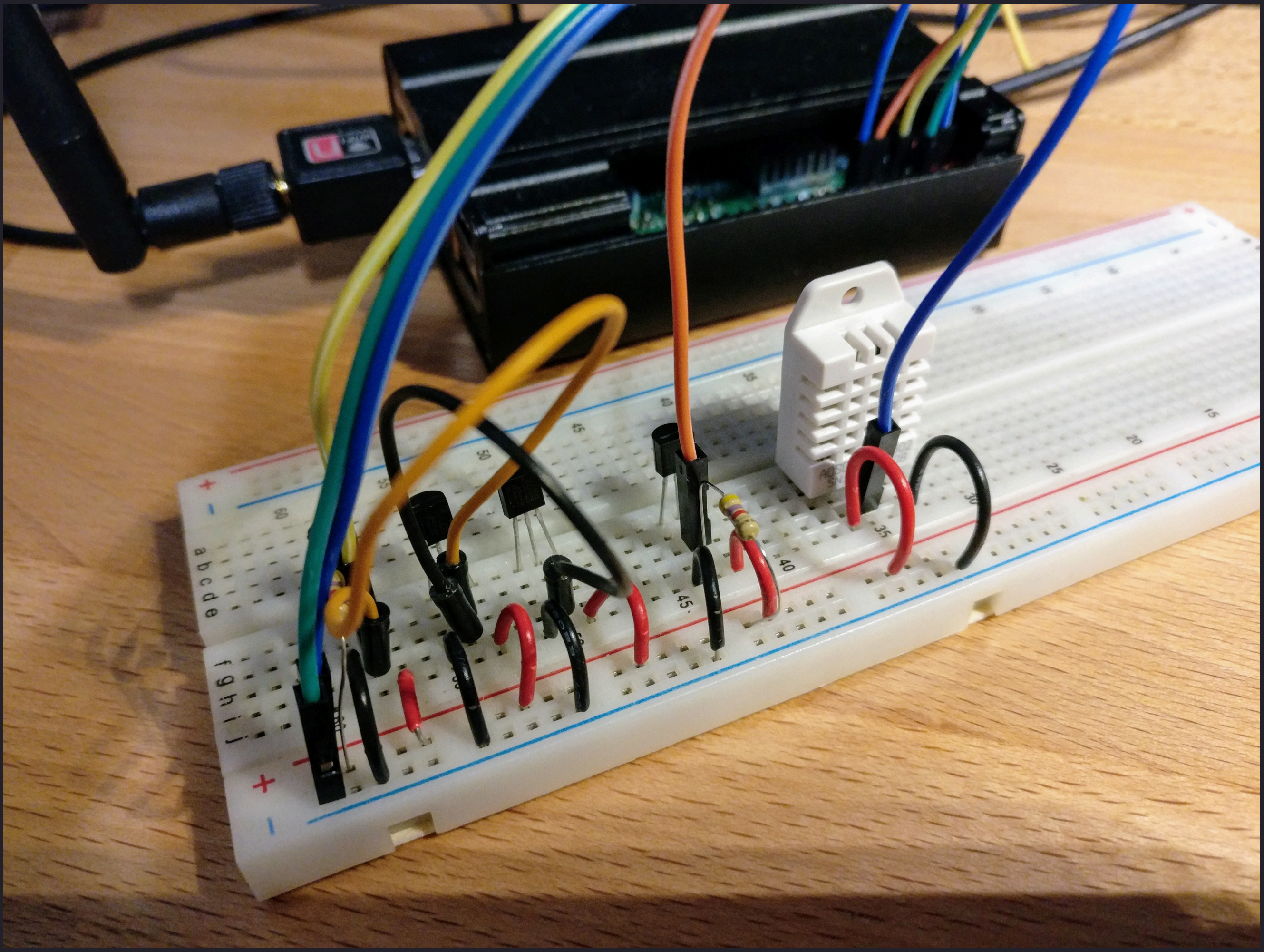
Multiple 1-Wire Buses on the Raspberry Pi

2018-03-27 :: [Updated :: 2023-02-16]

#hardware #raspberrypi #onewire #iot



The DS18B20 is a popular temperature sensor that uses the 1-Wire protocol for communication. Recent versions of the Linux kernel include a kernel driver for this protocol, making it relatively convenient to connect one or more of these devices to a Raspberry Pi or similar device. 1-Wire devices can be daisy chained, so it is possible to connect several devices to your Pi using only a single GPIO pin, and you’ll find many articles out there that describe how to do so.

Occasionally, it may be necessary to have more than a single chain of connected devices. For example, you may have reached the limits on the size of your 1-Wire network, or you may simply need to route your cables in a way that makes a single chain difficult. You can enable *multiple* 1-Wire buses on your Raspberry Pi to handle these situations.

For a single 1-Wire bus, you add the following to `/boot/config.txt` :

```
dtoverlay=w1-gpio
```

This will enable the 1-Wire bus on GPIO 4. To enable *multiple* 1-Wire buses, you will use multiple `dtoverlay` statements and the `gpiopin` parameter to the `w1-gpio` overlay. For example, to enable 1-Wire buses on GPIO 4 and GPIO 17, you would use:

```
dtoverlay=w1-gpio,gpiopin=4
dtoverlay=w1-gpio,gpiopin=17
```

In the picture at the top of this post, there are four DS18B20 sensors. Three are connected to the 1-Wire bus on GPIO 4, and one is connected to the 1-Wire bus on GPIO 17. Looking in `/sys/bus/w1/devices` , I see two w1_bus_master devices (and the four temperature sensors):

```
$ ls /sys/bus/w1/devices/
28-041722cbacff  28-0417231547ff  w1_bus_master1
28-041722ce24ff  28-04172318c0ff  w1_bus_master2
```

I can check the temperature on all four devices like this:

```
$ cat /sys/bus/w1/devices/28-*/w1_slave
50 01 4b 46 7f ff 0c 10 e8 : crc=e8 YES
50 01 4b 46 7f ff 0c 10 e8 t=21000
50 01 4b 46 7f ff 0c 10 e8 : crc=e8 YES
50 01 4b 46 7f ff 0c 10 e8 t=21000
57 01 4b 46 7f ff 0c 10 38 : crc=38 YES
57 01 4b 46 7f ff 0c 10 38 t=21437
57 01 4b 46 7f ff 0c 10 38 : crc=38 YES
57 01 4b 46 7f ff 0c 10 38 t=21437
```

You may have noted that there is also a DHT22 sensor in the picture. Much like the 1-Wire overlay, the kernel driver for the DHT22 can be associated with an arbitrary GPIO pin like this:

```
dtoverlay=dht11,gpiopin=27
```

8 Comments - *powered by utteranc.es*

dibdib99 commented on Oct 8, 2020

Hi,
The DS18B20 Dallas device has a Programmable Resolution from 9 Bits to 12 Bits.
This is selected by writing to byte 4 of the scratchpad memory. How do you write to
this device using the w1 bus interface?
The power on resolution is 12 bits (0.0625 deg C) which takes 750ms to do a conversion. As the accuracy os only 0.5 deg C I would like to use only 9 bits or resolution (0.5 deg) when a conversion takes only 93 ms.
Thanks
Ian davidson

larsks commented on Oct 17, 2020 Owner

Hi, thanks for the question! I don't have an answer off the top of my head, but it looks like this question has been asked elsewhere: https://raspberrypi.stackexchange.com/questions/71563/how-to-set-precisionof-ds18b20-via-w1-therm

HuthjaminBen commented on Jan 15, 2021

Hi,
how many GPIO-pins may I use as one-wire-bus-pins? All? (That would be fantastic!) Am I able to only read out the sensors connectet to one specified pin without caring about the ID of the sensor? For example may i read all temperatures from pin 17, without knowing their IDs?
Thanks in advance
Ben

larsks commented on Jan 15, 2021 Owner

> how many GPIO-pins may I use as one-wire-bus-pins?

I think you can use any available gpio. Your best bet is just to try it.

> Am I able to only read out the sensors connectet to one specified pin without caring about the ID of the sensor?

It's been a while since I wrote this, but I don't believe you need to know the id of the sensors; the driver discovers the available devices on each bus.

GerfriedC commented on Jan 25, 2021

Is this feature of connecting several 1-wire branches available with the Pico too?

larsks commented on Jan 25, 2021 Owner

@GerfriedC It is generally possible to support multiple 1-wire buses on a microcontroller...e.g., the standard Arduino onewire library accepts a pin number in the constructor, so can create a bus for any spare GPIO pin.

I can't comment specifically on the Pico because I don't have one, and every place I've looked is out of stock.

In any case, the solution wouldn't have anything to do with this article, which is specific to Linux.

GerfriedC commented on Jan 25, 2021

Now I have tested and two questions:

Now I have tested and two questions:

- for what topologies *pullup=on* is required?
- I'm measuring only 1.6V on the 3V3 pin between GPIO22 and 10
  May be the RPi Zero has not enough power for two 1-wire strands?
  many thanks

HuthjaminBen commented on Jan 25, 2021

I've tried it in 3 GPIO Pins and it worked, but than I noticed, that this wont let my project become easier, so I did not test the max of GPIO Pins.
Ive also tried it with a RasPi ZeroW (2 Pins) - it worked fine. As I understood the pullup just makes sure, that the pin definitly goes to "+", so that the 1-wire-master is able to send a kind of "start-comand" to the sensor by bridging the pin to zero - if You use power supply for the sensor (not that "parasite - power - thing") it should work even with less than 1,6V.

Write Preview

Sign in to comment

Styling with Markdown is supported

Sign in with GitHub

Lars Kellogg-Stedman :: Theme made by panr