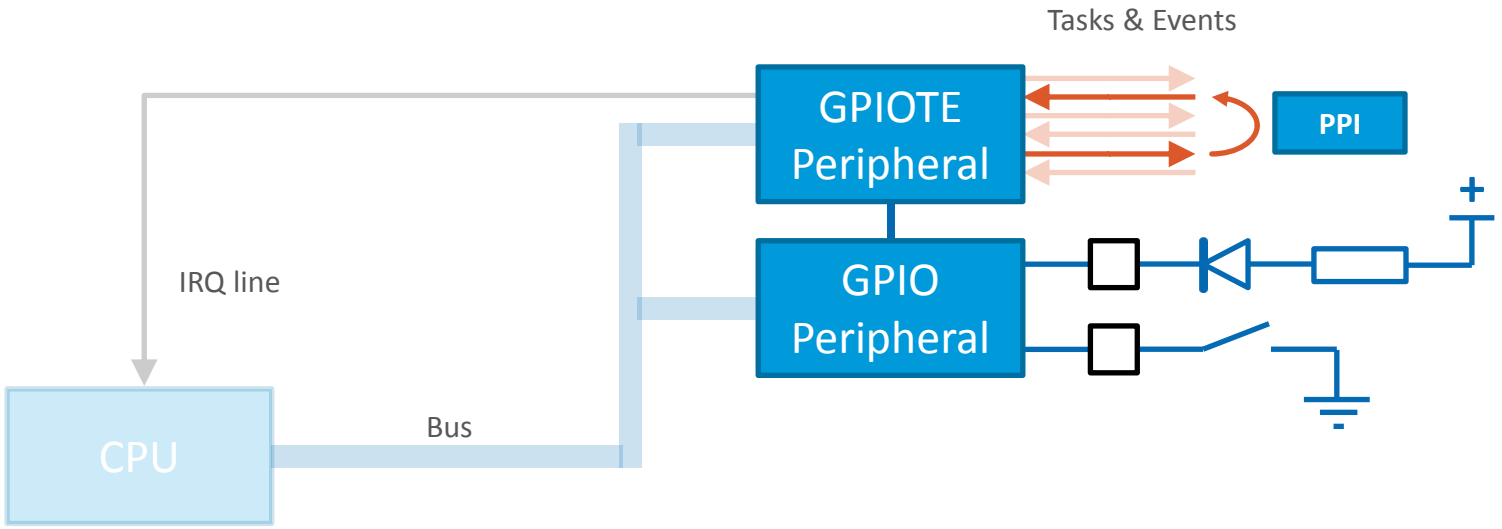
GPIOTE events



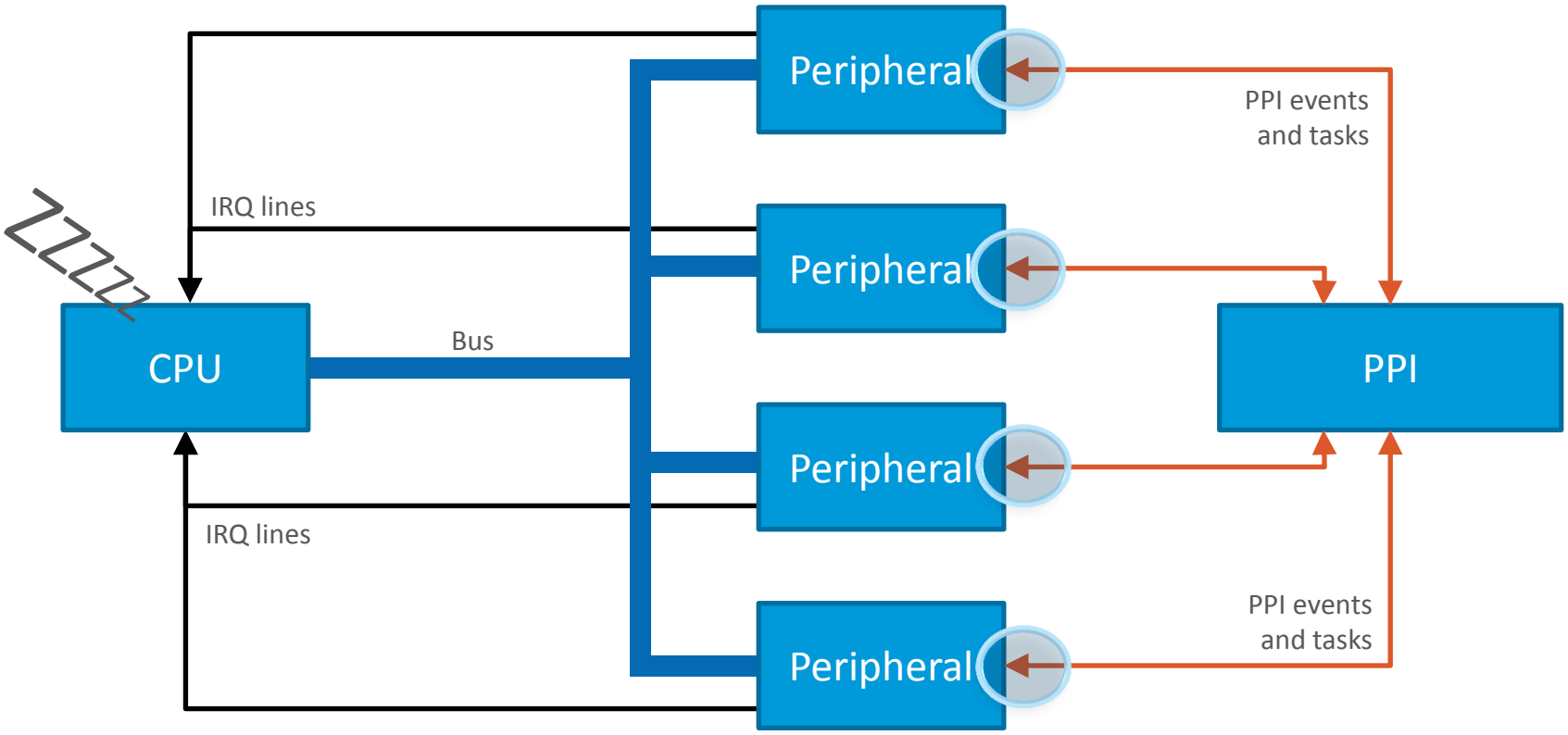
GPIOTE tasks



Linking an event to a tasks



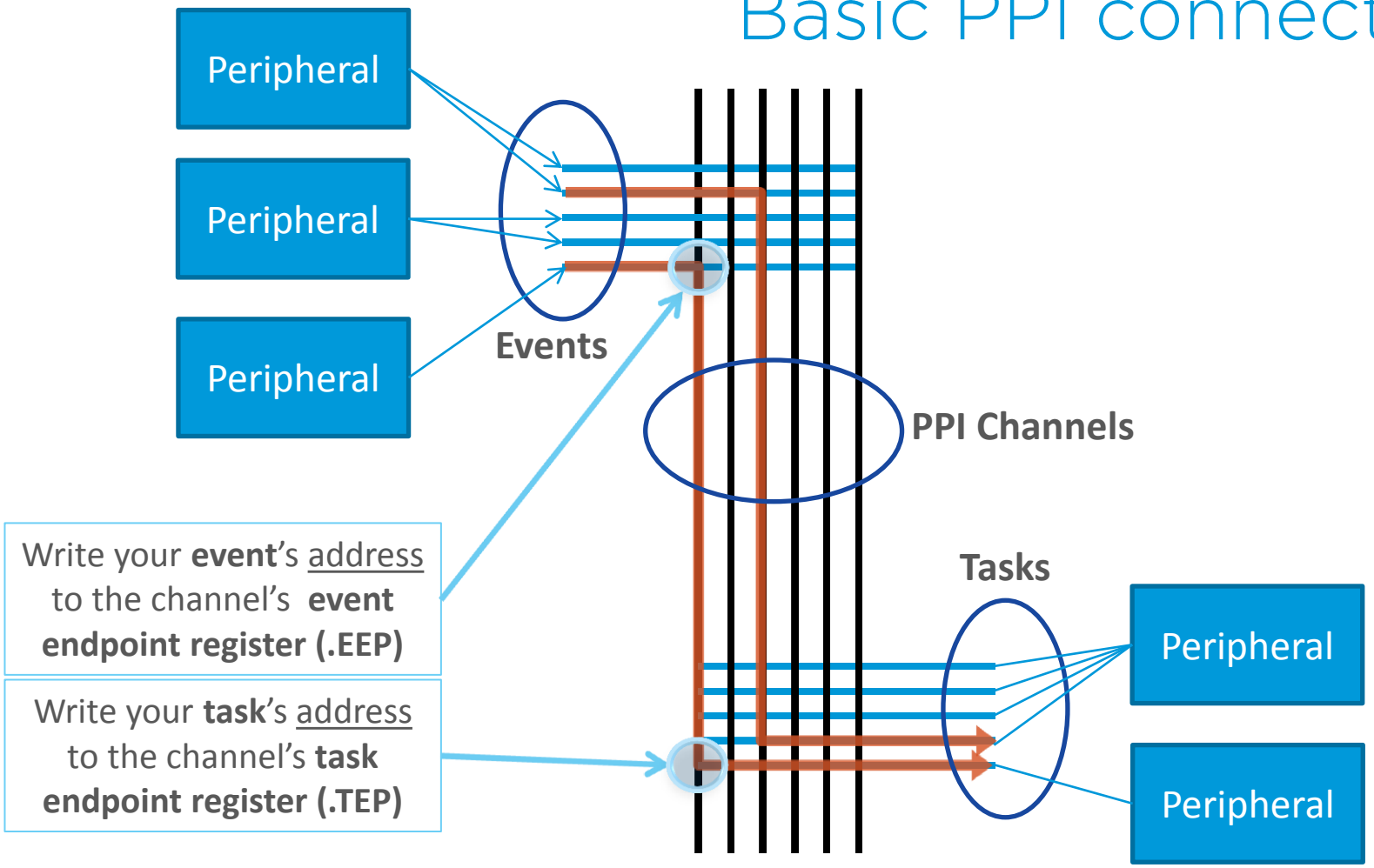
Whole system with Programmable Peripheral Interconnect (PPI)



Outline

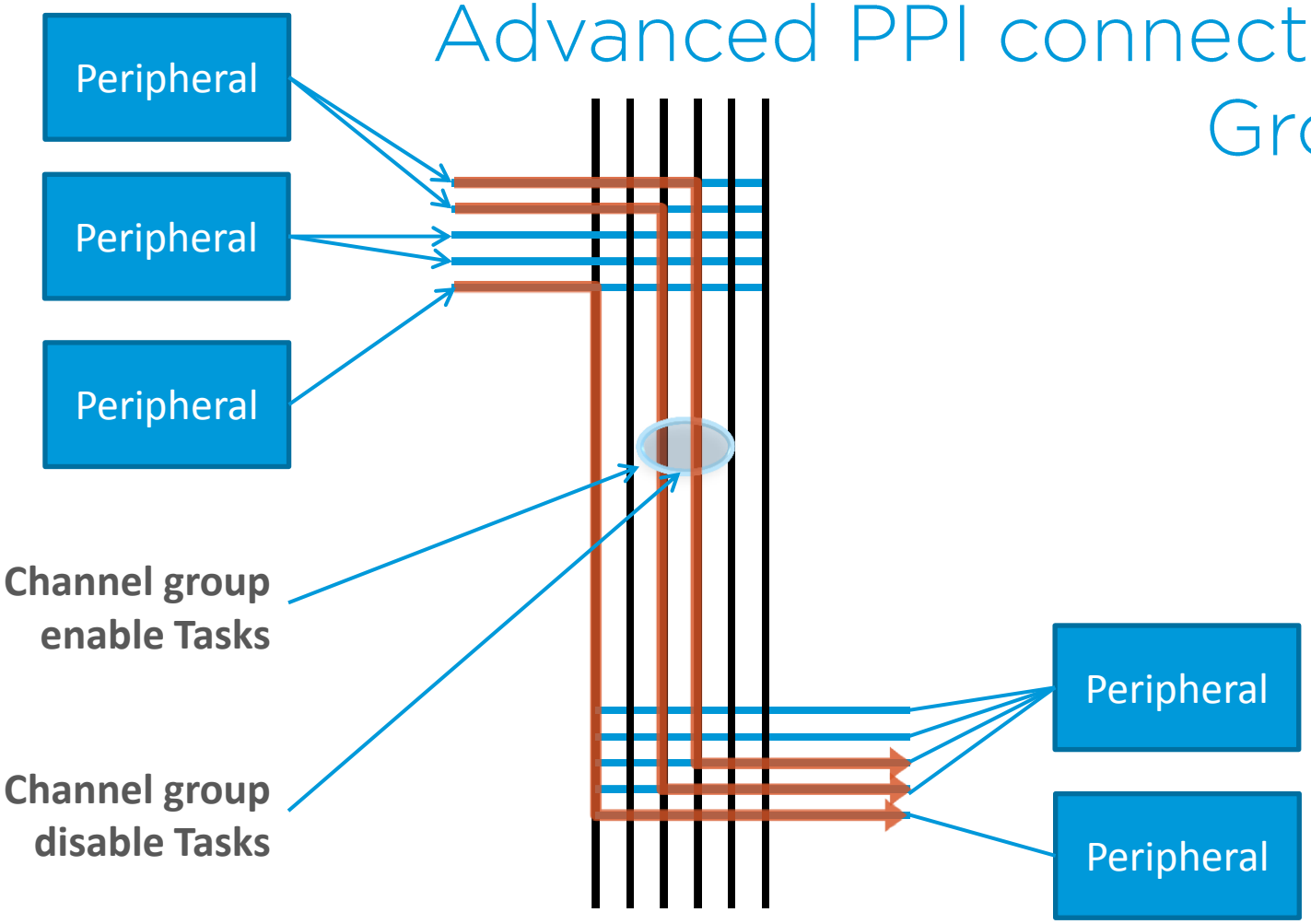
- Tasks and events
- **PPI**
- GPIO and GPIOTE
- EasyDMA
- Demo

Basic PPI connections



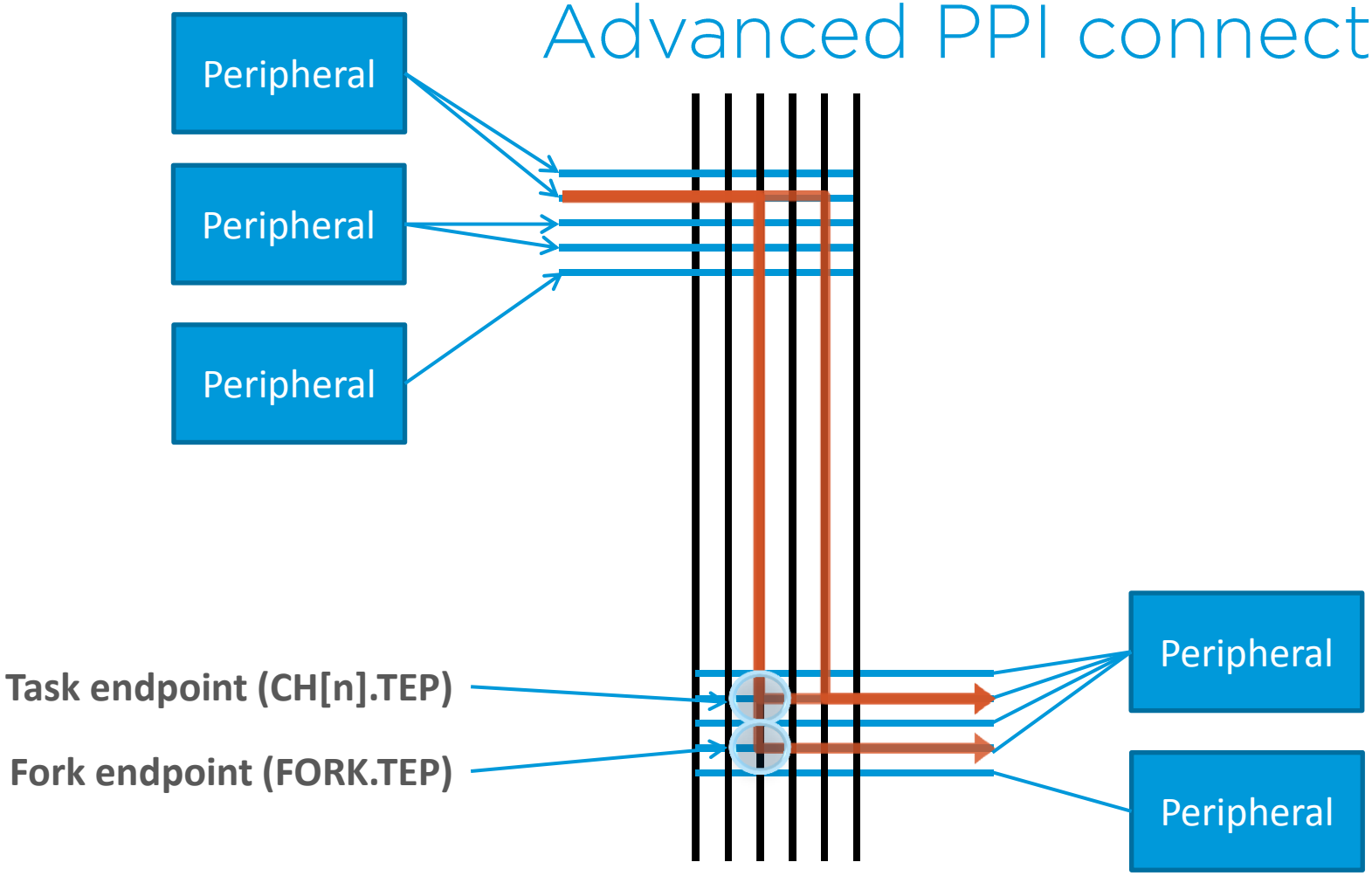
Advanced PPI connections: Groups

Advanced PPI connections: Groups



Advanced PPI connections: Fork

Advanced PPI connections: Fork



PPI features

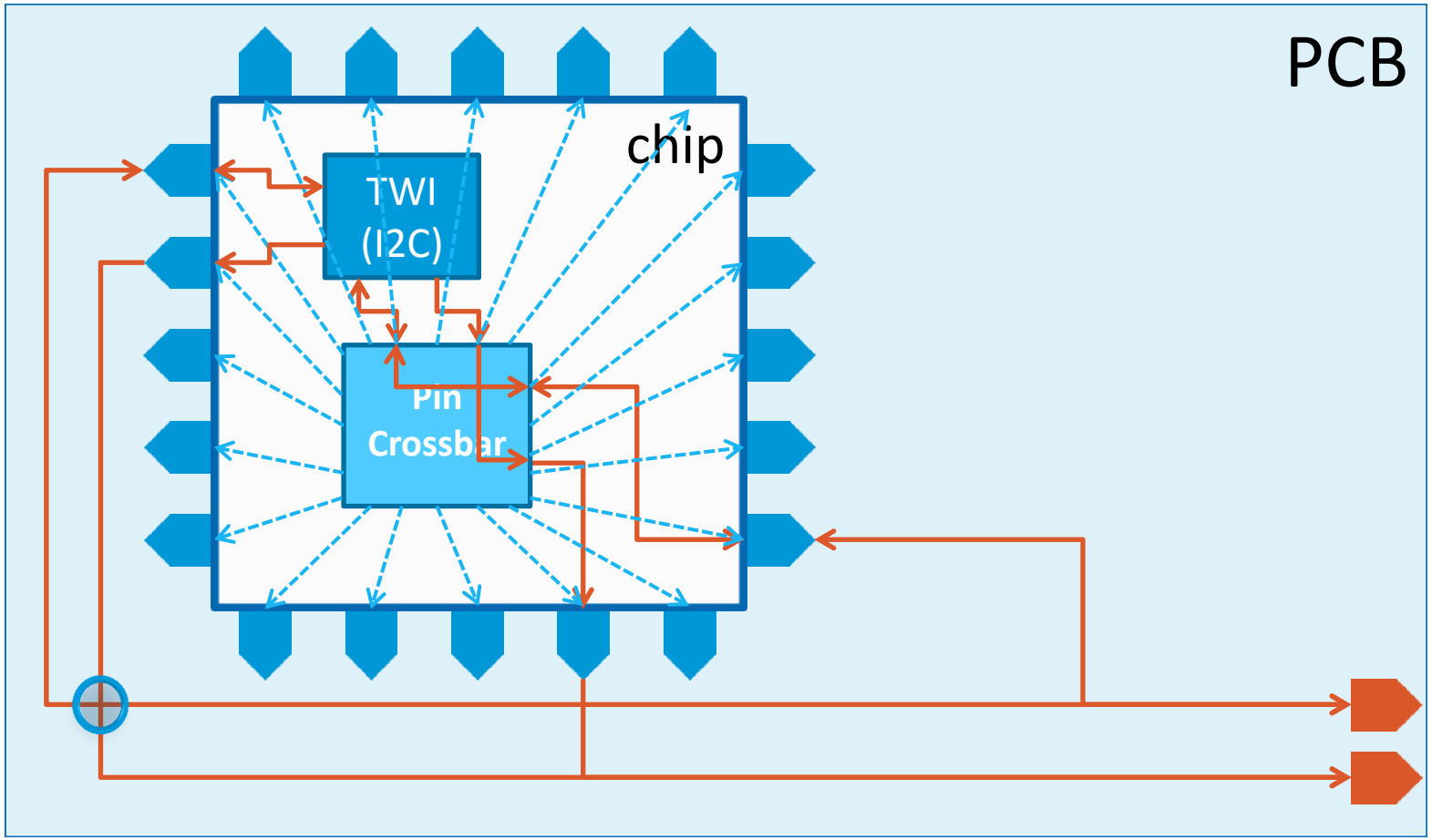
Feature	nRF51	nRF52
Programmable Channels	16 (channel 0-15)	20 (channel 0-19)
Preprogrammed Channels	12 (channel 20-31)	12 (channel 20-31)
Channel groups	4	6
Fork	Not available	All 32 channels can be forked

- Any channel can be included in any group
- A channel can be part of more than one group
- Multiple events can control the same task (but uses multiple channels)
- An event can be connected to several channels

Outline

- Tasks and events
- PPI
- **GPIO and GPIOTE**
- EasyDMA
- Demo

Peripherals pin access



GPIOTE

General Purpose IO Tasks and Events

- Can send tasks off chip
- Can convert external events to internal events
- Fully integrated with PPI and IRQ system

GPIOTE Input mode: Receive external events

- Generate event on rising, falling or toggle of external signal
- Does not require 16Mhz clock
  - Keeps peripherals power domain on, resulting in increased leakage
- Port event can be used to detect inputs without extra leakage
  - Port event has new Latch mode where input pin(s) triggering the event is logged.
  - Latch status is set when input condition is met and has to be cleared by user

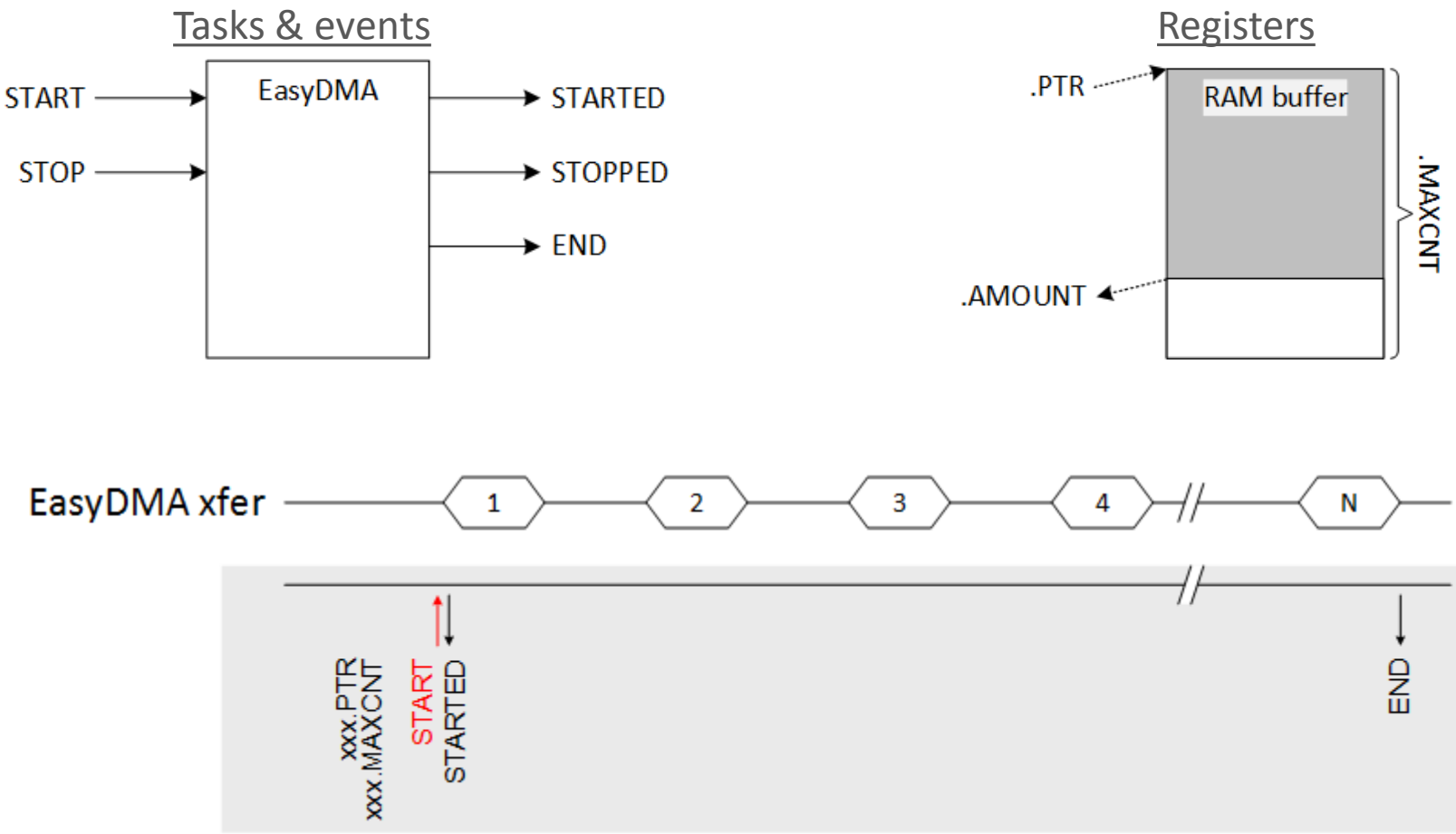
GPIOTE Output mode: Send events off chip

- Generate signals for external devices based on any internal event
- TASKS_OUT: can be programmed to set output high, low or to toggle value
- TASKS_SET: New task in nRF52832: Sets output high
- TASKS_CLEAR: New task in nRF52832: Sets output low

Outline

- Tasks and events
- PPI
- GPIO and GPIOTE
- **EasyDMA**
- Demo

Interacting with EasyDMA

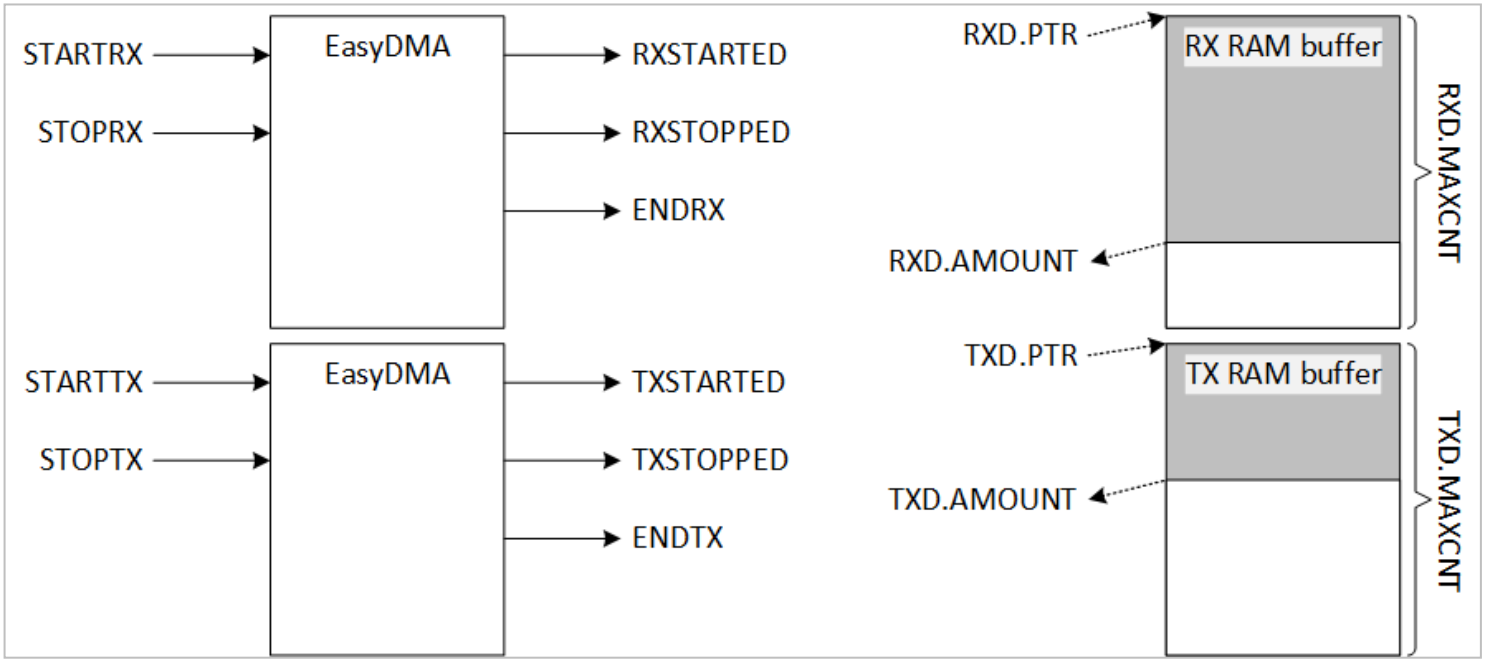


Interacting with EasyDMA

Control registers	Status registers	
xxx.PTR		Start of RAM buffer
xxx.MAXCNT		Size of RAM buffer
xxx.LIST		Autolog configuration
	xxx.AMOUNT	RAM buffer consumed

Tasks	Events	
START	STARTED	.PTR & .MAXCNT captured
STOP	STOPPED	
	END	Done with RAM access

Multiple EasyDMA channels



Outline

- Tasks and events
- PPI
- GPIO and GPIOTE
- EasyDMA
- **Demo**

PPI and GPIOTE demo

- Code to show how to set up a PPI connection from one input to one output using GPIOTE
  - Use SDK drivers
  - May be included in a SoftDevice project later on so should preferably support this as well
  - Connection between a button and a LED for visual check

Source code

```
#define GPIO_OUTPUT_PIN_NUMBER BSP_LED_0  /**< Pin number for output. */
#define GPIO_INPUT_PIN_NUMBER BSP_BUTTON_0  /**< Pin number for output. */

static void led_blinking_setup()
{
    uint32_t input_evt_addr;
    uint32_t gpiote_task_addr;
    nrf_ppi_channel_t ppi_channel;

    // Get the instance allocated for both OUT task and IN event
    gpiote_task_addr = nrf_drv_gpiote_out_task_addr_get(GPIO_OUTPUT_PIN_NUMBER);
    input_evt_addr = nrf_drv_gpiote_in_event_addr_get(GPIO_INPUT_PIN_NUMBER);

    // Get a PPI channel from the PPI pool
    APP_ERROR_CHECK(nrf_drv_ppi_channel_alloc(&ppi_channel));
    // Tie the GPIOTE IN event and OUT task through allocated PPI channel
    APP_ERROR_CHECK(nrf_drv_ppi_channel_assign(ppi_channel, input_evt_addr, gpiote_task_addr));
    // Enable the allocated PPI channel
    APP_ERROR_CHECK(nrf_drv_ppi_channel_enable(ppi_channel));

    // Enable OUT task and IN event
    nrf_drv_gpiote_out_task_enable(GPIO_OUTPUT_PIN_NUMBER);
    nrf_drv_gpiote_in_event_enable(GPIO_INPUT_PIN_NUMBER, false);
}

/**
 *
 */
int
{
    led_blinking_setup();
    while (true)
    {
        // No need to do anything other than sleep,
        __WFE();
    }
}
```

Development tools

- Keil
  - IDE with
    - C-compiler
    - Debugger
    - Programming support
- nRFGo studio
  - Programming support
- nrfjprog
  - Command line programming tool