

TI C2000 Toolbox Components overview

This document describes how library components are organized and provides explanation on what features each component category has.

TI C2000 Setup

[This component](#) is needed in every model that relies on TI C2000 Toolbox library. It enables the user to choose a target platform (device), create a CCS project, build and flash the control code and configure the system settings. Detailed description can be found in the [dedicated document](#). **Important:** It is recommended to select *target platform* before configuring peripheral components since component mask appearance and many available options may depend on selected platform.

Peripheral configuration components

There are two ways to configure the peripherals of the target MCU:

1. Target-oriented approach
2. Application-oriented approach

Target-oriented approach

With target-oriented approach, user configures the peripherals similarly to how it would be done through manual coding, meaning that user would configure the register values directly and interact with the peripherals looking from the target MCUs perspective. For example, if ePWM peripheral is considered, user's input will be directly mapped onto the counter period. If GPIO DO/DI peripheral is considered, user will choose the number of the GPIO pin to manipulate/read.

This approach is well suited for users that have experience in manual programming of the target platform and would like to maintain full control of the peripheral configuration, along with full flexibility that target platform offers.

Components that offer target-oriented approach are named the same way the peripheral is named in manufacturer's resources: ADC, ePWM, eQEP, etc.

Components that allow for this type of peripheral configuration are:

- [GPIO DO](#)
- [GPIO DI](#)
- [ADC](#)
- [ePWM](#)
- [eQEP](#)

Application-oriented approach – Generic components

Similarly to components that aim for Target-oriented approach, Application-oriented components enable the user to configure the target peripherals. However, these components allow the user to focus on application of interest, which is exemplified in

two categories of tasks that are automatically done by the component, that would otherwise be required to be done by the control developer:

1. Register fields are hidden from the users and peripheral is configured through simplified set of properties. For example:
 - Instead of configuring TBPRD register (Time-Base Period of ePWM), user can simply select the switching frequency. Value of TBPRD register is calculated in background based on clock frequency of selected device and desired switching frequency.
 - Instead of configuring the DBCTL register (Dead-Band Generator Control), user can enter the desired dead-time duration. Again, clock frequency of selected device is used to calculate value of edge delay registers.
 - Instead of manually configuring the SOC (start of conversion) for ADCs, these are automatically configured to be triggered by the selected trigger source.

NOTE: ePWM peripheral clock is set to be the maximum achievable clock for selected device in [TI C2000 Setup](#) component.

2. Interface between HIL device and MCU is taken into consideration when configuring the peripherals that have multiple modules, thus meaning that the user can select the HIL pin number, and is not required to consult the interface board documentation whenever new peripheral module is added to the control:
 - Instead of choosing the ADC pin, user selects the HIL analog output pin. Component will automatically choose the ADC module and channel that is connected to this pin.
 - Instead of choosing PWM module, component will provide the user with combo widget that shows the HIL digital inputs that are connected to the available PWM channels.
 - If GPIO peripheral is used, user can select the HIL DI/DO pin.

NOTE: It is recommended to select interface type before choosing target HIL pin since pin availability and connection routing may change.

Application-oriented components are well suited for users that are not familiar with low-level configuration of target MCU, as well as users that are looking for ultra-rapid prototype development, before proceeding with more detailed peripheral configuration via target-oriented components or by coding manually. Components that provide the user with application-oriented approach have (*Generic*) in their name, suggesting that they are configured in a generic manner, using properties that could be associated with any peripheral that provides the functionality in question. Generic components that are currently available in the library are:

- [GPIO DO \(Generic\)](#)
- [GPIO DI \(Generic\)](#)
- [ADC \(Generic\)](#)

- [ePWM \(Generic\)](#)
- [SCI Setup \(Generic\)](#)
- [SCI Receive \(Generic\)](#)
- [SCI Send \(Generic\)](#)

Auxiliary components

There is a separate category in component library called *Aux* with components that support and extend the overall functionality of the toolbox. While not central to the main operation, these components can play a very useful role in configuration, integration, and overall system usability.

Two components are available:

- [State reset](#)
- [User code](#)
- CPU Utilization – outputs time required to execute *step* code of specified *execution rate* relative to the *execution rate* (period) of the task.