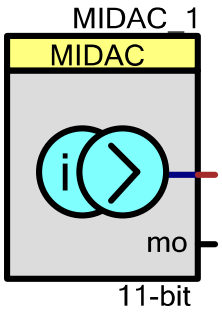
9 to 11 bit Modulated Current DAC (MIDAC)

Version 1.1

Features

- Three ranges 2047 uA, 255 uA, and 32uA
- Current source
- Selectable 9,10 or 11 bit resolution
- Use as voltage or current DAC



General Description

The MIDAC component is a 9 to 11 bit current DAC (Digital to Analog Converter). The MIDAC increases resolution of the standard 8-bit current DAC by modulating the current with a PWM. By using a fixed load resistor, this DAC can be configured for voltage output.

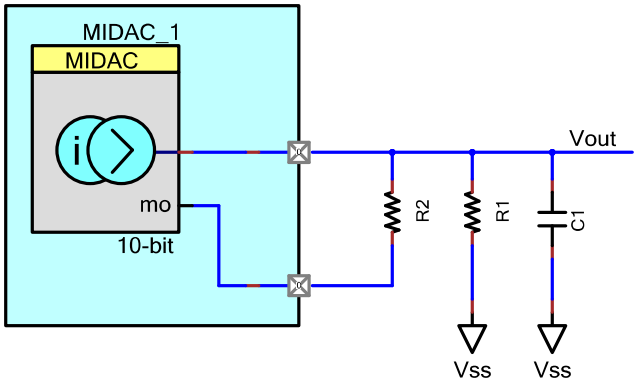


Figure 1 Usage configuration.

In the above configuration, R1 is the load resistor. To calculate the size of R1, simply divide the maximum output voltage required by the current range that is being used.

$$R1 = \frac{V\max}{I}$$

Equation 1

After R1 is calculated, R2 may also be calculated with Equation 2 below.

$$R2 = R1 * \frac{Vcc + (V\ \max/256)}{(V\ \max/256)}$$ Equation 2

Example:

If the user requires a DAC output voltage between 0 and 1.024 volts, using the 256uA current range, and has a Vcc = 3.3V solve for R1 and R2. (Using Equation 1 and 2 above.)

$$R1 = \frac{1.024V}{256uA} = 4K\Omega$$

$$R2 = 4K * \frac{3.3 + (1.024/256)}{(1.024/256)}$$ = 3.3 MΩ

To calculate the value of the filter capacitor, "C1" in figure 1, you need to know how much attenuation is required and the modulation frequency. The table below provides these values.

Resolution	9	10	11
Modulation Frequency	6 MHz	3 MHz	1.5 MHz
Attenuation	6 dB	12 dB	18 dB

First calculate the cutoff frequency of the filter.

$$F_c = \frac{F_{modulation}}{10^{\frac{Atten}{20}}}$$ Equation 3

Using the cutoff frequency "F_c" calculate C with the equation below.

$$C1 = \frac{1}{2\pi R1 F_c}$$ Equation 4

To calculate the approximate setting time, the Equation 5 below.

$$t = -\ln(\frac{0.5}{2^{B-8}}) * R1 * C1$$ Equation 5

Note: “B” is the resolution setting in the component.

When to use a MIDAC

Use the MIDAC when the standard 8-bit DACs do not provide sufficient resolution for the application.

Input/Output Connections

This section describes the various input and output connections for the MIDAC. An asterisk (*) in the list of I/O’s states that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

output – Analog

The output terminal is the current source output. It should be connected to a load or a fixed resistor if creating a voltage source. When the highest current range is selected (2048 uA), the output should only be connected to a specific set of pins that provide a direct low resistive path. The best ports to use are P0[6], P0[7], P3[0], or P3[1].

mo – Output

This output must be connected to an external resistor whose value is determined by Equation 2 above. The other end of the resistor is connected to the same node as the analog output.

Parameters

Drag a MIDAC onto your design and double-click it to open the Configure dialog.

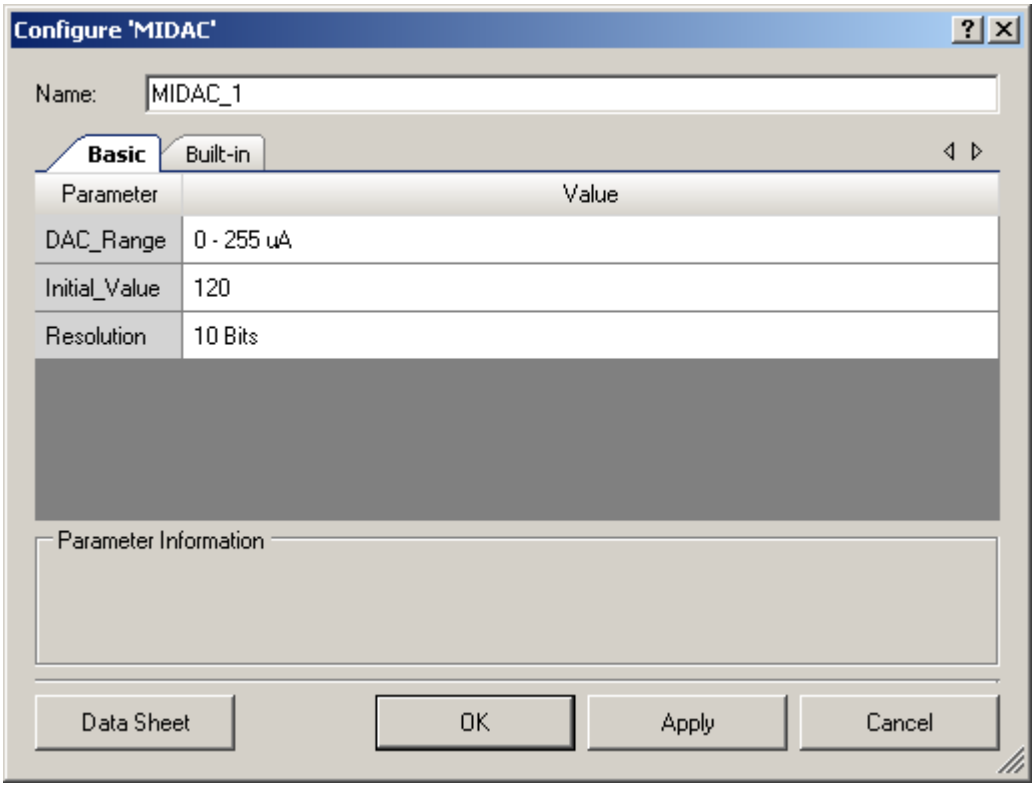


Figure 2 Configure MIDAC Dialog

The MIDAC provides the following parameters.

DAC_Range

This parameter enables the designer to set one of three current ranges as the default value. The range may be changed at anytime during runtime with the SetRange() function. If the highest current range, "0 - 2040uA" is selected, the output should be routed to one of the special pins that provide a low resistive path. These pins are P0[6], P0[7], P3[0], and P3[1].

Range	Lowest Value	Highest Value 9 / 10 / 11 bits	Step Size 9 / 10 / 11 bits
32 uA	0.0 uA	31.9 / 31.97 / 31.98 uA	62.5 / 31.25 / 15.625 nA
255 uA	0.0 uA	255.50 / 255.75 / 255.875 uA	500 / 250 / 125 nA
2047 uA	0.0 uA	2044 / 2046 / 2047 uA	4 / 2 / 1 uA

Initial_Value

This is the initial value the MIDAC will present after the Start() command is executed. The SetValue() function or a direct write to the DAC register will override the default value at anytime. Legal values are between 0 and $2^{bits} - 1$.

Resolution

The Resolution parameter allows the designer to select between 9, 10, or 11 bits of resolution.

Resources

The MIDAC consumes one UDB and one viDAC8 block and one UDB.

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name "MIDAC_1" to the first instance of a component in a given design. You can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is "MIDAC".

Function	Description
void MIDAC_Start(void)	Initialize the MIDAC with default customizer values. Enable and power up the MIDAC.
void MIDAC_Stop(void)	Disables the MIDAC and sets it to the lowest power state.
void MIDAC_SetRange(uint8 range)	Sets full scale range for MIDAC.
void MIDAC_SetValue(uint16 value)	Sets value between 0 and 2^{bits} - 1 depending on the resolution parameter.
void MIDAC_Init(void)	Initializes or restores default MIDAC configuration
void MIDAC_Enable(void)	Enables the MIDAC.
void MIDAC_SaveConfig(void)	Empty function. Provided for future usage.
void MIDAC_RestoreConfig(void)	Empty function. Provided for future usage.
void MIDAC_Sleep(void)	Stops and saves the user configuration.
void MIDAC_WakeUp(void)	Restores and enables the user configuration.

void MIDAC_Start(void)

Description: Initialize the MIDAC with default customizer values. Enable and power up the MIDAC to the given power level. A power level of 0 is the same as executing the stop function.

Parameters: None

Return Value: None

Side Effects: None

void MIDAC_Stop(void)

Description: Powers down MIDAC to lowest power state and disables output.

Parameters: None

Return Value: None

Side Effects: None

void MIDAC_SetRange(uint8 range)

Description: Sets full scale range for MIDAC.

Parameters: (uint8) range: Sets full scale range for MIDAC. See table below for ranges.

Power Setting	Notes
MIDAC_RANGE_32uA	Set full scale range to 32 uA
MIDAC_RANGE_255uA	Set full scale range to 255 uA
MIDAC_RANGE_2047A	Set full scale range to 2047 uA

Return Value: None

Side Effects: Changing range may require R1 to be changed as well for proper operation.

void MIDAC_SetValue(uint16 value)

Description: Sets value to output on MIDAC.

Parameters: (uint16) value: Maximum value is dependent on the resolution parameter

Resolution	Range
9-Bits	0 to 511
10-Bits	0 to 1023
11-Bits	0 to 2047

Return Value: None

Side Effects: None

void MIDAC_Init(void)

Description: Initializes/Restores default MIDAC configuration.

Parameters: None

Return Value: None

Side Effects: All registers will be set to their initial values. This will re-initialize the component. Calling the Init() function requires a call to SetValue() if you intend to set a new value other than what is currently in the register.

void MIDAC_Enable(void)

Description: Enables MIDAC.

Parameters: None

Return Value: None

Side Effects: None

void MIDAC_Sleep(void)

Description: Stops the operation. Saves the user configuration and the component enable state. Should be called just prior to entering sleep.

Parameters: None

Return Value: None

Side Effects: None

void MIDAC_Wakeup(void)

Description: Restores the configuration registers and component enable state. Should be called just after awaking from sleep.

Parameters: None

Return Value: None

Side Effects: Calling this function without previously calling MIDAC_Sleep() may lead to unpredictable behavior.

void MIDAC_SaveConfig(void)

Description: Saves the user configuration.

Parameters: None

Return Value: None

Side Effects: None

void MIDAC_RestoreConfig(void)

Description: Restores the user configuration.

Parameters: None

Return Value: None

Side Effects: None

Sample Firmware Source Code

The following is a C language example demonstrating the basic functionality of the MIDAC reference component. This example assumes the component has been placed in a design with the default name "MIDAC_1."

Note If you rename your component you must also edit the example code as appropriate to match the component name you specify.

<Insert meaningful and/or useful sample code here>

```
#include <device.h>

void main()
{
    MIDAC_1_Start();              // Enable MIDAC
    MIDAC_1_SetValue(500);        // Set output to 500uA in 11-bit mode
}
```

Functional Description and Block diagram

The MIDAC component requires a current DAC, a PWM, and at least one external resistor. A second external resistor can be used to simulate a voltage DAC output. Depending on how the output is used, it may be advisable to use an external resistor to filter out the noise generated by the PWM output. This noise should be less than the full scale output divided by 256. The output of the current DAC is connected to both the modulation resistor (R1) and the load (R2) in parallel. See Figure 4 below for the schematic. With resistor R2 connected to the DAC's output and Vss, the current DAC operates as a voltage DAC. The second resistor connected to the PWM must be much larger than the resistor connected to ground. The PWM acts to modulate the load by less than the value of the DACs lsb. The PWM's clock operates at 12 MHz, but the period is dependent on the resolution setting.

Resolution	Clock Frequency	Period	Dither Frequency
9	12 MHz	2	6 MHz
10	12 MHz	4	3 MHz
11	12 MHz	8	1.5 MHz

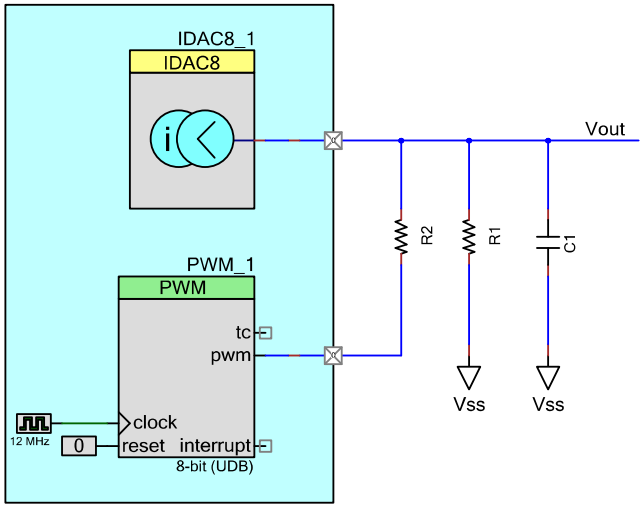


Figure 3 MIDAC schematic

Global Variables

Variable	Description
MIDAC_initVar	Indicates whether the MIDAC has been initialized. The variable is initialized to 0 and set to 1 the first time MIDAC_Start() is called. This allows the component to restart without reinitialization after the first call to the MIDAC_Start() routine. If reinitialization of the component is required, then the MIDAC_Init() function can be called before the MIDAC_Start() or MIDAC_Enable() function.

Registers

The functions provided support most of the common runtime functions that are required for most applications. The register reference below provides a brief description for the advanced user.

Table 1 MIDAC_CR0

Bits	7	6	5	4	3	2	1	0
Value	reserved			mode	Range[1:0]		hs	reserved

- mode: Sets DAC to either voltage or current mode.
- range[1:0]: DAC range settings.

- hs: Use to set data speed.

Table 3 MIDAC_CR1

Bits	7	6	5	4	3	2	1	0
Value	reserved		mx_data	reset_udb_en	mx_idir	idirbit	Mx_ioff	ioffbit

- mx_data: Select data source.
- reset_udb_en: DAC reset enable.
- mx_idir: Mux selection for DAC current direction control.
- idirbit: Register source for DAC current direction.
- mx_off: Mux selection for DAC current off control.
- ioffbit: Register source for DAC current off

Table 5 MIDAC_Data

Bits	7	6	5	4	3	2	1	0
Value 9-Bit	Data[8:1]							
Value 10-Bit	Data[9:2]							
Value 11-Bit	Data[10:3]							

- 9-Bit Mode Data [8:1]: DAC data register
- 10-Bit Mode Data [9:2]: DAC data register
- 11-Bit Mode Data [10:3]: DAC data register

Table 5 MIDAC_Dither_PWM_Period_LSB

Bits	7	6	5	4	3	2	1	0
Value 9-Bit	0	0	0	0	0	0	0	1
Value 10-Bit	0	0	0	0	0		1	1
Value 11-Bit	0	0	0	0	0	1	1	1

Table 5 MIDAC_Dither_PWM_Compare1_LSB

Bits	7	6	5	4	3	2	1	0
Value 9-Bit	0	0	0	0	0	0	0	Data[0]
Value 10-Bit	0	0	0	0	0	0	Data[1:0]	
Value 11-Bit	0	0	0	0	0	Data[2:0]		

- 9-Bit Mode Data [0]: DAC data register
- 10-Bit Mode Data [1:0]: DAC data register
- 11-Bit Mode Data [2:0]: DAC data register

DC and AC Electrical Characteristics

The following values are indicative of expected performance and based on initial characterization data.

DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
	Output current					
Iout	High	Vdda ≤ 2.7 V, RL 300 Ω	-	-	2.047	mA
	Medium	RL 600 Ω	-	-	255	µA
INL	Integral non linearity	RL 600 Ω, CL=15 pF , 9-bit	-	± 0.6	-	LSB

		RL 600 Ω, CL=15 pF , 10-bit	-	± 1.0	-	LSB
		RL 600 Ω, CL=15 pF , 11-bit		± 2	-	LSB
DNL	Differential non linearity	RL 600 Ω, CL=15 pF , 9-bit	-	0.35	-	LSB
		RL 600 Ω, CL=15 pF , 10-bit	-	0.6	-	LSB
		RL 600 Ω, CL=15 pF , 11-bit	-	1.0	-	LSB
Ezs	Zero scale error		-	0	±1	LSB
Eg	Gain error	Uncompensated	-	-	2.5	%
		Temperature compensated	-	-	TBD	%
IDAC_ICC	DAC current low speed mode	Code = 0	-	-	200	µA
IDAC_ICC	DAC current high speed mode	Code = 0	-	-	1000	µA

AC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
Fdac	Update rate		-	-	4	Msps
Tsettle	Settling time to 0.5LSB	Full scale transition, 600 Ω load, CL = 15 pF				
	Fast mode	Independent of IDAC range setting (Iout)	-	-	100	ns
	Slow mode	Independent of IDAC range setting (Iout)	-	-	1000	ns

Reference Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes
1.0	First release
1.1	Changed library tab to Concept