



What are you looking for?

Currency  
GBP ▼

Login / Signup

Account

0  
Cart

Raspberry Pi ▼

Maker Store ▼

micro:bit ▼

Arduino ▼

Gifts ▼

Sale!

Tutorials

Blog

Super Fast Shipping  
from just £2.99

## Maker Advent Calendar Day #8: Tracking Temps!

By The Pi Hut • Dec 8, 2022 • 27 comments

Welcome to day eight of your [12 Projects of Codemas Advent Calendar](#). Today we're going to be sensing the temperature of our local environment and using these temperature readings to make fun projects!

Temperature sensors are another component that you'll likely walk past every single day. The thermostat for your home's heating system, whether a traditional version or a modern *smart* thermostat, will use something similar to what you find in your box today. Ovens, cars, offices and many other everyday items and locations use temperature sensors.

With winter very much here and getting colder every day, it's a great time to learn how to keep an eye on temperature with code and your Pico!

### Box #8 Contents

In this box you will find:

- 1x Custom waterproof temperature sensor with male jumper wires

- 1x 4.7k resistor
- 3x Male to male jumper wires



## Today's Project

Today we'll use our temperature sensor to measure the ambient temperature in your home, then move on to more exciting projects like sensing liquid temperature and creating an alarm to let you know when your home is getting too cold!

The sensor in your box is a custom version of our popular waterproof temperature sensor. The metal tip houses the sensor inside (specifically a *DS18B20-compatible 1-wire* sensor) and is sealed within a length of wire, allowing you to dunk this into water and other liquids to check the temperature.

We've added male jumper wires to this version to make it easier to use with a breadboard too.

### What is 1-wire?

**1-wire** is a serial communication protocol for devices to communicate to a controller...

*...ergh, let's make that a bit more human!* 1-wire allows a special device like our temperature sensor to *talk* to a controller, like our Pico, enabling it to send data to the Pico to be used in a program.

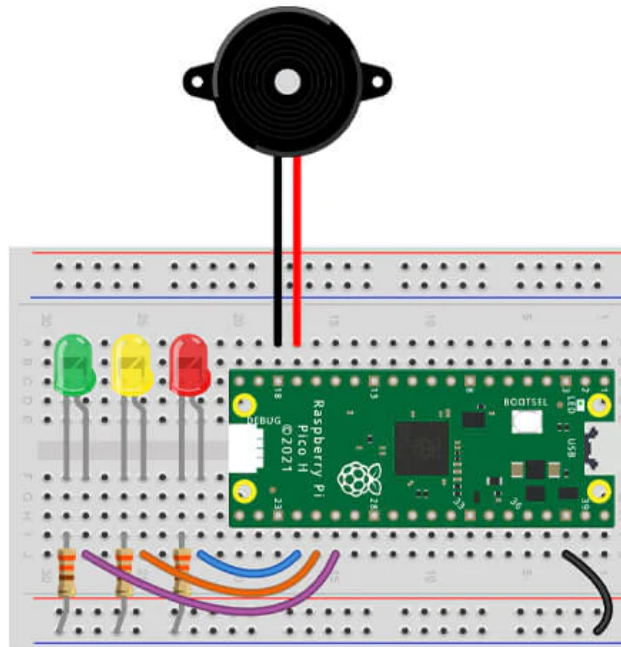
Despite it's name, you actually need at least two wires to use 1-wire devices, and in some examples (like our sensor) we're using three wires. The name is related to the data communication - you only need one wire to allow the devices to talk to each other both ways.

## Construct the Circuit

We're once again going to use the LEDs and buzzer along with the temperature sensor (they just pair so well with so many sensors and components!), so remove the PIR circuit from yesterday leaving just the LEDs and

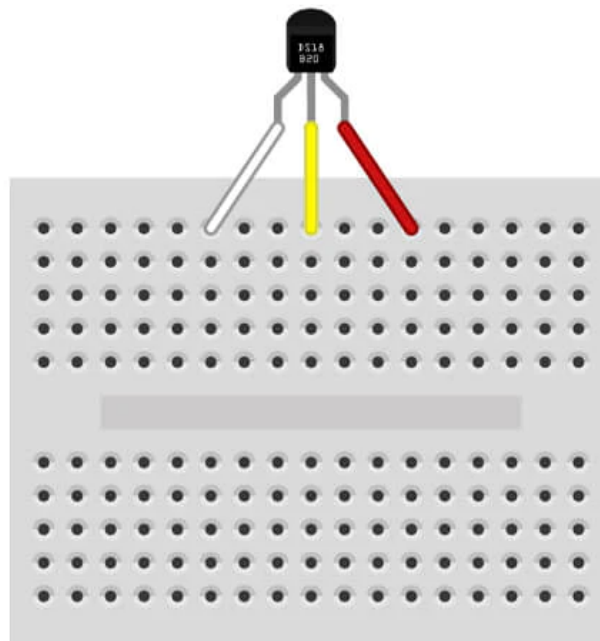
buzzer on the main breadboard.

Your starting circuit should look like this:

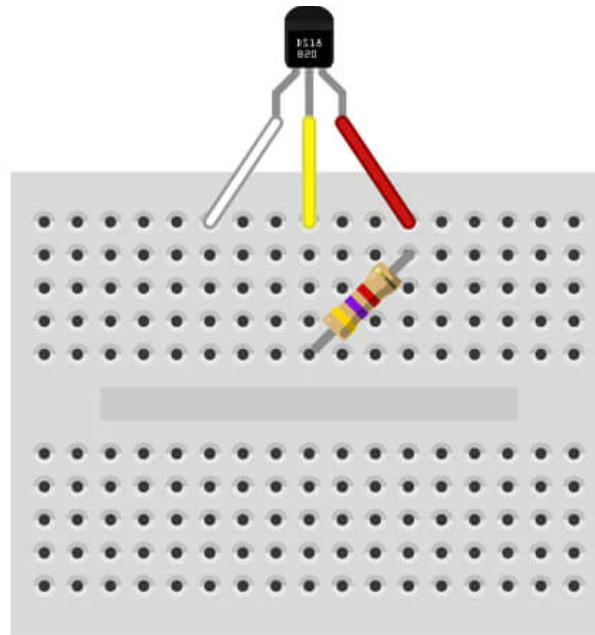


Now to add the temperature sensor to the mini breadboard. Your sensor should have three wires - red, yellow and white (the white wire may be black depending on the batch).

Place the wires into the top edge of the mini breadboard in the order white-yellow-red as per the image below:



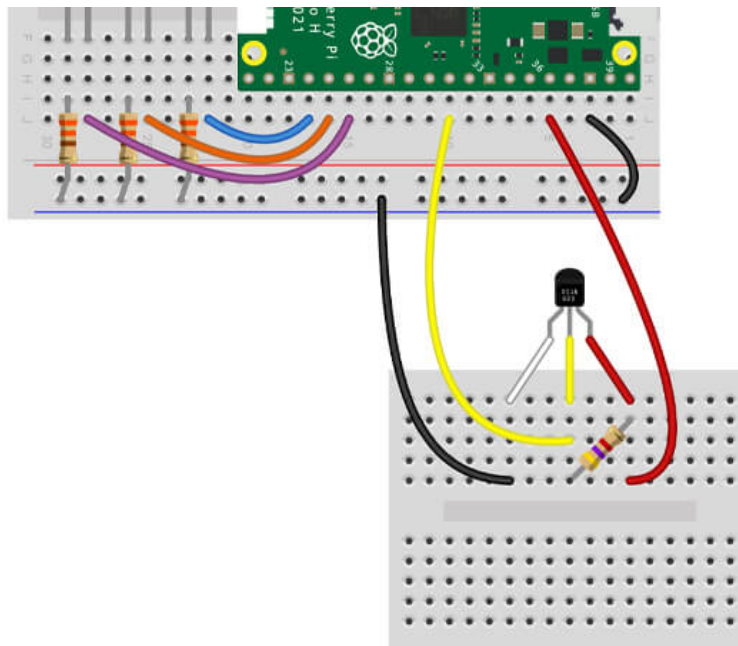
Next, add the resistor to connect the yellow and red wires, in the same way we show below:



Now we need to add three wires to complete the circuit and connect the sensor to our Pico:

- Connect the **left** leg to the blue **GND** lane
- Connect the **middle** leg to **GPIO 26** (*physical pin 31*)
- Connect the **right** leg to **3.3V** on *physical pin 36*

Your circuit should look similar to this:



## Activity 1: Simple Temperature Sensing

Let's start with a minimal program to take a single temperature reading from the sensor.

### Library imports

This sensor relies on some new **imports** that do a lot of the hard work for us - the **onewire library** and **ds18x20 library**. Luckily these are now built into MicroPython so we don't need to worry about downloading anything.

## 1-wire Code

We set up the sensor's pin as an input as we've done before with other sensors, however using this particular 1-wire sensor requires some extra code to get it working.

We have to tell MicroPython that we're using a 1-wire **DS18B20** sensor with the following line in our code below: **sensor = ds18x20.DS18X20(owewire.OneWire(SensorPin))**

We also need to scan for 1-wire devices on the GPIO pin we have set up. 1-wire devices such as our sensor have a unique registration number in their **read-only memory (rom)**. The following line in our code below scans for sensors (**roms**) connected to our pin: **roms = sensor.scan()**.

## For loop

We then use a **while loop** which contains a **for loop**. We used a **for loop** yesterday with the motion sensor when using the **range function**, but this time our range will be **roms** (the number of roms our scan found, which will just be our single sensor). The for loop is saying "*for every rom I find, do this*".

The code then converts the sensor's temperature reading to centigrade, and then prints this whilst also adding the **°C** symbol to the end of it. The code commentary should make everything clear - give it a try!

## The Code

```
# Imports
import onewire, ds18x20, time
from machine import Pin

# Set the data pin for the sensor
SensorPin = Pin(26, Pin.IN)

# Tell MicroPython we're using a DS18B20 sensor, and which pin it's on
sensor = ds18x20.DS18X20(owewire.OneWire(SensorPin))

# Look for DS18B20 sensors (each contains a unique rom code)
roms = sensor.scan()

while True: # Run forever

    sensor.convert_temp() # Convert the sensor units to centigrade

    time.sleep(2) # Wait 2 seconds (you must wait at least 1 second before taking a r

    for rom in roms: # For each sensor found (just 1 in our case)

        print((sensor.read_temp(rom)), "°C") # Print the temperature reading with °C a

        time.sleep(5) # Wait 5 seconds before starting the loop again
```



## Activity 2: Temperature Indicators

Now that we have the basics covered, let's use these readings to drive our LEDs to give us visual indication of the temperature.

The example below adds our LEDs back in and uses a **while loop** with multiple **if statement** conditions to drive the LEDs depending on the temperature - red if it goes below 18°C, amber if it's at a comfortable 18°C-22°C, and green if it's over 22°C.

We've covered while loops, if statements, if/elif and ranges in the previous boxes, so the code in this example should feel very familiar.

Grab a few cups of water at different temperatures (**no boiling water!**) and try dipping the waterproof end of the sensor into each one, watching the readings change and the different LEDs light up. **Keep the Pico end of things as far away as possible...water and electronics aren't the best of friends...**

### The Code

```
# Imports
import onewire, ds18x20, time
from machine import Pin

# Set up the LED pins
red = Pin(18, Pin.OUT)
amber = Pin(19, Pin.OUT)
green = Pin(20, Pin.OUT)

# Set the data pin for the sensor
SensorPin = Pin(26, Pin.IN)

# Tell MicroPython that we're using a DS18B20 sensor, and which pin it's on
sensor = ds18x20.DS18X20(onewire.OneWire(SensorPin))

# Look for DS18B20 sensors (each contains a unique rom code)
roms = sensor.scan()

while True: # Run forever

    time.sleep(5) # Wait 5 seconds between readings

    for rom in roms: # For each sensor found (just 1 in our case)

        sensor.convert_temp() # Convert the sensor units to centigrade
        time.sleep(1) # Always wait 1 second after converting

        reading = sensor.read_temp(rom) # Take a temperature reading

        print(reading) # Print the reading

        if reading <= 18: # If reading is less than or equal to 18
```

```
        red.value(1) # Red ON
        amber.value(0)
        green.value(0)

    elif 18 < reading < 22: # If reading is between 18 and 22

        red.value(0)
        amber.value(1) # Amber ON
        green.value(0)

    elif reading >= 22: # If reading is greater than or equal to 22

        red.value(0)
        amber.value(0)
        green.value(1) # Green ON
```

## Activity 3: Temperature Alarm

We can now introduce our buzzer along with an alarm function in our code again to make a fun temperature alarm project.

Whilst yesterday's motion sensor alarm project is great for spotting people moving in an area, this alarm can be used to keep an eye on the temperature in a specific room/area.

Most of us have thermostats in our homes, but there's great fun to be had creating your own temperature sensing alarm project to monitor a specific area - be it for your own health or monitoring the environment for plants and pets.

### The Code

The alarm example below is similar to yesterday's motion alarm, using an alarm **function** to light the LEDs and sound the buzzer when the temperature goes below 18°C - change the 18 to whatever is relevant for your project/scenario.

We've added as many comments as possible to help remind you what each section is doing:

```
# Imports
import onewire, ds18x20, time
from machine import Pin, PWM

# Set up the LED pins
red = Pin(18, Pin.OUT)
amber = Pin(19, Pin.OUT)
green = Pin(20, Pin.OUT)

# Set up the Buzzer pin as PWM
buzzer = PWM(Pin(13))

# Start PWM duty to 0% at program start
buzzer.duty_u16(0)
```

```
# Set the data pin for the sensor
SensorPin = Pin(26, Pin.IN)

# Tell MicroPython that we're using a DS18B20 sensor, and which pin it's on
sensor = ds18x20.DS18X20(OneWire.OneWire(SensorPin))

# Look for DS18B20 sensors (each contains a unique rom code)
roms = sensor.scan()

def alarm(): # Our alarm function

    buzzer.duty_u16(10000) # Buzzer duty (volume) up

    for i in range(5): # Run this 5 times

        buzzer.freq(5000) # Higher pitch

        # LEDs ON
        red.value(1)
        amber.value(1)
        green.value(1)

        time.sleep(0.2) # wait 1 second

        buzzer.freq(1000) # Lower pitch

        # LEDs OFF
        red.value(0)
        amber.value(0)
        green.value(0)

        time.sleep(0.2) # wait 1 second

    buzzer.duty_u16(0) # Buzzer duty (volume) off

while True: # Run forever

    time.sleep(5) # Wait 5 seconds between readings

    for rom in roms: # For each sensor found (just 1 in our case)

        sensor.convert_temp() # Convert the sensor units to centigrade
        time.sleep(1) # Always wait 1 second after converting

        reading = sensor.read_temp(rom) # Take a temperature reading

        print(reading) # Print the reading

        if reading < 18: # If reading is less than or equal to 18

            alarm() # Call our alarm function
```



## Day #8 Complete!

Another day, another component completed! You now have a temperature sensor in your growing arsenal of parts to make projects with, and we're sure you'll use this one time and time again.

As you're probably starting to realise, a lot of these sensors and components are coded in a very similar way. Sometimes we need the help of imported libraries, sometimes we need resistors/additional hardware, but the fundamental way we write code for them (with **loops**, **if statements** and **variables**) is pretty consistent.

So what did we cover on day #8? Today you have:

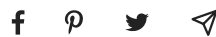
- Built a circuit with a temperature sensor
- Learnt how to use a temperature sensor with MicroPython and the Pico
- Used your first 1-wire component and the 1-wire library
- Created a temperature monitor and temperature alarm system
- Learnt more about *for loops*

As always, keep your circuit safe somewhere until tomorrow (**don't take anything apart just yet**) and we'll see you again tomorrow for more fun!

---

We used [Fritzing](#) to create the breadboard wiring diagram images for this page.

---



### Featured Products



**Maker Advent Calendar - The 12 Projects of Codemas (inc. Raspberry Pi Pico H)**

**£40** incl. VAT

**ADD TO CART**

### 27 comments

**Sarah**

February 9, 2023 at 2:25 pm

Hi, we are getting this error. We've tried swapping the yellow and red wires as someone above suggested and it didn't change anything. The error comes immediately on pressing run so changing the time.sleep to be longer makes no difference.



Traceback (most recent call last):

File "", line 16, in

File "ds18x20.py", line 20, in convert\_temp

File "onewire.py", line 23, in reset

**OneWireError:**

Hi, we are getting this error. We've tried swapping the yellow and red wires as someone above suggested and it didn't change anything. The error comes immediately on pressing run so changing the time.sleep to be longer makes no difference.

Traceback (most recent call last):

File "", line 16, in

File "ds18x20.py", line 20, in convert\_temp

File "onewire.py", line 23, in reset

OneWireError:

**The Pi Hut**

January 11, 2023 at 1:15 pm



@Simeon – That's just a packing error (luckily it was the 'good' way and you weren't missing anything). It might come in handy later on in your maker journey so keep hold of it :)

@Simeon – That's just a packing error (luckily it was the 'good' way and you weren't missing anything). It might come in handy later on in your maker journey so keep hold of it :)

**Simeon**

January 11, 2023 at 12:45 pm



When I opened my day 8 box I found 2 resistors but there was only one that you used, wondering if this was a packaging error or we use them later in another project?

When I opened my day 8 box I found 2 resistors but there was only one that you used, wondering if this was a packaging error or we use them later in another project?

**Rowan**

January 11, 2023 at 12:43 pm



I had the same issue as Kat, first time in the whole kit I've had any issues! And I solved it by unplugging and plugging all the pins back in. And making sure the sensor was in Pin 26 (physical pin 31) and that I called it.

Loving this kit! Having the most fun and learning so much.

I had the same issue as Kat, first time in the whole kit I've had any issues! And I solved it by unplugging and plugging all the pins back in. And making sure the sensor was in Pin 26 (physical pin 31) and that I called it.

Loving this kit! Having the most fun and learning so much.

**Sebastien**

December 23, 2022 at 8:39 am



@Rob you can find the implementation of `convert_temp` on the github of micropython-lib:  
<https://github.com/micropython/micropython-lib/blob/master/micropython/drivers/sensor/ds18x20/ds18x20.py>  
 It seems to send some data to the sensor, probably low level commands.

While searching for it, I've found this implementation which looks really similar, and it has a method to convert celsius into fahrenheit:

[https://github.com/robert-hh/Onewire\\_DS18X20/blob/master/ds18x20.py#L96-L97](https://github.com/robert-hh/Onewire_DS18X20/blob/master/ds18x20.py#L96-L97)

@Rob you can find the implementation of `convert_temp` on the github of micropython-lib:  
<https://github.com/micropython/micropython-lib/blob/master/micropython/drivers/sensor/ds18x20/ds18x20.py>  
 It seems to send some data to the sensor, probably low level commands.

While searching for it, I've found this implementation which looks really similar, and it has a method to convert celsius into fahrenheit:

[https://github.com/robert-hh/Onewire\\_DS18X20/blob/master/ds18x20.py#L96-L97](https://github.com/robert-hh/Onewire_DS18X20/blob/master/ds18x20.py#L96-L97)

### The Pi Hut

December 20, 2022 at 6:01 pm



@Rob – Unfortunately the MicroPython documentation doesn't say a lot about this but you can run the conversion in your code if needed. Here are the official docs:([https://mpython.readthedocs.io/en/master/library/mPython/ds18x20.html#ds18x20.DS18X20.convert\\_temp](https://mpython.readthedocs.io/en/master/library/mPython/ds18x20.html#ds18x20.DS18X20.convert_temp))

@Rob – Unfortunately the MicroPython documentation doesn't say a lot about this but you can run the conversion in your code if needed. Here are the official docs:([https://mpython.readthedocs.io/en/master/library/mPython/ds18x20.html#ds18x20.DS18X20.convert\\_temp](https://mpython.readthedocs.io/en/master/library/mPython/ds18x20.html#ds18x20.DS18X20.convert_temp))

### The Pi Hut

December 20, 2022 at 5:57 pm



@Markus – An entry on StackExchange explained this well – "The 1-wire bus requires that the control signal be pulled high so the master device can pull it low to ask for data, and the slave device can pull it low to give the data. This allows you to have multiple 1-wire devices on the same "one wire" (source: <https://arduino.stackexchange.com/questions/30822/the-use-of-4-7kohm-resistor-with-ds18b20-temperature-sensor>).

You'll also see a comment suggesting that the resistor doesn't need to be 4.7K, or in some instances you don't need a resistor at all, but it's one of those things we've 'always done' so we tried it, it worked as expected, so we stuck with what we know :)

@Markus – An entry on StackExchange explained this well – "The 1-wire bus requires that the control signal be pulled high so the master device can pull it low to ask for data, and the slave device can pull it low to give the data. This allows you to have multiple 1-wire devices on the same "one wire" (source: <https://arduino.stackexchange.com/questions/30822/the-use-of-4-7kohm-resistor-with-ds18b20->

temperature-sensor).

You'll also see a comment suggesting that the resistor doesn't need to be 4.7K, or in some instances you don't need a resistor at all, but it's one of those things we've 'always done' so we tried it, it worked as expected, so we stuck with what we know :)

### The Pi Hut

December 20, 2022 at 5:53 pm

@Ben Yes, you could use multiple sensors technically. The MicroPython documentation doesn't cover this in much detail so it's not something we've tried yet, but the short example on this page shows how you could print the temperature value of each device:



<https://mpython.readthedocs.io/en/master/library/mPython/ds18x20.html>

Re damaging pins, it can happen, but it's quite rare and shouldn't have happened if you've only been using the parts in the calendar. You may want to use another analogue component on the same pin to check functionality (ruling out this particular sensor).

@Ben Yes, you could use multiple sensors technically. The MicroPython documentation doesn't cover this in much detail so it's not something we've tried yet, but the short example on this page shows how you could print the temperature value of each device: <https://mpython.readthedocs.io/en/master/library/mPython/ds18x20.html>

Re damaging pins, it can happen, but it's quite rare and shouldn't have happened if you've only been using the parts in the calendar. You may want to use another analogue component on the same pin to check functionality (ruling out this particular sensor).

### Rob

December 20, 2022 at 4:37 pm



Silly question(s)... is there more info on the 'convert\_temp()' function? I've been searching, but I'm not finding much other than it's a function that converts the data to a temp.

Also, is there a param that can convert the temp to Fahrenheit or would you need to write code to do the calculation?

Silly question(s)... is there more info on the 'convert\_temp()' function? I've been searching, but I'm not finding much other than it's a function that converts the data to a temp.

Also, is there a param that can convert the temp to Fahrenheit or would you need to write code to do the calculation?

### Markus

December 20, 2022 at 4:37 pm



Am I right, that the resistor is used as a pullup to provide a high signal, when nothing is received

from the sensor? In this case the dimension doesn't really matters, it just has to be big enough.

Is this the right interpretation?

Am I right, that the resistor is used as a pullup to provide a high signal, when nothing is received from the sensor? In this case the dimension doesn't really matters, it just has to be big enough. Is this the right interpretation?

**Ben**

December 20, 2022 at 4:37 pm

One quick question re: the ROMS piece, in order to understand how this works better.



Does this mean that one could have more than one sensor on the same PIN, discover the ROM IDs and then cycle through multiple sensors?

If that is the case, when one issues the `convert_temp()` does that impact all sensors on the Pin?

Or is the limitation that you can still only have 1 sensor per Pin, it's just that you need the ROM of the sensor in order to read the output?

One quick question re: the ROMS piece, in order to understand how this works better.

Does this mean that one could have more than one sensor on the same PIN, discover the ROM IDs and then cycle through multiple sensors?

If that is the case, when one issues the `convert_temp()` does that impact all sensors on the Pin?

Or is the limitation that you can still only have 1 sensor per Pin, it's just that you need the ROM of the sensor in order to read the output?

**Ben**

December 20, 2022 at 4:37 pm

I had the same issue as Kat and Ivan and couldn't fix it, even by moving to Pin 27, or 26.



However, i moved to Pin 22 or 28 and all was good.

Could there be something in the theory that these other pins have become "damaged" in some way? They seem, to work fine as Pin.IN or Pin.OUT.

Kind Regards

Ben

I had the same issue as Kat and Ivan and couldn't fix it, even by moving to Pin 27, or 26.

However, i moved to Pin 22 or 28 and all was good.

Could there be something in the theory that these other pins have become "damaged" in some way?

They seem, to work fine as Pin.IN or Pin.OUT.

Kind Regards

Ben

**Ivan**

December 19, 2022 at 10:59 am



@Kat – I was getting the same error as you and couldn't really fix it but finally changed the pin I was using for sensing the temperature (27 instead of 26) and the project code started working like a charm.

One possibility may be that pin 26 has been damaged by one of the previous projects and cannot read digital signals any more (I did test it with PWM and is still operational though).

Hope that helps!

@Kat – I was getting the same error as you and couldn't really fix it but finally changed the pin I was using for sensing the temperature (27 instead of 26) and the project code started working like a charm.

One possibility may be that pin 26 has been damaged by one of the previous projects and cannot read digital signals any more (I did test it with PWM and is still operational though).

Hope that helps!

**Kat**

December 19, 2022 at 10:55 am



Thanks Ben, that resolved it for me.

I swapped the Red and Yellow wires around and it worked!

Thanks Ben, that resolved it for me.

I swapped the Red and Yellow wires around and it worked!

**Alessio Graziano**

December 19, 2022 at 10:55 am



Hello, I keep having the same error already reported by Kat. Do you have ideas on how to fix it? I take this opportunity to thank you for creating a similar product, I'm really enjoying it!! Thank you.

Hello, I keep having the same error already reported by Kat. Do you have ideas on how to fix it?

I take this opportunity to thank you for creating a similar product, I'm really enjoying it!! Thank you.

**The Pi Hut**

December 15, 2022 at 11:50 am



@Keith We give you more jumper wires each day as we thought it might be handy for those who might fancy building up their own grand project with all parts included, rather than re-using each day and not having enough to combine things.

We considered challenges, and occasionally we prompt readers to try adding other parts back in, but we're also aware that many users will be brand new to this and may get stuck. We didn't



want any disappointment or confidence-crushers.

@Keith We give you more jumper wires each day as we thought it might be handy for those who might fancy building up their own grand project with all parts included, rather than re-using each day and not having enough to combine things.

We considered challenges, and occasionally we prompt readers to try adding other parts back in, but we're also aware that many users will be brand new to this and may get stuck. We didn't want any disappointment or confidence-crushers.

### The Pi Hut

December 15, 2022 at 11:50 am



@Huw Millington If you can send us a quick message via [support.thepihut.com](mailto:support.thepihut.com), we'll get a replacement resistor sent out to you :)

@Huw Millington If you can send us a quick message via [support.thepihut.com](mailto:support.thepihut.com), we'll get a replacement resistor sent out to you :)

### Keith

December 15, 2022 at 11:46 am

Thanks – 8 down (4 to go) and it's all worked well. 2 comments:



- 1) What's with all the jumper wires?
- 2) It might have been nice to have a daily challenge. It wouldn't have to be anything big, just enough to encourage you to mess with the code a bit. For instance, the last programme from today could be amended so that the green LED stays on to indicate the kit is running, and the red and amber flash alternately for the alarm. It would encourage users not to simply copy and paste the code without even reading it.

Thanks – 8 down (4 to go) and it's all worked well. 2 comments:

- 1) What's with all the jumper wires?
- 2) It might have been nice to have a daily challenge. It wouldn't have to be anything big, just enough to encourage you to mess with the code a bit. For instance, the last programme from today could be amended so that the green LED stays on to indicate the kit is running, and the red and amber flash alternately for the alarm. It would encourage users not to simply copy and paste the code without even reading it.

### Huw Millington

December 15, 2022 at 11:46 am



I was missing the 4.7k resistor today (or it went flying somewhere when extracting the temperature sensor's wire from the bag) so I pressed yesterday's 10k one into service again. It worked, but I suspect the readings were not quite accurate: it measures about 5c in my glass of iced diet Coke which I would have expected to be slightly lower.

I was missing the 4.7k resistor today (or it went flying somewhere when extracting the temperature sensor's wire from the bag) so I pressed yesterday's 10k one into service again. It worked, but I suspect

the readings were not quite accurate: it measures about 5c in my glass of iced diet Coke which I would have expected to be slightly lower.

**Ben**

December 13, 2022 at 10:06 am



Aha! Further to my previous comment, it looks like the issue was that the sensor wires were in the wrong order on the breadboard, and that was what was causing the problem in the code. Fixed (for me).

Aha! Further to my previous comment, it looks like the issue was that the sensor wires were in the wrong order on the breadboard, and that was what was causing the problem in the code. Fixed (for me).

**Ben**

December 13, 2022 at 10:06 am



I'm getting the same error as @Kat even if I just run the `sensor.convert_temp()` outwith the loop on it's own? So I don't think adding to the sleep time afterwards will help as it's throwing the error before getting to it....

I'm getting the same error as @Kat even if I just run the `sensor.convert_temp()` outwith the loop on it's own? So I don't think adding to the sleep time afterwards will help as it's throwing the error before getting to it....

**The Pi Hut**

December 12, 2022 at 10:15 am



@Kat We haven't seen that before, but it may be that the sensor is taking longer than usual to convert the temperature. Try increasing the `time.sleep()` after the `sensor.convert_temp()` line to see if that helps (maybe try 5 or 10 seconds first, just to see if it helps)

@Kat We haven't seen that before, but it may be that the sensor is taking longer than usual to convert the temperature. Try increasing the `time.sleep()` after the `sensor.convert_temp()` line to see if that helps (maybe try 5 or 10 seconds first, just to see if it helps)

**Kat**

December 12, 2022 at 10:11 am

I get this.. any ideas?



```
>>> %Run -c $EDITOR_CONTENT
Traceback (most recent call last):
  File "", line 16, in
  File "ds18x20.py", line 20, in convert_temp
  File "onewire.py", line 23, in reset
OneWireError:
```

```
>>>
```

I get this.. any ideas?

```
>>> %Run -c $EDITOR_CONTENT
```

Traceback (most recent call last):

File "", line 16, in

File "ds18x20.py", line 20, in convert\_temp

File "onewire.py", line 23, in reset

OneWireError:

```
>>>
```

### The Pi Hut

December 9, 2022 at 12:41 am



@REP – This is something we discovered when investigating the DS18X20 driver for the calendar. If you take a look around the middle of this page (<https://mpython.readthedocs.io/en/master/library/mPython/ds18x20.html>) you will see at least 750ms between the convert\_temp line and taking a reading is recommended. We then found the same information on similar sensor datasheets which stated that this is how long it takes the sensor to convert the temperature ('conversion time'). If we tried before this, we may not get a reliable reading (or a reading at all). We bumped it up to 1 second for a bit of a buffer. Hope that helps.

@REP – This is something we discovered when investigating the DS18X20 driver for the calendar. If you take a look around the middle of this page (<https://mpython.readthedocs.io/en/master/library/mPython/ds18x20.html>) you will see at least 750ms between the convert\_temp line and taking a reading is recommended. We then found the same information on similar sensor datasheets which stated that this is how long it takes the sensor to convert the temperature ('conversion time'). If we tried before this, we may not get a reliable reading (or a reading at all). We bumped it up to 1 second for a bit of a buffer. Hope that helps.

### The Pi Hut

December 9, 2022 at 12:23 am



@Adan – Indeed, many makers spotted this (which means everyone is paying close attention which is great!). We used one of the other many GND pins instead, which will work just fine as they're all the same, but it didn't match the instructions. Thanks everyone for letting us know, the diagram has been updated.

@Adan – Indeed, many makers spotted this (which means everyone is paying close attention which is great!). We used one of the other many GND pins instead, which will work just fine as they're all the same, but it didn't match the instructions. Thanks everyone for letting us know, the diagram has been updated.

### REP

December 9, 2022 at 12:33 am





Hi there,

Why is it the case that you have to wait 1 second after `sensor.convert_temp()`?

Hi there,

Why is it the case that you have to wait 1 second after `sensor.convert_temp()`?

**Adan**



December 9, 2022 at 12:20 am

Probably already had this a 1000 times but I think the wiring diagram is wrong as the ground goes to physical 27

Probably already had this a 1000 times but I think the wiring diagram is wrong as the ground goes to physical 27

## Leave a comment

All comments are moderated before being published.

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.




**SUBMIT**

## Related Posts



# MicroPython Skill Builders

---

## #4 Functions with Potentiometers

### MicroPython Skill Builders - #4 Functions with Potentiometers

Tony Goodhew • Jul 4, 2023

Welcome to the fourth instalment of our MicroPython Skill Builders series by Tony Goodhew, aiming to improve your coding skills with MicroPython whilst introducing new components and coding techniques -...

[Read more](#)



# MicroPython Skill Builders

---

## #3 LEDs & Procedures

### MicroPython Skill Builders - #3 LEDs and Procedures

Tony Goodhew • Jun 21, 2023



This is the third instalment of our MicroPython Skill Builders series by Tony Goodhew, aiming to improve your coding skills with MicroPython whilst introducing new components and coding techniques - using a Raspberry...

[Read more](#)



### MicroPython Skill Builders - #2 LEDs & Lists

Tony Goodhew • May 30, 2023

This is the second instalment of our MicroPython Skill Builders series by Tony Goodhew, aiming to improve your coding skills with MicroPython whilst introducing new components and coding techniques - using...

[Read more](#)

---

#### Handy Links

[All Products](#)

[FAQs](#)

[Popular Searches](#)

[Search](#)

[Site Reviews](#)

#### Got any questions?

[Contact Us / Support Portal](#)

[Can I Cancel My Order?](#)

[Has My Order Shipped Yet?](#)

[Where Is My Order?](#)

[Do You Ship To {insert country name}](#)

[How Much Is Shipping?](#)



## Terms & Conditions

[Delivery](#)

[Lithium Shipping](#)

[Pre-Orders](#)

[Privacy Statement](#)

[Policies](#)

[Terms of Service](#)

[Company Info](#)

[FAQ](#)

[Klarna FAQ](#)

[Quick Start Guide](#)

[Search](#)

[Support Portal](#)

## Our Store Sections

[Raspberry Pi](#)

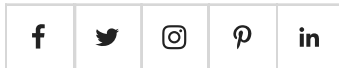
[Maker Store](#)

[micro:bit](#)

[Arduino](#)

[Gifts](#)

## Follow us



We accept



© The Pi Hut 2023