

Diario de Rorschach hoy es 3 de Octubre de 2022:

Me puse a investigar como empezar a hacer el gatito claramente empezando por como hacer un robot cuadripedo, empeze viendo videos de cuadrupedos, empezando por este:

https://www.youtube.com/watch?v=RQOVxuyS2_Y

A la mitad del video entendí que era más un video de muestra de robots avanzados, pero lo vi por si dijera algo que me pudiera servir, spoiler no habia nada asi que pase al siguiente:

https://www.youtube.com/watch?v=XdNCb2R3_zg

Este video fue bastante interesante ya que explicaba un poco cual era la ciencia sobre el movimiento del robot hablando de que predecía hacia donde iría y como actuaria la gravedad en él para saber como poner las patas, me gustaría primero hablar con el profesor para ver que tan viable es esto. Además es un proyecto open source el cual se puede permitir ver como esta hecho por dentro, además que en su canal tiene más videos hablando del tema asi que quiza encuentro algo piola. Procedo a adjuntar el gitlab con el contenido del robot open source y el segundo video de el mismo canal que vi:

https://gitlab.com/anflores/pavlov_mini

<https://www.youtube.com/watch?v=d26RmQo-feY>

Saqué bastante información y nisiquiera termine el video, hice un modelo para la pierna y para el cuerpo, después lo termine.

Diario de Rorschach 5 de Octubre de 2022:

Hoy tocaba proyecto informático, me dispuse a hablar con el Game Master dijo que era demasiado complejo, había que meterse en componentes muy difíciles de conseguir, mucha cantidad y un sistema nuevo para mi equipo, así que vamos a seguir investigando.

<https://blog.arduino.cc/2020/10/10/make-your-own-3d-printed-arduino-powered-quadruped-robot-for-under-60/>

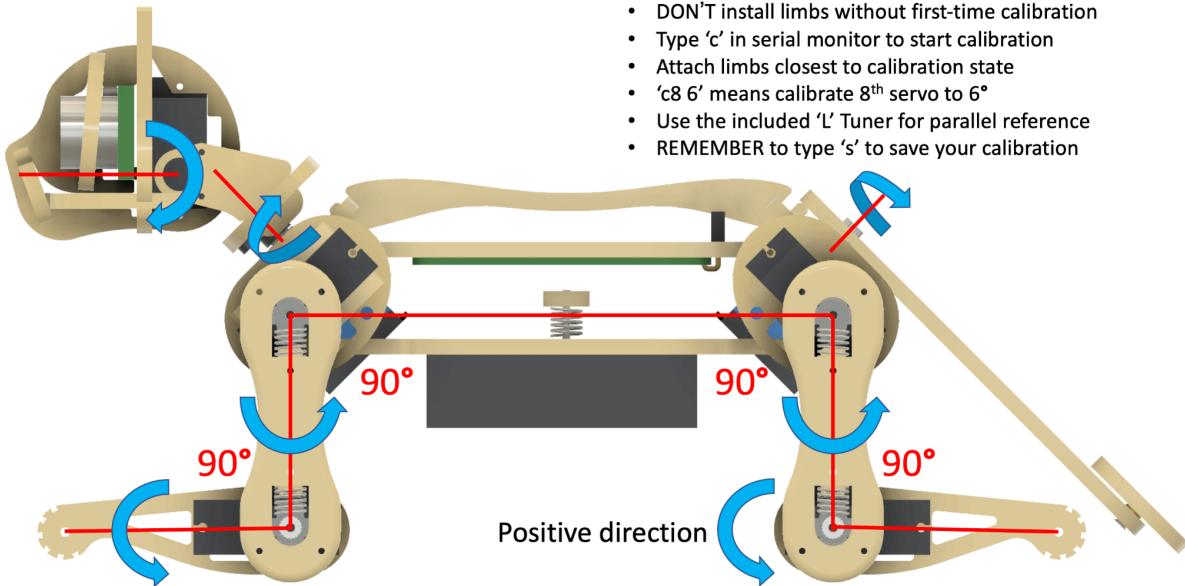
<https://www.instructables.com/3D-Printed-Arduino-Powered-Quadruped-Robot/>

Como los rodamientos salieron una locura seguimos viendo:

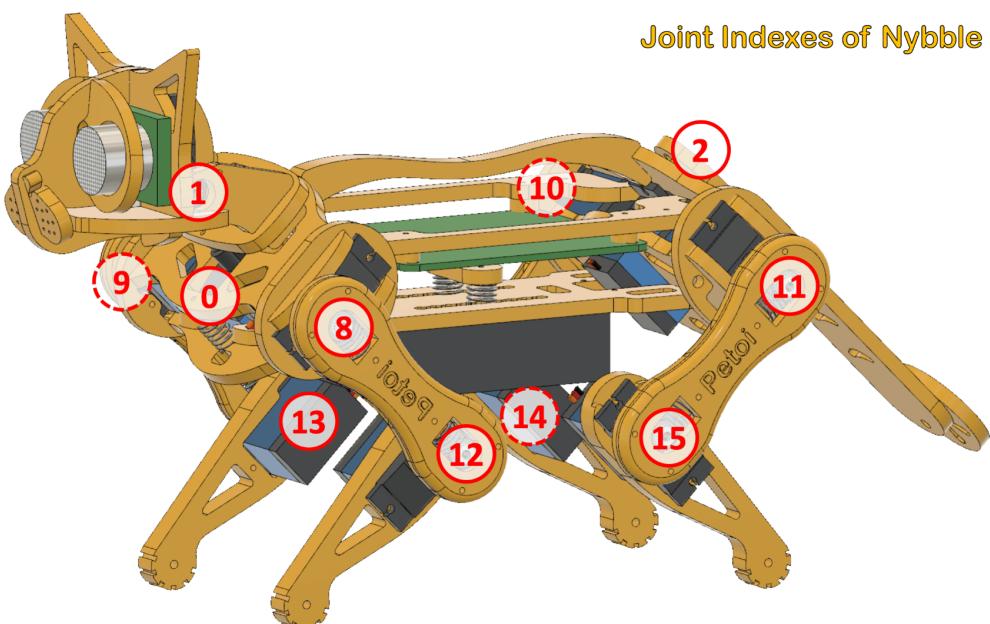
<https://www.instructables.com/circuits/howto/an+quadruped+robot/>

Nuestro compañero Matías termino encontrando el modelo que íbamos a usar con los modelos 3D y los componentes que necesitaremos

Zero Angle/Calibration State

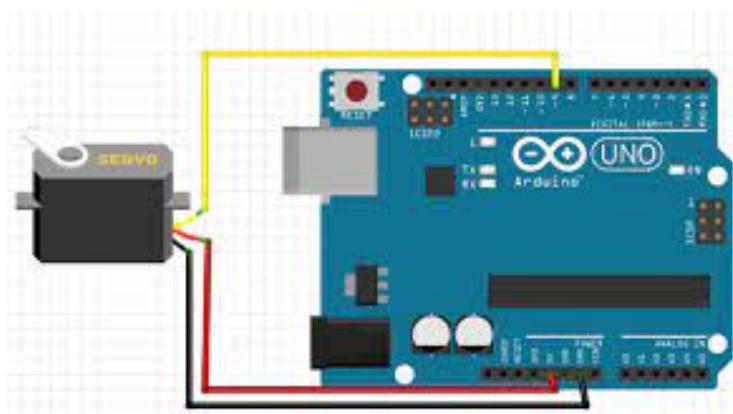


Joint Indexes of Nibble



Diario de Rorschach 12 de Octubre de 2022:

Hoy toca empezar el cableado por lo que empezamos viendo cómo conectar el servomotor, el principal componente, compensamos viendo modelos de conexión.



Y conectamos un servomotor al arduino, luego lo conectamos a

la pc y empezamos a investigar la librería <SERVO>

Diario de Rorschach 19 de Octubre de 2022:

Hoy conectamos los 2 servomotores, que conformarán una pierna, luego usamos un video que mostró Matias para analizar el movimiento del gato y así asimilar como se moverá y sus ángulos. Esperaremos probarlo cuando tengamos el modelo impreso y crearemos el sistema de movimiento ya que tenemos una pierna con la que haremos las demás pero nos falta las demás

```
// Lo primero que hace el código de arduino es llamar a una biblioteca para llamar
// servomotores llamada <Servo.h>
#include <Servo.h>
// Luego declaramos los servomotores que tenemos, por ahora solo tenemos 2 pero luego
// incluiremos más
Servo muslo_1;
Servo pant_1;

void setup() {
    // Declararemos el pin en el que está conectado los Servos
    muslo_1.attach(9);
    pant_1.attach(11);
}

void loop() {
    // put your main code here, to run repeatedly:
    // Aquí modificamos los ángulos de cada servo y ponemos un delay entre estos
    muslo_1.write(0);
    pant_1.write(15);
    delay(2000);
    muslo_1.write(80);
    pant_1.write(0);
    delay(2000);

}
```

Actualización

Agregando cosas

```
// Lo primero que hace el código de arduino es llamar a una biblioteca para llamar
```

```

// servomotores llamada <Servo.h>
#include <Servo.h>
// Luego declaramos los servomotores que tenemos, por ahora solo tenemos 2 pero luego
// incluiremos mas
Servo muslo_1;
Servo pant_1;

// Creamos listas con servos adentro separando entre los servos de los muslos y las
// Pantorrillas
Servo muslos[] = { muslo_1 };
Servo pant[] = { pant_1 };

void setup() {
    // Declararemos el pin en el que esta conectado los Servos
    muslo_1.attach(9);
    pant_1.attach(11);
}

void loop() {
    // Llamamos una funcion dandole el muslo 1 y la pantorrilla 1 como servos a mover
    mover_pata( muslos[0], pant[0] );

}

void caminar(){
    // Esto es una prueba de una funcion a la que llamar cada ves que se va a mover
    // mover_pata(muslo[0], pant[0]);
}

// Creamos una funcion para mover las patas dando los datos de 2 servos
void mover_pata(Servo muslo, Servo pant){
    // Cuando el muslo este en la posicionA (Pie parado) lo lleve a la posicionB (Pie adelantado)
    if (muslo.read() == 0){
        muslo.write(80);
        pant.write(0);
    }
    // Cuando el muslo este en la posicionB (Pie adelantado) lo lleve a la posicionA (Pie parado)
    else{
        muslo.write(0);
        pant.write(15);
    }
}

```

 Como Hacer un GATO ROBOT con Arduino - PCBway

Diario de Rorschach 27 de Octubre de 2022:

Hoy se siguió con el código creando un sistema de caminata que hasta no tener la pierna completa no se puede probar

```
// Lo primero que hace el código de arduino es llamar a una biblioteca para llamar
// servomotores llamada <Servo.h>
#include <Servo.h>
// Luego declaramos los servomotores que tenemos, por ahora solo tenemos 2 pero luego
// incluiremos mas
Servo muslo_1;
Servo pant_1;

// Para probar si funcionaba
#define boton 4

// Creamos listas con servos adentro separando entre los servos de los muslos y las
// Pantorrillas
Servo muslos[] = { muslo_1 };
Servo pants[] = { pant_1 };

// Creamos listas con los angulos para cada situación específica
int angulos_1[] = { 80, 0, 80, 0};
int angulos_2[] = { 0, 15, 0, 15};
int angulos_if[] = {angulos_1, angulos_2};
int angulos_else[] = {angulos_2, angulos_1};
int angulos_todos[] = {angulos_if, angulos_else};

void setup() {
    // put your setup code here, to run once:
    muslo_1.attach(9);
    pant_1.attach(11);
    pinMode(boton, INPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    while ( digitalRead(boton) == HIGH ){
        caminar(muslos, pants, angulos_todos, 0);
    }else{
        caminar_reset(muslos, pants, angulos_todos, 1);
    }
}

void caminar(Servo muslo[], Servo pant[]){
```

```

mover_pata(muslos, pants);
}

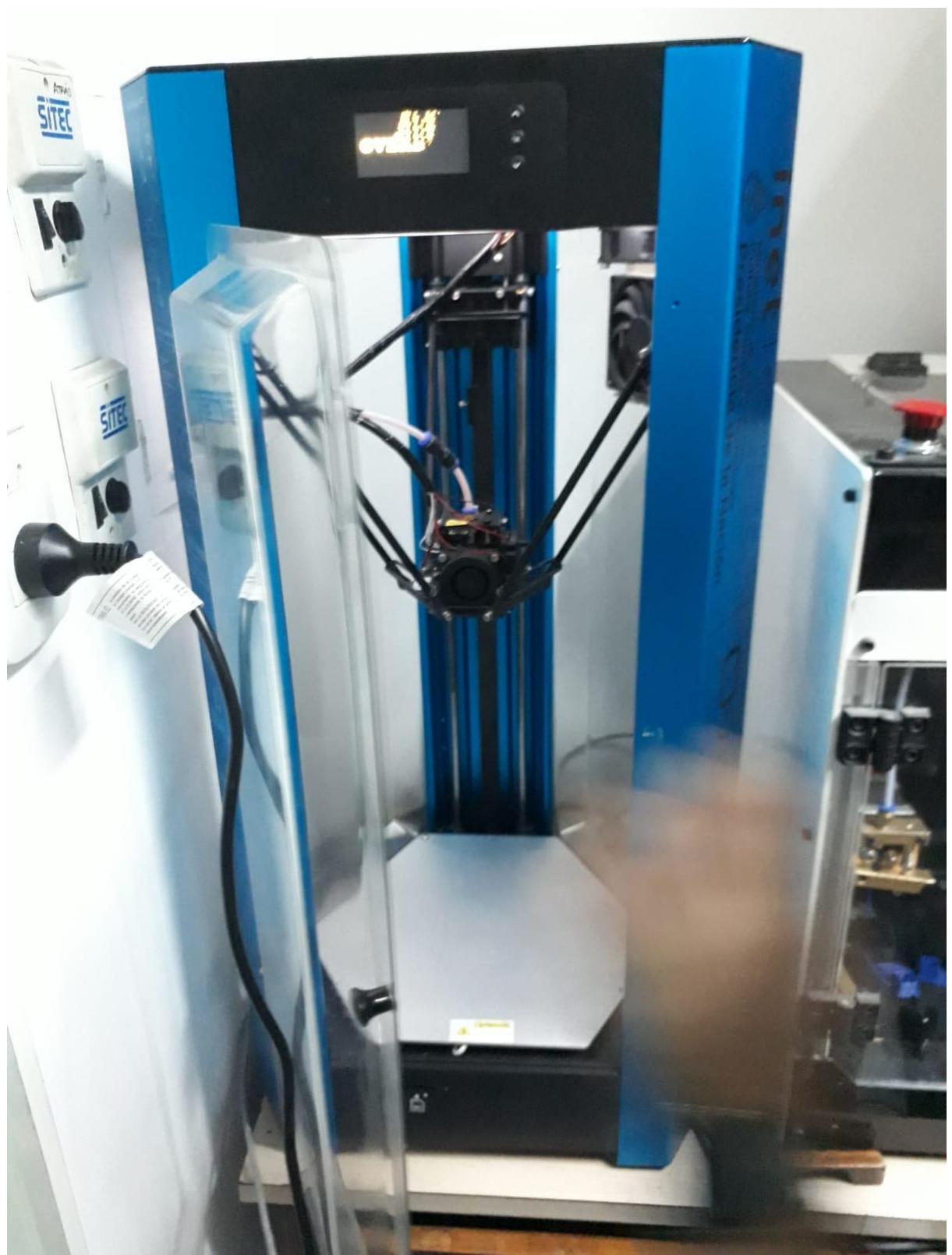
void mover_pata(Servo muslo[], Servo pant[], int angulos_todos[], int reset){
if (muslo[0].read() == 0 and reset == 0){
for (i==0;i<=3;i++){
    muslo[i].write(angulos_todos[0][i]);
    pant[i].write(angulos_todos[1][i]);
}
}else{
for (i==0;i<=3;i++){
    muslo[i].write(angulos_todos[0][-i]);
    pant[i].write(angulos_todos[1][-i]);
}
}
if (reset == 1){
for (i==0;i<=3;i++){
    muslo[i].write(0);
    pant[i].write(15);
}
}
}
}

```

Luego presencié como hacer la pierna en la impresora 3D procedo a dejar las imágenes:

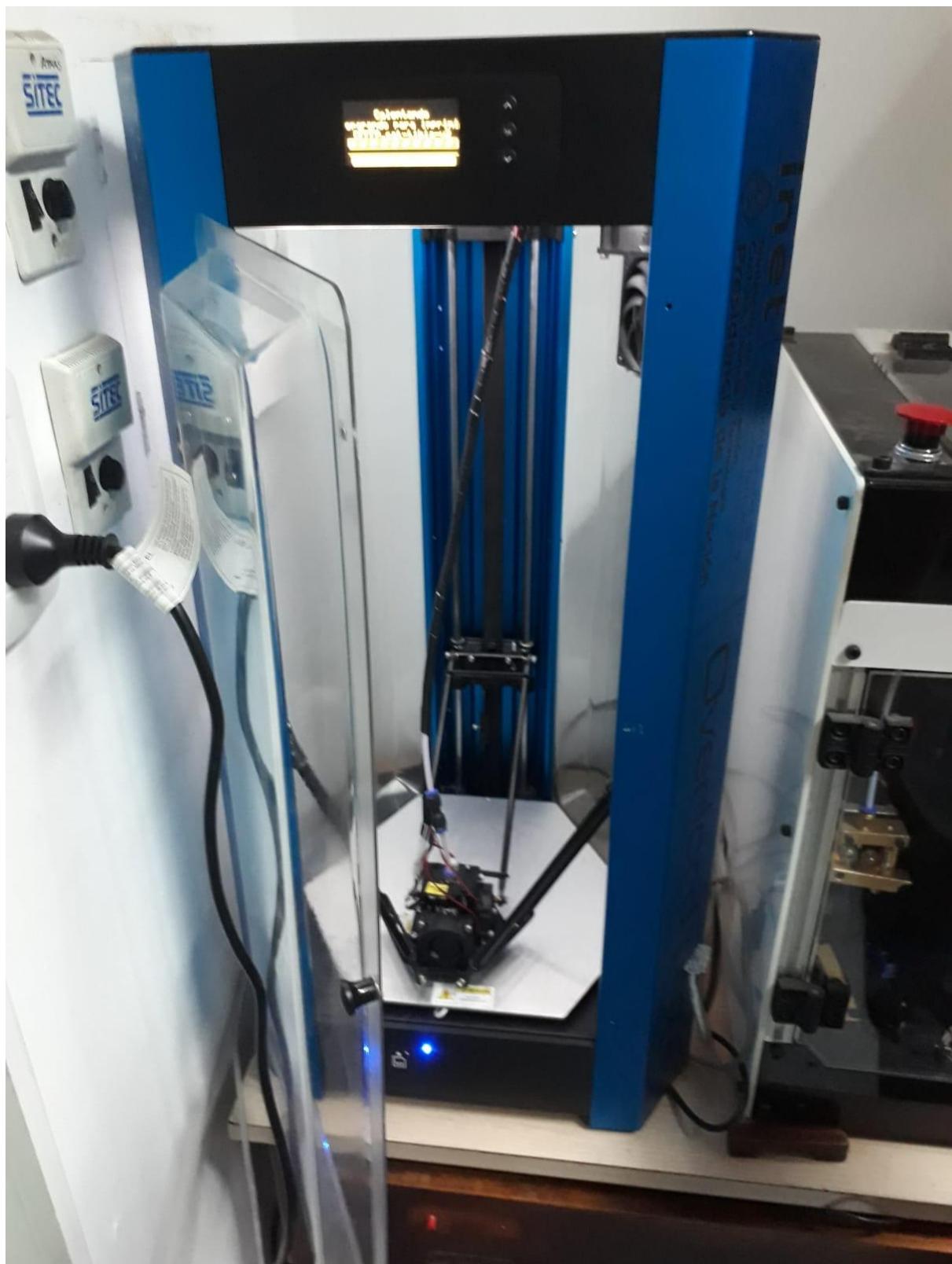


Primero se pone este spray en el suelo donde se imprimira la pieza para que se pegue en el y no se quede en la punta del la impresora.

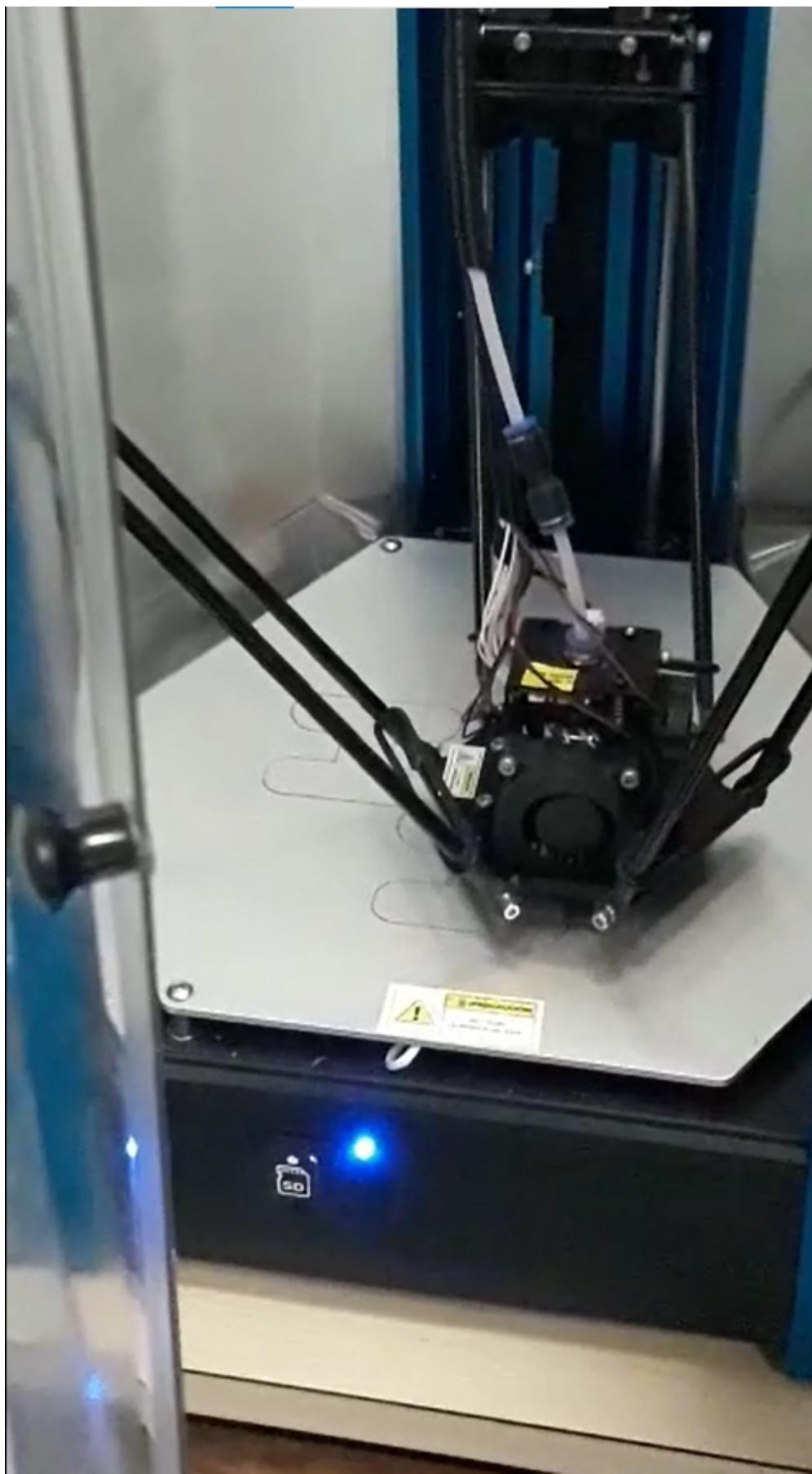




Luego se calienta el piso para que se derrita el spray y se logre pegar



Como vemos aca para esto se utiliza la punta de la impresora



Por ultimo vemos como empieza a imprimir la pieza, en este caso las uniones entre los servomotores de las piernas

Diario de Rorschach 3 de Noviembre de 2022:

Hoy mis compañeros armaron la pierna, como solo tenemos una pierna procedí a hacer un código provvisorio para probar si la pierna funcionaba

Tiene muchos errores que hay que solucionar

```
#include <Servo.h>
Servo muslo_1;
Servo pant_1;

#define boton 4

int reinicio = 0;
Servo muslos[] = { muslo_1 };
Servo pants[] = { pant_1 };
int angulos_1[] = { 80, 0, 80, 0 };
int angulos_2[] = { 0, 15, 0, 15 };

/*int angulos_if[] = {angulos_1[], angulos_2[]};
//int angulos_else[] = {angulos_2, angulos_1};
//int angulos_todos[] = {*angulos_if[], *angulos_else[]};

void setup() {
    // put your setup code here, to run once:
    muslo_1.attach(9);
    pant_1.attach(11);
    pinMode(boton, INPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    if ( digitalRead(boton) == HIGH ){
        reinicio = 1;
    }else{
        reinicio = 0;
    }
    mover_pata(muslos, pants, angulos_todos, reinicio);

}

// Aca modificamos la funcion para que agarre las listas y la opcion de si tiene que caminar
// o no
```

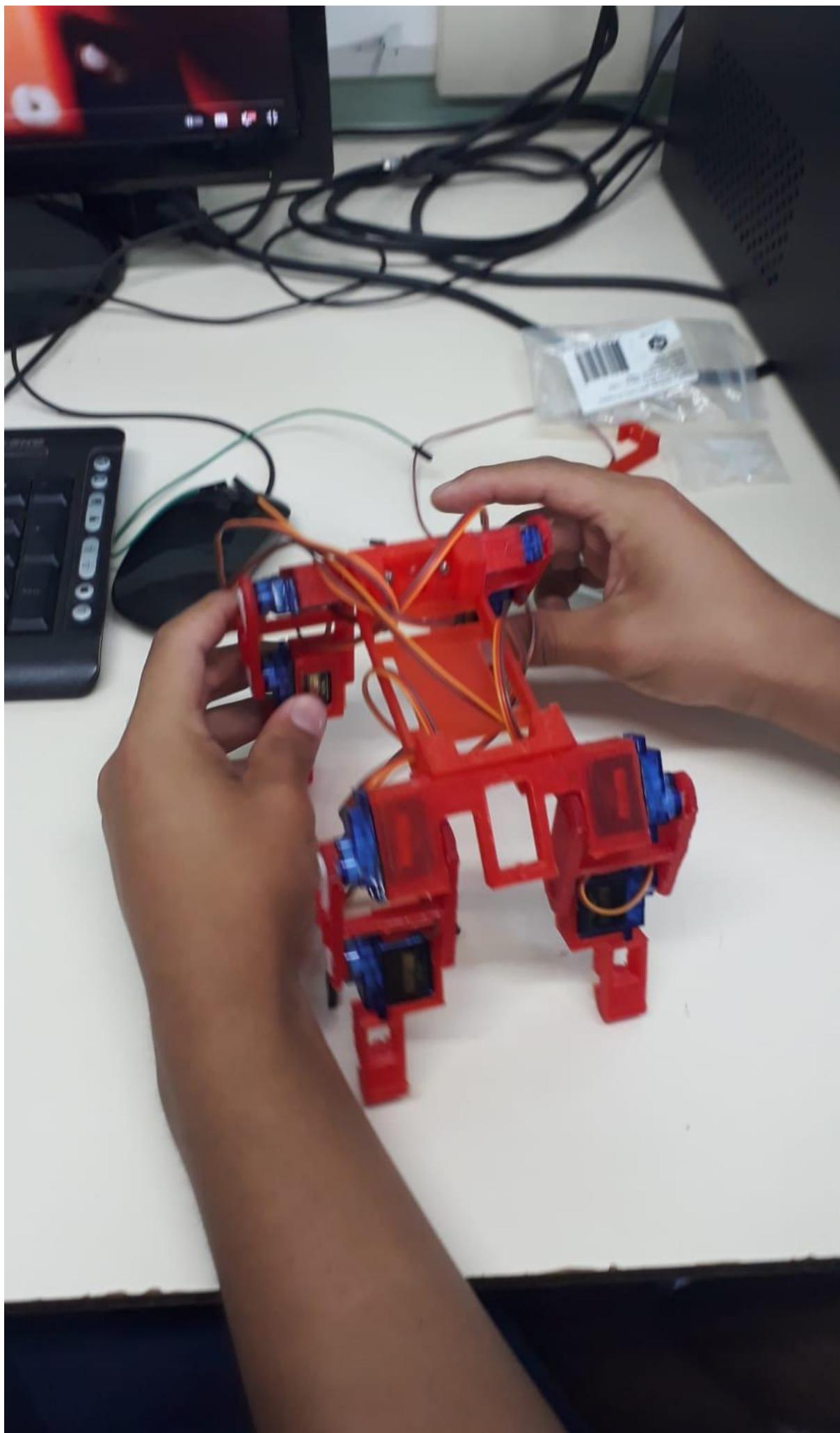
```

void mover_pata(Servo muslo[], Servo pant[], int angulos_todos[], int reset){
// Si el muslo esta en la posicion parado y esta hecho para que camines ejecuta
// La lista de angulos para caminar
if (muslo[0].read() == 0 and reset == 0){
    for (int i=0;i<=3;i++){
        muslo[0].write(angulos_todos[0][i]);
        pant[0].write(angulos_todos[1][i]);
    }
}
// Si no pasa eso que ejecute el comando para que camine el otro pie
else{
    for (int i=0;i<=3;i++){
        muslo[0].write(angulos_todos[0][-i]);
        pant[0].write(angulos_todos[1][-i]);
    }
}
// Si el reset esta activo que reinicie las piernas
if (reset == 1){
    muslo[0].write(0);
    pant[0].write(15);
}
}

```

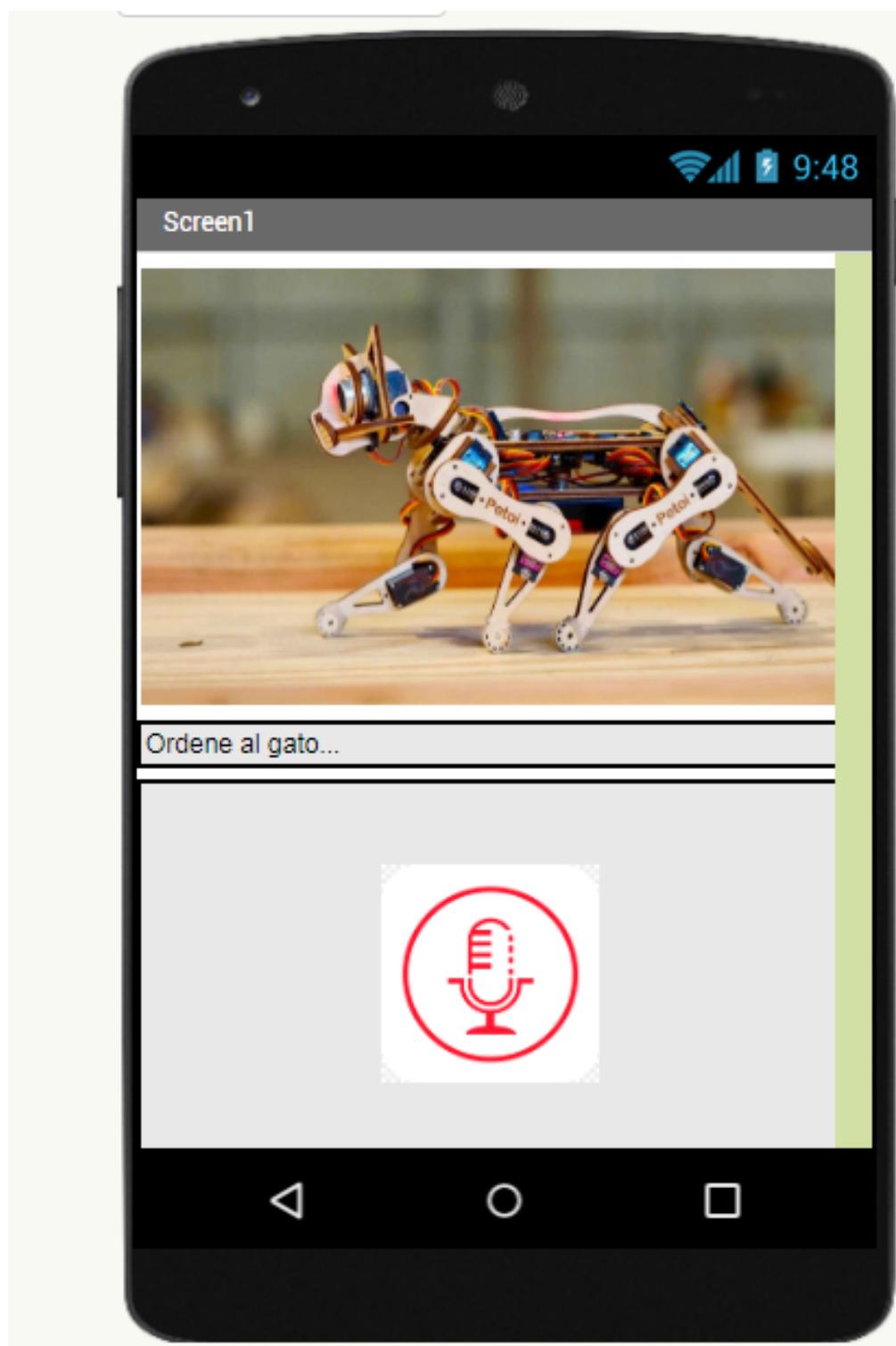
Diario de Rorschach hoy es 10 de Noviembre de 2022:

Hoy mientras mis compañeros se disponen a juntar las piezas del robot empezare a investigar sobre el app inventor para hacer la aplicación del celular.



(esta foto esta un poco adelantada pero no tenia antes)

 Tutorial: Tu primera app con MIT App Inventor 2



Esta seria la interfaz con un boton donde con el audio de google podrias decir tu orden, que luego se escribiria en la zona de “ordene el gato...”



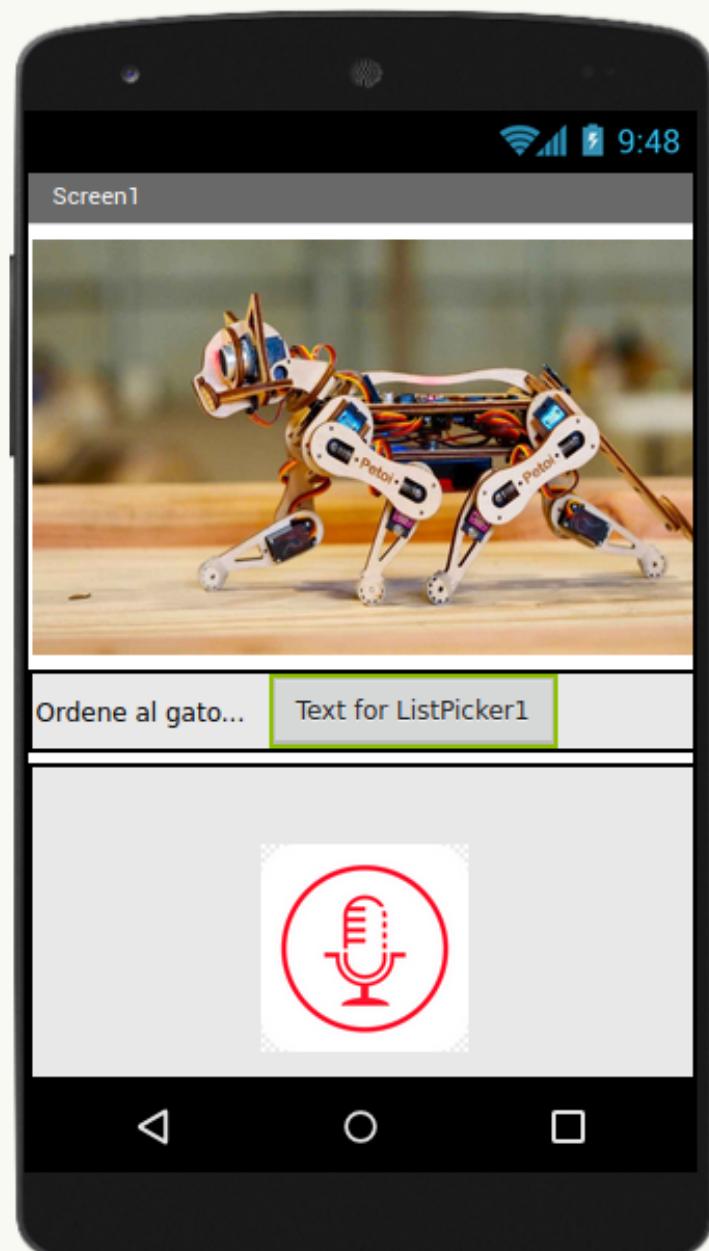
Este es el mecanismo que cuando tocas el boton llama al dispositivo que escucha y lo convierte en un mensaje

Diario de Rorschach hoy es 17 de Noviembre de 2022:

Hoy es el anteultimo dia de trabajo, la app esta lista solo hay que probar que el bluetooth funcione correctamente, el diseño es simple pero funcional, mientras tanto mis compañeros están terminando de conectar por los que los voy a ir a ayudar

Display hidden components in Viewer

Phone size (505,320) ▾

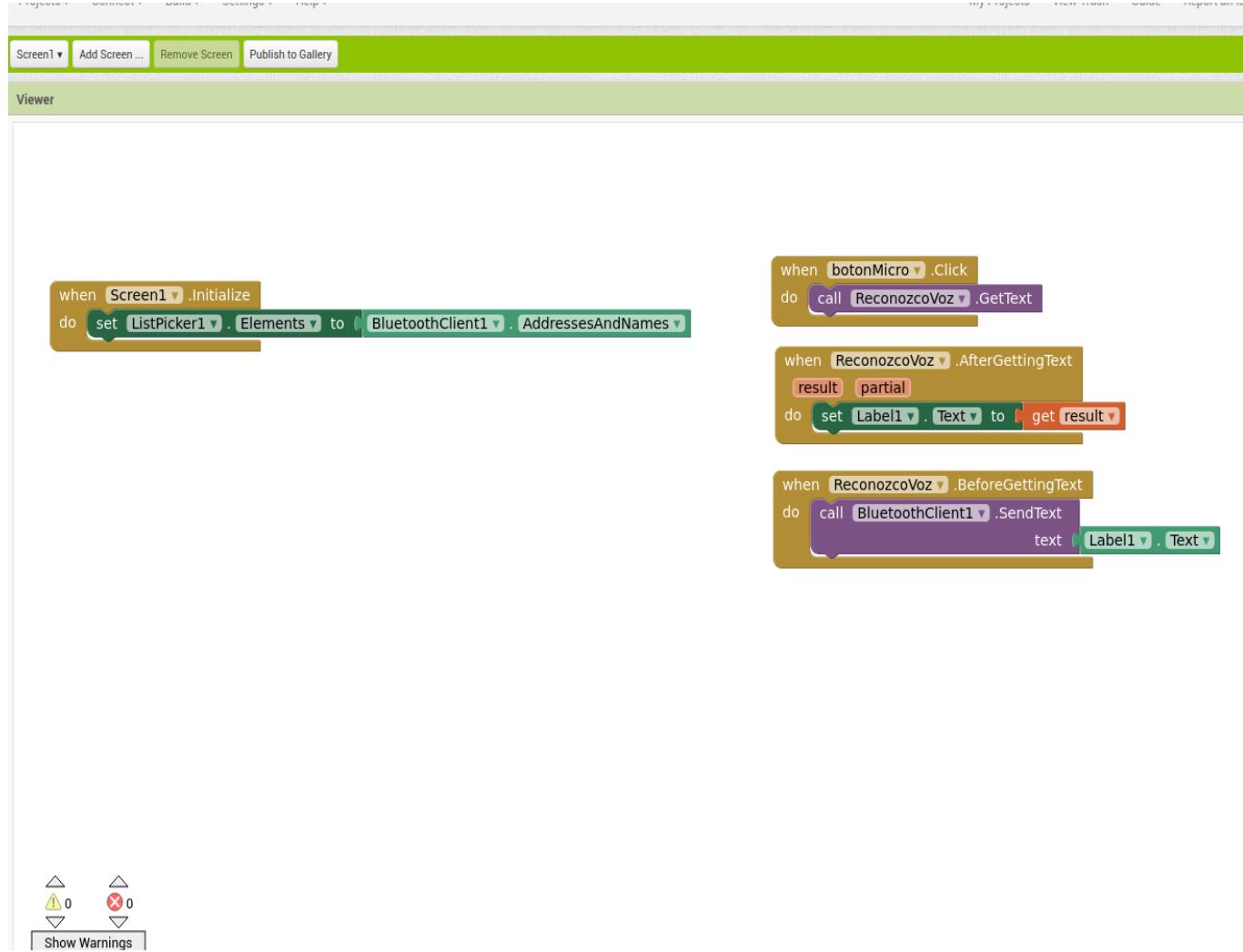


Non-visible components



ordenHablada ReconozcoVoz BluetoothClient1





El cambio de la interfaz fue para agregar bluetooth por donde se conectara con el arduino con su placa de Bluetooth

Lo de abajo es el codigo para conectar a bluetooth,

When screen1 .initialize... este crea una lista con los dispositivos bluetooth a tu alcance

when botonMicro.click... esto es cuando toques el boton te reconozca la voz y la convierta en texto

When reconoczcoVoz.AfterGettingText... Establecer el texto en el label1

When reconoczcoVoz.BeforeGettingText... Cuando ese texto se establezca se lo va a mandar como resultado al dispositivo bluetooth conectado

```
#include <SoftwareSerial.h>
SoftwareSerial BT( 10 , 11 );

void setup() {
    // put your setup code here, to run once:
    BT.begin( 38400 );
    Serial.begin( 38400 );
}

void loop() {
    // put your main code here, to run repeatedly:

    if( BT.available() )
    {
        Serial.println( BT.read() );
    }

    if( Serial.available() )
    {
        BT.write( Serial.read() );
    }
}
```

Este es un código para configurar la placa de bluetooth del arduino

Diario de Rorschach hoy es 24 de Noviembre de 2022:

Solucioné unos errores de código y ahora vamos a probar si funciona

Procedí a agregarle los otros servomotores que no estaban antes y agregarles sus respectivos pines, y corregí errores en las listas agregando * y **, esto también en las listas.

Se solucionó el llamado de los ángulos para que llame a los ángulos correctos.

Luego hago una prueba para ver su funcionamiento, tambien le agrego un delay para que no intente ir directamente al siguiente ángulo y convulsione, hay que ver que tan bien este el delay con el tiempo pero eso para despues

```
#include <Servo.h>
Servo muslo_1;
Servo pant_1;
Servo muslo_2;
Servo pant_2;
Servo muslo_3;
Servo pant_3;
Servo muslo_4;
Servo pant_4;

#define boton 4

int reinicio = 0;

Servo muslos[] = { muslo_1 , muslo_2 , muslo_3 , muslo_4};
Servo pants[] = { pant_1 , pant_2 , pant_3 , pant_4 };

int angulos_1[] = { 80, 0, 80, 0};
int angulos_2[] = { 0, 15, 0, 15};

int* angulos_if[] = {angulos_1, angulos_2};
int* angulos_else[] = {angulos_2, angulos_1};
int** angulos_todos[] = {angulos_if, angulos_else};

void setup() {
    // put your setup code here, to run once:
    muslo_1.attach(8);
    pant_1.attach(7);
    muslo_2.attach(3);
    pant_2.attach(2);
```

```

muslo_3.attach(4);
pant_3.attach(5);
muslo_4.attach(9);
pant_4.attach(6);
pinMode(botón, INPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    if ( digitalRead(botón) == HIGH ){
        reinicio = 1;
    }else{
        reinicio = 0;
    }
    mover_pata(muslos, pants, angulos_todos, reinicio);
}

```

```

void mover_pata(Servo* muslo, Servo* pant, int*** angulos_todos,
int reset){
    if (muslo[0].read() == 0 and reset == 0){
        for (int i=0;i<=3;i++){
            muslo[i].write( ((angulos_todos[0])[0])[i] );
            pant[i].write( ((angulos_todos[0])[1])[i] );
            digital
        }
    }else{
        for (int i=0;i<=3;i++){
            muslo[i].write( ((angulos_todos[1])[0])[i] );
            pant[i].write( ((angulos_todos[1])[1])[i] );
        }
    }
    if (reset == 1){
        for (int i=0;i<=3;i++){

```

```
    muslo[i].write( ((angulos_todos[0])[0])[1] );
    pant[i].write( ((angulos_todos[0])[1])[1] );
}
}

delay(100);
}
```

#####PROBANDO

```
#include <Servo.h>
Servo muslo_1;
Servo pant_1;
Servo muslo_2;
Servo pant_2;
Servo muslo_3;
Servo pant_3;
Servo muslo_4;
Servo pant_4;

void setup() {
    // put your setup code here, to run once:
    muslo_1.attach(8);
    pant_1.attach(7);
    muslo_2.attach(3);
    pant_2.attach(2);
    muslo_3.attach(4);
    pant_3.attach(5);
    muslo_4.attach(9);
    pant_4.attach(6);
}

void loop() {
    // put your main code here, to run repeatedly:
    muslo_1.write(0);
```

}

Diario de Rorschach hoy es 30 de Noviembre de 2022:

Hoy iba a ser dia de terminar el projecto pero por razones que ya sabes no va a poder ser por lo que procedo a adjuntar hasta donde llegamos.

