Explanation of the left-right learning setup for testing cognitive abilities of common shrews (*Sorex araneus*)

Author: Lara Vrbanec

Date: 30.10.2020

Contents

Objective

The objective of the left- right learning setup is to test the spatial memory of the common shrew in the outside enclosure, without the disturbance of the experimenter. The animal is presented with two identical opaque tunnels, one of which always provides the food reward. The aim is to see how many trials animal needs to learn the rewarding arm, and how high is the mistake (how often the animal will sample the non-rewarding arm) after the learning objective is reached.

Left right learning setup design

Setup consist of two identical opaque tunnels (1 x 10cm) connected to the automatic food dispenser. Every tube consists of two IR sensors, which record the direction of animal movement (entry / exit), and date and time of the activation. The information is saved as a .txt file on a SD card as ' sensor,1a,timestamp,[date].

Tunnel design

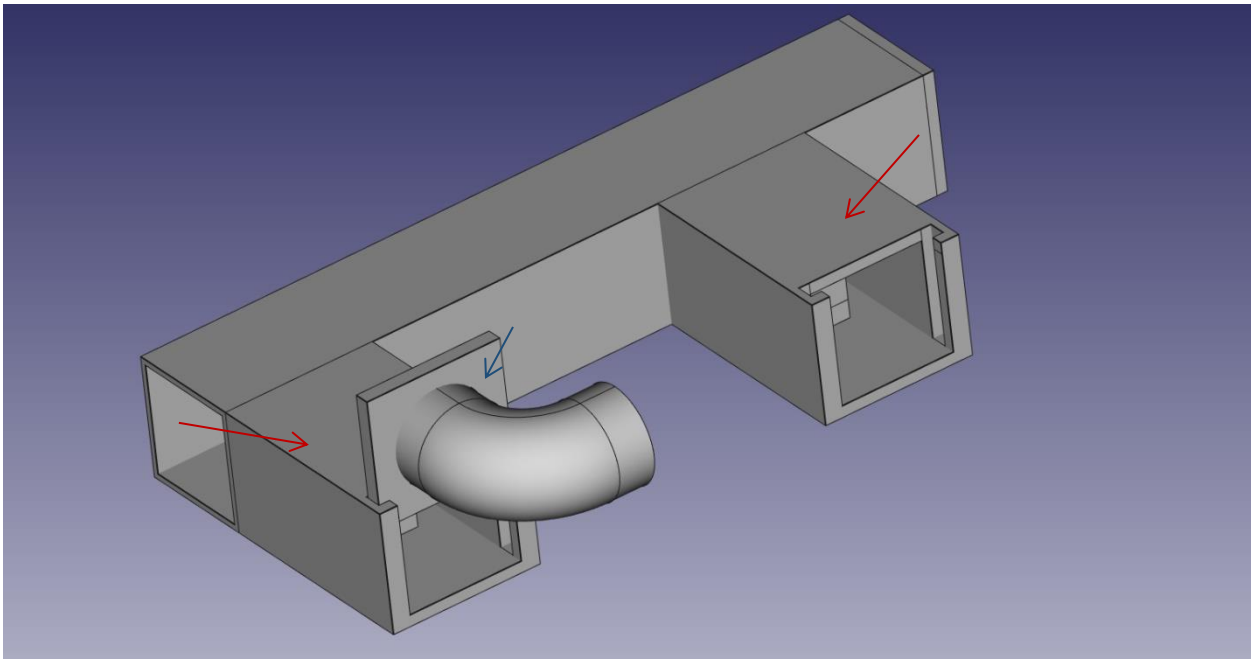


Figure 1. 3D model of the right / left tunnel created in software FreeCAD. Two constructions on the right marked with red arrows are used as IR sensor holders, and for the stabilization of the test setup. The lid, marked with blue arrow is used to prevent the moisture from entering the setup as the setup is intended for use in outdoor conditions.

Automatic food dispenser design

The dispenser consists of thee parts (i) body that holds the mealworms (ii) casing and (iii) lid. The body is intended to connect to the NEMA-17 size stepper motor, which allows for the rotation of the dispenser and the dispensing of mealworms due to the gravity.

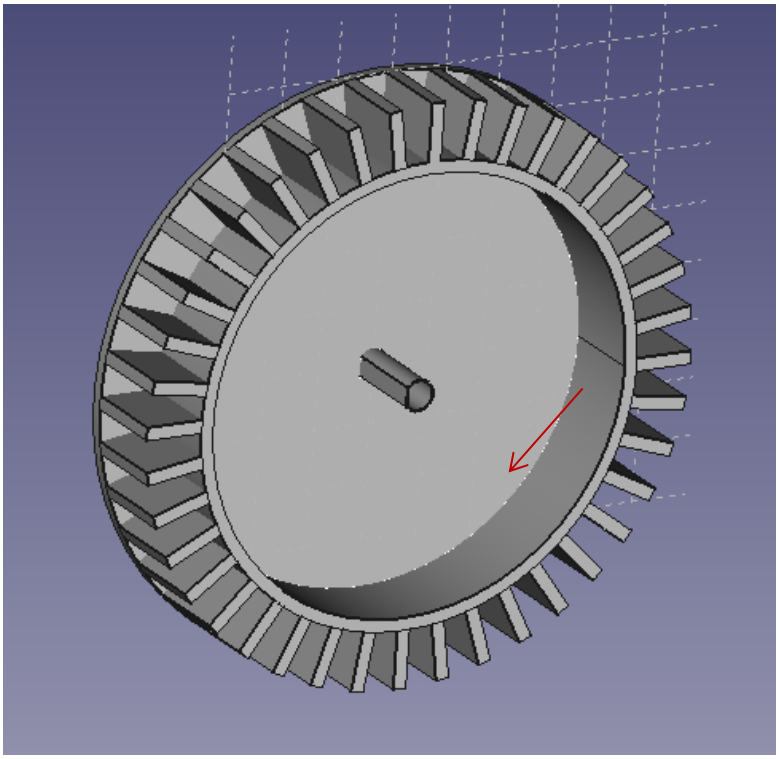
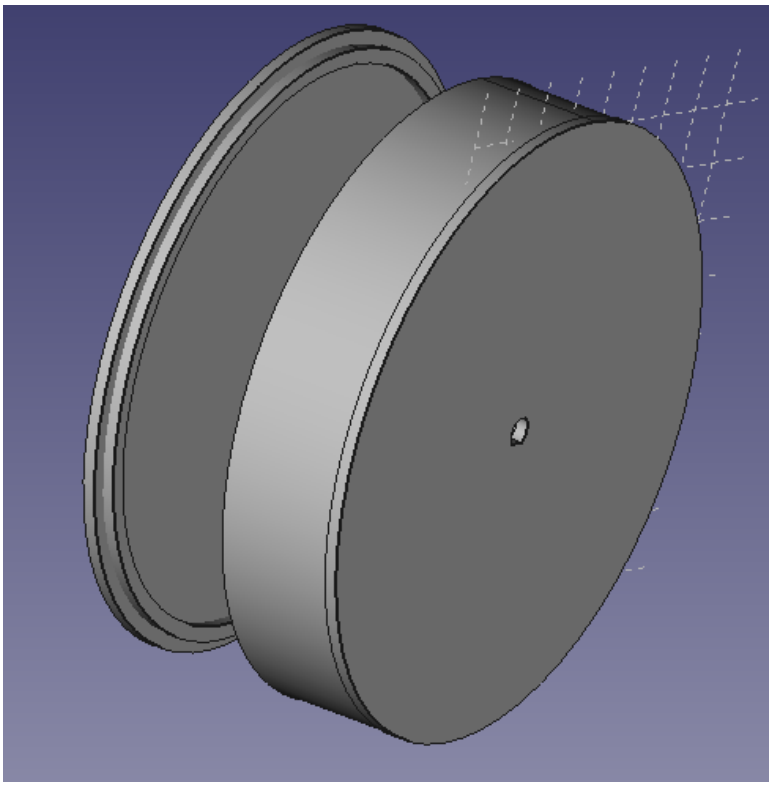


Figure 2. 3D model of the body, created in software FreeCAD. The rotating body holds up to 100 mealworms. The body is intended to fit on the NEMA-17 size stepper motor trough the holder in the middle (red arrow).



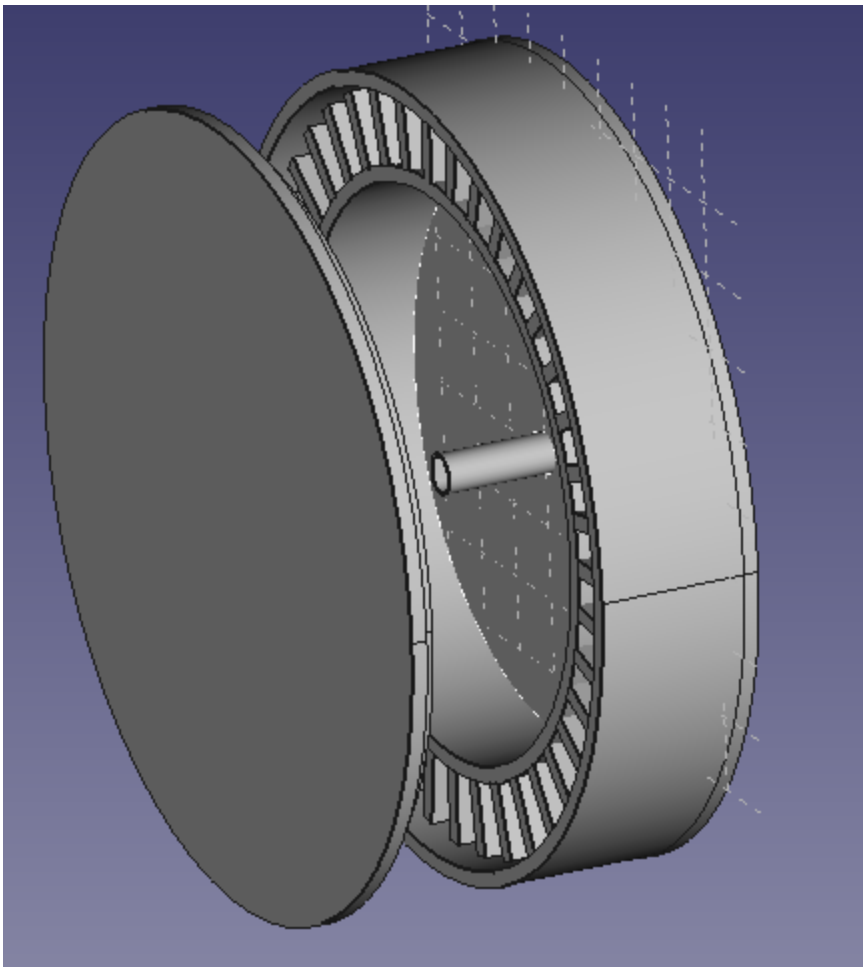


Figure 4. 3D model of the whole food dispenser, created in software FreeCAD.

Electronics components and Supplies

- 1x Arduino (we used Arduino Uno)

- 1x Adafruit Motor Shield V2

- 4x obstacle avoidance IR sensors (we used Waveshare)

- 1x stepper motor (we used NEMA-17 size - 200 steps/rev, 12V 350mA)

- RTC for Arduino (we used tiny RTC I2C module)

- Micro SD card module

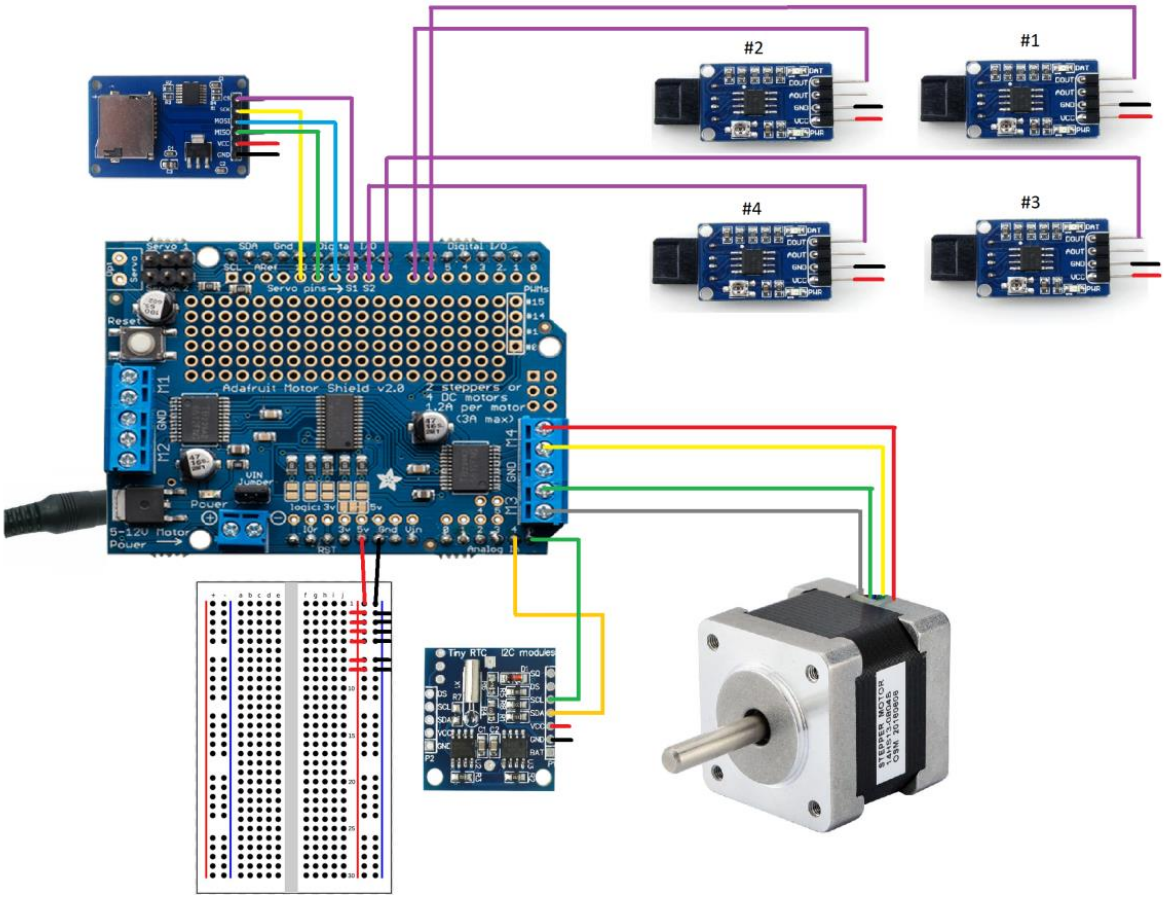
- Bread board

- Jumper cables

- Power source for Arduino (we used 12V/5A) and for Raspberry Pi (we used official SUB charger 5.1V 3A)

- micro SD card

Schematics



Arduino code

When the animal enters the tunnel (sensor2 activated after sensor1) the motor dispenses the food and saves the unix time of animal entrance on the SD card.

```
// OS Windows 10, Arduino 1.8.10

// Author Lara Vrbanec

// Date 06/04/2020

// code sources SD_card https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/ RTC https://lastminuteengineers.com/ds1307-rtc-arduino-tutorial/

// sensors Waveshare_IR_sensor MicroSD_card_adapter Tiny_RTC_12C_module(DS1307)


// When IR sensor1 is activated after IR sensor2, the motor is rotated and the unix time is saved on a micro SD card
```

```
// When IR sensor3 is activated after IR sensor4, the unix time is
saved on SD card


#include <SPI.h>  // library for SPI SD

#include <SD.h>   // library for SD

#include <Wire.h> // library for RTC

#include "RTClib.h" // library RTClib by Adafruit for RTC

#include <Adafruit_MotorShield.h> // library for Adafruit motor shield
v2



#define ledPin 4

#define motor 2

#define IR1 6

#define IR2 7

#define IR3 8

#define IR4 9



RTC_DS1307 rtc;

File myFile;  // name of a file to be saved

Adafruit_MotorShield AFMS = Adafruit_MotorShield(); // Create the
motor shield object with the default I2C address

Adafruit_StepperMotor *Motor = AFMS.getStepper(200, motor); // Connect
a stepper North with 200 steps per revolution (1.8 degree) to motor
port #2 (M3 and M4)


const int chipSelect = 10; // SD of SD card module. Other SD pins are
pre-defined

const int OneMotorRotation = 15; // define n degrees by which the
motor will move

boolean seq1 = false; // used to create a sequence of sensors

boolean seq2 = false; // used to create a sequence of sensors
```

```
// blink out an error code (number of blinks equal the error number)
void error(uint8_t errno) {
    while(1) {
    uint8_t i;
    for (i=0; i<errno; i++) {
      digitalWrite(ledPin, HIGH);
      delay(100);
      digitalWrite(ledPin, LOW);
      delay(100);
    }
    for (i=errno; i<10; i++) {
      delay(200);
    }
  }
}

void setup()
{
  // Open serial communications
  Serial.begin(9600);

  // create with the default frequency 1.6KHz for motor shield
  AFMS.begin();


  // Define IR sensor imput (digital)
  pinMode(IR1,INPUT); //
  pinMode(IR2,INPUT); //
  pinMode(IR3,INPUT); //
  pinMode(IR4,INPUT); //
```

```
// Set speed of motor
Motor -> setSpeed(20);  // 10 rpm

// Check if SD card started normally

Serial.print("Initializing SD card...");
if (!SD.begin()) {
  Serial.println("initialization failed!");
  error(1);
  return;
}
Serial.println("initialization done.");

// Create a filename on SD when Arduino re-starts
  char filename[15]; //create a filename with 15 characters
  strcpy(filename, "TRC00.TXT");
  for (uint8_t i = 0; i < 100; i++) {
  filename[3] = '0' + i/10;
  filename[4] = '0' + i%10;
  if (! SD.exists(filename)) {
    // only open a new file if it doesn't exist
    myFile = SD.open(filename, FILE_WRITE);
      // write if it cannot open the SD
      if( ! myFile ) {
        Serial.print("Couldnt create ");
        Serial.println(filename);
      error(2);
      return; //try again the loop
      }
      Serial.print("Opened ");
```

```
        Serial.println(filename);
        break;  // leave the loop!
      }
    }


   // Check if RTC  started normally

  Serial.print("Initializing RTC...");
  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    error(3);
    while (1);
  }
  if (!rtc.isrunning()) {
    Serial.println("RTC lost power, lets set the time!");
      // Comment out below lines once you set the date & time.
      // Following line sets the RTC to the date & time this sketch
was compiled
      // rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); //uncomment
if you want to set the time

      // Following line sets the RTC with an explicit date & time
      // for example to set January 27 2017 at 12:56 you would call:

      // rtc.adjust(DateTime(2017, 1, 27, 12, 56, 0));
      }
  Serial.println("initialization done.");


}
```

```
void loop()
{

  //if the animal choses correct side
  if (digitalRead(IR1)==0) {
    seq1 = true;
    Serial.println("Sensor IR1 detected movement");
  }

  if (seq1 == true && digitalRead(IR2)==0) { // if BOTH
      Serial.println("Sensor IR2 detected movement after IR1");


       // move motor
       Motor->step(OneMotorRotation, FORWARD , SINGLE); // 50 is 90
degrees as the motor has 200 steps


       // get the unix time
       DateTime currentTime = rtc.now();
           // if the file opened okay, write to it:
               if (myFile) {
               Serial.println("writing to txt file...");
               myFile.print("Choice,true,");
myFile.print("Timestamp,");
               myFile.println(currentTime.unixtime()); //write to SD
               myFile.flush(); // without .flush nothing can be
printed on a card
               }
               if (! myFile) {
               error(4);
               return;
               }
     seq1 = false;
```

```
    }

  //if the animal choses wrong side
  if (digitalRead(IR3)==0) {
    seq2 = true;
    Serial.println("Sensor IR3 detected movement");
  }

  if (seq2 == true && digitalRead(IR4)==0) { // if BOTH
      Serial.println("Sensor IR4 detected movement after IR3");
        // get the unix time
           DateTime currentTime = rtc.now();
           // if the file opened okay, write to it:
               if (myFile) {
               Serial.println("writing to txt file...");
               myFile.print("Choice,false,");
myFile.print("Timestamp,");
               myFile.println(currentTime.unixtime()); //write to SD
               myFile.flush(); // without .flush nothing can be
printed on a card
               }
               if (! myFile) {
               error(4);
               return;
               }
      seq2 = false;
      }

}
```