S/C - Sin/cos encoder

SD103 | Posted on November 8, 2021 | Updated on May 27, 2025

Simon STROBL
Product Director
imperix • in

Table of Contents

The sin/cos encoder (S/C) block retrieves the Sine and Cosine signals of a sin/cos encoder connected to the Motor Interface.

The B-Box RCP supports up to two sin/cos encoders through the Motor Interface for B-Box RCP. This type of sensor encodes the position of the rotor using two signals in quadrature (sin and cos). The principle is similar to an incremental encoder: the signals in quadrature are periodic, and the sensor produces a fixed number of periods per revolution (PPR). Unlike an incremental encoder, signals in quadrature are analog, which allows computing the angle within one electrical period of the sensor. As a result, a sin/cos encoder offers a better resolution at the same PPR. Additionally, some sin/cos encoders also provide an index signal (ns) equivalent to the Z reset signal of an incremental encoder. The Motor Interface does not support absolute sin/cos encoders.

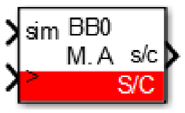
Version 3.7 beta of the ACG SDK for Simulink does not implement an angle decoder. Instead, the S/C block provides the raw analog signals of the sensor read by some ADCs, and the control must implement angle decoding.

The S/C block is available starting from version 3.7.1.4 of the SDK. The Motor Interface for B-Box RCP is **required** to use this driver.

Simulink block

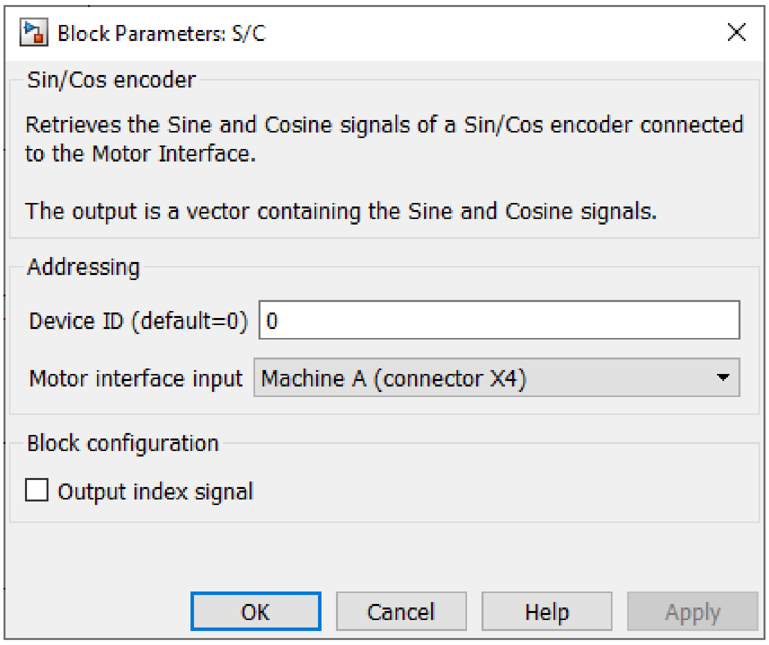
Signal specification

- The output `s/c` is a vector containing the Sine and Cosine signals in [V].
- The optional output `ns` corresponds to the index signal.
- The `sim` input is used in simulation and represents the actual Sine and Cosine signals, computed by the simulation plant model.
- The `>` input signal needs to be connected to the sampling clock generated by the CONFIG block to account for the exact sampling instant in simulation.



Parameters

- `Device ID` selects which B-Box/B-Board to address when used in a multi-device configuration.
- `Motor Interface input` selects which connector of the Motor Interface is used as an input.
- `Output index signal` defines if the index signal is output or not.


Block Parameters: S/C

Sin/Cos encoder

Retrieves the Sine and Cosine signals of a Sin/Cos encoder connected to the Motor Interface.

The output is a vector containing the Sine and Cosine signals.

Addressing

Device ID (default=0) 0

Motor interface input Machine A (connector X4)

Block configuration

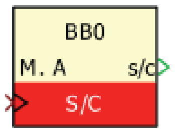
☐ Output index signal

PLECS block

Signal specification

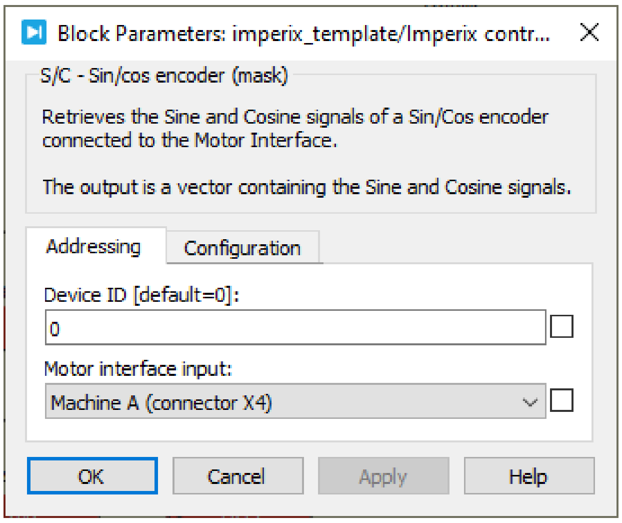
- The output `s/c` is a vector containing the Sine and Cosine signals in [V].
- The optional output `ns` corresponds to the index signal.
- The `sim` input is used in simulation and represents the actual Sine and Cosine signals, computed by the simulation plant model.

- The > input signal needs to be connected to the sampling clock generated by the CONFIG block to account for the exact sampling instant in simulation.



Parameters

- `Device ID` selects which B-Box/B-Board to address when used in a multi-device configuration.
- `Motor Interface input` selects which connector of the Motor Interface is used as an input.
- `Output index signal` defines if the index signal is output or not.



Block Parameters: imperix_template/Imperix contr...

S/C - Sin/cos encoder (mask)

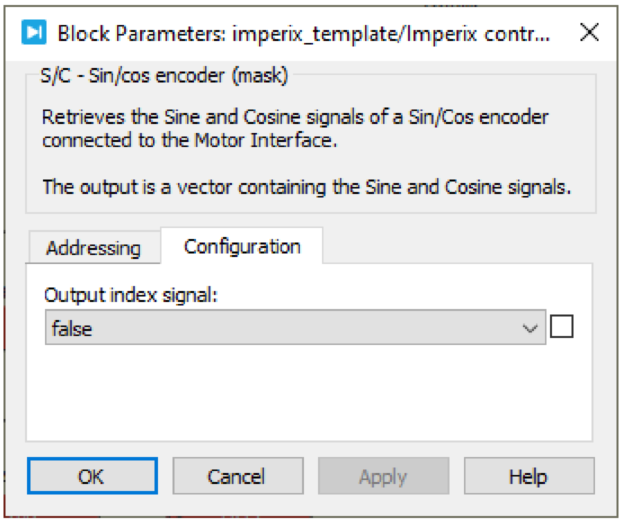
Retrieves the Sine and Cosine signals of a Sin/Cos encoder connected to the Motor Interface.

The output is a vector containing the Sine and Cosine signals.

Addressing | Configuration

Device ID [default=0]: 0

Motor interface input: Machine A (connector X4)



Block Parameters: imperix_template/Imperix contr...

S/C - Sin/cos encoder (mask)

Retrieves the Sine and Cosine signals of a Sin/Cos encoder connected to the Motor Interface.

The output is a vector containing the Sine and Cosine signals.

Addressing | Configuration

Output index signal: false

C++ functions

`MotInt_EnableMotorInterface` **— Enable the drivers of the Motor Interface**

```cpp
void MotInt_EnableMotorInterface(unsigned int device=0);
```
Code language: C++ (cpp)

Enables the drivers of the Motor Interface.

It has to be called in `UserInit()`.

Parameters

- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Sc_GetSin` — Read the sine signal

```cpp
float Sc_GetSin(tMotIntMachine machine, unsigned int device=0);
```
Code language: C++ (cpp)

Read the value of the sine signal.

It has to be called during the control interrupt.

Parameters

- `machine`: the machine to configure (*MACHINE_A* or *MACHINE_B*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Sc_GetCos` — Read the cosine signal

```cpp
float Sc_GetCos(tMotIntMachine machine, unsigned int device=0);
```
Code language: C++ (cpp)

Reads the value of the cosine signal.

It has to be called during the control interrupt.

Parameters

- `machine`: the machine to configure (*MACHINE_A* or *MACHINE_B*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Sc_GetNs` — Read the index signal

```cpp
float Sc_GetNs(tMotIntMachine machine, unsigned int device=0);
```
Code language: C++ (cpp)

Reads the value of the index signal.

It has to be called during the control interrupt.

Parameters

- `machine`: the machine to configure (*MACHINE_A* or *MACHINE_B*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).