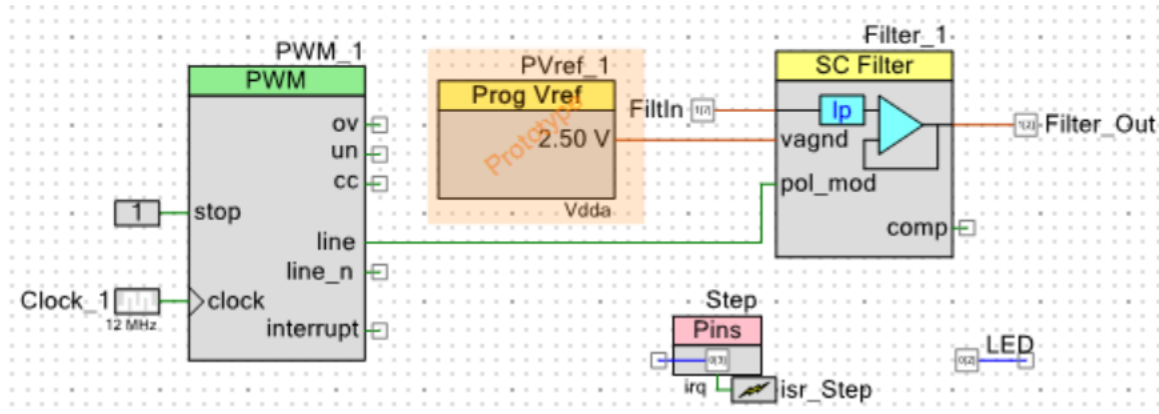


## Stepped Switched-Cap Filter in PSoC Analog Coprocessor

### Project overview

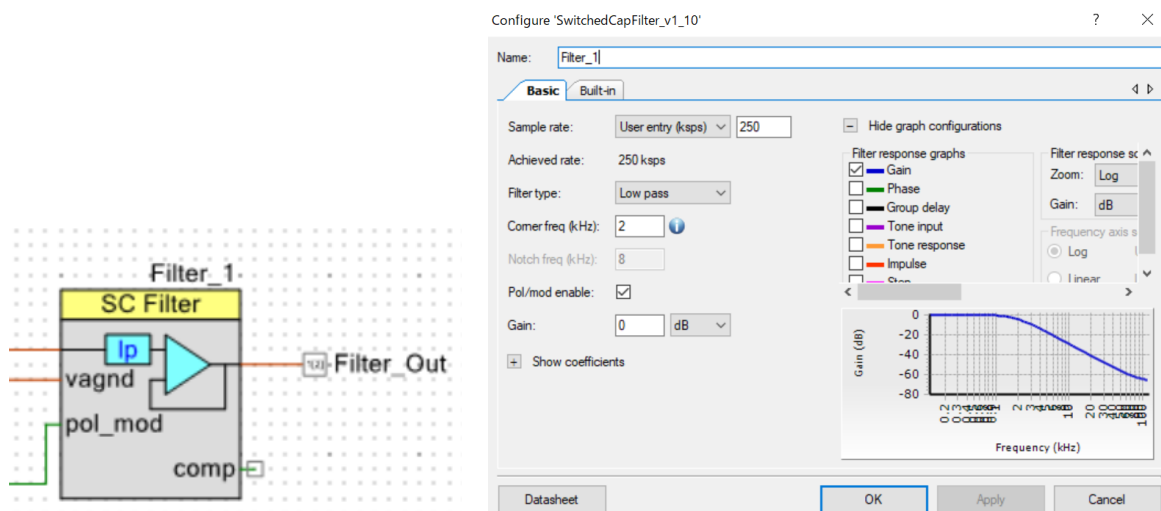
This project achieves programmable cutoff frequency for Low Pass Filter with Switched-Cap Filter component.

The project was developed in PSoC Creator 4.2. While 4.3 is available, it has a issue that won't let any switched-capacitor filters be implemented.



- SC Filter component achieves Low pass filter. To open Configuration tab, it is seen a setting for corner frequency and transfer function calculated automatically.

In this project, Corner frequency set in the configuration is just for initialization value. It will be set some of frequency required in main.c.



- Project steps through 2-pole low pass filters at 2.0, 3.0, and 4.0kHz.

Before set corner frequency required in main.c, it should get a capacitor value against each corner frequency.

The calculation of them is complicated, so it must be useful to figure out from register for capacitor value which is set by corner frequency decided in component configuration.

For example, the capacitor values are held in register 0x40340208 for UAB half A and register 0x40340308 for UAB half B, shown here for filter with corner frequency = 2.0 kHz.

It sets 2kHz for corner frequency in configuration, and then check into these registers as below.

Address:	0x40340208	Address Sp	Address:	0x40340308	Address Sp
0x40340208	43	0c 03 18	0x40340308	04	01 00 1a

Bunch of capacitor values against each corner frequency are prepared in main.c, and each capacitor values are set into registers depending on a processing condition as below.

```
int main()
{
    PVref_1_Start();
    Filter_1_Start();
    ABCF0add = 0x40340208;
    ABCF0[0] = 0x18030c43; //2k
    ABCF0[1] = 0x1c060d86; //3k
    ABCF0[2] = 0x1f090f09; //4k

    ABCF1add = 0x40340308;
    ABCF1[0] = 0x1a000104; //2k
    ABCF1[1] = 0x1d0001c7; //3k
    ABCF1[2] = 0x1a000208; //4k
}
```

```
// Step through configs
if(ConfigCount < 3)
{
    CY_SET_REG32(ABCF0add, ABCF0[ConfigCount]);
    CY_SET_REG32(ABCF1add, ABCF1[ConfigCount]);
    ConfigCount += 1;
}
```

## **Substantial in this project**

In this project, there is some auxiliary equipment to work it correctly. It would be required depending on user's application.

It left a little description about them below.

- **The programmable voltage reference (PVRef\_1)**

There is a polarity of transfer function. That would be required if it's necessary to take a care of the polarity for a project.

It would be required with positive polarity, if DC value is important for a project.

Also it must be in negative polarity, if sharp attenuation is required.

For example, the case that LPF is used for DA converter to change PWM output to Analog value, or it's required to be filtered just a specific frequency.

- **Digital Pin output**

It should be 0.5% accuracy with IMO. It's actually quite stable, but that could be decreased it till 0.02%.