

# TI C2000 Toolbox How to blink an LED?

This document demonstrates the minimum required workflow in order to deploy the control code to the target platform using Typhoon [Schematic Editor](#), using the LED blinking as an example.

## Install the TI C2000 Toolbox package

TI C2000 Toolbox is available in a form of Typhoon package. To see how packages are installed, see [Package Manager](#). Also, please view the [First time setup document](#).

## Build and deploy program

This section provides step-by-step guide on how to build and deploy the desired control code to the target platform. Once the package is installed, TI C2000 Toolbox library is available and it can be seen among [user-defined libraries](#) (Figure 1).

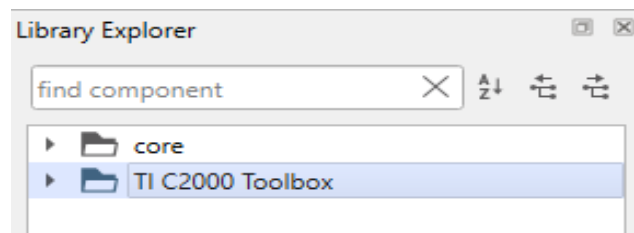
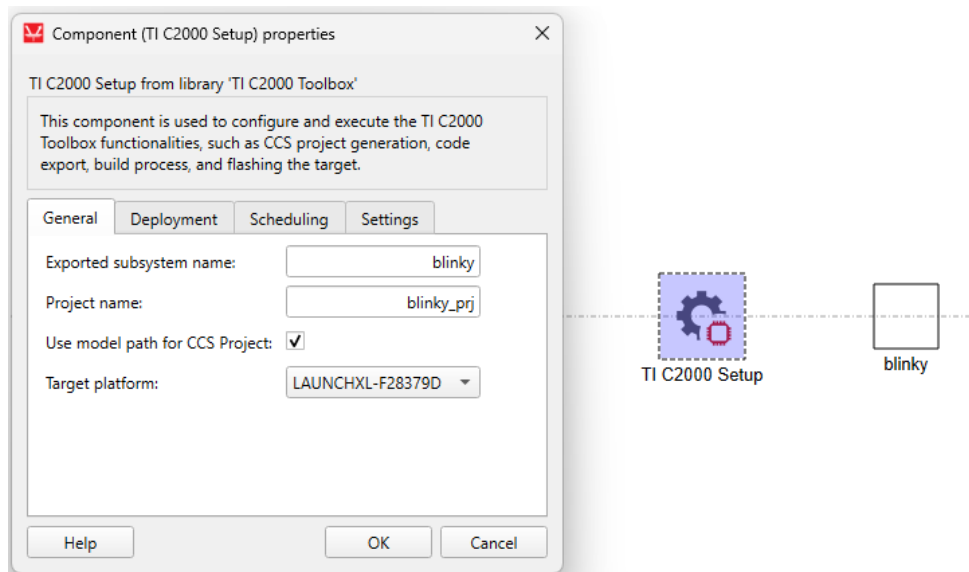


Figure 1 TI C2000 Toolbox in Schematic Editor Library Explorer

### Prepare model for deployment

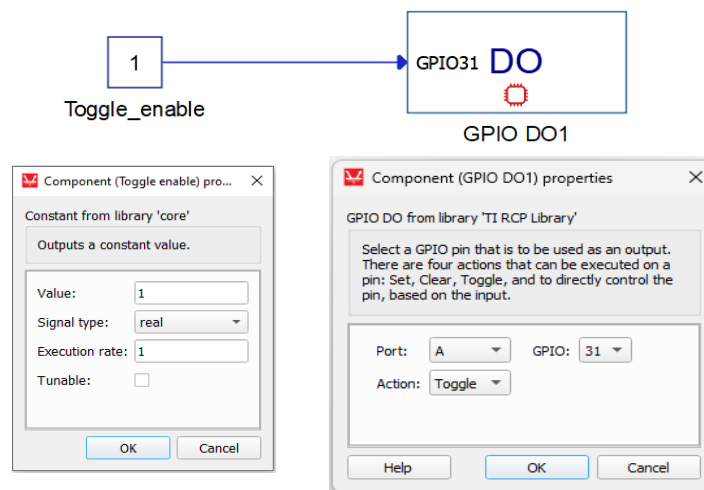
First, [TI C2000 Setup](#) component must be placed and configured on the schematic root. It is important to provide correct **target platform before** configuring other components in the schematic further. Once exported subsystem, project name and target platform are configured, it is suitable to build the schematic. In order to export the code, all of the components need to be encapsulated into a single [subsystem](#).



**Figure 2 Model prepared for export**

### Build the schematic

To blink an LED, we can use toggle command, available in [GPIO DO](#) component. MCUs digital output will be toggled in each code execution step. Therefore, if goal is to toggle an LED at 1Hz frequency, execution rate of constant component should be 1s. [LAUNCHXL-F28379D](#) development kit connects GPIO DO pin 31 to a blue LED that is visible on the LaunchPad.



**Figure 3 Toggle pin GPIO31**

Since this pin is not connected with HIL device, [GPIO DO](#) component is used.

## Scheduling

Once schematic is configured, it is necessary to choose a source for scheduling. For each given *execution rate* in the schematic, separate task is defined. Pre-generated *scheduler* is provided with the CCS project which dictates the task execution at correct rate. An interrupt source needs to be provided to the scheduler to execute the tasks. By default, CPU timer invokes the scheduler on fastest execution rate and takes into account all the slower execution rate tasks that will be executed on multiples of the fastest one.

For this example, simple *CPU Timer Based Scheduler* is used.

**IMPORTANT NOTE:** In case of other non-default scheduling sources, it is important to configure a peripheral component in such way it would provide an interrupt to the scheduler at expected execution rate (for example, if ADC interrupt is used, sampling needs to be triggered and *end-of-conversion* interrupt generated at the fastest execution rate).

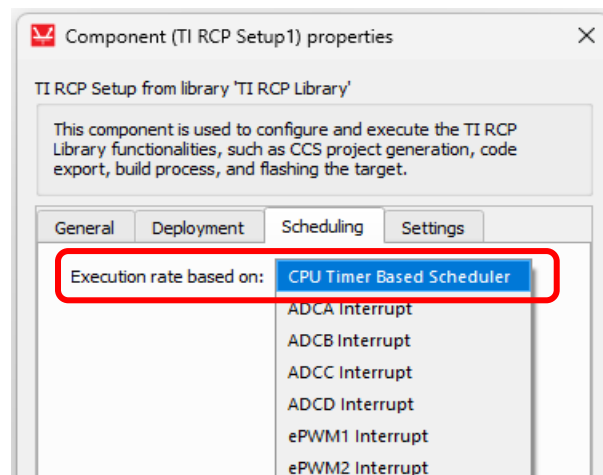
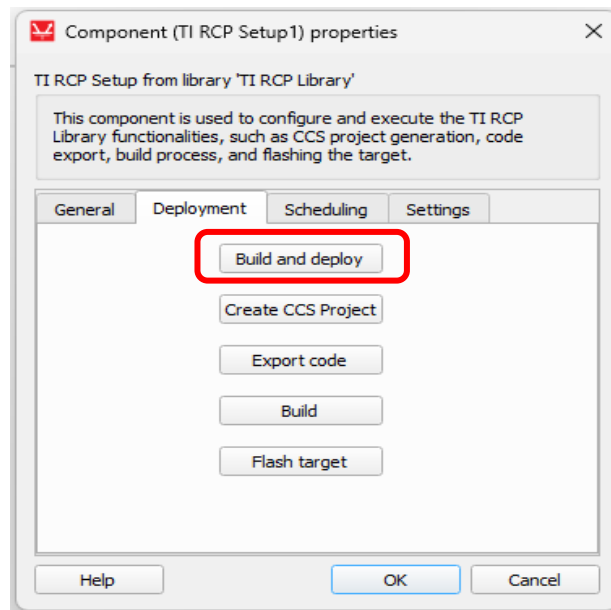


Figure 2 Scheduling source

## Build and deploy

Once [TI C2000 Setup](#) properties and schematic are configured, *Build and deploy* button is pressed. After the target is flashed, blinking LED will be visible on the launchpad.



**Figure 5 Build and deploy button**