

# Light Control for Solar Simulator

## Technical Specifications



Author: **Simone Pilon** <[s.pilon@uva.nl](mailto:s.pilon@uva.nl)>

Noël Research Group - 2023 - University of Amsterdam

# Table of Contents

Usage.....	3
Arduino Serial Monitor .....	3
List of Commands.....	4
'Sx=y' Set variable x to value y .....	4
'Rx' Read variable x .....	4
Construction.....	5
Principle of operation .....	5
3D-printed parts.....	5
Electrical connections .....	6
Bill of Materials .....	8

# Usage

Once properly connected, the device can be controlled from the front panel (for wiring diagrams see the 'Construction' section):

1. Turn on the mains switch (bottom right).
2. The screen lights up, use the + and – buttons to select the desired light intensity.
3. Press the 'start' button for 3 seconds to activate the lights and cooling fans. Press the button again for 3 seconds to turn the lights off again. The fans will stay on until the device is reset (e.g.: turn off mains power).



Once the mains power has been activated (step 1), remote/automation control is possible by connecting a computer to the Arduino board within the device. The USB port of the Arduino control board is accessible from the right side of the box without opening the lid. Once connected, commands can be sent via any serial interface.

## Arduino Serial Monitor

To control the device remotely via the Arduino Serial Monitor, start the Arduino IDE (tested on version 2.1.0), then connect the device to the USB port of the computer. Make sure the "Serial Monitor" is open, by clicking on the top-right button to activate it. Then, select the correct USB port from the menu on the left. If asked, select "Arduino UNO" as the board.

Once connected, it is possible to send and receive data from the device.

Note: if you are assembling a new device or replacing the Arduino board, the firmware will need to be uploaded to the board before sending any commands: open the firmware sketch (solar\_simulator/solar\_simulator.ino file) in the Arduino IDE and press the 'upload' button on the top left. If there are no errors, the message 'upload successful' should appear at the bottom of the screen.



To test if the Arduino board is working as expected, type 'R1' in the Serial monitor message bar and press enter. The message 'LIGHT\_CONTROL\_0' should appear right below it.

# List of Commands

This is a list of commands which can be sent to the device via the serial interface:

## 'Sx=y' Set variable x to value y

Sending the ASCII character 'S' tells the microcontroller that a variable value needs to be changed. An integer number must follow the 'S' character, indicating the variable number (see table below). After a delimiter character (any non-digit except '.' Will do), the value for the variable must be given. The type depends on the variable.

Note: in case of a string variable, the command must terminate with a newline.

## 'Rx' Read variable x

Sending the ASCII character 'R' tells the microcontroller that a variable value needs to be printed to serial. An integer number must follow the 'R' character, indicating the variable number (see table below).

Variable number	Access	Type	Description
0	RESERVED	-	Serial.parseInt returns 0 on error To avoid undesired variable access, this value is not used.
1	READ/WRITE	String [20]	Device identifier This can be used to distinguish similar serial devices.
2	READ/WRITE	Int [0-1]	Lights On/_Off state Setting to 1 turns on the lights and the fans. Setting to 0 will turn the lights off only.
3	READ/WRITE	Int [0-1]	Fans On/_Off state Controls the fans, if the lights are on the fans will not be turned off.
4	READ/WRITE	Int [0-1]	Manual interface lock On/_Off state Setting this to 1 prevents users from changing the device settings via the manual interface.
5	READ/WRITE	Int [0-100]	Light intensity Percentage of light intensity for the lights. According to the power supply manufacturer, values below 10% should be avoided. Do not set this to 0 to turn the lights off, use the Lights On/_Off variable instead.

# Construction

## Principle of operation

Four high power LEDs are powered via an HLG-480H-36AB power supply. This is equipped with a DIM+/- input which allows to control light intensity by sending a PWM signal proportional to the desired output power (and hence light intensity).

The DIM signal is controlled by an Arduino microcontroller via a PWM signal, which is shifted from the 5v at the output pin of the Arduino to the ~12v required by the power supply via an operational amplifier on a small custom electronic board.

The Arduino also controls 2 medium power MOSFET transistors on the custom board, which control the fans and main circuit breaker. Since the light intensity cannot be turned off completely via the DIM signal, a circuit breaker interrupts the connection between the power supply and mains, ensuring a safe off condition.

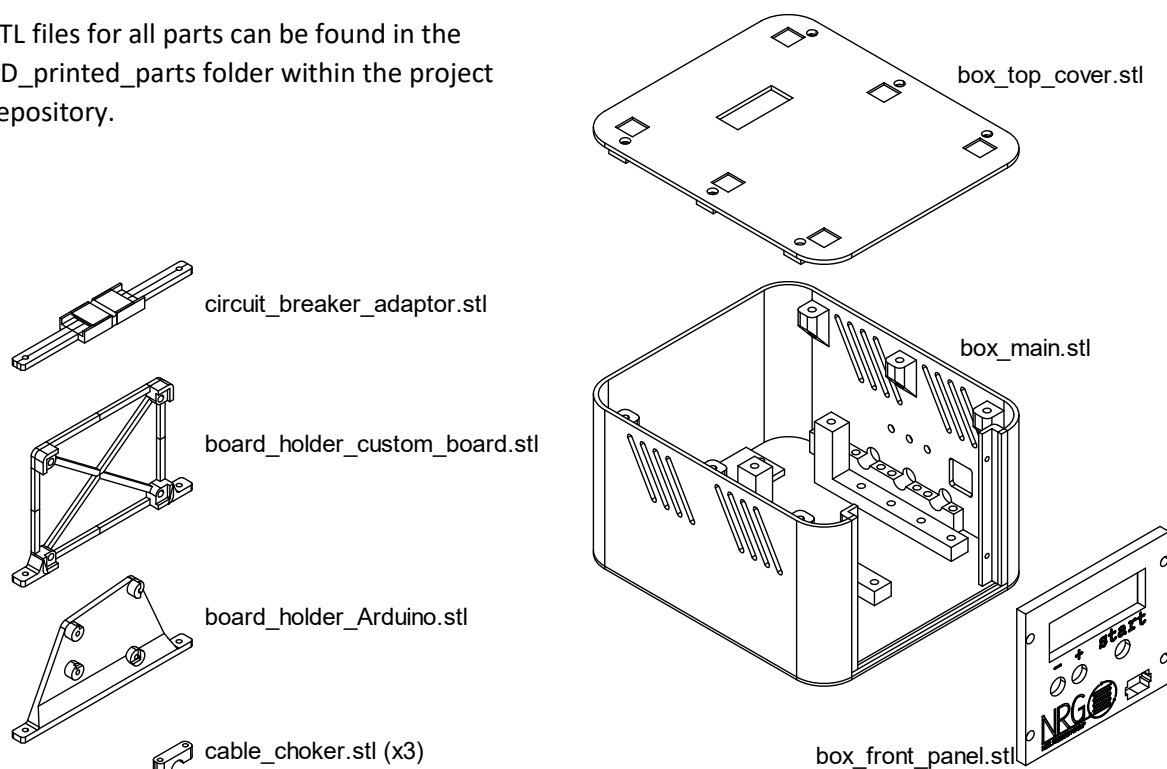
The 12v power supply operates separately from the LED power supply.

## 3D-printed parts

Most components are housed in a 3D-printed box consisting of 3 parts held together by 10 M3x12mm cylindrical head DIN 912 screws. The screws are threaded into M3 brass plastic inserts which are embedded into the 3D-printed unthreaded holes with a soldering iron.

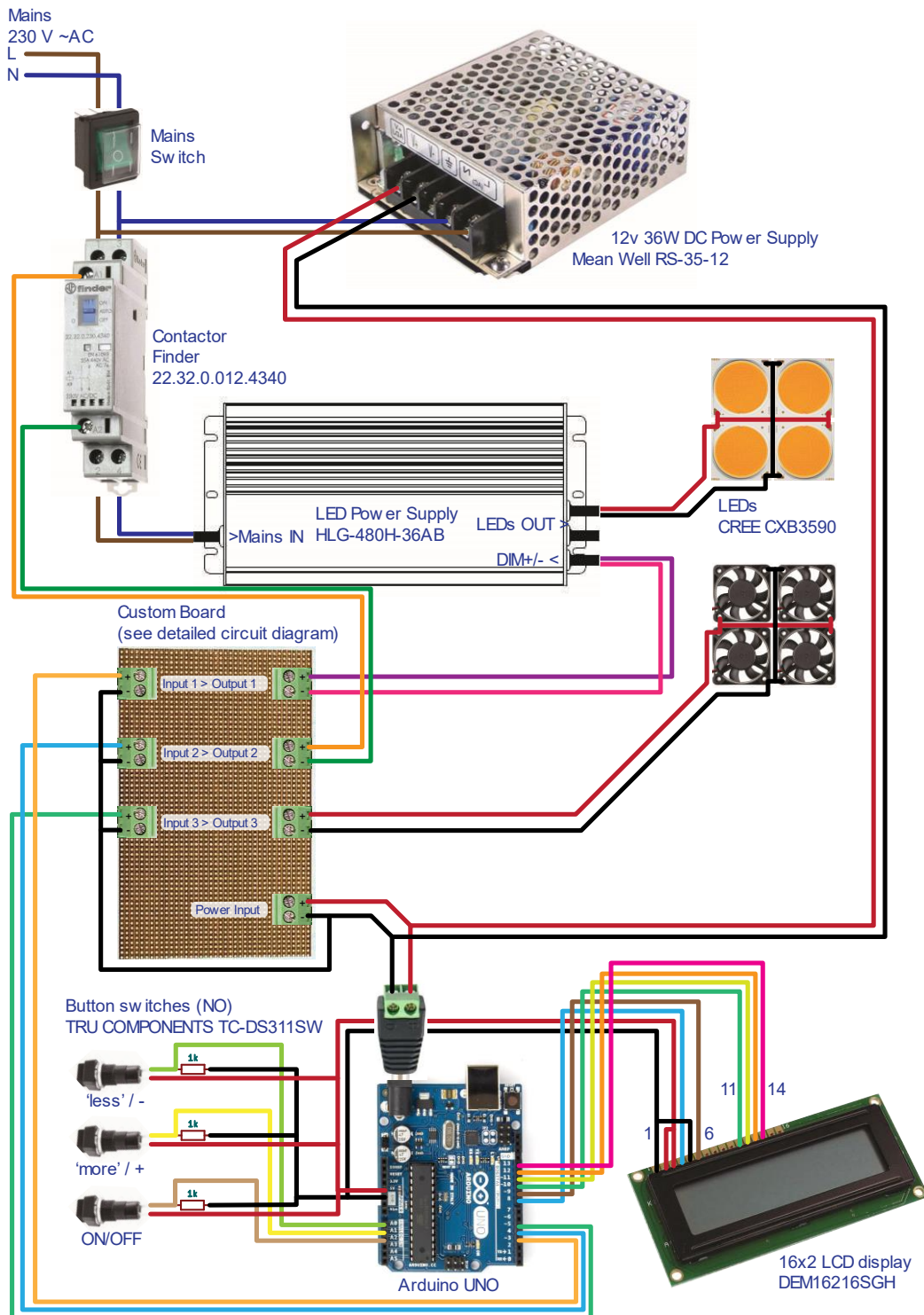
The circuit breaker, Arduino board and a custom board are secured to the main box body via adaptors which fit into the rail system at the bottom of the box. All screws used are M3x12mm cylindrical head DIN 912 and all holes are made to fit M3 inserts.

STL files for all parts can be found in the 3D\_printed\_parts folder within the project repository.



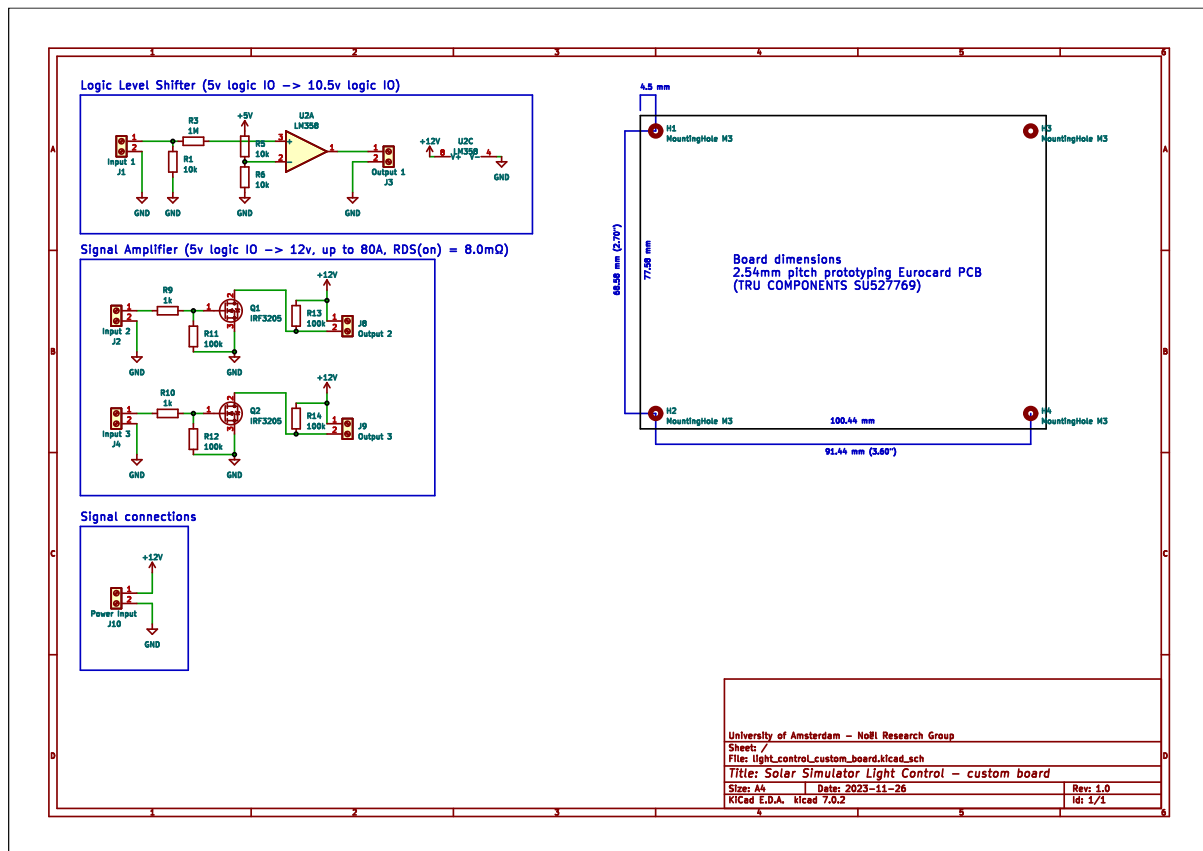
# Electrical connections

This is the diagram for all electrical connections required for the controller to operate (find a full-size image in docs/schematics):



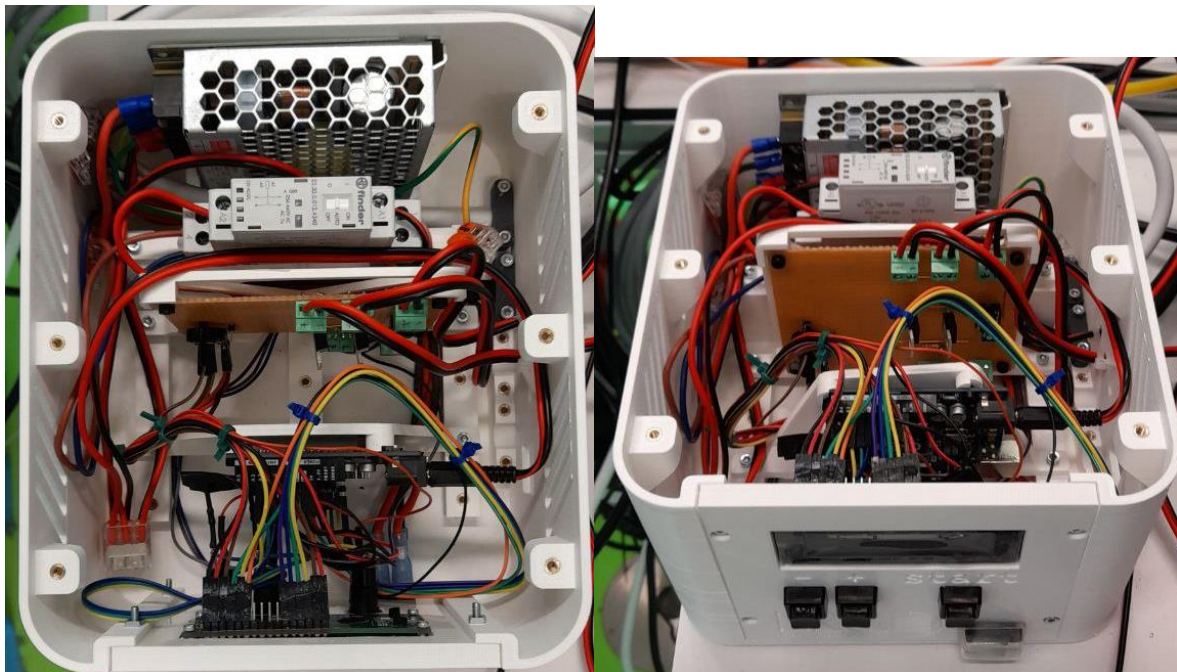


The connections within the custom board are reported below (find a full-size image in docs/schematics, as well as a KiCad project file):



The board was soldered by hand from discrete components.

This is a top-view of the fully assembled device for reference:



# Bill of Materials

- ❑ LED Power Supply HLG-480H-36AB
- ❑ 12v 36W DC power supply Mean Well RS-35-12
- ❑ LEDs CREE CXB3590 (4)
- ❑ Contactor Finder 22.32.0.012.4340
- ❑ Cooling fans 40mm 12v (4)
- ❑ Press button TRU COMPONENTS TC-DS311SW (3)
- ❑ LCD display 16x2 DEM16216SGH
- ❑ Arduino UNO microcontroller
- ❑ Mains switch Omron A8L-21-11N2
- ❑ Custom Board components:
  - Screw terminal block - 2 contacts - TE Connectivity 282837-2 (7)
  - 1k $\Omega$  1/4W resistor (5)
  - 10k $\Omega$  1/4W resistor (3)
  - 100k $\Omega$  1/4W resistor (4)
  - 1M $\Omega$  1/4W resistor (1)
  - Power MOSFET IRF3205 (2)
  - Operational amplifier LM358 with socket
  - Prototyping Eurocard PCB TRU COMPONENTS SU527769
- ❑ Dupont Jumper wires
- ❑ Brass insert for plastic M3  $\varnothing$ 5.0mm x 5.0mm (50)
- ❑ Cylindrical head screw M3x12mm DIN 912 (50)