INC - Incremental encoder input

SD100 | Posted on November 8, 2021 | Updated on May 27, 2025

Simon STROBL
Product Director
imperix

Table of Contents

- Simulink block
  - Signal specification
  - Parameters
- PLECS block
  - Signal specification
  - Parameters
- C++ functions

The incremental encoder interface (INC) block decodes the A, B, Z signals from an incremental encoder for motor drive applications.

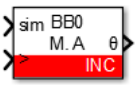
The B-Box supports up to two differential incremental encoders through the Motor Interface for B-Box RCP. These devices provide two signals in quadrature (usually called A and B), as well as an optional reset line (usually called Z). The decoder module counts all 4 edges of the A and B inputs, leading to an angular resolution 4 times superior to the PPR value usually specified for a given encoder. The position counter can be reset either at a specified value or using the Z signal provided by the sensor. Finally, the position is latched similarly and simultaneously to the sample-&-hold feature of the ADC inputs.

The INC 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. Alternatively, the B-Box RCP and the B-Board Pro also support incremental encoders directly through their GPI pins. In this case, please refer to the Angle decoder module (DEC).

Simulink block

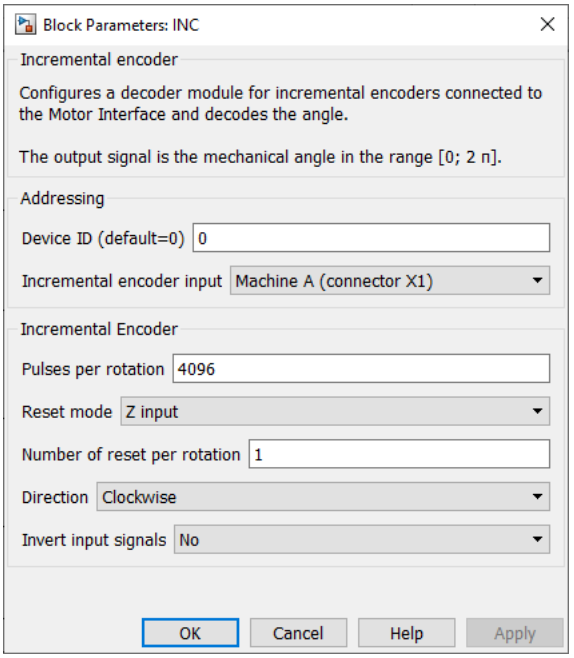
Signal specification

- The output signal is the mechanical angle in the range [0; 2π].
- The sim input signal is used in simulation and represents the actual angle value in radian, computed by the simulation plant model.
- The > input signal needs to be connected to the CONFIG block to account for the exact sampling instant in simulation.



Parameters

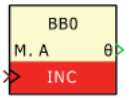
- `Device ID` selects which B-Box/B-Board to address when used in a multi-device configuration.
- `Incremental encoder input` selects which connector of the Motor Interface is used.
- `Pulses per rotation` defines the number of pulses of the A or B signal during one complete rotation of the incremental encoder, as given by the manufacturer.
- `Reset Mode` selects the counter reset mode. If *Z input* is selected, the pulse counter value is reset on the rising edge of the Z signal. If *maximum value* is selected, it is reset as soon as it has reached the number of pulses per rotation.
- `Number of pulse per rotation` configures the number of Z pulses per complete rotation of the encoder, as given by the manufacturer. It can only be used if the *ZINPUT* reset mode is selected.
- `Direction` defines if an increasing angle corresponds to a *clockwise* or *counterclockwise* rotation.
- `Invert input signals` configures the decoder to consider the inverted logical value of the considered digital inputs.



PLECS block

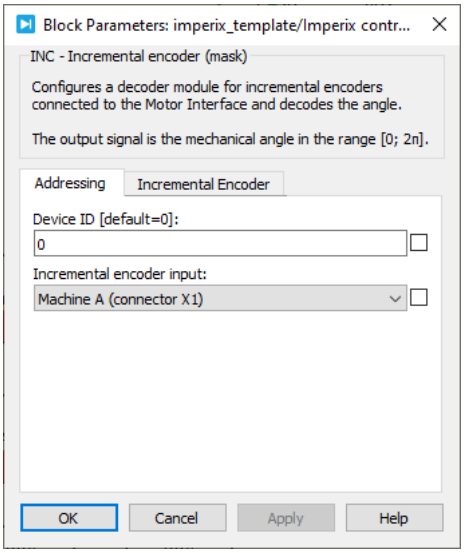
Signal specification

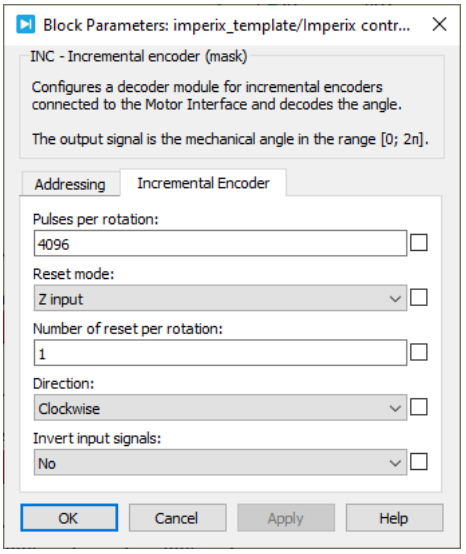
- The output signal is the mechanical angle in the range [0; 2π].
- The sim input signal is used in simulation and represents the actual angle value in radian, computed by the simulation plant model.
- The > input signal needs to be connected to the CONFIG block to account for the exact sampling instant in simulation.



Parameters

- `Device ID` selects which B-Box/B-Board to address when used in a multi-device configuration.
- `Incremental encoder input` selects which connector of the Motor Interface is used.
- `Pulses per rotation` defines the number of pulses of the A or B signal during one complete rotation of the incremental encoder, as given by the manufacturer.
- `Reset Mode` selects the counter reset mode. If *Z input* is selected, the pulse counter value is reset on the rising edge of the Z signal. If *maximum value* is selected, it is reset as soon as it has reached the number of pulses per rotation.
- `Number of pulse per rotation` configures the number of Z pulses per complete rotation of the encoder, as given by the manufacturer. It can only be used if the *ZINPUT* reset mode is selected.
- `Direction` defines if an increasing angle corresponds to a *clockwise* or *counterclockwise* rotation.
- `Invert input signals` configures the decoder to consider the inverted logical value of the considered digital inputs.





C++ functions

`MotInt_EnableMotorInterface` — Enable the drivers of the Motor Interface

```
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).

`Inc_ConfigurePulsePerRotation` — Set the number of pulses per rotation

```
void Inc_ConfigurePulsePerRotation(tMotIntMachine machine, unsigned int pulsePerRotation, unsigned int device=0);C
```

Defines the number of pulses of the A or B signal during one complete rotation of the incremental encoder, as given by the manufacturer.

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

Parameters

- `machine`: the machine to configure (*MACHINE_A* or *MACHINE_B*).
- `pulsePerRotation`: the number of pulses of the A or B signal during one complete rotation of the encoder.
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Inc_ConfigureResetMode` — Set the counter reset mode

```
void Inc_ConfigureResetMode(tMotIntMachine machine, tIncResetMode resetMode, unsigned int device=0);Code language: C+
```

Defines if the pulse counter is reset on the rising edges of the Z signal, or when it reaches the PPR value.

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

Parameters

- `machine`: the machine to configure (*MACHINE_A* or *MACHINE_B*).
- `resetMode`: the reset method of the pulse counter (*ZINPUT* or *MAXVALUE*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Inc_ConfigureResetPerRotation` — Set the number of Z pulses per rotation

```
void Inc_ConfigureResetPerRotation(tMotIntMachine machine, unsigned int resetPerRotation, unsigned int device=0);C
```

Configures the number of Z pulses per complete rotation of the incremental encoder, as given by the manufacturer. It can only be used if the *ZINPUT* reset mode is selected.

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

Parameters

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

`Inc_ConfigureDirection` — Set the rotation direction

```
void Inc_ConfigureDirection(tMotIntMachine machine, tMotIntDirection direction, unsigned int device=0);Code language
```

Configures what rotation direction leads to an increasing angle (positive direction). If clockwise is selected, the angle is increased when A leads B.

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

Parameters

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

`Inc_ConfigureInputPolarity` — Set the polarity of the input signals

```
void Inc_ConfigureInputPolarity(tMotIntMachine machine, tIncPolarity polarity, unsigned int device=0);Code language:
```

Defines the polarity of the incremental encoder signals. If inverted is selected, the decoder considers the inverse of the input signal.

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

Parameters

- `machine`: the machine to configure (*MACHINE_A* or *MACHINE_B*).
- `polarity`: the encoder signals polarity, normal (*NORM)* or inverted (*INV*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Inc_GetAngle` — Get the mechanical angle

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

Returns the value of the mechanical position angle in the range [0; 2π].

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).