Arduino Clock

Amir Ghorbani

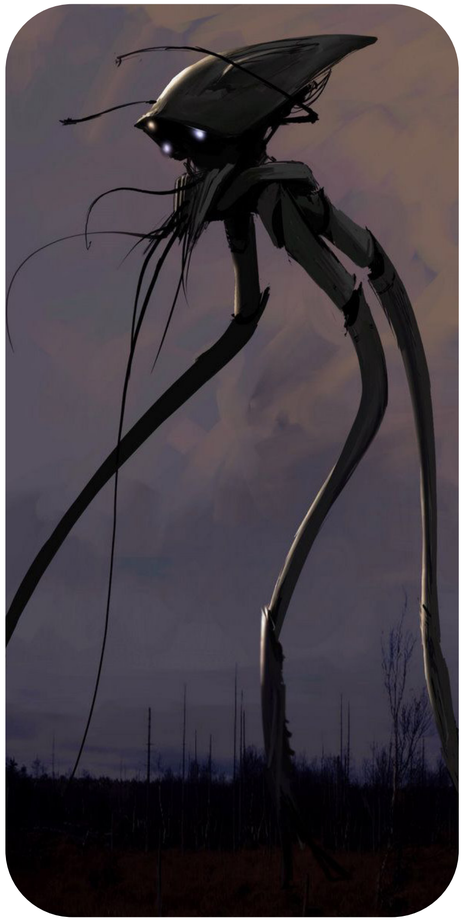
March 2021

Y1 USE18105 Lights, Codes, Making 20-21

Tutor: Nick Rothwell

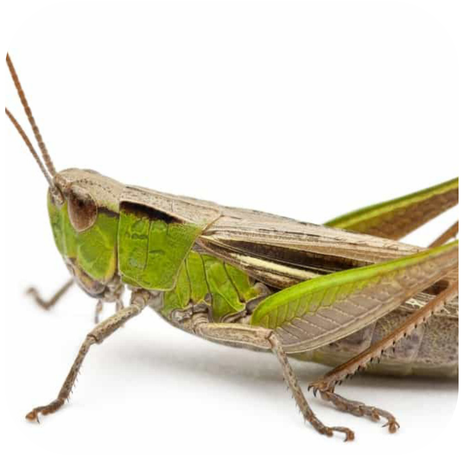
Design Concept












"Intelligent objects are certainly not a product of the 21th century. Since the introduction of solid state transistors in the '60s we've been witnessing the "smartifications" of appliances and toys. Moore's law made possible an exponential advancement in electronics, allowing us to manufacture always cheaper and smaller devices. So why has this term become such a trend topic over the past years? The answer to this question is to be found in tools, rather than technologies.

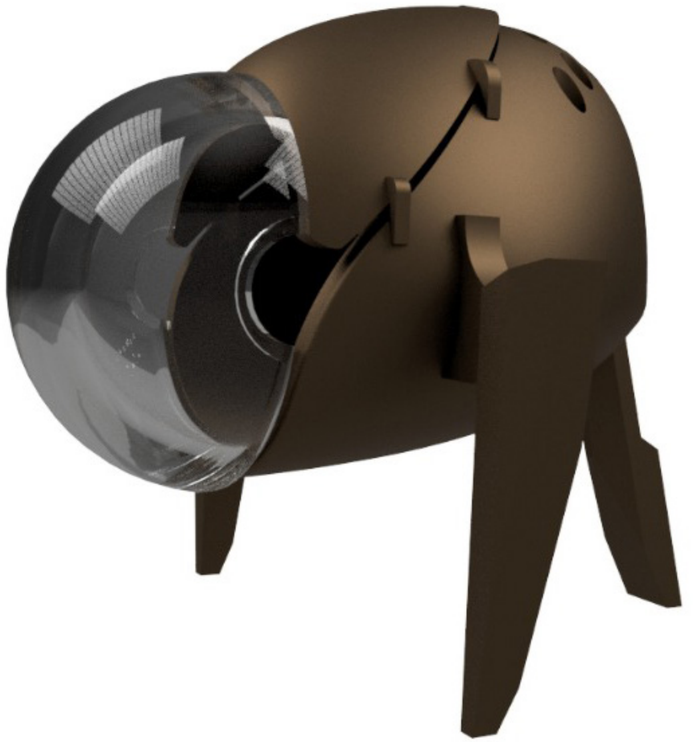
Electronic prototyping platforms such as Arduino, along with the online communities that formed around them, gave creatives, equipment and support for testing their ideas. The subsequent rise of interest in smart objects also encouraged manufacturing companies and design studios to take a chance, and invest in the launch of their own product."

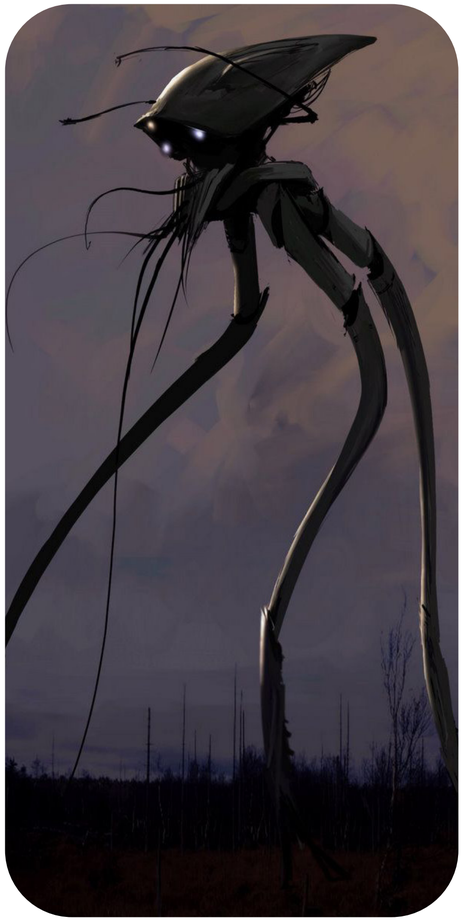
https://medium.com/@orgonomyprod/intelligent-objects-are-certainly-not-a-product-of-the-21th-century-f03f94d7eed1

The Story of “The Bimmer”

Year 2095, The rise of interest in smart physical objects has encouraged a manufacturing company called “Ardu” to invest on its new product “The Bimmer”.

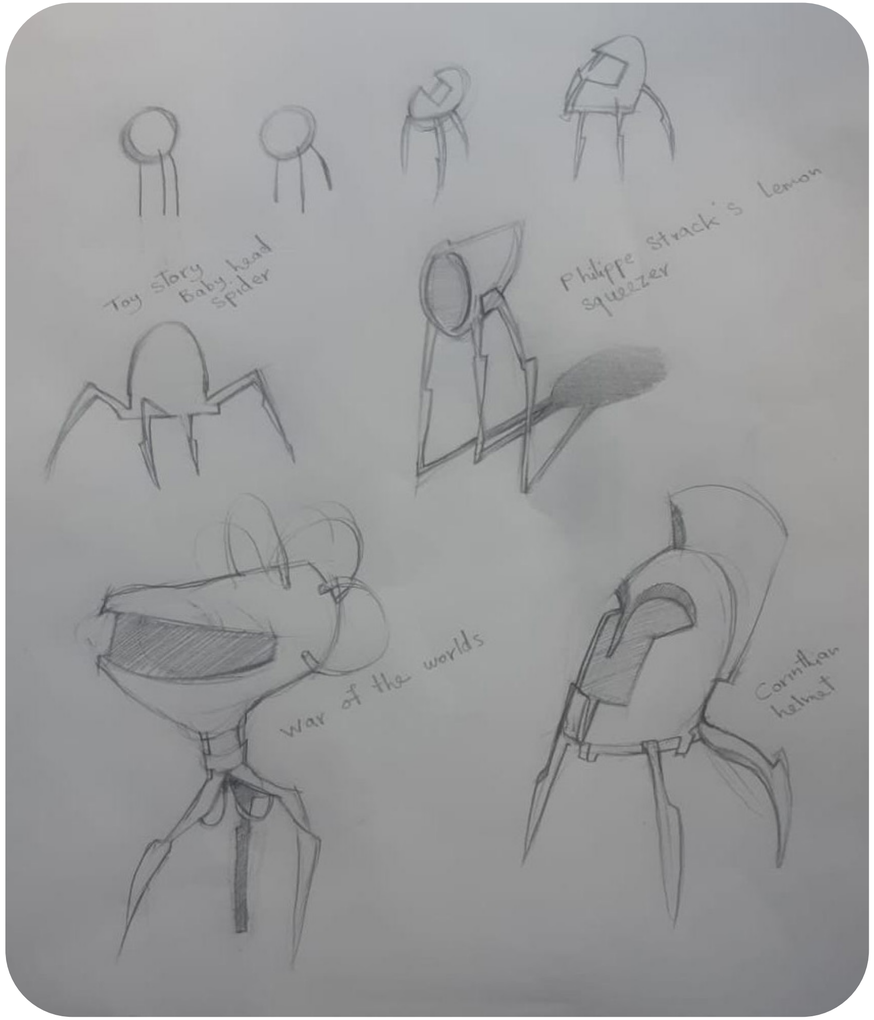
“The Bimmer” is an intelligent carrier. It scans all the electronic components and adjusts its own dimensions to fit the electronics. The Bimmer’s intelligent body parts also transform it to a fighter. It can engage in battle while it is on mission which is carrying all the essential electronic parts.



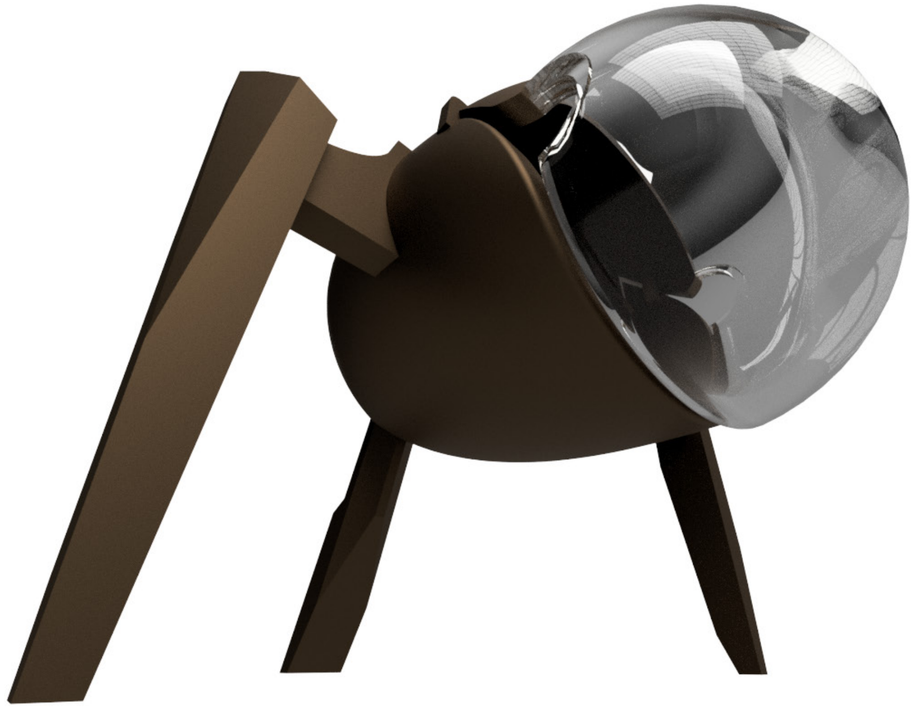


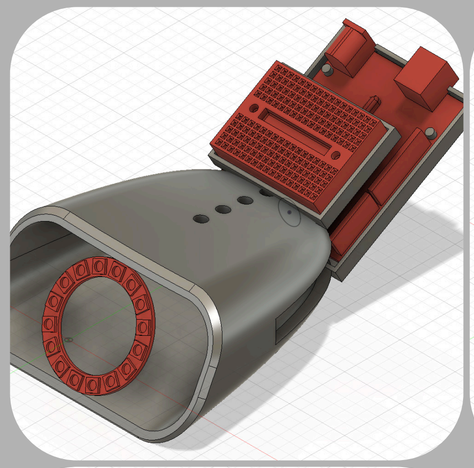
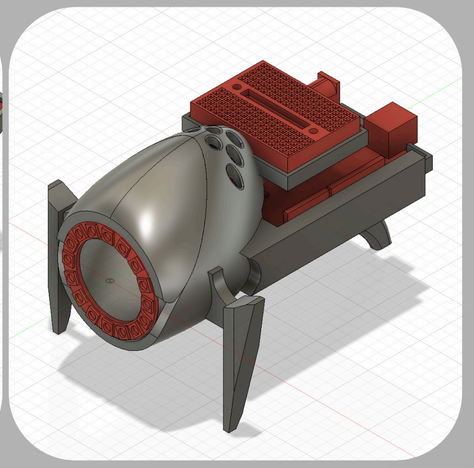
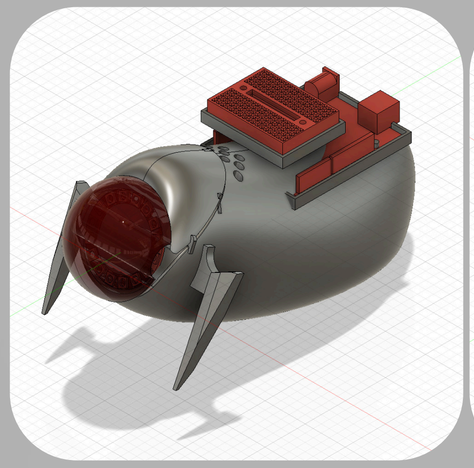
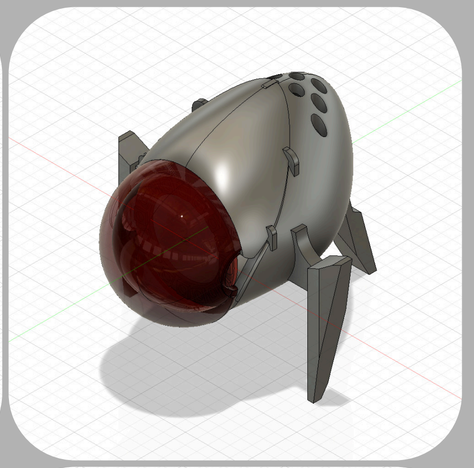
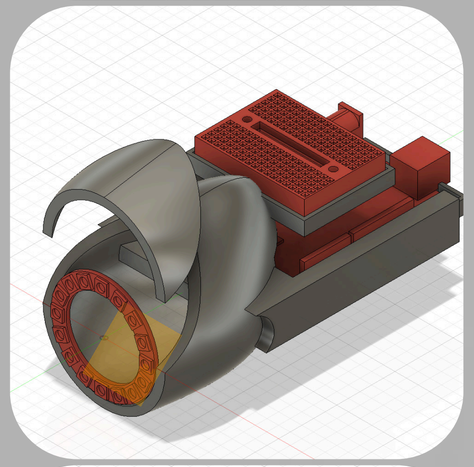
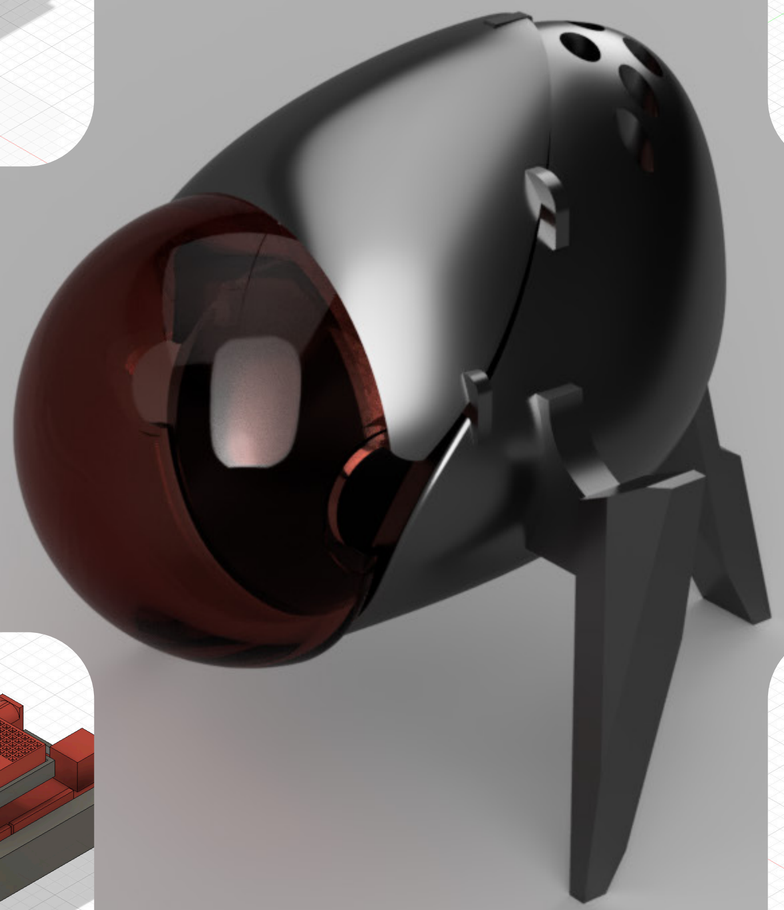
The story of The Bimmer and 3 main images in my mood board have encouraged me to come up with these initial sketches.

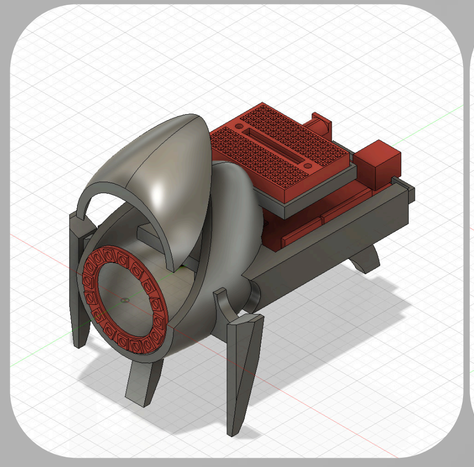
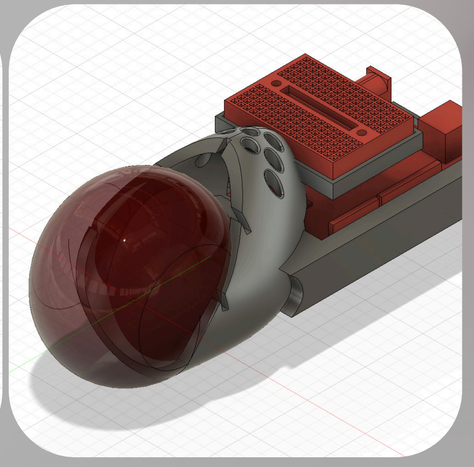
Science Fiction Creatures
Exposed Electronic Components
Artificial Intelligence

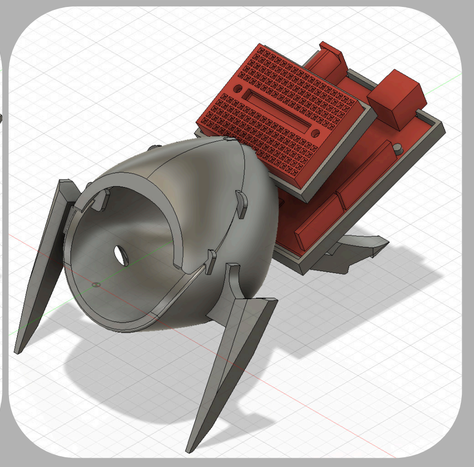


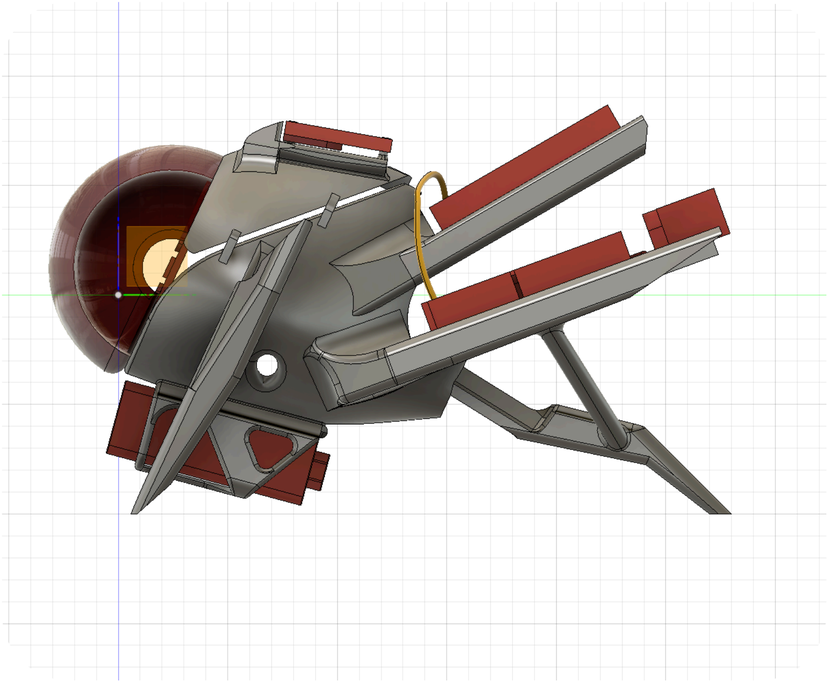
Fusin360

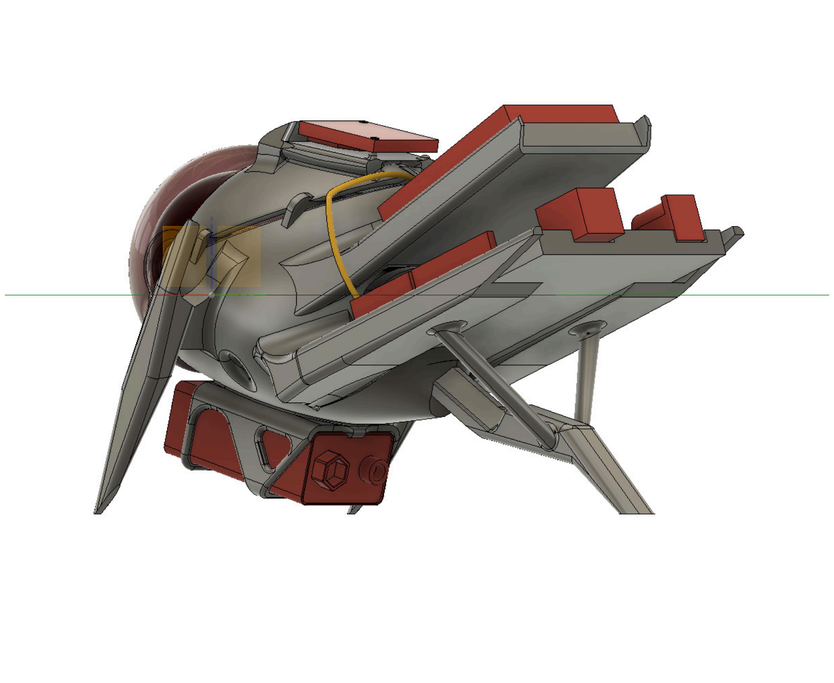


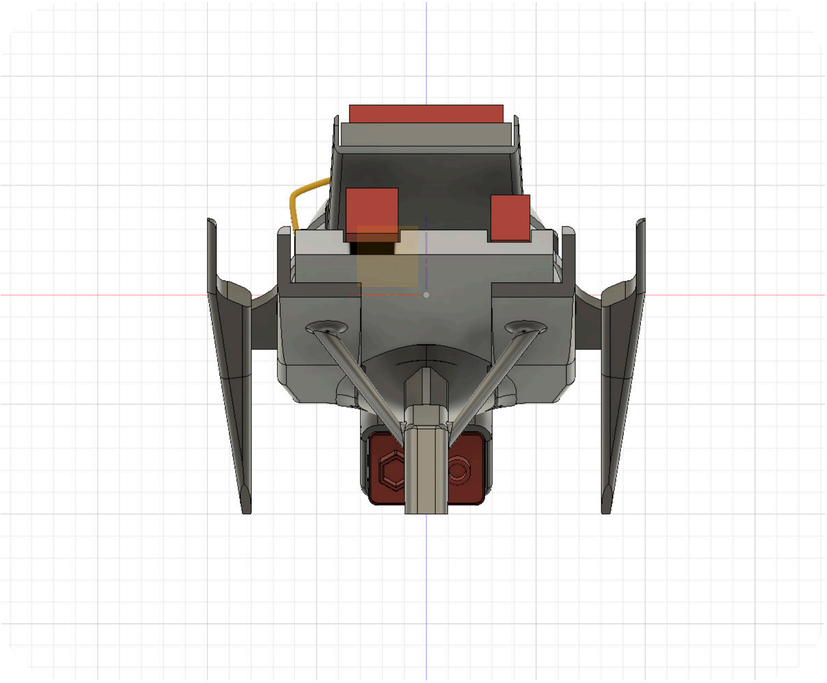







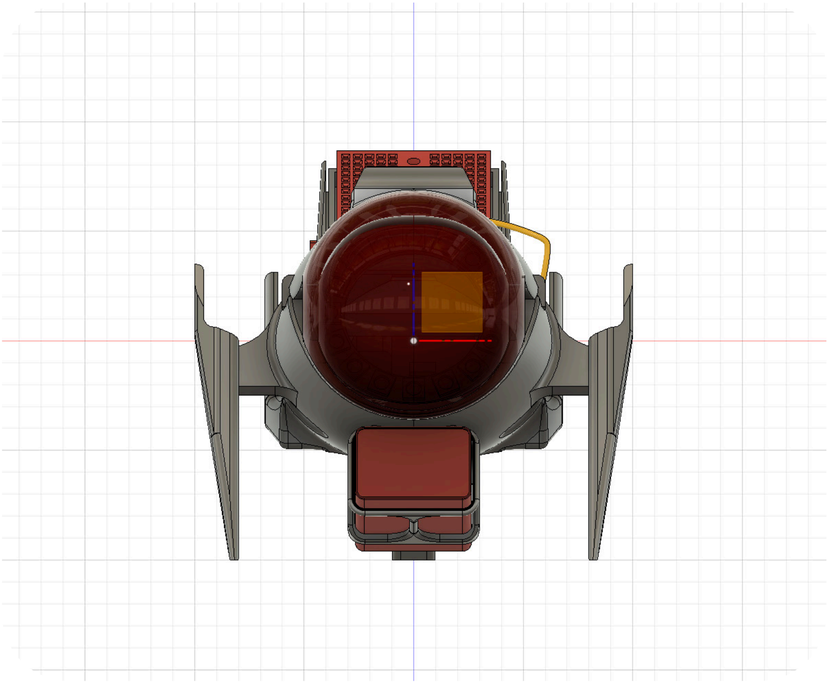



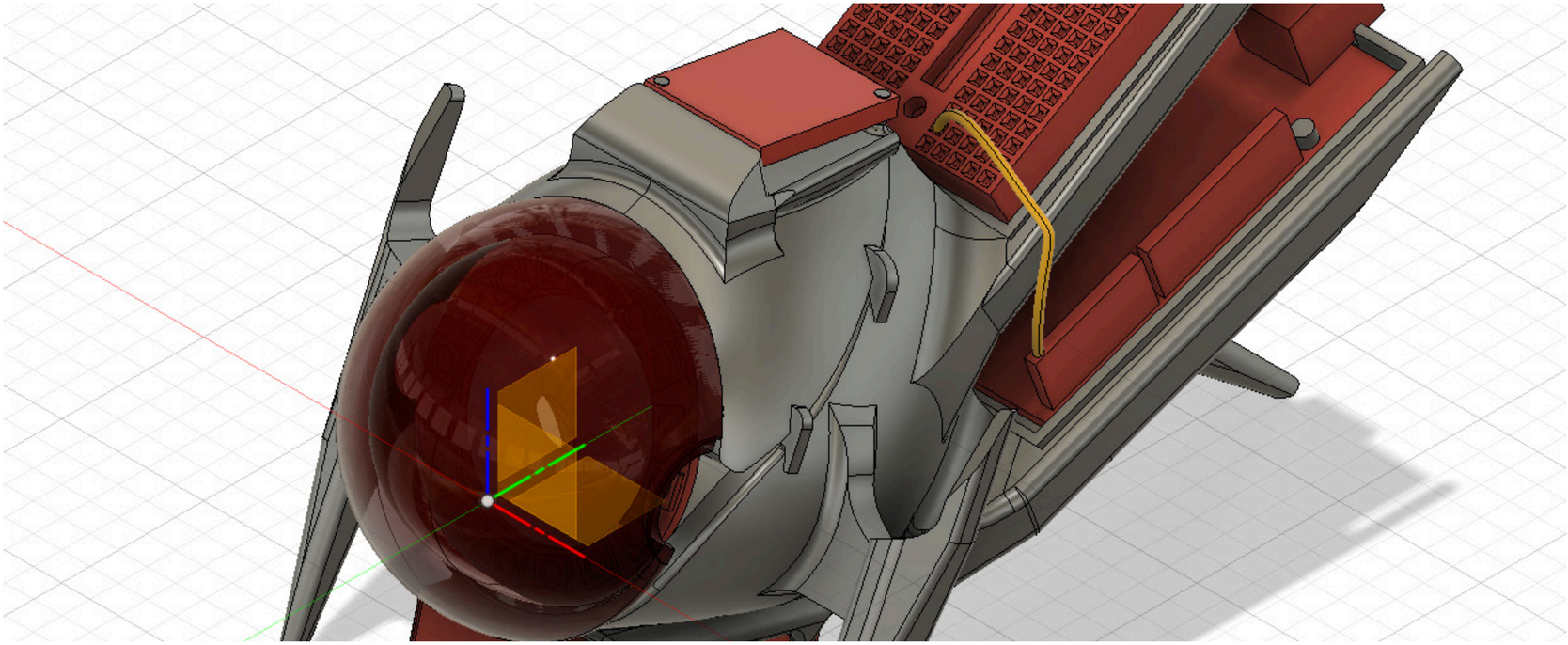


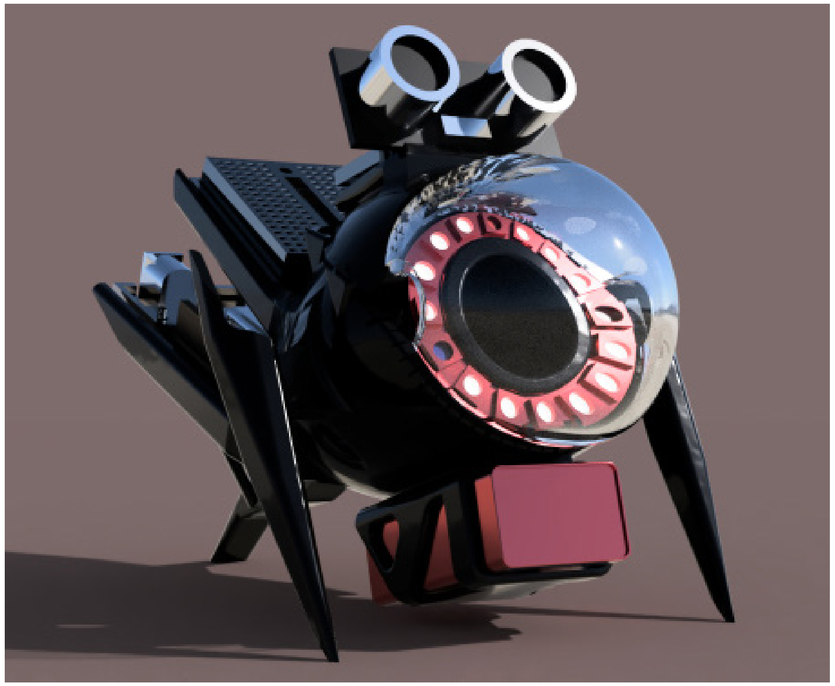


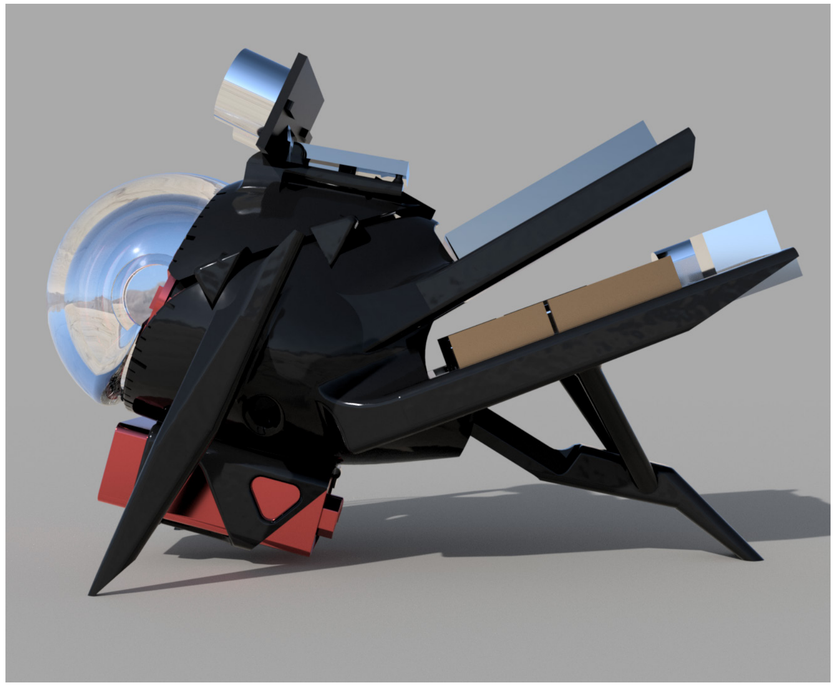


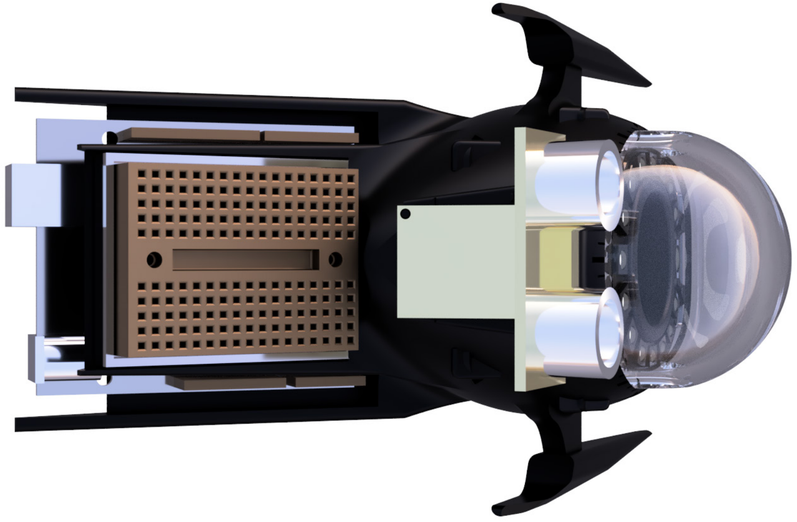




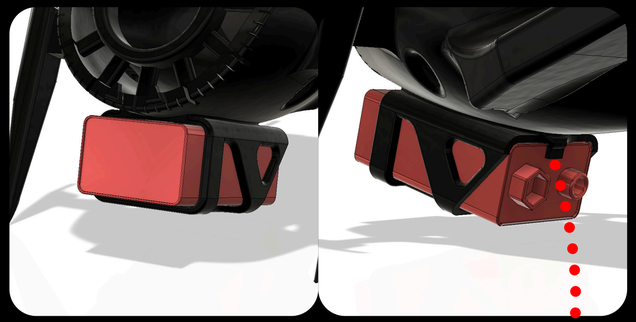








Electronic Component fitting



01 - Battery slides in its cage.

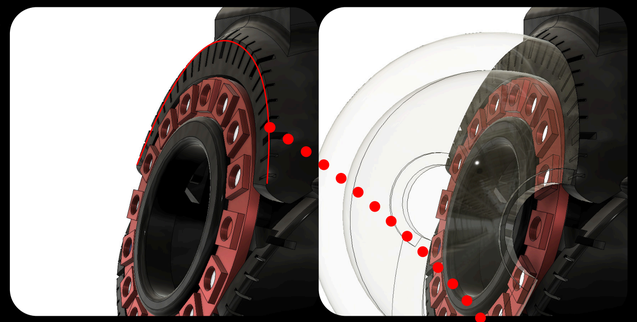
Battery stopper.

02 - NeoPixel fits in around the body ring.

NeoPixel stoppers.

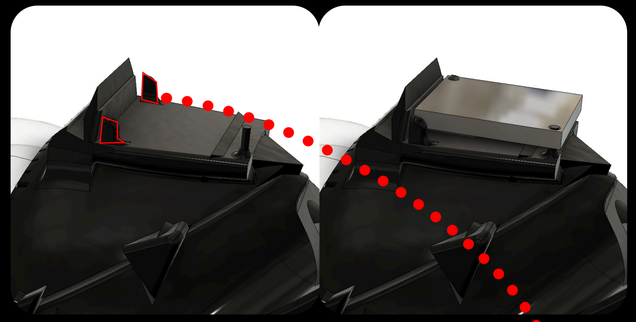
03 - Top part of the body sits in place.

Top body fits inside 5 of these.



04 - The dome attaches to the top body part.

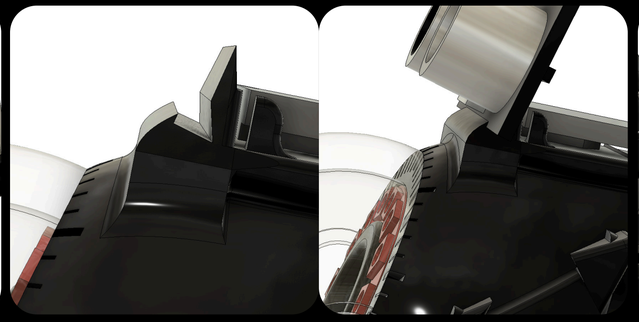
Dome is attached to this face of the top body.



05 - Time board fits in its place.

2 stoppers.

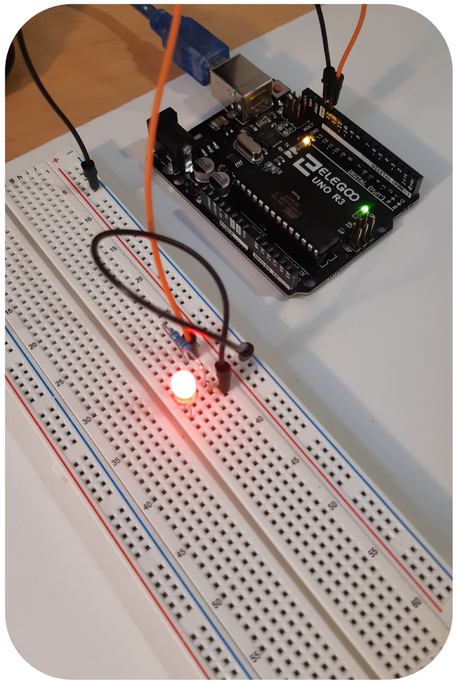
06 - Bread board is inside a case. The case slides in.

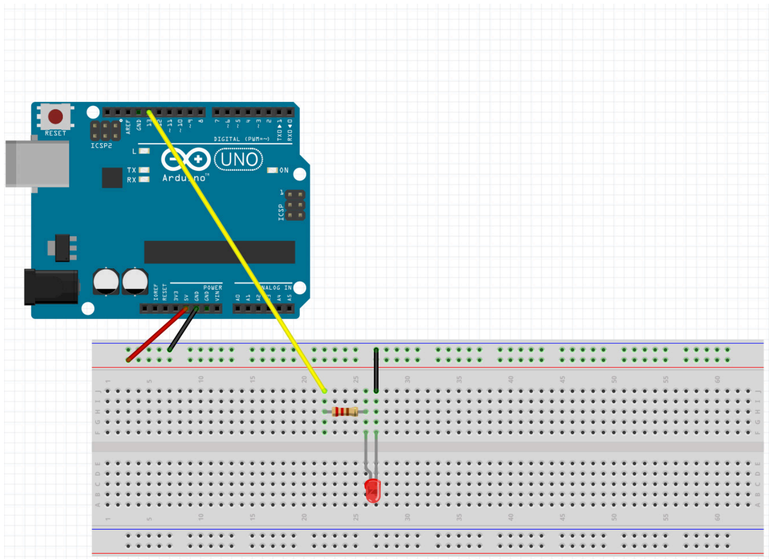


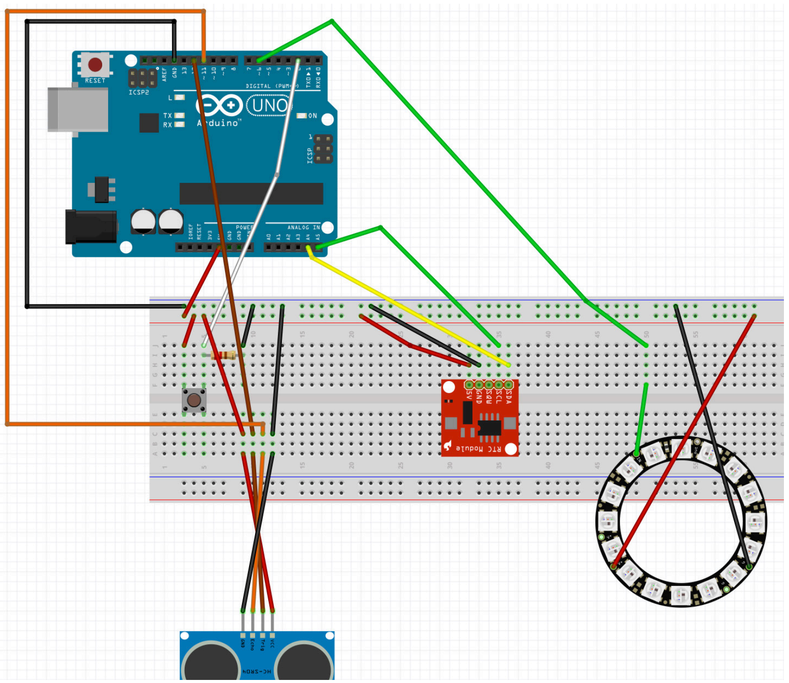
07 - Ultrasonic sensor fits in its place.

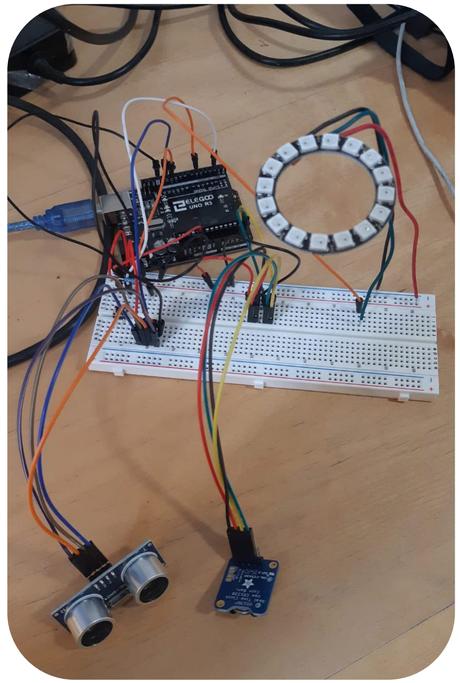
08 - Arduino is inside a case. The case slides in.

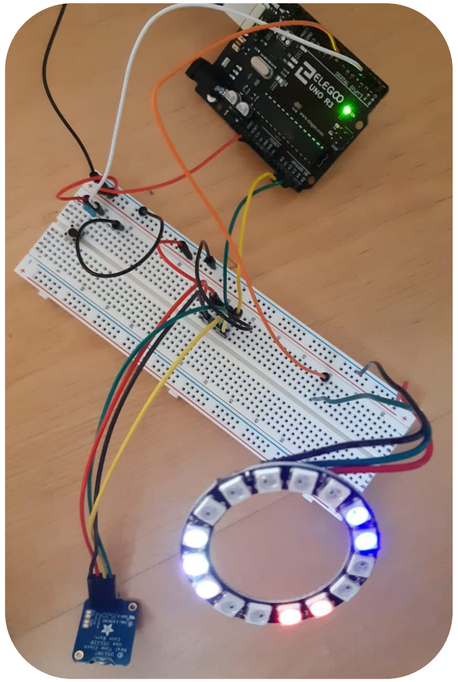
Arduino

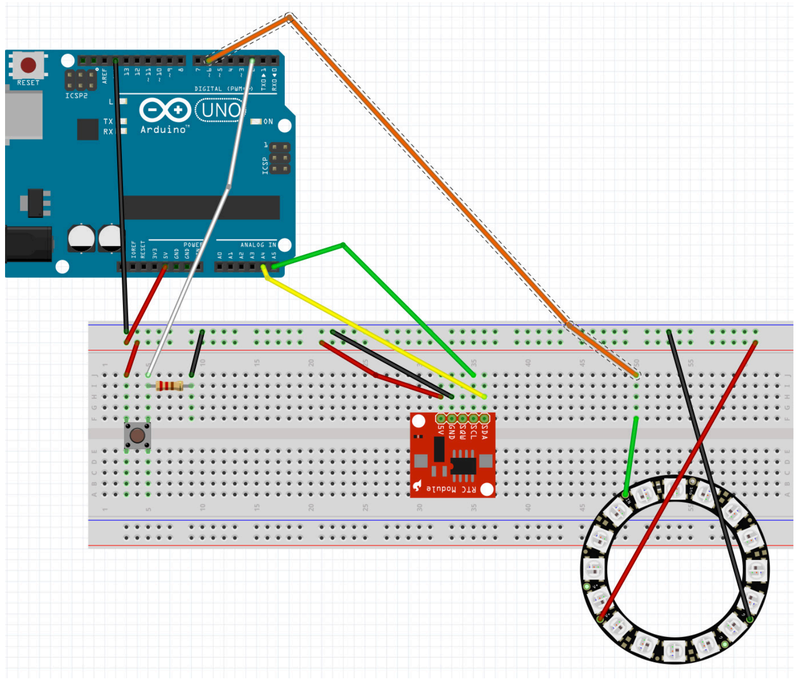


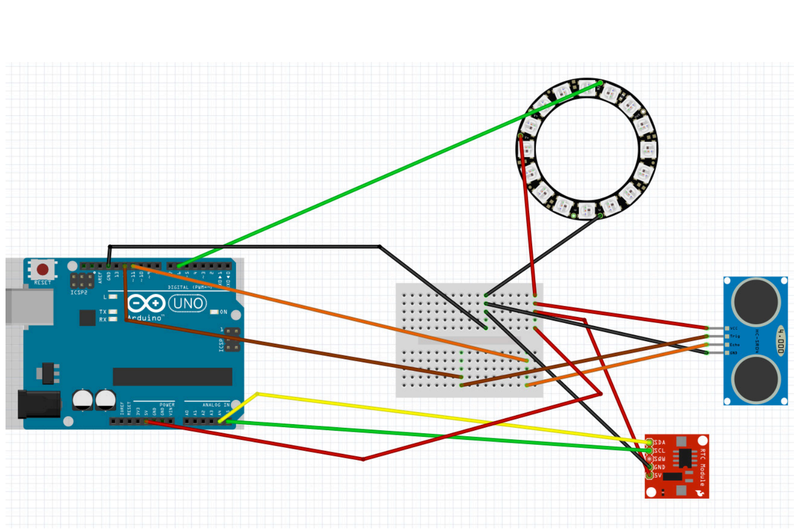


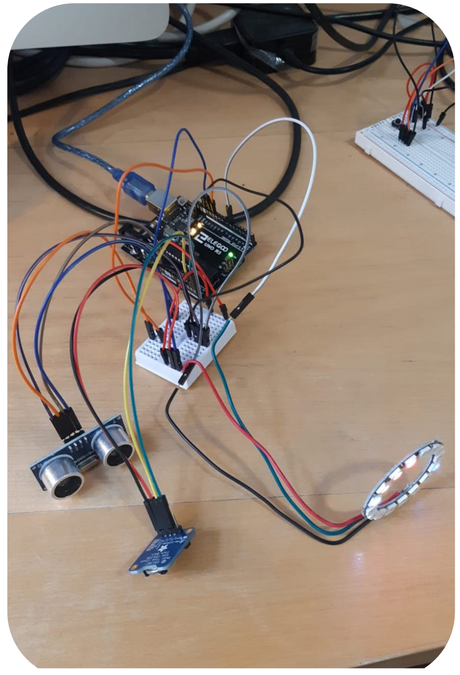


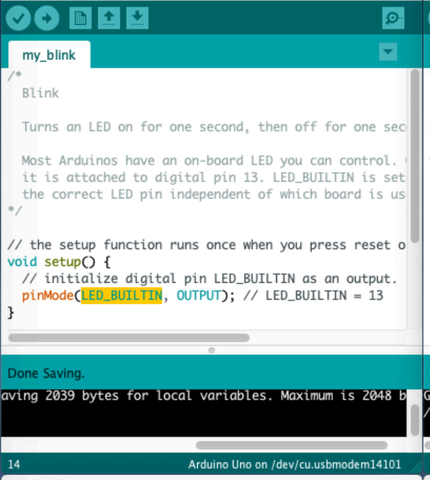


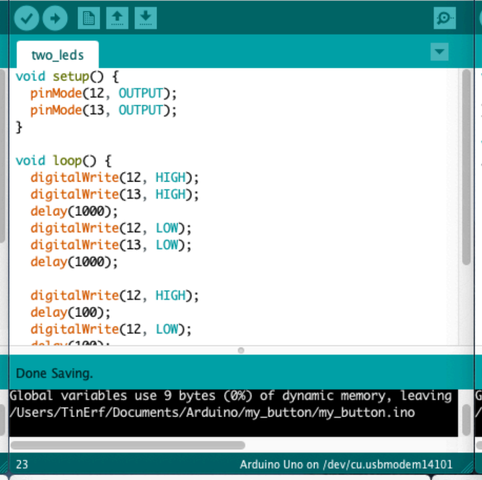


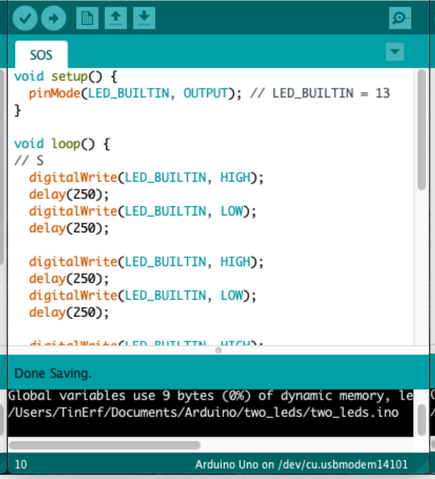


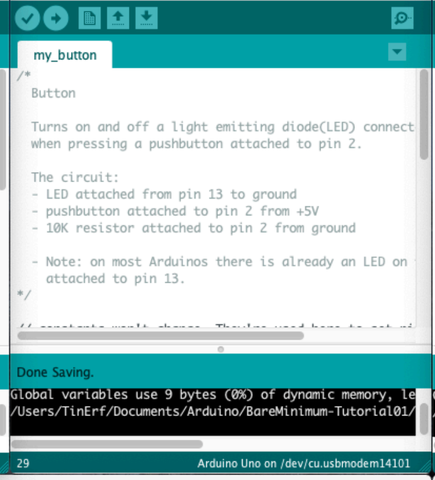


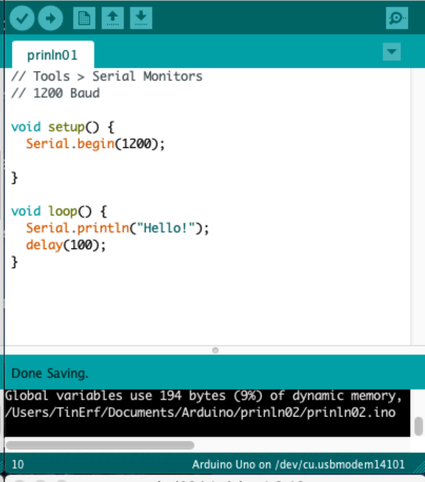












println02 | Arduino 1.8.13

println02

```
// Tools > Serial Monitor
// 1200 Baud

void setup() {
  Serial.begin(1200);
}

void loop() {
  for (int i = 0; i < 100; i++) {
    Serial.print("A");
  }
  Serial.println("Hello!");
  delay(1000);
}
```

Done Saving.

Global variables use 196 bytes (9%) of dynamic memory,
/Users/TinErf/Documents/Arduino/my_button03/my_button03

11 Arduino Uno on /dev/cu.usbmodem14101

my_button03 | Arduino 1.8.13

my_button03

```
const int buttonPin = 2;     // the number of the push
const int ledPin =  13;      // the number of the LED

// variables will change:
int buttonState = LOW;         // variable for reading
int lastButtonState = LOW;


void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);

  Serial.begin(1200);
```

Done uploading.

Global variables use 198 bytes (9%) of dynamic memory,
/Users/TinErf/Documents/Arduino/neopixel01/neopixel01.i

11 Arduino Uno on /dev/cu.usbmodem14101

/dev/cu.usbmodem14101

Send

```
15:51:44.821 -> 53 Second(s) == PixelNumber 14
15:51:47.799 -> ----------
15:51:47.799 -> 15 Hour(s) == PixelNumber 10
15:51:47.799 -> 51 Minute(s) == PixelNumber 13
15:51:47.833 -> 56 Second(s) == PixelNumber 14
15:51:50.814 -> ----------
15:51:50.814 -> 15 Hour(s) == PixelNumber 10
15:51:50.814 -> 51 Minute(s) == PixelNumber 13
15:51:50.814 -> 59 Second(s) == PixelNumber 15
15:51:53.844 -> ----------
15:51:53.844 -> 15 Hour(s) == PixelNumber 10
15:51:53.844 -> 52 Minute(s) == PixelNumber 13
15:51:53.844 -> 2 Second(s) == PixelNumber 0
15:51:56.828 -> ----------
15:51:56.828 -> 15 Hour(s) == PixelNumber 10
15:51:56.862 -> 52 Minute(s) == PixelNumber 13
15:51:56.862 -> 5 Second(s) == PixelNumber 1
15:51:59.853 -> ----------
15:51:59.853 -> 15 Hour(s) == PixelNumber 10
15:51:59.853 -> 52 Minute(s) == PixelNumber 13
15:51:59.888 -> 8 Second(s) == PixelNumber 2
```

Autoscroll Show timestamp Newline 57600 baud Clear output

neopixel01 | Arduino 1.8.13

neopixel01

```
#include <Adafruit_NeoPixel.h>

Adafruit_NeoPixel strip(16, 6, NEO_GRB + NEO_KHZ800);

void setup() {
    strip.begin();
    strip.setBrightness(10);
}

void loop() {
    strip.clear();

    strip.setPixelColor(0, 255, 0, 0);

    strip.show();
}
```

Done compiling.

Global variables use 41 bytes (2%) of dynamic memory, l
/Users/TinErf/Documents/Arduino/neopixel02/neopixel02.i

13 Arduino Uno on /dev/cu.usbmodem14101

neopixel02 | Arduino 1.8.13

neopixel02

```
void setup() {
    strip.begin();
    strip.setBrightness(10);
}

void loop() {
    strip.clear();

    for(int i = 0; i < 16; i++) {
      if(i % 4 == 0) {
        strip.setPixelColor(i, 255, 0, 0);
      } else {
        strip.setPixelColor(i, 0, 100, 0);
      }
    }
```

Done Saving.

Global variables use 41 bytes (2%) of dynamic memory, l
/Users/TinErf/Documents/Arduino/my_clock_ds1307_01/my_c

7 Arduino Uno on /dev/cu.usbmodem14101

button_neopixel | Arduino 1.8.13

button_neopixel

```
  strip.begin();
  strip.setBrightness(10);

  Serial.begin(9600);
}

void loop() {
  strip.clear();

  Serial.println(millis()/1000);

  int pixelNumber = (millis()/1000) % 16;

  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
```

Done compiling.

Global variables use 220 bytes (10%) of dynamic memory,
/Users/TinErf/Documents/Arduino/button_neopixel02/butto

button_neopixel02 | Arduino 1.8.13

button_neopixel02

```
#include <Adafruit_NeoPixel.h>

Adafruit_NeoPixel strip(16, 6, NEO_GRB + NEO_KHZ800);


// constants won't change. They're used here to set pin numb
const int buttonPin = 2;     // the number of the pushbutton
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = LOW;         // variable for reading the p
int lastButtonState = LOW;


void setup() {
```

Done Saving.

Global variables use 41 bytes (2%) of dynamic memory, leaving
/Users/TinErf/Documents/Arduino/my_strandtest01/my_strandtest

sketch_feb15b_05 | Arduino 1.8.13

sketch_feb15b_05

```
}

void singlePixel(int pixelNo, uint32_t color) {
  // uint32_t is a type of value that can represent an entire colour
  strip.clear();
  strip.setPixelColor(pixelNo, color);
  strip.show();
}

void black() {
  strip.clear();
  strip.show();
}

void loop() {
```

/Users/TinErf/Documents/Arduino/clock_plus_neo_01/clock_plus_neo_01.ino

myclock_plus_neo_01 | Arduino 1.8.13

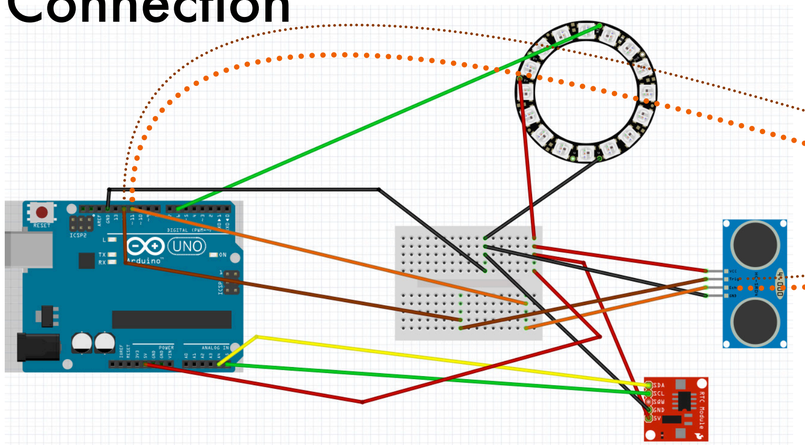
myclock_plus_neo_01

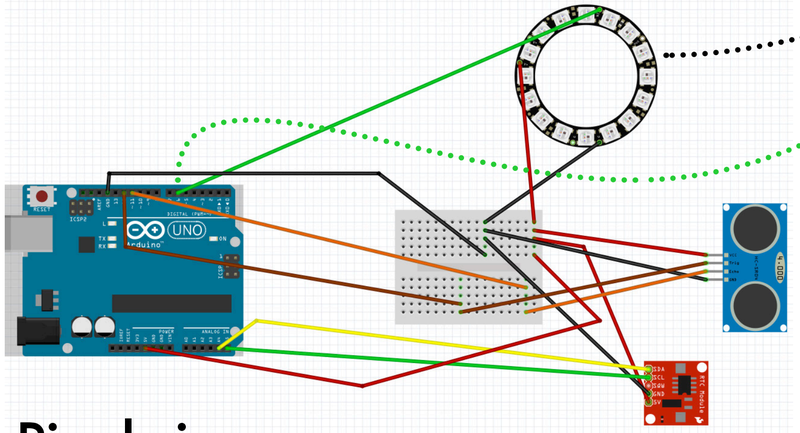
```
  strip.clear();

  int hPixel = map(h, 0, 24, 0, 16);
  Serial.print("----------");
  Serial.println();
  Serial.print(h, DEC);
  Serial.print(" Hour(s) == PixelNumber ");
  Serial.print(hPixel, DEC);
  Serial.println();

  int minPixel = map(m, 0, 60, 0, 16);
  Serial.print(m, DEC);
  Serial.print(" Minute(s) == PixelNumber ");
  Serial.print(minPixel, DEC);
  Serial.println();
```

Done Saving.

Sketch uses 7482 bytes (23%) of program storage space. Maximum is 32256 bytes.
Global variables use 584 bytes (28%) of dynamic memory, leaving 1464 bytes for local variables. Maximum is 2048 bytes.

Ultrasonic Connection





Pixel ring Connection

16 LEDs

```
1  // Ultrasonic variables//
2  int trigPin=12;
3  int echoPin=11;
4  int pingTravelTime; // How long does it take for the ping to go from the sensor hit the target and return.
5  float pingTravelDistance;
6  int intDistanceToTarget;
7  float distanceToTarget;
8
9  // Neo Pixel //
10 #include <Adafruit_NeoPixel.h>
11 int LED_COUNT = 16;
12 int LED_PIN = 6;
13 Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);
14
15 // Date and time functions using a DS1307 RTC connected via I2C and Wire lib
16 #include "RTClib.h"
17 RTC_DS1307 rtc;
18
19 void setup () {
44
45 void myTimeHMS () {
64 void myTimeMillis() {
91 void ultrasonic () {
107
108 void loop () {
```

Time board

Showing hour, minute and second LEDs

Sensor function

An LED goes one round of pixel ring every second

setup () {

```
19 void setup () {
20
21   Serial.begin(57600);
22
23   // Ultrasonic pins////////////////////////////////
24   pinMode(trigPin,OUTPUT);
25   pinMode(echoPin,INPUT);
26
27   // RTC //
28   #ifndef ESP8266
29   while (!Serial); // wait for serial port to connect. Needed for native USB
30   #endif
31   if (! rtc.begin()) {
32     Serial.println("Couldn't find RTC");
33     Serial.flush();
34     abort();
35   }
36   if (! rtc.isrunning()) {
37     Serial.println("RTC is NOT running, let's set the time!");
38     rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
39   }
40
41   // Neo Pixel ////////////////////////////////////
42   strip.begin();
43 }
```

Pin 12

Pin 11

Time board

myTimeHMS () {

```
45 void myTimeHMS () {
46   DateTime now = rtc.now();
47
48   int s = now.second();
49   int m = now.minute();
50   int h = now.hour();
51
52   strip.clear();
53   strip.setBrightness(50);
54   // Second // Mapping seconds to 16 LEDs // Every 60 seconds = 1 round of NeoPixel
55   int secPixel = map(s, 0, 60, 0, 16);
56   strip.setPixelColor(secPixel, strip.Color(255, 0, 0));
57   // Minute // Mapping minutes to 16 LEDs // Every 60 minutes = 1 round of NeoPixel
58   int minPixel = map(m, 0, 60, 0, 16);
59   strip.setPixelColor(minPixel, strip.Color(255, 255, 255));
60   // Hour // Mapping hours to 16 LEDs // Every 12 hour = 1 round of NeoPixel
61   int hPixel = map(h%12, 0, 12, 0, 16);
62   strip.setPixelColor(hPixel, strip.Color(0, 0, 255));
63   strip.show();
64 }
```

- Set the brightness of all LEDs to 50
- Map: 60 seconds = 1 round of Pixel ring
- Red LED shows the second
- Map: 60 minutes = 1 round of Pixel ring
- White LED shows the minute
- Map: 12 hours = 1 round of Pixel ring
- Blue LED shows the hour

myTimeMillis () {

```
65 void myTimeMillis() {
66   // Milli // The red LED goes one round every second and when it hits the second, minute and hour  LEDs it blinks
67   DateTime now = rtc.now();
68   int s = now.second();
69   int m = now.minute();
70   int h = now.hour();
71   int secPixel = map(s, 0, 60, 0, 16);
72   int minPixel = map(m, 0, 60, 0, 16);
73   int hPixel = map(h%12, 0, 12, 0, 16);
74
75   strip.clear();
76   int milPixel = map(millis(), 0, 1000, 0, 16);
77
78   // When fast LED hit the second_LED the brightness changes to 255
79   if (milPixel%16 == secPixel) {
80     strip.setBrightness(255);
81     strip.setPixelColor(milPixel%16, 255, 0, 0);
82     // When fast LED hit the minute_LED the brightness changes to 255
83     } else if (milPixel%16 == minPixel) {
84       strip.setBrightness(255);
85       strip.setPixelColor(milPixel%16, 255, 255, 255);
86       // When fast LED hit the hour_LED the brightness changes to 255
87     } else if (milPixel%16 == hPixel) {
88       strip.setBrightness(255);
89       strip.setPixelColor(milPixel%16, 0, 0, 255);
90     } else {
91       // Normally the fast LED's brightness is 50
92       strip.setBrightness(50);
93       strip.setPixelColor(milPixel%16, 255, 0, 0);
94     }
95     strip.show();
96 }
```

- Map: 1 second = 1 round of Pixel ring
- If fast LED hit the Second_LED brightness changes to 255 and its colour is red
- If fast LED hit the Minute_LED brightness changes to 255 and its colour is white
- If fast LED hit the Hour_LED brightness changes to 255 and its colour is blue
- White LED shows the minute
- Otherwise the brightness of fast LED is 50 and its colour is red

ultrasonic () {

```
97 void ultrasonic () {
98   // Sending and listening
99   digitalWrite(trigPin,LOW); // Launch an ultrasonic signal it will come out it will hit the target.
100  delayMicroseconds(10);
101  digitalWrite(trigPin,HIGH);
102  delayMicroseconds(10);
103  digitalWrite(trigPin,LOW);
104  pingTravelTime = pulseIn(echoPin,HIGH);
105  pingTravelDistance=(pingTravelTime*34.3/1000); // Speed of sound = 1234.8 km/h = 34.3 cm/ms
106  distanceToTarget=pingTravelDistance/2;
107  intDistanceToTarget = round(distanceToTarget);
108
109  Serial.print("Distance to Target is: ");
110  Serial.print(intDistanceToTarget);
111  Serial.println(" cm.");
112 }
```

This launch an ultrasonic signal,
it will come out,
it will hit the target,
it will come back

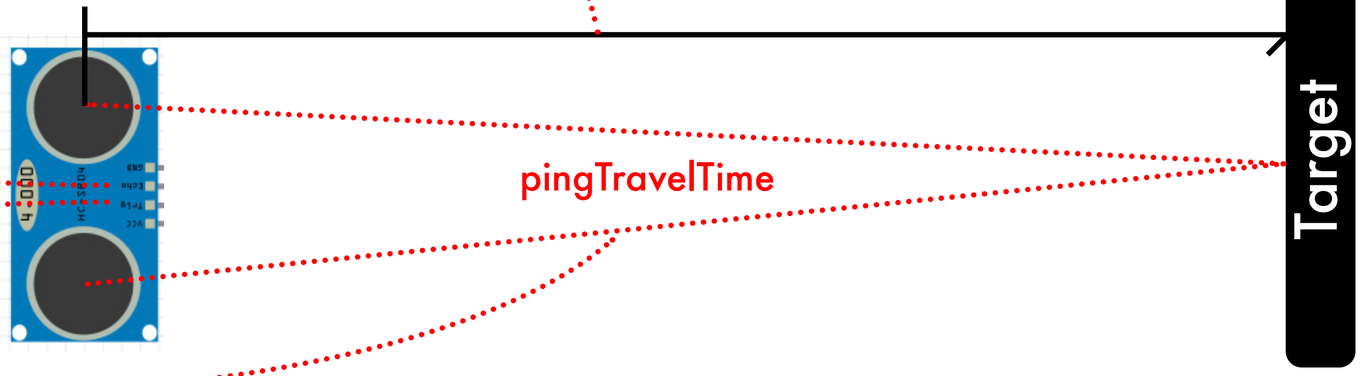
Put the trigPin LOW for 10 ms

Put the trigPin HIGH for 10 ms

Put the trigPin LOW again

Speed = Distance / time
Distance = Speed * time

We only need half of pingTravelDistance



loop () {

```
114 void loop () {
115   ultrasonic (); // My function
116
117   int maxDistance = 30;
118   int minDistance = 5;
119   if (intDistanceToTarget > maxDistance) {
120     myTimeHMS(); // My function
121     myTimeMillis(); // My function
122   } else {
123     strip.clear();
124     if (intDistanceToTarget <= minDistance) {
125       for (int i = 0; i <=16; i++) {
126         strip.setPixelColor(i, strip.Color(255, 0, 0));
127       }
128       // When an object is within 5 to 30 cm distance of the Ultrasonic the neoPixel reacts //////////
129     } else if(intDistanceToTarget <= maxDistance & intDistanceToTarget > minDistance) {
130       int sonicPixel = map(intDistanceToTarget, maxDistance, minDistance, 0, 16);
131       strip.setPixelColor(sonicPixel, strip.Color(255, 255, 255));
132     }
133     strip.show();
134     }
135 }
```

Start reading distance to any target

If the distance to target > 30cm

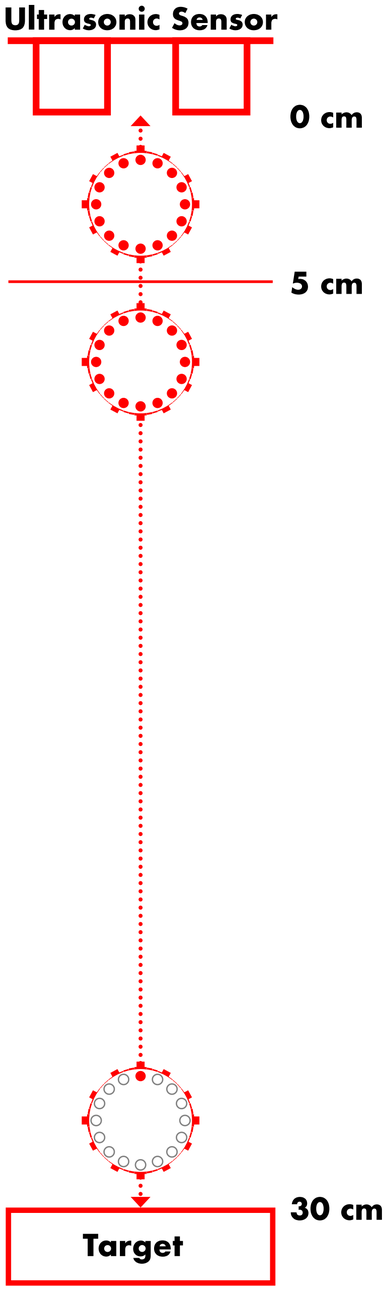
These two functions will run

If the distance to target < 5cm
All 16 LEDs will be red

But if the distance to target >= 5cm
and <= 30cm

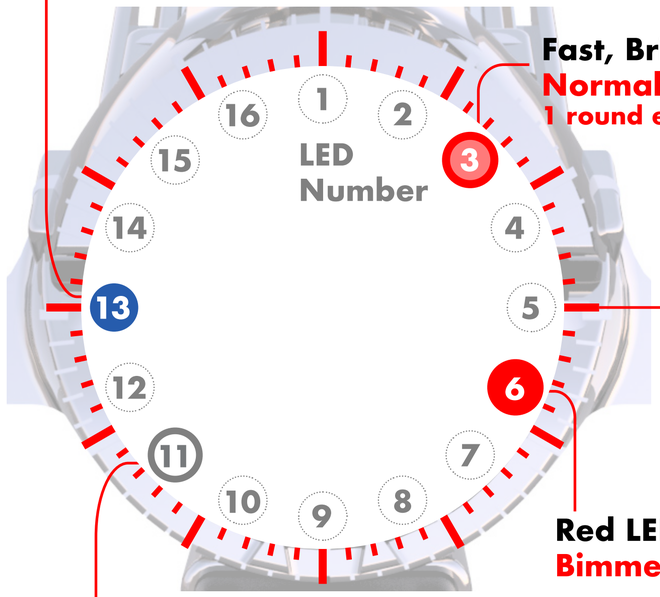
Map: (30cm - 5cm) to (LED 0 - LED 16)
1 white LED will be mapped

Manual



Blue LED Shows
Bimmer Hour Counter

Example:
The blue LED Number is 13.
The Hour is between 9h to 9.75h.



Fast, Bright, Red LED shows
Normal Second Counter
1 round every 1 second

Use the clock indicator to read the time quickly

Red LED shows
Bimmer Second Counter

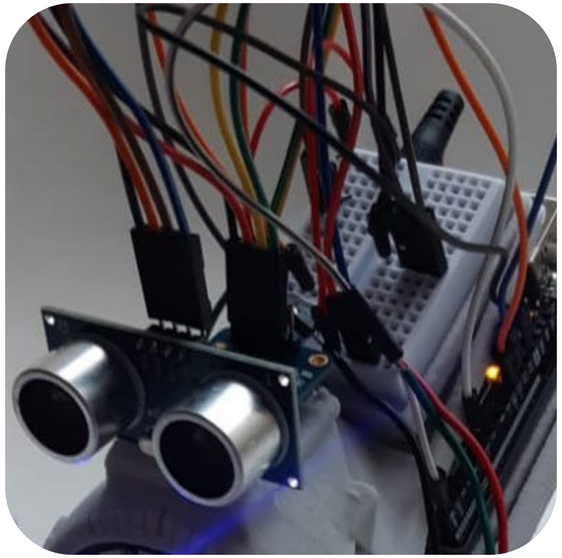
Example:
The red slow LED Number is 6.
The Second is between 18.750s to 22.500s.

White LED shows
Bimmer Minute Counter

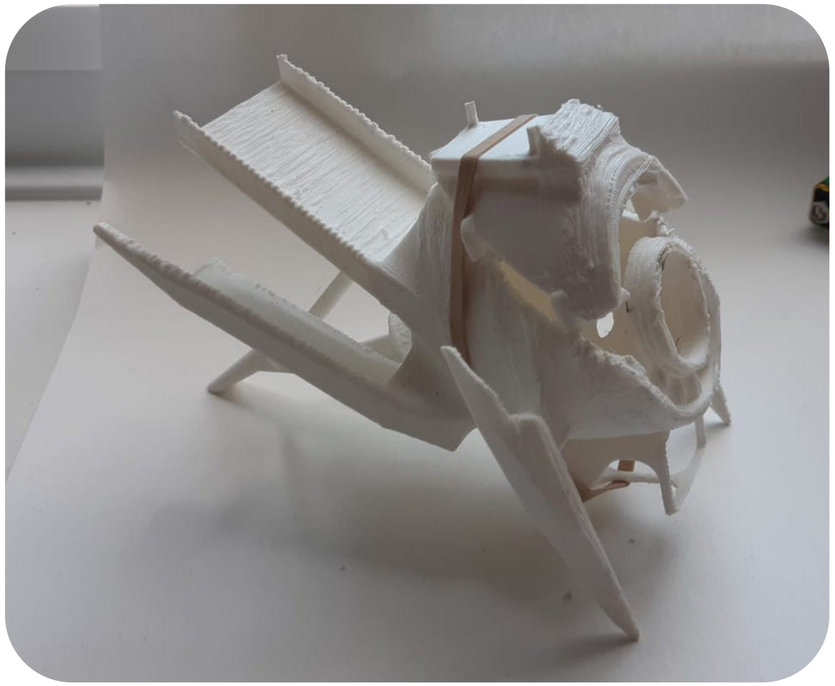
Example:
The white LED Number is 11.
The Minute is between 37.500m to 41.250m.

LED Number	Time (hours)
1	Between 0 - 0.75
2	0.75 - 1.5
3	1.5 - 2.25
4	2.25 - 3
5	3 - 3.75
6	3.75 - 4.5
7	4.5 - 5.25
8	5.25 - 6
9	6 - 6.75
10	6.75 - 7.5
11	7.5 - 8.25
12	8.25 - 9
13	9 - 9.75
14	9.75 - 10.5
15	10.5 - 11.25
16	11.25 - 0

LED Number	Time (seconds or minutes)
1	Between 0 - 3.750
2	3.750 - 7.500
3	7.500 - 11.250
4	11.250 - 15
5	15 - 18.750
6	18.750 - 22.500
7	22.500 - 26.250
8	26.250 - 30
9	30 - 33.750
10	33.750 - 37.500
11	37.500 - 41.250
12	41.250 - 45
13	45 - 48.750
14	48.75 - 52.500
15	52.500 - 56.250
16	56.250 - 0



3d Print



First round of 3d printing (Plastic)

First round of 3d printing is to check the dimensions, fittings and more refinements.

I changed the dimension of the Aduino case and breadboard case and also added more parts to hold the Pixel ring and Ultrasonic sensor.

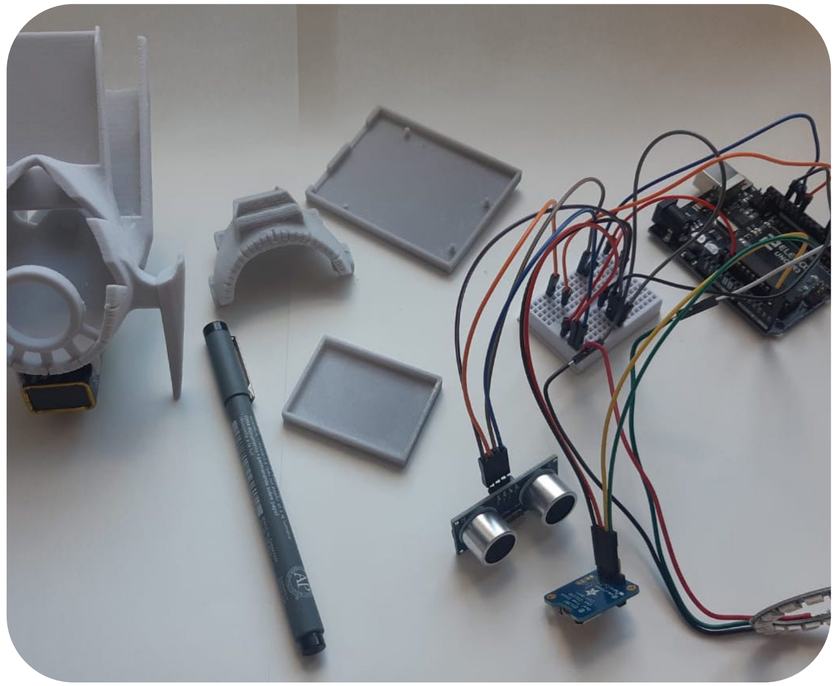
Second round of 3d printing (Glue and Powder)

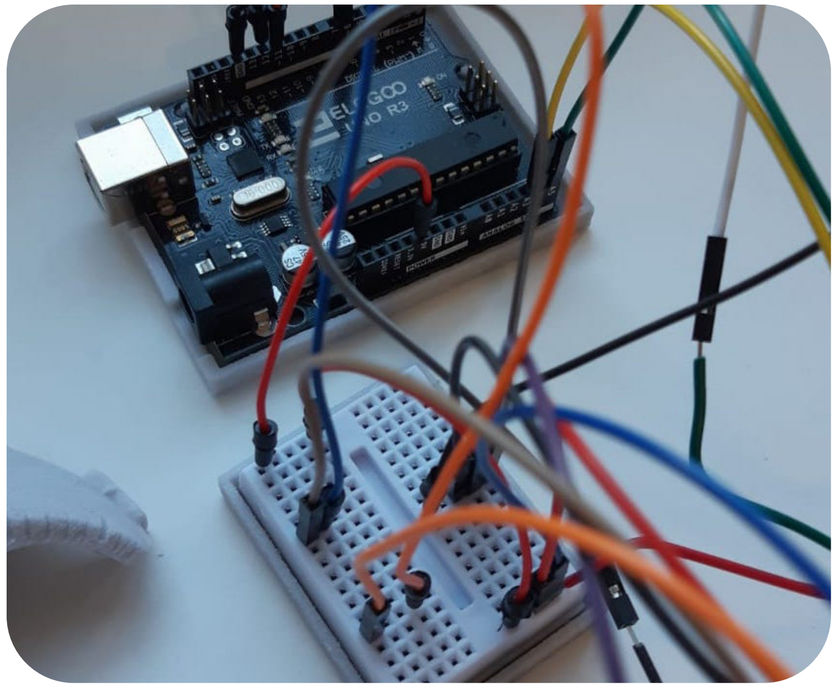
The second round of 3d print is more satisfying. The Electronic parts are fitting in their places.

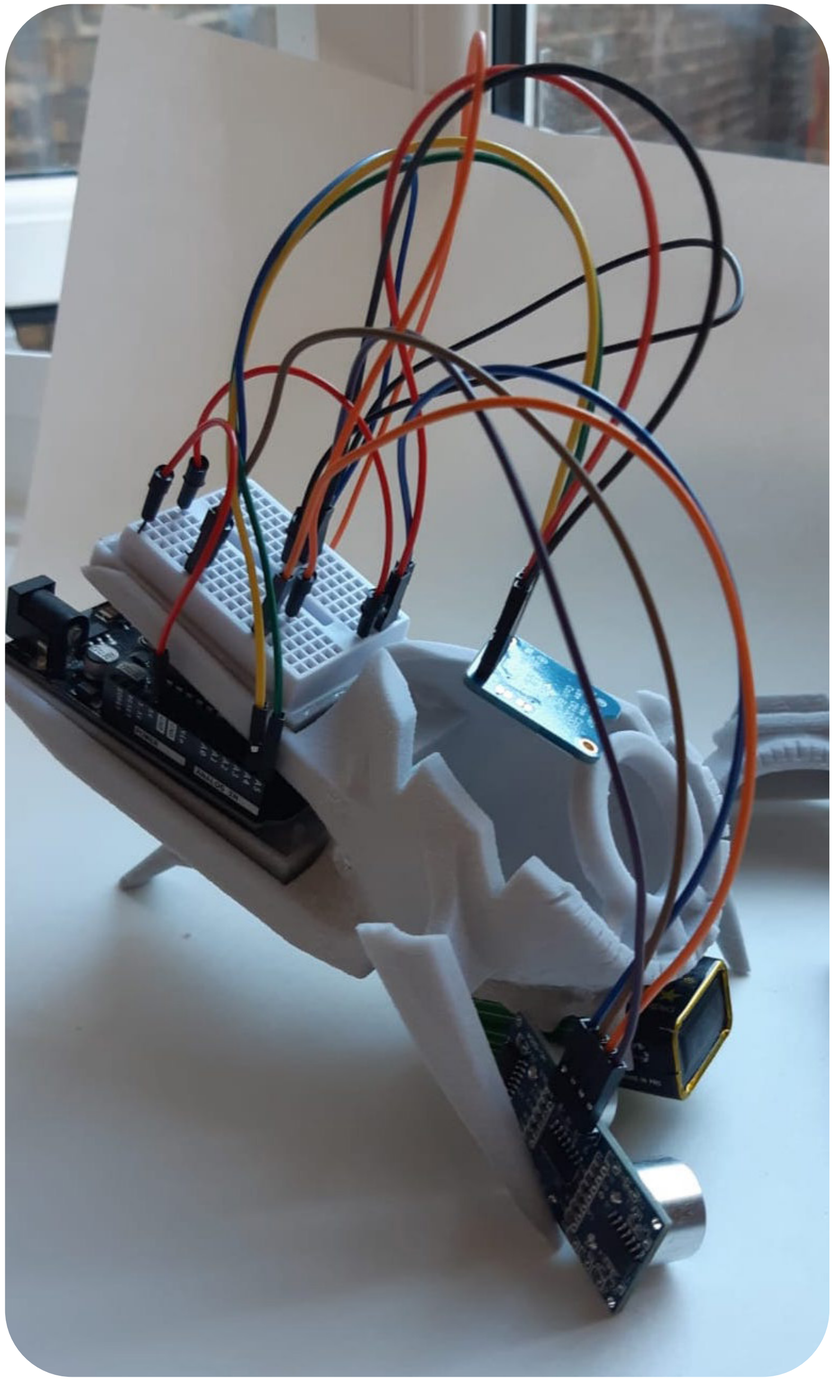
Two broken parts:
1. The battery case
2. internal Pixel ring holder.

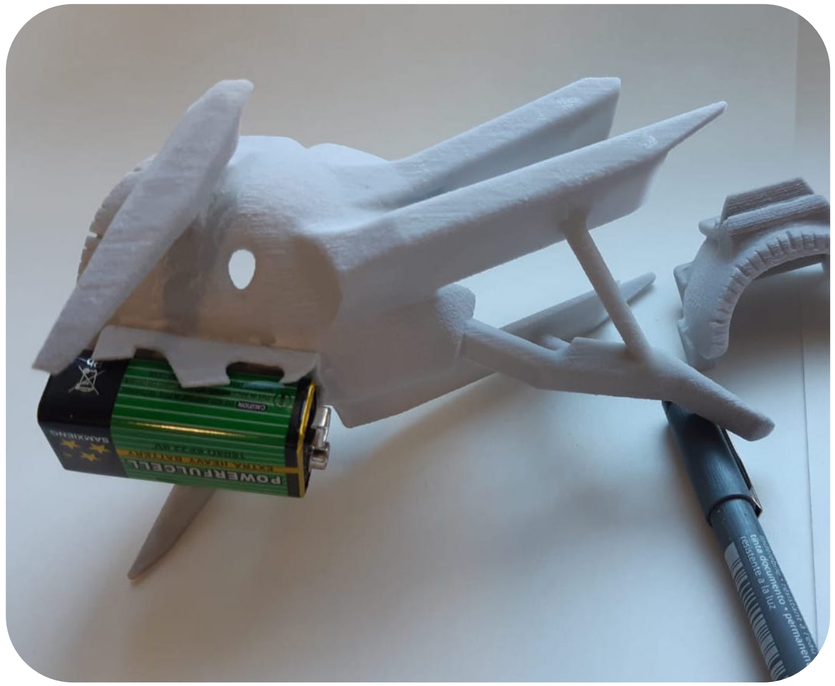
I was able to manage the broken parts and fit the electronics, as it shows on following images.

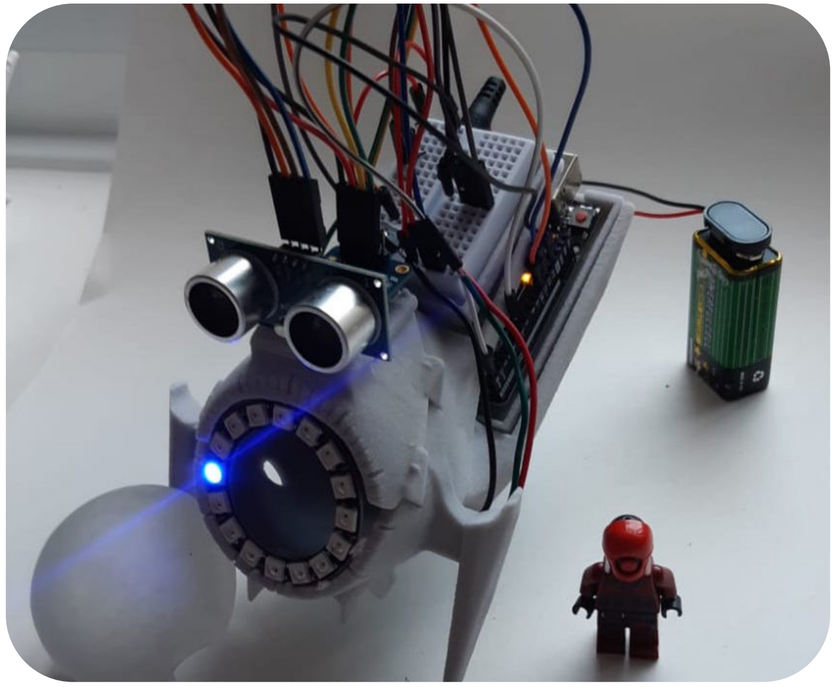
I have attached the related video to this pdf file.

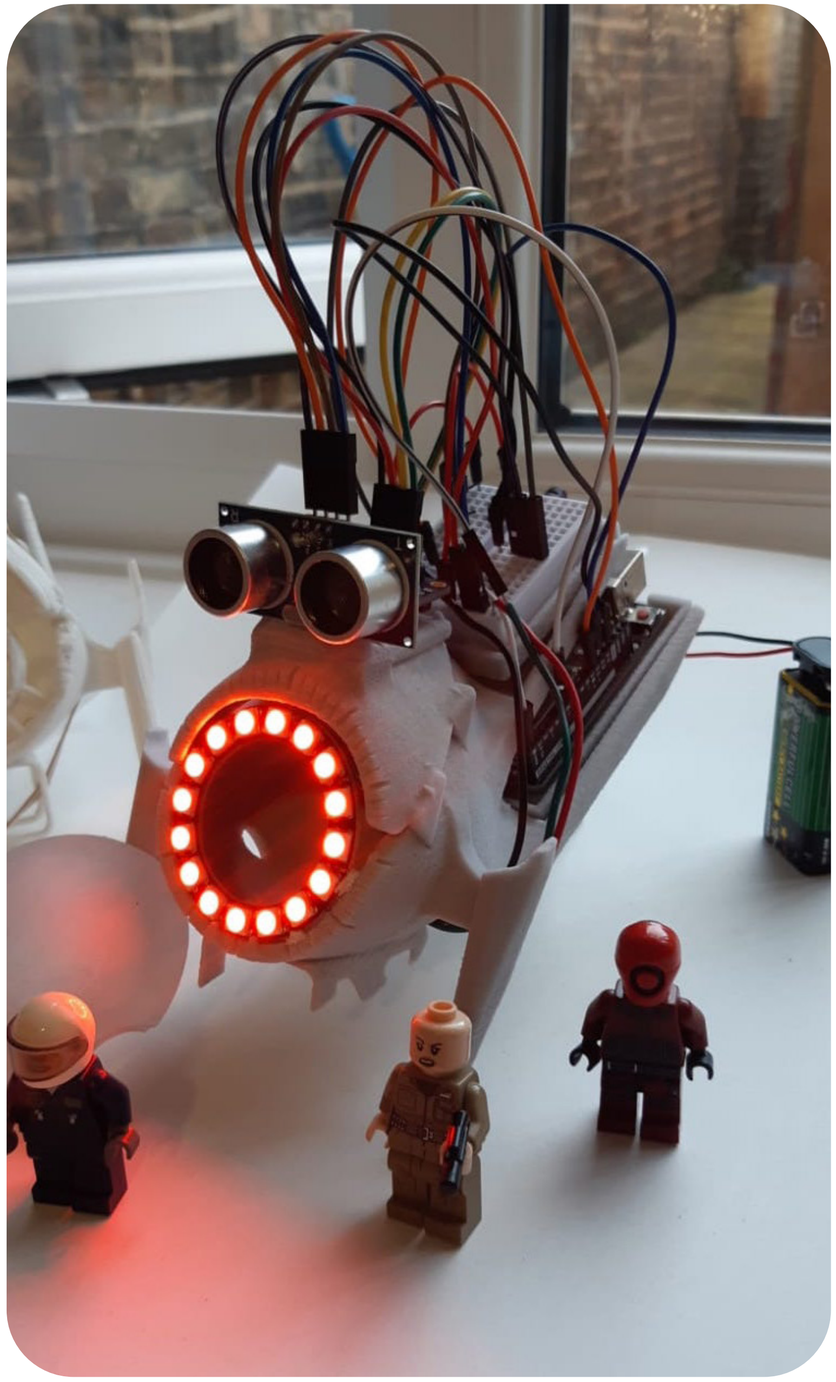




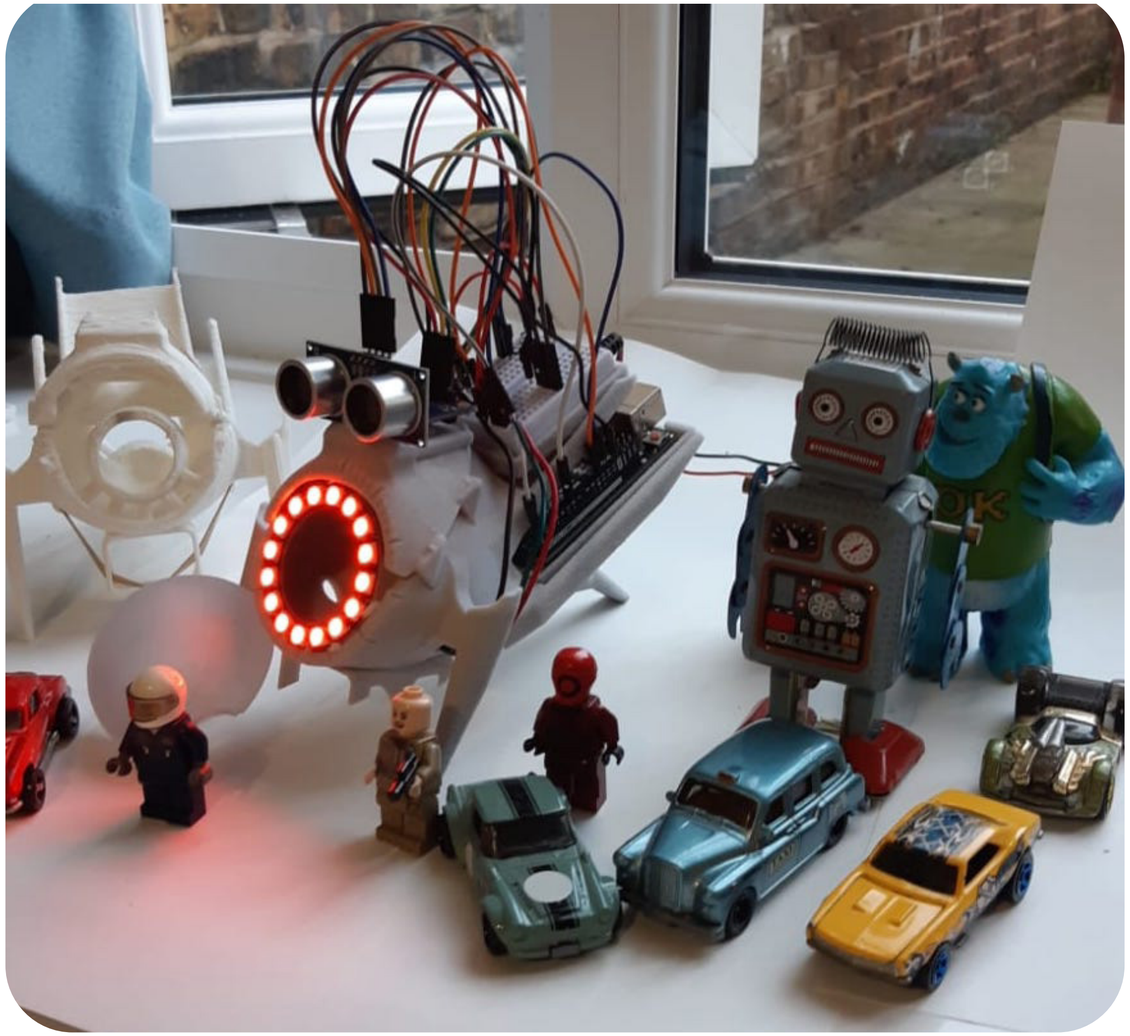



POWERFULCELL





Thank you



Arduino Clock

Amir Ghorbani

March 2021

Y1 USE18105 Lights, Codes, Making 20-21

Tutor: Nick W