Basic Neuroscience

An inexpensive Arduino-based LED stimulator system for vision research

Petteri Teikari\textsuperscript{a,b,\ast}, Raymond P. Najjar\textsuperscript{a,b,2}, Hemi Malkki\textsuperscript{c,3}, Kenneth Knoblauch\textsuperscript{a,b,4}, Dominique Dumortier\textsuperscript{d,5}, Claude Gronfier\textsuperscript{a,b,\ast\ast,1}, Howard M. Cooper\textsuperscript{a,b,\ast\ast,1}

\textsuperscript{a} Stem Cell and Brain Research Institute, INSERM U846, 18 avenue Doyen Lépine, 69500 Bron, France
\textsuperscript{b} Université Claude Bernard, Lyon I, Lyon, France
\textsuperscript{c} Cognitive and Systems Neuroscience, Center for Neuroscience, Swammerdam Institute for Life Sciences, University of Amsterdam, P.O. Box 94232, 1090 GE Amsterdam, The Netherlands
\textsuperscript{d} École Nationale des Travaux Publics de l’Etat, LASH, DGBR, Rue Maurice Audin, 69518 Vaulx-en-Velin, France

HIGHLIGHTS

- We propose an inexpensive system to control light intensity for research applications in vision research.
- The system is based on light emitting diodes (LEDs) and open-source Arduino microcontroller.
- The visual stimulator has been used in applications including rodent pupillometry, cognitive learning, and human psychophysics.
- Applications that can be easily implemented for students, educational purposes and universities with limited resources.

ARTICLE INFO

Article history:
Received 24 June 2012
Received in revised form 11 September 2012
Accepted 13 September 2012

Keywords:
LED
Arduino
Visual stimulator
Microcontroller
Pupillometry
Education

ABSTRACT

Light emitting diodes (LEDs) are being used increasingly as light sources in life sciences applications such as in vision research, fluorescence microscopy and in brain–computer interfacing. Here we present an inexpensive but effective visual stimulator based on light emitting diodes (LEDs) and open-source Arduino microcontroller prototyping platform. The main design goal of our system was to use off-the-shelf and open-source components as much as possible, and to reduce design complexity allowing use of the system to end-users without advanced electronics skills. The main core of the system is a USB-connected Arduino microcontroller platform designed initially with a specific emphasis on the ease-of-use creating interactive physical computing environments. The pulse-width modulation (PWM) signal of Arduino was used to drive LEDs allowing linear light intensity control. The visual stimulator was demonstrated in applications such as murine pupillometry, rodent models for cognitive research, and heterochromatic flicker photometry in human psychophysics. These examples illustrate some of the possible applications that can be easily implemented and that are advantageous for students, educational...
1. Introduction

Light-emitting diodes (LEDs) have become more and more common as a low-cost and flexible way to provide light stimuli in vision research (Nygård and Frumkes, 1982; Schoolfield and Murdoch, 1987; Watanabe et al., 1992; Pokorny et al., 2004; Fadda and Falsini, 1997; Demontis et al., 2005; da Silva Pinto et al., 2011; Rogers et al., 2012). LEDs offer easier and more versatile control of light characteristics (Schubert and Kim, 2005) compared to traditional light sources such as xenon, mercury, metal halide and halogen lamps which often require various auxiliary devices in experimental settings such as a set of filters for spectral tuning and shutters to control exposure duration. LEDs are close to being monochromatic light sources (half-bandwidths of 20–30 nm) and they provide roughly linear light output (Svilainis, 2009) in response to pulse-width modulation (PWM, Narra and Zinger, 2004) control signal over an extended dynamic range. The response time of LED chips itself is very fast (in the order of nanoseconds). The whole system response of the LED driver and the LED chip together is slower, but still sufficient for most purposes, the response time being in the order of microseconds (Albeau et al., 2008).

An additional advantage of LEDs is their characteristic low electromagnetic interference (EMI) emission (Svilainis, 2012), as compared to cathode ray tube (CRT) monitors that can cause significant electromagnetic interference to electrical cables (Hogg, 2006), making them especially suitable to be used with electrophysiological techniques such as electroretinography (ERG, Fadda and Falsini, 1997), magnetoencephalography [MEG, Wilson et al., 2009], and electroencephalography [EEG, da Silva Pinto et al., 2011] in which the physiological potentials are typically very weak and are prone for artifacts.

LEDs are typically controlled with a discrete on–off PWM signal, in which the on-time (duty cycle) is in theory linearly proportional to the light output. More detailed discussion on the advantages of LEDs over traditional light sources can be found in regard to the noise and stability characteristics (Brophy, 1967; Rumyantsev et al., 2004; Salzberg et al., 2005), and to biological applications such as in widefield microscopy (Holman, 2007) and visual neuroscience (da Silva Pinto et al., 2011).

Despite the widespread use of LEDs in scientific research, there is only a handful of papers dedicated for the design of LED stimulators for research purposes (Watanabe et al., 1992; Pokorny et al., 2004; Fadda and Falsini, 1997; Demontis et al., 2005; Nishimura et al., 2006; Albeau et al., 2008; da Silva Pinto et al., 2011; Rogers et al., 2012). Additionally, the existing approaches have relied on either off-the-shelf proprietary laboratory equipment (e.g. Fadda and Falsini, 1997), or advanced electronics from the end-users (e.g. Watanabe et al., 1992; Demontis et al., 2005). da Silva Pinto et al. (2011) recently described an open-source portable LED stimulator promoting the reproducibility of their device by providing the hardware schematics, layouts, parts lists and application source freely on their support website (http://sites.google.com/site/mculestledstimulator/), the cost of their hardware parts being around $100 (~$76).

However, based on our experience the described implementation of a LED stimulator might be technically too challenging for vision research laboratories without staff with good electronics and programming skills. We sought to simplify the design and propose an even simpler and low cost alternative for the design of da Silva Pinto et al. (2011) using the open source electronics prototyping platform Arduino [http://arduino.cc/], for an in-depth review of the technology see Wheat (2011), and for applications and programming see Margolis (2011) which is based on flexible, easy-to-use hardware and software. There is a lively community of Arduino users contributing and benefiting from their collective research (see Arduino Playground for example, http://arduino.cc/playground/).

Arduino has been employed in scientific applications such as LED light control of an open-source in vivo multispectral imaging system for rodents (Sun et al., 2010), teaching violin bowing by providing an interface for combining motion sensing and vibrotactile feedback (van der Linden et al., 2011), smart textile design with optical fiber integration to clothes (Parkova et al., 2011), wearable computing (Buechley and Eisenberg, 2008; Isyama et al., 2011), and in cross-disciplinary teaching of biology and computer science (Grasel et al., 2010).

Following the example set by da Silva Pinto et al. (2011), we provided all the hardware designs and application source code freely on our project website http://code.google.com/p/arduino-v-neusci/ distributed under the GNU General Public License (GPL, version 3, http://www.gnu.org/copyleft/gpl.html).

2. Materials and methods

2.1. Design of the LED stimulator

2.1.1. Overview

The LED stimulator is designed around the Arduino prototyping microcontroller platform (or other Arduino-compatible boards such as the Maple®) consisting of five components: a computer with chosen programming environment, the Arduino prototyping platform, the constant-current LED driver, the LED itself, and a power supply (Fig. 1).

The computer is used for online control of the light intensity, either setting the light intensity manually by the experimenter or making the light intensity depend on the subject input (e.g. if the subject of the experiment controls the light intensity with a joystick) or some external trigger such as some other stimulus presentation or sensor reading.

The Arduino is used in analogous fashion to a data acquisition device [DAQ, see for example Stankus et al., 2010] to read analog/digital inputs from external switches, buttons, joysticks,
sensors, etc. [see for example D’Ausilio, 2011], and providing PWM signal out to control the LED drivers.

The maximum current of the PWM output of the Arduino Uno used in this study is only 50 mA whereas typical high-power LEDs can be driven with currents ranging from 350 mA up to several amperes. The use of proprietary LED driver makes the system more modular, allowing the stimulator system to be equipped with the most suitable LED driver for the desired application. Most LED drivers in the market are designed to produce constant-current output for driving LEDs in contrast to previous voltage-controlled schemes with complex feedback circuits (Watanabe et al., 1992).

The modular interface between the LED driver and the LED in practice means that the LED can be chosen freely from available options ensuring that the LED can withstand the maximum current of the LED driver. The power supply can be then any $5\,\text{V}_{\text{DC}}$ source preferably a battery or linear power supply to minimize the possible electromagnetic interference from switched power supplies. Additionally, exactly the same PWM-based driving scheme is used for electrical motors in robotics (Warren et al., 2011), in computer numerical control (CNC) machining (e.g. http://www.linuxcnc.org/), in 3D-printing (e.g. RepRap, Jones et al., 2011), and in speed control of DC motors (e.g. http://arduino.cc/it/Main/ArduinoMotorShieldR3).

2.1.2. Hardware

2.1.2.1. Microcontroller. The Arduino Uno is an open-source microcontroller board based on the ATMega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs) and 6 analog input. It contains everything needed to support the microcontroller, and it can be simply connected to a computer with an Universal Serial Bus (USB) cable to get started. Alternatively, a more advanced implementation Arduino Mega2560 can be chosen with more input/output pins (54 pins compared to 14 pins of Arduino Uno). The design described here is compatible with other Arduino boards including the Arduino Mega2560.

The Arduino Uno can be programmed with the Arduino Integrated Development Environment (IDE) software based on the Processing IDE (http://processing.org). The C-based simple program code for the Arduino is referred to as a sketch. Collection of sketches for specific functionalities are referred to as libraries (http://arduino.cc/playground/Code/Library), often simplifying the programming task of the end-user hiding the code inside the libraries (‘black box’ approach). The Arduino can be programmed within the limitations of the 32 KB memory and the processing power of the microcontroller (Wheat, 2011). Arduino can function autonomously without being connected to a computer, or alternatively programmed to respond mainly to commands sent from the computer via various software interfaces (http://arduino.cc/playground/Main/InterfacingWithSoftware) or to the data acquired from the input channels (http://arduino.cc/playground/Main/InterfacingWithHardware). Additionally, the Arduino can be programmed and controlled via third-party programming environments such as Splish (Kato, 2010).

The hardware functionality of Arduino can be extended with external plug-ins referred as shields (http://shielddlist.org) commercially manufactured or created by the user. The functions of shields include for example adding networking capabilities such as ethernet (http://arduino.cc/en/Main/ArduinoEthernetShield), Bluetooth (http://iteadstudio.com), ZigBee (http://arduino.cc/it/Main/ArduinoXbeeShield); TFT touch panel capability (http://www.adafruit.com/products/376); data logging capability without the need of a computer (http://www.ladyada.net/make/logshield); joystick and button combination for reading user input (http://www.sparkfun.com/products/9760); motor drive shields (http://arduino.cc/en/Main/ArduinoMotorShield); and specific shields with LED drivers included for LED control (http://www.chesterstgaram.com; http://neuroelec.com/hp-rgb-led-shield). In practice, the ethernet shield gives the Arduino an IP address allowing it to be controlled over the internet, the Bluetooth shield allows wireless communication with mobile phones running on Android (Amarino project, http://www.amarino-toolkit.net/) and iOS/iPhone (e.g. http://www.sparkfun.com/tutorials/152), and the ZigBee shield interaction with wireless sensor networks for example Faludi (2010).

As Arduino is released as an open-source hardware project, there are various non-commercial and commercial implementations of the same idea with inter-compatible interfaces. In practice, some of the functions can be stripped out for cheaper or smaller platforms as with Oak Micros om328p (http://www.arduino.cc/playground/Main/om328p), or alternatively upgrading some components as for example with the case of Maple Rev. 5 allowing higher PWM resolution and higher PWM frequency. Furthermore some Arduino-compatible modifications are optimized for certain applications as for example for light control (Lightduino, http://www.toastedcircuits.com/), Brain–Computer Interfacing (BCI, http://pceeg.sourceforge.net), and Global Positioning System (GPS) data logging platform for wireless telemetry (ArduinoMetry, http://www.digitalmiser.com/projects/arduemetry). Interested readers are referred to the work of Mellis and Buechley (2012) for description of the collaborative modification process of Arduino-compatible boards.

2.1.2.2. LEDs. The LED light chips come in various form factors ranging from small surface-mounted devices (SMD) LEDs to larger high-power LED sources that are often mounted on metal bases allowing easy soldering and connection with the heat sink. Recently, solderless LED sockets have been introduced both for high power LEDs (e.g. Molex, http://www.molex.com/; Tyco Electronics, http://www.te.com/catalog/minf/en/763) and for LED light modules (e.g. Xicato, http://www.xicato.com/; Vexa Lumaera, http://www.vexica.com/; Philips Fortimo, http://www.lighting.philips.com) making LEDs easier to implement in a research setting.


2.1.2.3. LED driver. LED drivers are widely available in roughly three different groups, based on their form factor as for LEDs: (1) the surface-mounted devices (SMD) that are very small (1–3 mm × 1–3 mm) designed for mass produced electronics, (2) the “medium-sized” LED driver requiring either soldering or being supplied with factory-soldered wires, and (3) LED drivers designed for architectural lighting not often requiring any soldering.

The first group of SMD LED drivers offer the most technical flexibility and performance, but can be hard to implement in environments without advanced soldering equipment and staff. Commercial vendors for SMD LED drivers include Analog Devices (http://www.analog.com), Diodes Inc. (http://www.diodes.com), Infineon Technologies (http://www.infineon.com), Maxim...
The second group from which our XP Power (http://www.xppower.com) was selected (see Section 2.2.3) provides a compromise between the technical performance in ease-of-use in environment with only basic soldering tools while still providing good solutions for scientific applications. Commercial vendors include, LuxDrive producing BuckPuck (http://www.luxdrive.com), and Recom (http://www.recom-international.com).

The third group often contain all the accessory electronics around the LED drivers like resistors, capacitors, coils and all the connectors ready assembled. The LED drivers from this group might not be suitable for applications requiring fast response times if they are optimized for architectural lighting with high reactive impedance damping their response to electromagnetic interference (EMI) emission. Commercial vendors include Integrated System Technologies (http://www.istl.com), Osram GmbH (http://www.osram-os.com) and Philips (http://www.advance.philips.com).

Special categories of LED drivers include dedicated LED flash circuits mainly used to drive the LED flashes used in camera phones that could be used as well as in scientific applications such as in the flash electroretinogram (Viswanathan et al., 2011), requiring pulsed lights with a lower frequency compared to PWM pulsing. Additionally, Nishigaki et al. (2006) used a custom-built LED driver circuit providing stroboscopic illumination in fluorescent cell imaging in which the reduced light exposure was shown to reduce the phototoxicity of the illuminating light to the imagined cell. Dedicated LED flash circuit manufacturers include among others Analog (e.g. ADP 1650), Maxim (e.g. MAX1583), National Semiconductor (e.g. LM2754), and Texas Instruments (e.g. TPS61310).

Additionally one can search for further information from magazines devoted to LED technology, those including for example LED Professional (http://www.led-professional.com), LEDs Magazine (http://www.ledsmagazine.com), LED Journal (http://www.ledjournal.com), and more general Electronic Engineering Times (http://www.eetimes.com).

A cost estimate of the proposed LED stimulator system with exemplary LED components is shown in Table 1.

Table 1: Cost budget of the system. Update later, just to give view for the reader of the cost of the proposed system which is very low. URL: Uniform Resource Locator.

<table>
<thead>
<tr>
<th>Component</th>
<th>URL</th>
<th>Cost (£)</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED Driver (e.g. XP Power)</td>
<td>URL (<a href="http://www.xppower.com">http://www.xppower.com</a>)</td>
<td>~13</td>
<td>~17</td>
</tr>
<tr>
<td>LED Optics</td>
<td>URL (<a href="http://www.eetimes.com">http://www.eetimes.com</a>)</td>
<td>~1–5</td>
<td>~1.3–6.5</td>
</tr>
<tr>
<td>Heat sink</td>
<td>URL</td>
<td>~3.5</td>
<td>~4.6</td>
</tr>
<tr>
<td>Thermal adhesive</td>
<td>URL (<a href="http://www.computerpower.com">http://www.computerpower.com</a>)</td>
<td>~10</td>
<td>~13</td>
</tr>
<tr>
<td>High-Power LED, Lumiled</td>
<td>URL (<a href="http://www.lumileds.com">http://www.lumileds.com</a>)</td>
<td>~3–7</td>
<td>~3.9–9.2</td>
</tr>
<tr>
<td>DC Power Supply (linear)</td>
<td>URL</td>
<td>~10–20</td>
<td>~13–26</td>
</tr>
</tbody>
</table>

(required, for example, for Matlab and for LabVIEW. The use of Python facilitates the integration of the Arduino system described here, for example to existing electrophysiological (Spacek et al., 2009) and visual psychophysics (Peirce, 2007) software packages written in Python. The access to Arduino is very straightforward under Unix-systems (Linux, Mac) as it functions as any serial interface, thus allowing one to read and write to it as if it was a file. Additionally, there is a wrapper library called pySerial (http://pyserial.sourceforge.net/) that handles the serial interface between the computer and the Arduino, a library that works across all operating systems.

Additionally to fully open-source Python implementation, LabVIEW virtual instruments (Vis, “LabVIEW programs”) were developed by us (available from: http://code.google.com/p/arduino-v-neusci/) to demonstrate simple LED intensity control with PWM along with an implementation of the heterochromatic flicker photometry paradigm (see Section 2.2.3). The VI development was done using the NI LabVIEW Interface for Arduino Toolkit (LIFA, http://ni.com/arduino). The LIFA comes with its own Arduino sketch (adiosrv.pde) that need to be uploaded to your Arduino board before running our Vis or the example Vis provided by National Instruments (http://www.ni.com/). The LIFA supports sampling rates of 200 Hz for wired communication and for 25 Hz wireless communication with either Bluetooth or Xbee/ZipBee. A simplified Matlab implementation was done using the Matlab ArduinoIO Package (Giampietro Campa, MathWorks, Inc., http://www.mathworks.com/matlabcentral/fileexchange/32374-matlab-support-package-for-arduino-aka-arduinoio-package). We chose to use the sketch file adiosrv.pde from the provided files, the sketch file functioned in practice as an analog/digital input and output server for Matlab. The provided Matlab class arduino.m was installed for Matlab allowing the access to the input and output pins of Arduino. In other words, the sketch file was uploaded to Arduino Uno using the default IDE allowing us to write and read digital and analog values from the Arduino under Matlab. For full details on the ArduinoIO package, one is referred to documentation of the Arduino IO Package, and http://code.google.com/p/arduino-v-neusci/ for the examples developed by us. Alternatively, the Arduino calls could have been implemented directly using the lower level serial port communication capability of Matlab. The Matlab implementation is useful for example with the widely used psychophysics package Psychophysics Toolbox for Matlab (Brainard, 1997).

2.2. Light control experiments

The use of the Arduino in neuroscience applications is demonstrated using three typical applications implemented in our laboratories, allowing also the demonstration of different technical characteristics of the Arduino LED stimulator system.

2.2.1. Mouse pupilillometry

Technically the simplest demonstration application presented by us, is the light stimuli delivery from LEDs equipped with interference filters (Fig. 2) to measure the murine pupillary light reflex (Mure et al., 2007; Hussain et al., 2009). Recently similar LED-based chromatic pupilillometry was employed by Herbst et al. (2011) to study the human pupillary light reflex (PLR). The mechanical shutter used previously by Mure et al. (2007) could to be replaced either with an external transistor–transistor logic (TTL) 5 V trigger signal to Arduino’s digital input, or with an integrated timing function directly for the used Python code (Arduino sketch GuiArduinoLED.ino and GuiArduinoLED.py the demonstration Python GUI front-end in http://code.google.com/p/arduino-v-neusci/). The corresponding Matlab implementation is GuiArduinoLED.m, and
the LabVIEW virtual instrument GuiArduinoLED.vi. The board schematics corresponding to the example code is shown in Fig. 3.

2.2.2. Cue light conditioning in rodents

In addition to driving LED intensity with the Arduino, it can be used to relay trigger signals from external sources or to generate the trigger directly from the code. One example of this kind of use, is to turn on and off a cue light based on the behavior of the mouse in the experimental paradigm of Malkki et al. (2010). The mouse in the “operant nose poke task” has to collect a sucrose pellet while the cue LED light is on by poking in the reward tray (Fig. 4), the cue light turning off upon successful collection or after a 30 s timeout. In contrast to the mouse PLR setup above, in this application Arduino can be run autonomously without any need of a control computer. Simplified breadboard schematics are shown in Fig. 5, and the electronics schematic shown in Fig. 6.

![Fig. 2. Photo of LED light source (via fiber optics on the left) and pupillometer (Arrington ViewPoint, right below) equipped with additional infrared LEDs (above right) for measuring consensual rodent PLR.](image1)

![Fig. 3. Schematics for Arduino UNO in rodent pupillometry. The digital input (D2) reads the signal initiating light stimulus corresponding to the use of mechanical Uniblitz shutter in previous setup (Mure et al., 2007). The switch is connected to the ground (GND) via a pulldown 10 kΩ resistor. The LED intensity is adjusted via a LED driver by the PWM output (PWM4 from digital pin 11) of Arduino Uno. Arduino schematic was developed using the Fritzing (http://fritzing.org/) software.](image2)

![Fig. 4. Behavioral testing chamber for mice from Malkki et al. (2010). The behavioral box with green LED stimulus lights, two operant levers (shown in the withdrawn state) and food magazine (in the middle) on the front panel. Operant nose poke task: (1) cue light on → 30 s time window during which the animal can collect a sucrose pellet by poking in the reward tray → cue light goes after collection of reward or the 30 s timeout → 5–25 s pseudorandom inter-trial interval during.](image3)
joystick/input Shield shields (e.g., DFRobot RB-DI-08 and Sparkfun Joystick Shield) for Arduino.

Software. The software front-end was developed by the authors using Python (heterochromaticFlicker.py) for the flicker demonstration (available from http://code.google.com/p/arduino-v-neusci/), and the corresponding Arduino sketch for the heterochromatic flicker fusion task being heterochromaticFlicker.ino. Alternative implementation was also provided for LabVIEW (file heterochromaticFlicker.vi).

3. Results

The quality of the output PWM signal was measured using a data acquisition card (DAQ, National Instruments™ USB-6210) with its supplied NI LabVIEW SignalExpress software for interactive data logging. The measurement results are shown in Fig. 9 with various different duty cycles.

In the cue light task (Section 2.2.2), the cue light (digital output) was triggered with a digital input signal, thus in some applications the latency from trigger to the output is of an interest. The measurement was done as above for the quality of the PWM signal. The measurement with a sampling rate of 2.5 kHz at a baud rate of 57,600 bps yields a latency of ≈2.5 ms, and at a baud rate of 115,200 bps yields a latency of ≈1.2 ms. Furthermore, if needed, the digital input–output behavior could be optimized for Arduino for example using external libraries such as the digitalwritefast (http://code.google.com/p/digitalwritefast/).

The linearity of the light output (irradiance) as a function of the duty cycle of output PWM signal was measured using a radiometer (International Light™ IL1700). The results of the measurements are shown in Fig. 10 for Luxeon K2 Cool White (light output of 120 lm at 1000 mA, LXX2-PW14-V00), using the default 8-bit PWM output of Arduino to drive XP Power™ 1000 mA constant current supply (LDU2430S1000), and with the Timer1 library (http://arduino.cc/playground/Code/Timer1) for Arduino allowing 10-bit PWM resolution the Luxeon K2 LED being directly connected to the digital output pin of Arduino (50 mA maximum output current). The Python code (bit10PWM.py) and the corresponding Arduino sketch (bit10PWM.ino) for 10-bit PWM implementation are available for download from: http://code.google.com/p/arduino-v-neusci/. Additionally, the Timer1 library could be used if the PWM frequency, or the flicker frequency of the modulating envelope would like to be modified, the latter needed in systems to measure critical flicker fusion (CFF) frequency as done for example in Gregori et al. (2011) and in Teikari et al. (unpublished results). The reason for direct connection to the output pin was that we did not have LED driver sensitive enough for low duty cycles.

4. Discussion

The inexpensive Arduino-based LED stimulator system developed using off-the-shelf components described here was shown to behave according to design criteria. The pulse-width modulation (PWM) signal provided by the Arduino was shown be of a good quality, thus leading the irradiance output of LEDs to be linear as a function of the duty cycle of PWM signal. The latency from digital input to digital output was shown to be sufficiently short for our applications, further supported by the more detailed timing precision analysis done by D’Ausilio (2011) for Arduino in behavioral neuroscience application.

Our approach was designed to be easily set up even without advanced skills in electronics in contrast to the project described by da Silva Pinto et al. (2011) that required the end-user to have the access on equipment making printed board circuits (PCBs). The
use of USB-powered and interfaced ensured good compatibility with future computers compared to the solution of Rogers et al. (2012) interfacing the LEDs via parallel port that is becoming obsolescent in modern desktop computers and laptops. Furthermore, the approach is highly modular so that the Arduino outputs PWM signal that can be used to drive the majority of LED drivers allowing the end-users to select their components easily from commercial solutions based on their individual needs. Additionally, the Arduino can be controlled using various programming environments and it is not limited to some proprietary environments such as LabVIEW.

By choosing open-source solutions both in respect to hardware (Arduino) and programming environment (Python), the capabilities to modify the proposed LED stimulator are greatly broadened and allowing novel ways of interaction between the developers and end/users (De Paoli and Storni, 2011; Mellis and Buechley, 2012). Active user communities exist both around Arduino (e.g. Arduino Playground, http://arduino.cc/playground/) and Python (e.g. http://www.python-forum.org/). Python has gained popularity recently in general-purpose programming as well as in scientific computing (e.g., Langtangen, 2009). Python-based open-source scientific software has been developed for example for visual psychophysics [PsychoPy, Peirce, 2007], stimulus presentation in vision research [Vision Egg, Straw, 2008], and in brain–computer interfacing (BCI) systems [Pyff, Venthur et al., 2010]. Additionally, various solutions exist for integrating existing program code developed for example in C/C++(http://cython.org/), Java (http://jython.org/), and Matlab [OMPC, Jurica and van Leeuwen, 2009] to Python program code.

The major limitation of the Arduino-based LED stimulator described herein is the narrow dynamic range of the light output that is only 8 bits (≈2.4 log units) or 10 bits (≈3.0 log units) optimizing timer–function of Arduino. This is over a magnitude lower compared to the previous report of 4.5 log unit dynamic range (Demontis et al., 2005), although the dynamic range obtained by us is sufficient for various applications. Given the modular structure of the designed system, the Arduino could be replaced with an
Arduino-compatible Maple Rev. 5 (Leaflabs, http://leaflabs.com/) that is based on ARM (Advanced RISC Machine) processor family providing a 16-bit PWM output (dynamic range of ≈4.8 log units). Given that the hardware design of Arduino is open-source, advanced users can modify the existing solutions and design Arduino-compatible (for the current list of compatible boards see http://arduino.cc/playground/Main/SimilarBoards) boards for example with improved timer functions allowing higher dynamic ranges for light output.

In some applications requiring the measurement environment to be as free from artifacts as possible, as the case in precision metrology and in electrophysiology, the possible electromagnetic interference (EMI) caused by rapid switching (i.e. the pulse width modulation) of the LED current can cause artifacts. The mechanism of the EMI creating in PWM-driven LED systems is similar as for PWM-driven electrical motors as analyzed by Zhong et al. (1996). The magnitude of EMI emission depend on the LED current, current fall and rise times, cable shielding, grounding, and geometry of electrical conductors in the research setup (Ott, 2009). For proper measurement of EMI-induced artifacts on electrophysiological settings, one needs to do the measurements in situ with the actual cable geometry. For comparison, in the system of da Silva Pinto et al. (2011), the EMI emitted by the system was low enough for the system to be used in EEG recordings, thus we assumed that in our applications the EMI emitted would not pose a problem. Additionally, the used commercial components Arduino Uno and XP LED Driver were Federal Communications Commission (FCC) certified for low EMI emissions.

![Arduino Uno schematic](Fritzing.org)

**Fig. 8.** Schematics for Arduino UNO in heterochromatic flicker technique used in the ocular media density measurement by the authors (Teikari et al., unpublished results). Arduino schematic was developed using the Fritzing (http://fritzing.org/) software.

![10-bit PWM signals](Timert1 library)

**Fig. 9.** The measured 10-bit PWM signals from Arduino Uno at different duty cycles (indicated on the left of the traces) using the Timer1 library. The PWM frequency was the default 400 Hz (period of one cycle 2.5 ms). The sampling rate for acquisition was 50 kHz.
In order to reduce the EMI emission one can either improve the design of the switching LED driving circuit (e.g. Wang and Xu, 2012), or use a LED driving circuit that doesn’t output pulsed current. A recent commercial solution is the provided by IST (Integrated System Technologies Limited, Aldridge, UK) with its iDrive Quad LED driver that according to the specifications provides DC current output without switching with a 16-bit dynamic range. The IST iDrive Quad, however, is not directly compatible with our PWM-based solution as it can be only driven with constant voltage (1–10V) or network lighting control protocols Digital Addressable Lighting Interface (DALL, http://www.dali-ag.org/) and DMX512/RDM (e.g. http://www.opendmx.net/) mainly used in architectural lighting control. Additionally, the dynamic range of the system could be increased by adding a switching transistor like insulated gate bipolar transistor (IGBT) to the output of the iDrive Quad or similar constant current supply, and switch the output current driving the transistor with the PWM output of Arduino.

When comparing the Arduino to existing data acquisition device (DAQ) solutions, the Arduino cannot provide the same level of temporal precision as dedicated precision DAQs. However, for most of the visual neuroscience solutions the DAQ approaches are often overqualified and overpriced as pointed out by Puts et al. (2005). The Arduino as a DAQ device was tested at site “Measuring Stuff: The Arduino DAQ Chronicles” (https://sites.google.com/site/measuringstuff/the-arduino), giving estimates for analog input sampling rate via serial connection to the hard drive at baud rate of 9600bps to be approximately 26 samples per second, and at baud rate of 115,200bps to be approximately 517 samples per second. The obtained analog input sampling rate is sufficient to replace oscilloscopes and logic analyzers in some applications as done demonstrated in the ArduinoScope project (http://code.google.com/p/arduinooscope/).

The sample rate can be further increased for burst writing the data only to the on-board SRAM of the Arduino (2 KB in Arduino Uno, and 8 KB in Arduino Mega) reaching sample rates of 8300 samples per second. The latter option of by-passing serial communication can be useful for on-board calculation based on analog input signal not requiring data logging. Additionally, the analog input could be used for example to correct the light output depending on the measured photoreceptor membrane voltage (Djupsund et al., 1995), LED junction temperature measured with various means (Siegal, 2006; Qu et al., 2010), or light output measured with a photodiode (Watanabe et al., 1992).

The computational performance of the Arduino platform in practice depends on the development of Atmel® (http://www.atmel.com) microcontrollers. For embedded applications where the Atmel-based performance is insufficient, an alternative would be to use either ARM-based Arduino-compatible boards such as Maple Rev. 5 (Leaflabs, http://leaffabs.com/), or recent ARM-based board computers such as BeagleBoard (http://beagleboard.org/) and Raspberry Pi (http://www.raspberrypi.org/). The ARM-based board computers in practice are capable of doing many of the same things as Arduino while not being designed for the same target audience. ARM-based board computers provide more computing power and Linux as an embedded operating system directly on the board at a low price, Raspberry Pi having a retail price of 25 (~20€), BeagleBoard and Raspberry Pi have less input/output pins compared to Arduino boards, thus the ARM-based board computers can be complemented with Arduino boards. BeagleBoard for example has an extension written to exploit Arduino sketches (https://github.com/prpplague/Arduino-for-Userspace-Linux). Similarly project for Raspberry Pi have been developed allowing Arduino sketches to be run on Raspberry Pi (https://projects.drogen.net/raspberry-pi/wiringpi/) extending the usability of the existing Arduino code.

In conclusion, the LED visual stimulator based on the Arduino microcontroller prototyping platform was found to be suitable for controlling LEDs in several neuroscience applications. The low-cost and ease of use of the system and ease of use makes it an attractive alternative to DAQ-based systems. The system can be used for a wide range of applications in educational and research environments with limited budgets or technical skills, and in particular for researchers, students and universities with limited resources.

Conflict of interest

PT has a received a free sample XBee shield from Farnell.

Acknowledgements

Financial support: ANR-09-MNPS-040, Retina France, Unadev, Rhône-Alpes Cible. Rhône-Alpes Cluster HVN, Federation des Aveugles et Handicapés Visuels de France, Ministère de l’Enseignement Supérieur et de la Recherche Français. We would like to thank Tuomas Poikonen for helpful comments on the manuscript, and David Weber for the help on mouse pupillometry.