



# **INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA**

TESIS

**“DISEÑO DE UNA PLATAFORMA  
COMPUTACIONAL PARA LA GESTIÓN  
ENERGÉTICA DE AIRES  
ACONDICIONADOS,  
VIDEOPROYECTORES”.**

PARA OBTENER EL TÍTULO DE INGENIERO EN  
**SISTEMAS COMPUTACIONALES**

PRESENTA

**MIGUEL ANGEL TORRES MEDEL**

ASESOR

**MSC. JORGE CRUZ SALAZAR**

MISANTLA, VER.

FEBRERO, 2021







INSTITUTO TECNOLÓGICO SUPERIOR DE MISANTLA  
DIVISIÓN DE ESTUDIOS PROFESIONALES  
AUTORIZACIÓN DE IMPRESIÓN DE TRABAJO DE TITULACIÓN

FECHA: 28 de Abril de 2021.

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN  
DE TESIS PROFESIONAL.

A QUIEN CORRESPONDA:

Por medio de la presente hago constar que el (la) C:

MIGUEL ANGEL TORRES MEDEL

pasante de la carrera de INGENIERÍA EN SISTEMAS COMPUTACIONALES con No. de Control 162T0372 ha cumplido satisfactoriamente con lo estipulado por el **Manual de Procedimientos para la Obtención del Título Profesional de Licenciatura** bajo la opción Titulación Integral (Tesis Profesional)

Por tal motivo se **Autoriza** la impresión del Tema titulado:

**“DISEÑO DE UNA PLATAFORMA COMPUTACIONAL PARA LA GESTIÓN ENERGÉTICA DE AIRES ACONDICIONADOS, VIDEOPROYECTORES”**

Dándose un plazo no mayor de un mes de la expedición de la presente a la solicitud del Acto de Recepción para la obtención del Título Profesional.

ATENTAMENTE

ING. GERBACIO TLAXALO ESPINOZA  
DIVISIÓN DE ESTUDIOS PROFESIONALES



Archivo.

## **Agradecimientos**

Agradecido con la vida y el destino que me trajo aquí, que me está dando la oportunidad de llegar a los últimos escalones de mi formación profesional, dándome fuerza para seguir adelante y llegar a la meta de finalizar mi tesis y mi carrera profesional.

## **Dedicatoria**

A mis padres Eric Torres Sosa, Lorena Medel Casco y familiares, a ellos que gracias a su esfuerzo de cada día hoy estoy aquí, que siempre me han brindado su apoyo y nunca me han dejado solo, gracias por su apoyo incondicional, ya que lo único que quieren es que Yo llegue muy lejos, gracias por los brindado hasta el día de hoy, y por ellos puedo finalizar este proyecto, los quiero.

A mi asesor M.S.C. Jorge Cruz Salazar por su tiempo, por su dedicación y sobre todo por su paciencia, brindando siempre buenos consejos para poder mejorar y gracias a eso poder terminar los últimos pasos de mi carrera profesional.

## **Resumen**

En las aulas del Instituto Tecnológico Superior de Misantla hay una mala aplicación en cuanto a la eficiencia energética, ya que dentro de ellas aparatos electrónicos como los son aires acondicionados y videoproyectores, en diversas ocasiones se ha detectado que tanto alumnos y personal de la institución se retiran de las salas sin haber apagado dichos aparatos, esto lo hacen por en ocasiones los controles no están a su disposición no sirven o simplemente por decidía, esto da como resultado un mal consumo de energía eléctrica que representa un gasto muy elevado que es innecesario.

El presente proyecto, “Diseño de una plataforma computacional para la gestión energética de aires acondicionados, videoproyectores”, pretende desarrollar un sistema el cual permita manipular los aparatos antes mencionados y poder ver el estado en el que se encuentran, ya sea encendidos o apagados y tener un control más precisos sobre estos.

Para el desarrollo de dicha plataforma se pretende hacer uso de una tarjeta ESP32 para crear un sistema embebido que a través de señales infrarrojas podrá controlar los aparatos con ayuda de una página Web y también hacer uso de un Smartphone que tenga comunicación infrarroja para así poder de igual forma controlar aires acondicionados y videoproyectores con ayuda de una aplicación.

## CONTENIDO

<b>CAPITULO I</b> .....	<b>1</b>
<b>GENERALIDADES Y DESCRIPCIÓN DE LA EMPRESA</b> .....	<b>1</b>
<b>1.1 Introducción</b> .....	<b>2</b>
<b>1.2 Problemática</b> .....	<b>3</b>
<b>1.3 Hipótesis</b> .....	<b>4</b>
<b>1.4 Objetivos</b> .....	<b>5</b>
1.4.1 Objetivo general .....	5
1.4.2 Objetivos específicos .....	5
<b>1.5 Justificación</b> .....	<b>6</b>
<b>1.6 Propuesta de solución</b> .....	<b>6</b>
<b>1.7 Metodología</b> .....	<b>6</b>
<b>1.8 Alcances y limitaciones</b> .....	<b>8</b>
1.8.1 Alcances .....	8
1.8.2 Limitaciones .....	8
<b>1.9 Antecedentes de la institución</b> .....	<b>9</b>
1.10 Misión .....	11
1.11 Visión .....	11
1.12 Ubicación geográfica de la institución .....	11
1.13 Macro localización .....	11
1.14 Micro localización .....	12
1.15 Áreas de prácticas .....	13
1.16 Taller industrial .....	13
1.17 Centro de cómputo .....	13
1.18 Laboratorio de electrónica .....	14
1.19 Laboratorio de electromecánica .....	14
1.20 Laboratorio de bioquímica .....	14
1.21 Laboratorio asignado para la investigación .....	15
<b>CAPITULO II</b> .....	<b>16</b>

<b>MARCO TEÓRICO Y ESTADO DEL ARTE .....</b>	<b>16</b>
<b>Marco Teórico .....</b>	<b>17</b>
<b>2.1 ¿Qué es el infrarrojo?.....</b>	<b>17</b>
<b>2.2 Características de los sensores infrarrojos.....</b>	<b>17</b>
<b>2.3 LED infrarrojo.....</b>	<b>18</b>
<b>2.4 Aplicaciones.....</b>	<b>18</b>
<b>2.5 ¿Qué es un protocolo?.....</b>	<b>18</b>
<b>2.6 Protocolos infrarrojos. ....</b>	<b>19</b>
2.6.1 Protocolo RC5 .....	19
2.6.2 Protocolo ITT IR .....	19
2.6.3 Protocolo IrDA .....	19
2.6.4 Protocolo NEC y SIRC .....	20
<b>2.7 Arduino (IDE).....</b>	<b>20</b>
<b>2.8 Android Studio .....</b>	<b>20</b>
<b>2.9 ¿Qué es Firebase? .....</b>	<b>21</b>
<b>2.10 Tarjeta ESP32.....</b>	<b>21</b>
2.10.1 Características de ESP32 .....	21
<b>2.11 Arduino Uno .....</b>	<b>22</b>
<b>2.12 Cables Dupont.....</b>	<b>22</b>
<b>2.13 Dispositivos automatizables (infrarrojo).....</b>	<b>22</b>
2.13.1 Aire acondicionado Split.....	22
2.13.2 Proyectores .....	22
<b>2.14 Estado del Arte.....</b>	<b>23</b>
<b>CAPITULO III .....</b>	<b>28</b>
<b>DESARROLLO.....</b>	<b>28</b>
<b>3.1 Requisitos de software y equipo .....</b>	<b>29</b>
3.1.1 Laptop.....	29
3.1.2 Smartphone .....	29



3.1.3 Tarjetas y Componentes .....	30
3.1.4 Investigar y seleccionar el entorno de desarrollo.....	30
3.1.5 Descargas .....	30
3.1.6 Lenguaje de programación.....	40
3.1.7 Instalación de la tarjeta ESP32 en el IDE de Arduino.....	41
3.1.8 Librerías de Arduino .....	43
<b>3.2 Diseño.....</b>	<b>45</b>
3.2.1 Captura de señales infrarrojas .....	45
<b>3.3 Implementación.....</b>	<b>46</b>
3.3.1 Creación de página web en Arduino .....	46
3.3.2 Creación de proyecto en AndroidStudio .....	47
3.3.3 Permisos de Internet .....	48
3.3.4 Dependencias .....	48
3.3.5 Conexión con Firebase .....	49
3.3.6 Habilitar inicio de sesión en Firebase .....	52
3.3.7 Función de login .....	52
3.3.8 Función onCreate .....	53
<b>3.4 Verificación .....</b>	<b>54</b>
3.4.1 Conexión de hardware y software.....	54
3.4.2 Envío de pulsos con LED infrarrojo.....	55
3.4.3 Inicio de sesión de la aplicación.....	56
3.4.4 Autenticación con Firebase.....	57
<b>CAPITULO IV.....</b>	<b>62</b>
<b>RESULTADOS Y CONCLUSIONES .....</b>	<b>62</b>
4.1 Resultados.....	63
4.2 Conclusiones.....	64
4.3 Referencias .....	65
4.4 Anexos.....	66

## Índice de figuras

FIGURA 1 ITSM.....	9
FIGURA 2 LOCALIZACIÓN .....	12
FIGURA 3 UBICACIÓN .....	12
FIGURA 4 CENTRO DE COMPUTO.....	15
FIGURA 5 DESCARGA ANDROID.....	31
FIGURA 6 TIEMPO DE DESCARGA .....	31
FIGURA 7 INICIO DE INSTALACIÓN .....	31
FIGURA 8 PROCESO DE INSTALACIÓN 1 .....	32
FIGURA 9 PROCESO DE INSTALACIÓN 2 .....	32
FIGURA 10 PROCESO DE INSTALACIÓN 3 .....	33
FIGURA 11 PROCESO DE INSTALACIÓN 4 .....	34
FIGURA 12 INICIO DE ANDROID.....	34
FIGURA 13 DESCARGA DE ARDUINO .....	35
FIGURA 14 TIEMPO DE DESCARGA ARDUINO .....	35
FIGURA 15 PROCESO DE INSTALACIÓN ARDUINO.....	36
FIGURA 16 PROCESO DE INSTALACIÓN ARDUINO 2.....	36
FIGURA 17 PROCESO DE INSTALACIÓN ARDUINO 3.....	37
FIGURA 18 PROCESO DE INSTALACIÓN ARDUINO 4.....	37
FIGURA 19 PROCESO DE INSTALACIÓN ARDUINO 5.....	38
FIGURA 20 INICIO DE ARDUINO.....	39
FIGURA 21 PREFERENCIAS DE ARDUINO .....	41
FIGURA 22 MENÚ PREFERENCIAS .....	41
FIGURA 23 OPCIÓN DE PLACAS .....	42
FIGURA 24 LIBRERÍA ESP32.....	42
FIGURA 25 TARJETAS DISPONIBLES DE PLATAFORMA ESP32.....	43
FIGURA 26 OPCIÓN DE ADMINISTRADOR DE BIBLIOTECAS .....	43
FIGURA 27 BIBLIOTECA WIFI .....	44
FIGURA 28 BIBLIOTECA IRREMOTE.....	44
FIGURA 29 CAPTURA DE CÓDIGOS RAW .....	45
FIGURA 30 TARJETA ARDUINO UNO.....	45
FIGURA 31 CÓDIGOS RAW.....	46
FIGURA 32 PÁGINA WEB EN ARDUINO .....	46
FIGURA 33 CREACIÓN DE PROYECTO EN ANDROID .....	47
FIGURA 34 PERMISOS DE INTERNET.....	48
FIGURA 35 DEPENDENCIAS.....	48
FIGURA 36 CONEXIÓN CON FIREBASE .....	49
FIGURA 37 AUTENTICACIÓN.....	50
FIGURA 38 DEPENDENCIAS DE FIREBASE .....	51
FIGURA 39 HABILITAR SERVICIO DE GOOGLE.....	52
FIGURA 40 FUNCIÓN LOGIN .....	52
FIGURA 41 FUNCIÓN ONCREATE .....	53
ILUSTRACIÓN 42 ESP32 .....	54
ILUSTRACIÓN 43 ARDUINO UNO.....	54
ILUSTRACIÓN 44 PULSOS.....	55
ILUSTRACIÓN 45 CLIMA.....	55
FIGURA 46 INICIO DE SESIÓN DE APLICACIÓN .....	56

FIGURA 47 LISTA DE USUARIOS .....	57
FIGURA 48 MENÚ DE SALAS .....	58
FIGURA 49 CONTROL DE APARATOS .....	59
FIGURA 50 APLICACIÓN WEB .....	59
ILUSTRACIÓN 51 FIREBASE ON .....	60
ILUSTRACIÓN 52 FIREBASE OFF .....	60
ILUSTRACIÓN 53 GRAFICO .....	61

## **INDICE DE TABLAS**

TABLA 1 LAPTOP .....	29
TABLA 2 SMARTPHONE .....	29
TABLA 3 TARJETAS Y COMPONENTES .....	30
TABLA 4 ENTORNOS DE DESARROLLO .....	30

# **CAPITULO I**

## **GENERALIDADES Y DESCRIPCIÓN DE LA EMPRESA**

## 1.1 Introducción

Actualmente en el transcurso de los años el hombre ha tenido la experiencia con la mayoría de los cambios climatológicos que nos brinda la naturaleza, el hombre ha tomado en cuenta también la evolución de los acontecimientos antiguos, así como hoy en día el cambio es muy radical debido a la tecnología estas nos brindan mayor accesibilidad a la mayoría de nuestras necesidades.

Como han pasado los años las necesidades han estado cambiando constantemente. lo que antes era buscar un lugar con las mejores condiciones para poder llevar una vida conforme al gusto de cada persona, ahora es muy común encontrar equipos que nos mantengan nuestros lugares en las mejores condiciones requeridas por el hombre, estos equipos de enfriamiento son de mayor uso en lugares calurosos.

Las personas de hoy en día suelen ser un poco distraídas y descuidadas, debido a que cuando realizan una compra de algún equipo electrónico nuevo, ya que lo de hoy es la tecnología, existe gran variedad que cuenta con un dispositivo de control o mando remoto, pero no siempre se toman la responsabilidad de tenerle ciertos cuidados al dispositivo remoto no obstante que suelen pasar ocasiones en las cuales el dispositivo se rompe, sufre alguna desconfiguración, o se le acaban las pilas.

Los equipos de control remoto siempre llegan a tener un desperfecto con el paso del tiempo por ello se llevará a cabo la implementación del mayor uso de la tecnología para realizar un dispositivo que nos brinde el apoyo de poder monitorear cualquier dispositivo que requiera de un mando a distancia controlado por señales infrarrojas, así como también tener el control de dispositivos electrónicos a través de una tarjeta para tener el monitoreo de todos ellos en un solo mando.

## 1.2 Problemática

En las aulas de Instituto Tecnológico Superior de Misantla existe una gran área de oportunidad en referencia a la eficiencia energética ya que normalmente cuentan con aire acondicionado de tres toneladas de enfriamiento, un video proyector y un sistema de iluminación con 6 lámparas, que son usadas a discreción de los usuarios, se ha detectado en diversas ocasiones que los usuarios se retiran dejando alguno o todos los elementos activados ya sea ser por decidía o porque no cuentan con algún control o medio para desactivarlos o activarlos y en algunas ocasiones estos controles presentan fallas por lo que no toda la responsabilidad es del usuario, como resultado de esto la energía eléctrica consumida representa un gasto económico innecesario así como una mayor huella de carbono para el instituto.

Para dar un panorama de la problemática, podemos decir que un clima de tres toneladas equivale a un consumo en watts de 10,550.6; el proyector consume en promedio 450 watts, ya que no son de tecnología led y seis lámparas promedio de 25 watts multiplicado por seis equivale a 150 watts, si sumamos el consumo de los tres elementos considerados en cada hora de tiene  $10550.6 + 450 + 150 = 11025.6$  watts, consultando los costos de CFE se puede estimar un gasto promedio de \$11.03 pesos por cada hora del salón y en un uso promedio de 8 horas es igual a 88.24 pesos por cada día en cada salón de acuerdo a la tarifa 1 de la comisión federal de electricidad.

### **1.3 Hipótesis**

Es posible desarrollar una aplicación móvil como herramienta para el fácil uso correspondiente de los dispositivos que se encuentran en las aulas del centro de cómputo del Instituto Tecnológico Superior de Misantla, como los son los aires acondicionados, videoproyectores, luces y pantallas para videoproyectores, para poderle dar un mejor uso en implementación de nuevas tecnologías.

## 1.4 Objetivos

### 1.4.1 Objetivo general

Realizar una plataforma que sea accesible desde una aplicación móvil o un navegador web, que permita el control de videoproyectores y aires acondicionados y su gestión de energética.

### 1.4.2 Objetivos específicos

- Realizar la búsqueda de información de diversas fuentes confiables, analizar en contenido publicado para clasificar la información que sea más valiosa o menos relevante para el proyecto.
- Seleccionar la tarjeta de desarrollo con la cual se estará trabajando, analizar los pros y contra que tiene en comparación con otras existentes en el mercado, elección del lenguaje de programación que mejor se adapte a las necesidades del proyecto y, por último, la compra de componentes.
- Investigar los diversos protocolos de infrarrojos que cada uno de los componentes electrónicos usan.
- Diseñar el hardware que se va a necesitar y aplicar ingeniería inversa para la adquisición de información.
- Implementar los diversos protocolos de comunicación infrarroja y de wifi de los aparatos electrónicos.
- Seleccionar un IDE para el desarrollo de la aplicación móvil y web, la primera contendrá diversas interfaces para el manejo de dispositivos, así como la programación necesaria para que operen de manera adecuada y la segunda tendrá una interfaz donde se podrá controlar los aparatos electrónicos.
- Programar los componentes adquiridos y tarjeta de desarrollo para establecer una comunicación estable y confiable.
- Realizar pruebas para revisar el funcionamiento de la aplicación y corregir todos los posibles errores de funcionamiento.



## 1.5 Justificación

En las instalaciones del centro de cómputo del ITSM se da un control sobre el manejo de los aparatos electrónicos de las salas, como los son los aires acondicionados o videoproyectores, pero en ocasiones tanto alumnos como personal de la institución los dejan al descuido. Esta aplicación servirá como un medio facilitador de manejo de dichos aparatos, permitiendo una manipulación más fácil de ellos.

Esta aplicación tiene la finalidad de ser una herramienta donde personal de la institución o algún alumno designado pueda observar el estado actual de los aparatos, para así saber si están siendo utilizados de una manera adecuada y así no hacer un mal uso del consumo eléctrico y ayudar a que los aparatos tengan un lapso de vida más largo.

## 1.6 Propuesta de solución

Para resolver el problema del mal uso que se les dan a los aparatos electrónicos de las salas del centro de cómputo del ITSM, se desarrollara una aplicación móvil y web donde se podrán manipular y ver el estado de estos aparatos con ayuda de una tarjeta ESP32 e infrarrojos, esto ayudara a observar si los aparatos están siendo utilizados de una manera adecuada y así prolongar más su tiempo de vida.

## 1.7 Metodología

Modelo en Cascada, también llamado Lineal secuencial, es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

### Requisitos del software y equipo

En esta fase se realiza una búsqueda de información sobre los diversos protocolos que los aparatos electrónicos usan; se hace la búsqueda para elegir la tarjeta de desarrollo que sea mejor para el trabajo que se va a realizar, al igual que la búsqueda de componentes que se van a ocupar; por otra parte, se debe de encontrar el lenguaje de programación que más se preste con lo que se va a desarrollar

## **Diseño**

En esta etapa se hace el diseño del hardware que se a ocupar y una vez diseñado aplicar ingeniería inversa para la adquisición de información, que en este caso serán los códigos raw que emiten los controles de los aparatos mediante el infrarrojo.

## **Implementación**

En esta fase con ayuda de los lenguajes de programación que se van a usar se va a hacer la implementación de los protocolos de comunicación de infrarroja y de wifi, esto se va a poder realizar con el IDE que se elija, con esto todos los componentes adquiridos y la tarjeta de desarrollo deben tener una buena comunicación.

## **Verificación**

La fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.

Se hizo la verificación de que cada botón que se creó en la aplicación funcionara correctamente, esto probando cada uno y observado su comportamiento.

## 1.8 Alcances y limitaciones

### 1.8.1 Alcances

- El sistema es para Android y web, pero es factible programarse en otros entornos.
- La aplicación aun que es desarrollada para dispositivos con sistema operativo Android, los usuarios que tengan IOS también pueden tener acceso al control de los aparatos por medio de la tarjeta ESP32 a través de la web.
- La aplicación controla aires acondicionados y videoproyectores del centro de cómputo del ITSM.
- Desde el servidor FireBase donde se observa el estado del aparato de puede apagar o encender de igual manera que en la aplicación.

### 1.8.2 Limitaciones

- Solo disponible para las versiones superiores a Android 6.0
- La aplicación en Android solo funciona con dispositivos que tengan infrarrojo.
- Solo se puede iniciar sesión con una cuenta de Google.

## 1.9 Antecedentes de la institución



*Figura 1 ITSM*

El Instituto Tecnológico Superior de Misantla (ITSM), fue creado mediante el decreto el 26 de noviembre del 1994, con el propósito de contribuir a impulsar y consolidar el Programa de Desarrollo de la Educación Superior Tecnológica en el estado; y como un organismo público descentralizado del Gobierno del Estado, con personalidad jurídica y patrimonio propios, ubicado en Misantla Veracruz, con los siguientes objetivos:

- ✓ Formar profesionales e investigadores aptos para la aplicación y generación de conocimientos científicos y tecnológicos.
- ✓ Realizar investigación científica y tecnológica que permita el avance del conocimiento, el desarrollo de la enseñanza tecnológica y el mejor aprovechamiento social de los recursos naturales y materiales; y que se traduzca en aportaciones concretas para el mejoramiento y eficacia de la producción industrial y de servicios, y a la elevación de la calidad de vida de la comunidad.
- ✓ Colaborar con el sector público, privado y social en la consolidación del desarrollo tecnológico y social de la comunidad.
- ✓ Promover la cultura regional y nacional.

En su inicio el Instituto Tecnológico Superior De Misantla contaba con 11 maestros y 17 empleados administrativos; en ese entonces, el tecnológico no tenía instalaciones propias, las clases iniciaron en la Casa de la Cultura y después en la Escuela de Bachilleres “Alfonso Reyes” de Misantla.<sup>16</sup>

La matrícula inicial fue de 166 alumnos en dos carreras, ingeniería industrial y licenciatura en informática. Hoy a 20 de años de su fundación se cuenta con un edificio totalmente equipado, (ver Figura No.1) con 9 carreras profesionales y 2 maestrías, una matrícula 2243 alumnos y un staff de poco más de 200 empleados.

Las carreras con las que cuenta son:

1. Ingeniería industrial
2. Ingeniería en Sistemas computacionales
3. Ingeniería bioquímica
4. Ingeniería electromecánica
5. Ingeniería civil
6. Ingeniería en tecnologías de la información y comunicaciones
7. Ingeniería ambiental
8. Ingeniería en gestión empresarial
9. Ingeniería Petrolera

Las carreras del tecnológico tienen reconocimiento nacional de calidad, se tiene un programa de movilidad internacional donde participan más de 50 alumnos, los cuales cuentan con visas para viajar y estudiar en los Estados Unidos y en otros países.

Las maestrías son:

- Maestría en Ingeniería industrial
- Maestría en Sistemas Computacionales

Las maestrías cuentan con reconocimiento nacional de posgrados de calidad, por esa razón, los alumnos de posgrado reciben una beca mensual mayor a nueve mil pesos; se tiene 100 alumnos y es el primer Tecnológico que imparte programas de maestría fuera del estado.

### **1.10 Misión**

Ofrecer un servicio de Educación Superior Tecnológica, pertinente y de calidad, para la formación de profesionistas competitivos; que, cimentados en un desarrollo integral y armónico, les permita responder en un contexto de globalización, a los retos que demanda el País. Así mismo, compartir con la población en general los beneficios del conocimiento, la cultura científica y tecnológica; en particular, proporcionar servicios directos a los diversos sectores, mismos que contribuyan al bienestar que demanda la sociedad.

### **1.11 Visión**

Ser en nuestra Región y zona de influencia, un Instituto Tecnológico de prestigio en la formación de profesionistas competitivos acordes con la demanda que exige el desarrollo del país; sustentado en la calidad de los servicios educativos infraestructura, ambiente armónico, difusión de la cultura y vinculación con los sectores social, público y privado.

### **1.12 Ubicación geográfica de la institución**

El instituto tecnológico superior de Misantla, la casa de estudios más grande de la región ubicada en la ciudad de Misantla estado de Veracruz, sus coordenadas son  $19^{\circ}55'51.86''N$   $96^{\circ}51'6.09''O$  y está situado a una altura de 400 m. sobre nivel del mar.

### **1.13 Macro localización**

La ciudad de Misantla limita al norte con el municipio de Nautla, al sur con el municipio de Chiconquiaco, al este con los municipios de Yecuatla y Colipa, al oeste con Altotonga y Atzalan, al suroeste con Tenochtitlán y al noroeste con Martínez de la Torre. (Ver Figura No. 2).

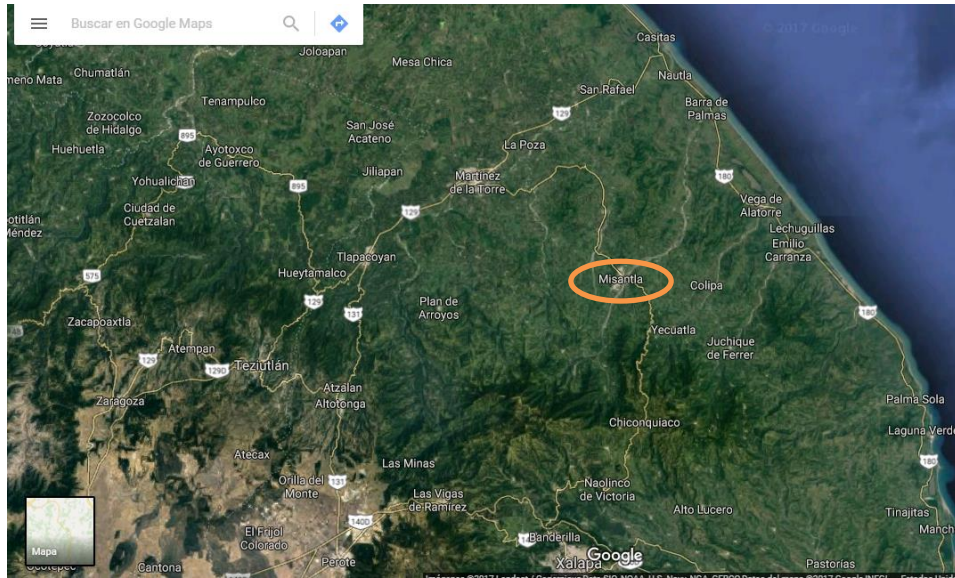


Figura 2 Localización

### 1.14 Micro localización

La casa de estudios del Instituto Tecnológico Superior de Misantla se encuentra entre las calles Gabriela Mistral y Josefa Ortiz de Domínguez, a Km 1.8 carretera a Loma del Cojolite. (Ver Figura No. 3).

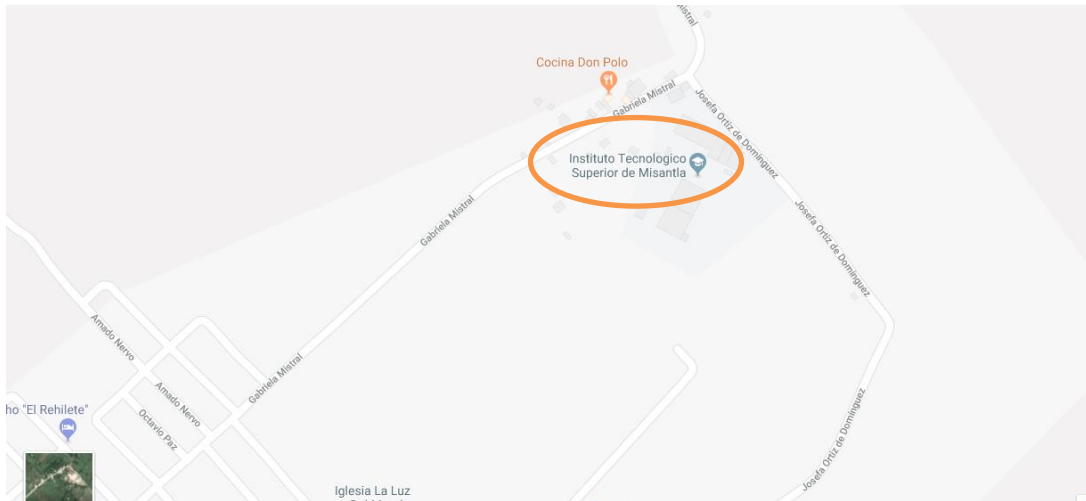


Figura 3 Ubicación

### **1.15 Áreas de prácticas**

El instituto tecnológico superior de Misantla se encuentra organizado en cinco áreas de prácticas: taller industrial, centro de cómputo, taller de electromecánica, laboratorio de electrónica y laboratorio de bioquímica.

### **1.16 Taller industrial**

Su objetivo es controlar, regular y fijar las condiciones de la realización de todas las actividades efectuadas dentro del taller Industrial, para obtener un mayor rendimiento de los materiales, herramientas, equipos e instalaciones, contribuyendo a mantener un orden dentro de la instalación, logrando con esto un mejor nivel en el servicio.

La infraestructura del Taller Industrial del ITSM comprende:

- Salón.
- Cubículo.
- Sanitarios.
- Laboratorio de manufactura.
- Laboratorio de metrología.
- Laboratorio de ingeniería de métodos.
- Módulo de préstamos de equipo

### **1.17 Centro de cómputo**

Apoyar con criterios de calidad, eficiencia y seguridad, a las actividades donde se requiera el uso de equipo de cómputo en las áreas académicas, de investigación y administrativas de la institución, satisfaciendo las necesidades de información del Instituto, de manera veraz y oportuna, facilitando las herramientas necesarias para acceder al desarrollo tecnológico de la informática.

El laboratorio de cómputo presta a los usuarios los siguientes servicios:

- Internet.
- Registro de usuarios.
- Reservación de equipo de cómputo.



- Préstamo de material de consulta y aplicación.
- Servicio de mantenimiento preventivo y correctivo a equipos del instituto.
- Mantenimiento a sistemas de información.
- Asesoría a usuarios.
- Servicio de impresión.
- Apoyo a departamentos.
- Soporte técnico.

### **1.18 Laboratorio de electrónica**

El laboratorio tiene como objetivo proporcionar el servicio de préstamo y mantenimiento de equipo como son manuales, herramientas, componentes e instalación con que cuenta el laboratorio para obtener los mejores resultados individuales y de conjunto en el aspecto educativo. (Ver Figura No. 6).

### **1.19 Laboratorio de electromecánica**

El objetivo del laboratorio de electromecánica es regular el préstamo del material y equipo existente del ITSM con la finalidad de satisfacer las necesidades de información y prácticas de los usuarios que lo soliciten.

### **1.20 Laboratorio de bioquímica**

El objetivo del laboratorio de bioquímica es regular el préstamo del material y equipo existente en el laboratorio de bioquímica del ITSM con la finalidad de satisfacer las necesidades de información y prácticas de los usuarios que lo soliciten.

El servicio que ofrece es: facilitar el material, reactivos y equipo para realizar sus prácticas única y exclusivamente dentro del laboratorio de ing. Bioquímica y con la orientación de un docente y auxiliar.

### **1.21 Laboratorio asignado para la investigación**

La investigación se realizó en el centro de cómputo del Instituto Tecnológico Superior de Misantla, en dónde se presta servicio de material para que, profesores y alumnos del área, puedan realizar sus diversas prácticas en las asignaturas, (Ver Figura No.4). Este laboratorio está a cargo del MSC. Jorge Cruz Salazar.



*Figura 4 Centro de Computo*

# **CAPITULO II**

## **MARCO TEÓRICO Y ESTADO DEL ARTE**

## **Marco Teórico**

### **2.1 ¿Qué es el infrarrojo?**

El infrarrojo es un tipo de luz que el ojo humano no alcanza a percibir. La luz infrarroja brinda información como por ejemplo puede mostrar cuanto calor tiene algún objeto, o también se usa para permitir la comunicación entre nodos, esto se hace usando una serie de leds infrarrojos llamadas redes de infrarrojos. Se trata de emisores y receptores de las ondas infrarrojas que hay entre ambos dispositivos, tanto como el emisor y el receptor se tienen que “ver” uno al otro para poder realizar una comunicación adecuada. Esta interacción entre dispositivos infrarrojos se utiliza principalmente para realizar intercambio de datos entre dispositivos móviles, ya que el rango de velocidad y el tamaño de los datos que se envía y que se recibe es pequeño.

Existen tres tipos de comunicación infrarroja, uno de ellos es el modo de punto a punto, este consiste en que tanto como el emisor y el receptor deben de estar lo más cerca posible y estén alineados de una forma correcta. El otro tipo de comunicación infrarroja es el casi difuso que en este modo todas las estaciones están comunicadas por medio de superficies reflectantes y no es necesario tener una línea de visión entre dos estaciones, pero si deben estarlos con la superficie de reflexión. Y por último se encuentra el modo difuso que es más flexible, en lo que es localización y posición de cada estación, pero su flexibilidad está a costa excesivas emisiones ópticas.

### **2.2 Características de los sensores infrarrojos**

Un detector infrarrojo es un tipo de dispositivo optoelectrónico que tiene la capacidad de medir la radiación electromagnética infrarroja que emiten los cuerpos que se encuentran dentro de su campo de visión. Se trata de un tipo de radiación que emiten todos los cuerpos de forma independiente a que exista otro tipo de luz ambiental. De este modo, permite observar espacios y objetos sin necesidad de que exista luz visible o de otro tipo en el entorno.

## 2.3 LED infrarrojo

Un led infrarrojo son un tipo específico de diodo emisor de luz que produce luz en el espectro infrarrojo. La luz en el rango infrarrojo no es percibida por el ojo humano, pero puede ser detectada por una variedad de dispositivos electrónicos, esto convierte al LED como algo ideal para controles remotos.

La longitud de onda de la luz emitida por un LED infrarrojo es cerca de 1000 nm con un ancho de banda de unos 50 nm. Esto quiere decir que un LED con valor de 1000 nm puede producir luz entre 950 y 1050 nm.

Un led infrarrojo funciona convirtiendo la corriente eléctrica en luz infrarroja; mientras que los detectores infrarrojos hacen lo contrario al detectar luz infrarroja y la convierten en una corriente eléctrica. La corriente que genera un detector de infrarrojo es una señal que indica que existe ese tipo de luz.

## 2.4 Aplicaciones

- Uso doméstico
- Ciencias médicas y biológicas
- Cámaras de vigilancia y seguridad
- Seguridad aérea y territorial
- Automovilismo
- Periféricos de TI y productos de consumo

## 2.5 ¿Qué es un protocolo?

En las telecomunicaciones o en informática, un protocolo se refiere a un sistema de normas que regulan la comunicación entre dos o más sistemas que transmiten información a través de diversos medios físicos. Un protocolo dicho de otra manera son códigos de comunicación entre sistemas informáticos, estos están definidos con base en una sintaxis, una semántica y una sincronización, así como de métodos de recuperación de errores.

## **2.6 Protocolos infrarrojos.**

### **2.6.1 Protocolo RC5**

El protocolo RC5 es utilizado para la comunicación de datos. Este protocolo cuenta con 32 direcciones diferentes con 64 instrucciones para cada dirección, teniendo un total de 2048 instrucciones diferentes. Los dispositivos que usan este protocolo utilizan una dirección propia que permite controlar otros dispositivos con un solo mando a distancia, este código se compone de 14 bit.

### **2.6.2 Protocolo ITT IR**

El protocolo ITT IR no usa una frecuencia modulada para enviar ordenes, sino que es un protocolo más antiguo que envía mensaje con 14 pulsos. Este protocolo tiene mucha finalidad y tiene la ventaja de que consume muy poca energía. Lo malo de este protocolo es que puede generar ordenes que pueden ser falsas. El código que emite este protocolo tiene dirección de 4 bits y una longitud de comando de 6 bits.

### **2.6.3 Protocolo IrDA**

El protocolo IrDA es el máximo extendido, ya que está construido en los teléfonos móviles, ordenadores portátiles, impresoras y asistentes digitales personales. Este protocolo se conforma por otros tres protocolos que son obligatorios que son:

- IrPHY(físicas).
- IrLAP (Link Access).
- IrLMP (Enlace de Gestión).

IrDA cuenta con diversos enlaces infrarrojos para apoyar diferentes velocidades de transmisión de datos:

- Infrarrojo serie soporta 115.2 kbps.
- Medio infrarrojo soporta 0.576 Mbps.
- Fast Infrared soporta 4.0 Mbps.
- Infrarrojo más rápido soporta 14 Mbps.

### **2.6.4 Protocolo NEC y SIRC**

El protocolo NEC y SIRC fueron diseñados por Sony, el protocolo NEC utiliza una frecuencia de 38 KHz y también usa un pulso de codificación de distancia. Este protocolo utiliza 8 bits con 8 comandos. El protocolo SIRC posee tres versiones, una de ellas es de 12 bits, la otra versión es de 15 bits y la última versión es de 20 bits. SIRC se utiliza más en los dispositivos de entretenimiento, como TV, reproductores de DVD y equipo de música.

## **2.7 Arduino (IDE)**

Arduino es una plataforma abierta que hace fácil la programación de un microprocesador. Los microprocesadores usan sensores para escuchar el mundo físico y los actuadores para interactuar con el mundo físico.

Una de las principales características de Arduino (IDE) es que es portable y su respectivo lenguaje de programación tiene una sencillez para facilidad de uso. Arduino consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Además, incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware. Arduino es una tecnología que tiene una rápida curva de aprendizaje con básicos conocimientos de programación y electrónica, esto permite que se puedan desarrollar proyectos en el ámbito de las Smart Cities, el internet de las cosas, dispositivos wearables, salud, ocio, educación, robótica, y entre otras más.

## **2.8 Android Studio**

Este entorno de desarrollo integrado (IDE) está basado en la herramienta de IntelliJ IDEA, y cuenta con un potente editor de códigos y multitud de funciones que permiten una mayor productividad durante el desarrollo de la aplicación. También ofrece un flexible sistema de compilación, un emulador de gran rapidez y herramientas para identificar problemas de compatibilidad, rendimiento o usabilidad. Además, permite realizar cambios de código y aplicar nuevos recursos a la aplicación sin necesidad de reiniciarla, y es compatible con C++, NDK y Google Cloud Platform.2.12.

Este software fue diseñado para la creación de aplicaciones móviles con sistema operativo Android y, hoy en día, es una herramienta imprescindible para cualquier desarrollador. Sin embargo, su uso se extiende mucho más allá de la creación, ya que permite realizar muchas otras acciones de suma importancia para el posterior desarrollo y actualización de la aplicación.

Gracias a su sencilla e intuitiva interfaz se pueden crear y desarrollar aplicaciones de forma fácil, sin invertir demasiado esfuerzo, tanto si eres un experto o un aspirante a desarrollador.

## 2.9 ¿Qué es Firebase?

Es una plataforma digital que se utiliza para facilitar el desarrollo de aplicaciones web o móviles de una forma efectiva, rápida y sencilla, la cual es utilizada por sus diversas funciones como una técnica de Marketing Digital para aumentar la base de usuarios y generar mayores beneficios económicos.

Su principal objetivo, es mejorar el rendimiento de las aplicaciones mediante la implementación de diversas funcionalidades que van a hacer de la aplicación en cuestión, mucho más manejable, segura y de fácil acceso para los usuarios

## 2.10 Tarjeta ESP32

La tarjeta ESP32 pertenece a una familia de chips SoC que tienen un costo bajo y su consumo de energía es menor. Cuenta con tecnología WI-Fi y Bluetooth de un modo dual integrada. La tarjeta ESP32 cuenta con un microprocesador Tensilica Xtensa LX6 en sus modelos más simples y de doble núcleo e incluye interruptores de antena, balun de radiofrecuencia, amplificador de potencia, amplificador receptor de bajo ruido, filtros y módulos de administración de energía.

### 2.10.1 Características de ESP32

- Procesador dual core Xtensa® LX6 de 32 bits
- Es compatible Arduino con el plugin adecuado (Enseguida vamos)
- Velocidad de reloj: Entre 160 Mhz y 240 Mhz
- 520 kb de RAM
- Wifi integrado: Acces point & Station
- Bluetooth 4.2 2.4 GHz; BT 2.0 y 4.0 BLE
- 36 GPIO pins.
- 16 x Analog-to-Digital Converter (ADC) de 12 bits de resolución y se pueden programar con límite de entrada a 1V, 2 V y 4V
- 2 x Digital to Analog converter DAC de 8 bits.
- Pueden definirse hasta 16 canales de PWM.
- 2 x UART o puertas serie
- 2 x I2C channels y 4 x SPI channels.



## **2.11 Arduino Uno**

Arduino Uno es una tarjeta basada en un microcontrolador llamado atmega328. esta tarjeta cuenta con 14 pines de entrada/salida digital, 6 entradas análogas, un resonador cerámico de 16 MHz, un conector para USB tipo hembra, un Jack para fuente de poder, un conector ICSP y botón de reset.

## **2.12 Cables Dupont**

Los cables Dupont para prototipos, son cables con un conector en cada punta, las cuales hay de diversos tipos ya sean Macho-Macho, Macho-Hembra y Hembra-Hembra, estas puntas se usan normalmente para interconectar varios componentes en una placa de pruebas. Se utilizan de manera general para lograr transferir señales eléctricas de cualquier parte de la placa de prototipos.

## **2.13 Dispositivos automatizables (infrarrojo)**

### **2.13.1 Aire acondicionado Split**

Es un sistema básico que funciona con una unidad externa y una unidad interna controlado a control remoto.

### **2.13.2 Proyectores**

Un proyector está diseñado para recibir la señal de vídeo desde algún dispositivo externo, normalmente un reproductor de DVD, un reproductor de Blu-Ray o una computadora (y proyectar esa señal en una pantalla). Esto se hace mediante la visualización de la imagen representada en la señal de vídeo en una pantalla pequeña en el interior del propio proyector, que se proyecta en una pantalla mediante una luz brillante y una lente. La lente es un pedazo de vidrio con una forma muy específica diseñada para tomar la pequeña imagen y convertirla en una dramáticamente más grande.

## 2.14 Estado del Arte

En esta sección se presentará los trabajos realizados sobre la tecnología Infrarroja mencionando datos relevantes de dicho trabajo.

(Pujols) [1] La mayoría de los dispositivos electrónicos modernos que se usan en el hogar pueden ser controlados remotamente: televisiones, radios, reproductores de multimedia, acondicionadores de aire, puertas, ventanas, luces, entre otras. En el caso de los televisores, radios y acondicionadores de aire, la tecnología más utilizada para la transmisión de los comandos es la transmisión por luz infrarroja. En promedio, se tienen cuatro (4) controles remotos por hogar, que usan en su mayoría infrarrojos para transmitir. Existen en el mercado infinidad de “controles remotos universales”, que ofrecen configuraciones preestablecidas para trabajar con múltiples dispositivos a la vez, permitiendo tener en un mismo dispositivo el control remoto de la TV, del DVD y el receptor de la señal por cable. También existen aplicaciones para teléfonos móviles que tienen la misma finalidad que los “controles remotos universales”. El enfoque de este proyecto es combinar las ventajas que ofrecen las aplicaciones para teléfonos móviles y la comodidad de tener un dispositivo dedicado a una sola función, sumándole la flexibilidad de poder crear o rediseñar tu propio control remoto, a través del mismo dispositivo.

(González) [2] Actualmente, el control remoto de aparatos electrónicos es algo habitual. Es notoria la cantidad de aparatos electrónicos con mando a distancia que hoy en día podemos encontrar en los hogares. Resulta evidente que los mandos a distancia han sido una revolución en todos los ámbitos pues estos dispositivos inalámbricos permiten al usuario ejecutar de manera remota tareas cotidianas como: subir el volumen, comenzar la reproducción de un video, avanzar a la siguiente dispositiva de una presentación, etc.

(Aldean Pacalla y col.) [3] Este Proyecto de Titulación está dirigido a la comunidad en general, para ayudar en el monitoreo y control de los equipos de climatización e iluminación de forma remota, ya que el descuido del usuario por dejar los equipos encendidos genera consumo energético que refleja costo elevado en la planilla eléctrica. El presente proyecto se propone un prototipo en hardware y software libre en Arduino y App Inventor para el control y monitoreo de los equipos de una forma remota, se comprobó a través de experimentaciones la correcta lectura y control de la temperatura. La metodología ágil, Scrum se aplicó en el desarrollo del proyecto por la cual realizamos casos de prueba a las historias de usuario. Esta herramienta ayudará a medir con precisión la temperatura de los equipos de climatización y el control de la iluminación, de esta manera se permite ahorrar energía eléctrica ya que envía una alerta al dispositivo móvil del estado que se encuentra el equipo, ya sea encendido o apagado.

(Nacho) [4] La domótica es un área relativamente nueva que incorpora nuevas tecnologías en los hogares, en particular las que tienen que ver con la automatización, el ahorro energético, la seguridad y el control, para diversas funciones de comodidad y confort dentro del hogar. Inicialmente se brinda toda la teoría necesaria para la adecuada comprensión de la investigación, conceptos básicos sobre la automatización de sistemas, en esta a su vez se describe el funcionamiento de todos los dispositivos electrónicos que se utilizaron en el sistema de control, así como una breve descripción de los sensores que se utilizaron en el mismo. El presente proyecto se basa en el desarrollo de un sistema prototipo aplicado a una maqueta en pequeña escala que permitirá el bienestar y seguridad dentro de cualquier área que sea implementado, capaz de lograr el resguardo de recursos tangibles y el confort para realizar actividades sin esfuerzo alguno, donde se utilizaron sensores de temperatura, de movimiento y otros dispositivos electrónicos que nos ayuden a realizar la automatización de estos aparatos eléctricos mediante una placa Arduino y el control mediante una aplicación móvil elaborada en App Inventor 2 a través de una conexión inalámbrica Bluetooth que será el nexo de la comunicación entre la aplicación con los sensores y actuadores deseados.

El desarrollo de una aplicación móvil que nos ayude a una comunicación e interacción con el controlador, utilizando comandos de voz para dar las órdenes que se desean realizar dentro de la vivienda. Todo ello se realiza utilizando herramientas de software y hardware libre y componentes que existen en nuestro mercado local.

(Mera y col.) [5] El presente trabajo expone el diseño e implementación de un sistema de control de luces incandescentes por medio de un control remoto universal ajustando el diseño al mercado de los hogares ecuatorianos en base a costes, simplicidad de uso y necesidades de la tecnología moderna para la iluminación. Inicialmente se describe los antecedentes de la iluminación y cómo en la actualidad las necesidades tecnológicas han evolucionado para dar paso a las soluciones actualmente propuestas por cualquier sistema de domótica, se explica el alcance de estas soluciones y hacia dónde están enfocadas, la justificación de la realización del presente proyecto y el sector de mercado al cual está dirigido, así como también las perspectivas hacia nuevas tecnologías de luminarias. Luego de haber explicado la problemática y en consecuencia precisado las soluciones demandadas por el proyecto, se detallan las herramientas y tecnologías a utilizar para llevar a cabo la implementación de este. Esto es el estudio de los protocolos existentes en la comunicación por infrarrojos y la selección del más apropiado con su respectivo módulo receptor. El uso del microcontrolador y sus prestaciones a los requerimientos de diseño con el software apropiado para el desarrollo de las instrucciones y procedimientos.

(Tamayo) [6] En el presente artículo se describe el diseño de un equipamiento electrónico para la centralización del control de los electrodomésticos del hogar de forma remota y simple. Se confeccionan tres tipos de módulos que abarcan en su conjunto el control de todos los electrodomésticos. Cada módulo se conecta de forma independiente con una aplicación para el sistema operativo Android mediante bluetooth. Uno de los módulos emplea el canal infrarrojo para comunicarse con los equipos de la casa que empleen mandos a distancia. Los otros dos módulos son utilizados para controlar los dispositivos equivalentes a cargas inductivas como ventiladores, lavadoras o bombas hidráulicas, así como para las cargas resistivas como lámparas. Además, se emplean sensores para complementar la automatización de los electrodomésticos. El diseño propuesto constituye una herramienta para la centralización de las principales funciones del hogar. Su flexibilidad le permite ser utilizado en cualquier vivienda y su precio relativamente económico favorece su implementación.

(Cárdenas Lancheros y col) [7] Este trabajo esta denominado “Aplicación mediante la cual se realice el control y gestión de un sistema de iluminación led basado en la tecnología inalámbrica LoRa”, con este, se desarrolló una herramienta capaz de llevar a cabo la gestión y el control de un sistema de iluminación led, la comunicación de tipo “Broadcast” se realiza a través de transceptores inalámbricos basados en tecnología LoRa (Long Range – Largo Alcance), la cual es una tecnología que permite realizar la comunicación a largas distancias y con un bajo consumo de energía, teniendo como cobertura 5 km en área urbana y 15 km en área rural. El control se efectúa mediante un aplicativo móvil vinculado al sistema transmisor, este a su vez envía la información de control y gestión a dos sistemas receptores, los cuales gobiernan cada uno luminaria led. Los estados en los cuales pueden entrar las luminarias son encendido, apagado y porcentajes de atenuación. El sistema abre paso a proporcionar un ahorro energético donde se implemente, ya que está diseñado para asignar comportamientos a las luminarias led de acuerdo con las necesidades de un usuario, evitando que estén todo el tiempo suministrando el 100 % de su capacidad luminosa, lo cual se ve reflejado finalmente en disminución de consumo energético.

(Colomer) [8] El método comprende procesar mediante un sistema electrónico una primera señal digital obtenida a partir de una señal de infrarrojos recibida (proveniente por ejemplo de un mando a distancia), para obtener una segunda señal digital, o señal digital original, con la cual se modula una señal de radiofrecuencia que se inyecta en un punto de una línea eléctrica. También comprende demodular dicha señal de radiofrecuencia en otro punto de la línea

eléctrica, para extraer dicha señal digital original, procesarla mediante un sistema electrónico, para obtener una señal digital modificada, con la cual generar una señal de infrarrojos, y emitir dicha señal de infrarrojos hacia un dispositivo a controlar (por ejemplo, un televisor). El dispositivo y el sistema, que incorpora varios de dichos dispositivos, están adaptados para aplicar el método propuesto.

(Alarcón) [9] El objetivo de nuestra aplicación es poder controlar las funciones básicas de nuestros aparatos electrónicos con un ordenador y un sistema de emisión/recepción de infrarrojos. De fácil manejo gracias a su interfaz clara y sencilla con botones, gráficos y formularios que nos permitirán registrarnos como usuarios para poder almacenar aparatos electrónicos y los comandos de estos, para después utilizarlos. También compartirlos con otros usuarios registrados si así lo deseamos. El resultado no pasa por la red, pero contiene lo necesario para poder expandirse y ser capaces en un futuro de apagar la televisión de nuestra casa desde la oficina con un clic y así evitar que se quede encendida consumiendo energía. Seguridad, facilidad, accesibilidad, ayuda al medio ambiente... esa es la base de este proyecto del que, gracias a la investigación y a su base, se pueden desarrollar versiones o ampliarse.

(Pérez) [10] El presente proyecto implementa el diseño, cálculo y la construcción de un equipo de efectos digitales para guitarra eléctrica, programados en una tarjeta de DSP controlada por un Arduino UNO y comandado por infrarrojos. Se estudian los diferentes tipos de efectos, se analizan diferentes configuraciones de hardware y sus limitaciones y se construye físicamente un equipo plenamente funcional. Para ello se han empleado los siguientes programas de diseño: PsPice (Simular la adaptación de la señal analógica), Fritzing (Diseñar las Protoboards), Eagle (diseño de la pcb), AutoCAD (Diseño del contenedor 3D), Entorno de programación de Arduino (Programa de control por infrarrojos/teclado y del DSP). Se ha fabricado la pcb por la fresadora del departamento de electrónica industrial y el contenedor con una impresora 3D de plástico. El equipo ofrece la información del efecto programado junto con los parámetros de este a través del monitor serie del Arduino y muestra amplia información ...

# **CAPITULO III**

## **DESARROLLO**

## 3.1 Requisitos de software y equipo

Para este proyecto, fue utilizado el siguiente equipo:

### 3.1.1 Laptop

<b>MARCA</b>	Lenovo ideapad 110-15ACL
<b>MODELO</b>	80TJ
<b>SISTEMA OPERATIVO</b>	Windows 10
<b>PROCESADOR</b>	AMD A6
<b>MEMORIA RAM</b>	8 GB DRR3
<b>TARJETA GRAFICA</b>	AMD Radeon™ R4 Graphics
<b>ALMACENAMIENTO SSD</b>	256 GB

*Tabla 1 Laptop*

### 3.1.2 Smartphone

<b>MARCA</b>	Samsung
<b>MODELO</b>	SM-G950U
<b>VERSIÓN ANDROID</b>	9
<b>CHIPSET</b>	Qualcomm MSM8998 Snapdragon 835
<b>CPU</b>	Octa-core 3 GHz Cortex-A53
<b>GPU</b>	Adreno 540
<b>MEMORIA RAM</b>	4 GB
<b>ALMACENAMIENTO</b>	64 GB
<b>TAMAÑO DE PANTALLA</b>	5.8 pulgadas
<b>RESOLUCIÓN DE PANTALA</b>	1440 x 2960 pixeles

*Tabla 2 Smartphone*



### 3.1.3 Tarjetas y Componentes

Tarjeta ESP32
Cables Dupont
Diodo Infrarrojo
Tarjeta Arduino Uno
Sensor receptor infrarrojo

Tabla 3 Tarjetas y componentes

### 3.1.4 Investigar y seleccionar el entorno de desarrollo

Al seleccionar un entorno de desarrollo se analizó los diversos entornos en los que se pueden crear aplicaciones móviles, al igual que los softwares que permiten el uso de tarjetas como la ESP32.

<b>Entorno de desarrollo</b>
Android Studio
Arduino

Tabla 4 Entornos de Desarrollo

### 3.1.5 Descargas

Una vez elegidos los entornos de desarrollo, se elige la mejor versión y también la más estable para poder descargarla.

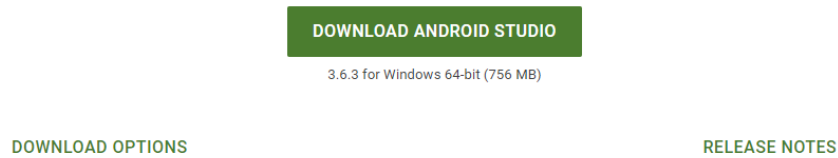
La versión seleccionada en Android Studio fue la 3.6.3, para poder descargar esta versión se hizo lo siguiente.

Para descargar Android Studio se entra al siguiente enlace:  
<https://developer.android.com/studio>

Una vez en la página oficial se buscó versión antes mencionada donde se muestra un botón que nos indica (Download Android Studio) para que el programa proceda con la descarga.

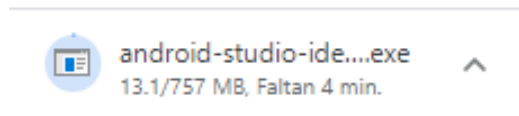
# android studio

Android Studio provides the fastest tools for building apps on every type of Android device.



*Figura 5 Descarga Android*

Cuando empieza la descarga del paquete de Android Studio tiene una capacidad de algo menos de 1GB, por lo que se debe de estar en espera para la terminación de la descarga.



*Figura 6 Tiempo de descarga*

Una vez que se haya completado el proceso de descarga es necesario llevar a cabo la debida instalación de Android Studio.

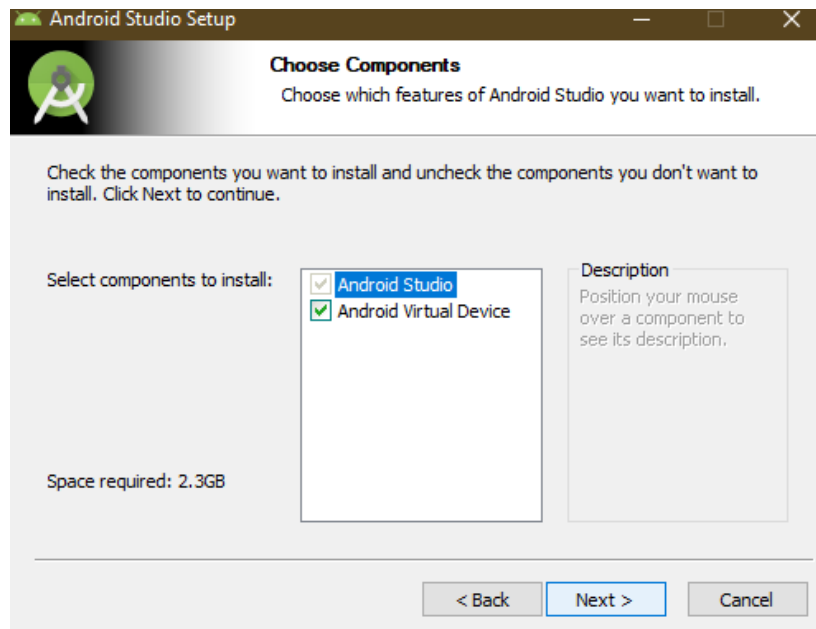


*Figura 7 Inicio de instalación*

Como siguiente paso pide descargar e instalar el JDK de Java en caso de no tenerlo es necesario descargarlo y se puede seguir con la instalación.

En la Próxima ventana del asistente, debemos de seleccionar los paquetes que deseamos instalar.

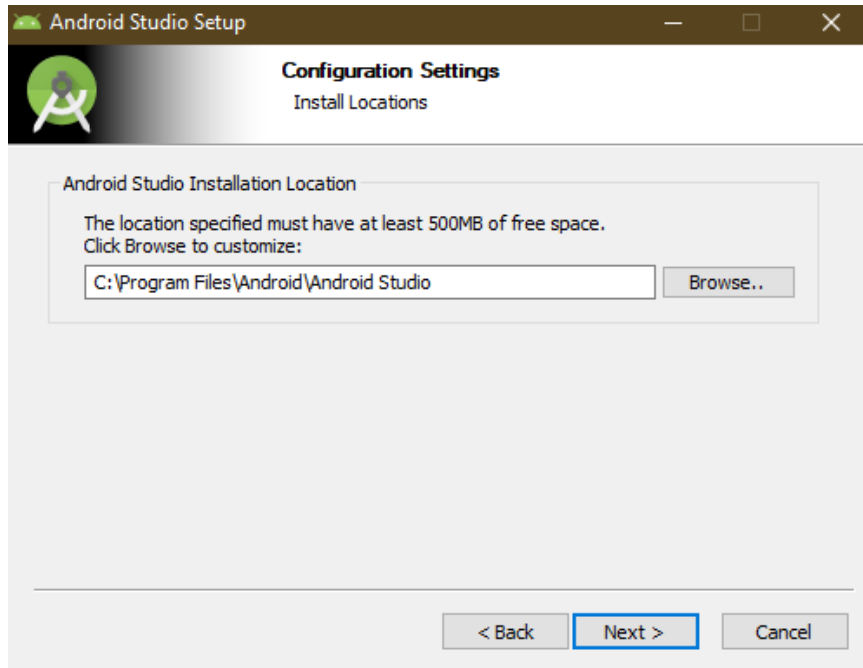
- El editor de Android Studio
- El SDK de Android
- El emulador de Android



*Figura 8 Proceso de instalación 1*

Seguimos con el asistente y como siguiente paso debemos de elegir la ruta donde vamos a instalar el editor de Android.

*Figura 9 Proceso de instalación 2*



Con todo lo anterior, el asistente está listo. El proceso de instalación esta completo.

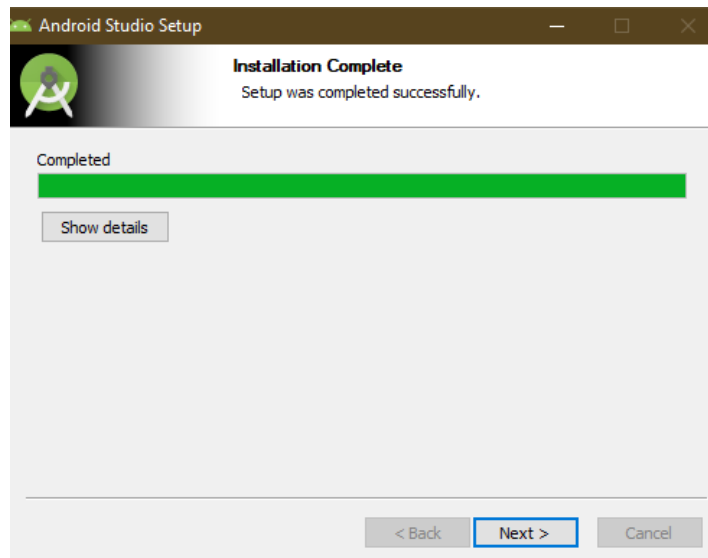


Figura 10 Proceso de instalación 3

Una vez finalizado el proceso de instalación presionamos en (Finish), ya tendremos nuestro IDE listo para empezar el proceso de programación de nuestro sistema.

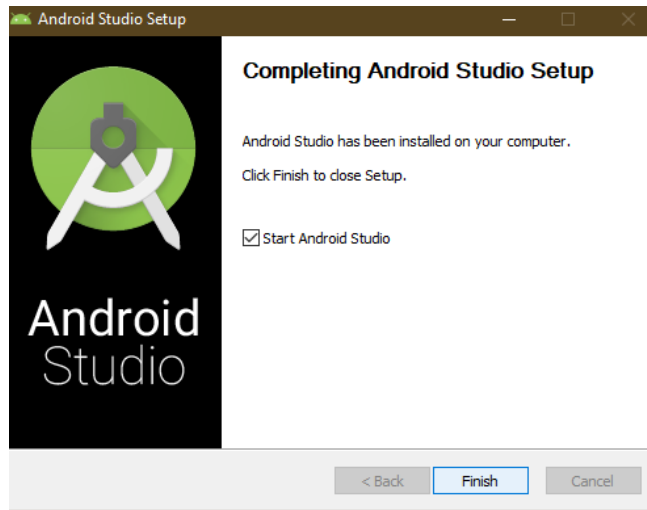


Figura 11 Proceso de instalación 4

Automáticamente Android estudio empezara iniciarse.



Figura 12 Inicio de Android

Para poder trabajar en Arduino es necesario llevar a cabo la descarga de este IDE. Este es muy ligero y sencillo y de muy fácil uso a comparación de otros Softwares.

Para llevar a cabo la descarga de ello se dirigió al siguiente enlace:  
<https://www.arduino.cc/en/Main/Software>

La plataforma para la cual se descargo es Windows:



Figura 13 Descarga de Arduino

Al dar clic en Windows Automáticamente comenzara la descarga:

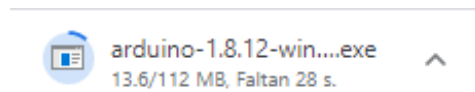


Figura 14 Tiempo de descarga Arduino

Una vez finalizada la descarga, Ejecutamos el archivo .exe y clic en el botón I Agree.

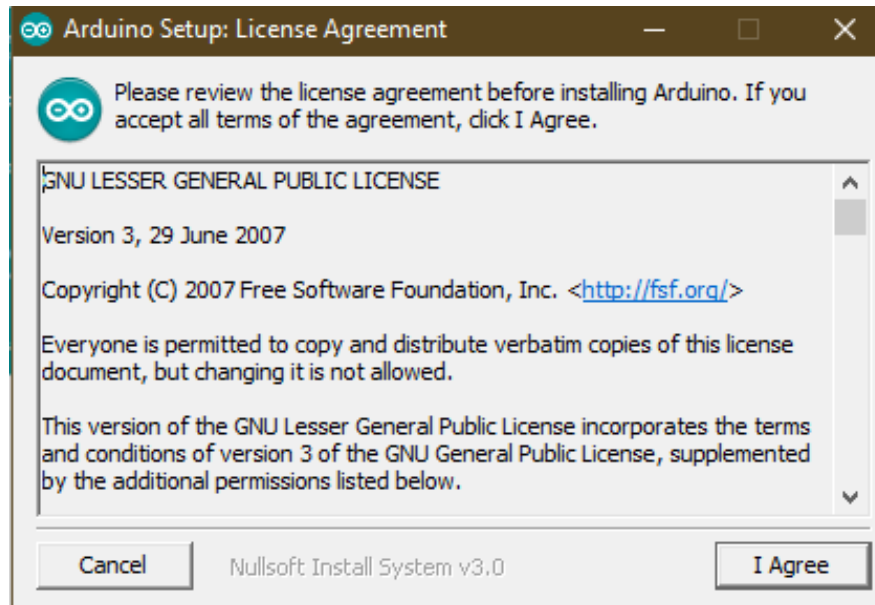


Figura 15 Proceso de Instalación Arduino

Seleccionamos los componentes de instalación de Arduino.

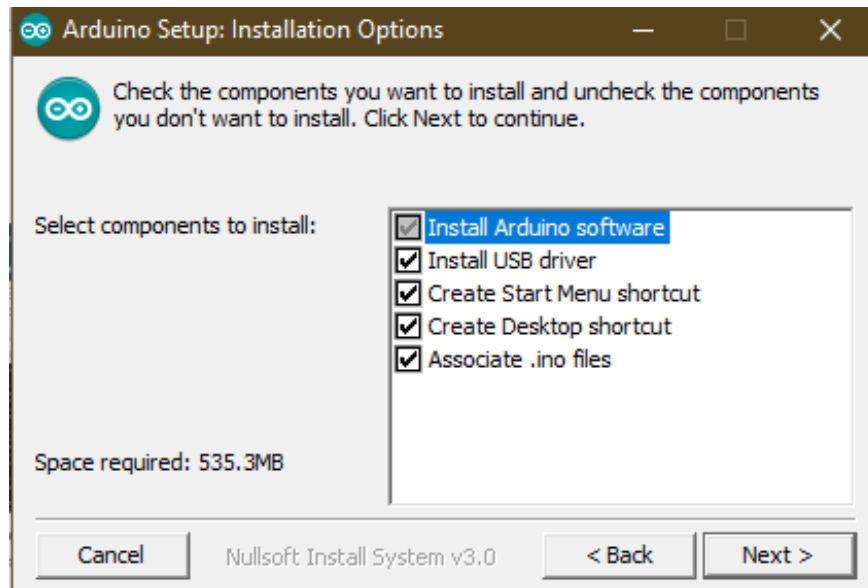


Figura 16 Proceso de Instalación Arduino 2

Una vez realizado los pasos anteriores, se procede a seleccionar la carpeta donde se desea instalar.

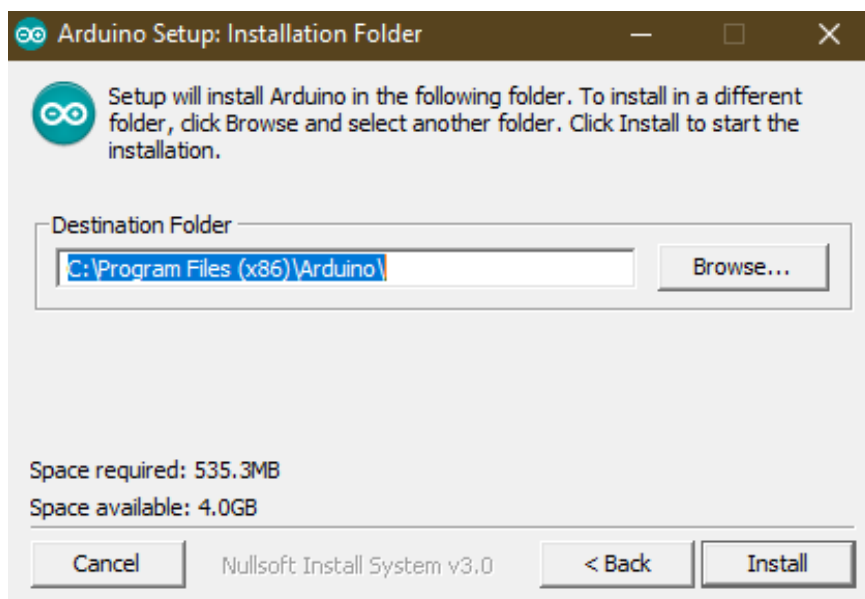


Figura 17 Proceso de Instalación Arduino 3

Seguido de haber seleccionado la carpeta de instalación, se procede a instalar.

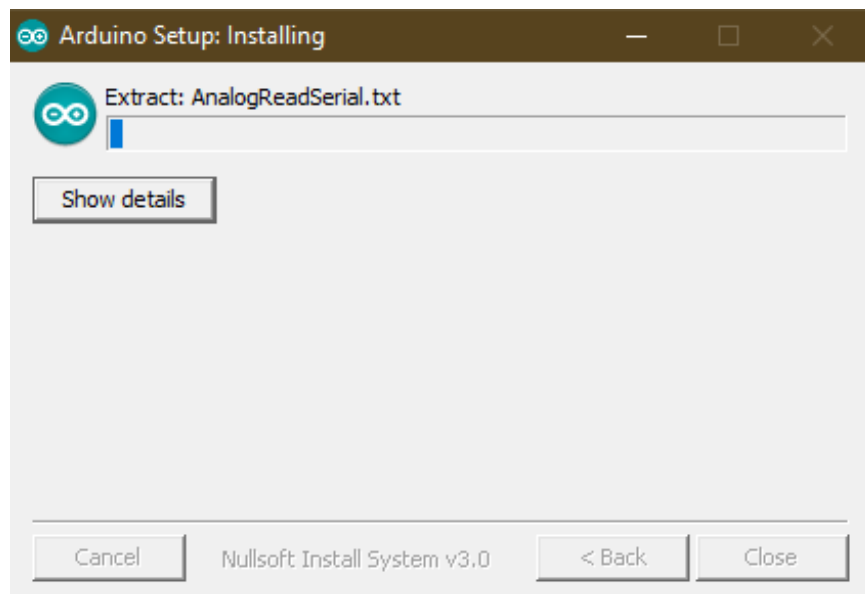
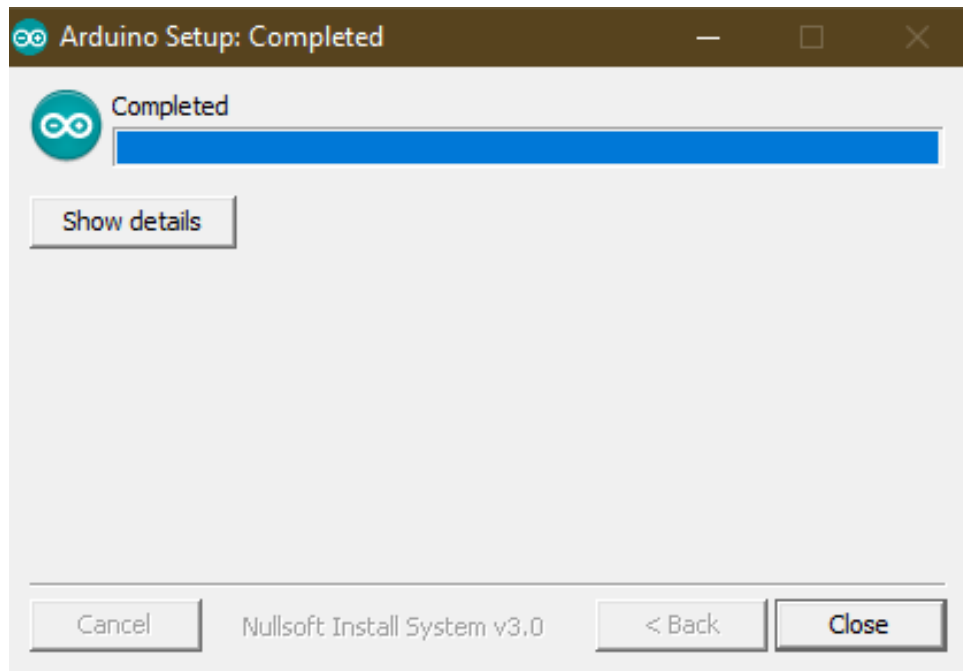


Figura 18 Proceso de Instalación Arduino 4



Por último, mostrara una ventana sobre la instalación completada, clic en (close) para terminar y automáticamente iniciara el programa.



*Figura 19 Proceso de Instalación Arduino 5*

Después de haberse instalado adecuadamente Arduino se abre automáticamente.

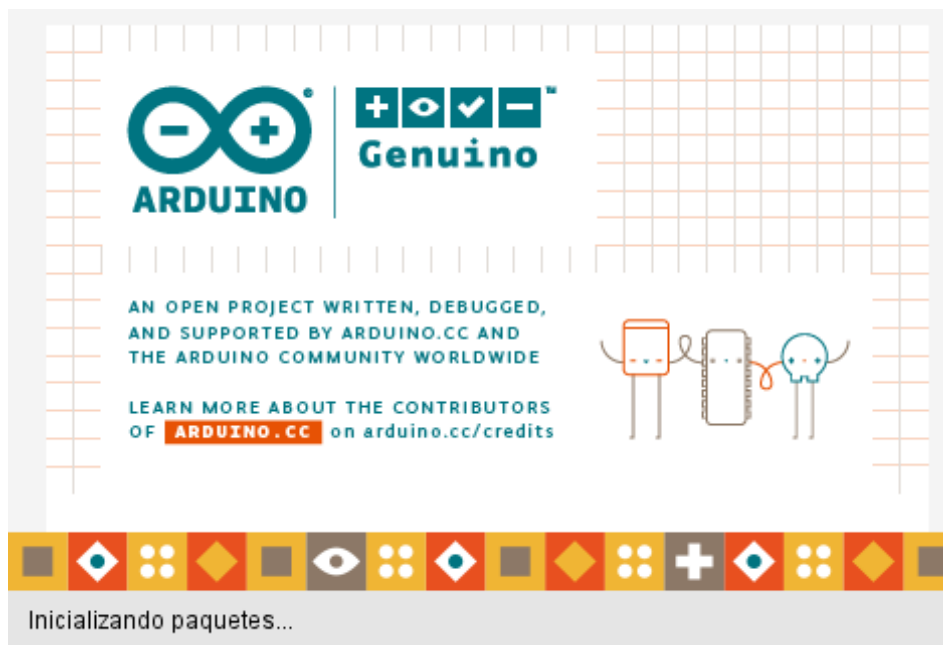


Figura 20 Inicio de Arduino

### 3.1.6 Lenguaje de programación

#### Kotlin

Kotlin es un lenguaje de programación de tipado estático para Java, Android, el navegador y próximamente LLVM, 100% interoperable con Java, creado por JetBrains, la compañía responsable, entre otros, de IntelliJ Idea.

#### Características de Kotlin

- **Compatibilidad:** Kotlin es totalmente compatible con el JDK a partir de la versión 6, esto es importante ya que garantiza que Kotlin puede ejecutarse en dispositivos Android antiguos sin problemas. Kotlin además es compatible con Android Studio y con las herramientas de compilación de Android.
- **Performance:** Una aplicación de Kotlin es tan rápida como una equivalente de Java, gracias a una estructura de bytecode muy similar. Con el soporte de Kotlin para funciones en línea, el código que usa lambdas y a menudo se ejecuta incluso más rápido que el mismo código escrito en Java.
- **Interoperabilidad:** Kotlin es 100% interoperable con Java, permitiendo utilizar todas las bibliotecas existentes de Android en una aplicación de Kotlin. Esto incluye el procesamiento de anotaciones, por lo que el DataBinding y Dagger también funcionan.
- **Tiempo de compilación:** Kotlin es compatible con la compilación incremental eficiente, por lo que, aunque hay alguna sobrecarga adicional para las compilaciones limpias, las creaciones incrementales suelen ser tan rápidas o más rápidas que con Java.
- **Curva de aprendizaje:** Para un desarrollador de Java, empezar con Kotlin es muy fácil. El convertidor automatizado de Java a Kotlin incluido en el complemento Kotlin ayuda con los primeros pasos. Kotlin Koans ofrecen una guía a través de las características clave de la lengua con una serie de ejercicios interactivos.

### 3.1.7 Instalación de la tarjeta ESP32 en el IDE de Arduino

Para poder instalar los servicios de la tarjeta, se abre el IDE de Arduino, se le da clic en la pestaña que dice Archivo y se selecciona la opción de preferencias.

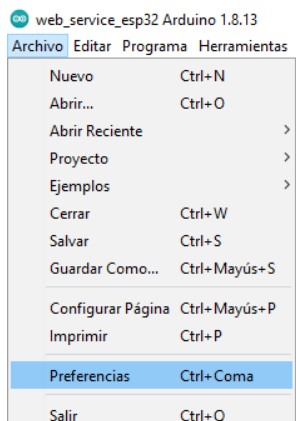


Figura 21 Preferencias de Arduino

Se abre una ventana y en campo de Gestor de URLs adicionales de tarjetas se pega el siguiente enlace: [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json).

Si ya hay un enlace puesto, solo se agrega una coma al final del primer link y se agrega el de la nueva tarjeta y se le da ok.

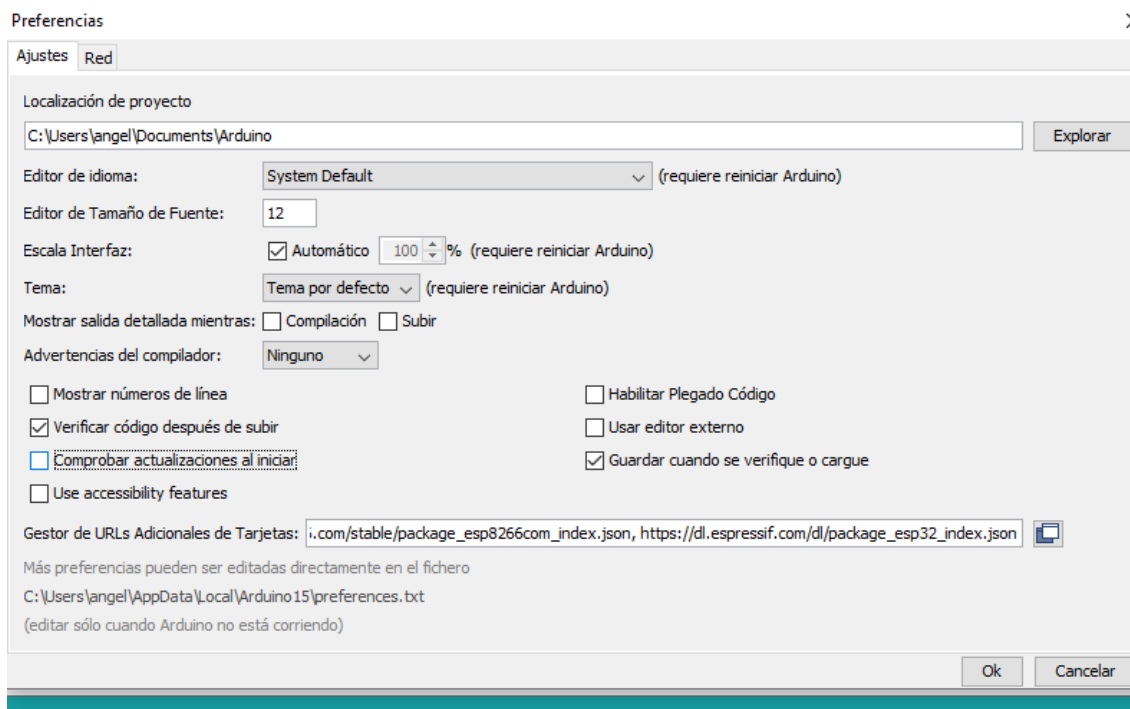


Figura 22 Menú Preferencias

Posteriormente se selecciona la pestaña de Herramientas y se selecciona la opción Placa, esta desplegara una lista de las placas disponibles, para agregar la ESP32 se escoge la opción Gestor de tarjetas.

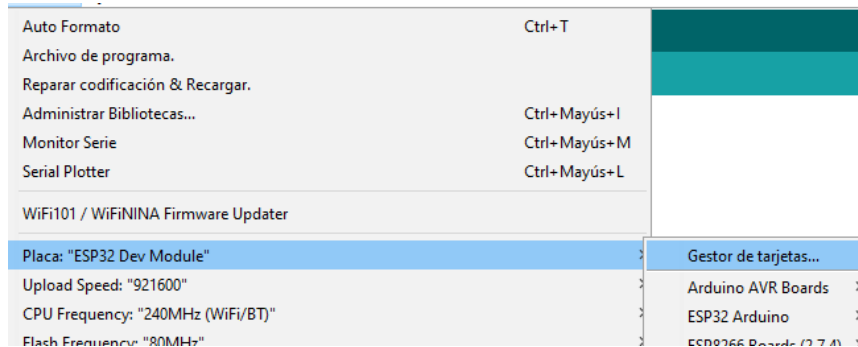


Figura 23 Opción de Placas

En la venta que se abre se escribe ESP32 y se le da en la opción de instalar.

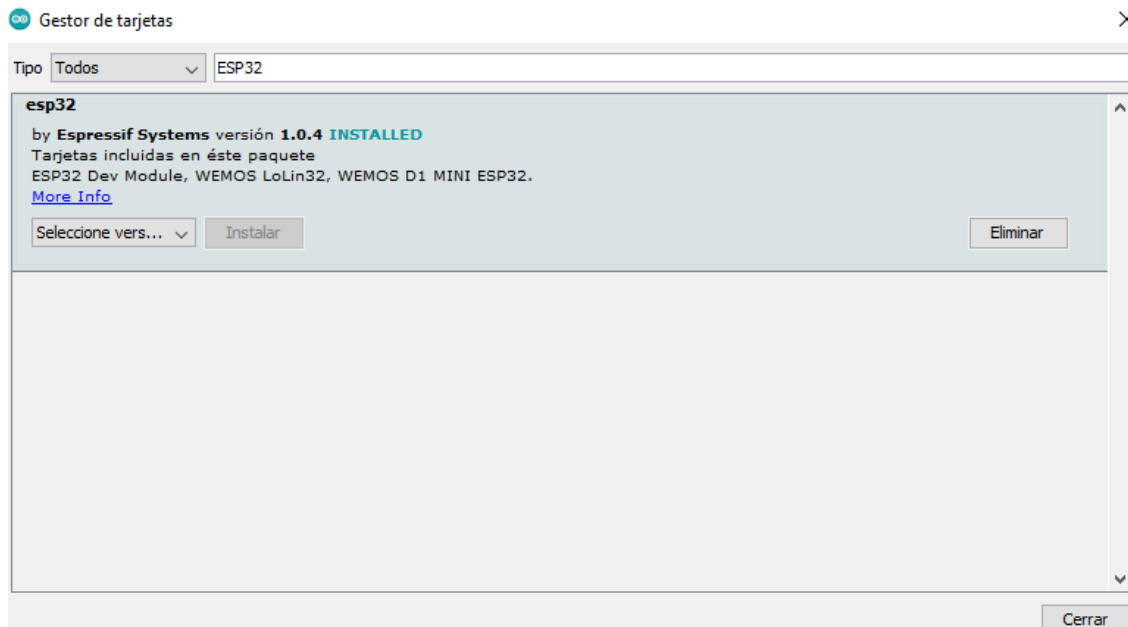


Figura 24 Librería ESP32

Una vez finalizada la instalación se vuelve a seleccionar la pestaña de Herramientas y la opción que dice Placa y ya se puede elegir entre las diversas placas basadas en la plataforma ESP32.

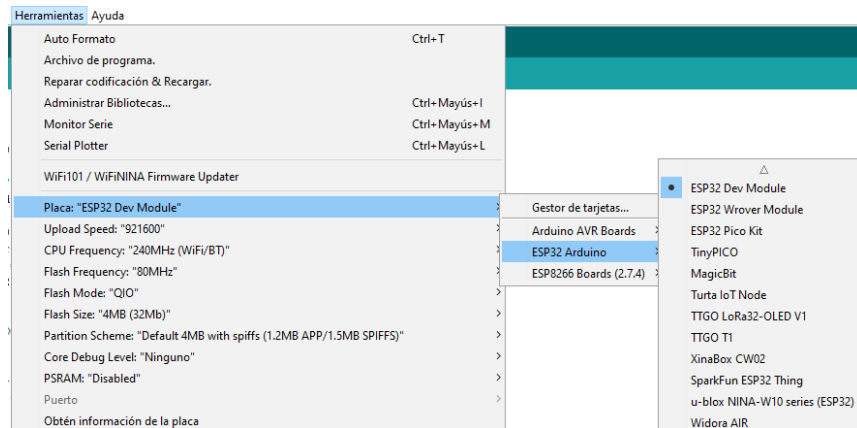


Figura 25 Tarjetas disponibles de plataforma ESP32

### 3.1.8 Librerías de Arduino

#### Librería WiFi.h

Con la librería WiFi permite que una placa ESP32 se conecte a Internet, ya que esta servir como servidor que acepta conexiones entrantes o como cliente que realiza conexiones salientes. La biblioteca admite el cifrado WEP y WPA2 Personal, pero no WPA2 Enterprise. También se debe tener en cuenta que, si el SSID no se transmite, el escudo no se puede conectar.

Para instalar la librería de WiFi, desde el mismo IDE de Arduino se seleccionó la pestaña de Herramientas y selecciono la opción de administrador de bibliotecas.

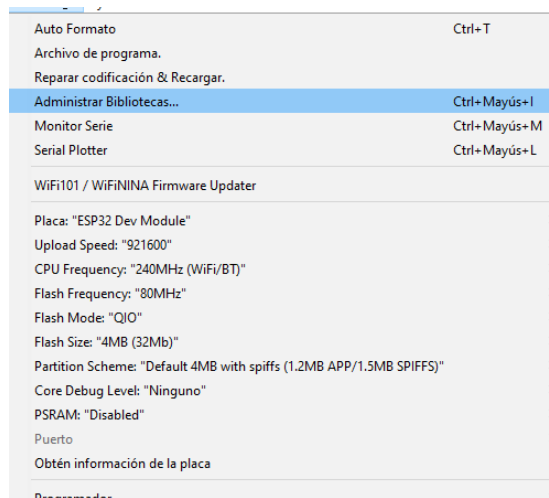


Figura 26 Opción de Administrador de Bibliotecas

Una vez estando en el administrador de bibliotecas se hizo la búsqueda de dicha librería y se instaló de manera rápida.

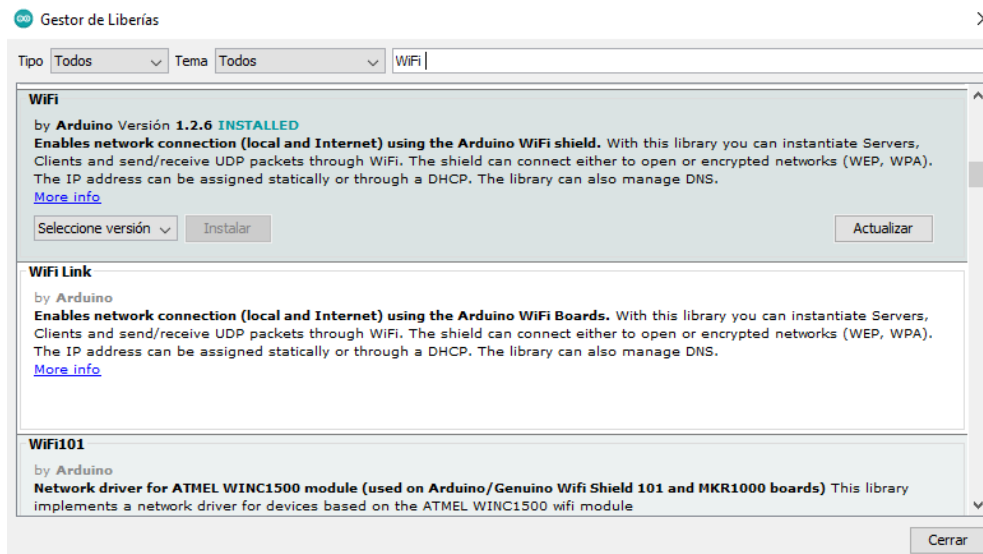


Figura 27 Biblioteca WiFi

## Librería IR Remote

IRremote es una de las librerías más usadas y completas para trabajar con protocolos de controles infrarrojos, tiene implementado varios protocolos de las marcas más conocidas como Sony, LG, Samsung, Sanyo, etc. y con esta librería se puede tanto enviar como recibir señales infrarrojas.

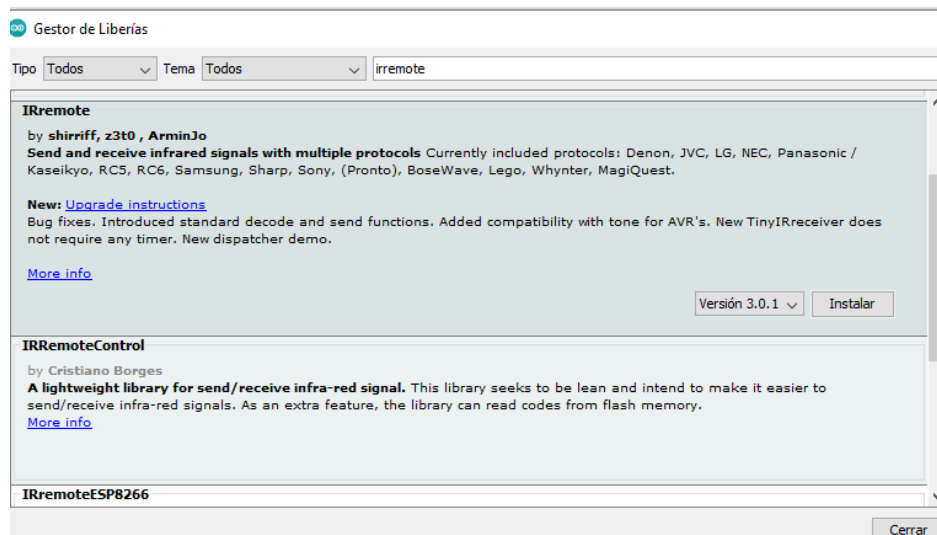


Figura 28 Biblioteca IRRemote

## 3.2 Diseño

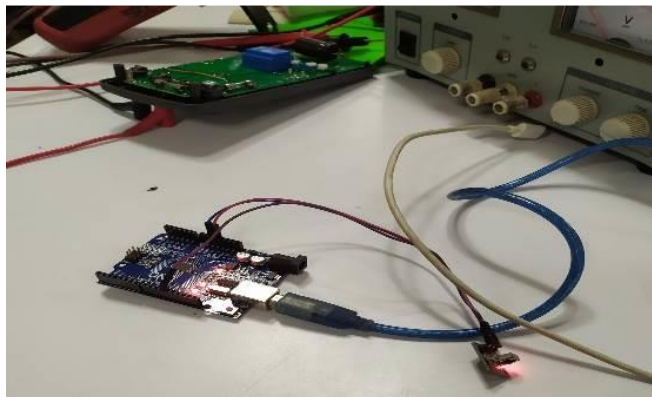
### 3.2.1 Captura de señales infrarrojas

Para capturar las señales infrarrojas emitidas por los controles remotos de aires acondicionados y videoproyectores, se utilizó un programa en codificado en la plataforma con ayuda de una placa Arduino uno y un receptor infrarrojo.



*Figura 29 Captura de códigos raw*

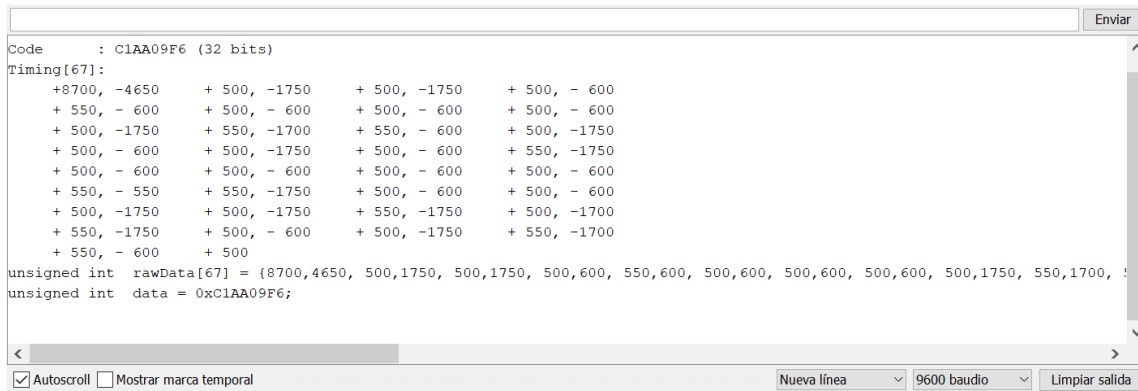
Esta señal es captada por el sensor infrarrojo para ser procesada por la aplicación en Arduino.



*Figura 30 Tarjeta Arduino Uno*



La aplicación hecha en Arduino muestra la señal que emiten los controles, estos datos son conocidos como raw.



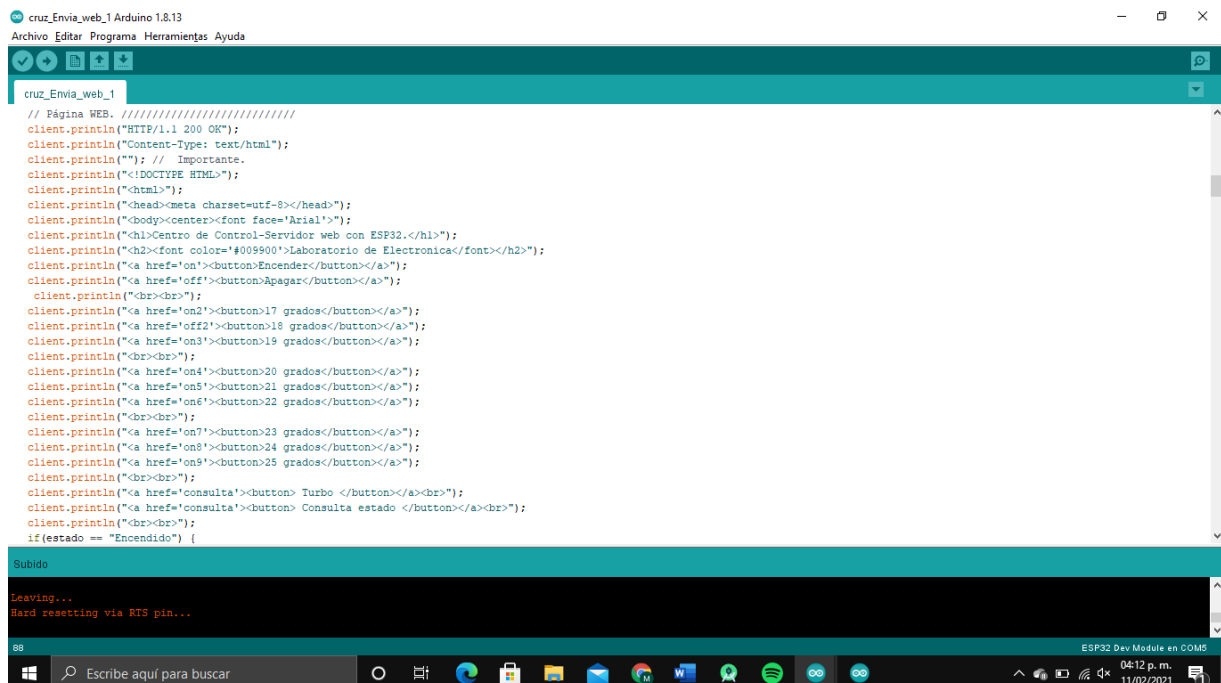
```
Code : C1AA09F6 (32 bits)
Timing[67]:
+8700, -4650 + 500, -1750 + 500, -1750 + 500, - 600
+ 550, - 600 + 500, - 600 + 500, - 600 + 500, - 600
+ 500, -1750 + 550, -1700 + 550, - 600 + 500, -1750
+ 500, - 600 + 500, -1750 + 500, - 600 + 550, -1750
+ 500, - 600 + 500, - 600 + 500, - 600 + 500, - 600
+ 550, - 550 + 550, -1750 + 500, - 600 + 500, - 600
+ 500, -1750 + 500, -1750 + 550, -1750 + 500, -1700
+ 550, -1750 + 500, - 600 + 500, -1750 + 550, -1700
+ 550, - 600 + 500
unsigned int rawData[67] = {8700,4650, 500,1750, 500,1750, 500,600, 550,600, 500,600, 500,600, 500,600, 500,1750, 550,1700, ...};
unsigned int data = 0xC1AA09F6;
```

Figura 31 Códigos raw

### 3.3 Implementación

#### 3.3.1 Creación de página web en Arduino

Para emplear los códigos raw que se recolectaron, se creó una página web desde el IDE de Arduino.



```
// Página WEB. //////////////////////////////////////
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // Importante.
client.println("<DOCTYPE HTML>");
client.println("<html>");
client.println("<head<meta charset=utf-8></head>");
client.println("<body<center<font face=Arial>");
client.println("<h1>Centro de Control-Servidor web con ESP32.</h1>");
client.println("<h2<font color=#009900>Laboratorio de Electronica</h2>");
client.println("<a href=on1>buttonEncender</a>");
client.println("<a href=off1>buttonApagar</a>");
client.println("<br>");
client.println("<a href=on2>button17 grados</a>");
client.println("<a href=off2>button18 grados</a>");
client.println("<a href=on3>button19 grados</a>");
client.println("<br>");
client.println("<a href=on4>button20 grados</a>");
client.println("<a href=on5>button21 grados</a>");
client.println("<a href=on6>button22 grados</a>");
client.println("<br>");
client.println("<a href=on7>button23 grados</a>");
client.println("<a href=on8>button24 grados</a>");
client.println("<a href=on9>button25 grados</a>");
client.println("<br>");
client.println("<a href=consulta>button Turbo </a><br>");
client.println("<a href=consulta>button Consulta estado </a><br>");
client.println("<br>");
if(estado == "Encendido") {
```

Figura 32 Página Web en Arduino

### 3.3.2 Creación de proyecto en AndroidStudio

Para el comienzo de la aplicación, se creó un nuevo proyecto en AndroidStudio, aquí se seleccionó que la aplicación se haría en el lenguaje de programación kotlin y se le dio un nombre.

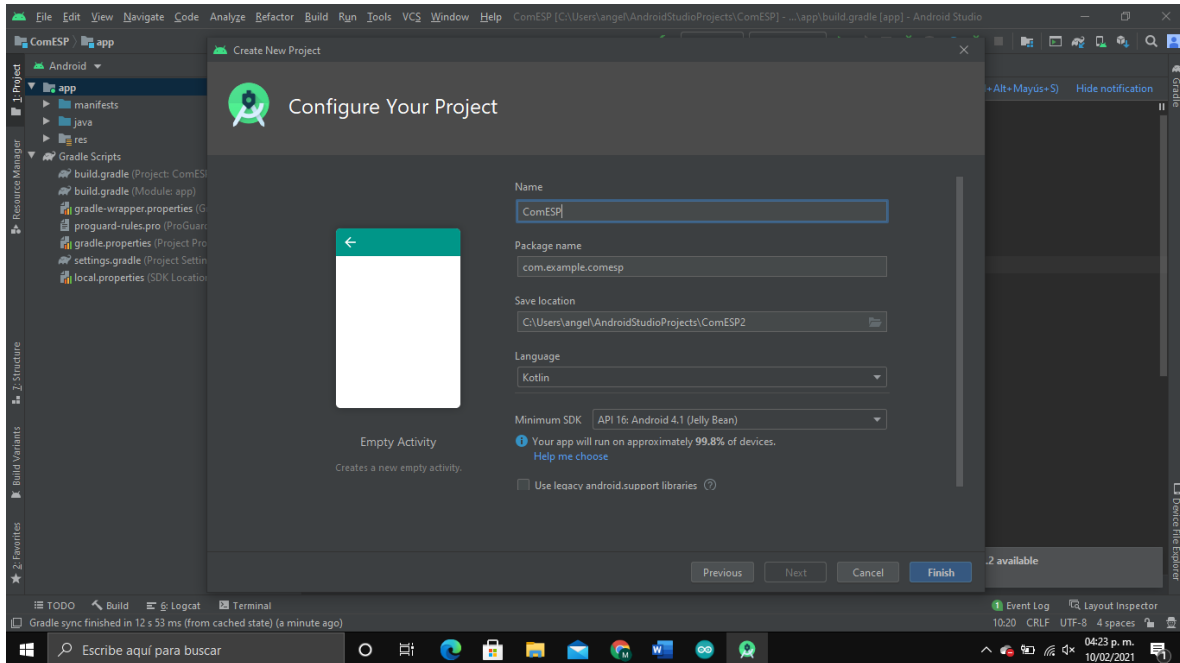
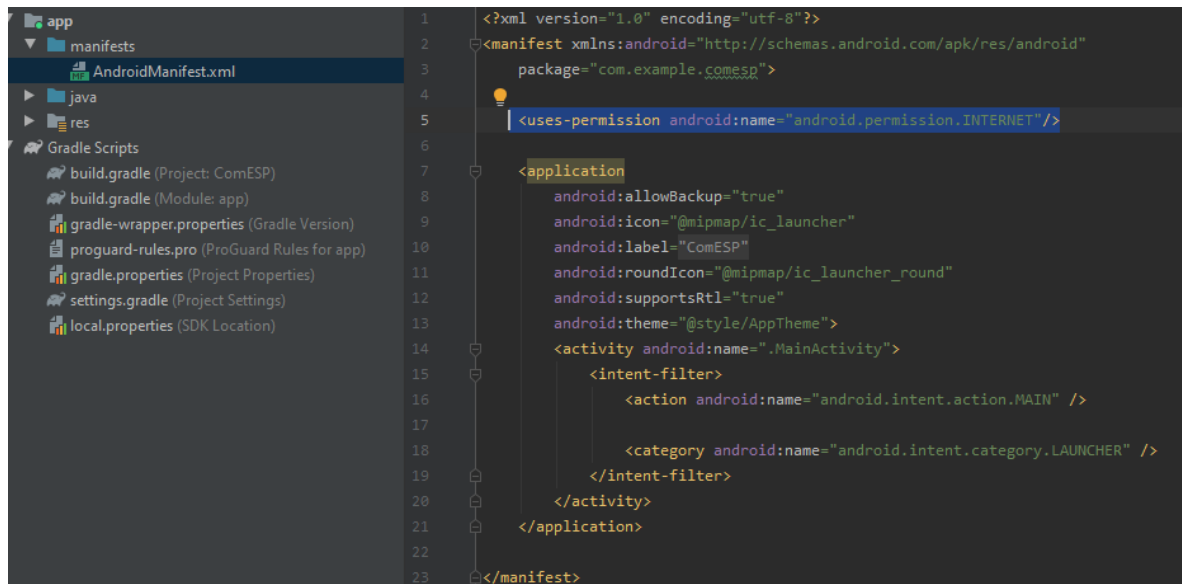


Figura 33 Creación de Proyecto en Android

### 3.3.3 Permisos de Internet

Para el buen funcionamiento de la aplicación se le agregaron los correspondientes permisos de Internet. Estos permisos se agregan en el archivo AndroidManifest.xml.

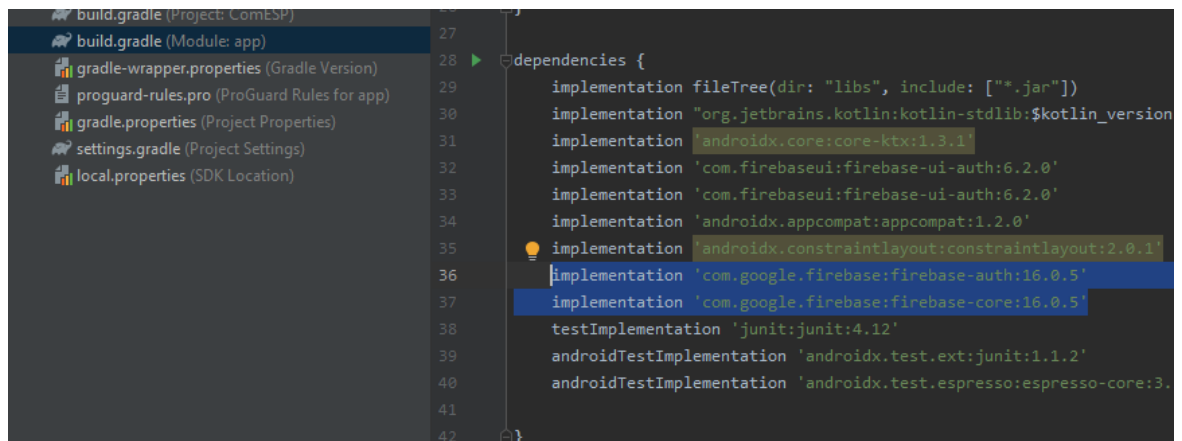


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3 package="com.example.comesp">
4
5 <uses-permission android:name="android.permission.INTERNET"/>
6
7 <application
8     android:allowBackup="true"
9     android:icon="@mipmap/ic_launcher"
10    android:label="ComESP"
11    android:roundIcon="@mipmap/ic_launcher_round"
12    android:supportRtl="true"
13    android:theme="@style/AppTheme">
14    <activity android:name=".MainActivity">
15        <intent-filter>
16            <action android:name="android.intent.action.MAIN" />
17
18            <category android:name="android.intent.category.LAUNCHER" />
19        </intent-filter>
20    </activity>
21 </application>
22
23 </manifest>
```

Figura 34 Permisos de Internet

### 3.3.4 Dependencias

En el archivo build.gradle se agregaron las dependencias que se utilizaron para trabajar con ayuda Firebase.



```
27 dependencies {
28     implementation fileTree(dir: "libs", include: ["*.jar"])
29     implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
30     implementation "androidx.core:core-ktx:1.3.1"
31     implementation 'com.google.firebase:firebase-core:16.0.5'
32     implementation 'com.google.firebase:firebase-auth:16.0.5'
33     implementation 'com.google.firebase:firebase-ui-auth:6.2.0'
34     implementation 'com.google.firebase:firebase-ui-core:6.2.0'
35     implementation 'androidx.appcompat:appcompat:1.2.0'
36     implementation 'androidx.constraintlayout:constraintlayout:2.0.1'
37     implementation 'com.google.firebase:firebase-core:16.0.5'
38     implementation 'com.google.firebase:firebase-auth:16.0.5'
39     testImplementation 'junit:junit:4.12'
40     androidTestImplementation 'androidx.test.ext:junit:1.1.2'
41     androidTestImplementation 'androidx.test.espresso:espresso-core:3.
42 }
```

Figura 35 Dependencias

### 3.3.5 Conexión con Firebase

Para poder conectar la aplicación con los servicios de Firebase, se seleccionó la pestaña Tools y la opción de Firebase.

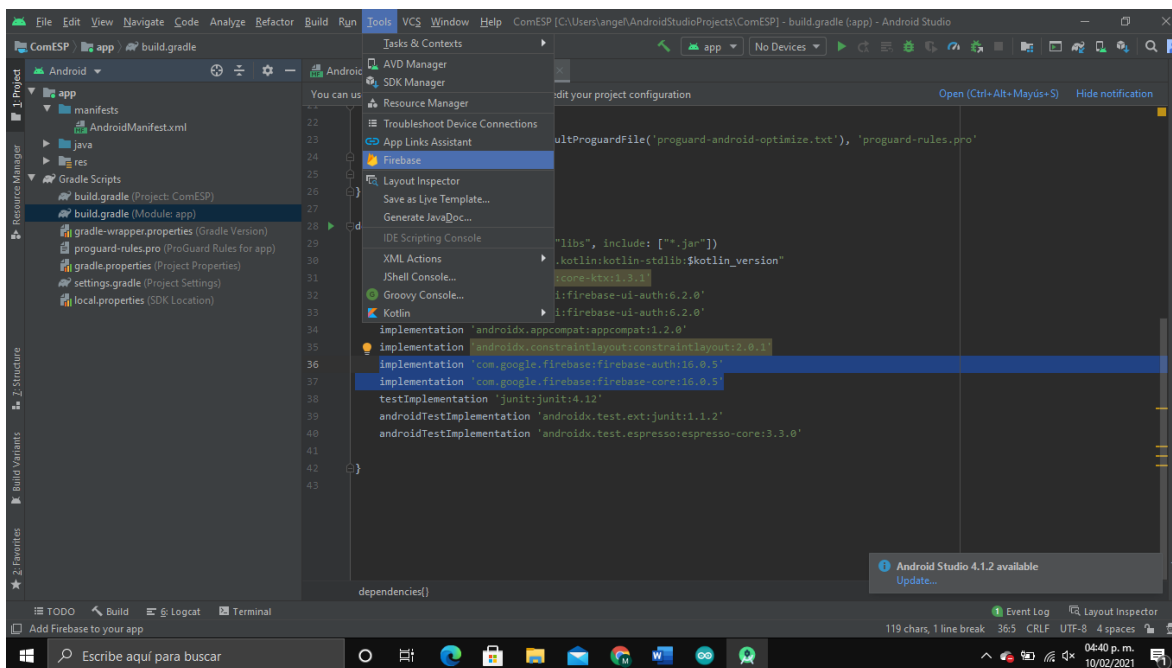


Figura 36 Conexión con Firebase

Una vez que se seleccionó la opción de Firebase, en el menú que aparece se lado derecho de la pantalla, se eligió la opción de autenticación.

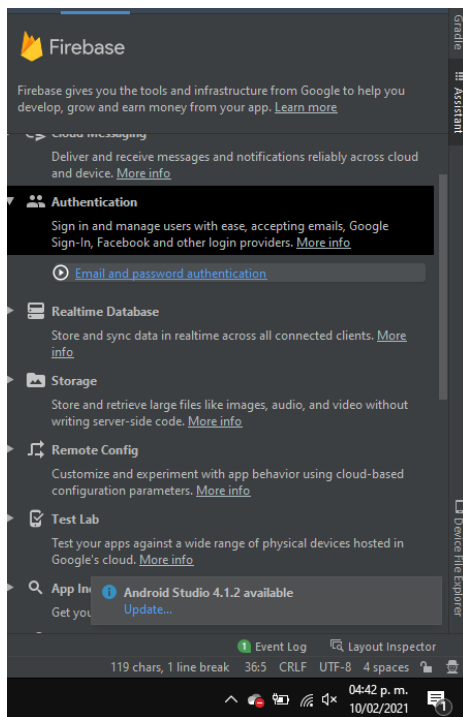


Figura 37 Autenticación

En la parte de autenticación se siguieron unos pasos que el mismo AndroidStudio proporciona.

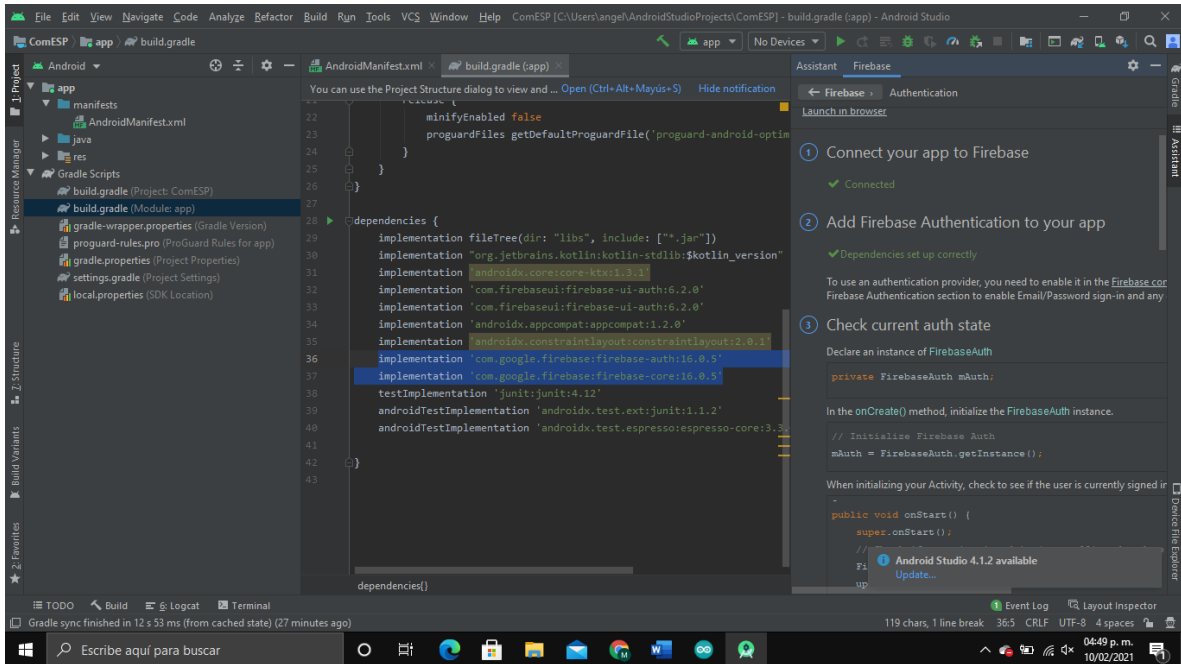


Figura 38 Dependencias de Firebase

### 3.3.6 Habilitar inicio de sesión en Firebase

Firebase brinda varias opciones para hacer un inicio de sesión, para poder iniciar sesión en la aplicación se escogió la de Google.

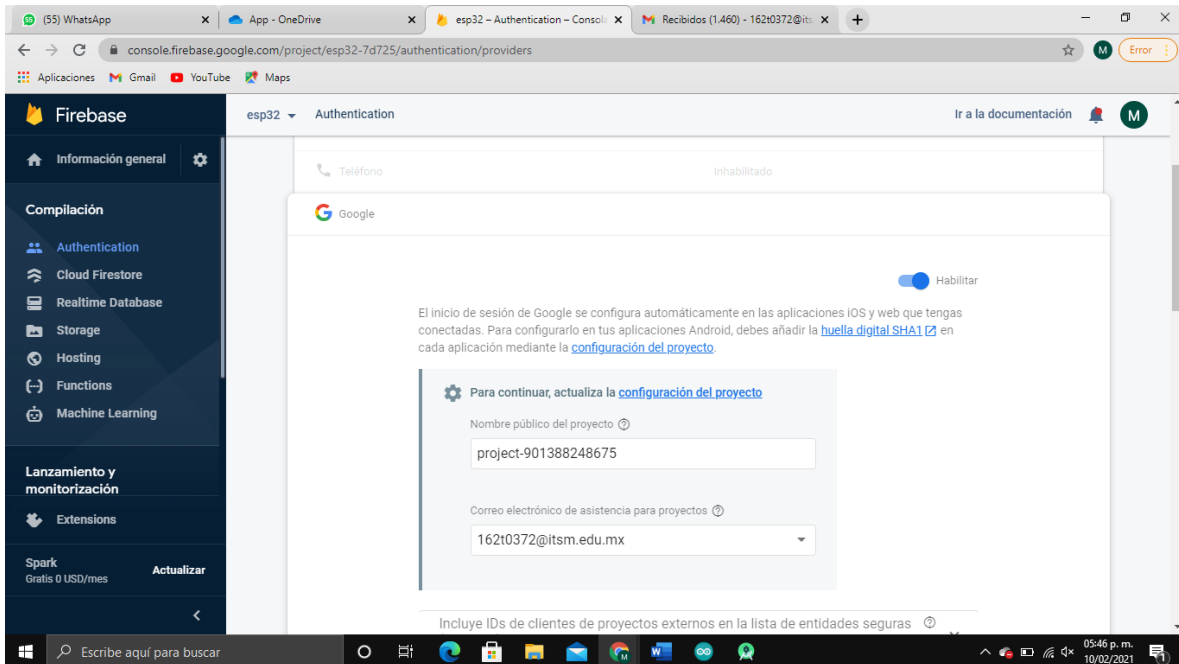


Figura 39 Habilitar Servicio de Google

### 3.3.7 Función de login

Esta función permite que el usuario pueda hacer un inicio de sesión sin la necesidad de crear alguna cuenta, ya que en esta función inicia la sesión con el correo de Google que el usuario tenga en su móvil.

```
//función de login
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if(requestCode == MY_REQUEST_CODE){
        val response = IdpResponse.fromResultIntent(data)
        if(resultCode == Activity.RESULT_OK){
            val user = FirebaseAuth.getInstance().currentUser
            Toast.makeText(context, this, text: ""+user!!.email,Toast.LENGTH_SHORT).show()
            btn_sing_out.isEnabled = true
        }else{
            Toast.makeText(context, this, text: ""+ response!!.error!!.message,Toast.LENGTH_LONG).show()
        }
    }
}
```

Figura 40 Función login

### 3.3.8 Función onCreate

En esta función se hace la selección de los servicios con los cuales se puede iniciar sesión en la aplicación este caso se eligió solo el inicio de sesión mediante una cuenta de correo de Google y también se creó un botón para que este permita el cierre de la sesión en caso de que el usuario así lo quiera.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    //se puede seleccionar los servicios para iniciar sesion
    provider = Arrays.asList<AuthUI.IdpConfig>{
        AuthUI.IdpConfig.GoogleBuilder().build()
    }
    tiposdelogin()
    btn_sing_out.setOnClickListener { it: View?
        AuthUI.getInstance().signOut( context: this)
        .addOnCompleteListener { it: Task<Void!>
            btn_sing_out.isEnabled = false
            tiposdelogin()
        }
        .addOnFailureListener{
            e -> Toast.makeText( context: this, e.message,Toast.LENGTH_SHORT).show()
        }
    }
}

} //fin del onCreate
```

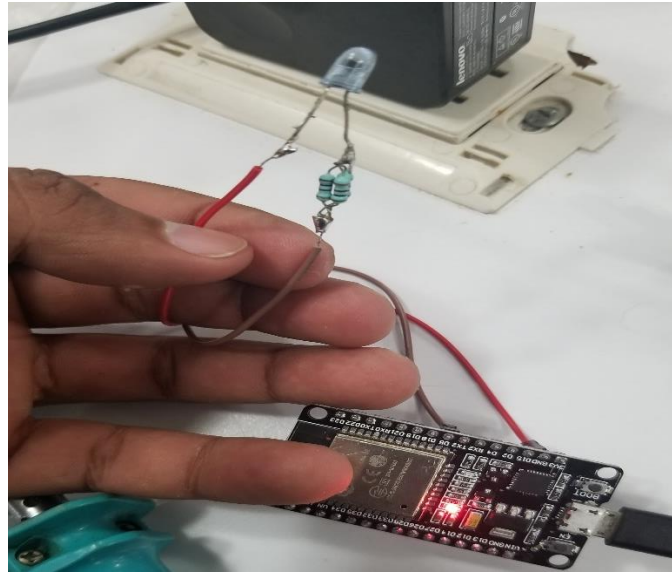
Figura 41 Función onCreate



## 3.4 Verificación

### 3.4.1 Conexión de hardware y software

Se realizó la conexión entre las tarjetas ESP32 y Arduino para comprobar si el LED infrarrojo manda los pulsos de una manera adecuada.



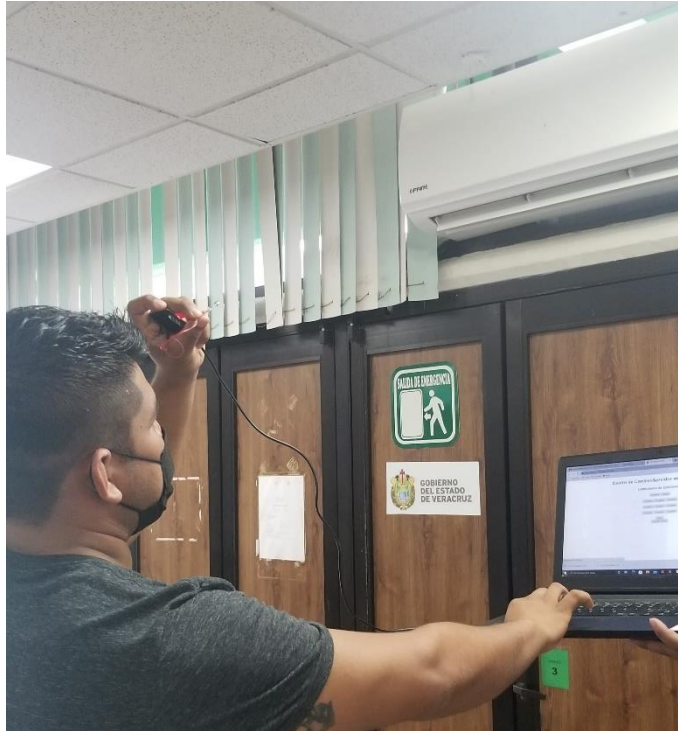
*Ilustración 42 ESP32*



*Ilustración 43 arduinoUNO*

### 3.4.2 Envío de pulsos con LED infrarrojo.

Se revisó que los pulsos se mandaran de una manera adecuada.



*Ilustración 44 pulsos*



*Ilustración 45 clima*

### 3.4.3 Inicio de sesión de la aplicación

Cuando se inicia la aplicación, esta permite que el usuario pueda iniciar sesión a través de su cuenta de correo de Google.

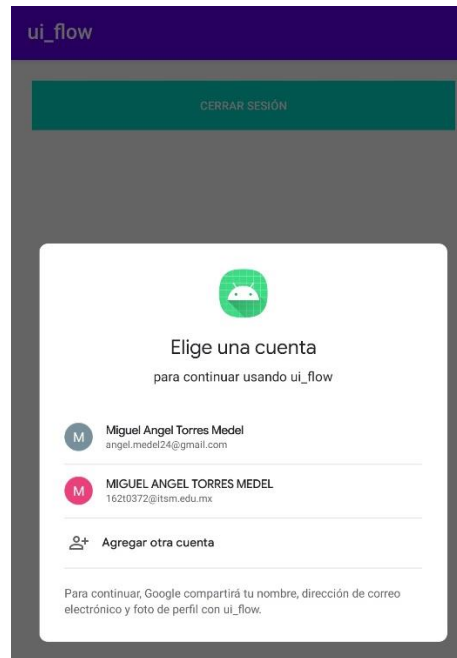


Figura 46 Inicio de sesión de aplicación

### 3.4.4 Autenticación con Firebase

Cada vez que un usuario agrega una dirección de correo, esta se va agregando a una lista en el servidor de Firebase, aquí se pueden ver todas las cuentas que han accedido a la aplicación.

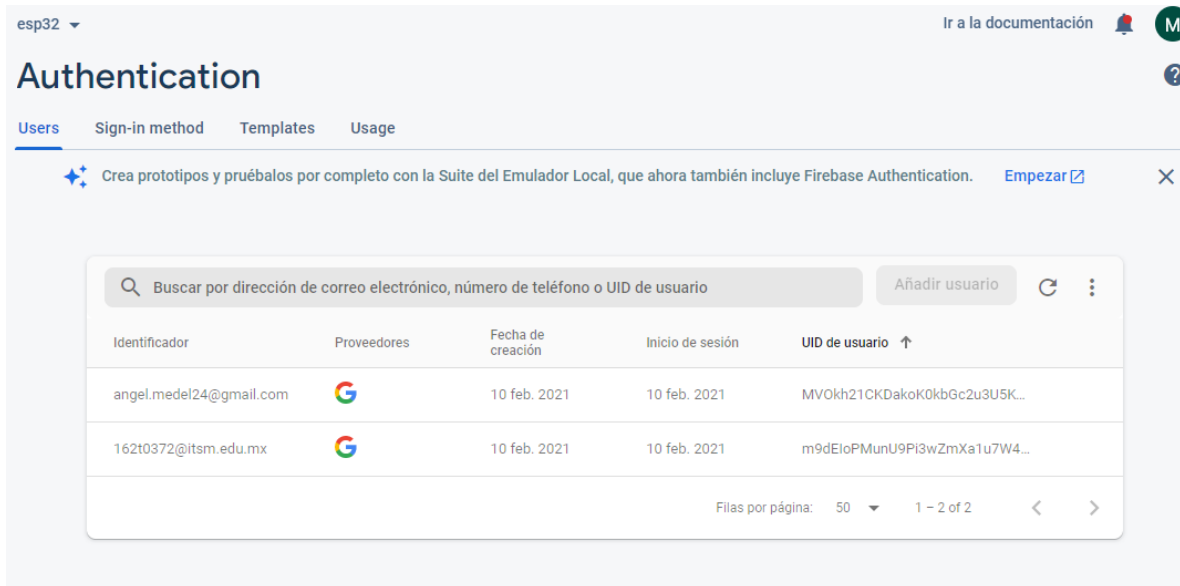


Figura 47 Lista de usuarios

Una vez que se entra a la aplicación muestra un menú de las salas disponibles del centro de cómputo.



*Figura 48 Menú de Salas*

Cuando se elige una sala en específico manda a otra pantalla donde muestra que se quiere prender si el videoprojector o el aire acondicionado.




Figura 49 Control de Aparatos


Para la aplicación web que se hizo desde Arduino se puede ingresar a ella gracias a una dirección ip que la ESP32 proporciona, esto es para usuarios que no cuenten con un sistema Android.



Figura 50 Aplicación Web


Para observar el comportamiento de los aparatos del centro de cómputo en Firebase se puede observar si el aparato este encendido que se representa con el numero 1 y apagado que se representa con el número 0.

 <https://tesiscomputo.firebaseio.com/>

**tesiscomputo**  
 **Electronica**  
└── **clima: 0**

*Ilustración 51 FireBase ON*

 <https://tesiscomputo.firebaseio.com/>

**tesiscomputo**  
 **Electronica**  
└── **clima: 1**

*Ilustración 52 Firebase OFF*

Una vez terminado el desarrollo con todos los procesos especificados anteriormente, se procede a instalar y verificar el funcionamiento del sistema, en la gráfica que se muestra a continuación se denota el funcionamiento del sistema después de 100 pruebas, en donde el valor de “100” es una prueba exitosa y el valor “0” significa que la prueba falló.

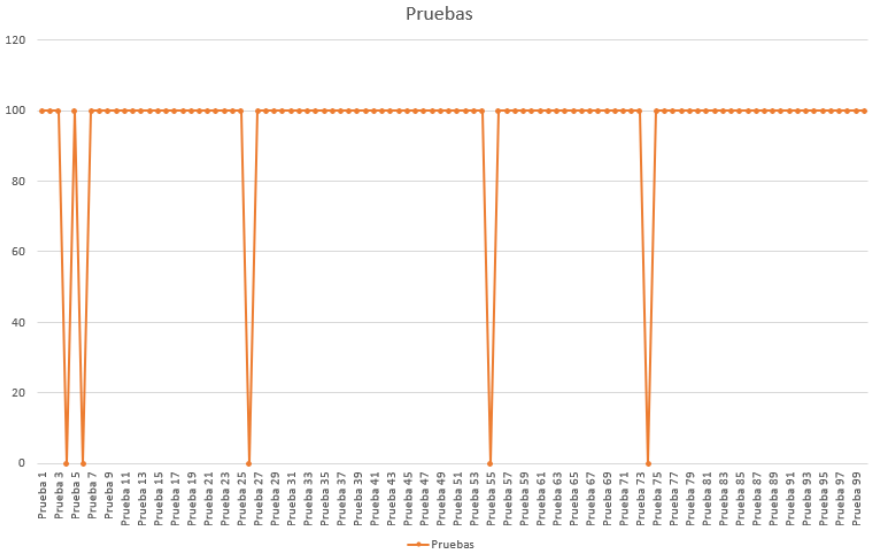


Ilustración 53 grafico



# CAPITULO IV

## RESULTADOS Y CONCLUSIONES

## 4.1 Resultados

De acuerdo los entornos de desarrollo antes mencionados que se usaron para crear la aplicación, AndroidStudio mostro ser un entorno que cumple con todo lo que se necesitó para el desarrollo, en cambio Arduino, presento ciertos problemas con las librerías, ya que no contaba con todas las necesarias, esto se solucionó descargando las librerías desde otras plataformas y así poder instalarlas en el IDE.

El smartphone que se utilizó en un inicio no pudo funcionar para el uso de la aplicación ya que este no contaba con un módulo infrarrojo, en ese caso se tuvo que instalar en un dispositivo que si tuviera modulo infrarrojo.

En la etapa del diseño, se pudieron obtener todos los códigos raw de los dispositivos electrónicos, las dos tarjetas que se ocuparon tuvieron un buen funcionamiento al igual que los sensores infrarrojos tuvieron un buen desempeño.

En la etapa de diseño Arduino fue eficiente para la creación de la página web que se pudo conectar con éxito en Firebase, para poder observar el estado de los aparatos y AndroidStudio también tuvo un buen desempeño y se pudo crear bien la app.

En la etapa de verificación se logró con éxito la conexión del hardware con el software y se realizaron 100 pruebas para verificar que los pulsos se mandaran de una manera adecuada, de estas 100 solo 5 fueron las que salieron mal y eso se debió a que había fallos en la red de internet y una latencia muy amplia.

## **4.2 Conclusiones**

Para concluir los resultados de la primera etapa, si se logró conseguir todo el software y hardware necesario que requería la creación de la página y la app.

Se concluye que los resultados de la etapa dos fueron satisfactorios, ya que los códigos raw obtenidos dieron buenos resultados a la hora de mandar los pulsos.

En el resultado de la etapa tres se concluye los entornos de desarrollo con los que se trabajaron y dieron buenos resultados en la creación de las aplicaciones, y de que también tuvieran un buen funcionamiento.

En el resultado de la etapa cuatro en conclusión tuvo buenos resultados ya que el hardware y software tuvieron buen funcionamiento a la hora de realizar las pruebas.

Por ultimo los resultados de la última etapa las aplicaciones tuvieron un buen funcionamiento al momento de utilizarse.

### **Conclusión final**

Con base en lo anterior se concluye que la hipótesis planteada de diseñar una plataforma computacional para la gestión energética de aires acondicionados, videoproyectores, ayuda a hacer un monitoreo del estado en el que se encuentran los aparatos electrónicos y así gestionar la energía eléctrica.

### 4.3 Referencias

- [1] Pujols, M. I. P. Control remoto infrarrojo táctil y ajustable, con interfaz de ajustes vía PC.
- [2] Allauca González, J. M. (2010). Control remoto con infrarrojos usando microcontroladores.
- [3] Aldean Pacalla, E. M., & García Ramos, J. L. (2019). Aplicación móvil y prototipo funcional en Arduino que permita el monitoreo y control remoto de los equipos de climatización e iluminación para hogares u oficinas (Bachelor's thesis, Universidad de Guayaquil Facultad de Ciencias Matemáticas y Físicas Carrera de Ingeniería en Sistemas Computacionales).
- [4] Nacho, R. (2016). Sistema de Control Domótico basado en Arduino, aplicación móvil y voz. La Paz: Universidad Mayor de San Andres, Facultad de Ciencias Puras y Naturales.
- [5] Mera, A., Carlos, J., & Yagual Castro, J. A. (2011). Diseño e implementación de un sistema de control de focos incandescentes en los hogares por medio de un control remoto universal (Bachelor's thesis).
- [6] Tamayo, D. A. L. (2019). Equipamiento para el control remoto y centralizado desde un teléfono móvil Android de los diferentes electrodomésticos existentes en los hogares.
- [7] Cárdenas Lancheros, E. A., & Castro Castro, O. J. Aplicación mediante la cual se realice el control y gestión de un sistema de iluminación led basado en la tecnología inalámbrica LoRa.
- [8] Colomer Campos, J. (2007). Método, dispositivo y sistema para la transmisión y recepción de señales de infrarrojos a través de una línea eléctrica.
- [9] Alarcón Vallejo, M. Á., Maroto Alpuente, P., & Pérez de Castro Martínez, S. (2011). Remotic: aplicación domótica por infrarrojos.
- [10] Pan Pérez-Vivancos, L. M. (2016). Dispositivo de efectos digitales para guitarra eléctrica controlado por Arduino y mando a distancia.

## 4.4 Anexos

```
#include <WiFi.h>
const char* ssid = "miguel123";
const char* password = "miguel123";

WiFiServer server(80);

#define LED2 5 // LED en terminal 2

String estado = "";

void setup() {
  Serial.begin(115200);
  pinMode(LED2, OUTPUT);

  // Conecta a la red wifi.
  Serial.println();
  Serial.print("Conectando con ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Conectado con WiFi.");
```

```

// Inicio del Servidor web.
server.begin();
Serial.println("Servidor web iniciado.");

// Esta es la IP
Serial.print("Esta es la IP para conectar: ");
Serial.print("http://");
Serial.println(WiFi.localIP());
}

void loop() {
// Consulta si se ha conectado algún cliente.
WiFiClient client = server.available();
if (!client) {
return;
}

Serial.print("Nuevo cliente: ");
Serial.println(client.remoteIP());

// Espera hasta que el cliente envíe datos.
while(!client.available()){ delay(1); }

////////////////////////////////////

// Lee la información enviada por el cliente.
String req = client.readStringUntil('\r');
Serial.println(req);

// Realiza la petición del cliente.

```

```

if (req.indexOf("on2") != -1)
    {
        send17();
    }
if (req.indexOf("off2") != -1){
    send18();
}
if (req.indexOf("consulta") != -1){
    if (digitalRead(LED2)){estado = "Encendido a 17";}
    else {estado = "Encendido a 18";}
}

////////////////////////////////////
// Página WEB. //////////////////////////////////
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // Importante.
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head><meta charset=utf-8></head>");
client.println("<body><center><font face='Arial'>");
client.println("<h1>Centro de Control-Servidor web con ESP32.</h1>");
client.println("<h2><font color='#009900'>Laboratorio de Electronica</font></h2>");
client.println("<a href='on'><button>Encender</button></a>");
client.println("<a href='off'><button>Apagar</button></a>");
client.println("<br><br>");
client.println("<a href='on2'><button>17 grados</button></a>");
client.println("<a href='off2'><button>18 grados</button></a>");
client.println("<a href='on3'><button>19 grados</button></a>");
client.println("<br><br>");

```

```

client.println("<a href='on4'><button>20 grados</button></a>");
client.println("<a href='on5'><button>21 grados</button></a>");
client.println("<a href='on6'><button>22 grados</button></a>");
client.println("<br><br>");
client.println("<a href='on7'><button>23 grados</button></a>");
client.println("<a href='on8'><button>24 grados</button></a>");
client.println("<a href='on9'><button>25 grados</button></a>");
client.println("<br><br>");
client.println("<a href='consulta'><button> Turbo </button></a><br>");
client.println("<a href='consulta'><button> Consulta estado </button></a><br>");
client.println("<br><br>");
if(estado == "Encendido") {
    client.print("<a href='off2'><button>17 grados </button></a><br><br>");
    client.println("<img src='http://iesromerovargas.com/lampon.png'><br>");
} else {
//client.print("<a href='on2'><button>18 grados</button></a><br><br>");

}
client.println("</font></center></body></html>");

Serial.print("Cliente desconectado: ");
Serial.println(client.remoteIP());
client.flush();
client.stop();
}

//*****
*****

```



```

//Método que envia los pulsos correspondientes a la cadena de bits del infrarrojo
void pulseIR(long microsecs) {
    cli(); //Interrupcion

    while (microsecs > 0) {
        //Pulsos a 38 Khz
        digitalWrite(LED2, HIGH); // this takes about 3 microseconds
to happen
        delayMicroseconds(10); // hang out for 10 microseconds
        digitalWrite(LED2, LOW); // this also takes about 3
microseconds
        delayMicroseconds(10); // hang out for 10 microseconds

        microsecs -= 26;
    }

    sei(); //Sale de la interrupcion
}

```

```

//*****
*****

```

```

void sendOn(){
pulseIR(5960);
delayMicroseconds(7260);
pulseIR(580);
delayMicroseconds(1620);
pulseIR(480);
delayMicroseconds(1660);
pulseIR(560);
delayMicroseconds(1620);

```

pulseIR(460);  
delayMicroseconds(1660);  
pulseIR(560);  
delayMicroseconds(1560);  
pulseIR(520);  
delayMicroseconds(1680);  
pulseIR(540);  
delayMicroseconds(1580);  
pulseIR(500);  
delayMicroseconds(1680);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1560);  
pulseIR(580);  
delayMicroseconds(1620);

```
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(1620);
```

```
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1620);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(1560);  
pulseIR(520);  
delayMicroseconds(640);  
pulseIR(460);  
delayMicroseconds(640);  
pulseIR(480);  
delayMicroseconds(620);  
pulseIR(460);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(640);  
pulseIR(460);  
delayMicroseconds(640);  
pulseIR(460);  
delayMicroseconds(640);  
pulseIR(460);  
delayMicroseconds(640);  
pulseIR(460);  
delayMicroseconds(640);
```

pulseIR(460);  
delayMicroseconds(1680);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(480);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(1680);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(480);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(1660);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(460);  
delayMicroseconds(640);  
pulseIR(460);  
delayMicroseconds(1680);  
pulseIR(540);  
delayMicroseconds(1580);

```
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(1660);  
pulseIR(560);  
delayMicroseconds(1560);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1680);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(460);  
delayMicroseconds(1680);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1560);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1680);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);
```

```
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1560);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1680);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1560);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1700);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1560);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1680);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1560);
```

```
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1680);  
pulseIR(540);  
delayMicroseconds(1560);  
pulseIR(520);  
delayMicroseconds(7300);  
pulseIR(520);  
}
```

```
void sendTurbo(){  
pulseIR(6040);  
delayMicroseconds(7220);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);
```



pulseIR(540);  
delayMicroseconds(580);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(540);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1580);

```
pulseR(540);  
delayMicroseconds(1600);  
pulseR(540);  
delayMicroseconds(580);  
pulseR(520);  
delayMicroseconds(600);  
pulseR(520);  
delayMicroseconds(560);  
pulseR(540);  
delayMicroseconds(1600);  
pulseR(560);  
delayMicroseconds(540);  
pulseR(580);  
delayMicroseconds(520);  
pulseR(580);  
delayMicroseconds(540);  
pulseR(560);  
delayMicroseconds(540);  
pulseR(560);  
delayMicroseconds(1600);  
pulseR(520);  
delayMicroseconds(640);  
pulseR(460);  
delayMicroseconds(1620);  
pulseR(580);  
delayMicroseconds(520);  
pulseR(580);  
delayMicroseconds(1600);  
pulseR(500);  
delayMicroseconds(1640);
```

```
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1620);
```

pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);

pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(540);

```
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(7300);  
pulseIR(520);  
}
```

```
void sendOFF(){
pulseIR(6000);
delayMicroseconds(7240);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(580);
delayMicroseconds(1600);
pulseIR(500);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(1620);
pulseIR(580);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
pulseIR(540);
delayMicroseconds(580);
```

```
pulseIR(520);
delayMicroseconds(580);
pulseIR(520);
delayMicroseconds(600);
pulseIR(500);
delayMicroseconds(600);
pulseIR(500);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1580);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(1640);
pulseIR(540);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
pulseIR(580);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(540);
```



```
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(520);
delayMicroseconds(1640);
pulseIR(540);
delayMicroseconds(1580);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(580);
delayMicroseconds(1580);
pulseIR(580);
delayMicroseconds(1560);
pulseIR(580);
delayMicroseconds(1620);
pulseIR(540);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
```

pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);

```
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(540);
```

pulseIR(560);  
delayMicroseconds(1560);  
pulseIR(540);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);

```
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(7240);  
pulseIR(560);  
}
```

```
void send16(){  
pulseIR(6020);  
delayMicroseconds(7220);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1560);
```

pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(600);  
pulseIR(480);  
delayMicroseconds(620);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1600);

pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1560);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);

```
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(500);  
delayMicroseconds(600);
```



```
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);
```

```
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1600);
```

```
pulseIR(560);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(560);
pulseIR(520);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(580);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(1600);
pulseIR(520);
delayMicroseconds(580);
pulseIR(500);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(560);
pulseIR(560);
delayMicroseconds(1560);
pulseIR(540);
delayMicroseconds(600);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(1600);
```

```
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(7240);  
pulseIR(560);  
}
```

```
void send17(){  
pulseIR(6060);  
delayMicroseconds(7200);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1580);
```

```
pulseIR(520);  
delayMicroseconds(620);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);
```

pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(1640);

pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(500);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);

pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);



pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(600);

```
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(7260);  
pulseIR(580);  
}
```

```
void send18(){  
  pulseIR(6020);  
  delayMicroseconds(7200);  
  pulseIR(580);  
  delayMicroseconds(1600);  
  pulseIR(540);  
  delayMicroseconds(1600);  
  pulseIR(560);  
  delayMicroseconds(1600);  
  pulseIR(520);  
  delayMicroseconds(1620);  
  pulseIR(580);  
  delayMicroseconds(1620);  
  pulseIR(480);  
  delayMicroseconds(1620);  
  pulseIR(580);  
  delayMicroseconds(1600);  
  pulseIR(500);  
  delayMicroseconds(1640);  
  pulseIR(560);  
  delayMicroseconds(560);  
  pulseIR(540);  
  delayMicroseconds(540);  
  pulseIR(560);  
  delayMicroseconds(540);  
  pulseIR(580);  
  delayMicroseconds(560);  
  pulseIR(520);  
  delayMicroseconds(560);
```

```
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);
```

```
pulseIR(540);
delayMicroseconds(580);
pulseIR(540);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
pulseIR(580);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(1620);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(520);
delayMicroseconds(1620);
pulseIR(580);
delayMicroseconds(540);
```

```
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1620);
```

```
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(580);  
delayMicroseconds(540);
```

```
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);
```



```
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(7280);  
pulseIR(540);  
}
```

```
void send19(){  
pulseIR(6000);  
delayMicroseconds(7240);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);
```

```
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);
```

pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(560);

```
pulseR(540);
delayMicroseconds(540);
pulseR(560);
delayMicroseconds(1600);
pulseR(540);
delayMicroseconds(1600);
pulseR(560);
delayMicroseconds(1600);
pulseR(540);
delayMicroseconds(1600);
pulseR(580);
delayMicroseconds(1580);
pulseR(540);
delayMicroseconds(1600);
pulseR(580);
delayMicroseconds(1600);
pulseR(540);
delayMicroseconds(1600);
pulseR(560);
delayMicroseconds(540);
pulseR(560);
delayMicroseconds(560);
pulseR(540);
delayMicroseconds(540);
pulseR(540);
delayMicroseconds(560);
pulseR(580);
delayMicroseconds(540);
pulseR(540);
delayMicroseconds(560);
```

```
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);
```

```
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1660);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);
```

```
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1620);
```

```
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(7220);  
pulseIR(580);  
}
```

```
void send20(){  
pulseIR(6040);  
delayMicroseconds(7220);  
pulseIR(500);  
delayMicroseconds(1660);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
}
```



```
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1600);
```

```
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(1620);
```

pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);

pulseIR(520);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1620);

```
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);
```

```
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(540);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(7260);  
pulseIR(560);  
}
```

```
void send21(){
pulseIR(6040);
delayMicroseconds(7220);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(520);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(1620);
pulseIR(580);
delayMicroseconds(1600);
pulseIR(500);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(500);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(560);
pulseIR(560);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(540);
```

```
pulseIR(560);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(540);
pulseIR(540);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(580);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1580);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1580);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(540);
pulseIR(580);
delayMicroseconds(520);
pulseIR(560);
delayMicroseconds(560);
pulseIR(560);
delayMicroseconds(540);
```



```
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);
```

```
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1600);
```

```
pulseIR(560);
delayMicroseconds(540);
pulseIR(580);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(560);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(580);
pulseIR(520);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(540);
pulseIR(580);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(580);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(1600);
```

```
pulseIR(540);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(580);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(520);
delayMicroseconds(600);
pulseIR(520);
delayMicroseconds(560);
pulseIR(520);
delayMicroseconds(580);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(580);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(540);
```

```
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(7280);  
pulseIR(560);  
}
```

```
void send22(){  
pulseIR(6040);  
delayMicroseconds(7220);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);
```

pulseIR(580);  
delayMicroseconds(1620);  
pulseIR(480);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(620);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);

```
pulseIR(520);  
delayMicroseconds(1660);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(480);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(480);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(620);  
pulseIR(480);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);
```

```
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(500);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1620);  
pulseIR(480);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(620);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(620);
```



pulseIR(480);  
delayMicroseconds(1660);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1580);

```
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);
```

```
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);
```

```
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(7280);  
pulseIR(520);  
}
```

```
void send23(){  
pulseIR(6020);  
delayMicroseconds(7220);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1600);
```

```
pulseIR(520);
delayMicroseconds(600);
pulseIR(500);
delayMicroseconds(580);
pulseIR(540);
delayMicroseconds(560);
pulseIR(540);
delayMicroseconds(600);
pulseIR(480);
delayMicroseconds(600);
pulseIR(500);
delayMicroseconds(600);
pulseIR(520);
delayMicroseconds(580);
pulseIR(520);
delayMicroseconds(580);
pulseIR(520);
delayMicroseconds(1620);
pulseIR(580);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(500);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(1620);
```

pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1580);

```
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(1600);
```

pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1620);



```
pulseIR(580);
delayMicroseconds(1600);
pulseIR(500);
delayMicroseconds(620);
pulseIR(480);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(520);
delayMicroseconds(1620);
pulseIR(560);
delayMicroseconds(1580);
pulseIR(540);
delayMicroseconds(1620);
pulseIR(580);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(520);
pulseIR(580);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(580);
pulseIR(560);
delayMicroseconds(560);
pulseIR(560);
delayMicroseconds(540);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(580);
delayMicroseconds(540);
```

```
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(7240);  
pulseIR(580);  
}
```

```
void send24(){
pulseIR(6000);
delayMicroseconds(7240);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(520);
delayMicroseconds(1620);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(520);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1580);
pulseIR(520);
delayMicroseconds(1640);
pulseIR(560);
delayMicroseconds(1600);
pulseIR(540);
delayMicroseconds(1600);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(580);
pulseIR(500);
delayMicroseconds(580);
pulseIR(560);
delayMicroseconds(540);
pulseIR(560);
delayMicroseconds(560);
```

```
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(480);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(580);
```

```
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(540);
```

```
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);
```

```
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);
```

```
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);
```



```
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(7240);  
pulseIR(580);  
}
```

```
void send25(){  
pulseIR(5980);  
delayMicroseconds(7220);  
pulseIR(540);  
delayMicroseconds(1620);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(1600);
```

```
pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);
```

```
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(580);  
delayMicroseconds(1560);  
pulseIR(580);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(580);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(540);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(600);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(1600);
```

```
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(1660);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(560);  
delayMicroseconds(1620);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(540);
```

```
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(520);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(580);  
pulseIR(520);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(540);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1600);
```

pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1620);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(500);  
delayMicroseconds(1640);  
pulseIR(560);  
delayMicroseconds(1600);  
pulseIR(520);  
delayMicroseconds(1640);  
pulseIR(540);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(1600);  
pulseIR(500);  
delayMicroseconds(600);  
pulseIR(540);  
delayMicroseconds(1600);

```
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(520);  
pulseIR(580);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1600);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(580);  
delayMicroseconds(560);  
pulseIR(540);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(560);  
pulseIR(560);  
delayMicroseconds(1580);  
pulseIR(560);  
delayMicroseconds(540);  
pulseIR(560);  
delayMicroseconds(1600);
```

```
pulseR(540);  
delayMicroseconds(580);  
pulseR(540);  
delayMicroseconds(1580);  
pulseR(580);  
delayMicroseconds(1580);  
pulseR(560);  
delayMicroseconds(7220);  
pulseR(540);  
}
```