

CARRERA: TECNICATURA SUPERIOR EN TELECOMUNICACIONES

MATERIA: ELECTRONICA MICROCONTROLADA

DOCENTES:

- JORGE E. MORALES, INGENIERO ELECTRICISTA ELECTRÓNICO (UNIVERSIDAD NACIONAL DE CÓRDOBA)
- C. GONZALO VERA, TECNICO SUPERIOR EN MECATRONICA (U.T.N.)

ESP32



CORDOBA, 29 DE JULIO DEL 2022.-

GRUPO N°2:

- DARIO ARRIOLA
- DANIEL RODRIGUEZ
- JEREMIAS CASTRO
- CARLA ARGENTINA WAYAR

¿Qué es el ESP32?

Es un chip de la familia SoC, que fue desarrollado por la empresa china espressif system, este incluye wifi, bluetooth y un montón de características.

El ESP32 es el sucesor del ESP8266, tiene más GPIO con más funcionalidades en comparación con el ESP8266.. Combina capacidades inalámbricas Wi-Fi y Bluetooth y dual-core.

Con el ESP32 puede decidir qué pines son UART, I2C o SPI; solo necesita configurar eso en el código. Esto es posible gracias a la característica de multiplexación del chip ESP32 que permite asignar múltiples funciones al mismo pin. Si no los establece en el código, los pines se utilizarán de manera predeterminada, como se muestra en la figura a continuación (la ubicación del pin puede cambiar según el fabricante).

¿Cómo podemos usar el ESP32?

Podemos usarlo de dos maneras diferentes: de forma independiente para agregarlo a nuestras placas personalizadas o ya agregado a nuestro kit de desarrollo listo para usar, esta segunda opción es la más rápida para comenzar a programarlo y ver resultados.

Existen muchos kits de desarrollo diferentes que incluyen el chip ESP32 en sus placas nosotros nos centraremos en uno de los más básicos y fáciles de usar.

I. ESPECIFICACIONES TECNICAS:

Numero de núcleos	2 (doble núcleo)
Wifi	2,4 GHz hasta 150 Mbits/s
Bluetooth	BLE (Bluetooth Low Energy) y Bluetooth heredado
Arquitectura	32 bits
Frecuencia de reloj	Hasta 240 MHz
RAM	512KB
PINES	30 o 36 (depende del modelo)

Periféricos	<p>interfaz periférico con DMA que incluye táctil capacitivo</p> <p>ADC (convertidor analógico a digital)</p> <p>DAC (convertidor de digital a analógico)</p> <p>I²C (Circuito Inter-Integrado)</p> <p>UART (receptor/transmisor asíncrono universal)</p> <p>SPI (interfaz periférica en serie)</p> <p>I²S (Sonido Interchip Integrado)</p>
-------------	---

pág. 2

	<p>RMII (interfaz independiente de medios reducidos)</p> <p>PWM (modulación de ancho de pulso).</p>
<p>Bajo consumo de energía. -</p> <p>Seguridad: aceleradores de hardware para AES y SSL/TLS.-</p>	

Para dar un poco más de detalle, se menciona algunas características de la versión de 30 pines:

esta pequeña placa dispone de 30 pines de conexión que podemos utilizar o bien utilizando cables o pinchando en una protoboard, dispone de un puerto micro USB que usaremos para llevar a cabo su programación o para suministrarle energía.

Su voltaje de alimentación es de 5V cuando usamos el micro USB o el pin VIN, pero todos sus pines de entrada y salida funcionan a 3,3 V.

PROCESADOR:

El cerebro principal de esta placa es un procesador de dos núcleos de 32 bits en concreto el procesador Xtensa Dual Core 32-Bits LX6 con 600 DMIPS.

COPROCESADOR:

realiza operaciones de ultra-bajo consumo cuando el dispositivo se encuentra en modo Deep Sleep.

La frecuencia de operación es de 240 Mhz, ejecuta instrucciones 15 veces más rápido que una placa arduino uno, nos permite realizar operaciones con números reales de forma muy eficiente, así como realizar operaciones de números grandes de forma instantánea.

TIPOS DE MEMORIA:

Memorias Internas y externas

Las memorias internas son las que se encuentran ya incluidas dentro del chip y las externas son las que podemos incluir para expandir la capacidad de el sistema.

Memoria ROM: dispone de una memoria ROM de 448 KiB destinados al arranque y gestion del wifi y bluetooth, esta memoria es de solo escritura, es decir que no la podemos reprogramar.

Memoria SRAM interna: 520 KiB destinados a datos e instrucciones temporales, para el procesador es mucho mas fácil acceder a esta que, a la memoria externa, lo que aumenta la velocidad de procesamiento. Memoria RTC

SRAM: 16 KiB utilizados por el co-procesador para realizar operaciones de ultra-bajo consumo. Memoria Efuse: 256

Bits utilizados por el propio sistema y los 768 Bits restantes reservados para otras aplicaciones.

Memoria Flash Empotrada (Embedded Flash): Memoria variable destinada al almacenamiento del programa (sketch). La cantidad de esta memoria varía dependiendo del chip que estemos usando.

pág. 3

Existe la posibilidad de agregar más memoria cuando la memoria es insuficiente para nuestra aplicación es posible añadir más memoria de forma externa. Se pueden agregar hasta 16 MiB de memoria flash externa y hasta 8MiB de memoria SRAM externa.

SEGURIDAD:

El ESP32 cuenta con aceleradores de algoritmos orientados a la encriptación y soporta

los siguientes tipos de cifrado:

- SHA
- RSA
- AES
- RGN

CONECTIVIDAD INALÁMBRICA:

Dispone de wifi integrado de alto rendimiento en la banda de los 2.4 GHz que permite alcanzar velocidades de hasta 150Mbps/s asi como tambien conexion bluetooth compatible con bluetooth.

Dispone de un sensor de efecto HALL para detectar campos magnéticos y en algunas versiones antiguas incluye un sensor de temperatura

II. TIPOS DE VERSIONES:

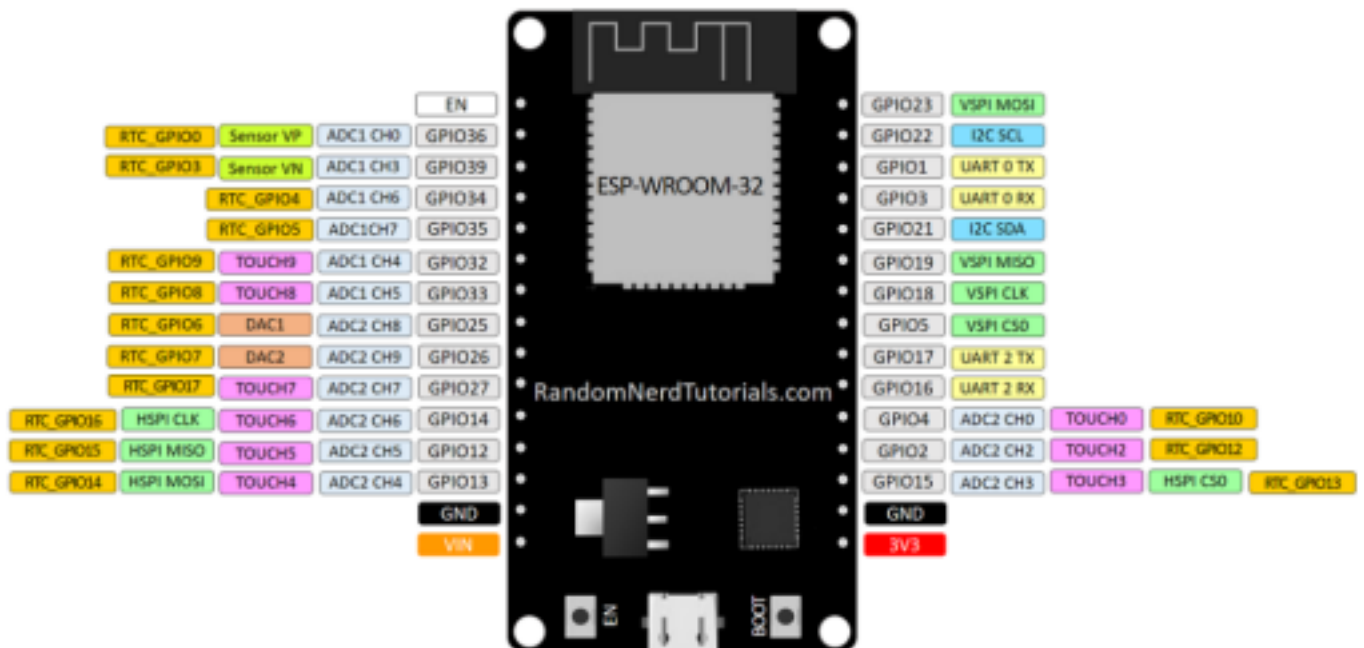


TIPOS DE PLACAS

VERSION DE 30 PINES

ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs



- **CONFIGURACIÓN DE PINES Y NÚMERO DE PINES:** Para usar correctamente el ESP32 en sus proyectos, debe tener acceso al pinout de la placa (como un mapa que muestra qué pin corresponde a qué GPIO y sus características). Así que asegúrese de tener acceso al pinout de la placa que está obteniendo. De lo contrario, puede terminar usando el ESP32 incorrectamente. Recomendamos consultar nuestra Guía de referencia de ESP32 GPIO que muestra cómo usar los ESP32 GPIO correctamente.
- **INTERFAZ USB A UART Y CIRCUITO REGULADOR DE VOLTAJE:** La mayoría de las placas de desarrollo con funciones completas tienen estas dos funciones. Esto es importante para conectar fácilmente el ESP32 a su computadora para cargar el código y aplicar energía.
- **BOTONES BOOT Y RESET** para poner la placa en modo intermitente o resetear (reiniciar) la placa. • **CONECTOR DE BATERÍA.** Si desea alimentar su ESP32 con baterías, hay placas de desarrollo que vienen con conectores para baterías LiPo; esto puede ser más útil. También puede alimentar un ESP32 "normal" con baterías a través de los pines de alimentación.
- **CARACTERÍSTICAS ADICIONALES.** Hay placas de desarrollo ESP32 con características adicionales como una pantalla OLED , un módulo LoRa , un módulo SIM800 (para GSM y GPRS), un soporte de batería o una cámara.

NOTA: Los pines resaltados en verde están bien para usar. Los resaltados en amarillo están bien para usar, pero debe prestar atención porque pueden tener un comportamiento inesperado principalmente en el arranque. No se recomienda utilizar los pines resaltados en rojo como entradas o salidas.

III. ENTORNO DE PROGRAMACION:

El ESP32 se puede programar en diferentes entornos de programación. Puedes usar:

- IDE DE ARDUINO
- ESPRESSIF IDF (MARCO DE DESARROLLO DE IOT)
- MICROPITÓN
- JAVASCRIPT
- LUA

Compatible con Arduino IDE: puede programar el ESP32 con Arduino IDE utilizando el núcleo de Arduino.(Instrucciones de instalación de Windows, Mac OS X y Linux). También puede usar otros IDE para programar el ESP32 con el núcleo de Arduino (como VS Code con la extensión PlatformIO , por ejemplo).

Compatible con MicroPython : puede programar el ESP32 con firmware MicroPython (Comience con MicroPython en ESP32).-

ESP-IDF

Espresif IoT Development Framework. Official development framework for ESP32.

MICROPYTHON:



MicroPython

Arduino IDE



ESP32-CAM

El ESP32-CAM es un módulo que se puede usar con multitud de proyectos, y con Arduino. Es un completo módulo con un microcontrolador integrado, lo que lo puede hacer funcionar de forma independiente. Además de la conectividad WIFI+Bluetooth, este módulo también cuenta con una cámara de video integrada y con una ranura microSD para almacenar

Este módulo no es nada caro y puede tener multitud de aplicaciones. Desde algunas simples de IoT, como otras más avanzadas para el seguimiento y reconocimiento de imagen usando IA, e incluido como sistema de vigilancia para comprobar lo que sucede en un lugar de forma remota estés donde estés

Características:

El módulo ESP32-CAM tiene algunas características técnicas muy interesantes que puedes ver en el datasheet del fabricante.

- Conectividad WiFi 802.11b/g/n + Bluetooth 4.2 con BLE. Soporta carga de imagen por WiFi
- Conexiones UART, SPI, I2C y PWM, Tiene 9 pines GPIO
- Frecuencia de reloj: Hasta 160 Mhz
- Potencia del cómputo del microcontrolador: hasta 600 DMIPS.
- Memoria: 520 KB de SRAM + 4MB de PSRAM + SD slot para tarjetas
- Extras: tiene múltiples modos de reposo, firmware actualizable por OTA, y LED para uso de memoria flash built-in
- Cámara: soporta cámaras OV2640 que puede venir en el pack o comprarse independientemente.

Este tipo de cámaras tienen:

- 2 MP en su sensor
- Tamaño de arreglo UXGA de 1622x1200 px
- Formato de salida YUV422, YUV420, RGB565, RGB555 y compresión de datos de 8-bit
- Puede transferir imagen entre 15 y 60 FPS

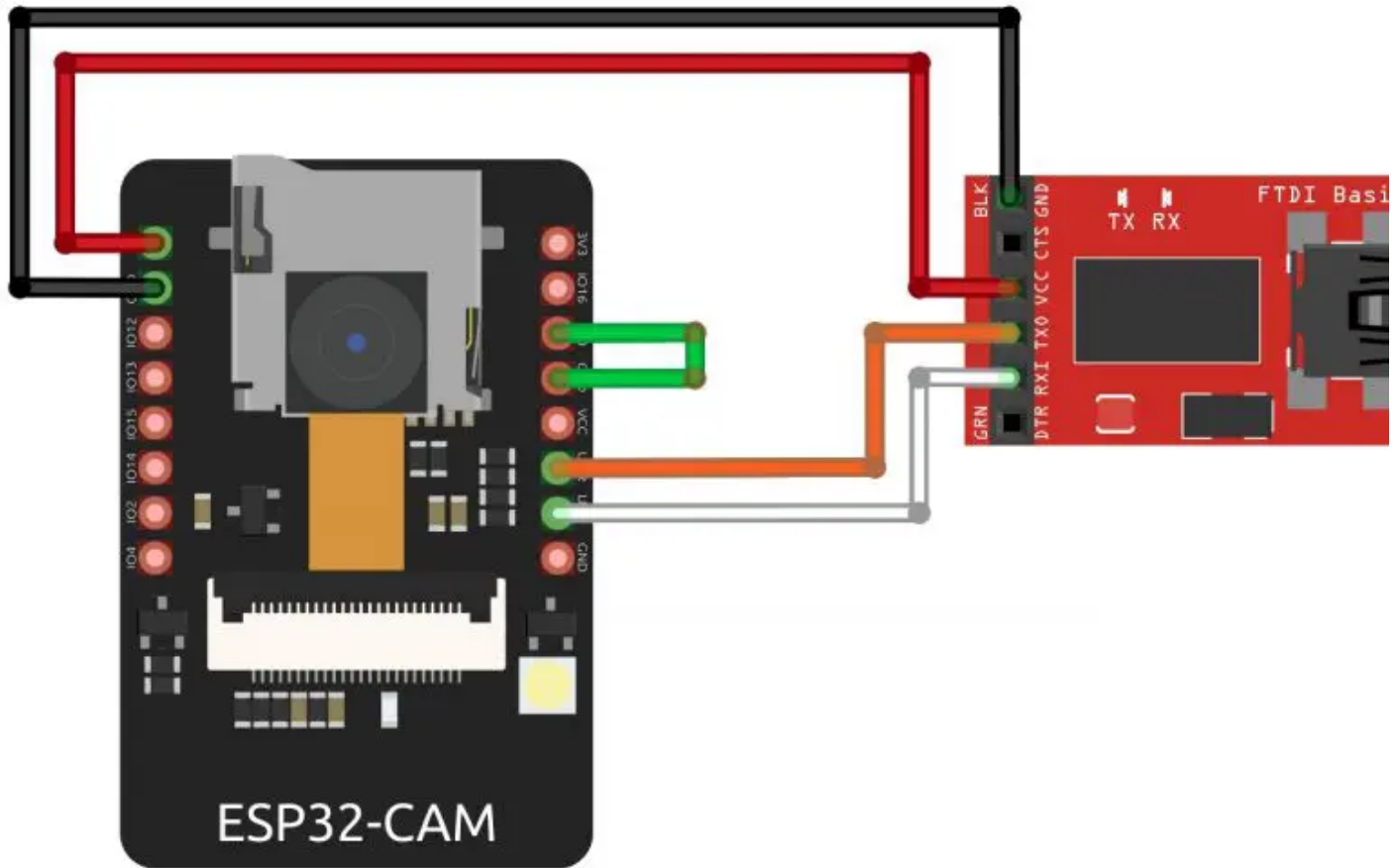


El pinout del ESP32-CAM es muy simple, como puedes observar en el diagrama anterior. Y la cámara, va conectada al conector que hay habilitado para ello.

Se puede usar un adaptador externo FTDI para conectar este módulo y que te sea más sencillo su control. Eso permite el uso de un puerto tipo mini USB en vez del cableado del ESP32-CAM. Para usar uno de estos módulos se puede conectar así:

- Configura el módulo FTDI para trabajar a 3.3v
- Puentear el pin GPIO y GND del módulo ESP32-CAM
- El pin 3v3 del módulo debe ir conectado a Vcc del FTDI
- GPIO 3 (UOR) del módulo irá al TX del FTDI
- GPIO 1 (UOT) del módulo va a RX del FTDI
- Y el otro GND del ESP32-CAM con GND del módulo FTDI

Integración con arduino IDE



Para poder integrar con FTDI, la conexión es muy sencilla. Tan solo tienes que hacer lo siguiente:

- Conecta la conexión de 5v del módulo ESP32-CAM al Vcc del módulo FTDI
- Conecta GND del módulo ESP32-CAM a GND del módulo FTDI.
- TX0 de la placa FTDI va a GPIO 3 (U0RXD)
- RXI de la placa FTDI va a GPIO 1 (U0TXD)
- y puentea GPIO 0 y GND de la placa ESP32-CAM

Ahora ya se puede conectar al equipo mediante USB a través del módulo FTDI.

Otra opción es conectarlo a Arduino directamente, sin usar el módulo FTDI. Pero veamos el caso con el FTDI...

1 - El siguiente paso es instalar la biblioteca ESP32 para poder aprovechar ésta. Para eso, desde Arduino IDE ve a Archivo > Preferencias > Allí, en el campo para añadir URL, añade: https://dl.espressif.com/dl/package_esp32_index.json y pulsa Aceptar. Ahora ve a Herramientas > Placa > Gestor de placas > busca ESP32 y pulsa instalar "ESP32 by Espressif Systems».

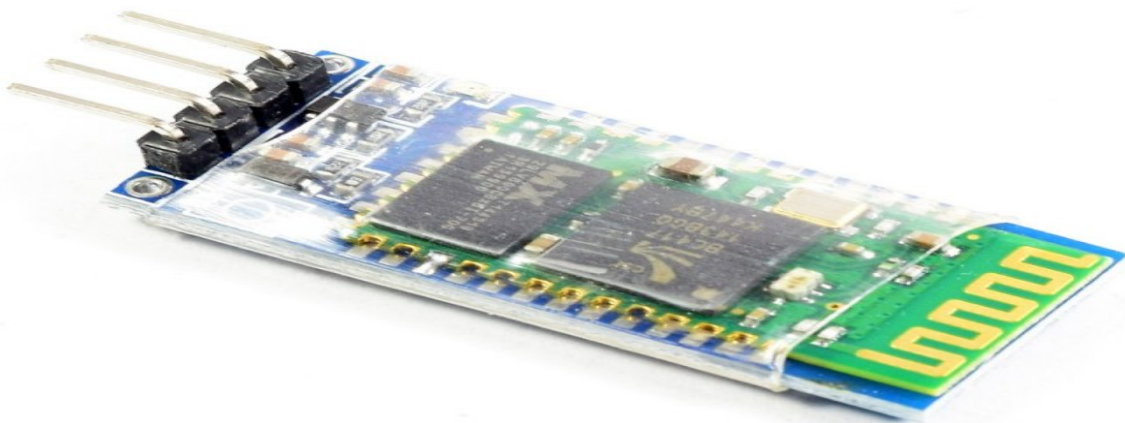
2 - Luego abre Arduino IDE > Herramientas > Placas > selecciona AI-Thinker ESP32-CAM (deberás tener instalado el add-on ESP32 para que aparezca dicha opción en el menú). Después ve a Herramientas > Puerto y selecciona COM, donde está conectada tu placa.

3- Ahora, puedes cargar un sketch en la placa, para que sea más simple, usa uno de los ejemplos vea Archivo > Ejemplo > ESP32 > Camera > CameraWebServer. Una vez hecho, cuando aparezca el mensaje de que se cargó con éxito, quita el cable del pin GPIO 0 de GND y presiona el botón Reset de la placa.

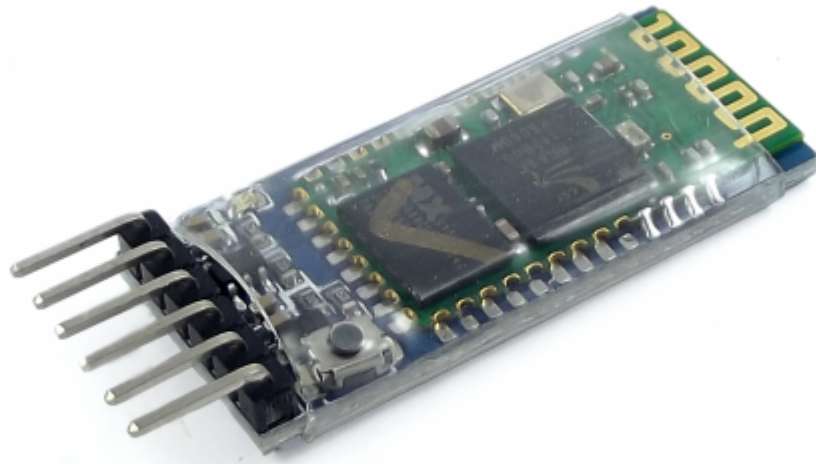
4- Por último, puedes usar y comenzar a ver los resultados en la interfaz web... Cuando lo ejecutes, te dará en el monitor una URL con una IP que debes insertar en tu navegador web para acceder. Desde ella se pueden ajustar parámetros y ver lo que se ve desde el sensor de la cámara.

Módulo Bluetooth HC-06 y el HC-05

Fisicamente se diferencian por el número de pines. En el HC-06 tiene un conector de 4 pines mientras que el HC-05 trae uno de 6 pines.



HC-06



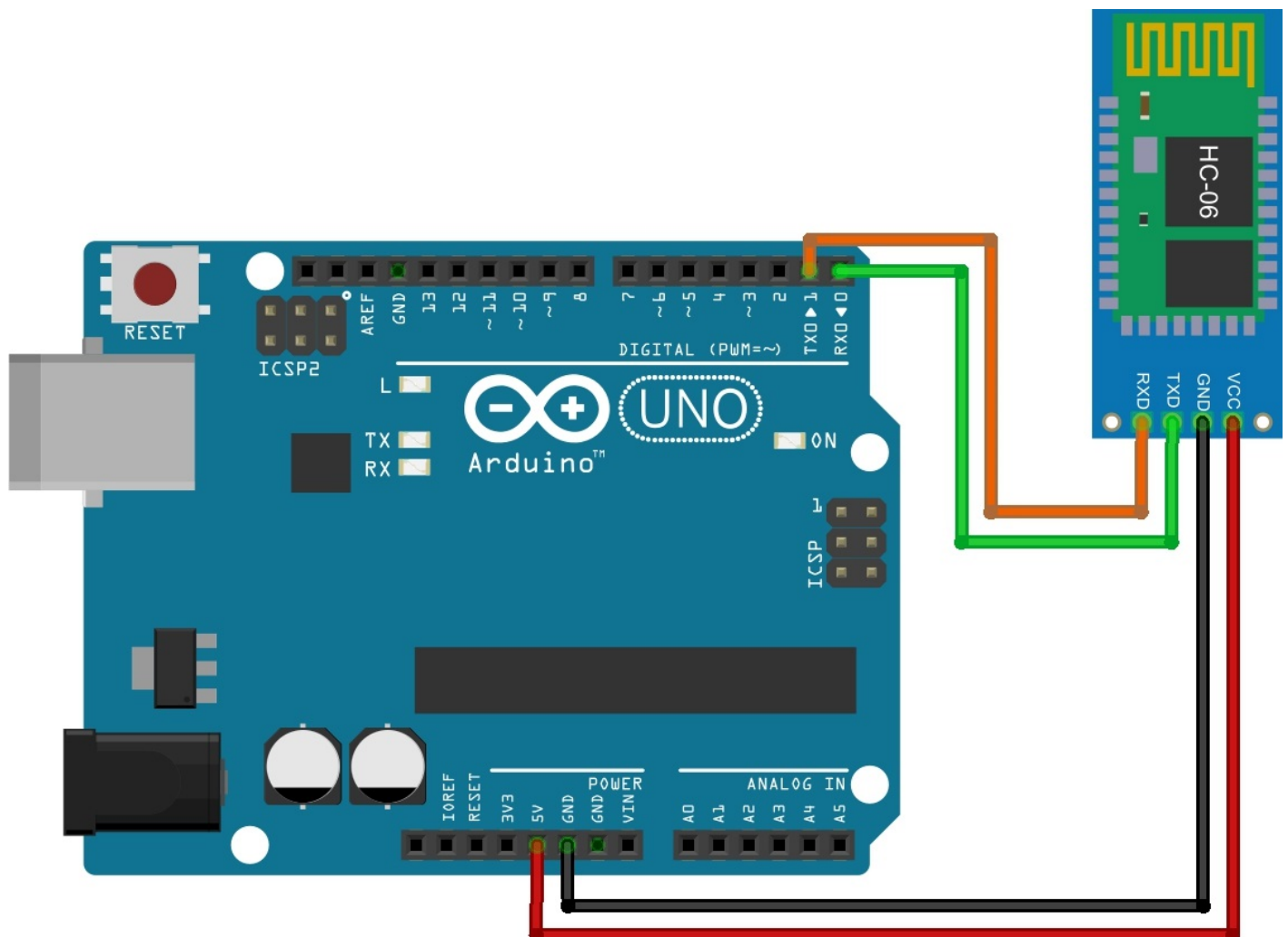
HC-05

El HC-06 tiene 4 pines:

- Vcc, Voltaje positivo de alimentación, aquí hay que tener cuidado por que hay módulos que solo soportan voltajes de 3.3V, pero en su mayoría ya vienen acondicionados para que trabajen en el rango de 3.3V a 6V
pero es bueno revisar los detalles técnicos de nuestro módulo antes de hacer las conexiones
- GND, Voltaje negativo de alimentación, se tiene que conectar al GND del arduino al GND de la placa que se esté usando
- TX pin de transmisión de datos, por este pin el HC-06 transmite los datos que le llegan desde la PC o Móvil mediante bluetooth, este pin debe ir conectado al pin RX del Arduino
- RX pin de recepción, a través de este pin el HC-06 recibirá los datos del Arduino los cuales transmitirán por bluetooth, este pin debe ir conectado al pin TX del arduino

Procedemos a realizar las siguientes conexiones entre nuestro módulo bluetooth

En este caso es un Arduino UNO R3



Para cargar el programa al Arduino, desconectamos los pines RX0 y TX0 del Arduino, pues internamente el Arduino trabaja con los mismos pines para cargar el programa y si están conectados al módulo Bluetooth, no nos dejara cargar (para evitar este inconveniente se puede usar el software serial y usar otros pines)

Ahora en el IDE de Arduino copiamos el siguiente código y luego lo cargamos a nuestro Arduino

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available())
  {
    char dato=Serial.read();
    Serial.print("Dato recibido: ");
    Serial.println(dato);
  }
}
```

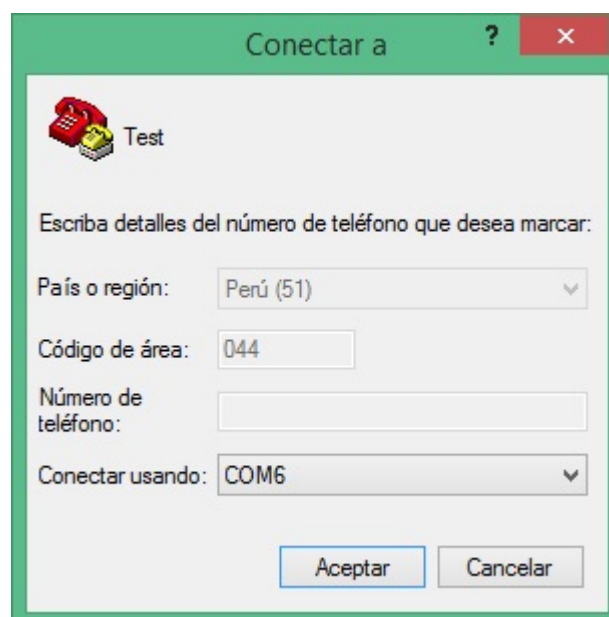
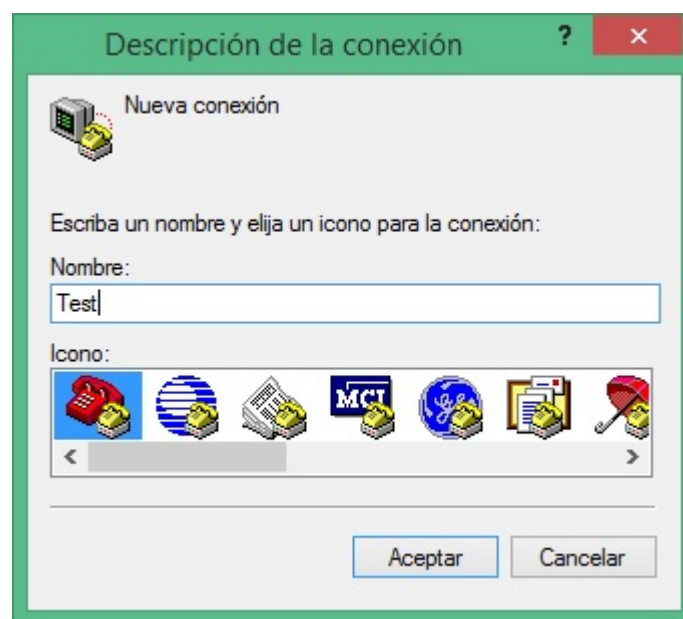

Después de terminar de cargar, nuevamente volvemos a conectar los pines RX y TX

El programa que hemos cargado al Arduino básicamente devolverá por Bluetooth cualquier dato que reciba a través de él con la variable "dato"

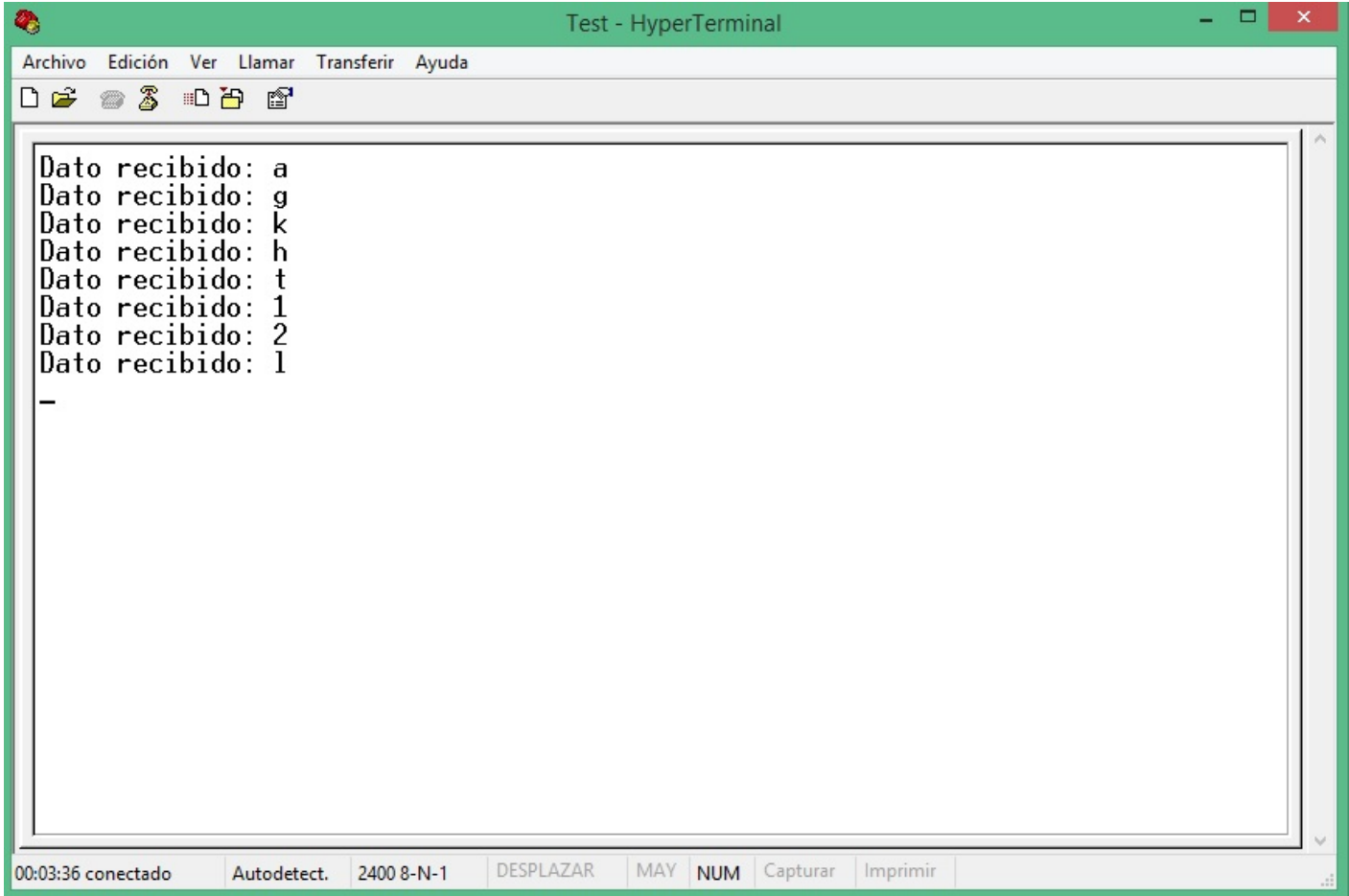
```
char dato= Serial.read();
```

Para conectarse al Módulo Bluetooth desde la PC usaremos el Hyperterminal

Abrimos y le asignamos cualquier nombre a la conexión y luego elegimos el puerto COM perteneciente al bluetooth que en nuestro caso es el COM6



Al darle en aceptar Hyperterminal hará la conexión al HC-06, cuando se establezca la conexión el HC-06 dejará de parpadear y en la parte inferior del Hyperterminal indicará CONECTADO, ahora podemos enviar por teclado cualquier carácter y el arduino nos devolverá el carácter enviado



The screenshot shows a HyperTerminal window titled "Test - HyperTerminal". The menu bar includes "Archivo", "Edición", "Ver", "Llamar", "Transferir", and "Ayuda". The toolbar contains icons for file operations and communication. The main text area displays the following received data:

```
Dato recibido: a  
Dato recibido: g  
Dato recibido: k  
Dato recibido: h  
Dato recibido: t  
Dato recibido: 1  
Dato recibido: 2  
Dato recibido: 1  
-
```

The status bar at the bottom shows "00:03:36 conectado", "Autodetect.", "2400 8-N-1", "DESPLAZAR", "MAY", "NUM", "Capturar", and "Imprimir".

También podemos hacer la conexión desde Proteus, Proton, Micro code Studio, Java, Labview, entre otros

