2020

Developing a Computer-Controlled Treat Dispenser for Canine Operant Conditioning

Walker Arce
University of Nebraska-Lincoln

Jeffrey R. Stevens
University of Nebraska-Lincoln, jstevens5@unl.edu

Journal of
open hardware

Arce, W and Stevens, JR. 2020. Developing a Computer-Controlled Treat Dispenser for Canine Operant Conditioning. *Journal of Open Hardware,* 4(1): 6, pp. 1–10. DOI: https://doi.org/10.5334/joh.27

HARDWARE METAPAPER

Developing a Computer-Controlled Treat Dispenser for Canine Operant Conditioning

Walker Arce* and Jeffrey R. Stevens†

When performing canine operant conditioning studies, the delivery of the reward can be a limiting factor of the study. While there are a few commercially available options for automatically delivering rewards, they generally require manual input, such as using a remote control, in accordance with the experiment script. This means that human reaction times and transmission distances can cause interruptions to the flow of the experiment. The potential for development of non-supervised conditioning studies is limited by this same factor. To remedy this, we retrofitted an off-the-shelf treat dispenser with new electronics that allow it to be remotely controllable as well as act as an experiment computation, data storage, and networking center. We present a fully integrated dispenser driver board with a complementary Raspberry Pi. With rather simple modifications, the commercial treat dispenser can be modified into a computer-controlled dispenser for canine cognition experiments or for other forms of canine training or games.

Keywords: Canine; computer-controlled dispenser; operant conditioning; retrofitted dispenser
Funding Statement: This research was supported by a National Science Foundation grant to JRS (SES-1658837), including a Research Experiences for Undergraduates (REU) stipend to WA, and by a University of Nebraska-Lincoln Research Council Grant-in-Aid to JRS.

Metadata Overview

Main design files: https://github.com/unl-cchil/canine_treat_dispenser.
Target group: Scientists in canine cognition.
Skills required: Desktop 3D printing, surface mount electronics assembly.
Replication: No builds known to the authors so far.
See section "Build Details" for more detail.

(1) Overview

Introduction

Operant conditioning is a critical tool used to train pet dogs (McKinley and Young 2003). In particular, positive reinforcement involves providing a reward in the presence of a behavior to increase the expression of that behavior (McKinley and Young 2003). Clicker training is a classic example of employing operant conditioning in dogs, where a command is given and, if the dog expresses the correct behavior (or an approximation of that behavior), the dog handler presses the clicker and follows up with a food treat or other reward. These forms of operant conditioning are often used in training obedience and in reducing problem behaviors (Reisner 2016).

Operant conditioning is also used by researchers interested in studying aspects of dog cognition (McKinley and Young 2003). Clicker training, however, is often not ideal for these studies because the clicker and food dispensing must be implemented by a person, which may distract the dog from the cognitive task. For that reason, studies of dog cognition often use automated treat dispensers that are triggered by the researchers, sometimes with remote controls. For example, canine cognition can be tested by dispensing a treat when the animal subject touches a visual indicator on a screen (see demonstration video at https://www.youtube.com/watch?v=veKvqE5ipu4). This allows researchers to present a range of different stimuli to the dogs, record their responses, and reinforce the responses with food treats.

The *PetSafe Treat & Train Manners Minder Remote Reward Dog Trainer* (PetSafe 2020) is a commercial treat dispenser for training pet dogs at home using operant conditioning techniques. A user can dispense treats using a radio frequency (RF) remote control. The *Treat & Train* (PetSafe 2020) is a commonly used treat dispenser for canine operant conditioning, and it is generally controlled using its RF remote, dispensing a single treat per button press (Hiby et al. 2004, Yin et al. 2008, Hardin et al. 2015, Protopopova et al. 2016, Majikes 2017, Wallis et al. 2017, Edwards 2019). However, the inconsistencies in the human operating the remote, as well as the remote's

* Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, US
† Department of Psychology, Center for Brain, Biology and Behavior, University of Nebraska-Lincoln, US
Corresponding author: Jeffrey R. Stevens (jeffrey.r.stevens@gmail.com)

range, can limit the effectiveness of the treat dispenser. There have been instances of these dispensers being retrofitted to automatically dispense treats, but there have not been any published examples of the method to do so (Wallis et al. 2017). There have, however, been published builds of food-dispensing devices for other species, such as fruit flies (Wayland et al. 2018).

Our aim here is to describe the construction, operation, and reliability of a dispenser that not only is directly controlled by a computer (rather than a remote control) but is a self-contained experimenting system when a Raspberry Pi is integrated into the *Treat & Train*. To do this, we created a custom-printed circuit board to drive the dispenser.

We integrated a Raspberry Pi 4 single-board computer (Raspberry Pi Foundation 2020) and our custom-printed circuit board into the existing *Treat & Train* (PetSafe 2020) dispenser to create a computer-controlled adaptation of this commonly used device. Our creation allows stimulus generation using common experiment generation software such as *PsychoPy* (Peirce 2007), controls reward delivery, and performs data collection and transfer in a single, self-contained unit. This paper outlines the methods performed to retrofit the *Treat & Train* dispenser, the hardware and software designed, and the results of reliability testing in the dispensing of treats. All documentation and firmware are available at https://github.com/unl-cchil/canine_treat_dispenser.

Overall Implementation and Design

Treat & Train System

The *Treat & Train* dispenser consists of a rotating wheel with cutouts that allow treats to fall from the treat storage area to a feeding receptacle. A motor rotates the wheel and a photo interrupter sensor detects whether a treat was dispensed (**Figure 1**). This means that applying a constant current source to an infrared (IR) transmitter (LED) and an IR receiver and monitoring the output of the receiver will indicate if there is an object blocking the LED. In the case of our system, this would indicate that a treat has been dispensed, which is the output we are trying to detect. Additionally, there is a photo interrupter in the base of the dispensing wheel that can detect the movement of the wheel.

To power the IR LED, we inserted a current-limiting resistor in series with the LED power supply. On the negative end we wanted to monitor the output voltage, which should be proportional to the amount of IR light interacting with it. To do this, we placed a resistor divider of sufficiently high resistance to measure the output voltage. This allows the circuit to act as a voltmeter and gives us sufficient resolution to detect treats crossing the IR beam.

The *Treat & Train* has a brushed DC motor on-board that drives a gear and pulley system to the main dispensing wheel. The stock dispenser drives this motor at 6 V_{DC} in one direction. For our purposes, we allowed more flexibility by

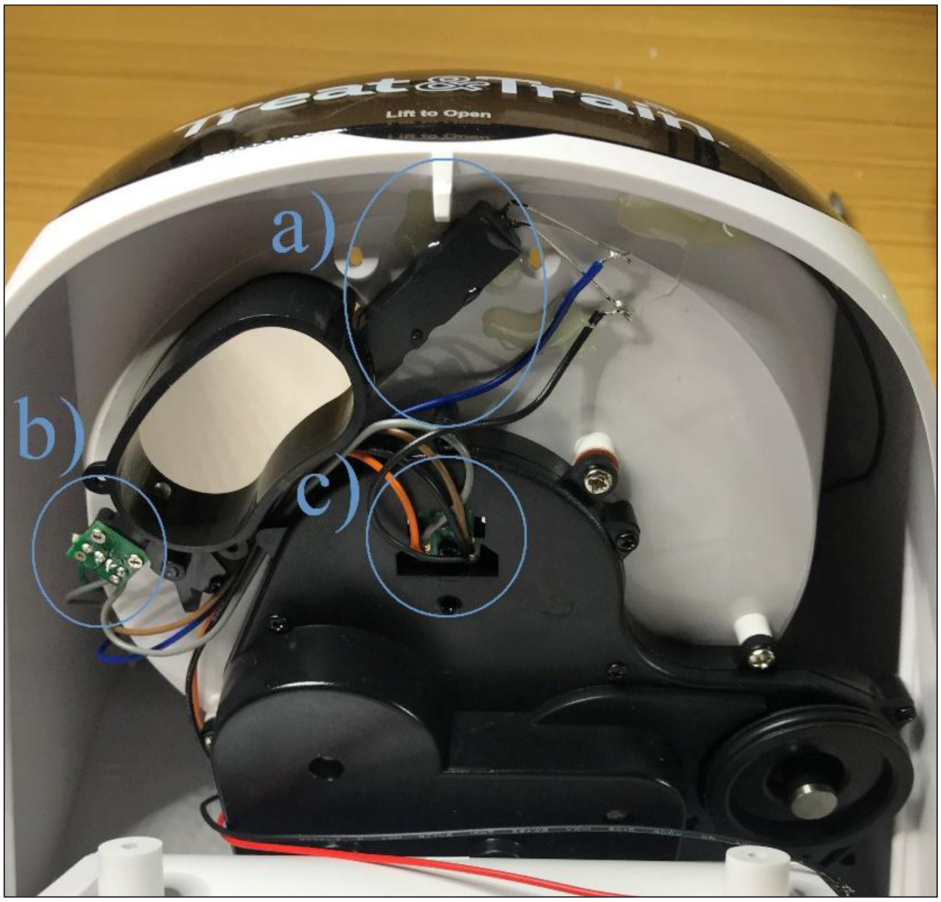

Figure 1: Location of the two IR transmitter-receiver pairs, where **(a)** is the treat IR receiver, **(b)** is the treat IR LED—so (a) and (b) constitute a single photo interrupter—and **(c)** is the wheel photo interrupter, with both parts integrated into a single package.

using a motor driver chip to drive the motor and control its speed. To drive a 6 V_{DC} motor from a USB power supply, which is set to 5 V_{DC}, we needed to boost the voltage. To allow some excess capacity/voltage for future implementations, we designed a boost regulator with an output voltage of 12 V_{DC}.

Hardware Development

We developed a simple driver board with a USB interface, microcontroller, power stage, motor interface, and an interface for the *Treat & Train* sensors. We used a microcontroller to drive these circuits, an ARM Cortex-M0+, which is supported by the Arduino IDE for easy programmability and customization. This system was then paired with a Raspberry Pi 4 to create a self-contained dispenser. Additionally, we combined this system with a television display and a USB infrared touchscreen device that is used as mouse input. The result is a complete operant system in which stimuli are presented on the television display, responses are recorded by the touchscreen, and treats are automatically dispensed based on the responses (see demonstration video at https://www.youtube.com/watch?v=veKvqE5ipu4). The connection diagram for this is shown in **Figure 2**.

Next, we designed the interfaces to the *Treat & Train* main board, which had all of its internal connections severed to be interfaced to our circuits. The first circuit is the digital interface for the RF remote 'click' signal that dispenses treats. This is triggered by connecting the positive motor output of the *Treat & Train* main board to a transistor circuit that would energize when the main board tries to drive the motor. This retains RF remote control functionality. The second circuit is the analog voltage sensing resistor divider, for which we used 1 MΩ resistors to keep the current drawn low. These are shown in **Figure 3**.

For the custom board, we used an ARM Cortex-M0+ microcontroller (ATSAMD21E from Microchip), with a TB6612 motor driver (**Figure 4**). The USB voltage was regulated to 3.3 V_{DC} using an AP2112k low dropout regulator and the 12 V_{DC} for the motor was generated using an LMR6201 1.6 MHz step-up converter with a 1.4 A switch. Finally, the right-angle USB connector was mounted such that it would be exposed out of the back wall of the *Treat & Train*.

This board was designed to work with the "Sparkfun Qwiic Micro" bootloader (Microsoft 2019) that is available through the *Boards Manager* in the Arduino IDE. Reference this firmware when determining what the

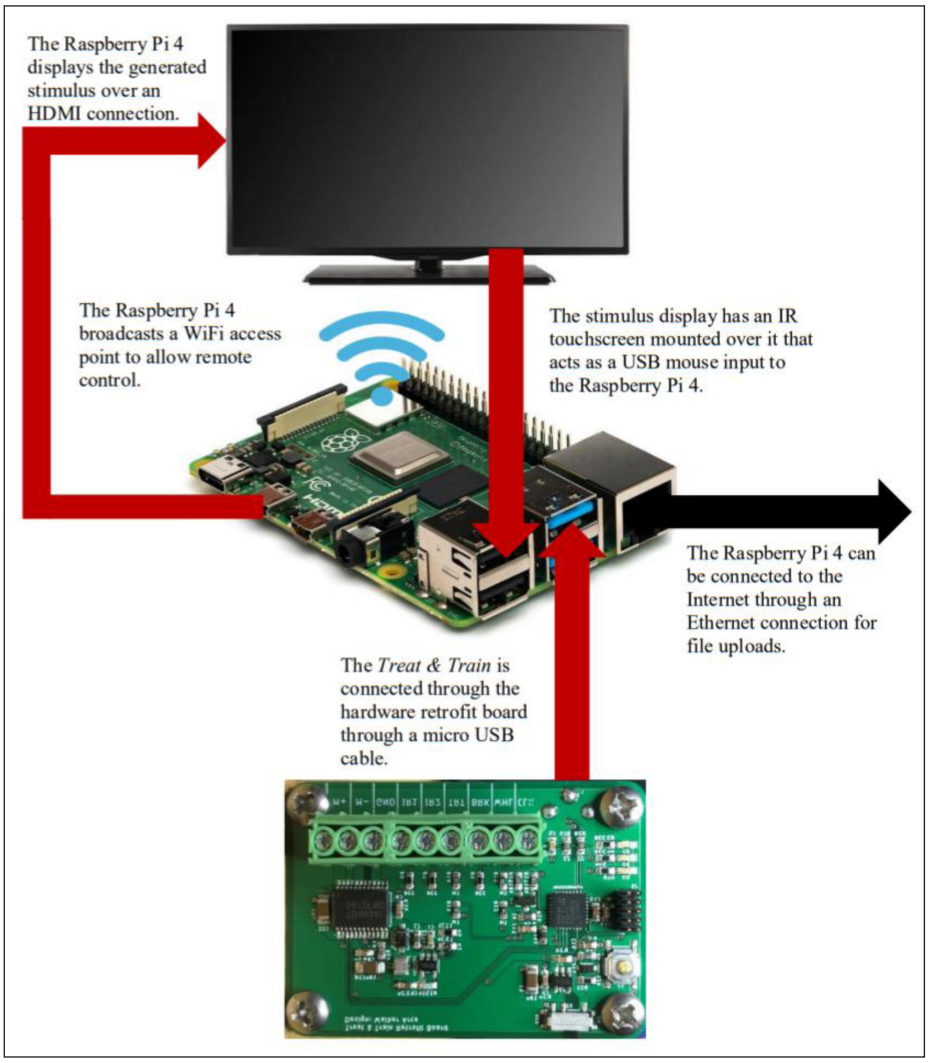

Figure 2: Connection diagram for the developed hardware.

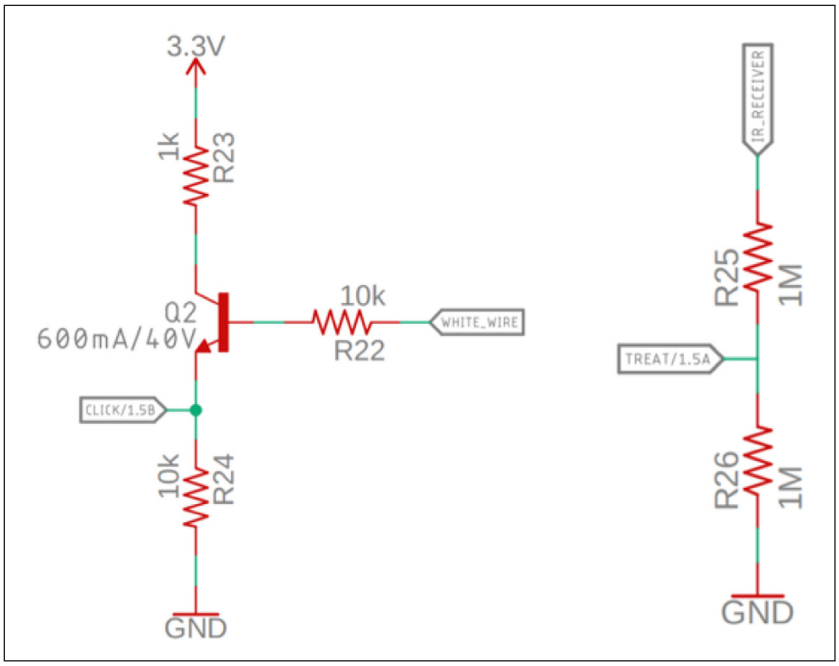

Figure 3: (Left) Transistor to detect the RF remote being activated, (Right) resistor divider circuit to read the IR receiver output voltage. Note: "WHITE WIRE" is the hardware connection to the RF receiver; "CLICK" is the output signal for the RF receiver; "TREAT" is the voltage reading from the output of the IR receiver.

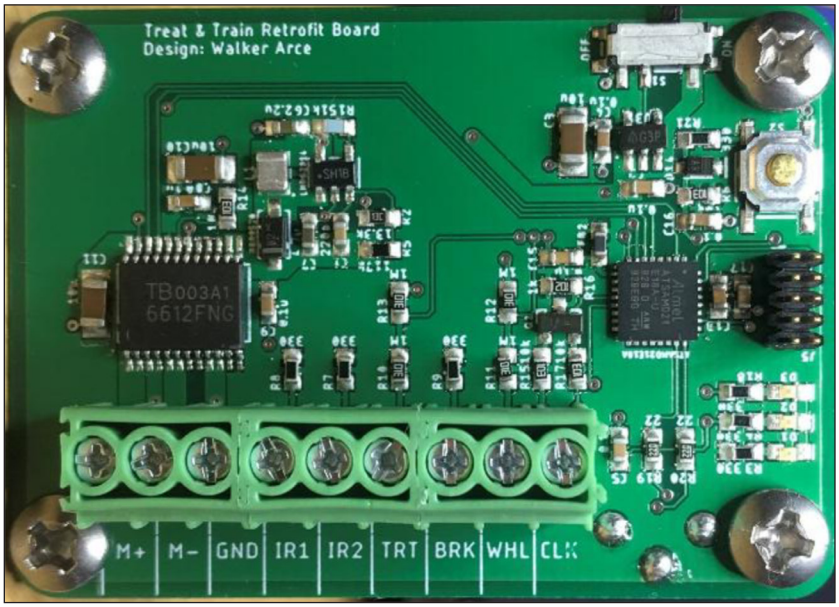

Figure 4: Custom printed circuit board.

Arduino pin names refer to if using this firmware with another processor. The pins that are broken out on the screw terminals are shown in **Table 1** and are valid for firmware version 0.4.

The internal connections from the microcontroller to the motor driver are specified in **Table 2**.

Software Development

Our implementation will have two sets of software: one set that runs on the microcontroller and another that runs on a host computer, in this case the Raspberry Pi. We composed the first set in the Arduino IDE 1.8.12 (Arduino 2020) using an open-source bootloader and the second using Python (Python Software Foundation 2019), which allows for cross-platform compatibility and testing.

The firmware for the current implementations is at version 0.4 and was designed using the Arduino IDE 1.8.12 (Arduino 2020). The firmware running on the custom-printed circuit board operates using a defined USB protocol driven by byte codes. This can be easily expanded upon by adding cases to the switch statement for the command byte (**Table 3**). A response of 0x00 indicates the command successfully executed and a response of 0x01 indicates the command failed. Commands are processed in the order they are received.

Table 1: Custom board pin descriptions and functions, applicable to firmware version 0.4.

Pin Name	Arduino Pin	Function	Description
M+	N/A	Output	Connected to positive motor pin
M-	N/A	Output	Connected to negative motor pin
GND	N/A	Input	Common ground pin
IR1	N/A	Output	3.3V power supply for IR LEDs, current limited to 5mA
IR2	N/A	Output	3.3V power supply for IR LEDs, current limited to 5mA
TRT	A1	Input	IR receiver sensor from treat sensor
BRK	N/A	Output	3.3V power supply for photointerrupter sensors, current limited to 1mA
WHL	A3	Input	IR receiver sensor output from wheel sensor
CLK	D8	Input	RF remote input for 'Dispense' button

Table 2: Arduino pin names and descriptions.

Pin Name	Arduino Pin	Description
AIN1	D7	First input for motor driver, invert from other input to drive motor in one direction or the other.
AIN2	D6	Second input for motor driver, invert from other input to drive motor in one direction or the other.
PWMA	D3	PWM input that controls the speed of the motor

Table 3: USB protocol byte descriptions.

Name	Value	Function
Start Byte	67 (Dec), 0x43, 'C'	Signals the system to start listening for input
Command Byte	1 byte	Command defined below
Data Byte	1 byte	Defined based on commands
End Byte	69 (Dec), 0x45, 'E'	Signals the end of the current command

There are a few pre-defined commands for this version of the firmware, one of which is not accessible through USB input (**Table 4**).

On the Pi, a Python class was implemented to send the above commands over a USB serial link to the driver board. This class was used with *PsychoPy* in the experiment script.

Dispenser Development

To run entire experimental setups, perform the visualization of input stimulus, control stimulus devices, and log generated data, we integrated a Raspberry Pi 4 (Raspberry Pi Foundation 2020) single-board computer running the default Raspberry Pi OS (Raspberry Pi Foundation 2020). This allowed us to achieve three goals:

1 Wireless Access Point
   – The Raspberry Pi 4 can host its own WiFi access point, which allows the experimenter to connect to the Pi and run experiments, download results, and interact with the main system.
2 Decoupled Interfaces
   – The experimenter's screen and the screen used by the canine are separate and require no extra hardware to achieve. By connecting to the Pi using SSH (Secure Shell), the command line can be exposed to run the experiments. The canine screen can be connected to the HDMI0 of the Pi and the SSH session can be started over a wireless connection.
3 Simple Setup
   – The system is expected to be easy to setup and maintain. We can use robust packages such as *piap* (Houlahan 2019) and SD card backups to easily restore the dispenser and get new dispensers up and running.

Achieving the above three criteria allows for a launching point towards more complex and integrated solutions for automated reward delivery and data logging.

The final dispenser with the new electronics and functionality had its connectors exposed on the back wall of the dispenser. The connections include the USB connector to the retrofitted control board, the Raspberry Pi 4 USB connectors, an HDMI port, Ethernet jack, and the USB-C power supply input port. This allows the Raspberry Pi 4 to display stimuli on a screen connected to the HDMI port and accept mouse inputs from an IR touchscreen adapter through on of the Pi USB ports. The Pi can also be network connected through the Ethernet jack. A short USB cable from the Pi provides power and serial communications

Table 4: USB commands that follow Table 3 protocol.

Command	Command Byte	Data	Function
Dispense	'D'	'0'–'9'	Command to dispense up to 9 treats.
Print	'P'	'0'	Prints current value of sensors, data field is set to zero in ASCII ('0').
Update RF	'B'	'0'–'9'	Updates the number of treats that will be dispensed when using the RF remote.
Dispense	'A'	0–255	Command to dispense up to 255 treats.
Motor Update	'M'	0–255	Updates the motor speed used when operating the dispenser in any mode. Default: 127
Wheel Test	'F'	'0'	Performs the wheel test routine to verify operation of the firmware and sensors.
RF Dispense	N/A	N/A	By clicking the 'Dispense' button on the RF remote, the system will dispense the defined number of treats. Default: 1 treat

to the driver board. This USB cable can be removed from the Pi, and the dispenser can be connected directly to a standard PC—bypassing the Raspberry Pi—to operate the retrofitted dispenser. This is shown in **Figure 5**.

The dispenser, when turned on, creates a WiFi access point called "operant-canine-xx", with "xx" being the dispenser number, i.e. "01" with a local address of `pi@192.168.0.1` and the password `raspberry`, which are the default values in the *pi-ap* configuration. This can be modified in the *pi-ap* software. By connecting to this AP, the Raspberry Pi can be accessed through SSH using a program such as PuTTY (Tatham 2020).

To use the RF remote to trigger the system, the *Treat & Train* main board must first be turned on before powering on the retrofitted board. If the *Treat & Train* is turned on after the retrofitted board, the 'CLK' signal will go high, as it is controlled through a transistor that is not turned off when the board is first powered on. Do not use anything other than a computer USB port for powering the system. The retrofitted board does not power the *Treat & Train* main board, so it will still require four D cell batteries to operate with backwards compatibility. This should be fixed in the next revision and the RF receiver should be powered by the custom printed circuit board.

(2) Quality Control

Safety

Operators of this dispenser should be aware of four hazards:

- Finger pinches.
- Electric shock.
- Burns when assembling electronics.
- Damage to *Treat & Train*.

Finger Pinches

The dispenser wheel turns continuously from a DC motor to dispense treats through an eyelet that, if not careful, can cause the user's finger to be pinched when trying to fix a jam.

Electric Shock

The powering of electronics using an external power source, such as when testing a newly developed circuit, can cause electric shocks if not careful. Work tables should be placed where there is no risk of splashing water. All cables used to connect sensors and power should be inspected for damage and replaced if necessary.

Burns during Assembly

The assembly of the described electronics includes the use of soldering irons and hot air rework tools. These tools use high temperatures to melt solder and pose a risk of burning when used improperly. Ensure that the user's manuals are read and adhered to when using new tools.

Damage to Treat & Train

When retrofitting an off-the-shelf device, there is always a risk of damaging or destroying the on-board electronics. Before applying power to any on-board sensors, double check your power supply and connections to prevent short circuits and over-voltage/over-current situations.

General Testing

Considering one line of canine behavior and psychology research includes the design of numerical preference studies (Ward 2007, Baker 2012, Miletto Petrazzini et al. 2020), we tested the dispenser's accuracy of dispensing treats. This gives benchmarks that allow for future work that may be needed to meet specific performance deficits that exist in the current setup and for the dispenser to be adapted to new types of studies.

For this testing, we wanted to measure the failure rate of the dispensing commands, meaning the amount over or under that it dispenses from the target amount. To do this, we performed a total of 100 tests split into 10 tests of increasing dispensing targets from 1 to 10 treats, in steps of one treat. These tests were performed using two of the described dispensers, which were filled up to the "1 cup" line with Pet Botanics Training Reward treats, which are cylindrical treats that fit through the larger dispenser wheel. Reliability data are available at https://github.com/unl-cchil/canine_treat_dispenser.

The results for this testing are shown in **Tables 5** and **6**. We tracked the mean number of treats dispensed for each dispenser tested (columns Actual One and Two), the number of treats dispensed that were over or under the expected number in each set of tests (columns Errors One

Figure 5: Peripheral interface for the dispenser.

Table 5: Results of the treat dispensing trials.

	Total Treats Dispensed	Correct Number Dispensed	Over-Dispensing	Under-Dispensing
First Dispenser	100	68	31	1
Second Dispenser	100	78	22	0

Table 6: Results of the reliability testing.

Dispense	Actual One	Actual Two	Errors One	Errors Two	Mean Actual	Mean Error
1	1.10	1.00	1	0	1.05	0.50
2	2.40	2.20	4	2	2.30	3.00
3	3.40	3.30	4	3	3.35	3.50
4	4.10	4.00	1	0	4.05	0.50
5	5.20	5.30	2	3	5.25	2.50
6	6.40	6.50	4	5	6.45	4.50
7	7.60	7.10	6	1	7.35	3.50
8	8.50	8.60	4	4	8.55	4.00
9	9.50	9.20	4	2	9.35	3.00
10	10.00	10.20	2	2	10.10	2.00
Mean Error			32.0%	22.0%		27.0%

and Two), the mean number of treats dispensed averaged across the two dispensers (column Mean Actual), and the mean number of failures between the two dispensers (Mean Error).

Each dispensing command has a potential error rate of 32% in the first dispenser and 22% in the second dispenser. This means that over- or under-dispensing occurs roughly every third and fourth event, respectively.

(3) Application

Use Case(s)

The modified *Treat & Train* was designed to be used in canine cognition studies and be controlled by a script to perform the reward dispensing in response to input from the canine subject. This described tools can also reasonably act as a platform for new dispenser development, which can be specifically designed for canine cognition research, such as precise dispensers.

Further, this may be used by dog owners who are training their dogs for particular tasks or who would like to develop games for their dogs to play. Existing commercial dispensers such as PupPod (PupPod 2020) allow for inputs from embedded sensors in toys. But they do not connect to touchscreens, which could provide rich possibilities for games. Our dispenser allows a wider range of potential stimuli to be presented to and responses to be recorded from dogs to trigger a reward.

Since the Raspberry Pi 4 has an on-board Bluetooth modem, this allows for integration of other custom devices such as inertial measurement units (IMUs), which measure its own movement in terms of acceleration in the X, Y, and Z axes, to allow a canine's position and movement to control treat dispensing (Protopopova et al. 2016, Majikes 2017). Additionally, the sound output of the Pi can allow for sound triggers in the operant conditioning protocol (Protopopova et al. 2016, Edwards 2019).

Reuse Potential and Adaptability

With the included Raspberry Pi 4, custom sensors can be easily integrated through Python scripting to trigger the dispensing command. The developed Python class and custom hardware allow for control of other dispensers, such as for bird or rat pellets. Common off-the-shelf dispensers use a single motor to dispense sequential treats, in a similar fashion as the *Treat & Train*; due to this the presented hardware would work on those systems.

(4) Build Details

Availability of Materials and Methods

Building the custom hardware requires the use of a soldering iron, hot air rework gun, reflow oven, precise tweezers, and a vise. All the parts for this part of the build are available from DigiKey and a PCB manufacturer.

The retrofitting operation requires a 3D printed Raspberry Pi mount, a Dremel for exposing ports on the *Treat & Train*, and various M3 screws. Additionally, the extra parts, such as HDMI cables and ports, for the Raspberry Pi 4 are available from Amazon.

Ease of Build

The parts used in the build are readily available from online vendors such as Adafruit and DigiKey. The custom version requires the operation of more advanced tools such as a hot air gun or reflow oven, pick and place or precise tweezers, and solder paste and flux. We estimate the build can be performed in around eight hours with sufficient planning and experience from start to finish for the DIY version.

Operating Software and Peripherals

For the custom hardware, Arduino IDE 1.8.12 was used as the development environment. For the Raspberry Pi 4, the Raspberry Pi OS was used as well as *pyserial* (version 3.4), *PsychoPy* (version 3.2.4), and *pi-ap* (version 1.10.02).

Dependencies

This build requires the use of *pi-ap* (Houlahan 2019), Raspberry Pi OS (Raspberry Pi Foundation 2020), and Arduino IDE (Arduino 2020). The driving software to send commands over the USB port is written in Python 3.7.6 (Python Software Foundation 2019).

Hardware Documentation and File Location

Name: GitHub
Persistent identifier: https://github.com/unl-cchil/canine_treat_dispenser
Licence: Creative Commons CC-BY-SA, 4.0
Publisher: Jeffrey R. Stevens
Date published: 29/07/2020
Software is stored together with the hardware files, in the same repository.
Licence: Creative Commons CC-BY-SA, 4.0
Date published: 29/07/2020

(5) Discussion

Conclusions

This study presented a method by which a common, off-the-shelf, canine treat dispenser can be modified to conduct operant conditioning studies using a computer interface. Using the presented version, the common Arduino platform was used to interface to the onboard photo interrupters and control the dispensing motor, achieving the same functionality as the original dispenser but controlled by a computer. This computer, a Raspberry Pi 4, was integrated to the new hardware to run a Linux distribution and operate the generation of stimulus through *PsychoPy* (Peirce 2007). In adding this central computing platform, all results from the experiments can be centrally stored and easily accessed for future processing and reporting.

By making this dispenser DIY and open-source, researchers may expand on the work and include new trigger methods ranging from sound to ultrasonic distance sensing. By having a single-board computer housed within the dispenser, experiment control software can be generated and run by using the tested and supported *PsychoPy* Python package (Peirce 2007). Additionally, this setup allows for cross-platform development, meaning that any software already generated using the *PsychoPy* package is supported. The inclusion of the Raspberry Pi 4 also makes the entire setup mobile, so experiments can be conducted outside of the laboratory.

The demonstrated computer-controlled dispenser can be readily used and assembled with minimal skill and can act as a jumping off point for automated canine operant conditioning studies. Since automated reward dispenser are difficult to build from scratch, our proposed modifications of the *Treat & Train* would allow for researchers to integrate an automated dispenser into their studies (Hardin et al. 2015, Protopopova et al. 2016, Majikes 2017, Wallis et al. 2017, Robinson et al. 2020). Additionally, since the use of positive reinforcement training has been shown to indicate an increased level of obedience, our dispenser can be used in at-home situations for canine training exercises (Hiby et al. 2004).

The average error rate between the two dispensers is 27%, which is an error every fourth dispensing command. These errors often happen when two treats stack in the

output hole and, when it dispenses, both fall through, leading to dispensing one extra treat. If, for instance, a circular or thicker treat were used, this issue could be mitigated. Alternatively, modifying the dispensing wheel to limit the number of treats grabbed could also help. As for situations where the treat would jam the movement of the wheel, the force of the turning wheel would cut the treat in half, as the treats used in testing were pliable enough to allow for this. This resolved the issue when it presented itself, but this may not be the case for all treat types.

For studies in which precise treat dispensing is not required, our work allows researchers to relatively inexpensively adapt an off-the-shelf product into a treat dispenser that can be controlled either by a personal computer or a Raspberry Pi, thus facilitating operant conditioning studies in dogs.

Future Work

Our results indicate that the developed dispenser is most likely not suitable for precise dispensing experiments, such as numerical preference experiments. Due to this, our focus will be to prove out the developed dispenser for canine cognition studies and stress test the current design. Additionally, to make the developed printed circuit board more easily integrated, it would be beneficial to adapt it into a Raspberry Pi HAT (Hardware Attached on Top), in which the board is directly attached to the Raspberry Pi.

Additional File

The additional file for this article can be found as follows:

- **Data file.** CSV file of reliability testing data. DOI: https://doi.org/10.5334/joh.27.s1

Acknowledgements

We thank Harold Tay for comments on this manuscript.

Competing Interests

The authors have no competing interests to declare.

Author Contributions

Design, assembly, testing, paper writing and editing, Walker Arce.

Oversight, funding, paper writing and editing, Jeffrey R. Stevens.

References

Arduino. (2020). Arduino. Available at: https://www.arduino.cc/.

Baker, J. M., Morath, J., Rodzon, K. S. and **Jordan, K. E.** (2012). A shared system of representation governing quantity discrimination in canids. *Frontiers in Psychology*, 3: 387. DOI: https://doi.org/10.3389/fpsyg.2012.00387

Edwards, T. L. (2019). Automated canine scent-detection apparatus: technical description and training outcomes. *Chemical Senses*, 44(7): 449–455. DOI: https://doi.org/10.1093/chemse/bjz039

Hardin, D. S., Anderson, W. and **Cattet, J.** (2015). Dogs can be successfully trained to alert to hypoglycemia samples from patients with type 1 diabetes. *Diabetes Therapy*, 6(4): 509–517. DOI: https://doi.org/10.1007/s13300-015-0135-x

Hiby, E. F., Rooney, N. J. and **Bradshaw, J. W. S.** (2004). Dog training methods: their use, effectiveness and interaction with behaviour and welfare. *Animal Welfare*, 13(1): 63–70.

Houlahan, T. (2019). pi-ap. Available at: https://github.com/f1linux/pi-ap.

Majikes, J. J. (2017). Computational and design techniques for a semi-autonomous computerized dog-training system with timing and accuracy performance comparable to a professional dog trainer. (Unpublished doctoral dissertation, North Carolina State University).

McKinley, S. and **Young, R. J.** (2003). The efficacy of the model–rival method when compared with operant conditioning for training domestic dogs to perform a retrieval–selection task. *Applied Animal Behaviour Science*, 81(4): 357–365. DOI: https://doi.org/10.1016/S0168-1591(02)00277-0

Miletto Petrazzini, M. E., Mantese, F., and **Prato-Previde, E.** (2020). Food quantity discrimination in puppies (*Canis lupus familiaris*). *Animal Cognition*, 23: 703–710. DOI: https://doi.org/10.1007/s10071-020-01378-z

Microsoft. (2019). uf2-samdx1. Available at: https://github.com/microsoft/uf2-samdx1.

Peirce, J. W. (2007). PsychoPy—Psychophysics software in Python. *Journal of Neuroscience Methods*, 162(1–2): 8–13. DOI: https://doi.org/10.3758/s13428-018-01193-y

PetSafe. (2020). Treat & Train™ Remote Reward Dog Trainer. Available at: https://intl.petsafe.net/en-gb/support/treat-train-remote-reward-dog-trainer.

Protopopova, A., Kisten, D. and **Wynne, C.** (2016). Evaluating a humane alternative to the bark collar: Automated differential reinforcement of not barking in a home-alone setting. *Journal of Applied Behavior Analysis*, 49(4): 735–744. DOI: https://doi.org/10.1002/jaba.334

PudPod. (2020). Puzzle Toy for Dogs. PupPod. Available at: https://puppod.com/.

Python Software Foundation. (2019). Python Release Python 3.7.6. Python.org. Available at: https://www.python.org/downloads/release/python-376/.

Raspberry Pi Foundation. (2020). Teach, Learn, and Make with Raspberry Pi – Raspberry Pi. Available at: https://www.raspberrypi.org/.

Reisner, I. (2016). The learning dog: A discussion of training methods. In J. Serpell (Ed.), *The Domestic Dog: Its Evolution, Behavior and Interactions with People*, 2: 211–226. Cambridge University Press.

Robinson, C., Brulé, E., Jackson, J., Torjussen, A., Kybett, J. and **Appshaw, T.** (2020). Tricks and treats: Designing technology to support mobility assistance dogs. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA on 25–30 April 2020, pp. 1–14.

Tatham, S. (2020). Download PuTTY: latest release (0.74). Available at: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html.

Wallis, L. J., Range, F., Kubinyi, E., Chapagain, D., Serra, J. and **Huber, L.** (2017). Utilising dog-computer interactions to provide mental stimulation in dogs especially during ageing. In: *Proceedings of the Fourth International Conference on Animal-Computer Interaction*, New York, NY, USA on November 2017, pp. 1–12. DOI: https://doi.org/10.1145/3313831.3376188

Ward, C. and **Smuts, B. B.** (2007). Quantity-based judgments in the domestic dog (*Canis lupus familiaris*). *Animal Cognition*, 10(1): 71–80. DOI: https://doi.org/10.1007/s10071-006-0042-7

Wayland, M. and **Landgraf, M.** (2018). A Cartesian coordinate robot for dispensing fruit fly food. *Journal of Open Hardware*, 2(1): 3. DOI: https://doi.org/10.5334/joh.9

Yin, S., Fernandez, E. J., Pagan, S., Richardson, S. L. and **Snyder, G.** (2008). Efficacy of a remote-controlled, positive-reinforcement, dog-training system for modifying problem behaviors exhibited when people arrive at the door. *Applied Animal Behaviour Science*, 113(1–3): 123–138. DOI: https://doi.org/10.1016/j.applanim.2007.11.001

How to cite this article: Arce, W and Stevens, JR. 2020. Developing a Computer-Controlled Treat Dispenser for Canine Operant Conditioning. *Journal of Open Hardware*, 4(1): 6, pp. 1–10. DOI: https://doi.org/10.5334/joh.27

Published: 28 October 2020

]u[*Journal of Open Hardware* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS