

A MOBILE INTERFACE USING ANDROID® DEVICES FOR OPERANT LABORATORY COURSES

INTERFAZ MÓVIL CON DISPOSITIVOS ANDROID® PARA CURSOS EN LABORATORIOS OPERANTES

Carlos A. Pérez-Herrera, Rogelio Escobar and Brissa Gutiérrez
National Autonomous University of Mexico

Abstract

Microcontroller input/output boards communicating with computers have been used for controlling operant conditioning chambers. Although these interfaces can be used in laboratory courses that accompany lectures on behavior analysis, in many universities classrooms equipped with computers are limited. Use of smartphones and tablets has increased in recent years and, because of the processing capacities of such devices, they could replace computers where resources are limited. This paper describes the use of an Arduino® Uno microcontroller board in combination with Android® devices, as a wireless, portable, and inexpensive interface for laboratory courses on operant conditioning. The interface consists of a free Android® appli-

The editorial process related to this manuscript was managed by Dr. Kennon A. Lattal, the Editor for English Language Submissions, who designated anonymous reviewers and decided upon publication.

Rogelio Escobar and Brissa Gutiérrez, Laboratorio de Análisis de la Conducta, Facultad de Psicología, Universidad Nacional Autónoma de México.

Those interested in using the interface described can contact the second author for technical support (rescobar@unam.mx). This project was part of the undergraduate honors thesis of the first author. Preparation of this manuscript was financed by grant IT300517 awarded by PAPIIT, DGAPA UNAM to the second author. We thank Nadia M. Santillán and Rodrigo Benavides for their help during different stages of the project.

cation that communicates with an operant chamber using Bluetooth to generate basic schedules of reinforcement and store real-time data for subsequent analysis.

Keywords: interface, Arduino®, Android®, Bluetooth, instrumentation

Resumen

Las tarjetas de inputs y outputs basadas en microcontroladores que se comunican con computadoras se han utilizado para controlar cámaras de condicionamiento operante. Aunque estas interfaces pueden usarse en cursos de análisis de la conducta, en muchas universidades es limitado el acceso a laboratorios con computadoras. El uso de smartphones y tablets ha aumentado en los últimos años y debido a la capacidad de procesamiento de dichos dispositivos podrían usarse para reemplazar a las computadoras cuando los recursos son limitados. Este trabajo describe el uso de una tarjeta de microcontrolador Arduino® Uno en combinación con dispositivos Android® como una interfaz portátil, inalámbrica y de bajo costo para cursos de laboratorio sobre condicionamiento operante. La interfaz consiste en una aplicación gratuita para dispositivos Android® que se comunica con una cámara de condicionamiento operante por medio de Bluetooth y permite generar programas de reforzamiento simples y registrar datos en tiempo real para su posterior análisis.

Palabras clave: interfaz, Arduino®, Android®, Bluetooth, instrumentación

During the first undergraduate courses on behavior analysis at Columbia University, started by Keller and Schoenfeld during the 1940s, laboratory demonstrations of principles of behavior accompanied traditional lectures. Students observed the effects of a variety of schedules of reinforcement using rats as subjects. The purpose of laboratory courses was that students could learn, first hand, about the control of behavior accomplished with reinforcement schedules while receiving training in using laboratory equipment. Keller and Schoenfeld (1949) observed:

We concluded quickly that laboratory work must be an integral part of elementary instruction... Laboratory work, we agreed, can make for a more mature understanding of a science and its methods. It is sounder pedagogically, permitting more active participation of the student in the educational process. It gives concreteness to the lecture material and a sense of proof and security about the facts that are learned (p. 166).

During the following decades, the success of the program at Columbia University inspired the creation of new undergraduate and graduate courses in behavior analysis with laboratory courses in other universities in the United States and eventually in Brazil and Mexico during the 1970s (Colotla & Ribes, 1981; Todorov, 2006). Teaching laboratory courses, however, in most cases requires costly specialized laboratory equipment that may be difficult to acquire. This problem is particularly severe in universities in developing countries where laboratory equipment is usually imported.

Previous papers have offered a solution to this issue by showing that inexpensive microcontroller boards can be used to control operant-conditioning chambers (e.g., Varnon & Abramson, 2013) in laboratory demonstrations. Microcontroller boards allow users to detect responses using, for example, a switch and activate devices like LEDs, buzzers, motors, etc. A single program, loaded into the microcontroller, can be used to read the state of a switch (i.e., detect responses) and present a variety of stimuli. Limited memory in these boards, however, restricts the procedures that can be executed and how events are displayed to the user. For this reason, Escobar and Pérez-Herrera (2015) described how an Arduino® microcontroller board can be used in combination with Visual Basic Express Edition, running in a PC, as an interface for conducting operant research. Such an interface is adequate not only for research purposes but also for teaching laboratories associated with undergraduate courses in behavior analysis. Using the interface in laboratory courses, however, could be problematic because it requires computers running Visual Basic. In many universities, especially in developing countries, not only are resources for acquiring laboratory equipment scarce, but also access to classrooms equipped with computers may be limited.

The use of smartphones and tablets has increased significantly in recent years. One of the most popular operating systems for these devices is Android®, used in over a billion devices around the world (www.android.com). Its popularity depends on, among other things, the fact that it is an open-source platform, thus the price of devices equipped with Android® starts, approximately, at 60 US dollars (1,200 Mexican pesos) at the time this paper was written. Because of the processing and connectivity capacities of mobile devices, they could replace computers in laboratory courses and communicate with microcontroller boards. The present paper describes the design of an interface for controlling operant-conditioning chambers using Arduino® Uno microcontroller boards in combination with Android® devices for laboratory courses. This Arduino-Android interface could be used for generating simple schedules of reinforcement by instructors and students without knowledge of electronics or computer programming.

Table 1.

List of components and costs

| Components | Cost | |
|---------------------|--------------|---------------|
| | US Dollars | Mexican Pesos |
| HC-06 Bluetooth | 5.00 | 100.00 |
| Arduino UNO® | 25.00 | 455.00 |
| LED 10mm | 0.16 | 3.00 |
| Buzzer 3.7 KHZ | 1.58 | 29.81 |
| Breadboard | 2.67 | 48.69 |
| Micro Switch | 0.27 | 5.00 |
| 4 AA Battery Holder | 1.95 | 35.50 |
| 4 AA Battery | 2.07 | 37.74 |
| 2.1-mm Power Plug | 0.38 | 7.00 |
| 220-Ohm Resistor | 0.05 | 0.91 |
| Total | 39.13 | 700.00 |

Interface Components and Hardware Connections

Arduino Uno® is the most popular microcontroller board and can be purchased from numerous distributors. These boards can be used in combination with modules designed to extend its capabilities for specific applications. Some modules allow wireless communication between Arduino® boards and other devices. One example, used in the present design, is the Bluetooth to Serial Port Module: HC-06, which is a module that enables wireless communication with devices equipped with Bluetooth 2.1 or higher at a maximum distance of 10 meters. This module is inexpensive and easy to configure (see Table 1 for a list of the components and prices). The module is commercialized with a “breakout board,” that is, a board that contains the module and is equipped with labeled pins that work with Arduino® boards. Breakout boards containing the HC-06 modules are common and they are generally labeled only “HC-06” by vendors. Figure 1 shows the Arduino UNO® board and the breakout board with the HC-06. The connections shown are necessary to change the default name and password as described in the section below entitled *Change Default Name and Password*.

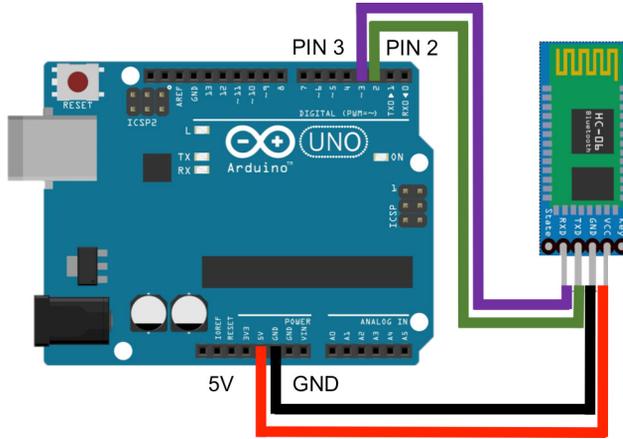


Figure 1. Arduino® Uno board and HC-06 Breakout Bluetooth Module. The connections shown are necessary to change the default name and password. The Pin VCC in module is connected to Arduino® Pin labeled 5V, Pin GND is connected to GND Pin in the Arduino. Pins RXD and TXD in the module are connected to Arduino® Pins 3 and 2, respectively.

The operant-conditioning chambers that we used for testing the Arduino-Android interface in the classroom were equipped with low-current output devices that were connected directly to the Arduino® board without relays or transistors. Food reinforcement was delivered manually by the students. A 10-mm diffused LED was used as a houselight, and a 3.7 kHz buzzer signaled students when the food reinforcer was to be delivered. This interface also can be used with 28-V DC devices commonly used in commercial operant-conditioning chambers by adding an external power supply in combination with an array of relays (see Escobar & Pérez-Herrera, 2015, for details).

A microswitch with a metal lever was used to record responses. With this arrangement, output and input pins in the Arduino Uno® board were connected directly to the operant-conditioning chamber as shown in Figure 3. A solderless breadboard was used to facilitate connections and to avoid soldering the devices.

The Arduino® Uno board is connected to 4 AA batteries using a 2.1 mm center-positive connector (see the arrangement in Figure 3) to function independently of a computer. The Bluetooth Module can be detected using Bluetooth settings in Android® devices. After the device is detected it should be paired with the Android® device using PIN code 1234 (or any custom code assigned previously). The price

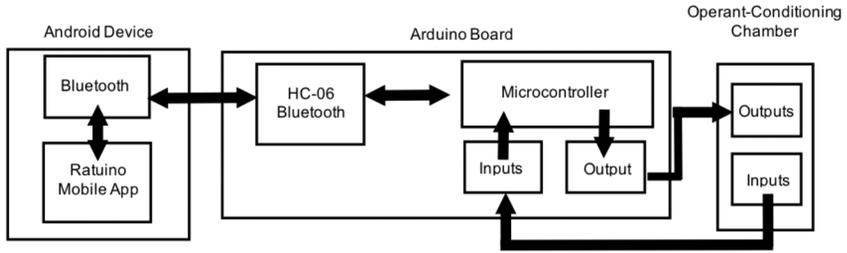


Figure 2. Scheme of the interface structure. The Android® application uses Bluetooth for establishing bidirectional serial communication with the microcontroller in the Arduino® Uno board by means of the Bluetooth module. The microcontroller activates the outputs in the operant-conditioning chamber, and detects the changes in input values.

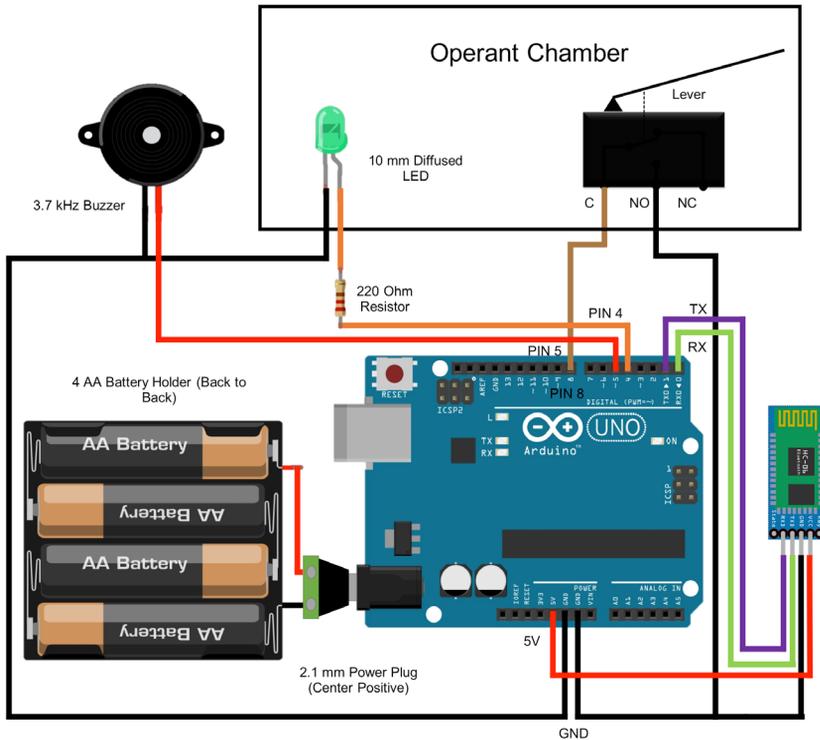


Figure 3. Schematics of the connections between Arduino® Uno board with the Bluetooth HC-06 module, and the devices in an operant-conditioning chamber. In the example, only a houselight (LED) and a lever were used in the experimental chamber. A buzzer signaled when reinforcement had to be delivered manually. Four AA batteries were used to power on the board independently of a computer. Pins RXD and TXD in the module are connected to Arduino® Pins TX (1) and RX (0), respectively. These Arduino® pins allow for serial communication with other devices.

of the breadboard, LED, buzzer, battery holder, and lever used in our tests was approximately 40 US dollars (700 Mexican pesos). A detailed list of the components appears in Table 1. All these resources can be reused multiple times in different courses.

The interface structure is shown in Figure 2. Using an Android® application, an Android® device establishes communication with an Arduino Uno® board by means of the Bluetooth module. The microcontroller on the Arduino Uno® board controls outputs and detects changes in inputs in the operant-conditioning chamber.

Initial Configuration

The first step in setting up the interface is to download and Install Arduino® IDE (see www.arduino.cc for instructions) in a computer (Mac or PC). The next steps described in this section are only required during the initial setup. After the Arduino® Uno board, Bluetooth module, and Android® device are connected and configured, the interface functions as described in the section *Android® Application*. If more than one interface will be used in the classroom, it is necessary to assign a different name and password to each Bluetooth module to avoid confusing the devices. If only one device is needed, the default configuration can be used (see *Connecting the Bluetooth Device*). A PC or Mac computer is required only for setup.

Change Default Name and Password

Download the file `Change_Name.ino` from the authors' website (http://analisdelaconducta.net/wp-content/uploads/2018/01/Change_Name.zip) and open it using Arduino® IDE. At this point, connect the Bluetooth module to the Arduino UNO® board following the diagram in Figure 1. Afterwards, connect the Arduino UNO® board to the computer. In the Arduino® IDE, verify that the board Arduino/Genuino UNO is selected in Tools->Board and that the correct serial port is selected in Tools->Port.

Quoted text in Lines 19 and 20 in the `Change_Name.ino` file can be changed to create a custom name and password for the Bluetooth Module. Once the name and password are modified, upload the code in `Change_Name.ino` to the Arduino UNO® board by clicking the “upload” icon. As noted in the code, the internal LED in the Arduino® board first will light up, then it will turn off, and afterwards it will blink indicating that the name and password have been changed. Disconnect the Arduino board from the computer, and disconnect RX and TX pins from the HC-06 Bluetooth Module.

Arduino®-Android® Interface Setup

Download the file `Arduino_Android_Mobile.ino` (http:// analisisdelaconducta.net/wp-content/uploads/2018/01/Arduino_Android_Mobile.zip) from the authors' website. Open the file within the Arduino® IDE, and upload it to the Arduino UNO® board. Connect the HC-06 Module as shown in Figure 3.

An application for Android® devices can be downloaded for free from Google Play Store (<https://play.google.com/store/apps/details?id=com.Ratuino.ratuino-mobile>). The application was programmed in Android Studio® using Java programming language. The minimum requirements for Android® devices are: Android® operating system 3.2, Bluetooth 2.1, and 268 Kb available internal memory. After installing the application, the icon of the application "Ratuino Mobile" will be added to the device.

Before running the application, Bluetooth connectivity in the Android® device must be on. In the first screen of the application (see left panel in Figure 4), session data are entered (e.g., subject name, session number, session duration, reinforcers per session, and schedule of reinforcement). The application is preloaded with four simple schedules of reinforcement: fixed ratio, fixed interval, variable ratio, and variable interval. The user can specify the values of these schedules. Generating variable schedules, based on Fleshler and Hoffman's (1962) progression, requires specifying number of intervals, and mean duration. After the values are entered, clicking on the upper "magnifying glass" icon opens a window for selecting active Bluetooth devices. Select the name used for the Bluetooth module. After selecting the device, a new layout is set to the screen (see center panel in Figure 4). On this screen, click on "start session" button to execute the selected schedule of reinforcement. When the schedule is running, number of responses, reinforcers, and session time are shown (see right panel in Figure 4). With devices where the layout cannot be shown entirely on the screen, scrolling through the screen shows all the information available. Pressing the green button with the label "Reinforce" signals the delivery of a response-independent reinforcer. This button is useful during response shaping or for shaping responses other than lever pressing. Table 2 shows a summary of the instructions necessary to configure the interface.

Once the "start session" button is clicked, the application creates a folder named "Ratuino" in the root directory of the internal memory of the device, that can be found with a file-manager app. Within the folder, a .txt file with the data from the session is recorded for subsequent data analysis. The .txt file includes information of the session, followed by an array of values consisting of a code for the experimental

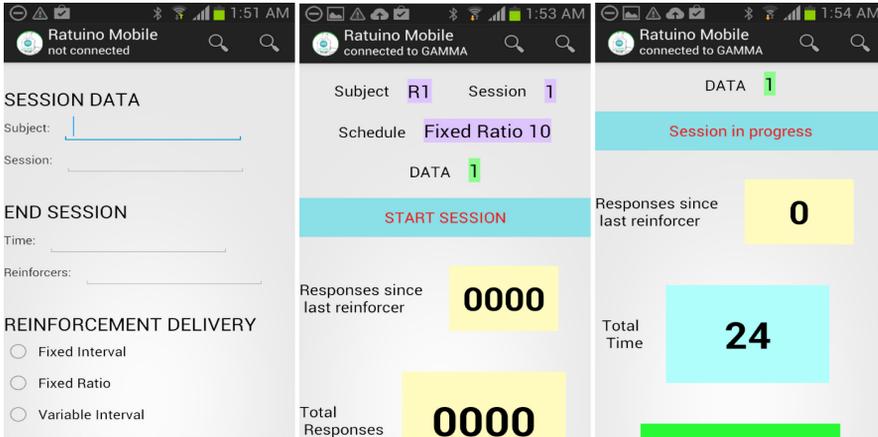


Figure 4. Sections of the layouts displayed on Android® devices by the Android® application for running simple schedules of reinforcement. Data layout (left) is used for loading the experimental session and selecting reinforcement schedule type and values. Confirm layout (center) is used for verifying the information and starting the session. Session layout (right) shows relevant data during the experimental session. Scrolling is necessary to show hidden sections of the layout.

event and the time stamp in milliseconds separated by a period. This .txt file can be imported and analyzed in Excel® using the period to split the values into columns.

Testing

The accuracy of standalone Arduino® Uno boards in behavioral research was assessed previously (e.g., D'Ausilio, 2012). A concern with the Arduino®-Android® interface, however, was that communicating with Android® devices via Bluetooth® could reduce accuracy. Therefore, several tests were performed. The tests analyzed the maximum number of responses per second that were recorded during a series of 500 simulated responses. Each test was conducted using a stand-alone Arduino® board that was programmed to activate an electromechanical relay (a simulated response) and turn it off and on 500 times in succession. The duration of the simulated response and the interresponse time were varied from 50 to 60, 70, and 80 ms. Ten tests were conducted with each combination of duration and interresponse time. Changes in the state of the relay, that is, the simulated response, were detected and recorded with the Arduino®-Android® interface connected to a smartphone (Samsung Galaxy® Model S3 mini GT-18190 with Android® 4.1 Jelly Bean) running the application. Table 3 shows the results of these tests. All 500 responses were ac-

Table 2.

Summary of instructions for using the Arduino®-Android® interface.

| Steps | Instructions |
|---|--|
| Initial Configuration | |
| 1 | Download and Install Arduino® IDE (www.arduino.cc). |
| 2 | Connect the Arduino Uno® board, the Bluetooth module, and the Android® device. |
| Change Default Name and Password (Optional) | |
| 1 | Download the file Change_Name.ino (http:// analisisdelaconducta.net/wp-content/uploads/2018/01/Change_Name.zip) |
| 2 | Connect the Bluetooth module and Arduino UNO® board (Figure 1). |
| 3 | Connect the Arduino UNO® board to the computer, open the file Change_Name.ino with the Arduino IDE and verify that the board and serial port are correct using Tools-> Board (Arduino/Genuino UNO), and Tools->Port (Serial Port) |
| 4 | Change the text in Lines 19 and 20 in the file to create a custom name and password. |
| 5 | Click the "Upload" icon. |
| 6 | The internal LED in the Arduino board will blink, indicating that the name and password have been changed. |
| Arduino® - Android® communication setup | |
| 3 | Download the file Arduino_Android_Mobile.ino (http:// analisisdelaconducta.net/wpcontent/uploads/2018/01/Arduino_Android_Mobile.zip) |
| 4 | Connect the Arduino board, open the file and upload it to the board. |
| 5 | Download and install the application Arduino-Android Operant Control from Google Play Store (https://play.google.com/store/apps/details?id=com.Ratuino.ratuinomobile). |
| 6 | Activate the Bluetooth function in the Android device and run the application. |
| Running the application | |
| 7 | Enter the session data information in the first screen of the application (see left panel in Figure 4). |
| 8 | Specify the values of the schedule. |
| 9 | Click on the magnifying glass icon located on the upper right side of the screen, which opens a window for selecting active Bluetooth devices. |
| 10 | Select the correct Bluetooth device. |
| 11 | A new layout will appear in the screen (see center panel in Figure 4). In this screen wait for the data to stabilize and click the "Start session" button. |
| 12 | A new layout with several data will appear in the screen (see right panel in Figure 4). |
| 13 | The application will create a folder named "Ratuino" in the root directory of the internal memory of the Android device, in which a .txt file will record data from the session. |

Table 3.
Speed Tests

| Response duration (ms) | Inter-response time (ms) | | | |
|------------------------|--------------------------|--------------|-------------|-------------|
| | 50 | 60 | 70 | 80 |
| 50 | 468.2(5.22) | 470(5.16) | 474.4(1.26) | 474.2(0.78) |
| 60 | 495.2(6.19) | 490.8(10.79) | 498(2.58) | 498.4(2.45) |
| 70 | 498.2(2.85) | 498.8(2.52) | 500(0) | 500(0) |
| 80 | 500(0) | 500(0) | 500(0) | 500(0) |

Note. The mean number of simulated responses detected during 10 tests of 500 simulated responses in each combination of response duration and inter-response time. Standard deviation is shown in parenthesis.

curately detected when the minimum response duration was 70 ms with a minimum interresponse time of 70 ms. Fewer responses were detected when these durations were shortened. In summary, up to 7 responses per second were recorded accurately. This resolution is sufficient for replicating a variety of operant conditioning experiments. Ferster and Skinner (1957), with pigeons as subjects, added a graph that represented 0.25, 0.5, 1, and 3 responses per second, along with each cumulative record they presented. This suggests that response rate peaked at approximately 3 responses per second. With differential-reinforcement-of- high-rate schedules, however, response rates of pigeons could reach more than 15 responses per second (Skinner, 1957). The peak rate with a rat likely is somewhat lower than this. In such rare cases of extremely high local response rates, the Arduino®-Android® interface would be inadequate for accurately recording responses.

It is important to note that when smartphones are used, incoming calls could interfere with the functioning of the interface by adding delays, and occasionally crashing the program. To avoid such problems, calls should not be made or received during experimental sessions. Incoming calls can be avoided by using the smart-phone in Airplane mode and activating only Bluetooth communication. It is also strongly recommended to avoid opening other apps while the Arduino®-Android® interface is in operation, as they could interfere unpredictably with its operation.

Use in the Classroom

The Arduino® - Android® interface was tested in a laboratory course on the experimental analysis of behavior for 12 undergraduate students halfway through their Bachelor degree at the National Autonomous University of Mexico. Android de-

Table 4.
Descriptive statistics of questionnaire scores

| Item | Statement | Mean | Range |
|------|--|------|-------|
| 1 | Using rats in the laboratory course improved learning of the topics reviewed during class. | 4.6 | 3-5 |
| 2 | Using reinforcement schedules improved learning of the topics reviewed during class. | 4.8 | 4-5 |
| 3 | Working with electronic equipment (Arduino-Android interface) and assembling it improved my learning experience of the topics reviewed in class. | 4.6 | 3-5 |
| 4 | I liked working with electronic equipment (Arduino-Android interface) and rats in the laboratory course. | 4 | 1-5 |
| 5 | Laboratory courses with electronic equipment (Arduino-Android interface) and rats are better than other courses based on students' oral presentations. | 4.7 | 2-5 |
| 6 | It was difficult using the Arduino® board in combination with Android devices like smartphones and tablets. | 2.9 | 1-5 |

Note. Items scored 1 (strongly disagree) to 5 (strongly agree).

vices used included smartphones and tablets. The 16-week course consisted of 3 hours per week of lectures accompanied by readings of a textbook: *Behavior Analysis and Learning* (Pierce & Cheney, 2004), and a 2-hour-per-week laboratory in which the interface was used. The interface was used for shaping lever pressing, and for generating fixed-ratio, fixed-interval, variable-ratio, and variable-interval schedules of reinforcement. To assess perceived value, effectiveness, and ease of use, a 6-item questionnaire prepared by the authors was used. The students rated each item using a Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). The questionnaire was sent to the students and returned by email. Ten students returned the questionnaires. Table 4 shows an English translation of the items, and the mean and range of responses for each item. Items 1 and 2 evaluated perceived importance of laboratory courses involving rats and reinforcement schedules. In response to these items, all students reacted positively to the laboratory course. In response to Item 3, students reported that the use of electronic equipment was important for improving learning during the classes ($M = 4.6$, range, 3-5). In response to Item 4, 80% of the students reported that working with electronic equipment was a positive experience ($M = 4$, range, 1-5). Nine students reported that working with electronic

equipment and rats was better than other techniques, for example students' oral presentations, commonly used in similar courses ($M = 4.7$, range, 2-5). Fifty percent of the students reported that they found difficult working with the interface. One explanation for such results is that some of the devices used by students had limited hardware and software specifications that produced a nonoptimal experience with the app. These issues, however, were helpful to determine the current minimum requirements of the interface.

Conclusions

The Arduino®-Android® interface could be useful for generating basic schedules of reinforcement for demonstrations in graduate and undergraduate laboratory courses. When used in combination with operant chambers, it could help students learn about the systematic effects of schedules of reinforcement on behavior and about basic electronics and construction of laboratory equipment. A limitation of the interface, however is that the basic schedules of reinforcement included in the Android® application cannot be modified. Future developments could include methods to modify the schedules.

Other inexpensive devices have been used for generating schedules of reinforcement in laboratory courses. Varnon and Abramson (2013), for example, described that a Propeller® microcontroller could be used for demonstrations of learning principles in laboratory courses. Their interface, however, lacked a graphical user interface. The use of an Android® app to control the Arduino® microcontroller board in our design eliminated this limitation.

As noted by previous authors, laboratory courses that accompany lectures on principles of behavior are beneficial to students in many ways. One is that laboratory courses, in which principles of behavior are observed in real environments with nonhuman animals, facilitate understanding the principles reviewed during lectures (Elcoro & Trundle, 2013).

Another advantage noted by Gomes Lopes, Lopes Miranda, Sousa do Nascimento and Dias Cirino (2008) is that laboratory courses shape research-related behavior that is important for solving problems in professional practice, and for training future researchers. Gomes Lopes et al. also noted that undergraduate laboratory courses could help disseminating experimental psychology and, more specifically, behavior analysis.

Lattal, McFarland, and Joyce (1990), noted that undergraduate laboratory courses could be significant for the future of behavior analysis because it is during

these courses that many psychology students are attracted to behavior analysis. Latal et al. found, however, that less than half of the courses on Psychology of Learning courses in the surveyed colleges and universities included laboratory courses. Furthermore, only a few of these courses had an orientation consistent with behavior analysis. Even though the potential significance of laboratory courses for the survival and growth of behavior analysis, laboratory courses in which nonhuman animals are used for demonstrations are now rare. Many of these courses have been eliminated, replaced with computer simulations, or replaced with other practices. Among the reasons for such tendency are the restrictive policies and expenses associated with maintaining animals, and the high costs of specialized electronic equipment required for demonstrations. The Arduino®-Android® interface could help lessening one of these problems by reducing the cost of the equipment required to conduct laboratory demonstrations with non-human subjects.

A popular cost-effective solution for the problem of using live animals and specialized operant equipment in laboratory courses, is using computer simulations (see Graf, 1995). Two notable examples of currently used simulations that mimic the effects of a variety of reinforcement schedules are Sniffy the virtual rat and CyberRat. Although these simulations solve the issues associated with setting up laboratory courses, Elcoro and Trundle (2013) found that when students were asked to choose between working with simulations or real rats, they choose live rats. Furthermore, students described that they enjoyed working with live rats more than with simulations, and even that course topics were more interesting when they were accompanied by demonstrations with live rats. Elcoro and Trundle's findings suggest that although simulations are a cost-effective alternative to traditional laboratory courses, they cannot replace the experience of modifying the behavior of a real subject. Tools like the Arduino®-Android® interface described in this manuscript could help bringing back the demonstrations of behavioral phenomena with live rats into the laboratory courses.

We are hoping that this low-cost interface using free-distribution Arduino® code and a free Android® application will help in disseminating behavior analysis by facilitating the pedagogic practice of accompanying lectures on principles of behavior analysis with hands-on laboratory courses using nonhuman animal subjects.

References

- Colotla, V. A., & Ribes, I. E. (1981). Behavior analysis in Latin America: a historical review. *Spanish-Language Psychology, 1*, 121-136.

- D'Ausilio, A. (2012). Arduino: A low-cost multipurpose lab equipment. *Behavior Research Methods*, 44, 305-313. <https://doi.org/10.3758/s13428-011-0163-z>
- Elcoro, M., Trundle, M., (2013). Students preferences for live versus virtual rats in a learning course. *International Journal for the Scholarship of Teaching and Learning*, 7(1), Article 16. <https://doi.org/10.20429/ijstl.2013.070116>
- Escobar, R., & Pérez-Herrera, C. A. (2015). Low-cost USB interface for operant research using Arduino® and Visual Basic. *Journal of the Experimental Analysis of Behavior*, 103, 427-435. <https://doi.org/10.1002/jeab.135>
- Ferster, C. B., & Skinner, B. F. (1957). *Schedules of reinforcement*. Prentice-Hall.
- Fleshler, M., & Hoffman, H. S. (1962). A progression for generating variable-interval schedules. *Journal of the Experimental Analysis of Behavior*, 5, 529-530. <https://doi.org/10.1901/jeab.1962.5-529>
- Gomes Lopes, M., Lopes Miranda, R., Sousa do Nascimento, S., & Dias Cirino, S. (2008). Discutindo o uso do laboratório de análise do comportamento no ensino de psicologia. *Revista Brasileira de Terapia Comportamental e Cognitiva*, 10(1), 67-79.
- Graf, S.A., (1995). Three nice labs, no real rats: a review of three operant laboratory simulations. *The Behavior Analyst*, 18, 301-3016. <https://doi.org/10.1007/bf03392717>
- Keller F. S., & Schoenfeld W. N. (1949). The psychology curriculum at Columbia College. *American Psychologist* 4, 165-172. <https://doi.org/10.1037/h0057770>
- Lattal, K.A., McFarland, J.M., & Joyce, J. H. (1990). What is Happening in Psychology of Learning Courses? *The Behavior Analyst*, 13, 121-130. <https://doi.org/10.1007/bf03392529>
- Skinner, B. F. (1957). The experimental analysis of behavior. *American Scientist*, 45, 343-371.
- Todorov, J. C. (2006). Behavior analysis in Brazil. *Avances en Psicología Latinoamericana*, 24, 29-36
- Varnon, C., & Abramson, C. I. (2013). The propeller experiment controller: low-cost automation for classroom experiments in learning and behavior. *Innovative Teaching*, 2(2), 1-18. <https://doi.org/10.2466/07.08.it.2.2>

Recibido Enero 22, 2018 /
 Received January 22, 2018
 Aceptado Abril 23, 2018 /
 Accepted April 23, 2018