.AIM / DESCRIPTION

The aim of this project was to

-create a simple way of visualizing the 2020 presidential election:
The red and blue LEDs stand for votes coming in (Republicans/ Democrats respectively, each blinking when changes happened since last update), the RGB LED stands and lights in the color of the presently leading candidate (measured in electoral votes)

-learn the basics of Python and it's *serial* library
-learn the basics and create a webscraper using BeautifulSoup
-further my knowledge regarding Arduinos (Elegoo)
-have fun

The hardware and software build were subsequently finished within 2 days!

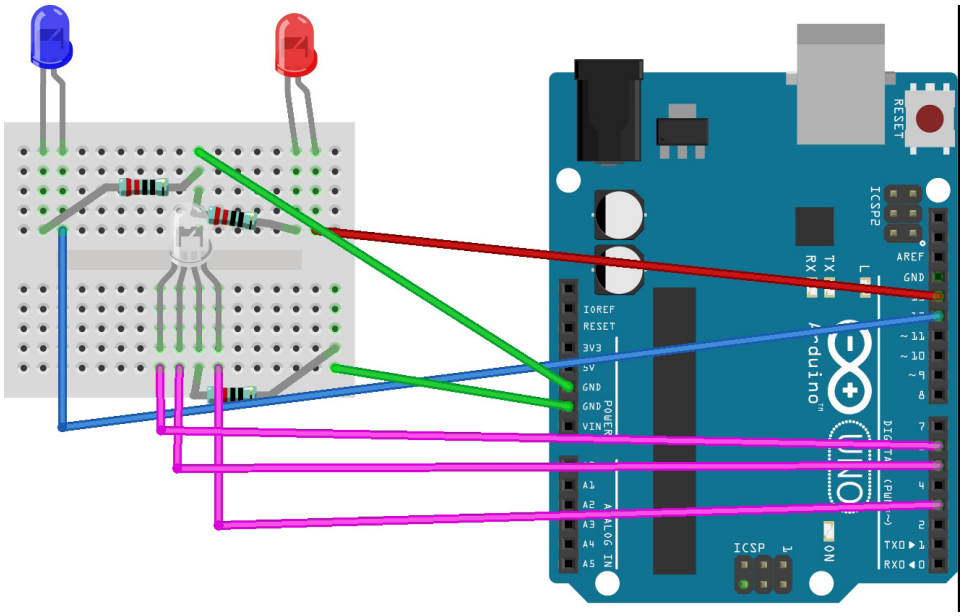
.BILL OF MATERIALS & PARTS

1x Elegoo UNO & Power Supply
1x Breadboard
6x Breadboard Jumper Wire
1x RGB LED
1x Red LED
1x Blue LED
3x Resistor 2k

.NECESSARY TOOLS

-Arduino compatible IDE

.CIRCUIT DIAGRAM



.SHORT DESCRIPTION OF SOFTWARE ROUTINE

.Python:

I start out initializing the connection using the *serial* library and its *serial.Serial*-function (**Important**: add a timer of a few seconds to wait for an established connection!)
The scraping of the relevant values follows by reading in and parsing (Thanks *BeautifulSoup*!) the html file of an article by the news site *The Guardian*, specifically this article archived here:

http://web-old.archive.org/web/20201107084804/https://www.theguardian.com/us-news/ng-interactive/2020/nov/05/us-election-2020-live-results-donald-trump-joe-biden-presidential-votes-arizona-nevada-pennsylvania-georgia

After scraping and parsing, the incoming values get compared to the ones from the last update; with change detected comes the execution of the *arduino.write-*command, sending the event-specific chars to the USB-connected Arduino. (**Note:** wanting to send more than a single char comes with a lot of further complication!)

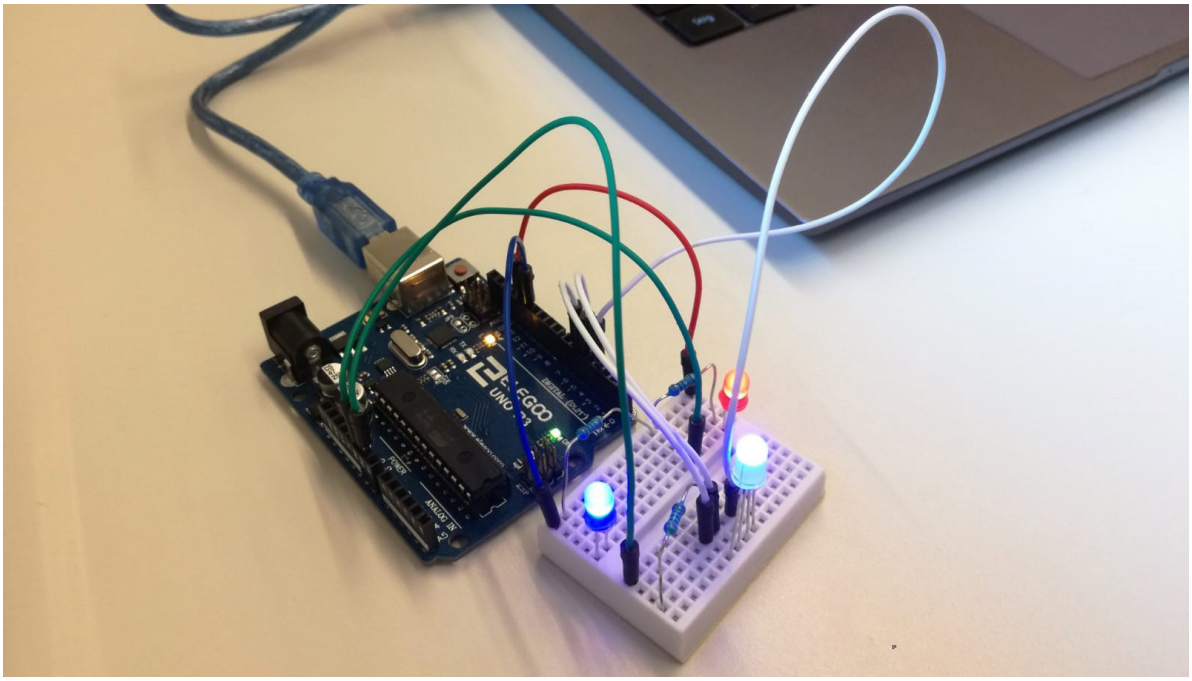
The whole update-routine will repeat itself after a 30 to 40 second pause.

.Arduino:

To communicate between Python and the arduino, the *serial* library and it's subsequent *write*-function is used.

In the *setup* routine, first the Serial input is flushed and the LED pins are assigned.
In *loop*, I check for the incoming chars sent by my IDE used for the Python program and trigger the applicable LED routine.

.IN WORKING CONDITION:



06.11.2020