

Video Article

A Precise and Autonomous System for the Detection of Insect Emergence Patterns

Meghan M. Bennett^{1,2}, Joseph P. Rinehart³, George D. Yocum³, Ian Yocum³

¹Department of Biological Sciences, North Dakota State University

²School of Life Sciences, Arizona State University

³Red River Valley Agricultural Research Center, USDA-ARS

Correspondence to: Meghan M. Bennett at mbenne15@asu.edu

URL: <https://www.jove.com/video/58362>

DOI: [doi:10.3791/58362](https://doi.org/10.3791/58362)

Keywords: Emergence, rhythms, falling-ball, Megachile, microcontroller, Arduino

Date Published: 12/5/2018

Citation: Bennett, M.M., Rinehart, J.P., Yocum, G.D., Yocum, I. A Precise and Autonomous System for the Detection of Insect Emergence Patterns. *J. Vis. Exp.* (), e58362, doi:10.3791/58362 (2018).

Abstract

Existing systems to measure insect emergence patterns have limitations; they are only partially automated and are limited in the maximum number of emerging insects they can detect. In order to obtain precise measurement of insect emergence, it is necessary for systems to be semi-automated and able to measure large numbers of emerging insects. We addressed these issues by designing and building a system that is automated and can measure emergence of up to 1200 insects. We modified the existing "falling-ball" system using Arduino microcontrollers to automate data collection and expand the sample size through multiple data channels. Multiple data channels enable the user to not only increase their sample size, but also allows for multiple treatments to be run simultaneously in a single experiment. Furthermore, we created an R script to automatically visualize the data as a bubble plot, while also calculating the median day and time of emergence. The current system was designed using 3D printing so the user can modify the system to be adjusted for different species of insects. The goal of this protocol is to investigate important questions in chronobiology and stress physiology, using this precise and automated system to measure insect emergence patterns.

Video Link

The video component of this article can be found at <https://www.jove.com/video/58362/>

Introduction

Precisely measuring the timing of terrestrial insect emergence in experimental settings is notoriously difficult and requires some degree of automation. Several mechanisms have been designed in the past, incorporating either a "falling-ball" principle, using falling balls and sensors, or a "bang-box" using a funnel-type system^{1,2,3}. There are two limitations with existing designs: 1) data collection is only partially automated, and 2) sample size or the number of emergent insects that can be detected is limited. These problems decrease the precision of data collection, which is important to studying timing of eclosion and/or emergence patterns. We addressed these problems by designing a system that is automated and not limited by sample size, enabling the user to better visualize emergence rhythms in response to environmental cues.

Our system is an improvement to the falling-ball principle, the most recent version of which used infrared sensors to detect insect emergence in six-minute increments². Our system still uses infrared sensors, but also incorporates an Arduino microcontroller to record the date and time of each emergence event to the nearest second. Data is automatically stored to a secure digital (SD) card, which can be exported as a comma-delimited file for analysis. The analysis is automated by using a custom R script, which will graph the data as a bubble plot and identify the median time and day of emergence.

Multiple channels enable the user more flexibility in data acquisition. For example, our multiple channel design not only minimizes the impact of a "clogged" sensor, but also can be used to increase sample size. Furthermore, multiple channels allow the user to designate treatments to specific channels so they can run simultaneously in an experiment. Using all six channels allows for approximately 1200 emerging bees to be recorded in a single experiment. To our knowledge, this is the largest sample size of any current system measuring insect emergence and has allowed us to observe fine-scale emergence patterns in response to environmental cues. Finally, our system benefits from the fact that the majority of the parts are 3D printed. This creates precisely sized components, which reduces the likelihood of errors occurring (such as detector clogging) during operations. It also allows for customization for other research systems.

The goal of this protocol is to custom build a precise and automated system to measure insect emergence, to investigate questions in chronobiology and stress physiology. This system has been, and will continue to be, critical in investigating unanswered questions related to insect emergence patterns in response to environmental cues. Here we describe its assembly and use for emergence detection of the alfalfa leaf-cutting bee, *Megachile rotundata* in lab-based experimental settings. The system is automated using a programmable microcontroller and customizable using 3D printed parts. Printed racks hold in place tubes containing bee nest cells following a metal BB. Upon emergence, the

metal BB is freed from the rack, passing through an infrared sensor recording the date and time of emergence to an SD card. The current design is optimized for *M. rotundata*, but with minor adjustments could be adapted for other species of insects.

Protocol

1. System Construction

- Using PLA filament, print the following number of parts for each channel being constructed: 1 collector manifold (collector_manifold.stl), 1 end cap (end_cap.stl), 6 platform supports (platform_support.stl), 4 tube rack base plates (base_plate.stl), and 4 tube rack face plates (face_plate.stl). Make sure printer bed is large enough to print an item before printing. All *.stl files are available in Supplemental data.
- With 3 platform supports and a 33 x 30 cm piece of corrugated plastic, use hot glue to assemble 2 tube rack platforms per channel being constructed, as shown in **Figure 2**. The corrugated plastic can be scored on one side at each corner to allow for bending.
- Install electronics into the collector manifold.
 - Solder a 120 Ω resistor to the anode (longer leg) of both the infrared emitter and the infrared detector, and a ~5 cm length of 22 GA wire to both cathodes. Use different colors of wires to avoid confusion in later steps.
 - Carefully insert the detector into one socket of the collector manifold (highlighted in blue in **Figure 3**) and the emitter into the second socket (highlighted in red). Both components should fit snugly.
 - Feed the detector wires through the cabling channel (highlighted in yellow in **Figure 3**) and pull all four wires through the access hole (highlighted in green). Ensure no bare wires are touching, using hot glue to secure them in place.
 - Solder all four wires to a RJ45 (Ethernet) jack, using the back row of pins. Both anodes should be soldered to the left-most pin, the cathode of the emitter to the right-most pin, and the cathode of the detector to either of the center pins (**Figure 4**).
 - Secure the RJ45 jack over the collector manifold access hole (highlighted in green in **Figure 3**) with hot glue, ensuring no bare wires are touching inside the manifold.
- Construct falling ball collector (1 per channel being constructed) as shown in **Figure 5**
 - With one wired collector manifold, one end cap, and a 24 x 30 cm section of corrugated plastic, use hot glue to connect the base of the unit (red, green, and light grey components of **Figure 5**).
- Use an 8 x 27 cm section of corrugated plastic to add a falling ball ramp to the collector (dark grey component of **Figure 5**). The end cap and collector manifold designs include ledges to ensure correct placement. Check for a smooth transition from the ramp to the collector to avoid jams during use.
- Construct the central processor for the system (as detailed in **Figure 6**).
 - Print a custom printed circuit board for system construction. All files required for PCB board printing are available in supplemental data.
 - Solder female headers on the through-holes labeled for the following installations: Arduino Nano, temp, clock, SD module, and liquid crystal display (LCD) screen (unlabeled 2 x 5 through-hole area in the upper left corner of the PCB board).
 - Snap in and solder six RJ45 jacks along the lower edge of the PCB board.
 - Solder six 470k ohm pull-down resistors into the through-hole sites located just above the RJ45 jacks.
 - Install the Arduino Nano, DHT-temperature and humidity sensor, clock, and SD module onto the PCB board. DHT-temperature and humidity sensor should be tested before use in experiments to ensure accuracy.
 - Connect a 10-conductor ribbon wire to the LCD screen connector of the PCB board. Solder the other end of the ribbon wire to the LCD screen so that the screen pins correspond to the Arduino pins, as noted in **Figure 4**. Further details on LCD wiring are available at <https://Learn.adafruit.com/character-lcds/wiring-a-character-lcd>.
- System programming
 - Download and install the latest version of the Arduino IDE for the correct operating system from www.arduino.cc.
 - On first use, install Arduino libraries for the real time clock (github.com/adafruit/RTClib) and the temperature/humidity sensor (github.com/adafruit/DHT-sensor-library). Set the clock to the current local time by using the ds1307 script included with the library.
 - Upload the system Arduino script, available in supplemental data.

2. System Use

- Assemble the system as shown in **Figure 7**. For each channel being used, one falling-ball collector (assembled in step 1.4) should be flanked on either side by a rack platform (assembled in step 5.1). Use packaging tape to hold together pieces and to create a smooth rounded edge on the rack platform.
- Configure unused channels to avoid false positive signals. Since the system relies on a low signal to detect an event (infrared detector not receiving a signal from the infrared emitter), unused channels must be configured appropriately to avoid false positive signals. This may be accomplished by one of two ways.
 - Deactivate unused channels in the software by commenting out the loops corresponding to the unused channels. In Arduino IDE, this can be accomplished by adding `/*` prior to the unneeded loops and `*/` at their end.
 - Deactivate unused channels through a simple hardware accommodation. Simply solder together wires #6 and #8 (usually the solid brown and solid green wires of a commercially available cat 6 cable) and insert into the empty RJ45 jack on the central processor.
- Load and place tube racks immediately prior to running an experiment.
 - Ensure that all holes contain a 0.5 mL microcentrifuge tube with the cap removed and that the tubes fit snugly.
 - Fill each tube with one insect brood cell, pupal case or cocoon, one airsoft pellet, and finally one metal BB. Make sure the flat edge side (cap) of the brood cell is facing toward the airsoft pellet and metal BB. Affix the tube rack faceplate, with the rounded edge towards the bottom of the rack, using 1/4 inch nylon screws.

3. Place tube racks on the rack platform, with the opening facing towards the falling-ball collector. Racks should be placed at the very edge of the platform so that a metal BB can fall freely into the collector without bouncing against another portion of the structure (**Figure 7**). When placing the rack, start with the opening facing upwards and then gently rotate into place to ensure metal BBs are not released. The racks are designed so that the tubes will slant slightly backwards when properly placed, reducing the chance of accidental release of the metal BBs.
4. Insert a SD card into the adapter and then start the central processor by plugging a micro-USB connector into the Arduino, and the other end into any appropriate USB adapter. The LCD screen will display numbers one through six when ready. Drop a single metal BB into the ball collector of each channel and watch for the corresponding count to appear on the screen and for the correct time to display at the bottom of the screen.
 1. If the correct time is not displayed, repeat steps 1.6.3 and 1.6.4 to reset the clock.
 2. If the test metal BB is not recorded, the collector is blocked. Check visually for blockage and restart the system.
 3. If a channel "counts" up one event every second, this indicates that the channel is not properly connected. Check all connections and restart the system.

3. Experiment End and Data Analysis

1. After emergence has ended (see Results and **Figures 8 and 9** for examples of the time scale), power down the apparatus by unplugging the Arduino. Racks may be disassembled and cleaned for reuse.
2. During the experiment, data is stored on the SD card in a comma-delimited file (CSV) accessible by the R programming language. Use the SD card to transfer data to the computer, and RStudio to auto-generate bubble plots of the data.
 1. Both event and temperature data are saved in the same file for data integrity. Hence, some processing must be completed prior to analysis. Import the comma delimited file into a spreadsheet program. Columns I and J are the date and time of emergence for bees; make them columns A and B by cutting and pasting columns A-E into a second spreadsheet, and save as a separate file, this is the temperature data.
 2. Title column A with, "Date" and column B "Time," and sort the data by column A then by B. Save as a CSV file.
 3. Download and install the latest version of the RStudio from <https://www.r-project.org/>. Help with using RStudio for uploading and analyzing data can be found here at <https://cran.r-project.org/doc/manuals/r-release/R-intro.html>.
 4. Using the R script available in the supplemental data, upload the data into RStudio. Change the working destination in the R script to match where the excel *.CSV file is located. Run the script and select the data file to analyze. Type "plot" into the R console. The bubble plot will be located in the working destination named "High-res;" rename this file to save as a high-resolution tiff (300 dpi) file.

Representative Results

Emergence of *M. rotundata* is asynchronous without exposure to an environmental cue, with emergence occurring uniformly throughout the day⁴. However, when exposed to a square-wave thermoperiod (4° C thermoperiod), emergence becomes synchronous to the thermophase^{4,5}. This result is similar to other studies where insects have been found to use thermoperiod cues to regulate emergence, including the flesh fly *Sarcophaga crassipalpis*⁶, the onion fly *Delia atiqua*⁷ and the boll weevil *Anthonomus grandis grandis*⁸. One study has shown that stress during development affects the synchrony of adult emergence in *S. crassipalpis*⁹. Here, we present results from *M. rotundata* that were exposed to a stress during development, to test the hypothesis that this treatment causes the desynchronization of adult emergence.

Successful run

The user should watch the LCD screen before opening the incubator to make sure insects are no longer emerging. Once the experiment is completed, the SD card is removed and the data can be exported into RStudio as a comma-delimited file to be visualized as a bubble plot, as described earlier. **Figure 8** displays bee emergence under a 4° C thermoperiod after exposure to a cold stress during development. The red crosshairs indicate the median time and day of emergence and the file name is the title. This R script should be used to visualize the data, but should not serve as the sole analysis. To analyze emergence response to an environmental cue, the data can be analyzed for rhythmicity (see *Analysis*).

Complication

When a sensor is clogged with metal BBs, the lack of a signal is repeatedly counted, giving rise to multiple false data points. **Figure 9** demonstrates the same dataset presented in **Figure 8**, but with one of the six channels clogged with BBs, therefore creating the large bubble on the graph. In the event of a clogged sensor, data from this channel can be easily removed from the analysis. Incorporating multiple channels in an experiment is beneficial in minimizing the impact of a clogged sensor.

Analysis

Analyzing data for presence of synchronization can be done by calculating "parameter R," a scalar statistic that identifies if emergence is rhythmic or arrhythmic^{10,11,12}. This is done by calculating the highest number of emerging adults in an 8-hour window, dividing this number by the number of adults emerging outside the 8-hour window, then multiplying by 100. All individuals that emerged should be pooled to calculate the number of emerging adults for each hour of the day. The theoretical range of parameter R is from 0 (all emergence occurs within the gate) to 200 (emergence is distributed uniformly throughout the day)¹⁰. R values < 60 are considered rhythmic emergence, 60 < R < 90 are weakly rhythmic, and R > 90 are arrhythmic. R values > 150 indicate uniform distribution of emergence¹⁰. **Figure 8** shows that emergence is rhythmic with parameter R = 20.21 < 60. Due to the fact that this type of data is distributed around a repeating 24-hour clock, circular statistics must be employed for a more robust analysis (described in detail in Bennett *et al.*, 2018⁵). This can be accomplished via circular statistics packages available for RStudio (Package 'circular'- CRAN.R-project.org).

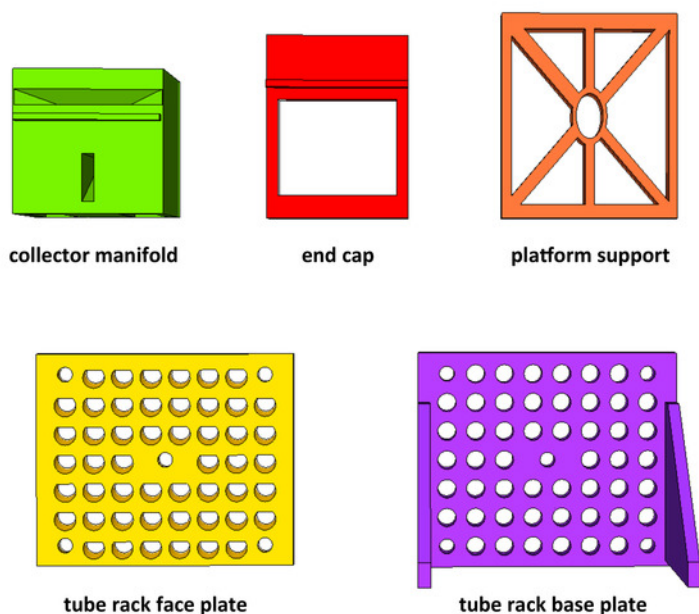


Figure 1: Additive manufactured components. Using PLA filament, 3D print the required parts for the system. For each channel being constructed, parts needed are 1 collector manifold (green), 1 end cap (red), 6 platform supports (orange), 4 tube rack base plates (purple), and 4 tube rack face plates (yellow). [Please click here to view a larger version of this figure.](#)

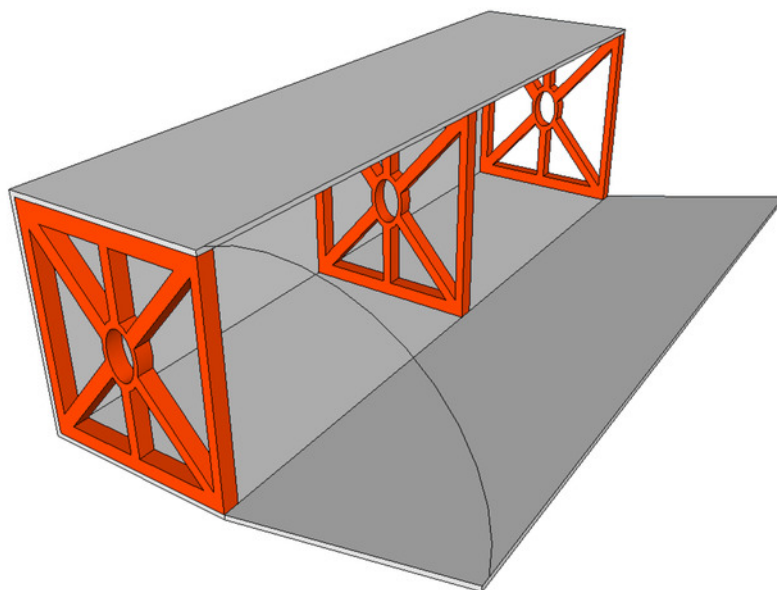


Figure 2: Tube rack platform assembly. Use hot glue to assemble two tube rack platforms per channel being constructed. Use three platform supports (shown in orange) with a section of corrugated plastic (shown in grey). [Please click here to view a larger version of this figure.](#)

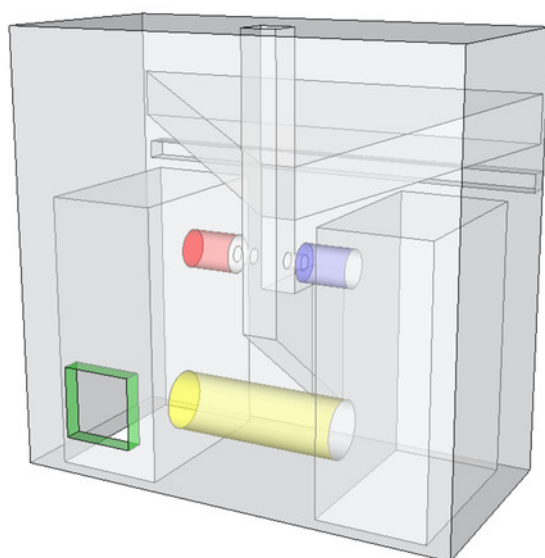


Figure 3: Collector manifold x-ray. Insert one infrared detector into one socket of the collector (shown in blue) and the emitter into the second socket (shown in red). Feed the detector wires through the cabling channel (shown in yellow) and pull all four wires through the access hole (highlighted in green). Ensure no bare wires are touching, using hot glue to secure them in place. [Please click here to view a larger version of this figure.](#)

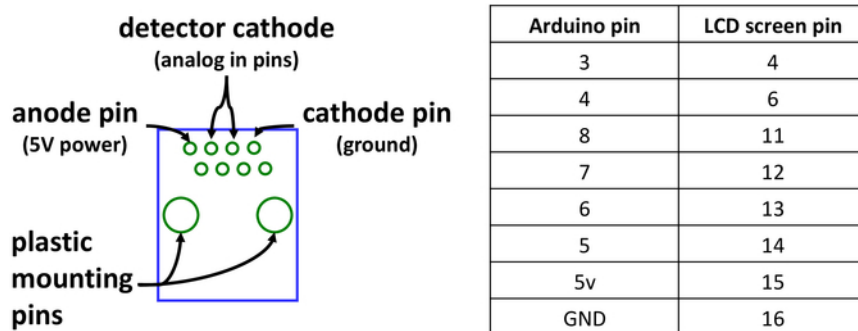


Figure 4: Connector wiring. Wiring diagram for the RJ45 jack prior to affixing to the collector manifold, as seen from the bottom of the jack and wiring table for connecting the LCD screen to the central processor. [Please click here to view a larger version of this figure.](#)

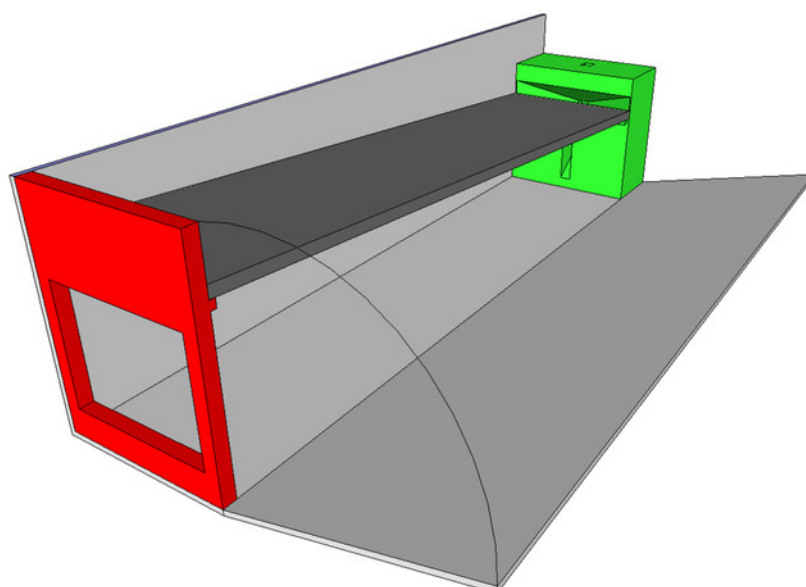


Figure 5: Ball collector assembly. Using one end cap (shown in red), one collector manifold (shown in green) and a 24 x 30 cm piece of corrugated plastic (shown in light grey) assemble the shell of the ball collector assembly. Use an 8 x 27 cm piece of corrugated plastic (shown in dark grey) to add a ramp. [Please click here to view a larger version of this figure.](#)

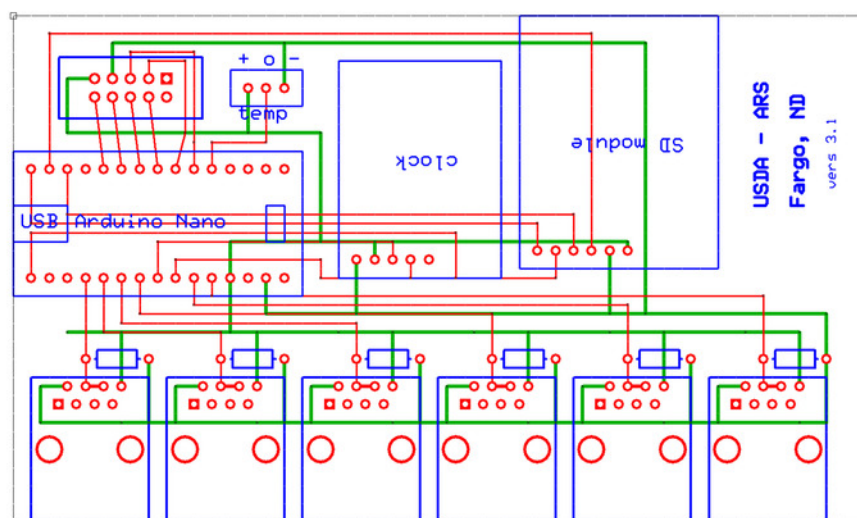


Figure 6: Central Processor PCB board. The PCB board for the central processor consists of a bottom layer (depicted in green), a top layer (depicted in red), and a silkscreen layer (depicted in blue). Solder female headers to all pass-through holes, except for those for the RJ45 jacks (along the bottom) and for the pull-down resistors (directly above the RJ45 pads). [Please click here to view a larger version of this figure.](#)

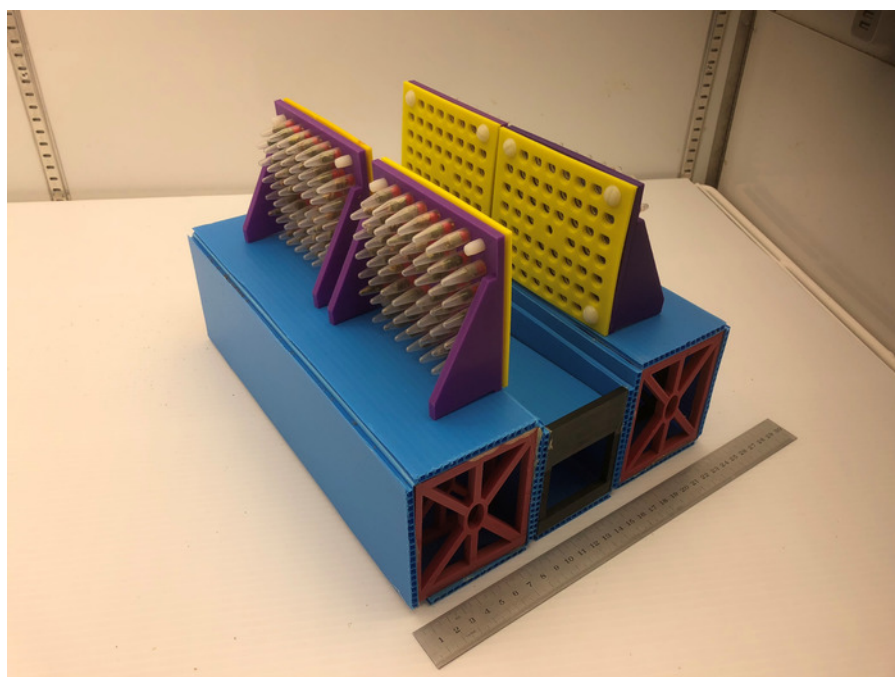


Figure 7: Final assembly. When in use, the apparatus should be assembled with a tube rack platform on either side of each ball collector being used. Tube racks with attached faceplates should be positioned so that they are at the very edge of the Tube Rack Platform, reducing the possibility of falling BBs bouncing off of the apparatus. The footprint of the assembled apparatus is approximately 25 cm x 35 cm, with a height of 20 cm. [Please click here to view a larger version of this figure.](#)

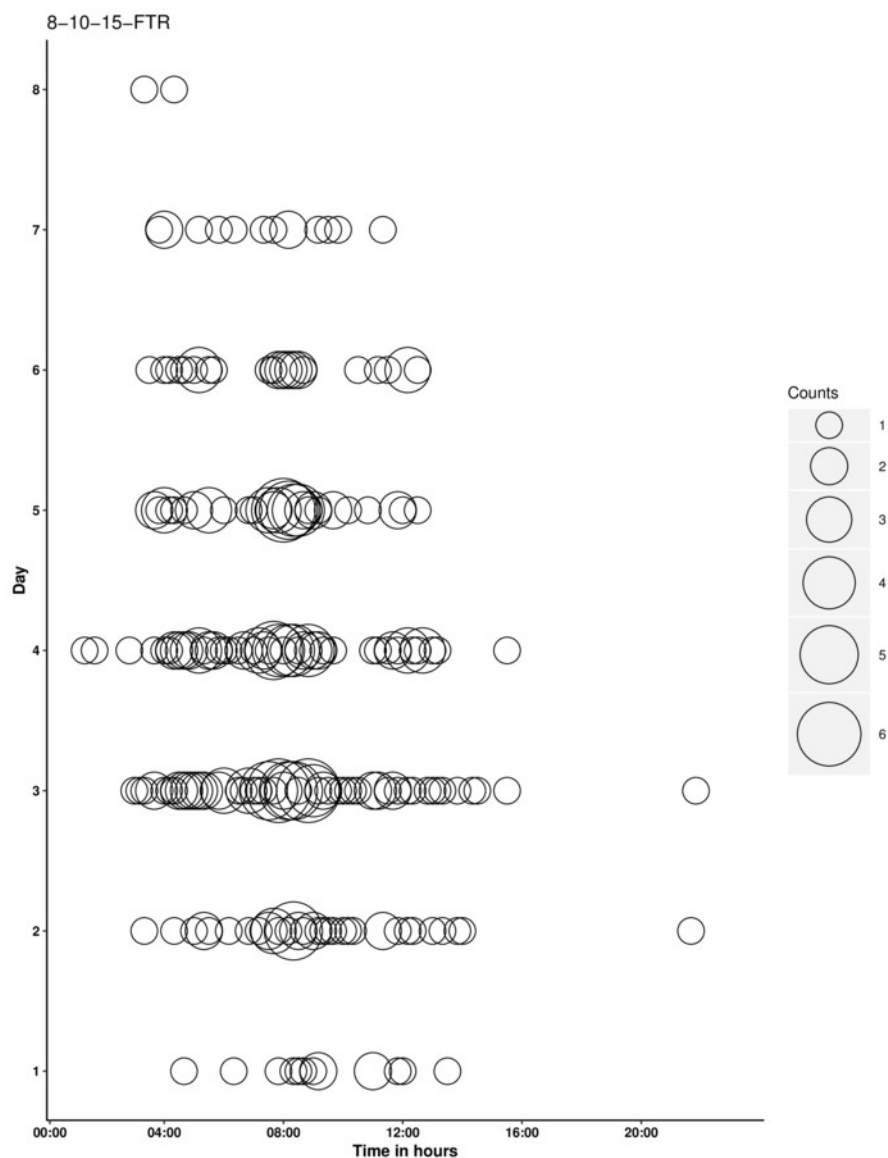


Figure 8: Graph of a typical experimental run after processing in R. [Please click here to view a larger version of this figure.](#)

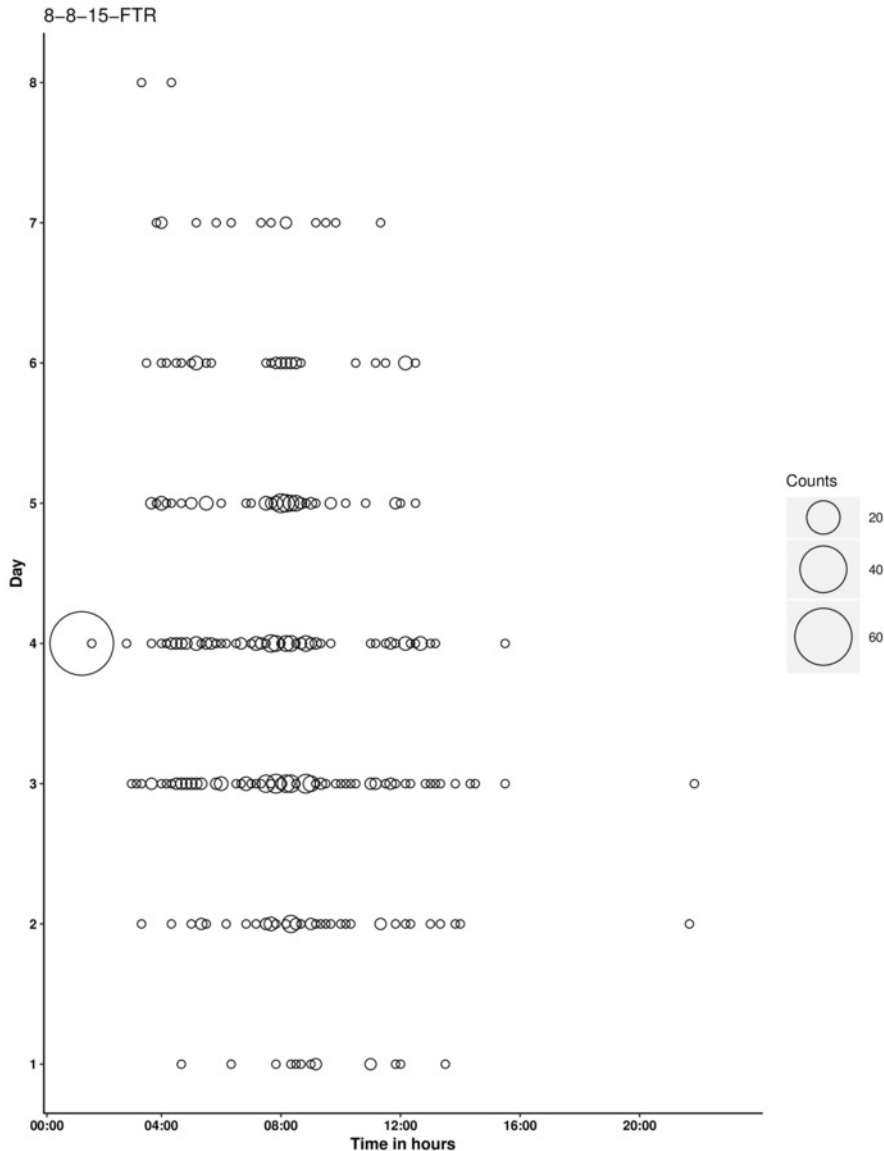


Figure 9: Graph of an experiment that suffered from a clogged detector, as shown by the relatively large bubble on day 4. The clogged channel can be removed from the analysis, thereby preserving the remaining data points. [Please click here to view a larger version of this figure.](#)

Discussion

We present a protocol for the assembly and use of a system that allows for precise measurement of the timing of insect emergence. This system solves two problems, which limited previous designs: partial automation and limited sample size. We solved these problems by automating data collection using microcontrollers, which also enabled us to increase sample size by using multiple channels. The current design has six channels that can hold a total of 1200 bees. Additional channels can be added or subtracted if needed, allowing not only for increased sample size, but also for simultaneously investigating the effects of multiple treatments. Critical steps, modifications, limitations and future applications are discussed below.

The only part of the system that is not automated is loading the racks with brood cells, metal BBs, and airsoft pellets at the beginning of the experiment. Although the racks are designed so they lean back slightly to prevent metal BBs from falling when the racks are standing upright, care must be taken when placing the racks to prevent the accidental release of metal BBs. Also, make sure the racks are flush with the edge of the shelf, so the falling trajectory of the metal BB aligns with the runway. Finally, leaf debris should be wiped clear from the runway, and the undercarriage holding the metal BBs from previous experiments should be cleared to prevent blocking of the sensor. Data is automatically recorded to an SD card as a CSV file, and the script is written so that the Arduino will not run unless an SD card is present. The data file is manually imported into RStudio and visualized using the previously mentioned R script. This script will automatically graph the data as a bubble plot and identify the median time and day of emergence. The Arduino script is written to append event data to the end of the file, which prevents data loss in the event of a power failure. However, this also means that once data is extracted from the SD card, all files should be cleared prior to the next experiment.

Modifications to the SketchUp files can be made to adjust the size of the racks for insects of different sizes, with different sized tubes being used in the modified racks. Furthermore, the size of the airsoft pellet is important because it prevents the insect from leaving the tube, and pellets of different sizes may be needed as well. A wide variety of alterations can be made to the R script to change the appearance of the bubble plots, and other graphic parameters.

We reduced the risk of false positives by writing a debounce code that disables any given channel for one second after a metal BB is detected thereby preventing a single metal BB from being counted as multiple data points. Although, this creates the possibility of a data point being missed if many bees emerge at once, but the fact that the channels are independent reduces this risk. Another limitation of the current system is that individual data points are not discernable, *i.e.*, a falling metal BB cannot be traced back to a specific individual. Furthermore, the current system measures emergence but not eclosion rhythms in *M. rotundata*, but would measure eclosion rhythms in species where emergence and eclosion are synonymous. Finally, the current design is not weatherproof, limiting its use to controlled environments.

Future applications include examining the effects of other abiotic and biotic environmental cues for timing emergence of *M. rotundata*. Furthermore, because insects occupy diverse environments, relevant environmental cues vary between species. Thus, incorporation of more insect species is important to investigating how circadian systems evolved across taxa. Little is known about how developmental conditions affect the timing of adult emergence; therefore, our system can be used to decipher the effects of treatments on emergence. Furthermore, combinations of environmental cues can affect insect responses, thus future experiments should incorporate multiple environmental cues to understand their relative effects on emergence. Lastly, deployment in the field to observe how natural settings mediate emergence rhythms is of interest. The ease of use of this system, and its unique combination of additive manufacturing, open-source programming, and observable biological traits, make it a candidate for use in an educational setting.

Disclosures

The authors have nothing to disclose.

Acknowledgements

We would like to acknowledge the Insect Cryobiology and Ecophysiology working group in Fargo, ND for their helpful feedback on experiments using the described system.

References

1. Lankinen, P. Geographical variation in circadian eclosion rhythm and photoperiodic adult diapause in *Drosophila littoralis*. *Journal of Comparative Physiology A*. **159**, 123-142 (1986).
2. Watari, Y. Comparison of the circadian eclosion rhythm between non-diapause and diapause pupae in the onion fly, *Delia antiqua*. *Journal of Insect Physiology*. **48**, 83-89 (2002).
3. Zimmerman, W.F., Pittendrigh, C.S., Pavlidis, T. Temperature compensation of the circadian oscillation in *Drosophila pseudoobscura* and its entrainment by temperature cycles. *Journal of Insect Physiology*. **14**, 669-684 (1968).
4. Yocum, G.D., Rinehart, J.P., Yocum, I.S., Kemp, W.P., Greenlee, K.J. Thermoperiodism synchronizes emergence in the alfalfa leafcutting bee (Hymenoptera: Megachilidae). *Environmental Entomology*. **45**, 245-251 (2016).
5. Bennett, M. M., Rinehart, J. P., Yocum, G.D., Doetkott, C., Greenlee, K.J. Cues for cavity nesters: Investigating relevant Zeitgebers for emerging leafcutting bees, *Megachile rotundata* (Hymenoptera: Megachilidae). *Journal of Experimental Biology*. **221**, jeb175406 (2018).
6. Miyazaki, Y., Goto, S.G., Tanaka, K., Saito, O., Watari, Y. Thermoperiodic regulation of the circadian eclosion rhythm in the flesh fly, *Sarcophaga crassipalpis*. *Journal of Insect Physiology*. **57**, 1249-1258 (2011).
7. Watari, Y., Tanaka, K. Effects of background light conditions on thermoperiodic eclosion rhythm of onion fly *Delia antiqua*. *Entomological Science*. **17**, 191-197 (2014).
8. Greenberg, S.M., Armstrong, J.S., Setamou, M., Coleman, R.J., Liu, T.X. Circadian rhythms of feeding, oviposition, and emergence of the boll weevil (Coleoptera: Curculionidae). *Insect Science*. **13**, 461-467 (2006).
9. Yocum, G.D., Zdarek, J., Joplin, K.H., Lee, R.E., Smith, D.C., Manter, K.D., Denlinger, D.L. Alteration of the eclosion rhythm and eclosion behavior in the flesh fly, *Sarcophaga crassipalpis*, by low and high temperature stress. *Journal of Insect Physiology*. **40**, 13-21 (1994).
10. Winfree, A. Integrated view of resetting a circadian clock. *Journal of Theoretical Biology*. **28**, 327-374 (1970).
11. Watari, Y., Tanaka, K. Interacting effect of thermoperiod and photoperiod on the eclosion rhythm in the onion fly, *Delia antiqua* supports the two-oscillator model. *Journal of Insect Physiology*. **56**, 1192-1197 (2010).
12. Short, C.A., Meuti, M.E., Zhang, Q., Denlinger, D.L. Entrainment of eclosion and preliminary ontogeny of circadian clock gene expression in the flesh fly, *Sarcophaga crassipalpis*. *Journal of Insect Physiology*. **93 - 94**, 28-35 (2016).