

# TI C2000 Toolbox SCI Send (Generic)

This document describes *SCI Send (Generic)* component from TI C2000 Toolbox library.

## Short description

*SCI Send (Generic)* component enables user to configure data transmitting operation of the *Serial Communication Interface* peripheral. It is used to specify the type of connection with the receiver and message configuration and scheduling.

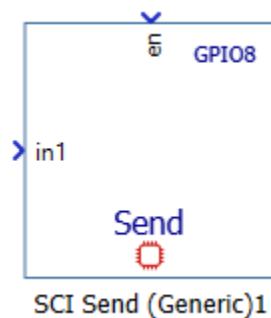


Figure 1. SCI Send (Generic) - component icon.

## Detailed overview

**NOTE:** It is recommended to select *target platform* on [TI C2000 Setup](#) component before configuring the component.

### Component properties:

- Tab **General**:
  - Access port – specifies type of connection:
    - **DIO** (digital inputs/outputs) – communication is executed between digital inputs of the HIL device and digital outputs of controller board header pins. In this case, serial communication must be configured on the HIL side as well (HIL is a receiver), by configuring [Serial Setup](#) and [Serial Receive](#) components in the model, outside of exported subsystem. It is not required to have an USB connection to the controller after flashing the code.
    - **USB** – communication is executed between controller USB connector and HIL SCADA (the PC application, not the HIL device). In this case, serial communication must be configured on the HIL SCADA side as well (SCADA is a receiver), by configuring [SCI Setup widget](#) and [SCI Read widget](#) on SCADA panel. It is required to have an USB connection with the controller.

- Receiver (HIL device or HIL SCADA) and transmitter (controller) must have the same setup parameters (such as connection type, baud rate, parity, stop bits, data bits) and start-of-frame, end-of-frame, message configuration settings – for both types of connections!
  - Receiving HIL DI number – visible when *Access port* is set to *DIO*. Specifies the number of HIL digital input to which the controller will send messages. It lists all available HIL digital inputs that are connected to the digital outputs of the controller that have SCI mux option as transmitter (TX). Also, the same HIL DI must be selected in [Serial Setup](#) (property *UART rx DI pin*).
  - Interface type - select interface board that is used, currently supported boards are '[HIL TI Launchpad Interface](#)' and '[HIL TI uGrid Launchpad Interface](#)', '[HIL DSP 180 Interface](#)' and '[HIL DSP Interface](#)'. Visible when *Access port* is set to *DIO*.
  - Controller index - visible when '[HIL TI uGrid Launchpad Interface](#)' is selected and *Access port* is set to *DIO*. Specifies which MCU slot on the interface board is used. It is dependent on *Interface type* property value and *Target platform* selected in *TI C2000 Setup* component.
  - Enable active TX input – enables additional input for a signal that will enable data transmit by applying a value greater than zero
  - Sending period – additional timing delay between two consecutive message payloads, in milliseconds. If *Serial Send* is used to perform serial communication with HIL SCADA via **USB** connection, the value should be set to at least 250 milliseconds, since that is the “execution rate” of SCADA widgets that read the received messages from the MCU. In this way, an overflow of received messages from the MCU is avoided.
  - Execution rate – desired rate at which the controller will send data. This value must be compatible with other components of the same subsystem: the value must be a multiple of the fastest execution rate in the circuit. To specify the execution rate, you can use either decimal (e.g. 0.001) or exponential values (e.g. 1e-3) in seconds. The aspect of limited baud rate and amount of data specified for exchange must be considered while determining execution rate. Typically, an execution rate of 1 ms should be sufficient.
- **Tab Data:**
    - Start of frame – bytes used for start of frame which is sent before the message payload and is used for synchronization,
    - End of frame - bytes used for end of frame which is sent after the message payload and is used for synchronization,
    - Message configuration - Defines the message payload for transmitting operation. It specifies message configuration that controller will send. For more information on how to configure the payload check the [Message](#)

[Configuration](#) section (from the documentation for core serial communication components).

**NOTE:** Data escape option is not available in SCI Send (Generic) component! Hence, if [Serial Receive](#) component is used on receiver side, this option must be disabled.

#### Component inputs:

- in1 – for each message defined in *Message configuration* property separate input is created and named after the message. Dimension is also defined separately for each output in *Message configuration*.
  - Supported types: uint, int and real,
  - Vector support: yes, if *Access port* is set to *DIO*, otherwise no.
- en – exists if *Enable active TX input* is set to *True*. Input signal that enables the component to transmit data. If this input is set to '0', the controller does not transmit. For values higher than '0', the controller executes transmitting operation.
  - Supported types: uint, int and real,
  - Vector support: no.

**NOTE:** For utilizing serial communication between *HIL SCADA* and controller (through USB connection between PC and controller – if *Access port* is set to **USB**), a SCADA counterpart is available in form of [SCI Read widget](#). *SCI Read* widget writes messages defined by *SCI Send (Generic)* component to COM port to which the controller is connected. Number of [SCI Read widgets](#) should match number of messages in *Message configuration* property. **Dimension of every message must be 1 in this case!** *Start of frame*, *End of frame* and *Message configuration* properties should be copied to [SCI Setup widget](#) properties. In this case, target development board **must be configured** to enable the serial communication through USB connection. Details on how to do this can be found [here](#).

**NOTE:** In context of serial communication between HIL device and controller (through digital pins – if *Access port* is set to **DIO**), HIL transmits messages to the controller, hence [Serial Receive](#) component must be configured properly in the model (outside of the exported subsystem) – it must have the same *Start of frame*, *End of frame* and *Message configuration* properties. Also, the same HIL DI number must be used by both sides (specified in *Receiving HIL DI number* property of SCI Receive component and *UART rx DI pin* property of [Serial Setup](#) component).

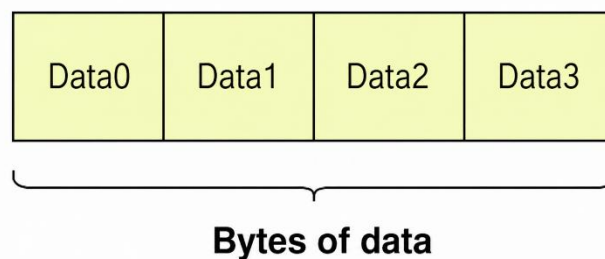
## . Backward incompatibility resolution

With *TI C2000 Toolbox* version 1.0.0 *SCI Send (Generic)* suffered major refactoring. Many new properties are introduced (see [Component properties](#) section) and as a result, models containing this component might have some backward incompatible

changes. In case of *SCI Send (Generic)* component, the following changes might affect old models:

- **New input port:** *en* (see *Component inputs* section). Since all inputs to the components must be connected to successfully generate the code, it is required to provide an enabling signal to *SCI Send (Generic)* component. Best practice would be to apply some signal that is aligned with the simulation start (such as the sampling of HIL digital output whose state changes on simulation start) – see example models. It helps avoid TX buffer overflow.
- **Message configuration changed:** Before 1.0.0 version, all messages defined in *SCI Send (Generic)* component had fixed configuration:
  - Data type: real 32-bit/4-byte float,
  - Dimension: 1,
  - Endianness: little endian

For example, if message payload was defined with one signal (input to the component), sent bytes were arranged like on Fig2:



*Figure 2. Old message configuration.*

On new version, message payload can be customized with several parameters, (such as data type, dimension, endianness, etc.), as described [here](#).

Now, message payload can have different configurations, as described [here](#).