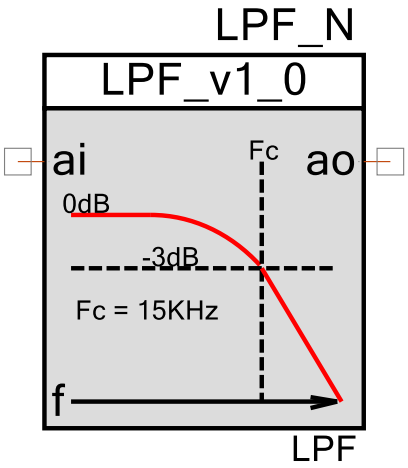
# LPF
1.0




## Features

- Analog component for a Low-Pass Filter (LPF) with 4 configurable corner frequencies and bypass.
- Run-time setting of corner frequency or bypass.

## General description

The LPF Component implements a simple interface on the PsoC5 for a settable Low-Pass Filter (LPF) for analog signal processing.

The LPF component is a PSoC fixed function analog block that is a first-order single RC circuit with 2 'R' setting values and 2 'C' setting values. This provides 4 configurable corner frequencies as shown in the table below. The corner frequency (Fc) is the frequency where the output is at 70.7% of the input. At frequencies above Fc, the output slope is at -20dB per decade.

**Table 1 - Corner Frequency and RC values**

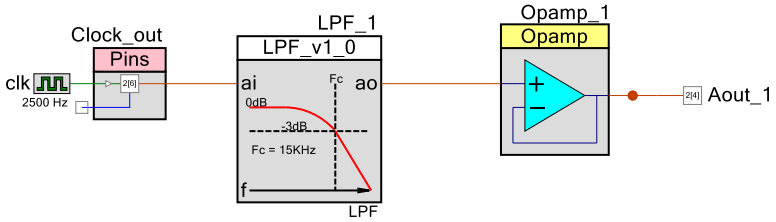
| Fc enum value | 'R' value | 'C' value | Corner Frequency (Fc) |
|---|---|---|---|
| 3 | 1040K Ω | 10pF | 153 KHz |
| 2 | 1040K Ω | 5pF | 77 KHz |
| 1 | 208K Ω | 10pF | 31 KHz |
| 0 | 208K Ω | 5pF | 15 KHz |

The LPF component can be stopped or disabled. When this occurs, the LPF is placed in BYPASS. This mode has no practical analog filtering. The entire frequency spectrum is passed from the input to the output.

**When to use LPF component**

The LPF Component was developed to attenuate analog frequencies past the selected corner frequency for further down-stream analog processing.

The LPF can also be used in combination with an Opamp in follower mode to lower the slew-rate of a digital signal. Here is a circuit example:




Use of this custom component requires the designer to configure the project to reference the component library supplied. Refer to "How to Access Custom Libraries and Components" for methods of access available.

**Device Families Supported**

At this time, the following are the PsoC devices that support this component:

- PsoC5 (Note: There are only 2 of these blocks available.)

**Implementation Limitations**

This version of the LPF Component has the following limitations:

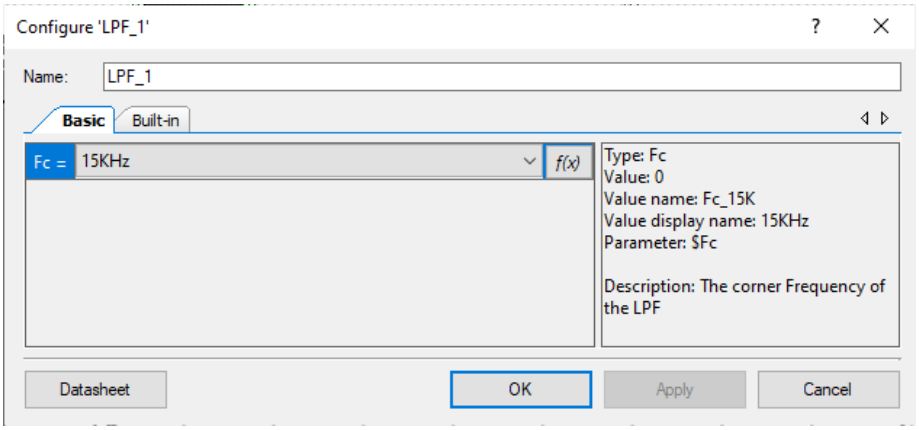
- The output impedance of the LPF is high (> 200K Ω). This means it is highly affected by additional circuit resistance and/or capacitance. Connecting the LPF output directly to a PSoC pin will affect the corner frequency by lowering it. This is because the PSoC pin capacitance ranges from 3pF to 12pF. If there are any other circuit capacitances like PCB parasitics, it will lower the corner frequency further.
  Therefore I highly recommend connecting the output of the LPF to the input of an internal Opamp. The input capacitance of the internal Opamp is <<1pF.

**Input-output connections**

LPF Component has only two analog connections (Ain and Aout).

**Parameters and Settings**

## Basic tab




This tab provides following parameters:

**Fc =**

- There are 4 settable corner frequencies as listed in Table 1 - Corner Frequency and RC values

# Application Programming Interface (API)

The API calls supported:

**Table 2 - Supported API calls**

| API call | Description |
|---|---|
| uint8 LPF_Init() | Initialization |
| void LPF_Start() | Initialization and Enable |
| void LPF_Stop() | Stop (bypass) the LPF |
| void LPF_Enable() | Enable the LPF |
| void LPF_Disable() | Disable (bypass) the LPF |
| char *LPF_GetFc_Str() | Get the current selected Fc in string form |
| uint8 LPF_SetFc() | Set the Corner Frequency (Fc) |

**uint8 LPF_Init(LPF_1_cf_e fc)**

This is the initialization function.

It is used for LPF_Start() and LPF_SetFc().

*Parameters:*
fc = corner frequency enum value.

*Returns:*
STATUS_GOOD = No error
STATUS_BAD_PARAM = Bad fc parameter

**void LPF_Start(void)**

This is the Start function that calls the initialization with the user fault Fc enum value and Enables it.

*Parameters:*
none

*Returns:*
none

**void LPF_Stop(void)**

This issues LPF_2_Disable()

*Parameters:*
none

*Returns:*
none

**void LPF_Enable(void)**

This is the Enable function sets the LPF amux to point to the LPF

*Parameters:*
none

*Returns:*
none

**void LPF_2_Disable(void)**

This is the Disable function sets the LPF amux to bypass around the LPF.

*Parameters:*
none

*Returns:*
none

**void LPF_2_GetFc_Str(void)**

This is the Disable function sets the LPF amux to bypass around the LPF.

*Parameters:*
none

*Returns:*
Pointer to char of selected corner frequency string

**uint8 LPF_SetFc(LPF_1_cf_e fc)**

This set the corner frequency Fc.

*Parameters:*
fc = corner frequency enum value.

*Returns:*
STATUS_GOOD = No error
STATUS_BAD_PARAM = Bad fc parameter

# Resources

The LPF Component utilizes the following resources.

| Function | Resource Type | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Fixed Func | Datapath Cells | Macro cells | Status Cells | Control Cells | DMA Channels | Interrupts | GPIO |
| LPF | 1 | | | | | | | |

# Sample Firmware Source Code

A simple sample project is available called “LPF_Demo_v1_0_1” using the LPF component.

# Component Changes

| Version | Description of changes | Reason for changes/impact |
|---|---|---|
| 1.0 | first release of the component | |

# References

| | |
|---|---|
| LPF_Demo_v1_0_1.pdf | |
| How to Access Custom Libraries and Components.pdf | |