How to start an electronic device using a timer button

Created by: David Rubio G.

Blog post: https://soceame.wordpress.com/2025/03/09/how-to-start-an-electronic-device-using-a-timer-button/

Blog: https://soceame.wordpress.com/

GitHub: https://github.com/DRubioG

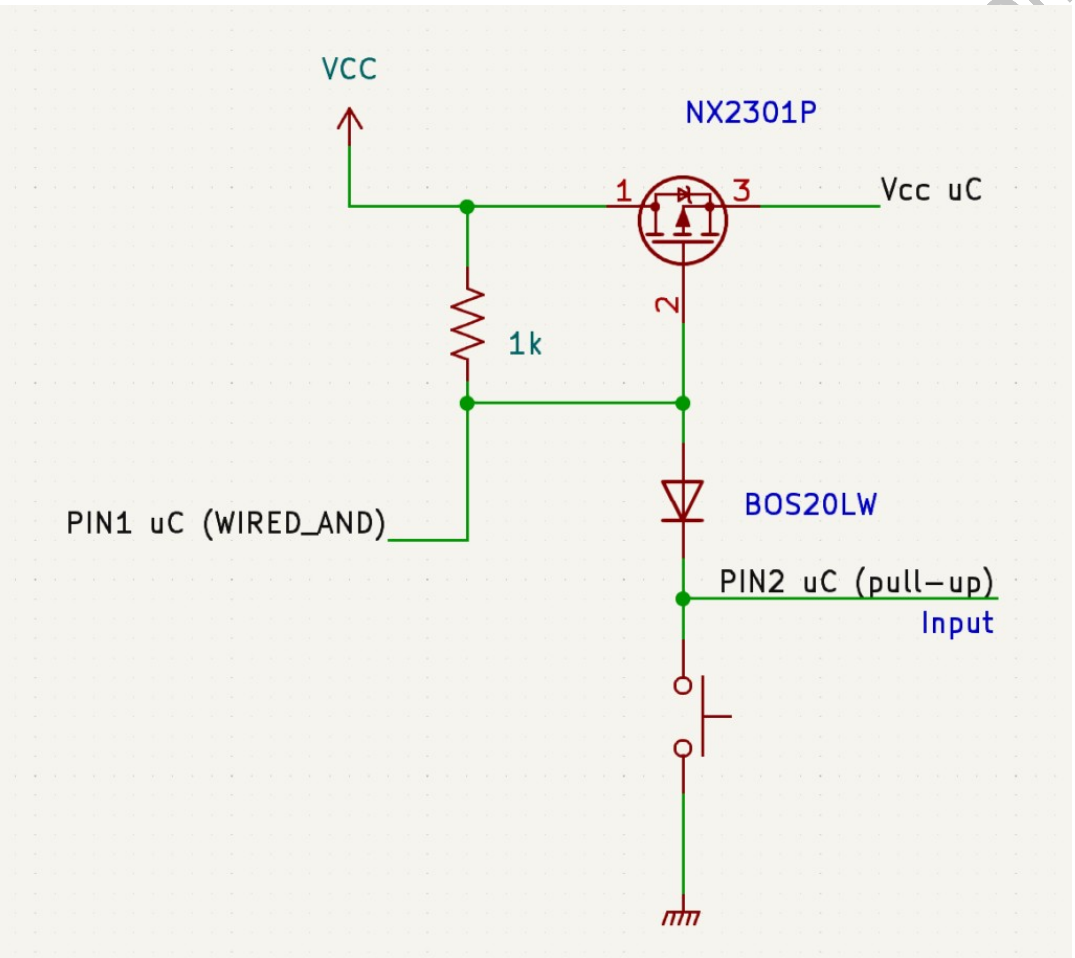
Last modification date: 09/03/25

I don't know if you have ever thought about how to start a mobile phone, a tablet, or any other electronic device using a single button. Because when you use a switch it is easy, because you use it as a current switch, which lets the current pass depending on the state it is in. But a button, how can you do it?

Well, that is what I am going to explain to you now, based on a previous work experience that I had.

Description of the model

In the following image you can see a diagram of an electronic circuit that performs that function and that I am going to explain to you below.



In the schematic you can see that there is a power supply (Vcc), with a P-type MOSFET transistor, which is placed between the general power line and the microcontroller power line (uC). This transistor is coupled by the base to a WIRED_AND pin of the microcontroller and a diode that is connected with the anode to the transistor, with the cathode to the button and to a microcontroller pin through a pull-up connection.

https://soceame.wordpress.com/2025/03/09/how-to-start-an-electronic-device-using-a-timer-button/

So, the microcontroller has three pins connected to the schematic, one for power and two pins, one for writing (PIN1 or WIRED_AND) and another for reading (PIN 2 or button reading pin).

Explanation

Now comes the explanation. To do this, you first have to know that a microcontroller executes the code that is recorded in Flash memory from the moment it is powered.

Steps:

1. The button is pressed, which powers the microcontroller, which starts executing the microcontroller's boot code, including the time-based interrupt routine. *This routine has the sole purpose of setting the microcontroller's WIRED_AND PIN1 to a low level to leave the power input transistor permanently active
2. So, while the button is pressed, the microcontroller is executing the boot routine, until two circumstances occur: **the user lifts his finger from the button**, which would cut off the microcontroller's power and the microcontroller turns off; **or the time stipulated by the programmer for the boot passes** (which can be 1, 2, 3, … seconds), **this is controlled by the time-based interrupt routine** that starts running at startup. This routine does is that if the time interruption routine is executed, the PIN1 of WIRED_AND of the microcontroller is set to **low level**, which leaves the power supply transistor permanently active, so that the finger can now be lifted from the button.

And this is how a microcontroller is started using a button.

But this does not end here, because it can also be turned off using the same system.

To do this.

1. Two interruption routines are put in place, one to read the value of the button and another for the time. In this way, the microcontroller will trigger an interruption at the same time that the button is pressed, and it will launch the time routine. This routine has to be deactivated if the finger is lifted from the button, so either the same interruption routine that activated the time routine or a new routine only for when the finger is lifted from the button has to be used.
2. If the timing routine is executed, then this routine sets PIN1 of WIRED_AND to high, which causes the input transistor to turn off and the microcontroller to shut down.

I may not have explained it in the best way, but I think you can understand how this system works (and that it can be modified so that the same function can be performed in another way).