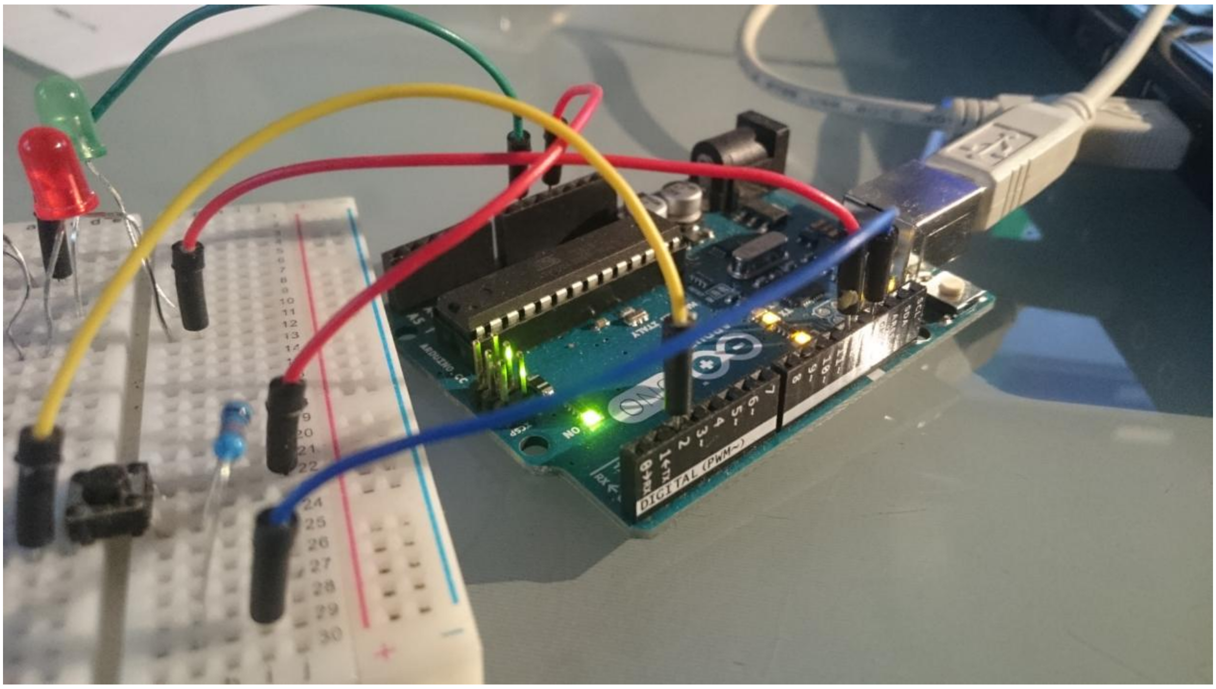
ARDUINO - RUBY via Serial Communication



Introduction

Having worked with Arduino on many projects and having more recently learned to develop for the web, the thing I immediately wanted to do was to interface both the hardware side of things with the digital one.

So I fiddled with Arduino, Serial communication and Ruby to do just that.

The code can be viewed here.

How it was built

I used Ruby for the computer side of things because I'm most fluent in it, but this could have been accomplished with another language.

I found two approaches to interface Ruby with Arduino:

1. Use the Dino gem, which essentially uses pre-made Ruby code from the Dino team to operate. This is the easiest option, but that means that to change Arduino's actions requires to change the Ruby code. Since I already know how to code these actions in Arduino, I would rather code the Arduino's actions in Arduino, and leave only the terminal/browser part to be handled by Ruby.

2. Use the serialport gem, which simply provides a class for using RS-232 serial ports. That way, I can write the code to control the Arduino in Arduino language, and write the code which controls the terminal / browser in Ruby. I pass instructions from one to the other by sending a code from one side, and listening for it on the other side.

I followed this tutorial for the Ruby part, replacing

```
message = sp.gets.chomp

 puts message
```

by:

```
message = sp.gets
p message unless message.nil?
```

which was returning an error.

Credits

- serialport gem to use RS-232 serial ports
- Viget tutorial for the serial communication with Ruby