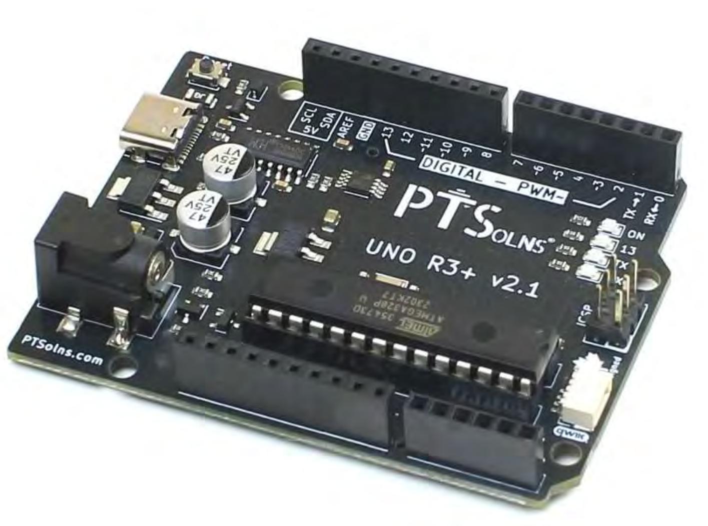
Uno R3+

1 DESCRIPTION



The PTSolns *Uno R3+* based on the ATmega328P, is a popular microcontroller development board with additional unique features. The *Uno R3+* continues to be compatible with a range of shields, yet it has two extra female header pins. These pins are used to supply SDA and SCL that are shifted from the onboard logic level of 5 V down to 3.3 V. Along with the onboard QWIIC® connector, this shifted 3.3 V I2C bus makes it convenient to connect many common sensors and modules that operate on 3.3 V logic No need for extra external components, all the logic level shifting between 5 V and 3.3 V on the I2C bus is done automatically onboard.

The *Uno R3+* replaces the previous bulky port with a modern USB-C port. Meanwhile the standard power barrel jack remains onboard. The board has a new layout of components, such as for example, the grouping of the four LEDs (Power LED, Pin 13 controlled LED, TX and RX communication LEDs). Furthermore, the *Uno R3+* allows for a larger current draw on the 3.3 V pin with a rated maximum of 0.8A. This lets the user manage their project's power budget more easily by allowing a larger current draw.

This version of the *Uno R3+* contains the ATmega328P as a 28-Pin through-hole component that can be removed from the board and used elsewhere, as in for example the popular entry-level project "Make Your own Uno". The *Uno R3+* comes with a pre-installed out-of-the-box ready set of tests. This gets simple projects started quickly and allows for testing the proper function of the board. A large active online community supports this microcontroller and its programming using the Arduino IDE software. Many tutorials, projects, schematics, and sketches can be easily found online in blogs, forums and the like.

This *Plus* version offers all of the features users come to learn and expect, and adds additional ones that make this board stand out.

All *Uno R3+* boards are individually inspected and marked with a quality control sticker (either on the PCB, or on the packaging).

Table of Contents

2 DOCUMENT REVISION HISTORY

Current document revision is Rev 1.

Changes to Rev 1

1. Updated images.
2. Added Section 1.1.1.
3. Updated Section 6.1.
4. Added to references in Section 7.
5. Updated Section 5, added Sections 5.1 and 5.2.
6. Added Section 6.4.
7. Added Section 6.5.

3 PRODUCT FEATURES

This section highlights notable features of the *Uno R3+*.

3.1 I/O Connectors & Input Power

The *Uno R3+* has three input/output (I/O) connectors as follows:

- USB-C
- Power barrel jack (2.1 mm x 5.5 mm)
- QWIIC®

These connectors are shown in Figure 1. Note that the QWIIC connector adheres to the standardization as outlined by SparkFun Electronics: https://www.sparkfun.com/qwiic

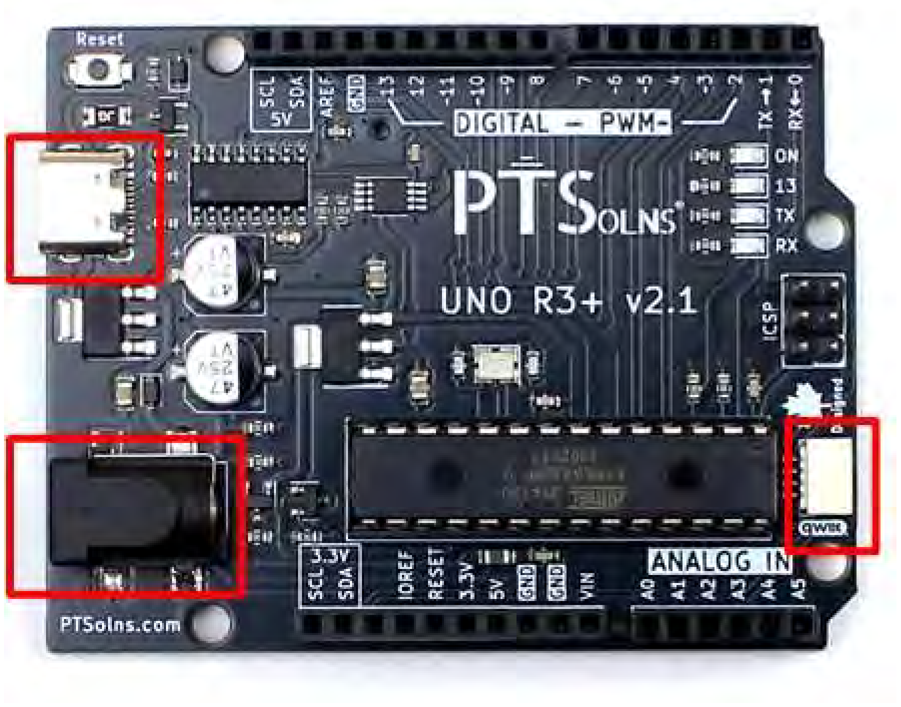


Figure 1: The three I/O connectors on the Uno R3+.

The *Uno R3+* can be powered in multiple ways, with the most common two methods being the USB-C port and the power barrel jack. If the user connects both at the same time, the board automatically selects the voltage from either source depending on the voltage level of the power jack. The user can also power the board using the "VIN" pin on the female header with the same voltage input ratings as for the power barrel jack (i.e. Vin = 7-12 V max).

The board can also be powered directly by supplying a clean and steady 5 V DC to the "5V" pin on the female header. This bypasses the onboard 5 V voltage regulator, and can be done if an external 5 V source is available. **Note that the ATmega328P should not exceed an input voltage of 5.5 V and therefore the user needs to exercise caution when powering the *Uno R3+* from the 5 V pin directly.**

3.1.1 Note on USB-C Cables

From a data transfer perspective USB-C cables can be categorized into two types:

1. Data transfer capable
2. Not data transfer capable

Only the first type of USB-C cable can be used to program not just the *Uno R3+*, but indeed any microcontroller. This type of cable provides power to the board, as well as facilitates data transfer between the computer and the board. **Using this type of cable is essential in programming a microcontroller**.

The second type of USB-C cable does not facilitate the transfer of data but can merely be used to power the board. Therefore, this type of USB-C cable cannot be used to program a microcontroller.

How to tell if a USB-C Cable is Data Transfer Capable?

One can easily and quickly check if a particular USB-C cable is data transfer capable by simply trying to program the *Uno R3+*. If the USB-C cable is not data transfer capable, then upon plugging it into the computer with the other end into the *Uno R3+* no port will appear.

3.2 I2C Bus

The *Uno R3+* offers an I2C bus on the SDA and SCL lines, inherently at the ATmega328P 5 V logic level. For the ATmega328P the 5 V SDA and SCL pins are the A4 and A5 analog pins, respectively. For convenience these pins are also available on the top-left female header breakout marked as “SCL” and “SDA” in a box with “5 V”, as seen in Figure 1.This is to clearly indicate to the user that this part of the I2C bus operates on 5 V logic.

The *Uno R3+* has an onboard Logic Level Shifter (LLS) that shifts the 5 V I2C bus down to 3.3 V. These corresponding I2C pins are available on the bottom-left female header breakout marked as “SCL” and “SDA” in a box with “3.3V”, as seen in Figure 1. This is to clearly indicate to the user that this part of the I2C bus operates on the 3.3 V logic.

The I2C bus at 3.3 V is also made available in the form as a QWIIC connector. This connector is available to the right of the ATmega328P marked with the trademark “QWIIC” logo, as seen in Figure 1. Note that the QWIIC connector adheres to the standardization as outlined by SparkFun Electronics: https://www.sparkfun.com/qwiic

The user can connect multiple peripherals to the *Uno R3+* via I2C simultaneously. It is possible to connect to the 5 V I2C, shifted 3.3 V I2C as well as the QWIIC connector all at once as long as the following requirements are met:

- The total number of I2C connected peripherals does not exceed 126 units.
- The power budget of the project does not exceed any current rating (See Section 5).
- Each device address is unique (alternatively an I2C MUX can be used to connect peripherals with the same address on the same I2C bus).
- There is not too much capacitance introduced by long connecting wires. The rise-time of the I2C signals as determined by the corresponding time constant must be within acceptable range (See discussion below).

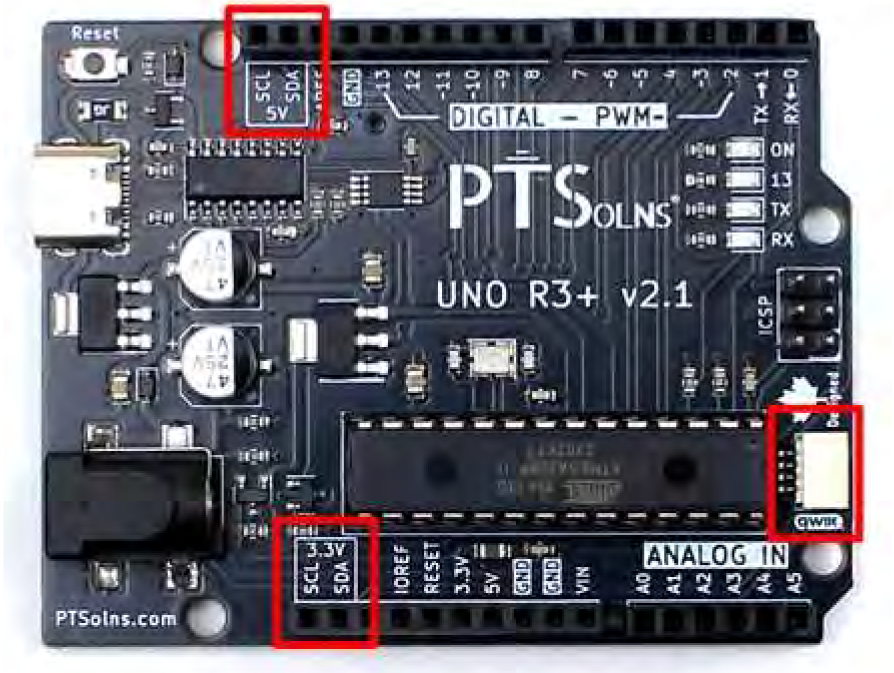


Figure 1: I2C pins at 5 V and 3.3 V, as well as the QWIIC connector, on the Uno R3+.

The default connection for the 3.3 V SDA and SCL pins on the *Uno R3+* is such that it is enabled. This means, the *Uno R3+* out-of-the-box will have a functioning 3.3 V I2C bus as seen near the bottom-left (female header pins) and middle-right (QWIIC connector) in Figure 1. However, it is possible to disconnect these pins by cutting the jumper pads labelled “3.3V SDA” and “3.3V SCL” as shown in Figure 2. When these pins are cut the 3.3V SDA and SCL pins and QWIIC connector are floating.

Furthermore, there are additional pull-up resistors on the 5 V I2C bus each of 10 kΩ. These can also be disconnected by cutting the jumper pads labelled “10k Pull-up 5V SCL” and “10k Pull-up 5V SDA” as shown in Figure 2. A full schematic of the Logic Level Shifter (LLS) showing the 5 V and the 3.3 V sides of the I2C bus, as well as the four jumper pads, can be seen in Figure 3.

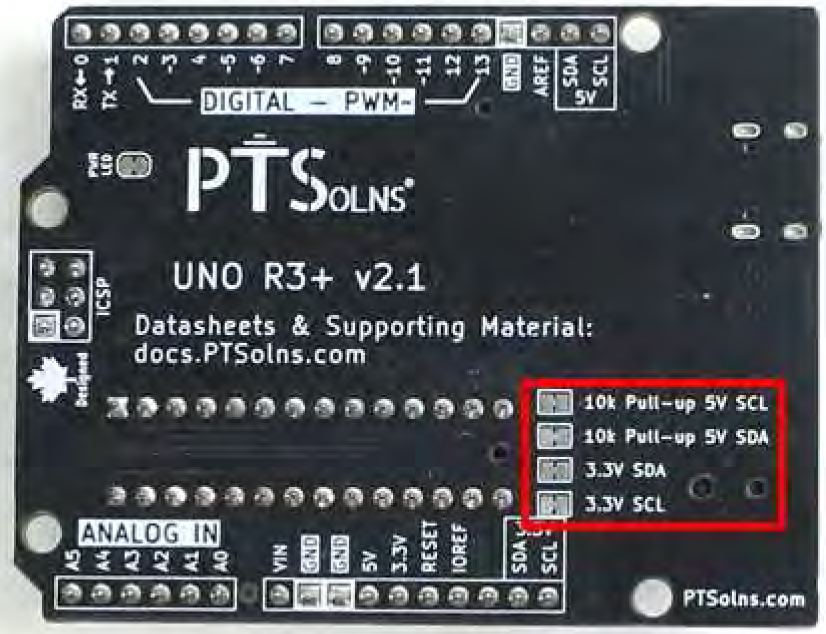


Figure 2: Jumper pads on the back of the Uno R3+ (all closed by default).

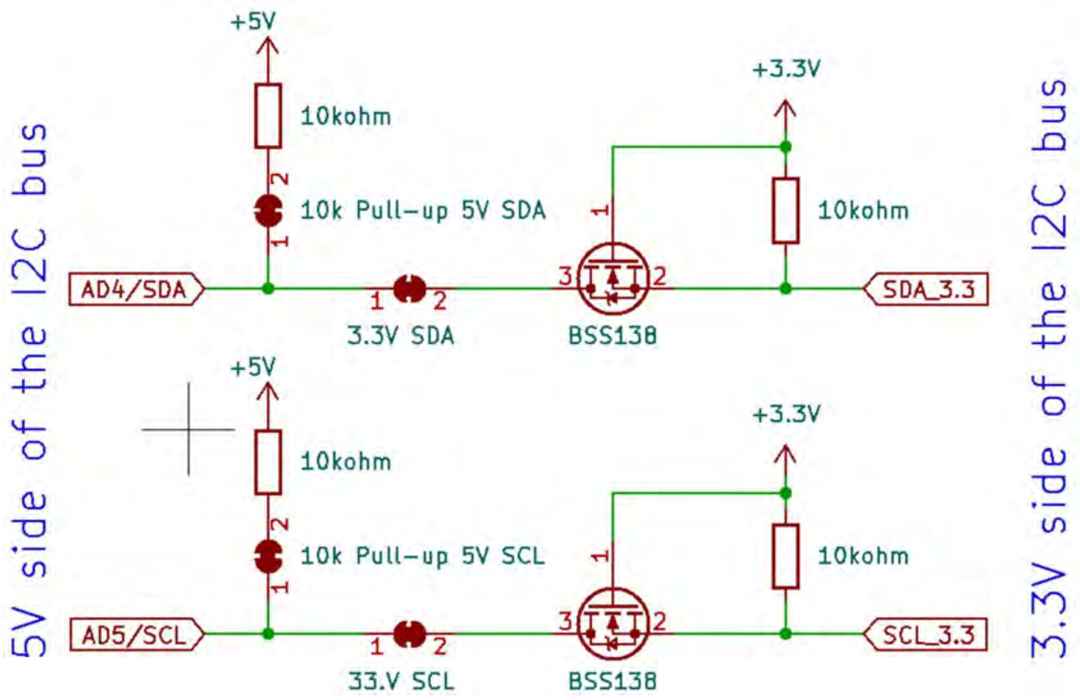


Figure 3: Logic Level Shifter (LLS) schematic on the Uno R3+ showing the 5 V and the 3.3 V sides of the I2C bus.

The user is made aware that if too many I2C connected peripherals are used at once, or the connecting wires are long (around 1.5m or longer), the I2C communication may start to break down. This is not a deficiency of the *Uno R3+* but rather a characteristic of I2C communication itself. When the SDA signal rise-time starts to take too long and becomes sluggish (because the time constant is too large, which is a function of resistance and capacitance on the I2C bus), then data may become corrupt. Often this will result in the microcontroller freezing and a restart may be required. This scenario is easily diagnosed with an oscilloscope. The user can try the follwoing simple changes to improve the I2C bus quality:

- Reduce the communication wire on the SDA and SCL lines.
- Add or decrease pull-up resistors on the SDA and SCL lines. Note that many peripherals have their own pull-ups onboard, and when connecting many of such modules on the I2C bus that the total resistance acts as if in parallel, thus decreasing the overall resistance. The user may have to decrease the resistance or disable entirely. Alternatively, there may not be enough resistance and so larger resistance may need to be added.
- Reduce the clock speed of the SCL line. The slower the communication speed, the less sluggish or slow rise-time of the SDA line impacts the communication. Typically, the default SCL clock speed is 100 kHz. Although there is no theoretical lower limit for I2C, some peripherals have issues working below 10 kHz.

3.3 ICSP Breakout

The *Uno R3+* has an In-Circuit Serial Programming (ICSP) breakout section via a 2X3 Pin male header, as shown in Figure 4. The pin breakout of the ICSP is shown in the schematic in Figure 5, with the following definitions:

- CIPO: Controller In, Peripheral Our (formerly called MISO)
- SCK: Serial Clock
- COPI: Controller Out, Peripheral In (formerly called MOSI)
- Reset: Reset

For orientation of the pins, the ground "GND" pin is outlined with a white printing, best seen from the back of the board.

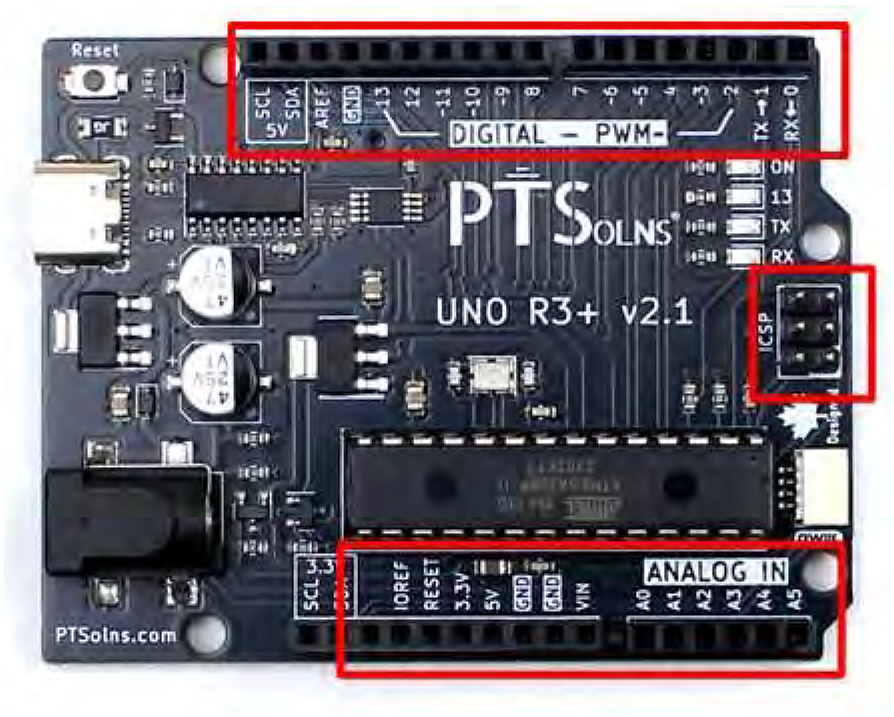


Figure 4: Breakout pins of the Uno R3+.

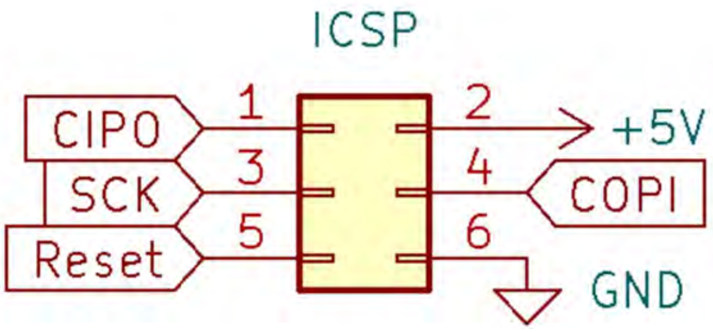


Figure 5: ICSP breakout pins.

3.4 Compatibility with Shields (Stackability)

The *Uno R3+* is fully stackable and thus compatible with a range of common shields. Therefore, all of the regular breakout pins both be the female headers as well as the male header are available, as shown in Figure 4. The *Uno R3+* does have two extra pins on the female header labelled with “SCL” and “SDA” in a box with “3.3V” (See Section 3.2). For most shields these two extra pins do not interfere with a shield stacked on top.

3.5 Component Arrangement & Pinout Diagram

The *Uno R3+* has a unique component arrangement. As an example, the four onboard LEDs (Power LED, Pin 13 controlled LED, TX and RX communication LEDs). Some notable component arrangements, including the LEDs, Reset button and the ATmega328P, are shown in Figure 6.

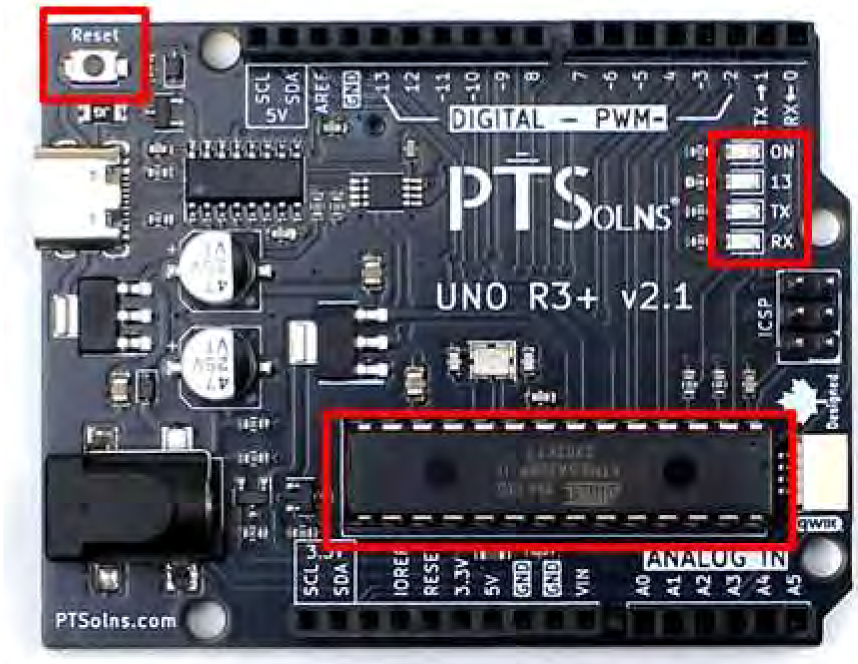


Figure 6: Notable component arrangements of the Uno R3+.

The *Uno R3+* has 14 digital Input/Output (I/O) pins, referred to as “DIO”, of which six are PWM capable. There are six analog input pins, and various power related pins. All of the pins are shown in the pinout diagram in Figure 7.

Note the following:

- A tilde next to a DIO indicates that pin is PWM capable.
- DIO 13 also controls the onboard LED labelled “13” as shown.
- The Reset pin is connected to the push button as well as the pinout on the ICSP.

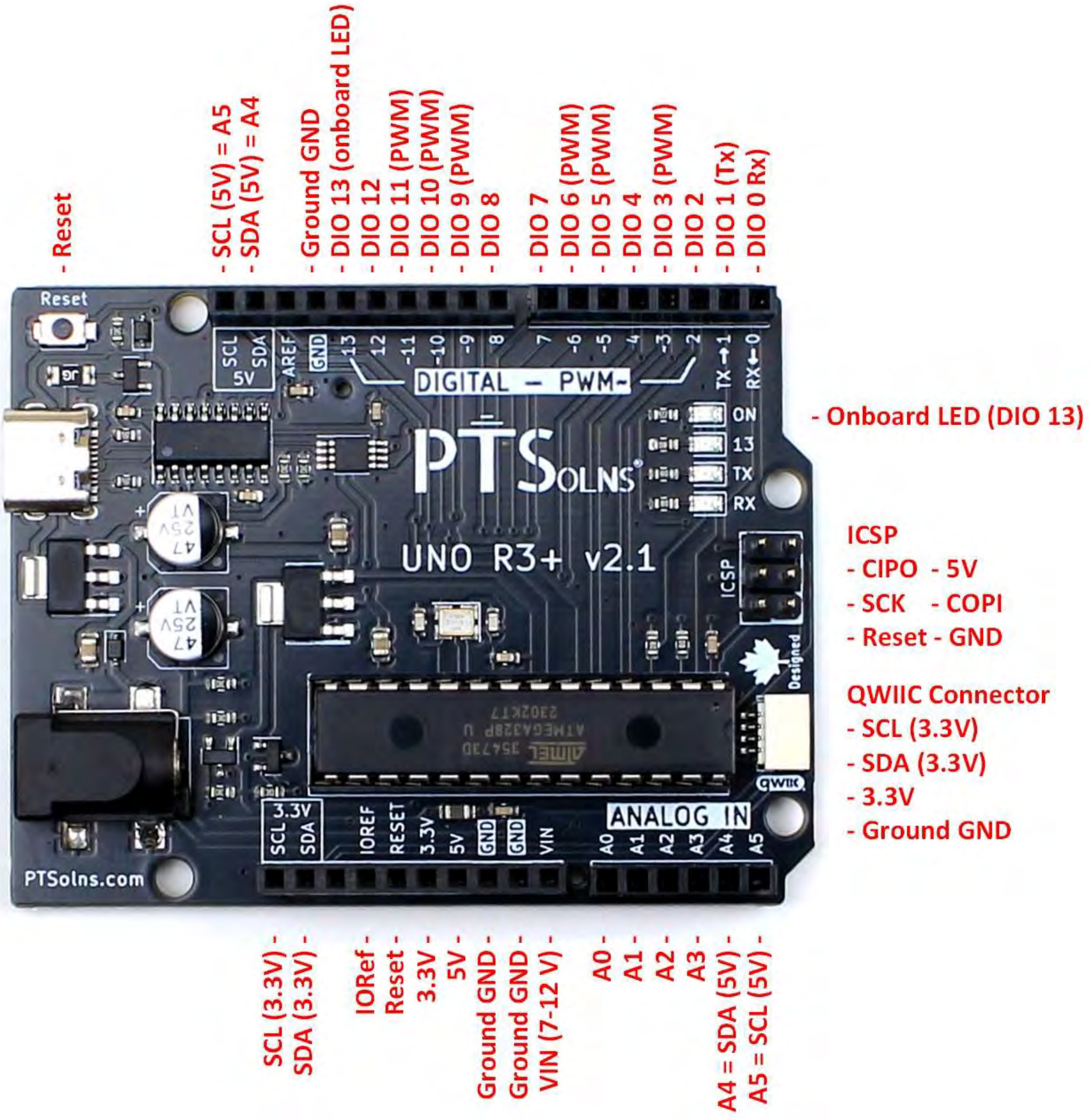


Figure 7: Pinout diagram for the Uno R3+.

3.6 Mark of Authenticity

Authentic PTSolns PCBs have a black solder mask color and are marked with the “PTSolns” logo in white silkscreen printing. The “Canadian Designed” symbol, consisting of the Canadian Maple Leaf with the word “Designed” underneath, can also be found on the PCB in white silkscreen printing. The “PTSolns” trademark and the “Canadian Designed” symbols are shown in Figure 8.

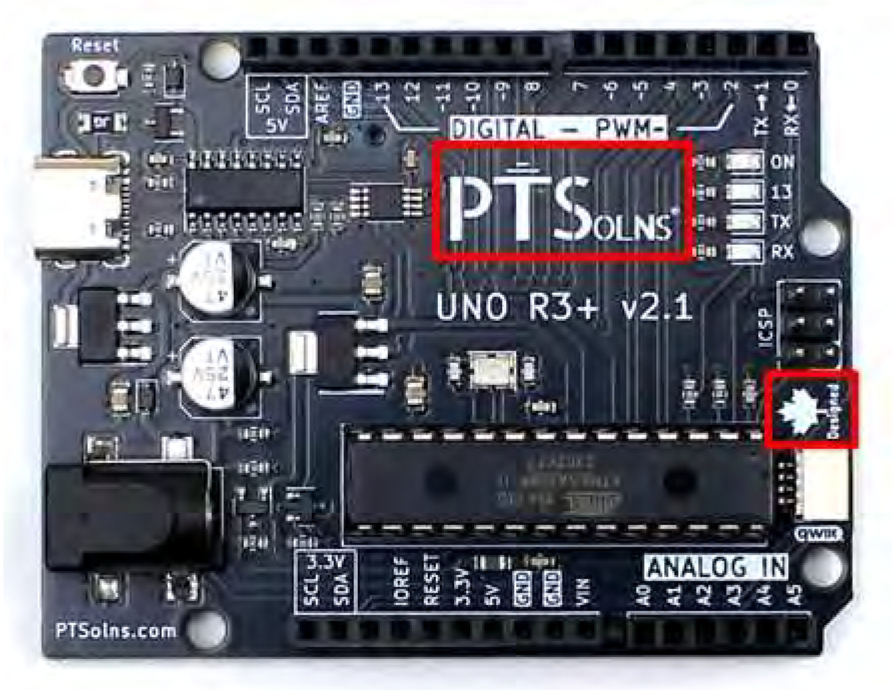


Figure 8: The "Canadian Designed" symbol found on authentic PTSolns PCBs.

4 PHYSICAL PROPERTIES

The physical properties of the *Uno R3+* are outlined in Table 1.

Table 1: Physical Properties.

	Quantity	Value	Reference
PCB	Length (longest)	68.6 mm	Figure 9
	Width (longest)	53.3 mm	Figure 9
	Thickness	1.6 mm	--
	Weight (with IC)	24 g	--
	Color	Black	--
	Silkscreen	White	--
Material	Lead free HASL-RoHS surface finish		--
	FR-4 base		--
Mounting Holes	4x each with 3.2mm diameter		Figure 10

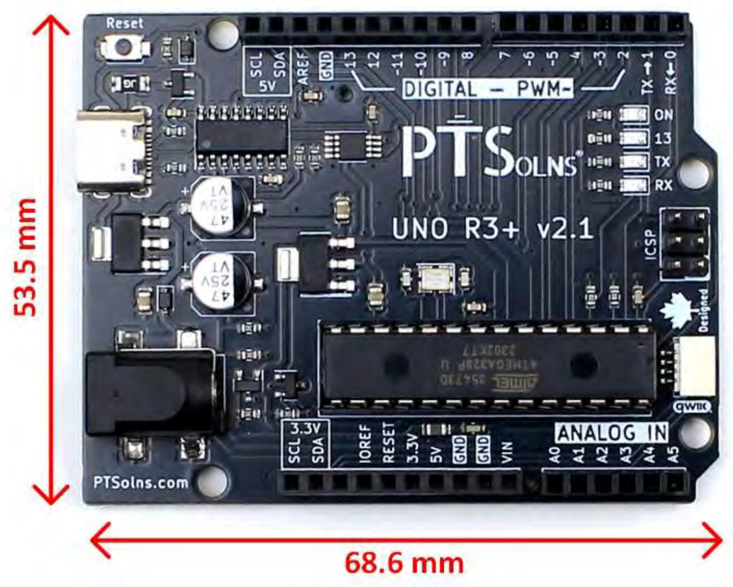


Figure 9: Dimensions of the Uno R3+.

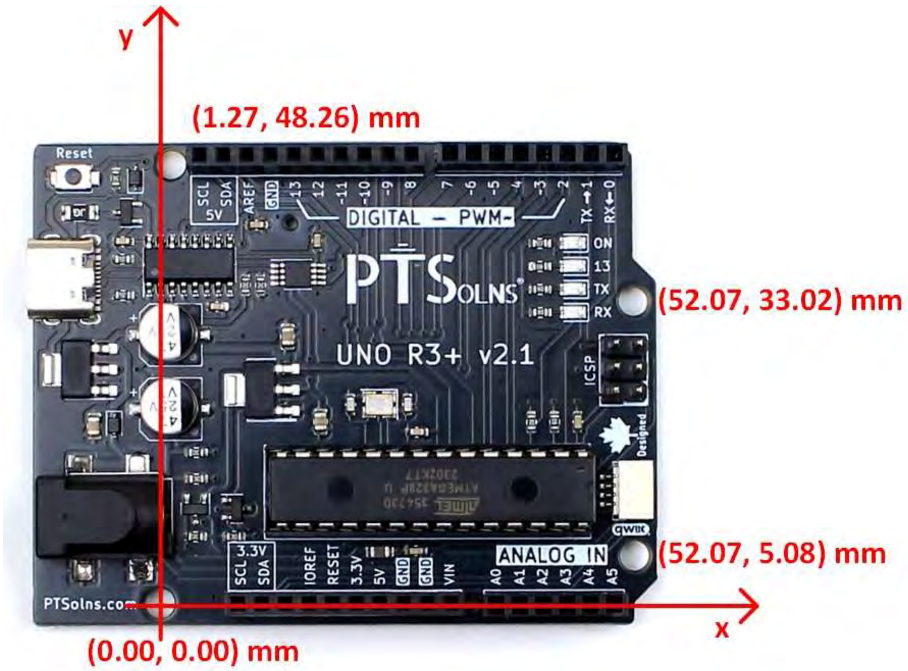


Figure 10: Positions of mounting holes. Coordinate grid zeroed on the bottom-left mounting hole.

5 ELECTRICAL PROPERTIES

The electrical ratings for the *Uno R3+* are outlined in Table 2.

Table 2: Electrical ratings for the *Uno R3+*.

Type of trace	Rating
Input voltage on USB-C	5 V
Input voltage on power barrel jack	7-12 V
Operating current draw on any single GPIO	20 mA
Max current on any single GPIO	40 mA **(Do not operate at this level for extended periods)**
Max combined current on all GPIO	200 mA **(IF OPERATING in "Stable" CONDITIONS. See Section 5.1 for important details)**
Max current draw on 3.3 V pin	800 mA **(IF OPERATING in "Stable" CONDITIONS. See Section 5.1 for important details)**
Max current draw on 5 V pin	800 mA **(IF OPERATING in "Stable" CONDITIONS. See Section 5.1 for important details)**
Max combined current draw on all GPIO and power pins (Total External Current Draw (TECD))	800 mA **(IF OPERATING in "Stable" CONDITIONS. See Section 5.1 for important details)**

The current ratings for the two power pins, the 3.3 V and 5 V pins, are discussed in more detail in Section 5.1. The voltage ratings are discussed in Section 5.2.

5.1 Current Rating

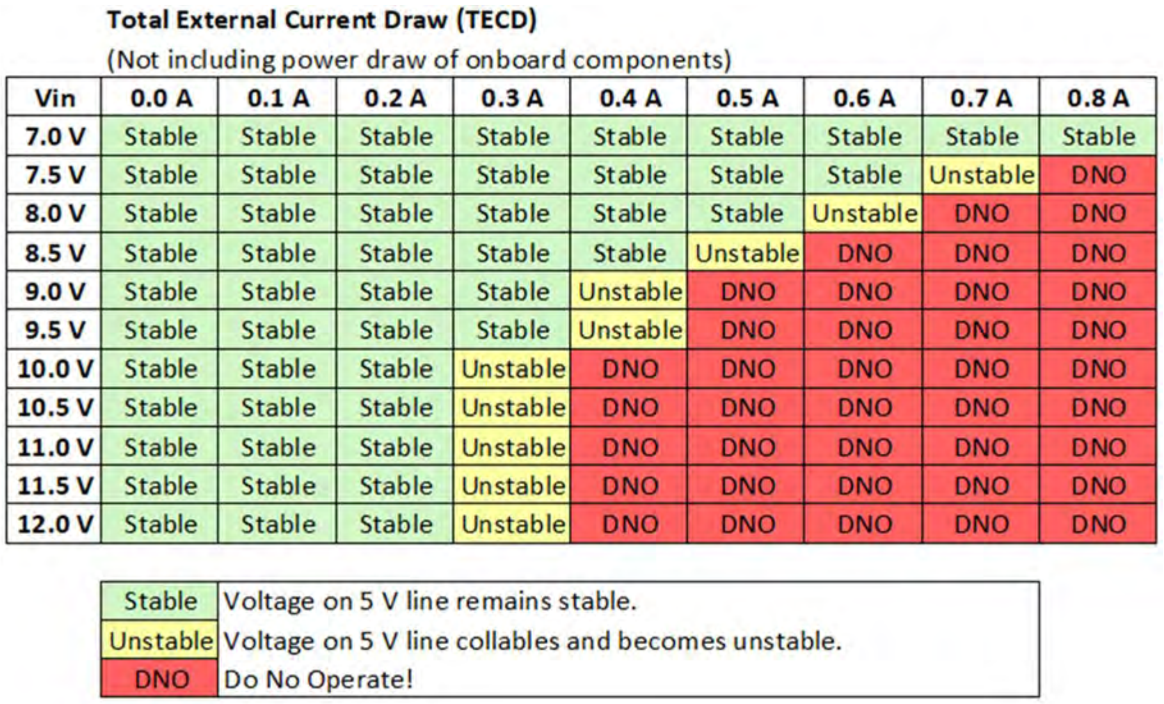
The onboard components on the *Uno R3+* consume approximately 20 to 40 mA, depending on the LEDs and the ATmega328P microcontroller demand. External current draws can be made by employing any of the General-Purpose Input/Output (GPIO) pins, such as the digital or analog pins, as well as the 3.3 V and 5 V power pins. The 3.3 V and 5 V power pins can be used to power external sensors and modules. The combined current draw of any GPIO and power pins used is the Total External Current Draw (TECD).

The TECD is delivered by an onboard 5 V voltage regulator. Depending on the input voltage (Vin) supplied to the regulator, as well as the TECD, the *Uno R3+* may operate in the "Stable" region or the "Unstable" region. This is shown in Table 3. In this context, "Stable" is defined such that the 5 V line remains constant, with a small ripple, within acceptable tolerances. "Unstable" is defined such that the 5 V line starts to drop below unacceptable tolerances, collapses, or rises above 5 V plus an acceptable tolerance. The user should only operate the *Uno R3+* in the "Stable" region. In extreme unstable regions (e.g. Vin = 12 V, TECD = 800 mA), the voltage regulator may allow the input voltage through to the 5 V line. This can cause damage to components downstream, and the user must avoid such extreme conditions at all times.

As an example, at an input voltage of Vin = 7 V, the TECD can reach the maximum of 800 mA. With increasing input voltages, the TECD in which the *Uno R3+* operates in the "Stable" regions starts to reduce.

Therefore, the user should take care that all current draws on external pins (GPIO, 3.3 V and 5 V power pins) remains in the "Stable" region for a given input voltage Vin, as outlined in Table 3.

Table 3: TECD Operating Conditions.



Total External Current Draw (TECD)
(Not including power draw of onboard components)

Vin	0.0 A	0.1 A	0.2 A	0.3 A	0.4 A	0.5 A	0.6 A	0.7 A	0.8 A
7.0 V	Stable	Stable	Stable	Stable	Stable	Stable	Stable	Stable	Stable
7.5 V	Stable	Stable	Stable	Stable	Stable	Stable	Stable	Unstable	DNO
8.0 V	Stable	Stable	Stable	Stable	Stable	Stable	Unstable	DNO	DNO
8.5 V	Stable	Stable	Stable	Stable	Stable	Unstable	DNO	DNO	DNO
9.0 V	Stable	Stable	Stable	Stable	Unstable	DNO	DNO	DNO	DNO
9.5 V	Stable	Stable	Stable	Stable	Unstable	DNO	DNO	DNO	DNO
10.0 V	Stable	Stable	Stable	Unstable	DNO	DNO	DNO	DNO	DNO
10.5 V	Stable	Stable	Stable	Unstable	DNO	DNO	DNO	DNO	DNO
11.0 V	Stable	Stable	Stable	Unstable	DNO	DNO	DNO	DNO	DNO
11.5 V	Stable	Stable	Stable	Unstable	DNO	DNO	DNO	DNO	DNO
12.0 V	Stable	Stable	Stable	Unstable	DNO	DNO	DNO	DNO	DNO

Stable	Voltage on 5 V line remains stable.
Unstable	Voltage on 5 V line collables and becomes unstable.
DNO	Do No Operate!

NOTE 1: Drawing full allowed current and temperature of 5 V voltage regulator

If the TECD is high the voltage regulator will get hot. This is unavoidable and depends greatly on the input voltage. The component's temperature depends on how much power (in Watts) it has to dissipate. This power is a result of the voltage difference between the Vin and Vout of the voltage regulator, times the current being draw. This can be expressed in the following formula:

$$Power_{dissipate\ on\ volt.reg} = (V_{in} - V_{out}) * I$$

The input voltage (discussed more in Section 5.2) ranged from 7 V to 12 V. The most power entering in the voltage regulator needing to be dissipated in terms of heat is when Vin = 12 V and the TECD is the maximum of 800 mA. In that case, the power to be dissipated becomes (12 V – 5 V)*800 mA = 5.6 W. This is a lot of power for such a small SMD component. The component will heat up! The 5 V voltage regulator will automatically shut off when the internal component temperature reaches 145 degrees C. The component restarts when the temperature is below a threshold.

The *Uno R3+* was developed with the temperature and current ratings in mind. The traces within the PCB that carry the current have been made very wide to reduce the temperature. This is particularly the case for the trace to the 5 V power pin. Furthermore, thermal vias were added below both the 5 V and 3.3 V voltage regulator. These vias take the surplus heat from under the component and bring it to the other side of the PCB where there is more surface area to dissipate the heat.

5.2 Voltage Rating

The 5 V voltage regulator determines the maximum and minimum acceptable voltage input a user can supply. The maximum is set to 12 V. However, the regulator can accept short momentary voltage spikes up to 14 V caused by the external power source. The user should be careful to never exceed the 12 V rating as otherwise damage to the components can result. If the input voltage is too high the regulator can malfunction, allowing high input voltage on the 5 V line, causing other components downstream on the 5 V line to be damaged, including the ATmega328P. Therefore, if wanting to supply the *Uno R3+* with a 12 V source, particularly from a battery or an unstable buck/boost converter, ensure that the voltage is not above the 12 V rating.

The minimum input voltage is specified as 7 V. With 7 V the *Uno R3+* can reliably be operated. That being said, the voltage can likely be a little bit lower. It depends on how much current is being drawn through the 5 V voltage regulator, as well as the fuse settings on the ATmega328P. The higher the current draw, the larger the dropout voltage is. At 800 mA current draw, the regulator drops ~1.45 V. An input voltage of 6 V might work well for very low external current draws. The user is encouraged to experiment if in their project setup a lower input voltage is acceptable.

6 USAGE AND APPLICATION

This section presents important information regarding the first-time usage of the *Uno R3+*.

6.1 CH340 Driver

The *Uno R3+* uses the common CH340 IC to facilitate communication between what is plugged into the USB-C port (e.g. user's laptop) and the microcontroller (ATmega328P). This IC is required when programming the *Uno R3+.* The driver for the CH340 typically must be installed the first time it is needed in any project. Many boards and modules make use of the CH340 so chances are that the driver is already installed. However, if the driver is not yet installed, the user must first install it in order to program the *Uno R3+.* This installation typically only must be done once.

Instructions on how to install the CH340 driver (on a Windows machine) can be found in the following reference:

https://www.youtube.com/watch?v=UUQ84VKg3oM

Additionally, SparkFun Electronics has written a comprehensive tutorial on this topic, and the user is referred to their excellent documentation on this. Find the link here:

https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all

There are many other tutorials available online that can be found by searching "CH340 driver installation instructions" or similar.

Once the CH340 driver is installed, the *Uno R3+* can be programmed. See Section 3.1.1 regarding using a proper UCB-C that is data transfer capable. See Section 6.4 for details on how to use Arduino IDE to program the *Uno R3+*.

6.2 Inserting the ATmega328P IC

When inserting the ATmega328P IC into the corresponding 28-Pin IC socket the orientation is important. The user must take care to never inset the IC the wrong way around. The ATmega328P has two markings to indicate the Pin 1 and the orientation/Pin 1. Figure 11 shows these two markings.

The *Uno R3+* also has two markings on and near the corresponding 28-Pin IC socket, indicated by a white silkscreen dot which can be seen through the slots of the IC socket. The ATmega328P IC must be inserted into the IC socket with all the orientation and Pin 1 markings aligned. As in the case of the image in Figure 12, the markings are all on the right side (towards the QWIIC connector). The writing on the ATmega328P should be upside when viewing the *Uno R3+* in the same orientation as in Figure 12.

NOTE: It is recommended that the power to the *Uno R3+* be disconnected whenever inserting or removing the ATmega328P IC.

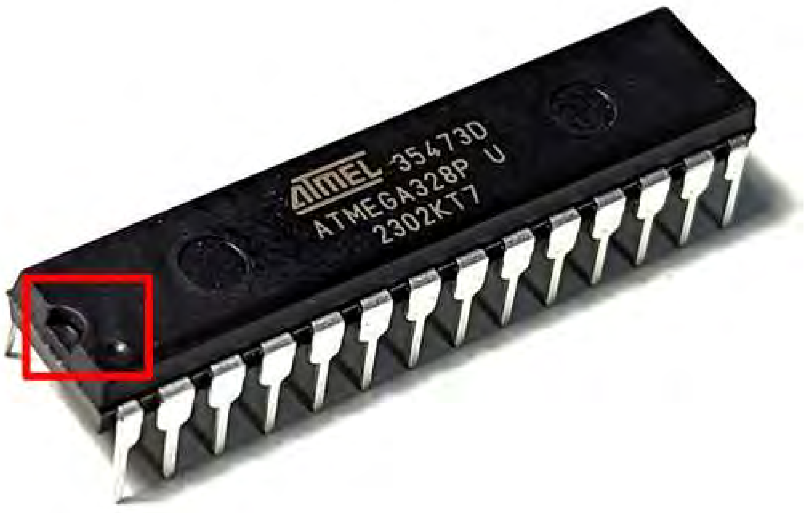


Figure 11: ATmega328P Pin 1 marking (dot indent) and half-circle marking (half-circle indent).

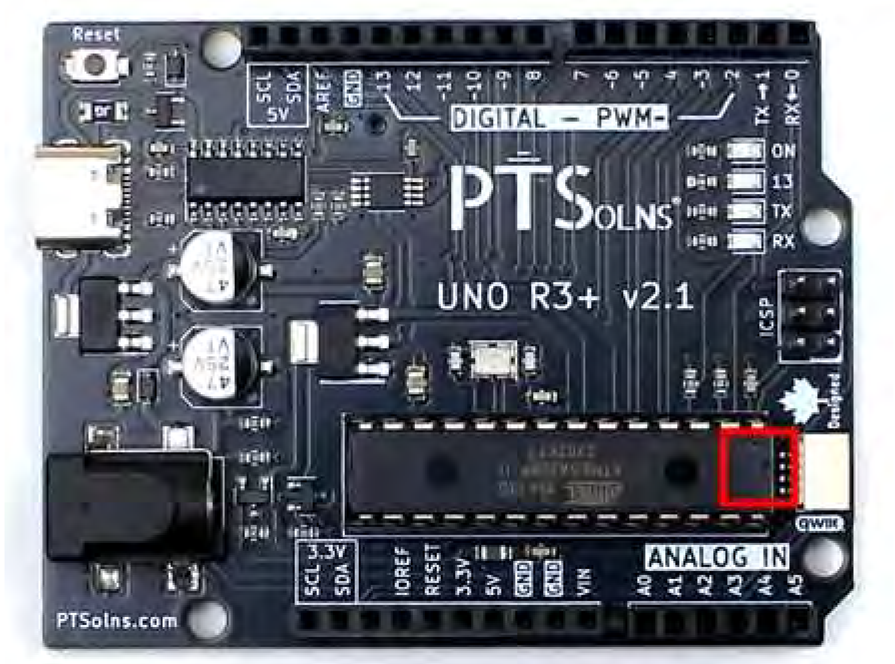


Figure 12: Pin 1 marking (white silkscreen dot) and orientation marking (half-circle indent on IC socket) on the Uno R3+.

6.3 Out-of-the-Box Ready Examples

The *Uno R3+* comes pre-installed with a sketch that allows the user to perform several tests without any further programming. These pre-programmed tests can be used to check the working order of the *Uno R3+*, or to get started quickly making some simple examples. These out-of-the-box ready tests include:

1) Onboard (and Pin 13) LED blinking in unique pattern.
2) Reset makes onboard LED (and Pin 13) blink in fast pattern of four for one cycle.
3) I2C scanner searches for connected devices (5 V or 3.3 V I2C bus, or QWIIC connector) every 5 seconds and prints the results to Serial monitor on baud rate 9600.
4) Pin 9 is on a fading in and out cycle that can drive an external LED accordingly.
5) Analog read on Pin A0 and displayed to Serial monitor on baud rate 9600.

Each of these tests is explained in further detail below. If the user wants to restore the pre-installed testing sketch, it can be downloaded from the PTSolns documentation repository sub-domain:

https://docs.PTSolns.com/Products/PTS-00194_Uno_R3_Plus/Sketches/Uno_R3plus_GetStarted.ino

6.3.1 Test 1: Onboard LED (Pin 13) Blink Unique Pattern

With the *Uno R3+* powered, the onboard LED (marked “13”) blinks in a regular unique pattern. The LED will illuminate for 100 mS and turn OFF for 200 mS. If the reset is pressed (see Test 2) the pattern changes momentarily before returning to the same pattern.

The onboard LED (marked “13”) is also connected to Pin 13, available on one of the pins of the female header (also marked “13”). As a further test, the positive terminal of a standard LED can be put into the female header Pin 13, with the negative terminal of the LED attached to a resistor (in the range of ~200 Ω to 1000 Ω, give or take). The other free end of the resistor can be plugged into one of the ground (marked “GND”) pin in a female header. The LED and resistor can either be soldered together, or a breadboard can be used to make the electrical connection.

6.3.2 Test 2: Onboard LED (Pin 13) Blink Reset Pattern

The reset button triggers a momentarily different pattern consisting of four rapid blinks of the onboard LED (marked “13”) of 50 mS ON and 50 mS OFF. This tests that the *Uno R3+* is restarting properly.

Upon reset the *Uno R3+* output several messages to the Serial monitor on baud rate 9600. To see these messages, load the Arduino IDE software and plug in the *Uno R3+* (ensure that the CH230 driver is installed – see Section 6.1). Turn on the Serial monitor (select baud rate 9600) and read the output window.

6.3.3 Test 3: I2C Scanner

Every five seconds the I2C bus is scanned for any connected peripherals. This includes the 5 V and 3.3 V I2C bus available on the corresponding female header pins, an the QWIIC connector (See Section 3.2 on I2C bus). The results of the scan are printed to the Serial monitor on baud rate 9600. To see these results, load the Arduino IDE software and plug in the *Uno R3+* (ensure that the CH230 driver is installed – see Section 6.1). Turn on the Serial monitor (select baud rate 9600) and read the output window.

The user can connect several I2C peripherals at once, on any or all of the three connection points (5 V I2C pins, 3.3 V I2C pins, QWIIC connector). All the device addresses will be displayed in the Serial monitor.

6.3.4 Test 4: Pin 9 Fade

Pin 9 available on the female header (marked "~9") is a PWM capable pin (that's what the tilde "`" indicates). PWM allows a pin to be driven at different duty cycles. This, among many other examples, can be used to dim, or fade, an external LED. In a similar fashion as outlined in Test 1, connect an LED plus resistor to Pin 9 and ground (marked "GND") and observe the LED fading pattern. Ensure that the LED positive terminal is in Pin 9 and that the negative terminal goes toward GND through the resistor.

6.3.5 Test 5: Analog A0 Read

The analog pin A0 is programmed to be continuously reading any input connected to it. The read input value is displayed in the Serial monitor on baud rate 9600. To see these results, load the Arduino IDE software and plug in the *Uno R3+* (ensure that the CH230 driver is installed – see Section 6.1). Turn on the Serial monitor (select baud rate 9600) and read the output window.

The user can plug a wire directly into the female header pin A0 and the other end into:

- A0 to Ground (marked "GND")
- A0 to 3.3 V
- A0 to 5 V
- A0 free floating

The output as displayed in the Serial monitor will read different values accordingly. A properly working *Uno R3+* should produce the following results:

- A0 to Ground (marked "GND") -> Output around 0
- A0 to 3.3 V -> Output around 660, plus or minor a few
- A0 to 5 V -> Output around 1023, plus or minor a few
- A0 free floating -> Output ranges widely

6.4 Arduino IDE to Program the *Uno R3+*

To program the *Uno R3+* the user is encouraged to install the Arduino IDE software (free), which can be found here:

https://www.arduino.cc/en/software

There are several ways a microcontroller can be programmed, but using Arduino IDE is one of the easiest ways to get started quickly, with a large active online community and many tutorials and forums. The following settings are shown using Arduino IDE version 2.2.1. Future versions of the software may vary slightly.

To upload a new sketch onto the *Uno R3+* using Arduino IDE, the user must select the following settings:

- **Board selection**: In the “Select other board and port…” option (Figure 13), in the “BOARDS” search, type and then select “Arduino Uno”, as shown in Figure 14. **NOTE**: this selection can also be done under the “Tools” and “Board:” option.
- **Port selection**: Under “Tools” and “Port:” select the port that is being used with the USB-C cable that was inserted to the computer, as shown in Figure 15. To see the port being used open the Device Manager under section “Ports (COM & LPT)” or similar. The correct port should be labelled with “CH340”. As an example, see Figure 16. There are several reasons why this port or even the entire Ports section may not be showing. See Section 6.1 for details on CH340 driver installation, or Section 0 for troubleshooting help.
- **Programmer selection**: Under “Tools” and “Programmer:” select “Arduino as ISP”, as shown in Figure 17.

For common troubleshooting relating to programming the microcontroller, see Section 0.

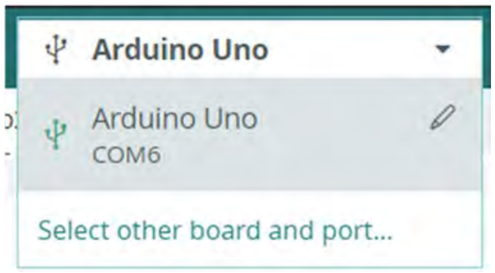
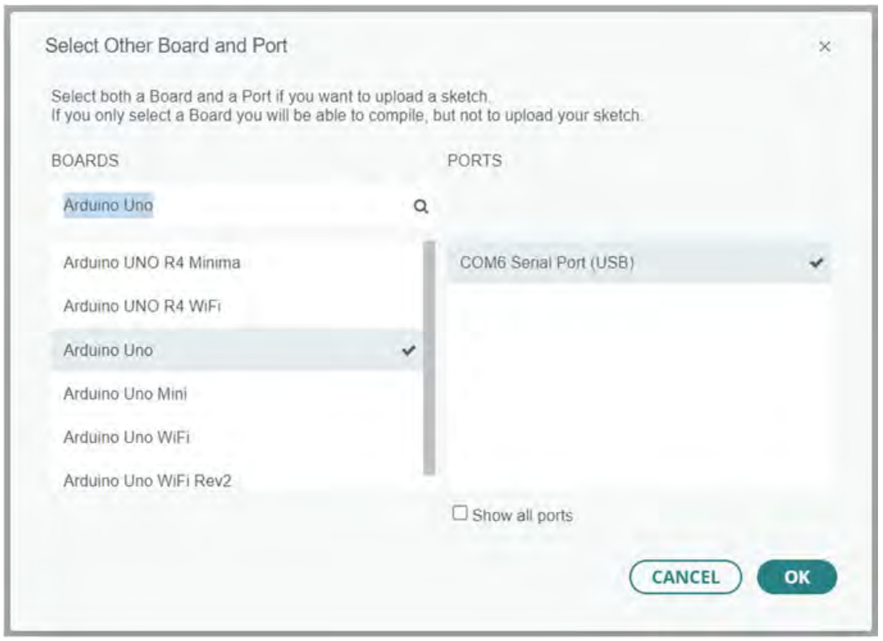


Figure 13: Select other board and port...


Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

Figure 14: "Arduino Uno" board selection for the Uno R3+.

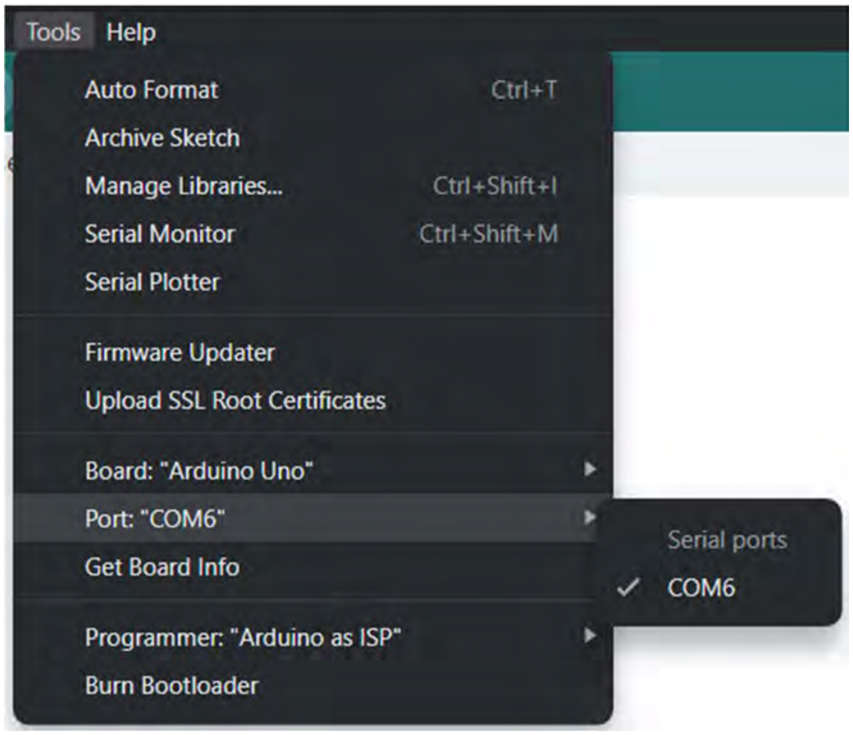


Figure 15: Port selection for the Uno R3+.

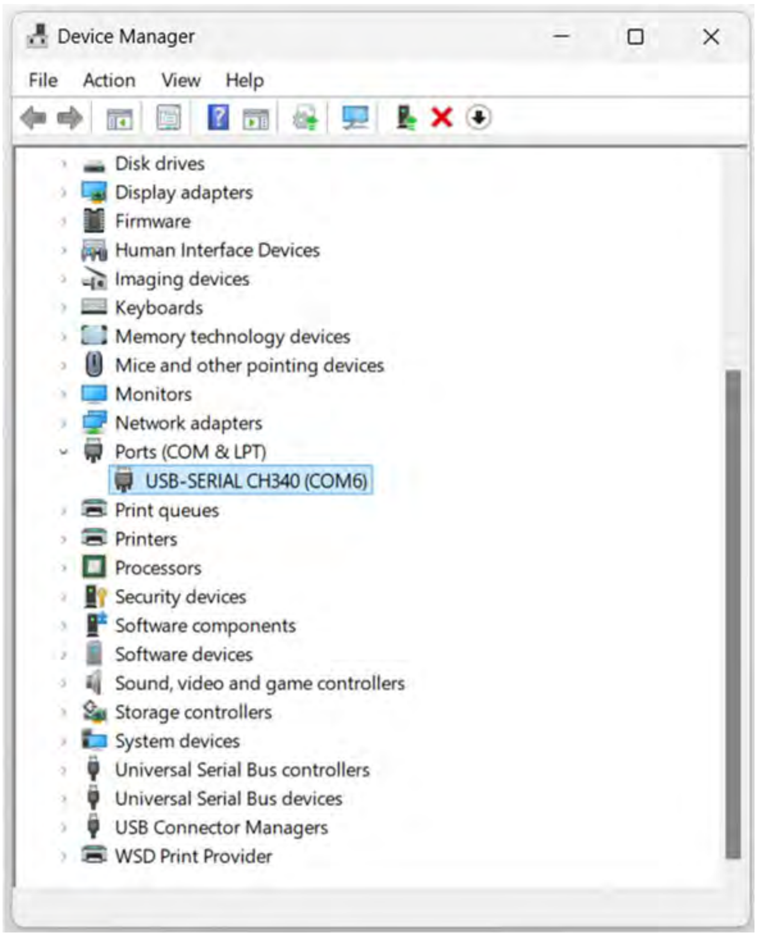


Figure 16: Device Manager, Ports.

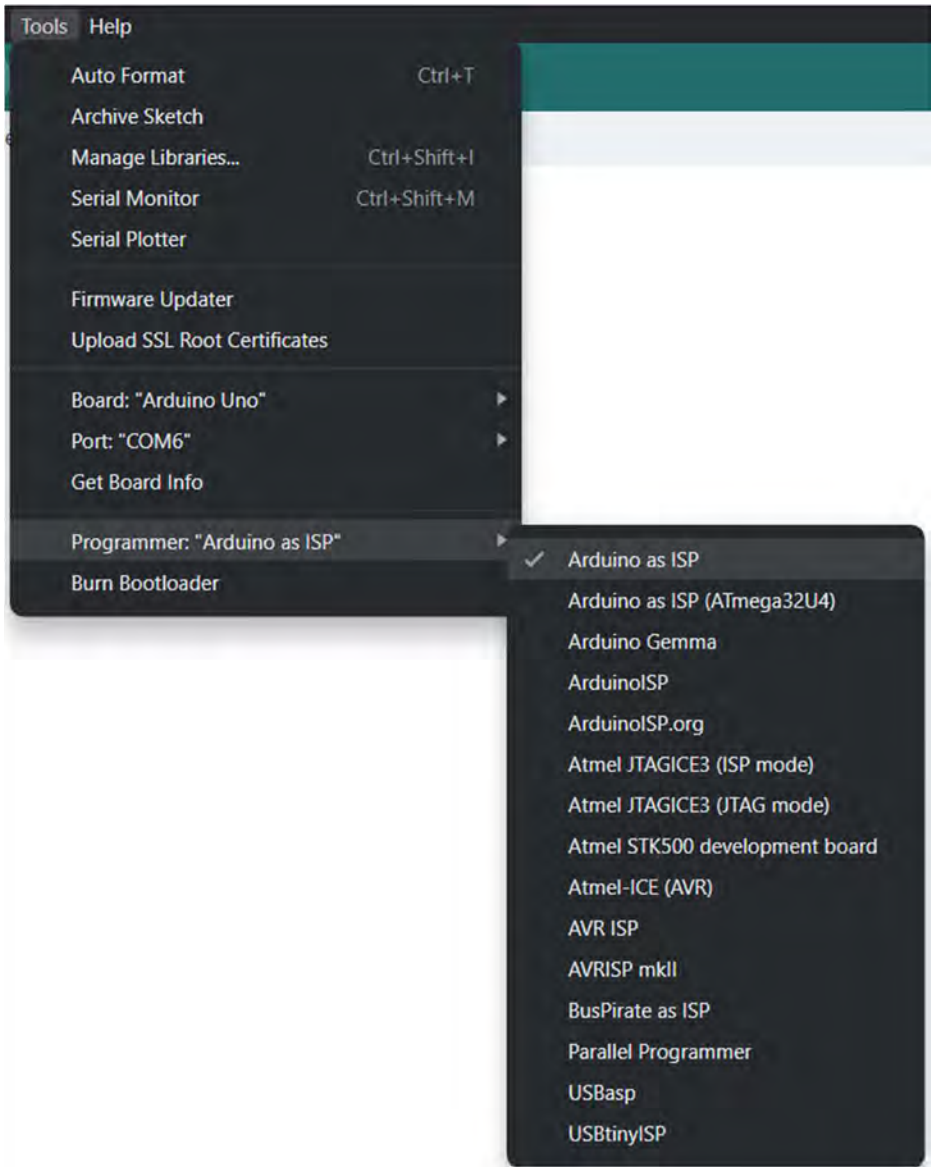


Figure 17: Programmer selection for the Uno R3+.

6.4.1 Blink Sketch Example

The following sketch can be uploaded to the *Uno R3+* to make the onboard LED (on Pin 13) blink in a repeating pattern and print a message to the Serial Monitor. Simply copy/past the below sketch into Arduino IDE, set-up the board settings (see Section 0) and press “Upload”.

```
void setup() {
      Serial.begin(9600);
      pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
      digitalWrite(LED_BUILTIN, HIGH);
      Serial.println("LED ON");
      delay(1000);
      digitalWrite(LED_BUILTIN, LOW);
      delay(1000);
      Serial.println("LED OFF");
}
```

Breakdown of the code:

The `void setup()` function is executed only once at the startup (either powering ON the board, or restarting the board). Inside this function are several declarations and initializations of other routines needed. In the above example the Serial Monitor is initiated on baud 9600 (ensure to select baud 9600). Furthermore, the pinMode command is used to declare that Pin 13 (LED_BUILTIN) is set as an OUTPUT. This allows the LED to be powered ON and OFF.

The `void loop()` function runs repeatedly over and over ad infinitum and is the crux of what makes microcontrollers to versatile. In the above example the LED Pin 13 (LED_BUILTIN) is toggled HIGH and LOW with each a delay of 1000 microsecond (or 1 second). Furthermore, the simple print statement is used to print to the Serial Monitor the status of the LED.

The above example is very simple but can be used as a starting point to expand the user’s knowledge. There are many ready examples that come with the Arduino IDE download. It is recommended to use the Serial Monitor as a tool to help debug. The Serial Monitor can be used to print values such as variables or constants, but also simple “here” messages when a certain part of the sketch was reach/executed.

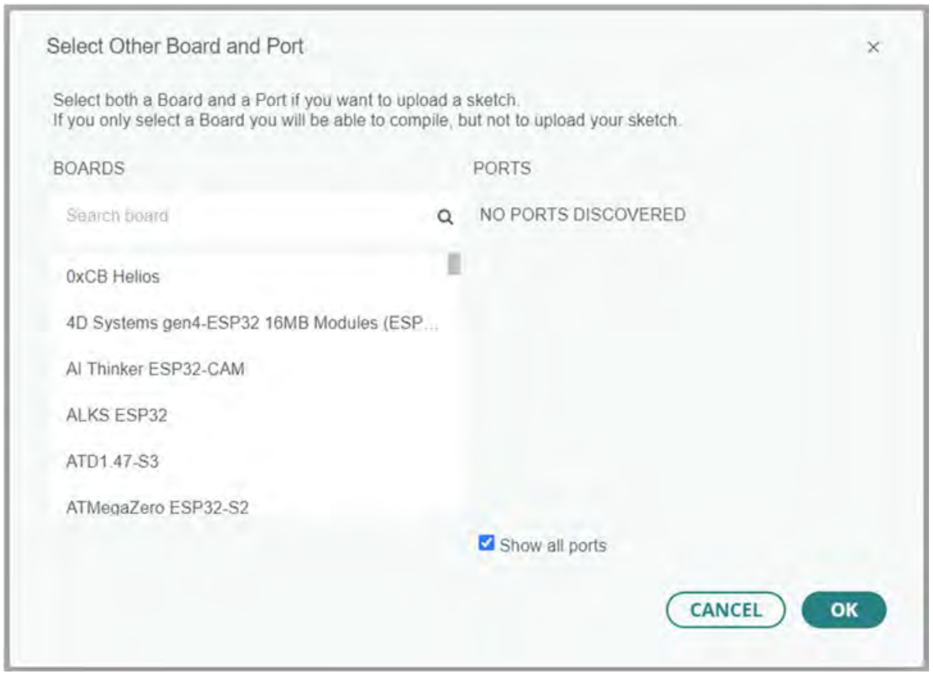
6.5 Troubleshooting

The following items is a list of troubleshooting items that commonly arise. The user is encouraged to go through these common items in order to diagnose the issue and quickly find a solution. The list below was made using Arduino IDE version 2.2.1. Future versions of the software may produce different error messages or symptoms.

Description: Incorrect USB-C cable type (not data transfer capable)

Symptom: No port showing / Not the correct port showing

Solution: One possible cause of not seeing a port (Figure 18) or seeing some ports but not the correct one is that the USB-C cable being used is not capable to transfer data. Not every USB-C cable is capable to transfer data from the computer to the microcontroller. For details on data transfer capabilities as they relate to USB-C cables, see Section 0. The solution in this instance is to replace the USB-C cable with a different one that is capable of data transfer.


Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

BOARDS

PORTS

NO PORTS DISCOVERED

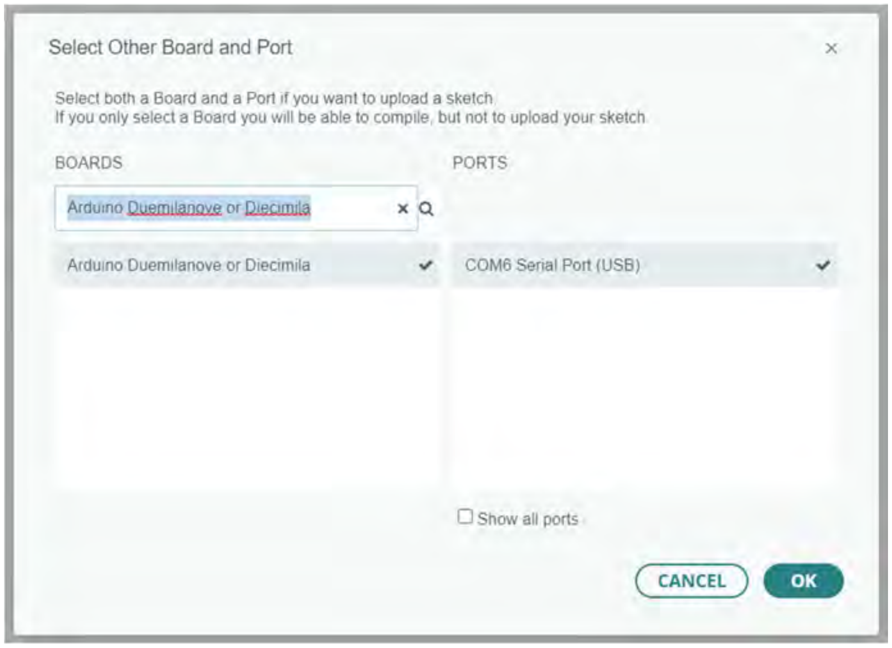
0xCB Helios
4D Systems gen4-ESP32 16MB Modules (ESP…
AI Thinker ESP32-CAM
ALKS ESP32
ATD1.47-S3
ATMegaZero ESP32-S2

Figure 18: No port showing / Not the correct port showing.

Description: Incorrect Board selection.

Symptom: “avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x00”

Solution: One possible cause of this error message is incorrect board selection. As an example, in Figure 19 the “Arduino Duemilanove or Diecimila” board was selected, which will not work with the *Uno R3+*. The solution to this is to select the correct board, which is “Arduino Uno” as shown in Figure 14.



Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch
If you only select a Board you will be able to compile, but not to upload your sketch

BOARDS: Arduino Duemilanove or Diecimila ✓

PORTS: COM6 Serial Port (USB) ✓

Figure 19: Incorrect board selection.

Description: Incorrect Port selection.

Symptom: “avrdude: stk500_recv(): programmer is not responding”

“avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x8c”

Solution: One possible cause of the error messages is that the wrong port was selected. If multiple USB sources are plugged at once it is possible to select the wrong one. To find out which port is the correct one, open the Device Manager to the “Ports (COM & LPT) or similar. If there are no other issues, there should be a port labelled with “CH340”. If this is not showing, see Section 6.1.

Description: CH340 driver not installed.

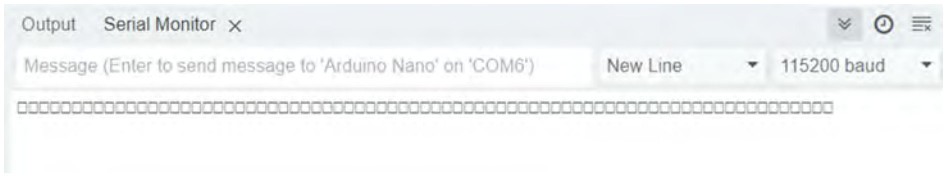
Symptom: "avrdude: ser_open(): can't open device "\\.\COM6": The system cannot find the file specified."

Solution: One possible cause of this error is that the CH340 driver is not installed. The solution is to install the driver, as outlined in Section 6.1.

Description: Incorrect baud selected.

Symptom: Nothing/Junk is printing in the Serial Monitor

Solution: A mismatch in baud rate settings. In the sketch where the Serial Monitor is initiated ("Serial.begin(baud)") the specified baud has to match the baud setting in the Serial Monitor. When the baud rate is not set properly the results can look similar as in Figure 20. The solution is to match the baud rate. Look at the void setup() and find the "Serial.begin(XXX)" command. Note what the baud rate is (e.g. XXX = 9600). Open the Serial Monitor and ensure that the same baud rate is selected.



115200 baud

```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
```

Figure 20: Incorrect baud selected.

Description: Port busy / used elsewhere.

Symptom: "avrdude: ser_open(): can't open device "\\.\COM6": Access is denied."

Solution: One possible cause is that multiple instances of Arduino IDE with multiple Serial Monitors are opened, and the same port (in this example Port COM 6) is used in two or more of them. The solution to this is to close all the other Serial Monitors that are not in use. In severe cases Arduino IDE can be restarted entirely to solve the issue.

If all else fails, contact our support team:

https://ptsolns.com/pages/contact

7 REFERENCES

This section lists relevant references.

- ATmega328P datasheet by Microchip Technology:
  https://www.microchip.com/en-us/product/atmega328p

- Arduino IDE software (See Section 6.1):
  https://www.arduino.cc/en/software

- PTSolns' Documentation Repository Sub-Domain:
  https://docs.PTSolns.com

- PTSolns website:
  https://ptsolns.com/

- *Uno R3+* default installed testing sketch (See Section 6.3):
  https://docs.PTSolns.com/Products/PTS-00194_Uno_R3_Plus/Sketches/Uno_R3plus_GetStarted.ino

- QWIIC connector standardization by SparkFun Electronics (See Sections 3.1 and 3.2):
  https://www.sparkfun.com/qwiic

- CH340 driver installation tutorial (See Section 6.1):

  By PTSolns:
  https://www.youtube.com/watch?v=UUQ84VKg3oM

  By Sparkfun:
  https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all

- PTSolns support:
  https://ptsolns.com/pages/contact

The following lists PTSolns shields recommended for use in conjunction with the *Uno R3+.*

- Proto-Shield datasheet:
  https://docs.ptsolns.com/datasheets/Datasheet_PTS-00153_Proto-Shield.pdf

- Interface-Shield datasheet:
  https://docs.ptsolns.com/datasheets/Datasheet_PTS-00156_Interface-Shield_Kit.pdf

- NRF-Shield datasheet:
  https://docs.ptsolns.com/datasheets/Datasheet_PTS-00154_NRF-Shield.pdf