Video Article

Feeding Experimentation Device (FED): Construction and Validation of an Open-source Device for Measuring Food Intake in Rodents

Katrina P. Nguyen[1], Mohamed A. Ali[1], Timothy J. O'Neal[1], Ilona Szczot[1], Julia A. Licholai[1,2], Alexxai V. Kravitz[1,3]

[1]National Insttitute of Diabetes and Digestive and Kidney Diseases

[2]National Center for Complementary and Integrative Health

[3]National Institute on Drug Abuse, National Institutes of Health

Correspondence to: Alexxai V. Kravitz at alexxai.kravitz@nih.gov

URL: https://www.jove.com/video/55098
DOI: doi:10.3791/55098

Keywords: Behavior, Issue 120, automation, electronics, feeding, food intake, home cage, microcontroller, open-source

Date Published: 2/21/2017

Citation: Nguyen, K.P., Ali, M.A., O'Neal, T.J., Szczot, I., Licholai, J.A., Kravitz, A.V. Feeding Experimentation Device (FED): Construction and Validation of an Open-source Device for Measuring Food Intake in Rodents. *J. Vis. Exp.* (120), e55098, doi:10.3791/55098 (2017).

Abstract

Food intake measurements are essential for many research studies. Here, we provide a detailed description of a novel solution for measuring food intake in mice: the Feeding Experimentation Device (FED). FED is an open-source system that was designed to facilitate flexibility in food intake studies. Due to its compact and battery powered design, FED can be placed within standard home cages or other experimental equipment. Food intake measurements can also be synchronized with other equipment in real-time via FED's transistor-transistor logic (TTL) digital output, or in post-acquisition processing as FED timestamps every event with a real-time clock. When in use, a food pellet sits within FED's food well where it is monitored via an infrared beam. When the pellet is removed by the mouse, FED logs the timestamp onto its internal secure digital (SD) card and dispenses another pellet. FED can run for up to 5 days before it is necessary to charge the battery and refill the pellet hopper, minimizing human interference in data collection. Assembly of FED requires minimal engineering background, and off-the-shelf materials and electronics were prioritized in its construction. We also provide scripts for analysis of food intake and meal patterns. Finally, FED is open-source and all design and construction files are online, to facilitate modifications and improvements by other researchers.

Video Link

The video component of this article can be found at https://www.jove.com/video/55098/

Introduction

With the rise of global obesity over the latter part of the 20th century, there is renewed attention on the mechanisms underlying feeding[1,2,3,4]. Typically, food intake is weighed manually[5], or with commercially-available feeding systems. Commercial systems are excellent, but provide limited flexibility in modifying their designs or code. Here, we describe the Feeding Experimentation Device (FED): an open-source feeding system for measuring food intake with fine temporal resolution and minimal human interference[6]. FED is battery powered and fully contained within a 3D printed case that can fit inside of standard colony rack caging or other scientific equipment.

In its steady state, FED operates in a low-power mode with a food pellet resting in its food well. The presence of the pellet is monitored via an infrared beam. When a mouse removes a pellet, a photointerrupter sensor sends a signal to the microcontroller and the time-stamp is logged on the onboard secure digital (SD) card. Concurrently, a transistor-transistor logic (TTL) output provides a real-time output of pellet retrieval. Immediately following this event, the motor rotates to dispense another pellet, and the system returns to its low power mode. Due to its open-source nature, FED can be modified and improved to fit specific research needs. For example, the code can be easily altered to limit feeding to specific times of the day, or to stop dispensing when a number of pellets has been reached, without requiring human interference.

Here, we outline the step-by-step instructions for the construction, validation, and use of FED for measuring food intake in mice. We provide a list of all components to construct a system. Importantly, no prior experience in electronics is needed to construct FED.

Protocol

NOTE: This protocol is written for components specifically named in the Table of Materials. While similar functionality can be achieved using other hardware, FED was programmed for the Arduino Pro microcontroller (henceforth termed: microcontroller) and listed accessories. Other microcontrollers may work equally well, but will require the user to modify the code to support them. Offline data analysis was coded using the Python programming language.

1. Preparation and Software Installation

1. Procure electronic components needed to construct FED (see **Table 1** andFed Github BoM.xlxs at: https://github.com/KravitzLab/FED/tree/master/doc).
   NOTE: Alternative suppliers may be used for many parts on this table, provided they have equivalent specifications.
2. Print all 3D designed components (**Figure 1**, available at: https://github.com/KravitzLab/FED/wiki/3D-Printed-Components). 3D printers with a 200 micron resolution should be capable of printing FED.
3. Download and install the Integrated Development Environment (IDE) platform to program the microcontroller.
4. Download and install additional libraries to enable functionality of motor shield and data logger (available at: https://github.com/KravitzLab/fed/tree/master/fed-arduino).
5. Procure tools needed for assembly (*e.g.* a soldering iron, heat gun, solder, wire strippers, needle-nosed pliers, and both flat-head and screwdrivers).

2. Soldering Electrical Components

NOTE: Use heat shrink tubing to protect all soldered joints. Prior to soldering connections, slide a piece of shrink wrap tubing (~2 cm) tubing around one of the wires. After soldering the connection, center the tubing on the connection point and use a heat gun to heat shrink the tubing.

1. **Preparing connectors (Figure 2A)**
   1. Prepare four 2-pin JST connector pairs and label both male and female sides "A", "B", "C", and "D", respectively. Remove the red wire from both sides of connector pair "D".
   2. Prepare one 3-pin JST connector pair and label both male and female sides "E".

2. **Microcontroller and stackable shields (Figure 2B)**
   1. Solder female stackable headers with sockets on the top side of the microcontroller. Clip protruding wire from headers on the bottom of the microcontroller.
   2. Solder female stackable headers with sockets on the top side of the SD data logging shield. Leave protruding wires at bottom of the shield.
   3. Solder male headers onto the motor shield with pins protruding from the bottom.
   4. Place a coin cell battery into the slot of SD shield to provide power to the real-time clock module.

3. **External power button (Figure 2C)**
   NOTE: A latching metal pushbutton has five connections: power, ground, normally closed (NC1), normally open (NO1), and common (C1).
   1. Solder the 2-pin male connector "A" to C1 (use red wire) and ground (use black wire). Heat-shrink all connections.
   2. Solder the 2-pin male connector "B" to + (use red wire) and NO1 (use black wire). Heat-shrink all connections.

4. **Photointerrupter (Figure 2D)**
   1. Solder photointerrupter (the black part) to breakout board.
   2. Solder a 4.7K resistor to the front of the breakout board.
   3. Solder the male 3-pin connector "E" to the back of the breakout board: red wire to PWR, green wire to GND, and white wire to SGL.
   4. Trim loose wires on back of photointerrupter break out board.

5. **Boost board (Figure 2E)**
   1. Solder the 2-pin female connector "A" to 5V and Ground pins on the boost board.
   2. Solder the black wire from male connector "D" to the additional GND pin on the boost board.

6. **BNC output cable (optional: Figure 2F)**
   1. Solder the 2-pin connector "C" to the terminals of a BNC cable (red wire to central pin, black wire to outside pin).
      NOTE: For assembly, the 2-pin connector must fit through the nut on the BNC plug. We use a smaller connector, or shave down the JST connector with a razor blade to make it fit.

7. **Motor Shield (Figure 2G)**
   1. Twist the red and black wires of the female connector "B" together and solder to V_{in}.
   2. Solder the black wire of the female connector "C" to the ground pin next to ARef, and the red wire of this connector to pin 3.
   3. Solder the black wire of the female connector "D" to the ground pin next to V_{in}.
   4. Solder the green wire of the female connector "E" to the ground pin next to 5V, the red wire of this connector to 5V, and the white wire of this connector to pin 2.

3. Software Upload

1. Connect the FTDI breakout board to the programming pins of the microcontroller, and then connect FTDI breakout board to computer via micro USB cable.
2. Open the IDE (integrated development environment) program.
3. Select the correct microcontroller board for software upload through Tools > Board dropdown menu.
4. Select ATMega 328 (5V, 16mHz) through the Tools > Processor menu.
5. Select the port that the microcontroller is connected to through Tools > Port > COM# (will vary depending on which port is currently in use).
6. Click the "upload" button to upload the FED sketch to the board (available at: https://github.com/KravitzLab/fed/tree/master/fed-arduino).

4. Hardware Assembly

1. **Stepper motor and motor shield (Figures 1C and 3A and 3B)**
    1. Secure the 5V stepper motor onto the 3D printed motor mount with two #6 x ¼" sheetmetal screws (**Figures 1C** and **3A**).
    2. Insert rotating disk into motor mount and push down to securely attach to stepper motor shaft (**Figure 3B**).
    3. Twist on 3D printed food silo onto the motor mount making sure the pellet leveler arm is over the hole in the motor mount.
    4. Twist on connected pieces from above (steps 4.1.1 - 4.1.3) to the top of the printed base, with the stepper motor positioned towards the back of the base and the hole positioned in the front.
    5. Cut the 5-pin connector from the stepper motor wires and strip ~2 mm from the end of each wire.
    6. Connect wires from stepper motor to the terminal block connectors on the motor shield: red to ground, orange and pink to one motor port (*e.g.*, M1), and blue and yellow to the other motor port (*e.g.*, M2).

2. **External power button**
    1. Remove the nut from the power button and insert the power button into the hole in the right side of the base. Secure button in place with hex nut.

3. **Photointerrupter (Figure 3C)**
    1. Place the photointerrupter into its 3D printed housing.
    NOTE: use a heat gun to heat up the housing if the photointerrupter does not seat all the way in.
    2. String the 3-pin male connector "E" from the photointerrupter (PWR, GND, and SGL) through the front middle hole of the 3D printed base.
    3. Secure the housing into the FED base with two 1" nylon screws and corresponding nuts.

4. **BNC output cable (optional)**
    1. Insert BNC connector into hole on the left side of the FED base. Secure in place with nut.
    2. If BNC connector is not used, plug hole with 3D printed plug.

5. Battery and boost board (**Figure 3D**)
    1. Connect 3.7 V battery pack to the DC/DC boost converter module via the JST 2-pin connection. The blue LED on the Boost board will illuminate if battery is charged.

6. **Mounting boards inside of housing (Figure 3E)**
    1. Mount microcontroller inside of the base with FTDI connections facing the power switch, using #4 x ¼" steel sheet metal screws.
    2. Stack motor shield and data logging shield on top of the microcontroler.
    3. Screw the Boost board into the case using #2 x ¼" steel sheet metal screws. Mount Boost with the micro-SD slot pointing down. FED can be charged through this port without opening the case.
    4. Connect the five connectors, "A" male to "A" female, "B" male to "B" female, etc.
    5. Place the battery inside the 3D printed base and close by sliding the back cover.
    6. Slide on the 3D printed face plate.

5. Validation and Data Acquisition

NOTE: Prior to powering on a FED system, ensure an SD card is inserted on the SD shield, otherwise FED will not dispense pellets. Additionally, ensure power jumper on the motor shield (just above the power block) is in place.

1. **Power on FED system with the power pushbutton and test device functionality.**
    1. Fill food silo with 20 mg food pellets before powering on.
    NOTE: The power switch should light up, as should LEDs on the microcontroller, SD shield, and motor shield. If there is no pellet in the well, one should dispense.
    2. Manually remove 5 - 10 pellets from the food well and confirm that replacement pellets are dispensed.

2. Remove the SD card and verify that data was logged properly. Data should be acquired in a comma-separated value (.CSV) that is named according to the variable FILENAME in the code.
3. Place FED unit inside experimental setting, power on, and ensure that a pellet is dispensed into the food well.
4. Over the course of data acquisition, check FED daily to verify that it is working properly by confirming that the LED light on the power switch is on (this indicates that the battery has enough charge) and a pellet is sitting in the food well (indicates that there are no problems with pellet dispensing).
5. After data acquisition, retrieve SD card and access .csv file.
NOTE: Analysis scripts for meals and patterns of feeding are available at: https://github.com/KravitzLab/fed.

Representative Results

Validation tests involving the use of animals were reviewed and approved by the Animal Care and Use Committee at the National Institute of Diabetes and Digestive and Kidney Diseases. To demonstrate the use of FED for measuring home cage feeding, adult female C57BL/6 mice (*n* = 4) were individually housed with *ad libitum* access to water and standard laboratory chow under a 12/12 h light/dark cycle (lights on at 05:00). After a one week habituation period, the food hopper was removed and replaced with a FED for five days of validation testing. At the end of the testing period, FEDs were removed from cages and feeding data from each SD card was analyzed using custom, freely-available scripts and a spreadsheet program. As shown in **Figure 4A**, pellet retrieval for individual mice (top panel; rasters indicate single pellet retrieval events) and average pellet retrieval across all mice in 30 min bins (bottom panel; line indicates mean ± SEM) show continuous feeding across the validation testing period, with clearly visualized circadian rhythmicity. To quantify the accuracy of FED's data logging capacity, each FED system was given 1000 pellets for the validation testing period and the remaining pellets were manually counted and compared to logged data on the SD card. FED logged 95.35 ± 1.25% of pellets that were dispensed to the SD card (**Figure 4B**).

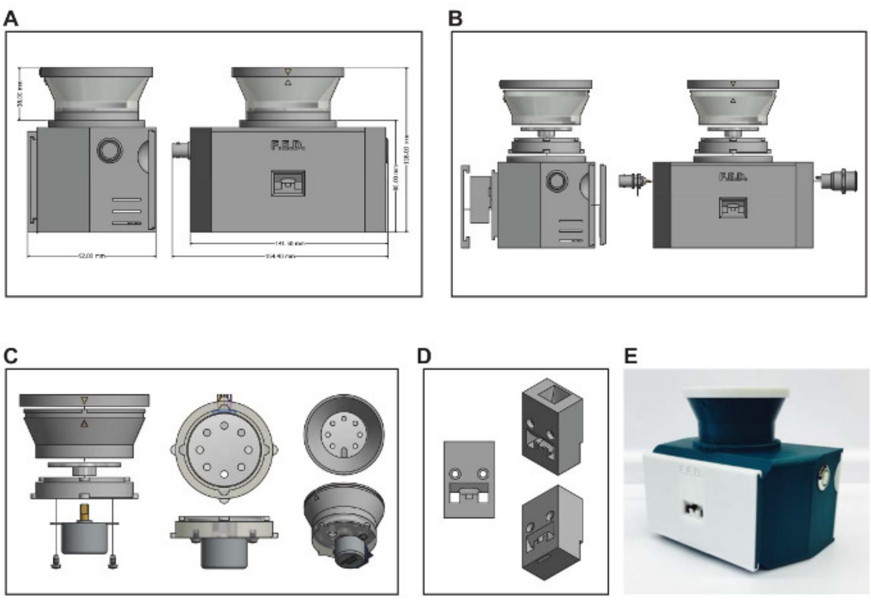

Figure 1: 3D Designed FED Components. (**A**) Measurements of assembled FED. (**B**) Exploded view of 3D printed components in an assembled FED. (**C**) Exploded view of complete pellet hopper assembly (left), assembly of stepper motor to rotating disk (middle), and assembly of food silo to rotating disk (right). (**D**) Design of food well with arms to secure the photointerrupter for pellet detection. (**E**) Photograph of fully assembled FED. Please click here to view a larger version of this figure.

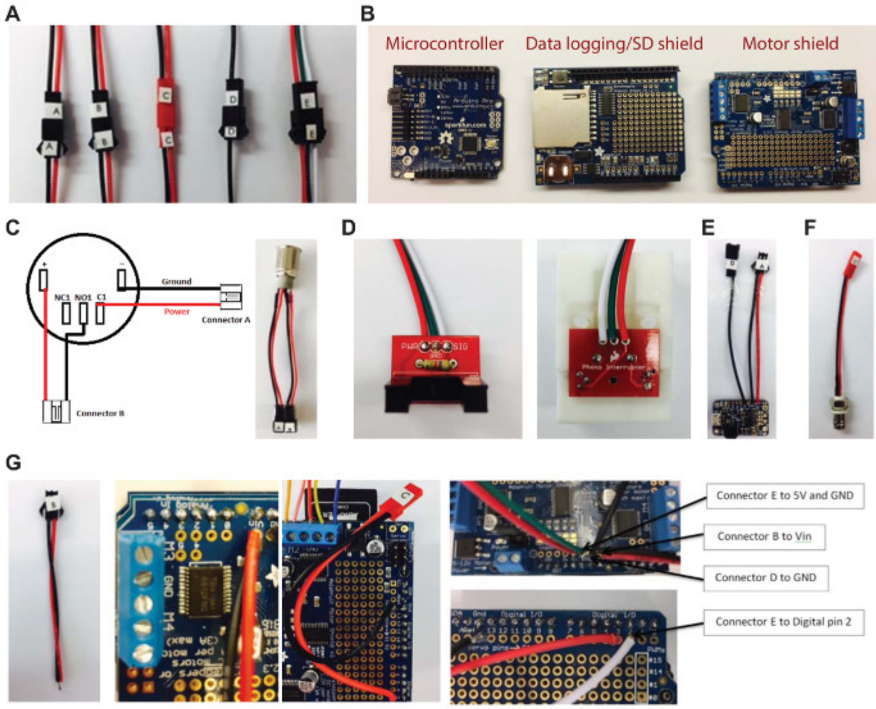

Figure 2: Wiring of FED Electrical Components. Wiring and assembly of (**A**) Connectors (**B**) Microcontroller, Data logging/SD shield, Motor shield, (**C**) Power button, (**D**) Photointerrupter, (**E**) Boost board, (**F**) BNC output cable, (**G**) Wiring of motor shield. Please click here to view a larger version of this figure.

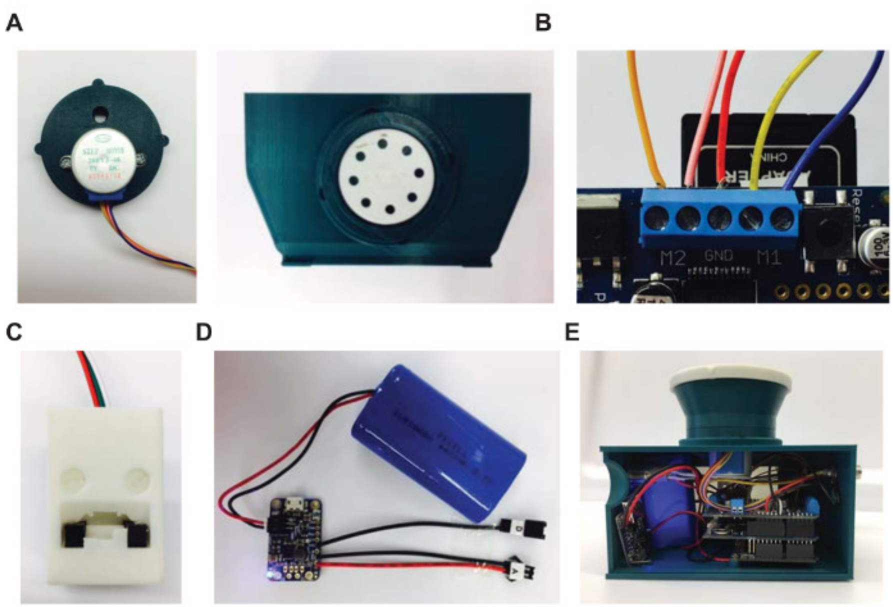

Figure 3: Assembly of FED Hardware. (**A**) Mounting stepper motor (left) into motor mount and attaching pellet disk (right). (**B**) Wiring stepper motor to motor shield. (**C**) Inserting photointerrupter into 3D printed housing. (**D**) Attaching battery to Boost board. (**E**) Boards assembled inside FED housing. Please click here to view a larger version of this figure.

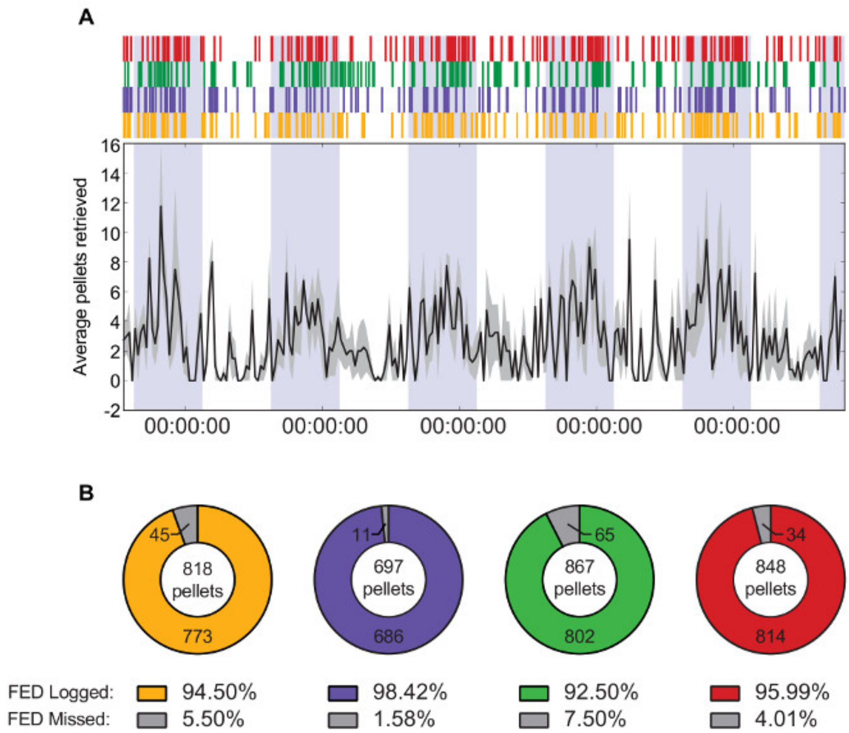

Figure 4: Representative Food Intake Data Collected *via* FED. (**A**) Food intake data collected *via* FED and visualized using open-source analysis scripts. Top panel: individual rasters indicate single pellet retrieval events, with rows containing individual mice. Bottom panel: line indicates mean ± SEM of food intake across mice (n = 4). (**B**) Accuracy of individual FED units in logging dispensed pellets. Colors correspond with rasters in Figure 3A. Please click here to view a larger version of this figure.

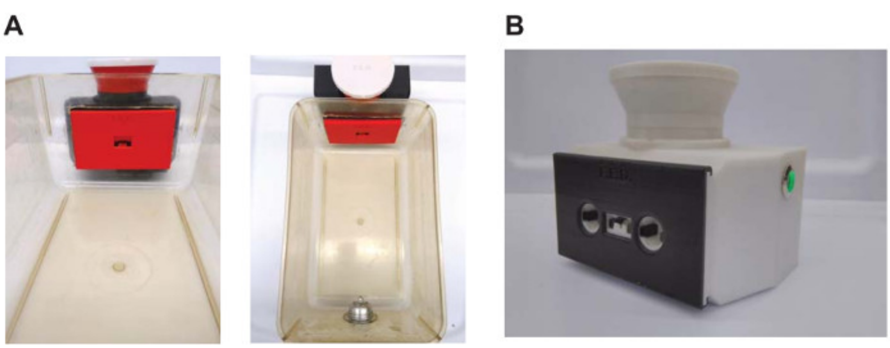

Figure 5: Modified uses for FED. (**A**) Externally mounted FED reduces floor space in rodent cages. (**B**) Changing the front-face configuration allows for the addition of two nose poke ports for operant training. Please click here to view a larger version of this figure.

Supplemental File 1: Hardware Schematic. Graphic schematic showing the electronic connections of FED. Please click here to download this file.

Supplemental File 2: Sample Output Code. Sample .CSV file showing output code from FED. Please click here to download this file.

Supplemental Files 3-6: Analysis Scripts. Four analysis scripts are provided for analyzing FED data: eating_rate.py, meals.py, meal_bars.py, and plotmice.py. Input parameters and description is provided in the commented sections at the top of each script. Please click here to download this file.

Discussion

The Feeding Experimentation Device (FED) is a flexible food intake monitoring system. Here, we describe detailed instructions on fabricating and troubleshooting the device, including the assembly of 3D printed hardware, soldering of electrical components, and uploading of sketches onto the microcontrollers. Though it is important to follow all steps outlined in the protocol carefully, there are critical steps that deserve extra attention in each section to ensure a successful end product. The 3D printed rotating disk should fit snugly onto the stepper motor shaft and be able to rotate with minimum resistance from neighboring parts. When soldering pin connections, be sure that the solder joint is secure without having excess solder. Ensure that all additional libraries installed must be located in the correct directory before uploading the script onto the microcontroller. The process of assembling FED involves tasks that are easily achievable, even with no prior experience in 3D printing or electronics.

There are few limitations with FED. First, each FED uses an internal SD card to log and store data. SD cards can be a cumbersome means to track and store data from many FEDs. We are exploring wireless data transfer to alleviate this issue, but the current iteration of FED relies on local storage on SD cards. Second, ~ 5% of the time FED dispenses two pellets instead of one. While pellet jams have not occurred in our hands with enough frequency to count, this possibility should be strictly monitored as we cannot rule them out completely. Users should never leave mice with FED as their only food source without checking FED's functionality daily. Despite these limitations, we believe FED's error rate is acceptable for most research studies. That said, in future work we will seek to decrease these errors further. Finally, the tolerance of 3D printing can vary depending on the specific 3D printer and material. As such, the 3D design files we provide may need to be tweaked for printing on other models of 3D printer, or with other materials. As one helpful tip, we have found that a heat gun can be used after printing to soften 3D printed parts for minor adjustments.

A single FED can be assembled for approximately $350. Of this price, we estimate ~$200 for commercial 3D printing out of PLA material, and ~ $150 for purchasing the electronic components. 3D printing costs vary greatly depending on the quality and material. It is possible to reduce this cost considerably if a user prints their own parts. FED could also be printed or machined out of more durable materials such as stainless steel or aluminum, although this would be costlier.

We chose to build FED with a microcontroller on a programmable circuit board with an open-source platform. For this, we chose the open source microcontroller (see **Table of Materials**) as it has a low power draw. However, our code should work on any board that has sufficient input/output pins. Changing microcontroller boards will likely require minor changes to the code. We provide all design files and code online, and our design is open-source to provide researchers with flexibility to modify FED to suit their needs, including the use of other microcontroller boards.

There is a growing trend towards open-source hardware for scientific research. Various groups have made open-source devices for neuroscience research, including OpenControl, a software for video tracking animals during behavioral tasks[7]; ROBucket, an Arduino-based operant chamber for liquid outcome delivery[8]; ELOPTA, a PICmicro-based operant device for pellet outcome delivery[9]; and BEEtag, an image-based tracking system for bumblebees[10]. FED complements these with its unique form factor and low cost. A key aim for FED was to make it open-source. Open-source hardware allows researchers to perform minor tweaks to designs (for example to modify the dimensions), but is particularly powerful when users perform major modifications to expand the functionality of the hardware. For example, it is fairly easy to modify the design of FED for external mounting on a cage, which both protects FED and increases floor space within the cage (**Figure 5A**). Additionally, we designed FED to have room for nose-pokes on each side of the feeding well for operant training, and have successfully implemented these with minor design changes (**Figure 5B**). We will post such developments on our website (https://github.com/KravitzLab/fed), where we also look forward to hosting other such improvements from the research community.

Disclosures

Authors declare no conflict of interests, financial or otherwise.

Acknowledgements

This work was supported by the Intramural Research Program of the National Institutes of Health (NIH), The National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK). We thank the NIH Section on Instrumentation and the NIH Library for assistance with 3D printing.

References

1. Ellacott, K. L., Morton, G. J., Woods, S. C., Tso, P., & Schwartz, M. W. Assessment of feeding behavior in laboratory mice. *Cell Metab.* **12** (1), 10-17 (2010).
2. Betley, J. N. *et al.* Neurons for hunger and thirst transmit a negative-valence teaching signal. *Nature.* **521** (7551), 180-185 (2015).
3. van den Heuvel, J. K. *et al.* Neuropeptide Y activity in the nucleus accumbens modulates feeding behavior and neuronal activity. *Biol Psychiatry.* **77** (7), 633-641 (2015).
4. Cone, J. J., Roitman, J. D., & Roitman, M. F. Ghrelin regulates phasic dopamine and nucleus accumbens signaling evoked by food-predictive stimuli. *J Neurochem.* **133** (6), 844-856 (2015).
5. Ulman, E. A., Compton, D., Kochanek, J. Measuring food and water intake in rats and mice. *ALN Mag.* 17-20 (2008).
6. Nguyen, K. P., O'Neal, T. J., Bolonduro, O. A., White, E., & Kravitz, A. V. Feeding Experimentation Device (FED): A flexible open-source device for measuring feeding behavior. *J Neurosci Methods.* **267** 108-114 (2016).
7. Aguiar, P., Mendonca, L., & Galhardo, V. OpenControl: a free opensource software for video tracking and automated control of behavioral mazes. *J Neurosci Methods.* **166** (1), 66-72 (2007).
8. Devarakonda, K., Nguyen, K. P., & Kravitz, A. V. ROBucket: A low cost operant chamber based on the Arduino microcontroller. *Behav Res Methods.* **48** (2), 503-509 (2016).
9. Hoffman, A. M., Song, J., & Tuttle, E. M. ELOPTA: a novel microcontroller-based operant device. *Behav Res Methods.* **39** (4), 776-782 (2007).
10. Crall, J. D., Gravish, N., Mountcastle, A. M., & Combes, S. A. BEEtag: A Low-Cost, Image-Based Tracking System for the Study of Animal Behavior and Locomotion. *PLoS One.* **10** (9), e0136487 (2015).

