



Universidad de Valladolid

E.T.S. Ingeniería Informática

Trabajo fin de grado

Grado en Ingeniería Informática

Desarrollo de una interfaz gráfica para un sistema electrónico de caracterización óptica de diodos emisores de luz

Autor:

D. Juan José Colino Barrigón

Tutor:

D. Iván Santos Tejido

Resumen

En este Trabajo de Fin de Grado (TFG) se describe el proceso de análisis, diseño y desarrollo de una interfaz gráfica para un sistema de caracterización óptica de diodos emisores de luz (LED del inglés Light-emitting diode). La interfaz ha sido implementada en Python usando la librería gráfica GTK+3 para Linux.

Palabras clave

Diseño de interfaz, Linux, LED, Caracterización Óptica, GTK+

Agradecimientos

Me gustaría agradecer a mi familia en especial a mi padre y a mi madre el apoyo que me han prestado durante mis años de estudiante, siendo una parte imprescindible en mi desarrollo personal y profesional.

También me gustaría agradecer a mi tutor D. Iván Santos Tejido por su ayuda y guía durante el desarrollo del trabajo.

Índice

Resumen	3
Palabras clave	3
Agradecimientos	5
Índice	6
Introducción	9
Motivación.....	9
Conocimientos previos.....	9
Definiciones.....	9
LED.....	11
Intefaces gráficas de usuario	13
Descripción del sistema inicial.....	16
Monocromador	17
Fotodiodo.....	18
Microcontrolador	18
PC	18
Objetivos	18
Estructura de la memoria	19
Captura de requisitos	21
Requisitos funcionales	21
Requisitos no funcionales.....	21
Análisis funcional del sistema inicial	23
Interfaz y funcionamiento inicial.....	23
Configuración inicial.....	24
Selección de ganancia	25
Realización de una medida puntual.....	26
Realización de un barrido	27
Estructura de navegación.....	32
Programa del PC.....	33
La función main()	34

La función muestraMenu().....	34
La función irPosicion().....	34
La función selectorGanancia()	34
La función barrido().....	34
Programa del micro.....	35
Protocolo de comunicación micro – PC.....	35
Diseño de la solución.....	38
Diseño de la interfaz.....	38
Pantalla de configuración inicial.....	40
Pantalla del menú principal.....	41
Pantalla de medida puntual.....	42
Pantalla de barrido	43
Diseño de clases	44
Clase monocromador	44
Clases de la interfaz gráfica	45
Implementación.....	47
Python.....	47
Matplotlib.....	47
GTK+3	47
Manual de usuario	50
Configuración inicial.....	50
Navegación entre pantallas	51
Realizar una medida puntual	53
Realizar un barrido.....	54
Manual de instalación.....	57
Conclusiones.....	59
Referencias	62
Apéndice	63
Contenido del archivo empaquetado entregable	63

Introducción

Motivación

El objetivo principal del presente Trabajo Fin de Grado es diseñar y construir una interfaz gráfica para un sistema existente de caracterización óptica de diodos emisores de luz (LED). El sistema toma como punto de partida el trabajo realizado por D. Jorge Fernández Lucas en el diseño y desarrollo del sistema inicial, y supone una mejora del mismo.

Conocimientos previos

Definiciones

LED

Del inglés Light-emitting diode, un LED es un componente optoelectrónico pasivo emisor de luz.

Longitud de onda

La longitud de onda es la distancia real que recorre una perturbación (una onda) en un determinado intervalo de tiempo. Aplicado a los LED se refiere a la longitud de onda de la luz emitida por el mismo a mayor intensidad. Un LED emite luz en varias longitudes de onda pero una es dominante.

Espectro electromagnético

Se refiere a la a la distribución energética del conjunto de las ondas electromagnéticas. Cuando se refiere a los LED se refiere a la radiación electromagnética que emite. Es una característica de cada tipo de LED.

Regiones del espectro

Cada una de las partes en las que se divide el espectro electromagnético para su estudio. También se denominan bandas. Estas incluyen los rayos gamma, rayos x, ultravioleta, luz visible, infrarrojos (IR), microondas y ondas de radio.

Región IR

Una región del espectro electromagnético. Una de las regiones en las que existen diodos comerciales que emiten en dichas longitudes de onda. Su uso se suele centrar en comunicaciones ya que la emisión en IR no es visible.

Región visible

Una región del espectro electromagnético. Así misma se divide en subregiones por colores visibles. Es una de las regiones en las que existen diodos comerciales que emiten en dichas longitudes de onda. Su uso se centra en iluminación y señalización.

Región UV

Una región del espectro electromagnético. Es una de las regiones en las que existen diodos comerciales. Su uso se centra en iluminación de sustancias fluorescentes.

Monocromador

Es un dispositivo es un dispositivo óptico que permite seleccionar y transmitir una estrecha banda de longitudes de onda a partir de una fuente emisora que produzca una amplia gama de longitudes de onda. Es un dispositivo de filtrado, recibe como entrada una emisión en varias longitudes de onda y emite como salida una selección menor de longitudes de onda.

GUI (Graphical User Interface)

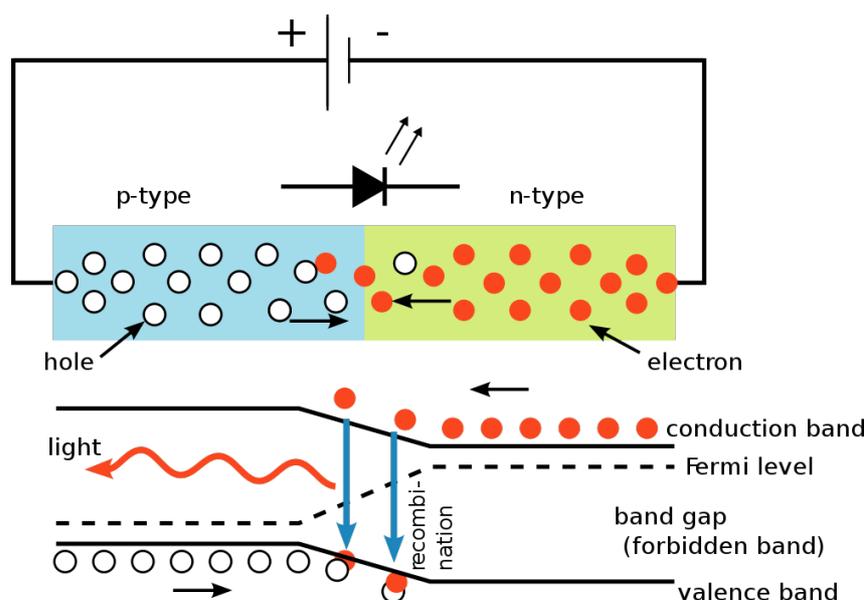
Las interfaces gráficas (GUI, del inglés graphical user interface) de usuario son un tipo de interfaz que permite a los usuarios de un sistema interactuar con él a través de iconos gráficos e indicadores visuales, en oposición a las interfaces basadas en texto.

LED

Los LED y su funcionamiento

Los LED (light-emitting diode) son componente optoelectrónicos pasivo emisor de luz. Básicamente son dispositivos semiconductores con una unión pn que emite luz cuando es activada. Cuando se les aplica una tensión suficiente a sus extremos los electrones son capaces de recombinarse con los huecos dentro del dispositivo, emitiendo luz al hacerlo.

Este efecto se denomina electroluminiscencia y el color de la luz (la longitud de banda en la que emite) depende la distancia de la banda de energía. La banda de energía es la zona intermedia de la unión pn donde los electrones no pueden encontrarse. Los electrones saltan a través de dicha banda. Según los materiales usados en la construcción de la unión pn la banda de energía varía dando lugar a diferentes tipos de LED para diferentes usos.



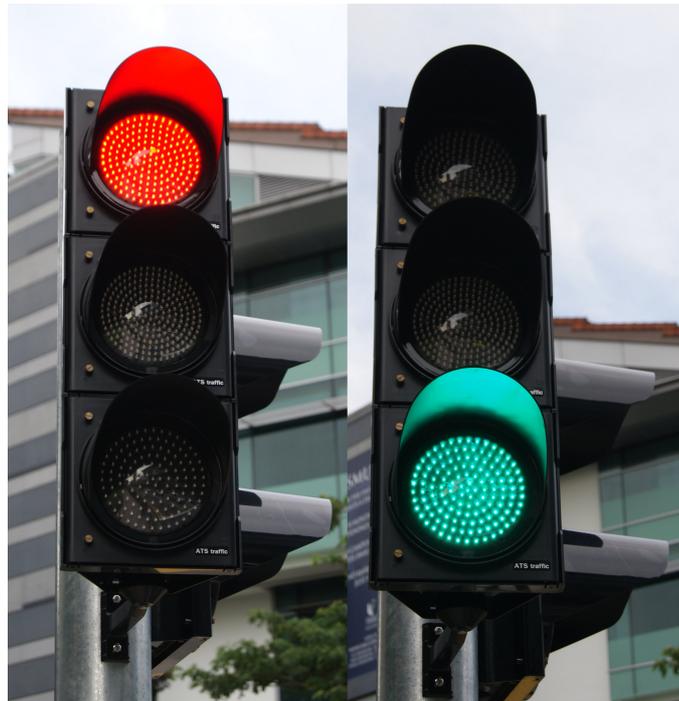
Funcionamiento de un LED con el diagrama del circuito en la parte superior y el de banda en la parte inferior.

Los materiales más comunes para los LED son el GaAs (Arseniuro de galio), el GaAsP (Fosfo- Arseniuro de galio) y el AlGaAs (Arseniuro de galio-aluminio).

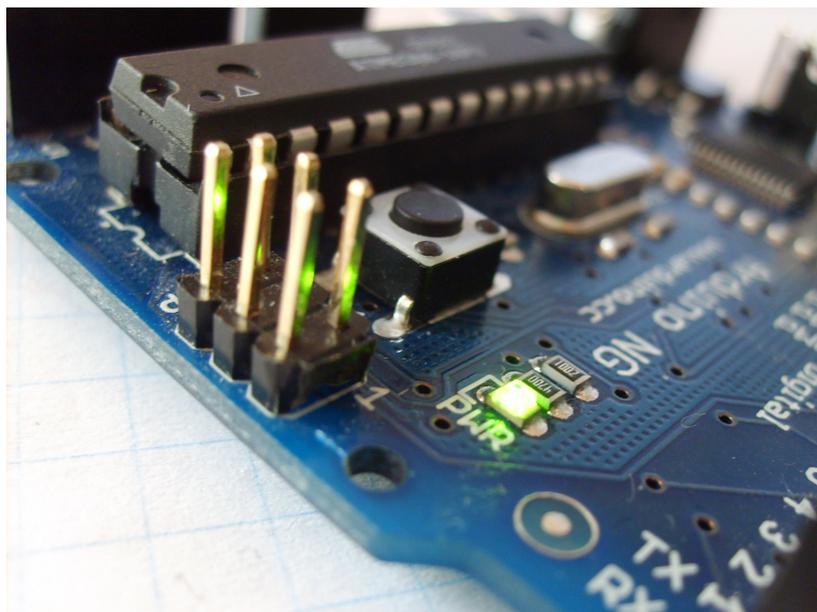
Además de sus características de funcionamiento los LED comerciales se acompañan de otros componentes aparte de los semiconductores, como son los encapsulados, terminales, componentes ópticos difusores de luz, etc...

Usos y aplicaciones

Los LED comerciales tienen muchas y variadas aplicaciones. Principalmente se usan por su capacidad de emitir luz. Esto les convierte en indicadores en sistemas electrónicos y displays, usados tanto para indicar si una radio está encendida como en señalización de tráfico.



LEDs usados en señalización de tráfico en un semáforo.



LED usado en una placa de un microcontrolador para indicar estado.

También tienen usos en iluminación, desde iluminación ambiental de hogares de hasta estadios deportivos. Sus características de bajo consumo, duración, robustez física y alta velocidad de encendido los hace muy atractivos para estas aplicaciones.



Bombilla LED.

También tradicionalmente han tenido y tienen aplicación en telecomunicaciones cuando se usan emitiendo en bandas invisibles por su capacidad de cambiar de estado (on/off switching).

También son usados como sensores de luz, no solo de luz visible sino como sensores de luz no visible. Esto tiene aplicaciones en sistemas de visión asistida como los sistemas de visión nocturna y por computador.

Pese a sus ventajas también presentan ciertos inconvenientes en su uso, siendo los principales su mayor coste de fabricación (respecto a otras tecnologías de iluminación) y dependencia de la temperatura ambiente.

Interfaces gráficas de usuario

Las interfaces gráficas (GUI, del inglés graphical user interface) de usuario son un tipo de interfaz que permite a los usuarios de un sistema interactuar con él a través de iconos gráficos e indicadores visuales, en oposición a las interfaces basadas en texto.

Las acciones realizadas en una interfaz gráfica de usuario son realizadas normalmente a través de la manipulación directa de elementos gráficos.

A continuación realizaremos un repaso de la reciente historia de las interfaces gráficas de usuario, sus fundamentos y su contraposición con las interfaces basadas en línea de comandos.

Historia de las interfaces gráficas de usuario

El primer precursor de las GUI fue desarrollado en Standform Research Institute en Menlo Park, en el estado de California. Allí desarrollaron un sistema de texto basado en hipervínculos que eran manipulados con un ratón para el sistema On-Line System (NLS). El concepto fue refinado y desarrollado por los investigadores en Xeros PARC que usaron una interfaz gráfica de usuario como interfaz principal para el Xerox Alto computer, que vio la luz en 1973. La mayoría de los GUI multipropósito actuales derivan de este sistema.

El PARC user interface introdujo los conceptos de ventana, menú, radio button y check box. Más tarde se introdujo el concepto de icono. El sistema utilizaba un ratón para complementar al teclado.

Tras esto el llegaron el Xerox 8010 Star Information System en el 81, el Apple Lisa en el 83 y el Apple Macintosh 128K en el 84. En el 85 vieron la luz el Atari ST y el Commodore Amiga.



Apple Macintosh 128K.

Apple, IBM y Microsoft se basaron en las ideas de Xerox para el desarrollo de sus productos. Las especificaciones de IBM Common User Access formaron las bases de las interfaces de usuario que aparecería en Microsoft Windows, IBM OS/2 Presentation Manager, y Unix Motif toolkit and window manager.

A pesar de las dificultades para la implantación de los GUI debido a la poca disponibilidad de software que corriera en ellos y las limitaciones en prestaciones de los computadores las campañas de marketing de Apple (con su anuncio en la SuperBowl XVIII) y Microsoft con Windows 95 ayudaron a implantar definitivamente las interfaces gráficas de usuario.

En 2007 con el iPhone y en 2010 con la introducción del iPad, Apple introduce nuevos y populariza las interfaces post-WIMP (Windows, Icons, Menus, Pointing-device) con soporte multi-táctil, que suponen un hito muy importante en el desarrollo de los dispositivos móviles actuales.

Fundamentos de las GUI

La mayoría de interfaces gráficas de usuario utilizan metáforas e ideas desarrolladas en el final de los años 70 y comienzos de los 80 en California. Conceptos como las ventanas que contienen otros elementos, los botones, hipervínculos datan de esta época. Estos conforman el **lenguaje visual** de las interfaces gráficas de usuario.

Muchos de estos elementos se toman de una **metáfora de escritorio**. Así aparecen las metáforas de documentos, carpetas, espacio de escritorio, papelera de reciclaje, entre otros.

Tradicionalmente las GUI siempre han ido acompañadas de un elemento de hardware (HID, human interaction device) que se usa como **puntero**, normalmente en la forma de ratón, aunque también hay punteros tipo lápiz + tableta.

Los GUI's ofrecen un **gestor de ventanas** que facilita las interacciones entre ventanas, aplicaciones y el sistema operativo.

GUI's vs CLI's

Las interfaces gráficas de usuario nacen en parte por las dificultades presentadas por las interfaces por línea de comandos (CLI, del inglés Command Line Interface). Para operar una CLI el operador tiene que recordar los comandos que están disponibles y su funcionamiento. Pese al uso de nombres mnemónicos y descriptivos la tarea presenta una curva de aprendizaje empinada. Además cuando se usan comandos largos es bastante fácil equivocarse lo que puede dar a grandes dolores de cabeza y problemas reales.

```

mars@marsmain ~ $ pwd
/home/mars
mars@marsmain ~ $ cd /usr/portage/app-shells/bash
mars@marsmain /usr/portage/app-shells/bash $ ls -al
total 130
drwxr-xr-x  3 portage portage 1024 Jul 25 10:06 .
drwxr-xr-x 33 portage portage 1024 Aug  7 22:39 ..
-rw-r--r--  1 root  root   35808 Jul 25 10:06 ChangeLog
-rw-r--r--  1 root  root   27802 Jul 25 10:06 Manifest
-rw-r--r--  1 portage portage 4645 Mar 23 21:37 bash-3.1_p17.ebuild
-rw-r--r--  1 portage portage 5977 Mar 23 21:37 bash-3.2_p39.ebuild
-rw-r--r--  1 portage portage 6151 Apr  5 14:37 bash-3.2_p48-r1.ebuild
-rw-r--r--  1 portage portage 5988 Mar 23 21:37 bash-3.2_p48.ebuild
-rw-r--r--  1 portage portage 5643 Apr  5 14:37 bash-4.0_p10-r1.ebuild
-rw-r--r--  1 portage portage 6230 Apr  5 14:37 bash-4.0_p10.ebuild
-rw-r--r--  1 portage portage 5648 Apr 14 05:52 bash-4.0_p17-r1.ebuild
-rw-r--r--  1 portage portage 5532 Apr  8 10:21 bash-4.0_p17.ebuild
-rw-r--r--  1 portage portage 5660 May 30 03:35 bash-4.0_p24.ebuild
-rw-r--r--  1 root  root   5660 Jul 25 09:43 bash-4.0_p28.ebuild
drwxr-xr-x  2 portage portage 2048 May 30 03:35 files
-rw-r--r--  1 portage portage 468 Feb  9 04:35 metadata.xml
mars@marsmain /usr/portage/app-shells/bash $ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
<herd>base-system</herd>
<use>
  <flag name='bashlogger'>Log ALL commands typed into bash; should ONLY be
  used in restricted environments such as honeypots</flag>
  <flag name='net'>Enable /dev/tcp/host/port redirection</flag>
  <flag name='plugins'>Add support for loading builtins at runtime via
  'enable'</flag>
</use>
</pkgmetadata>
mars@marsmain /usr/portage/app-shells/bash $ sudo /etc/init.d/bluetooth status
Password:
* status: started
mars@marsmain /usr/portage/app-shells/bash $ ping -q -c1 en.wikipedia.org
PING rr.esams.wikimedia.org (91.198.174.2) 56(84) bytes of data.

--- rr.esams.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 49.820/49.820/49.820/0.000 ms
mars@marsmain /usr/portage/app-shells/bash $ grep -i /dev/sda /etc/fstab | cut --fields=-3
/dev/sda1      /boot
/dev/sda2      none
/dev/sda3      /
mars@marsmain /usr/portage/app-shells/bash $ date
Sat Aug  8 02:42:24 MSD 2009
mars@marsmain /usr/portage/app-shells/bash $ lsmode
Module          Size  Used by
rndis_wlan      23424  0
rndis_host      8696  1 rndis_wlan
cdc_ether        5672  1 rndis_host
usbnet           18688  3 rndis_wlan,rndis_host,cdc_ether
parport_pc      38424  0
fglrx            2388128 20
parport         39648  1 parport_pc
iTCO_wdt         12272  0
i2c_i801         9380  0
mars@marsmain /usr/portage/app-shells/bash $ █

```

Interfaz CLI.

Por el contrario las GUI ofrecen una manera más compartimentada de introducir ordenes al sistema a través de widgets contenidos en las ventanas. Los menús ayudan al usuario a ver todas las opciones disponibles sin tener conocer los comandos de antemano y la integración de ayuda en el mismo software es más sencilla y eficiente.

Descripción del sistema inicial

El sistema inicial ha sido desarrollado por D. Jorge Fernández Lucas como Trabajo de Fin de Grado en el Grado en Ingeniería de Tecnologías de Telecomunicación durante el curso 2014 – 2015 con título “Diseño y fabricación de un sistema electrónico para la caracterización de diodos emisores de luz”. En la presente sección se describe dicho sistema en su estado inicial, sobre el cuál parte este TFG, para dotar de una interfaz gráfica al mismo.

El sistema cuenta con varias partes diferenciadas. El dispositivo central es un monocromador que se encarga de filtrar ciertas longitudes de ondas. El sistema cuenta también con un porta-diodos cuya función es la de alimentar y sujetar en posición LED's para su estudio a través del sistema. Un fotodiodo se encarga de medir la intensidad de luz en una cierta longitud de onda filtrada por el monocromador. El sistema está gobernado por un microcontrolador que se encarga de las operaciones a bajo nivel (componentes y sistemas electrónicos) y un ordenador personal que se encarga de las operaciones de entrada y salida.

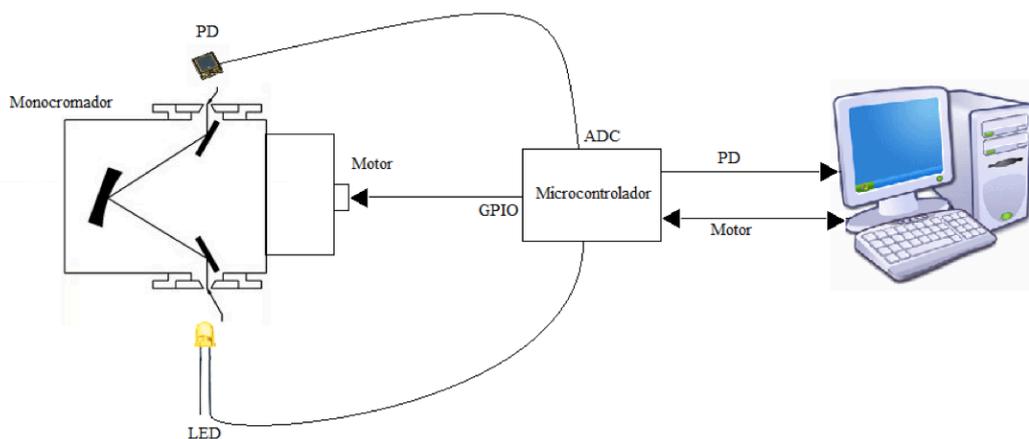


Diagrama del sistema inicial. Del la memoria del TFG de D. Jorge Fernández Lucas.

A continuación se detallan los distintos componentes del sistema. En siguientes capítulos se detallan otras partes importantes del funcionamiento del sistema.

Monocromador

Tal y como lo definimos antes un monocromador es un dispositivo es un dispositivo óptico que permite seleccionar y transmitir una estrecha banda de longitudes de onda a partir de una fuente emisora que produzca una amplia gama de longitudes de onda.

En nuestro caso el sistema cuenta con un monocromador comercial de la empresa TVC modelo MonoSpec 10 con una longitud focal de 100mm.

En ambos lados del monocromador se encuentran dos placas de circuito impreso. Una contiene el fotodiodo para realizar las medidas y la otra actúa de soporte para el LED a analizar. El monocromador también cuenta con un dial analógico que muestra la posición actual del monocromador. Este dial se mueve al mismo tiempo que lo hace el monocromador, es decir, mientras el sistema está obteniendo medidas. También cuenta con un conector para interactuar con la placa del microcontrolador.

Fotodiodo

El sistema cuenta con un fotodiodo que se encuentra en la abertura de salida del monocromador. Este dispositivo electrónico de tipo diodo se encarga de medir la intensidad de luz que incide sobre él. Se conecta con la placa del microcontrolador para a su vez ser conectado a un sistema de amplificación y posteriormente al conversor A/D del microcontrolador.

Microcontrolador

El microcontrolador del sistema se encarga de realizar el control a bajo nivel de los componentes del sistema, es decir, los componentes electrónicos como el LED, el fotodiodo ya el monocromador.

El microcontrolador utilizado es un LPC2103 basado en un núcleo ARM7 que posee un cristal de cuarzo que permite una frecuencia máxima de funcionamiento de 14.7456 MHz.

PC

El PC se encarga de realizar las operaciones de entrada y salida con el usuario. Ejecuta un programa que se encarga de leer la entrada del usuario, comunicarse con el microcontrolador para que es realice las tareas a bajo nivel y mostrar la salida al usuario una vez las operaciones y cálculos han sido completados.

El PC utiliza Linux como sistema operativo en la distribución Ubuntu en la versión 15.04 LTS.

Esta será la parte objetivo de nuestras modificaciones para cambiar la interfaz basada en línea de comandos por una interfaz gráfica de usuario.

Objetivos

El objetivo del TFG es el de ampliar los conocimientos y poner en práctica lo aprendido durante el Grado en Informática sobre desarrollo de software en general y sobre las interfaces gráficas de usuario en particular.

A través de un proceso de análisis del sistema existente, diseño de la solución, implementación y testeo de una solución se busca sustituir la interfaz basada en línea de comandos por una interfaz gráfica de usuario para un sistema Linux.

Estructura de la memoria

La presente memoria se estructura en los siguientes apartados:

1. **Introducción:** En esta capítulo se trata sobre la motivación del TFG y sobre los conocimientos previos que aporta la base conceptual y teórica. También se describe el sistema inicial y se discuten los objetivos y la estructura de la memoria.
2. **Captura de requisitos:** En este capítulo se enumeran y describen los requisitos funcionales y no funcionales de la interfaz a desarrollar.
3. **Análisis funcional** del sistema inicial: En este capítulo se detalla el funcionamiento del sistema inicial, su interfaz y su estructura de navegación. Finalmente se describen tres componentes importantes del mismo, el programa que se ejecuta en el PC, el que se ejecuta en el microcontrolador y el protocolo que utilizan ambos para comunicarse.
4. **Diseño de la solución:** En este capítulo se discuten los procesos de diseño de interfaz y de las clases que formarán la GUI desarrollada.
5. **Implementación:** En este capítulo se describen y justifican las elecciones de los módulos software usados para implementar la GUI. Se habla de la librería gráfica y de ploteo y del lenguaje de programación utilizado en la implementación.
6. **Manual de usuario:** Este capítulo describe las operaciones que puede realizar el sistema y cómo realizarlas a modo de manual de usuario.
7. **Manual de instalación:** En este capítulo se describe el proceso de instalación del sistema con la nueva interfaz gráfica de usuario.
8. **Conclusiones:** En este capítulo se discuten las conclusiones extraídas del desarrollo del TFG.
9. **Referencias:** En este capítulo se muestran las referencias ordenadas al estilo Harvard.
10. **Apéndice:** En este capítulo se incluyen los apéndices. Notablemente el contenido de los ficheros en el empaquetado entregable.

Captura de requisitos

A continuación se detallan los requisitos funcionales y no funcionales del desarrollo de la interfaz gráfica del sistema. Los requisitos ha sido obtenidos de las reuniones mantenidas con D. Iván Santos, en calidad de tutor del proyecto.

Requisitos funcionales

1. La interfaz gráfica debe permitir la realización de la funcionalidad que ahora mismo permite la interfaz por línea de comandos (CLI)
 - a. Realización de medidas puntuales
 - b. Realización de barridos
 - c. Representación gráfica
 - d. Exportación de datos en formato TXT y en imagen.
2. El sistema se comunicará con el microcontrolador siguiendo el protocolo actual existente, sin posibilidad de modificar el mismo.

Requisitos no funcionales

1. La aplicación de la interfaz gráfica deberá correr bajo un sistema operativo Linux.
2. El sistema deberá estar correctamente documentado para permitir su futura expansión y su mantenimiento.
3. El trazado de gráficas deberá ser intuitivo y fácil de expandir o mejorar. Se preferirá el uso de alguna librería de alto nivel.

Análisis funcional del sistema inicial

Interfaz y funcionamiento inicial

El interfaz inicial está basada en línea de comandos (CLI) a través de una metáfora de menú con selector de opciones. Está pensado para ser ejecutado en un entorno Linux en una distribución Ubuntu 12.04 LTS. Tras ejecutar la aplicación el usuario es guiado a través de una serie de menús y mensajes en pantalla.

El usuario introduce los datos para la configuración necesaria y para la ejecución de funcionalidad a través de teclado. El sistema realiza la validación de la entrada y la ejecución de la funcionalidad del sistema, mostrando los resultados en forma textual y gráfica (a través de una ventana que renderiza una representación gráfica de los resultados).

El sistema permite la ejecución de dos operaciones a nivel general que se describen en los siguientes apartados. Antes poder realizar cualquiera de estas operaciones es necesario informar al sistema de la posición inicial del monocromador.

Configuración inicial

Nada más ejecutar la aplicación el sistema pide al usuario que introduzca la posición inicial del monocromador. Esta información es necesaria ya que el microcontrolador no puede conocer este dato por si mismo con el diseño de bajo nivel inicial. Una vez el usuario introduce este valor, que puede ser leído de un display analógico presente en uno de los laterales del monocromador, el sistema muestra el menú principal.

```
Jorge@Jorge-VPCEB4M1E: ~/workspace/Eclipse/TFG/TFGv1/Debug
/-----/
/                               /
/                               /
/-----/
MENÚ PRINCIPAL
/-----/

La posición actual del monocromador es 325.0.
Equivale a: 650.0 nm.
           6500 Å.

Menú principal:
=====
0 Salir.
1 Ir a una longitud de onda.
2 Seleccionar ganancia del fotodiodo.
3 Hacer un barrido.
=====

Escoja una opción: █
```

Configuración inicial

Selección de ganancia

El sistema permite seleccionar la ganancia de la amplificación de la señal que se obtiene del fotodiodo que se utiliza como lector. Esta ganancia puede tomar 4 valores posibles. El sistema no aplica ninguna ganancia por defecto pero permite ajustarla a través de esta pantalla antes de realizar una medida.

```
jorge@jorge-VPCEB4M1E: ~/workspace/Eclipse/TFG/TFGv1/Debug
/=====/
/          FUNCIÓN SELECTOR DE GANANCIA          /
/=====/

Seleccionar ganancia:
=====
1.- Ganancia x1.
2.- Ganancia x4.
3.- Ganancia x16.
4.- Ganancia x64.
=====

Escoja una opción:
3

Ganancia seleccionada: x16.
Pulse enter para continuar: |
```

Selección de ganancia

Realización de un barrido

El sistema permite realizar un barrido para obtener mediadas para un intervalo de longitudes de onda tomando medidas con un cierto paso o intervalo y con una determinada ganancia de amplificación en la señal de medida.

Primeramente el sistema muestra la posición actual del monocromador y pide al usuario que introduzca la posición inicial y final del barrido (en angstroms).

```
jorge@jorge-VPCEB4M1E: ~/workspace/Eclipse/TFG/TFGv1/Debug
/=====/
/          FUNCIÓN HACER UN ABRRIDO          /
/=====/

La posición actual del monocromador es 325.0.
Equivale a: 650.0 nm.
           6500 Å.

Introduza posición inicial (en ångströms):5600
Introduza posición final (en ångströms):6500█
```

Posiciones del barrido

Tras esto el sistema pide al usuario que elija entre las 4 opciones disponibles para el paso del barrido, 2.5 A, 5 A, 10 A y 20 A.

```
jorge@jorge-VPCEB4M1E: ~/workspace/Eclipse/TFG/TFGv1/Debug
/=====/
/                               /
/                               /
/=====/

Seleccionar resolución:
=====
1.- 2.5 A      1 semipaso.
2.- 5.0 A      2 semipaso.
3.- 10 A       4 semipaso.
4.- 20 A       8 semipaso.
=====

Escoja una opción:
2
```

Selector de resolución

El siguiente paso es seleccionar la ganancia para el barrido a través de un menú similar al anterior.

```
jorge@jorge-VPCEB4M1E: ~/workspace/Eclipse/TFG/TFGv1/Debug
/=====/
/          FUNCIÓN SELECTOR DE GANANCIA          /
/=====/

Seleccionar ganancia:
=====
1.- Ganancia x1.
2.- Ganancia x4.
3.- Ganancia x16.
4.- Ganancia x64.
=====

Escoja una opción:
3

Ganancia seleccionada: x16.
Pulse enter para continuar: |
```

Selector de ganancia

Finalmente se pide al usuario que introduzca el nombre del fichero donde se exportarán los datos, dando uno por defecto, que contiene información referente al barrido (la fecha, las posiciones inicial y final, paso y ganancia).

```
Jorge@Jorge-VPCEB4M1E: ~/workspace/Eclipse/TFG/TFGv1/Debug
/          FUNCIÓN HACER UN ABRRIDO          /
/-----/
La posición actual del monocromador es 325.0.
Equivale a: 650.0 nm.
           6500 Å.
Datos del barrido:
/-----/
La posición inicial es: 5600.0 Å
La posición final es: 6500.0 Å
El incremento es: 5.0 Å
La ganancia seleccionada es: 64
/-----/
Por defecto, el nombre del fichero en el que se van a almacenar los datos es:
      B20141211101827-i560.0-f650.0-R0.5-Gx64
¿Desea cambiar el nombre del fichero? (S/N):s
Introduzca el nuevo nombre del fichero: Prueba
```

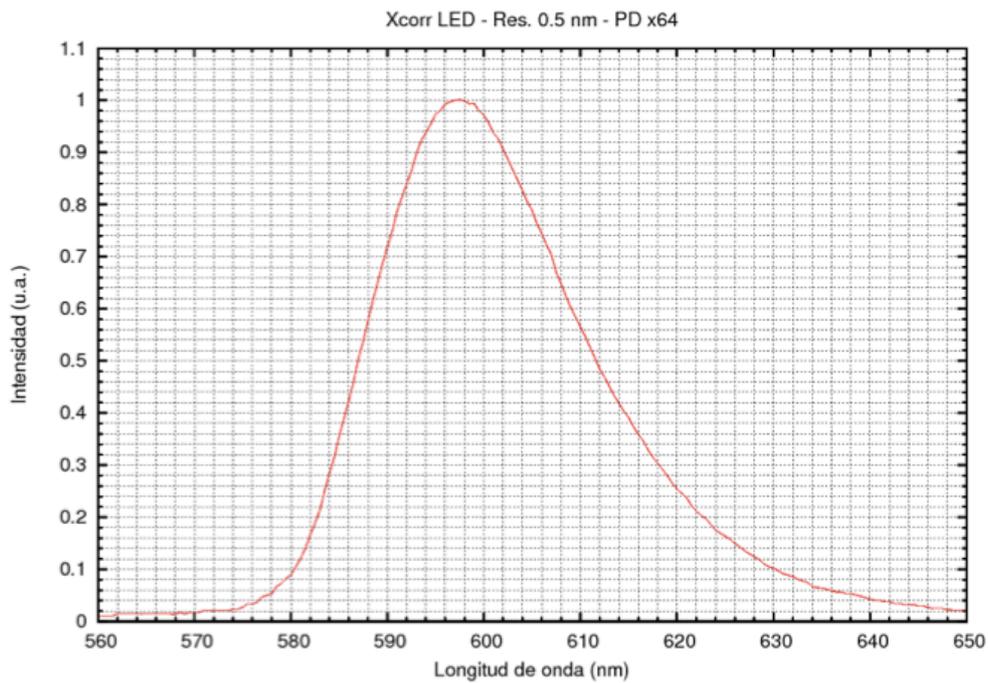
Fichero de salida

Una vez llegado a este punto el sistema realiza las llamadas necesarias al microcontrolador para obtener las medidas y escribir el fichero de salida. También se encarga de llamar a una aplicación de representación gráfica, gnuplot, para mostrar una gráfica en pantalla con los resultados del barrido.

```
jorge@jorge-VPCEB4M1E: ~/workspace/Eclipse/TFG/TFGv1/Debug
Introduzca el nuevo nombre del fichero: Prueba
El nombre del fichero es: Prueba.
Realizando barrido.
=====
5600.0 115
5605.0 116
5610.0 115
5615.0 115
5620.0 116
5625.0 116
5630.0 116
5635.0 116
5640.0 116
5645.0 116
5650.0 116
5655.0 116
5660.0 116
5665.0 116
5670.0 116
5675.0 116
5680.0 117
5685.0 117
5690.0 117
```

Obtención de medidas

Tras finalizar el barrido se vuelve al menú inicial para que el usuario pueda realizar otra operación en el caso de que lo desee.



Ploteo de datos

Estructura de navegación

A continuación se muestra un esquema con la estructura de navegación de la interfaz CLI del sistema inicial.

Pantalla principal :

1. Salir
2. Ir a una longitud de onda
3. Seleccionar una ganancia del fotodiodo
4. Hacer un barrido

Salir:

1. Confirmar...

Ir a una longitud de onda:

1. Introducir longitud de onda en angstroms...

Selector de ganancia:

1. x1
2. x4,1
3. x16
4. x66,13

Hacer un barrido:

- Introduzca la posición inicial...
- Introduzca la posición final...
 - a. Selección de Resolución/paso:
 - i. 2.5 A
 - ii. 5 A
 - iii. 10 A
 - iv. 20 A
 - b. Selección de ganancia:
 - i. x1
 - ii. x4,1
 - iii. x16
 - iv. x66,13
 - c. Introduzca nombre de archivo de salida...

Programa del PC

El programa "main_pc.c" es el encargado de realizar las tareas de la interfaz de comandos (CLI) y de realizar las llamadas al microcontrolador para realizar las operaciones.

El programa ofrece al usuario la posibilidad de realizar dos tareas básicas, la realización de una medida puntual y la realización de un barrido. También permite dos tareas de control como son salir del programa y ajustar la ganancia. El funcionamiento general de dichas tareas ha sido descrito al comienzo del presente capítulo. En este apartado analizaremos con más detalle las operaciones realizadas por el programa ejecutado en el PC aunque estas no estén relacionadas con la funcionalidad o el usuario.

En líneas generales el programa realiza una serie de tareas iniciales, como son configurar la comunicación por el puerto serie, la configuración de los flags de inicio del microcontrolador y otras tareas de configuración e iniciación de recursos y variables.

Tras esto el programa muestra el menú principal al usuario y espera a que este le indique qué operación realizar. Una vez el usuario ha seleccionado una de estas operaciones (Salir, Ajustar la ganancia, Obtener una medida puntual o un barrido) el sistema procede a realizarla. Tras terminar se vuelve a mostrar el menú principal y se comienza de nuevo el proceso (salvo si se ha seleccionado la opción de salir, como es obvio).

El siguiente diagrama de flujo muestra las funciones que se realizan a lo largo del programa.

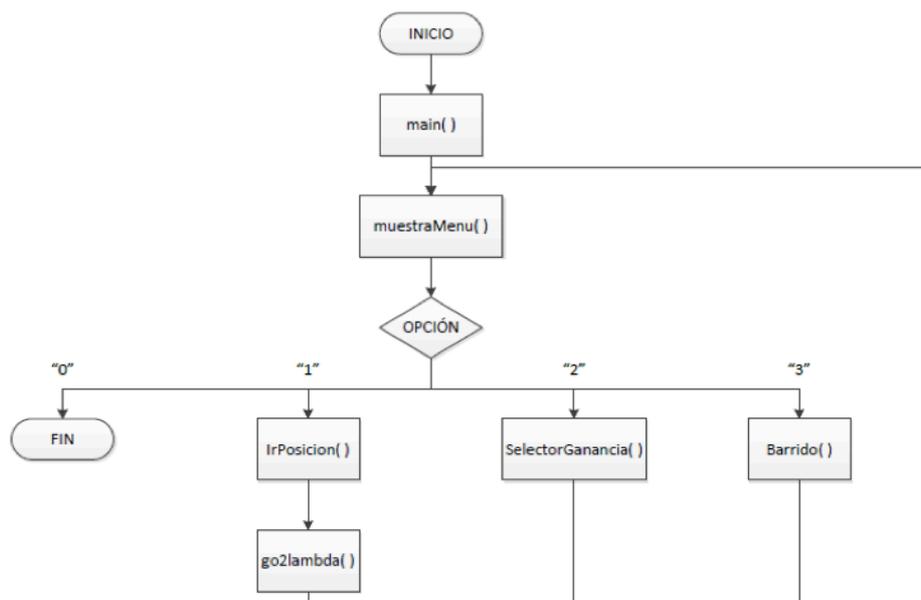


Diagrama de flujo de la ejecución de "main_pc.c". Del TFG de D. Jorge Fernández Lucas.

La función `main()`

La función `main()` se encarga de inicializar la conexión serial. También realiza la configuración de flags en el micro y espera a la respuesta de este para continuar con su ejecución.

Una vez hecho esto la función pide al usuario la posición inicial del monocromador y la guarda internamente. Entonces muestra el menú principal llamando a la función `muestraMenu()`

La función `muestraMenu()`

La función `muestraMenu()` se encarga de mostrar por pantalla las opciones disponibles y de llamar a la función adecuada en cada caso, salvo en el caso de que la opción sea salir, caso en el cual se cierra la aplicación.

La función `irPosicion()`

Esta función se encarga de la lectura puntual en una longitud de onda. Lee dicha posición del usuario y se encarga de hacer las llamadas correspondientes al micro para ir a esa posición (llamando a la subrutina `go2lambda()`) y realizar la medida correspondiente. Una vez la medida se ha mostrado por pantalla se devuelve el control a la función `muestraMenu()` para volver a mostrar el menú principal.

La función `selectorGanancia()`

La función `selectorGanancia()` se encarga de pedir al usuario la ganancia que desea seleccionar para amplificar la señal procedente del fotodiodo. Hay 4 opciones posibles que corresponden con unas ganancias de $\times 1$, $\times 4,1$, $\times 16$ y $\times 66,13$ aproximadamente. Una vez el usuario ha seleccionado una ganancia la función se encarga de comunicarse con el micro para indicárselo. Una vez terminada la tarea y recibido el "OK" del micro se vuelve al menú principal.

La función `barrido()`

La función `barrido` se encarga de realizar un barrido en un rango de longitudes de onda, con un cierto paso o resolución y con una determinada ganancia. A través de menús de opciones se obtienen valores necesarios para el barrido (posición inicial, final, paso y ganancia) y el nombre de un fichero donde guardar los resultados de barrido.

Una vez se han leído del usuario todos estos parámetros la función se encarga de realizar las llamadas correspondientes al micro para realizar el

barrido, incluyendo operaciones de avance y/o retroceso, lectura, control de LED, etc... correspondientes. Una vez obtenidos los resultados estos se almacenan en el fichero de salida y se llama a un software externo (gnuplot) para representar dichos datos gráficamente en una ventana aparte. Una vez finalizado el barrido se vuelve al menú principal.

Para más detalles sobre la implementación de las funciones anteriores puede consultarse la memoria del TFG de D. Jorge Fernández Lucas, en su capítulo 4.

Programa del micro

El programa "main_micro.c" implementa la funcionalidad que se ejecuta en el microcontrolador. Se encarga de realizar las operaciones a bajo nivel a petición del PC y de las operaciones de inicialización y control.

En líneas generales su funcionamiento es el siguiente. Primeramente se realiza configura los pines del microcontrolador. Luego realiza una vuelta inicial del motor paso a paso que controla el monocromador para colocar el motor en una posición conocida. Tras estos pasos de configuración el programa se pone a la espera de recibir comandos por parte del PC.

Cuando llega un comando el programa lo despacha a una función que se encarga de procesarlo y informa al PC cuando este ha finalizado como se ha descrito anteriormente.

Se pueden consultar más detalladas de estas operaciones en la memoria del TFG de *D. Jorge Fernández Lucas*, en el capítulo 4. No nos extendemos en un análisis más pormenorizado del programa del micro puesto que los detalles tienen más bien que ver con la electrónica de control y no son relevantes para el presente TFG.

Protocolo de comunicación micro – PC

La comunicación serial entre el microcontrolador y el PC está sujeta a un protocolo de comunicación que fue desarrollado por D. Jorge Fernández Lucas. A continuación se describen los mensajes que componen el protocolo y las reglas de comunicación.

La comunicación siempre parte desde el PC hacia el micro salvo en el inicio de la aplicación que es el micro el que indica al PC que está listo enviando la cadena "OK".

Los comandos están relacionados con operaciones que realiza el micro, que son requeridas por el PC.

- **Ax** (Avanzar x semipasos): Este comando indica al micro que tiene que avanzar x semipasos la posición del monocromador.
- **Rx** (Retroceder x semipasos): Este comando indica al micro que tiene que retroceder x semipasos la posición del monocromador.
- **E** (Encender LED): Indica al micro que ha de encender el LED para poder medir.
- **O** (Apagar LED): Indica al micro que tiene que apagar el LED. Usamos la letra O de off ya que A se encuentra ocupada por el comando avanzar.
- **M** (Medir): Indica al micro que mida en la posición actual. El micro devolverá la cadena "Mx" con x la medida obtenida.
- **Gx** (Ganancia): Indica al micro la ganancia que desea aplicar a la señal procedente del fotodiodo. El valor de x puede ser 1, 2, 3 o 4.
 - X = 1 -> Ganancia 1
 - X = 2 -> Ganancia 4,1
 - X = 3 -> Ganancia 16
 - X = 4 -> Ganancia 66,13

Tras la ejecución de la operación el micro responde con la letra de la operación seguida de "OK" en la misma cadena, por ejemplo "EOK" para indicar que se ha encendido el LED correctamente o "A24OK" para indicar que se avanzado el monocromador 24 pasos correctamente.

En el caso de avanzar y retroceder, en el caso de que se alcance del final de carrera se enviará el mensaje "FcA" (final de carrera por arriba) o "FcB" (final de carrera por abajo) al PC y posteriormente "OK".

Diseño de la solución

Diseño de la interfaz

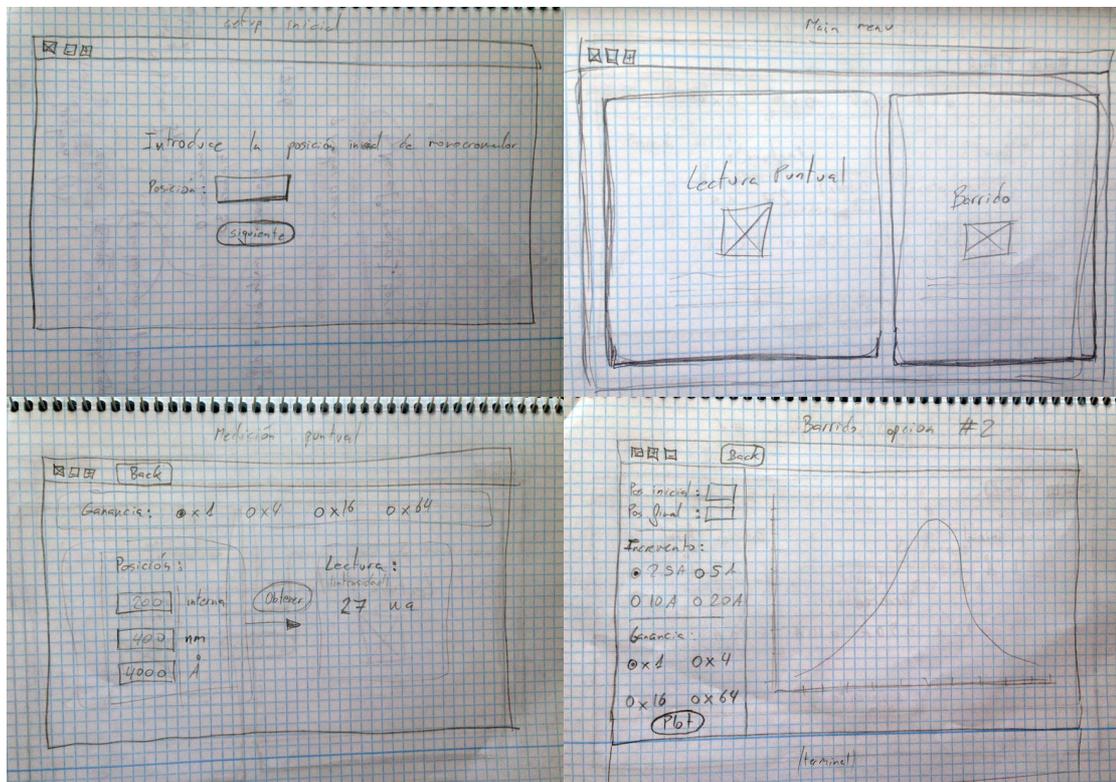
Para el diseño de la interfaz se ha partido del sistema inicial y su interfaz por línea de comandos (CLI). Se han reconocido las funciones o tareas principales; realizar una medida puntual y realizar un barrido. A estas hay que sumar la tarea inicial de configuración de la posición inicial del monocromador.

También en la memoria del TFG de D. Jorge Fernández Lucas, se dan ideas sobre una posible interfaz.

Basándonos en estos datos nos decidimos a implementar una interfaz basada en pantallas por tareas. Así cada pantalla permitirá realizar una tarea global, mostrando solo las opciones relativas a esa tarea. Esto ayuda al usuario a situarse en el contexto de la tarea a realizar y a reducir la complejidad, ya que solo las opciones relevantes para la tarea en cuestión serán mostradas. Será el usuario el que navegue a través de las pantallas para realizar las tareas requeridas.

Durante el diseño de las pantallas se han seguido las 10 heurísticas para el diseño de interfaces de Jakob Nielsen (Nielsen, Jakob, 1995)

Para el diseño de la interfaz primeramente se realizaron los bocetos de las pantallas y posteriormente se depuraron en la fase de codificación a la vez que se detectaba algún problema o inconsistencia y a medida que se conocían las opciones que ofrecía la librería de aplicaciones gráficas.



Bocetos iniciales de la interfaz.

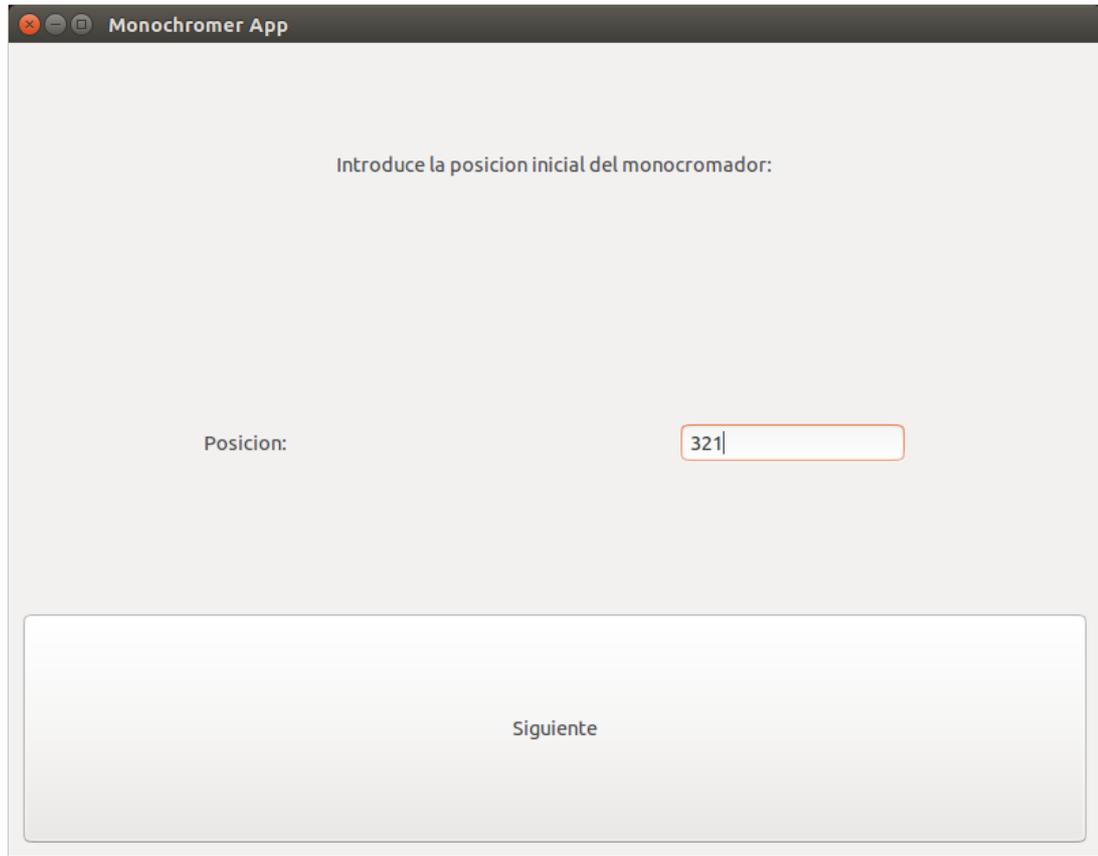
A continuación se muestra un esquema con las pantallas de la aplicación y se comentan detalladamente cada una de ellas.

Pantalla de configuración inicial

- _ Pantalla del menú principal
 - _ Pantalla de medida puntual
 - _ Pantalla de barrido

Pantalla de configuración inicial

Tarea principal: realizar la configuración inicial del sistema.



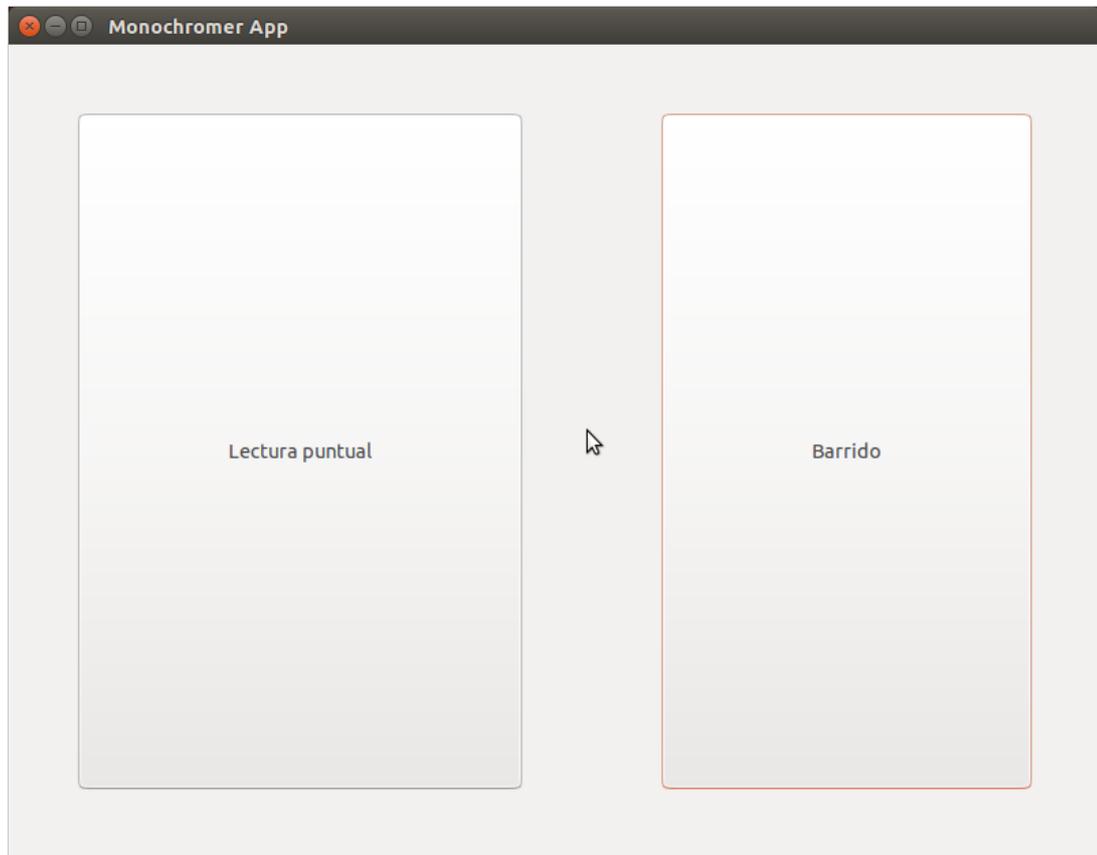
The screenshot shows a window titled "Monochromer App" with a dark header. The main content area is light gray and contains the text "Introduce la posición inicial del monocromador:" centered at the top. Below this, the label "Posicion:" is positioned to the left of a text input field. The input field contains the number "321" and has a vertical cursor at the end. At the bottom of the window, there is a large, light gray button labeled "Siguiete".

Pantalla de configuración inicial.

En esta pantalla el usuario debe introducir la posición inicial del monocromador mostrada en un pequeño dial analógico en el mismo. Tras introducir el número se puede pasar a la siguiente pantalla.

Pantalla del menú principal

Tarea principal: mostrar las tareas realizables por el usuario y permitir acceder a ellas.

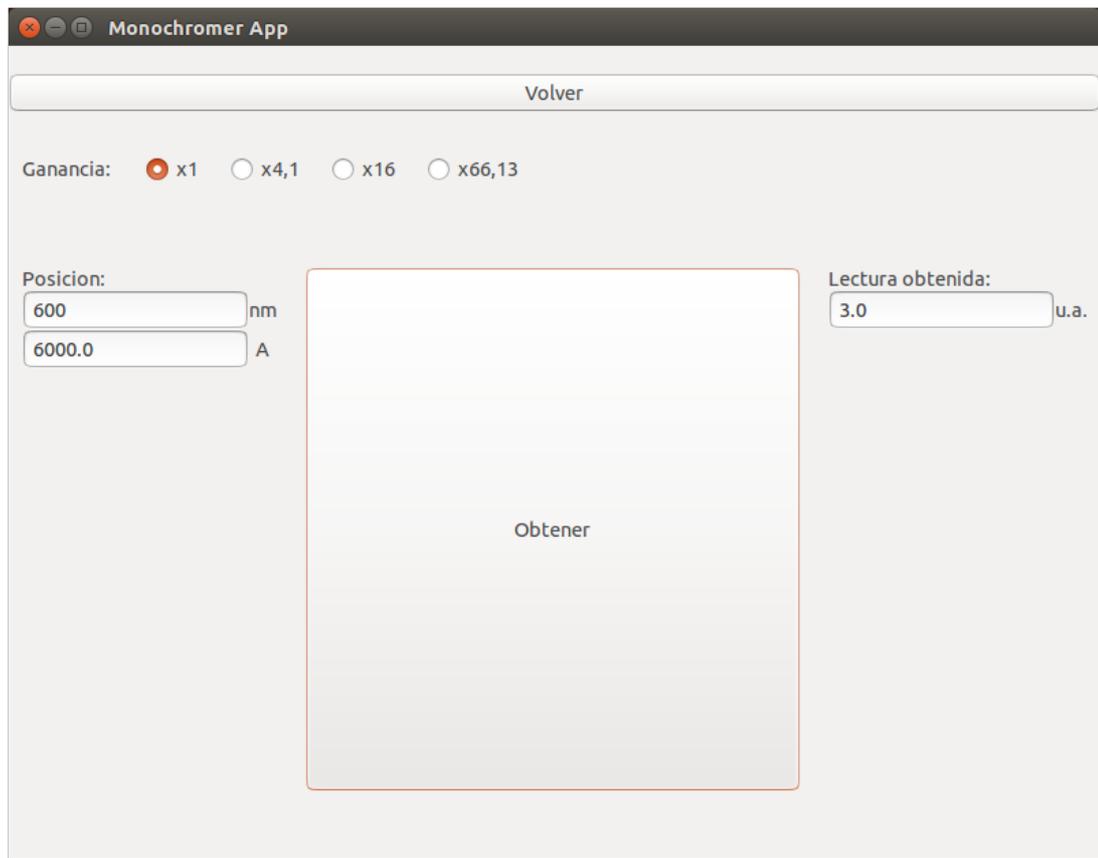


Pantalla del menú principal.

En esta pantalla el usuario ve las opciones de menú y selecciona una. Las opciones se marcan con dos grandes botones. De esta manera se postulan como los elementos más prominentes de la ventana a modo de "call to action". Su gran tamaño ayuda a que se puedan clicar con facilidad.

Pantalla de medida puntual

Tarea principal: realizar una medida puntual en cierta longitud de onda.



The screenshot shows a window titled "Monochromer App". At the top, there is a "Volver" button. Below it, the "Ganancia" section has four radio buttons: "x1" (selected), "x4,1", "x16", and "x66,13". The "Posicion:" section contains two input fields: "600 nm" and "6000.0 A". To the right, the "Lectura obtenida:" section has an input field with "3.0 u.a.". In the center, there is a large button labeled "Obtener".

Pantalla de medida puntual.

En esta pantalla el usuario puede realizar una medida puntual. Primeramente se muestra el botón de volver para que el usuario pueda volver al menú principal. Esto favorece la exploración de la aplicación y el principio de "user control and freedom" de las heurísticas de Nielsen. Este botón está colocado en el mismo sitio en la pantalla de barrido por el principio de consistencia ("Consistency and standards" para Nielsen), acciones similares deben comportarse de manera similar y seguir estándares.

En la parte superior de la pantalla se encuentra el selector de ganancia. Sus ítems se encuentran uno cerca del otro y en orden para mostrar al usuario su relación. Como solo se puede seleccionar una opción se ha elegido un widget de tipo radio button para prevenir errores (que se seleccionen más de dos opciones a la vez). El selector de ganancia presenta también una selección por defecto que muestra el estado del sistema (principio de visibilidad del estado del sistema de Nielsen). También aplica el principio de "reconocimiento antes que memoria" mostrando las opciones disponibles sin que el usuario tenga que recordar cuales son sus valores.

En la parte central de la pantalla se disponen los bloques lógicos de entrada de posición, botón de acción "Obtener" y bloque de salida de datos. Los dos campos de entrada de posición están juntos para mostrar su relación. Además al introducir el número en uno de los campos, el otro se actualiza automáticamente y si se introduce un valor inválido (una letra por ejemplo) el campo se resetea. Este comportamiento sigue el principio de prevención de errores de Nielsen.

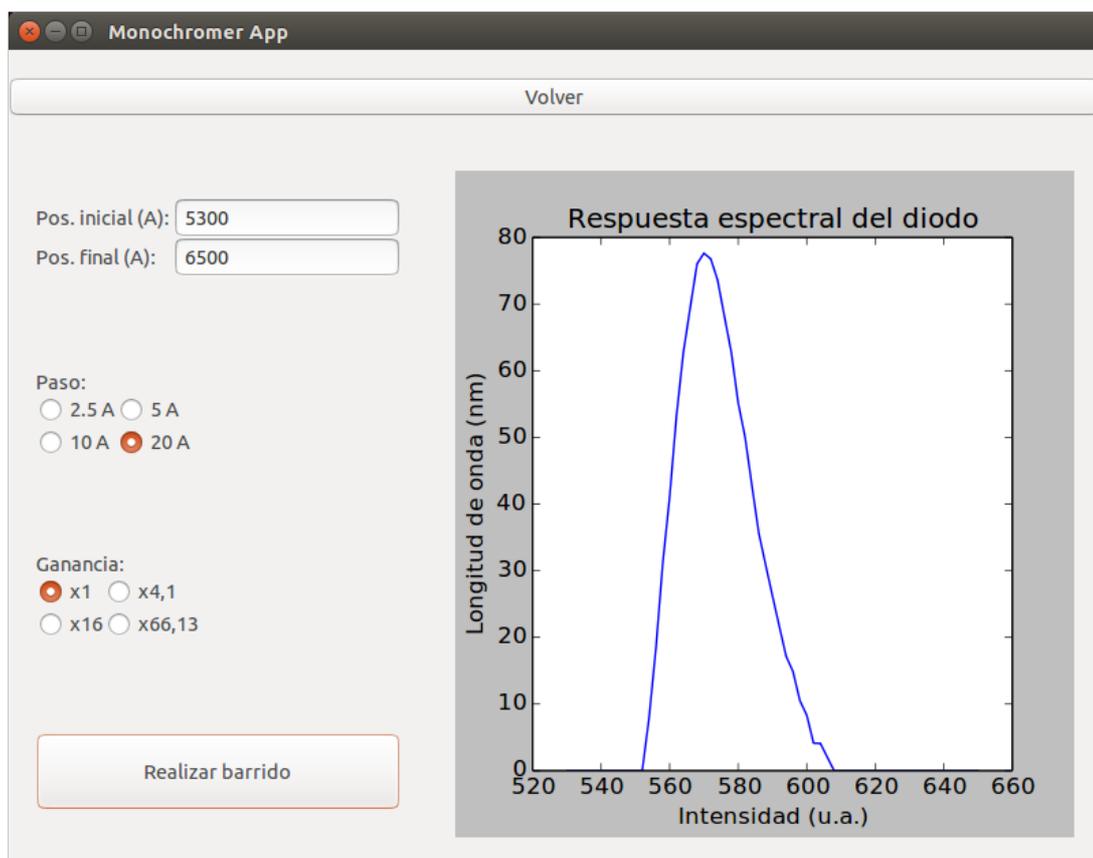
El botón de la acción "obtener" se encuentra deshabilitado hasta que un valor válido se introduce en los campos de posición por el mismo principio anterior de prevención de errores.

Una vez el usuario pulsa el botón el resultado aparece en el campo de la derecha. Este campo no es editable pero sí seleccionable, lo que permite al usuario copiar el texto para pegarlo en otra aplicación.

La tarea puede repetirse infinitas veces hasta que el usuario cierra la ventana (y el programa se cierra también) o vuelve al menú principal.

Pantalla de barrido

Tarea principal: realizar un barrido de medidas en un rango de longitudes de onda de finido por el usuario.



Pantalla de barrido.

Al comienzo de esta pantalla se muestra de nuevo el botón de volver, en el mismo lugar que en la pantalla de medida puntual, por consistencia.

Se utiliza un layout muy parecido a la otra pantalla, con la introducción de datos y opciones en la parte izquierda y la obtención de resultados en la parte de la derecha. Los campos para la entrada de datos están agrupados por bloques semánticos. En los campos de texto se introduce la posición inicial y final del barrido y en los radio buttons se introducen la ganancia y el paso del barrido.

Un vez pulsado el botón de "Realizar barrido" los resultados se muestran en el gráfico de la derecha. Además se guardan dos ficheros en la carpeta *exports*, una imagen en formato PNG y un archivo de texto el resultado del barrido. El nombre se forma a partir de la fecha y hora y los parámetros del barrido.

Diseño de clases

El programa ha sido diseñado en torno a una clase que modela el monocromador. Esta clase actúa como proxy del monocromador ofreciendo métodos que se comunican con el micro a través del puerto serie para realizar la tarea requerida. Son entonces las clases de la interfaz (casi todas clases que heredan de *Gtk.Window*, la clase que representa las ventanas en la librería gráfica) las que utilizan una única instancia de la clase *Monocromador* para realizar acciones como obtener medidas o ajustar la ganancia.

Clase monocromador

La clase *Monocromador* abstrae las operaciones y el comportamiento del monocromador y su microcontrolador. Guarda información sobre el estado del mismo (posición actual, parámetros de ganancia, puerto serie, etc...) y ofrece métodos al exterior para realizar operaciones (como *puntualMeasure()* para realizar una medida puntual).

A continuación detallamos los métodos de la clase *Monocromador*.

Primeramente en su método de creación la clase realiza las tareas iniciales de configuración de la comunicación serie con el micro, configuración inicial de los flags del micro y otras tareas de inicialización.

Después la clase define unos sencillos métodos de conversión de unidades (semipasos, angstroms y nm) de longitud de onda. Estas funciones son usadas por otros métodos de la clase.

La clases define entonces unos métodos que se organizan de forma jerárquica en cuanto a su relación de uso de unos con los otros.

Primeramente se definen las llamadas funciones o métodos de **nivel 0**. Estos métodos representan acciones básicas con el micro, como son encender o apagar el LED, mover el monocromador hacia arriba o hacia abajo o ajustar la ganancia. Estas funciones se encarga de enviar los mensajes correspondientes al micro y de actualizar el estado de la instancia. Dichos métodos son privados para la clase.

Sobre las funciones de nivel 0 se define una función de **nivel 1** que utiliza únicamente funciones de nivel 0, `go2lambda()`. Esta función utiliza la función `move()` de nivel 0 para mover el monocromador a una posición en concreto. Este método también es privado.

Finalmente se definen los métodos públicos de la clase que abstraen comportamientos de más alto nivel, como realizar un barrido o realizar una medida puntual. Estos métodos de **nivel 2** solo utilizan funciones de niveles inferiores.

Clases de la interfaz gráfica

La interfaz gráfica se implementa en cuatro clases que heredan de la clase `Gtk.Window` de la librería gráfica `Gtk+3`. Cada una de estas clases crea los widgets necesarios para la misma y los dota de la funcionalidad requerida.

Cada una de estas clases ventana corresponde con una de las cuatro ventanas de la aplicación. Dichas clases (salvo `mainWindow`) tienen acceso a una instancia de la clase `Monocromador` para poder enviar mensajes de control al mismo para realizar las operaciones requeridas por el usuario.

- `PunctualMeasureWindow`
- `BarridoWindow`
- `MainWindow`
- `InitialSetupWindow`

A parte de estas clases ventana se implementa una clase que hereda de `Gtk.Entry` para implementar un tipo de entrada de texto que modele el comportamiento de las entradas de texto pareadas para introducir la posición en la ventana de medida puntual. Esta clase se llama `PositionEntry`.

Implementación

Python

El sistema ha sido implementado utilizando el lenguaje de programación Python en su **versión 2.7**. Python es un lenguaje de programación interpretado y multiparadigma que soporta programación imperativa, orientada a objetos y funcional. También usa tipado dinámico y es multiplataforma. El lenguaje pone énfasis en la legibilidad del código y la transparencia siguiendo una filosofía propia.

Es un lenguaje muy usado en varios ámbitos incluyendo el científico y académico y que cuenta con un amplio rango de librerías y utilidades.

Hemos elegido este lenguaje por disponer de bindings adecuados para Gtk+3, las librerías adecuadas para el procesamiento gráfico de datos y por ser bastante popular en el ámbito científico y académico. También su facilidad de uso lo hace adecuado para que pueda ser modificado y mejorado por desarrolladores de cuya rama principal no sea la computación ya que es un proyecto encajado en el departamento de telecomunicaciones.

Matplotlib

Para el ploteo de datos se ha utilizado la librería Matplotlib. Pese a que en el sistema inicial el ploteo se realizaba a través de gnuplot, la integración de este en el sistema de ventanas GTK+3 presentaba dificultades.

Matplotlib se ajusta a las necesidades de ploteo que tenemos y está ampliamente documentado y mantenido. Esto permite que las capacidades gráficas de la aplicación puedan ser adaptadas de manera sencilla para incluir nueva funcionalidad en futuros desarrollos. La naturaleza interpretada de Python facilita esta tarea de probar diferentes visualizaciones y comprobar los resultados prácticamente "al vuelo". Puede consultar más información de Matplotlib [aquí](#).

GTK+3

La interfaz ha sido implementada usando la librería **GTK+3** para el escritorio Gnome. Esta es una de las más populares para sistemas Linux y está ampliamente documentada. Se han utilizado los bindings para Python que se encuentran en el módulo **PyGObject**.

Gtk+3 funciona como un sistema basado en eventos. Primeramente se define la interfaz a través de widgets y ventanas y se conecta el manejador de

eventos. Los widgets se conectan con manejadores de señales de modo que cuando el usuario realiza acciones en la interfaz, estas emiten señales que son capturadas por el manejo de eventos que llama a manejador de ese evento en concreto. Así cuando el usuario hace click en un botón el manejador de eventos llama al manejador definido para ese botón que será el que realice cambios en la interfaz u otras operaciones requeridas.

La creación de los widgets y su disposición ha sido realizada de manera manual en el código, sin utilizar ninguna herramienta para la creación de interfaces, como Glade. Al ser una aplicación simple se ha preferido evitar su uso por el sobre coste inicial que necesita (en tiempo de desarrollo).

Puede consultarse más información sobre GTK+3 y sus bindings para Python [aquí](#).

Manual de usuario

El sistema permite realizar dos operaciones básicas, realizar medidas puntuales a partir de una determinada longitud de onda o realizar un barrido en un rango de longitudes de onda para posteriormente mostrar el resultado gráficamente.

A continuación se describe la pantalla de configuración inicial y el sistema de navegación de la aplicación que permite acceder a las operaciones mencionadas anteriormente. Más adelante se detallan las instrucciones para realizar dichas funciones al detalle.

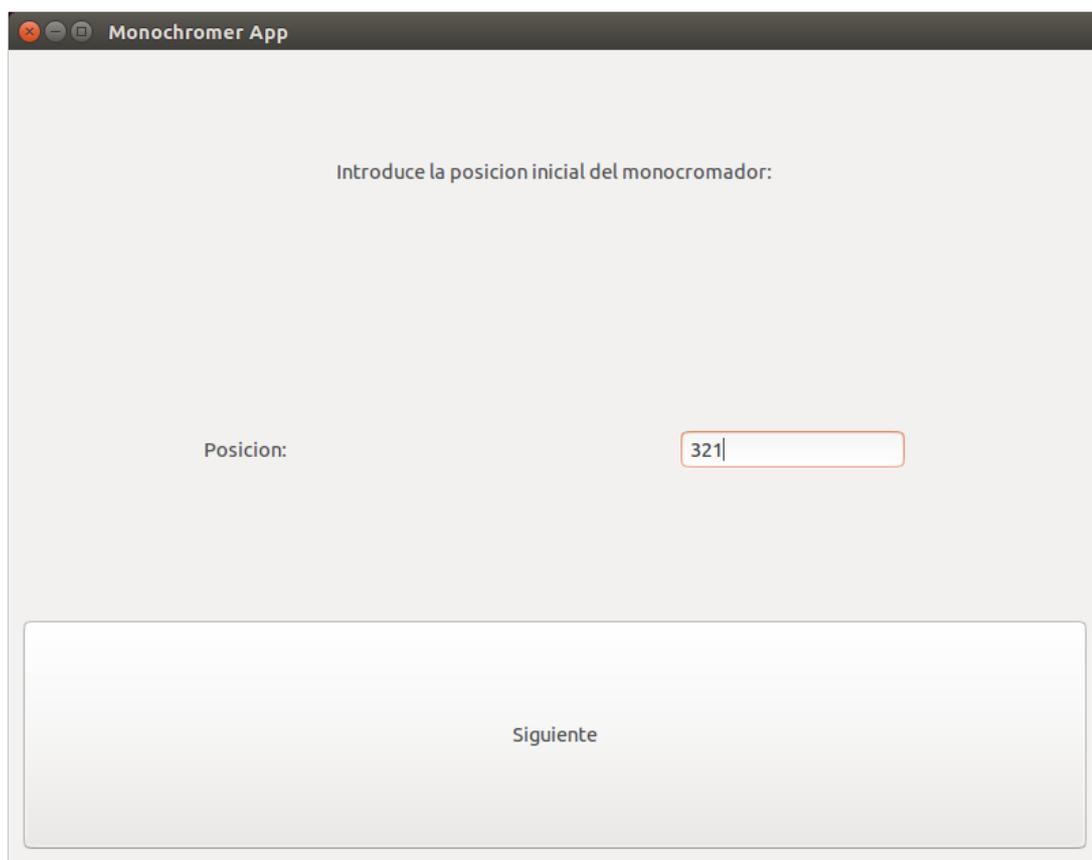
Configuración inicial

La primera pantalla tras arrancar la aplicación es la pantalla de configuración inicial. Para que el sistema funcione correctamente (las medidas sean precisas) se ha de introducir el número correspondiente al dial analógico mostrado en el monocromador.



El dial analógico del monocromador. Dicho número deberá ser introducido al comenzar la aplicación.

El número representa la posición actual del monocromador. Una vez introducido el número se puede pasar a la pantalla del menú principal.



Pantalla de configuración inicial.

Pasos para efectuar la configuración inicial:

1. Introduzca el número indicado en el dial analógico (ver imagen anterior) del monocromador en el campo *Posición inicial*:
2. Pulse *ENTER* o el botón *Siguiente*.

Navegación entre pantallas

La navegación entre pantallas se realiza a través de botones que hacen pasar de una ventana a otra. La estructura de ventanas es la siguiente:

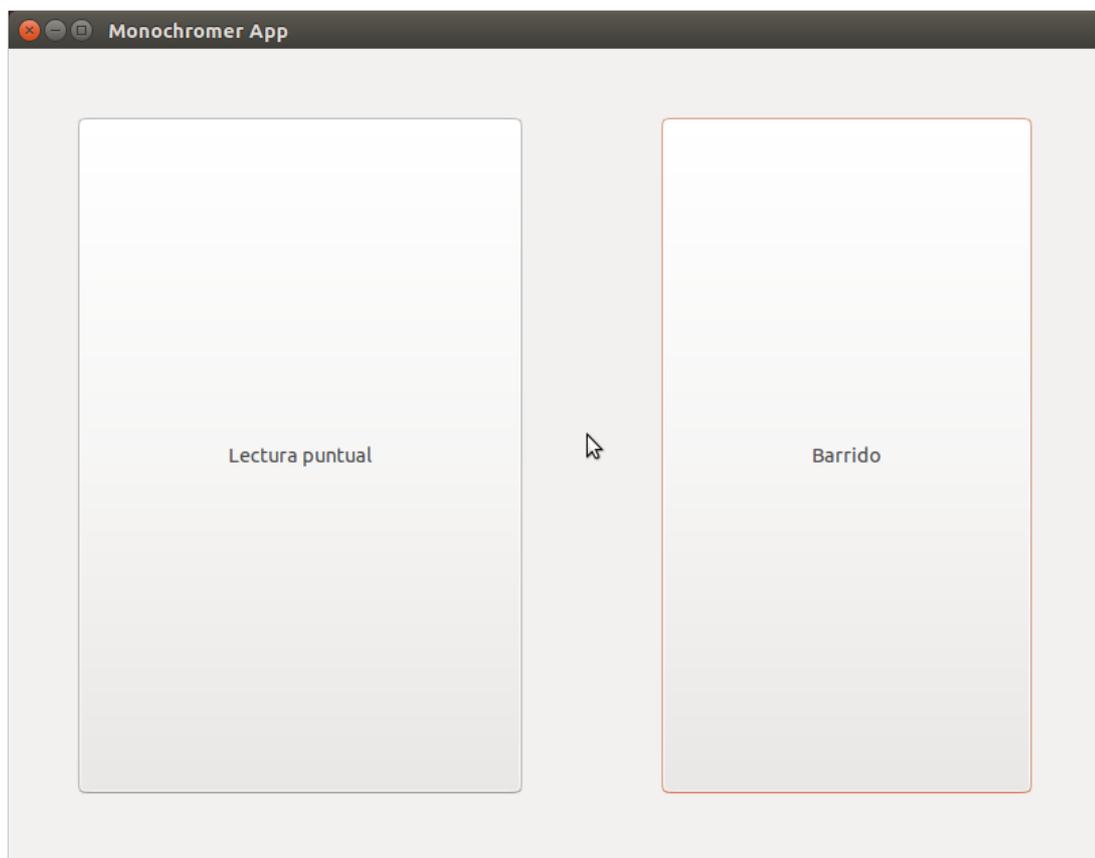
Pantalla de configuración inicial

- _ Pantalla del menú principal
 - _ Pantalla de medida puntual
 - _ Pantalla de barrido

La pantalla de configuración inicial solo es accesible nada más ejecutar la aplicación. Una vez se ha introducido el valor de la posición inicial del monocromador y pasado al menú principal no se podrá volver a dicha ventana.

Para navegar entre las diferentes ventanas realice los siguientes pasos:

1. Desde la pantalla del *menú principal* pulse el botón "*Lectura puntual*" para acceder a la pantalla de *lectura puntual* o el botón "*Barrido*" para acceder a la *pantalla de barrido*.
2. Para volver al menú principal pulse el botón volver en la parte superior de la ventana de *lectura puntual* o *barrido*. Tenga en cuenta que no es posible navegar entre pantallas mientras se están realizando operaciones de lectura/barrido.

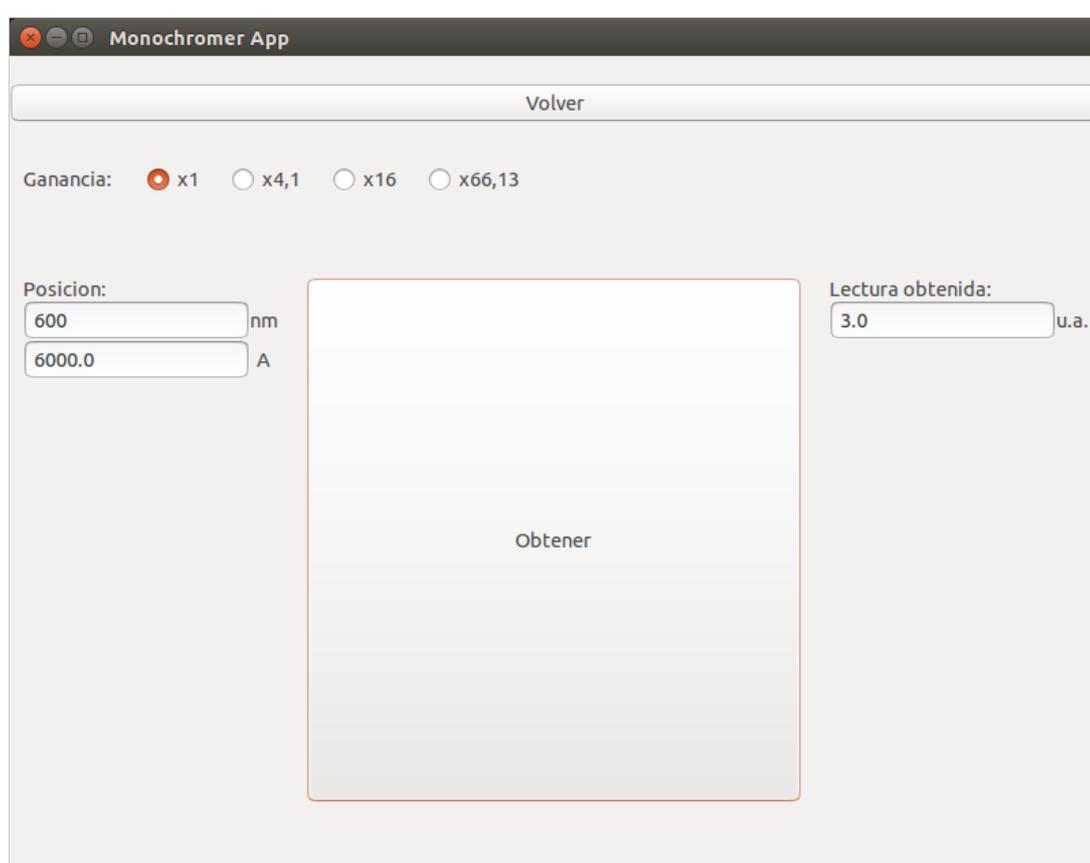


Pantalla de menú principal.

Realizar una medida puntual

Para realizar una medida principal primeramente tiene que encontrarse en la *pantalla de medida puntual*. Consulte el apartado "Navegación entre pantallas" del presente manual de usuario.

Una vez se encuentre en la *pantalla de medida puntual* podrá realizar medidas puntuales y ver el resultado de las mismas. Podrá introducir la longitud de onda en la cual desea obtener la medida en dos unidades, angstroms y nanómetros. También podrá ajustar la ganancia de amplificación con la que se realizará la medida. La ganancia seleccionada por defecto es "x1", es decir, sin amplificación.



The screenshot shows the 'Monochromer App' interface. At the top, there is a 'Volver' button. Below it, the 'Ganancia' section has four radio buttons: 'x1' (selected), 'x4,1', 'x16', and 'x66,13'. The 'Posicion:' section has two input fields: the first contains '600' with 'nm' to its right, and the second contains '6000.0' with 'A' to its right. In the center is a large 'Obtener' button. On the right, the 'Lectura obtenida:' section has an input field containing '3.0' with 'u.a.' to its right.

Pantalla medida puntual.

Para seleccionar una ganancia realice los siguientes pasos:

1. Seleccione la ganancia deseada en el panel de ganancia marcado por la etiqueta "Ganancia".

Para realizar una medida puntual realice los siguientes pasos:

1. Introduzca la longitud de onda donde quiere realizar la medida en uno de los campos de "Posición", dependiendo de la unidad en la que desee

hacerlo (nanómetros marcado con la etiqueta "nm" o Angstroms marcado con la etiqueta "A"). El campo de la otra unidad se actualizará automáticamente según vaya introduciendo el valor.

2. Pulse el botón "Obtener".
3. La medida obtenida aparecerá en el campo "Lectura obtenida" en la parte derecha de la pantalla. Tenga en cuenta que la operación puede llevar algunos segundos dependiendo de la posición del monocromador respecto a la posición objetivo donde se quiere realizar la medida.

Realizar un barrido

Para realizar barrido primeramente tiene que encontrarse en la *pantalla de barrido*. Consulte el apartado "Navegación entre pantallas" del presente manual de usuario.

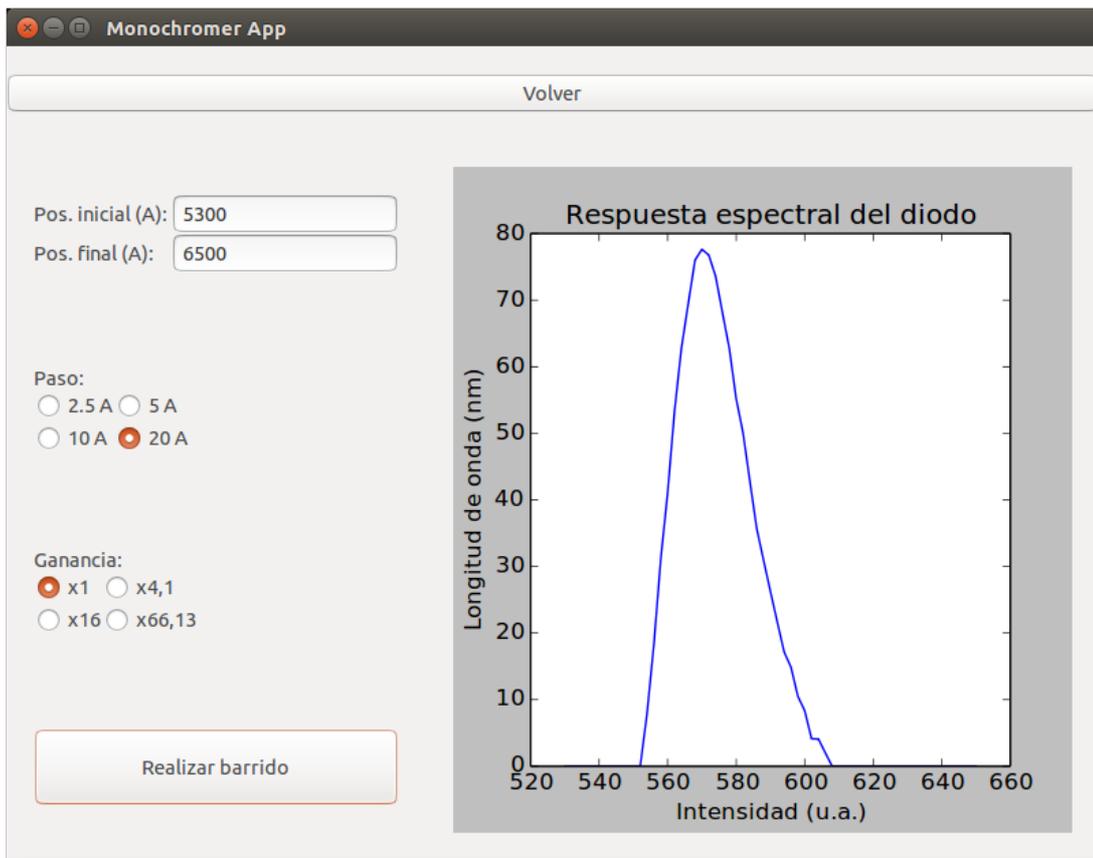
Una vez se encuentre en la *pantalla de barrido* podrá realizar un barrido en un rango de longitudes de onda y ver el resultado de las mismas en un gráfico en la pantalla. Podrá ajustar la ganancia de amplificación con la que se realizará la medida y el paso en el que se realizará el barrido. La ganancia seleccionada por defecto es "x1", es decir, sin amplificación. El paso seleccionado por defecto es de 2.5 Angstroms.

Para cambiar el paso y/o la ganancia realice los siguientes pasos:

1. Utilice el selector de ganancia marcado por la etiqueta "Ganancia" para seleccionar la ganancia elegida.
2. Utilice el selector de paso marcado por la etiqueta "Paso" para seleccionar el paso elegido.

Para realizar un barrido realice los siguientes pasos:

1. Compruebe los parámetros de ganancia y de paso. En caso de que desee cambiarlos proceda con los pasos del apartado anterior "Para cambiar el paso y/o ganancia".
2. Introduzca la posición inicial y la posición final para el barrido en el campo "Pos. inicial" y en el campo "Pos. final" respectivamente.
3. Pulse el botón "Realizar barrido" que se encuentra en la parte inferior izquierda de la ventana.
4. El resultado del barrido aparecerá en una representación gráfica en la parte inferior derecha de la pantalla.
5. En la carpeta "exports" aparecerán los ficheros TXT y PNG con los resultados del barrido.



Pantalla de barrido.

Manual de instalación

Para instalar el sistema primero se han de instalar ciertas librerías/módulos para Python 2:

- [PyGObject](#) y sus dependencias:
 - GTK+3
 - Python 2 (2.6 or later)
 - gobject-introspection
- [Matplotlib](#) 1.3 o superior
- [PySerial](#) 2.7
- Módulos de Python math, termios, struct, fcntl, os (normalmente instalados por defecto junto con Python)

Todos estas librerías/modulos están disponibles como empaquetados para las distribuciones de Linux más comunes como Ubuntu (14.04 LTS), donde ha sido desarrollado y testeado el sistema. Consulte más información sobre cómo instalar paquetes de software en Ubuntu [aquí](#).

Para ejecutar el programa se ha de navegar al directorio donde se ha descomprimido el empaquetado a través de la interfaz de comandos. Una vez en el directorio raíz se ha de ejecutar el programa con **permisos de root** (por el acceso al puerto serie):

```
>>> sudo python MonochromatorApp.py &
```


Conclusiones

El desarrollo de interfaces de usuario está cobrando mayor importancia en los últimos años. Con la democratización de la computación a las masas en forma de ordenadores de sobremesa, portátiles y últimamente a sistemas móviles el perfil de usuario ha cambiado drásticamente. Ahora los usuarios de programas sistemas de información presentan unos conocimientos mucho más diversos en cuanto a sistemas de información. Lo sistemas son más generalistas y han de ser diseñados con la usabilidad y experiencia de usuario en mente.

En el caso concreto del desarrollo de la interfaz para el sistema de caracterización de LED's sobre el que versa este TFG, la interfaz gráfica supone una mejora global al sistema a varios niveles.

La nueva interfaz mejora la experiencia de usuario al ofrecer una interfaz gráfica, en contraste con al interfaz basada en comandos. Las interfaces gráficas aportan factores importantes como una mayor habilidad para comunicar las opciones disponibles del sistema (discoverability). Proporcionan también una estructuración lógica de elementos de feedback y actuadores con respecto a la funcionalidad el sistema, es decir ordenan de forma lógica los botones, etiquetas y widgets. También mejoran la usabilidad haciendo el la interacción con los sistemas de información más fácil y eficiente.

Durante el desarrollo del TFG gran parte del trabajo ha sido comprender y entender el funcionamiento del sistema base. Primeramente a través de la memoria del sistema y después a través del código de las aplicaciones tanto del PC como del microcontrolador. Es una buena experiencia el tener que bucear en el código escrito por otra persona y valorar aspectos como la sencillez en los algoritmos, claridad en implementarlos o la cantidad de comentarios en el código.

Pese a que la memoria ha supuesto una parte central para comprender el sistema y el código los comentarios incrustados en el mismo han sido de gran ayuda. Esto pone de manifiesto la importancia de la documentación de los sistemas software, ya sea en manuales o documentos adjuntos o en el mismo código. Muchas veces es más interesante programar de una manera que se más fácil de entender para alguien externo al proyecto que hacerlo de la manera más breve, condensada o eficiente posible. Muchas veces esta supuesta eficiencia es ínfima con la potencia de los sistemas de computación actuales y no merece la pena; seguramente en futuras actualizaciones o resoluciones de bus se aprecie mucho más la claridad. Esto es especialmente importante en desarrollos grandes donde son varios los programadores que participan en el desarrollo y el mantenimiento del software.

El desarrollo del TFG se desarrolla navegando entre varios sub-sistemas. El sistema original tiene una parte de hardware que a su vez se divide en varios sub-sistemas. Algunos de estos son comerciales como el monocromador mientras que otros han sido desarrollados ad-hoc para el sistema, como las placas del fotodiodo o de acople del diodo LED. El nivel de software también se distribuye a varios niveles. En el más bajo se encuentra la programación del microcontrolador para interactuar con los sistemas electrónicos de bajo nivel. Por encima se encuentra el software que se ejecuta en el PC en un lenguaje de alto nivel, Python en nuestro caso, con librerías gráficas especializadas. Trabajar con todos estos sub-sistemas para obtener un comportamiento emergente ha sido muy gratificante y estimulante.

Referencias

- Fernández Lucas, (2015) Jorge. Diseño y fabricación de un sistema electrónico para la caracterización óptica de diodos emisores de luz. ETSI Telecomunicación, Universidad de Valladolid, 2015.
- Jakob Nielsen, (1995) 10 Usability Heuristics for User Interface Design. Nielsen Norman Group. Available from:
<http://www.nngroup.com/articles/ten-usability-heuristics/>
- <https://wiki.gnome.org/Projects/PyGObject>
- <http://matplotlib.org/>
- https://en.wikipedia.org/wiki/Graphical_user_interface
- https://en.wikipedia.org/wiki/Light-emitting_diode

Apéndice

Contenido del archivo empaquetado entregable

- **Memoria.pdf**: la memoria en formato pdf.
- **docs**: contiene diverso material de referencia
- **IMG**: las imágenes utilizadas en la memoria.
- **src**: el código fuente y ejecutable de la app y las carpetas de ejecución.
 - **exports**: carpeta para los archivos exportados por el app.
 - MonochromatorApp.py

