

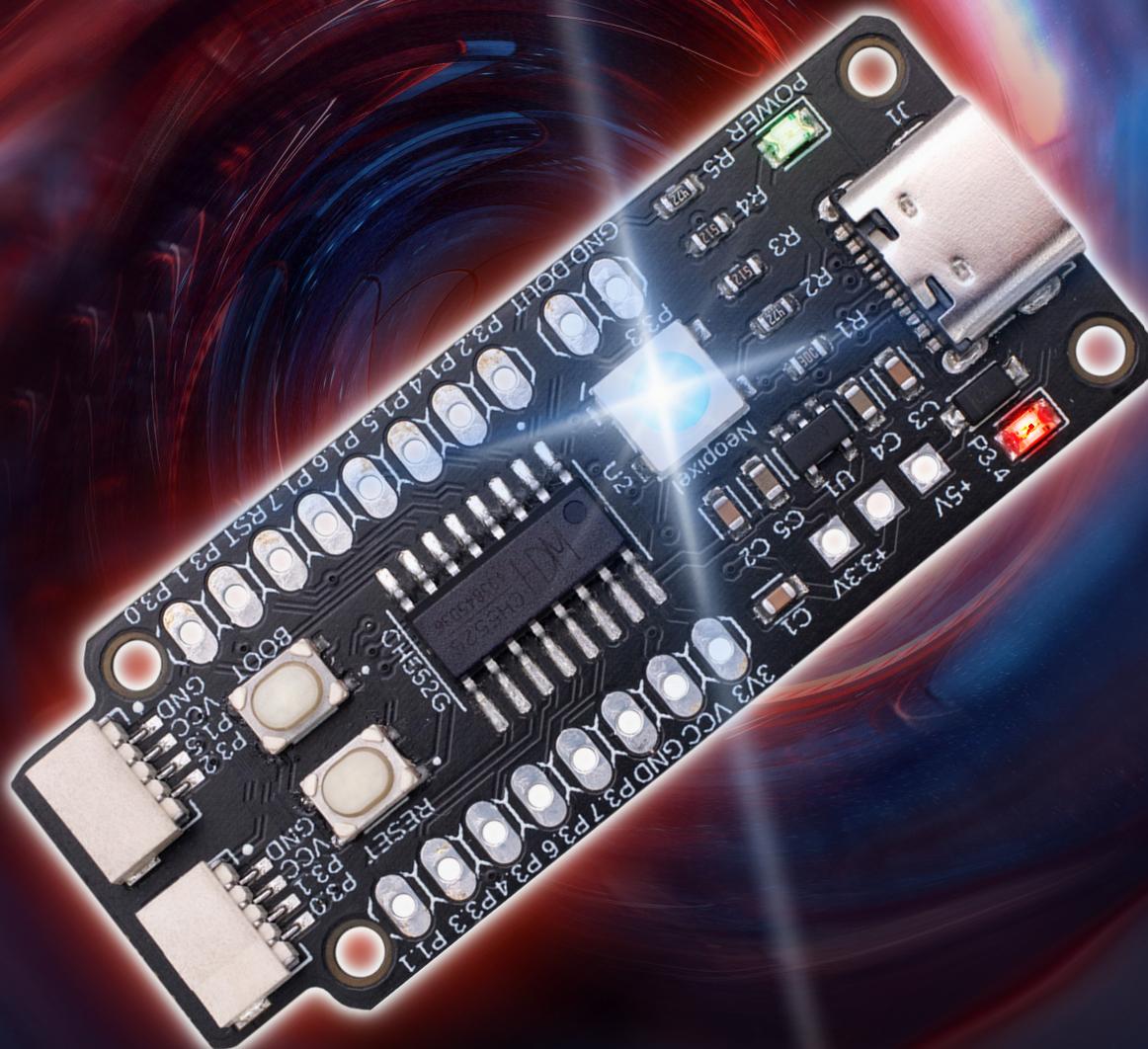
MANUAL DE USUARIO

Versión 1.2 Fecha de actualización: 12/04/2023

COCKET NOVA CH552G

DEVELOPMENT BOARD

La innovación comienza con 8 bits.



Descripción

La Cocket Nova Development Board está basada en el microcontrolador CH552G, el cuál se caracteriza por tener un núcleo mejorado e8051 y ser compatible con el set de instrucciones MCS51, siendo el 79% de sus instrucciones ciclos de un solo byte. Es capaz de trabajar a 24 MHz e incluye una memoria ROM interna de 16 Kb y 256 bytes de RAM.

El microcontrolador incluye internamente un convertidor analógico-digital (ADC), botón táctil detector de capacitancia, tres sistemas de timers y capturadores de señal, PWM, puerto dual serial asíncrono SPI, controlador de dispositivo USB y un transceptor de velocidad completa.

La tarjeta de desarrollo permite programar el CH552G a través de un puerto USB tipo C y usar todos los pines del microcontrolador con la facilidad de conectar la tarjeta a una protoboard, también incluye un LED Neopixel conectado al pin 3.3 junto con 2 headers para conectar externamente más LEDs neopixel, un LED Built-in conectado al pin 3.4, botón de reinicio y de boot, un grupo de pines para seleccionar el voltaje de alimentación del MCU y dos conectores JST de 1 mm compatible con protocolos QWIC, STEMMA QT y similares.

Características

El microcontrolador CH552G posee un núcleo e8051 compatible con el set de instrucciones MCS51, siendo este núcleo de 8 a 15 veces más veloz que el estándar MC51.

✦ MEMORIA

- 16 KB de memoria ROM, puede dividirse en 14 KB para programa y 2KB para el bootloader o programación ISP.
- 128 bytes para memoria flash .
- 256 bytes de memoria RAM interna.

✦ PERIFÉRICOS

- 2 salidas PWM de 8 bits.
- 4 canales de 8 bits para ADC.
- 6 canales para detección de capacitancia, soporta hasta 15 botones táctiles.
- 17 Pines GPIO.
- 14 grupos de interrupciones y un watch dog configurable en 8 bits.
- Botón de reset externo y de boot para programación.
- Señal de reloj externa a través de una multiplexación de los GPIO

✦ USB

- Controlador y transceptor interno USB, permite la programación vía USB sin necesidad de controladores externos, también soporta USB 2.0 a una velocidad máxima de 12 mbps.

✦ ENERGÍA

- Voltaje de alimentación de 5V o 3.3V.
- Voltaje de alimentación mínimo de 2.8V y máximo de 3.5V.

✦ COMUNICACIÓN

- 2 grupos de UART, el grupo UART 0 es el puerto serial estándar.
- SPI, contiene un FIFO interno y es compatible con modos esclavo y maestro.
- I2C simulable en los GPIO (SoftI2C).
- 2 Conectores JST de 4 pines: compatibles con protocolos tipo QWIIC, STEMMA QT o similares, se puede alimentar el microcontrolador a través de este conector o dar energía a otros módulos, uno de los conectores ofrece comunicación con los GPIOs 3.0 y 3.1 mientras que el otro tiene conexiones a los GPIOs 1.5 y 3.2.

- Selector de voltaje de alimentación a través de un jumper switch para que el microcontrolador trabaje con 5V o 3.3V.
- LED RGB Neopixel conectado al GPIO 3.3 y a una tira header que funciona para conectar más LEDs Neopixel a la salida de datos.
- LED Built-in conectado al pin 3.4.

Requisitos de la tarjeta

- ✦ Cable USB tipo C de datos.
- ✦ Arduino IDE u otro software de compilación compatible con el CH552G.
- ✦ Software Zadig. 
- ✦ Controlador CH372DRV. 



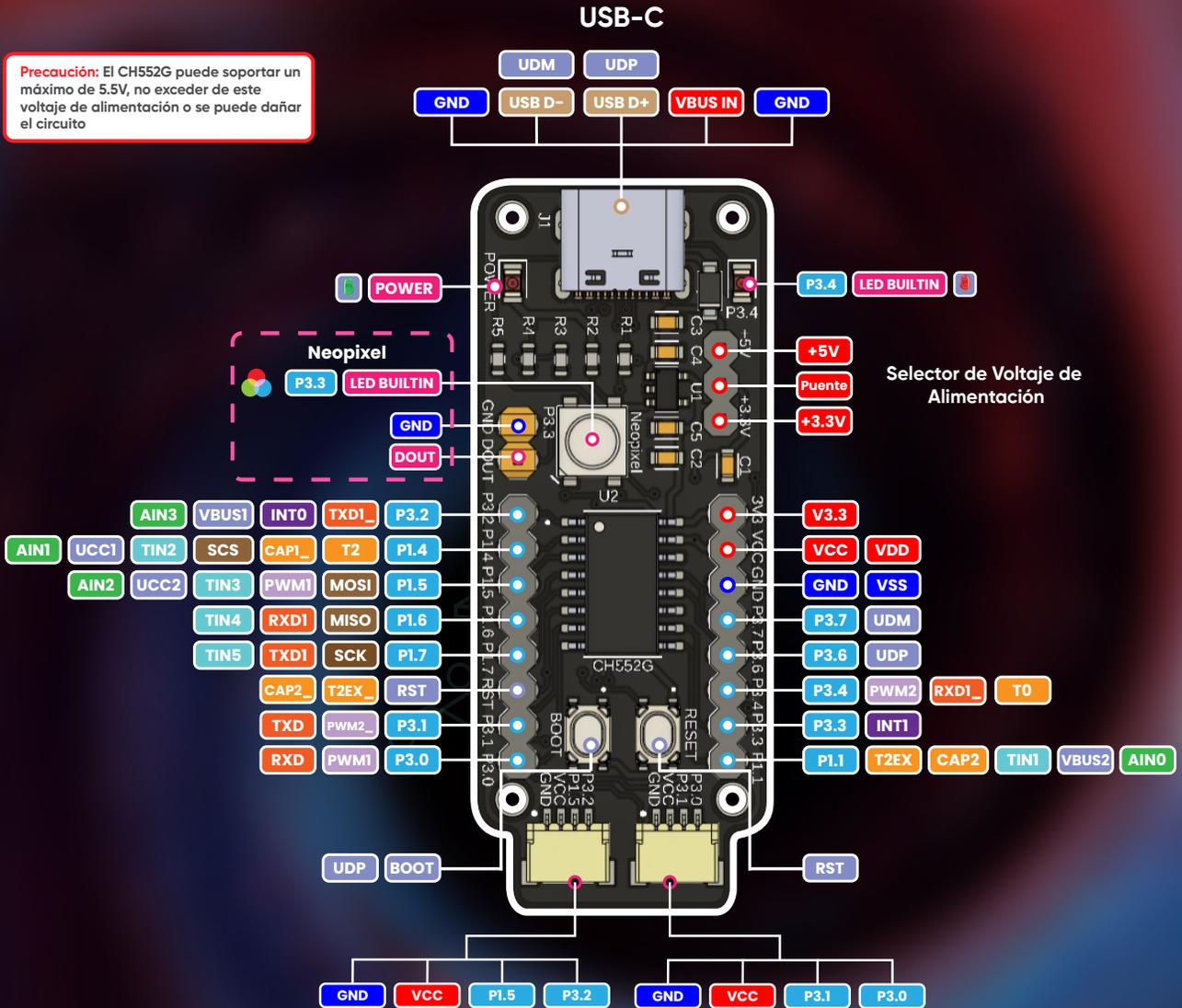
Contenido

✦ Descripción	01
✦ Características	02
✦ Requisitos de la tarjeta	03
✦ Contenido	04
✦ Pinout de la tarjeta	05
✦ Condiciones recomendadas de operación	06
✦ Power Tree	
✦ Diagrama a bloques	
✦ Operación	07
• Selector de Voltaje	
• Conector JST	
✦ Programación con Arduino IDE	09
• Instalación del CH552G en el IDE	
• Instalar el driver CH372	10
• Carga de código	12
• Ejemplo usando I2C y el conector JST	15
✦ Problemas comunes	16
✦ Información Adicional	17

Pinout de la tarjeta

VISTA FRONTAL

Precaución: El CH552G puede soportar un máximo de 5.5V, no exceder de este voltaje de alimentación o se puede dañar el circuito



Legenda:

- | | | |
|---|---|---|
| ■ Alimentación | ■ Analógico | ■ Timers |
| ■ GND y VSS | ■ UART/USART | ■ Interrupción |
| ■ Componentes | ■ PWM | ■ Sistema (Reset, Programación, USB, Boot) |
| ■ GPIO | ■ SPI | ■ Botón touch capacitivo |
| ■ LED | ■ LED RGB | ■ LED |

Precaución: Cuando alimentes la tarjeta a través del conector JST o conectes un módulo como salida, asegurate de seleccionar el voltaje correcto con el selector.

Condiciones recomendadas de operación

Símbolo	Descripción	Mínimo	Típico	Máximo	Unidades
V_{USB}	Voltaje de alimentación vía USB	3.14	3.3/5	5.255	V
V_{In}	Voltaje de alimentación desde los pines	2.7	3.3/5	5.5	V
T_{op}	Temperatura de operación	-40	25	85	°C

Power Tree

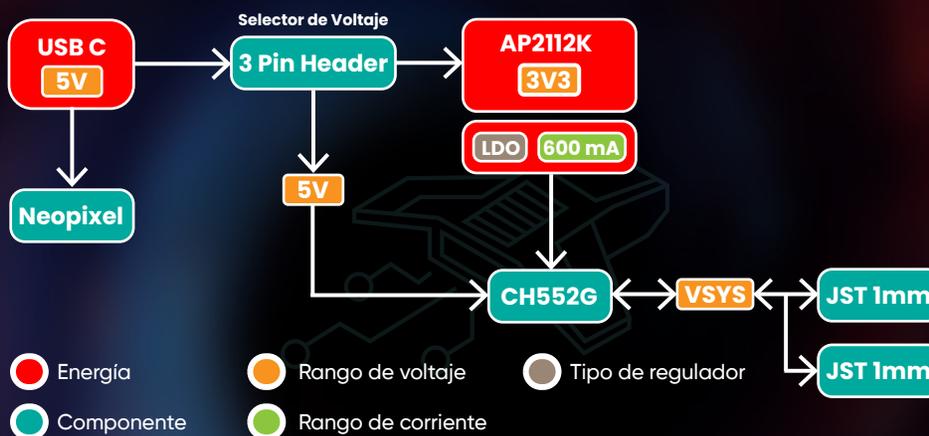
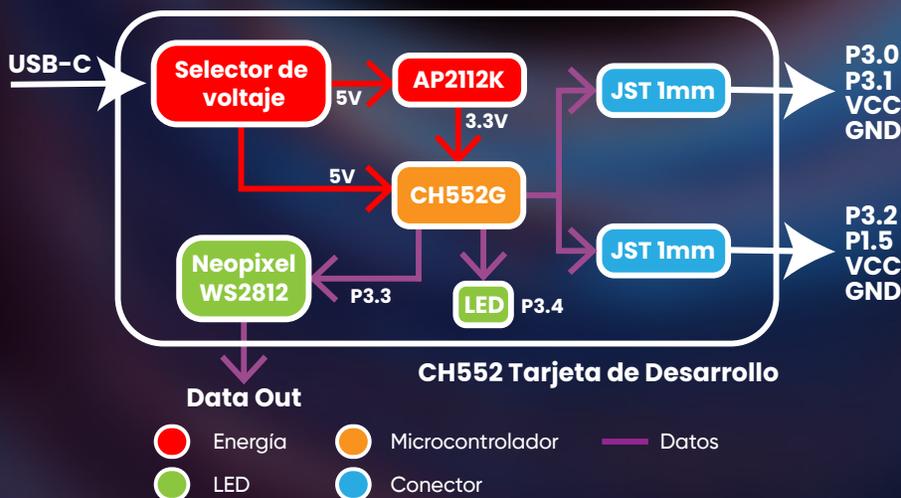


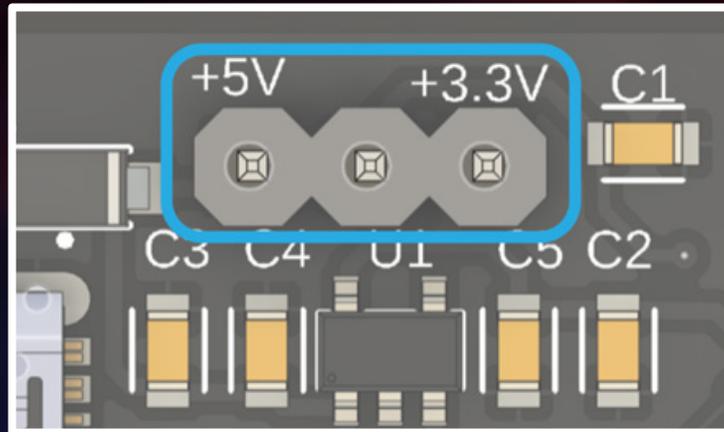
Diagrama de bloques



Operación

✦ Selector de Voltaje

La tarjeta de desarrollo contiene 3 pines junto con un jumper switch, estos 3 pines definen el voltaje de operación con el que va a trabajar la tarjeta.



El pin central entre el pin de +5V y +3.3V, es el puente de alimentación. Por lo que si se conectan a través del jumper switch el pin central y +5V, la tarjeta funcionará con 5 volts, de lo contrario, al conectar el pin de +3.3V con el pin central se activará el regulador APK2112K y la tarjeta se alimentará con 3.3V.



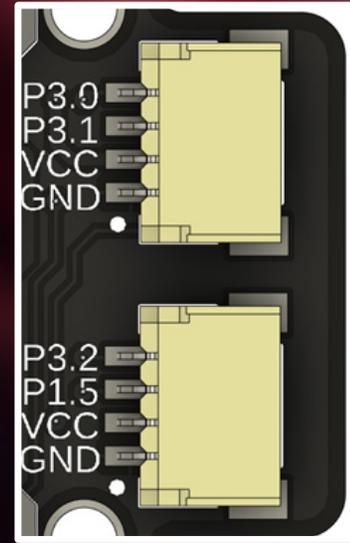
NOTA: Dependiendo del voltaje con el que se vaya a trabajar, se debe asegurar que esté el jumper switch en la posición correcta para evitar daños a los módulos, componentes que se conecten a la tarjeta e incluso a la tarjeta misma.

✦ Conectores JST

La Cocket Nova contiene dos conectores JST de 1 mm conectados a diferentes pines, el primer conector está conectado directamente a los GPIO 3.0 y 3.1 del microcontrolador, mientras que el segundo está conectado a los pines 3.2 y 1.5.

Ambos conectores están conectados paralelamente al voltaje de alimentación seleccionado a través del jumper switch y son compatibles con protocolos QWIC, STEMMA QT o similares.

Ya que tienen la misma distribución de pines, sin embargo, hay que asegurarse que el voltaje del selector coincida con el del sistema para evitar daños a los circuitos. De la misma manera, también se puede alimentar la tarjeta desde estos puertos y utilizar otras funciones como PWM y comunicación serial.



✦ LEDs Built-IN

Existen dos LEDs directamente conectados al microcontrolador, el primero está conectado al pin 3.4 mientras que el segundo es un LED Neopixel conectado al pin 3.3, este LED ofrece una salida de dos headers conectados a la salida de datos del Neopixel y a la tierra de la tarjeta respectivamente para poder conectar externamente más LEDs.

Para usar esta salida hay que conectar el pin DOUT al pin DIN del siguiente LED en la fila. Para la alimentación se puede usar el pin VCC siempre y cuando los LEDs externos se puedan alimentar con este voltaje, de lo contrario se necesitará alimentarlos con una fuente externa.

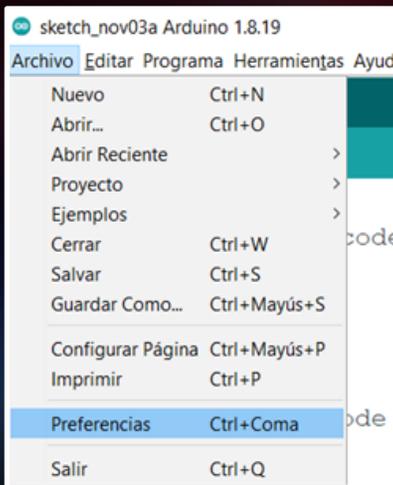


Programación con Arduino IDE

✦ Instalación del CH552G en el IDE

Dentro del Arduino IDE ir a la siguiente ruta:

Archivo>Preferencias>"Gestor de URLs Adicionales de Tarjetas"



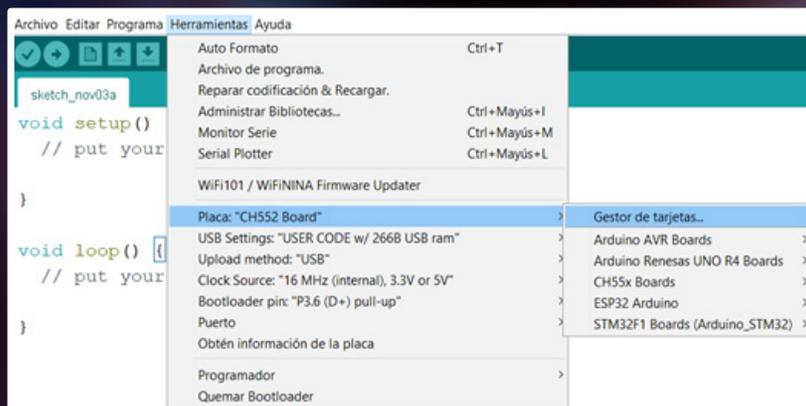
Y agregar la siguiente URL:



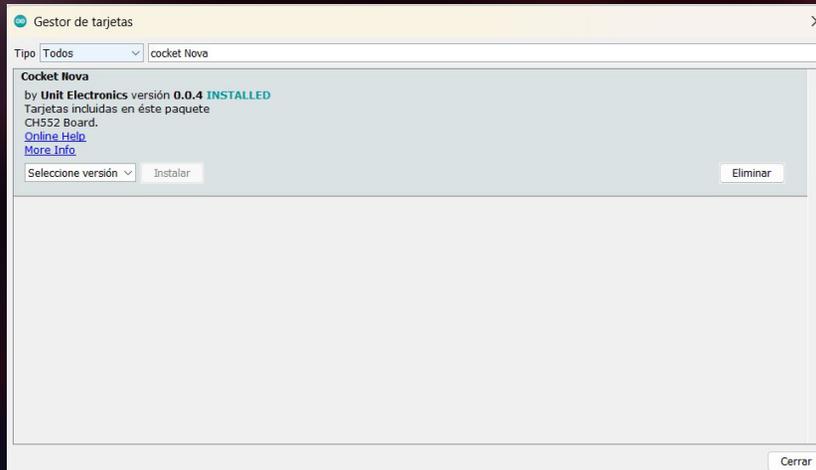
https://raw.githubusercontent.com/UNIT-Electronics/Uelectronics-CH552-Arduino-Package/refs/heads/develop/package_duino_mcs51_index.json

Después de esto hay que instalar los microcontroladores de la serie CHXXX en la siguiente ruta:

Herramientas>Placa:...>Gestor de tarjetas



Y dentro del gestor de tarjetas hay que buscar “Ch55xduino” e instalar las tarjetas.



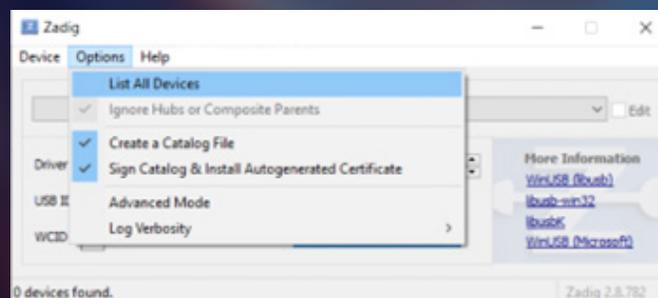
✦ Instalar el driver CH372

Este driver es el que permitirá cargar el código a nuestro microcontrolador, para ello requerimos descargar el [driver CH372](#) y el [software Zadig](#).

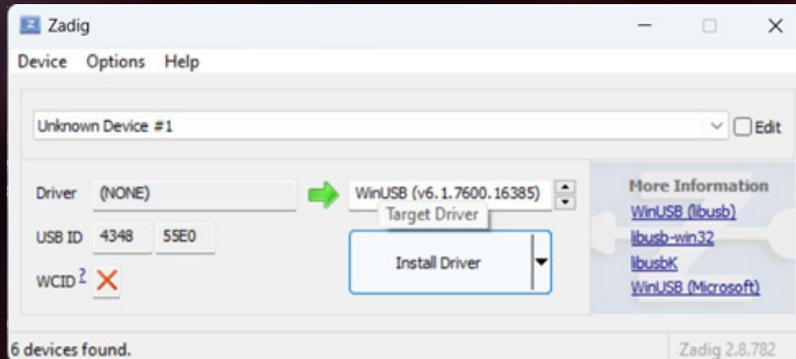
Abriremos el software Zadig y antes de conectar la tarjeta a la computadora, mantén presionado el botón “Boot”, sin soltar el botón conecta la tarjeta a la computadora y después suelta el botón, de esta manera el microcontrolador habrá entrado en modo de programación.

Dentro de Zadig iremos a la siguiente ruta:

Options>List all Devices

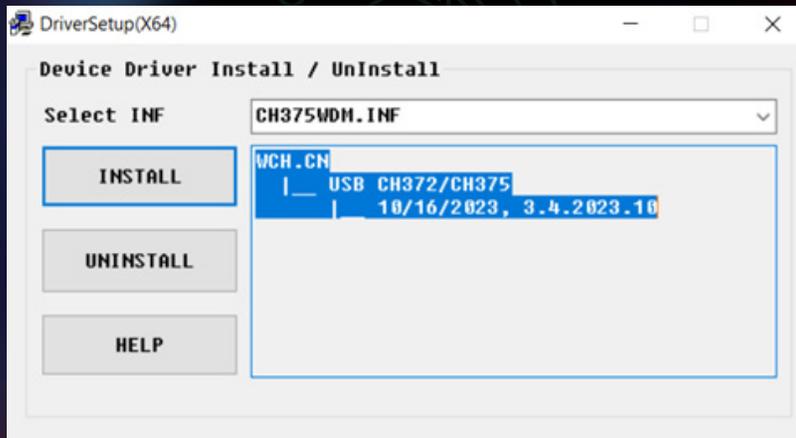


Sabremos que la computadora nos detecta la tarjeta porque en el menú desplegable nos aparecerá un dispositivo llamado “USB Module” o “Unknown Device”.



Seleccionamos el dispositivo que nos aparezca de los anteriores y elegimos el driver WinUSB, después de esto le daremos “Install Driver” y esperaremos a que termine la instalación.

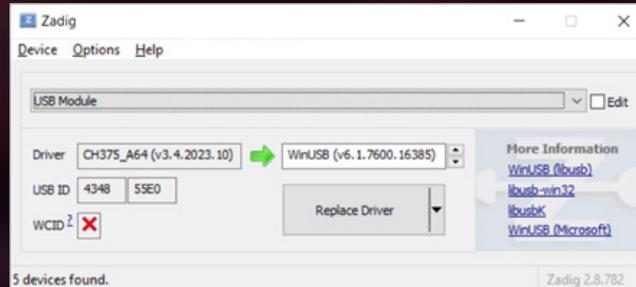
Una vez terminada la instalación, abrimos el instalador del driver CH375 y damos clic en instalar.



Desconectamos la tarjeta de la computadora y volveremos a oprimir el botón de “Boot” mientras conectamos la tarjeta como en el paso anterior. Esta vez, Zadig nos detectará la tarjeta como “USB Module” y nos marcará el driver CH375.

Continuación >>>

De esta manera nuestra tarjeta ya estará lista para que le podamos cargar algún programa, por último desconectamos la tarjeta en lo que preparamos el código.



✦ Carga de código

El código que se usará de ejemplo es para encender el LED neopixel que se encuentra en la tarjeta, este LED se encuentra conectado al pin 3.3 del CH552; sin embargo, al momento de programar, el IDE no nos reconocerá el pin 3.3 por el punto decimal. Entonces para todos los pines del CH552 se deberán definir sin el punto. Por ejemplo: el pin 3.3 se deberá definir como el 33.

EL código de ejemplo se encuentra en el siguiente enlace:



https://github.com/UNIT-Electronics/ch552_neopixel/tree/main/software/neopixel_lite

Al ser un proyecto, es necesario descargar todos los archivos que se encuentran en el enlace y guardarlos dentro de una misma carpeta, hay que abrir los archivos con extensión .C e .ino en el IDE de Arduino. Dentro del archivo con extensión .C se encuentra básicamente toda la codificación, aquí mismo vamos a cambiar algunos parámetros para hacer funcionar el LED:

```
#define delay 500
#define NeoPixel 1 // Number Neopixel connect
#define level 50 // Illumination level 0 to 255
```

Al modificar estos parámetros indicaremos que solo se va a encender un solo LED Neopixel y con un brillo de valor 50 en la escala del 0 al 255. Lo único que hace falta modificar es el pin de salida de datos que en este caso es el P3.3, para esto, dentro de las carpetas que descargamos se encuentra una que se llama src dentro de la siguiente ruta:

C:\...\ch552_neopixel-main\software\neopixel_lite

Y dentro de esta carpeta se encuentra un archivo llamado config.h, vamos a editar este archivo usando el bloc de notas o algún editor de texto y modificaremos el PIN_NEO de la siguiente manera:

```
#define PIN_NEO P33 // pin connected to NeoPixel
```

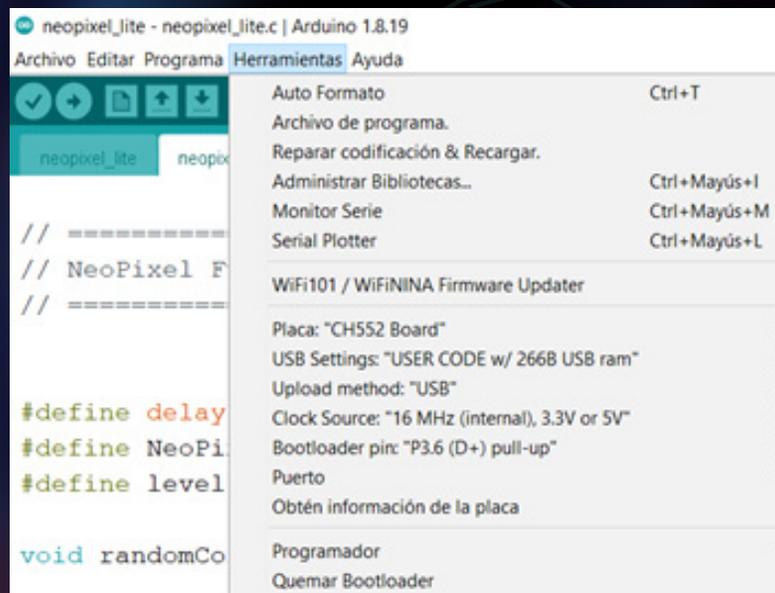
Guardamos y cerramos el archivo y después de esto iremos a nuestro programa en el Arduino IDE.

Iremos a la siguiente ruta seleccionaremos la tarjeta CH552:

Herramientas>Placa”...”>CH55x Boards>CH552 Board

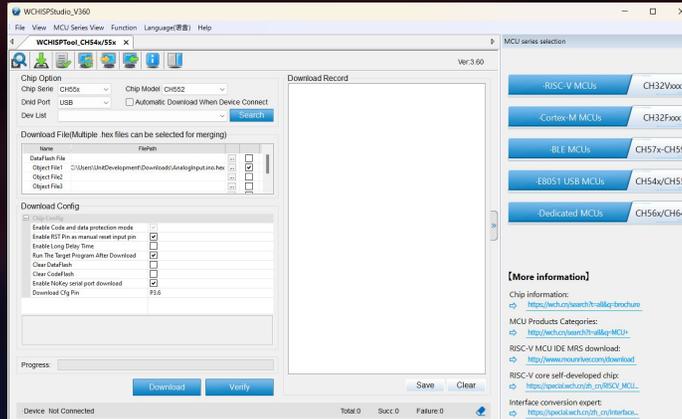
También dentro de la pestaña de herramientas modificaremos los siguientes parámetros:

USB Settings: USER CODE w/ 266B USB ram
Clock Source: 16MHz (internal), 3.3V or 5V

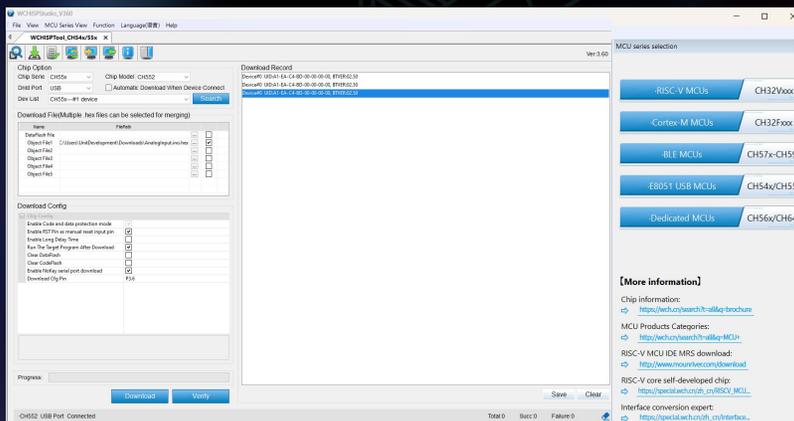


Es posible que el microcontrolador necesite estar más tiempo en modo Boot para poder cargar el programa con Arduino IDE. Para resolver esto, es necesario descargar **WCHISP Studio**, que es el software de configuración del fabricante.

Una vez descargado, dentro del menú de Chip Option seleccionamos la serie CH552x y el modelo CH552 como se muestra a continuación:



Después de esto, conectamos la tarjeta mientras presionamos el botón BOOT, y WCHISP Studio nos la reconocerá, mostrándonos el historial de sucesos y el nombre del dispositivo.



Una vez hecho esto, regresamos al Arduino IDE y cargamos el código. Si el código se cargó correctamente veremos como se enciende y apaga el LED oscilando entre colores de forma aleatoria:



✦ Ejemplo usando I2C y el conector JST

El CH552G no contiene puertos nativos para protocolo I2C; sin embargo, se puede usar este protocolo como SoftI2C el cual utiliza los GPIOs del microcontrolador para simular los pines SDA y SCL del protocolo I2C.

Para este ejemplo utilizaremos un **UNIT Módulo Hub I2C QW/ST 5 Puertos** y un **display OLED**, primero descargamos el código del siguiente enlace y al igual que con el código del LED Neopixel guardaremos todas las carpetas y archivos dentro de una misma carpeta:

https://github.com/UNIT-Electronics/OLED_ch552/tree/main/oled_hola_mundo

Dentro del archivo config.h que se encuentra en:

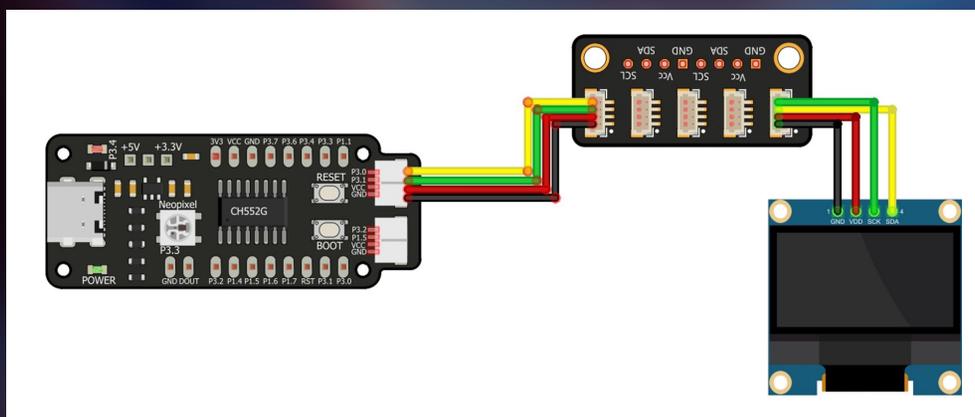
```
C:\...\OLED_ch552-main\OLED_ch552-main\oled_hola_mundo\src
```

Modificaremos los pines de SDA y SCL y se los asignaremos a los pines 3.1 y 3.0 de la tarjeta respectivamente:

```
#define PIN_SDA    P31    // I2C SDA
#define PIN_SCL    P30    // I2C SCL
```

Usaremos la misma configuración de programación y el mismo método para cargar el programa que usamos para el LED Neopixel.

Para las conexiones, primero conectaremos la tarjeta al módulo de expansión, el cual nos permite conectar varios dispositivos con I2C para controlarlos a partir de un mismo Hub y hacer conexiones de manera rápida.



Problemas comunes

✦ La tarjeta no enciende:

Asegúrate de que esté bien conectada, el cable USB no tenga falso contacto y tu fuente de alimentación esté dando el suficiente voltaje para alimentar la tarjeta.

✦ El LED Neopixel no enciende:

Revisa que el código se haya cargado correctamente y de que estén configurados correctamente en el programa el número de leds y la intensidad del brillo.

✦ Los Neopixel conectados externamente al puerto de salida de datos no encienden o el nivel de luz es muy bajo:

Verifica que la fuente de alimentación que estás utilizando es capaz de suministrar la corriente y voltaje necesario para alimentar todos los leds, de la misma manera hay que revisar que el pin DOUT de la tarjeta esté conectado al pin DIN del siguiente Neopixel.



PRECAUCIÓN: No conectar una fuente de alimentación de más de 5 volts directamente al conector USB, ya que se puede dañar el microcontrolador. Los Neopixel externos deben ser alimentados externamente en caso de requerir más voltaje.

✦ El LED Built-In del pin 3.4 no enciende:

Asegúrate que dentro de tu código esté configurado el pin 3.4 como salida y que esté haciendo las operaciones adecuadas.

✦ El programa no se carga en el microcontrolador:

Mantén presionado el botón de BOOT mientras conectas tu tarjeta de desarrollo, después de esto carga el programa. Si el problema persiste, vuelve a instalar los drivers ó cambia la configuración de la programación en el IDE que estás usando. También revisa que tu cable USB sea capaz de transmitir datos.

Si no carga el programa y el IDE te reconoce la tarjeta como un puerto COM, pulsa una vez el botón reset y sin desconectar la tarjeta carga el código.

También puede ser que el programa tarde demasiado para subirse y el microcontrolador salga del modo Boot, para esto, conecta la tarjeta mientras presionas el botón Boot cuando el IDE este por terminar la compilación y antes de que empiece a cargar el programa, asegúrate de soltar el botón justo antes de que lo cargue.
Por último, si no funciona ninguno de estos métodos, reinicia tu computadora

✦ El módulo conectado al conector JST no enciende:

Asegúrate de que el selector de voltaje esté configurado al voltaje que deseas manejar y que el módulo que conectes funcione con los mismos niveles de voltaje que puede otorgar la tarjeta de desarrollo.

Información Adicional

✦ Para más información de como usar la Cocket Nova y códigos de ejemplo, visita:



Cocket Nova & Arduino IDE



Ejemplos de Cocket Nova CH552 usando SDCC



GitHub, repositorio oficial de Cocket Nova



Guía de inicio rápido de Cocket Nova



Página oficial de UNIT Electronics



Página oficial de WinChipHead



Recursos de hardware de Cocket Nova

COCKET NOVA CH5520

DEVELOPMENT BOARD

La innovación comienza con 8 bits.



Desarrollado por el equipo de UNIT

Encuentra más productos en

 uelectronics.com