




## General IO Monitor for Arduino UNO

Before any of the free Apps for Arduino can be run for the first time, you will need to have Arduino drivers installed via the [Arduino ide software page](#).

You will also need to load the Arduino Uno App Developer Firmware from [this Wiki page](#) to your Arduino Uno.

You can manually find the firmware for different target devices by going to the [Wiki](#) and searching for scada slaves.

The zip file will contain three files:

-  `Arduino_Uno_App_Developer_Comp.fcfx` <Component source code
-  `Arduino_Uno_App_Developer_Firmware.fcfx` <Firmware source code
-  `Arduino_Uno_App_Developer_Firmware.hex` <Compiled firmware

Once loaded on to Arduino, any of the Arduino apps can be run without reloading the firmware.

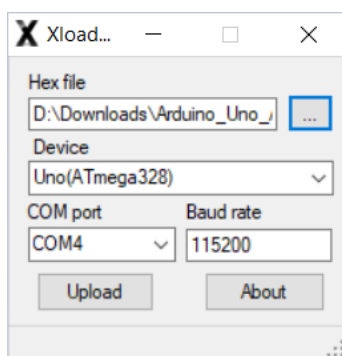
Download the firmware from the first link.

Extract the zip file into a folder of your choice.

Of course, you can use Flowcode to load the firmware then send it to the Arduino.

If you do not have Flowcode, then download XLoader from [here](#).

XLoader enables you to upload hex files very easily:




Use the three dots on XLoader to browse to where you extracted the `Arduino_Uno_App_Developer_Firmware.hex` zip file.

Select **Arduino\_Uno\_App\_Developer\_Firmware.hex**

Make sure Device is showing **Uno(ATmega328)**

Run Device **Manager** by right-clicking on the windows key and left-click on Device manager.

Expand >  Ports (COM & LPT)

Look for either Arduino Uno or USB Serial Device (depends on which drivers were used) and remember the Com port number.

Select correct com port in XLoader, then click on Upload.

You should see the RX and TX LEDs on the Arduino flash while data is being transferred.

If it is not, then make sure no other software has taken the port over.

After the transfer is complete, you are now able to run any of the App developer's software.

Run the App by double-clicking on General IO Monitor Arduino UNO bat file.

You can also right-click on the bat file and select **Send to**, then Desktop (**create a short-cut**)

Right-click in the panel and select **Properties** & Select the correct com port for Arduino.

If the App Developer is running successfully, then both RX and TX LEDs on the Arduino will constantly flash.

If they do not, then see Troubleshooting.

The mouse scroll wheel can zoom in & out.

When happy with the size, right-click, select **Look at the origin**, Which will centralise the App

The General IO monitor has a mixture of five inputs, five outputs, five analogue inputs and one PWM output.

The number within brackets represents either the Digital (D4 to D13) or Analogue (A0 to A5)

The Digital input is just monitoring pins (D4 to D8) for a voltage source.

If nothing is connected to the digital input pins, then the level could be either 1 or 0, depending on any external fields present.

This is because the inputs are floating due to their high impedance.

For the input LED to light, the input must be equal or greater than  $0.6V_{cc}$ , e.g., 3V or greater if  $V_{cc}$  is 5V & no greater than  $V_{CC} + 0.5V$

For the LED to be off, the voltage must be less than  $0.3V_{cc}$ , e.g., less than 1.5V if  $V_{cc}$  is 5V

The analogue also has a high impedance, so it will float if no voltage source is connected, this will cause random values to be displayed.

The analogue input must have no more than  $V_{CC}+0.5$  on the input.

The only way you can have greater than  $0.5+V_{cc}$  is to use a suitable potential divider, which is beyond the scope of these instructions.

Finally, a PWM on D3 that runs at 31KHz, with a duty of 0 to 100%.

A PWM is a digital signal that can have a high portion changed in width compared to the low portion, thereby changing the average equivalent level.

E.g 0% == 0V, 50% == 2.5V & 100% == 5V

## Troubleshooting.

When App is logging, and you get unexpected results.

Unexpected results would mainly be down to a communication issue.

Other causes could be hardware issues, e.g. faulty potentiometer or Arduino or a software bug.

There are two ways of determining if it is a communication issue.

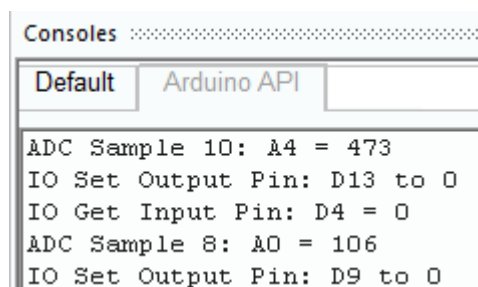
If both TX and RX LEDs are not flashing while App Developer is logging.

Or open the console (View menu, select Consoles).

Undock the console so the App Developer panel can be made larger.

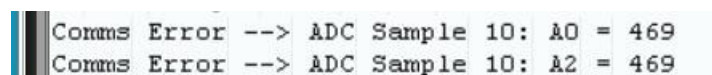
Select **Arduino API** tab.

If communication OK, you will see something like:



```
Consoles .....
Default  Arduino API
-----
ADC Sample 10: A4 = 473
IO Set Output Pin: D13 to 0
IO Get Input Pin: D4 = 0
ADC Sample 8: A0 = 106
IO Set Output Pin: D9 to 0
```

If there is an issue with communication or firmware, then you will see:



```
Comms Error --> ADC Sample 10: A0 = 469
Comms Error --> ADC Sample 10: A2 = 469
```

If it is a communication issue:

Is the correct com port selected within properties?

Has some other software taken over the port?

Is only one instance of App developer run at a time?

With the hardware issue, so long as communication is confirmed working, measure the pin's voltage that is giving unexpected results.

It should be steady and not floating.

If the voltage is correct, then make sure the Arduino has the Arduino App developer firmware programmed.

For further advice, there will be help available via the [forums](#).