**TECNOLÓGICO NACIONAL DE MEXICO**

**INSTITUTO TECNOLÓGICO DE LA LAGUNA**

****

**REPORTE DE PRACTICA**

**UNIDAD 4: Microcontroladores PRÁCTICA 15: Bluetooth**

**DOCENTE: LAMIA HAMDAN M.**

| **NUM DE CONTROL** | **NOMBRE** |
| --- | --- |
| 19130514 | Isaias Gerardo Cordova Palomares |
| 19130545 | Oscar Martinez Ruiz |
| 17130763 | Raúl Martín Ayala Salais |
| 19130541 | Pedro Lopez Ramirez |
| 19130535 | Ivan Herrera Garcia |
| 18131263 | Gerardo Alberto Orozco Villegas |

**FECHA DE ENTREGA:** 6/11/2022

**TABLA DE CONTENIDO**

[**1.INTRODUCCIÓN**](#_555xbbp1n8y5) **3**

[**2. COMPETENCIA A DESARROLLAR**](#_30j0zll) **3**

[**3. CIRCUITO LÓGICO Y/O PROGRAMA**](#_vyaxf5uqhjrg) **4**

[**4. METODOLOGÍA**](#_uikunnlbl3c) **4**

[**5. RESULTADOS**](#_4j5zl4o09ir7) **7**

[**6.CONCLUSIONES Y RECOMENDACIONES**](#_swudbzgg3llj) **9**

[**7. REFERENCIAS**](#_s8u6prgrx9vv) **9**

# **1.INTRODUCCIÓN**

Conocer y aplicar los conocimientos de comunicaciones conectando el dispositivo de bluetooth a la computadora utilizando Arduino.

Para ello pretendemos utilizar un componente llamado ESP32 en su placa Dev.

[1] El ESP32 es un microcontrolador de tipo “low-cost & low-power” el cual viene equipado con Bluetooth y Wifi. Es sucesor del ESP8266, el cual también viene con Wifi integrado, pero más sin embargo viene más limitado de funcionalidades que el ESP 32..

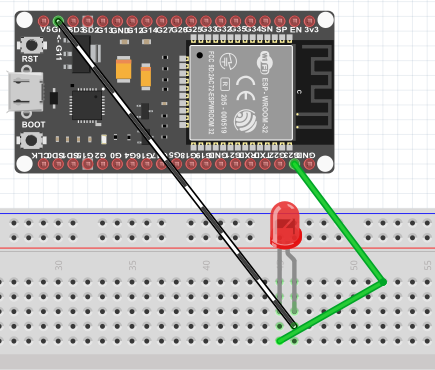
Entrando en detalle más técnicos, el ESP 32 cuenta con una antena integrada, amplificador de bajo ruido,filtros y módulo de administración de energía. Esta placa se utiliza con chips Bluetooth y Wifi de modo dual de 2.4 GHz y TSMC con tecnología de baja potencia de 40 nm, las mejores propiedades de potencia y RF, que es seguridad, confiable y escalable para una variedad de aplicaciones.

[1]

# **2. COMPETENCIA A DESARROLLAR**

* Identifica las características eléctricas de un microcontrolador.
* Conoce la arquitectura interna del microcontrolador.
* Comprende la estructura de registros del microcontrolador.
* Analiza dispositivos de entrada/salida y puertos del microcontrolador.
* Organiza y clasifica información proveniente de fuentes diversas.

# **3. CIRCUITO LÓGICO Y/O PROGRAMA**



*Montaje del circuito en think cad/fritzing*

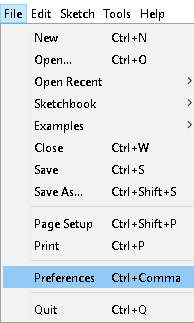
# **4. METODOLOGÍA**

**Material necesario:**

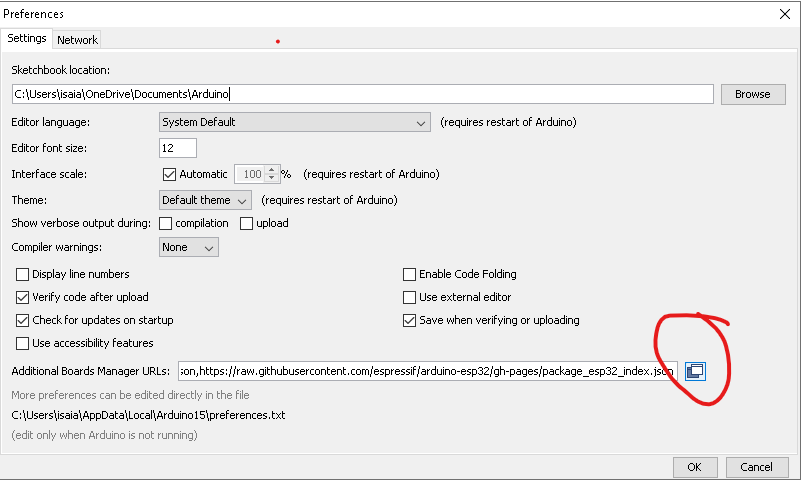
* Cables
* IDE Arduino
* ESP32 Dev
* Led Rojo
* Celular

Al ser la primera vez que manejamos este componente, tuvimos que hacer algunos ajustes para que nuestro dispositivo ESP32 fuera detectado por el ordenador e IDE.

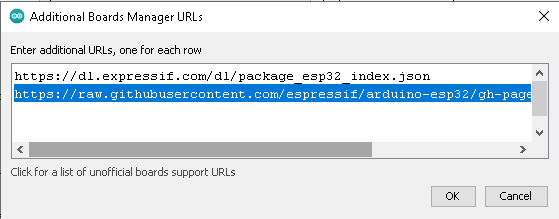
1. **Ir al apartado de preferencias en : Archivo > Preferencias**

****

1. **Dar click en el icono de la parte derecha en donde dice “Placas adicionales”**

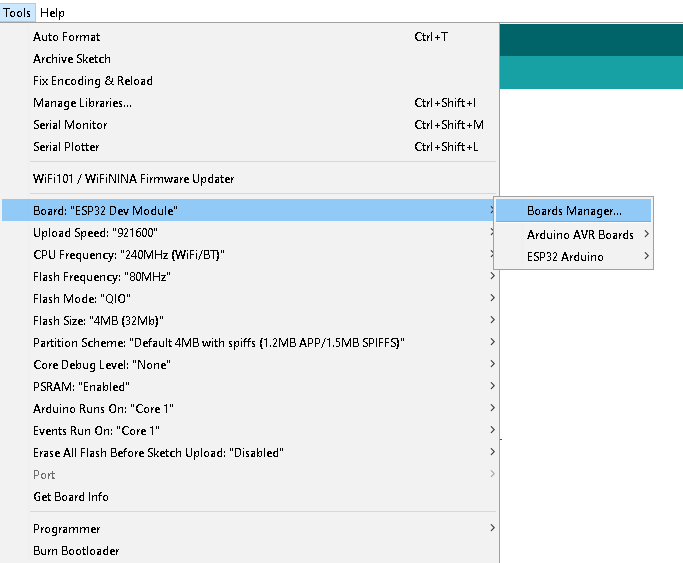
****

1. **Añadimos los siguientes repositorios**

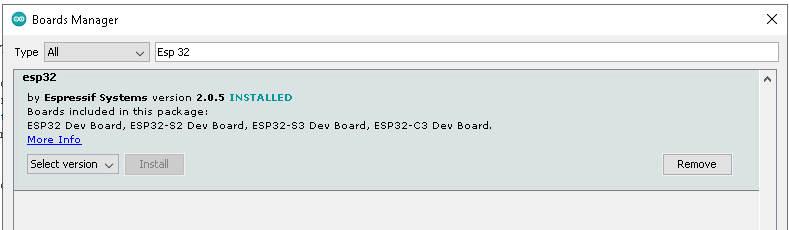
****

Se tuvo problemas con el primer repositorio, asi que principalmente el que nos ayudo fue el segundo repositorio : <https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json>

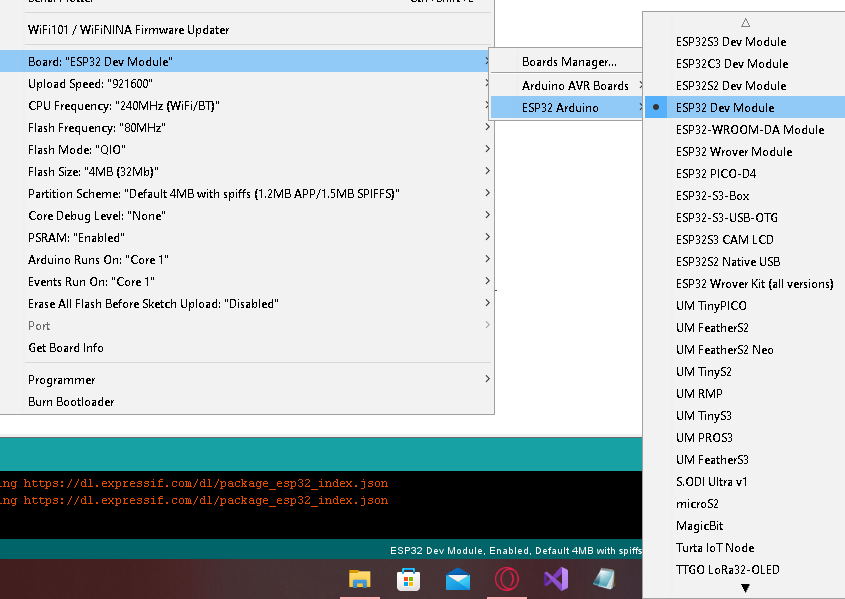
1. **Ir al administrador de placas en : Herramientas > Placa > Administrador de Placas**

****

1. **Buscar “ESP 32” y darle en instalar**

****

1. **Con esto ya tendremos instalado los drivers necesarios para que podamos trabajar con el ESP 32 en el IDE de Arduino.**

****

# **5. RESULTADOS**

Al poder por fin lograr una conexión bluetooth entre un dispositivo y el ESP32, el resultado que se logró es que nosotros al mandarle un ‘0’ el led se apaga y al mandar un ‘1’ el led se enciende.



*Led apagado al mandarle un ‘0’ al esp32*

**

*Led Encendido al haberle mandado un ‘1’ al esp32*

**Codigo**

/\* SISTEMAS PROGRAMABLES AGOSTO-DICIEMBRE 2022

\* UNIDAD 4 : Practica 15

Programa para el control del ENCENDIDO/APAGADO de un LED usando el ESP32

\*/

#include "BluetoothSerial.h" // Incluimos la libreroa

// Verifica que la conexion sea exitosa

#if !defined(CONFIG\_BT\_ENABLED) || !defined(CONFIG\_BLUEDROID\_ENABLED)

#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it

#endif

#define LED 23

BluetoothSerial BT; // Objeto de Tipo SETUP SERIAL

void setup() {

Serial.begin(9600); // Inicialización de la conexión en serie para la depuración

BT.begin("ESP32\_LED\_Control"); // Nombre de su dispositivo Bluetooth.

Serial.println("El dispositivo Bluetooth está listo para emparejarse");

pinMode (LED, OUTPUT); // Cambie el pin LED a OUTPUT

}

void loop() {

if (BT.available()) // Compruebe si recibimos algo de Bluetooth

{

int valor = BT.read(); // Lee lo que recibimos

Serial.print("Recibido: ");

Serial.println(valor);

// Para practica 15 y 16 solo cambian los ascii

/\*VALORES EN ASCII DEL :

0 : 48

1 : 49\*/

if (valor == 49){ // 1 en ASCII

digitalWrite(LED, HIGH); // LED Encendido

BT.println("LED encendido"); // Envía el mensaje de texto a través de BT Serial

}

if (valor == 48){ // 0 en ASCII

digitalWrite(LED, LOW); // LED Apagado

BT.println("LED apagado"); // Envía el mensaje de texto a través de BT Serial

}

}

delay(20); // espera 0.02 segundos.

}

# **6.CONCLUSIONES Y RECOMENDACIONES**

En general fue una práctica un tanto compleja, por diversos motivos, pero principalmente es el hecho de que ningún integrante del equipo había manejado antes un dispositivo de estos con anterioridad. Por lo que tuvimos que investigar a fondo el cómo funciona este componente, afortunadamente no tardamos mucho en dar con la solución a este problema de vinculación, por lo que después de haber sabido el cómo vincular el dispositivo se completó la práctica con normalidad. Asi mismo, el usar por primera vez el componente ESP32 nos trajo problemas ya que ningún integrante del equipo lograba una conexión entre su equipo (Laptop) con el dicho componente (Puertos COM). Al final se soluciono investigando diversos programas para lograr la conexión.

# 

# **7. REFERENCIAS**

(no date) *Workshop studio*. Available at: https://catalog.us-east-1.prod.workshops.aws/workshops/5b127b2f-f879-48b9-9dd0-35aff98c7bbc/en-US/module1/esp32 (Accessed: October 30, 2022).

Espressif (2022) *Arduino-esp32/bluetoothserial.h at master · espressif/arduino-ESP32*, *GitHub*. Available at: https://github.com/espressif/arduino-esp32/blob/master/libraries/BluetoothSerial/src/BluetoothSerial.h (Accessed: October 30, 2022).

Capps, D. *et al.* (2019) *Esp32 bluetooth classic with Arduino IDE - Getting Started*, *Random Nerd Tutorials*. Available at: https://randomnerdtutorials.com/esp32-bluetooth-classic-arduino-ide/#:~:text=To%20connect%20to%20the%20ESP32,Bluetooth%20devices%2C%20including%20the%20ESP32test (Accessed: October 30, 2022).

*Electrosoftcloud* (no date) *ElectroSoftCloud*. Available at: https://www.electrosoftcloud.com/ (Accessed: October 30, 2022).

(no date) *Cátedra Digital - Agosto Diciembre 2021*. Available at: https://catedig2.itlalaguna.edu.mx/ (Accessed: October 30, 2022).