Making the clock

Hardware needed:

Here are some suggestions, links may or may not work.

2 mini breadboards:
http://vakits.com/mini-solderless-breadboard-170-points-1193
https://www.adafruit.com/product/65
1 Arduino Nano or similar:
http://www.yourduino.com/sunshop/index.php?l=product_detail&p=84
http://vakits.com/arduino-nano-328p
1 Stepper Motor 28BYj-48, 5v
https://www.amazon.com/ELEGOO-28BYJ-48-ULN2003-Stepper-Arduino/dp/B01CP18J4A
(I used these stepper motor, but not the driver board)
1 L293D IC: https://www.adafruit.com/product/807
Wires, 22AWG size: http://vakits.com/breadboard-jumper-wire-pack

3D printed parts

Tools:
Sandpaper, pliers, screwdriver, hacksaw

Note:

You can print the 608 bearings from the file, or just buy some.
It can be tricky to get the right settings for the bearing to be usable
Check it out here: https://www.myminifactory.com/object/3d-print-85854

All parts, apart the bearings, are easy to print, no support needed, however to make sure all gears spins freely you might need to clean the bores and the shafts
Use sandpaper, cut a small piece and make a small roll and pass it thru the bores

Code for the Ardunio at the end of this document

Parts

All the parts needed

Note about the shafts

All shafts are printed flat on the bed in two halves
This allows to print long and narrow parts and the result is much stronger then printing a high tower.
The top shaft is over 100 mm in length with a diameter of 8 mm
I added a hollow core in the shafts; the result is a lot more longitudinal strands of PLA from on end to the other. This makes strong shafts.
The thing to remember is that each half is unique, so that the treads can match up.

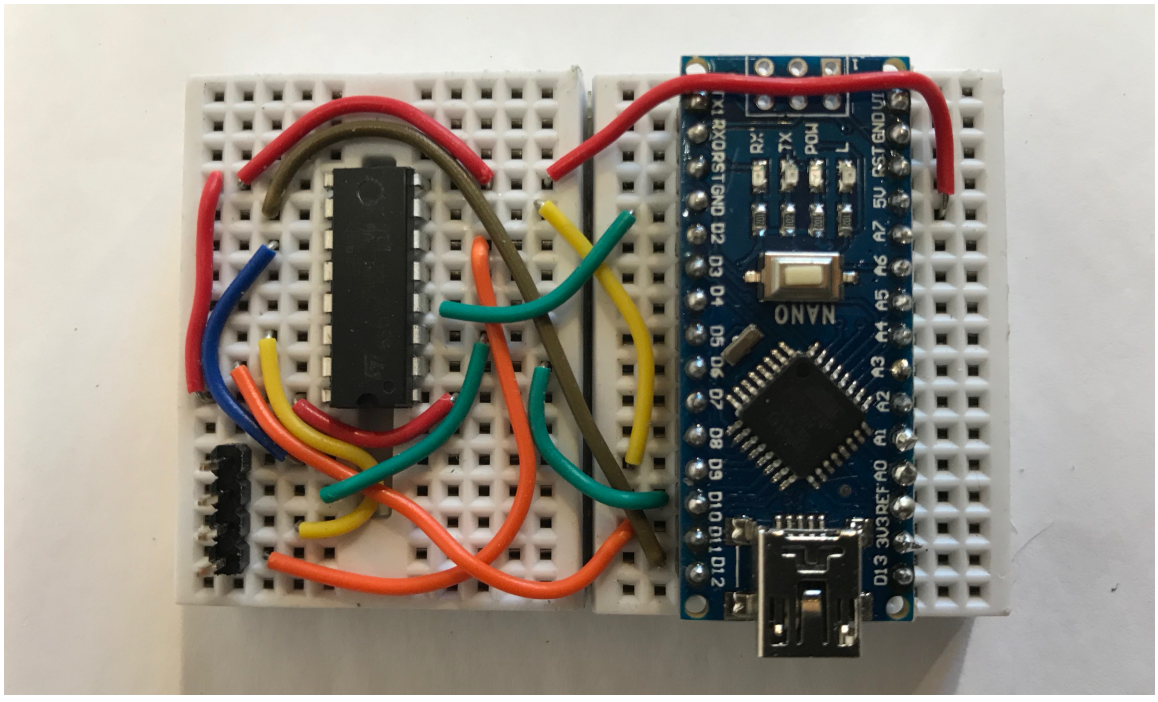
Lower Frame

Parts needed: Shaft, 2 halves, lower back frame, gear15-40, gear10-40, gear12-45, lower front frame

Lower Frame assembly

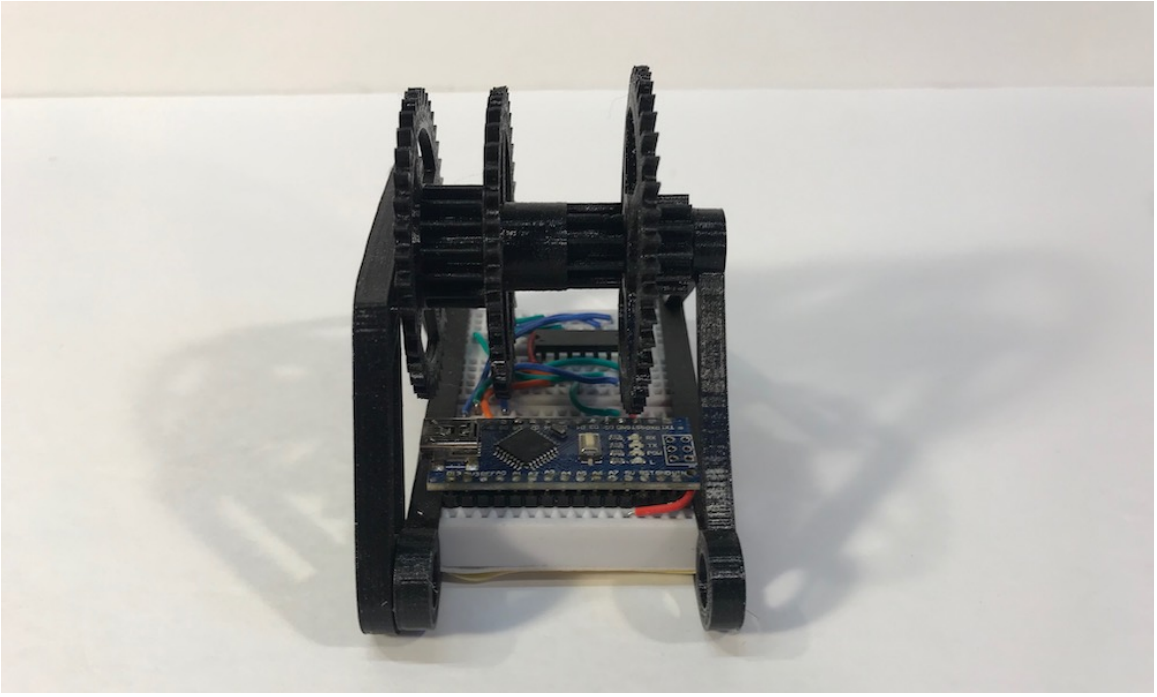
In order, from left to right:
Lower back frame, gear15-40, gear10-40, gear12-45, lower front frame. Notice the different frame shape. Make sure all gears spin freely on the shaft

Board assembly



Bottom assembly complete

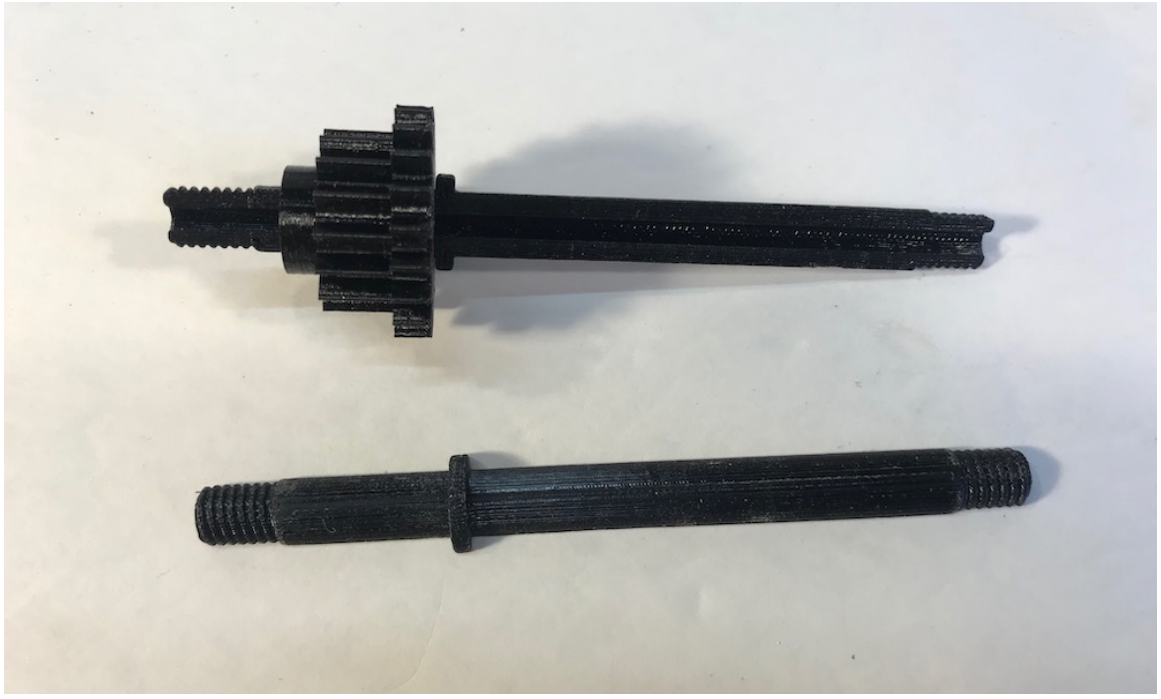
An additional bracket on the back is used to secure the boards in place



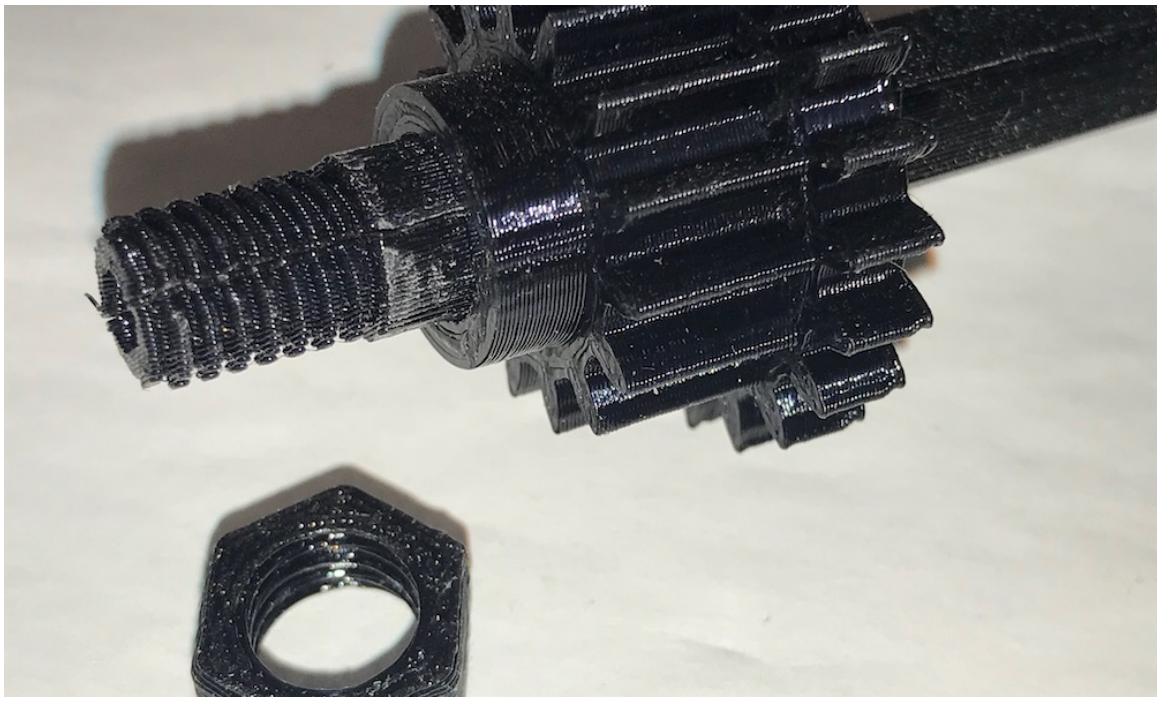
Top Shaft and frame

Parts:

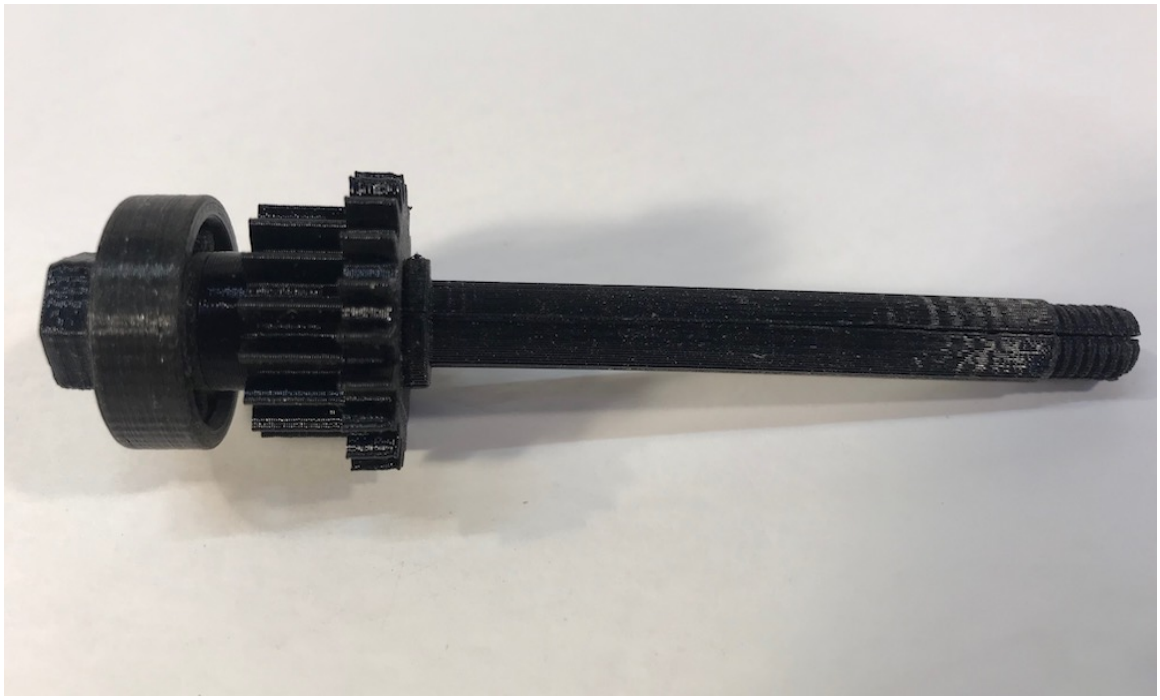
Top shaft details. Notice the two halves



Detail

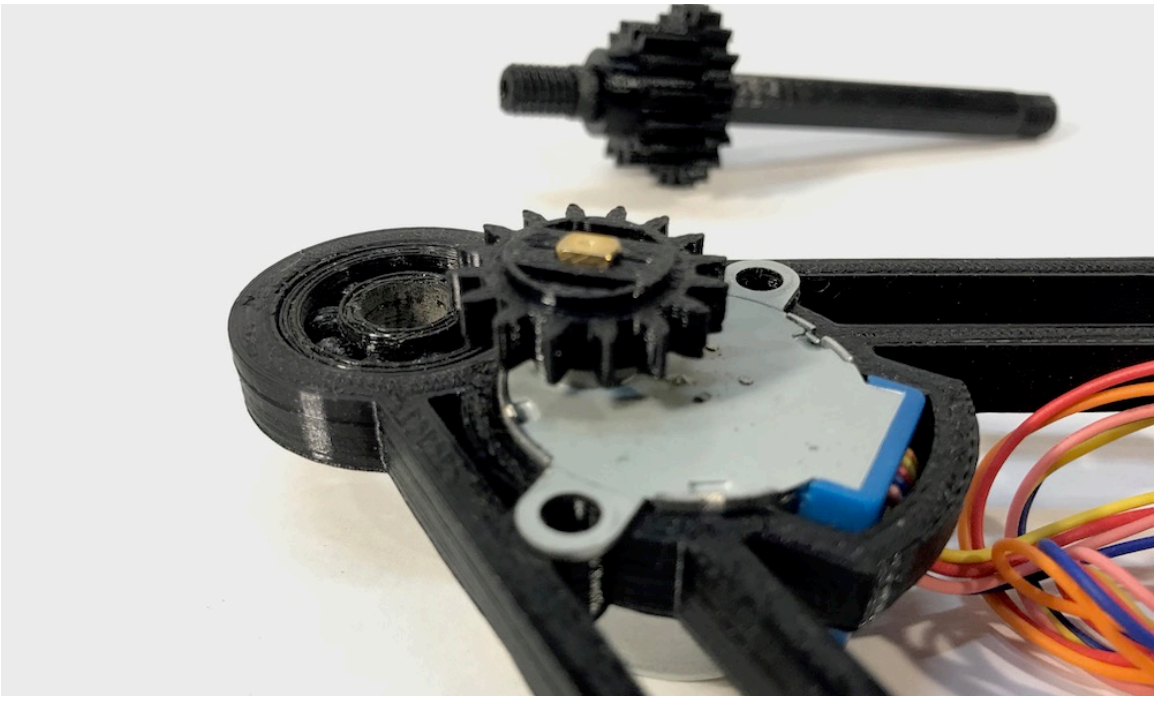


With bearing installed

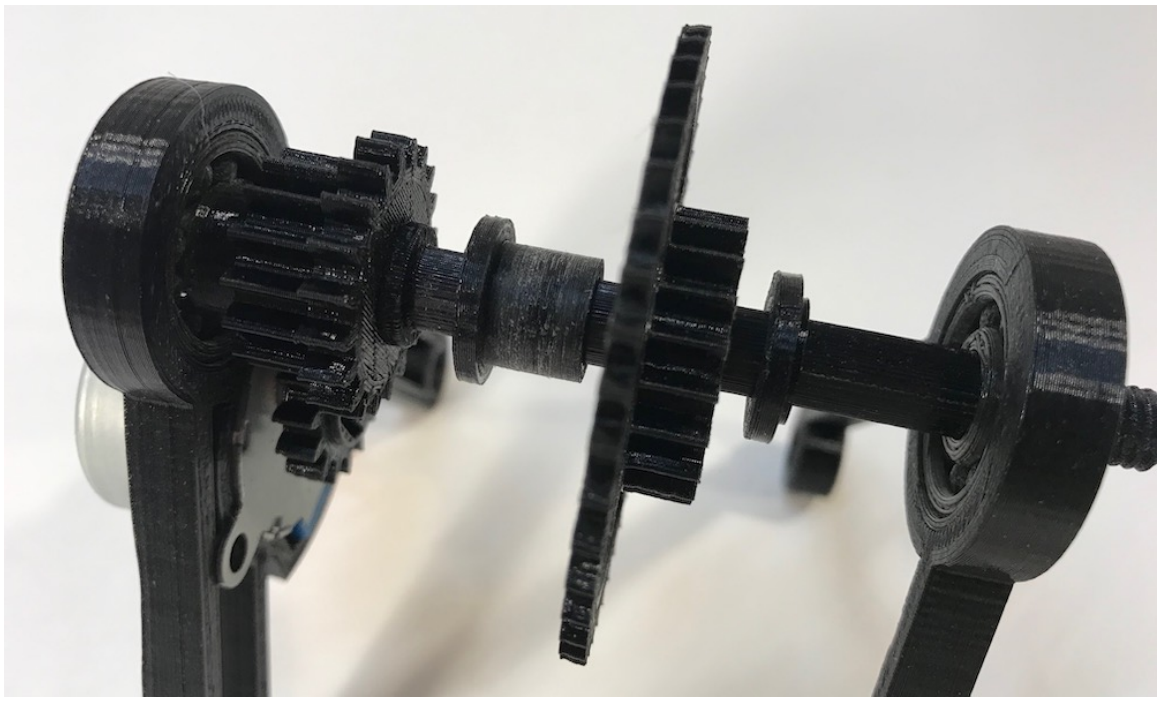


Two top frames with motor and gear installed
Two 608 sized 3D printed bearings

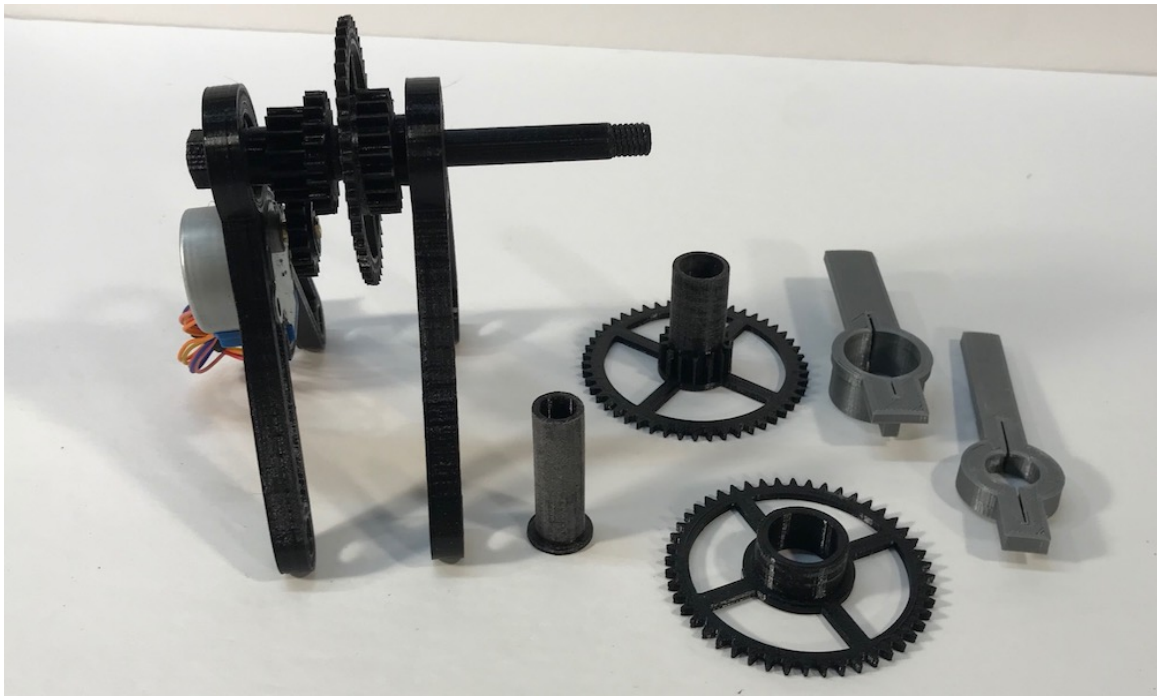
Another view on the motor and gear, no screws needed for the motor



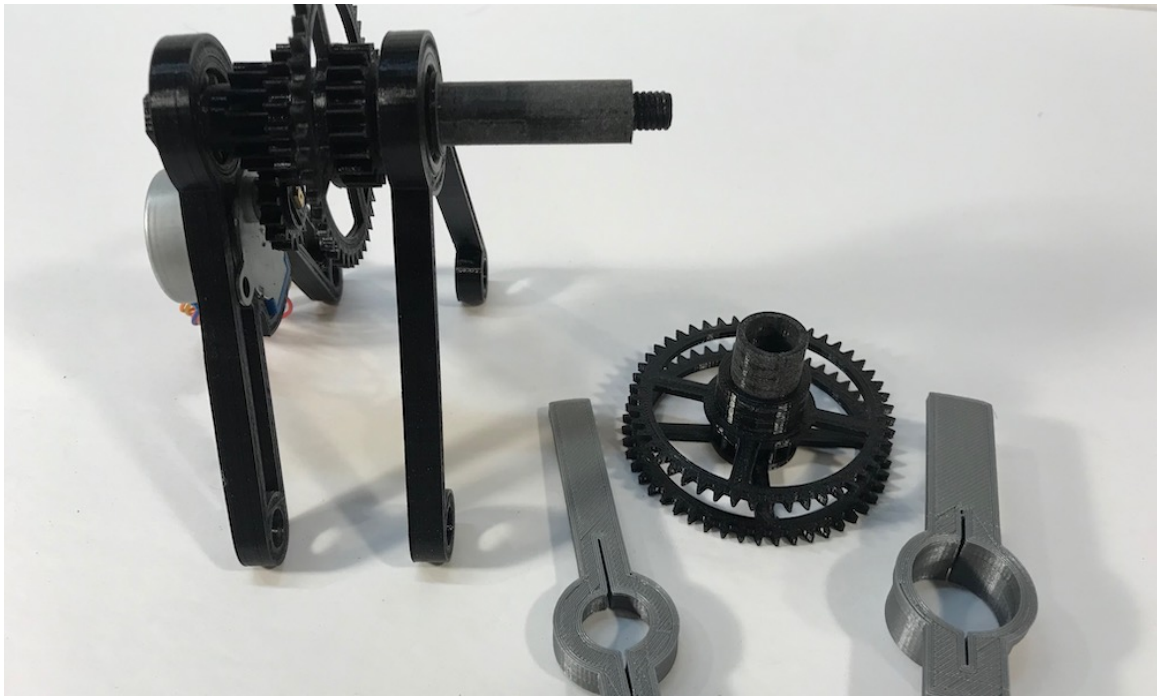
Next
Install the top shaft in the back frame, the small bushing, the 20-45 gear, the spacer, the front top frame, the 20-45 gear need to spin freely.



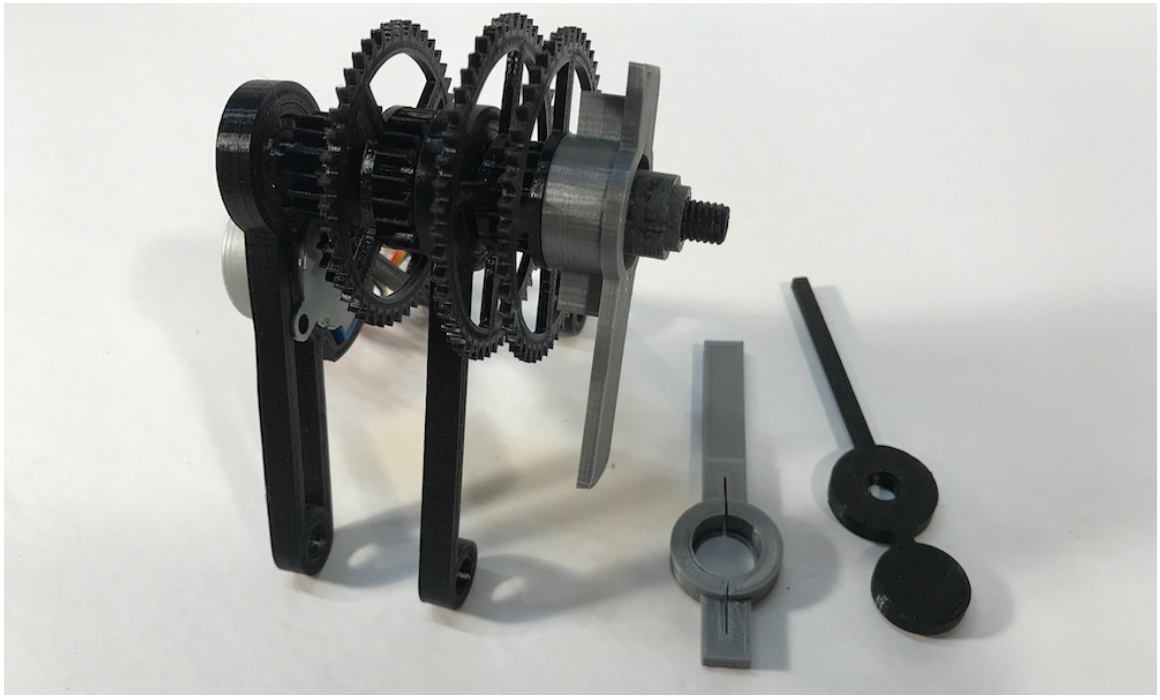
Then to the front part: The long bushing, the minute gear, the hour gear and hands



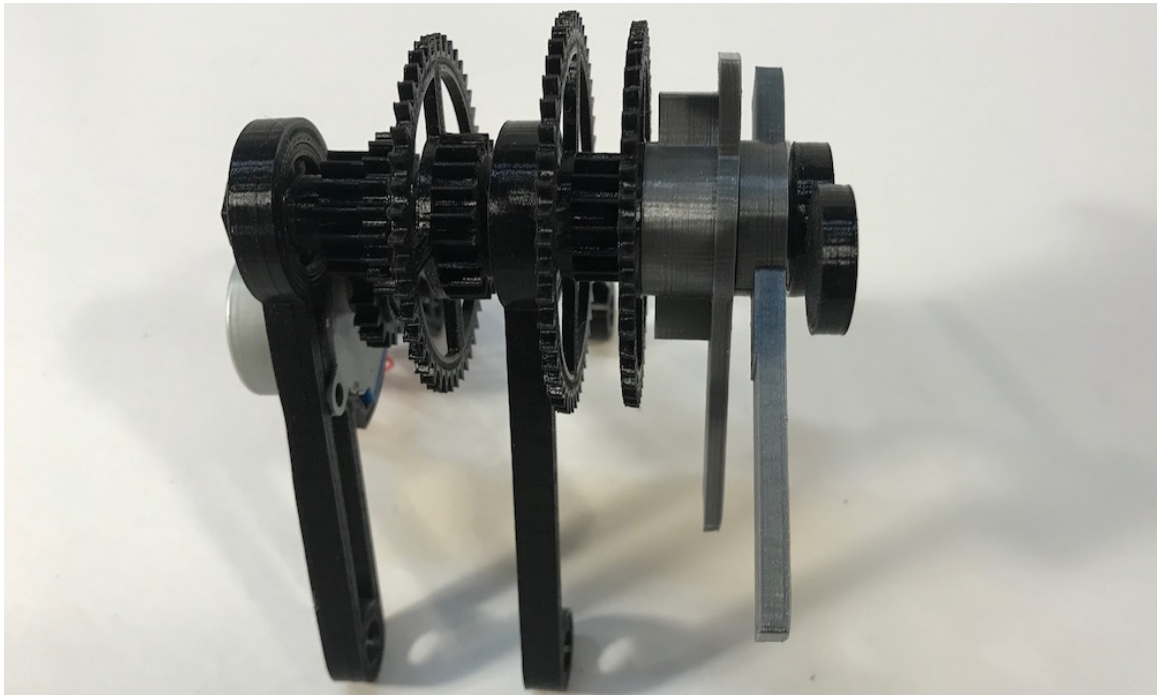
The hour gear has to spin on the minute gear that spins on the bushing



Press on the hour hand

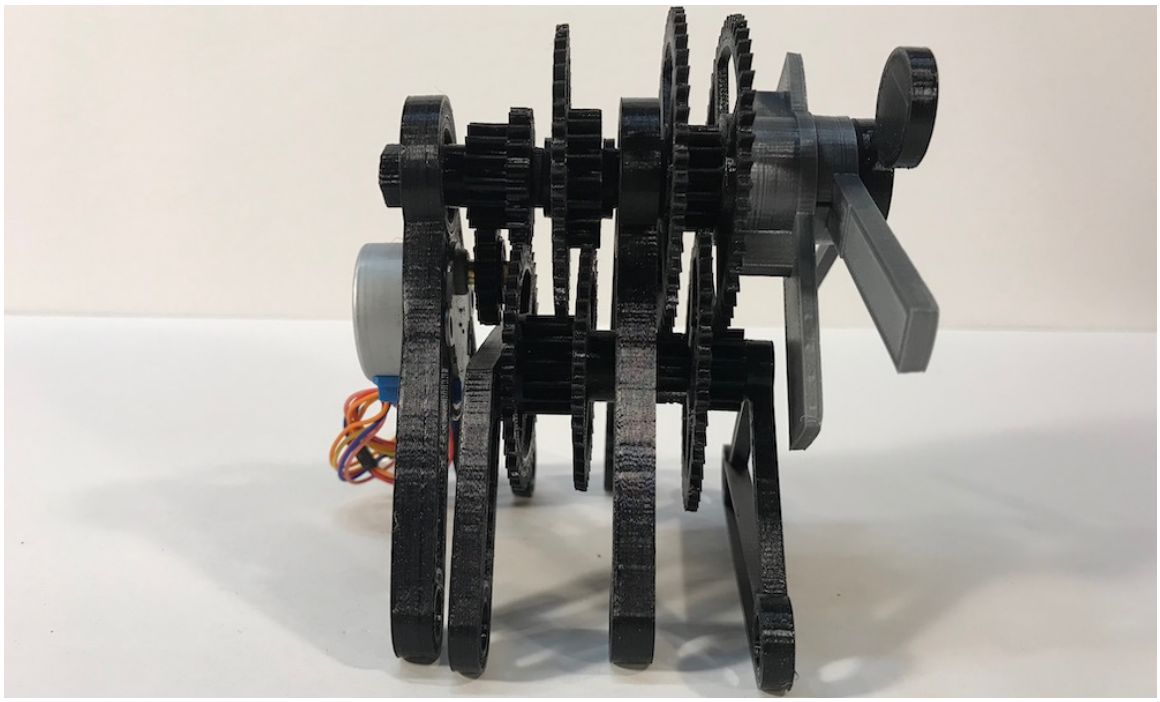


Then press the minute hand, finally the second hand is screwed on

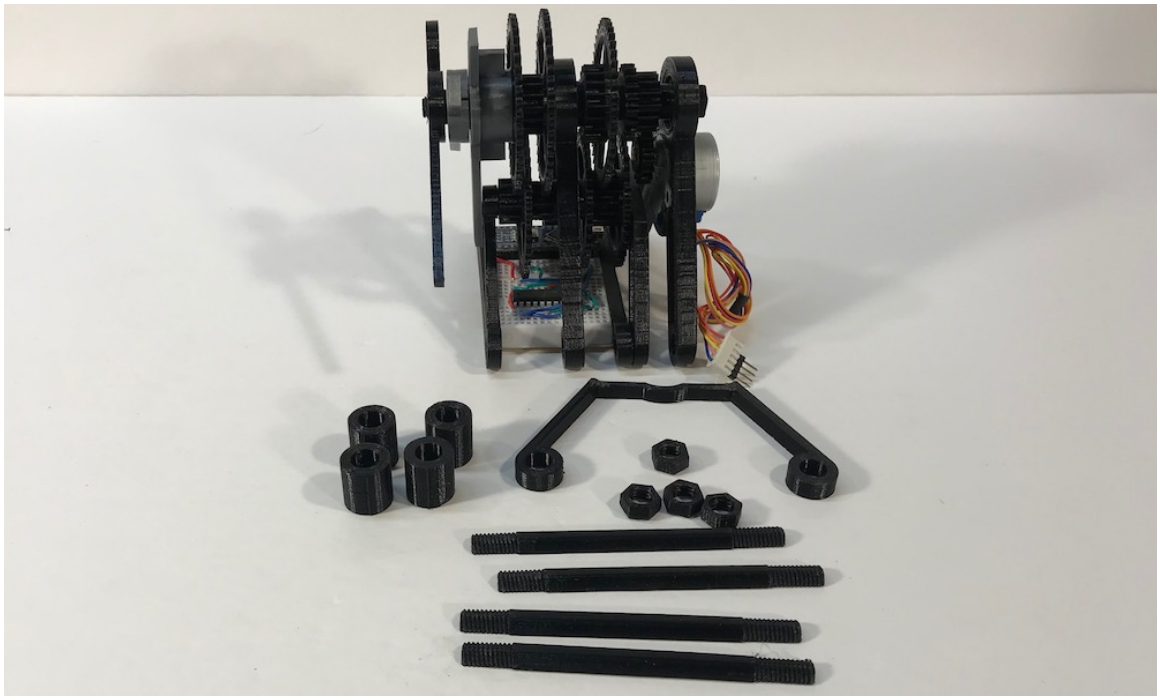


Final assembly

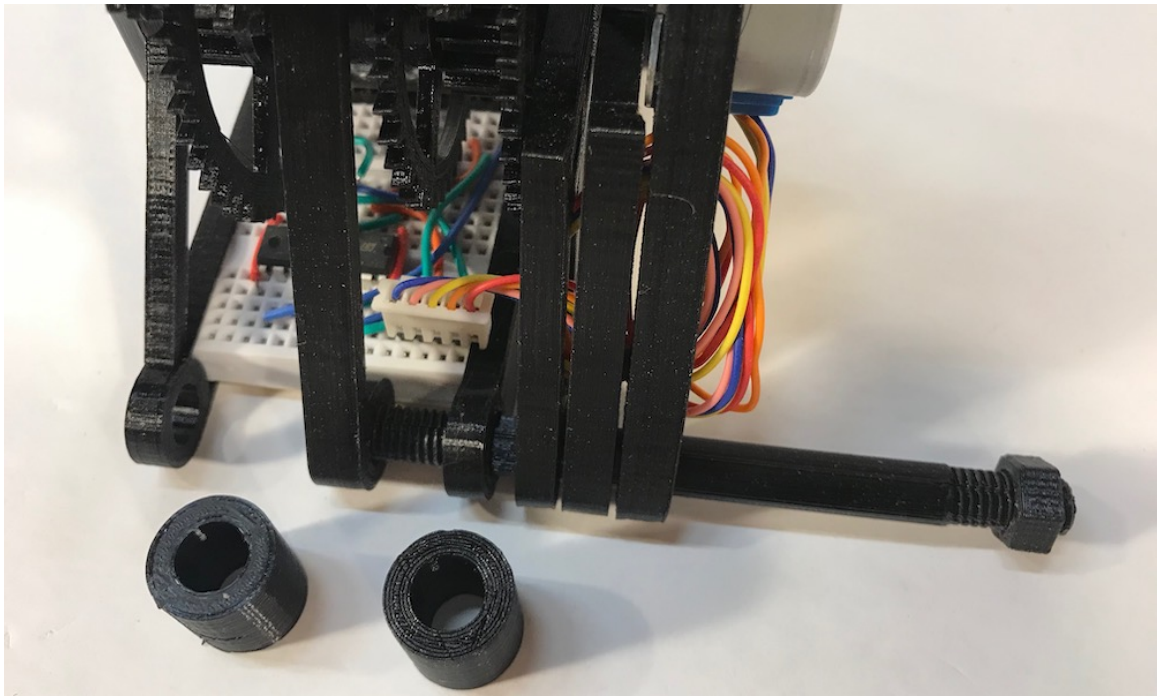
Top frame assembly and lower frame assembly



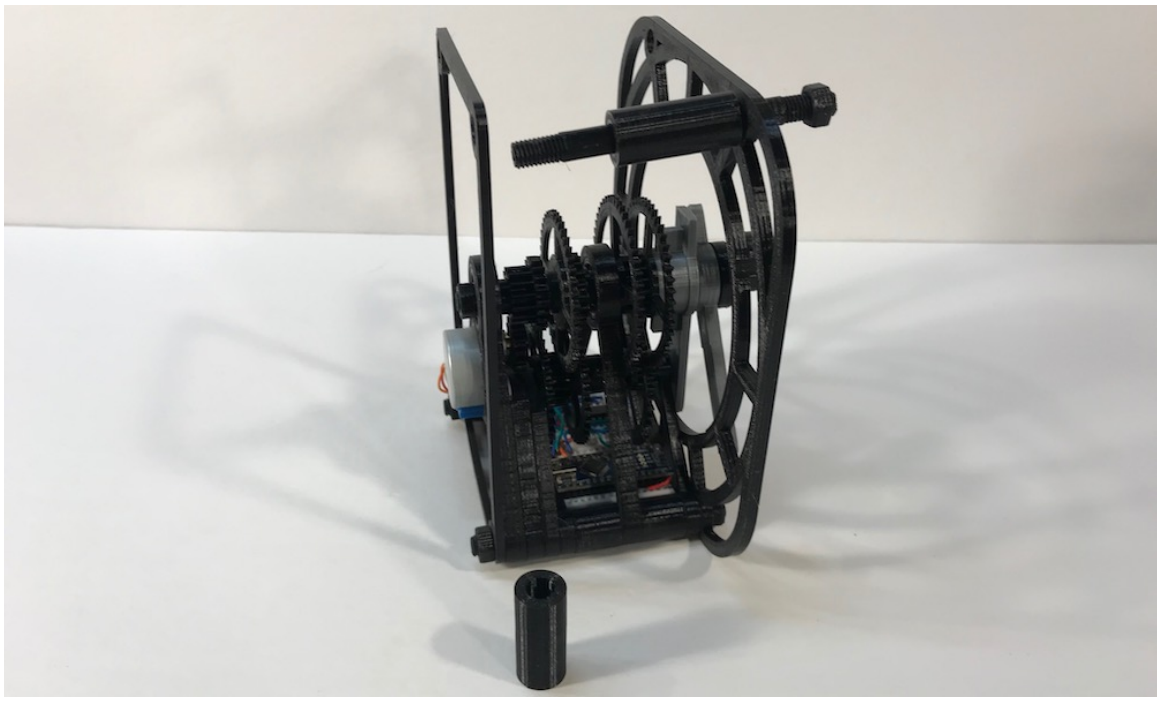
And the extra frame parts



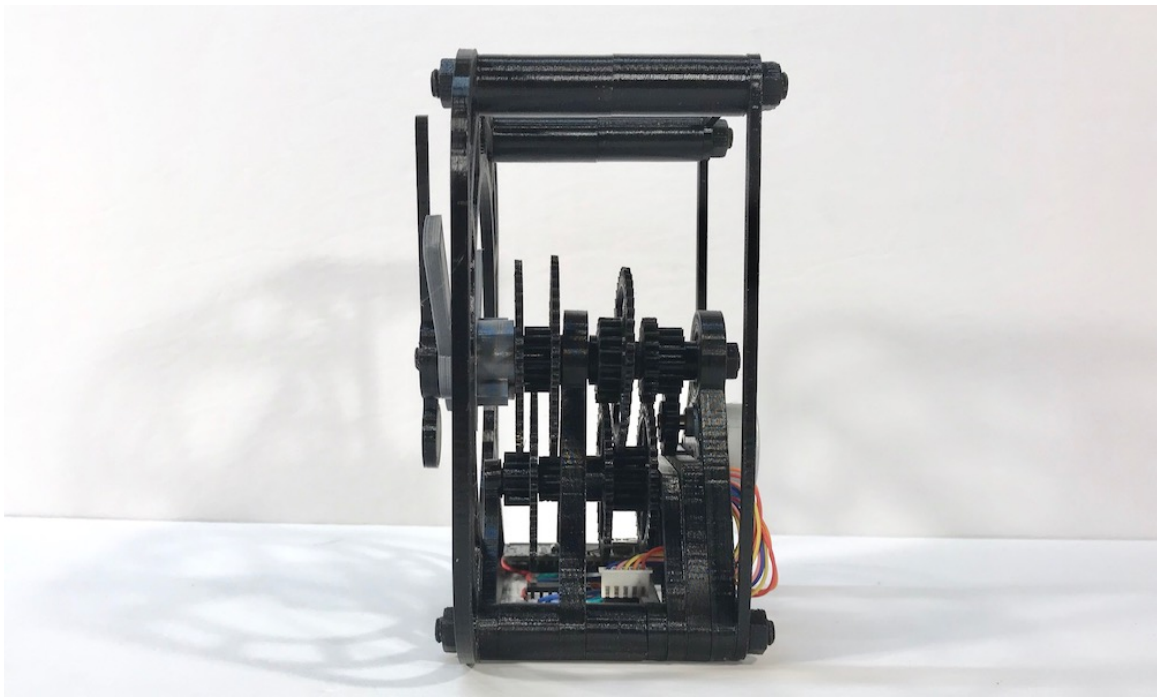
Plug in the motor, install the rear bracket, the studs and spacers



Finally install the front and back face



Done!



The program:

```
/*Jacques Favre with:
http://www.arduino.cc/en/Tutorial/BlinkWithoutDelay
https://learn.adafruit.com/adafruit-arduino-lesson-16-stepper-motors
https://www.arduino.cc/reference/en/language/functions/communication/serial/print/
Driving  a stepper motor to power a clock and print time to console
 Blink a LED
 */
 // stepper library
 #include <Stepper.h>

 // constants won't change. Used here to set a pin number:
const int ledPin =  LED_BUILTIN;// the number of the LED pin
//and for stepper
int in1Pin = 12;
int in2Pin = 11;
int in3Pin = 10;
int in4Pin = 9;

// 32 step motor
Stepper motor(32, in1Pin, in2Pin, in3Pin, in4Pin);

// Variables will change:
int ledState = LOW;           // ledState used to set the LED

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0;       // will store last time LED was updated

// constants won't change:
const long interval = 1000;          // interval at which to blink (milliseconds)

// to print time to console
unsigned long time;

void setup() {
  // set the digital pin as output:
  pinMode(ledPin, OUTPUT);

  // set stepper output
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
```

```
  pinMode(in3Pin, OUTPUT);
  pinMode(in4Pin, OUTPUT);

  //set motor speed
  motor.setSpeed(450);
 Serial.begin(9600);
}

void loop() {
  // here is where you'd put code that needs to be running all the time.

  // check to see if it's time to blink the LED; that is, if the difference
  // between the current time and last time you blinked the LED is bigger than
  // the interval at which you want to blink the LED.
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {

    //print time to console

   Serial.print("Time: ");
  time = millis();
  Serial.println(time);    //prints time since program started

    // save the last time you blinked the LED
    previousMillis = currentMillis;

   //move 32 steps
    motor.step(32);

    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);

  }
}
```