

# Modulo motorizado para ilustrar la difracción, atenuación y absorción de ondas electromagnéticas en el espectro infrarrojo; utilizando tecnologías libres y de bajo costo.

Diego Parra,\* Julian Salamanca<sup>†</sup>

Universidad Distrital, Calle 3 No 26A-40 Bogotá-Colombia  
Grupo de Física e Informática “FISINFOR”

March 21, 2016

## Abstract

This writing describes in detail the construction of a motor vehicle taking advantage of the technological wonders of semiconductors to control hardware with a ATMEGA328P-PU microcontroller with serial communication via Bluetooth to a computer whose operating system is GNU-Linux, equipped with a side infrared sensor illustrating the diffraction phenomenon and calculates the wavelength emitted by the LED diode emitting infrared; also it has a pair of emitter sensors - infrared receiver on the front of the vehicle, the infrared sensor front and infrared emitter front illustrate the law of decay density radiansa with the square of the distance to the radiating source, the reflectance and transmittance that occurs due to the interaction of the radiation with matter.

This project also has instructions for using the control software free infrarossi and their respective installation, whose function is to control the actions taken by the infrarossi vehicle, both forward and capture data from the side and front sensors and control the radiant source photon in the infrared, for examination. It is noteworthy that all this communication between the motor module infrarossi and control software comes pre designed to be via bluetooth.

**Keywords:** infrarossi motorized module, control software free infrarossi, law decay density radiansa with the inverse square, reflectance, diffraction, serial communication, bluetooth, GNU-Linux, software, hardware.

## Resumen

El presente escrito, describe detalladamente la construcción de un vehículo motorizado aprovechando las maravillas tecnológicas de los semiconductores para el control de hardware con un microcontrolador atmega328P-PU, con comunicación serial vía bluetooth con un ordenador cuyo sistema operativo es GNU-Linux, equipado con un sensor infrarrojo lateral que ilustra el fenómeno de difracción y calcula la longitud de onda emitida por el diodo led emisor en infrarrojo; también tiene un par de sensores emisor – receptor infrarrojo en la parte del frente del vehículo, el sensor infrarrojo frontal y el emisor infrarrojo frontal ilustran la ley del decaimiento de la densidad de radiansa con el cuadrado de la distancia a la fuente radiante, la reflectancia y transmitancia que ocurre debido a la interacción de esta radiación con la materia.

Este proyecto cuenta también con las indicaciones para el uso del software de control free infrarossi y su respectiva instalación, cuya función es controlar las acciones que realiza el vehículo infrarossi, tanto avanzar y capturar datos de los sensores lateral y frontal como de controlar la fuente radiante de fotones en el infrarrojo, para su respectivo análisis. Cabe mencionar que toda esta comunicación entre el modulo motorizado infrarossi y su software de control viene pre diseñada para que sea vía bluetooth.

**Descriptores:** Modulo motorizado infrarossi, software de control free infrarossi, ley de decaimiento de la densidad de radiansa con el inverso del cuadrado, reflectancia, difracción, comunicación serial, bluetooth GNU-Linux, software, hardware.

---

\*diegoestudianteud1@gmail.com

<sup>†</sup>jasalamanca@udistrital.edu.co

# 1 Introducción

En la actualidad, una cantidad de instrumentos de laboratorio para la enseñanza en física y especialmente de fenómenos electromagnéticos, transporte e inyección de energía a los portadores de carga en los semiconductores, fenómenos ondulatorios, etc.; son muy costosos, incluso estos instrumentos son exclusivos de los departamentos de ciencias en diferentes universidades y laboratorios, lo que dificulta el contacto de la población a estas manifestaciones físicas; un caso especial es el estudio de las propiedades de las ondas electromagnéticas y su dualidad onda – partícula, en determinadas frecuencias “caso clásico” o en paquetes de energía discreta según su longitud de onda “caso cuántico”; específicamente en cuestión de frecuencias o longitudes de onda que corresponden al espectro electromagnético infrarrojo, concretamente cuando se habla de difracción, atenuación, absorbancia, transmitancia, reflectancia, ley de decaimiento de la densidad de radianza con el inverso del cuadrado de la distancia a la fuente; solo se tiene como marco de estudio de estos fenómenos fenómenos ondulatorios infrarrojos a la espectrometría infrarroja, que es usada habitualmente por estudiantes de medicina, química, biología, etc., muy especializados, con instrumentos muy costosos y precisos; lo que deja sin aproximarse a los demás estudiantes de ciencias exactas a estos tópicos de la física que están presentes en la vida diaria y que no son visibles al ojo humano.

El objetivo de este trabajo de grado es diseñar un instrumento de laboratorio que sea capaz de ilustrar tres fenómenos físicos de la radiación electromagnética y su dualidad onda – partícula; los cuales son: difracción en infrarrojo, reflectancia en el infrarrojo y ley del decaimiento de la irradiancia con el inverso del cuadrado de la distancia a la fuente. Esto con el fin de dejar referencia que en la actualidad no se estudia de manera cómoda los fenómenos ondulatorios y de interacción de esta manifestación física de la energía con la materia, en el espectro de esta longitud de onda como lo es el infrarrojo.

Por esta razón se elabora muy cuidadosamente un instrumento que llene las expectativas de aprendices y docentes de carreras afines a estos asuntos; el cual es económico, con materiales de fácil acceso, que es capaz de aproximar al educando a fenómenos como lo es la difracción, la ley del decaimiento de la densidad de radianza con el inverso del cuadrado de la distancia a la fuente radiante, la reflectancia y transmitancia que ocurre debido a la interacción de esta radiación con la materia, en el rango infrarrojo del espectro.

A este instrumento se le denominó *modulo motorizado infrarossi*, el cual es un vehículo de tracción electromagnética, con comunicación serial vía bluetooth con un ordenador GNU-Linux, equipado con un sensor infrarrojo lateral para ilustrar el fenómeno de difracción, un sensor infrarrojo frontal y un emisor infrarrojo frontal para ilustrar la ley del decaimiento de la densidad de radianza con el cuadrado de la distancia a la fuente radiante, la reflectancia y transmitancia que ocurre debido a la interacción de esta radiación con la materia; el modulo infrarossi es capaz de acercar de manera cualitativa y cuantitativa al estudio de estos contenidos, pues aparte del instrumento que obtiene los datos, también cuenta con un software de control y análisis de datos obtenidos, a este software se le designó el nombre de *software de control free infrarossi*, el cual complementa el modulo motorizado infrarossi, convirtiéndolo en una herramienta de laboratorio muy precisa, económica, fácil de utilizar, con su propio repositorio en github, con buena documentación, fácil de manipular e instalar.

## 2 Diseño experimental

El modulo motorizado infrarossi ver figuras 33 , 32 , 31 y 30, en el anexo B; es un vehículo con chasis de madera, batería de 9V del tipo cuadrada recargable, modulador de señal de 9V a 5V voltaje continuo, tracción electromagnética con transmisión de tipo engranaje trasera con diferencial cero de eje posterior fijo y eje delantero fijo, cuatro llantas de cinco centímetros de diámetro y dos centímetros de ancho, comunicación bluetooth entre vehículo y ordenador, dos sensores de intensidad lumínica en el rango infrarrojo del espectro electromagnético de estructura epoxy cinco milímetros, uno del tipo radar sin barrido basado en el principio de electro-recepción[1] activa con disposición espacial en la parte media del sector frontal, y en la zona lateral derecha a siete centímetros del sector frontal reposa el sensor del tipo radar sin barrido basado en el principio de electro-recepción pasiva, control de rapidez en el avance del vehículo y control de hardware del vehículo.

### 2.1 Disposición eléctrica

El modulo infrarossi es un vehículo didáctico para la enseñanza experimental en física ver figuras 24 y 23 , los diferentes sistemas electrónicos del vehículo pueden removese en cuatro sistemas: control de hardware y comunicación bluetooth con ordenador ver figura 25, control rapidez del vehículo ver figura 26 , sensor electro-recepción

pasiva ver figura 27 , y, un sensor de electro-recepción activa ver figura 28 .

### 2.1.1 Control de hardware

El modulo infrarossi tiene un sistema de control de hardware ver figura 25, que le indica al vehículo cuando debe avanzar, cuando debe detenerse, la rapidez con que debe avanzar, la distancia que debe avanzar, activar y detener el sensor de electro-recepción pasiva y enviar los datos vía bluetooth al ordenador, activar y detener el sensor de electro-recepción activa y enviar los datos vía bluetooth al ordenador, enviar señales del proceso que esta realizando con el encendido y apagado de diodos leds de colores y mantener una comunicación continua con el ordenador vía bluetooth.

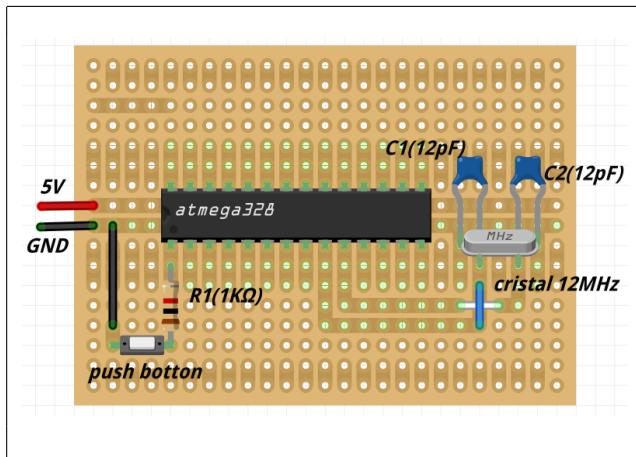


Figura 1: Esquema tarjeta perforada con microcontrolador atmetga 328, cristal oscilador  $12\text{MHz}$ , dos condensadores cerámicos de  $12\text{pF}$ , resistencia de  $1\text{k}\Omega$  y un botón; los cables de color negro son tierra, los de color rojo son voltaje, el cable azul y blanco son puentes. Figura generada en Fritzing.

El control de hardware ver figura 7 , está basado en un microcontrolador atmega[2] 328P-PU de 28 pines, con un voltaje de operación de  $5\text{V}$ , una corriente máxima de  $40\text{mA}$  por pin de salida, la suma de corriente en todas las salidas del microcontrolador no excede los  $150\text{mA}$ , trabaja en una frecuencia de  $12\text{ MHz}$ . Este dispositivo consta de un modulador de frecuencia, un modulador de señal de  $9\text{V}$  a  $5\text{V}$  voltaje continuo, un modulo de comunicación tipo bluetooth, un botón de reinicio y un led indicador de procesos.

El modulador de frecuencia como se muestra en las figuras 1 y 2 , sincroniza el reloj interno del microcontrolador, calibrando los tiempos en los procesos del microcontrolador con los indicados por el fabricante y que reposan en la pagina oficial del proyecto arduino<sup>1</sup> .

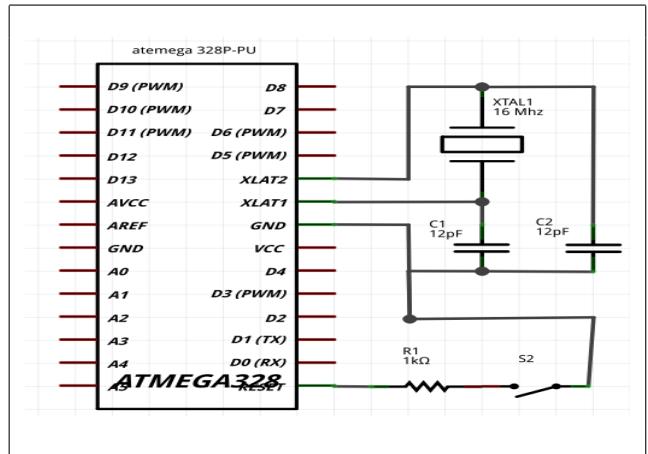


Figura 2: Esquema eléctrico microcontrolador atmetga 328, cristal oscilador  $12\text{MHz}$ , dos condensadores cerámicos de  $12\text{pF}$ , resistencia de  $1\text{k}\Omega$  y un botón. Figura generada en Fritzing.

El botón de reinicio o switch de reset como se muestra en las figuras 1 y 2, este interruptor reinicia el firmware cargado dentro del microcontrolador cuando surge algún conflicto interno en los procesos de este dispositivo.

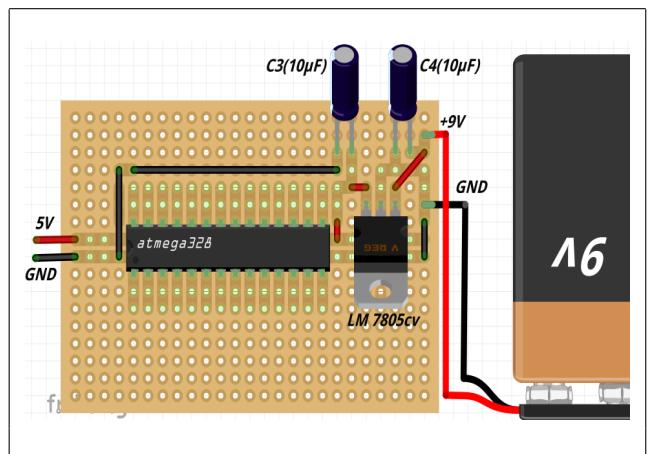


Figura 3: Esquema tarjeta perforada microcontrolador atmetga 328, transistor LM 7805 cv, dos condensadores electrolíticos de  $10\mu\text{F}$ , conectados a batería de  $9\text{V}$ . Figura generada en Fritzing.

<sup>1</sup>Página oficial proyecto arduino, [on line] <https://www.arduino.cc/>

El modulador de señal como se muestra en las figuras 3 y 4 , filtra la señal de entrada de 9V voltaje continuo a la señal de trabajo óptima del microcontrolador y los diferentes sistemas electrónicos cual es de 5V voltaje continuo.

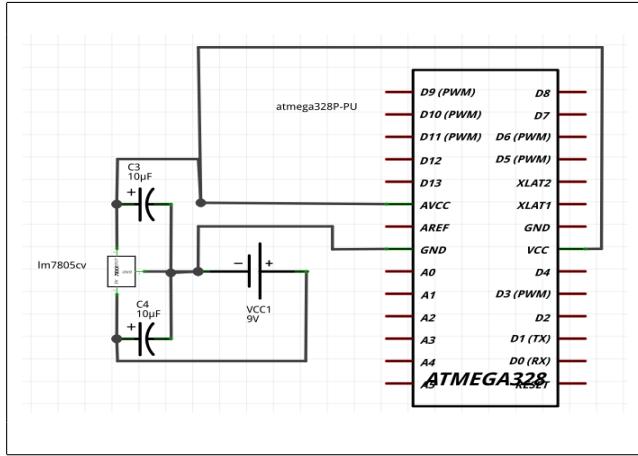


Figura 4: Esquema eléctrico microcontrolador atmeg328, transistor LM 7805 cv, dos condensadores electrolíticos de  $10\mu F$ , conectados a batería de 9V. Figura generada en Fritzing.

La comunicación del microcontrolador con el ordenador como se muestra en las figuras 5 y 6 la ejecuta el modulo bluetooth hc-05; de esta manera el software de control free infrarossi tiene completo dominio del hardware del vehículo motorizado infrarossi.

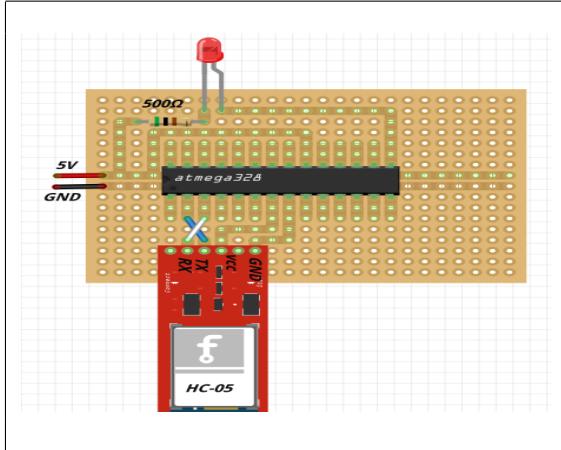


Figura 5: Esquema tarjeta perforada microcontrolador atmeg328, modulo bluetooth hc-05, diodo led de 3mm y una resistencia de  $500\Omega$ . Figura generada en Fritzing.

El avance espacial del vehículo como el encendido y apagado de los diferentes sensores del dispositivo, junto con la recolección de datos de los sensores y la verificación del

proceso que realiza la tarjeta, son enviados por el modulo hc - 05 al ordenador, para su respectivo análisis; de igual forma las ordenes provenientes del software de control via bluetooth son recibidas por el modulo hc - 05 y enviadas al microcontrolador para su inmediata ejecución.

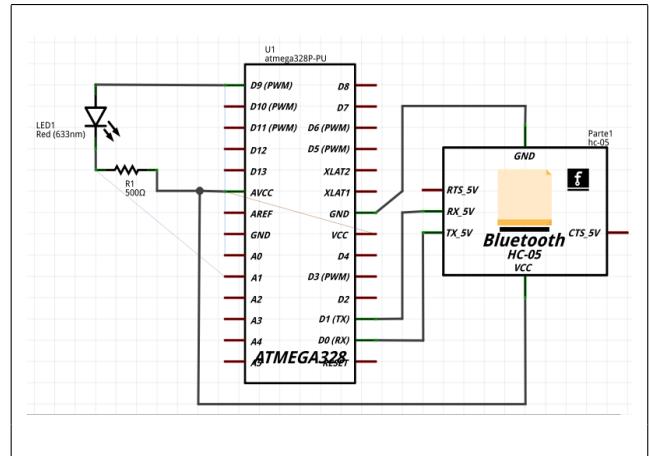


Figura 6: Esquema eléctrico microcontrolador atmeg328, modulo bluetooth hc-05, diodo led de 3mm y una resistencia de  $500\Omega$ . Figura generada en Fritzing.

El diodo led que se observa en las figuras 5 y 6 es un indicador lumínico de los procesos que esta realizando el microcontrolador.

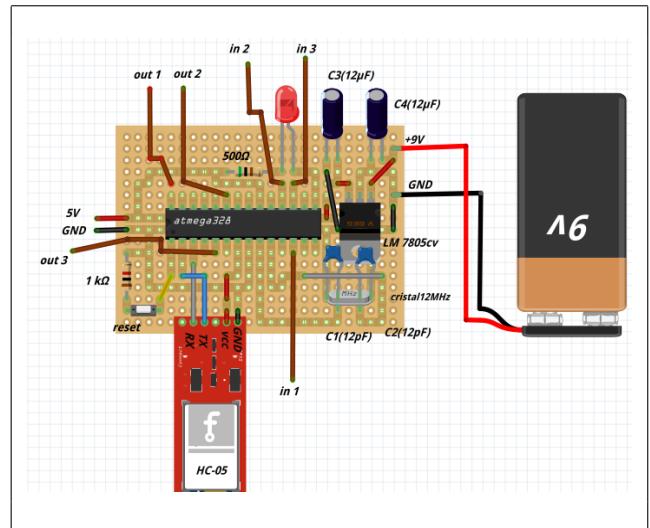


Figura 7: Esquema en tarjeta perforada del microcontrolador atmeg328, con las disposiciones del sistema de control de hardware. Figura generada en Fritzing.

El control de hardware que se observa en las figuras 7, 8 y 9, consta de un modulador de señal LM 7805cv, modulador de frecuencia con oscilador 12 Mhz, el pin IN2 es la conexión a led de color 3mm, el pin IN3 es la conexión a led de color 3mm, pin de salida a +5v, pin de salida a GND, modulo bluetooth hc-05, botón de reinicio, microcontrolador atmega 328P-PU, el pin OUT1 es la salida al sensor de electro-recepción activa, el pin OUT2 es la salida al sensor de electro-recepción pasiva, el pin OUT3 es el pin de salida al sistema de control de velocidad, el pin IN1 es la salida al diodo emisor infrarrojo; la placa perforada tiene dimensiones de  $6 \times 4,5 \text{ cm}^2$ , con un acabado en velcro, lo que facilita el montaje de la misma en el vehículo.

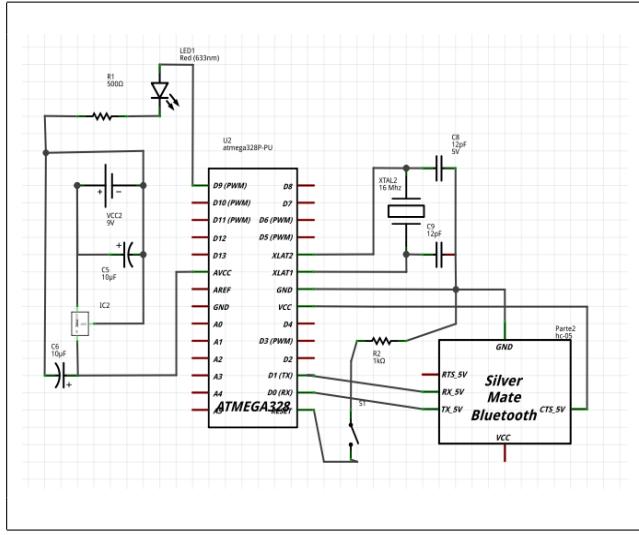


Figura 8: Esquema eléctrico microcontrolador atmetga 328, con las disposiciones del sistema de control de hardware. Figura generada en Fritzing.

### 2.1.2 Control de avance

El sistema de control de avance ver figura 26 , es el que da el movimiento del vehículo únicamente hacia adelante, pues no consta de un sistema de reversa, esta peculiaridad del vehículo permite que este avance una cantidad mínima de distancia para la recolección de datos de los distintos sensores de electro-recepción que dispone el vehículo.

El sistema de control en el avance del vehículo infrarossi como se observa en las figuras 10 , 11 y 12 , consta de dos salidas a GND, dos salidas a +5V, una entrada a GND, una entrada a +5V, un tip 122, una resistencia de  $5k\Omega$ , diodo 1N4001, el pin OUT3 es la salida al pin D3(PWM) del microcontrolador atmega, salida M1 para el motor, salida M2 para el motor; un motor de corriente directa de medio

vatio de potencia con sistema de transmisión de engranaje de eje fijo como se observa en la figura 30.

El pin D3(PWM) del microcontrolador envía una señal de  $24mA$  al pin base del tip[4] 122, durante 37 ms, dando un avance de 2mm al vehículo motorizado infrarossi.

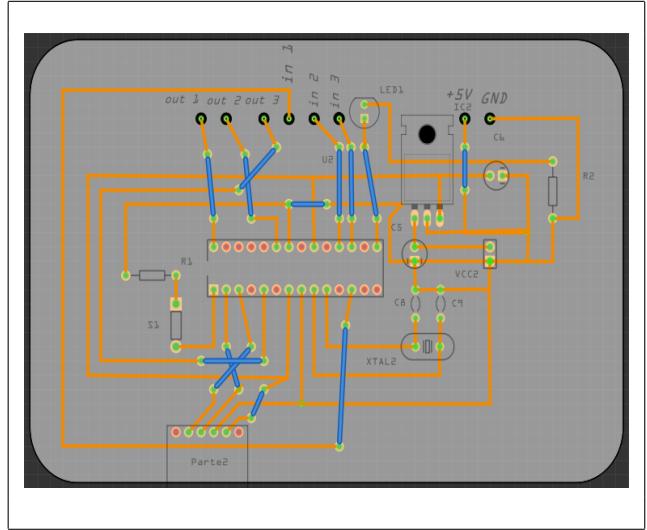


Figura 9: Esquema de tarjeta impresa para el microcontrolador atmetga 328P-PU, con las disposiciones del sistema de control de hardware. Figura generada en Fritzing.

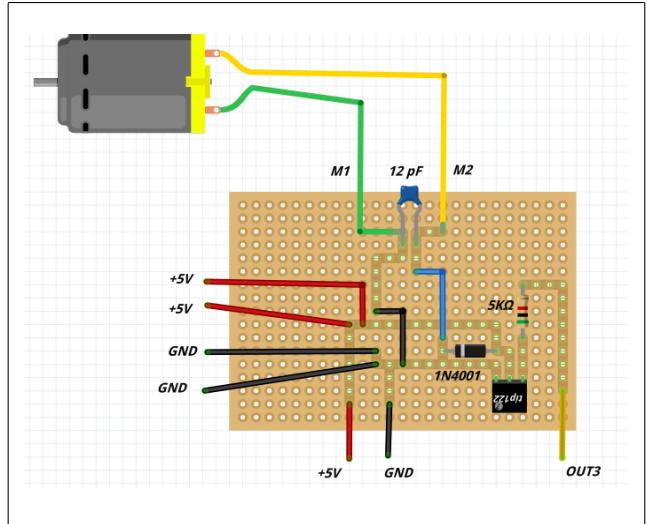


Figura 10: Esquema tarjeta perforada del sistema de control de avance. Figura generada en Fritzing.

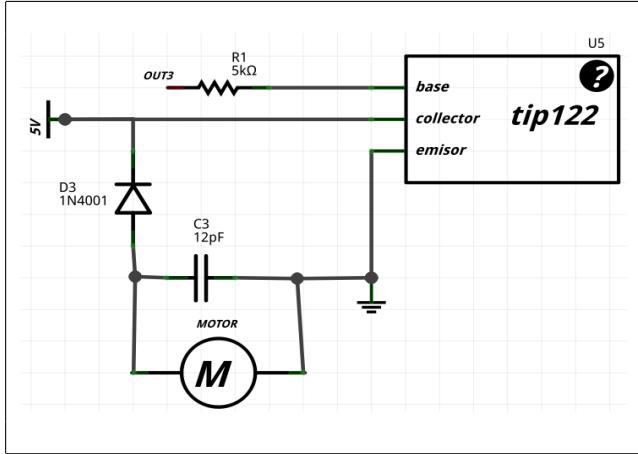


Figura 11: Esquema del circuito eléctrico del sistema de control de avance. Figura generada en Fritzing.

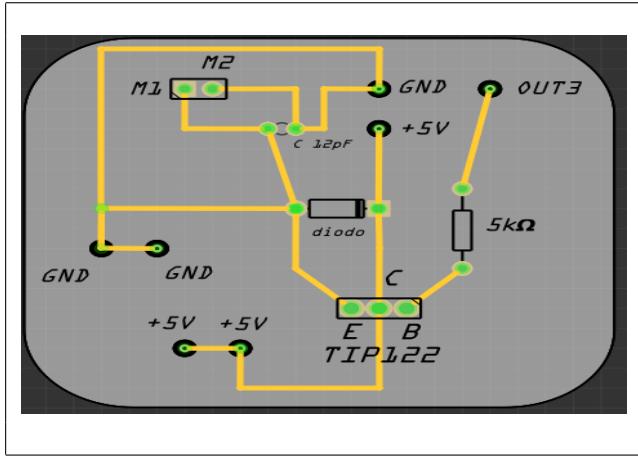


Figura 12: Tarjeta de circuito impreso el sistema de control de avance. Figura generada en Fritzing.

### 2.1.3 Sensor de electro-recepción pasiva

El sistema de electro-recepción pasiva como se observa en la figura 27, es un sistema de adquisición de información energética de radiación electromagnética en el espectro infrarrojo de manera pasiva, solo recibe señales electromagnéticas en estas longitudes de onda; la información que recibe el sensor es enviada al microcontrolador y este la envía al ordenador a través del dispositivo bluetooth para el análisis de estos datos por la interfaz gráfica free infrarossi.

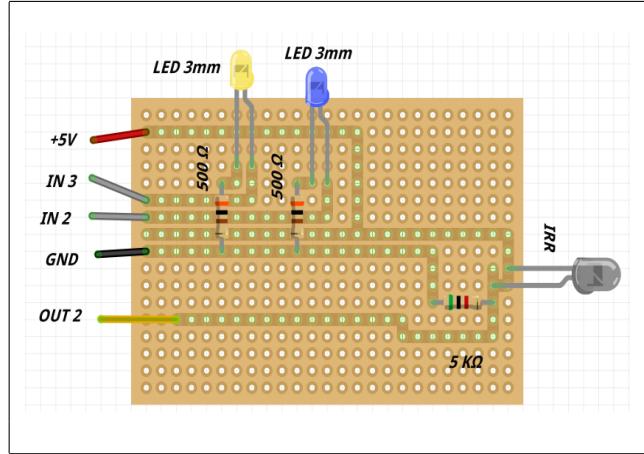


Figura 13: tarjeta perforada con el montaje del sensor de electro-recepción pasiva. Figura generada en Fritzing.

El sistema de electro-recepción pasiva como se observa en las figuras 13 y 14 , consta de un diodo receptor infrarrojo o IRR, dos led de 3mm, dos resistencias de  $500\Omega$ , una resistencia de  $5k\Omega$ , el pin IN2 es la salida al pin D10(PWM) del microcontrolador, el pin IN3 es la salida al pin D11(PWM) del microcontrolador, el pin OUT2 es la salida al pin (A0) del microcontrolador, un pin de entrada +5V y un pin de entrada a GND.

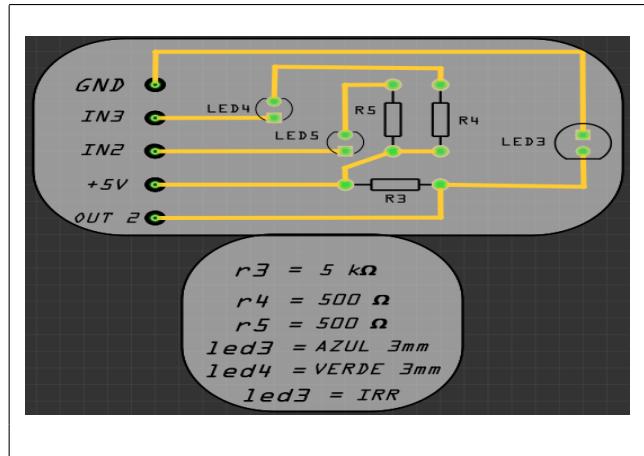


Figura 14: Placa de circuito impreso para el sensor de electro-recepción pasiva. Figura generada en Fritzing.

El sensor esta calibrado para medir  $40\mu W/mV$ , recolectando 4 datos por segundo, los diodos led son solo indicadores del proceso que esta realizando el microcontrolador y de la intensidad de señal recibida por el sensor, cantidad máxima medible  $20mW/\Omega$ .

#### 2.1.4 Sensor de electro-recepción activa

El sensor de electro-recepción activa como se observa en la figura 28 . esta basado en el principio de electro-recepción activa sin barrido, el cual consiste en un sensor de recepción por reflexión, donde una fuente fotónica ubicada al lado del sensor envía un flujo de energía electromagnética en la longitud de onda de  $940\text{nm}$  por el espacio y al encontrar una barrera que le impide el paso, esta interactúa con el obstáculo dando una reflexión del flujo irradiado en sentido contrario, fruto de esta interacción es la reducción de la intensidad de la señal. Ahora el sensor infrarrojo percibe esta señal transformando la energía lumínica en energía eléctrica, esta información es enviada al microcontrolador quien a su vez la envía por el bluetooth al software de control free infrarossi para su análisis.

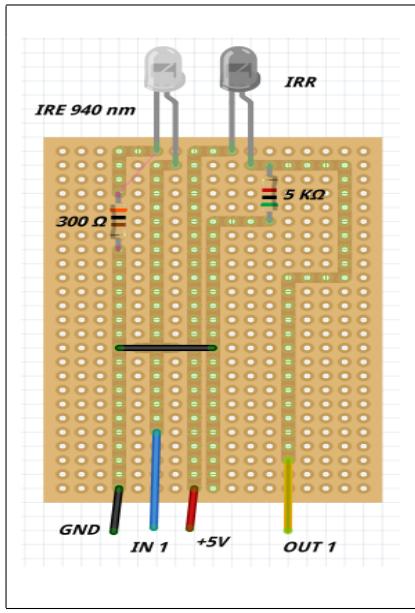


Figura 15: Esquema tarjeta perforada del radar de electro-recepción activa sin barrido, el cable de color negro es el pin de salida a tierra ( GND ), el cable de color azul es el pin de salida al pin D6(PWM) del microcontrolador, el cable de color rojo es el pin de conexión a ( $+5\text{V}$ ), el cable de color naranja es la conexión al pin A5 del microcontrolador. Figura generada en Fritzing.

El sistema de electro-recepción activa como se observa en las figuras 15 y 16 , consta de un diodo receptor infrarrojo o IRR, un diodo emisor infrarrojo o IRE, una resistencia de  $300\Omega$ , una resistencia de  $5\text{k}\Omega$ , un pin de entrada  $+5\text{V}$  y un pin de entrada a GND, el pin IN1 es la conexión al pin D6(PWM) del microcontrolador, el pin OUT1 es la salida al pin A5 del microcontrolador.

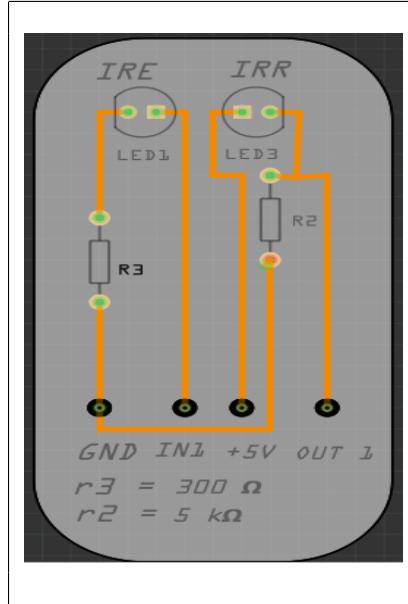


Figura 16: Placa de circuito impreso para el sensor de electro-recepción activa. Figura generada en Fritzing.

El emisor infrarrojo de longitud de onda  $940\text{nm}$  tiene una densidad de flujo radiante sobre ángulo sólido de  $20\text{mW}/\Omega$ .

El sensor de electro-recepción activa esta calibrado para medir  $40\mu\text{W}/\text{mV}$ , recolectando 4 datos por segundo, máxima intensidad medible  $20\text{mW}/\Omega$  , con máximo alcance en la distancia de detección de 60 cm a la fuentes radiante.

## 2.2 Disposición mecánica

La disposición mecánica del vehículo motorizado free infrarossi como se observa en las figuras 29 , 30 , 31 , 32 y 33 , esta compuesta por un chasis de madera, dos pares de llantas, dispone de tres bases macizas forradas en velcro, para facilitar el montaje de los diferentes sensores y sistemas de control, la base del sistema de control es hueca con espacio suficiente para la batería de  $9\text{V}$  de tipo cuadrada recargable, el montaje del vehículo, un capó de material fommy negro para aislar el ruido en el sensor de electro-recepción activa, dos ejes fijos, un sistema de transmisión de tipo engranaje y sistema de avance de 2 mm (únicamente hacia adelante), los sistemas de control se puedan quitar y poner en el vehículo debido al material de velcro, el sistema de control aparte de ser móvil, puede ser removido el microcontrolador y programado con futuras actualizaciones del firmware que provee el software free infrarossi para el vehículo motorizado infrarossi y ampliar su funcionalidad como instrumento de laboratorio, estos

sensores de adquisición de datos no solo sirven para ilustrar la propiedad de difracción, atenuación y absorción de las ondas electromagnéticas, sino que por su naturaleza activa y pasiva, podrían ser utilizados para ilustrar la magnitud de la aceleración gravitacional terrestre, las características en el movimiento de un péndulo, entre otras.

### 3 Control de hardware a través del software free infrarossi

#### 3.1 ¿Qué es free infrarossi?

Free infrarossi es un programa creado en Colombia, en la ciudad de Bogotá, como tesis de grado para optar por el título de licenciado en física en la Universidad Distrital Francisco José de Caldas, el cual fue desarrollado con software libre y en un entorno libre como lo es GNU-Linux, no es multiplataforma, fue diseñado únicamente para este sistema operativo; se vale de lenguajes de programación, como C, C++, python 2.7, bash; y de programas como arduino, octave, latex, gnuplot, blueman-manager entre otros, fusionados en una interfaz amigable y fácil de utilizar.

Free infrarossi es el laboratorio virtual del instrumento de laboratorio infrarossi que ilustra la propiedad de difracción, atenuación y absorción de ondas electromagnéticas en el espectro infrarrojo, esta diseñado para ser utilizado tanto por estudiantes como docentes de muy diversas ramas de las ciencias y la ingeniería o como una herramienta muy útil para los educadores y alumnos de media vocacional.

#### 3.2 Licencia

Programa de control de hardware e ilustración física de las propiedades de las ondas electromagnéticas en el espectro infrarrojo.

Copyright (C) 2016-01-01 Universidad Distrital Francisco Jose, Diego Alberto Parra Garzón, Dr. Julian Andres Salamanca Bernal.

El programa free infrarossi es software libre; puedes redistribuirlo y / o modificarlo bajo los términos de la Licencia Pública General GNU publicada por la Fundación para el Software Libre; ya sea la versión 3 de la Licencia, o (a su elección) cualquier versión posterior.

Este programa se distribuye con la esperanza de que sea útil,

pero SIN NINGUNA GARANTÍA; ni siquiera la garantía implícita de COMERCIALIZACIÓN o IDONEIDAD PARA UN PROPÓSITO PARTICULAR. Vea la Licencia Pública General GNU para más detalles.

Debería haber recibido una copia de la Licencia Pública General de GNU junto con este programa; si no, escriba a la Free Software Foundation, Inc., 51 Franklin Street, Quinto Piso, Boston, MA 02110-1301 EE.UU..

Si usted hace alguna modificación en esta aplicación, deberá siempre mencionar el autor original de la misma.

#### 3.3 Instalación

La instalación de este software es el mismo para distribuciones basadas en Debian<sup>2</sup>.

- a. Abrir una terminal<sup>3</sup> como se observa en la figura 17 y escribir sin comillas "sudo su" y luego oprimir la tecla enter.
- b. Escribir la contraseña de administrador, presione enter.
- c. Escribir sin comillas "aptitude install -y git", presione enter.
- d. Escribir sin comillas "cd Documentos" presione enter.
- e. Escribir sin comillas "git clone https://github.com/Nombre\_Usuario/Free-infrarossi", oprimir enter.
- f. Escribir sin comillas "chmod +777 Free-infrarossi", presione la tecla enter.
- g. Cerrar la terminal y dirigirse al navegador de archivos en la carpeta Documentos/Free-infrarossi/install, como se observa en la figura 18 .
- h. Dar click derecho del mouse en el archivo instalador.py como se observa en las figuras 19 y 20 , click izquierdo en la opción propiedades; en la pestaña de general debe decir abrir archivo con python 2.7 y en la pestaña permisos debe estar seleccionada la casilla

<sup>2</sup>Página oficial proyecto Debian <https://www.debian.org/>

<sup>3</sup>Procedimiento: <http://www.comoinstalarlinux.com/como-abrir-una-terminal-en-ubuntu-linux-mint-centos-debian/>

permitir ejecutar este archivo como programa, de no ser cambiar las opciones.

- i. Cerrar la ventana y abrir el archivo INSTALADOR.py.
- j. Escribir la clave administrador y presionar enter.
- k. Escribir 1 y presionar enter.
- l. Escribir nuevamente 1 y enter.
- m. Una vez finalizada la instalación hay que reiniciar el pc.

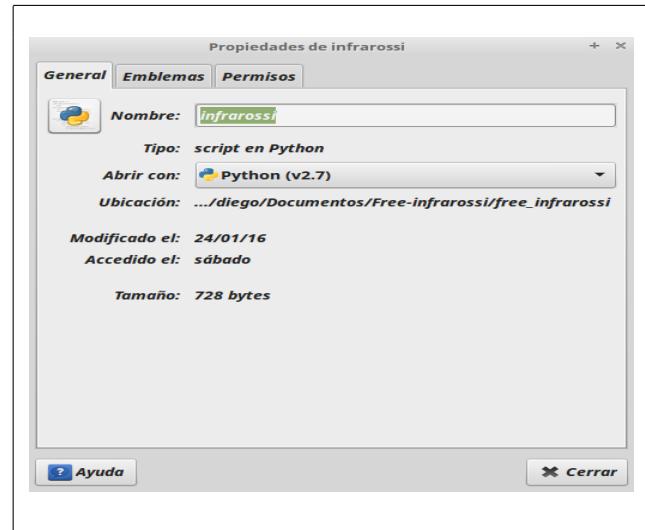


Figura 19: Instalación, propiedades free infrarossi

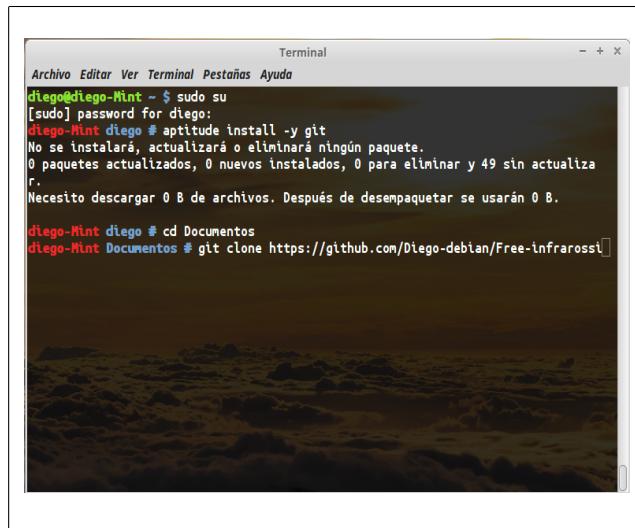


Figura 17: Instalación, comandos en terminal de Linux mint 17.2

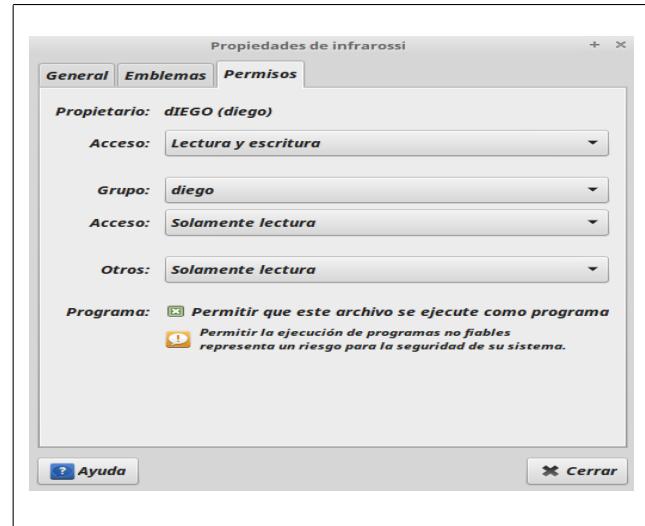


Figura 20: Instalación, propiedades free infrarossi



Figura 18: Instalación, vista navegador de archivos.

### 3.4 Primer uso free infrarossi

Abrir una terminal del S.O. escribir en la terminal sin comillas '*infrarossi*' , oprimir la tecla enter, escribir la clave de administrador y se despliega la ventana del software free infrarossi, como se aprecia en la figura 21.



Figura 21: Instalación, comandos en terminal de Linux mint 17.2

### 3.5 Carga de firmware en integrado atmega 328

La ventana del programa free infrarossi, en la parte inferior derecha hay tres botones, oprimir el botón de firmware como se muestra en la figura 22 , y conectar la tarjeta micro controladora arduino uno al pc; oprimir el botón continuar y esperar que cargue el firmware en la tarjeta, una vez hecho esto retirar el micro controlador de la tarjeta y colocarlo en el montaje del vehículo.

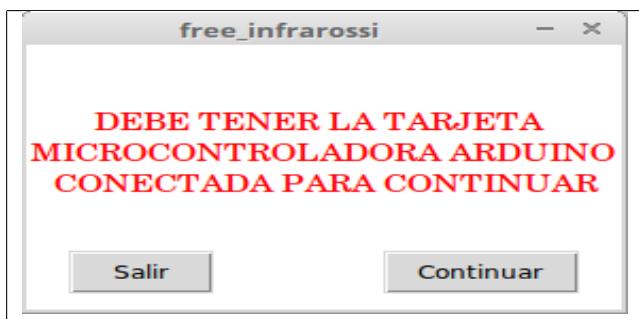


Figura 22: Ventana de instalación del firmware, que ofrece el software free infrarossi, para el vehículo motorizado infrarossi.

### 3.6 Test del instrumento infrarossi y su software de control free infrarossi

C.7

## Referencias

- [1] Pedraja, F. (2012). Modelo computacional de *Gymnotus omarorum*: Un pez eléctrico de pulso con órgano distribuido.
- [2] Atmel. (2015). Atmel 8-BIT Microcontroller with 4/8/16/32KBYTES in-system programmable flash. atmega328P-PU data sheet [On line]. Disponible en: [http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Summary.pdf](http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf)
- [3] Semiconductor, F. (2012). LM7805 Data Sheet. [Online]. Disponible en: <http://pdf.datasheetcatalog.net/datasheet/fairchild/LM7805.pdf>
- [4] Semiconductor, F. (2008). Tip120/tip121/tip122 npn epitaxial darlington transistor. TIP120 datasheet, Oct. [On line]. Disponible en: <http://pdf.datasheetcatalog.net/datasheet/fairchild/TIP122.pdf>.
- [5] Phototransistor T1. (2006). Pt333-3B Data Sheet. [Online]. Disponible en: <http://media.digikey.com/pdf/Data%20Sheets/Everlight%20PDFs/PT333-3B.pdf>
- [6] Knörig, A., Wettach, R., & Cohen, J. (2009, February). Fritzing: a tool for advancing electronic prototyping for designers. In Proceedings of the 3rd International Conference on Tangible and Embedded Interaction. ACM.
- [7] Gift, N., & Jones, J. M. (2008). Python for Unix and Linux system administration. “O'Reilly Media, Inc.”.
- [8] Alexander, C. K., Sadiku, M. N., Bermúdez, A. V., & Pedraza, C. R. C. (2006). Fundamentos de circuitos eléctricos. McGraw-Hill.

## Anexos

A Diseño eléctrico

#### A.1 Esquema eléctrico equivalente del vehículo

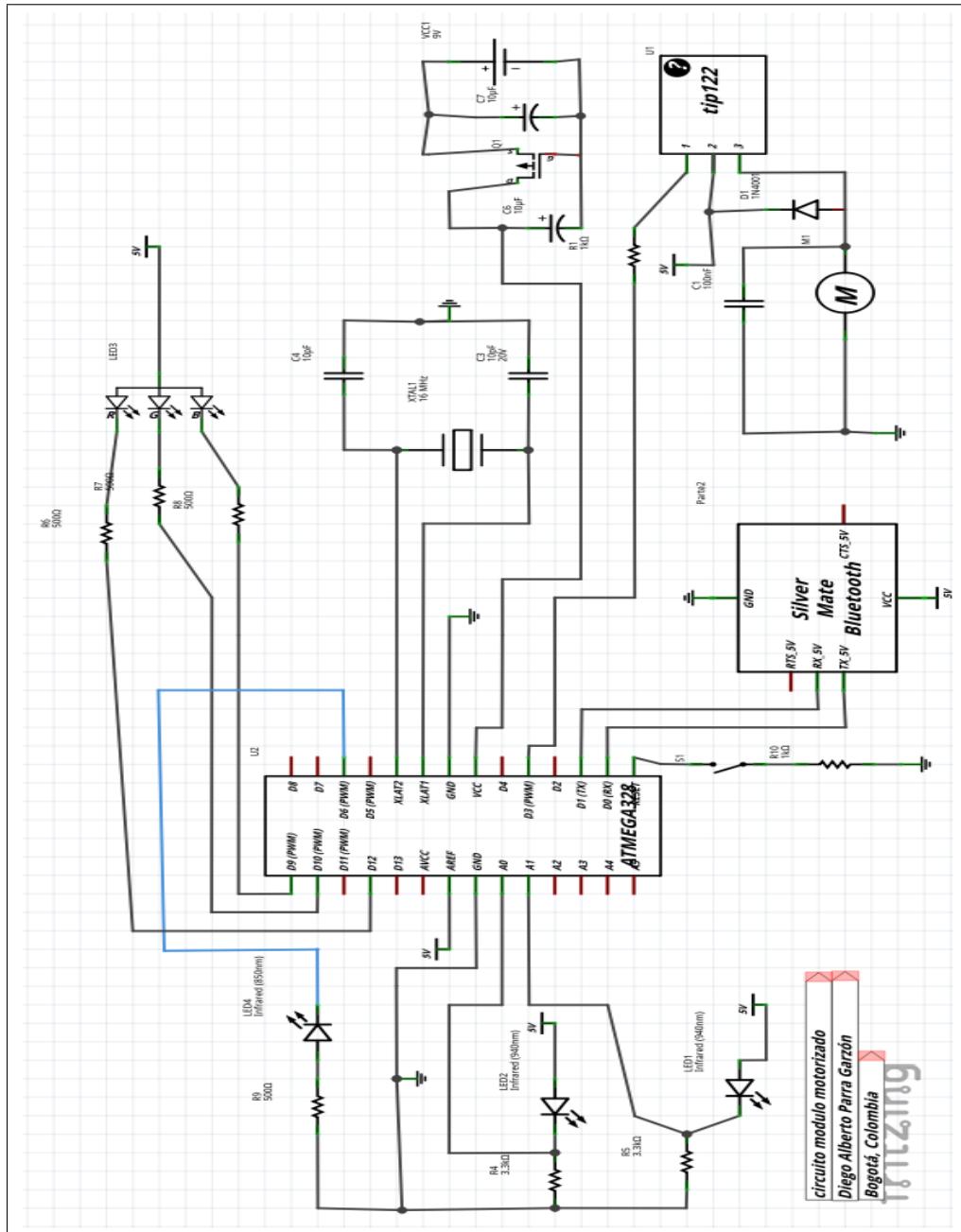


Figura 23: Esquema eléctrico equivalente del vehículo.

## A.2 Montaje eléctrico en protoboard

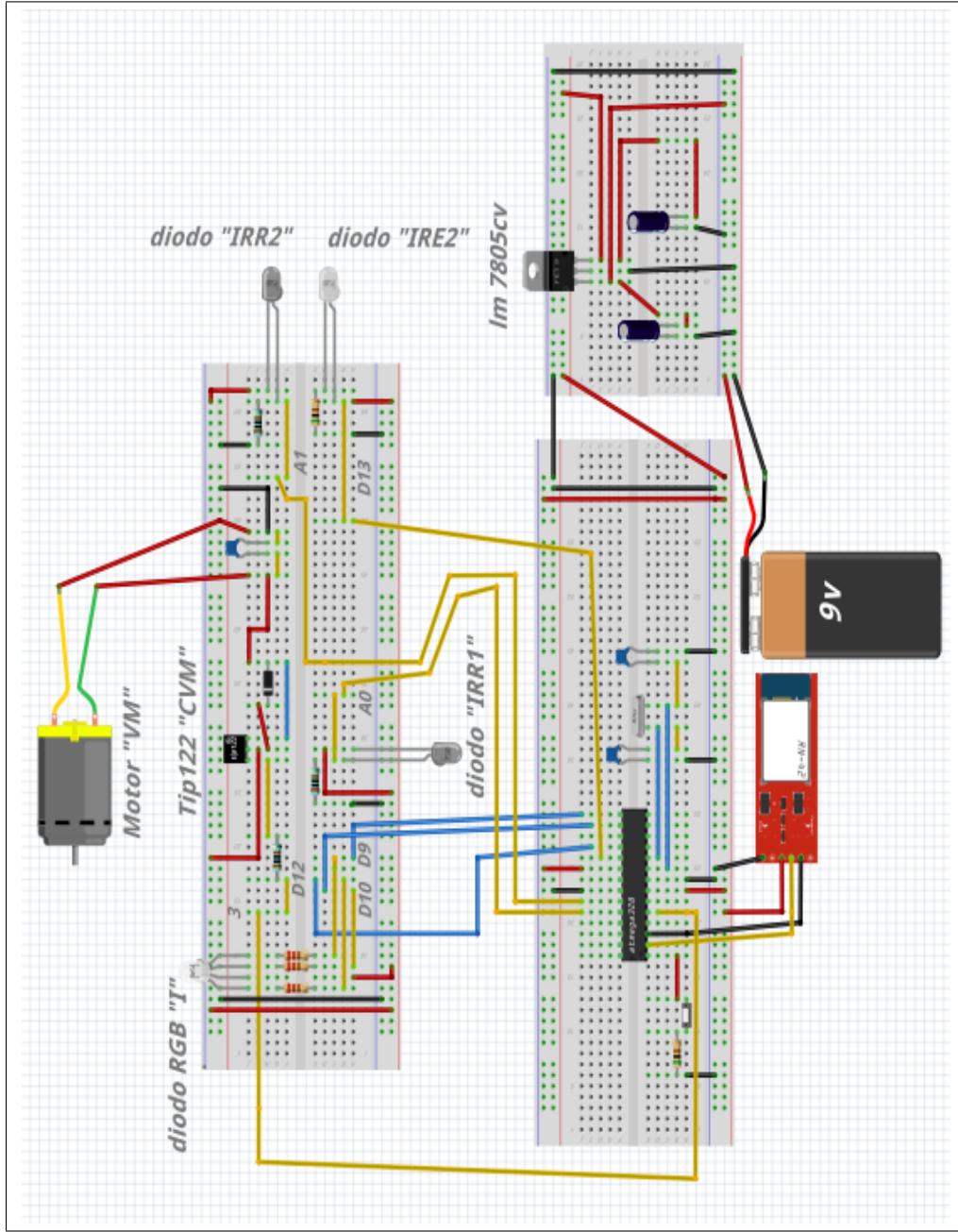


Figura 24: Montaje en protoboard.

### A.3 Montaje eléctrico, sistema control de hardware

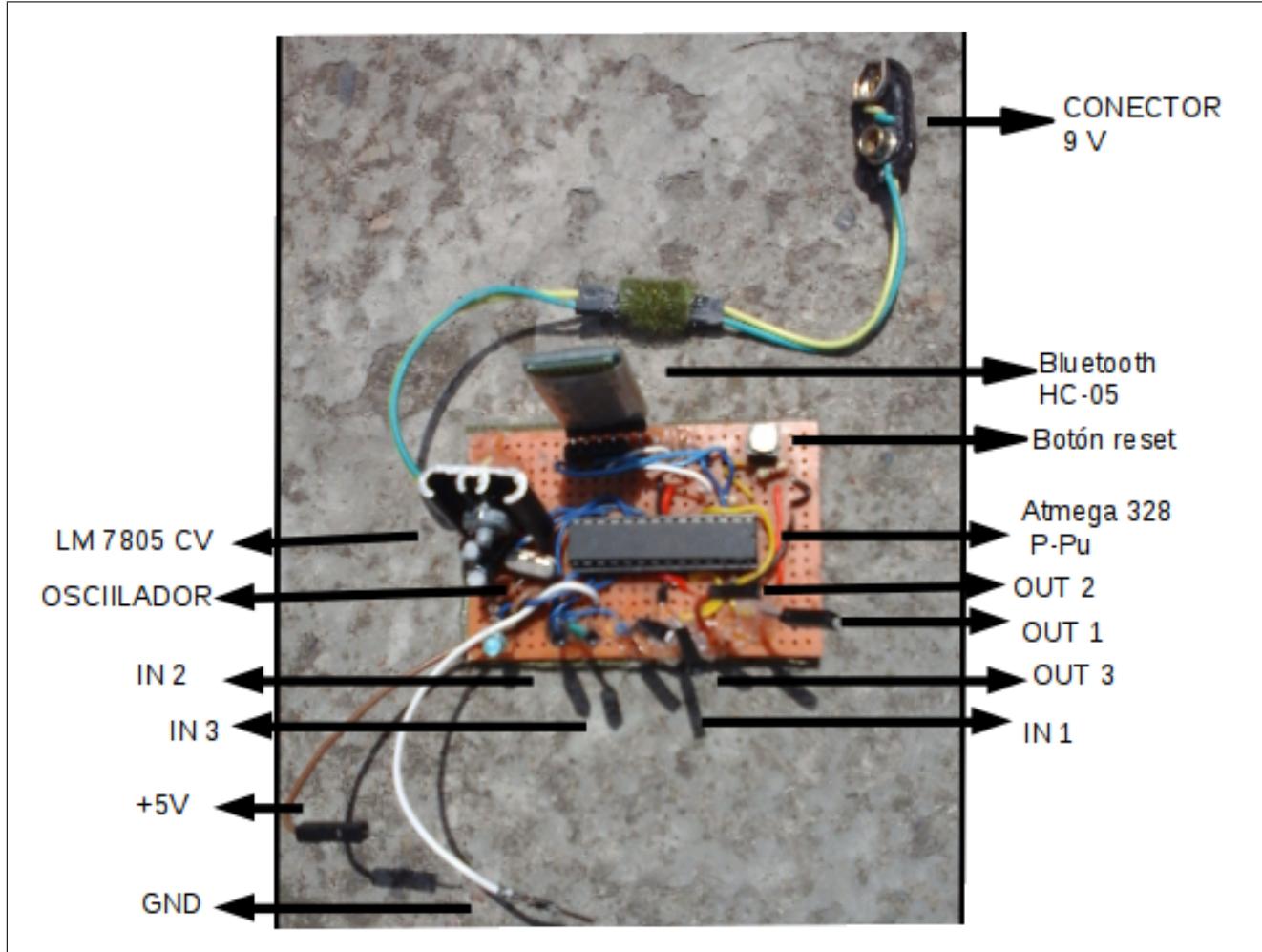


Figura 25: Sistema control de hardware, con modulador de señal LM 7805cv, modulador de frecuencia con oscilador 12 Mhz, el pin IN2 es la conexión a led de color 3mm, el pin IN3 es la conexión a led de color 3mm, pin de salida a +5v, pin de salida a GND, modulo bluetooth hc-05, botón de reinicio, microcontrolador atmega 328P-PU, el pin OUT1 es la salida al sensor de electro-recepción activa, el pin OUT2 es la salida al sensor de electro-recepción pasiva, el pin OUT3 es el pin de salida al sistema de control de velocidad, el pin IN1 es la salida al diodo emisor infrarrojo.

#### A.4 Montaje eléctrico, sistema control de avance del vehículo infrarossi

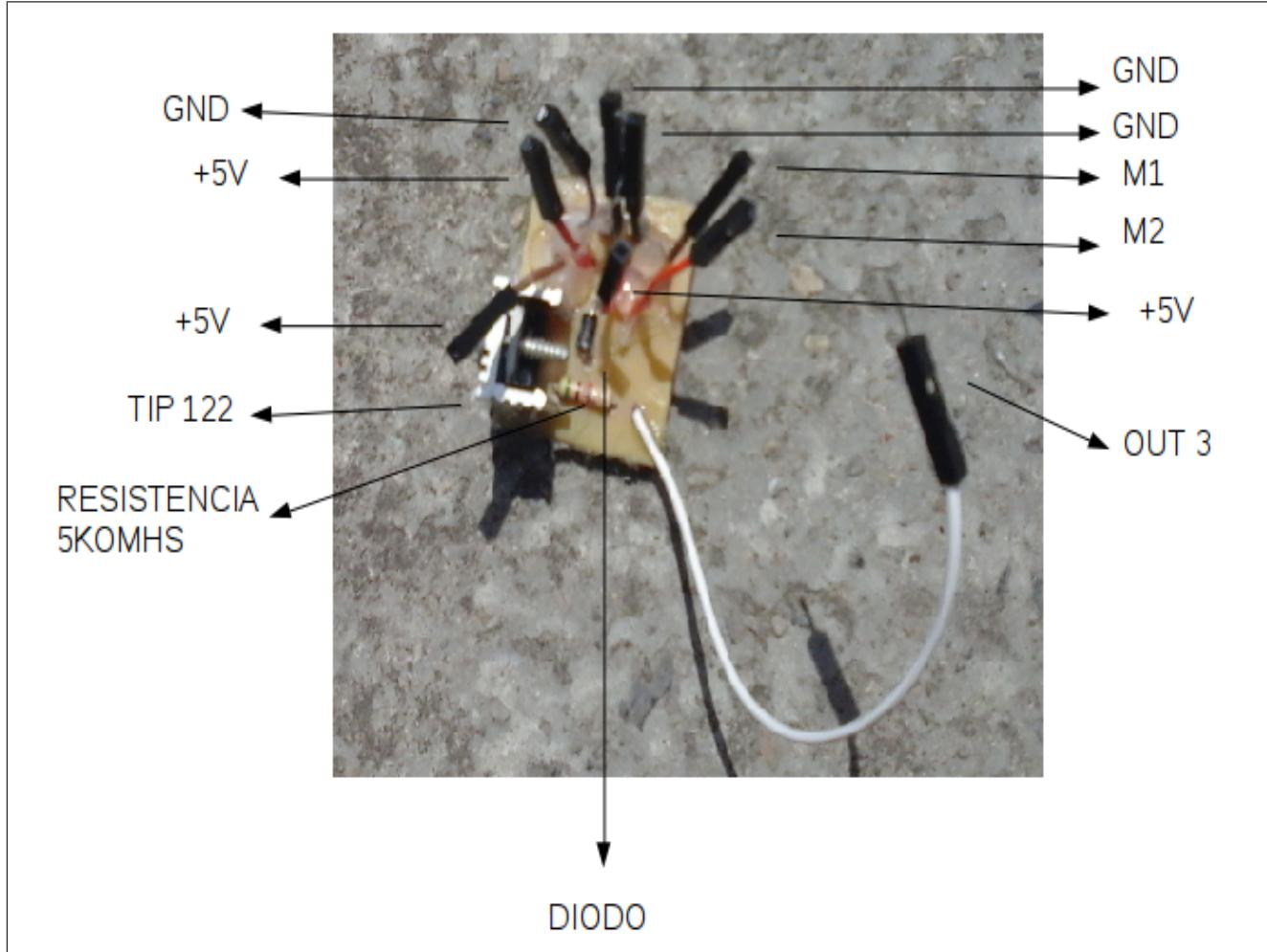


Figura 26: Sistema control de avance del vehículo infrarossi, consta de dos salidas a GND, dos salidas a +5V, una entrada a GND, una entrada a +5V, un tip 122, una resistencia de  $5k\Omega$ , diodo 1N4001, el pin OUT3 es la salida al pin D3(PWM) del microcontrolador atmega, salida M1 para el motor, salida M2 para el motor.

#### A.5 Montaje eléctrico, sensor electro-recepción pasiva

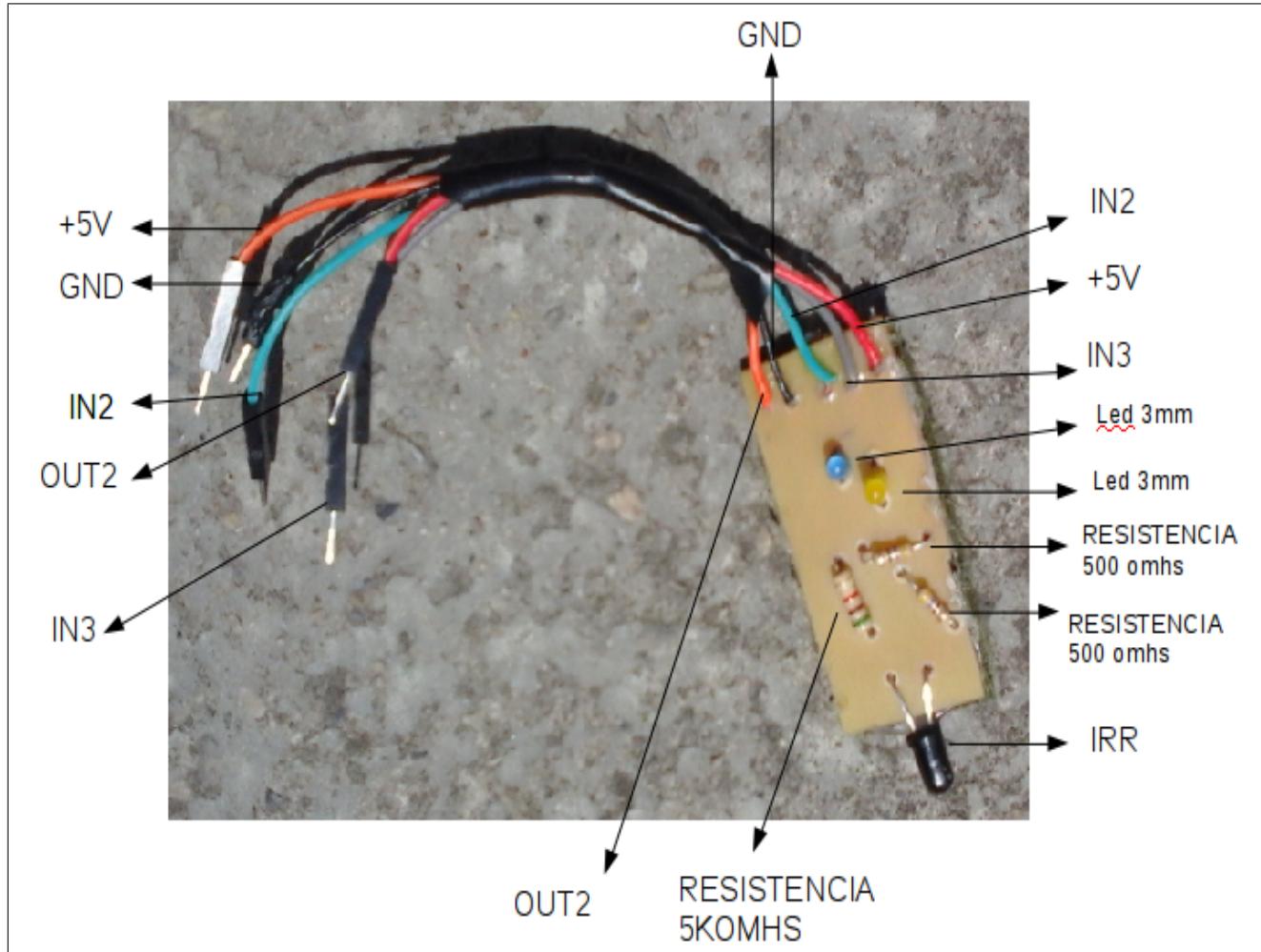


Figura 27: El sistema de electro-recepción pasiva consta de un diodo receptor infrarrojo o IRR, dos led de 3mm, dos resistencias de  $500\Omega$ , una resistencia de  $5k\Omega$ , el pin IN2 es la salida al pin D10(PWM) del microcontrolador, el pin IN3 es la salida al pin D11(PWM) del microcontrolador, el pin OUT2 es la salida al pin (A0) del microcontrolador, un pin de entrada +5V y un pin de entrada a GND.

#### A.6 Montaje eléctrico, sensor electro-recepción activa

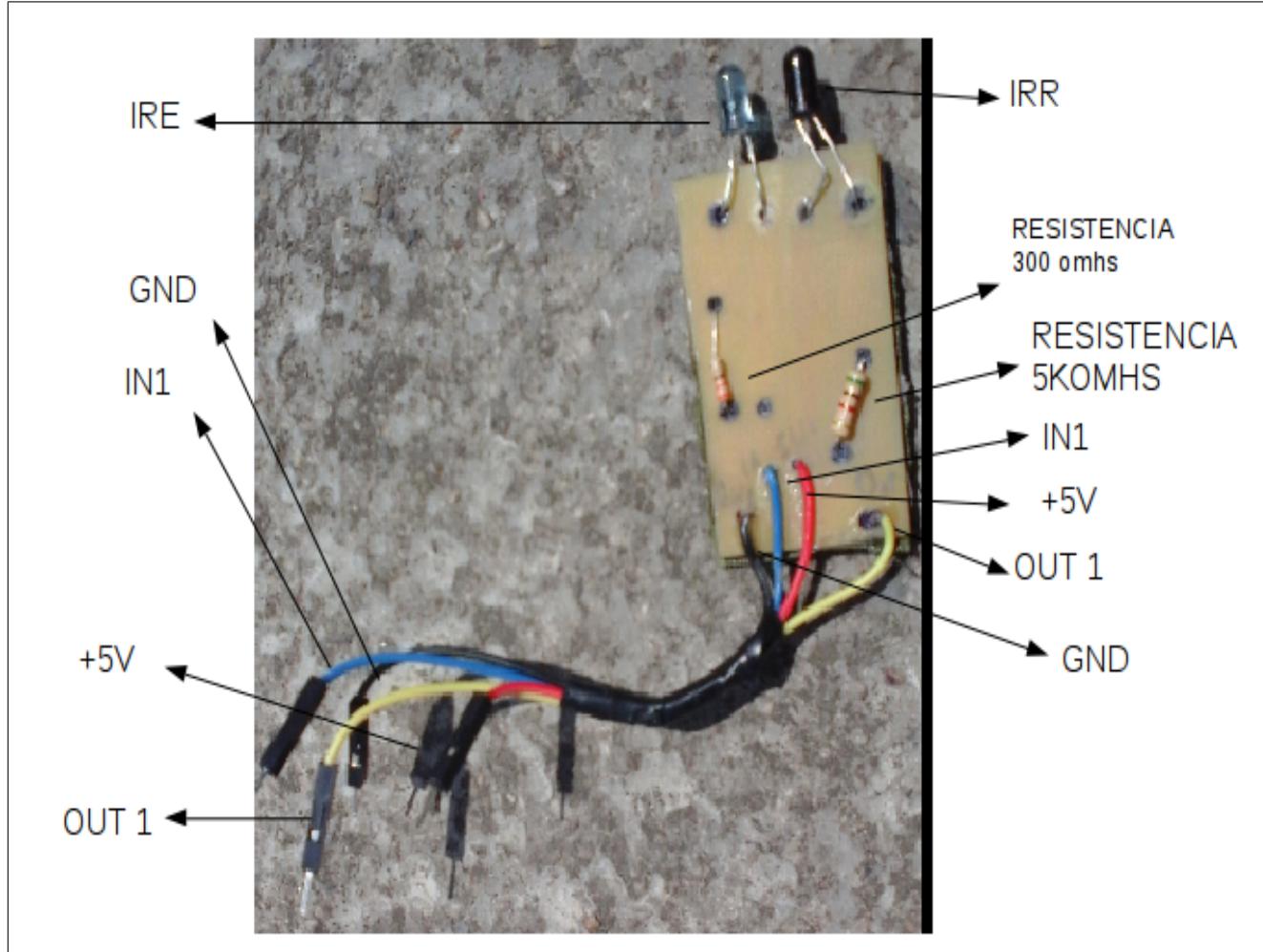


Figura 28: El sistema de electro-recepción activa consta de un diodo receptor infrarrojo o IRR, un diodo emisor infrarrojo o IRE, una resistencia de  $300\Omega$ , una resistencia de  $5k\Omega$ , un pin de entrada +5V y un pin de entrada a GND, el pin IN1 es la conexión al pin D6(PWM) del microcontrolador, el pin OUT1 es la salida al pin A5 del microcontrolador.

## B Diseño mecánico

### B.1 Montaje mecánico, dimensiones del chasis.

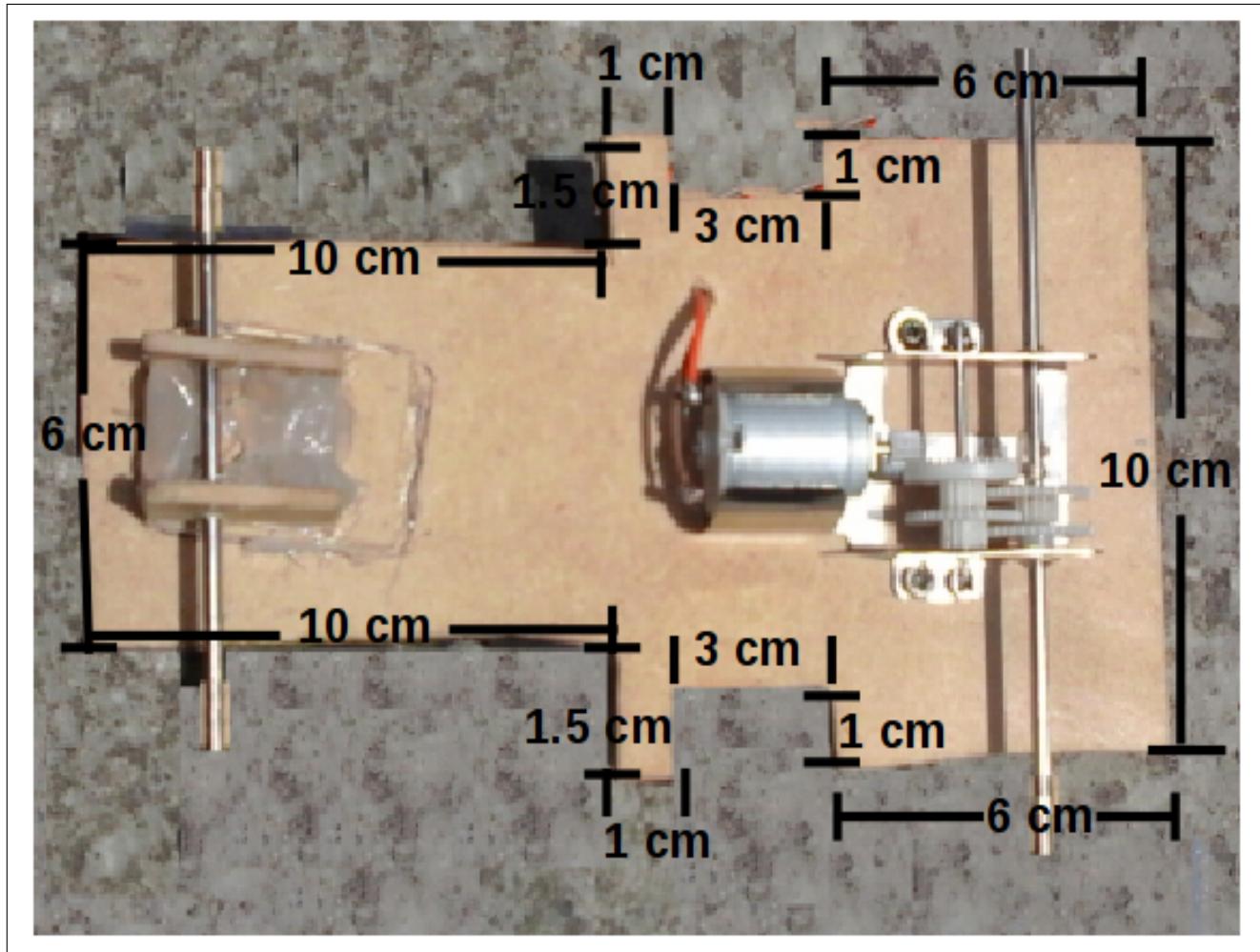


Figura 29: Chasis de madera con dimensiones en centímetros, la caja del eje delantero tiene dimensiones dos centímetros de ancho, 3 centímetros de largo y 2,5 centímetros de alto; el eje delantero debe estar a 1 centímetro de separación del chasis.

## B.2 Montaje mecánico, vista inferior

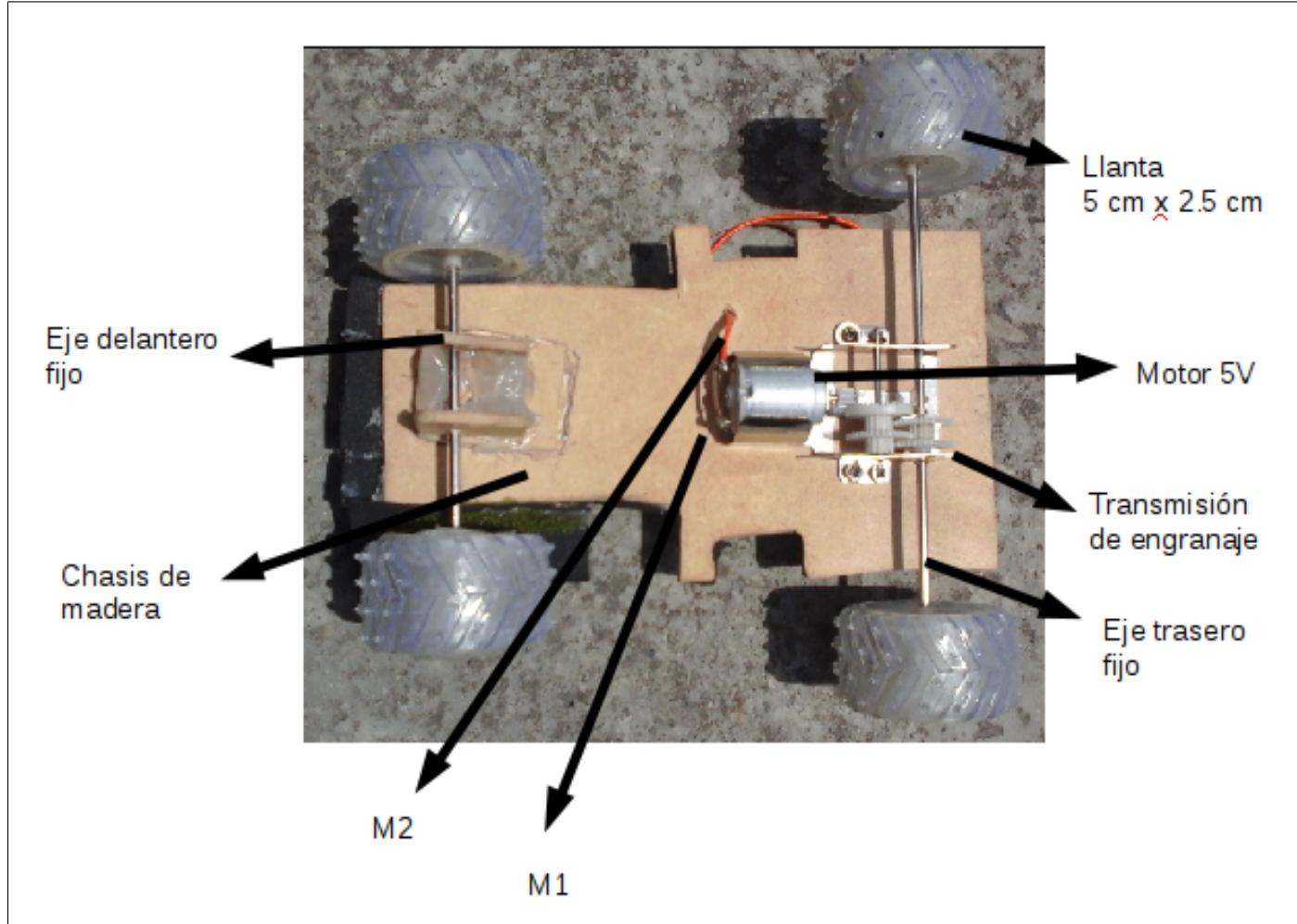


Figura 30: Vista inferior del vehículo con eje delantero fijo, eje trasero fijo, transmisión de engranaje, motor de 5 V, llantas de 5 cm de diámetro por 2.5 de ancho, conexión del motor M1 y conexión del motor M2.

### B.3 Montaje mecánico, vista superior

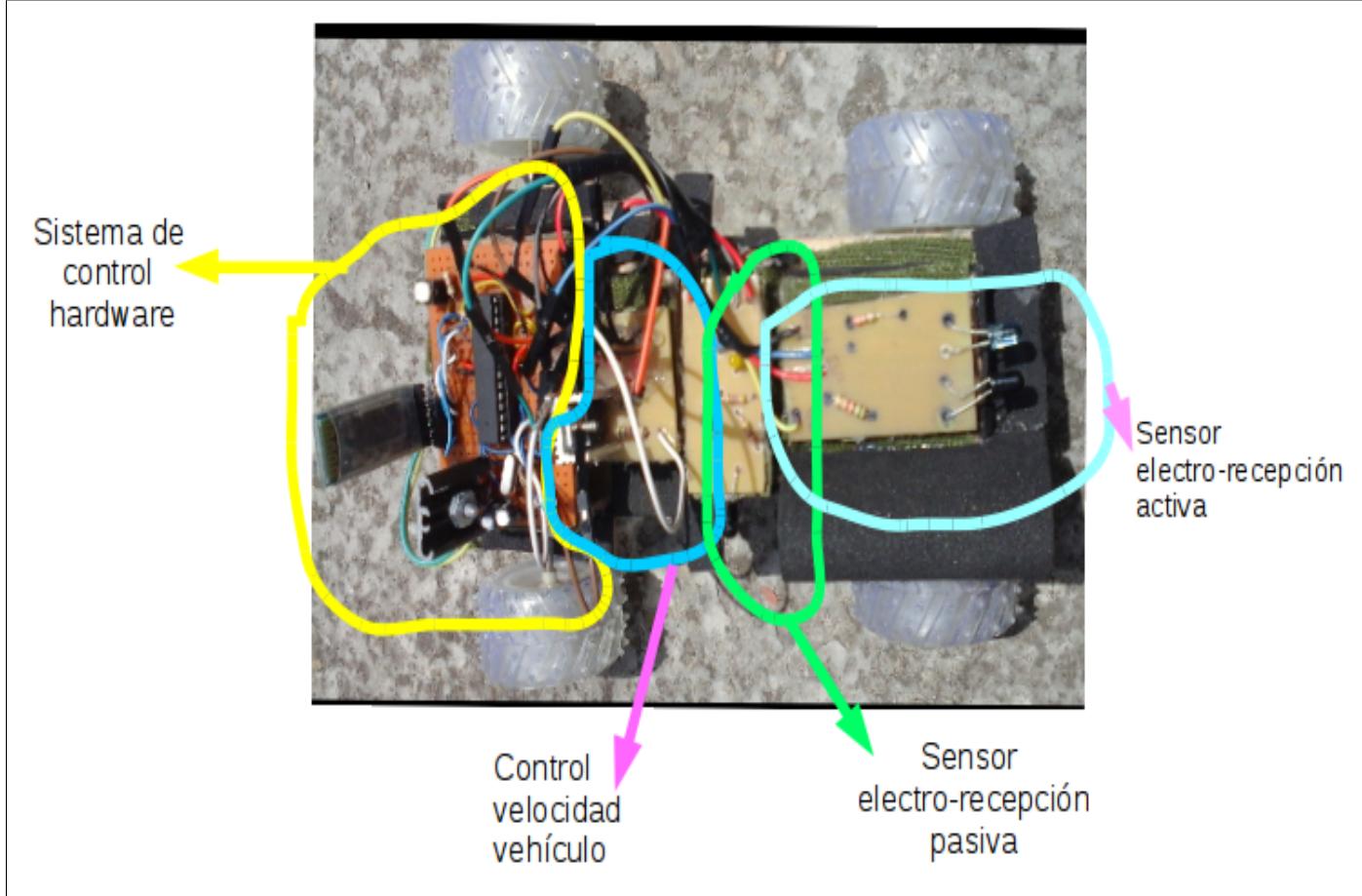


Figura 31: Vista superior del vehículo donde se observa el sistema de control de hardware, el sistema de control de avance del vehículo, el sensor de electro-recepción pasiva y el sensor de electro-recepción activa.

#### B.4 Montaje mecánico, vista superior del chasis.

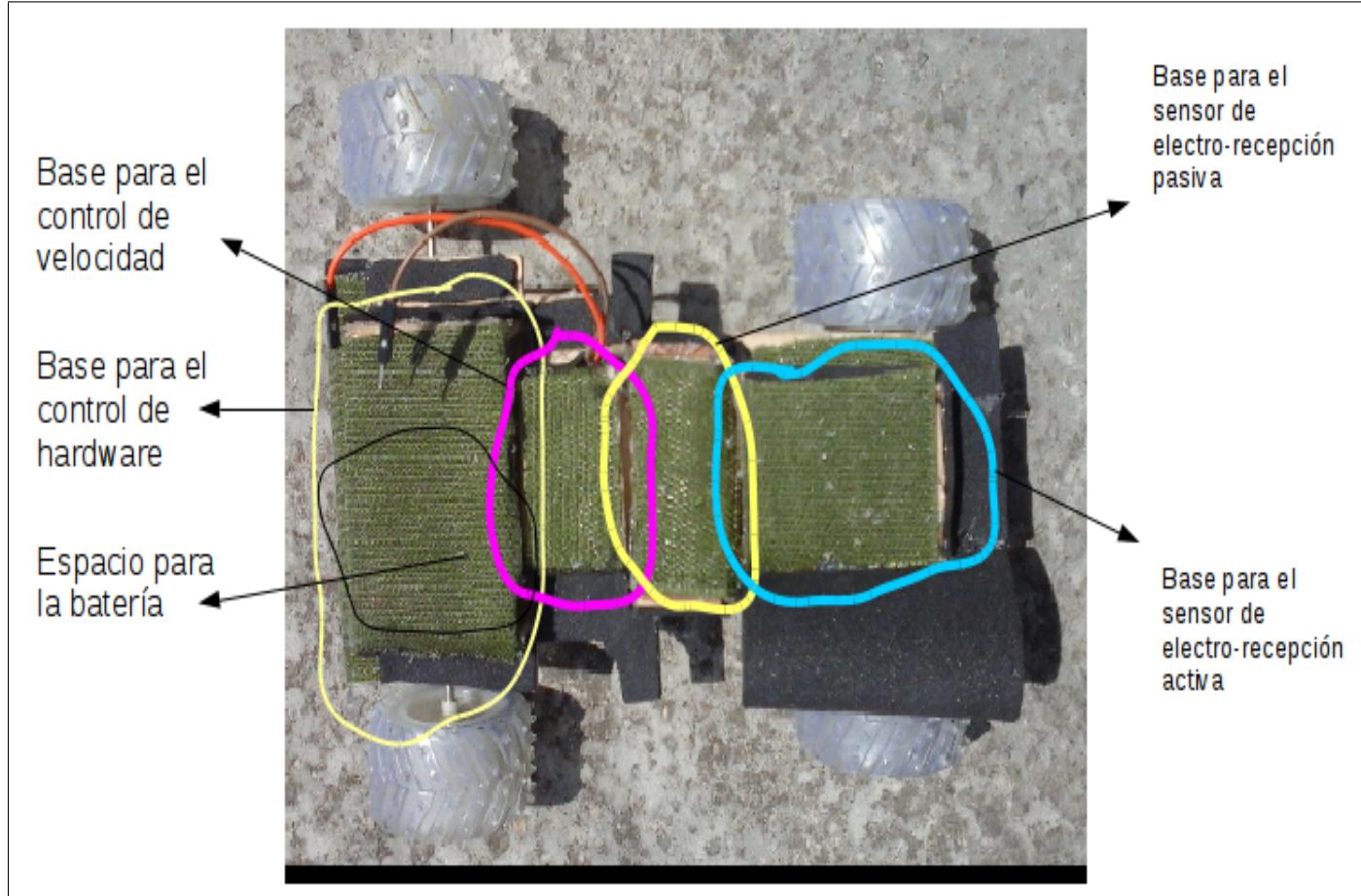


Figura 32: Chasis vista superior las bases tienen las mismas dimensiones de los sensores tanto de ancho como de largo, el alto de las bases de 1.5 centímetros, con velcro, la base para el control de hardware tiene 2.3 cm de alto pues debajo de esta hay un espacio para guardar la batería de 9 voltios.

### B.5 Montaje mecánico.

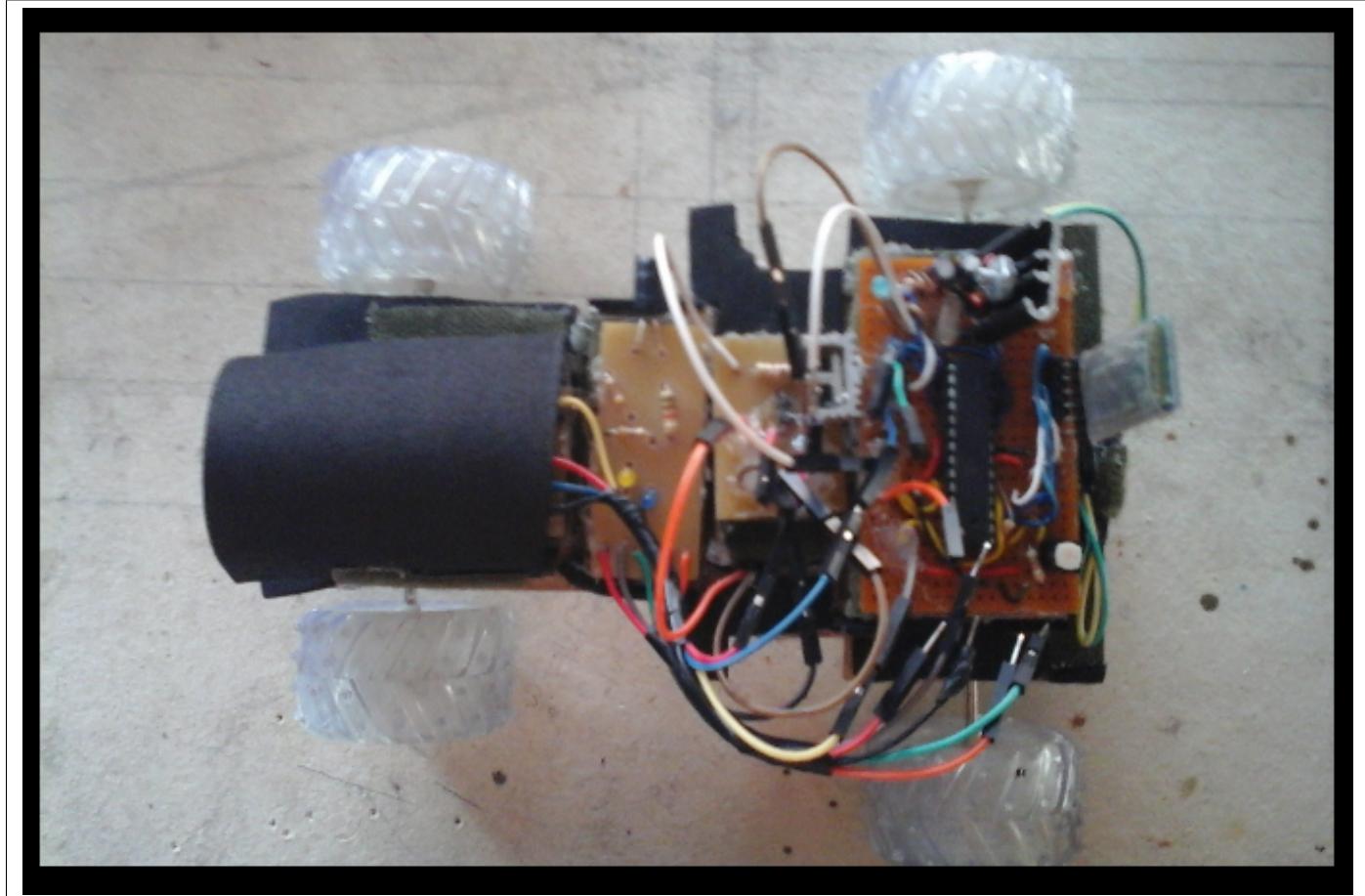


Figura 33: Chasis vista superior las bases tienen las mismas dimensiones de los sensores tanto de ancho como de largo, el alto de las bases es de 1.5 centímetros, con velcro, la base para el control de hardware tiene 2.3 cm de alto pues debajo de esta hay un espacio para guardar la batería de 9 voltios.

## C Códigos del programa free infrarrosi

### C.1 LICENSE

Este código comienza en la pagina que sigue.



56 have designed this version of the GPL to prohibit the practice for those  
57 products. If such problems arise substantially in other domains, we  
58 stand ready to extend this provision to those domains in future versions  
59 of the GPL, as needed to protect the freedom of users.  
60

61 Finally, every program is threatened constantly by software patents.  
62 States should not allow patents to restrict development and use of  
63 software on general-purpose computers, but in those that do, we wish to  
64 avoid the special danger that patents applied to a free program could  
65 make it effectively proprietary. To prevent this, the GPL assures that  
66 patents cannot be used to render the program non-free.  
67

68 The precise terms and conditions for copying, distribution and  
69 modification follow.  
70

## 71 TERMS AND CONDITIONS

### 72

#### 73 0. Definitions.

74 "This License" refers to version 3 of the GNU General Public License.  
75

76 "Copyright" also means copyright-like laws that apply to other kinds of  
77 works, such as semiconductor masks.  
78

79 "The Program" refers to any copyrightable work licensed under this  
80 License. Each licensee is addressed as "you". "Licensees" and  
81 "recipients" may be individuals or organizations.  
82

83 To "modify" a work means to copy from or adapt all or part of the work  
84 in a fashion requiring copyright permission, other than the making of an  
85 exact copy. The resulting work is called a "modified version" of the  
86 earlier work or a work "based on" the earlier work.  
87

88 A "covered work" means either the unmodified Program or a work based  
89 on the Program.  
90

91 To "propagate" a work means to do anything with it that, without  
92 permission, would make you directly or secondarily liable for  
93 infringement under applicable copyright law, except executing it on a  
94 computer or modifying a private copy. Propagation includes copying,  
95 distribution (with or without modification), making available to the  
96 public, and in some countries other activities as well.  
97

98 To "convey" a work means any kind of propagation that enables other  
99 parties to make or receive copies. Mere interaction with a user through  
100 a computer network, with no transfer of a copy, is not conveying.  
101

102 An interactive user interface displays "Appropriate Legal Notices"  
103 to the extent that it includes a convenient and prominently visible  
104 feature that (1) displays an appropriate copyright notice, and (2)  
105 tells the user that there is no warranty for the work (except to the  
106 extent that warranties are provided), that licensees may convey the  
107 work under this License, and how to view a copy of this License. If  
108 the interface presents a list of user commands or options, such as a  
109 menu, a prominent item in the list meets this criterion.  
110

111  
112     1. Source Code.  
113  
114     The "source code" for a work means the preferred form of the work  
115     for making modifications to it. "Object code" means any non-source  
116     form of a work.  
117  
118     A "Standard Interface" means an interface that either is an official  
119     standard defined by a recognized standards body, or, in the case of  
120     interfaces specified for a particular programming language, one that  
121     is widely used among developers working in that language.  
122  
123     The "System Libraries" of an executable work include anything, other  
124     than the work as a whole, that (a) is included in the normal form of  
125     packaging a Major Component, but which is not part of that Major  
126     Component, and (b) serves only to enable use of the work with that  
127     Major Component, or to implement a Standard Interface for which an  
128     implementation is available to the public in source code form. A  
129     "Major Component", in this context, means a major essential component  
130     (kernel, window system, and so on) of the specific operating system  
131     (if any) on which the executable work runs, or a compiler used to  
132     produce the work, or an object code interpreter used to run it.  
133  
134     The "Corresponding Source" for a work in object code form means all  
135     the source code needed to generate, install, and (for an executable  
136     work) run the object code and to modify the work, including scripts to  
137     control those activities. However, it does not include the work's  
138     System Libraries, or general-purpose tools or generally available free  
139     programs which are used unmodified in performing those activities but  
140     which are not part of the work. For example, Corresponding Source  
141     includes interface definition files associated with source files for  
142     the work, and the source code for shared libraries and dynamically  
143     linked subprograms that the work is specifically designed to require,  
144     such as by intimate data communication or control flow between those  
145     subprograms and other parts of the work.  
146  
147     The Corresponding Source need not include anything that users  
148     can regenerate automatically from other parts of the Corresponding  
149     Source.  
150  
151     The Corresponding Source for a work in source code form is that  
152     same work.  
153  
154     2. Basic Permissions.  
155  
156     All rights granted under this License are granted for the term of  
157     copyright on the Program, and are irrevocable provided the stated  
158     conditions are met. This License explicitly affirms your unlimited  
159     permission to run the unmodified Program. The output from running a  
160     covered work is covered by this License only if the output, given its  
161     content, constitutes a covered work. This License acknowledges your  
162     rights of fair use or other equivalent, as provided by copyright law.  
163  
164     You may make, run and propagate covered works that you do not  
165     convey, without conditions so long as your license otherwise remains

166 in force. You may convey covered works to others for the sole purpose  
167 of having them make modifications exclusively for you, or provide you  
168 with facilities for running those works, provided that you comply with  
169 the terms of this License in conveying all material for which you do  
170 not control copyright. Those thus making or running the covered works  
171 for you must do so exclusively on your behalf, under your direction  
172 and control, on terms that prohibit them from making any copies of  
173 your copyrighted material outside their relationship with you.  
174

175 Conveying under any other circumstances is permitted solely under  
176 the conditions stated below. Sublicensing is not allowed; section 10  
177 makes it unnecessary.

178 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

181 No covered work shall be deemed part of an effective technological  
182 measure under any applicable law fulfilling obligations under article  
183 11 of the WIPO copyright treaty adopted on 20 December 1996, or  
184 similar laws prohibiting or restricting circumvention of such  
185 measures.

187 When you convey a covered work, you waive any legal power to forbid  
188 circumvention of technological measures to the extent such circumvention  
189 is effected by exercising rights under this License with respect to  
190 the covered work, and you disclaim any intention to limit operation or  
191 modification of the work as a means of enforcing, against the work's  
192 users, your or third parties' legal rights to forbid circumvention of  
193 technological measures.

194 4. Conveying Verbatim Copies.

197 You may convey verbatim copies of the Program's source code as you  
198 receive it, in any medium, provided that you conspicuously and  
199 appropriately publish on each copy an appropriate copyright notice;  
200 keep intact all notices stating that this License and any  
201 non-permissive terms added in accord with section 7 apply to the code;  
202 keep intact all notices of the absence of any warranty; and give all  
203 recipients a copy of this License along with the Program.

205 You may charge any price or no price for each copy that you convey,  
206 and you may offer support or warranty protection for a fee.

208 5. Conveying Modified Source Versions.

210 You may convey a work based on the Program, or the modifications to  
211 produce it from the Program, in the form of source code under the  
212 terms of section 4, provided that you also meet all of these conditions:

214 a) The work must carry prominent notices stating that you modified  
215 it, and giving a relevant date.

217 b) The work must carry prominent notices stating that it is  
218 released under this License and any conditions added under section  
219 7. This requirement modifies the requirement in section 4 to  
220 "keep intact all notices".

221  
222       c) You must license the entire work, as a whole, under this  
223       License to anyone who comes into possession of a copy. This  
224       License will therefore apply, along with any applicable section 7  
225       additional terms, to the whole of the work, and all its parts,  
226       regardless of how they are packaged. This License gives no  
227       permission to license the work in any other way, but it does not  
228       invalidate such permission if you have separately received it.  
229

230       d) If the work has interactive user interfaces, each must display  
231       Appropriate Legal Notices; however, if the Program has interactive  
232       interfaces that do not display Appropriate Legal Notices, your  
233       work need not make them do so.  
234

235       A compilation of a covered work with other separate and independent  
236       works, which are not by their nature extensions of the covered work,  
237       and which are not combined with it such as to form a larger program,  
238       in or on a volume of a storage or distribution medium, is called an  
239       "aggregate" if the compilation and its resulting copyright are not  
240       used to limit the access or legal rights of the compilation's users  
241       beyond what the individual works permit. Inclusion of a covered work  
242       in an aggregate does not cause this License to apply to the other  
243       parts of the aggregate.  
244

245       6. Conveying Non-Source Forms.  
246

247       You may convey a covered work in object code form under the terms  
248       of sections 4 and 5, provided that you also convey the  
249       machine-readable Corresponding Source under the terms of this License,  
250       in one of these ways:  
251

252       a) Convey the object code in, or embodied in, a physical product  
253       (including a physical distribution medium), accompanied by the  
254       Corresponding Source fixed on a durable physical medium  
255       customarily used for software interchange.  
256

257       b) Convey the object code in, or embodied in, a physical product  
258       (including a physical distribution medium), accompanied by a  
259       written offer, valid for at least three years and valid for as  
260       long as you offer spare parts or customer support for that product  
261       model, to give anyone who possesses the object code either (1) a  
262       copy of the Corresponding Source for all the software in the  
263       product that is covered by this License, on a durable physical  
264       medium customarily used for software interchange, for a price no  
265       more than your reasonable cost of physically performing this  
266       conveying of source, or (2) access to copy the  
267       Corresponding Source from a network server at no charge.  
268

269       c) Convey individual copies of the object code with a copy of the  
270       written offer to provide the Corresponding Source. This  
271       alternative is allowed only occasionally and noncommercially, and  
272       only if you received the object code with such an offer, in accord  
273       with subsection 6b.  
274

275       d) Convey the object code by offering access from a designated

276 place (gratis or for a charge), and offer equivalent access to the  
277 Corresponding Source in the same way through the same place at no  
278 further charge. You need not require recipients to copy the  
279 Corresponding Source along with the object code. If the place to  
280 copy the object code is a network server, the Corresponding Source  
281 may be on a different server (operated by you or a third party)  
282 that supports equivalent copying facilities, provided you maintain  
283 clear directions next to the object code saying where to find the  
284 Corresponding Source. Regardless of what server hosts the  
285 Corresponding Source, you remain obligated to ensure that it is  
286 available for as long as needed to satisfy these requirements.  
287

288 e) Convey the object code using peer-to-peer transmission, provided  
289 you inform other peers where the object code and Corresponding  
290 Source of the work are being offered to the general public at no  
291 charge under subsection 6d.

292 A separable portion of the object code, whose source code is excluded  
293 from the Corresponding Source as a System Library, need not be  
294 included in conveying the object code work.

295 A "User Product" is either (1) a "consumer product", which means any  
296 tangible personal property which is normally used for personal, family,  
297 or household purposes, or (2) anything designed or sold for incorporation  
298 into a dwelling. In determining whether a product is a consumer product,  
299 doubtful cases shall be resolved in favor of coverage. For a particular  
300 product received by a particular user, "normally used" refers to a  
301 typical or common use of that class of product, regardless of the status  
302 of the particular user or of the way in which the particular user  
303 actually uses, or expects or is expected to use, the product. A product  
304 is a consumer product regardless of whether the product has substantial  
305 commercial, industrial or non-consumer uses, unless such uses represent  
306 the only significant mode of use of the product.  
307

308 "Installation Information" for a User Product means any methods,  
309 procedures, authorization keys, or other information required to install  
310 and execute modified versions of a covered work in that User Product from  
311 a modified version of its Corresponding Source. The information must  
312 suffice to ensure that the continued functioning of the modified object  
313 code is in no case prevented or interfered with solely because  
314 modification has been made.

315 If you convey an object code work under this section in, or with, or  
316 specifically for use in, a User Product, and the conveying occurs as  
317 part of a transaction in which the right of possession and use of the  
318 User Product is transferred to the recipient in perpetuity or for a  
319 fixed term (regardless of how the transaction is characterized), the  
320 Corresponding Source conveyed under this section must be accompanied  
321 by the Installation Information. But this requirement does not apply  
322 if neither you nor any third party retains the ability to install  
323 modified object code on the User Product (for example, the work has  
324 been installed in ROM).

325 The requirement to provide Installation Information does not include a  
326 requirement to continue to provide support service, warranty, or updates

331 for a work that has been modified or installed by the recipient, or for  
332 the User Product in which it has been modified or installed. Access to a  
333 network may be denied when the modification itself materially and  
334 adversely affects the operation of the network or violates the rules and  
335 protocols for communication across the network.

336  
337 Corresponding Source conveyed, and Installation Information provided,  
338 in accord with this section must be in a format that is publicly  
339 documented (and with an implementation available to the public in  
340 source code form), and must require no special password or key for  
341 unpacking, reading or copying.

342  
343 7. Additional Terms.

344  
345 "Additional permissions" are terms that supplement the terms of this  
346 License by making exceptions from one or more of its conditions.  
347 Additional permissions that are applicable to the entire Program shall  
348 be treated as though they were included in this License, to the extent  
349 that they are valid under applicable law. If additional permissions  
350 apply only to part of the Program, that part may be used separately  
351 under those permissions, but the entire Program remains governed by  
352 this License without regard to the additional permissions.

353  
354 When you convey a copy of a covered work, you may at your option  
355 remove any additional permissions from that copy, or from any part of  
356 it. (Additional permissions may be written to require their own  
357 removal in certain cases when you modify the work.) You may place  
358 additional permissions on material, added by you to a covered work,  
359 for which you have or can give appropriate copyright permission.

360  
361 Notwithstanding any other provision of this License, for material you  
362 add to a covered work, you may (if authorized by the copyright holders of  
363 that material) supplement the terms of this License with terms:

364  
365 a) Disclaiming warranty or limiting liability differently from the  
366 terms of sections 15 and 16 of this License; or

367  
368 b) Requiring preservation of specified reasonable legal notices or  
369 author attributions in that material or in the Appropriate Legal  
370 Notices displayed by works containing it; or

371  
372 c) Prohibiting misrepresentation of the origin of that material, or  
373 requiring that modified versions of such material be marked in  
374 reasonable ways as different from the original version; or

375  
376 d) Limiting the use for publicity purposes of names of licensors or  
377 authors of the material; or

378  
379 e) Declining to grant rights under trademark law for use of some  
380 trade names, trademarks, or service marks; or

381  
382 f) Requiring indemnification of licensors and authors of that  
383 material by anyone who conveys the material (or modified versions of  
384 it) with contractual assumptions of liability to the recipient, for  
385 any liability that these contractual assumptions directly impose on

386        those licensors and authors.

387  
388        All other non-permissive additional terms are considered "further  
389        restrictions" within the meaning of section 10. If the Program as you  
390        received it, or any part of it, contains a notice stating that it is  
391        governed by this License along with a term that is a further  
392        restriction, you may remove that term. If a license document contains  
393        a further restriction but permits relicensing or conveying under this  
394        License, you may add to a covered work material governed by the terms  
395        of that license document, provided that the further restriction does  
396        not survive such relicensing or conveying.

397  
398        If you add terms to a covered work in accord with this section, you  
399        must place, in the relevant source files, a statement of the  
400        additional terms that apply to those files, or a notice indicating  
401        where to find the applicable terms.

402  
403        Additional terms, permissive or non-permissive, may be stated in the  
404        form of a separately written license, or stated as exceptions;  
405        the above requirements apply either way.

406  
407        8. Termination.

408  
409        You may not propagate or modify a covered work except as expressly  
410        provided under this License. Any attempt otherwise to propagate or  
411        modify it is void, and will automatically terminate your rights under  
412        this License (including any patent licenses granted under the third  
413        paragraph of section 11).

414  
415        However, if you cease all violation of this License, then your  
416        license from a particular copyright holder is reinstated (a)  
417        provisionally, unless and until the copyright holder explicitly and  
418        finally terminates your license, and (b) permanently, if the copyright  
419        holder fails to notify you of the violation by some reasonable means  
420        prior to 60 days after the cessation.

421  
422        Moreover, your license from a particular copyright holder is  
423        reinstated permanently if the copyright holder notifies you of the  
424        violation by some reasonable means, this is the first time you have  
425        received notice of violation of this License (for any work) from that  
426        copyright holder, and you cure the violation prior to 30 days after  
427        your receipt of the notice.

428  
429        Termination of your rights under this section does not terminate the  
430        licenses of parties who have received copies or rights from you under  
431        this License. If your rights have been terminated and not permanently  
432        reinstated, you do not qualify to receive new licenses for the same  
433        material under section 10.

434  
435        9. Acceptance Not Required for Having Copies.

436  
437        You are not required to accept this License in order to receive or  
438        run a copy of the Program. Ancillary propagation of a covered work  
439        occurring solely as a consequence of using peer-to-peer transmission  
440        to receive a copy likewise does not require acceptance. However,

441 nothing other than this License grants you permission to propagate or  
442 modify any covered work. These actions infringe copyright if you do  
443 not accept this License. Therefore, by modifying or propagating a  
444 covered work, you indicate your acceptance of this License to do so.  
445

446       10. Automatic Licensing of Downstream Recipients.

447       Each time you convey a covered work, the recipient automatically  
448 receives a license from the original licensors, to run, modify and  
449 propagate that work, subject to this License. You are not responsible  
450 for enforcing compliance by third parties with this License.  
451

452       An "entity transaction" is a transaction transferring control of an  
453 organization, or substantially all assets of one, or subdividing an  
454 organization, or merging organizations. If propagation of a covered  
455 work results from an entity transaction, each party to that  
456 transaction who receives a copy of the work also receives whatever  
457 licenses to the work the party's predecessor in interest had or could  
458 give under the previous paragraph, plus a right to possession of the  
459 Corresponding Source of the work from the predecessor in interest, if  
460 the predecessor has it or can get it with reasonable efforts.  
461

462       You may not impose any further restrictions on the exercise of the  
463 rights granted or affirmed under this License. For example, you may  
464 not impose a license fee, royalty, or other charge for exercise of  
465 rights granted under this License, and you may not initiate litigation  
466 (including a cross-claim or counterclaim in a lawsuit) alleging that  
467 any patent claim is infringed by making, using, selling, offering for  
468 sale, or importing the Program or any portion of it.  
469

470       11. Patents.

471       A "contributor" is a copyright holder who authorizes use under this  
472 License of the Program or a work on which the Program is based. The  
473 work thus licensed is called the contributor's "contributor version".  
474

475       A contributor's "essential patent claims" are all patent claims  
476 owned or controlled by the contributor, whether already acquired or  
477 hereafter acquired, that would be infringed by some manner, permitted  
478 by this License, of making, using, or selling its contributor version,  
479 but do not include claims that would be infringed only as a  
480 consequence of further modification of the contributor version. For  
481 purposes of this definition, "control" includes the right to grant  
482 patent sublicenses in a manner consistent with the requirements of  
483 this License.  
484

485       Each contributor grants you a non-exclusive, worldwide, royalty-free  
486 patent license under the contributor's essential patent claims, to  
487 make, use, sell, offer for sale, import and otherwise run, modify and  
488 propagate the contents of its contributor version.  
489

490       In the following three paragraphs, a "patent license" is any express  
491 agreement or commitment, however denominated, not to enforce a patent  
492 (such as an express permission to practice a patent or covenant not to  
493 sue for patent infringement). To "grant" such a patent license to a  
494

496 party means to make such an agreement or commitment not to enforce a  
497 patent against the party.  
498

499 If you convey a covered work, knowingly relying on a patent license,  
500 and the Corresponding Source of the work is not available for anyone  
501 to copy, free of charge and under the terms of this License, through a  
502 publicly available network server or other readily accessible means,  
503 then you must either (1) cause the Corresponding Source to be so  
504 available, or (2) arrange to deprive yourself of the benefit of the  
505 patent license for this particular work, or (3) arrange, in a manner  
506 consistent with the requirements of this License, to extend the patent  
507 license to downstream recipients. "Knowingly relying" means you have  
508 actual knowledge that, but for the patent license, your conveying the  
509 covered work in a country, or your recipient's use of the covered work  
510 in a country, would infringe one or more identifiable patents in that  
511 country that you have reason to believe are valid.  
512

513 If, pursuant to or in connection with a single transaction or  
514 arrangement, you convey, or propagate by procuring conveyance of, a  
515 covered work, and grant a patent license to some of the parties  
516 receiving the covered work authorizing them to use, propagate, modify  
517 or convey a specific copy of the covered work, then the patent license  
518 you grant is automatically extended to all recipients of the covered  
519 work and works based on it.  
520

521 A patent license is "discriminatory" if it does not include within  
522 the scope of its coverage, prohibits the exercise of, or is  
523 conditioned on the non-exercise of one or more of the rights that are  
524 specifically granted under this License. You may not convey a covered  
525 work if you are a party to an arrangement with a third party that is  
526 in the business of distributing software, under which you make payment  
527 to the third party based on the extent of your activity of conveying  
528 the work, and under which the third party grants, to any of the  
529 parties who would receive the covered work from you, a discriminatory  
530 patent license (a) in connection with copies of the covered work  
531 conveyed by you (or copies made from those copies), or (b) primarily  
532 for and in connection with specific products or compilations that  
533 contain the covered work, unless you entered into that arrangement,  
534 or that patent license was granted, prior to 28 March 2007.  
535

536 Nothing in this License shall be construed as excluding or limiting  
537 any implied license or other defenses to infringement that may  
538 otherwise be available to you under applicable patent law.  
539

## 540 12. No Surrender of Others' Freedom.

### 541

542 If conditions are imposed on you (whether by court order, agreement or  
543 otherwise) that contradict the conditions of this License, they do not  
544 excuse you from the conditions of this License. If you cannot convey a  
545 covered work so as to satisfy simultaneously your obligations under this  
546 License and any other pertinent obligations, then as a consequence you may  
547 not convey it at all. For example, if you agree to terms that obligate you  
548 to collect a royalty for further conveying from those to whom you convey  
549 the Program, the only way you could satisfy both those terms and this  
550 License would be to refrain entirely from conveying the Program.

551  
552     13. Use with the GNU Affero General Public License.  
553

554       Notwithstanding any other provision of this License, you have  
555 permission to link or combine any covered work with a work licensed  
556 under version 3 of the GNU Affero General Public License into a single  
557 combined work, and to convey the resulting work. The terms of this  
558 License will continue to apply to the part which is the covered work,  
559 but the special requirements of the GNU Affero General Public License,  
560 section 13, concerning interaction through a network will apply to the  
561 combination as such.

562  
563     14. Revised Versions of this License.  
564

565       The Free Software Foundation may publish revised and/or new versions of  
566 the GNU General Public License from time to time. Such new versions will  
567 be similar in spirit to the present version, but may differ in detail to  
568 address new problems or concerns.

569  
570       Each version is given a distinguishing version number. If the  
571 Program specifies that a certain numbered version of the GNU General  
572 Public License "or any later version" applies to it, you have the  
573 option of following the terms and conditions either of that numbered  
574 version or of any later version published by the Free Software  
575 Foundation. If the Program does not specify a version number of the  
576 GNU General Public License, you may choose any version ever published  
577 by the Free Software Foundation.

578  
579       If the Program specifies that a proxy can decide which future  
580 versions of the GNU General Public License can be used, that proxy's  
581 public statement of acceptance of a version permanently authorizes you  
582 to choose that version for the Program.

583  
584       Later license versions may give you additional or different  
585 permissions. However, no additional obligations are imposed on any  
586 author or copyright holder as a result of your choosing to follow a  
587 later version.

588  
589     15. Disclaimer of Warranty.  
590

591       THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY  
592 APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT  
593 HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY  
594 OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,  
595 THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
596 PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM  
597 IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF  
598 ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

599  
600     16. Limitation of Liability.  
601

602       IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING  
603 WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS  
604 THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY  
605 GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE

606 USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF  
607 DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD  
608 PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),  
609 EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF  
610 SUCH DAMAGES.

611

## 612 | 17. Interpretation of Sections 15 and 16.

613

614 If the disclaimer of warranty and limitation of liability provided  
615 above cannot be given local legal effect according to their terms,  
616 reviewing courts shall apply local law that most closely approximates  
617 an absolute waiver of all civil liability in connection with the  
618 Program, unless a warranty or assumption of liability accompanies a  
619 copy of the Program in return for a fee.

620

## END OF TERMS AND CONDITIONS

622

623

624

## C.2 copyright

Este código comienza en la pagina que sigue.

- 1 Programa de control de hardware e ilustración física de las propiedades de las ondas electromagnéticas en el espectro infrarrojo.
- 2 Copyright (C) 2016-01-01 Universidad Distrital Francisco Jose, Diego Alberto Parra Garzón, Dr. Julian Andres Salamanca Bernal.
- 3 El programa free infrarroso es software libre; puedes redistribuirlo y / o modificarlo bajo los términos de la Licencia Pública General GNU publicada por la Fundación para el Software Libre; ya sea la versión 3 de la Licencia, o (a su elección) cualquier versión posterior.
- 4 Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTÍA; ni siquiera la garantía implícita de COMERCIALIZACIÓN o IDONEIDAD PARA UN PROPÓSITO PARTICULAR. Vea el Licencia Pública General GNU para más detalles.
- 5 Debería haber recibido una copia de la Licencia Pública General de GNU junto con este programa; si no, escriba a la Free Software Foundation, Inc., 51 Franklin Street, Quinto Piso, Boston, MA 02110-1301 EE.UU..
- 6

### C.3 README.md

Este código comienza en la pagina que sigue.

- 1 Programa de control de hardware e ilustración física de las propiedades de las ondas electromagnéticas en el espectro infrarrojo.
- 2 Copyright (C) 2016-01-01 Universidad Distrital Francisco Jose, Diego Alberto Parra Garzón, Dr. Julian Andres Salamanca Bernal.
- 3 El programa free infrarroso es software libre; puedes redistribuirlo y / o modificarlo bajo los términos de la Licencia Pública General GNU publicada por la Fundación para el Software Libre; ya sea la versión 3 de la Licencia, o (a su elección) cualquier versión posterior.
- 4 Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTÍA; ni siquiera la garantía implícita de COMERCIALIZACIÓN o IDONEIDAD PARA UN PROPÓSITO PARTICULAR. Vea el Licencia Pública General GNU para más detalles.
- 5 Debería haber recibido una copia de la Licencia Pública General de GNU junto con este programa; si no, escriba a la Free Software Foundation, Inc., 51 Franklin Street, Quinto Piso, Boston, MA 02110-1301 EE.UU..
- 6

#### C.4 instalador.py

Este código comienza en la pagina que sigue.

```

1  infrarr#/usr/bin/python
2  #!*- coding:utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los términos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.

15
16 import os
17 import time
18 class Instalador:
19     def Presentacion(self):
20         os.system("clear")
21         print chr(27)+"[2;32m"+"\n\n\t @@@@@@@@CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
22         print "\t\t Instalador de free_infrarossi"
23         print "\t @@@@@@@@CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
24
25         print chr(27)+"[0m"+"\n \n \n \t Bienvenido al software free_infrarossi, el
26         cual le permitira tener\n\t varias horas de diversión; este software fue
27         diseñado para el control,\n\t recolección y análisis de datos del vehículo
28         motorizado infrarossi,\n\t el cual permite ilustrar el estudio de las
29         propiedades de las ondas\n\t electromagnéticas en el espectro infrarrojo,
30         tales como la difracción,\n\t atenuación y absorción. Disfrutara de una
31         interfaz amigable al usuario. "
32         print "\n\n\t @@@@@@@@CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
33         print chr(27)+"[5;36m"+"\t\t DESEA CONTINUAR CON LA INSTALACION: "
34         print chr(27)+"[0m"+"\t @@@@@@@@CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
35         print "\t Oprima 1 para si \n\t Oprima 2 para no"
36         Pr1 = int(raw_input("Ingrese su respuesta: "))
37         if Pr1 == 1:
38             self.Instalar()
39         elif Pr1 ==2:
40             self.exit()
41             os.system("killall python")
42         else:
43             self.exit()
44             os.system("killall python")
45         def Instalar(self):
46             os.system("clear")
47             print chr(27)+"[2;32m"+"\n\n ¿Que desea hacer ?\n\n"
48             print chr(27)+"[0m"+"\n \t oprima 1 para instalar free_infrarossi"
49             print "\n \t Oprima 2 para desinstalar free_infrarossi"
50             print chr(27)+"[3;36m"+"\n \t Oprima cualquier tecla para SALIR del
51             instalador\n\n"
52             Pr2 = int(raw_input(chr(27)+"[5;32m"+"Ingrese su petición : "+chr(27)+"[0m"))
53             if Pr2 == 1:
54                 os.system("apt-get update ")

```

```

49         os.system("apt-get install xterm bluez* gcc g++ emacs gnuplot gnuplot-qt evince octave python-matplotlib python-numpy python-tk python-gnuplot python-serial python-visual* libgtkglextmm* arduino fritzing binutils")
50         archi = open('/etc/bash.bashrc', 'a+')
51         archi.write("\nalias infrarossi='cd ~/Documentos/Free-infrarossi/free_infrarossi && ./infrarossi'")
52         archi.close()
53         print chr(27)+"[5;32m"+"INSTALACION TERMINADA "
54         print "Disfrute su software \n reinicie su pc"
55
56     elif Pr2 == 2:
57         os.system("apt-get --purge remove emacs gnuplot gnuplot-qt evince octave python-matplotlib python-scipy python-numpy python-tk python-gnuplot python-serial python-visual* libgtkglextmm* arduino fritzing")
58         os.system("apt-get autoremove")
59         os.system("apt-get update")
60         os.system("bash unistall.sh")
61         os.system("clear")
62         print "\t\t\t DESINSTALACION EXITOSA "
63         print "Se ha creado una copia de respaldo del archivo"+chr(27)+"[3;36m"+ "\n\n\n /etc/bash.bashrc como /etc/bash.bashrc.respaldo\n" + chr(27)+ "[0m"+ "\n\n\n si tiene algun problema con este archivo despues de la desinstalación;\n solamente ejecute este comando en la terminal con permisos\n de administrador y sin las comillas \n\n\t "+chr(27)+"[5;31m"+ "+ANOTELO QUE ES \n\t MUY IMPORTANTE"+chr(27)+"[0m"+ "\n\n\t "+chr(27)+"[3;33m"+ "cp /etc/bash.bashrc.respaldo /etc/bash.bashrc"+chr(27)+"[0m"+ "\n\n"
64         time.sleep(60)
65         print "Desinstalacion completada ---"
66         print "reinicio su pc"
67         self.exit()
68
69     else:
70         print "cerrando el instalador"
71         self.exit()
72
73     def exit(self):
74         exit()
75         exit()
76         exit()
77
78     def __init__(self):
79         self.Presentacion()
80         self.__del__()
81
82     def __del__(self):
83         print chr(27)+"[5;33m"+"FIN DEL PROGRAMA"
84         os.system("exit")
85         os.system("rm logs.txt")
86         time.sleep(4)
87
88 if __name__ == "__main__":
89     Instalador()
90

```

## C.5 INSTALADOR.py

Este código comienza en la pagina que sigue.

```
1 #!/usr/bin/python
2 #!-*- coding:utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15
16 import os
17 os.system('sudo python instalador.py')
18 os.system("infrarossi= 'cd ~/Documentos/Free-infrarossi/free_infrarossi &&
19 ./infrarossi'")
```

→

## C.6 unistall.py

Este código comienza en la pagina que sigue.

```
1 #!/bin/bash
2 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
3 # los términos de la licencia pública general de GNU, según es publicada
4 # por la free software fundation bien la versión 3 de la misma licencia
5 # o de cualquier versión posterior. (según su elección).
6 # Si usted hace alguna modificación en esta aplicación, deberá siempre
7 # mencionar el autor original de la misma.
8 # Autor:
9 # Universidad Distrital Francisco Jose
10 # Grupo de fisica e informatica
11 # Diego Alberto Parra Garzón
12 # Dr Julian Andres Salamanca Bernal
13 # Colombia, Bogota D.C.
14
15 cp /etc/bash.bashrc /etc/bash.bashrc.respaldo
16 rm -rf ~/Documentos/Free-infrarossi
17 cat /etc/bash.bashrc | grep -n "infrarossi" | cut -d ":" -f 1,1 > logs.txt
18 NUM=`cat logs.txt`
19 LET=`echo d`
20 LINEA=`echo -e "$NUM$LET"`
21 sed -i "$LINEA" /etc/bash.bashrc
22
```

## C.7 infrarossi.py

Este código comienza en la pagina que sigue.

```

1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.

15
16 import numpy as np
17 import pylab as pl
18 import serial
19 import os
20 import subprocess
21 import math
22 import time
23 import Gnuplot
24 from Tkinter import *
25 import tkMessageBox
26 import Tkinter
27 import shutil

28
29 class App:
30     def Modulo(self):
31         bicho = Tk()
32         bicho.geometry("430x180+180+60")
33         bicho.config(bg="white")
34         bicho.title("Infrarossi")
35         bicho.resizable(width=0, height=0)

36
37     def Salir():
38         tkMessageBox.showinfo("Infrarossi", message= "¡ Cerrando el programa !")
39         os.system("rm MAC.txt MACD.txt puerto.txt conexion.txt direccion.txt")
40         os.system("rm dispo.txt log.txt mac.txt macd.txt texput.log x.log")
41         os.system("rm MAC.txt MACD.txt puerto.txt")
42         os.system("rm bin/conexion.txt bin/direccion.txt")
43         os.system("sh bin/d_Blush &")
44         exit()

45
46     def Verifica():
47         tkMessageBox.showinfo("Infrarossi", message= "¡ Conectando con el dispositivo, por favor espere !")
48         os.system('xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue -e sh c_Blush ')
49         Conecta()

50
51     def Conecta():
52         os.system('rfcomm -a > conexion.txt | cut -d ":" -f 1,1

```

```

    conexion.txt > direccion.txt ')
53     puerto = open('direccion.txt', 'r')
54     self.puerto = puerto.read(7)
55     puerto.close()
56     self.puerta = "/dev/" +self.puerto
57     print self.puerta
58
59     Conectar()
60
61 def Conectar():
62     tkMessageBox.showinfo("Infrarossi", message= " ! Verificando conexion i ")
63     print self.puerta
64     try:
65         arduino = serial.Serial(self.puerta, 9600)
66         time.sleep(4)
67         arduino.write("aa")
68         Valido()
69         Valido()
70     except:
71         tkMessageBox.showinfo("Infrarossi", message= " ! No hay conexion i ")
72         # os.system('xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue -e sh c_Blu.sh &')
73
74
75     def Valido():
76 # ..... Botones menu de inicio .....
77     x1 = int(150)
78     y3 = int(300)
79     lblRapidez = Label(bicho, text="\nMENU\nDE INICIO", fg = ("black"), bg =
80                         = ("white"), font = ("Century Schoolbook L",10)).place(x=15+x1, y=280
81                         -y3)
80     btnComenzar = Button(bicho, text= "Difraccion", width=6, height=1,
81     command= Difraccion).place(x=20+x1, y=350-y3)
81     btnDetener= Button(bicho, text= "Atenuacion", width=6, height=1,
81     command= Atenuacion).place(x=20+x1, y=380-y3)
82     btnLimpiar = Button(bicho, text= "Absolucion", width=6, height=1, command=
82     Absolucion).place(x=20+x1, y=410-y3)
83
84
85
86 #----- Funcion limpiar pantalla
87
88 def Reset():
89     tkMessageBox.showinfo("Infrarossi", message= " ! Limpiando, por favor espere i ")
90     os.system("rm MAC.txt MACD.txt puerto.txt conexion.txt direccion.txt")
90     os.system("sh bin/d_Blu.sh &")
91     os.system("python infrarossi.py &")
92     os.system("rm MAC.txt MACD.txt puerto.txt")
93     exit()
94
95 #----- Funciones menu
96
97     def Difraccion():

```

```

97         tkMessageBox.showinfo("Infrarossi", message= "Espere por favor,      ↵
98             Preparando todo para empezar con el experimento.")
99     #         os.system("python bin/Difraccion.py &")
100
101
102     def Atenuacion():
103         tkMessageBox.showinfo("Infrarossi", message= "Espere por favor,      ↵
104             Preparando todo para empezar con el experimento.")
105         os.system("python bin/Atenuacion.py &")
106     #         os.system("xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue -e
107         python bin/Atenuacion.py & ")
108
109     def Absorcion():
110         tkMessageBox.showinfo("Infrarossi", message= "Espere por favor,      ↵
111             Preparando todo para empezar con el experimento.")
112         os.system("python bin/Absorcion.py & ")
113
114     #----- Bluetooth desconectado
115     -----
116     def Bl_off():
117         tkMessageBox.showinfo("Infrarossi", message= "!Bluetooth desconectado!")
118         os.system("rm MAC.txt MACD.txt puerto.txt conexion.txt direccion.txt")
119         os.system("sh bin/d_Blu.sh &")
120         os.system("python infrarossi.py &")
121         os.system("rm MAC.txt MACD.txt puerto.txt")
122         exit()
123
124     # ----- Definiendo Funcion firmware
125     -----
126
127     def Firmware():
128         tkMessageBox.showinfo("Infrarossi", message= "Conekte la tarjeta      ↵
129             microcontroladora arduino uno, con un microcontrolador listo para su
130             uso.\n\nProcediendo con el instalador del firmware")
131         os.system("python bin/firmware/G_firmware.py &")
132
133
134
135
136     #----- Botones Bluetooth
137     -----
138     y1 = int(300)
139     lblBlue = Label(bicho, text="BLUETOOTH ", fg = ("black"), bg = ("white"),
139                      font = ("Century Schoolbook L",10)).place(x=10, y=320-y1)
139     btnConectar= Button(bicho, text= " ON ", width=5, height=1, command=

```

```
140     Verifica).place(x=20, y=350-y1)
141 #         btnConectar= Button(bicho, text= " ON ", width=5, height=1, command=
142 Valido).place(x=20, y=350-y1)                                     ↵
143         btnDesconectar= Button(bicho, text= " OFF ", width=5, height=1, command=
144             Bl_off).place(x=20, y=380-y1)                                     ↵
145     btnSalir=Button(bicho, text = "Salir", command=Salir, height=1, width =5).
146     place(x=20, y=410-y1)                                             ↵
147
148
149 #----- Botones firmware -----
150 y2 = int(270)
151 btnfirmware=Button(bicho, text = "Firmware", command=Firmware, height=1, width =
152     =5).place(x=340, y=380-y2)                                     ↵
153 btnDocumentacion=Button(bicho, text = "Ayuda", command=Documentacion, height=1 ↵
154 , width =5).place(x=340, y=350-y2)
155     btnLimpiar = Button(bicho, text= "limpiar", width=5, height=1, command=
156     Reset).place(x=340, y=320-y2)
157     bicho.mainloop()
158
159
160 def __init__(self):
161     self.Modulo()
162     self.__del__()
163
164 def __del__(self):
165     print ("PROGRAMA TERMINADO")
166
167 if __name__ == "__main__":
168     App()
```

## **C.8 infrarossi**

Este código comienza en la pagina que sigue.

```
1 #!/usr/bin/python
2 # -*- coding:utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de física e informática
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15
16 import os
17 import time
18
19
20 os.system("sudo python
~/Documentos/Free-infrarossi/free_infrarossi/infrarossi.py")
```

→

## C.9 c\_Blue.sh

Este código comienza en la pagina que sigue.

```
1 #!/bin/bash
2 #/etc/init.d/bluetooth restart
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de física e informática
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15 rfcomm release rfcomm0
16 hcitool dev > log.txt | grep -e 'hci0' log.txt > mac.txt
17 cut -d "0" -f 2,3 mac.txt > MAC.txt
18 MAC=`cat MAC.txt`
19 echo "La mac del dispositivo es : $MAC"
20 hcitool scan > dispo.txt | grep -e "HC-05" dispo.txt > macd.txt
21 cut -d "H" -f 1 macd.txt > MACD.txt
22 #/etc/init.d/bluetooth status
23 #/etc/init.d/bluetooth force-reload
24 #/etc/init.d/bluetooth reset
25 rm log.txt mac.txt dispo.txt macd.txt
26 MACD=`cat MACD.txt`
27 echo "dispositivo HC-05 encontrado MAC: $MACD"
28 rfcomm connect -i $MACD
29
30
31
```

## C.10 g\_p\_Abs.py

Este código comienza en la pagina que sigue.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los términos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Dr Julian Andres Salamanca Bernal
13 # Diego Alberto Parra Garzón
14 # Colombia, Bogota D.C.

15
16 import numpy as np
17 import os
18 import serial
19 import subprocess
20 import math
21 import time
22 import Gnuplot
23 import shutil
24 import matplotlib.pyplot as pl
25 class App:
26     def Analisis(self):
27         # os.system("octave bin/estadistica1.m")
28         os.system("octave estadistica2.m")
29         self.BX = np.loadtxt("datos/BX.dat")
30         self.BY = np.loadtxt("datos/BY.dat")
31         self.error_CM1 = np.loadtxt("datos/ECM1.dat")
32         self.q1 = np.loadtxt("datos/q1.dat")
33         self.Amplitud = np.loadtxt("datos/Amplitud.dat")
34         self.Ampli = round(self.Amplitud, 6)
35         self.T = np.loadtxt("datos/Transmitancia.dat")
36         self.A = np.loadtxt("datos/Vteo.dat")
37         self.V_A = np.loadtxt("datos/V_aproximado.dat")
38         f = np.loadtxt("datos/F.dat")
39         a = np.loadtxt("datos/a.dat")
40         b = np.loadtxt("datos/b.dat")
41         self.Yest = np.loadtxt("datos/Yest.dat")
42         self.Yteo = np.loadtxt("datos/Yteo.dat")
43         self.ECM = np.loadtxt("datos/ECM.dat")
44         self.error_CM11 = round(self.error_CM1, 11)
45         self.X = f[:,0]
46         self.V = f[:,1]
47         self.a = round(a, 6)
48         self.b = round(b, 3)
49         print self.X, self.V, "\n\n\n", self.a, " X^(," ,self.b, ")   "
50
51     def Grafica(self):
52         pl.subplot(221)
53         pl.title('Distancia vs Intensidad \n')
54         pl.xlabel('Distancia [m]')
55         pl.ylabel('Intensidad [W]')

```

```

56     pl.plot(self.X, self.V, 'g:o')
57     pl.axis([0, 0.65, -0.001, 0.022])
58     pl.text(0.2, 0.019, r'Datos recolectados')
59
60     def Grafical(self):
61         pl.subplots_adjust(left=0.13)
62         pl.subplots_adjust(bottom=0.13)
63         pl.subplots_adjust(right=0.97)
64         pl.subplots_adjust(top=0.87)
65         pl.subplots_adjust(wspace=0.37)
66         pl.subplots_adjust(hspace=0.68)
67
68     def Grafica2(self):
69         pl.subplot(222)
70         pl.title('APROXIMACION')
71         pl.xlabel('Distancia [m]')
72         pl.ylabel('Intensidad [W]')
73         pl.plot(self.X, self.A, 'B', self.q1, self.BY, 'y', self.q1, self.BX, 'b')
74         pl.plot(self.X, self.V_A, 'Y', self.X, self.V, 'g:o')
75         pl.axis([0, 0.65, -0.001, 0.022])
76         pl.text(0.15, 0.015, r' ' + str(0.000277) + ' + X^(' + str(-2) + ')')
77         pl.text(0.3, 0.017, r'Azul')
78         pl.text(0.2, 0.010, r' ' + Amarillo)
79         pl.text(0.19, 0.008, r' ' + str(self.Ampli) + ' X^(-2)')
80
81
82     def Grafica3(self):
83         pl.subplot(212)
84         pl.plot(self.X, self.V, 'g:o', self.X, self.V_A, 'Y', self.X, self.A, 'B',
85         self.q1, self.BY, 'y', self.q1, self.BX, 'b')
86         pl.subplot(212)
87         pl.xlabel('Distancia [m]')
88         pl.ylabel('Intensidad [W]')
89         pl.title('Distancia vs Intensidad irradiada \n')
90         pl.legend(loc='upper left')
91         pl.axis([0, 0.65, -0.001, 0.022])
92         pl.text(0.3, 0.019, r' ideal = ' + str(0.000277) + ' + X^(' + str(-2) + ')')
93         pl.text(0.3, 0.016, r' Exp = ' + str(self.Ampli) + ' X^(-2)')
94         pl.text(0.12, 3.5, r'V1' )
95         pl.text(0.01, 2, r'V2*' )
96         pl.text(0.3, 0.013, r'Fp = ' + str(round(self.T,4)))
97         pl.text(0.3, 0.010, r'ECM = ' + str(round(self.error_CM1,11)))
98         pl.savefig('datos/Absolucion.png')
99         pl.show()
100
101     def Ordenar(self):
102         os.system("python bin/o_Carpetas2.py")
103
104     def __init__(self):
105         self.Analisis()
106         self.Grafical()
107         self.Grafica2()
108         self.Grafica3()
109     # self.Ordenar()

```

```
110         self.__del__()
111
112     def __del__(self):
113         print ("PROGRAMA TERMINADO")
114
115
116 Iniciar = App()
117
```

### C.11 g\_p\_Ate.py

Este código comienza en la pagina que sigue.

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Dr Julian Andres Salamanca Bernal
13 # Diego Alberto Parra Garzón
14 # Colombia, Bogota D.C.
15 import numpy as np
16 import os
17 import serial
18 import subprocess
19 import math
20 import time
21 import Gnuplot
22 import shutil
23 import matplotlib.pyplot as pl
24 class App:
25     def Analisis(self):
26         # os.system("octave datos/estadistical.m")
27         os.system("octave estadistical.m")
28         f = np.loadtxt("datos/F.dat")
29         a = np.loadtxt("datos/a.dat")
30         b = np.loadtxt("datos/b.dat")
31         self.Yest = np.loadtxt("datos/Yest.dat")
32         self.Yteo = np.loadtxt("datos/Yteo.dat")
33         self.ECM = np.loadtxt("datos/ECM.dat")
34         self.ECM1 = round(self.ECM, 11)
35         self.X = f[:,0]
36         self.V = f[:,1]
37         self.a = round(a, 6)
38         self.b = round(b, 3)
39         self.error = ((-2-float(self.b))**2)**0.5
40         print self.X, self.V, "\n\n\n", self.a, " X^(," ,self.b, ",)" 
41
42     def Grafica(self):
43         pl.subplot(221)
44         pl.title('Distancia vs Intensidad \n')
45         pl.xlabel('Distancia [m]')
46         pl.ylabel('Intensidad [W]')
47         pl.plot(self.X, self.V, 'o --')
48         pl.axis([0, 0.65, -0.001, 0.022])
49         pl.text(0.2, 0.019, r'Datos recolectados')
50
51
52
53     def Grafical(self):
54         pl.subplots_adjust(left=0.13)
55         pl.subplots_adjust(bottom=0.13)
```

```

56         pl.subplots_adjust(right=0.97)
57         pl.subplots_adjust(top=0.87)
58         pl.subplots_adjust(wspace=0.37)
59         pl.subplots_adjust(hspace=0.68)
60
61
62     def Grafica2(self):
63         pl.subplot(222)
64         pl.title('APROXIMACION')
65         pl.xlabel('Distancia [m]')
66         pl.ylabel('Intensidad [W]')
67         pl.plot(self.X, self.V, 'B')
68         pl.plot(self.X, self.Yest, 'R')
69         pl.axis([0, 0.65, -0.001, 0.020])
70         pl.text(0.3, 0.017, r'Azul')
71         pl.text(0.15, 0.015, r'+' + "(" + str(self.a) + ")" + '*X^(+' + str(self.b) + ')')
72         pl.text(0.2, 0.010, r'Rojo')
73         pl.text(0.19, 0.008, r'+' + "(" + str(self.a) + ")" + '*X^(-2)')
74
75
76     def Grafica3(self):
77         pl.subplot(212)
78         pl.plot(self.X, self.V, 'K')
79         pl.plot(self.X, self.Yeo, 'R')
80         pl.plot(self.X, self.Yest, 'B')
81         pl.subplot(212)
82         pl.xlabel('Distancia [m]')
83         pl.ylabel('Intensidad [W]')
84         pl.title('Distancia vs Intensidad irradiada \n')
85         pl.legend(loc='upper left')
86         pl.axis([0, 0.65, -0.001, 0.022])
87         pl.text(0.3, 0.02, r'Azul I_r=' + str(self.a) + '*X^(+' + str(self.b) + ')')
88         pl.text(0.3, 0.017, r'Rojo I_r=' + str(self.a) + '*X^(-2)')
89         pl.text(0.3, 0.014, r'ECM = ' + str(self.ECM1))
90         pl.text(0.3, 0.011, r'E_exp = ' + str(self.error))
91         pl.savefig('datos/Atenuacion.png')
92         pl.show()
93
94     def Ordenar(self):
95         os.system("python bin/o_Carpetas.py")
96
97     def __init__(self):
98         self.Analisis()
99         self.Grafica()
100        self.Grafica1()
101        self.Grafica2()
102        self.Grafica3()
103    # self.Ordenar()
104    self.__del__()
105
106    def __del__(self):
107        print ("PROGRAMA TERMINADO")
108
109
110 Iniciar = App()

```

## **C.12 Estadistica1.m**

Este código comienza en la pagina que sigue.

```

1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Dr Julian Andres Salamanca Bernal
13 # Diego Alberto Parra Garzón
14 # Colombia, Bogota D.C.
15
16 f = load('datos/dats1.dat'); #-----entrada de datos a través del archivo .dat
17 X = f(:,1)/100 #-----datos de distancia en metros
18 Y = f(:,2)/1000000 #-----datos de intensidad en vatios
19 ff = [X, Y] # -----voltaje como función X e Y
20 U = log(X); # -----logaritmo natural de la distancia
21 V = log(Y); #----- logaritmo natural del voltaje
22 U2 = U .* U; #----- el cuadrado del logaritmo de la distancia
23 UV = U .* V; # .....logaritmo del producto del voltaje y la
24 distancia
25 z = size(f); #----- tamaño de filas y columnas del archivo .dat
26 z1 = z(:,1); #----- tamaño de columnas del archivo .dat
27 sumX = sum(X); #----- sumatoria de todos los datos de la
28 distancia en metros
29 sumY = sum(Y); #----- sumatoria de todos los datos del voltaje
30 en voltios
31 sumU = sum(U); #----- sumatoria de todos los datos del
32 logaritmo de la distancia
33 sumV = sum(V); #----- sumatoria de todos los datos del
34 logaritmo del voltaje
35 sumU2 = sum(U2); #----- sumatoria de todos los datos del
36 cuadrado del logaritmo de la distancia
37 sumUV = sum(UV); #----- sumatoria de todos los datos del
38 producto de el logaritmo del voltaje y la distancia
39 promX = sumX/z1; #----- el promedio de la distancia
40 promY = sumY/z1; #----- el promedio del voltaje
41 promU = sumU/z1; #----- promedio del logaritmo de la distancia
42 promV = sumV/z1; #----- promedio del logaritmo del voltaje
43 promU2 = sumU2/z1; #----- promedio del logaritmo la distancia
44 promUV = sumUV/z1; #----- promedio del cociente del logaritmo
45 de la distancia y el voltaje.
46 #-----Primer calculo-----
47 Suv = promUV - promU*promV;
48 Su2 = promU2 - promU*promU;
49 #b = (Suv / Su2) - 0.5
50 b = (Suv / Su2)
51 A = (promV - b*promU)
52 #a = exp(A)/5 + 0.02
53 a = exp(A)
54 Yest = a* (X .^ b);

```

```
47 Yteo= a * (X .^(-2));
48 error = Y .- Yest;
49 ECM = sum(error .^(2)) /z1
50 save -ascii 'datos/a.dat' a;
51 save -ascii 'datos/F.dat' ff;
52 save -ascii 'datos/b.dat' b;
53 save -ascii 'datos/ECM.dat' ECM;
54 save -ascii 'datos/Yest.dat' Yest;
55 save -ascii 'datos/Yteo.dat' Yteo;
56 #plot (X, Y, 'o')
57 #pause
58
59
```

### **C.13 Estadistica2.m**

Este código comienza en la pagina que sigue.

```

1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Dr Julian Andres Salamanca Bernal
13 # Diego Alberto Parra Garzón
14 # Colombia, Bogota D.C.

15
16
17 f = load('datos/dats1.dat'); #-----entrada de datos a través del archivo .dat
18 X = f(:,1)/100 #-----datos de distancia en metros
19 Y = f(:,2)/1000000 #-----datos de voltaje en voltios
20 ff = [X, Y]; # -----voltaje como función X e Y
21 U = log(X); # -----logaritmo natural de la distancia
22 V = log(Y); #----- logaritmo natural del voltaje
23 U2 = U .* U; #----- el cuadrado del logaritmo de la distancia
24 UV = U .* V; # .....logaritmo del producto del voltaje y la distancia
25 z = size(f); #----- tamaño de filas y columnas del archivo .dat
26 z1 = z(:,1); #----- tamaño de columnas del archivo .dat
27 sumX = sum(X); #----- sumatoria de todos los datos de la distancia en metros
28 sumY = sum(Y); #----- sumatoria de todos los datos del voltaje en voltios
29 sumU = sum(U); #----- sumatoria de todos los datos del logaritmo de la distancia
30 sumV = sum(V); #----- sumatoria de todos los datos del logaritmo del voltaje
31 sumU2 = sum(U2); #----- sumatoria de todos los datos del cuadrado del logaritmo de la distancia
32 sumUV = sum(UV); #----- sumatoria de todos los datos del producto de el logaritmo del voltaje y la distancia
33 promX = sumX/z1; #----- el promedio de la distancia
34 promY = sumY/z1; #----- el promedio del voltaje
35 promU = sumU/z1; #----- promedio del logaritmo de la distancia
36 promV = sumV/z1; #----- promedio del logaritmo del voltaje
37 promU2 = sumU2/z1; #----- promedio del logaritmo la distancia
38 promUV = sumUV/z1; #----- promedio del cociente del logaritmo de la distancia y el voltaje.
39 #-----Primer calculo-----
40 Suv = promUV - promU*promV;
41 Su2 = promU2 - promU*promU;
42 #b = (Suv / Su2) - 0.5
43 b = (Suv / Su2)
44 A = (promV - b*promU)
45 #a = exp(A)/5 + 0.02
46 a = exp(A)

```

```

47 Yest = a* (X .^ b);
48 Yteo= a * (X .^(-2));
49 error = Y .- Yest;
50 ECM = sum(error .^(2)) /z1
51 save -ascii 'datos/a.dat' a;
52 save -ascii 'datos/F.dat' ff;
53 save -ascii 'datos/b.dat' b;
54 save -ascii 'datos/ECM.dat' ECM;
55 save -ascii 'datos/Yest.dat' Yest;
56 #-----segundo calculo-----
57 Vteo = 0.000277 * (X .^(-2))
58 q1 = [0.01:0.001:5];
59 Pprueba = (Y .* Y)/125 #----- Potencia en el diodo
60 Voltaje_entrante = 100*(Y ./ Vteo)
61 Pteori = (Vteo .* Vteo)/125
62 DV = Vteo .- Y;
63 T = DV ./ Vteo;
64 sT = sum(T);
65 Tp = sT / z1;
66 Trasmitancia = Tp
67 Reflectancia = 1 - Tp
68 EEp= (Reflectancia)*(0.000277);
69 densidad_voltaje = EEp
70 Voltaje_aproximado = (EEp)*(X .^(-2));
71 BX = (0.000277)*(q1 .^(-2));
72 BY = EEp * (q1 .^(-2));
73 V_estimado = EEp*(X .^(-2));
74 error1 = Y .- V_estimado;
75 ECM1 = sum(error1 .^(2)/z1)
76 save -ascii 'datos/Yteo.dat' Yteo;
77 save -ascii 'datos/Vteo.dat' Vteo;
78 save -ascii 'datos/Amplitud.dat' EEp;
79 save -ascii 'datos/Transmitancia.dat' Tp;
80 save -ascii 'datos/V_aproximado.dat' Voltaje_aproximado;
81 save -ascii 'datos/q1.dat' q1;
82 save -ascii 'datos/BX.dat' BX;
83 save -ascii 'datos/BY.dat' BY;
84 save -ascii 'datos/ECM1.dat' ECM1;
85
86 #plot (q1, BX, '-', q1, BY, 'o-', X, Voltaje_aproximado, "-")
87 #pause
88
89

```

## C.14 Atenuacion.py

Este código comienza en la pagina que sigue.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los términos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15 import serial
16 import os
17 import subprocess
18 import math
19 import time
20 import Gnuplot
21 from Tkinter import *
22 import tkMessageBox
23 import Tkinter
24 import shutil
25
26 class Gramo():
27     def Atenua(self):
28         bicho = Tk()
29         bicho.geometry("280x170+200+90")
30         bicho.config(bg="white")
31         bicho.title("Infrarossi")
32         bicho.resizable(width=0, height=0)
33
34         def Verifica():
35             print "ola"
36
37         def Salir():
38             tkMessageBox.showinfo("Infrarossi", message= "Saliendo . . . ")
39             arduino = serial.Serial("/dev/rfcomm0", 9600)
40             arduino.write('aa')
41             exit()
42             exit()
43
44         def Grafica():
45             os.system("python g_p_Ate.py &")
46
47         def Comenzar1():
48             tkMessageBox.showinfo("Infrarossi", message= "Se procede a capturar
49             datos, para detener el proceso cierre la ventana de captura de datos
50             'de color azul'")
51             os.system("xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue -e python
52             bin/c_p_Ate.py &")
53             # os.system("python bin/c_p_Ate.py")
54
55 # -----CONFIGURACION DE VENTANA

```

```
53         X=8
54         Y=10
55         lblTitulo = Label(bicho, text="ATENUACION", fg = ("blue"), bg = ("white"),
56                               font = ("Century Schoolbook L",23)).place(x=30, y=20)
57         btnConectar1 = Button(bicho, text= " INICIAR ", width=5, height=1,
58                               command= Comenzar1).place(x=20+X, y=100+Y)
59         btnSalir = Button(bicho, text= " SALIR ", width=5, height=1, command=
60                             Salir).place(x=170+X, y=100+Y)
61         btnGrafica = Button(bicho, text= " GRAFICA ", width=5, height=1, command=
62                               Grafica).place(x=95+X, y=100+Y)
63     Verifica()
64     bicho.mainloop()
65
66
67     def __init__(self):
68         self.Atenua()
69         self.__del__()
70
71
72     def __del__(self):
73         print ("PROGRAMA TERMINADO")
74
75
76     modulo = Gramo()
```

## C.15 Absorcion.py

Este código comienza en la pagina que sigue.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los términos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Dr Julian Andres Salamanca Bernal
13 # Diego Alberto Parra Garzón
14 # Colombia, Bogota D.C.
15 import serial
16 import os
17 import subprocess
18 import math
19 import time
20 import Gnuplot
21 from Tkinter import *
22 import tkMessageBox
23 import Tkinter
24 import shutil
25
26 class Gramo():
27     def Absor(self):
28         bicho = Tk()
29         bicho.geometry("280x170+200+90")
30         bicho.config(bg="white")
31         bicho.title("Infrarossi")
32         bicho.resizable(width=0, height=0)
33
34         def Verifica():
35             print "ola"
36
37         def Salir():
38             tkMessageBox.showinfo("Infrarossi", message= "Saliendo . . . ")
39             arduino = serial.Serial("/dev/rfcomm0", 9600)
40             arduino.write('aa')
41             exit()
42             exit()
43
44         def Comenzar1():
45             tkMessageBox.showinfo("Infrarossi", message= "Se procede a capturar
46             datos, para detener el proceso cierre la ventana de captura de datos
47             'de color azul'")
48             os.system("xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue -e python
49             bin/c_p_Abs.py &")
50             # os.system("python bin/c_p_Ate.py")
51
52         def Grafica():
53             os.system("python g_p_Abs.py &")
54
55         # -----CONFIGURACION DE VENTANA

```

```
53         X=10
54         Y=10
55         lblTitulo = Label(bicho, text="ABSORCION", fg = ("blue"), bg = ("white"),
56                               font = ("Century Schoolbook L",23)).place(x=30, y=20)
57         btnConectar1 = Button(bicho, text= " INICIAR ", width=5, height=1, command=
58                               Comenzar1).place(x=20+X, y=100+Y)
59         btnSalir = Button(bicho, text= " SALIR ", width=5, height=1, command=
60                               Salir).place(x=170+X, y=100+Y)
61         btnGrafica = Button(bicho, text= " GRAFICA ", width=5, height=1, command=
62                               Grafica).place(x=95+X, y=100+Y)
63     Verifica()
64     bicho.mainloop()
65
66
67     def __init__(self):
68         self.Absor()
69         self.__del__()
70
71
72     def __del__(self):
73         print ("PROGRAMA TERMINADO")
74
75
76     modulo = Gramo()
```

## C.16 Difraccion.py

Este código comienza en la pagina que sigue.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los términos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15 import serial
16 import os
17 import subprocess
18 import math
19 import time
20 import Gnuplot
21 from Tkinter import *
22 import tkMessageBox
23 import Tkinter
24 import shutil
25
26 class App():
27     def Difrac(self):
28         bicho = Tk()
29         bicho.geometry("280x170+200+90")
30         bicho.config(bg="white")
31         bicho.title("Infrarossi")
32         bicho.resizable(width=0, height=0)
33
34     def Salir():
35         os.system('rfcomm -a > conexion.txt | cut -d ":" -f 1,1 > direccion.txt')
36         puerto = open('direccion.txt', 'r')
37         self.puerto = puerto.read(7)
38         puerto.close()
39         self.puerta = "/dev/" + self.puerto
40         print self.puerta
41         tkMessageBox.showinfo("Infrarossi", message="Saliendo . . . ")
42         arduino = serial.Serial(self.puerta, 9600)
43         arduino.write('aa')
44         exit()
45         exit()
46
47     def Comenzar1():
48         tkMessageBox.showinfo("Infrarossi", message="Se procede a capturar
49             datos, para detener el proceso cierre la ventana captura de datos
50             'color azul'")
51         os.system("xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue -e python
52             bin/c_p_Dif.py &")
53
54 # -----CONFIGURACION DE VENTANA

```

```
52         X=10
53         Y=10
54         lblTitulo = Label(bicho, text="DIFRACCION", fg = ("blue"), bg = ("white"),
55                               font = ("Century Schoolbook L",23)).place(x=30, y=20)
56         btnConectarl = Button(bicho, text= " INICIAR ", width=5, height=1,
57                               command= Comenzarl).place(x=20+X, y=100+Y)
58         btnSalir = Button(bicho, text= " SALIR ", width=5, height=1, command=
59                               Salir).place(x=160+X, y=100+Y)
60         bicho.mainloop()
61
62
63
64     def __init__(self):
65         self.Difrac()
66         self.__del__()
67
68
69 modulo = App()
70
```

## C.17 c\_p\_Abs.py

Este código comienza en la pagina que sigue.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los términos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.

15
16 import numpy as np
17 import os
18 import serial
19 import subprocess
20 import math
21 import time
22 import Gnuplot
23 import shutil
24 import matplotlib.pyplot as pl
25 class App:
26     def Verifica(self):
27         os.system('rfcomm -a > conexion.txt | cut -d ":" -f 1,1 > direccion.txt')
28         puerto = open('direccion.txt', 'r')
29         self.puerto = puerto.read(7)
30         puerto.close()
31         self.puerta = "/dev/" + self.puerto
32         print self.puerta
33         try:
34             arduino = serial.Serial(self.puerta, 9600)
35             arduino.write("aa")
36         except:
37             os.system("exit")
38
39     def Salir(self):
40         exit()
41
42     def Comenzar(self):
43         # Distancia de separación 28 cm
44         # Distancia de recolección 25 cm
45         # Paso en centímetros pausada 1 = 0,34
46         # Paso en centímetros pausada 2 = 0,30
47         # Paso en centímetros pausada 3 = 0,27
48         # Paso en centímetros pausada 4 = 0,245
49         # Paso en centímetros pausada 5 = 0.213
50         for n in range (0, 117):
51             os.system('rm datos/dat.dat')
52             #time.sleep(2)
53             arduino= serial.Serial(self.puerta, 9600)
54             print "aca va la lectura"

```

```
55         arduino.write("aa")
56         #time.sleep(1)
57         arduino.write('4')
58         arduino.close()
59         arduino=serial.Serial(self.puerta, 9600)
60         time.sleep(2)
61         arduino.write('zz')
62         for i in range(0, 140):
63             arduino=serial.Serial(self.puerta, 9600)
64             archi = open('datos/dat.dat', 'a+')
65             time.sleep(0.00005)
66             x = arduino.readline()
67             z = 0.21367*2*(140 - n)
68             xo = str(z)
69             yo = str(x)
70             print "paso numero", n
71             print "(cm) \t (microw)"
72             print('{0} {1}'.format(xo, yo))
73             archi.write (xo)
74             archi.write (" ")
75             archi.write (yo)
76             archi.close()
77         else:
78             os.system("octave bin/prom1.m")
79             archi = open('datos/prom.dat', 'a+')
80             print("aca va la pausa")
81             Lectura = archi.read()
82             archi.close()
83             archil = open('datos/dats1.dat', 'a+')
84             archil.write(Lectura)
85             archil.close()
86             arduino.write('aa')
87             arduino.close()
88
89     def Grafw(self):
90         os.system("python g_p_Abs.py")
91
92     def Ordenar(self):
93         os.system("python bin/o_Carpetas2.py")
94
95
96
97     def __init__(self):
98         self.Verifica()
99         self.Comenzar()
100        self.Grafw()
101        self.Ordenar()
102        self.__del__()
103
104    def __del__(self):
105        print ("PROGRAMA TERMINADO")
106
107
108 Iniciar = App()
109
```

## C.18 c\_p\_Dif.py

Este código comienza en la pagina que sigue.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los términos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15 import numpy as np
16 import os
17 import serial
18 import subprocess
19 import math
20 import time
21 import Gnuplot
22 import shutil
23 import matplotlib.pyplot as pl
24 class App:
25     def Verifica(self):
26         os.system('rfcomm -a > conexion.txt | cut -d ":" -f 1,1 > direccion.txt')
27         puerto = open('direccion.txt', 'r')
28         self.puerto = puerto.read(7)
29         puerto.close()
30         self.puerta = "/dev/" + self.puerto
31         print self.puerta
32         try:
33             arduino = serial.Serial(self.puerta, 9600)
34             arduino.write("aa")
35         except:
36             os.system("exit")
37
38     def Salir(self):
39         exit()
40
41     def Comenzar(self):
42         # Distancia de separación 28 cm
43         # Distancia de recolección 25 cm
44         # Paso en centímetros pausada 1 = 0,34
45         # Paso en centímetros pausada 2 = 0,30
46         # Paso en centímetros pausada 3 = 0,27
47         # Paso en centímetros pausada 4 = 0,245
48         # Paso en centímetros pausada 5 = 0,213
49         for n in range(0, 117):
50             os.system('rm datos/dat.dat')
51             #time.sleep(2)
52             arduino = serial.Serial(self.puerta, 9600)
53             print "aca va la lectura"
54             arduino.write("aa")

```

```

55             #time.sleep(1)
56             arduino.write('4')
57             arduino.close()
58             arduino=serial.Serial(self.puerta, 9600)
59             time.sleep(2)
60             arduino.write('zz')
61             for i in range(0, 140):
62                 arduino=serial.Serial(self.puerta, 9600)
63                 archi = open('datos/dat.dat', 'a+')
64                 time.sleep(0.00005)
65                 x = arduino.readline()
66                 z = 0.21367*2*(140 - n)
67                 xo = str(z)
68                 yo = str(x)
69                 print "paso numero", n
70                 print "(cm) \t (microw)"
71                 print '{0} {1}'.format(xo, yo)
72                 archi.write (xo)
73                 archi.write (" ")
74                 archi.write (yo)
75                 archi.close()
76             else:
77                 os.system("octave bin/prom1.m")
78                 archi = open('datos/prom.dat', 'a+')
79                 print("aca va la pausa")
80                 Lectura = archi.read()
81                 archi.close()
82                 archil = open('datos/dats1.dat', 'a+')
83                 archil.write(Lectura)
84                 archil.close()
85                 arduino.write('aa')
86                 arduino.close()
87
88             def Analisis(self):
89 #               os.system("python bin/Estadistica2.py")
90 #               os.system("python bin/estadis2.py")
91             time.sleep(3)
92
93
94             def Ordenar(self):
95                 os.system("python bin/o_Carpetas1.py")
96
97
98
99             def __init__(self):
100                 self.Verifica()
101                 self.Comenzar()
102                 self.Analisis()
103                 self.__del__()
104
105             def __del__(self):
106                 print ("PROGRAMA TERMINADO")
107
108
109             Iniciar = App()

```

## C.19 c\_p\_Ate.py

Este código comienza en la pagina que sigue.

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15 import numpy as np
16 import os
17 import serial
18 import subprocess
19 import math
20 import time
21 import Gnuplot
22 import shutil
23 import matplotlib.pyplot as pl
24 class App:
25     def Verifica(self):
26         os.system('rfcomm -a > conexion.txt | cut -d ":" -f 1,1 > direccion.txt')
27         puerto = open('direccion.txt', 'r')
28         self.puerto = puerto.read(7)
29         puerto.close()
30         self.puerta = "/dev/" + self.puerto
31         print self.puerta
32         try:
33             arduino = serial.Serial(self.puerta, 9600)
34             arduino.write("aa")
35         except:
36             os.system("exit")
37
38     def Salir(self):
39         exit()
40
41     def Salir(self):
42         exit()
43
44     def Comenzar(self):
45         # Distancia de separación 28 cm
46         # Distancia de recolección 25 cm
47         # Paso en centímetros pausada 1 = 0,34
48         # Paso en centímetros pausada 2 = 0,30
49         # Paso en centímetros pausada 3 = 0,27
50         # Paso en centímetros pausada 4 = 0,245
51         # Paso en centímetros pausada 5 = 0.213
52         for n in range(0, 117):
53             os.system('rm datos/dat.dat')
54             #time.sleep(2)
```

```

55         arduino= serial.Serial(self.puerta, 9600)
56         print "aca va la lectura"
57         arduino.write("aa")
58         #time.sleep(1)
59         arduino.write('4')
60         arduino.close()
61         arduino=serial.Serial(self.puerta, 9600)
62         time.sleep(2)
63         arduino.write('zz')
64         for i in range(0, 140):
65             arduino=serial.Serial(self.puerta, 9600)
66             archi = open('datos/dat.dat', 'a+')
67             time.sleep(0.00005)
68             x = arduino.readline()
69             z = 0.21367*2*(140 - n)
70             xo = str(z)
71             yo = str(x)
72             print "paso numero", n
73             print "(cm) \t (microw)"
74             print('{0} {1}'.format(xo, yo))
75             archi.write (xo)
76             archi.write (" ")
77             archi.write (yo)
78             archi.close()
79         else:
80             os.system("octave bin/prom1.m")
81             archi = open('datos/prom.dat', 'a+')
82             print("aca va la pausa")
83             Lectura = archi.read()
84             archi.close()
85             archil = open('datos/dats1.dat', 'a+')
86             archil.write(Lectura)
87             archil.close()
88             arduino.write('aa')
89             arduino.close()
90
91     def Grafw(self):
92         os.system("python g_p_Ate.py")
93
94     def Ordenar(self):
95         os.system("python bin/o_Carpetas.py")
96
97     def __init__(self):
98         self.Verifica()
99         self.Comenzar()
100        self.Grafw()
101        self.Ordenar()
102        self.__del__()
103
104    def __del__(self):
105        print ("PROGRAMA TERMINADO")
106
107
108 Iniciar = App()
109

```

## C.20 Estadis2.py

Este código comienza en la pagina que sigue.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los términos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15 from matplotlib.widgets import RectangleSelector
16 from numpy import *
17 import numpy as np
18 import matplotlib.pyplot as pl
19 import os
20 import subprocess
21 import math
22 import time
23 import shutil
24 import Gnuplot
25 from matplotlib.widgets import Cursor
26 from pylab import *
27 class Estadistica:
28     def Cargar(self):
29         self.f= np.loadtxt('datos/dats1.dat')
30         self.x , self.y = np.loadtxt('datos/dats1.dat', unpack=True, usecols=[0,1])
31 #        self.x , self.y = np.loadtxt('dats1.dat', unpack=True, usecols=[0,1])
32         self.y = self.y/10
33         self.n = np.size(self.x)
34         self.k = np.ceil(1 + np.log2(self.n)) # comando ceil redondea el numero al
35         #mayor entero
36         self.c = np.sort(self.y) #comando sort guarda los datos de y
37 #        self.lon = self.c[110]/8
38         print self.n
39 #        print "El tamaño de la muestra es: ", self.n , "\nEl tamaño del intervalo es "
40 #        self.k, self.lon
41
42         def Grafica(self):
43             def onselect(eclick, erelease):
44                 print(' startposition : (%f, %f)' % (eclick.xdata, eclick.ydata))
45                 print(' endposition   : (%f, %f)' % (erelease.xdata, erelease.ydata))
46                 print(' used button   : ', eclick.button)
47                 self.xinicial = round(eclick.xdata,2)
48                 self.yinicial= round(eclick.ydata,11)
49                 self.xfinal = round(erelease.xdata,2)
50                 self.yfinal = round(erelease.ydata,11)
51                 print self.xinicial, self.yinicial , self.xfinal, self.yfinal
52
53             def toggle_selector(event):
54                 print(' Key pressed.')
55                 if event.key in ['Q', 'q'] and toggle_selector.RS.active:

```

```

54         print(' RectangleSelector deactivated.')
55         toggle_selector.RS.set_active(False)
56     if event.key in ['A', 'a'] and not toggle_selector.RS.active:
57         print(' RectangleSelector activated.')
58         toggle_selector.RS.set_active(True)
59
60     pl.xlabel('Distancia [m]')
61     pl.ylabel('Intensidad [microW]')
62     pl.title('LONGITUD DE ONDA DIODO INFRARROJO \n')
63     fig = figure
64     ax = subplot(111)
65     # pl.legend(loc='upper left')
66     ax.plot(self.x, self.y, 'o--')
67     cursor = Cursor(ax, useblit=True, color='red', linewidth=2)
68     toggle_selector.RS = RectangleSelector(ax, onselect, drawtype='line')
69     connect('key_press_event', toggle_selector)
70     pl.subplots_adjust(right=0.97)
71     pl.subplots_adjust(left=0.18)
72     pl.subplots_adjust(bottom=0.13)
73     pl.subplots_adjust(top=0.87)
74     pl.subplots_adjust(wspace=0.32)
75     pl.subplots_adjust(hspace=0.71)
76     pl.show()
77
78     def Estadistica(self):
79         self.distancia = self.xfinal - self.xinicial
80         d = 4.81*10**(-6)
81         x = self.distancia*10**(-2)
82         y = 45*10**(-2)
83         tetharad = math.atan(x/y)
84         tethagra = math.degrees(tetharad)
85         sintetha= math.sin(tetharad)
86         self.lamda = round((2*d*sintetha)*10**(9),2)
87         self.error = round(pow(800-self.lamda,2),0.5), 2)
88         self.error1 = round(pow(pow(100-self.lamda*100/800, 2), 0.5), 3)
89         print tethagra, "\n ", sintetha
90         print "la longitud de onda aproximada para el diodo infrarrojo es", self.
91         lamda , "nanometros, con un error de ", self.error1 , " %"
92
93     def Grafical(self):
94         self.x1 = self.x/100
95         self.y1 = self.y/1000000
96         pl.subplot(221)
97         pl.plot(self.x1,self.y1, 'o--')
98         pl.title('Datos Capturados \n')
99         pl.xlabel('Distancia [m]')
100        pl.ylabel('Intensidad [W]')
101
102    def Grafica2(self):
103        pl.subplot(222)
104        pl.plot(self.x1, self.y1, 'o--')
105        pl.title('Patrones de Interferencia \n')
106        pl.xlabel('Distancia [m]')
107        pl.ylabel('Intensidad [W]')

```

```

108     pl.text(0.001, 0.0000030, r' x1 = ' + str(self.xinicial/100))
109     pl.text(0.001, 0.0000025, r' x2 = ' + str(self.xfinal/100))
110     pl.text(0.001, 0.0000020, r' x2-x1 = ' + str(self.distancia/100))
111 #    pl.ylim(0, 0.036)
112 #        pl.plot([self.xinicial/100, self.xinicial/100], [0, 0.000005], '-')
113 #        pl.plot([self.xfinal/100, self.xfinal/100], [0, 0.000005], '-')
114 #        pl.plot(self.x1, self.y1, 'o--')
115 #    pl.ylim(0, 0.022)
116
117     def Grafica3(self):
118         pl.subplot(212)
119         pl.plot(self.x1, self.y1, 'o--')
120         pl.plot([self.xinicial/100, self.xinicial/100], [0, 0.000005], '-')
121         pl.plot([self.xfinal/100, self.xfinal/100], [0, 0.000005], '-')
122 #        pl.plot([self.xinicial/100, self.xfinal/100], [0, 0.0000000001], '-')
123         pl.text(0.001, 0.0000030, r' Distancia entre asintotas '+str(self.distancia/ 100) + '[m]')
124         pl.text(0.001, 0.0000025, r' m:1 Lamda IRE =' + str(self.lamda) + ' [nm] +/- ' + str(self.error) + ' [nm]')
125         pl.text(0.001, 0.0000020, r' Error porcentual ' + str(self.error1) + ' %')
126 #    pl.ylim(0, 0.022)
127         pl.xlabel('Distancia [m]')
128         pl.ylabel('Intensidad [W]')
129         pl.title('LONGITUD DE ONDA DIODO INFRARROJO \n')
130
131     def Plotear(self):
132         pl.subplots_adjust(right=0.97)
133         pl.subplots_adjust(left=0.18)
134         pl.subplots_adjust(bottom=0.13)
135         pl.subplots_adjust(top=0.87)
136         pl.subplots_adjust(wspace=0.62)
137         pl.subplots_adjust(hspace=0.71)
138         pl.savefig('datos/Graficas.png')
139         pl.show()
140
141     def ordena(self):
142         os.system("python bin/o_Carpetas1.py")
143
144     def __init__(self):
145         self.Cargar()
146         self.Grafica()
147         self.Estadistica()
148         self.Grafical()
149         self.Grafica2()
150         self.Grafica3()
151         self.Plotear()
152         self.ordera()
153
154     esto = Estadistica()
155

```

## C.21 o\_Carpetas.py

Este código comienza en la pagina que sigue.

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15
16 import os
17 import time
18 import shutil
19 import Gnuplot
20
21 class o_Carpetas:
22
23     def Carpeta(self):
24         self.Carpeta = str(time.ctime())
25         os.mkdir(self.Carpeta)
26
27     def c_Carpeta(self):
28         archi = open("datos/name.dat","w")
29         archi.write(self.Carpeta)
30         archi.close()
31         time.sleep(3)
32         shutil.move("datos/Yteo.dat", self.Carpeta)
33         shutil.move("datos/Yest.dat", self.Carpeta)
34         shutil.move("datos/F.dat", self.Carpeta)
35         shutil.move("datos/a.dat", self.Carpeta)
36         shutil.move("datos/b.dat", self.Carpeta)
37         shutil.move("datos/ECM.dat", self.Carpeta)
38         shutil.move("datos/dats1.dat", self.Carpeta)
39         shutil.move("datos/Atenuacion.png", self.Carpeta)
40         os.system("sh bin/m_Carpeta.sh")
41         os.system("rm datos/dat.dat prom.dat")
42
43     def __init__(self):
44         self.Carpeta()
45         self.c_Carpeta()
46         self.__del__()
47
48     def __del__(self):
49         print ("PROGRAMA TERMINADO")
50
51
52 if __name__ == "__main__":
53     o_Carpetas()
```

## C.22 o\_Carpetas1.py

Este código comienza en la pagina que sigue.

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15
16 import os
17 import time
18 import shutil
19 import Gnuplot
20
21 class o_Carpetas:
22
23     def Carpeta(self):
24         self.Carpeta = str(time.ctime())
25         os.mkdir(self.Carpeta)
26
27     def c_Carpeta(self):
28         archi = open("datos/name.dat","w")
29         archi.write(self.Carpeta)
30         archi.close()
31         time.sleep(3)
32         shutil.move("datos/dats1.dat", self.Carpeta)
33         shutil.move("datos/Graficas.png", self.Carpeta)
34
35     def Llama(self):
36         os.system("sh bin/m_Carpeta1.sh")
37
38     def __init__(self):
39         self.Carpeta()
40         self.c_Carpeta()
41         self.Llama()
42         self.__del__()
43
44     def __del__(self):
45         print ("PROGRAMA TERMINADO")
46
47
48 if __name__ == "__main__":
49     o_Carpetas()
```

### C.23 o\_Carpetas2.py

Este código comienza en la pagina que sigue.

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15
16 import os
17 import time
18 import shutil
19 import Gnuplot
20
21 class o_Carpetas:
22
23     def Carpeta(self):
24         self.Carpeta = str(time.ctime())
25         os.mkdir(self.Carpeta)
26
27     def c_Carpeta(self):
28         archi = open("datos/name.dat","w")
29         archi.write(self.Carpeta)
30         archi.close()
31         time.sleep(3)
32         shutil.move("datos/Yteo.dat", self.Carpeta)
33         shutil.move("datos/Yest.dat", self.Carpeta)
34         shutil.move("datos/F.dat", self.Carpeta)
35         shutil.move("datos/Amplitud.dat", self.Carpeta)
36         shutil.move("datos/BY.dat", self.Carpeta)
37         shutil.move("datos/ECM1.dat", self.Carpeta)
38         shutil.move("datos/ql.dat", self.Carpeta)
39         shutil.move("datos/Transmitancia.dat", self.Carpeta)
40         shutil.move("datos/V_aproximado.dat", self.Carpeta)
41         shutil.move("datos/Vteo.dat", self.Carpeta)
42         shutil.move("datos/a.dat", self.Carpeta)
43         shutil.move("datos/b.dat", self.Carpeta)
44         shutil.move("datos/ECM.dat", self.Carpeta)
45         shutil.move("datos/dats1.dat", self.Carpeta)
46         shutil.move("datos/Absorcion.png", self.Carpeta)
47         os.system("sh bin/m_Carpeta2.sh")
48         os.system("rm datos/dat.dat prom.dat")
49
50     def __init__(self):
51         self.Carpeta()
52         self.c_Carpeta()
53         self.__del__()
54
55     def __del__(self):
```

```
56         print ("PROGRAMA TERMINADO")
57
58
59 if __name__ == "__main__":
60     o_Carpetas()
61
```

## C.24 m\_Carpeta.sh

Este código comienza en la pagina que sigue.

```
1 #!/bin/bash
2 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
3 # los términos de la licencia pública general de GNU, según es publicada
4 # por la free software fundation bien la versión 3 de la misma licencia
5 # o de cualquier versión posterior. (según su elección).
6 # Si usted hace alguna modificación en esta aplicación, deberá siempre
7 # mencionar el autor original de la misma.
8 # Autor:
9 # Universidad Distrital Francisco Jose
10 # Grupo de fisica e informatica
11 # Diego Alberto Parra Garzón
12 # Dr Julian Andres Salamanca Bernal
13 # Colombia, Bogota D.C.
14
15 ARCHIVO=`cat datos/name.dat`
16 mv -f "$ARCHIVO" Carpetas/Atenuacion
17
18
```

## C.25 m\_Carpetas1.sh

Este código comienza en la pagina que sigue.

```
1 #!/bin/bash
2 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
3 # los términos de la licencia pública general de GNU, según es publicada
4 # por la free software fundation bien la versión 3 de la misma licencia
5 # o de cualquier versión posterior. (según su elección).
6 # Si usted hace alguna modificación en esta aplicación, deberá siempre
7 # mencionar el autor original de la misma.
8 # Autor:
9 # Universidad Distrital Francisco Jose
10 # Grupo de fisica e informatica
11 # Diego Alberto Parra Garzón
12 # Dr Julian Andres Salamanca Bernal
13 # Colombia, Bogota D.C.
14 ARCHIVO=`cat datos/name.dat`
15 mv -f "$ARCHIVO" Carpetas/Difraccion
16
17
```

## C.26 m\_Carpetas2.sh

Este código comienza en la pagina que sigue.

```
1 #!/bin/bash
2 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
3 # los términos de la licencia pública general de GNU, según es publicada
4 # por la free software fundation bien la versión 3 de la misma licencia
5 # o de cualquier versión posterior. (según su elección).
6 # Si usted hace alguna modificación en esta aplicación, deberá siempre
7 # mencionar el autor original de la misma.
8 # Autor:
9 # Universidad Distrital Francisco Jose
10 # Grupo de fisica e informatica
11 # Diego Alberto Parra Garzón
12 # Dr Julian Andres Salamanca Bernal
13 # Colombia, Bogota D.C.
14 ARCHIVO=`cat datos/name.dat`
15 mv -f "$ARCHIVO" Carpetas/Absorcion
16
17
```

## C.27 G\_firmware.py

Este código comienza en la pagina que sigue.

```

1 #!/usr/bin/python
2 #!-*- coding:utf-8 *-*
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15 import numpy as np
16 import pylab as pl
17 import serial
18 import os
19 import subprocess
20 import math
21 import time
22 import Gnuplot
23 from Tkinter import *
24 import tkMessageBox
25 import Tkinter
26 import shutil
27
28 class G_firmware:
29     def Difraccion(self):
30         def Salir():
31             tkMessageBox.showinfo("free_infrarossi", message= "! Cerrando el
32                                     programa i")
33             exit()
34
35         def Continuar():
36             tkMessageBox.showinfo("free_infrarossi", message= "Cargando el firmware
37             en la tarjeta microcontroladora. \n\n si no carga el firmware: \n\n*
38             Revise su conexión con la tarjeta programadora arduino uno. \n\n* Revise
39             el microcontrolador que este funcionando bien ")
40             os.system("python bin/firmware/firmware_free_infrarossi.py &")
41
42         bicho = Tk()
43         bicho.geometry("280x180+507+60")
44         bicho.config(bg="white")
45         bicho.title("free_infrarossi")
46         bicho.resizable(width=0, height=0)
47         yn= 400
48         xn = 350
49         # imgBoton2=PhotoImage(file="Imagenes/cap8.gif")
50         # btnLogo= Label(bicho, image=imgBoton2, height=150, width
51 =180).place(x=400+xn, y=215+yn)
52         # lblFisinfor = Label(bicho, text=" GRUPO DE FISICA E INFORMATICA ", fg =
53 =("black"), bg = ("white"), font = ("Century Schoolbook L",10)).place(x=360+xn,
54 =y=371+yn)
55         # lblInfo = Label(bicho, text="Dr. Julian Andres Salamanca\n Diego Alberto
56 ="

```

```
49 Parra Garzón", fg = ("black"), bg = ("white"), font = ("Century Schoolbook L",10)).place(x=400+xn, y=390+yn)
50     lblFisinfor = Label(bicho, text="DEBE TENER LA TARJETA
51         \nMICROCONTROLADORA ARDUINO\n CONECTADA PARA CONTINUAR", fg = ("red"), bg
52             = ("white"), font = ("Century Schoolbook L",11)).place(x=350-xn, y=440-
53                 yn)
54     btnSalir=Button(bicho, text = "Salir", command=Salir, height=1, width =5).
55         place(x=20, y=540-yn)
56     btnContinuar=Button(bicho, text = "Continuar", command=Continuar, height=1,
57         width =8).place(x=168, y=540-yn)
58
59     bicho.mainloop()
60
61
62
63
64     def __init__(self):
65         self.Difraccion()
66         self.__del__()
67
68     def __del__(self):
69         print ("PROGRAMA TERMINADO")
70
71
72
73
74 if __name__ == "__main__":
75     G_firmware()
```

## C.28 firmware\_free\_infrarossi.py

Este código comienza en la pagina que sigue.

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4 # los términos de la licencia pública general de GNU, según es publicada
5 # por la free software fundation bien la versión 3 de la misma licencia
6 # o de cualquier versión posterior. (según su elección).
7 # Si usted hace alguna modificación en esta aplicación, deberá siempre
8 # mencionar el autor original de la misma.
9 # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15 import serial
16 import os
17 import time
18
19 class Firmware:
20     def b_Puerto(self):
21         os.system("ls /dev/ttyACM* > puerto.txt")
22         archi=open("puerto.txt","r")
23         puerto = archi.readline()
24         self.puerto = str(puerto)
25         archi.close()
26         print self.puerto
27
28     def c_arch_Carga(self):
29         archi=open("bin/firmware/a_carga.sh", "a+")
30         archi.write("#!/bin/bash")
31         archi.write("\n")
32         archi.write("CARGA=`avrduude -F -V -c arduino -p ATMEGA328P -b 115200 -U
33         flash:w:bin/firmware/infrarossi.hex -P `")
34         archi.write(self.puerto)
35         archi.write(``)
36         archi.write("\n")
37         # archi.write("\n")
38         # archi.write("echo firmware instalado correctamente")
39         archi.write("\n")
40         archi.write("sleep 10")
41         archi.close()
42         time.sleep(1)
43         os.system("xterm -T infrarossi -geom 82x27+0+55 +cm -bg blue -e 'sh
44         bin/firmware/a_carga.sh' &")
45         time.sleep(8)
46
47     def limpiar(self):
48         os.system("rm bin/firmware/a_carga.sh bin/firmware/puerto.txt")
49
50     def __init__(self):
51         self.b_Puerto()
52         self.c_arch_Carga()
53         self.limpiar()
```

```
54
55 if __name__ == "__main__":
56     Firmware()
57
```

## C.29 infrarossi.cpp

Este código comienza en la pagina que sigue.

```
1 // Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
2 // los términos de la licencia pública general de GNU, según es publicada
3 // por la free software fundation bien la versión 3 de la misma licencia
4 // o de cualquier versión posterior. (según su elección).
5 // Si usted hace alguna modificación en esta aplicación, deberá siempre
6 // mencionar el autor original de la misma.
7 // Autor:
8 // Universidad Distrital Francisco Jose
9 // Grupo de fisica e informatica
10 // Diego Alberto Parra Garzón
11 // Dr Julian Andres Salamanca Bernal
12 // Colombia, Bogota D.C.
13 #include "Arduino.h"
14 void setup();
15 void Velo0();
16 void Velo1();
17 void Velo2();
18 void Velo3();
19 void Velo4();
20 void Velo5();
21 void Velo6();
22 void Vepa1();
23 void Vepa2();
24 void Vepa3();
25 void Vepa4();
26 void Vepa5();
27 void Inten1();
28 void Inten2();
29 void inicio();
30 void Menu();
31 void loop();
32 #line 1
33 int motor = 3;
34 const int sensor1 = A0;
35 const int sensor2 = A5;
36 const int ledRojo = 12;
37 const int ledAzul = 9;
38 const int ledVerde = 10;
39 const int ledEmisor = 6;
40 //const int resetea = 4;
41 //variables
42 long miliVolts1;
43 long intensidad1;
44 int brillol;
45
46
47 void setup()
48 {
49     Serial.begin(9600);
50     pinMode(ledAzul, OUTPUT);
51     pinMode(ledVerde, OUTPUT);
52     pinMode(ledRojo, OUTPUT);
53     pinMode(ledEmisor, OUTPUT);
54 //    pinMode(resetea, OUTPUT);
55 }
```

```
56
57 void Velo0()
58 {
59     //int velocidad = map(velocidad,'0', '5', 0, 255);
60     //velocidad = 0;
61     //analogWrite(motor, velocidad);
62     analogWrite(motor, 0);
63     delay(20);
64 }
65
66 void Velo1()
67 {
68     //int velocidad = map(velocidad,'0', '255', 0, 255);
69     //velocidad = '150';
70     //analogWrite(motor, velocidad);
71     analogWrite(motor, 135);
72     //delay(20);
73 }
74
75 void Velo2()
76 {
77     //int velocidad = map(velocidad,'0', '255', 0, 255);
78     //velocidad = '160';
79     //analogWrite(motor, velocidad);
80     analogWrite(motor, 140);
81     //delay(20);
82 }
83
84 void Velo3()
85 {
86     //int velocidad = map(velocidad,'0', '255', 0, 255);
87     //velocidad = '170';
88     //analogWrite(motor, velocidad);
89     analogWrite(motor, 145);
90     //delay(20);
91 }
92
93 void Velo4()
94 {
95     //int velocidad = map(velocidad,'0', '255', 0, 255);
96     //velocidad = '180';
97     //analogWrite(motor, velocidad);
98     analogWrite(motor, 150);
99     //delay(20);
100 }
101
102 void Velo5()
103 {
104     //int velocidad = map(velocidad,'0', '255', 0, 255);
105     //velocidad = '190';
106     //analogWrite(motor, velocidad);
107     analogWrite(motor, 160);
108     //delay(20);
109 }
110 }
```

```
111 void Velo6()
112 {
113     //int velocidad = map(velocidad,'0', '255', 0, 255);
114     //velocidad = '200';
115     //analogWrite(motor, velocidad);
116     analogWrite(motor, 190);
117     //    delay(20);
118 }
119
120 void Vepa1()
121 {
122     analogWrite(motor, 150);
123     delay(55);
124     Velo0();
125     delay(1000);
126     // digitalWrite(resetea, HIGH);
127 }
128
129
130 void Vepa2()
131 {
132     analogWrite(motor, 150);
133     delay(50);
134     Velo0();
135     delay(1000);
136 }
137
138 void Vepa3()
139 {
140     analogWrite(motor, 150);
141     delay(45);
142     Velo0();
143     delay(1000);
144 }
145
146 void Vepa4()
147 {
148     analogWrite(motor, 150);
149     delay(40);
150     Velo0();
151     delay(1000);
152 }
153
154 void Vepa5()
155 {
156     analogWrite(motor, 150);
157     delay(37);
158     Velo0();
159     delay(1000);
160 }
161
162 void Inten1()
163 {
164     miliVolts1 = ((analogRead(sensor1) *5000L) /1023)*4; //obteniendo el valor sensor
165     intensidad1 =miliVolts1;
```

```

166    brillo1 = map(intensidad1, 0, 5000, 0, 255); //funcion map (mapeo) convierte la ↵
167    variable y le da un rango y un dominio ↵
168    brillo1 = constrain(brillo1, 0, 255); //funcion constrain o contenido en el ↵
169    intervalo de analogWrite (0, 255) ↵
170    digitalWrite(ledAzul, 255 - brillo1 ); //Salida del led si esta el obstaculo ↵
171    esta lejos ↵
172    digitalWrite(ledVerde, brillo1); //Salida del led si el obstaculo esta cerca ↵
173    Serial.print(" "); //salida al Serialporth ↵
174    Serial.print(intensidad1); ↵
175    Serial.println(" "); ↵
176    delay (3); ↵
177 }
178
179 void Inten2()
180 {
181     miliVolts1 = ((analogRead(sensor2) *5000L) /(1023))*4; //obteniendo el valor ↵
182     sensor ↵
183     intensidad1 =miliVolts1; ↵
184     brillo1 = map(intensidad1, 0, 5000, 0, 255); //funcion map (mapeo) convierte la ↵
185     variable y le da un rango y un dominio ↵
186     brillo1 = constrain(brillo1, 0, 255); //funcion constrain o contenido en el ↵
187     intervalo de analogWrite (0, 255) ↵
188     digitalWrite(ledEmisor, 255); ↵
189     digitalWrite(ledAzul, 255-brillo1 ); //Salida del led si esta el obstaculo esta ↵
190     lejos ↵
191     digitalWrite(ledVerde, brillo1); //Salida del led si el obstaculo esta cerca ↵
192     Serial.print(" "); //salida al Serialport ↵
193     Serial.print(intensidad1); ↵
194     Serial.println(" "); ↵
195     delay (3); ↵
196
197 //void resetear()
198 //{
199 //    digitalWrite(resetea, HIGH);
200 //    delay(150);
201 //    digitalWrite(resetea, LOW);
202 //    delay(150);
203 //}
204
205 void inicio()
206 {
207     digitalWrite(ledRojo, HIGH); // turn the LED on (HIGH is the voltage level)
208     digitalWrite(ledAzul, 255);
209     digitalWrite(ledVerde, 255);
210     delay(100); // wait for a second
211     digitalWrite(ledRojo, LOW); // turn the LED off by making the voltage LOW
212     digitalWrite(ledAzul, 255);
213     digitalWrite(ledVerde, 255);
214     delay(100);
215 }
216
217 void Menu()
218 {
219 }
```

```
214     char opcion = Serial.read();
215     switch (opcion)
216     {
217         case 'a':
218
219             while(opcion=='a')
220             {
221                 inicio();
222                 Velo0();
223                 Menu();
224             }
225             break;
226
227         case 'b':
228
229             while(opcion=='b')
230             {
231                 inicio();
232                 Velo1();
233                 Menu();
234             }
235             break;
236
237         case 'c':
238             while(opcion=='b')
239             {
240                 inicio();
241                 Velo2();
242                 Menu();
243             }
244             break;
245
246         case 'd':
247             while(opcion=='d')
248             {
249                 inicio();
250                 Velo3();
251                 Menu();
252             }
253             break;
254
255         case 'e':
256             while(opcion=='e')
257             {
258                 inicio();
259                 Velo4();
260                 Menu();
261             }
262             break;
263
264         case 'f':
265             while(opcion=='f')
266             {
267
```

```
269             inicio();
270             Velo5();
271             Menu();
272         }
273     break;
274
275 case 'g':
276     while(opcion=='g')
277     {
278         inicio();
279         Velo6();
280         Menu();
281     }
282     break;
283
284 case 'h':
285     while(opcion=='h')
286     {
287         Velo0();
288         Inten1();
289         Menu();
290     }
291     break;
292
293 case 'i':
294     while(opcion=='i')
295     {
296         Velo1();
297         Inten1();
298         Menu();
299     }
300     break;
301
302 case 'j':
303     while(opcion=='j')
304     {
305         Velo2();
306         Inten1();
307         Menu();
308     }
309     break;
310
311 case 'k':
312
313     while(opcion=='k')
314     {
315         Velo3();
316         Inten1();
317         Menu();
318     }
319     break;
320
321 case 'l':
322     while(opcion=='l')
323     {
```

```
324             Velo4();
325             Inten1();
326             Menu();
327         }
328         break;
329
330     case 'm':
331         while(opcion=='m')
332         {
333             Velo5();
334             Inten1();
335             Menu();
336         }
337         break;
338
339     case 'n':
340         while(opcion=='n')
341         {
342             Velo6();
343             Inten1();
344             Menu();
345         }
346         break;
347
348     case 'z':
349         while(opcion=='z')
350         {
351             Velo0();
352             Inten2();
353             Menu();
354         }
355         break;
356
357     case 'o':
358         while(opcion=='o')
359         {
360             Velo1();
361             Inten2();
362             Menu();
363         }
364         break;
365
366     case 'p':
367         while(opcion=='p')
368         {
369             Velo2();
370             Inten2();
371             Menu();
372         }
373         break;
374
375     case 'q':
376         while(opcion=='q')
377         {
378             Velo3();
```

```
379             Inten2();
380             Menu();
381         }
382         break;
383
384     case 'r':
385         while(opcion=='r')
386     {
387             Velo4();
388             Inten2();
389             Menu();
390         }
391         break;
392
393     case 's':
394         while(opcion=='s')
395     {
396             Velo5();
397             Inten2();
398             Menu();
399         }
400         break;
401
402     case 't':
403         while(opcion=='t')
404     {
405             Velo6();
406             Inten2();
407             Menu();
408         }
409         break;
410 //case 'u':
411 //    {
412 //        resetear();
413 //        Menu();
414 //    }
415 //    break;
416
417 case '1':
418     {
419         Vepa1();
420         Menu();
421     }
422     break;
423
424 case '2':
425     {
426         Vepa2();
427         Menu();
428     }
429     break;
430 case '3':
431     {
432         Vepa3();
433         Menu();
```

```
434 }  
435     break;  
436     case '4':  
437     {  
438         Vepa4();  
439         Menu();  
440     }  
441     break;  
442     case '5':  
443     {  
444         Vepa5();  
445         Menu();  
446     }  
447     break;  
448 }  
449 }  
450 }  
451  
452 void loop()  
453 {  
454     inicio();  
455     Menu();  
456 }  
457 }  
458 }  
459 }  
460 }
```

### **C.30 infrarossi.hex**

Este código comienza en la pagina que sigue.

1	:100000000C94A7000C94CF000C94CF000C94CF005C
2	:100010000C94CF000C94CF000C94CF000C94CF0024
3	:100020000C94CF000C94CF000C94CF000C94CF0014
4	:100030000C94CF000C94CF000C94CF000C94CF0004
5	:100040000C940F040C94CF000C940E070C945107E1
6	:100050000C94CF000C94CF000C94CF000C94CF00E4
7	:100060000C94CF000C94CF006303660369036C0308
8	:100070006F03720372037203720372037203DB
9	:100080007203720372037203720372037203C8
10	:100090007203720372037203720372037203B8
11	:1000A0007203720372037203720372037203A8
12	:1000B000720372037203720372037203720398
13	:1000C0007203720372037203D702DE027203E50247
14	:1000D000EC02F302FA02010308030F0316031D03E7
15	:1000E00024032B033903400347034E0355035C03EA
16	:1000F00072037203720372033203000000077B
17	:100100000002010000030406000000000000000DF
18	:10011000010204081020408001020408102001029E
19	:1001200004081020040404040404020202026B
20	:100130000202030303030300000000250028005C
21	:100140002B0000000000240027002A000D081124C5
22	:100150001FBECFEFD8E0DEBFCDBF11E0A0E0B1E021
23	:10016000E8E4F2E102C005900D92AA31B107D9F797
24	:1001700011E0AAE1B1E001C01D92A73DB107E1F78E
25	:1001800011E0CEE4D1E004C02297FE010E940409F0
26	:10019000CC34D107C9F70E9459050C9422090C945C
27	:1001A000000040E855E260E070E08DE291E00E94DE
28	:1001B0008C0761E089E00E94A00361E08AE00E9470
29	:1001C000A00361E08CE00E94A00361E086E00C9453
30	:1001D000A00360E070E0809104010E94FA0464E1F1
31	:1001E00070E080E090E00C947C0467E870E080911F
32	:1001F00004010C94FA046CE870E0809104010C9402
33	:10020000FA0461E970E0809104010C94FA0466E953
34	:1002100070E0809104010C94FA0460EA70E080912F
35	:1002200004010C94FA046EEB70E0809104010C94CC
36	:10023000FA0466E970E0809104010E94FA0467E321
37	:1002400070E080E090E00E947C040E94E90068EE8B
38	:1002500073E080E090E00C947C0466E970E08091AB
39	:1002600004010E94FA0462E370E080E090E00E94E2
40	:100270007C040E94E90068EE73E080E090E00C945A
41	:100280007C0466E970E0809104010E94FA046DE24A
42	:1002900070E080E090E00E947C040E94E90068EE3B
43	:1002A00073E080E090E00C947C0466E970E080915B
44	:1002B00004010E94FA0468E270E080E090E00E948D
45	:1002C0007C040E94E90068EE73E080E090E00C940A
46	:1002D0007C0466E970E0809104010E94FA0465E202
47	:1002E00070E080E090E00E947C040E94E90068EEB
48	:1002F00073E080E090E00C947C04AF92BF92CF92C8
49	:10030000DF92EF92FF920F931F938EE00E94DD0425
50	:1003100028E833E1DC010E94AA082FEF33E040E037
51	:1003200050E00E94E308CA01B90122E0660F771F7E
52	:10033000881F991F2A95D1F760932001709321019E
53	:10034000809322019093230160931C0170931D01FF
54	:1003500080931E0190931F011F921F921F922FEFF7
55	:100360002F93A12CB12C650118E8E12E13E1F12E99

```

56 : 100370000E010E020E030E0A9010E9463080F9047
57 : 10038000F900F900F906F3F710519F014F06FEF01
58 : 1003900070E077FF02C060E070E070931B01609333
59 : 1003A0001A018FEF90E0861B970BBC0189E00E9439
60 : 1003B000FA0460911A0170911B018AE00E94FA040C
61 : 1003C00067E071E08DE291E00E94A40540911C017C
62 : 1003D00050911D0160911E0170911F012AE030E0D3
63 : 1003E0008DE291E00E942C0667E071E08DE291E0E1
64 : 1003F0000E94C10563E070E080E090E01F910F91E2
65 : 10040000FF90EF90DF90CF90BF90AF900C947C0462
66 : 10041000AF92BF92CF92DF92EF92FF920F931F9312
67 : 1004200083E10E94DD0428E833E1DC010E94AA0890
68 : 100430002FEF33E040E050E00E94E308CA01B90129
69 : 1004400042E0660F771F881F991F4A95D1F7609386
70 : 1004500020017093210180932201909323016093E6
71 : 100460001C0170931D0180931E0190931F011F9228
72 : 100470001F921F922FEF2F93A12CB12C650118E82A
73 : 10048000E12E13E1F12E00E010E020E030E0A901C0
74 : 100490000E9463080F900F900F900F906F3F7105AF
75 : 1004A00019F014F06FEF70E077FF02C060E070E0C9
76 : 1004B00070931B0160931A016FEF70E086E00E9459
77 : 1004C000FA046FEF70E080911A0190911B01681B94
78 : 1004D000790B89E00E94FA0460911A0170911B0166
79 : 1004E0008AE00E94FA0467E071E08DE291E00E94E8
80 : 1004F000A40540911C0150911D0160911E01709155
81 : 100500001F012AE030E08DE291E00E942C0667E0B6
82 : 1005100071E08DE291E00E94C10563E070E080E04F
83 : 1005200090E01F910F91FF90EF90DF90CF90BF90E0
84 : 10053000AF900C947C0461E08CE00E94D9036FEFD3
85 : 1005400070E089E00E94FA046FEF70E08AE00E9498
86 : 10055000FA0464E670E080E090E00E947C0460E0D1
87 : 1005600070E08CE00E94FA046FEF70E089E00E9476
88 : 10057000FA046FEF70E08AE00E94FA0464E670E02B
89 : 1005800080E090E00C947C048DE291E00E94A30650
90 : 10059000992787FD9095A92FB92FFC01F197EA348F
91 : 1005A000F10508F09FC0EC5CFF4F0C9402090E941B
92 : 1005B0009B020E94E9000E94C402F9CF0E949B02A4
93 : 1005C0000E94F5000E94C402F9CF0E949B020E9483
94 : 1005D00001010E94C402F9CF0E949B020E94070100
95 : 1005E0000E94C402F9CF0E949B020E940D010E944A
96 : 1005F000C402F9CF0E949B020E9413010E94C40210
97 : 10060000F9CF0E94E9000E947D010E94C402F9CF47
98 : 100610000E94F5000E947D010E94C402F9CF0E9451
99 : 10062000FB000E947D010E94C402F9CF0E940101DB
100 : 100630000E947D010E94C402F9CF0E9407010E941E
101 : 100640007D010E94C402F9CF0E940D010E947D012C
102 : 100650000E94C402F9CF0E9413010E947D010E94F2
103 : 10066000C402F9CF0E94E9000E9408020E94C4025D
104 : 10067000F9CF0E94F5000E9408020E94C402F9CF3F
105 : 100680000E94FB000E9408020E94C402F9CF0E944F
106 : 1006900001010E9408020E94C402F9CF0E940701D2
107 : 1006A0000E9408020E94C402F9CF0E940D010E941C
108 : 1006B00008020E94C402F9CF0E9413010E9408029E
109 : 1006C0000E94C402F9CF0E9419015ECF0E942D0141
110 : 1006D0005BCF0E94410158CF0E94550155CF0E9427

```

111	:1006E000690152CF08950E949B020C94C40283308A
112	:1006F00081F028F4813099F08230A1F0089586309D
113	:10070000A9F08730B9F08430D1F4809180008F7DDA
114	:1007100003C0809180008F778093800089584B516
115	:100720008F7702C084B58F7D84BD08958091B0001D
116	:100730008F7703C08091B0008F7D8093B0000895C3
117	:10074000CF93DF9390E0FC01E05FFE4F2491FC012A
118	:10075000EC5DFE4F8491882349F190E0880F991F4A
119	:10076000FC01EE5BFE4FA591B491885C9E4FFC01AD
120	:10077000C591D4919FB7611108C0F8948C912095D0
121	:1007800082238C93888182230AC0623051F4F894CA
122	:100790008C91322F309583238C938881822B888390
123	:1007A00004C0F8948C91822B8C939FBFDF91CF91E2
124	:1007B00008950F931F93CF93DF931F92CDB7DEB7AA
125	:1007C000282F30E0F901E450FF4F8491F901E05FF8
126	:1007D000FE4F1491F901EC5DFE4F04910023C9F026
127	:1007E000882321F069830E9477036981E02FF0E07C
128	:1007F000EE0FFF1FE85CFE4FA591B4919FB7F894F0
129	:100800008C91611103C01095812301C0812B8C93C1
130	:100810009FBF0F90DF91CF911F910F9108951F926D
131	:100820000F920FB60F9211242F933F938F939F93A4
132	:10083000AF93BF938091250190912601A09127014C
133	:10084000B09128013091240123E0230F2D3720F4AB
134	:100850000196A11DB11D05C026E8230F0296A11D1A
135	:10086000B11D209324018093250190932601A0932C
136	:100870002701B09328018091290190912A01A0912C
137	:100880002B01B0912C010196A11DB11D809329016E
138	:1008900090932A01A0932B01B0932C01BF91AF91AB
139	:1008A0009F918F913F912F910F900FBE0F901F90AE
140	:1008B00018953FB7F8948091290190912A01A09151
141	:1008C0002B01B0912C0126B5A89B05C02F3F19F034
142	:1008D0000196A11DB11D3FBF6627782F892F9A2F42
143	:1008E000620F711D811D911D42E0660F771F881FE9
144	:1008F000991F4A95D1F70895CF92DF92EF92FF9218
145	:10090000CF93DF936B017C010E945904EB01C1146A
146	:10091000D104E104F10479F00E9459046C1B7D0BB1
147	:10092000683E7340A0F381E0C81AD108E108F108DD
148	:10093000C851DC4FECCFDF91CF91FF90EF90DF906B
149	:10094000CF900895789484B5826084BD84B5816029
150	:1009500084BD85B5826085BD85B5816085BDEEE6C7
151	:10096000F0E0808181608083E1E8F0E010828081A6
152	:1009700082608083808181608083E0E8F0E0808114
153	:1009800081608083E1EBF0E0808184608083E0EB34
154	:10099000F0E0808181608083EAЕ7F0E0808184601C
155	:1009A0008083808182608083808181608083808178
156	:1009B000806880831092C10008958E3008F08E50B8
157	:1009C00087702091060190E4299F90011124822BC9
158	:1009D00080937C0080917A00806480937A0080917B
159	:1009E0007A0086FDFCCF809178002091790090E01C
160	:1009F000922B08951F93CF93DF93182FEB0161E0A3
161	:100A00000E94A003209711F460E004C0CF3FD105FD
162	:100A100039F461E0812FDF91CF911F910C94D903BC
163	:100A2000E12FF0E0E450FF4FE491E330F9F048F4B7
164	:100A3000E130B9F0E230A1F584B5806284BDC8BD73
165	:100A400034C0E630E9F0E73019F1E43049F580913F

166	:100A50008000806280938000D0938B00C0938A00D6
167	:100A600024C084B5806884BDC7BD1FC0809180004C
168	:100A7000806880938000D0938900C093880015C05F
169	:100A80008091B00080688093B000C093B3000DC027
170	:100A90008091B00080628093B000C093B40005C024
171	:100AA000C038D1050CF0B5CFAFCFDF91CF911F91FA
172	:100AB00008950E94A2040E94D100C8E4D7E00E94D9
173	:100AC00073032097E1F30E944807F9CFCF92DF929A
174	:100AD000EF92FF920F931F93CF93DF936C017A01F4
175	:100AE000EB01E60EF71E00E010E0CE15DF0561F029
176	:100AF0006991D601ED91FC910190F081E02DC60144
177	:100B00000995080F191FF1CFC801DF91CF911F91EF
178	:100B10000F91FF90EF90DF90CF90089561157105D0
179	:100B200081F0DB010D900020E9F7AD014150510942
180	:100B3000461B570BDC01ED91FC910280F381E02D07
181	:100B4000099480E090E008950C948E05DC01ED910D
182	:100B5000FC910190F081E02D09940F931F93CF93A6
183	:100B6000DF93EC016DE00E94A6058C016AE0CE01E6
184	:100B70000E94A605800F911FDF91CF911F910F91C9
185	:100B800008950F931F93CF93DF93EC010E948E057E
186	:100B90008C01CE010E94AD05800F911FDF91CF9196
187	:100BA0001F910F9108958F929F92AF92BF92CF9213
188	:100BB000DF92EF92FF920F931F93CF93DF93CDB706
189	:100BC000DEB7A1970FB6F894DEBF0FBECDBF7C0194
190	:100BD000C42EE52FCB01D22E19A221E02D1510F045
191	:100BE0002AE0D22E8E010F5D1F4F8D2C912CA12C4F
192	:100BF000B12C6C2D7E2FA50194010E94C1088C2D73
193	:100C0000D29E80191124015011098A3014F4805D9C
194	:100C100001C0895CF801808321153105410551052A
195	:100C200021F0C22EE32FCA01E4CFB801C7010E9410
196	:100C30008E05A1960FB6F894DEBF0FBECDBFDF9133
197	:100C4000CF911F910F91FF90EF90DF90CF90BF90C9
198	:100C5000AF909F908F900895CF92DF92EF92FF9286
199	:100C60000F931F93CF93DF93EC016A017B01211552
200	:100C7000310579F4E881F9810190F081E02D642F4C
201	:100C8000DF91CF911F910F91FF90EF90DF90CF9068
202	:100C900009942A303105E9F477FF1AC06DE20E9409
203	:100CA000A6058C0144275527BA014C195D096E0928
204	:100CB0007F092AE0CE010E94D305800F911FDF91AA
205	:100CC000CF911F910F91FF90EF90DF90CF900895FB
206	:100CD0002AE0B701A601CE01DF91CF911F910F91BC
207	:100CE000FF90EF90DF90CF900C94D305FC0124850A
208	:100CF0003585F901E05CFF4F80819181F901EE5B60
209	:100D0000FF4F20813181821B930B8F7399270895A8
210	:100D1000FC0184859585FC01E05CFF4F4081518199
211	:100D2000FC01EE5BFF4F208131814217530741F0F8
212	:100D30000190F081E02DE80FF91F808190E0089587
213	:100D40008FEF9FEF0895FC0184859585FC01E05CA1
214	:100D5000FF4F40815181FC01EE5BFF4F20813181CB
215	:100D60004217530779F0A081B181A80FB91F8C9168
216	:100D7000208131812F5F3F4F2F73332731832083B1
217	:100D800090E008958FEF9FEF0895FC0181A18823E3
218	:100D900029F0A489B5898C9186FFFBCF11A2089513
219	:100DA000CF93DF93FC0126853785D901A05CBF4F27
220	:100DB0008D919C91119701968F739927E901CE5BD4

221	:100DC000DF4F4881598184179507D9F30D90BC9165
222	:100DD000A02DA20FB31F6C93A685B785A05CBF4F53
223	:100DE00011969C938E93A689B7892C9181E090E00F
224	:100DF000078C02C0880F991F0A94E2F7282B2C93C6
225	:100E000081E081A30488F589E02D8081806480835E
226	:100E100081E090E0DF91CF91089508951F920F92A5
227	:100E20000FB60F9211242F933F934F938F939F935D
228	:100E3000EF93FF938091C00082FD1CC04091C600DB
229	:100E40008091D3019091D40101968F7399272091BD
230	:100E5000D5013091D6018217930771F0E091D3014B
231	:100E6000F091D401ED56FE4F40839093D4018093CE
232	:100E7000D30102C08091C600FF91EF919F918F91A5
233	:100E80004F913F912F910F900FBE0F901F9018958B
234	:100E90008DE291E00E947606892B11F00C940D07EB
235	:100EA00008951F920F920FB60F9211242F933F9324
236	:100EB0008F939F93EF93FF9320918F013091900137
237	:100EC00080919101909192012817390731F4809116
238	:100ED000C1008F7D8093C10014C0E0919101F09119
239	:100EE0009201E15BFE4F20818091910190919201EE
240	:100EF00001968F739927909392018093910120938B
241	:100F0000C600FF91EF919F918F913F912F910F908C
242	:100F10000FBE0F901F901895CF92DF92EF92FF9225
243	:100F20000CF93DF93EC016A017B01411581EE5807F5
244	:100F300061057105F9F0EC89FD8981E090E008A078
245	:100F400002C0880F991F0A94E2F7808360E079E07D
246	:100F50008DE390E0A70196010E94C108215031095C
247	:100F6000410951095695479537952795211580E1F7
248	:100F7000380798F0EC89FD89108260E874E88EE10A
249	:100F800090E0A70196010E94C10821503109410952
250	:100F900051095695479537952795E889F989308302
251	:100FA000EA89FB89208319A2EE89FF89408181E0CB
252	:100FB00090E09C010C8C02C0220F331F0A94E2F7D0
253	:100FC000422B4083EE89FF8940819C010D8C02C039
254	:100FD000220F331F0A94E2F7422B4083EE89FF89E8
255	:100FE00040819C010E8C02C0220F331F0A94E2F74D
256	:100FF000422B4083EE89FF8920810F8C02C0880F2D
257	:10100000991F0A94E2F7809582238083DF91CF9124
258	:10101000FF90EF90DF90CF900895109230011092E2
259	:101020002F0188EE93E0A0E0B0E08093310190932F
260	:101030003201A0933301B09334018DE091E090939D
261	:101040002E0180932D0183E991E090933A018093E2
262	:1010500039018FE491E090933C0180933B0185EC52
263	:1010600090E090933E0180933D0184EC90E090935A
264	:10107000400180933F0180EC90E090934201809387
265	:10108000410181EC90E0909344018093430182EC14
266	:1010900090E0909346018093450186EC90E0909318
267	:1010A00048018093470184E08093490183E0809365
268	:1010B0004A0187E080934B0185E080934C0181E0F9
269	:1010C00080934D0108954F925F926F927F92AF92FD
270	:1010D000BF92CF92DF92EF92FF920F931F93CF9325
271	:1010E000DF93CDB7DEB729013A01621B730B840B86
272	:1010F000950B29893A894B895C892A193B094C0946
273	:101100005D09298B3A8B4B8B5C8B9B01AC01698908
274	:101110007A898B899C890E94B108E418F508060930
275	:101120001709A80197010E94E3082A0D3B1D4C1DD9

276 :101130005D1DCA01B901DF91CF911F910F91FF9001  
277 :10114000EF90DF90CF90BF90AF907F906F905F90C7  
278 :101150004F9008950E940809B7FF0895821B930BD2  
279 :101160000895DB018F939F930E941709BF91AF9160  
280 :10117000A29F800D911DA39F900DB29F900D1124F1  
281 :101180000895A1E21A2EAA1BBB1BFD010DC0AA1FC8  
282 :10119000BB1FEE1FFF1FA217B307E407F50720F0E0  
283 :1011A000A21BB30BE40BF50B661F771F881F991F5B  
284 :1011B0001A9469F760957095809590959B01AC01A4  
285 :1011C000BD01CF010895052E97FB1EF400940E94E7  
286 :1011D000FA0857FD07D00E94C10807FC03D04EF45F  
287 :1011E0000C94FA0850954095309521953F4F4F4FFC  
288 :1011F0005F4F089590958095709561957F4F8F4FC3  
289 :101200009F4F0895EE0FFF1F0590F491E02D099474  
290 :10121000A29FB001B39FC001A39F700D811D112437  
291 :10122000911DB29F700D811D1124911D08950E9482  
292 :101230000809A59F900DB49F900DA49F800D911D4E  
293 :0812400011240895F894FFCF7A  
294 :1012480001000000030001200000000000D0066635  
295 :0A125800057606A3068806C5060009  
296 :00000001FF  
297