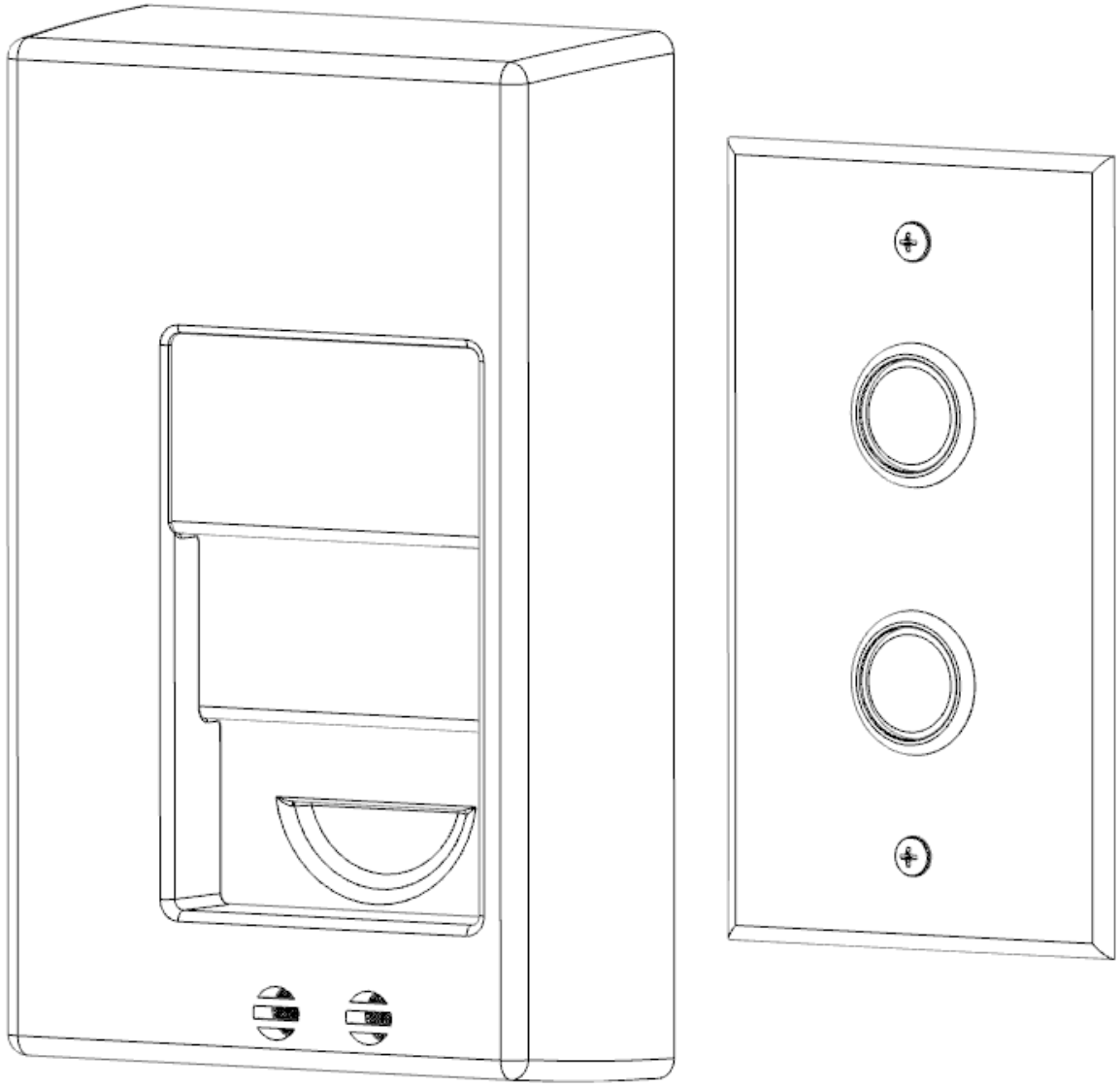
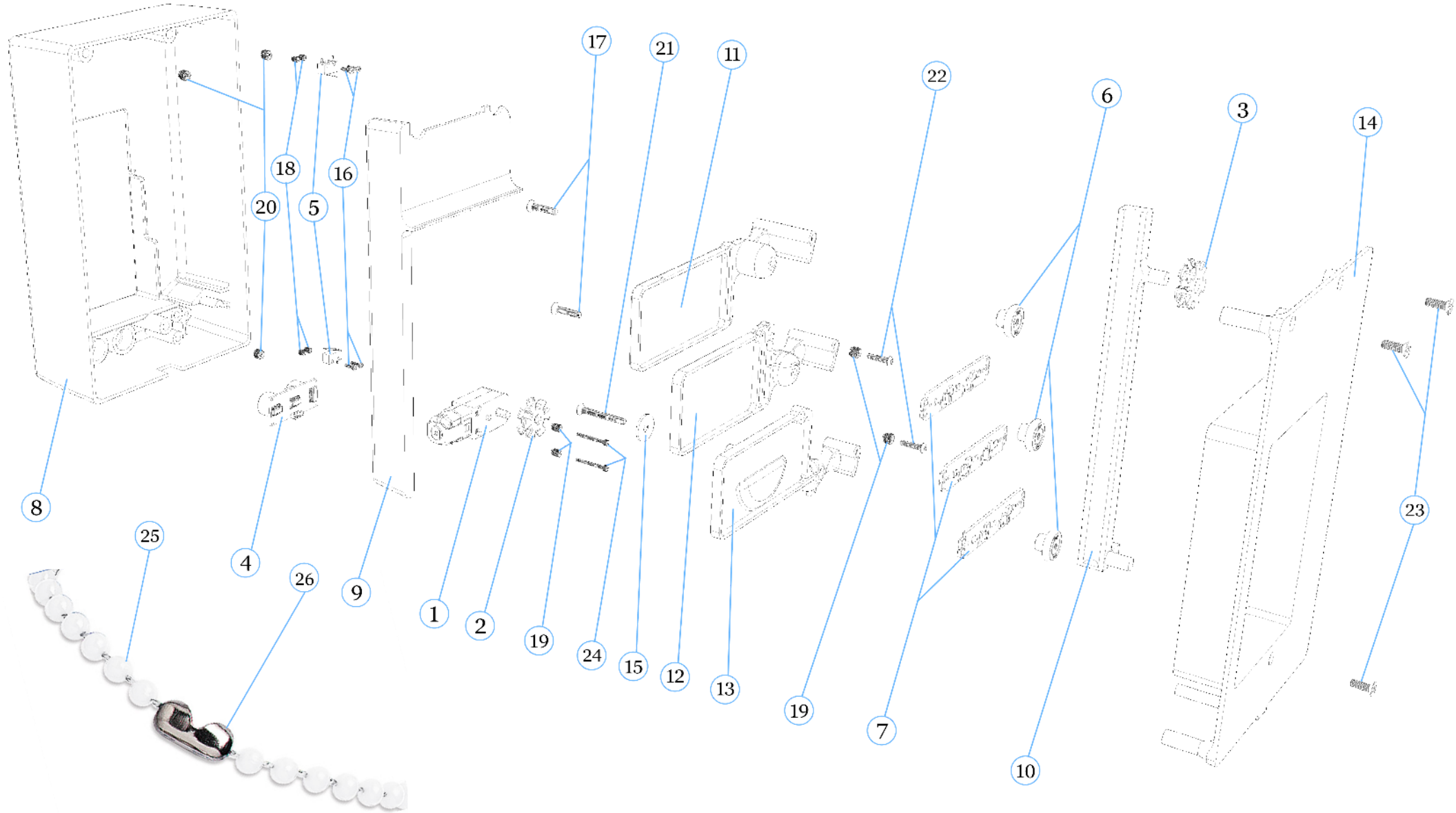
GermPass

Hall Call Unit Assembly Instructions

Designed and Illustrated by Les Martens

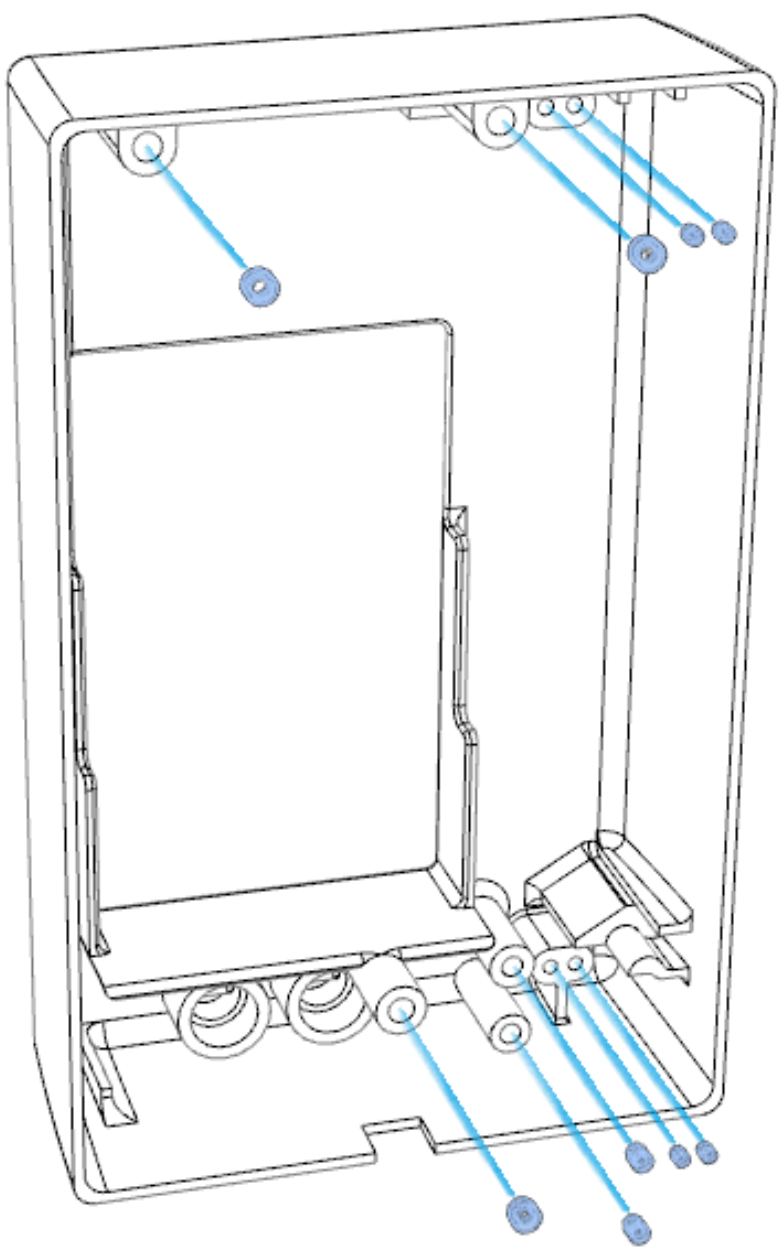


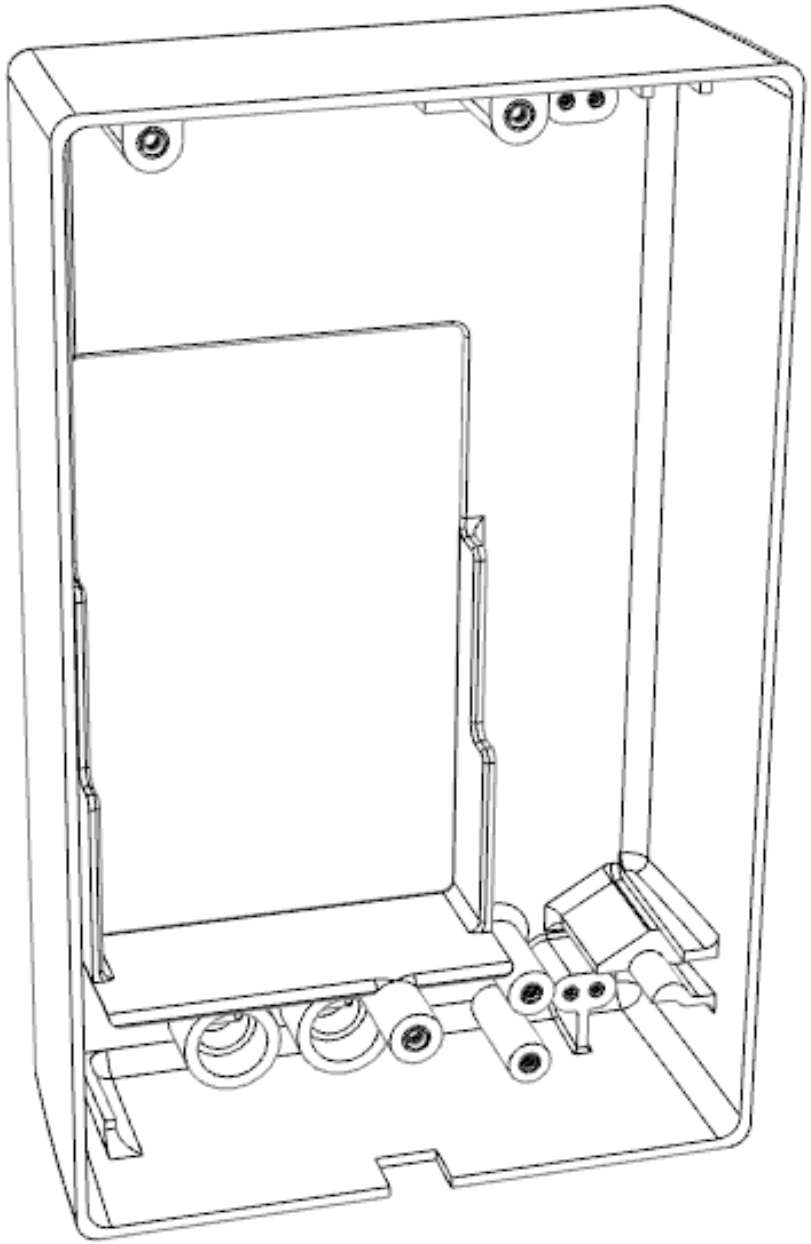
Bill Of Materials



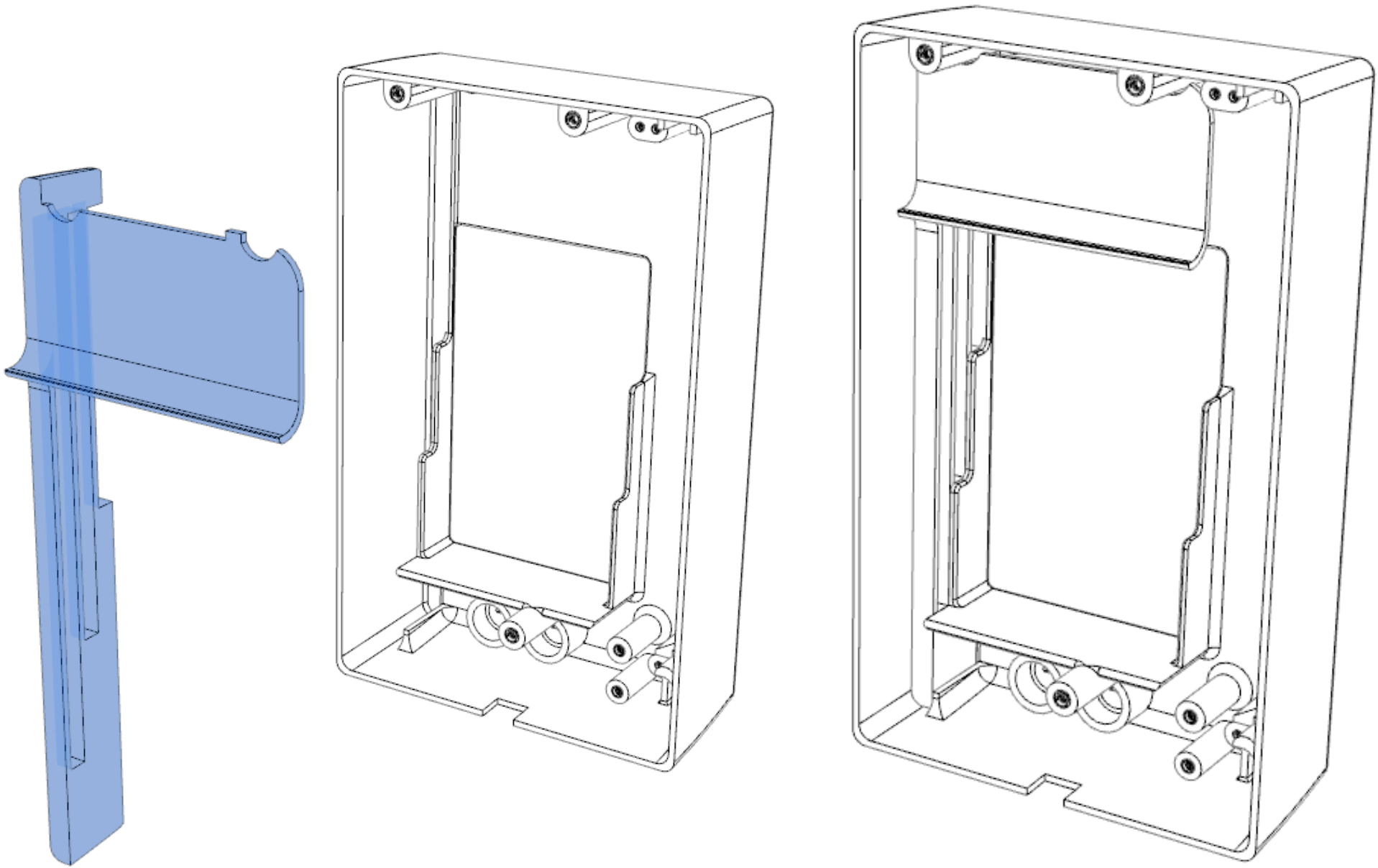
	Elevator Hall Call Unit			
Part No.	Name	Count	Make / Model	Description
1	TT Gear Motor	1	DC, 3v-6v, 1:48 Gear Ratio	Off Shelf - Plastic Drive Assembly
2	Drive Pulley	1	3D Printed in House Design	Pulley for Beaded Chain that fits on TT Motor Axle.
3	Support Pulley	1	3D Printed in House Design	Pulley for Beaded Chain that fits on Drive Rail Axle.
4	Ultrasonic Sensor	1	HC-SR04	Distance Sensor / Object Detection Sensor.
5	Limit Switch	2	CYT1073	Cylewet AC 1A 125V 3Pin SPDT Limit Micro Switch w/ Long Hinge Lever.
6	Shower Door Roller	3	CY103, 19mm Diameter	Glass Sliding Door Pulleys/Runners/Wheels,Strong Load-Bearing Capacity, Ultra-Quiet.
7	UVC 3x LED Light Board	3	LE-24V-3V-HC / 24Vdc / 350mA	3 LED Light Engine w/ integrated 12V DC driver, peak germicidal wavelength 265 nm.
8	HCU Outer Housing	1	3D Printed in House Design	The main body/chassis, outer product face with internal attach points.
9	HCU Guide Rail	1	3D Printed in House Design	Slotted design that the bare panel edge slides up and down in.
10	HCU Drive Rail	1	3D Printed in House Design	Vertical Bar which the panels are suspended from and the Shower Door Rollers glide al
11	HCU Top Panel	1	3D Printed in House Design	Upper most door in Automatic 3-door design.
12	HCU Middle Panel	1	3D Printed in House Design	Middle door in Automatic 3-door design.
13	HCU Bottom Panel	1	3D Printed in House Design	Lowest door in automatic 3-door design.
14	HCU Inner Housing	1	3D Printed in House Design	This sandwiches "fix point free" components into place and completes the outer housi
15	Bolt Seat Washer	1	3D Printed in House Design	This component gives the neccisarry thickness for a bolt head to be properly seated.
16	M2 x 12mm Round Head Hex Bol	4	Metric Hardware	Metric Bolt
17	M4 x 20mm Flat Head Hex Bolt	2	Metric Hardware	Metric Bolt
18	M2 Heat Set Inserts	4	Metric Hardware	Metric Bolt
19	M3 Heat Set Inserts	4	Metric Hardware	Metric Bolt
20	M4 Heat Set Inserts	3	Metric Hardware	Metric Bolt
21	M4 x 30mm Flat Head Hex Bolt	1	Metric Hardware	Metric Bolt
22	M3 x 12mm Round Head Hex Bol	2	Metric Hardware	Metric Bolt
23	M4 x 12mm Flat Head Hex Bolt	3	Metric Hardware	Metric Bolt
24	M3 x 25mm Round Head Hex Bol	2	Metric Hardware	Metric Bolt
25	Beaded Chain	1	Drive Chain	This component is simular to the beaded chain found on roller blinds
26	Beaded Chain Connector	1	Drive Chain Link	This connects one end of the beaded chain to the other end, creating a loop.
27	Nano			

1. Use a soldering iron to melt Heat Set Inserts (18,19,20) into the housing (8).

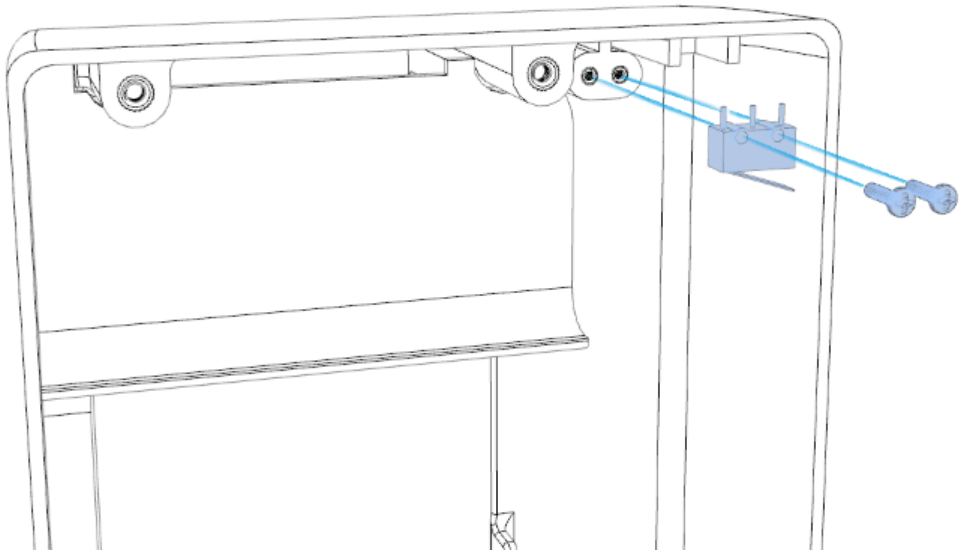


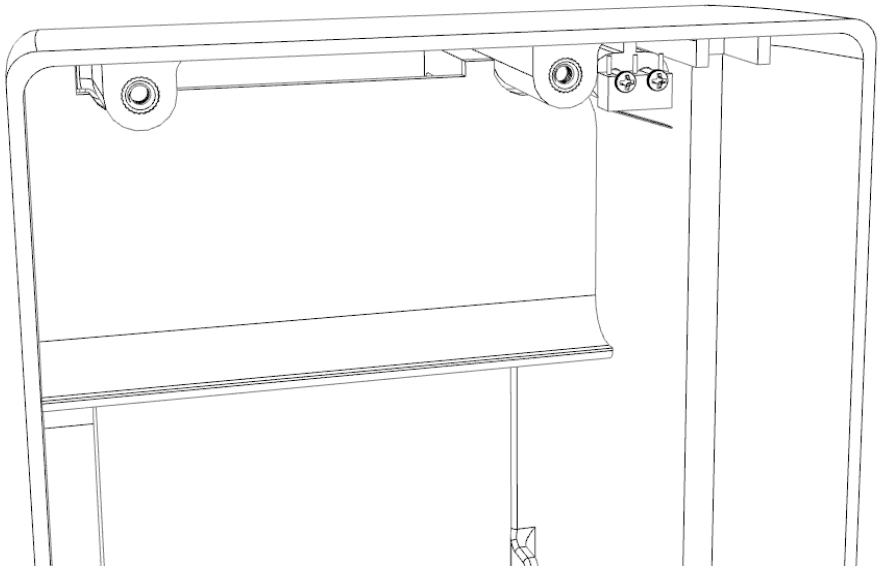


2. Insert the guide rail (9) into housing (8) without any fix points, “Fix Point Free.”

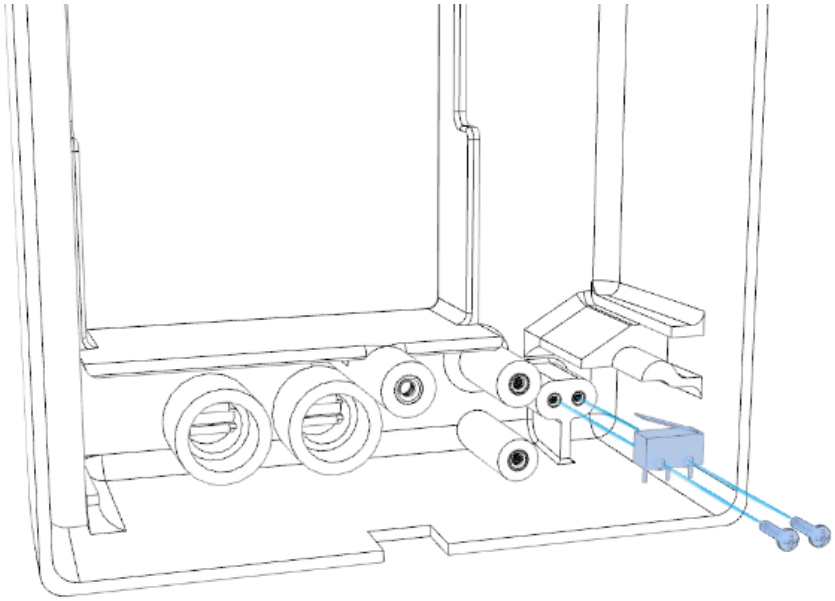


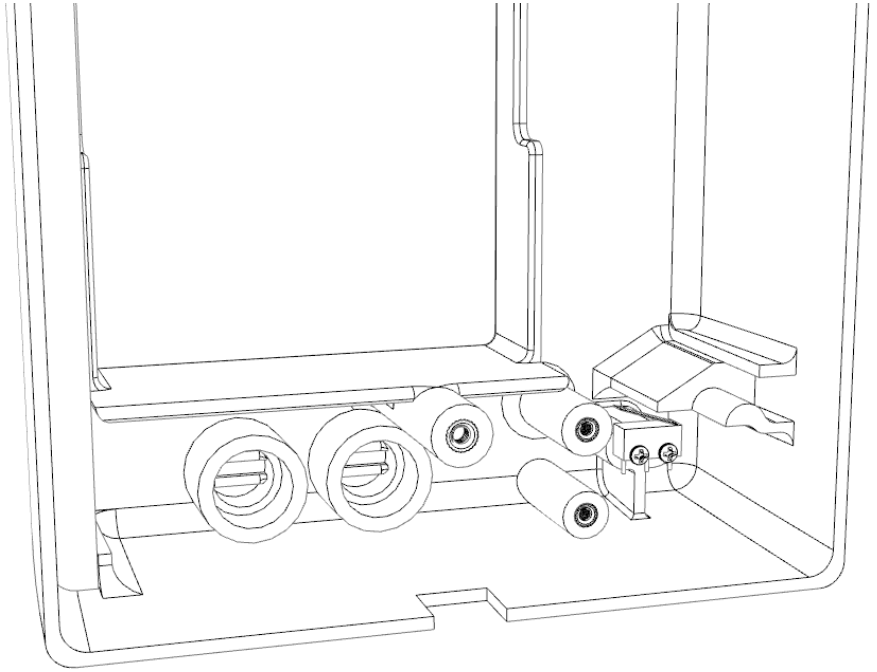
3. Affix the open limit switch (5) into the housing (8) using two screws (16).



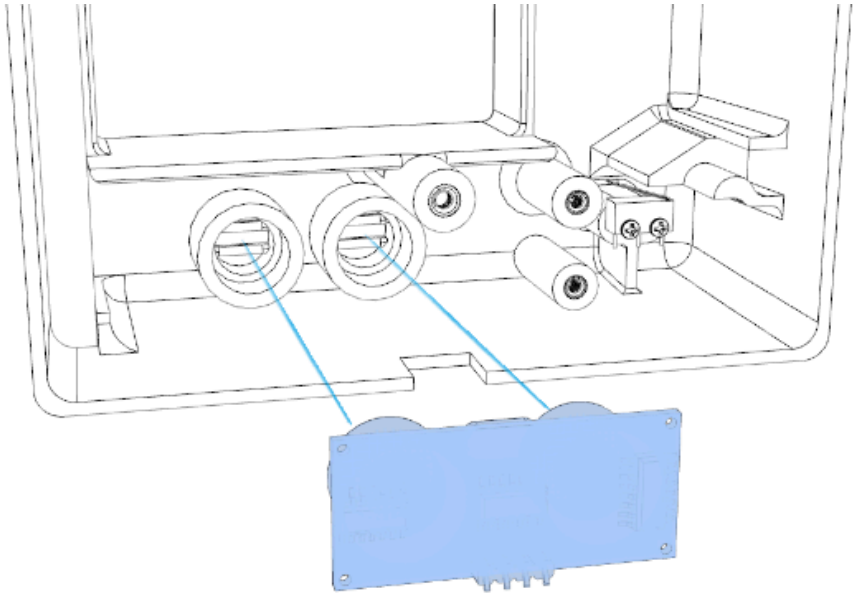


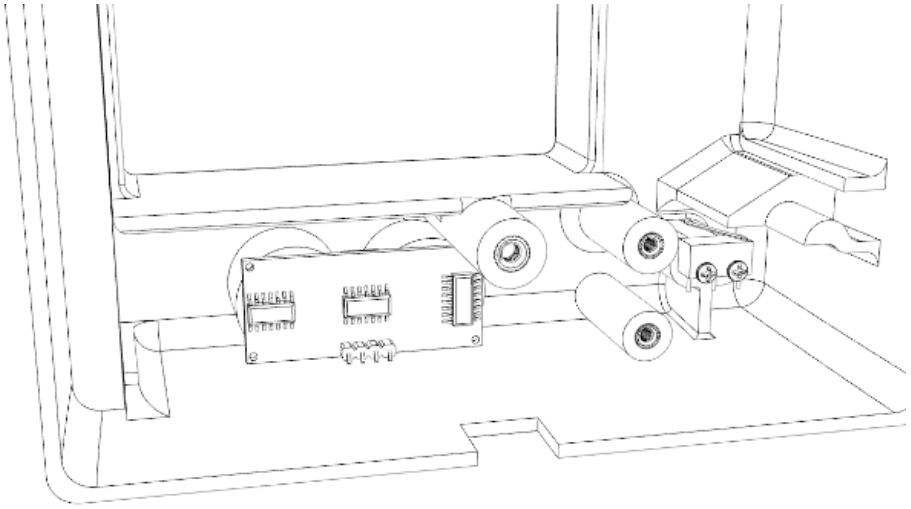
4. Affix the close limit switch (5) into the housing (8) using two screws (16).





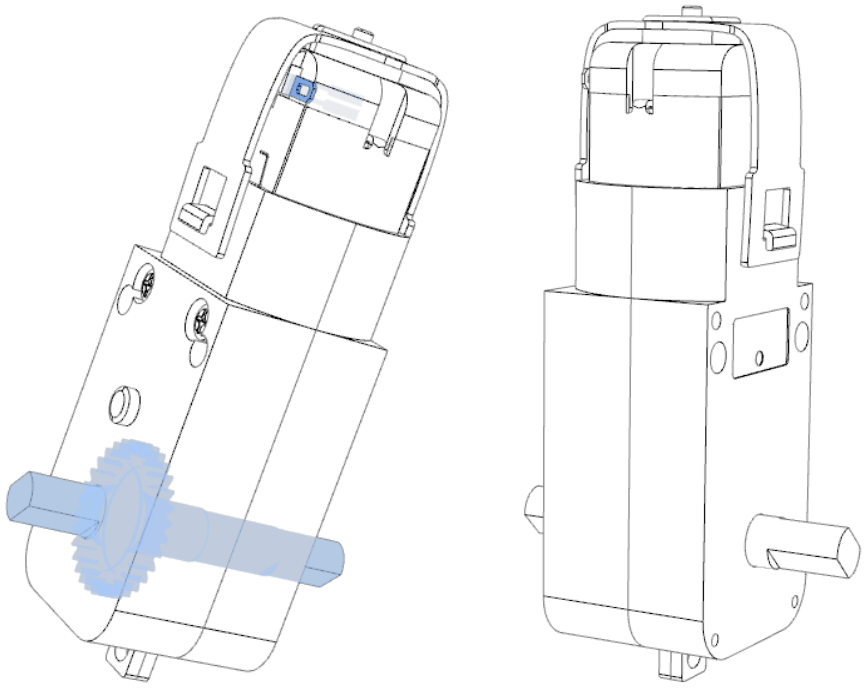
5. Insert motion sensor (4) fix-point free.



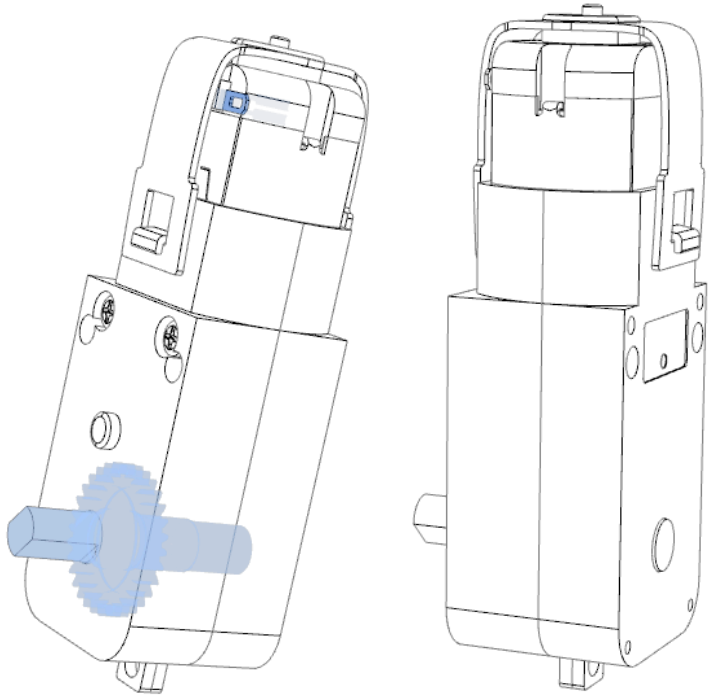


6. Use a Dremel on the TT motor (1) to shave off one side of the dual drive axle to be flush with housing. (Preferably the axle on the side of the motor without electrical leads)

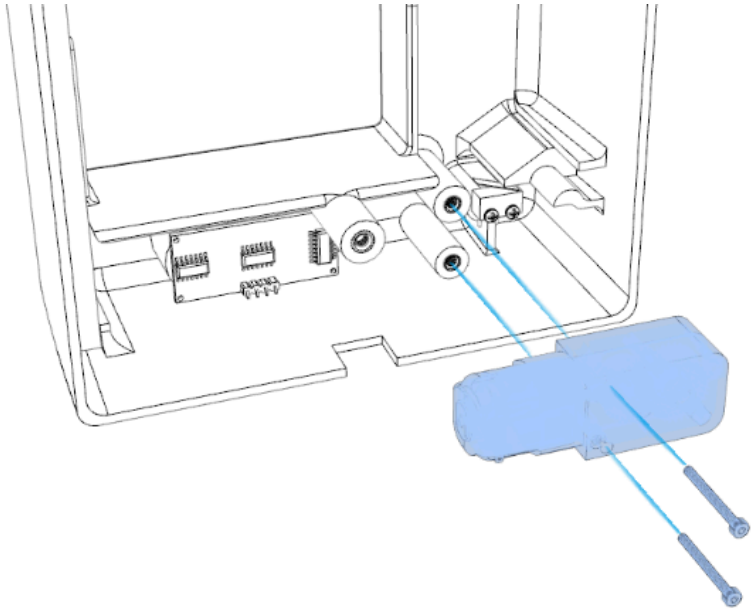
Before

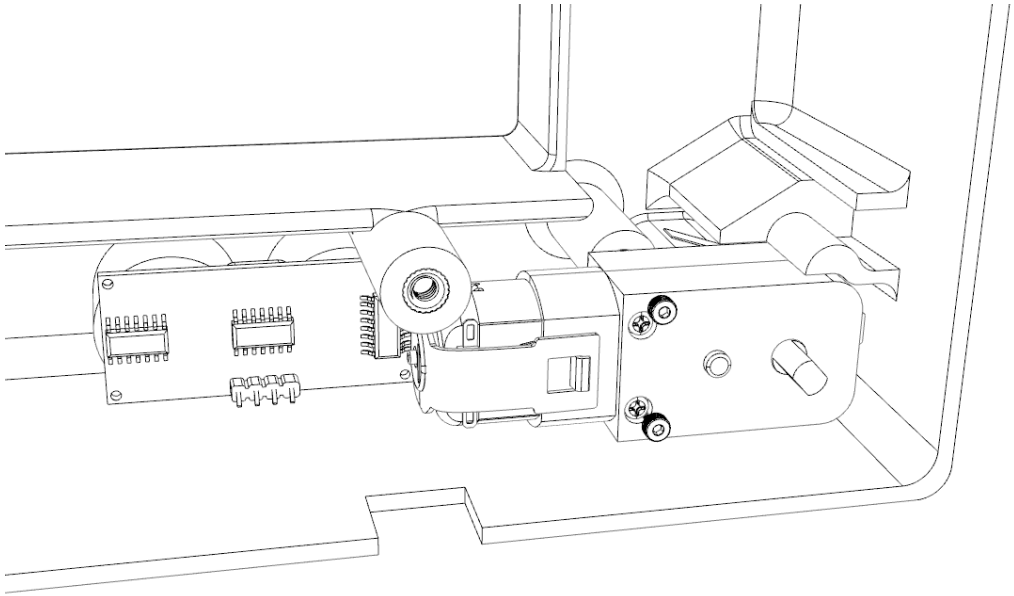


After

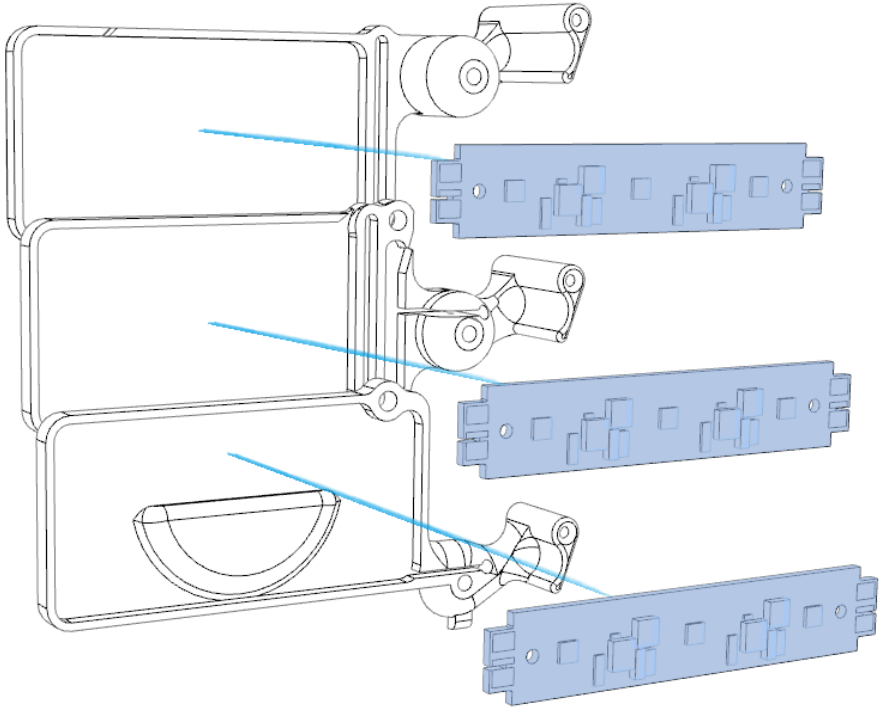


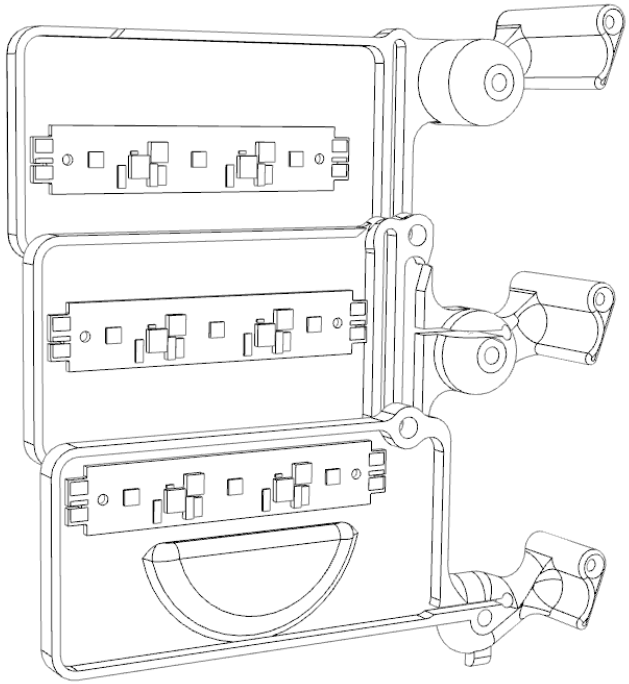
7. Use two screws (24) to mount the motor (1) into the housing (8).



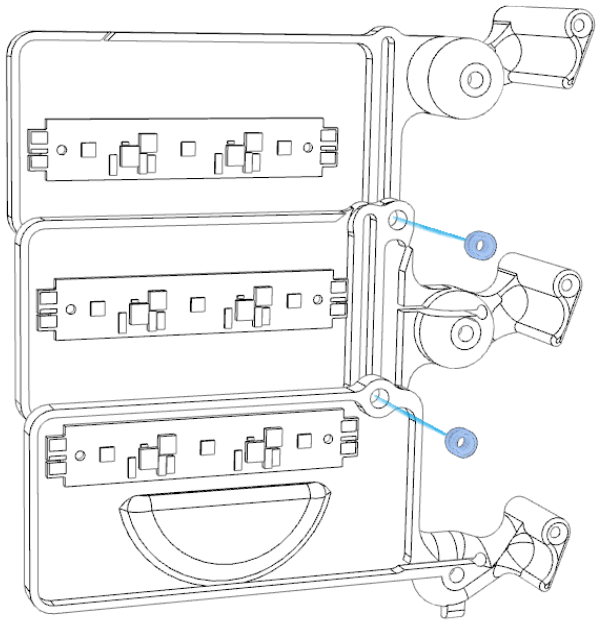


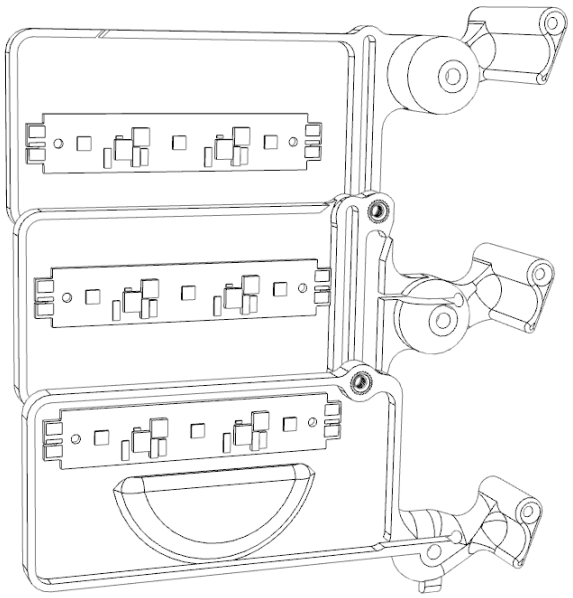
8. Glue UVC LED Strips (7) inside the door panels (11,12,13).



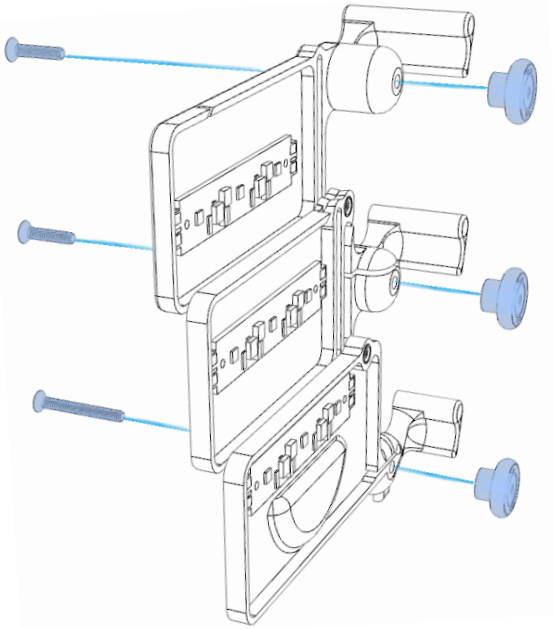


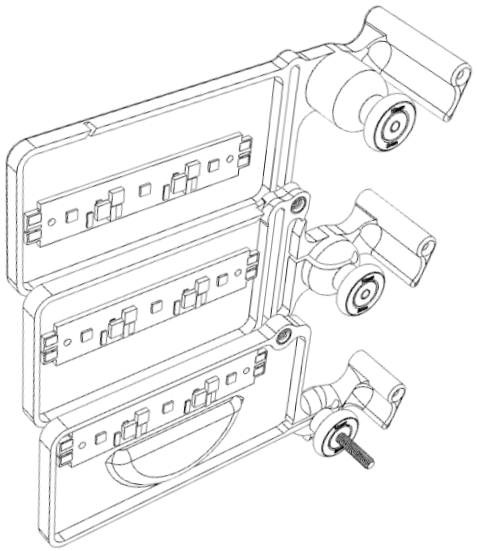
9. Use a soldering iron to place heat set inserts (19) into panels (12,13).



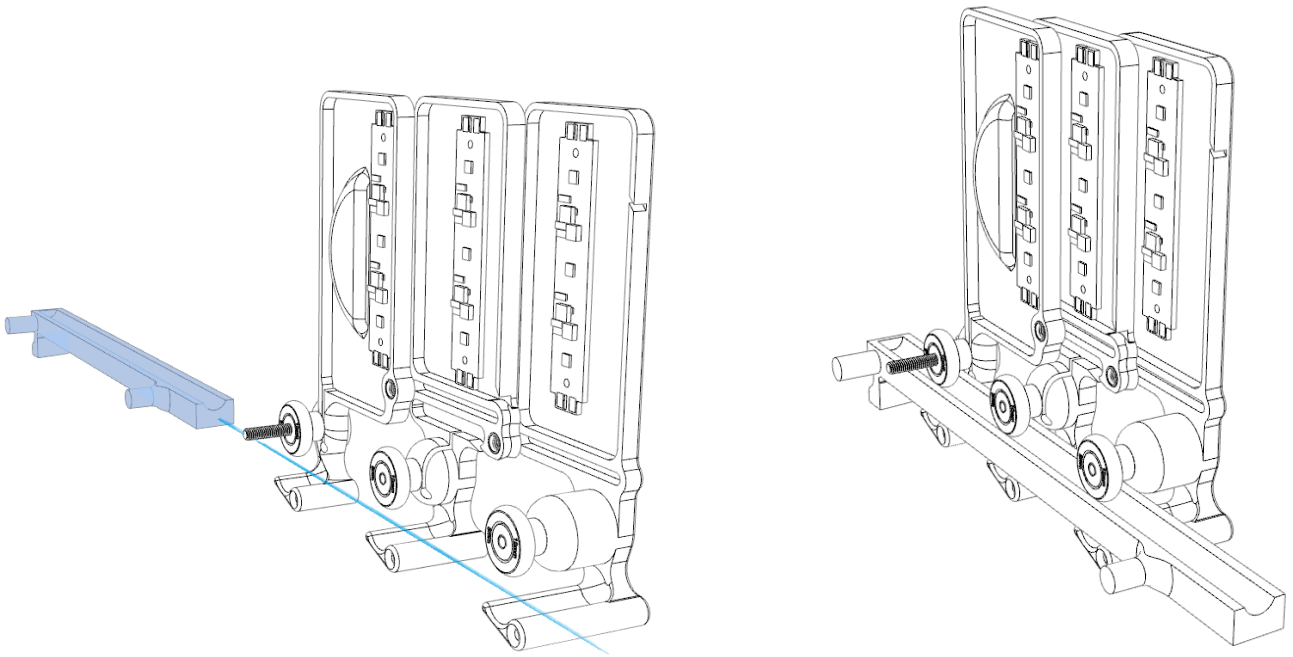


10. Screw (17,21) wheels (6) onto each panel (11,12,13).



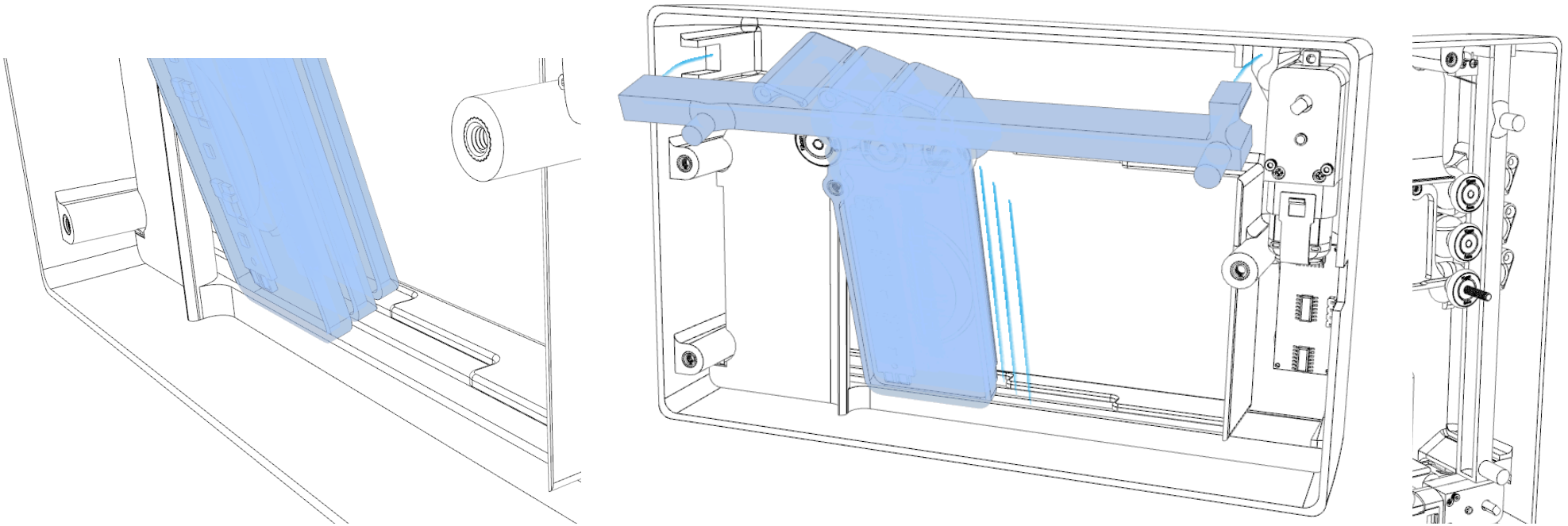


11. Slide each panel (11,12,13) onto the drive rail (10).

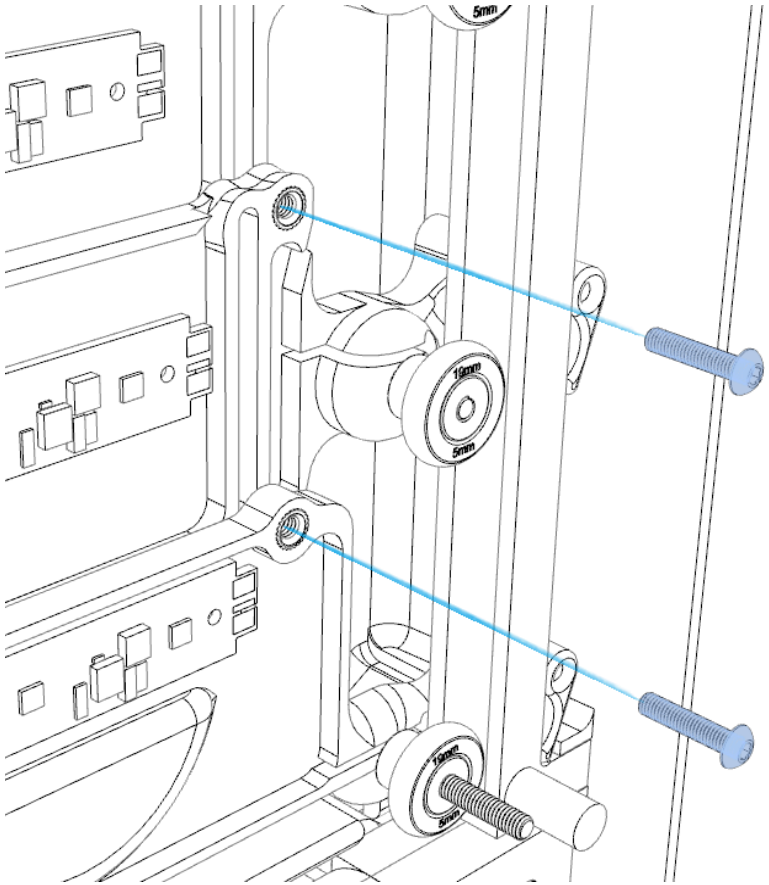


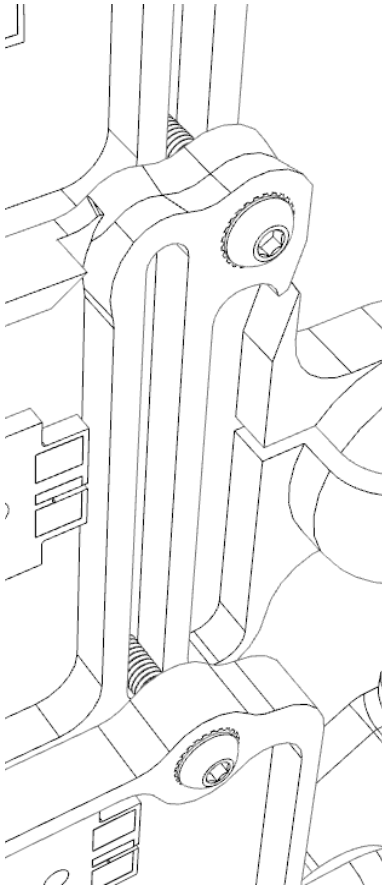
12. Apply graphite as needed to drive rail (10).

13. After setting the housing (8) on its side (guide rail down) gently place the outer edge of each panel (11,12,13) into its corresponding slot in the guide rail (9). Once the panels are in place, the drive rail (10) can now be slid into the other side of the housing, “fix-point free.”

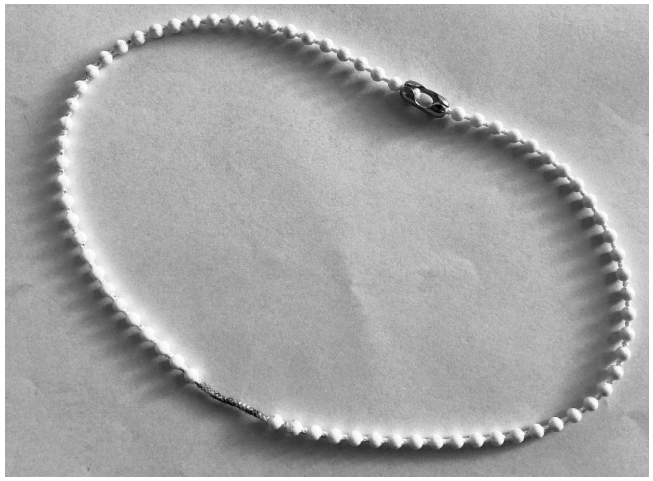


14. Insert panel guide screws (22).

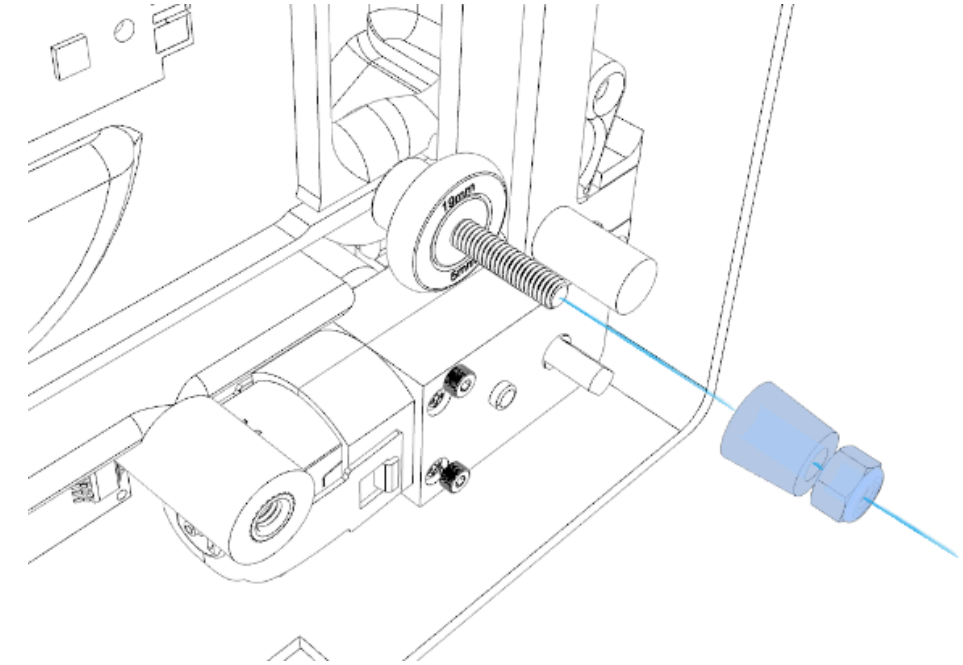


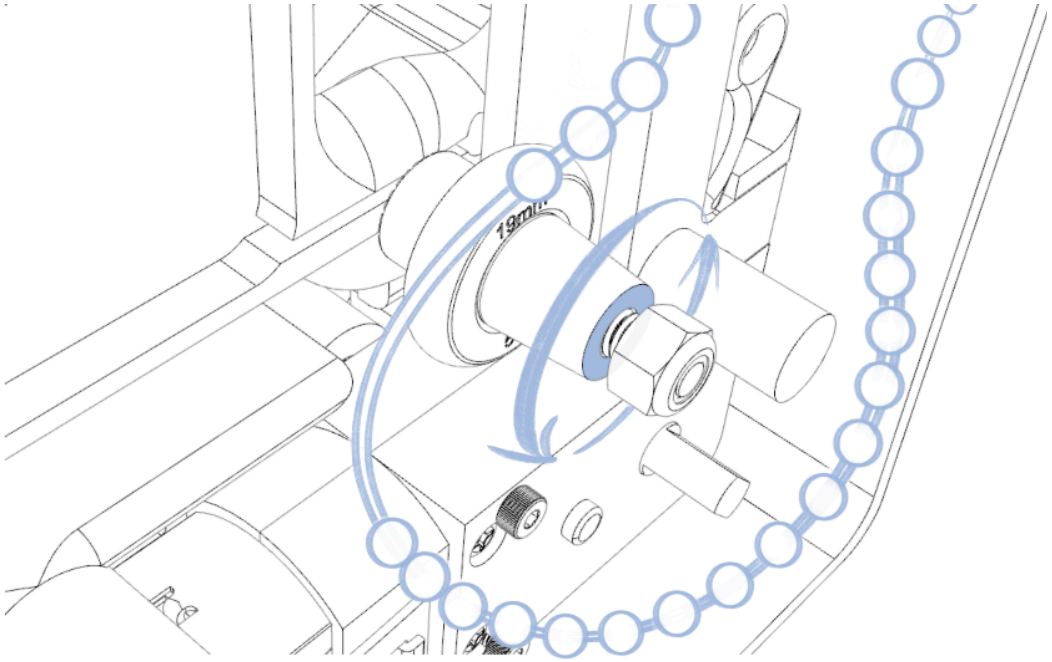


15. Cut the beaded chain (25) to length and create a loop using the connector (26). Snip the beads off a portion of string to create a gap of bare string without beads.

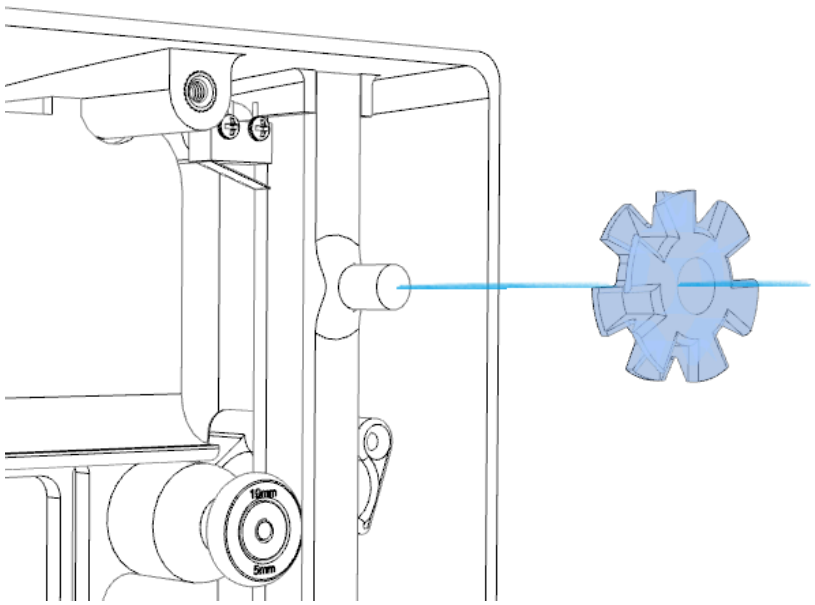


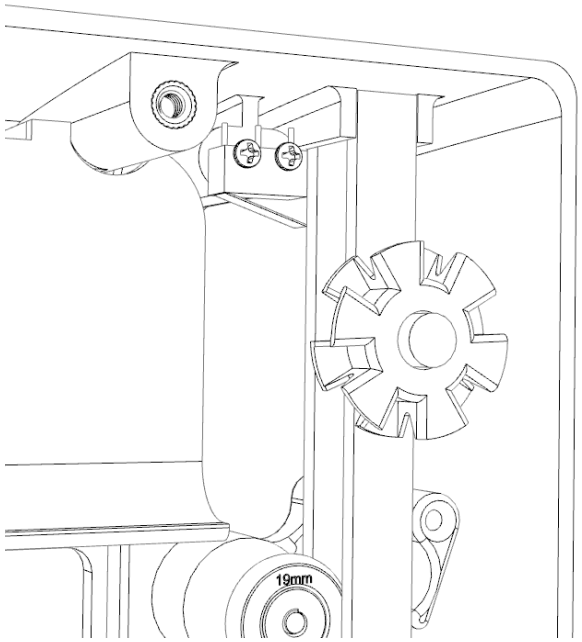
16. Wrap the gap of string in the beaded chain (25) around the drive panel (13) fix point bolt (21) one time and screw on locknut to keep in place. (Additional glue may be needed to keep this fix point from rotating.)



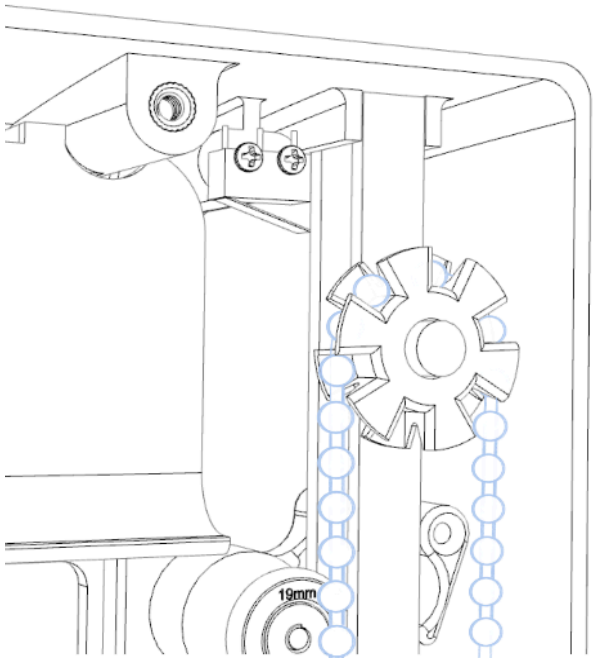


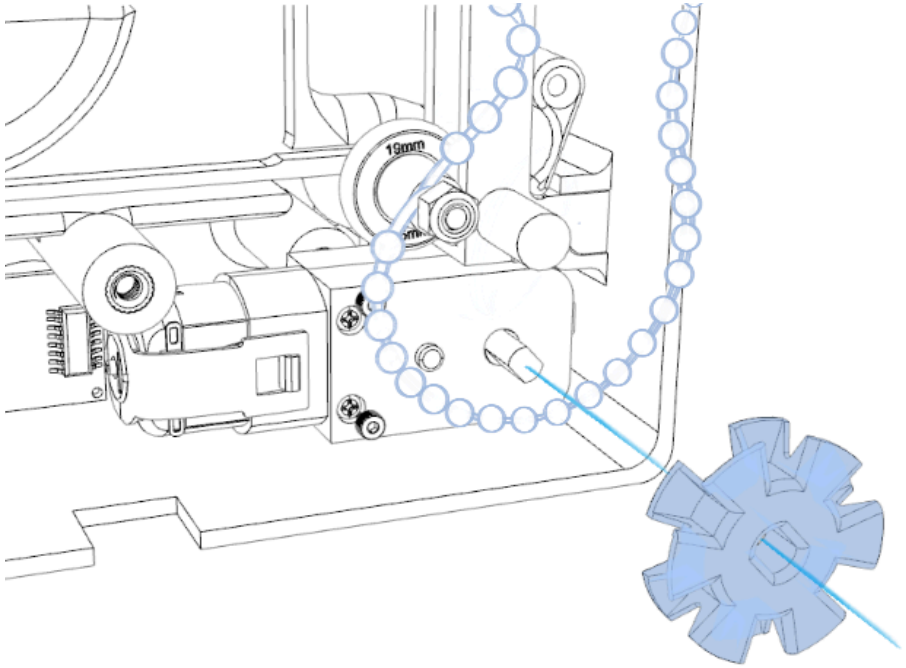
17. Slide the support pulley (3) onto the top drive rail (10) axle.



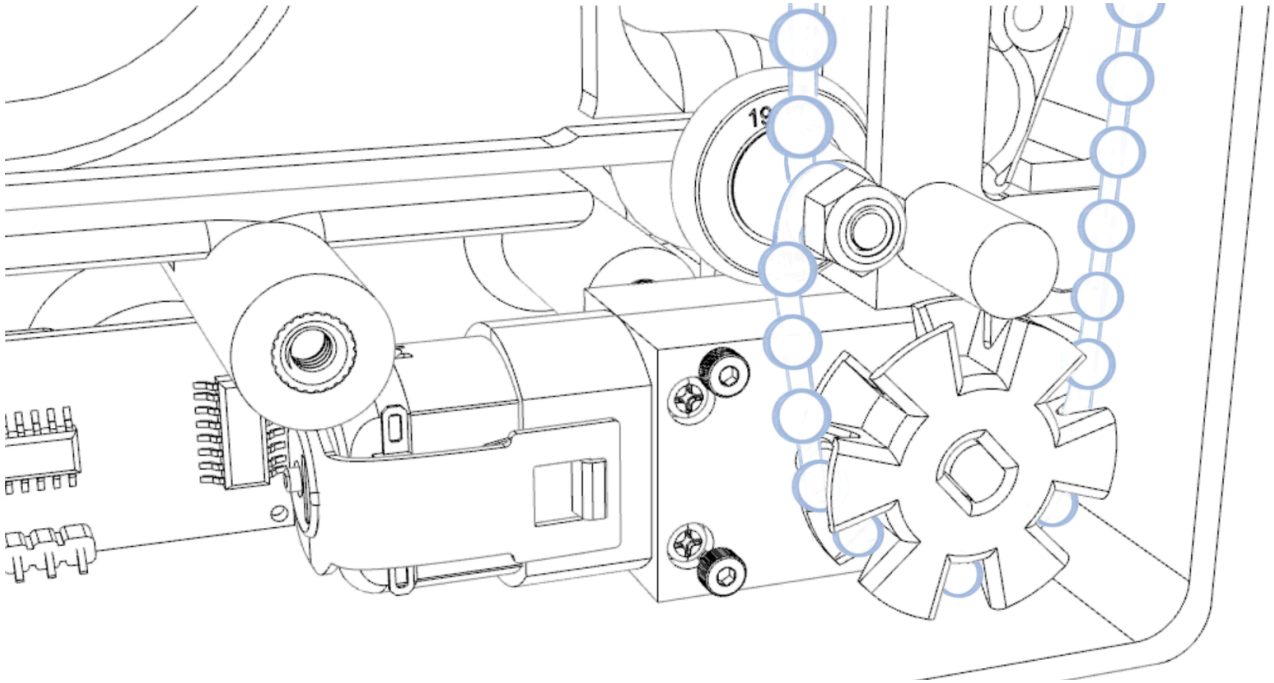


18. Wrap one end of the beaded chain (25) around the support pulley and the other around the drive pulley before mounting the drive pullet to the motor.





19. With the beaded chain (25) wrapped around the support (3) & drive (2) pulleys, Slide the drive pulley onto the motor (1) axle.



20. Upload code to Arduino

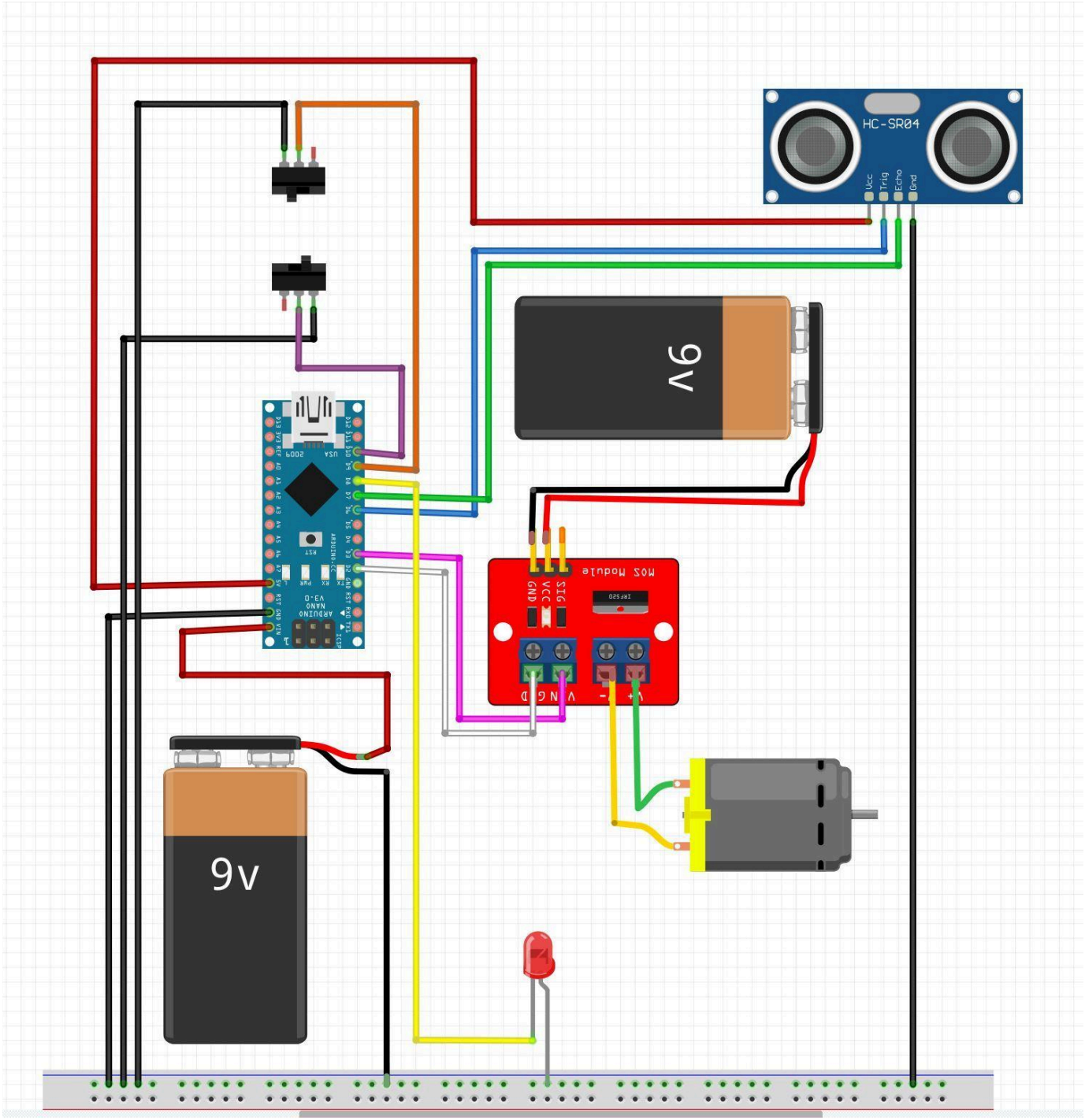
```
#include "src/MX1508/MX1508.h"
#include "src/HC_SR04/HCSR04.h"
// constants won't change
const int UVC_LED_PIN = 2;         // UVC LED signal Pin
const int OPEN_SWITCH_PIN = 9;     // Switch to detect the door is open
const int CLOSE_SWITCH_PIN = 10;   // Switch to detect the door is closed
const int TRIG_PIN = 6;            // Arduino pin connected to Ultrasonic Sensor's TRIG pin
const int ECHO_PIN = 7;            // Arduino pin connected to Ultrasonic Sensor's ECHO pin
const int LED_PIN = 8;             // Arduino pin connected to LED's pin
const uint8_t MOTOR_PIN_A = 3;         // Motor Pin A (PWM)
const uint8_t MOTOR_PIN_B = 4;         // Motor Pin 2
const int MAX_OPEN_SPEED = 255;    // Speed the motor should run to open the door - Range between 80-255
const int MAX_CLOSE_SPEED = -150; // Speed the motor should run to close the door  - Range between 80-255
const int MIN_OPEN_SPEED = 90;    // Speed the motor maintains after fully decelerating - Range between 80-255
const int MIN_CLOSE_SPEED = -65;   // Speed the motor maintains after fully decelerating - Range between 80-255
const int DISTANCE_THRESHOLD = 50; // Distance at which Motion sensor is triggered in centimeters
const int UVC_RUNNING_CYCLES = 3; // How many times the UVC LEDs Blink
const int OPEN_DELAY = 3; // Lower Number = Faster Deceleration - Range from 2-6
const int CLOSE_DELAY = 2; // Lower Number = Faster Deceleration - Range from 2-6
// variables will change:
float duration_us, distance_cm;
bool opened = false;
bool closed = false;
bool isOpening = false;
bool isClosing = false;
bool shouldCloseDoor = false;
bool uvc_cycle_running = false;
MX1508 motor(MOTOR_PIN_A, MOTOR_PIN_B, FAST_DECAY, PWM_1PIN);
HCSR04 sensor(TRIG_PIN, ECHO_PIN);
void setup()
{
Serial.begin(9600);                     // initialize serial port
pinMode(TRIG_PIN, OUTPUT);              // set arduino pin to output mode
pinMode(ECHO_PIN, INPUT);               // set arduino pin to input mode
pinMode(LED_PIN, OUTPUT);               // set arduino pin to output mode
pinMode(UVC_LED_PIN, OUTPUT);           // UVC LED
pinMode(OPEN_SWITCH_PIN, INPUT_PULLUP);  // "Opened" Switch
pinMode(CLOSE_SWITCH_PIN, INPUT_PULLUP); // "Closed" Switch
Serial.begin(96000);
Serial.begin(115200);
}
void loop()
{
tryProccessMotionSensorLogic();
tryProccessOpenSwitchLogic();
tryProccessCloseSwitchLogic();
delay(50); // 50 ms delay between loops
}
void tryProccessMotionSensorLogic()
```

```
{
if (shouldOpenDoor())
{
openDoor(isClosing);
}
else if (shouldCloseDoor && !ultraSonicSensorTriggered())
{
shouldCloseDoor = false;
closeDoor();
}
}
void tryProccessOpenSwitchLogic()
{
if (!opened && isOpenSwitchPressed()) // Only run the following if the door isn't currently opened and the open switch is pressed
{
opened = true;
stopDoor();  // stop the door
delay(5000); // wait 5 seconds
shouldCloseDoor = true;
}
else if (!isOpenSwitchPressed()) // Only run the following if the open switch is not pressed
{
opened = false;
}
}
void tryProccessCloseSwitchLogic()
{
if (!closed && isCloseSwitchPressed()) // Only run the following if the door isn't currently closed and the close switch is pressed
{
closed = true;
delay(200);
stopDoor();                          //  stop the door
runUvcLedCycle(UVC_RUNNING_CYCLES); // run the UVC LED Cycle
}
else if (!isCloseSwitchPressed()) // Only run the following if the close switch is not pressed
{
closed = false;
}
}
// Returns whether the door should open or not
bool shouldOpenDoor()
{
return !opened && !isOpening && !uvc_cycle_running && !shouldCloseDoor && ultraSonicSensorTriggered();
}
// Returns whether the sensor is triggered
bool ultraSonicSensorTriggered()
{
distance_cm = sensor.distanceInMillimeters() / 10;  // calculate the distance
return distance_cm <= DISTANCE_THRESHOLD; // return if the calculated distance is within proper range
}
```

```
// Returns whether the open switch is pressed or not
bool isOpenSwitchPressed()
{
return digitalRead(OPEN_SWITCH_PIN) == LOW;
}
// Returns whether the close switch is pressed or not
bool isCloseSwitchPressed()
{
return digitalRead(CLOSE_SWITCH_PIN) == LOW;
}
// Attempts to open the door
void openDoor(bool steadySpeed)
{
if (!opened && !isOpening) // only attempt to open the door if it isn't already open or opening
{
isOpening = true;
if (steadySpeed)
{
motor.motorGo(MIN_OPEN_SPEED);
}
else
{
// Start motor at max open speed and decelerate till open
for (int i = MAX_OPEN_SPEED; i >= MIN_OPEN_SPEED; i--)
{
motor.motorGo(i);
delay(OPEN_DELAY);
}
}
}
}
// Attempts to close the door
void closeDoor()
{
if (!closed && !isClosing) // only attempt to close the door if it isn't already closed or closing
{
isClosing = true;
// Start motor at max close speed and decelerate till closed
for (int i = MAX_CLOSE_SPEED; i <= MIN_CLOSE_SPEED; i++)
{
motor.motorGo(i);
delay(CLOSE_DELAY);
}
}
}
// Attemtps to stop the door
void stopDoor()
{
isOpening = false;
isClosing = false;
```

```
150) motor.stopMotor(); // stop closing rotation
151) }
152) // Runs the UVC LED Cycle - recursive
153) void runUvcLedCycle(int cyclesRemaining)
154) {
155) if (closed && cyclesRemaining > 0)
156) {
157) uvc_cycle_running = true;
158) delay(250);
159) digitalWrite(UVC_LED_PIN, HIGH); // turn on the UVC LED
160) delay(800);                      // For .8 seconds
161) digitalWrite(UVC_LED_PIN, LOW); // turn off the UVC LED
162) delay(200);                     // For .2 seconds
163) runUvcLedCycle(cyclesRemaining - 1); // run it again
164) }
165) else
166) {
167) uvc_cycle_running = false;
168) }
169) }
```

21. Connect the wiring harness, batteries, motion sensor and limit switches to the Arduino.



22. Test run unit.
23. Connect UVC LED strips (7) to the Arduino.
24. Insert the inner housing (14) which in most models pushes against and secures all “fix-point free” objects. Hold the inner housing into place with 3 screws (23).