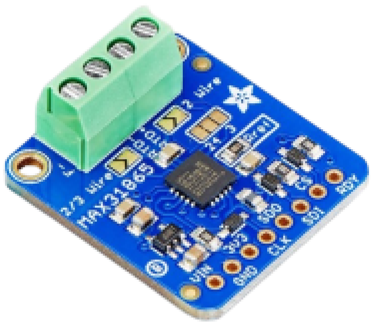
lamaPLC: Max31865 RTD to Digital Converter - PT100/PT1000 Platine



The **MAX31865** is an easy-to-use resistance-to-digital converter designed for platinum resistance temperature detectors (RTDs). An external resistor determines the sensitivity for the RTD used, and a precision delta-sigma ADC converts the RTD resistance ratio to the reference resistance into a digital output. The MAX31865's inputs are protected against overvoltage faults up to 45V. It also includes programmable detection for RTD open or short circuits and cable open or short circuits.

Key Features and Specifications

- **Sensor Compatibility:** Handles 2-, 3-, and 4-wire PT100 to PT1000 platinum RTDs.
- **High Accuracy:** Features a 15-bit ADC resolution, providing a nominal temperature resolution of 0.03125°C, with a total accuracy of 0.5°C max over all operating conditions.
- **Interface:** Communicates with microcontrollers using a 3 or 4-wire SPI-compatible interface.
- **Integrated Fault Detection:** Includes programmable detection for common errors like RTD open circuits, short circuits to voltage (out of range), or shorts across the RTD element, which increases system reliability.
- **Voltage Protection:** Inputs are protected against overvoltage faults up to ±45V.
- **Supply Voltage Range:** The main chip operates from 3.0V to 3.6V. Breakout boards often include voltage regulators and level shifters to support 3.3V to 5V microcontrollers such as Arduino or Raspberry Pi.
- **Conversion Time:** A maximum conversion time of 21ms for a 50Hz notch frequency filter.

Pinout

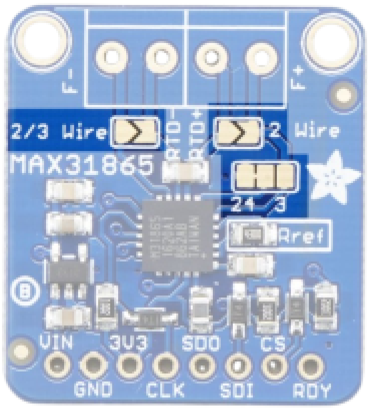
- Vin: This is the power pin. Since the chip operates at 3 VDC, we have included an on-board voltage regulator that accepts **3-5V DC** and safely steps it down. To power the board, supply it with the same logic-level voltage as your microcontroller — for example, 5V for a 5V microcontroller, such as an Arduino.
- 3Vo: This is the 3.3V output from the voltage regulator; you can draw up to 100mA from this if needed.
- GND: Common ground for power and logic.
- SCK: This is the SPI Clock pin, an input to the chip.
- SDO: This is the Serial Data Out / Microcontroller In Sensor Out pin, used for data sent from the MAX31865 to your processor.
- SDI: This is the Serial Data In / Microcontroller Out Sensor In pin, used for data sent from your processor to the MAX31865.
- CS: This is the Chip Select pin; drop it low to start an SPI transaction. It is an input to the chip.
- RDY (Ready): This is a data-ready indicator pin. You can use this pin to speed up your reads if you are writing your own driver. Our Arduino driver doesn't use it to save a pin.

If you'd like to support the development of the site with the price of a coffee — or a few — please do so here.

Here's a handy tip: you can quickly save this page as a PDF by clicking *“export to PDF”* in the menu on the right side of the screen.

2026/02/14 23:38

Configuration



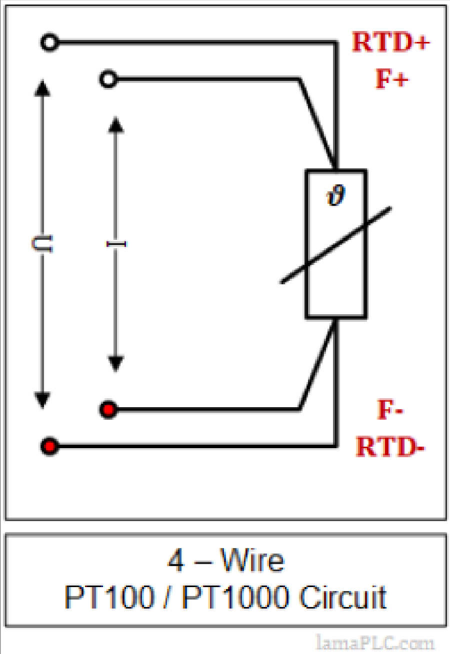
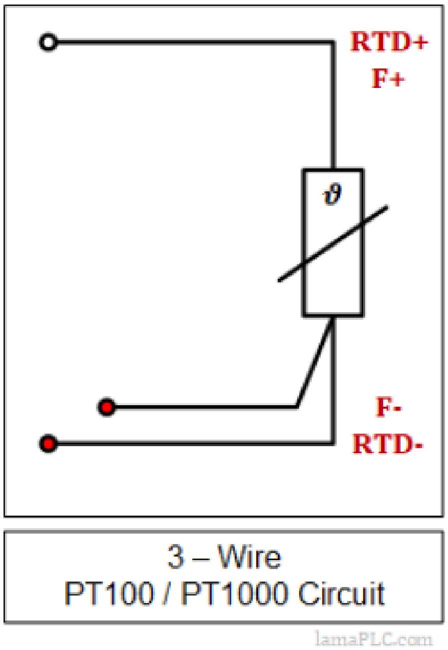
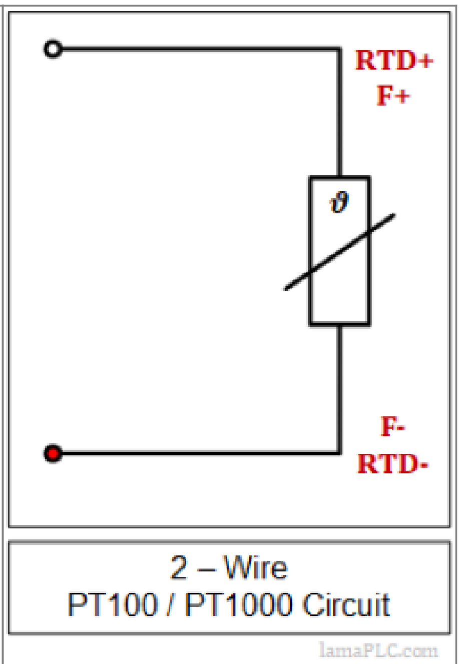
- By default, the sensor is configured for 4-wire RTD operation, but can be set for 2 or 3-wire. For a 4-wire setup, leave the jumpers as they are!
- For 3-wire usage: Solder the jumper labeled 2/3 Wire closed and cut the wire connecting the left side of the 2-way jumper just above Rref. Then, solder the right side labeled 3 closed.
- For 2-wire use: solder the two triangular jumpers below the terminal blocks closed, or connect short wire jumpers between the two terminal blocks on each side (essentially jumpering the two right-side terminal holes together, and the same for the left side).

PT100 / PT1000 sensors

RTDs (*Resistance Temperature Detectors*) are straightforward devices: simply a small strip of platinum that measures precisely 100 Ω or 1000 Ω at 0°C. Bonded to the PT100/PT1000 are two, three, or four wires.

Thus, the 4-wire RTD has two wires attached to each side of the sensor. Each wire has about 1Ω of resistance. When connected to the amplifier, the innovative amp measures the voltage across the RTD and across the wire pairs.

For example, the approximate resistances of a 4-wire PT100 RTD at 0 °C are as follows. (For a PT1000, the middle resistance would be about 1002Ω rather than 102Ω).

		
When the amp measures this sensor, it first assesses the resistance between one set of red and blue wires. It then measures the resistance between the red wires and the blue wires. To get the resistance of a single wire, divide each resistance value by two. The final calculation is 102 - 1 - 1 100Ω.	These are very similar to the 4-wire type, but only have one 'pair' of connected wires. This is because the wires for the RTD are all the same gauge and length; therefore, instead of two pairs, the amplifier reads one pair and uses that resistance for both wires.	It is as simple as it gets, with only one wire on each side. You might need to calibrate the sensor by placing it in an ice bath to measure the resistance at 0°C (around 102 Ω), then subtract 100 Ω to find the total resistance of the connection wires!

Connect the two ends of the PT100/PT1000 resistor to the RTD+ and RTD- terminals on the sensor module. For example, a resistance of 102 Ohms can be measured. In a 3-wire or 4-wire setup, the wire connections go to the F+ and F- terminals. These connections might differ from the resistance values of the respective sides by only a few Ohms, meaning the resistance between F+ and RTD+ or F- and RTD- may vary slightly, just a few Ohms.

°C	Ω	°C	Ω	°C	Ω	°C	Ω	°C	Ω	°C	Ω
-200	18,49	0	100,00	200	175,84	400	247,04	600	313,59	800	375,51
-190	22,80	10	103,90	210	179,51	410	250,48	610	316,80	810	378,48
-180	27,08	20	107,79	220	183,17	420	253,90	620	319,99	820	381,45
-170	31,32	30	111,67	230	186,82	430	257,32	630	323,18	830	384,40
-160	35,53	40	115,54	240	190,45	440	260,72	640	326,35	840	387,34
-150	39,71	50	119,40	250	194,07	450	264,11	650	329,51	850	390,26
-140	43,87	60	123,24	260	197,69	460	267,49	660	332,66		
-130	48,00	70	127,07	270	201,29	470	270,86	670	335,79		
-120	52,11	80	130,89	280	204,88	480	274,22	680	338,92		
-110	56,19	90	134,70	290	208,45	490	277,56	690	342,03		
-100	60,25	100	138,50	300	212,02	500	280,90	700	345,13		
- 90	64,30	110	142,29	310	215,57	510	284,22	710	348,22		
- 80	68,33	120	146,06	320	219,12	520	287,53	720	351,30		
- 70	72,33	130	149,82	330	222,65	530	290,83	730	354,37		
- 60	76,33	140	153,58	340	226,17	540	294,11	740	357,42		
- 50	80,31	150	157,31	350	229,67	550	297,39	750	360,47		
- 40	84,27	160	161,04	360	233,17	560	300,65	760	363,50		
- 30	88,22	170	164,76	370	236,65	570	303,91	770	366,52		
- 20	92,16	180	168,46	380	240,13	580	307,15	780	369,53		
- 10	96,09	190	172,16	390	243,59	590	310,38	790	372,52		

Basic resistance values in Ohm PT100 sensors according to DIN/IEC 751

2026/04/23 21:52
rtd, pt100, pt1000, sensor, temperature

SPI Wiring

Since this is an SPI-capable sensor, we can use either hardware or software SPI. To ensure consistent wiring across all Arduinos, we'll start with 'software' SPI. The following pins should be used:

- Connect the **Vin** to the **power supply**; 3.3V or 5V is fine. Use the same voltage as the microcontroller's logic. For most Arduinos, that is 5V.
- Connect **GND** to **common** power/data ground.
- Connect the **CLK** pin to Digital **#13**
- Connect the **SDO** pin to Digital **#12**
- Connect the **SDI** pin to Digital **#11**
- Connect the **CS** pin to Digital **#10**

Arduino code

To start reading sensor data, install the **Adafruit MAX31865** library from the Arduino library manager.

> https://www.adafruit.com/products/3328

This basic example uses the Adafruit library to read temperature from a PT100 sensor using a 3-wire configuration.

```
#include <Adafruit_MAX31865.h>

// Use hardware SPI: pass only the CS pin
```

```
Adafruit_MAX31865 thermo = Adafruit_MAX31865(10);

// Use 430.0 for PT100 and 4300.0 for PT1000
#define RREF      430.0
// The 'nominal' 0-degrees-C resistance of the sensor (100.0 for PT100)
#define RNOMINAL  100.0

void setup() {
  Serial.begin(115200);
  Serial.println("MAX31865 PT100 Test!");

  // Change to MAX31865_2WIRE or MAX31865_4WIRE as needed
  thermo.begin(MAX31865_3WIRE);
}

void loop() {
  uint16_t rtd = thermo.readRTD();
  float ratio = rtd;
  ratio /= 32768;

  Serial.print("RTD value: "); Serial.println(rtd);
  Serial.print("Ratio = "); Serial.println(ratio, 8);
  Serial.print("Resistance = "); Serial.println(RREF * ratio, 8);
  Serial.print("Temperature = ");
Serial.println(thermo.temperature(RNOMINAL, RREF));

  // Check for faults
  uint8_t fault = thermo.readFault();
  if (fault) {
    Serial.print("Fault 0x"); Serial.println(fault, HEX);
    thermo.clearFault();
  }
  Serial.println();
  delay(1000);
}
```

Sources

Adafruit MAX31865 RTD PT100 or PT1000 Amplifier

MAX31865, RTD, PT 100, PT 1000, temperature, SPI, Platinum, Arduino, code, sensor, Adafruit