# TMP - Temperature sensor

SD104  |  Posted on November 8, 2021  |  Updated on May 27, 2025

**Simon STROBL**
Product Director
imperix • in

Table of Contents

The temperature sensor (TMP) block provides access to the temperature measurement from a PT100 or PT1000 sensor for drive applications.

The B-Box RCP supports up to two temperature sensors through the Motor Interface for B-Box RCP. It can be either PT100 or PT1000 platinum resistors. These sensors have a standardized linear resistance-to-temperature characteristic (see the table below).

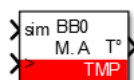| Characteristic | PT100 | PT1000 | Unit |
|---|---|---|---|
| Nominal resistance @ 0°C | 100 | 1000 | $\Omega$ |
| Temperature sensitivity | 0.385 | 3.85 | $\Omega$ / °C |

*Standard specifications of PT100 and PT1000 resistors*

The analog temperature measurement chain of the Motor Interface is agnostic to the type of resistor. Please refer to the datasheet on how to wire the sensors. As a result, the temperature sensor block does not require any knowledge of the sensor type, and parameters are normalized. The sensitivity of the sensors is 0.00385 pu/°C when normalized by the nominal resistance at 0°C. Similarly, the normalized offset is 1 pu.

The TMP 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 signal is the temperature in [°C].
- The `sim` input signal is used in simulation and represents the actual temperature, 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.
- `Temperature coefficient [pu/°C]` defines the sensitivity of the platinum resistor, normalized in per unit.
- `Resistance @ 0°C [pu]` defines the value of the platinum resistor at 0°C, normalized in per unit.

## PLECS block

## Signal specification

- The output signal is the temperature in [°C].
- The `sim` input signal is used in simulation and represents the actual temperature, 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.
- `Temperature coefficient [pu/°C]` defines the sensitivity of the platinum resistor, normalized in per unit.
- `Resistance @ 0°C [pu]` defines the value of the platinum resistor at 0°C, normalized in per unit.

# 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).

### `Tmp_AdjustSensorParams` — **Adjust the sensor parameters**

```cpp
void Tmp_AdjustSensorParams(tMotIntMachine machine, float tmpCoeff, float resistanceAtZeroK, unsigned int device=0
```

Adjusts the sensitivity and offset of the sensor for calibration purposes.

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

**Parameters**

- `machine`: the machine to configure (*MACHINE_A* or *MACHINE_B*).
- `tmpCoeff`: defines the sensitivity of the platinum resistor, normalized in per unit.
- `resistanceAtZeroK`: defines the value of the platinum resistor at 0°C, normalized in per unit.
- `device`: the id of the addressed device (optional, used in multi-device configuration only).

### `Tmp_GetTemperature` — **Get the temperature**

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

Gets the temperature reading in [°C].

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).