

Technical Brief

Which RTD Configuration is best for my Application?

A Resistance Temperature Detector (RTD) senses temperature by measuring resistance of an RTD element. As the temperature increases or decreases, the resistance also increases or decreases. This precise change in resistance will be influenced by the wires that connect it to the sensing circuit or transmitter.

The key to minimizing this influence is to eliminate the resistance of the wire, however, doing so increases complexity and cost.

There are three configurations, 2-wire, 3-wire, and 4-wire, which are commonly used in RTDs.



The 2-wire RTD is the simplest with a single wire connecting to both sides of the RTD element. This configuration is the least accurate of the three types. The resistance of each wire will be included by the measuring circuit and an error will result. The trade-off between accuracy and simplicity may be acceptable, depending on your application.

3-wire RTD connections

The 3-wire RTD configuration is the most commonly used RTD circuit design because it provides a good compromise of accuracy, cost and complexity. In this setup a third wire is used by the circuit to measure the resistance of the conductors on the 2-wire side, then approximate and negate the resistance of the RTD element wires. The result is much more accurate than a 2-wire configuration.

4-wire RTD connections

A 4-wire RTD is the most accurate, complex and expensive. In this configuration, a fourth wire is added to take the exact measurement of the conductors on both sides of the RTD element. The resistance is removed from the RTD measurement, resulting in the most accurate reading. 4-wire configurations are typically only required in settings such as a laboratory where extremely accurate measurements are needed.

For most industrial applications the 3-wire configuration is the best compromise and will meet most needs. If you have any questions, please feel free to contact one of our experts us at 800.232.5335 or info@teltru.com.









