



# Informe eléctrico y electrónico

PROYECTO DE MÓDULO 2º, 2º CUATRIMESTRE. GRUPO 10

Carla Marzo Beltrol

Cristina Meléndez Sancho

Sandra Ordovás Gracia

Carolina Jiménez de la Cruz

## ÍNDICE

1. INTRODUCCIÓN
  - 1.1. Descripción general del proyecto
  - 1.2. Objetivos
  
2. COMPONENTES ELECTRÓNICOS
  - 2.1. CPU
  - 2.2. Pantalla táctil
  - 2.3. Proyector
  - 2.4. Soporte físico (Tarjetas)
  - 2.5. Batería
  
3. ESQUEMA FUNCIONAL
  
  
4. FUNCIONAMIENTO REAL PRODUCTO
  - 4.1. Conexión RFID-Raspberry Pi
  
  
5. FUNCIONAMIENTO PROTOTIPO
  - 5.1. Esquema de componentes
  - 5.2. Arduino
  - 5.3. Lectura de las tarjetas
  - 5.4. Proyector
  - 5.5. Montaje final
  
  
6. BIBLIOGRAFÍA

## 1. INTRODUCCIÓN

### 1.1. DESCRIPCIÓN GENERAL DEL PRODUCTO

El producto diseñado consiste en un dispositivo capaz de guardar los sueños de los niños, es decir, archivar con lo que sueña cada día. Se trata de un diario del sueño en el que el niño no solo puede guardar lo que ocurre en sus sueños sino que también puede elegir con que soñar cada noche.

Para conseguir que el niño elija una temática e intente soñar con ella, se proyecta un pequeño vídeo con esa temática, de forma que al ser con lo último que piensa antes de dormir, tiene muchas probabilidades de soñar con eso. El niño se irá a la cama emocionado no solo por ver el emocionante vídeo sino por esperar impaciente a descubrir si ha soñado con ello y poderlo archivarlo a la mañana siguiente.

A nivel práctico, esta idea se materializa en una serie de componentes electrónicos que de manera conjunta posibilitan el correcto funcionamiento del dispositivo. Serán necesarios los siguientes componentes:

- **CPU:** es el componente que controla el resto de dispositivos, en el cual se sitúa toda la información: el software, las proyecciones...
- **PROYECTOR:** necesario para proyectar las imágenes
- **PANTALLA TÁCTIL:** con la cual el niño interactúa y archiva sus sueños
- **SOPORTE FÍSICO PARA CADA PROYECCIÓN:** se busca que el niño disponga de una serie de objetos físicos que pueda coleccionar y que a cada uno de ellos le corresponda una proyección. Para ello es necesario algún sistema físico o digital que identifique a cada una de las tarjetas (Tarjetas con código bidi, código de barras, RDIF....)
- **LECTOR DEL CÓDIGO DEL SOPORTE FÍSICO:** es el encargado de leer el código de los soportes físicos y transmitir esta información a la CPU.

### 1.2. OBJETIVOS

El objetivo principal de este trabajo es ser capaces de explicar de forma precisa el funcionamiento real de este producto. Mostrar que es viable eléctricamente realizar este diseño, cuáles son los componentes que se necesitan, cuáles son sus características técnicas y el modo en el que se transmite la información entre ellos.

También se busca que el juguete sea seguro ya que lo van a emplear los niños, para ello se buscará que los componentes estén bien protegidos y situados de forma fija para evitar que los niños puedan entrar en contacto con ellos y que puedan estropearse o desconectarse si cae al suelo o con cualquier golpe.

Otro objetivo es hacer visible el funcionamiento del dispositivo para poder presentar un prototipo funcional, pese a que no contenga los componentes finales.

## 2. COMPONENTES

### 2.1. CPU

A la hora de elegir una CPU u ordenador para el dispositivo, se han de establecer unas premisas, es decir, que condiciones ha de cumplir este para que sea el idóneo para nuestro producto. Estas son:

- Ha de tener dimensiones reducidas para que encaje en las dimensiones establecidas del producto
- Ha de tener peso reducido ya que el juguete en su conjunto no puede ser muy pesado ya que tiene que ser ligero para que puedan interactuar los niños.
- Precio reducido para que no encarezca el precio del producto. Esto puede suponer unas características sencillas en cuanto a memoria y procesador interno, pero eso no supone un problema ya que el uso será reducido, necesitando solo que almacene la aplicación y los videos.

LO QUE OFRECE EL MERCADO:



*Figura 1: diferentes tipos de mini CPUs*

### RASPERRY PI

Un dispositivo que cumple con todas estas premisas es el Raspberry Pi. Se trata de es un ordenador de placa reducida o placa única. El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz, un procesador gráfico (GPU) VideoCore IV, y 512 MiB de memoria RAM. El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa.



Figura 2: Placa Raspberry Pi

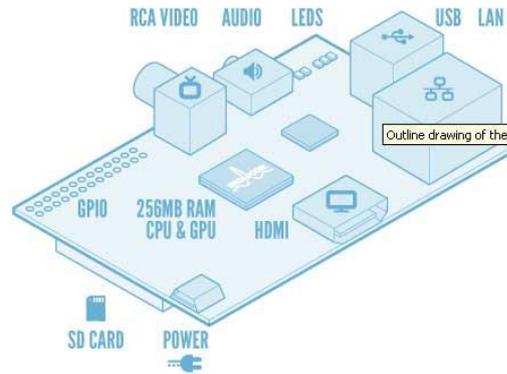


Figura 3: Puertos placa Raspberry Pi

El dispositivo cuenta con diferentes puertos para realizar las conexiones con el resto de dispositivos. Para el proyecto que se está desarrollando se necesitaría el puerto HDMI para conectar la pantalla y el puerto RCA video y USB para conectar el proyector en función de las características de éste. La batería iría conectada al puerto de POWER y el programa de la aplicación iría situado en la tarjeta SD.

Al estar basado en el software libre los precios de las placas son reducidos y las posibilidades son ilimitadas. Además se han diseñado muchos componentes electrónicos para utilizar a través de este sistema a muy buen precio también y lo que es más importante, muy fáciles de montar y manejar. A través de multitud de webs se pueden descargar softwares ya diseñados para poder incorporar en la placa. El Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux.

#### ALIMENTACIÓN:

La alimentación de la Raspberry Pi es importante. Se necesita un cargador de móvil o similar con conexión micro USB y que suministre más de 750 mA de corriente. Un puerto USB de ordenador generalmente es capaz de suministrar hasta 500 mA, así que tampoco se podrá conectar a un puerto USB 2.0 un cable USB a micro USB para alimentar la Raspberry Pi.

#### SISTEMA OPERATIVO:

A la hora de instalar el sistema operativo, se puede optar por la instalación en una tarjeta SD de todo lo que necesitemos. La Raspberry Pi puede utilizar tarjetas SD de clase 4 o superior gracias a su lector de tarjetas incluido. Pero es mejor contar con una tarjeta de memoria de baja capacidad (2 GB es un tamaño perfecto) e instalar todo en un pendrive USB o en un disco duro USB. En el segundo caso se puede conectar un Hub USB para poder alimentar discos que requieran más corriente u otros periféricos, ya que los puertos USB de la Pi no están pensados para alimentar dispositivos que consuman mucha energía. Siempre hay que dejar la tarjeta SD instalada, aunque se haga una instalación en un disco USB, ya que la información de arranque estará siempre en la tarjeta SD.

#### MINI PC

Un MiniPC es un ordenador en una barra USB, que utiliza Linux y que permite una experiencia consistente en cualquier pantalla o equipo de música, generalmente mediante un puerto HDMI, WiFi y conector de audio. Sirve de compañero para smartphones, tabletas, portátiles, así como añade capacidades a las actuales marcos MP4, televisores (para Smart TV), set top boxes y otros dispositivos que pueden utilizar almacenamiento masivo USB. Al tener USB OTG, permite conectar teclados, ratones y otros dispositivos USB. Su precio suele partir de 35 euros.



Figura 4: Mini PC Android 4.0.3 1GB Ram

Incluye puerto HDMI para poder enviar a la señal a un televisor opantalla, puerto USB y micro USB, ranura MicroSD para tarjeta de hasta 32gb con ficheros (videos,peliculas,música,etc...) E incluye tarjeta Wifi para poder conectarse a internet y a otros equipos.

Necesita alimentación mediante cable USB (5v,2A)

### Rikomagic MK802 IV

Los MK802 son un conjunto de aparatos comercializados por la empresa Rikomagic. Cuantan con un procesador Rockchip RK3188 de cuatro núcleos A9, lo que lo pone al nivel de un Tegra 3, pero le debemos sumar 2GB de RAM, 8GB de almacenamiento interno, y Bluetooth.



Figura 5: Rikomagic MK802 IV

## PRODUCCIÓN DEL PRODUCTO

Para la producción en serie de este producto no sería necesario emplear este tipo de dispositivos con tantas posibilidades y más complejos. El producto ya iría montado con su propio programa e interfaz preinstalado, el usuario no encontrará la interfaz de un ordenador completo como ocurre con los sistemas anteriores sino que el objetivo es que, al encender el aparato, se inicie directamente la aplicación del Archivador de sueños. Para conseguir esto se deberán utilizar microchips o también llamados, circuitos integrados en los que la información que incluye en su interior es introducida previamente y el usuario no puede alterar o modificar el programa.

Estos microchips son los que aparecen en multitud de productos en los que se quiere crear una interfaz para el usuario interactúe con él y para conseguirlo es necesario el uso de la electrónica. Un ejemplo podría ser los teléfonos móviles, los cuales ya tienen integrado el software en su interior y el usuario se limita a interactuar con él.

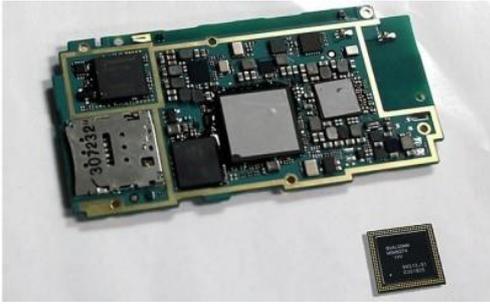


Figura 6: Imágenes circuitos integrados

## 1.2 PANTALLA

Una pantalla táctil es una pantalla que mediante un toque directo sobre su superficie permite la entrada de datos y órdenes al dispositivo, y a su vez muestra los resultados introducidos previamente; actuando como periférico de entrada y salida de datos, así como emulador de datos interinos erróneos al no tocarse efectivamente.

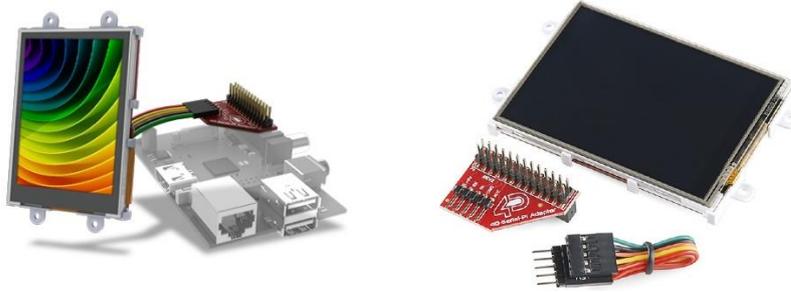
Según la tecnología que usen, hay dos tipos de pantallas táctiles de uso habitual:

Resistivas: Son más baratas y no les afectan el polvo ni el agua salada y, además de ser más precisas, pueden ser usadas con un puntero o con el dedo. Sin embargo, tienen hasta un 25% menos de brillo y son más gruesas, por lo que están siendo sustituidas por otras en los dispositivos móviles que precisan un tamaño y un peso ajustado y mayor brillo en la pantalla por la posibilidad de estar expuestos a la luz directa del sol.

Capacitivas: Basadas en sensores capacitivos, consisten en una capa de aislamiento eléctrico, como el cristal, recubierto con un conductor transparente, como el ITO (tin-doped indium oxide). Como el cuerpo humano es también un conductor eléctrico, tocando la superficie de la pantalla resulta una distorsión del campo electrostático de la pantalla, la cual es medida por el cambio de capacitancia (capacidad eléctrica). Diferentes tecnologías pueden ser usadas para determinar en qué posición de la pantalla fue hecho el toque. La posición es enviada al controlador para el procesamiento. La calidad de imagen es mejor, tienen mejor respuesta y algunas permiten el uso de varios dedos a la vez (multitouch). Sin embargo, son más caras y no se pueden usar con puntero normal, sino con uno especial para las pantallas capacitivas.

Existe una gran variedad de software dirigido al manejo de máquinas con pantallas táctiles y que puede ejecutarse en los principales sistemas operativos como son GNU/Linux, MacOS y Windows. En los tres casos existen versiones especiales de sus sistemas operativos que están adaptadas para su uso en tabletas y portátiles táctiles. Android es un sistema operativo de código abierto basado en el núcleo (kernel) de Linux, inicialmente desarrollado por Android Inc. y respaldado por Google.

En otro tipo de dispositivos como las PDAs o teléfonos con pantalla táctil también existen sistemas operativos como: PalmOS, Windows Phone, iOS, Android, BlackBerry OS, WebOS, Symbian OS, MeeGo o Maemo.



*Figura 7: Pantallas LCD para Raspberry Pi*

En las imágenes posteriores se muestran ejemplos de pantallas LCD táctiles para la placa Raspberry Pi, estas pantallas van conectados a la placa a través de un módulo intermedio de comunicación que es el que permite la entrada y salida de datos entre la pantalla y el ordenador.

### **1.3 PROYECTOR**

Este componente es fundamental para el producto ya que gracias a él el niño será capaz de ver las imágenes cada noche. Como se observó en el estudio de mercado, muchos de los dispositivos facilitadores del sueño infantil que encontramos hoy en día en el mercado, cuentan con un proyector, sin embargo estos proyectores se limitan a mostrar imágenes estáticas o un juego de luces en movimiento, pero en ninguno de los casos muestran secuencias de video. En esto se ha querido innovar y hacer que el producto sea diferente al de la competencia.

Una pregunta que era necesario plantearse era por qué no proyectaban vídeos siendo que esto puede ser mucho más llamativo que una imagen. La respuesta es que los proyectores pese a llevar años en el mercado, siempre han sido aparatos caros, de gran volumen y destinados a lugares como cines, aulas docentes o salas de presentaciones o conferencias. Sin embargo, hoy en día se pueden encontrar proyecciones de dimensiones muy reducidas, y de precio razonable que se puede emplear en muchas otras situaciones, o como en este caso para proyectar un video infantil.

Antes de mostrar lo que se encuentra en el mercado, es preciso conocer qué es un proyector y en qué principios se basa su funcionamiento:

Un proyector de vídeo o vídeo proyector es un aparato que recibe una señal de vídeo y proyecta la imagen correspondiente un sistema de lentes, permitiendo así mostrar imágenes fijas o en movimiento.

Todos los proyectores de vídeo utilizan una luz muy brillante para proyectar la imagen, y los más modernos pueden corregir curvas, borrones y otras inconsistencias a través de los ajustes manuales. La señal de vídeo de entrada puede provenir de diferentes fuentes, como un sintonizador de televisión (terrestre o vía satélite), un ordenador personal...

El funcionamiento de los proyectores se basa fundamentalmente en proyectar una imagen sobre una superficie utilizando un cañón de luz que lanza luz de colores sobre la pantalla. La diferencia principal entre los diferentes proyectores del mercado es la calidad de su lente en la que reside la nitidez de imagen que se obtiene.



*Figura 8: Proyector Yamaha*

### CÓMO FUNCIONA UN PROYECTOR:

Hay tres puntos imprescindibles a tener en cuenta a la hora de elegir un proyector, estos son:

1. **Resolución.** Mide el nivel de nitidez y la cantidad de detalle de la imagen proyectada. Las resoluciones más habituales son:
  - SVGA (800x600 px). Es la resolución más económica
  - XGA (1024x780 px). Es la resolución más común y con mejor relación calidad-precio.
  - WXGA (1280x800 px). Se emplea para realizar proyecciones de alta definición de formato panorámico. En esta resolución se consigue mayor detalle y área de proyección que en el resto.
2. **Luminosidad.** Indica la potencia de la lámpara. A mayor cantidad de lúmenes mayor brillo de la imagen, lo que también implica pantalla más grande y con más luz ambiente. Lo más corriente un proyector de:
  - 2500 lúmenes. Con ello se consiguen pantallas de 2 metros de ancho y una proyección sin necesidad de oscurecer.
  - 5000 lúmenes. Se consiguen pantallas de 3 a 5 metros de ancho.
3. **Conectividad.** Los conectores más habituales que se pueden encontrar en un proyector son:
  - VGA: es la conexión más expandida permite conectar el ordenador al proyector
  - HDMI: es una conexión digital de alta definición, ideal para proyectar películas
  - Video compuesto: es una conexión muy común para conectar vídeos DVD. Sin embargo, al tratarse de una conexión analógica muestra una imagen de peor calidad.



Figura 9: Puertos del proyector

4. **Óptica.** En esta característica hay que tener en cuenta dos aspectos importantes:
- Zoom. Es lo que permite poner la pantalla más grande o más pequeña.
  - Foco. Permite enfocar la imagen para que no aparezca borrosa.

TIPOS DE PROYECTOR:

#### Proyector de TRC

El proyector de tubo de rayos catódicos típicamente tiene tres tubos catódicos de alto rendimiento, uno rojo, otro verde y otro azul, y la imagen final se obtiene por la superposición de las tres imágenes (síntesis aditiva) en modo analógico.

Ventajas: es la más antigua, pero es la más extendida en aparatos de televisión.

Inconvenientes: al ser la más antigua, está en extinción en favor de los otros sistemas. Los proyectores de TRC son adecuados solamente para instalaciones fijas ya que son muy pesados y grandes, además tienen el inconveniente de la complejidad electrónica y mecánica de la superposición de colores.

#### Proyector LCD

El sistema de pantalla de cristal líquido es el más simple, por tanto uno de los más comunes y accesibles para el uso doméstico. En esta tecnología, la luz se divide en tres y se hace pasar a través de tres paneles de cristal líquido, uno para cada color fundamental (rojo, verde y azul); finalmente las imágenes se recomponen en una, constituida por píxeles, y son proyectadas sobre la pantalla mediante un objetivo.

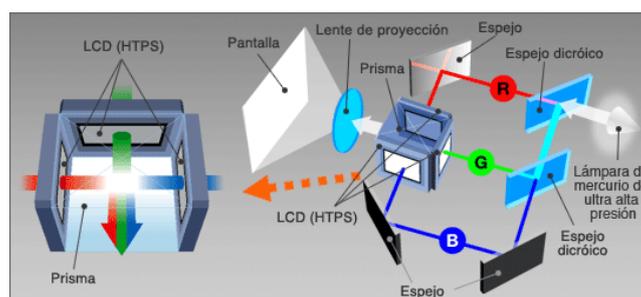


Figura 10: Esquema funcionamiento proyector LCD

Ventajas: es más eficiente que los sistemas DLP (imágenes más brillantes) y produce colores muy saturados.

Inconvenientes: es visible un efecto de pixelación, es probable la aparición de píxeles muertos y la vida de la lámpara es de aproximadamente 2000 horas.

### **Proyector 3D**

Proyector de última generación que muestra imágenes en una pantalla especial tratada de manera que las imágenes que proyecta envuelven al espectador dando la sensación de imagen envolvente.

### **MINI PROYECTORES:**

Dadas las dimensiones del producto a diseñar, el proyector ha de tener las dimensiones más reducidas posibles. En el mercado encontramos modelos que se ajustan a esta indicación:



*Figura 11: Imágenes mini proyectores*

Sin embargo todos estos proyectores pese a satisfacer la necesidad de un tamaño reducido, tienen un precio muy elevado y por lo tanto no son susceptibles a aplicarse al producto.

Visitando otras páginas webs y marcas menos conocidas se ha encontrado el modelo que por dimensiones, prestaciones y precio encajaría a la perfección.



**Dimensiones: 79x149x67mm (L x W x D)**

*Figura 12: Portable Projector*

### **Portable Projector For Kids**

#### **CARACTERÍSTICAS:**

- Vida útil: 30,000 Hours
- Fuente de alimentación: x AA Battery
- Fuente de luz: LED
- Contraste: 300: 1
- Luminosidad: 5 ANSI Lumens
- Resolución: 128x128
- Lámpara: 2W LED lamp
- Tamaño de las imágenes proyectadas: 5-100"
- Formato de audio: MP3, WMA, WAV
- Formato de imagen: JPG, GIF, BMP
- Formato de vídeo: WMA, WA

## **1.4 SOPORTE FÍSICO (TARJETAS)**

Para este producto se ha buscado desde el principio que los sueños fuesen reales, tangibles. Es decir, que los niños al elegir con que soñar no acudiesen simplemente a una base de datos y pulsasen a un botón con dicha temática, sino que cada tema tuviese su propio soporte físico. En un principio se optó por unas monedas, según las cuales, cada una representaría cada una de las temáticas, el niño elegiría una de las monedas y al insertarla en el archivador se proyectaría el video de la temática elegida.

El formato "moneda" se desechó y se optó por el formato "cromo" o "tarjeta". De forma que los niños pudiesen llegar a ampliar la colección, intercambiar sueños con los amigos, etc.

Para conseguir que por medio de una tarjeta el archivador fuese capaz de reconocer la temática y proyectar el video se plantearon una serie de posibilidades que se fueron descartando al estudiarlas y ver que no eran viables, hasta llegar a la solución idónea.

### **TARJETAS SD**

Secure Digital (SD) es un formato de tarjeta de memoria para dispositivos portátiles tales como cámaras fotográficas digitales, teléfonos móviles, computadoras portátiles e incluso videoconsolas (tanto de sobremesa como portátiles), entre muchos otros.

Esta idea de portátil y de capacidad de contener información hizo que fuese lo primero que se planteó. Según esto, a cada tarjeta le correspondería un video y al insertarlo este se proyectaría.

Sin embargo esta opción presentaba varios inconvenientes, en primer lugar, el tamaño reducido de las tarjetas que hacía que fuese muy fácil perderlas y en segundo lugar, el elevado precio que supondría tener un número elevado de tarjetas.

Analizando esta opción se llegó a las siguientes conclusiones:

- Los videos deberían estar dentro de la memoria del ordenador del proyector. Las tarjetas solo han de servir de código identificador, es decir, que a través de la tarjeta se

mande una señal al archivador que asocie a una proyección determinada y que tras ello comience la reproducción del vídeo.

- El tamaño de cada tarjeta debía ser como mínimo el tamaño de una tarjeta de crédito o cromo con el fin de no fuese tan sencillo perder las tarjetas.
- El precio de cada tarjeta debía ser lo más reducido posible, no solo teniendo en cuenta el precio final global del producto sino con vistas a poder comercializar las tarjetas de forma individual.

Siguiendo estas conclusiones se llegó a nuevas alternativas:

### **CÓDIGO BIDI (RQ) APLICADO A LA TARJETA**

El código QR (quick response code, «código de respuesta rápida») es un módulo útil para almacenar información en una matriz de puntos o un código de barras bidimensional. Se caracteriza por los tres cuadrados que se encuentran en las esquinas y que permiten detectar la posición del código al lector. La inclusión de software que lee códigos QR en teléfonos móviles, ha permitido nuevos usos orientados al consumidor.



*Figura 13: Código RQ*

Esta opción fue rápidamente descartada ya que los códigos QR pueden leerse desde PC, smartphone o tableta mediante dispositivos de captura de imagen, como puede ser un escáner o la cámara de fotos, programas que lean los datos QR y una conexión a Internet para las direcciones web. Tanto la cámara como la conexión a internet eran imposibles de trasladar al producto de forma viable.

### **CÓDIGO DE BARRAS APLICADO A LA TARJETA**

El código de barras es un código basado en la representación mediante un conjunto de líneas paralelas verticales de distinto grosor y espaciado que en su conjunto contienen una determinada información, es decir, las barras y espacios del código representan pequeñas cadenas de caracteres. De este modo, el código de barras permite reconocer rápidamente un artículo de forma única, global y no ambigua.



*Figura 13: Código de barras*

Según esta definición, esta podría ser la opción más idónea, sin embargo, para leer el código es necesario un escáner de código de barras. Con este escáner se consiguen leer los códigos de barras pasando un pequeño punto de luz sobre el símbolo del código de barras impreso. Solo se ve una fina línea roja emitida desde el escáner láser. Pero lo que pasa es que las barras oscuras absorben la fuente de luz del escáner y la misma se refleja en los espacios luminosos. Un dispositivo del escáner toma la luz reflejada y la convierte en una señal eléctrica.

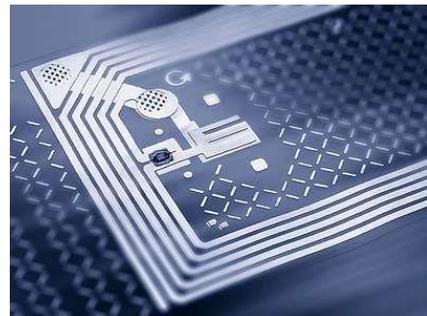


*Figura 14: Escáner código de barras*

Este escáner como componente individual en el mercado se vende con un aspecto formal y con unas dimensiones que imposibilitan el poder adoptar este sistema en el producto. Además el precio de estos escáner supera los 20€ lo cual hace que además no sea un sistema rentable económicamente.

## **TARJETAS RFID**

RFID (siglas de Radio Frequency IDentification, en español identificación por radiofrecuencia) es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio.



*Figura 15: Etiqueta RFID*

Un sistema RFID consta de los siguientes tres componentes:

Etiqueta RFID o transpondedor: compuesta por una antena, un transductor radio y un material encapsulado o chip. El propósito de la antena es permitirle al chip, el cual contiene la información, transmitir la información de identificación de la etiqueta. Existen varios tipos de etiquetas. El chip posee una memoria interna con una capacidad que depende del modelo y varía de una decena a millares de bytes. Existen varios tipos de memoria:

Solo lectura: el código de identificación que contiene es único y es personalizado durante la fabricación de la etiqueta.

De lectura y escritura: la información de identificación puede ser modificada por el lector.

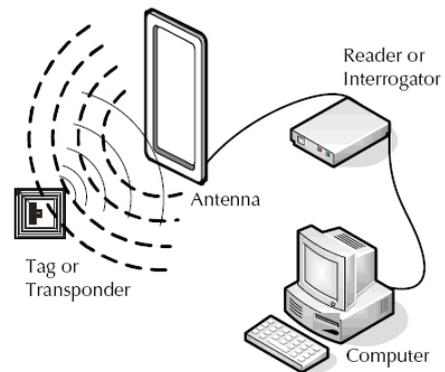
Anticolisión. Se trata de etiquetas especiales que permiten que un lector identifique varias al mismo tiempo (habitualmente las etiquetas deben entrar una a una en la zona de cobertura del lector).

Lector de RFID o transceptor: compuesto por una antena, un transceptor y un decodificador. El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones.

Cuando capta una señal de una etiqueta (la cual contiene la información de identificación de esta), extrae la información y se la pasa al subsistema de procesamiento de datos.

### ¿CÓMO FUNCIONA?

1. El interrogador genera un campo de radiofrecuencia, normalmente conmutando una bobina a alta frecuencia. Las frecuencias usuales van desde 125 KHz hasta la banda ISM de 2.4 Ghz, incluso más.
2. El campo de radiofrecuencia genera una corriente eléctrica sobre la bobina de recepción del dispositivo. Esta señal es rectificadora y de esta manera se alimenta el circuito.
3. Cuando la alimentación llega a ser suficiente el circuito transmite sus datos.
4. El interrogador detecta los datos transmitidos por la tarjeta como una perturbación del propio nivel de la señal.



*Figura 16: Esquema funcionamiento tarjetas RFID*

La señal recibida por el interrogador desde la tarjeta está a un nivel de -60 db por debajo de la portadora de transmisión. El rango de lectura para la mayoría de los casos está entre los 30 y 80 centímetros de distancia entre interrogador y tarjeta.

Se trata por tanto de un sistema sencillo de implementar y que ofrece infinidad de usos. Para poder aplicar este sistema al producto es necesario disponer de los dos elementos antes nombrados, por un lado las etiquetas RFID que se incorporarán en las tarjetas y por otro, el lector RFID que será el encargado de identificar las tarjetas y de transmitir la información a la CPU donde se asociará cada código de cada tarjeta con un determinado vídeo.

En el mercado podemos encontrar infinidad de opciones y de precios. Las tarjetas compradas en cantidades muy grandes pueden llegar a costar por debajo de 30 céntimos la unidad. En cuanto al lector, existen muchos modelos y precios, pudiéndose llegar a encontrar lectores por menos de 4€ la unidad, por lo que comprando al por mayor ese precio puede ser mucho menor.

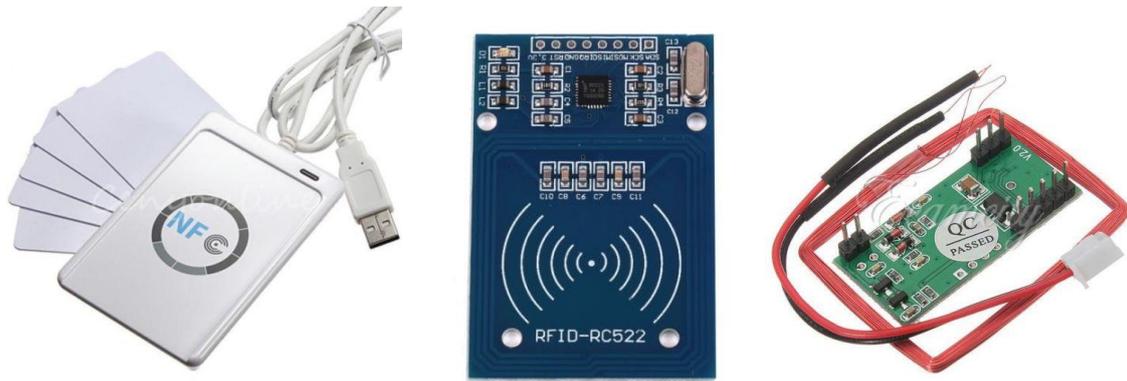


Figura 17: Diferentes tipos de lectores RFID

### 3. ESQUEMA FUNCIONAL

Tras haber elegido los componentes más apropiados para el Archivero de Sueños, es preciso establecer el esquema de funcionamiento a nivel global con el fin de observar cuales son los problemas y dificultades que pueden aparecer a la hora de hacer las conexiones entre los componentes.

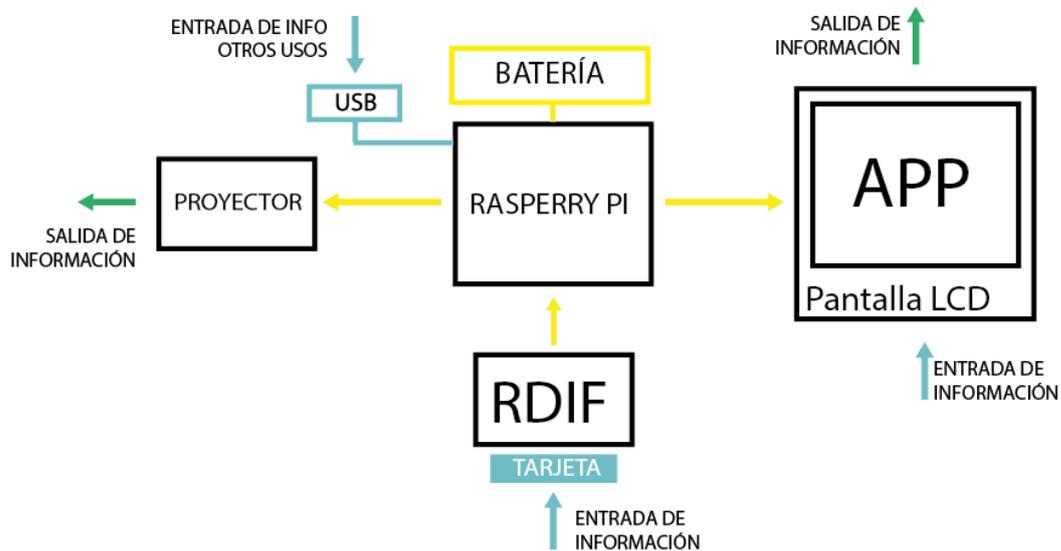


Figura 18: Esquema de componentes

Tal y como muestra el esquema, se ve como en el centro se encuentra el cerebro del Archivero, el Raspberry Pi el cual recibe información del usuario a través de la pantalla táctil y a través del lector RDIF, a su vez transmite información al proyector una vez que ha recibido la señal de qué ha de proyectarse en función de la tarjeta EDIF que se haya elegido.

Todos estos componentes van alimentados eléctricamente a través de la batería que va enchufada directamente sobre el Raspberry pi, el cual será quien alimente indirectamente al resto de componentes.

Además se ha decidido incorporar una ranura de lectura de tarjeta SD para dar la oportunidad de incorporar nuevas proyecciones o utilizar el proyector individualmente independientemente de la función del Archivador de Sueños.

Por lo que los componentes necesarios para el modelo real serían:

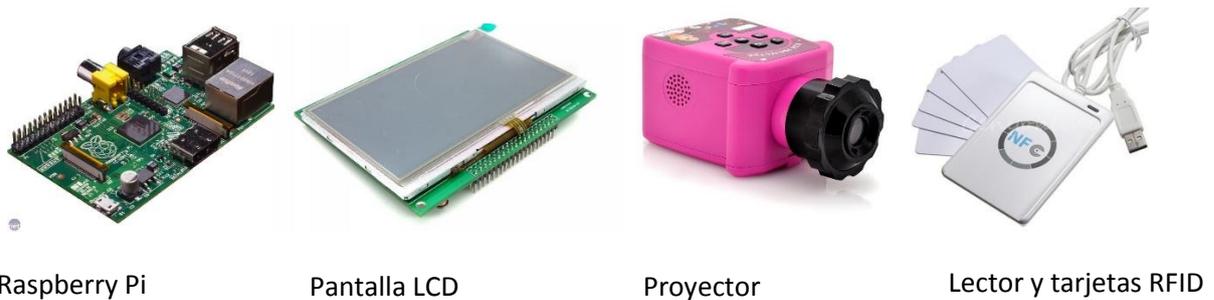


Figura 19: Componentes necesarios

#### 4. FUNCIONAMIENTO REAL PRODUCTO

##### 4.1. CONEXIÓN RFID-RASPERRY PI

Los elementos necesarios para llevar a cabo esta conexión son:

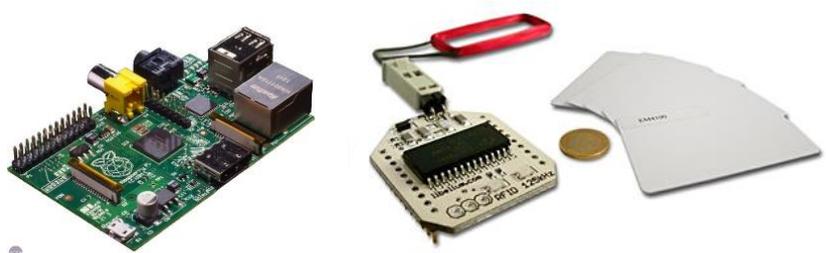


Figura 20: De izquierda a derecha, Raspberry pi, módulo lector RFID para Raspberry Pi y tarjetas.

Para llevar a cabo la conexión hay que seguir los siguientes pasos:

1. Conectar el módulo RFID al Raspberry Pi  
La configuración del hardware es muy fácil, sólo hay que enchufar el Raspberry Pi a la placa Raspberry Pi. Tras ello hay que programar el Raspberry Pi para que sea capaz de comunicarse con el módulo RFID a través del puerto serie (Serial.read ())
2. Lectura de las tarjetas

En esta parte se muestra un ejemplo de las tarjetas de solo lectura EM4100 Raspberry Pi. Utilizando el modo Auto Read, se observa como Raspberry Pi está esperando todo el tiempo, cuando se detecta una tarjeta, se lee su código, parpadea el LED e imprime el código de la salida estándar. Ejemplo:

Partiendo de los datos de una determinada tarjeta:

Command
0x87 - Set Auto Mode: EM4102 Mode - Parity decoded - Manchester RF/64
FF 01 09 87 01 03 02 00 10 20 30 40 37

El código de programación para conseguir la lectura de las tarjetas es el siguiente:

```
int led = 13;
byte data_1 = 0x00;
byte data_2 = 0x00;
byte data_3 = 0x00;
byte data_4 = 0x00;
byte data_5 = 0x00;
int val = 0;

void setup(){
    // Start serial port 19200 bps
    Serial.begin(19200);
    pinMode(led, OUTPUT);

    delay(500);

    // Setting Auto Read Mode - EM4102 Decoded Mode - No password
    // command: FF 01 09 87 01 03 02 00 10 20 30 40 37
    Serial.print(0xFF,BYTE);
    Serial.print(0x01,BYTE);
    Serial.print(0x09,BYTE);
    Serial.print(0x87,BYTE);
    Serial.print(0x01,BYTE);
    Serial.print(0x03,BYTE);
    Serial.print(0x02,BYTE);
    Serial.print(0x00,BYTE);
    Serial.print(0x10,BYTE);
    Serial.print(0x20,BYTE);
    Serial.print(0x30,BYTE);
    Serial.print(0x40,BYTE);
    Serial.print(0x37,BYTE);

    delay(500);
    Serial.flush();

    printf("\n");
```

```

        printf("RFID module started in Auto Read Mode\n");
    }

void loop(){

    val = Serial.read();
    while (val != 0xff){
        printf("Waiting card\n");
        val = Serial.read();
        delay(1000);
    }

    // Serial.read();    // we read ff
    Serial.read();    // we read 01
    Serial.read();    // we read 06
    Serial.read();    // we read 10
    data_1 = Serial.read();    // we read data 1
    data_2 = Serial.read();    // we read data 2
    data_3 = Serial.read();    // we read data 3
    data_4 = Serial.read();    // we read data 4
    data_5 = Serial.read();    // we read data 5
    Serial.read();    // we read checksum

    // Led blink
    for(int i = 0;i < 4;i++){
        digitalWrite(led,HIGH);
        delay(500);
        digitalWrite(led,LOW);
        delay(500);
    }

    // Printing the code of the card
    printf("\n");
    printf("EM4100 card found - Code: ");
    printf("%x",data_1);
    printf("%x",data_2);
    printf("%x",data_3);
    printf("%x",data_4);
    printf("%x",data_5);

    printf("\n\n");
}

int main (){
    setup();
    while(1){
        loop();
    }
    return (0);
}

```

Y con esto, al poner en funcionamiento del programa tendría que enviar el mensaje de esperando tarjeta hasta que reciba una señal distinta de 0 y una vez que ha recibido el código de la tarjeta lo reconoce y lo muestra como ocurre a continuación:

```
RFID module started in Auto Read Mode
Waiting card
Waiting card
Waiting card
Waiting card

EM4100 card found - Code: 24 00 95 BC 92

Waiting card
Waiting card
Waiting card
Waiting card

EM4100 card found - Code: 21 00 39 47 B0

Waiting card
Waiting card
Waiting card
Waiting card

EM4100 card found - Code: 24 00 7D 94 B5

Waiting card
Waiting card
```

**\*\* Nota:** el funcionamiento de este programa es meramente teórico, es decir, no se ha llevado a la práctica para comprobar su validez ya que no se adquirido una placa Raspberry Pi con la que poder probarlo. El código de programación es libre y está disponible en la página web de los vendedores de los componentes.

## 5. FUNCIONAMIENTO PROTOTIPO

### 5.1. ESQUEMA DE COMPONENTES

Para conseguir representar el funcionamiento el juguete se han adquirido una serie de componentes, algunos de ellos podrían coincidir con los que aparecen en el producto real y otros solo se emplean para la simulación.

LISTA DE COMPONENTES:

- Proyector “Portable Projector For Kids”
- Tablet
- Arduino
- 125 KHZ EM4100 RFID Módulo lector tarjeta Junta RDM630 UART (Compatible con Arduino)
- Tarjetas RFID 13.56 MHZ / NFC CARD



Figura 21: Componentes para el prototipo

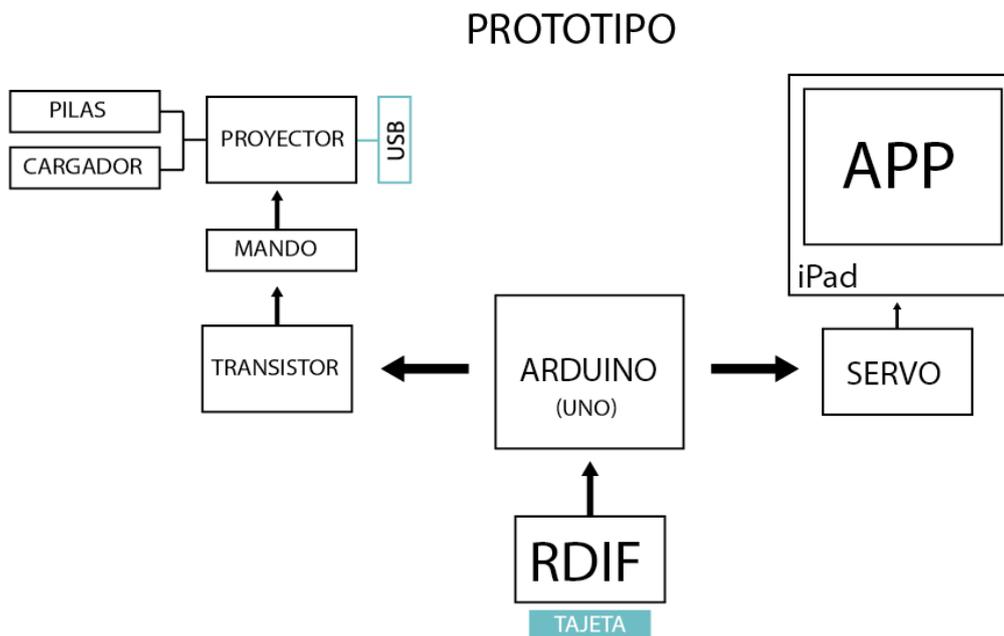


Figura 22: Esquema componentes del prototipo

La diferencia con el modelo real radica en la ausencia de la Raspberry Pi, esta se sustituye por la Tablet que actuará como cerebro de la operación y por el Arduino. El cual es necesario para transmitir el código de las pantallas al ordenador y que esté sea capaz de descifrar ese código y asociarle una secuencia de imágenes.

## 5.2. ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida.<sup>4</sup> Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

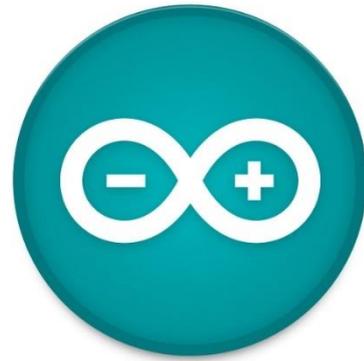


Figura 23: Logotipo Arduino

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data). Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, y controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador.

Las características de la placa más básica y la que se empleará en el prototipo son:

### ARDUINO UNO

- Microcontrolador: ATmega328
- Tensión de alimentación: 5 V
- Tensión de entrada recomendada: 7-12 V
- Límites de la tensión de entrada: 6-20 V
- Pines de entrada/salida digitales: 14 (de los cuales 6 proveen salidas PWM)
- Entradas analógicas: 6
- Corriente máxima por pin E/S: 40 mA
- Memoria Flash (memoria de programa): 32 KB (0,5 KB están ocupados)
- Memoria SRAM: 2 KB
- Memoria EEPROM: 1 KB
- Frecuencia de reloj: 16 MHz

Las partes de la placa en las que se colocan los diferentes puertos de entrada y de salida son:

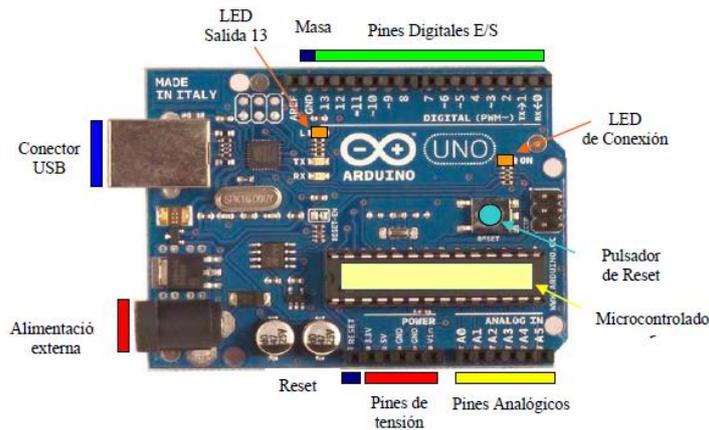


Figura 24: Placa arduino con los distintos pines

**Alimentación:** La placa puede ser alimentada o por una conexión USB o por una fuente de tensión externa. La tensión de suministro externo puede estar comprendida entre 6 V y 20 V, aunque se recomienda que esté comprendida entre 7 V y 12 V.

**Pines de tensión:** a través de estos pines se pueden obtener diversas tensiones de alimentación para posibles dispositivos electrónicos complementarios con la placa de Arduino.

**Vin:** tensión de salida igual que la tensión que alimenta la placa Arduino (la que entra por la alimentación externa).

**5 V:** salida de tensión a 5 V independientemente del valor de la tensión de alimentación.

**3,3 V:** salida de tensión de 3,3 V independiente del valor de la tensión de alimentación (puede suministrar como máximo 50 mA).

**Masa:** Es el pin respecto del cual están referenciadas todas las tensiones

**Pines digitales:** son 14 pines (del 0 al 13) que pueden utilizarse como entradas o salidas digitales. Las tensiones digitales operan entre 0 V (cero lógico) y 5 V (uno lógico). Los pines 3, 5, 6, 9, 10 y 11, además pueden operar como salidas moduladas en anchura de pulso (PWM es decir, Pulse Width Modulation).

**Pines analógicos:** son 6 pines (del 0 al 5) correspondientes a 6 entradas analógicas. Estas entradas analógicas pueden leer tensiones comprendidas entre 0 V y 5 V convirtiéndolas a valores digitales entre 0 y 1023 (convertor analógico digital de 10 bits de resolución).

**Pulsador de Reset:** cuando se acciona el Reset, el programa empieza a funcionar nuevamente desde el principio.

**LED de Conexión:** es un LED que indica cuando la placa está conectada a tensión.

**LED Salida 13:** es un LED que puede gestionarse su encendido o apagado desde el pin 13, cuando dicho pin está programado como salida digital

**Microcontrolador:** Es el circuito integrado que ejecuta secuencialmente todas las ordenes que están grabadas en su memoria de programa.

### 5.3. LECTURA DE LAS TARJETAS

Para conseguir mostrar el funcionamiento de las tarjetas RFID se han adquirido 3 tarjetas y un lector compatible con Arduino.

Para hacer la conexión el primer paso es conocer las características del lector y el modo que este ha de conectarse a la placa Arduino. El esquema de puertos del lector elegido está disponible en la web y es el que indica cómo ha de hacerse la conexión.

El lector adquirido es el modelo RDM6300, cuyas características técnicas son:

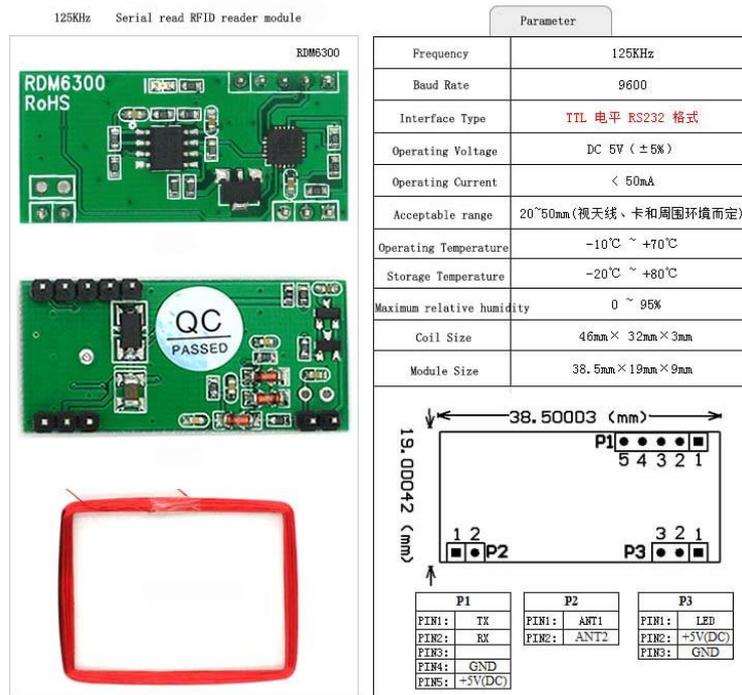
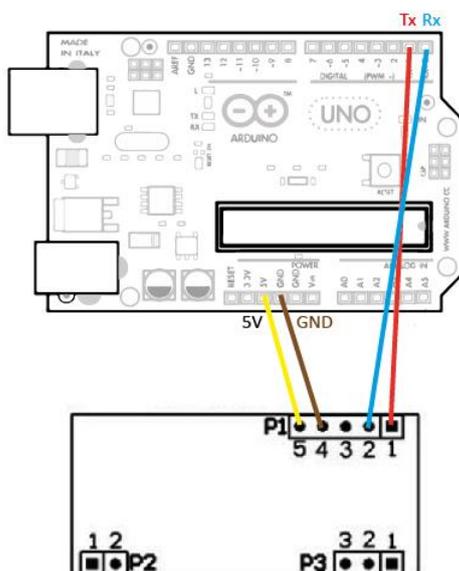


Figura 25: Ficha técnica lector RFID RDM 6300

Siguiendo el esquema con conexión establecida con el Arduino es la siguiente:



- El cable rojo es el que une el pin 1 de la serie P1 del lector con el pin digital Tx de la placa Arduino.
- El cable azul es el que une el pin 2 de la serie P1 del lector con el pin digital Rx de la placa Arduino.
- El cable marrón es el que une el pin 4 de la serie P1 del lector con la masa de la placa Arduino.
- El cable amarillo es el que une el pin 5 de la serie P1 del lector con el pin de tensión de 5V.

Figura 26: Conexión placa Arduino con el lector RFID

En la realidad el montaje queda así:



Figura 27 y 28: Fotos tomadas al circuito.

Para conseguir interpretar el código de las tarjetas se programó un sencillo programa para en primer lugar conocer el código de cada una de las tarjetas.

CÓDIGO:

```
char c;  
void setup(){  
  Serial.begin(9600);  
}  
void loop(){  
  c= Serial.read();  
  if ((c>10) && (c<128)){  
    Serial.write(c);  
  }  
}
```

La instrucción `if ((c>10) && (c<128))` es necesaria ya que es la que marca que solo escriba la información si le llega información real al Arduino. Ya que aunque no le llegue información, el Arduino no para de transmitir “no hay información”. Y esto imposibilitaría el ver el código de la tarjeta.

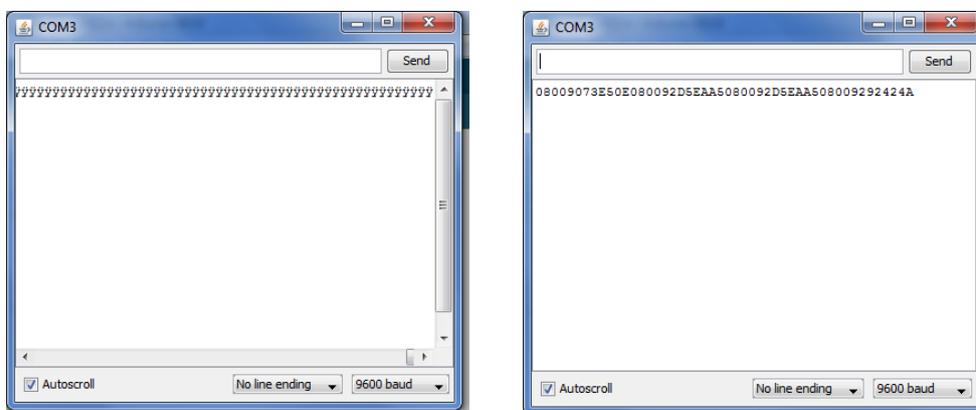


Figura 29: Pantalla resultante de la lectura del código

A través de este sencillo programa se consiguieron descifrar los códigos de las tarjetas:

TARJETA	CÓDIGO
1	08009073E50E
2	080092D5EAA5
3	08009292424A

El siguiente paso era transformar ese código en otro mensaje. Para comenzar se tradujo el código a 3 sencillos caracteres de forma que al acercar la tarjeta 1 aparecía en la pantalla la letra "A", con la tarjeta 2 la letra "B" y con la tarjeta 3 la letra "C".

Esto se consiguió con el siguiente código:

CÓDIGO TRANSMITIR A,B,C:

```
char c = 200;
char ultimoC;
boolean hayQueMandarMensaje = false;
boolean r4 = false;

void setup(){
  Serial.begin(9600);
}

void loop(){
  c = Serial.read();

  if((c == '7'))
    Serial.println("A");

  if((c == 'D'))
    Serial.println("B");

  if(c == '4'){
    if(r4){
      Serial.println("C");
      r4 = false;
    }
    else{
      r4 = true;
    }
  }
  ultimoC = c;
  delay(5);
}
```

Para conseguirlo de forma sencilla, se identificó un carácter único en cada serie con el fin de que el programa al identificar ese carácter ya fuese capaz de identificar la tarjeta. Para un número más elevado de tarjetas este sistema no es viable ya que no va a existir un carácter único en cada una de ellas. En este caso habría que hacer que reconociera el código entero lo cual es totalmente viable pero requiere una complejidad mayor a la hora de programarlo.

08009073E50E

080092D5EAA5

08009292424<sup>a</sup>

Una vez que se fue capaz de asociar el código interno de la tarjeta a un mensaje diferente a este, se buscó a traducirlo a algo más complejo que un simple carácter. Se comenzó asociando cada tarjeta a un color, a una imagen y por último a un video. Para conseguir esto se tuvo que emplear un programa distinto al que se emplea para programar en Arduino. El programa empleado fue **Processing**.



Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital.

*Figura 30: Imagotipo programa Processing*

## COLOR

Empleando este programa se comenzó asociando el código de la tarjeta a un color. Para ello se utilizó el siguiente código:

```
import processing.serial.*;
float redValue = 0;    // red value
float greenValue = 0; // green value
float blueValue = 0;  // blue value

Serial myPort;

void setup() {
  size(200, 200);

  println(Serial.list());
  // I know that the first port in the serial list is always my Arduino, so I open Serial.list()[0].
  // Open whatever port is the one you're using.

  myPort = new Serial(this, Serial.list()[0], 9600);

  myPort.bufferUntil('\n');
```

```

}

void draw() {
// set the background color with the color values:
background(redValue, greenValue, blueValue);
}

void serialEvent(Serial myPort) {
// get the ASCII string:
//String inString = myPort.readStringUntil('\n');
char r = myPort.readChar();

switch(r){
case 'A':
    redValue = 255;
    greenValue = 0;
    blueValue = 0;
    break;
case 'B':
    redValue = 0;
    greenValue = 255;
    blueValue = 0;
    break;
case 'C':
    redValue = 0;
    greenValue = 0;
    blueValue = 255;
}
}

```

## IMÁGENES

Para conseguir que a cada tarjeta le correspondiese una imagen distinta y basándose en un ejemplo que aparece en el propio programa, se implementó el siguiente código:

```

import processing.serial.*;

float redValue = 0;    // red value
float greenValue = 0; // green value
float blueValue = 0;  // blue value

Serial myPort;
PImage img;
void setup() {
size(640, 360);
img = loadImage("A.jpg");

// List all the available serial ports

```

```
println(Serial.list());
// I know that the first port in the serial list on my mac
// is always my Arduino, so I open Serial.list()[0].
// Open whatever port is the one you're using.
myPort = new Serial(this, Serial.list()[0], 9600);

myPort.bufferUntil('\n');
}

void draw() {
// set the background color with the color values:
//background(redValue, greenValue, blueValue);
image(img, 0, 0);
}

void serialEvent(Serial myPort) {
// get the ASCII string:
//String inString = myPort.readStringUntil('\n');
char r = myPort.readChar();

switch(r){
case 'A':
    redValue = 255;
    greenValue = 0;
    blueValue = 0;
    img = loadImage("A.jpg");
    break;
case 'B':
    redValue = 0;
    greenValue = 255;
    blueValue = 0;
    img = loadImage("B.jpg");
    break;
case 'C':
    redValue = 0;
    greenValue = 0;
    blueValue = 255;
    img = loadImage("C.jpg");
}
```

## VÍDEO

Por último para alcanzar el objetivo que es el de mostrar un video con cada tarjeta y avanzando en complejidad con respecto a los anteriores programas. Esto se consiguió a través del siguiente código:

```
import processing.video.*;
  PImage img;
  Movie myMovie;

void setup() {
  size(600, 600);
  myMovie = new Movie(this, "tarta.mov");
  myMovie.loop();
}

println(Movie.list());
myPort = new Movie(this, Movie.list()[0], 9600);
myPort.bufferUntil('\n');

void draw() {
  tint(255, 20);
  image(myMovie, mouseX, mouseY);
}

void movieEvent(Movie m) {
  m.read();
  char r=myPort.readChar()

switch(r){
  case 'A':
    img = myMovie (this, "tarta.MOV");
    break;
  case 'B':
    img = myMovie (this, "barco.MOV");
    break;
  case 'C':
    img = myMovie (this, "tarta.MOV");
  }
}
```

#### 5.4. PROYECTOR

Para conseguir controlar el proyector adquirido con las tarjetas Rfid, se comenzó convirtiendo los botones físicos de la carcasa del proyector en proyectores digitales que fuesen controlados desde el ordenador.

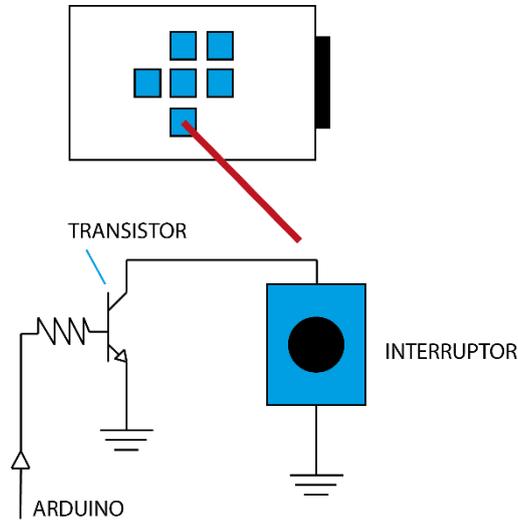


Figura 31: Interruptores del proyector conectados a la placa arduino

Para conseguir esa conversión se emplearon transistores, los cuales son dispositivos electrónico semiconductores utilizados para producir una señal de salida en respuesta a otra señal de entrada. Habrá un transistor por cada interruptor. Estos irán conectados a los pines digitales de salida del Arduino. El funcionamiento es sencillo, el Arduino enviará una señal eléctrica por el puerto correspondiente al interruptor que quiera accionar, al pasar corriente por ese canal el transistor actuará como interruptor permitiendo en ese momento el paso de la corriente y generando así el mismo efecto en el interruptor que si se pulsase físicamente.

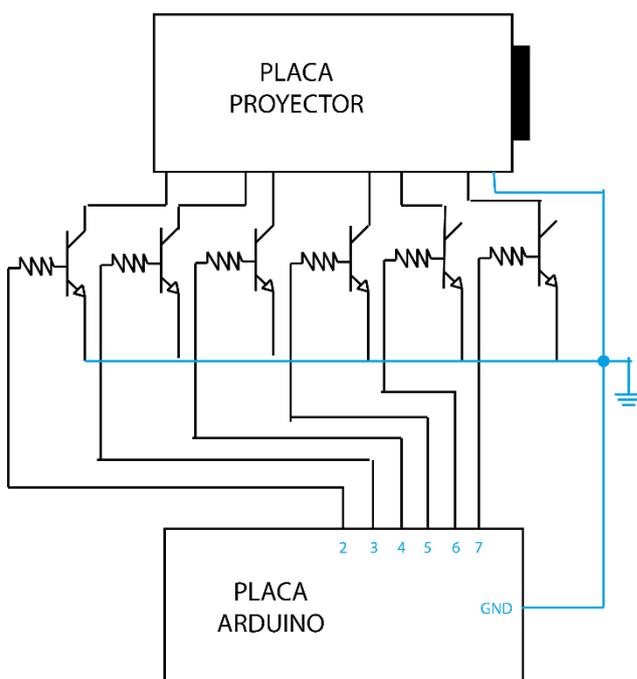


Figura 32: Esquema conexión proyector placa arduino

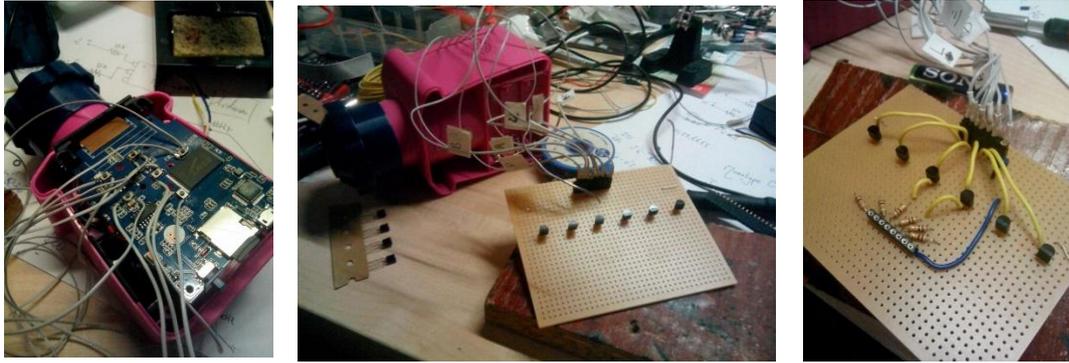


Figura 32: Secuencia de montaje

Para el montaje se emplearon los siguientes componentes:

- 10 resistencias BC548 (10K $\Omega$ )
- 10 transistores 10kohm
- Placa pcb perforada
- Tira de pines hembra
- Tira de pines machos

Con este montaje se consiguió controlar el proyector a través del ordenador a través de la programación de un programa que transmitiese información por cada uno de los puertos.

### 5.5. MONTAJE FINAL

Finalmente había que combinar el proyector con las tarjetas para que a través de estas se pudiesen accionar los diferentes videos.

Para ello se combinó el código de los dos programas para crear uno nuevo en el que a través de la información obtenida de la tarjeta, el Arduino fuese capaz de transmitir una secuencia de instrucciones que pusiesen en funcionamiento el proyector y activase un video según el código de cada tarjeta.



Figura 33:: Montaje final

## CÓDIGO DE PROGRAMACIÓN:

```
int incomingByte; // a variable to read incoming serial data into
boolean seHaPasadoTarjeta = false;
unsigned long tiempoAnterior = 0;
```

```
#define UP 0
#define RIGHT 1
#define DOWN 2
#define LEFT 3
#define CENTER 4
#define BACK 5
```

```
byte pines[] = {5, 7, 2, 3, 4, 6};
```

```
char c = 200;
char ultimoC;
boolean hayQueMandarMensaje = false;
boolean r4 = false;
```

```
void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  // initialize the LED pin as an output:
  //pinMode(ledPin, OUTPUT);
```

```
  for (byte i = 0; i < 6; i++){
    pinMode(pines[i], OUTPUT);
  }
}
```

```
void loop() {
  if(Serial.available() > 0)
    c = Serial.read();
  if(!seHaPasadoTarjeta){
    if((c == '7')){
      Serial.println("A");
      pulsacion(CENTER, 100);
      delay(100);
      pulsacion(CENTER, 100);
      seHaPasadoTarjeta = true;
      tiempoAnterior = millis();
    }
    if((c == 'D')){
      Serial.println("B");
      pulsacion(RIGHT, 100);
      seHaPasadoTarjeta = true;
      tiempoAnterior = millis();
    }
  }
}
```

```

}
if(c == '4'){
  if(r4){
    Serial.println("C");
    pulsacion(RIGHT, 100);
    r4 = false;
    seHaPasadoTarjeta = true;
    tiempoAnterior = millis();
  }
  else{
    r4 = true;
  }
}
ultimoC = c;
delay(5);
}
else{
  if((millis() - tiempoAnterior) > 5000){
    seHaPasadoTarjeta = false;
  }
}
}
void pulsacion(byte b, int t){ //Se le pasa el botón a ser pulsado y el tiempo
  digitalWrite(pines[b], HIGH);
  delay(t);
  digitalWrite(pines[b], LOW);
}

```

## 6. BIBLIOGRAFÍA:

### Raspberry Pi y mini PCs

- [es.wikipedia.org/wiki/Raspberry\\_Pi#Especificaciones\\_t.C3.A9cnicas](https://es.wikipedia.org/wiki/Raspberry_Pi#Especificaciones_t.C3.A9cnicas)
- [www.raspberrypi.org](http://www.raspberrypi.org)
- [www.xatakahome.com/trucos-y-bricolaje-smart/pitft-y-pitft-pibow-una-elegante-forma-de-integrar-una-pantalla-tactil-con-tu-raspberry-pi](http://www.xatakahome.com/trucos-y-bricolaje-smart/pitft-y-pitft-pibow-una-elegante-forma-de-integrar-una-pantalla-tactil-con-tu-raspberry-pi)
- [gizmologia.com/2013/11/mini-ordenadores-android](http://gizmologia.com/2013/11/mini-ordenadores-android)
- [www.digitaltrends.com/computing/can-a-50-mini-pc-replace-your-desktop/#!XAf0z](http://www.digitaltrends.com/computing/can-a-50-mini-pc-replace-your-desktop/#!XAf0z)

### Componentes:

- [comohacer.eu/como-funciona-un-proyector/](http://comohacer.eu/como-funciona-un-proyector/)
- [www.ebay.es](http://www.ebay.es)
- [www.cooking-hacks.com](http://www.cooking-hacks.com)
- [www.dealextreme.com](http://www.dealextreme.com)
- [www.ecojoven.com/dos/03/RFID.html](http://www.ecojoven.com/dos/03/RFID.html)

### Códigos de programación:

- [www.cooking-hacks.com/documentation/tutorials/raspberry-pi-rfid](http://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-rfid)
- [processing.org/reference/libraries/video/Movie.html](http://processing.org/reference/libraries/video/Movie.html)