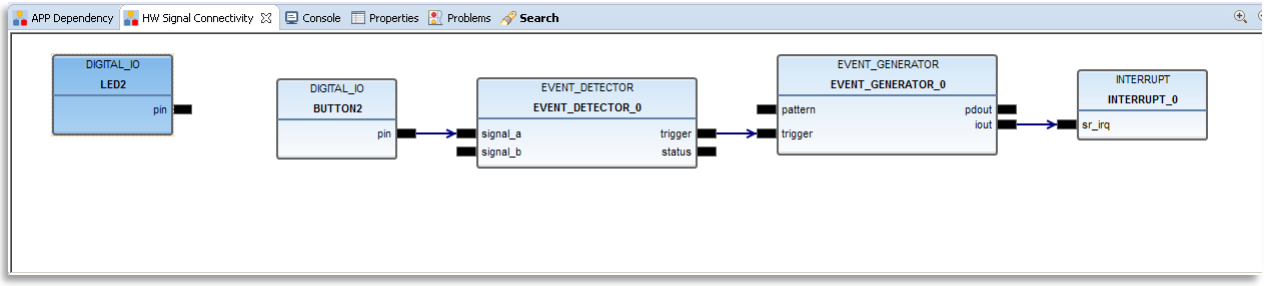
EXAMPLE NAME: XMC4500_Relax_Kit_External_Interrupt

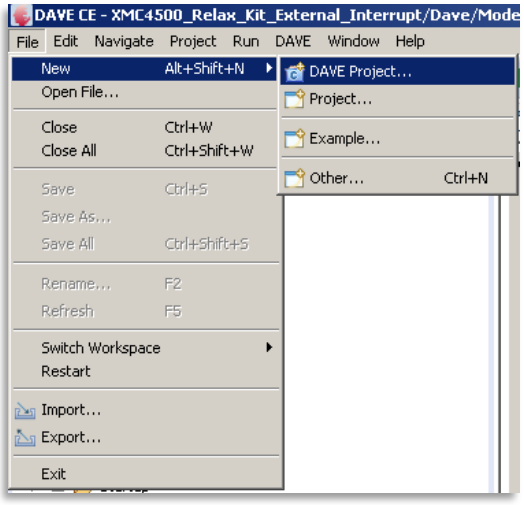
OVERVIEW:
This example shows how to configure *EVENT_DETECTOR* and *EVENT_GENERATOR* APPs to generate an interrupt on external events.

DESCRIPTION:
In this example, an EVENT_DETECTOR APP is configured to detect an event on the falling edge of an external input signal. The external input signal is coming from an instance of DIGITAL_IO APP, BUTTON2, which is connected to P1.15. Once the desired event is detected, the EVENT_GENERATOR APP triggers a service request that is handled by the INTERRUPT APP which toggles a led. The led is connected to P1.0 which is handled by an instance of DIGITAL_IO APP, LED2.



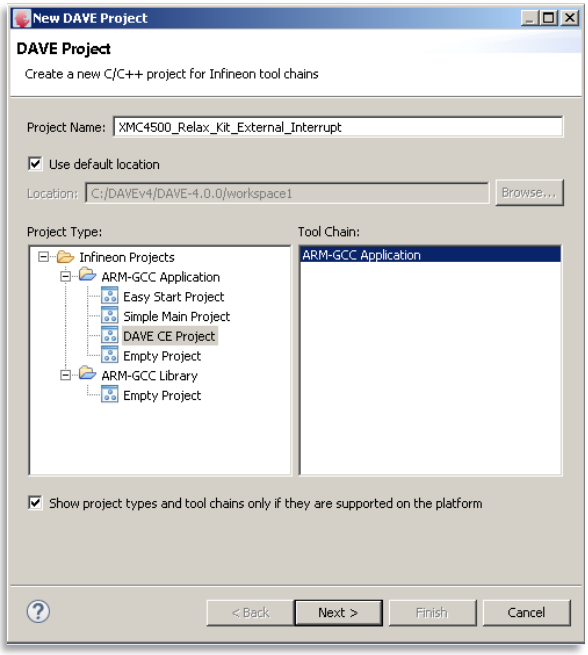
REQUIRMENTS:
Boards Required: XMC4500 Relex Kit Board (Order Nr: KIT_XMC45_RELAX_V1)

HOW TO CREATE THE PROJECT:
1. Create a new DAVE CE Project
   To create new projects go to **File** menu, select **New** and choose **DAVE Project**.



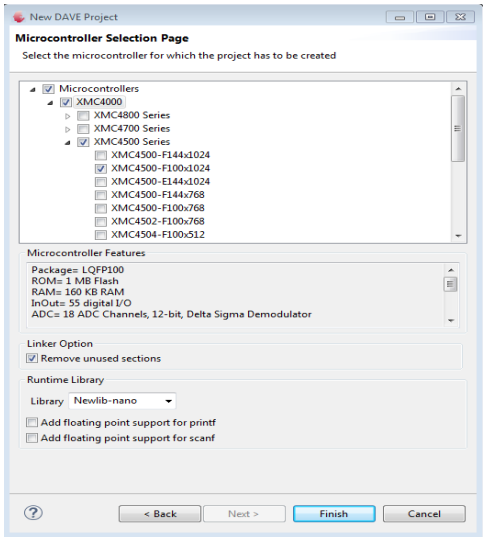
In the **New DAVE Project** window:

- in the *Project name:* field enter the name of the new project, i.e. XMC4500_Relax_Kit_External_Interrupt.
- in the *Project type:* section select **DAVE CE Project**.
- in the *Toolchains:* section select **ARM-GCC Application**
- click the Next > button



In the **Target Selection Page**:

- select the target device, i.e. **XMC4500-F100x1024**
- click the **Finish** button

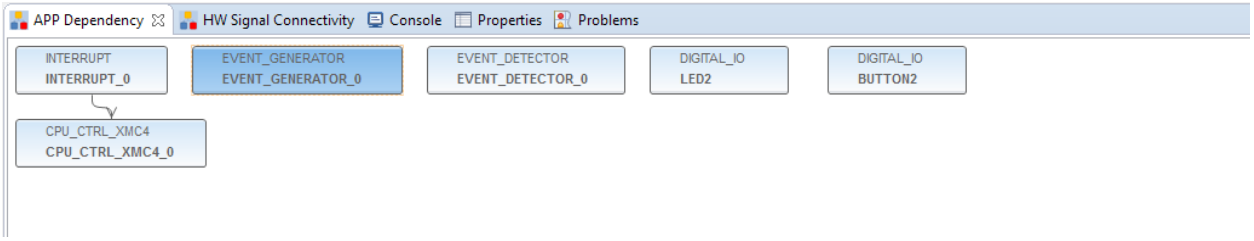


2. Add APPs to your project

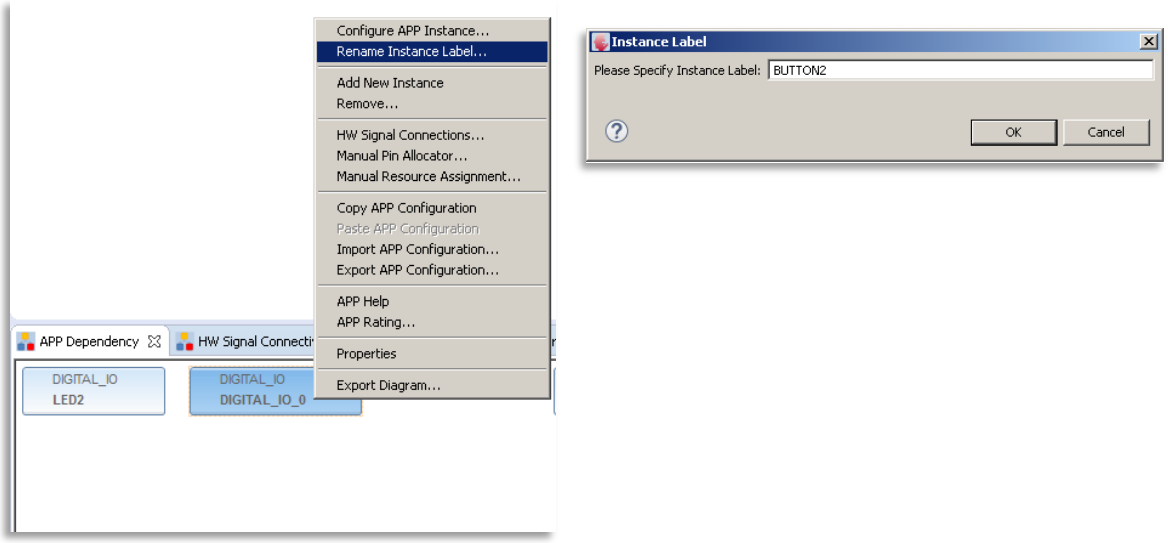
Use the *Add New App* button in toolbar or go to **DAVE** menu and select *Add New App*. In the **Add New APP** dialog, add the following APPs by double clicking:

- add an EVENT_GENERATOR APP
- add an EVENT_DETECTOR APP
- add two DIGITAL_IO APPs
- add an INTERRUPT APP
- click the **Close** button

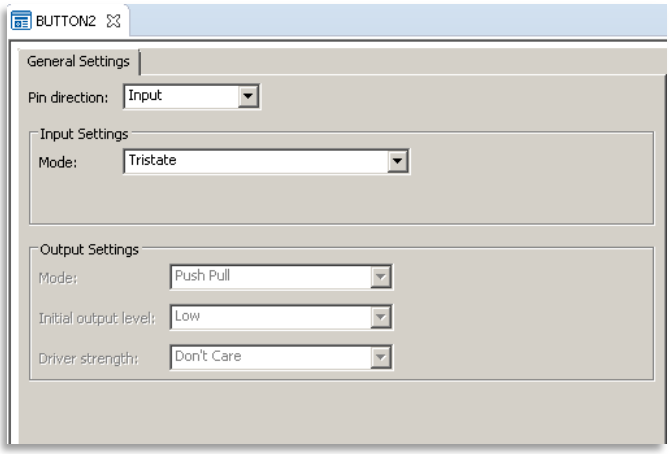
APP Dependency View is shown below:



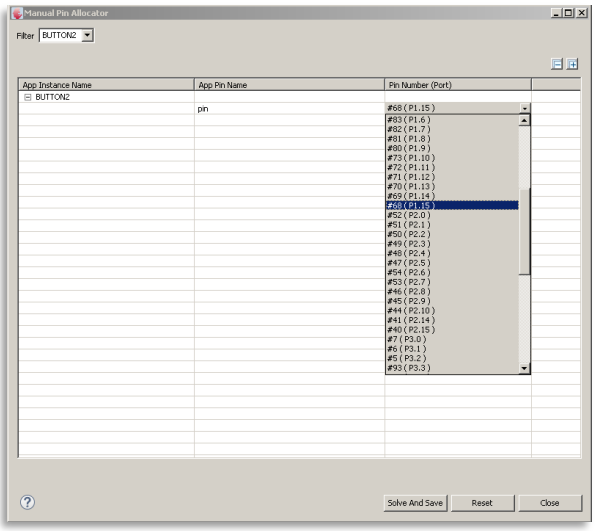
3. Assign names to APP instances
   - Right click on the DIGITAL_IO_0 APP instance, select *Rename Instance Label…* and set the name to BUTTON2.
   - Right click on the DIGITAL_IO_1 APP instance, select *Rename Instance Label…* and set the name to LED2.



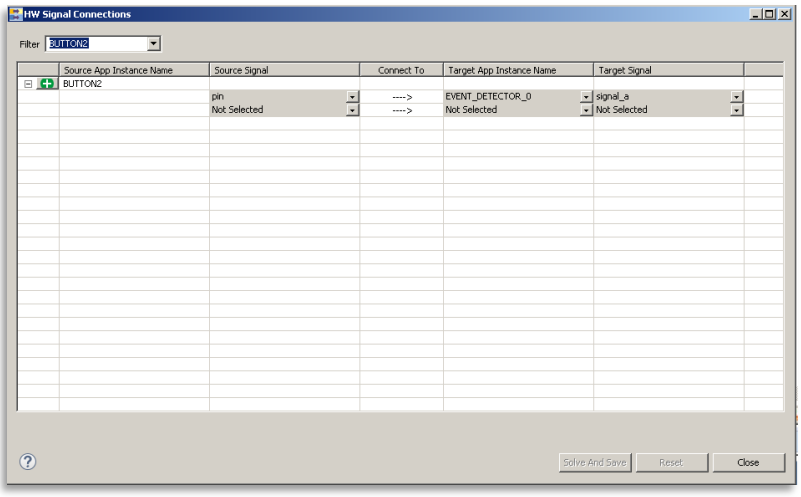
4. Configure APPs and signals
   - Double click on BUTTON2 APP instance to configure the APP:
     - select **Input** in the *Pin Direction:* selection.
     - keep default for the other of configuration settings.



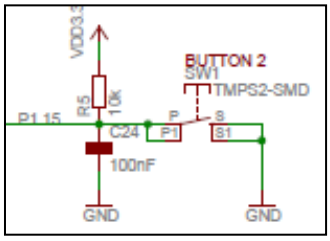
   - Right click on BUTTON2 APP instance, select *Manual Pin Allocator…*
     In the **Manual Pin Allocator** dialog:
     - select P1.15 in the *Pin Number (Port)* selection
     - click on **Solve and Save** button and finally on **Close** button.



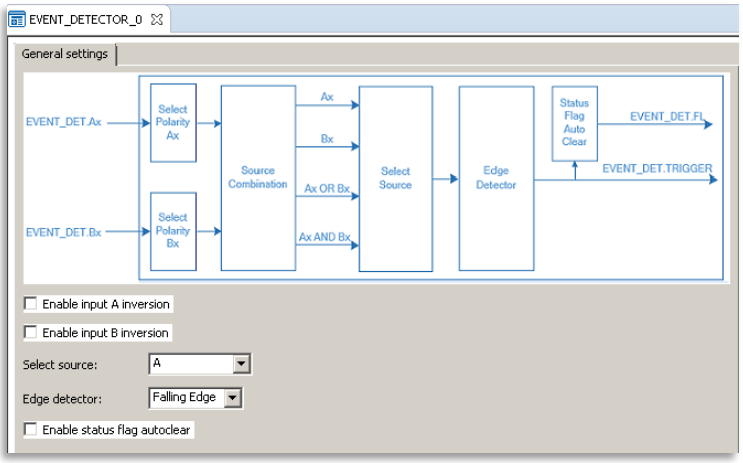
- Right click on BUTTON2 APP instance, select *HW Signal Connections…*
  In the **HW Signal Connections** dialog:
  - select **pin** in the *Source Signal* selection**.**
  - select **EVENT_DETECTOR_0** in the *Target APP Instance Name* selection*.*
  - select **signal_a** in the *Target Signal* selection*.*
  - Click on **Solve and Save** button and finally on **Close** button.



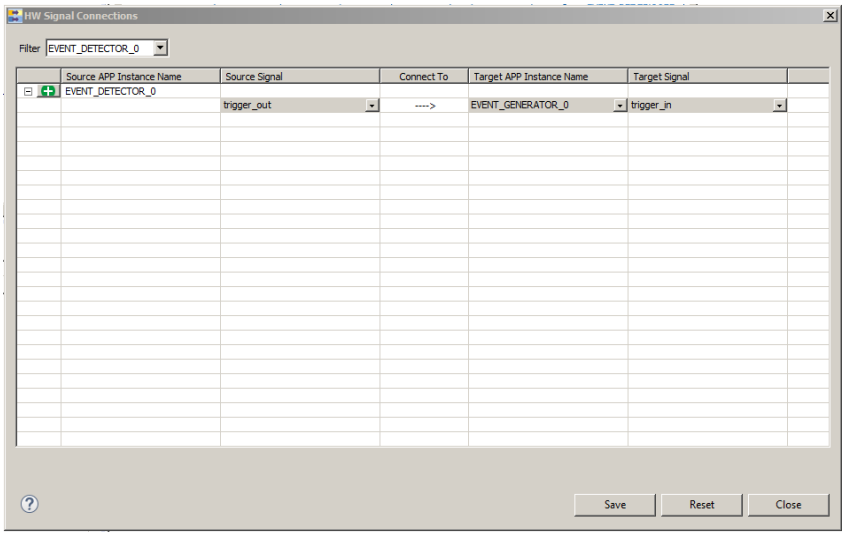
- Double click on EVENT_DETECTOR_0 APP instance to configure the APP:
  - select **A** in the *Select Source:* selection
  - select **Falling Edge** in the *Edge Detector:* selection
    The BUTTON2 is VDD terminated and only when pressed the voltage will go down.



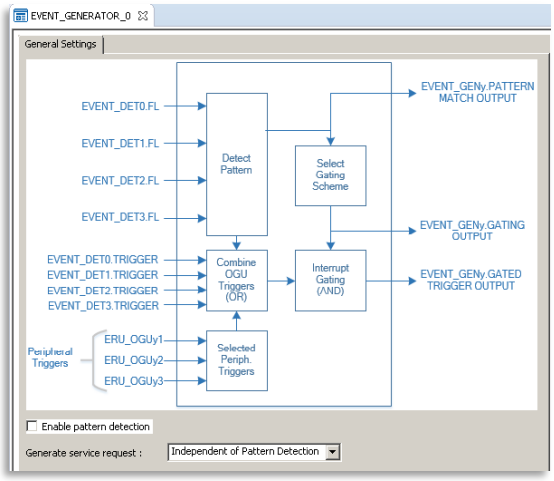
  - keep default for the other of configuration settings.



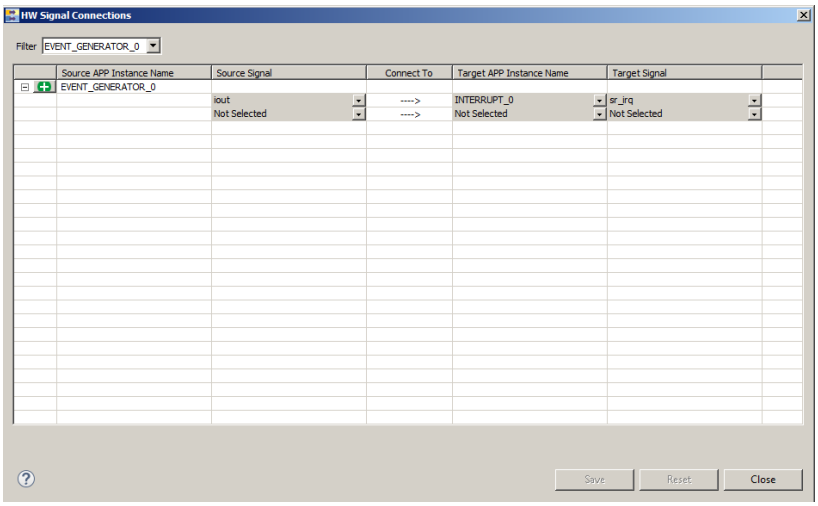
- Right click on EVENT_DETECTOR_0 APP instance, select *HW Signal Connections…*
  Connect the EVENT_DETECTOR_0 ::trigger_out output signal to EVENT_GENERATOR_0::trigger_in input signal.



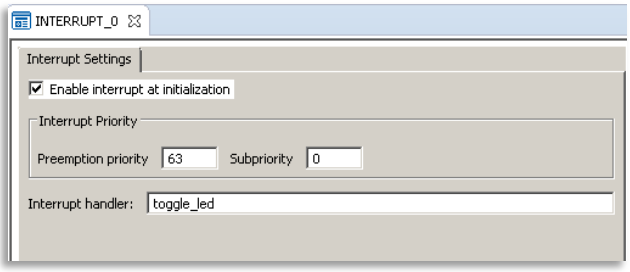
- Double click on EVENT_GENERATOR_0 APP instance to configure the APP:
  - Select **Independent of Pattern Detection** in *Generate service request:* selection
    In this way a service request will be triggered independent of the pattern detection result (featured not used in this example).
  - keep default for the other of configuration settings.



- Right click on EVENT_GENERATOR_0 APP instance, select *HW Signal Connections…*
  Connect the EVENT_GENERATOR ::iout output signal to INTERRUPT_0::sr_irq input signal.

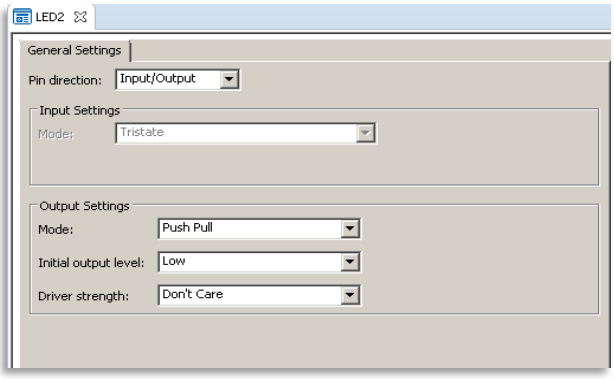


- Double click on INTERRUPT_0 APP instance to configure the APP:
  - enter **toogle_led** in the *Interrupt Handler:* input box.
  - keep default settings for the other configuration settings.

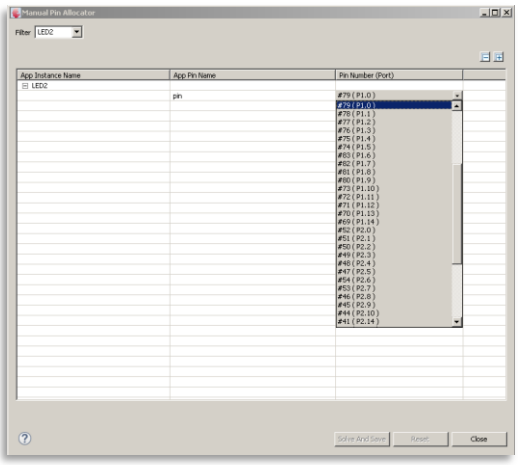


- Double click on LED2 APP instance
  - select **Input/Output** in *Pin Direction:* selection

- keep default for the other of configuration settings.



- Right click on LED2 APP instance
  In the **Manual Pin Allocator** dialog:
  - select P1.0 in the *Pin Number (Port)* selection
  - click on **Solve and Save** button and finally on **Close** button.



5. Generate the code

   Use the button  in toolbar or go to **DAVE** menu and select *Generate Code.*
   In the project a new folder *Generated* is created under *DAVE* folder. This folder contains a subfolder for every APP type used in the project. Each APP subfolder contains the generated APP configuration in the form of configuration structures, i.e. EVENT_DETECTOR_conf.c. In addition, the APP API library files are copied.

6. Add user code to main.c
   In this example, the only code that needs to be added is the implementation of the **toogle_led** function.

```
void toggle_led(void)
{
  DIGITAL_IO_ToggleOutput(&LED2);
}
```

7. Build and download to the microcontroller.