

## ***Índice***

Página 2:  
Desarrollo de Idea Fuerza

Página 2:  
Desafíos

Página 2:  
Descripción

Página 3:  
Introducción.

Página 3:  
Diagramas en bloques

Página 4:  
Descripción detallada

Página 6  
Descripción del Hardware utilizado.

Página 13:  
Circuitos.

Página 14:  
Listado de Componentes

Página 15:  
Máquinas de Estado

Página 17:  
Problemas y beneficios encontrados a lo largo del desarrollo

Página 18:  
Conclusiones

Página 18:  
Bibliografía

# Desarrollo de Idea Fuerza

## Descripción:

Se planificó para el proyecto “Máquina de escribir braille” que utilizara como hardware base, la mecánica de una impresora antigua de matriz de puntos, la cual posee sensores, motores y un cabezal que fue modificado para ser un “punzón” que puede realizar marcas en el papel distinguibles al tacto.

El aparato a diseñar tendrá dos modos de funcionamiento, según si se lo conecta a la computadora o no. El usuario podrá mandar a imprimir archivos desde la misma (con un software a programar en C++), o bien, desconectándolo de la computadora, el sistema pasará a ser una máquina de escribir autónoma, que responderá a un teclado común similar al teclado de computadora. Nos referimos a este teclado como el teclado PS/2 .

La idea surgió de proyectos relacionados con la temática realizados por algunos miembros del grupo durante la secundaria. Se reutilizará una impresora Epson modificada para ahorrar diseño de hardware. Acerca de la temática del proyecto, consideramos que es momento de darle la atención necesaria ya que puede ser de mucha utilidad para las personas no videntes.

## Desafíos:

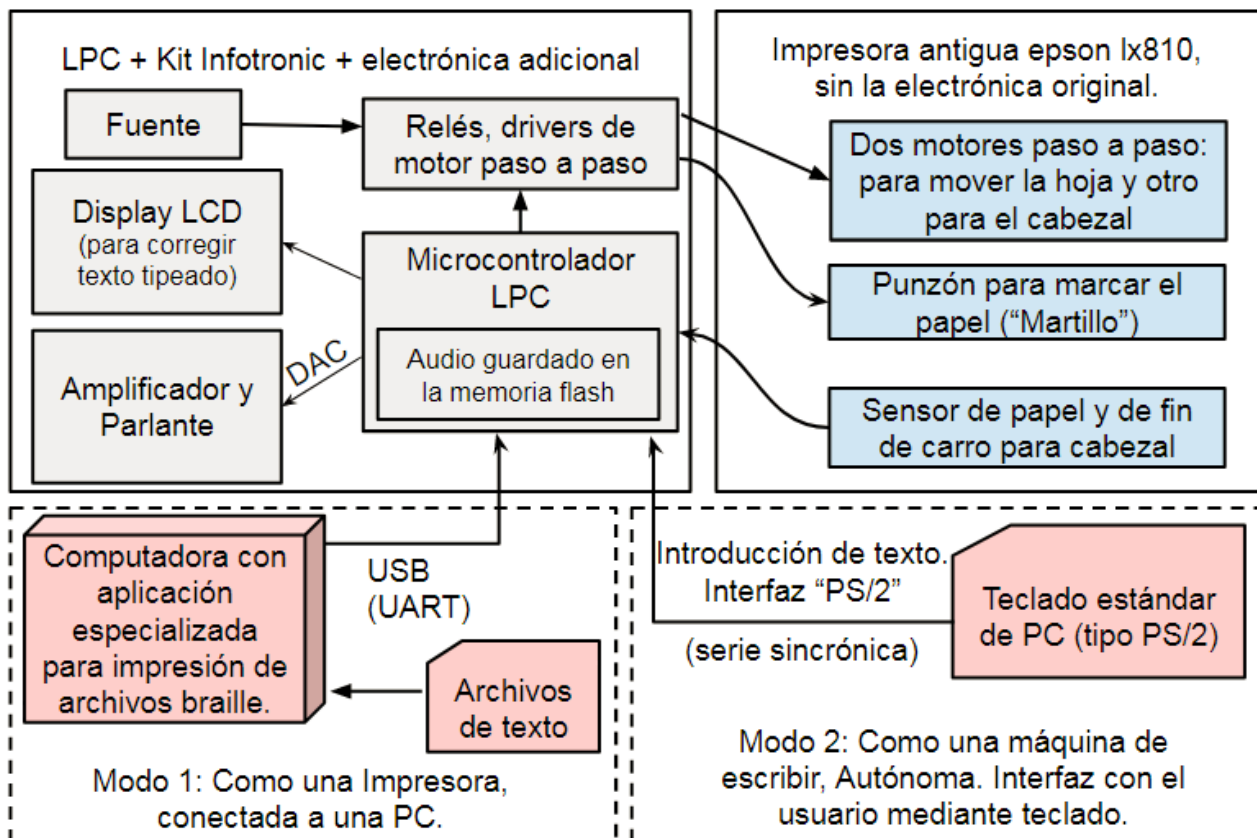
- Reproducción de archivos de audio digital.
- Comunicación, interpretación y compatibilidad con cualquier teclado ps/2.

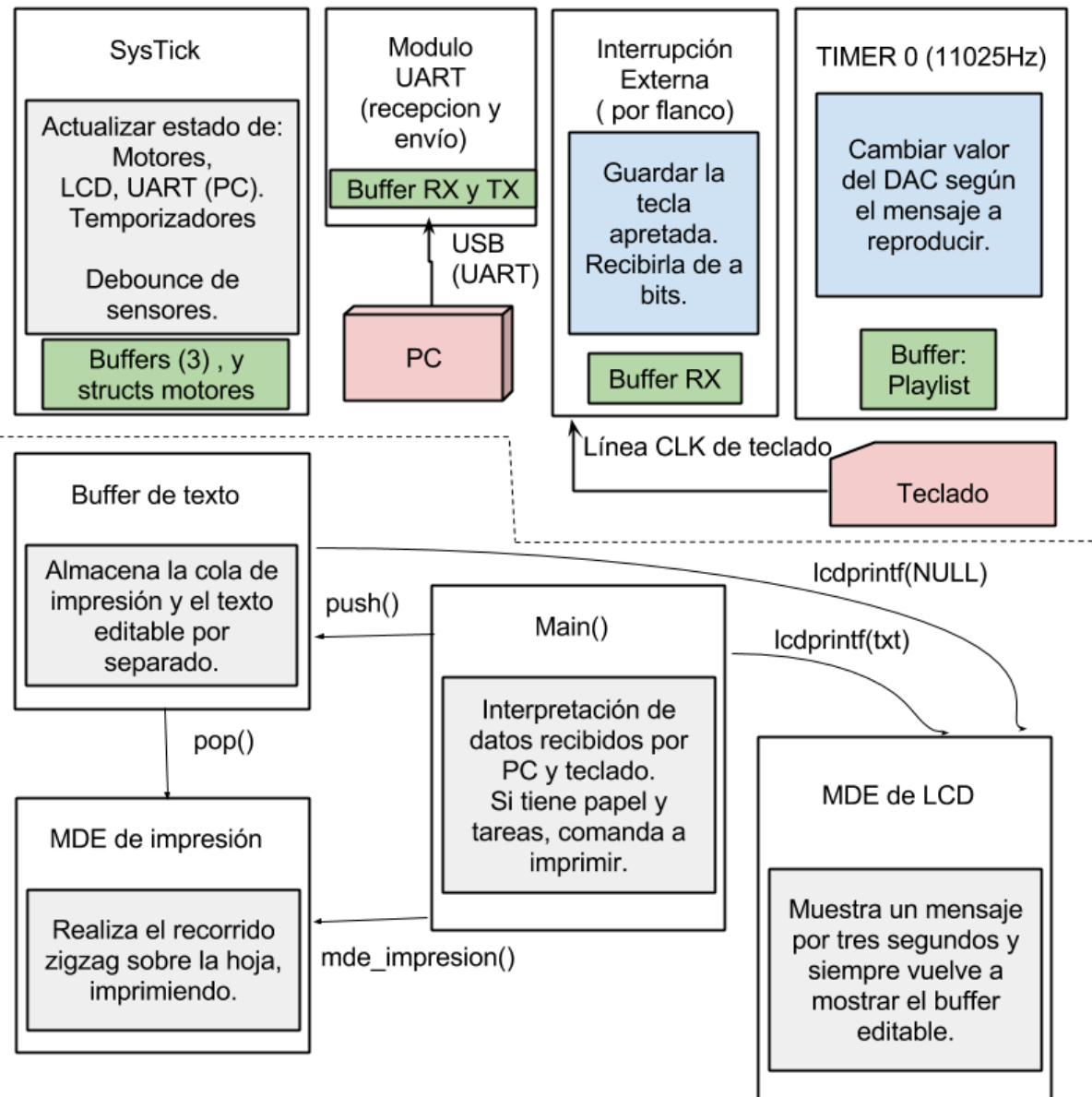
# Introducción

## Objetivos:

- Lograr con modificar una impresora de matriz de puntos, que funcione como una impresora Braille, utilizando componentes reciclados de impresoras en desuso. Que el aparato realice marcas en una hoja de papel, detectables al tacto.
- Permitir que el usuario podrá utilizar la impresora como maquina de escribir o, mediante una computadora, imprimir un documento de texto simple. Reproducir por sonido mensajes informativos, y mostrar en una pantalla el texto que se ingresa.
- Lograr ser compatible con teclados de computadora standard (los conexonado PS/2)

## Diagrama en bloques Hardware:



**Diagrama en bloques del software implementado para LPC:**

# Descripción detallada

**Descripción de los bloques:**

- Motores paso a paso: Se utilizan dos motores: uno mueve el cabezal horizontalmente y el otro desplaza el papel verticalmente.

- **Punzón para marcar papel:** Ubicado en el cabezal, es una bobina solenoidal que magnéticamente dispara una pieza metálica con punta que deja una marca en el papel distinguible al tacto. Necesita que la tensión VDD sea de 12V , 2A. Se lo conecta usando el relé de la placa infotronic #3.
- **Display LCD:** Cuando el aparato es usado como máquina de escribir, permite ver lo que se está escribiendo para corregir errores de tipeo antes de mandar a imprimir. Además muestra mensajes de estado.
- **Memoria FLASH:** Se refiere a la misma memoria que el LPC utiliza para el programa ejecutable. Cargando arrays constantes se almacenan archivos de audio digital. Se codificarán sin compresión (PCM) a frecuencia de 11025 Hz, 8 bits por muestra. El audio se graba y se codifica con herramientas especializadas de audio en una computadora. (hay 35 segundos de memoria)
- **Parlante y DAC:** Reproducen el audio digital pregrabado con el objetivo de informar mensajes de estado como, “listo para imprimir”, “no hay más papel”, y otros.
- **Teclado estándar de PC:** Cualquier teclado típico de computadora con esta conexión, es compatible con el proyecto y se lo conecta al aparato para introducir texto, e ingresar comandos.
- **Sensor de papel y sensor de fin de carro:** Sensores del tipo switch, similares a un botón, tienen solo dos estados. Uno detecta papel, y el otro calibra la posición absoluta del cabezal.

### **Especificaciones:**

Se debe utilizar una fuente de 12V como mínimo 2A.

Los motores paso a paso son EM-191 y EM-91 de la impresora Epson lx-810

Los drivers de motor paso a paso deben ser L298D , que tiene diodos de sobretensión incluidos.

# Descripción del Hardware utilizado.

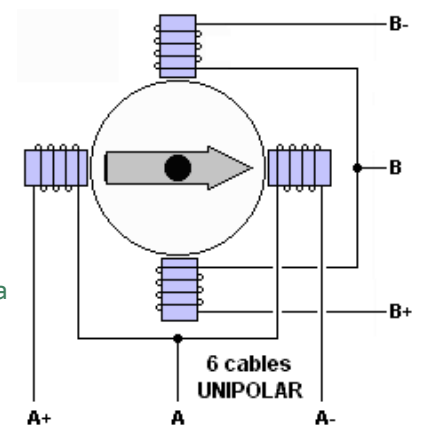
## 1: Mecánica de la impresora original:

Dos motores paso a paso unipolares de 4 bobinas, de 6 cables cada uno, y su cable paralelo en “Y” de tres conectores.

1. EM-91 de 40  $\Omega$  para mover el rodillo
2. EM-101 de 12  $\Omega$  para mover el carro

DR\_motores.h:

```
typedef struct
{
    motor_drive_t tipo_drive;
    // typedef enum: FULL_STEP, FULL_STEP_POWER
    uint8_t puertos[4][2];
    // Fila (A-B-C-D) - Columna (0:Puerto-1:Bit)
    motor_estado_t estado_corriente; // typedef enum:
    ENERGIZANDO, DEENERGIZANDO, DESUSO
    int8_t tiempo_transcurrido; // en ticks
    uint8_t tiempo_on; // tiempo que hay que prender la bobina
    uint8_t tiempo_off; // tiempo que hay que apagar la bobina
    int8_t estado_bob; // estado de la secuencia de bobinas
    int16_t pasos_pendientes;
    int16_t posicion;
} motorpap_t;
```



## 2: Bobina solenoidal

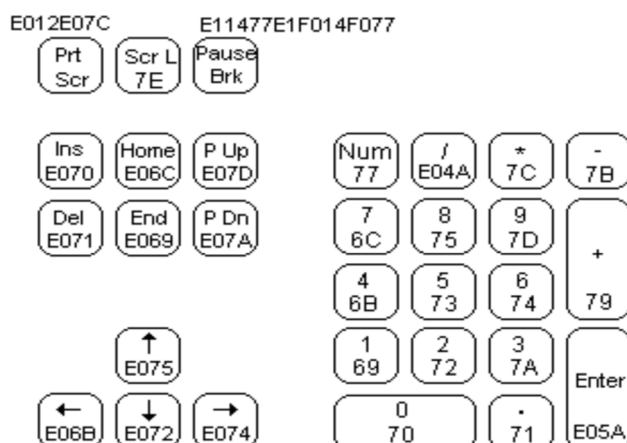
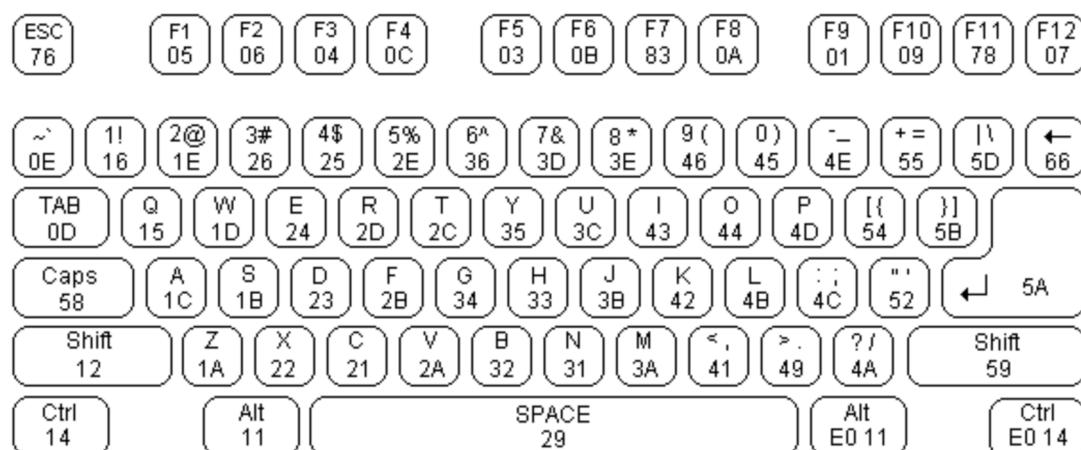


Se utiliza en el carro para golpear y marcar una hoja de papel, y así lograr la escritura de una letra braille. Al energizar la bobina con suficiente corriente, la pieza metálica del centro se desplaza rápidamente. Al desenergizarla, la pieza es empujada en dirección contraria por un resorte, volviendo a la posición original. Utiliza entre 1,5 y 2A.

### 3: Teclado standard PS/2

Un **teclado PS/2** para PC es un dispositivo que posee un microcontrolador cuyas funciones son rastrear teclas, ignorar sus rebotes y transmitir, mediante un protocolo estandarizado, los resultados de nuestras manipulaciones sobre las teclas.

El teclado PS/2 tiene un mapa de teclas, donde cada tecla tiene un código correspondiente, de uno o dos bytes, que luego será el mismo código que transmitirá para indicar que tecla se ha pulsado. A estos códigos se les llama **códigos de rastreo de teclado**. La siguiente imagen muestra los códigos de rastreo de cada una de las teclas:



```

#define CANTIDAD_CODIGOS 51
// Ñ = 0xEE
uint8_t const codigo_rastreo[CANTIDAD_CODIGOS][2] = {
    0x45, '0', 0x16, '1', 0x1e, '2', 0x26, '3', 0x25, '4', 0x2e, '5', 0x36, '6', 0x3d, '7',
    0x3e, '8', 0x46, '9',
    0x1c, 'a', 0x32, 'b', 0x21, 'c', 0x23, 'd', 0x24, 'e', 0x2b, 'f', 0x34, 'g', 0x33, 'h',
    0x43, 'i', 0x3b, 'j',
    0x42, 'k', 0x4b, 'l', 0x3a, 'm', 0x31, 'n', 0x4c, 0xEE, 0x44, 'o', 0x4d, 'p', 0x15, 'q',
    0x2d, 'r', 0x1b, 's',
    0x2c, 't', 0x3c, 'u', 0x2a, 'v', 0x1d, 'w', 0x22, 'x', 0x35, 'y', 0x1a, 'z', BLOQ_MAYUS,
    '^', TAB, '#', ENTER, '\n',
    0x41, ',', 0x49, '.', 0x4a, '_', 0x4e, '?', 0x5d, ';', 0x55, '!', 0x54, '(', 0x52, ')',
    0x0e, '"', 0x5b, '*', 0x29, ' ',
};

```

El teclado PS/2 envía el **código de rastreo** asociado a una tecla al ser ésta pulsada. Lo hace tantas veces como sea necesario, si se la mantiene pulsada con una cadencia, tal como lo indique su tiempo de repetición, que es programable. Finalmente envía el mismo código de rastreo con el prefijo del byte **F0h** al ser soltada la tecla, también conocido como **Break Code**.

**Por ejemplo:** para conseguir la letra **G Mayúscula** se debe mantener presionada la tecla Shift, pulsar la tecla G, soltar la tecla G y soltar la tecla Shift. Todos estos eventos se transmiten como esta secuencia de bytes:

### Keyboard 6 Pin (PS/2)



(6p Mini-DIN female (PS/2) at PC)



(6p Mini-DIN male at cable / keyboard)

1. Pulsar Shift -> 12h,
2. pulsar "G" -> 34h,
3. soltar "G" -> F0h 34h
4. y soltar Shift -> F0h 12h

Mensaje final: 1h, 34h, F0h, 34h, F0h, 12h.

### Conexión eléctrica:

El teclado PS/2 se conecta mediante cuatro cables. Dos de ellos son de alimentación (5 V) y los otros dos para las señales **Data** y **Clock**. El pinout de los conectores Mini-DIN PS/2, tanto los hembra de la PC, como los machos del teclado, es tal como se muestra en la imagen de la derecha.

Pin	Name	Dir	Description
1	DATA	↔	Key Data
2	n/c		Not connected
3	GND	—	Gnd
4	VCC	→	Power , +5 VDC
5	CLK	→	Clock
6	n/c		Not connected



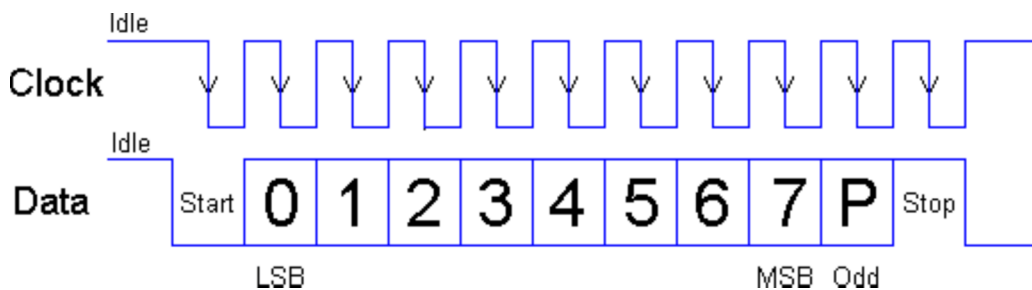
En el Teclado PS/2 las señales Data y Clock con de "**colector abierto**", por lo que para establecer una comunicación se deben colocar resistencias de pull-up en DATA y CLK. Esto se consigue con una configuración del µC.

El teclado PS/2 se comunica mediante un Protocolo Serie Síncrono. Utiliza por lo tanto, una señal de Clock que indica cuando esta disponible el bit a leer en el pin Data.

En reposo la señal de Clock está a nivel alto, a causa del pull-up. Cuando esta señal pasa a estar en 0, significa que el teclado enviara un dato en serie por el pin Data.

El mensaje completo se compone de 11 bits. Siendo el primero un bit de Start, a continuación los 8 bits del Dato a transmitir enviándose primero el LSB (ó bit menos significativo), el décimo indica la paridad (se usa la impar - odd) y el último es un bit de Stop.

Este gráfico representa las señales CLK y DATA desde el momento previo de envío de un mensaje del teclado al PC, hasta el fin del mensaje:



```
void EINT3_IRQHandler( void )
{
    static uint8_t bit = 0, paridad = 0, dato = 0;
    static bool_t mensaje_erroneo = FALSE;

    if( bit == 0 || TimerGet(10) == 0 )    // Bit de start
    {
        if( TimerGet(10) == 0 ) bit = 0;
        mensaje_erroneo = FALSE;
        paridad = 0;
        dato = 0;
        if( PS2DATA == ALTO ) mensaje_erroneo = TRUE; // Bit de start debe ser 0
        TimerStart( 10 , 4 , DEC, Teclado_PS2_Timeout );    // Prender timer para
resetear por timeout=20ms
    }

    if( bit >= 1 && bit <= 8 )    // Bits de datos
    {
        dato |= ( PS2DATA << (bit-1) );
        if( PS2DATA == 1 )
            paridad++;    // Cantidad de bits en uno
    }

    if( bit == 9 )
    {
        // Paridad impar, ignorar
    }
}
```

...

#### 4: Display alfanumérico genérico de cristal líquido.



Un display alfanumérico de matriz de puntos (dot-matrix) es un dispositivo de interfaz humana formado por una pantalla de cristal líquido o LCD (Liquid Crystal Display) sobre la que se pueden mostrar mensajes formados por distintos caracteres: letras, números, símbolos, etc. (figura 1). Se encuentran en distintos formatos, por ejemplo,  $2 \times 8$ ,  $2 \times 16$ ,  $4 \times 20$ , etc. El primer dígito indica el número de filas del display y el segundo el número de columnas, es decir,  $2 \times 16$  significa que tiene 2 filas y 16 columnas.

Estos dispositivos vienen gobernados por un microcontrolador, que normalmente va incorporado sobre la misma placa de circuito impreso que soporta el LCD (figura 2). En el mercado es muy habitual encontrarse con el controlador HD44780 de Hitachi, que es el que se describe a continuación aquí. El controlador se encarga de gestionar el display líquido: polarizar los puntos de la pantalla, generar los caracteres, desplazar la pantalla, mostrar el cursor, etc. El usuario se despreocupa de todos estos problemas y simplemente necesita conocer una serie de comandos o instrucciones de alto nivel (limpia display, posiciona cursor, etc.) que le permitirán mostrar mensajes o animaciones sobre la pantalla de forma sencilla. Para comunicarse con el controlador del display se dispone de una interfaz paralelo al exterior, de fácil conexión a otros microcontroladores o microprocesadores.

Descripción de los pines de un LCD alfanumérico

Nº PIN	SÍMBOLO	DESCRIPCIÓN
1	$V_{SS}$	Alimentación negativa (masa)
2	$V_{DD}$	Alimentación positiva (+5Vdc)
3	$V_O$	Ajuste del contraste
4	RS	Selección de Registro (Register Select) RS = 1 $\Rightarrow$ Registro de Datos RS = 0 $\Rightarrow$ Registro de Instrucciones
5	$R/\overline{W}$	Lectura / Escritura (Read/Write) $R/\overline{W}$ = 1 $\Rightarrow$ Lectura $R/\overline{W}$ = 0 $\Rightarrow$ Escritura
6	E	Habilitación del display (Enable) E = 1 $\Rightarrow$ Habilitado E = 0 $\Rightarrow$ Deshabilitado
7-14	DB[0..7]	Pines del 0 al 7 del bus de datos bidireccional

– Señales de alimentación:

Son tres señales correspondientes a los pines 1, 2 y 3. El pin 1 corresponde a la masa, el 2 a la alimentación positiva (normalmente +5Vdc) y el 3 al ajuste del contraste. Habitualmente, al dar alimentación y sin haber mandado todavía ningún comando, el display muestra en su pantalla la primera fila con todos sus caracteres en negro. Si no fuera así, se debería proceder al ajuste del contraste. Para ello, se debe instalar un potenciómetro de unos 10K $\Omega$  tal y como se observa en la figura 3. Cuanto más cercano a masa esté el voltaje en el pin 3 ( $V_O$ ) mayor será el contraste.

– Señales de control:

Los pines 4, 5 y 6 forman las señales de control del LCD. El pin 4 (RS) sirve para seleccionar el registro de datos (DR) o el de instrucciones (IR), poniendo RS=1 o RS=0, respectivamente. El pin 5 permite leer ( $R/\overline{W}$  = 1) o escribir ( $R/\overline{W}$  = 0) en el módulo LCD tanto datos como instrucciones. Y por último, el pin 6 (E) permite habilitar, con E=1, o deshabilitar el display (E=0). Sólo cuando está habilitado nos podemos comunicar con él.

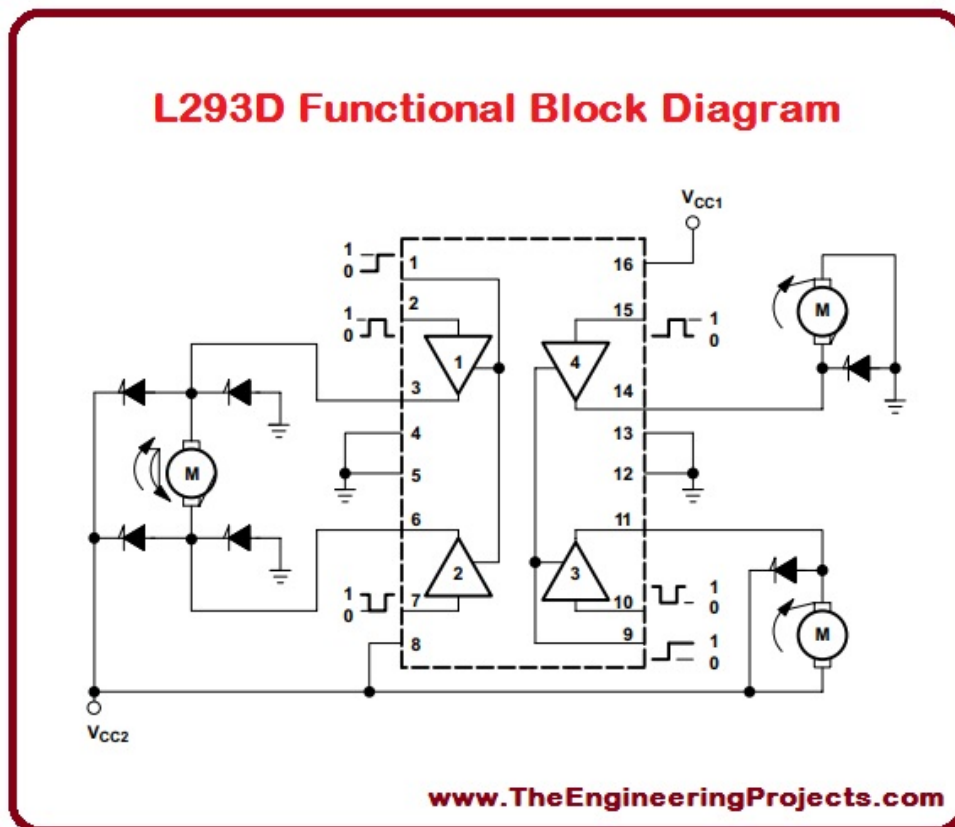
– Señales de datos:

Por último, los pines del 7 al 14 forman un bus de datos bidireccional de 8 bits (DB7 – DB0) por donde se pueden escribir datos e instrucciones y se puede leer el estado del display (si está o no ocupado, la posición actual del cursor, etc.). El LCD también

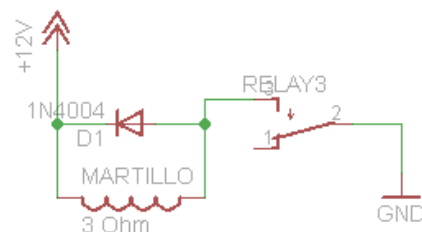
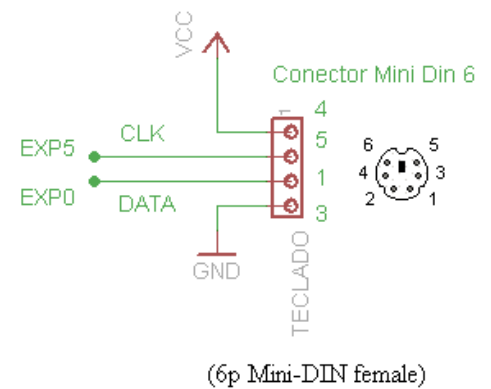
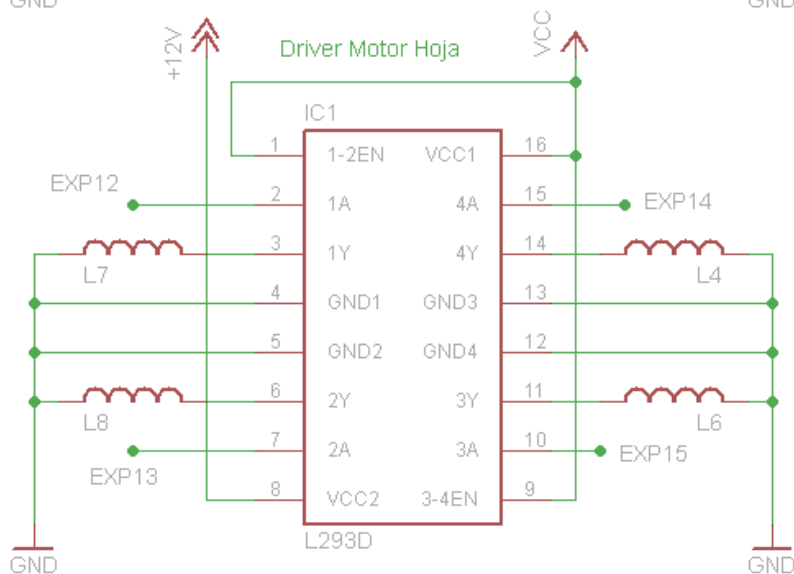
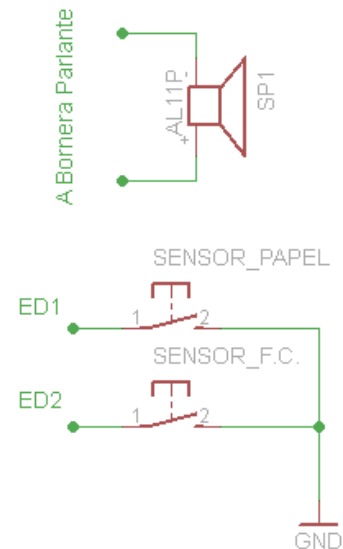
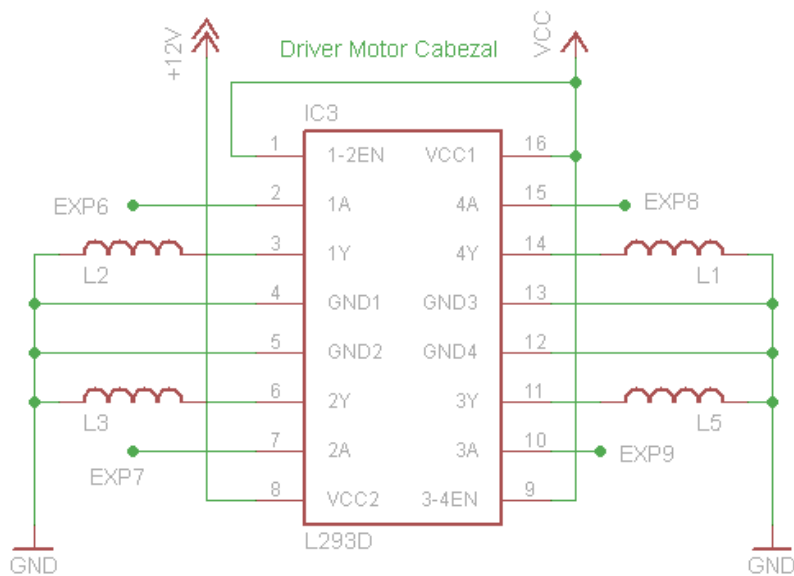
puede ser gobernado con un bus de datos de 4 bits (utilizando el nibble más significativo).

#### 5: Dos circuitos integrados L293D

Este circuito se ocupa de entregar 12V a las bobinas de un motor paso a paso a partir de una señal de microcontrolador (de mínima corriente) de 5V o menos. Se dedica un pin por cada bobina de cada motor.



## Circuitos



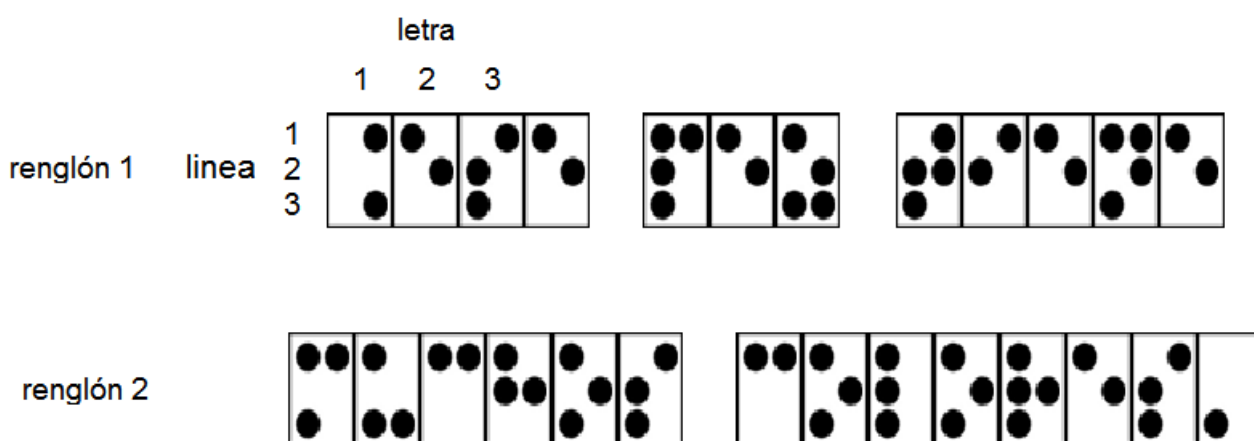
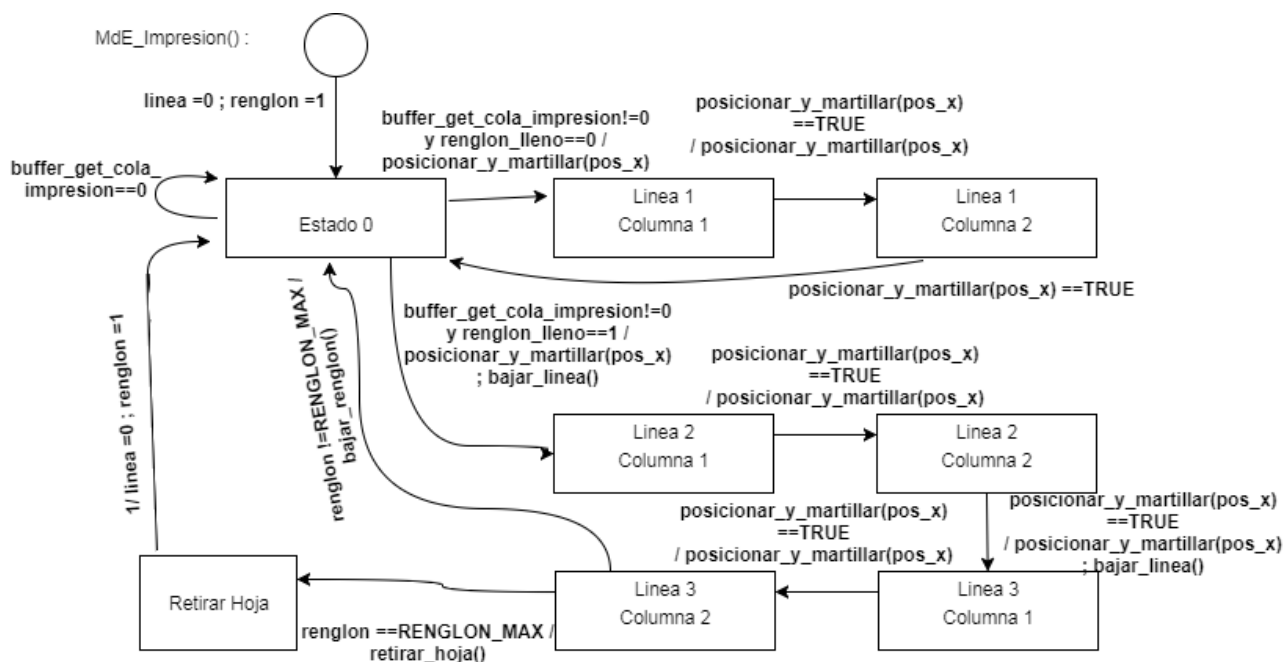
[www.ti.com/lit/ds/symlink/l293.pdf](http://www.ti.com/lit/ds/symlink/l293.pdf)

**Listado de componentes**

Cantidad	Material
2	Placas prototipo 10x15
2	L293D
1	Impresora matriz de puntos
1	Fuente
1	Parlante
1	Diodo 1n4004
1	Bobina solenoidal
1	LCD
1	Teclado PS2
10	Fichas y conectores varios
1	Capacitor de filtro

# Máquinas de Estado

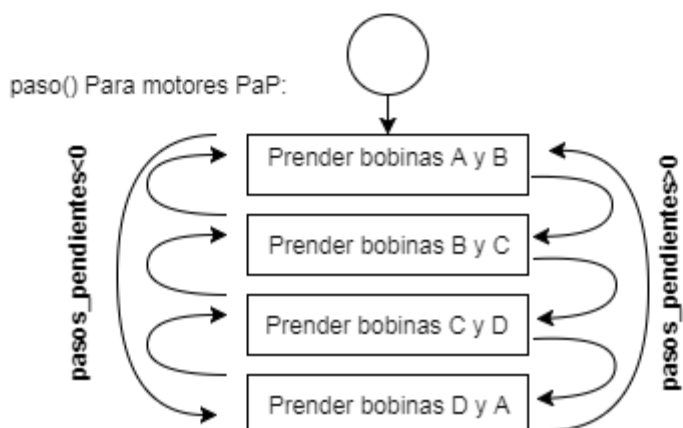
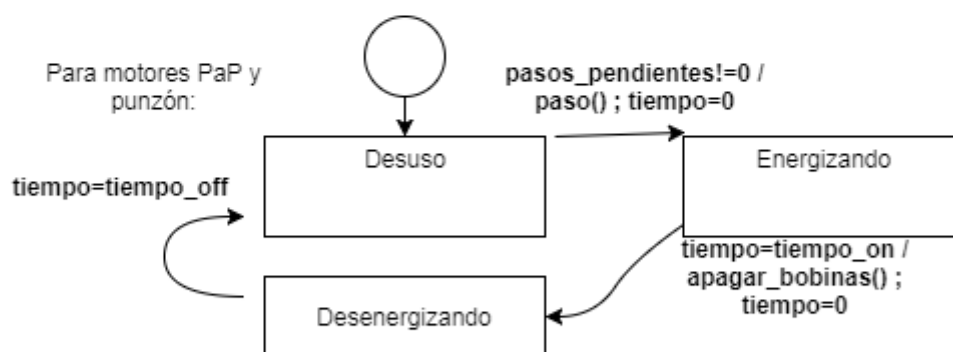
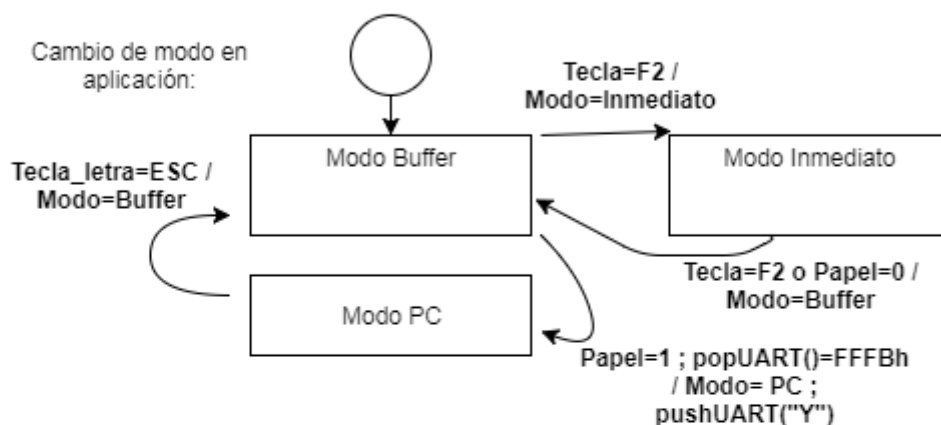
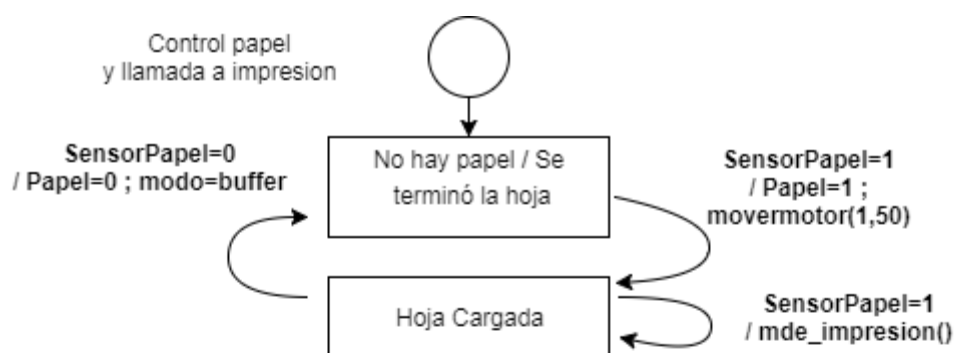
MdE\_Impresion()



Puntos:

1		4
2		5
3		6

uint8\_t letrabraille = 0b00111111  
Puntos: xx654321





# Problemas encontrados a lo largo del desarrollo

## Puesta en marcha

Se encontró que la bobina del punzón necesitaba una fuente bastante potente.

El conjunto mecánico del rodillo no estaba pensado para cambiar de dirección directamente.

El conector PS/2 (Minidin 6) es difícil de conseguir en negocios, además solo se consigue con pines rectos, no pudiendo ser soldados con cables.

## Algoritmos

Se encontró que no se puede usar más del 50% de FLASH de programa.

Se encontró que usando las librerías de QT de puerto, archivos y demás, es difícil saber con exactitud si a bajo nivel los datos se van a enviar en un formato esperado.

## Beneficios encontrados a lo largo del desarrollo del TPO

Se logró aprender una buena metodología para la abstracción y resolución de problemas que involucren un microcontrolador y electrónica básica.

Se conoció la plataforma de desarrollo que tiene buenos recursos (velocidad, memoria), que evitan el uso de técnicas de programación que busquen no usar memoria.

# Conclusiones

Los conocimientos adquiridos sobre programación de microcontroladores , junto con la comunicación con PC e interfaces de usuario, alcanzan para hacer un rango amplio de artefactos que resuelven problemas de la vida diaria.

La impresora braille tal como fue desarrollada es un aparato que ya adquirió muchas posibilidades de poder convertirse en una ayuda para el que lo necesite.

# Bibliografía

[http://picmania.garcia-cuervo.net/proyectos\\_teclado\\_ps2.htm](http://picmania.garcia-cuervo.net/proyectos_teclado_ps2.htm)

[www.ti.com/lit/ds/symlink/l293.pdf](http://www.ti.com/lit/ds/symlink/l293.pdf)