

README.md



CSL-light



This repository demonstrates how to control light sources with Arduino and Python, and output a trigger signal to synchronize a camera.

The codes rely on [Arduino](#) and [pyserial](#).

Pre-requisites

- Install RomiSerial and the Arduino software XXXX
- The light sources are already set-up. Refer to the example gallery for ideas.
- The light sources can be controlled by a trigger, or pulse-width modulated signal (PWM)
- The code was tested on Windows and Linux

Hardware

Here are the different hardware equipment the

Component	Quantity	Price per unit	Example
Arduino Uno	1	24€	Robotshop
Light source controller	tested up to 5	X	Thorlabs

Software

Software	Version we used	Download
Arduino	1.8.13	download
Python	3	install
CSL-serial	1.0	install

Codes and files provided

An [Arduino code](#) is provided to control the motors. An extra layer is added in Python to control it easily.

[CSLight](#) can be used the following way:

```
from serial import Serial
from CSLight import ControlLight

arduino_port = "COM5"
sec = 1000 #conversion ms to s
blue_param = {'pin': 11,
              'offset': 0.5*sec, #ms
              'period': 5*sec, #ms
              'duration': 2*sec, #ms
              'secondary': 1,
              'analog_value': 255,
              }

arduino_light = ControlLight(arduino_port)
arduino_light.add_digital_pulse(blue_param)

arduino_light.start_measurement()
time.sleep(300)
arduino_light.stop_measurement()
```

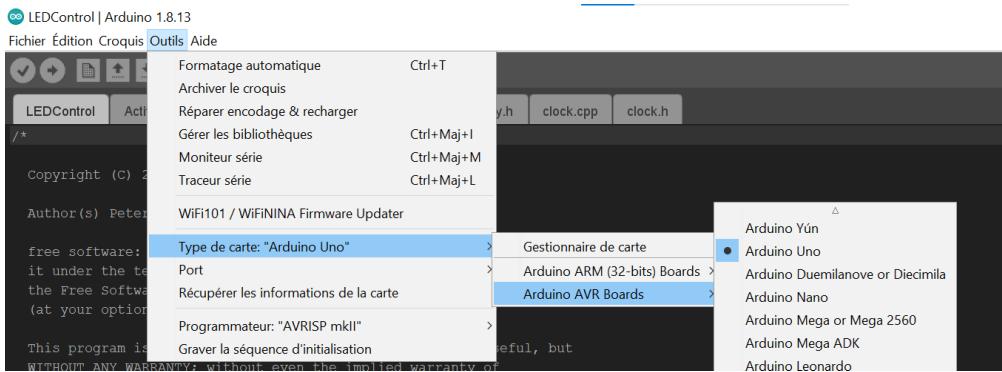
Instructions:

Download or clone the repository:

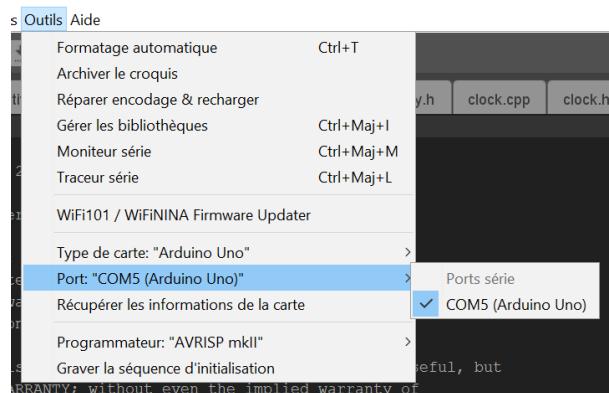
```
git clone XXXXXXXX
```

Control the LEDs

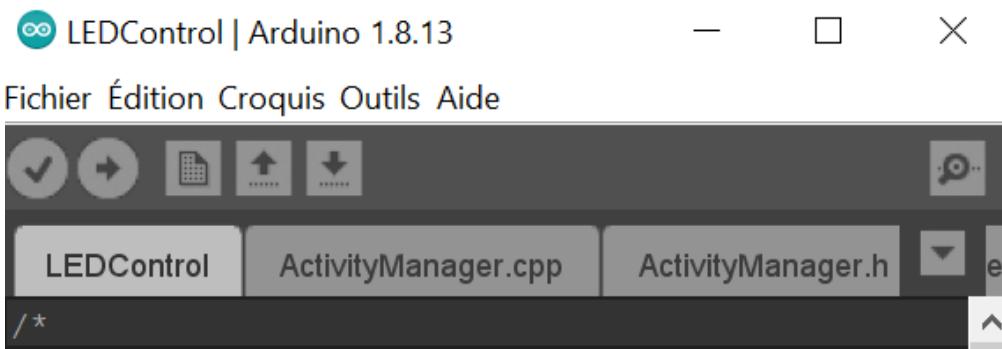
1. Get the wiring to connect the Arduino to the light source controller. To begin, connect the wire to **pin 11**.
2. Open the [LEDControl/LEDControl.ino](#) file.
3. Select the Arduino board type in the "Tools/card type"



1. Select the COM port. If the name of the board doesn't appear near any port, change the port USB until the name appears.

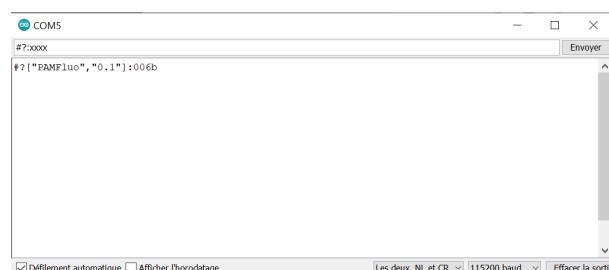


5. Press the check sign. If an error related to "RomiSerial" appears, verify that you have properly followed the instructions in the CSL-Serial repository.



6. If no error appears you can click the arrow to load the code in the Arduino.

7. To test that you can properly interact with the Arduino, click on the magnifying glass in the upper right to open the serial monitor. Select **115200 baud** and **Both NL & CR** and type: "#?:xxxx" and ensure you get this output:



8. type: #d[11,0,0,2,0,1,0,0,255]:xxxx You should see a character sequence appear.

Then type #b:xxxx to start the experiment. You should see the LEDs blink (frequency 0.5Hz). To stop the blinking, type #e:xxxx

Install the library

```
cd CSL-light  
python setup.py develop
```



1. Try running the code:

On Windows: python CSLlight/CSLight.py --port COMx by replacing "COMx" by the correct COM port identified in step 1.

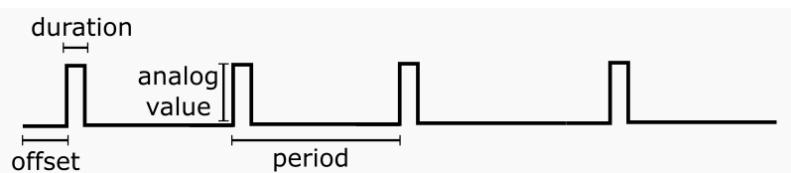
On Linux: python3 CSLlight/CSLight.py --port /dev/ttyACM0

You should see the LED blink.

2. Open the python code to see how it works. Open the python code [CSLight.py](#).

The code is commented and allows to control the frequency and amplitude of the LEDs. Set the parameters: The content of interest is after `if __name__ == "__main__":`

- replace the COM port with the one of your set-up ([tutorial](#)).
- input the correct ports for the LED control. The port 3 and 11 are good choices because they are PWM pins which allow to control the intensity level of the LEDs rather than only ON-OFF.
- you can change the other parameters that correspond to this scheme:



Examples of implementation



[How to make a gallery](#)

(Note: to build an LED controller refer to this [OpenUC2 repository](#), otherwise you might already use one of these [Thorlabs controllers](#)

License

This project is licensed under the [GNU General Public License v3.0](#)

