



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



MÁSTER ARTES VISUALES Y MULTIMEDIA
DEPARTAMENTOS DE PINTURA Y ESCULTURA
FACULTAD DE BELLAS ARTES

**DIGITOCAMERA.
DISPOSITIVO FOTOGRÁFICO
HÍBRIDO DIGITAL Y ANALÓGICO**

Trabajo Final de Máster

Realizado por:
Alexandre García Sauri
Dirigido por:
María José Martínez de Pisón Ramón
Valencia, Julio de 2015

Agradecimientos

Quiero agradecer primero a mi tutora María José Martínez de Pisón por creer desde el primer momento en el proyecto y animarme a seguir adelante. Por supuesto agradecer a todos los profesores del Máster Oficial de Artes Visuales y Multimedia, por los conocimientos que he recibido durante estos dos años y por el trato afectuoso; especialmente a los profesores Miguel Sánchez por iniciarme en los secretos de la electrónica y Salvador Sales por su atención y ayuda junto a David Barrera quien me orientó mucho para aprender a programar la aplicación de Android Studio.

También a mis compañeros del curso por aguantarme día a día y ayudarme en todo lo que sabían.

A los técnicos del plató y laboratorio de fotografía Josep e Idoia por su paciencia y su permisividad. Y muy especialmente al técnico del Máster Raul León por su ayuda en todas las fases del proyecto, aprendiendo como más me gusta, divirtiéndome.

Dar las gracias también a Carmen Gray y Florentini Pete, por su ayuda en el ámbito fotográfico, por presentarme a *Carmencita film lab* y rodearme de gente profesional a la vez que económica.

A la gente que me rodea y hace que me sienta vivo (ellos saben quienes son). A mis padres.

INDICE

Introducción.....	7
1. PLANTEAMIENTO DEL PROYECTO.....	11
1.1. CONTEXTO TEÓRICO.....	12
1.2. REFERENTES ARTÍSTICOS.....	19
2. DESCRIPCIÓN TÉCNICA.....	24
2.1. DISPLAYS	25
2.2 PROCEDIMIENTO DE CONSTRUCCIÓN	40
4.CONCLUSIÓN Y TRABAJO FUTURO	61
5.BIBLIOGRAFÍA.....	68

Introducción

Este Proyecto se inscribe en las líneas de investigación Cine expandido y Dispositivos fílmicos del máster AVM, así como a una línea de investigación personal; y se sitúa en la opción investigación aplicada.

Digitocamera tiene como punto de partida la modificación de una cámara fotográfica para desplazar la naturaleza de las imágenes que genera entre lo analógico y lo digital, utilizando para ello una cámara manufacturada de tipo lomográfico que nos permite experimentar sobre la imagen con diferentes efectos para representar la “realidad” desde una percepción diferente.

En este trabajo no nos centraremos en aspectos históricos de la evolución de las máquinas fotográficas, sino que experimentaremos con un objetivo claro, conseguir escribir digitalmente en la fotografía analógica.

Más que una pieza artística, *Digitocamera* es el prototipo de un artefacto con el que el usuario puede experimentar, porque el objetivo no es solo artístico.

Motivación

Las máquinas fotográficas han estado siempre en continua evolución, desde las primeras cámaras oscuras portátiles inventadas en el siglo XVI.

En la actualidad se detecta que la fotografía analógica está en las puertas de la obsolescencia. Es difícil competir con la comodidad y economía que procura lo digital, sin necesidad de comprar película ni revelar negativos. Y esta situación es perfecta para ser aprovechada: hay vitrinas llenas de aparatos analógicos en desuso considerados obsoletos, pero son atractivas piezas de mecánica que nos permiten experimentar.

Este proyecto se interesa en otras formas posibles de captar la luz que difieren de la hiperrealista. Los ensayos entran en la línea de trabajo de la arqueología mediática. *Media Archaeology* es un campo de estudio dentro de la cultura de los media, compuesto por un conjunto de teorías y métodos que investigan la historia de los medios de comunicación a través de sus raíces alternativas, sus caminos olvidados y desatendidas funciones, usando máquinas antiguas pero que todavía son útiles.

Dentro de ese contexto, en este trabajo se describirá el proceso seguido en la experimentación con cámaras analógicas, desde las tentativas previas hasta el proyecto central.

Objetivos

Los objetivos que guían estos ensayos son: cuestionar la obsolescencia en muchos aparatos analógicos, así como la supuesta novedad de la cultura

digital. También, investigar/experimentar en la relación imagen-texto dentro de la propia fotografía y las relaciones que se producen entre ambos.

El objetivo principal del proyecto es construir la máquina híbrida, experimentar con la toma de imágenes e impresión de textos para finalmente analizar estos ensayos, tanto del enfoque híbrido analógico/digital como de la relación imagen/texto.

Metodología

Dentro de las opciones de “Práctica artística como investigación” señaladas por Graeme Sullivan¹, se sigue un proceso empírico basado en la disciplina de las artes visuales y electrónicas; contempla una parte conceptual y reflexiva enfocada a los sistemas y al medio que tiene un carácter cualitativo, pero principalmente el proceso de investigación está guiado por el análisis de los ensayos, observación de errores y vías de solución, hasta la construcción del prototipo, esta segunda parte del trabajo tiene un carácter deductivo.

Estructura

La estructura a seguir se inicia con el planteamiento del proyecto y su contextualización en un breve marco teórico y referencial (dado que se trata de una investigación aplicada). El segundo capítulo se

¹ Sullivan, G., *Art Practice as Research: Inquiry in the Visual Arts*. Thousand Oaks (CA): Sage Publications, 2010.

centra en la descripción técnica de la cámara y su proceso de construcción, señalando los problemas y soluciones que han ido surgiendo al construirla, seguido de los referentes artísticos para acabar con las conclusiones y posibles proyectos futuros.

1. PLANTEAMIENTO DEL PROYECTO

Digitocamera es una interfaz háptica diseñada como dispositivo fotográfico que mezcla lo analógico y lo digital. Su rasgo distintivo es impresionar textos sobre la superficie fotosensible, mediante los valores proyectados por una pantalla LED². Esta pantalla está formada por 8 subdisplays de 7 segmentos LED cada uno. *Digitocamera* está programada en código abierto utilizando el microcontrolador Arduino³ y controlada mediante Bluetooth con una aplicación para móviles realizada con Android Studio.

Nos hemos servido de una cámara de fotos de plástico rígido llamada Holga Panoramic como base del proyecto donde construir el prototipo, un modelo no tan conocido de las cámaras *Lomography* que más adelante explicaremos.

Más que una pieza artística, *Digitocamera* se plantea como un nuevo instrumento artístico al servicio de cualquiera que esté interesado en los lenguajes

² Está compuesto de siete segmentos que se pueden encender y apagar individualmente. Cada segmento tiene la forma de una pequeña línea. Para escribir el valor deseado se iluminan los segmentos necesarios para configurarlo. Para más información ver: [Consultado: 25-06-2015] <http://es.wikipedia.org/wiki/Visualizador_de_siete_segmentos>

³ Arduino es una plataforma de hardware libre y una comunidad de usuarios, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares. Para más información ver: <http://www.arduino.cc/> [Consultado: 25-06-2015]

fotográficos que rompen los patrones clásicos (encuadres, equilibrio...) y las características de nitidez de las imágenes digitales. La mirada de la cámara reflejará composiciones más abstractas provocadas por desenfoques, fugas de luz, superposiciones, junto la composición imagen/texto...

Las Palabras clave del proyecto son: Dispositivos e Interfaces, Cámara Analógica, Imagen pobre, Imagen-texto, Display, Tangible, Hacking, Analógico/Digital, Fotomontaje.

1.1. CONTEXTO TEÓRICO

Uno de los conceptos en los que se apoya este proyecto es en la imagen precaria o más conocida como *poor image*. No buscamos la imagen hiperrealista de la última tecnología digital, sino experimentar con la fotografía analógica añadiéndole un factor digital.

Hito Steyerl describe la imagen pobre de la siguiente manera⁴:

“A menudo desafía el patrimonio, la cultura nacional, o incluso los derechos de autor. Se transmite como un señuelo, un índice, o como un recordatorio de

⁴ Steyerl, Hito. 2009. In Defense of the Poor Image. E-Flux Journal. [http://www.e-flux.com/journal/in-defense-of-the-poor image/](http://www.e-flux.com/journal/in-defense-of-the-poor-image/)>. [Consultado 1-6-2015].

lo que fue visual. Se burla de las promesas de la tecnología digital.

Las imágenes pobres suelen mostrar lo raro, lo obvio y lo increíble. Tienden a la abstracción.”

Otro concepto a tratar es el estilo de fotografía *low-fi*. La etiqueta *Low-fi* (del inglés *low fidelity*) es originariamente una categoría musical que define el gusto por grabar con medios de baja fidelidad, con el objetivo de buscar la reproducción fiel de lo que sucede dentro del estudio, al margen del sonido artificial de las grabaciones más comerciales. Pero también es cualquier proceso que no logra alcanzar la exactitud y la transparencia acorde al tiempo tecnológico en el que se encuentra. Las técnicas más populares incluyen el uso de cámaras de juguete o cámaras estenopeicas, por sus efectos estilísticos distintivos. Se puede considerar como una reacción a la facilidad percibida de la creación de fotografías técnicamente perfectas en la era digital. Algunos enfatizan un retorno a la película, mientras que otros utilizan la tecnología digital para lograr los mismos efectos. Aunque hoy en día ya podríamos decir que la nueva fotografía *low-fi* digital, éste sería es sin duda la que ofrecen las primitivas cámaras digitales con disquete de 3½, como la Sony digital Mavica de 1.3 megapíxeles, como la de los teléfonos móviles. Ésta sería sin duda la baja fidelidad más auténtica donde el rastro del píxel es insalvable. Pero tal vez sea un poco

pronto para que se pueda crear un fetichismo de esa sociedad digital.

Pero para comprender mejor las características de nuestro proyecto, profundizaremos en la historia de *Lomography*, ya que han sido unas cámaras muy comunes que han creado un estilo.

Los inicios nos sitúan en 1982 en San Petersburgo, Rusia. El general Igor Petrowitsch Kornitzky y el camarada Michael Panfilowitsch Panfiloff deciden copiar el diseño de una mini cámara japonesa, una Cosina CX-1, con el fin de producirla en gran cantidad con el fin de distribuirla para todo el pueblo soviético. Panfiloff era el director de una factoría soviética de armamento y óptica LOMO. Debe su nombre a la Unión de Ópticos y Mecánicos de Leningrado (Leningrádskoje Optiko Mechanitschéskoje Objedinénie) que luego fabricará en masa el aparato. Los dos, uno ingeniero y el otro profesor, decidieron fabricar una cámara pequeña y simple para complacer a los soviéticos. Las características particulares de esta cámara fotográfica compacta se encuentran en el objetivo, de una distancia focal de 32 mm, lo que se supone un gran angular medio. El objetivo, diseñado por un ingeniero óptico soviético es copia del Minotar 32, montado sobre la Minox. La particularidad del objetivo consiste sobre todo en la relativa luminosidad del diafragma (f:2,8), que unida a las pequeñas dimensiones

de la lente ofrece imágenes características.

Durante los años 80 esta fue la cámara típica del pueblo soviético, de los vietnamitas, cubanos y habitantes de la República Democrática Alemana.

La primera exposición lomográfica internacional fue en 1994 en Moscú y Nueva York simultáneamente. Se podían ver miles de fotografías sobre Nueva York en Moscú y viceversa.

Fue entonces cuando se fundó la primera Embajada Lomográfica en Berlín. Esto dio lugar a las lomoembajadas, puntos de encuentro donde se realizaban actos lomográficos locales, tours lomográficos... Actualmente hay más de 70 distribuidas por 30 países.

En 1997 el fenómeno da un salto a internet y crece a nivel acelerado. Se crea lomography.com, nacen lomoembajadas, se realizan encuentros internacionales, se producen nuevos modelos (Action Sampler, la SuperSampler, la Pop9), campeonatos a nivel mundial, lomoolimpiadas, publicación de libros, tiendas "Lomography Shop" y otros eventos.

En abril de 2005 el consorcio Lomo PLC anunció que las cámaras LC-A se dejarán de producir porque resultaba muy caro fabricarlas. La cámara Lomo LC-A+ fue lanzada al mercado durante el otoño del 2006. Ésta cámara, a diferencia de la original, es fabricada en

China. Venía con la lente original fabricada por LOMO PLC hasta mediados de 2007. Actualmente la cámara y el objetivo son fabricados en China. Vienen en 7 modelos diferentes.

Hoy en día se han multiplicado los modelos y marcas de cámaras lomograficas, que al mismo tiempo han sido modificadas y experimentadas por sus usuarios.

El tercer concepto que guía el proyecto es la relación imagen-texto, y aun siendo conocedores del debate habido entre el giro lingüístico (*linguistic turn*, Rosalind Krauss⁵) y el giro de la imagen (*pictorial turn*, W.J.T. Mitchell⁶), para el planteamiento de este proyecto se ha tenido más en cuenta el enfoque de Vilem Flusser:

“Es una lucha dialéctica; a medida que la cristiandad combate el paganismo, absorbe imágenes y ella misma se paganiza; a medida que la ciencia lucha contra las ideologías, absorbe imágenes y se ideologiza. La explicación de esta dialéctica es la siguiente: aunque los textos explican las imágenes a fin de comprenderlas, las imágenes, a su vez, ilustran los textos para hacer que su significado sea imaginable. Aunque el pensamiento conceptual analiza el pensamiento mágico para deshacerse de él, el pensamiento mágico se infiltra

⁵ Krauss, Rosalind. 1996 “Welcome to the Cultural Revolution” en *October*, v.77. MIT Press.

⁶ Mitchell, W. J. T. 1994. *Picture theory*. University of Chicago Press.

en el pensamiento conceptual a fin de imaginar sus conceptos”⁷.

Así pues en este proyecto, en contra de la opinión generalizada de que la fotografía es ante todo un registro físico/químico de lo real por medio de la luz, cuyo valor reside en la imagen, planteamos esa dialéctica entre imagen/texto. Como dice Joan Fontcuberta (2004) al fin y al cabo: “la fotografía puede ser inscripción y escritura a la vez”⁸.

Por otra parte el proyecto, por sus características tanto físicas como conceptuales, se relaciona con el campo de estudio denominado *Media Archaeology*; que se puede considerar como una sub-disciplina de la arqueología y también de la cultura de los media. En una conversación entre Jussi Parikka y Garnet Hertz sobre la arqueología de los medios, Parikka afirma:

“Para nosotros en *media studies* y *media arts* es muy frecuente prestar atención a las notas a pie de página de Foucault, Kittler y los *Dead Media* de Bruce Sterling porque proporcionan el contexto para hacer el análisis de los medios arqueológicos. Los medios

⁷ Flusser, Vilém. 1990. *Hacia una filosofía de la fotografía*, México: Trillas, p. 13-14.

⁸ Fontcuberta, Joan. 2004 "La fotografía será narrativa o no será", en *El Mundo*, suplemento *El Cultural*, 36. <<http://www.elcultural.es/revista/arte/La-fotografia-sera-narrativa-o-no-sera-por-Joan-Fontcuberta/9691>> [Consultado 20-5-2015]

arqueológicos se sitúan en algún lugar entre las teorías materialistas de los media y la insistencia en el valor de lo obsoleto y olvidado de las nuevas historias culturales que han surgido desde la década de 1980”⁹.

Como corriente artística su meta es explorar un conjunto de herramientas y prácticas experimentales; no es una escuela de pensamiento o una técnica específica, es como una actitud emergente y un grupo de tácticas relacionadas con la teoría de los medios contemporáneos que se caracteriza por el deseo de re-descubrir y hacer circular de nuevo tecnologías y medios casi en desuso socialmente. Su vinculación a la arqueología de Foucault se debe principalmente por el interés en los discursos secundarios, los conocimientos locales, y un cuestionamiento a la idea de progreso. Otros autores, como por ejemplo Siegfried Zielinski, Wolfgang Ernst, Thomas Elsaesser, y Erkki Huhtamo, están sobre todo interesados en movilizar historias y dispositivos que han sido marginados por el avance capitalista de los nuevos medios. Las huellas perdidas de las tecnologías se consideran temas importantes para ser experimentados y estudiados.

Nos parece de interés también, investigar sobre las patentes registradas de cámaras que se asemejen a

⁹ Parikka, Jussi y Hertz, Garnet, 2010. “CTheory Interview. Archaeologies of Media Art” < <http://www.ctheory.net/articles.aspx?id=631> > [Consultado: 01-06-2015]

nuestro proyecto. Son archivos que hemos encontrado en el historial de patentes de Google. Hay varias patentes similares a la investigación, pero la que más se parece es la registrada en 1976 como *device for recording camera data*¹⁰ por Tomoaki Kawamura, Syohei Ohtaki, Soichi Nakamoto, Kiyoshi Takahashi. En ella podemos observar que técnicamente es el mismo procedimiento solo que ellos no incorporaron la opción de texto (solo de fecha), aparte nosotros añadimos el enviar los datos desde un teléfono móvil mediante Bluetooth, con una aplicación Android.

1.2. REFERENTES ARTÍSTICOS

Las obras que más han influido en este proyecto mantienen algunas relaciones con la esfera del *Media Archeology*. Un ejemplo de estas prácticas es el caleidoscopio de Sebastián Díaz Morales en su obra *Simulacrum 2* (2007). Tal y como se menciona en el catálogo de la exposición *Máquinas de mirar*:

“En primer lugar, cuestionan el aparato y su finalidad estrictamente funcional. Impugnan su funcionamiento o, lo que es lo mismo, su obligatoria exhibición de un efecto. Liberan al aparato de la forzosa

¹⁰ Tomoaki Kawamura, Syohei Ohtaki, Soichi Nakamoto, Kiyoshi Takahashi.” *device for recording camera*
<<https://www.google.tl/patents/US3953868?dq=camera+data+back&hl=en&sa=X&ei=bDEdVeHxEMivUeOohKAI&ved=0CC8Q6AEwAw> > [Consultado: 2-4-2015]

y exclusiva demostración de un fenómeno óptico que corrobora algo ya conocido de antemano. En su lugar, los artistas ensanchan -y deconstruyen- la rígida definición de aparato óptico e inventan aparatos fenomenológicos -abiertos- en los que la demostración y el experimento forman una unidad indisociable”¹¹

También Zachary Lieberman junto Golan Levin en la obra *The Manual Input Workstation* (2004) utilizaron un proyector de opacos para hacer música¹². Un aparato que muchos piensan que ya está obsoleto.

El referente más destacado es Julius Von Bismarck y su obra *Image Fulgurator* (2007-08)¹³. Consiste en invertir el funcionamiento de una cámara de fotografía analógica; en vez de captar luz, el “Fulgurator” la emite. Un flash colocado en la parte trasera de la cámara proyecta la imagen que hay en la película.

Otro ejemplo de artistas vinculados al *Media Archeology* son Tayio Onorato & Nico Kebs, ambos exploran las posibilidades de la fotografía y de la realidad que conlleva. Han construido cámaras con

¹¹ Schmidt, Eva. 2009. “¿La obra de arte como aparato? Acerca del interés del arte posconceptual por los antiguos medios visuales” en *Máquinas de mirar o cómo se originan las imágenes. El arte contemporáneo mira a la Colección Werner Nekes*, Colonia: DuMont Buchverlag.

¹² Levin, G., y Lieberman, Z. “Manual Input Sessions (Demo video, 2004)” [video on-line] <<https://www.youtube.com/watch?v=3paLKLZbRY4>> [Consultado: 10-5-2015]

¹³ VON BISMARCK, J., “Image Fulgurator” [página web] <<http://juliusvonbismarck.com/bank/index.php?/projects/image-fulgurator/>> [Consultado: 10-5-2015]

libros y animales¹⁴.

En cuanto a las posibilidades de la fotografía estenopeica un referente es Ilan Wolff¹⁵. Se ha especializado en crear fotografías usando la técnica de la cámara oscura, con cámaras hechas a partir de viejas cajas o latas. Actualmente usa su furgoneta o habitaciones para crear imágenes de mayor formato.

Desde 1993 Ilan Wolff imparte talleres sobre la técnica de la cámara oscura pero también sobre la utilización de emulsiones fotográficas para la creación de imágenes en diferentes superficies. También interviene en proyectos fotográficos de carácter artístico y comercial.

Dentro del movimiento *Lomography* hemos seleccionado algunas intervenciones en cámaras, intervenciones simples pero que dan resultados complejos y que además, podríamos aplicar a nuestro proyecto fácilmente. La primera intervención consiste en cortar la mitad de la tapa que cubre el objetivo para obtener fotos con efecto “espejo”¹⁶.

¹⁴ Taiyo Onorato & Nico Krebs [página web] <<http://www.tonk.ch>> [Consultado: 10-5-2015]

¹⁵ Ilan Wolff [Pág. web] <<http://www.ilanwolff.com/es/>> [Consultado: 10-5-2015]

¹⁶ Para más información consultar la fuente: <http://www.lomography.es/magazine/244527-camera-modification-tips-holga-splitzer>

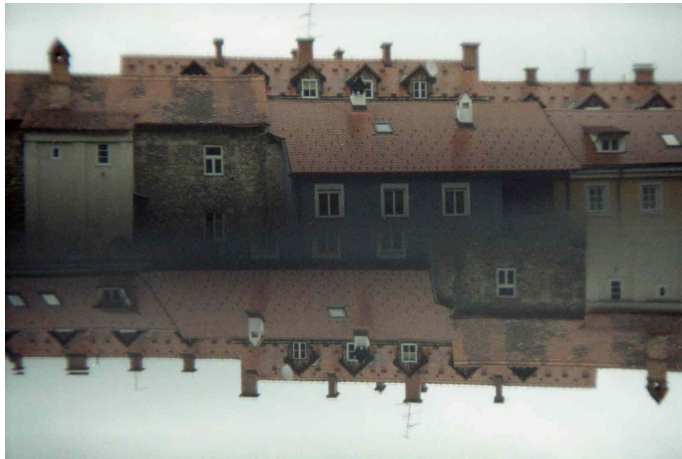


Figura 1. Autor: Lomofrue

La segunda intervención consta de un pequeño acople para poder colocar dos carretes de 35 dentro del espacio de uno de 120. Los resultados son imágenes compuestas por dos carretes¹⁷.



Figura 2. Autor: Arurin

¹⁷ Para más información consultar la fuente:
<<http://www.lomography.fr/magazine/238846-modifiez-votre-holga-120-pour-prendre-des-photos-dans-le-format-70-mm>>[Consultado: 2-7-2015]

Otro artista peruano afincado en Valencia que además ayudó en el proyecto es Pete Florentini, que modificando la cámara CANON Sure Shot 80 Tele Date conseguía disparar varias fechas en la misma fotografía.



Figura 3. Prueba realizada por Florentini Pete con CANON Sure Shot 80 Tele Date.

2. DESCRIPCIÓN TÉCNICA

En este capítulo se detallan los pasos seguidos para la construcción del prototipo, por lo que la descripción se aproxima a un cuaderno de taller, donde se señalan los problemas, ajustes y soluciones, adoptados tanto a nivel físico de construcción como de código.

Se han realizado diferentes ensayos para ir definiendo las posibilidades del proyecto: La primera idea era hackear la cámara que sirvió como fuente de creatividad, la compacta Olympus Mju 1, 35mm, F3.5 Epic Stylus, debido a que tiene la opción quartzdate (la fecha del día queda impresionada en la fotografía). Fue descartada por la dificultad de modificar el chip de la cámara para hacer que escribiese.

El primer problema que no se pudo resolver fue encontrar un display tan pequeño que esté acorde al tamaño de la película de 35mm. En la siguiente fotografía se puede observar con un recuadro rojo de donde sale proyectado la fecha para incidir en el papel fotográfico.



Figura 4 Cámara Olympus Mju 1. Por el recuadro rojo se proyecta la fecha.

2.1. DISPLAYS

Debido a este factor, la investigación dio un giro y se centró en diseñar el display acorde a las necesidades que la propia cámara requería. Para comprender cómo funciona el display se decidió empezar a programarlo desde cero, para evitar las funciones no deseadas. La composición de textos se realizaba con 2 botones; el primero permite incrementar el valor o letra (1,2,3,A,B,C..). El botón 2 guarda el valor anterior y activa el cambio de columna. Si este botón se mantiene pulsado más de un segundo, todos los valores del display quedarán guardados en la memoria de Arduino. Se puede observar un video de la practica siguiendo este link: <https://www.youtube.com/watch?v=L0xSBGxYc-w>

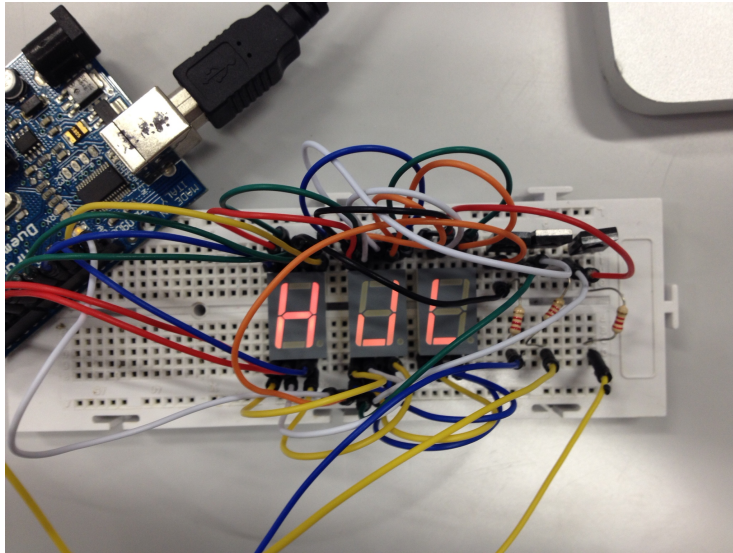


Figura 5. Primer prototipo de display. 3 displays de 7 segmentos de cátodo común

Para evitar tanto cableado, pines usados de Arduino y componentes electrónicos se decidió encargar un display JY-MCU TM1638 que viene con su propia librería¹⁸ programada por Ricardo Batista, de la que sobre todo hemos utilizado los métodos creados para manejar caracteres: `setDisplayToString(char*)`, `setDisplayToString(String)`, `clearDisplay`. Y también métodos referentes a la luminosidad de los valores como: `setupDisplay`.

Los primeros pasos fueron saber cómo escribir en el display desde Arduino y una vez conseguido, el reto fue cómo hacer encendidos tan rápidos, de hasta 8 letras, para que no se quemara el papel fotográfico. El

¹⁸ "tm1638-library - A library for interacting an arduino with a TM1638 or a TM1640" [Pág. Web] <<https://code.google.com/p/tm1638-library/>>[Consultado: 2-7-2015]

display no está pensado para programar tiempos tan pequeños, porque de normal se usa para un despertador, un anuncio publicitario en un escaparate..., y lo que nosotros buscábamos era un disparo parecido al de la propia cámara fotográfica, de un microsegundo. Finalmente jugando con delays y millis, se consiguió.

Para ver un video de la practica seguir el siguiente enlace:

<https://www.youtube.com/watch?v=GVsC-GURFE4>

El display está colocado con los números mirando hacia la película fotosensible, por ello se ha pensado realizar una aplicación para móviles con Android Studio. Con esta aplicación enviaremos los 8 valores deseados a la cámara, además evita que cada vez que se quiera cambiar el valor se tenga que abrir y cerrar la cámara, teniendo en cuenta que el carrete que introducimos en la cámara, ya no tiene el papel protector de la película fotosensible.

La aplicación consiste en 3 botones. El primer botón que está situado más arriba nos dice "Buscar dispositivo", el cual busca el dispositivo Bluetooth (BTcamera) y una vez seleccionado se comunica con él. El segundo botón es el del medio, nos dice "Escribir aquí...", cuando el usuario lo pulsa se despliega un teclado. En este teclado el usuario tendrá acceso a las

mayúsculas, minúsculas y números así como algún símbolo que más adelante se detalla. Se ha de tener en cuenta que no todas las letras se pueden componer, ya que disponemos de un display de 7 segmentos. Es por eso que las letras como: M, W, X no se pueden escribir correctamente. Se puede introducir hasta un máximo de 8 valores, que también pueden ser espacios en blanco. El tercer botón nos dice “Enviar”, y este botón envía los valores que estén previamente seleccionados al instante, en un disparo de un milisegundo.

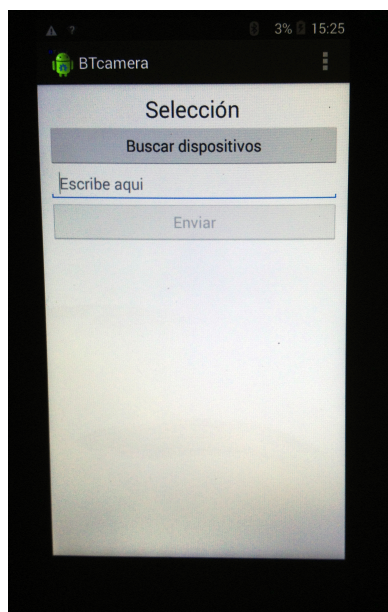


Figura 6. Imagen de la aplicación, al pulsar Escribe aquí aparece un teclado.

Para ver un video de la practica seguir el siguiente enlace:

<https://www.youtube.com/watch?v=kzD0hgjgytM>

A continuación se muestra la programación en Arduino, pero también se puede descargar la aplicación en el siguiente enlace:

<https://digitocamera.wordpress.com/code/>

Código Arduino:

```
#include <TM1638.h> //LIBRERIA DEL DISPLAY TM1638
#include <SoftwareSerial.h>
#define RxD 10
#define TxD 11
#define LED 9

SoftwareSerial BTSerial(RxD, TxD);
TM1638 module(2, 3, 4);
boolean stringComplete = false;
byte pinEstado = 0;
boolean blinky = false;
boolean dim = false;
String command = "";

boolean salir = false;
int cont = 0;

void setup(){
  pinMode(LED, OUTPUT);
  // Estado inicial
  digitalWrite(LED, LOW);

  BTSerial.begin(9600);
  BTSerial.flush();
  delay(500);
  // Configuramos el puerto serie de Arduino para Debug
  Serial.begin(9600);
  Serial.println("Ready");
```

```

}

void loop(){
    // Esperamos a recibir datos
    while (BTSerial.available()) {
        // obtener el nuevo byte:
        char inChar = (char)BTSerial.read();
        Serial.println(inChar);
        // añadirlo a inputString:
        command += inChar;
        //si el carácter de entrada es una línea nueva,
        // establecer un indicador
        // por lo que el bucle principal puede hacer algo:
        if (inChar == '\n') {
            stringComplete = true;
        }
    }
    if (stringComplete){
        Serial.println(command);
        module.setupDisplay(true, 0);
        module.setDisplayToString(command);
        delay(5); //funciones del display
        module.clearDisplay();
        BTSerial.flush();
        command="";
        stringComplete=false;
    } // Fin del tratamiento del comando
} // Fin Loop();

```

Para realizar la programación Android Studio se escribió el siguiente código que se divide en 3 bloques.

Bloque 1: STRINGS

```

<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">BTopen</string>
    <string name="Eleccion">Selección</string>
    <string name="btn1label">Buscar dispositivos</string>

```

```

    <string name="btn5label">Enviar</string>
    <string name="action_settings">Settings</string>
    <string name="btnbuscarmas">Buscar otros dispositivos
Bluetooth</string>
    <string name="NoBTSupport">Bluetooth no soportado.
Saliendo de la aplicación</string>
    <string name="Seleccion">Elija un dispositivo de la
lista</string>
    <string name="Buscando">Buscando Dispositivos
Bluetooth</string>
    <string name="title_activity_list">Busqueda
dispositivos</string>
    <string name="textoHint">Escribe aqui</string>
</resources>

```

Bloque 2: XML

```

<LinearLayout xmlns:android =
"http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="10dp"
tools:context="dabarvi.BTopen.MainActivity" >

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="@string/Eleccion"
    android:textSize="25sp" />

<Button
    android:id="@+id/btn1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="@string/btn1label"
    android:onClick="OnClick"/>

<EditText

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/texto"
        android:layout_gravity="center_horizontal"
        android:lines="1"
        android:linksClickable="false"
        android:maxLength="8"
        android:maxLines="1"
        android:singleLine="true"
        android:textAlignment="center"
        android:textSize="18dp"
        android:hint="@string/textoHint" />

<Button
    android:id="@+id/btn5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:onClick="OnClick"
    android:text="@string/btn5label" />

</LinearLayout>

```

Bloque 3: Main Activity_java

```

package dabarvi.BTopen;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.UUID;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;

```

```

import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {
    private static final String TAG = "PrincipalActivity";
    private static final int ListCode = 2;
    private static final int ENABLE_BLUETOOTH = 1;

    private BluetoothAdapter mBluetoothAdapter;
    private BluetoothDevice mBluetoothDevice;

    //Boton 1 btn1;

    private Button btn5;
    private EditText texto1;

    private boolean connected;

    // Lista de dispositivos Bluetooth
    private ArrayList<String> mArray = new
ArrayList<String>(120);
    private BluetoothSocket clientSocket;

    InputStream mInputStream;
    OutputStream mOutputStream;

    @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //btn1 = (Button) findViewById(R.id.btn1);

        btn5 = (Button) findViewById(R.id.btn5);
        texto1 = (EditText) findViewById(R.id.texto);

        if (savedInstanceState != null) {
            Bundle b = new Bundle();

```

```

b = savedInstanceState.getBundle("datos");
savedInstanceState = null;
if (b != null) {
    mBluetoothDevice = b.getParcelable("Device");
    connected = b.getBoolean("Connected");
    if (connected){
        resetConnection();

        if(connectRemoteDevice(mBluetoothDevice)){
            connected=true;
            btn5.setEnabled(true);
        }
        else {
            btn5.setEnabled(false);
            mBluetoothDevice = null;
        }

        connectRemoteDevice(mBluetoothDevice);
    }
}
else{
    btn5.setEnabled(false);
}

mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

if(mBluetoothAdapter == null){

    Toast.makeText(getApplicationContext(),
"@Strings/NoBTSupport", Toast.LENGTH_LONG).show();
    Log.d(TAG, "Not support Bluetooth");
    finish();
}

private void resetConnection() {
    if (mInputStream != null){
        try {mInputStream.close();} catch (Exception e) {}
        mInputStream = null;
    }
}

```



```

    }
    if (mOutputStream != null){
        try {mOutputStream.close();} catch (Exception e){}
    }
    if (clientSocket != null) {
        try {clientSocket.close();} catch (Exception e) {}
    }
}

@Override
protected void onResume() {
    super.onResume();
}

public void OnClick(View v){
    String mComando = new String();
    boolean esComando = false;
    String mString = new String();
    String blancos = "          "; //8 espacios en
blanco
    switch (v.getId()){
    case R.id.btn1:
        esComando=false;
        Log.d(TAG, "Dispositivos Bluetooth");

        if(!BluetoothAdapter.isEnabled()){
            Log.d(TAG,"Bluetooth apagado: Pedimos permiso para
            encenderlo.");
            startActivityForResult(new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE),
ENABLE_BLUETOOTH);
            // onActivityResult activa el Bluetooth
si el usuario asi lo decide
        }else{
            Log.d(TAG, "Bluetooth encendido.");
        }

        // Limpiamos la lista de dispositivos
detectados.
        mArray.clear();
        //btDeviceList.clear();

```

```

Intent i = new Intent(MainActivity.this,ListaActivity.class);

i.putStringArrayListExtra("dabarvi.exampleintent.Devices",
mArray);

        startActivityForResult(i, ListCode);
        break;

    case R.id.btn2:
        // Conmutar
        esComando=true;
        mComando = "C";
        break;

    case R.id.btn3:
        // Atenuar (Dim)
        esComando=true;
        mComando = "D";
        break;
    case R.id.btn4:
        // Parpadeo (Blink)
        esComando=true;
        mComando="B";
        break;
    case R.id.btn5:
        //Enviar
        esComando=true;
        //mComando="B";
        mString = texto1.getText().toString();

        if(mString.length()<8){
            mString.concat(blancos);

        }
        mString.substring(0,7);
        break;

    default:
        esComando=false;
        mComando="";
        break;
}
if (esComando) {

```

```

        mOutputStream = null;
        try {
            if (clientSocket.isConnected()){
                mOutputStream =
clientSocket.getOutputStream();
                for(int i = 0; i<7; i++){
                    //mComando =
Character.toString(mString.charAt(i));
                    mComando = mString.substring(i, i+1);
mOutputStream.write(mComando.getBytes());
                }
                mComando = "\n";
                mOutputStream.write(mComando.getBytes());
            }else{

Toast.makeText(getApplicationContext(), "Not
connected", Toast.LENGTH_LONG).show();
                }
            } catch (IOException e) {
                Log.d(TAG,e.getMessage());

                btn5.setEnabled(false);
            }
        }
    }

    @Override
    public void onActivityResult (int requestCode, int
resultCode, Intent data){
        if (resultCode == RESULT_OK && requestCode ==
ListCode) {
            if (data.hasExtra("Seleccion")) {
                Log.d(TAG, "Posicion " +
data.getIntExtra("Seleccion", 0));

                // Habilitar para depuración
                //Toast.makeText(this,
"Posicion " + data.getIntExtra("Seleccion", 0),
Toast.LENGTH_SHORT).show();
            }

            if (data.hasExtra("SelectedDevice")){
                Bundle b = data.getExtras();
                mBluetoothDevice =
(BluetoothDevice) b.getParcelable("SelectedDevice");
            }
        }
    }
}

```

```

        Log.d(TAG, "Dispositivo
seleccionado: " + mBluetoothDevice.getName());
        //Intentamos conectar con el dispositivo remoto

        if(connectRemoteDevice(mBluetoothDevice)){
            connected=true;

        }
    }
}

if (requestCode == ENABLE_BLUETOOTH){
    if (resultCode == RESULT_OK){
        Log.d(TAG, "Bluetooth: el
usuario acepta encenderlo");

    }else{
        Log.d(TAG, "Bluetooth: el
usuario NO acepta encenderlo");
        finish();
    }
}

@Override
protected void onPause() {
    super.onPause();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflar el menú; esto agrega elementos a la
barra de acción si está presente.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {

    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
}

```

```

        return super.onOptionsItemSelected(item);
    }

    private boolean connectRemoteDevice(BluetoothDevice
device){
        Log.d(TAG, "Conectando a " +
device.getName());
        boolean connect = false;
        try {
            // Conexion socket cliente.
            String mmUUID = "00001101-0000-1000-
8000-00805F9B34FB";
            this.clientSocket =
device.createRfcommSocketToServiceRecord(UUID.fromString(mmUU
ID));
            clientSocket.connect();
            connect = true;
        } catch (Exception e) {
            Log.d(TAG,e.getMessage());
            connect = false;
        }
        return connect;
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {

        Bundle b= new Bundle();
        b.putParcelable("Device", mBluetoothDevice);

        b.putBoolean("Connected", connected);
        outState.putBundle("datos", b);
        super.onSaveInstanceState(outState);
    }
}

```

La aplicación Android nos da también la posibilidad de elegir cuándo queremos mandar el texto (no está sincronizado al de la fotografía), se decidió así porque pensábamos que esta posibilidad nos ampliaba

más el juego de experimentación del usuario, ya que puedes pensar primero el texto, mandarlo y luego hacer la foto o bien hacer la foto, pensar el texto y luego mandarlo.

Teóricamente no es posible que las letras se activen de forma simultánea, pero por medio de la multiplexación¹⁹ se encienden tan rápido, una tras otra, que técnicamente se mantienen encendidas en el microsegundo del disparo fotográfico.

2.2 PROCEDIMIENTO DE CONSTRUCCIÓN

Primeramente se pensó en imprimir la cámara en 3d siguiendo una referencia²⁰ a la que se debía modificar la tapa trasera, pero la impresora 3d a la cual se tenía acceso en ese momento no imprimía con la precisión que requiere este proyecto, y además se mantenía el mismo problema de tamaño, por lo que se pensó en construir una cámara más grande²¹. Esta opción es viable pero a la hora de usar la película

¹⁹ La técnica de multiplexar varios displays de 7 segmentos se basa en el defecto que tenemos los humanos en los ojos (persistencia de la retina) en donde nos da la sensación de ver una imagen cuando esta ya ha desaparecido. Para mas información ver la fuente consultada: <<http://www.microcontroladorespic.com/tutoriales/Multiplexado/displays.html>> [Consultado 25-06-2015]

²⁰ Cámara hecha con impresora 3D: <<http://www.instructables.com/id/3D-Printed-Camera-OpenReflex/?ALLSTEPS>> [25-06-2015]

²¹ Página web de la cámara Holga, [Consultado 25-06-2015]
<http://www.lomography.es/magazine/news/2011/02/22/gil-e-a-holga-gigante>

fotosensible todo el proceso se encarecería debido al tamaño del papel. Por ello se llega a la conclusión de usar película fotosensible de 120mm. Hay varias marcas en el mercado distribuidas online por Lomography²², lo que facilita su compra con precios relativamente razonables.

De todos los modelos de Lomography se ha optado por utilizar la cámara analógica Holga Panoramic²³, debido a que su película fotográfica es de 120 mm, con un espacio de impresión más grande que el habitual (35mm). Esta mayor superficie del negativo se necesita porque el display mide 0,9 cm x 5,2 cm x 2,0 cm., es cierto que ópticamente se puede reducir su impresión, pero complicaría mucho la distribución de elementos en el interior de la cámara. A parte su rasgo característico es que está hecha con plástico, por lo que para hacer cualquier agujero es más fácil y controlable con la Dremel. La primera prueba realizada para ver si funcionaba, se realizó impresionando primero en la película fotosensible un disparo de 1 microsegundo y

²² Lomography es un Magazine, una tienda y una comunidad dedicada a la fotografía analógica. Para más información ver la fuente consultada: <<http://www.lomography.es/about>> [Consultado 25-06-2015]

²³ Holga es una cámara fotográfica con un formato medio 120 mm, hecha en China, conocida por su baja fidelidad y su estética plástica. La cámara Holga nació bajo la idea de un costo accesible siendo ampliamente producida por su sencillez de ensamblaje y vendida en muchas partes del mundo por su bajo costo. Para mas información ver la fuente consultada: <http://www.lomography.es/magazine/180528-holga-120-pan-the-lightweight-panoramic-miracle>>[25-06-2015]

después colocar esta película dentro de la cámara y realizar una fotografía en el mismo espacio que la impresión del texto. Este fue el resultado:



Figura 6. *Rude 1989*. Primera prueba

Viendo el resultado nos dimos cuenta de que el proyecto era viable, siempre y cuando se repitiesen las mismas condiciones de la primera prueba.

Un detalle que no podríamos dejar pasar es que *Digitocamera* fuese portátil. Por eso todo está alimentado con una pila de 9v. Se decidió esta alimentación porque sobrepasaba un poco lo que se necesita, pero así la batería será más duradera. A continuación se muestra un esquema del circuito con los componentes utilizados visualmente:

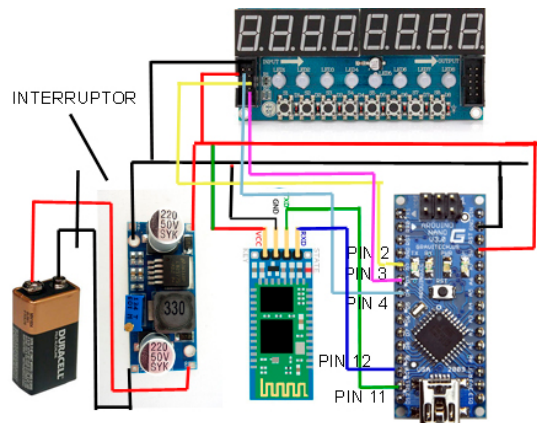


Figura 7. De izquierda a derecha: Pila de 9v, RioRand LM2596, Bluetooth, Arduino Nano (todos estos componentes van soldados a una placa encima del display TM1638)

Gracias a un regulador de voltaje RioRand LM2596 conectado a la pila, podemos ajustar la tensión con un potenciómetro incorporado para que no supere los 5v y así no quemar el circuito. Desde aquí soldamos a una placa dos carriles, uno positivo y otro negativo, donde alimentaremos el bluetooth, Arduino y el display.

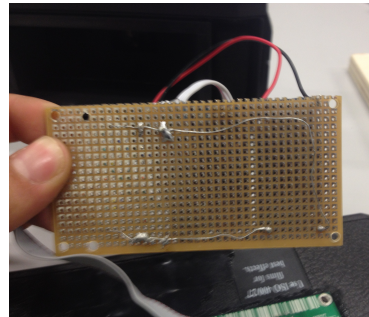


Figura 8. Placa del circuito vista por detrás

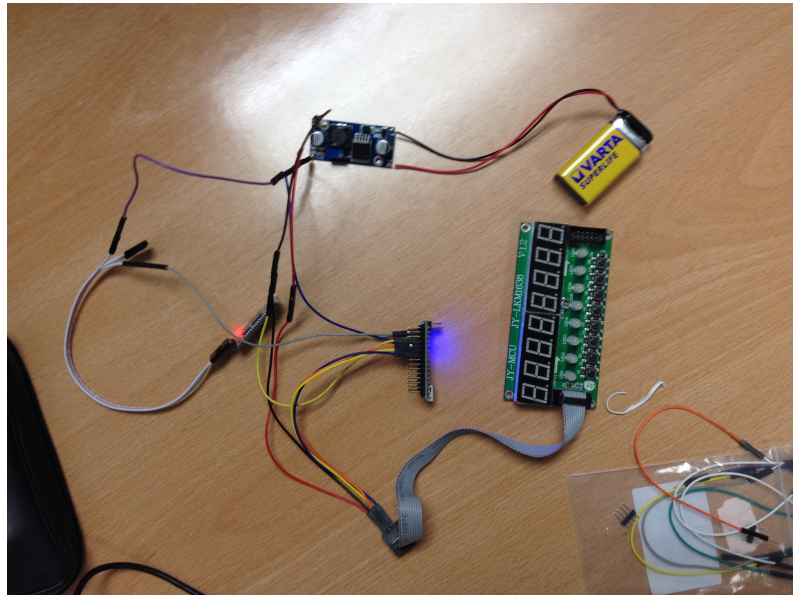


Figura 9. Circuito sin soldar

En la tapa trasera de la cámara Holga hay que insertar el display, nosotros lo hicimos perforando la tapa con el disco más fino de forma muy ajustada. La primera prueba con el display incrustado fue respetando el perímetro del chasis de la cámara para que la tapa conservara su totalidad, como se observa en la siguientes fotografías.



Figura 10. Perforando la tapa trasera con la Dremel.

Tras perforar, se colocó el display y para que no entrara nada de luz se selló con silicona negra para acuarios.

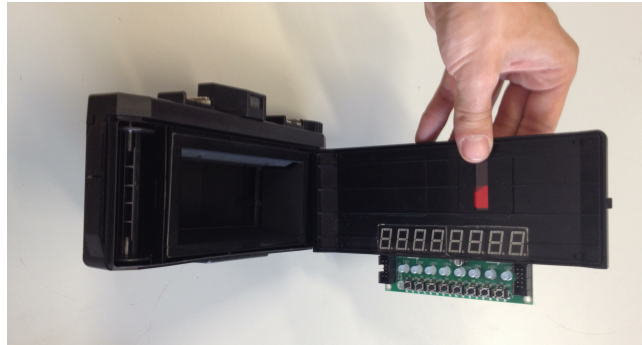


Figura 11. Primer prototipo con el display incrustado.

También se cubrió el recuadro rojo que servía para ver la parte posterior de la película fotosensible, ya que cada vez que metemos un carrete en *digitocamera* primero se tiene que retirar el papel protector trasero del carrete de 120mm, el mismo que se quita a la hora de revelar, sino lo hacemos no quedarán los valores impresos. Es por esto, que no tenemos ninguna referencia visual de cómo van pasando las fotos, ya que la ventanita, que de normal sirve para verlo, la hemos tapado con silicona negra. La solución más simple fue guiarnos por el sonido. Cuando vamos pasando el carrete, la cámara hace un sonido de “clicks”, por lo que dentro del laboratorio y con la luz apagada, contamos 70 “clicks” para una fotografía, 50 para que pase el espacio

del display por completo.

Sabiendo estos dos detalles, tenemos más posibilidades de experimentación. Por ejemplo podemos poner no solo 8 valores en una foto, sino que podríamos poner texto de principio a fin del carrete.



Figura 12. Primer prototipo. Observamos que se respetó el perímetro de la tapa trasera. Más tarde tuvimos que quitar el perímetro e imprimirlo en 3d.

La primera prueba salió mal porque las letras salían al revés y la base de los valores del display se cortaba haciendo ilegible las letras.



Figura 13. Segunda prueba errónea.

Fue necesario volver a sacar el display para subirlo 3 mm más, cortando el chasis de la tapa trasera por completo, asegurándonos que entraban todos los segmentos. Entonces se imprimió una pieza en la impresora 3D para simular el chasis de la cámara extraído para que no entrara luz del exterior.

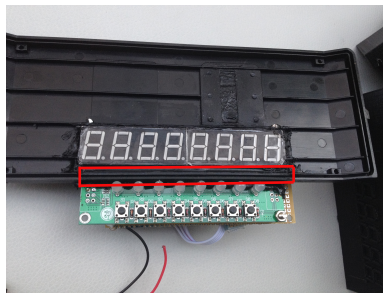


Figura 14. Pieza impresa en 3d con ranura para que acople en la cámara y no entre nada de luz.

Pero aquí surgió otro imprevisto: los pines de entrada y salida del display molestaban para acoplar bien ambas partes. La solución mas rápida fue desoldar los pines de la placa para colocarlos por la parte trasera.

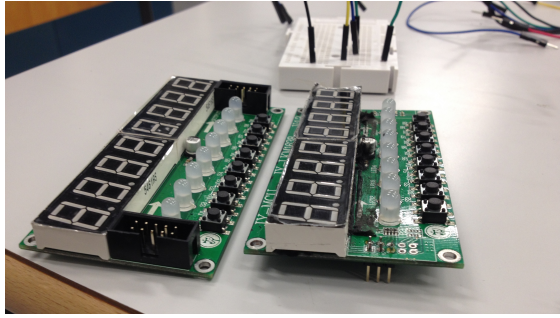


Figura 15. Display normal (izquierda). Display modificado (derecha)

Al mismo tiempo nos dimos cuenta de que las letras no estaban en la dirección correcta y eran ilegibles debido a los principios de inversión de la fotografía, los cuales hacen que la fotografía salga al revés y boca abajo respecto a su orientación física, por lo que se tuvo que reprogramar la librería TM16XXFonts del TM1638. Se barajó la posibilidad de hacer una prueba con el display colocado en la parte superior para ver en qué dirección salían, el problema es que así el display quedaba menos acoplado al papel fotográfico porque en las siguientes pruebas realizadas, la luz del display era demasiado intensa, y también filtraba luz del exterior por haber quitado la silicona negra.



Figura 16. Tercera prueba errónea .

Pese al fallo, se podía leer que no estaba en la dirección correcta y finalmente se colocaría el display en la parte inferior, quedando las letras en la parte superior de la fotografía, reprogramando otra vez la librería con las letras hacia abajo y en efecto espejo.

Para reprogramar la librería, se debe abrir el archivo TM16XXFonts para modificarlo acorde a lo que necesitamos, a continuación se muestra cómo es un abecedario en la dirección correcta, aunque también se puede descargar la librería del TM1638 con TM16XXFonts reprogramada en el enlace:

<https://digitocamera.wordpress.com/code/>

```

-- 0 --
|      |
5      1
-- 6 --
4      2
|      |
-- 3 -- .7

*/

#ifndef TM16XXFonts_h
#define TM16XXFonts_h

// definition for standard hexadecimal numbers
const byte NUMBER_FONT[] = {
    0b00111111, // 0
    0b0000110, // 1
    0b01101101, // 2-
    0b01001111, // 3
    0b01010110, // 4-
    0b01011011, // 5-
    0b01111100, // 6-
    0b00001110, // 7-
    0b01111111, // 8
    0b01011111, // 9-
    0b01111110, // A-
    0b01111100, // B
    0b00111001, // C
    0b01011110, // D
    0b01111001, // E
    0b01111000 // F-
};

const byte MINUS = 0b01000000;

// definition for error
const byte ERROR_DATA[] = {
    0b01111001, // E
    0b01100000, // r-
    0b01100000, // r-
    0b01100011, // o-
    0b01100000, // r

```



```

0,
0,
0
};

// definition for the displayable ASCII chars
const byte FONT_DEFAULT[] = {
0b00000000, // (32) <space>
0b10000110, // (33) !
0b00100010, // (34) "
0b01111110, // (35) #
0b01101101, // (36) $
0b00000000, // (37) %
0b00000000, // (38) &
0b00000010, // (39) '
0b00110000, // (40) (
0b00000110, // (41) )
0b01100011, // (42) *
0b00000000, // (43) +
0b00000100, // (44) ,
0b01000000, // (45) -
0b10000000, // (46) .
0b01010010, // (47) /
0b00111111, // (48) 0
0b00000110, // (49) 1
0b01101101, // (50) 2-
0b01001111, // (51) 3
0b01010110, // (52) 4-
0b01011011, // (53) 5-
0b01111100, // (54) 6-
0b00001110, // (55) 7-
0b01111111, // (56) 8
0b01011111, // (57) 9-
0b00000000, // (58) :
0b00000000, // (59) ;
0b00000000, // (60) <
0b01001000, // (61) =
0b00000000, // (62) >
0b01010011, // (63) ?
0b01011111, // (64) @
0b01111110, // (65) A -
0b01111111, // (66) B

```

```

0b00111001, // (67) C
0b00111111, // (68) D
0b01111001, // (69) E
0b01111000, // (70) F -
0b00111011, // (71) G-
0b01110110, // (72) H
0b00000110, // (73) I
0b00101111, // (74) J-
0b01110110, // (75) K-
0b00110001, // (76) L-
0b00010101, // (77) M
0b00111110, // (78) N-
0b00111111, // (79) O
0b01111100, // (80) P-
0b01100111, // (81) Q
0b00111000, // (82) R-
0b01011011, // (83) S-
0b01110001, // (84) T-
0b00110111, // (85) U-
0b00101010, // (86) V
0b00011101, // (87) W
0b01110110, // (88) X
0b01000111, // (89) Y-
0b01101101, // (90) Z-
0b00111001, // (91) [
0b01100100, // (92) \ (this can't be the last char on a
line, even in comment or it'll concat)
0b00001111, // (93) ]
0b00000000, // (94) ^
0b00001000, // (95) _
0b00100000, // (96) `
0b01111011, // (97) a-
0b01111100, // (98) b
0b01011000, // (99) c
0b01011110, // (100) d
0b01101111, // (101) e-
0b01111000, // (102) f-
0b01111101, // (103) g-
0b01110010, // (104) h-
0b00000010, // (105) i-
0b00100111, // (106) j-
0b01110101, // (107) k

```

```

0b00110000, // (108)l
0b01010101, // (109)m
0b01100010, // (110)n-
0b01100011, // (111)o-
0b01111100, // (112)p-
0b01100111, // (113)q
0b01100000, // (114)r-
0b01011011, // (115)s
0b01110001, // (116)t-
0b01010100, // (117)u-
0b00101010, // (118)v
0b00011101, // (119)w
0b01101110, // (120)x
0b01000111, // (121)y-
0b01101101, // (122)z-
0b01000110, // (123){
0b00000110, // (124)|
0b01110000, // (125)}
0b00000001, // (126)~
};
#endif

```

A continuación vemos la librería modificada, para apreciar el cambio, en este caso las letras saldrán del revés y boca abajo:

```

#ifndef TM16XXFonts_h
#define TM16XXFonts_h

// definition for standard hexadecimal numbers
const byte NUMBER_FONT[] = {
    0b00111111, // 0
    0b00000110, // 1
    0b01101101, // 2
    0b01001111, // 3
    0b01010110, // 4
    0b01011011, // 5
    0b01111100, // 6
    0b00001110, // 7
    0b01111111, // 8
    0b01011111, // 9
    0b01111110, // A
    0b01111100, // B
    0b00111001, // C
    0b01011110, // D

```

```

    0b01111001, // E
    0b01111000 // F
};

const byte MINUS = 0b01000000;

// definition for error
const byte ERROR_DATA[] = {
    0b01111001, // E
    0b01100000, // r
    0b01100000, // r
    0b01100011, // o
    0b01100000, // r
    0,
    0,
    0
};

// definition for the displayable ASCII chars
const byte FONT_DEFAULT[] = {
    0b00000000, // (32) <space>
    0b10000110, // (33) !
    0b00100010, // (34) "
    0b01111110, // (35) #
    0b01101101, // (36) $
    0b00000000, // (37) %
    0b00000000, // (38) &
    0b00000010, // (39) '
    0b00110000, // (40) (
    0b00000110, // (41) )
    0b01100011, // (42) *
    0b00000000, // (43) +
    0b00000100, // (44) ,
    0b01000000, // (45) -
    0b10000000, // (46) .
    0b01010010, // (47) /
    0b00111111, // (48) 0
    0b00000110, // (49) 1
    0b01101101, // (50) 2
    0b01001111, // (51) 3
    0b01010110, // (52) 4
    0b01011011, // (53) 5
    0b01111100, // (54) 6
    0b00001110, // (55) 7
    0b01111111, // (56) 8
    0b01011111, // (57) 9
    0b00000000, // (58) :
    0b00000000, // (59) ;
    0b00000000, // (60) <
    0b01001000, // (61) =
    0b00000000, // (62) >
    0b01010011, // (63) ?
    0b01011111, // (64) @
    0b01111110, // (65) A

```

```

0b01111111, // (66)    B
0b00111001, // (67)    C
0b00111111, // (68)    D
0b01111001, // (69)    E
0b01111000, // (70)    F
0b00111011, // (71)    G
0b01110110, // (72)    H
0b00000110, // (73)    I
0b00101111, // (74)    J
0b01110110, // (75)    K
0b00110001, // (76)    L
0b00010101, // (77)    M
0b00111110, // (78)    N
0b00111111, // (79)    O
0b01111100, // (80)    P
0b01100111, // (81)    Q
0b00111000, // (82)    R
0b01011011, // (83)    S
0b01110001, // (84)    T
0b00110111, // (85)    U
0b00101010, // (86)    V
0b00011101, // (87)    W
0b01110110, // (88)    X
0b01000111, // (89)    Y
0b01101101, // (90)    Z
0b00111001, // (91)    [
0b01100100, // (92)    \ (this can't be the last char
on a line, even in comment or it'll concat)
0b00001111, // (93)    ]
0b00000000, // (94)    ^
0b00001000, // (95)    _
0b00100000, // (96)    `
0b01111011, // (97)    a
0b01111100, // (98)    b
0b01011000, // (99)    c
0b01011110, // (100)d
0b01101111, // (101)e
0b01111000, // (102)f
0b01111101, // (103)g
0b01110010, // (104)h
0b00000010, // (105)i
0b00100111, // (106)j
0b01110101, // (107)k
0b00110000, // (108)l
0b01010101, // (109)m
0b01100010, // (110)n
0b01100011, // (111)o
0b01111100, // (112)p
0b01100111, // (113)q
0b01100000, // (114)r
0b01011011, // (115)s
0b01110001, // (116)t
0b01010100, // (117)u

```

```

0b00101010, // (118)v
0b00011101, // (119)w
0b01110110, // (120)x
0b01000111, // (121)y
0b01101101, // (122)z
0b01000110, // (123){
0b00000110, // (124}|
0b01110000, // (125)}
0b00000001, // (126)~
};
#endif

```

Una vez modificamos la librería hicimos nuevas pruebas y estos fueron los resultados:



Figura 17. Cuarta prueba. "Autoretrato rUdE HJL".



Figura 18. Quinta prueba. "HORTA ES"



Figura 19. Sexta prueba. "FUtUr"



Figura 20. Séptima prueba. Superposición de 2 fotografías. "Soy rUdE"



Figura 21. Octava prueba. "AtArDECE"



Figura 22. Novena prueba. "SAUri"

Para que todo quedase en el mínimo espacio posible se decidió separar la placa donde está soldado el circuito del display con unos separadores metálicos. Organizamos los componentes de la siguiente manera:

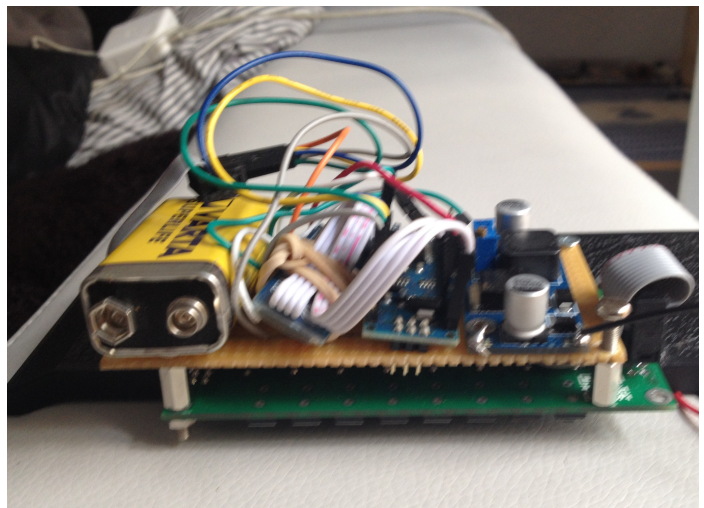
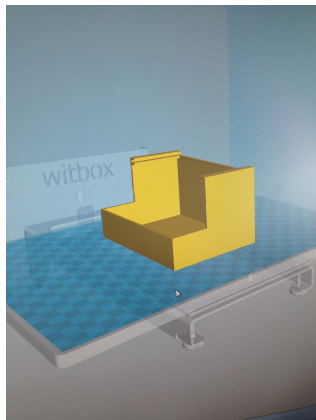


Figura 23. De izquierda a derecha: Pila de 9V, Bluetooth, Arduino Nano, Transformador.

Por último, para corregir las entradas de luz que todavía se observan, diseñamos una caja con Blender, la exportamos en un archivo .obj y la imprimimos en 3d acorde a las medidas para que encajase en la cámara. Añadiéndole un interruptor para facilitar el encendido y la simulación del display TM1638 en relieve, para que el usuario a la hora de componer las letras, le sea más fácil e intuitivo.

Figura 24. Caja impresa en 3D

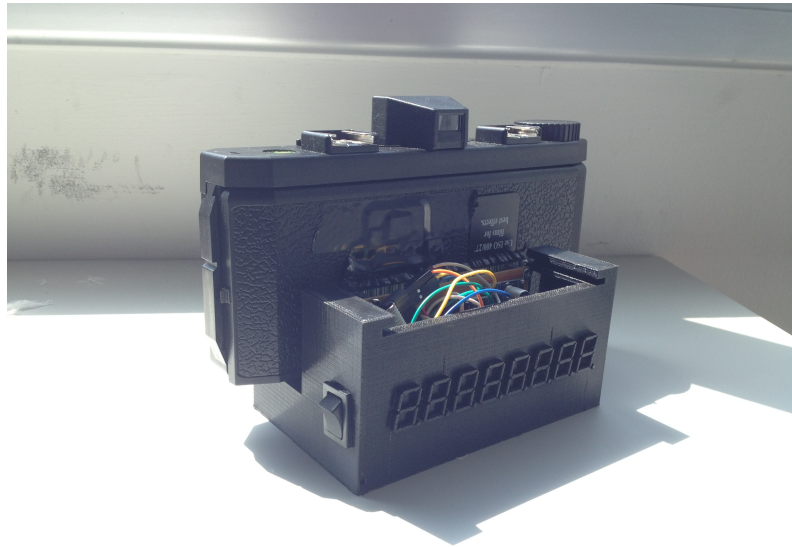


Figura 25. Prototipo a falta colocar la tapa que cubre.

4.CONCLUSION Y TRABAJO FUTURO

Como proyecto presentado al Trabajo Final de Máster, la finalidad es mostrar el funcionamiento de la cámara para valorar las cualidades estéticas de las imágenes/textos que realiza. Subrayando que más que una pieza artística, *Digitocamera* se plantea como un instrumento artístico al servicio de cualquiera que esté interesado en los lenguajes fotográficos particulares que genera.

Las conclusiones las abordaremos a raíz de los resultados obtenidos durante la experimentación. Cabe recalcar que ninguna imagen ha sido retocada con ningún software de edición de imagen.

Podemos decir con seguridad, después de todas las pruebas realizadas, que la luz del display no traspasa la película fijándola por la parte delantera (donde se fija la imagen), es decir, la fotografía no sufre modificaciones por culpa del display, ya que este está colocado por detrás. Lo que hemos corregido son algunas entradas de luz tapando el display con la caja negra impresa en 3D, ya que las entradas de luz tenían un tono verdoso procedente de la luz que traspasaba por él.

Son muchos los factores que determinan el resultado de impresión del display en el papel, por eso debemos tener en cuenta que al estar el display

colocado en la parte de abajo, luego en la fotografía revelada y positivada estará arriba. A la hora de fotografiar con un carrete de ISO400 (el más recomendable para esta cámara y más económico) por ejemplo un paisaje en el que hay mucha luz y el 1/3 superior de la fotografía (espacio donde quedarán impresos los valores deseados) será el cielo, si mandamos solo un disparo, es posible que el resultado después sea demasiado ligero como para leerlo. Según cual sea la intención buscada, se recomienda en estas ocasiones realizar dos o incluso tres disparos.

Al contrario, si vamos a fotografiar espacios con menos luz y al mismo tiempo el 1/3 superior de la fotografía tiene un fondo oscuro, es recomendable realizar solo un disparo desde la aplicación.

Con las pruebas realizadas en carrete de blanco y negro el resultado es bastante diferente, ya que solo tenemos estas dos tonalidades, se debe prever que si el fondo donde se impresionarán los valores es próximo al blanco, deberemos enviar más de tres disparos desde la aplicación.

Como podemos observar, en la primera imagen el texto es tan sutil que casi no se percibe, la segunda foto, se lee un poco mejor y está realizada con dos disparos mandados desde la aplicación Android. En la tercera imagen se realizaron 3 disparos.



Figura 26. Prueba realizada con un disparo "dANi GrAY"



Figura 27. Prueba realizada con dos disparos. "anticops"



Figura 28. Prueba realizada con tres disparos. "antitodo"

Un error que corregimos con la impresión 3D fue que nosotros mismos al no tener una referencia visual

de cómo se forman las letras, no siempre acertábamos bien, es decir, hay letras que se pueden repetir creando confusiones. En el siguiente ejemplo, se quiso escribir: “COLGADO” y como no teníamos referencia del display salió como si pusiese “COLGAOO”. Este error ahora teniendo la caja impresa en 3D, con la imitación del display TM1638 en relieve, lo podríamos haber corregido escribiendo “COLGAdO”.

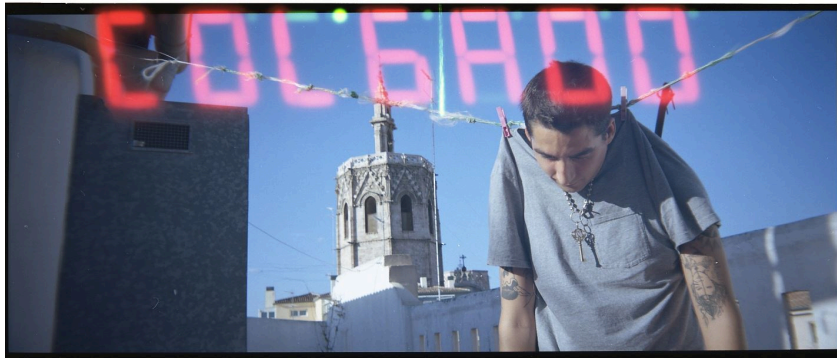


Figura 29. Prueba errónea “COLGAOO”

Otra conclusión que hemos sacado es que cuando se hacen la superposición de dos fotografías en un mismo texto, al haber tanta información, también sería aconsejable realizar dos disparos desde la aplicación Bluetooth, para que así quede con más intensidad. Por ejemplo:



Figura 30. Prueba errónea “ CHICOES3” con poca intensidad.

Los siguientes ensayos (casi como trabajo inminentemente futuro porque se realizará durante la semana anterior a la defensa del proyecto) se llevarán a cabo por la noche, porque al imprimir la pieza en 3D, añadimos una estabilidad a la cámara.

También se realizarán las pruebas guiándonos por los “clics” del carrete, con el objetivo de crear una frase, que esté colocada de principio a fin del carrete. Toda esta información será subida al blog del proyecto: <https://digitocamera.wordpress.com/>

Este blog, además de recoger el proyecto, mostrará fotografías de más pruebas y del prototipo final.

Y para finalizar, como trabajo futuro se debería dar visibilidad al proyecto. Por ejemplo en las redes sociales y si se diera el caso que gustase, crear un comunidad para realizar replicas mejoradas o con modificaciones.

BIBLIOGRAFÍA

Báztner, Nike; Nekes, Werner y Schmidt, Eva (eds.)
*Máquinas de mirar o cómo se originan las
imágenes. El arte contemporáneo mira a la
Colección Werner Nekes*, Colonia: DuMont
Buchverlag 2009

Flusser, Vilem. 1990. *Hacia una filosofía de la fotografía*,
México: Trillas.

Flusser, Vilem. 2010. «Television Image and Political
Space in the Light of the Romanian Revolution //
Lecture in Budapest, April 7, 1990» en *We shall
survive in the memory of others Vilém Flusser*.
[DVD] Centerfor Cultur and Communication
Foundation Budapest (Miklós Peternák) y Vilém
Flusser Archiv, Universität der Künste Berlin.

Fontcuberta, Joan. 2004 "La fotografía será narrativa o
no será", en *El Mundo* (Madrid), suplemento *El
Cultural*.

[http://www.elcultural.es/revista/arte/La-fotografia-
sera-narrativa-o-no-sera-por-Joan-
Fontcuberta/9691](http://www.elcultural.es/revista/arte/La-fotografia-sera-narrativa-o-no-sera-por-Joan-Fontcuberta/9691) [Consultado 20-5-2015]

Krauss, Rosalind. 1996 "Welcome to the Cultural
Revolution" en *October*, v.77. MIT Press.

- Mitchell, W. J. T. 1994. *Picture theory*. University of Chicago Press.
- Pink, Sarah. 1996. "Excursiones socio-visuales en el mundo del toreo", en María García Alonso (y otros) (eds.), *Antropología de los sentidos*. Madrid, Celeste Ediciones.
- Steyerl, Hito. "In defense of the Poor Image" en E-flux [En línea] <http://www.e-flux.com/journal/in-defense-of-the-poor-image/> [Consultado 1-6-2015]
- Sullivan, G., *Art Practice as Research: Inquiry in the Visual Arts*. Thousand Oaks (CA): Sage Publications, 2010.
- Tienda online cámaras Lomography. "cámaras Holga-Lomograpahy"
<http://shop.lomography.com/es/cameras/holga-cameras> [Consultado 25-06-2015]