DEC - Angle decoder input

SD007 | Posted on April 2, 2021 | Updated on May 27, 2025

Benoît STEINMANN
Software Team Leader
imperix

Table of Contents

The angle decoder (DEC) block decodes quadrature-encoded signals produced by incremental encoders for motor drive applications.

The B-Box RCP and B-Board PRO provide decoder inputs for quadrature-encoder speed/position sensor signals (usually called A and B), with or without a reset line (usually called Z). These inputs are either configurable as four independent inputs or two differential inputs. Each 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.

Performance specifications are available in the B-Box datasheet and in the B-Board datasheet. Alternatively, the Motor Interface for B-Box RCP also supports incremental encoders. In this case, please refer to the incremental encoder module (INC).

Information on how to use this block is available in Using the angle decoder modules (PN104).

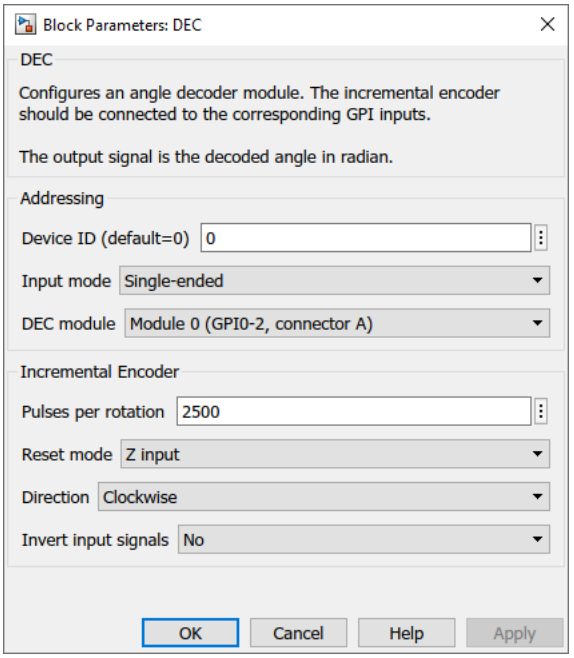
Simulink block

Signal specification

- The output signal is the decoded angle in radians.
- 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

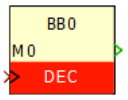
- `Device ID` selects which B-Box/B-Board to address when used in a multi-device configuration
- `Input mode` selects if the decoder considers one GPI input per encoder signal (*Single-ended*) or two GPI with complementary signals (*Differential*).
- `DEC module` selects which GPI inputs are used. See the B-Box datasheet or B-Board datasheet to check where to connect the encoder signals.
- `Pulses per rotation` defines the number of pulses of the A or B signal during one complete rotation of the 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.
- `Direction` defines, when A is leading B, if the angle increase (*Clockwise*) or decreases (*Counterclockwise*).
- `Invert input signals` configures the decoder to consider the inverted logical value of the considered GPI inputs.



PLECS block

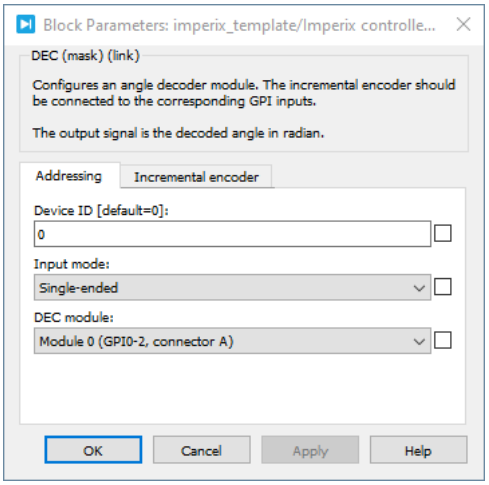
Signal specification

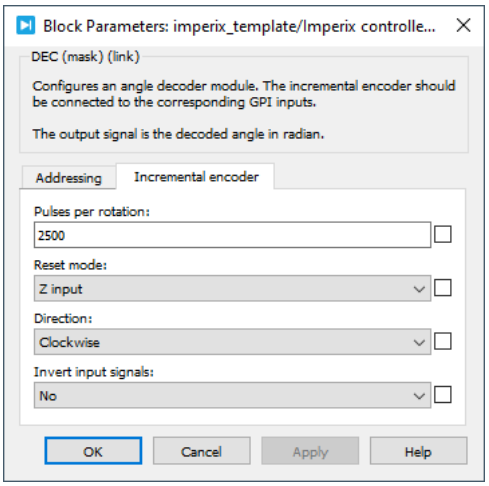
- The output signal is the decoded angle in radians.
- The target inport (only visible at the atomic subsystem level) is used in simulation and represents the actual angle value in radian, computed by the simulation plant model.
- The > input signal needs to be connected to the ADC output of the CONFIG 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
- `Input mode` selects if the decoder considers one GPI input per encoder signal (*Single-ended*) or two GPI with complementary signals (*Differential*).
- `DEC module` selects which GPI inputs are used. See the B-Box datasheet or B-Board datasheet to check where to connect the encoder signals.
- `Pulses per rotation` defines the number of pulses of the A or B signal during one complete rotation of the 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.
- `Direction` defines, when A is leading B, if the angle increase (*Clockwise*) or decreases (*Counterclockwise*).
- `Invert input signals` configures the decoder to consider the inverted logical value of the considered GPI inputs.





C++ functions

`Dec_ConfigureInputMode` **— Select between single-ended or differential inputs**

```
void Dec_ConfigureInputMode(tDecChannel channel, tDecInputMode inputMode, unsigned int device=0);Code language: C++ (
```

Configures the input mode for the selected decoder module to be used with single-ended or differential signals.

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

Parameters

- `channel`: the decoder module to configure (*DECODER_CHANNEL_0*, *DECODER_CHANNEL_1*, *DECODER_CHANNEL_2*or *DECODER_CHANNEL_3*).
- `inputMode`: selects if the decoder considers one GPI input per encoder signal (*SINGLEENDED*) or two GPI with complementary signals (*DIFFERENTIAL*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Dec_ConfigurePulsePerRotation` **— Set the number of pulses per rotation**

```
void Dec_ConfigurePulsePerRotation(tDecChannel channel, unsigned int pulsePerRotation, unsigned int device=0);Code
```

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

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

Parameters

- `channel`: the decoder module to configure (*DECODER_CHANNEL_0*, *DECODER_CHANNEL_1*, *DECODER_CHANNEL_2*or *DECODER_CHANNEL_3*).
- `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).

`Dec_ConfigureResetMode` **— Set the counter reset mode**

```
void Dec_ConfigureResetMode(tDecChannel channel, tDecResetMode 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

- `channel`: the decoder module to configure (*DECODER_CHANNEL_0*, *DECODER_CHANNEL_1*, *DECODER_CHANNEL_2*or *DECODER_CHANNEL_3*).
- `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).

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

```
void Dec_ConfigureResetPerRotation(tDecChannel channel, unsigned int resetPerRotation, unsigned int device=0);Code
```

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.

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

Parameters

- `channel`: the decoder module to configure (*DECODER_CHANNEL_0*, *DECODER_CHANNEL_1*, *DECODER_CHANNEL_2*or *DECODER_CHANNEL_3*).
- `resetPerRotation`: the number of Z pulses per rotation.
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Dec_ConfigureDirection` — Set the rotation direction

```
void Dec_ConfigureDirection(tDecChannel channel, tDecDirection direction, unsigned int device=0);Code language: C++ (
```

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

- `channel`: the decoder module to configure (*DECODER_CHANNEL_0*, *DECODER_CHANNEL_1*, *DECODER_CHANNEL_2*or *DECODER_CHANNEL_3*).
- `direction`: the positive direction, clockwise (*CW)* or counter-clockwise (*CCW*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Dec_ConfigureInputPolarity` — Set the polarity of the input signals

```
void Dec_ConfigureInputPolarity(tDecChannel channel, tDecPolarity polarity, unsigned int device=0);Code language: C++
```

Defines the polarity of the encoder signals. If inverted is selected, the decoder considers the inverted of the GPI signal.

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

Parameters

- `channel`: the decoder module to configure (*DECODER_CHANNEL_0*, *DECODER_CHANNEL_1*, *DECODER_CHANNEL_2*or *DECODER_CHANNEL_3*).
- `polarity`: the encoder signals polarity, normal (*NORM)* or inverted (*INV*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

`Dec_GetAngle` — Get the decoded position angle

```
float Dec_GetAngle(tDecChannel channel, unsigned int device=0);Code language: C++ (cpp)
```

Returns the value of the position angle decoded by the selected decoder module (in radians and in the range -2π to 2π).

It has to be called during the control interrupt.

Parameters

- `channel`: the decoder module to configure (*DECODER_CHANNEL_0*, *DECODER_CHANNEL_1*, *DECODER_CHANNEL_2*or *DECODER_CHANNEL_3*).
- `device`: the id of the addressed device (optional, used in multi-device configuration only).