LDRDuino, a construction guide: Measuring laser intensity underwater in a sample chamber with an open-hardware solution

Freya Whiteford
2250975w@student.gla.ac.uk

Chas Nelson
chas.nelson@gla.ac.uk

University of Glasgow, 2019

Abstract

An open-hardware laser intensity meter suitable for a submerged chamber was constructed using a light-dependent resistor connected to an Arduino Uno microcontroller. Here we present a step-by-step guide to constructing the device, as well as troubleshooting suggestions and areas for further improvement.

This design was originally based on that of [3].

1 Bill of Materials

- Arduino Uno or similar microcontroller with power source, which can be connected to a computer
- Adafruit ADS1115 16-bit analogue-to-digital converter (with associated drivers installed into Arduino IDE)
- Breadboard and jumper wires
- Light-dependent resistor of appropriate peak wavelength (here, 550nm) for laser to be measured, soldered onto long wire with header pins
- Resistor which allows best dynamic range for LDR used (here, 4.99k ohm - this will vary by LDR)
- Syringe (or equivalent sample holder in which to place the LDR at sample position)
- Electrical tape for rudimentary waterproofing

2 Construction

1. Assemble components as in Figure 1, connecting the analogue-to-digital converter to the Arduino Uno via I2C. If the Arduino Uno does not have individually labelled SDA and SCL pins, as some older models do not, connect the Adafruit ADC to analogue pins 4 and 5 as shown.

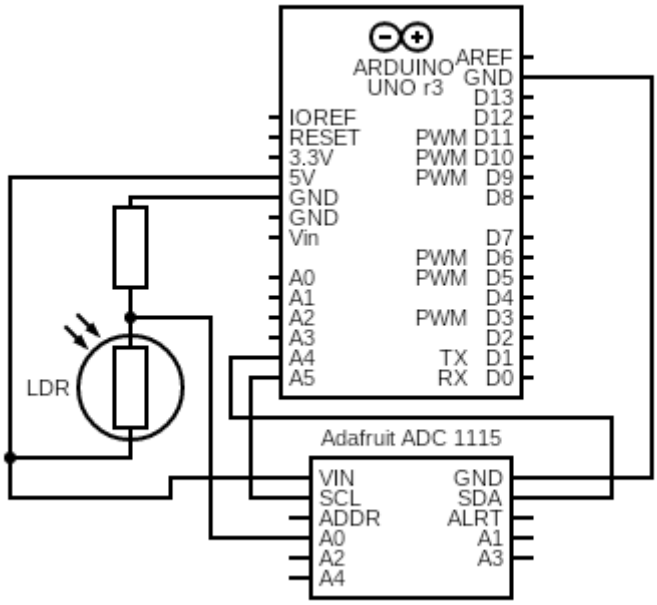

Figure 1: Circuit diagram of the Arduino, LDR and ADC

2. With the LDR on a long wire, thread it through a sample syringe so that it can be placed in sample position. Wind electrical tape tightly around the end of the syringe and the back of the LDR so that rudimentary waterproofing is achieved.

3. Via USB-B, connect the Arduino Uno to a computer with the Arduino IDE. [2]

4. Through the Arduino IDE, ensure the correct drivers are downloaded for the Adafruit ADS1115 ADC [1] and open the edited version of the example Adafruit sketch "singleended", titled "adcRead".

5. Uploading this sketch to the Arduino will now output the voltage across the LDR in mV to the serial monitor, which can be verified by opening the serial monitor through the Arduino IDE. Check that as light intensity incident upon the LDR increases, voltage also increases.

6. Now Python can be used to intercept the serial monitor. Running the LDRVoltage script should create a GUI which continuously outputs the voltage across the LDR.

7. With this script running, place the LDR at a position inside the beam path at which laser intensity can also be accurately measured using an external bolometer or similar device. For best results, ensure that the laser beam is entirely contained within the face of the LDR's sensor, and ideally place the LDR in the exact sample position to be measured.

8. Record the voltage across the LDR for a wide range of known incident laser powers, saving this data as a text file whose structure resembles that of the exemplar data (known power followed by LDR voltage).

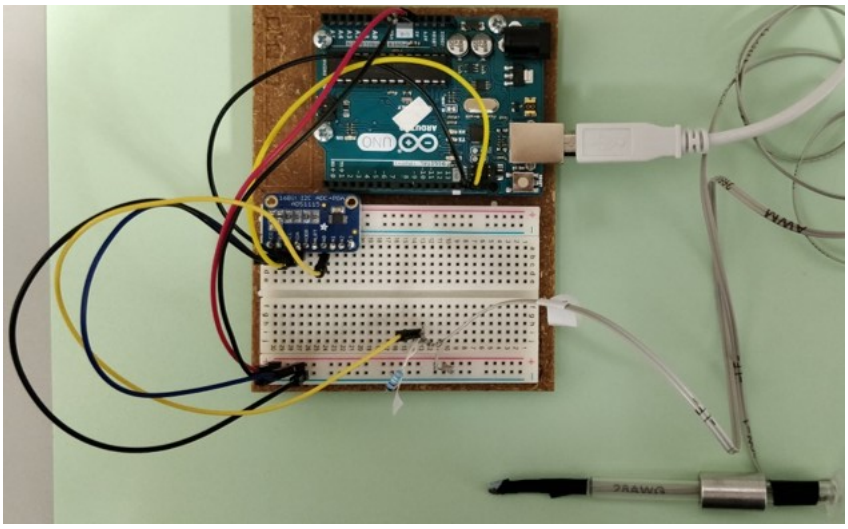

Figure 2: Assembled components

9. Feeding this calibration text file into the LDRPowerFromVoltage script will now return a live estimate of laser power incident upon the LDR through polynomial fitting.

3 Issues and improvements

1. The LDR is very sensitive to direction, and can misrepresent intensity if its sensor face is not entirely front-on to the laser beam.

2. Likewise, if the laser beam is not entirely contained within the face of the LDR, it is likely that some intensity will not be recorded. This could be improved with a larger sensor size to assist in alignment.

3. Although LDRs can be swapped out to match the wavelength of light to be measured, they should ideally be separately calibrated to account for sensitivity potentially differing between resistors. This could become impractical when dealing with a multi-wavelength laser system.

4 Troubleshooting

4.1 The Arduino Uno is not recognising the ADC when connected.

After ensuring that everything is wired properly, and the correct drivers are downloaded in the Arduino IDE, try re-loading the adcRead sketch to the Arduino Uno. If this fails, check if the I2C connection between the Arduino and ADC is working using a scanning sketch such as "i2cscanner". [4]

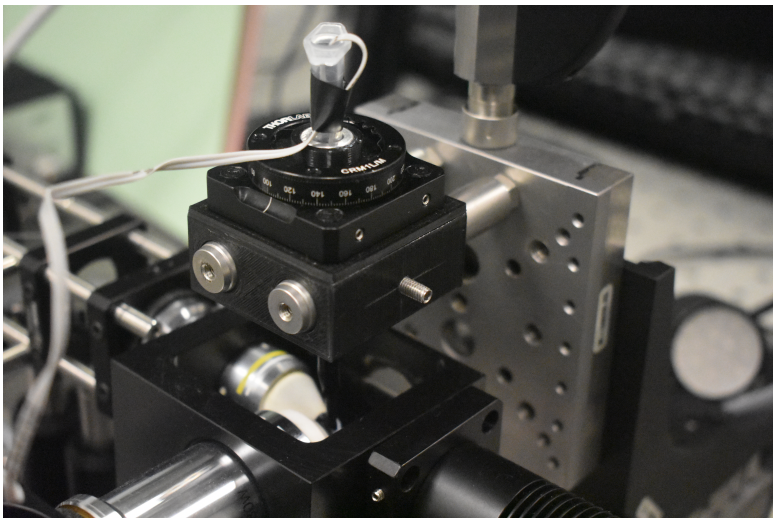

Figure 3: The device in situ at a sample chamber.

4.2 The ADC is connected properly but not registering any signal, or the signal is stuck at a high value.

This is likely to be a wiring or soldering issue. After checking that no ADC pins have been inadvertently connected to power or grounded, and that the correct analogue pin is being reported in adcRead, ensure that the ADC's header pins have been accurately soldered to its board for a secure connection.

4.3 The device is mis-estimating laser power.

- Depending on where calibration measurements were recorded for the LDR, intensity indications can vary. For best results, calibrate the LDR when it is in the exact sample position to be measured - if exact external power readings cannot be taken for that position, approximate by measuring at an earlier point in the beam path and applying the transmittance of the optical system between those two points.

- The polynomial degree of the fitting performed in LDRPowerFromVoltage can also be adjusted to account for the number of data points collected. If the device has difficulty measuring intensity while submerged, ensure there are plenty of calibration points recorded for low laser powers.

5 Bibliography

References

[1] *Adafruit ADS1115*. `https://learn.adafruit.com/adafruit-4-channel-adc-breakouts/arduino-code`.

[2] *Arduino Integrated Development Environment*. `https://www.arduino.cc/en/main/software`.

[3] Dirk Dormann. "IntensityCheck – The light measuring app for microscope performance checks and consistent fluorescence imaging". In: *PLOS ONE* 14.3 (Mar. 2019), pp. 1–19. DOI: 10.1371/journal.pone.0214659. URL: https://doi.org/10.1371/journal.pone.0214659.

[4] *I2C Scanner*. https://playground.arduino.cc/Main/I2cScanner.