

Modifying the Arduino Uno R4: Making it 3.3V-Friendly

 shabaz 10 Jul 2023

Table of Contents

- [Introduction](#)
- [What if you don't want to do this?](#)
 - [Inputs to the R4 board](#)
 - [Outputs from the R4 board](#)
- [Any Disadvantages?](#)
- [Arduino Uno R4 Minima Modifications](#)
- [What is Needed?](#)
- [Doing the Modifications](#)
- [Summary](#)

*Disclaimer: I *think* there is no harm in making the changes suggested in this blog post. Please don't blame me if it damages the Arduino, or any plugged-in devices, including your PC! If I have made mistakes, please let me know in the comments so everyone can benefit. Board-level modifications may invalidate any warranty.*

Introduction

The Arduino Uno R4 is designed for 5V logic levels. This is all well and good, but sometimes 3.3V logic levels are preferred or needed.

Logic-level translators could be used to perform the adaptation, among other methods.

This blog post discusses a more brutal approach, simply forcing the Uno R4 to run with a 3.3V supply instead of 5V. As a result, all inputs/outputs will be 3.3V logic level compatible, and 5V logic level operation will be gone (however, a benefit will be that some pins are still 5-V input tolerant!).

In other words, I was interested to see if the actual Arduino Uno Rev 4 board itself could be modified permanently for 3.3V operation in situations where 5V logic levels are not required at all.

In brief, the answer is yes, the Uno R4 can be converted. However, one could argue that you may as well purchase a different 3.3V-compatible Arduino, such as an Arduino MKR, or build your own custom microcontroller board, or just buy logic-level translators.

In any event, potential use-case, or lack of it, aside, this blog post will still discuss it. The modification is not too difficult; three changes need to be made to the board, but you will need a decent soldering iron tip since one of the components is 0402-sized.

This blog post applies to the Arduino Uno R4 Minima board. This blog does not cover the Arduino R4 WiFi version.



What if you don't want to do this?

You could use a different Arduino board! An example would be the Arduino MKR Zero, which has 3.3V logic operation by default. Otherwise, you could purchase a converter board of some sort, or you could build your own circuitry, which will be briefly discussed here:

Inputs to the R4 board

The normal Uno R4 logic input threshold for logic high is about 3.5-4V (depending on the input pin), although the I2C pins theoretically have a much lower threshold of 2.2V. I2C devices could interface at 3.3V (I believe; I have not tried it yet since the Arduino R4 is still new! and it depends on the Arduino I2C code too).

In any case, if you're not using I2C, then it will not be possible to connect 3.3V logic devices to the Arduino inputs without some form of level translation; consider using transistors or an IC such as M74VHC1GT50DTT1G for a single pin, or even any 74HC logic IC will do at a pinch (if you're not using a proper level translator chip). A 74HC86 quad XOR gate is convenient, and you can use it to optionally invert inputs as required too, which can be handy.

Outputs from the R4 board

For outputs from the Arduino other than I2C, one could use a resistor divider (potential divider) or, if high-speed operation is needed, logic-level translator circuitry based on active devices to drop the 5V outputs to 3.3V. An M74VHC1GT50DTT1G device can be used for a single pin. Otherwise, a device such as 74LVC240 could be used.

Any Disadvantages?

By performing the modifications described below, there are no significant disadvantages that I can think of other than that the 5V rail is gone. If you have any Arduino shields that require 5V power to come from the Arduino board, then they will no longer be compatible.

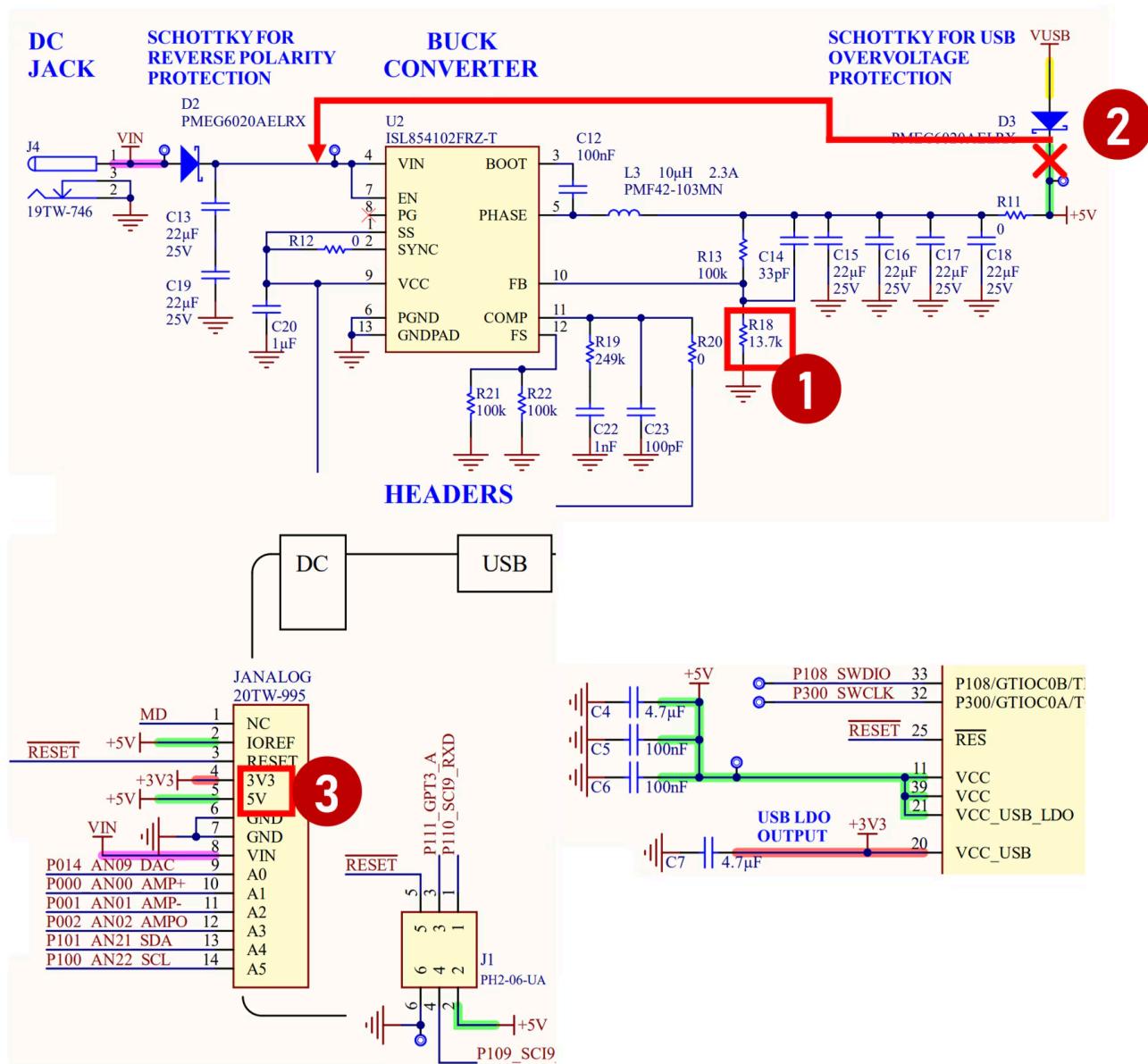
Arduino Uno R4 Minima Modifications

If you're not interested in the technical detail and want to go ahead and perform the modification, then skip to the next section!

Three modifications are needed:

- (1) A resistor is changed to convert the on-board DC-DC converter to generate 3.3V instead of 5V
- (2) A Schottky diode is moved to redirect the USB supply from the normally 5V rail to the input of the DC-DC converter instead
- (3) The 5V rail is shorted with the 3.3V rail since this is the simplest way of supplying power to the microcontroller for the USB functionality.

The circuit diagram below shows what the modifications are. The first mod is to replace the 13.7k resistor with a 22k one instead. The second mod is to desolder a Schottky diode so that one leg can be isolated and soldered elsewhere, as shown with the red arrow in the diagram below. The third modification bridges the 3.3V and 5V labeled connections so that the microcontroller's VCC_USB connection is at 3.3V (it's not strictly essential to perform this last modification, but I think it may as well be done).



What is Needed?

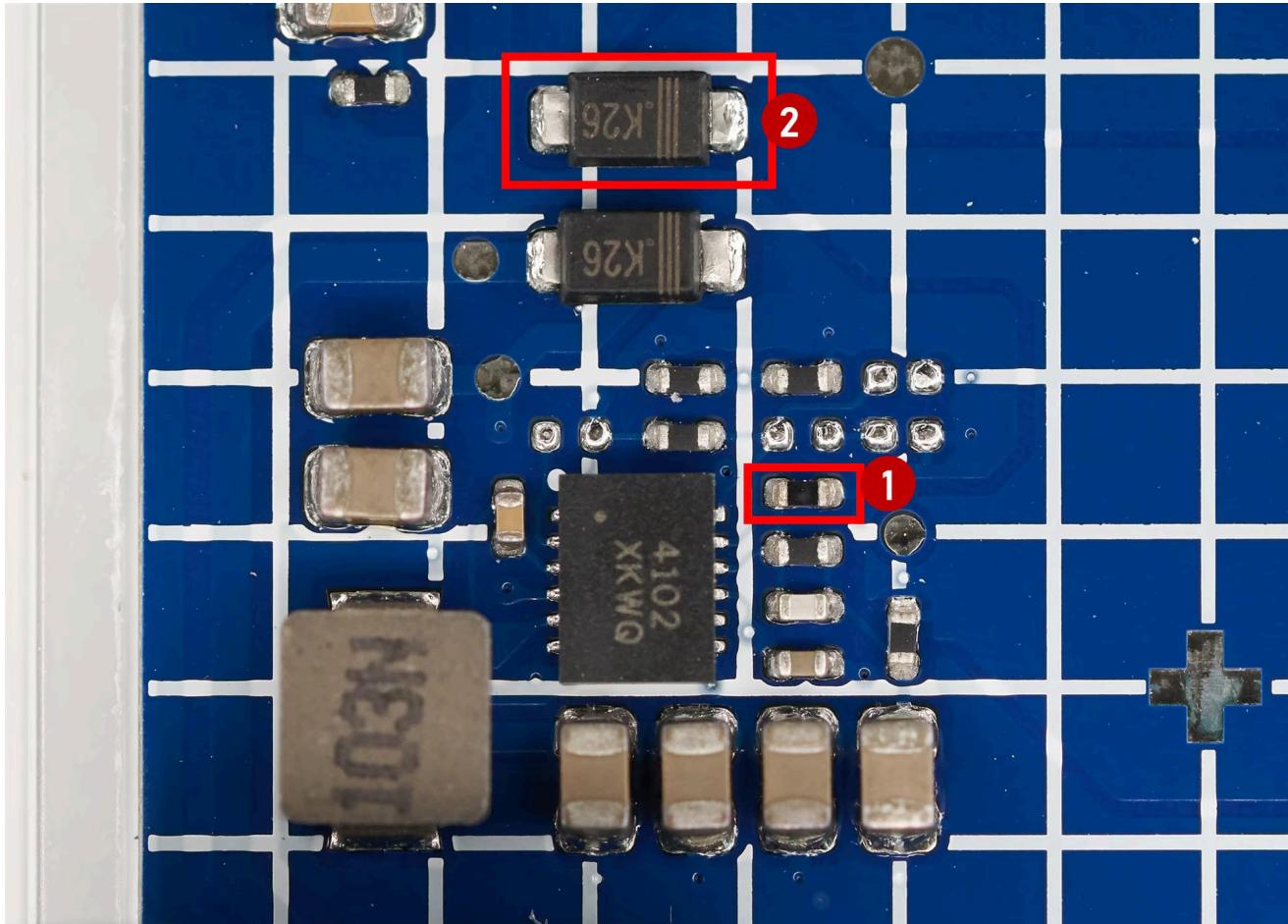
To perform the modifications, the following items are used:

- A 22k resistor (ideally 1% tolerance). The resistor should ideally be 0402 sized, but 0603 can work too. 0603 might be easier unless you have good tweezers (although you still need fairly good tweezers for 0603 size parts)
- A soldering iron with a small tip (1 mm tip max)
- Thin solder (ideally 0.38mm)
- Desoldering braid is useful for cleaning up the pads
- A head-mounted magnifier is very useful since the parts are quite small
- A small piece of Kapton tape (heat-resistant tape) to act as an insulator when isolating one end of the Schottky diode

(g) Optionally, a couple of 0603-sized zero-ohm resistors to make solder bridges more easily.

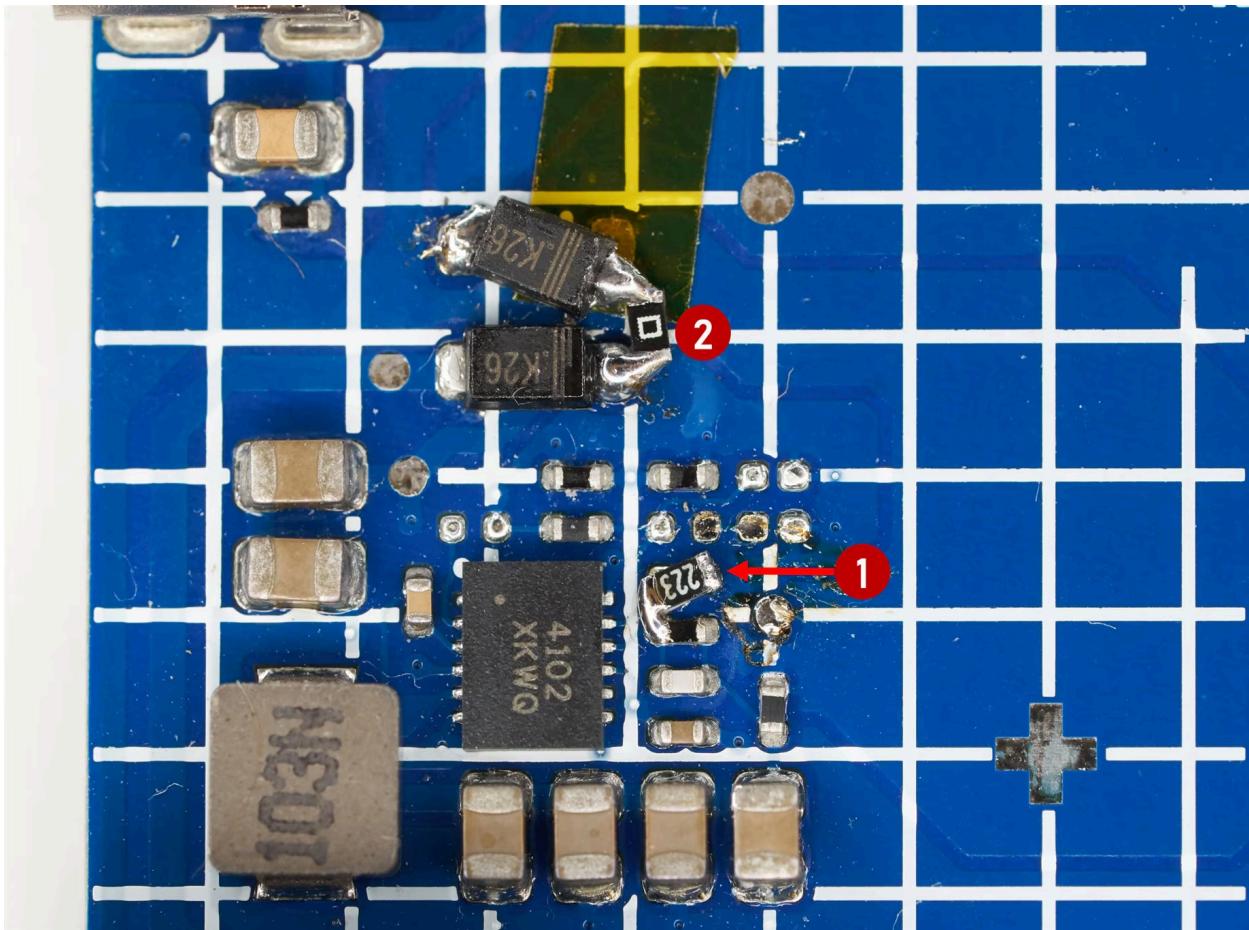
Doing the Modifications

First off, desolder the 13.7k resistor labeled (1) in the photo below, and discard it. Secondly, desolder the diode labeled (2), but keep it!



Replace the resistor with a 22k one, as shown in the photo below. If, like me, you only have an 0603 sized resistor instead of a 0402-sized one, then it is possible to pre-tin the right side of the resistor, place it at an angle, tack down the right side, and then bridge the left side with a solder blob, to attach it to the adjacent resistor.

For the earlier removed diode, place a small piece of Kapton tape on the right side footprint pad on the board, and then resolder the diode (with the correct orientation as shown in the photo). Bridge the right side to the lower diode. I used a 0603-sized zero-ohm link to help with that.



Examine with a magnifier, and ensure no other pads were accidentally shorted during the operation. There are several pads very close to the resistor, and they must not be accidentally bridged.

Finally, flip the board over (it lifts out of the plastic carrier easily) and bridge the 3.3V and 5V connections. I used another zero-ohm link for that:



Power up the board (ideally without initially involving your PC), and verify that the 5V and 3.3V header connections are now both at 3.3V. The Arduino R4 is now 3.3V-ready!

Summary

Three simple changes are needed to convert the Arduino Uno R4 to 3.3V logic levels. A 22k resistor is needed, but since the part to be removed is quite small (0402-sized), some experience will be needed working with surface-mount parts.

Please share any findings if you perform the modifications. Any feedback or improvements are welcome.

Thanks for reading!



[Fred27](#) over 1 year ago

Can I suggest one more step? Label it clearly as modified so that when someone - maybe you, maybe someone else - grabs it from a drawer in a year's time they aren't left wondering why it doesn't work (or where that smoke is coming from).



[misaz](#) over 1 year ago

MCU supports 1.8V also but this would be even more challenging 😊



[shabaz](#) over 1 year ago in reply to [misaz](#)

Hi! Yes, modification #3 would need to change to a more intricate one, whereas the shortcut I used only works fine for 3.3V levels. Also, there is a chance some other tweaks could be needed! It's definitely worth exploring one day, but I don't have any more boards to experiment with! This one will remain at 3.3V levels for a non-arduino project where I needed a board with this microcontroller on it, and it saves me having to build a custom board for that so soon!



[navadeepganeshu](#) over 1 year ago

Nice workaround. Yeah, often this makes sense while Arduino has to be incorporated in a system all with 3.3V. Or even sometimes I've seen people use Arduino just as a 5V supply for LCD display,...haha

Between, those are pleasing images of the PCB rework. Which camera do you use for these captures?



[shabaz](#) over 1 year ago in reply to [navadeepganeshu](#)

Hi Navadeep,

Thanks! Afterwards, I was thinking it might now also work with a single LiPo battery, but I've not tried that.

The photos were with a Sony A6300 (it is reasonable-cost in used condition).



[Fred27](#) over 1 year ago

Oops. I forgot to say "nicely done" first. That's a very clear and manageable modification. I must admit I'm surprised that 5V logic levels are still a thing in 2023.



[Jan Cumps](#) over 1 year ago in reply to [Fred27](#)

> I must admit I'm surprised that 5V logic levels are still a thing in 2023.

The Arduino designers did this on purpose. Their new designs are all 3V3 - and don't use the Arduino odd connector placement anymore. The R4 is made for maximum R3 compatibility: The classic R3 connector placement, and additional electronics to get that 5 V logic level.



[shabaz](#) over 1 year ago in reply to [Fred27](#)

Hi Fred,

Thanks! Very good point to label it to distinguish from unmodified boards!

I don't have any (or maybe at most one or two) Arduino shield boards, so much preferred the 3.3V levels.



[DAB](#) over 1 year ago

Nice mod.



[shabaz](#) over 1 year ago in reply to [DAB](#)

Hi DAB,

Thanks!

▼ [View More](#)