
Curso: 2019 - 2020

Talkative Gloves - Alexa

Tecnologías Emergentes

José Javier Alonso Ramos



**UNIVERSIDAD
DE GRANADA**

Índice

Info	3
Hardware necesario	3
Software necesario	3
Librerías	3
Código	4
Arduino	4
Raspberry	5
Construcción	6
Sensores	6
Materiales	6
Preparación	7
Guante	9

Info

El proyecto explica como convertir unos guantes normales en unos que nos dan la posibilidad de hablar tan solo con gestos de nuestra mano. Cada gesto se traducirá a una palabra o frase que será dicha por un altavoz. Esta implementación se dirige específicamente al uso de Alexa mediante gestos.

Hardware necesario

- Guantes
- Sensores de flexión
- Raspberry
- Arduino
- Altavoz
- Alexa (para probar su uso - no es indispensable)

Software necesario

Librerías

Para poder leer el puerto serie de arduino en raspberry desde un programa en python.

```
1 sudo -H pip3 install pyserial
```

Librería que permite pasar texto escrito a voz.

```
1 sudo -H pip3 install pyttsx3
```

Programa que usa la librería pyttsx3 para reproducir sonido desde el ordenador o, en nuestro caso, desde la raspberry.

```
1 sudo apt-get update && sudo apt-get install espeak
```

Código

Arduino

El código de arduino es muy básico. Tan solo tendremos que leer analógicamente los sensores de flexión y decidir cuáles son los dedos que están flexionados según el valor que devuelvan. El umbral a partir del cuál consideramos que están flexionados será arbitrario y, seguramente, diferente para cada dedo.

Lo que el arduino manda por el puerto serie a la raspberry es un String (una cadena de texto) de 5 dígitos (uno para cada dedo en orden: Pulgar, Ínlice, Corazón, Anular y Meñique) siendo el valor 1 si el dedo está flexionado y 0 si no lo está. De esta forma, una cadena de tipo '10111' indicaría que sólo el dedo índice está estirado o '00110' indicaría que estás jugando a ser *Spiderman*.

```
1 String mensaje = '';
2
3 int pulgar = analogRead(Pulgar);
4 if(pulgar < umbralPulgar)
5     mensaje += '1';
6 else
7     mensaje += '0';
8
9 int indice = analogRead(Indice);
10 if(indice < umbralIndice)
11     mensaje += '1';
12 else
13     mensaje += '0';
14
15 int corazon = analogRead(Corazon);
16 if(corazon < umbralCorazon)
17     mensaje += '1';
18 else
19     mensaje += '0';
20
21 int anular = analogRead(Anular);
22 if(anular < umbralAnular)
23     mensaje += '1';
24 else
25     mensaje += '0';
26
27 int menique = analogRead(Menique);
28 if(menique < umbralMenique)
```

```

29     mensaje += '1';
30 else
31     mensaje += '0';
32
33 Serial.println(mensaje);

```

Raspberry

Este código también es muy sencillo pero como usa bibliotecas no muy habituales lo vamos a repasar un poco. Lo puedes ver completo [aqui](#).

Lo primero que hacemos es inicializar el proceso de *Test to Speech* e indicar que la voz que vamos a usar en este caso es en *español*:

```

1 engine = pytsxs3.init()
2 engine.setProperty('voice', 'spanish')

```

Para poder ver todos los tipos de voces de las que dispone esta biblioteca podemos hacer lo siguiente:

```

1 voces = engine.getProperty('voices')
2 for i in range(len(voces)):
3     println('Voz: ', voces[i].id)

```

Ahora creamos una variable para leer el puerto serie al que se conecta el arduino. **ttyACM0** se trata del puerto serie por el que leemos los datos (puede que en tu raspberry sea otro), la **velocidad de comunicación** es a tu elección pero 115200 baudios es el valor que mejores resultados me ha dado y establecemos un tiempo de **timeout** de 3 segundos. Si tras este tiempo no hemos conseguido establecer conexión nos dará un error:

```

1 arduino = serial.Serial('/dev/ttyACM0', baudrate=115200, timeout=3.0)

```

Y por último leemos los datos que nos envía el arduino con la sentencia **readline** y hacemos un casting a String. La segunda línea formatea el texto de entrada para solo quedarnos con los 5 dígitos que nos interesan:

```

1 val = str(arduino.readline())
2 val = val[2:7]

```

Con esto hecho solo nos queda establecer distintas frases o palabras que queremos decir cuando se

de cierta combinación.

```
1 # Si flexionamos solo pulgar
2 if val == '10000':
3     engine.say('Hola')
4 # Si flexionamos solo índice
5 elif val == '01000':
6     engine.say('Adiós')
7 # Si flexionamos solo corazón
8 elif val == '00100':
9     engine.say('Soy un guante que habla')
10 ...
11 ...
12 # Si flexionamos índice, corazón y meñique
13 elif val == '01101':
14     engine.say('Tengo hambre')
15 ...
16
17 # Ejecutamos el TTS
18 engine.runAndWait()
```

Como hemos dicho antes, este ejemplo en concreto se hizo para comunicarse con el asistente de Amazon, Alexa. Por eso la única sentencia que lee el código de ejemplo en un principio es '11111' (puño cerrado). Esta sentencia reproduce la palabra 'Alexa', lo que invoca al asistente y sólo después de hacer esto el programa es capaz de leer más combinaciones para darle órdenes.

Construcción

Sensores

Como los sensores de flexión pueden resultar caros para un proyecto casero y más si queremos utilizar todos los dedos de la mano o incluso las dos manos vamos a ver como fabricar los nuestros propios.

Materiales

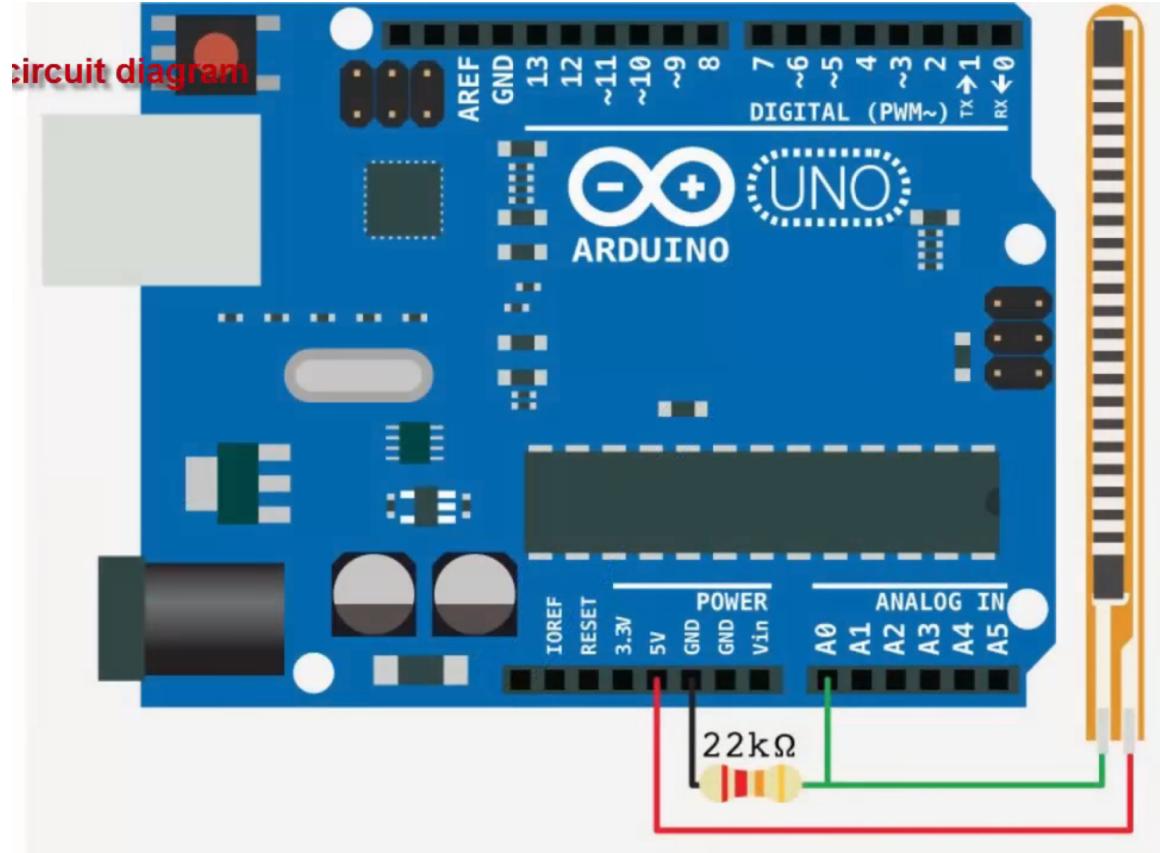
- Papel
- Papel de aluminio
- Lápiz o carboncillo
- Cables

Preparación

Tendremos un sensor para cada dedo. Cortamos un trozo de papel del ancho del dedo donde vaya a ir el sensor y de longitud el doble. Lo dobraremos por la mitad longitudinalmente y una mitad la coloreamos entera con el lápiz sin dejar ningún hueco en blanco y apretando lo suficiente como para dejar bastante carboncillo en ella. En la otra mitad pondremos dos tiras de papel de aluminio de manera longitudinal y sin que se toquen. Al extremo de estas dos tiras conectaremos dos cables que serán los que conectemos con el arduino.

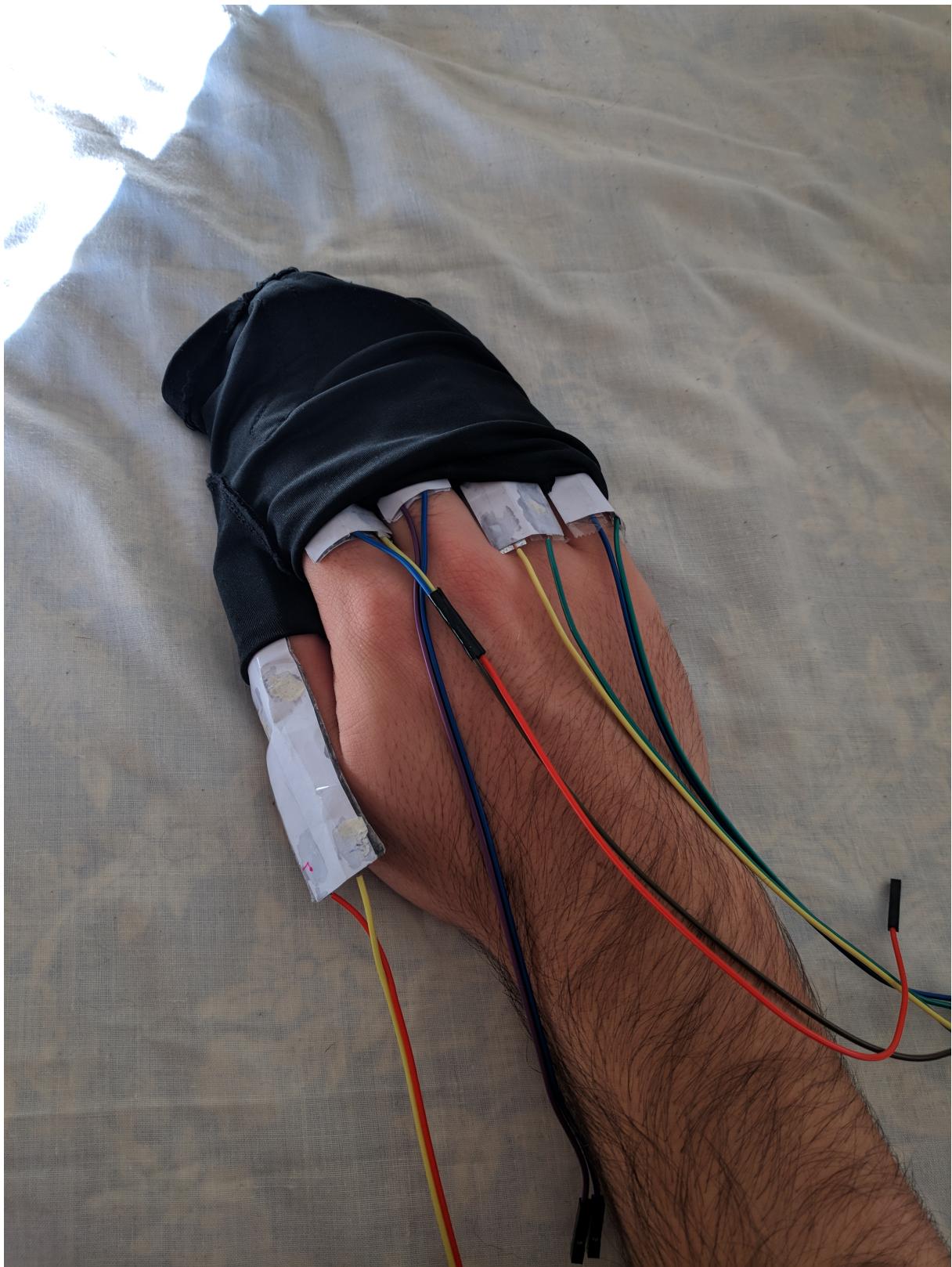


La conexión a arduino se realiza de la siguiente manera (da igual qué cable consideremos positivo o negativo):



Guante

Una vez listos los sensores solo queda introducirlos en el guante. Para un mejor funcionamiento es recomendable situarlos sobre los dedos y no debajo. Podemos coserlos o pegarlos al interior del guante pero asegúrate de que no se quedan muy rígidos y siguen siendo maleables.



Y este sería el aspecto del guante acabado. Como vemos no se notan para nada los sensores flexibles. Si trabajamos un poco más en la ocultación de los cables y la disposición del arduino y la raspberry podríamos tener un sistema vestible que pase totalmente desapercibido.



Figura1: Guante final