



Braillinox

“ Aprende jugando, lee sin limites”

Lider de proyecto Héctor Méndez

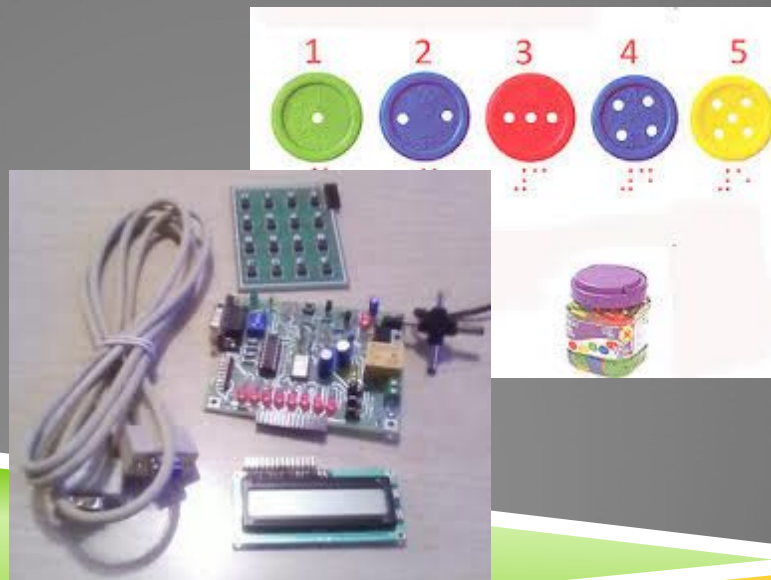
Objetivo



Desarrollar un dispositivo de bajo costo que aproveche la miniaturización de semiconductores a tal grado que en un espacio pequeño se pueda desarrollar una herramienta que ayude a los ciegos y débiles visuales a aprender alfabeto braille de una forma divertida, interesante y muy amigable.

Descripción del proyecto.

Se realizó un sistema didáctico para que ellos aprendan a leer el alfabeto Braille de un modo fácil y sencillo con juegos y retos que ayudaran a que nunca se les olviden las letras o las confundan en su composición, por eso el dispositivo estará dotado de una microcomputadora que auxiliará al sistema con patrones de reconocimiento de voz, micro salidas censadas y con actuadores lineales acondicionados para poder manejar salidas en el orbe de los nA.



Actividades realizadas al momento.

- 🔑 Planteamiento del problema.
- 🔑 Objetivo.
- 🔑 Desarrollo de la doc.
- 🔑 estructura
- 🔑 Etapas de potencia
- 🔑 Fuentes
- 🔑 Control con Raspberry Pi B
- 🔑 Procesado de señales
- 🔑 Scripts por terminal con Pocketsphinx
- 🔑 Pruebas.



Justificación del Proyecto.

Para una persona con facultades normales es fácil poder interactuar con el mundo mediante la escritura y la lectura, pero, para un ciego o débil visual no es posible ya que no existe una retroalimentación de cómo se hace, por esto se crea el idioma Braille (un alfabeto táctil), pero su aprendizaje es complicado y tardado en niños a temprana edad ya que necesita concentración y mucha paciencia, en adultos que por alguna circunstancia perdieron la visión también es difícil de aprender ya que no se tiene la sensibilidad necesaria para identificar los patrones de cajetín (cuadro generador), respondiendo a esta necesidad creemos conveniente implementar nuevas tecnologías a la enseñanza y lograr hacer más rápido el proceso de aprendizaje del Alfabeto táctil.

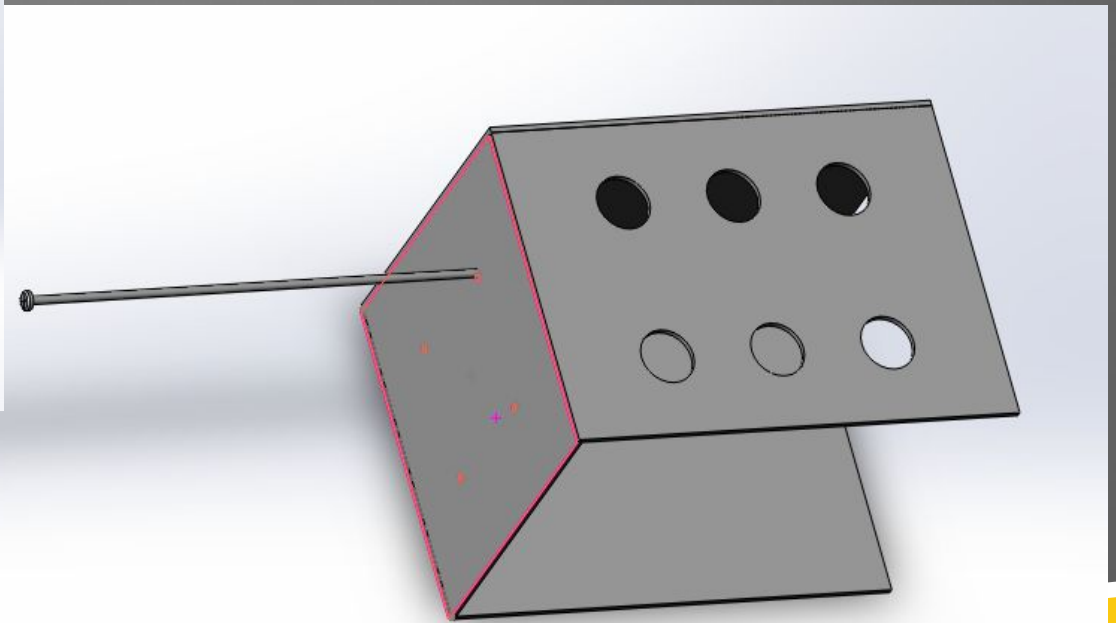
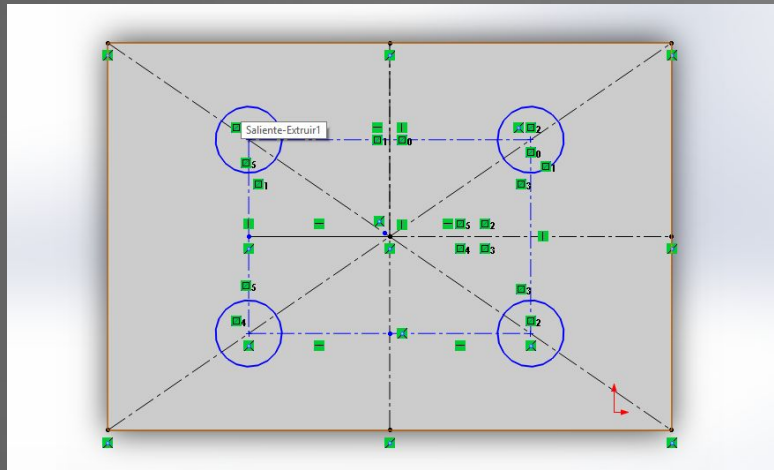
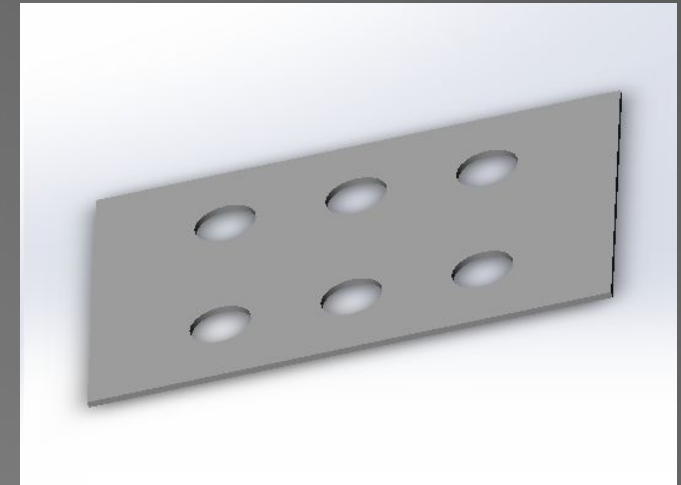


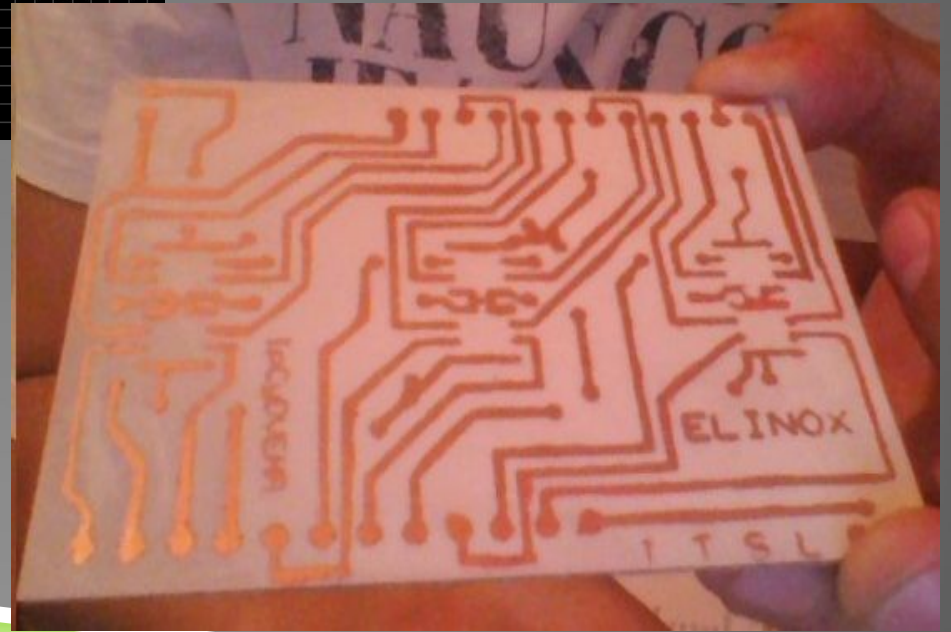
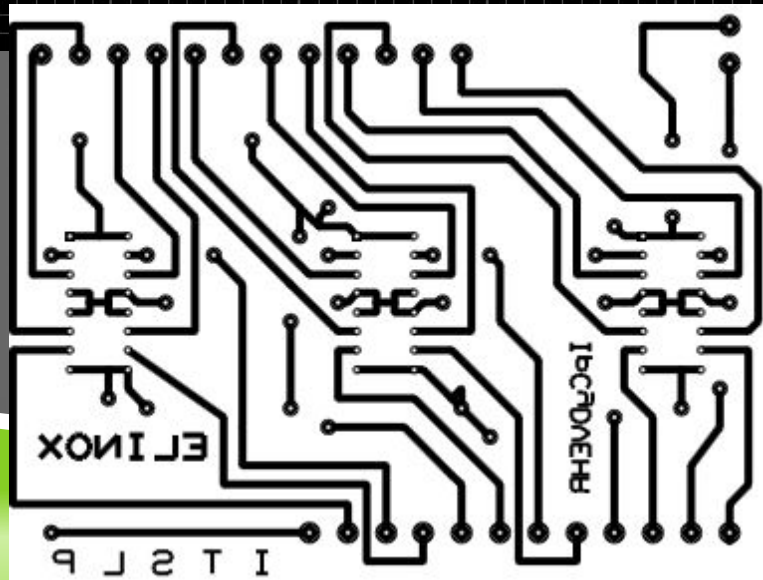
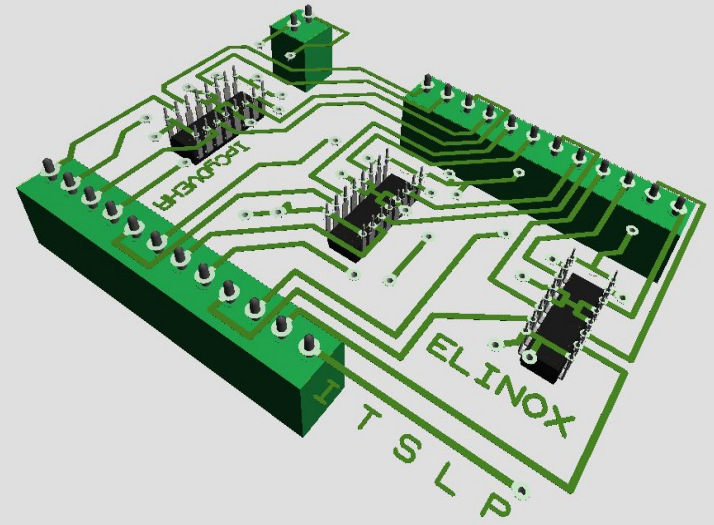
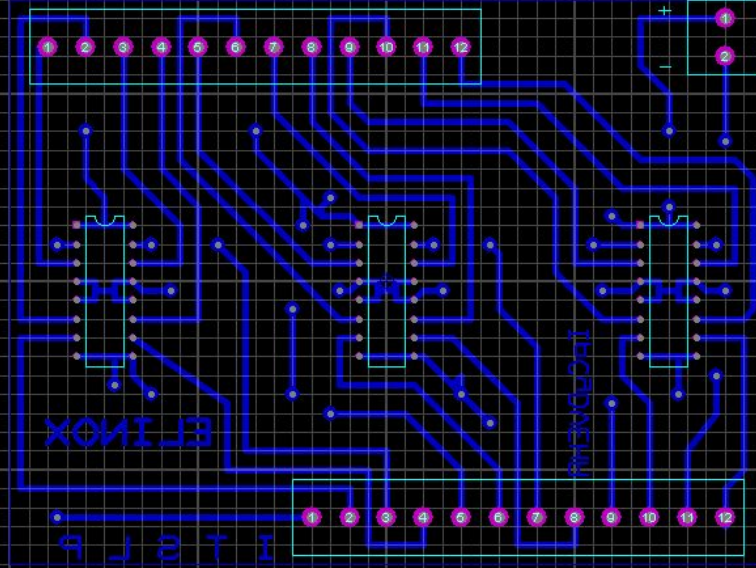
Delimitaciones.

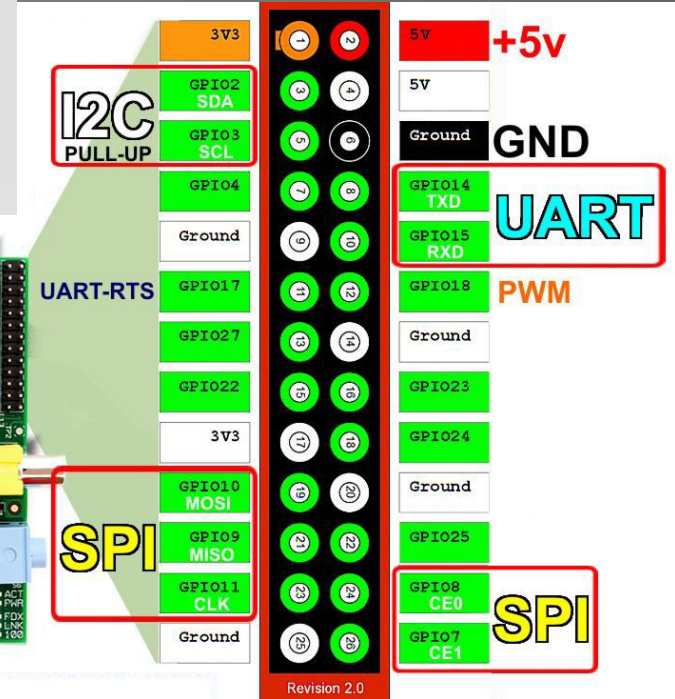
- ❏ El problema que se tiene principalmente con Brailin es que es un dispositivo enfocado a niños en edades tempranas y eso se debe corregir, tanto niños como adultos tienen la necesidad de aprender Braille (es infantil y no sirve para adultos).
- ❏ Requiere un mercado muy especial (institutos para no videntes y d é biles visuales).
- ❏ Una de las delimitaciones en este proyecto más relevantes es el análisis de los arm ó nicos para el patr ó n del reconocimiento de voz y el hardware y el software que se va a usar.
- ❏ S ó lo aplica para ciegos y d é biles visuales (no aplica en enseñanza para sordo ciegos).
- ❏ No es un dispositivo que por su composici ó n pueda costear una familia.
- ❏ Es costoso desarrollarlo.

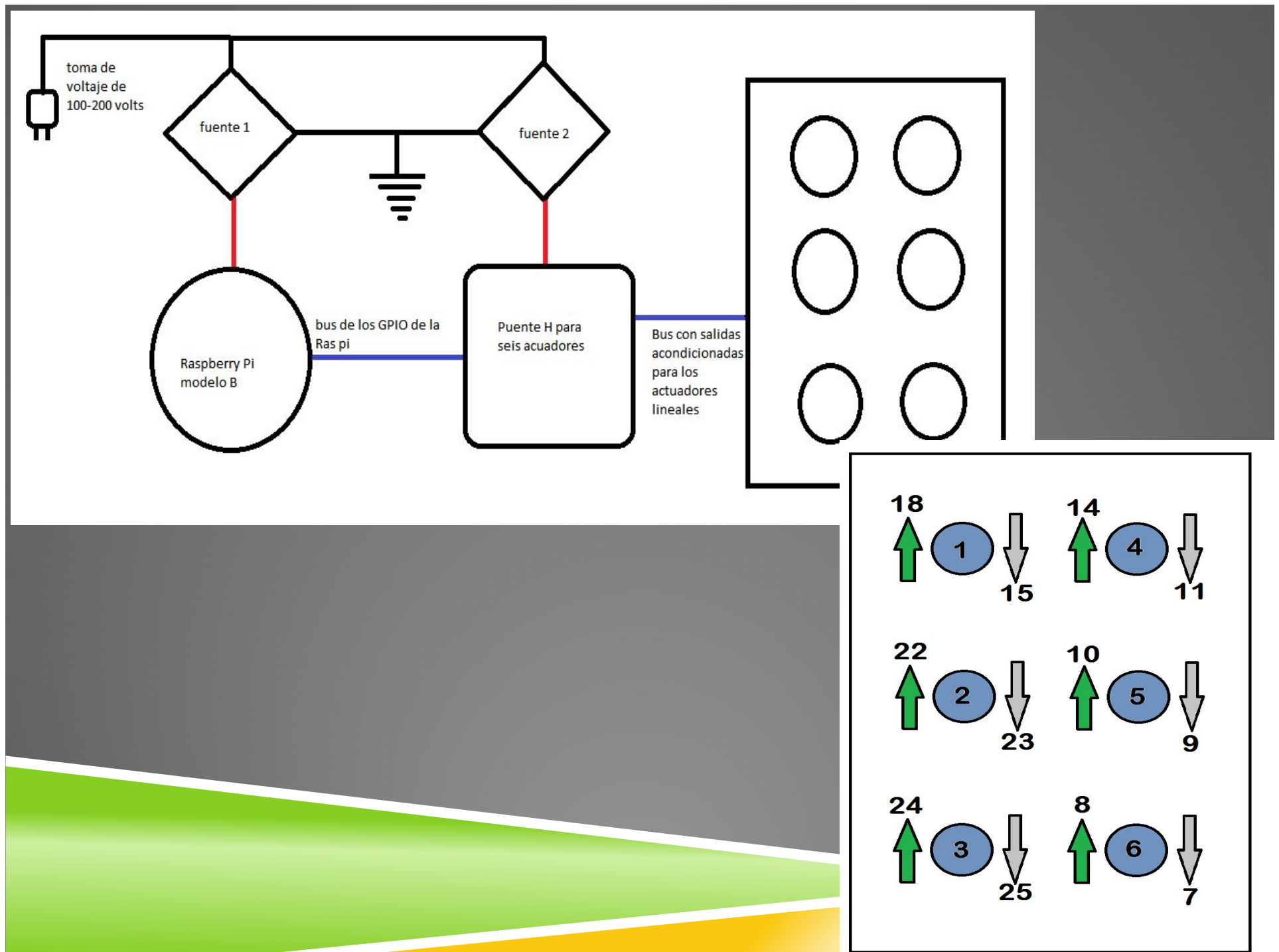


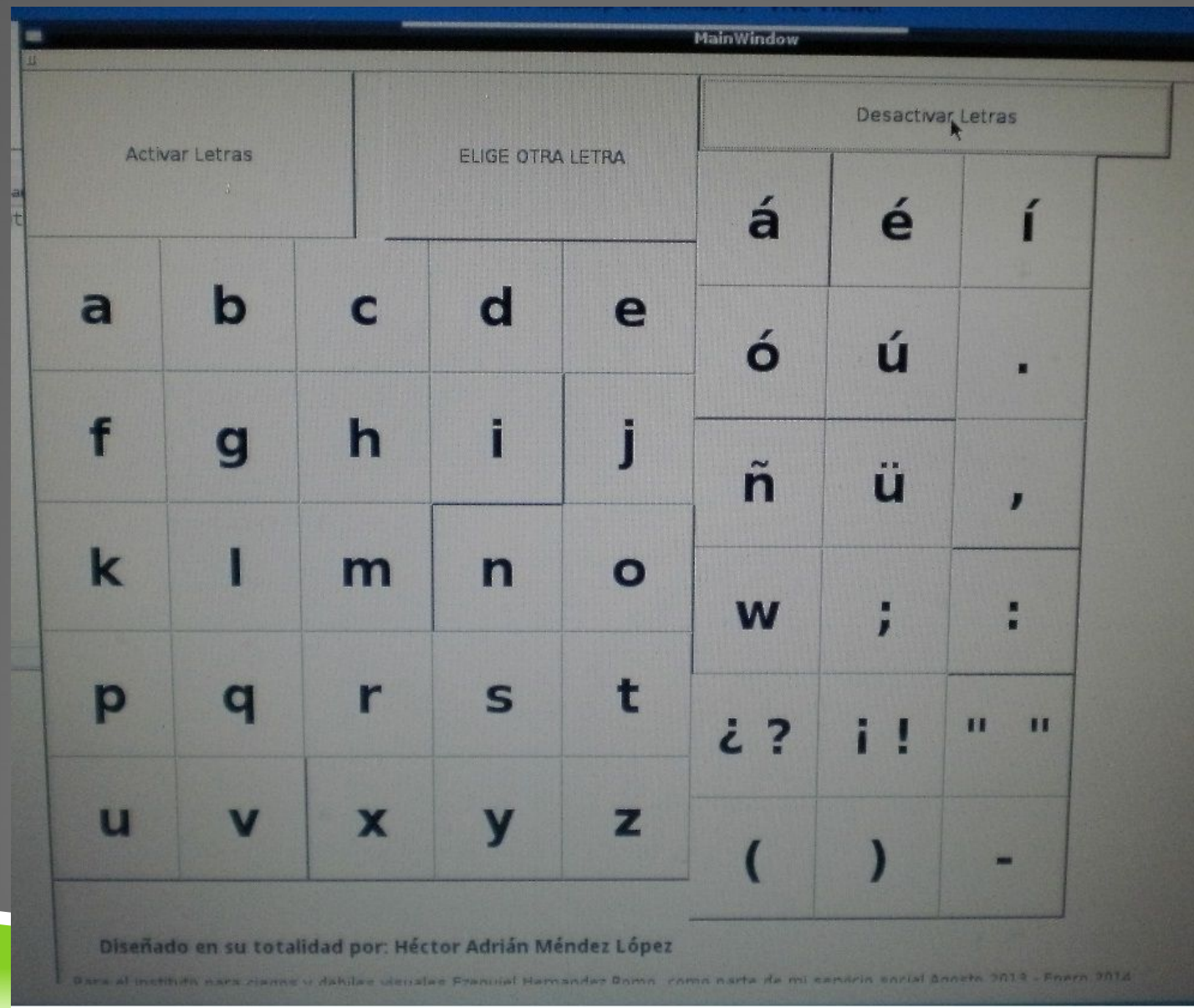
Desarrollo del Proyecto





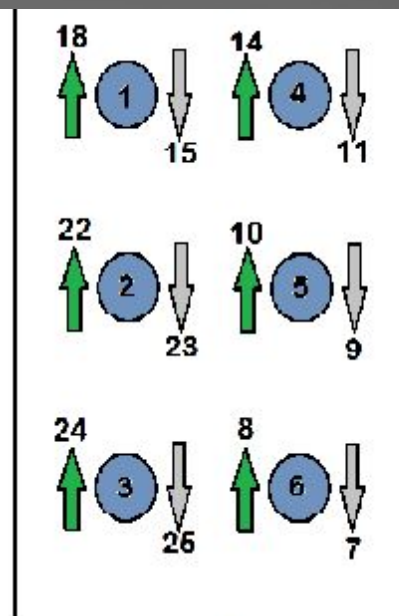






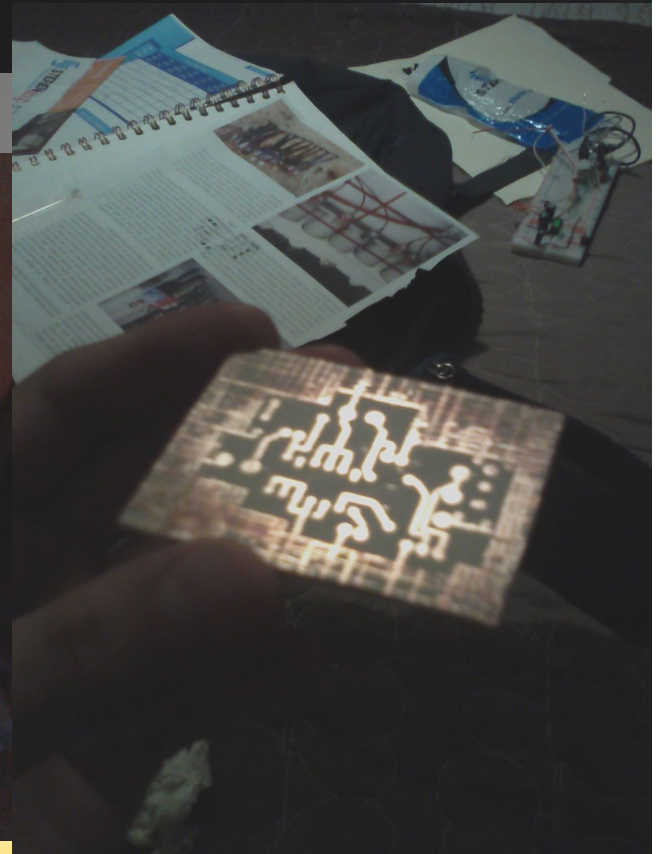
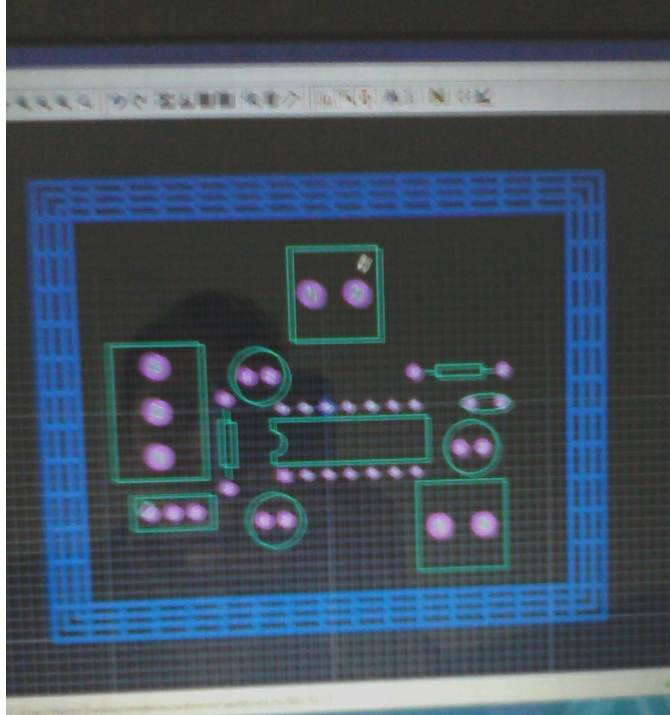
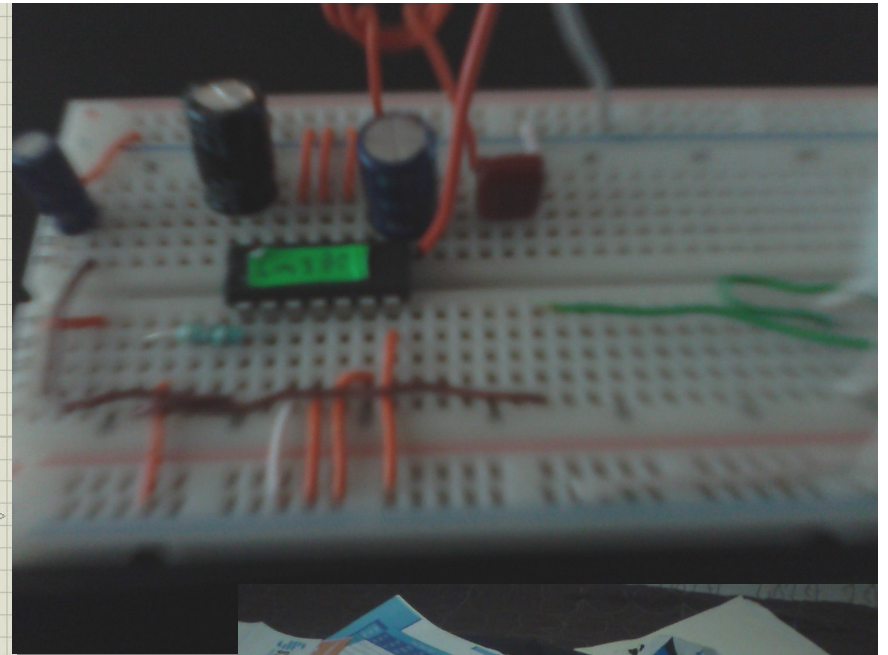
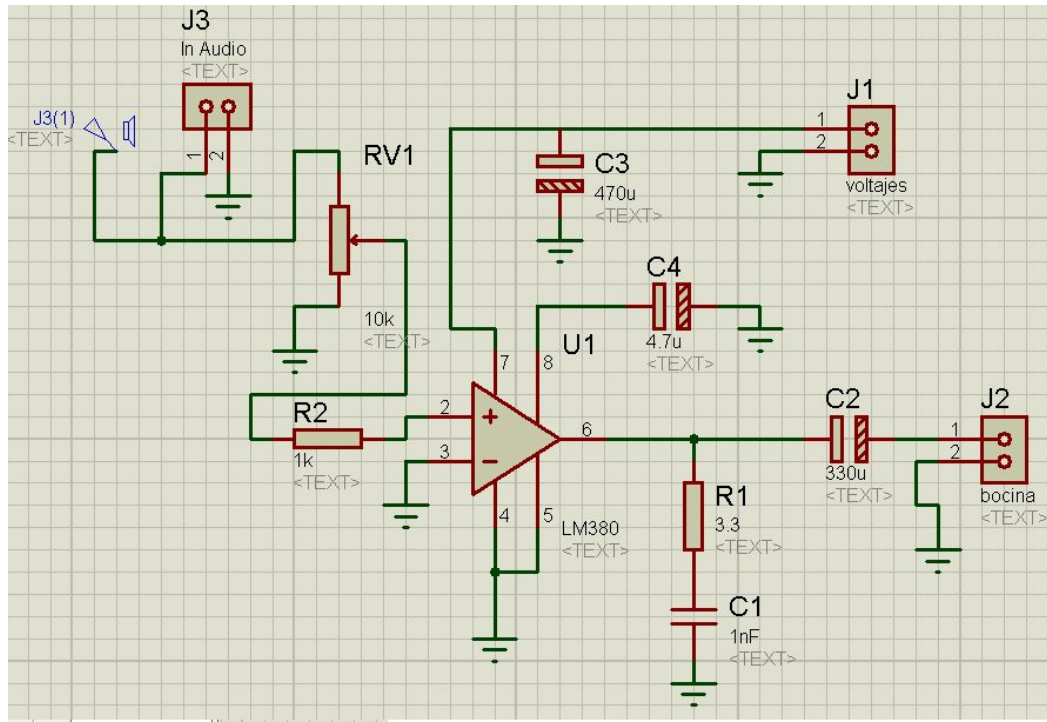


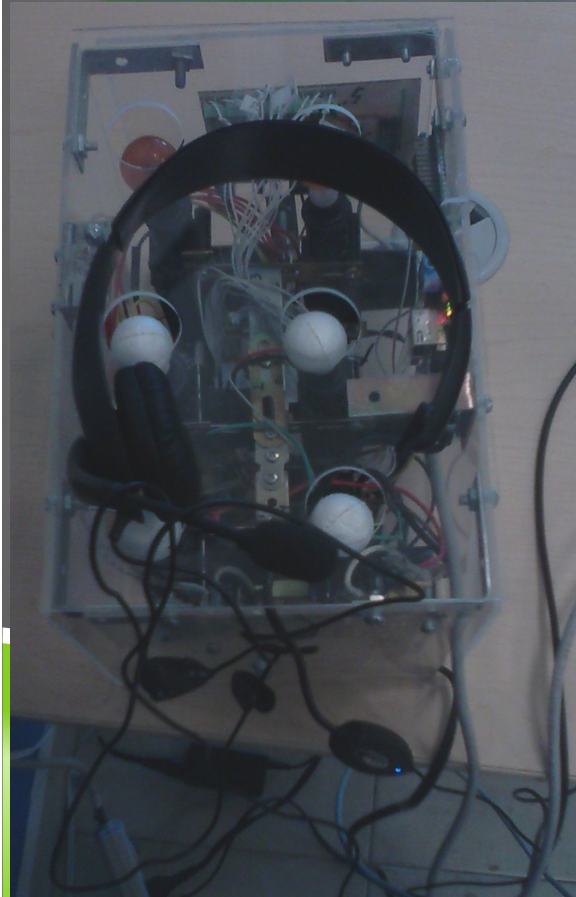
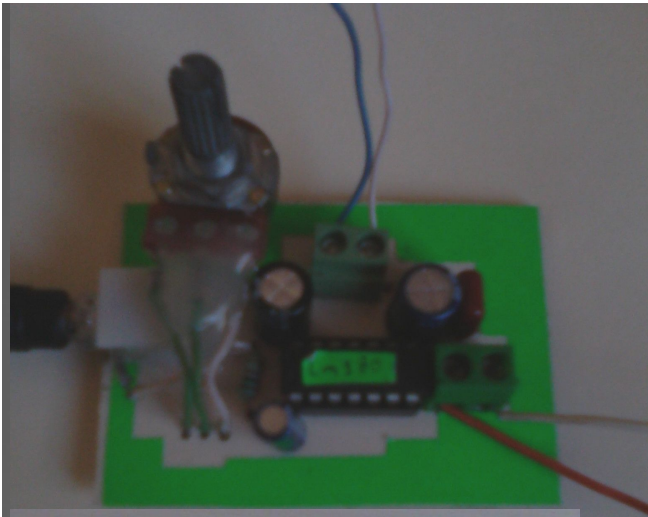
Digitaliza y añade filtros discretos pasa bajos y pasa altos.

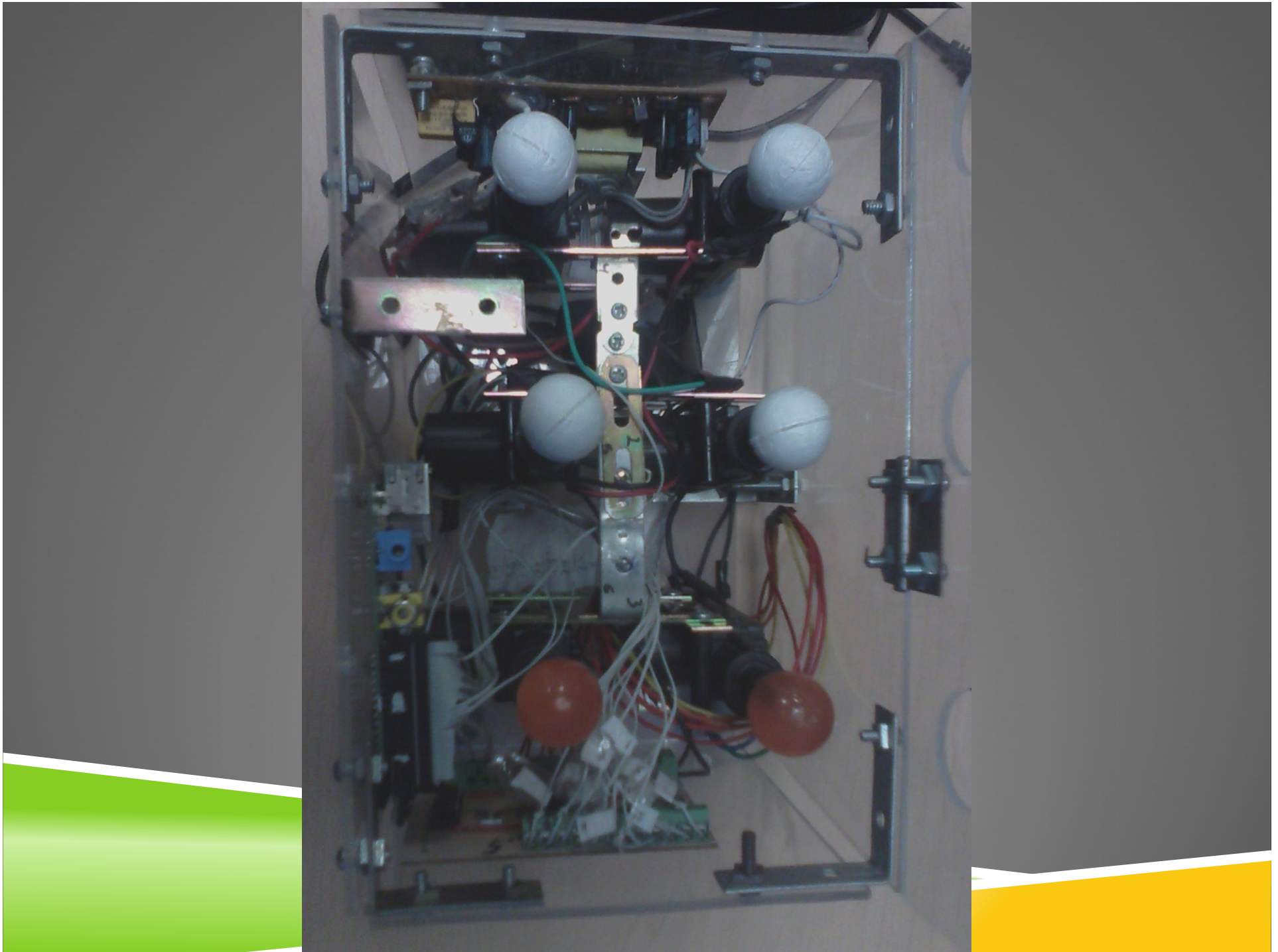


Finalmente se muestra la letra en alfabeto Braille según se necesite

El audio se digitaliza y entra a la Raspi, después de justificarse entra a una serie de filtros discretos para generar comparaciones en base a sentencias.







Paquetes de software libre

Raspbian

qt4

qtcreator

Bison

libasound2-dev

sphinxbase

pocketsphinx

Sphinx knowledge base generator

```
pocketsphinx_continuous -lm /home/pi//TAR3483/3483.lm -dict  
/home/pi/Descargas/TAR3483/3483.dic -hmm /home/pi/pocketsphinx-  
0.8/model/hmm/en_US/hub4wsj_sc_8k | perl -ne 'system("/bin/bash  
/usr/local/bin/prueba_sphinx.sh \"$_\"")'
```

primero al definir un lenguaje hay que detallar un diccionario por eso en la sentencia se ejecuta la instrucción “-dict”, luego “-lm” para describir un modelo matemático de lenguaje, y esto se debe relacionar con el diccionario de Pocketsphinx, para esto se usa “-hmm”, finalmente a todo eso se añade un script, el script con la sintaxis siguiente.

```
| perl -ne 'system("/bin/bash /la dirección de tu script.sh \"$_\"")'
```

El modo en el que se programa el script es en instrucciones por terminal y el script tiene una estructura de casos, cuando un caso se introduce y ese es igualado a lo introducido por la terminal en forma de línea de texto proveniente del audio, al compararse genera una acción en caso de que algo pase.

Ese es la estructura básica del script

```
#!/bin/bash
echo se recibe $@
echo
se recibe $@ >> /tmp/prueba_reconocimiento.log
for n in $(echo $@);
do
    case $n in
HOLA) #HOLA
    sudo echo 23 >/sys/class/gpio/export
    sudo chmod 777 -R /sys/class/gpio/gpio23
    sudo echo out >/sys/class/gpio/gpio23/direction
    sudo echo 25 >/sys/class/gpio/export
    sudo chmod 777 -R /sys/class/gpio/gpio25
    sudo echo out >/sys/class/gpio/gpio25/direction
    sudo echo 7 >/sys/class/gpio/export
    sudo chmod 777 -R /sys/class/gpio/gpio7
    sudo echo out >/sys/class/gpio/gpio7/direction
```

”

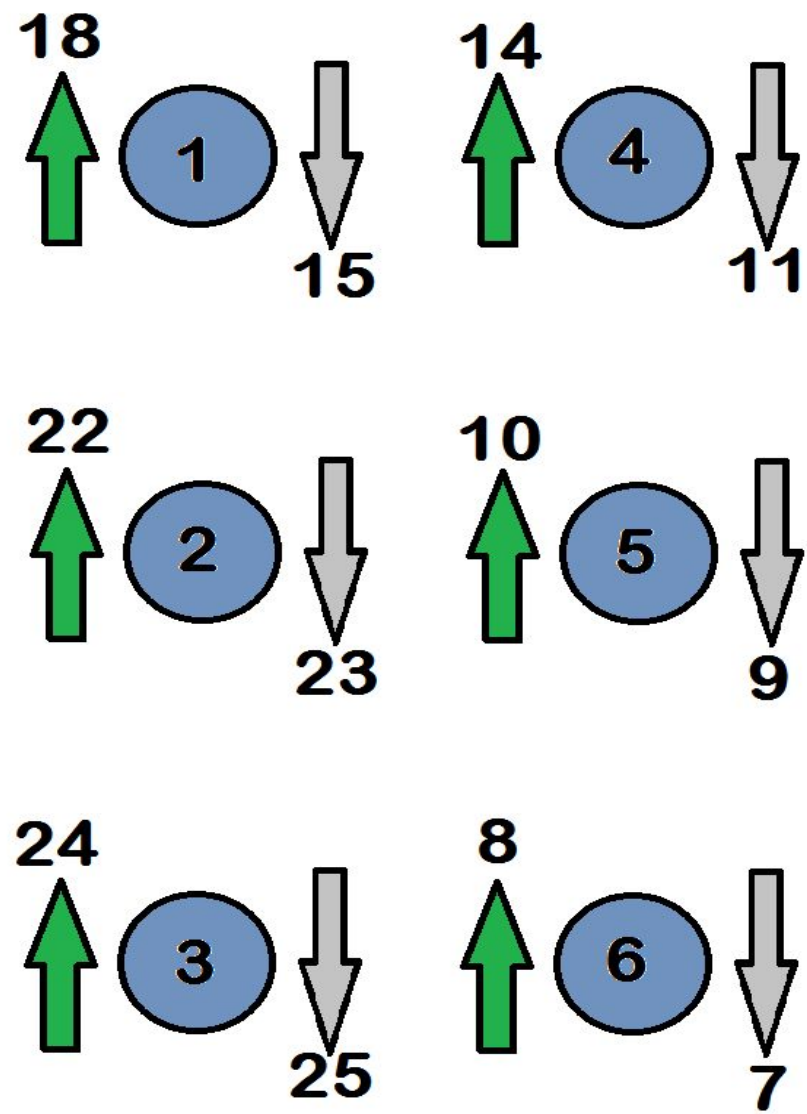
B) #B

```
mplayer / root/ audiobraillinux/ audio2.ogg  
echo 1 >/ sys/ class/ gpio/ gpio18/ value  
echo 1 >/ sys/ class/ gpio/ gpio22/ value  
mplayer / root/ audiobraillinux/ audio48.ogg  
echo 0 >/ sys/ class/ gpio/ gpio18/ value  
echo 0 >/ sys/ class/ gpio/ gpio22/ value
```

```
sleep 5s  
echo 1 >/ sys/ class/ gpio/ gpio7/ value  
echo 1 >/ sys/ class/ gpio/ gpio9/ value  
echo 1 >/ sys/ class/ gpio/ gpio11/ value  
echo 1 >/ sys/ class/ gpio/ gpio15/ value  
echo 1 >/ sys/ class/ gpio/ gpio23/ value  
echo 1 >/ sys/ class/ gpio/ gpio25/ value
```

```
sleep 1s  
echo 0 >/ sys/ class/ gpio/ gpio7/ value  
echo 0 >/ sys/ class/ gpio/ gpio9/ value  
echo 0 >/ sys/ class/ gpio/ gpio11/ value  
echo 0 >/ sys/ class/ gpio/ gpio15/ value  
echo 0 >/ sys/ class/ gpio/ gpio23/ value  
echo 0 >/ sys/ class/ gpio/ gpio25/ value
```

```
;;
```



E) #E

```
mplayer /root/audiobraillinux/audio5.ogg  
echo 1 >/sys/class/gpio/gpio18/value  
echo 1 >/sys/class/gpio/gpio10/value  
mplayer /root/audiobraillinux/audio51.ogg  
echo 0 >/sys/class/gpio/gpio18/value  
echo 0 >/sys/class/gpio/gpio10/value  
sleep 5s  
echo 1 >/sys/class/gpio/gpio7/value  
echo 1 >/sys/class/gpio/gpio9/value  
echo 1 >/sys/class/gpio/gpio11/value  
echo 1 >/sys/class/gpio/gpio15/value  
echo 1 >/sys/class/gpio/gpio23/value  
echo 1 >/sys/class/gpio/gpio25/value  
sleep 1s  
echo 0 >/sys/class/gpio/gpio7/value  
echo 0 >/sys/class/gpio/gpio9/value  
echo 0 >/sys/class/gpio/gpio11/value  
echo 0 >/sys/class/gpio/gpio15/value  
echo 0 >/sys/class/gpio/gpio23/value  
echo 0 >/sys/class/gpio/gpio25/value  
;;
```

Trabajo Futuro

Metas Logradas:

- 🔑 Enfocar el tema con un diseño que sea practico y de aprendizaje.
- 🔑 Realizar la elaboraci ó n en tiempo y forma.

Metas a Lograr:

- 🔑 Concluir el proyecto con calidad, buen diseño e innovador para su aplicaci ó n.



Conclusiones.

Hasta el momento se ha desarrollado el proyecto con las expectativas de diseño y aplicación con el que va enfocado el objetivo en un buen tiempo para seguir trabajando con este, para tener mayor calidad y eficacia con la que se pretende.

