What is Virtual Pins

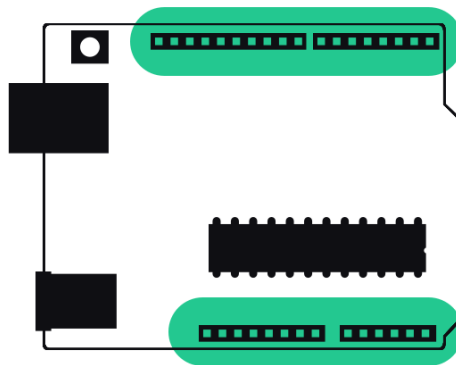Exchange any data between Blynk app and your hardware

Written by Pavel
Updated over a week ago

Virtual Pin is a concept invented by Blynk Inc. to provide exchange of any data between hardware and Blynk mobile app.

Virtual pins are different than Digital and Analog Input/Output (I/O) pins. They are **physical pins** on your microcontroller board where you connect sensors and actuators.
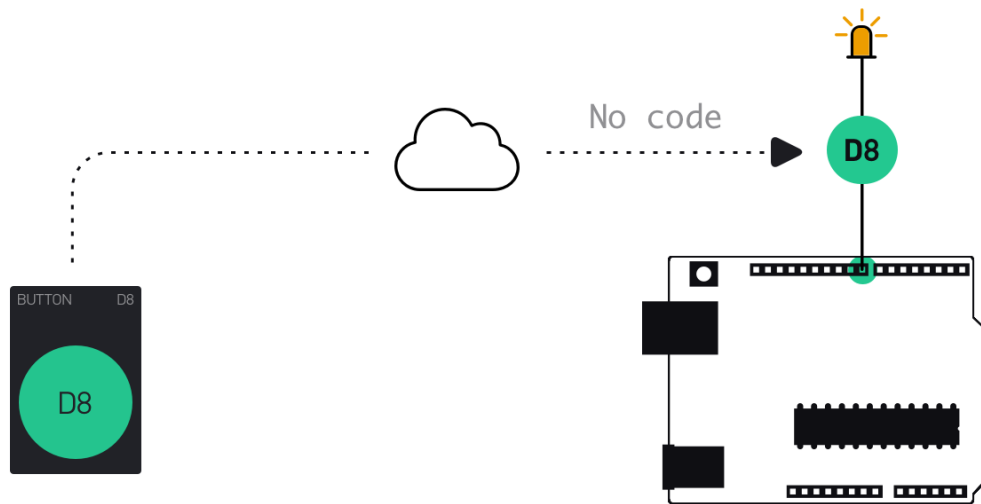

● Digital and Analog Pins

Blynk lets you control any hardware connected to Digital and Analog pins without having to write any additional code.

For example, if you need to turn On/Off LED connected to Digital pin, you don't have to write any code:

1. Just use BlynkBlink code for your hardware.
2. In the Blynk app - add **Button Widget** and set it to pin **D8**
3. That's it! No additional code is required. Simply press Play in the app.
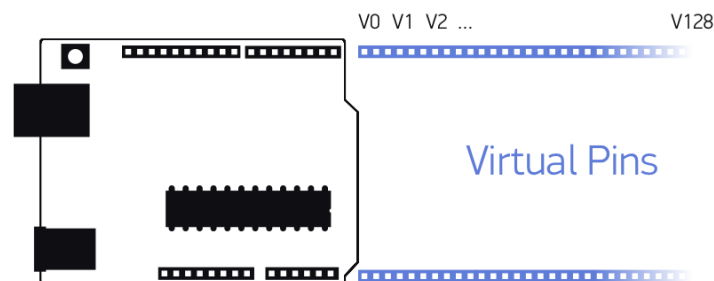
That was, easy, right? But what if you need more flexibility?

Virtual Pins

Virtual pins allow you to interface with any sensor, any library, any actuator.

Imagine that there are "virtual" pins that you can use

Think about Virtual Pins as a box where you can put any value, and everyone who has access to this box can see this value.

It's a very powerful feature to **display and send any data** from your hardware to Blynk app.

☝ **Remember, that virtual pins have no physical properties.**

There are two fundamental commands you need to know to use Virtual Pins:

To read data from Blynk app widgets

Use this block of code:

```
BLYNK_WRITE(V5) // V5 is the number of Virtual Pin
{
  int pinValue = param.asInt();
}
```

Where `param.asInt()` is the value from V5.

☞ Full article:**How to control anything with Blynk**

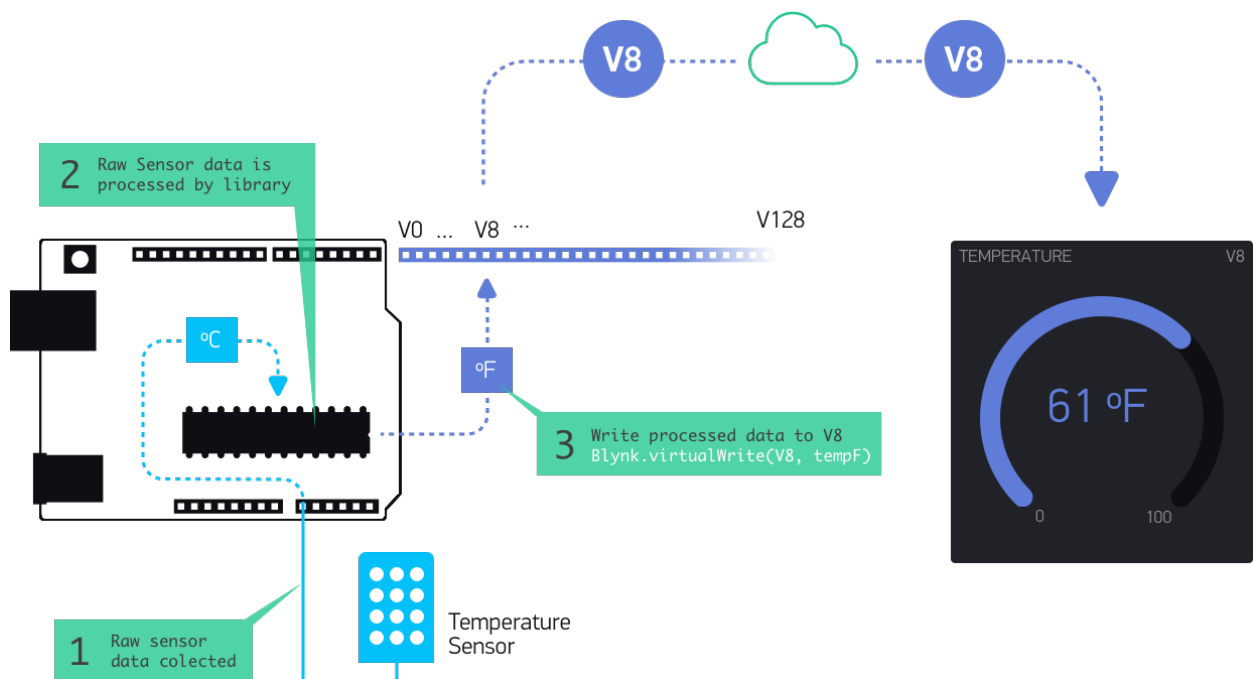## To send data from your hardware to Blynk app

Use this command `Blynk.virtualWrite(V5)` where V5 is the Virtual Pin you are using.

⚠ **WARNING:**
Don't place `Blynk.virtualWrite(V5)` **inside** `void loop()`

– Why?
– Read here

☞ Full article: **How to display any sensor data with Blynk**

If you have questions or something is not working, visit our 👥 community page.

Also, full 🏛 documentation is here.

Did this answer your question?