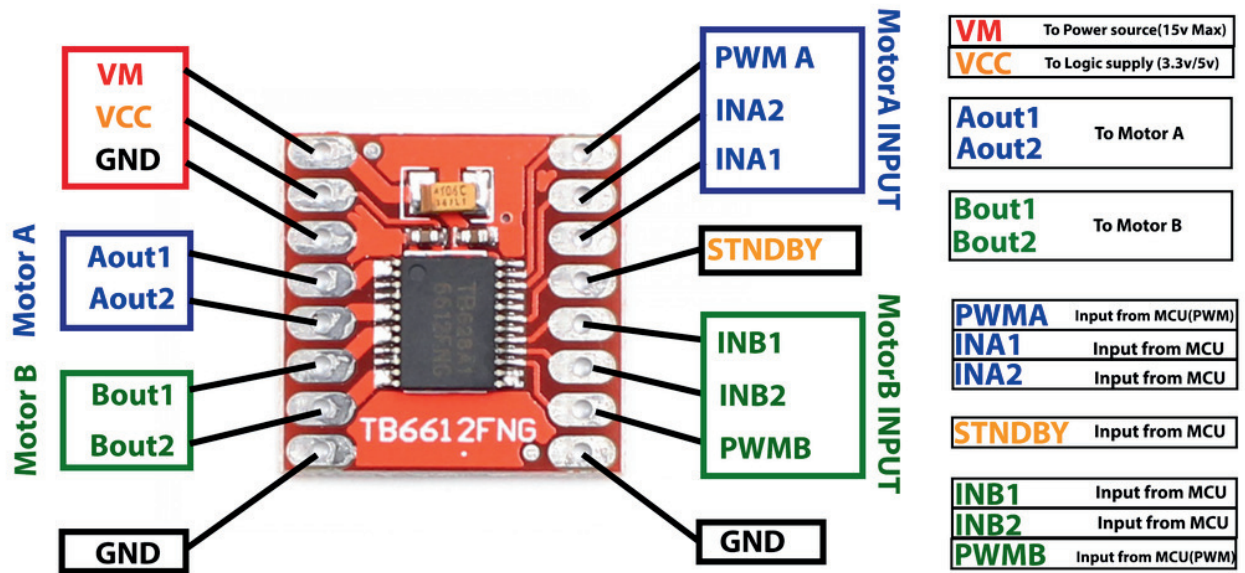
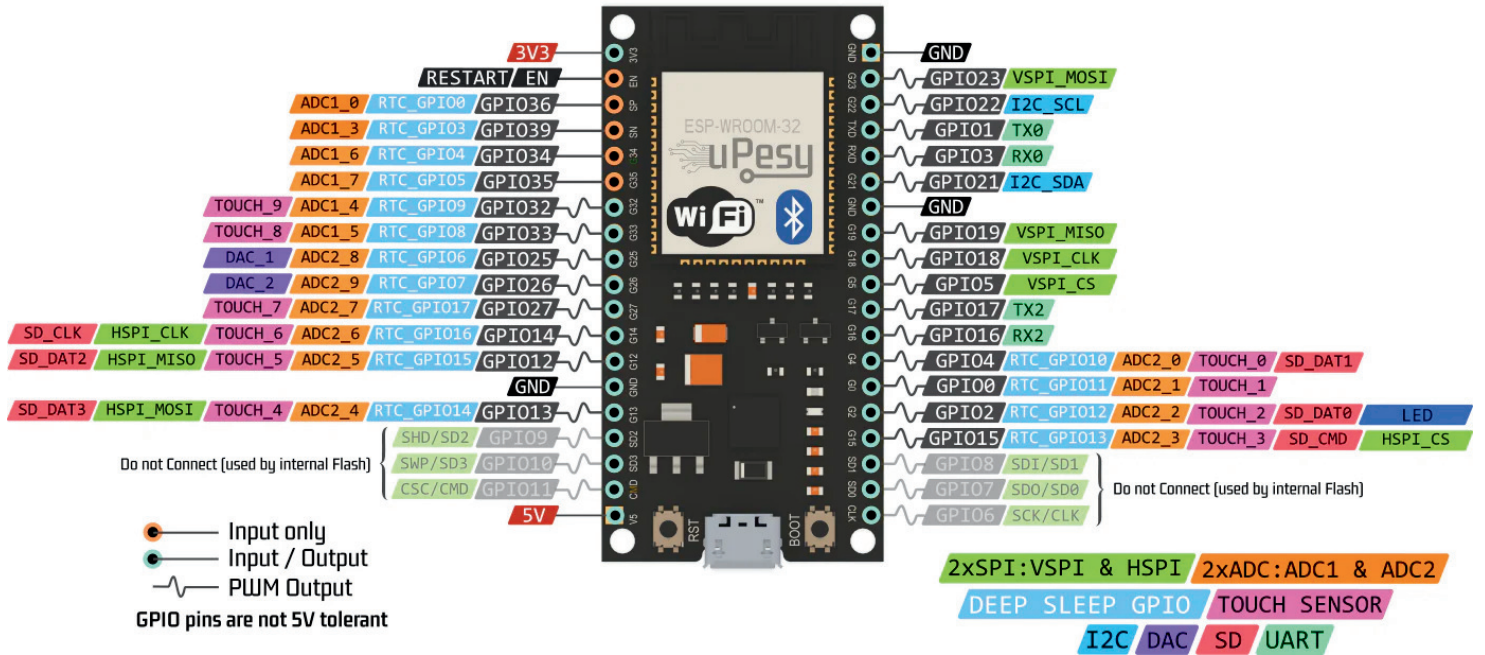


# ESP32 Wroom DevKit Full Pinout



ArtistTech

## Paso 4: Flujo de Trabajo Diario - Subir Cambios a GitHub

Este es el ciclo que seguirás cada vez que hagas cambios en tu código y quieras guardarlos en GitHub.

```
# 1. Revisa el estado de tus archivos
# Te mostrará qué archivos han sido modificados, creados o eliminados
git status

# 2. Añade los archivos que quieres guardar al "área de preparación" (staging)
# Para añadir todos los archivos nuevos o modificados en la carpeta actual:
git add .

# Si solo quieres añadir un archivo específico:
# git add nombre_del_archivo.ino

# 3. Guarda los cambios en tu repositorio local (commit)
# El mensaje debe ser descriptivo de los cambios que hiciste
git commit -m "Mi primer commit: Añadido código inicial del LED"

# 4. Sube los cambios guardados a tu repositorio en GitHub
# La primera vez que subes cambios, puede que necesites usar la opción -u
git push -u origin main

# Para las siguientes actualizaciones, solo necesitarás:
git push
```

¡Listo! Si ahora vas a tu repositorio en GitHub, verás todos tus archivos de código allí.

## Paso 5: Flujo de Trabajo - Bajar Cambios desde GitHub

Si trabajas en equipo o haces cambios desde otra computadora (o desde la propia web de GitHub), necesitarás actualizar tu carpeta local con los cambios del repositorio remoto.

```
# 1. Navega a la carpeta de tu proyecto
cd "C:/Users/TuUsuario/Documents/Arduino/MiProyectoLED"

# 2. Descarga los cambios del repositorio remoto y fúndalos con tu versión local
git pull origin main
```

## Resumen de Comandos Esenciales

- `git status` : ¿Qué ha cambiado?
- `git add .` : Prepara todos los cambios para guardarlos.
- `git commit -m "Mensaje"` : Guarda los cambios localmente.
- `git push` : Sube los cambios a GitHub.
- `git pull` : Baja los cambios desde GitHub.