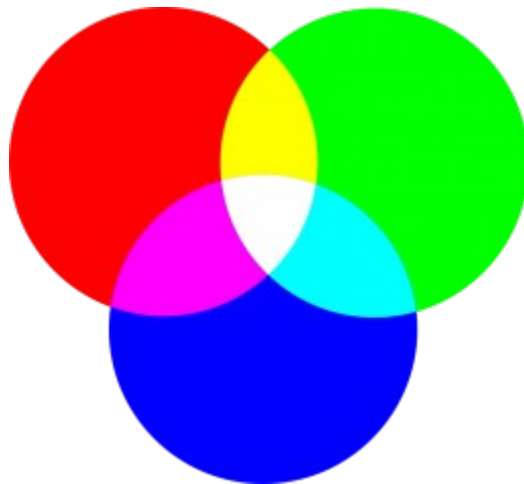


**UTN.BA**

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

# Display POV: Persistence of vision



## INDICE

1. Descripción general.....	3
1.1. Introducción teórica.....	3
1.2. Funcionamiento.....	3
1.3. Interface con usuario:.....	3



# UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

1.4. Operación:	3
2. Hardware	4
2.1. Diagrama en bloques	4
2.2. Descripción de hardware	5
2.2.1. Maqueta	5
2.2.2. Módulos adicionales	5
2.2.3. PCB: componentes principales	6
2.3. Circuito impreso	7
3. Firmware	8
3.1. Firmware Microcontrolador	8
4. Problemas durante el desarrollo	10
4.1. Hardware	10
4.2. Software PC	11
4.3. Firmware Microcontrolador	11
5. Software Utilizado	11
6. Referencias	12

# 1. Descripción general

## 1.1. Introducción teórica

El display persistente of vision está basado en el conocido fenómeno de la persistencia de la visión descubierto por Peter Mark Roget el cual demostró que una imagen permanecía en la retina una décima de segundo después de haber desaparecido. Es debido a este efecto que en la vida cotidiana vemos la ilusión de movimiento debido a que el cerebro entrelaza las imágenes que se superponen en la retina.

## 1.2. Funcionamiento

El fin de este proyecto es mostrar una imagen creada en una aplicación de computadora y transmitirla al display PoV mediante una conexión bluetooth. Para esto el display aprovecha el efecto de persistencia de la visión y mediante el montaje del pcb sobre el eje de un motor, gira a una frecuencia mayor a la frecuencia de retención de imágenes del ojo humano. Dicho pcb será detallado más adelante.

## 1.3. Interface con usuario:

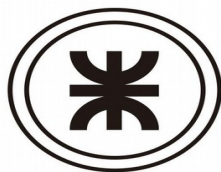
La aplicación "SPINO BRUSH" se encuentra disponible para las plataformas de Mac, Windows y Linux creada por Guillaume Chereau. Esta aplicación se encargará de crear y transmitir mediante una conexión de bluetooth (Requisito básico para utilizar esta aplicación) una imagen en coordenadas polares.

## 1.4. Operación:

Para lograr la conexión de la aplicación primero se deberá emparejar el dispositivo bluetooth con la computadora.

Luego se deberá establecer la conexión dentro de la aplicación. Una vez establecida la conexión el usuario contará con una serie de amplias herramientas para el diseño de su imagen así también contará con la posibilidad de cargar imágenes ya existentes.

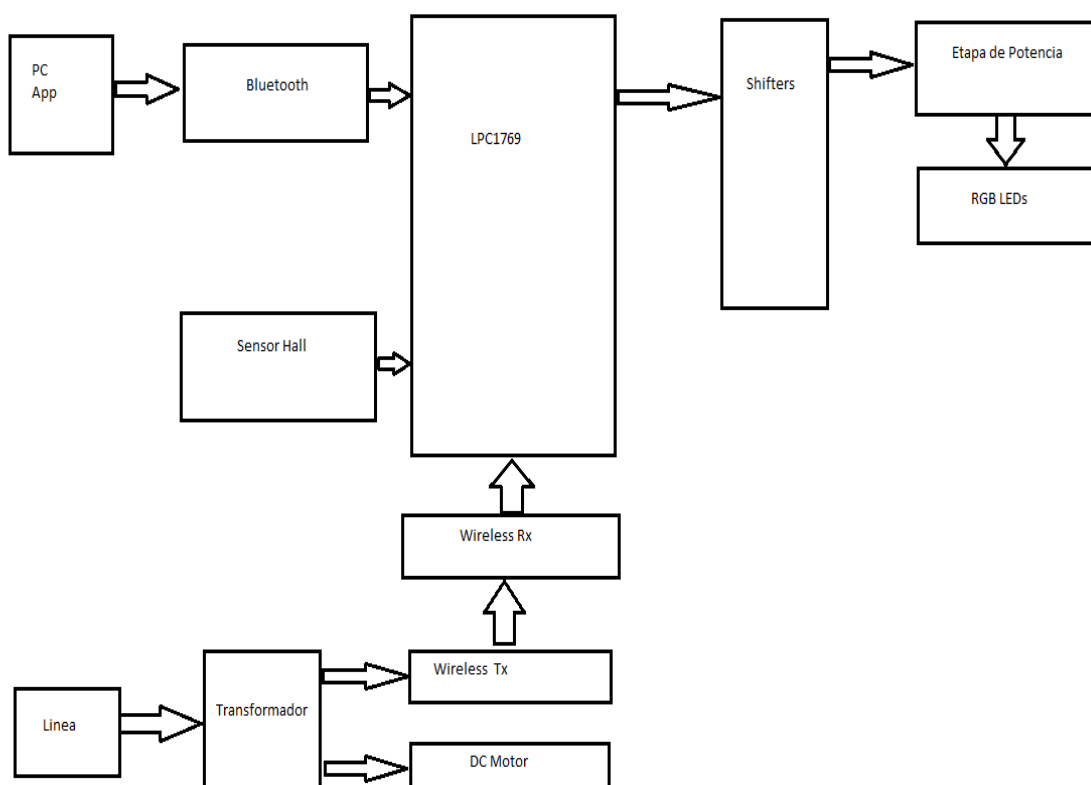
# 2. Hardware



# UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

## 2.1. Diagrama en bloques



## 2.2. Descripción de hardware

### 2.2.1. Maqueta

(Descargar [Maqueta3D](#))

- **Base:**

El diseño consiste en un gran cubo, compuesto de pino y acrílico, en el cual irán montadas todas las piezas del proyecto. En su interior se aloja el motor y el cableado, en el acrílico frontal se montan los interruptores



# UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

para el manejo por el usuario y por último sobre el acrílico superior se montan las piezas plásticas junto con la placa.

- **Piezas 3D:**

Se diseñaron 2 piezas 3D. La primera con el fin de sujetar y alinear la bobina encargada de la transmisión de energía hacia el PCB y la segunda encargada de la sujeción y alineación de la bobina de recepción de energía así como el agarre mecánico entre el motor y el PCB. (Ver [PLANO 1](#), [PLANO 2](#)).

## 2.2.2 Módulos adicionales

- **Módulo de carga inalámbrica:**

Se escogió un [módulo de carga inalámbrica](#) para transmitir la potencia necesaria para alimentar a la placa mediante el fenómeno de inducción magnética. Con esto evitamos problemas típicos de cableado, falsos contactos, ruidos y desgaste mecánico en la unión eléctrica de un sistema estático con uno en movimiento.

- **Motor de corriente continua:**

Se escogió el motor [SP6-1500](#) debido a que poseía las suficientes revoluciones por minuto para generar el efecto de persistencia de la visión.

- **Módulo de control de velocidad:**

Se escogió este [módulo](#) para poder hacer un arranque lento del motor, además de poder jugar con diferentes velocidades y ver cómo influye en el efecto del proyecto.

- **Sensor hall:**

Se escogió el modelo [A1301](#) para sensar la velocidad del motor de continua a través de un imán.

- **Imán de neodimio:**

Se escogió un [imán de neodimio](#) con el fin de utilizarlo como referencia para el sensor hall.



# UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

## 2.2.3 PCB: componentes principales

- **Microcontrolador:**

Se escogió el [LPC1769](#). Basado en un Cortex-M3 que contiene todo el firmware. Es el núcleo principal de la placa y se destinó a realizar todas sus tareas y funcionalidades.

- **Registro de desplazamiento (Shifter):**

Se escogió el integrado [MC74HC595](#). Se colocaron 12 registros de desplazamiento para poder controlar los 32 leds rgb, dispuestos de una forma matricial determinada acorde al manejo de la información de los mismos por parte del microcontrolador.

- **Módulo Bluetooth:**

Se escogió el modulo [HC-05](#). Se colocó 1 modulo bluetooth para lograr una conexión inalámbrica entre la aplicación de PC y el microcontrolador.

- **Tarjeta Micro SD:**

Se escogió una tarjeta [micro SD](#) con el fin de poder cargar una imagen estática sin la necesidad de la aplicación de PC.

- **Amplificador Operacional:**

Se escogió el operacional [MC33202](#). Se colocó este operacional rail to rail para que opere como smith trigger comparando la señal analógica de salida del sensor hall con una tensión estable de referencia.

- **Leds RGB:**

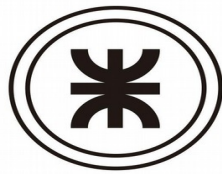
Se escogieron 32 [leds RGB 1206](#) para obtener una buena resolución en la imagen.

- **Regulador switching:**

Se escogió el integrado [A5972D](#) para la regulación de tensión de 5v de entrada a 3,3v de salida.

## 2.3. Circuito impreso

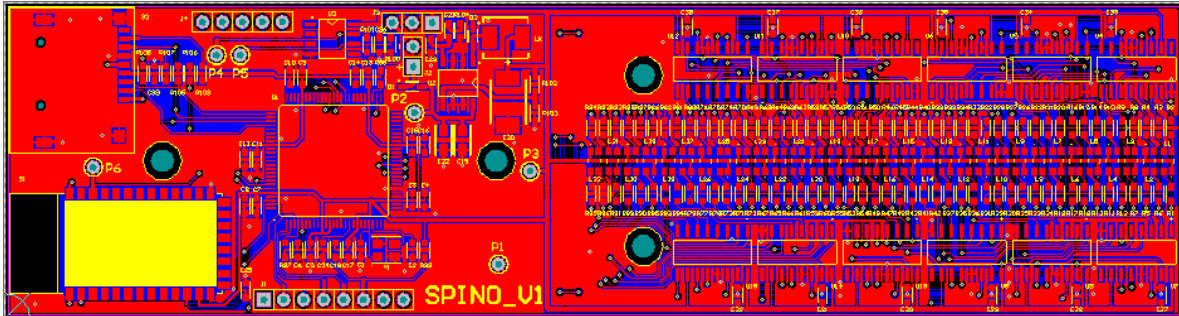
(Ver [Esquemático](#))



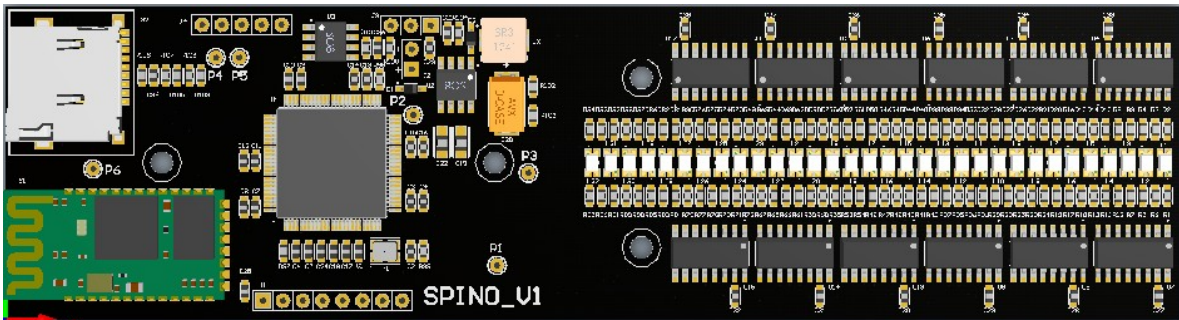
# UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

- **Diseño total**



- **Modelo 3D**  
(Descargar [PCB3D](#))



## 3. Firmware

### 3.1. Firmware Microcontrolador

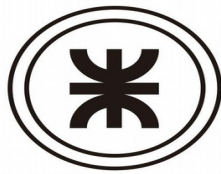
- **Funcionamiento general:**

El firmware fue diseñado en base a un sistema operativo en tiempo real, y si bien su propósito final es mostrar una imagen, podemos destacar y subdividir 4 áreas de su funcionamiento.

-Secuencia de inicio:

Al iniciar el programa, se concentrará específicamente en calibrar su frecuencia de refresco de imagen para poder trabajar adecuadamente (independientemente de poseer una imagen a imprimir). Esto se logra gracias a la señal del sensor de efecto hall,





**UTN.BA**

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

de la cual se toman sucesivas muestras que, luego de un simple filtro FIR, indicarán el periodo de trabajo del motor, es decir, el tiempo que tarda en dar una vuelta.

A partir de este momento, el sistema se encuentra listo y se habilita para realizar el resto de sus tareas habituales.

-Obtención de la imagen en formato fuente:

El sistema ahora estará a la espera de una conexión por bluetooth de la cual adquiera los datos necesarios para almacenar una imagen. Mientras no tenga éxito en la conexión, tomará su imagen fuente de una tarjeta micro SD por SPI. De tener éxito, el sistema se encargará de establecer comunicación, solicitar y validar la información requerida, esta última será enviada de forma comprimida para reducir el tamaño del paquete de información.

-Procesado y almacenamiento de la imagen en formato final:

Una vez recibido el dato, será momento de descomprimirlo, interpretarlo, traducirlo y almacenarlo para posteriormente poder imprimirlo. Cabe destacar que la información enviada por la aplicación de PC y la información requerida por el firmware son de tipos y tamaños completamente diferentes, por los que necesitan una transformación especial.

Por lo tanto, la información primero sufre descompresión por el algoritmo lz4, luego transformación de tipo y por último se almacena en memoria estática esperando ser solicitada.

-Impresión de la imagen en formato final:

Por último, conociendo ya a qué frecuencia deberá imprimir los datos, el sistema empezará a enviar serialmente la información y la señal de reloj a los registros de desplazamiento, y repetirá la secuencia recorriendo fila por fila de la imagen total.

De completar la secuencia de la imagen y no poseer una nueva que imprimir, el sistema, al tener una copia de la imagen almacenada, imprimirá indefinidamente dicha imagen.

Para ver información detallada de la programación de firmware, puede ver la documentación en [Doxygen](#).

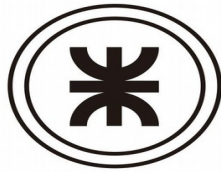
## 4. Problemas durante el desarrollo

### 4.1 Hardware

- **Manejo de leds:**

Para el manejo de los leds RGB tuvimos que utilizar una multiplexación debido a la gran cantidad de pines que representaban los 32 leds RGB





# UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

- **Ruteo:**

Debido a la alta densidad de trazas en el PCB por la cantidad de leds y la elección de utilizar solo 2 layers para reducir costos, encontramos dificultades a la hora de rutear por lo que utilizamos un layer para las trazas verticales y otro para las horizontales a modo de organización para facilitar el ruteo.

- **Motor:**

Nos encontramos con dificultades a la hora del montaje de las piezas 3d sobre el motor por el tamaño del eje del motor. Una vez solucionado este problema y mediante algunos testeos llegamos a la conclusión de que al estar alimentando al PCB por inducción el motor también se veía inmerso en este campo magnético por lo que se terminaba induciendo una corriente la cual calentaba el motor. Este efecto además de ser perjudicial para el motor también generaba una pérdida de potencia por lo que se escogió cambiar el motor por uno con caja reductora de plástico, de esta manera solucionamos el problema de la inducción del motor y además conseguimos un eje más largo lo cual aumento la estabilidad de las piezas finales.

- **Piezas 3D:**

Se presentaron inconvenientes a la hora del diseño de las piezas 3D ya que ningún integrante poseía el conocimiento para éste, y tampoco una impresora 3D. Luego de muchos intentos habiendo solucionado problemas de contrapeso y gracias a la disponibilidad de la impresora 3D del departamento de electrónica logramos diseñar estas piezas.

## 4.2 Software PC

- **Aplicación ya existente:**

Se optó por utilizar una aplicación ya existente para reducir el tiempo de desarrollo del proyecto, debido a esto tuvimos problemas a la hora de hacer ingeniería inversa para identificar el formato en el cual se enviaba la imagen.

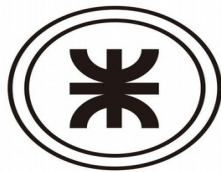
## 4.3 Firmware Microcontrolador

- **Tiempos de procesamiento:**

Se encontraron dificultades para hacer que el tiempo desde que se envía la imagen en la aplicación hasta que se ve realmente en el display fuera de percepción instantánea. Los algoritmos de descompresión y procesamiento, junto con los tiempos de interrupción del sistema operativo lograron superar al microcontrolador tanto en frecuencia de operación como en memoria requerida.

- **Memoria disponible:**

El gran almacenamiento de datos para contener copias de la imagen y la



# UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

memoria que requirió individualmente cada tarea del sistema operativo, saturaron en gran parte a la memoria del microcontrolador. Se solucionó reduciendo la cantidad de tareas del sistema operativo y agrupandolas de manera estratégica. No se optó por sobredimensionar el microcontrolador porque no se contaba con dicha opción.

## 5. Software Utilizado

- Programación de microcontrolador: MCUexpress
- Diseño 3D: SolidWorks
- Planos: SolidWorks
- Diseño PCB: Altium

## 6. Referencias

- [Repositorio General](#)
- [Hojas de datos](#)
- [Informe PCB](#)
- [Doxygen](#)
- [Manual Técnico](#)
- [Manual de Usuario](#)
- [Aplicación PC \(WEB del Creador\)](#)
- [BOM](#)
- [Esquemáticos](#)
- [Maqueta3D](#)