Primeros pasos con LILYGO TTGO T-A7670G ESP32 (4G GPRS y GPS)

Comienza con el ESP32 y el módulo T-A7670G 4G/GSM/GPRS. A lo largo de esta guía, aprenderá lo básico LILYGO TTGO T-A7670G ESP32 que combina el chip ESP32, el módulo A7670G (2G, A7670E, A7670SA), la ranura para tarjeta SD, la batería microSD, el GPS, la batería y el conjunto de antenas. Este tablero es ideal para proyectos de IoT que requieren conectividad 2G/3G/4G y ubicación GPS. También puede conectarse a Internet utilizando su plan de datos de tarjeta SIM y utilizar la aplicación GPS.

Características:

Este sistema de IoT ofrece una gama de placas, recorrido por varias redes 4G/3G/2G y Internet. Este tablero también puede enviar SMS y conectarse a Internet utilizando un plan de datos de tarjeta SIM. Si comprar la opción con GPS, podrás obtener la ubicación y la hora. Puede encontrar más información en el repositorio oficial de LILIGO T-A7670X Github.

- Este sistema admite varios modos de la placa para su región:
  - A7670E - Global (recomendamos esta versión con GPS)
  - A7670E - Europa / Tailandia Australia / America del Sur
  - A7670SA - América del Sur / Nueva Zelanda / Australia
  - Puedes comprar la tabla LILIGO TTGO T-A7670G aquí

Cuidado con la antena: Nuestro Nacional Tablero es un participante en programas de control de calidad y garantía de comportamiento en medio para evitar fallas de circuitos o Amazon, eBay, AliExpress y otros sitios. Si encuentra problemas con la calidad del producto, comuníquese con nosotros antes y nosotros le daremos respuestas satisfactorias.

Aprende ESP32 con Arduino IDE eBook : ¡Obtén consejos para programar el ESP32 con Arduino IDE!

SMART Home con Raspberry Pi, ESP32 y ESP8266 : aprende a construir un sistema completo de automatización del hogar.

Aprende Raspberry Pi Pico/Pico W con MicroPython : ¡Aprenda a programar el nuevo procesador de placa microcontroladora Raspberry Pi Pico/Pico W con MicroPython! Aprenda a programación MicroPython.

Aprende LORa : Comienza Ahora para proyectos ESP32 : Aprenda a construir interfaces prácticas de sensors LoRa para aplicaciones de IoT y M2M con ESP32, Nexpaio Dragino Library y con el IDE de Arduino.

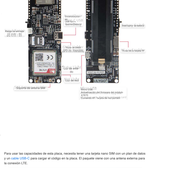
Seleccione una Configuración

Cookies helfen uns bei der Bereitstellung unserer Dienste. Durch die Inanspruchnahme unserer Dienste erklären Sie sich damit einverstanden, dass wir Cookies setzen.

Presentamos el LILYGO TTGO T-A7670G ESP32 (con GPS)

LILYGO TTGO T-A7670G ESP32 (con GPS)

La placa T-A7670G ESP32 es otra placa, recientemente lanzada por LILIGO. Este tablero combina el chip ESP32 con el módulo A7670G (2G, A7670E, A7670SA) y la placa. Esta opción que otra tablero puede enviar SMS y conectarse a Internet utilizando un plan de datos de tarjeta SIM. Si comprar la opción con GPS, podrás obtener la ubicación y la hora. Puede encontrar más información en el repositorio oficial de LILIGO T-A7670X Github.



Además del sistema A7670G, la placa también viene con algunas características interesantes: cuál es soporte de batería para una batería de litio recargable, un circuito de carga de batería donde puede cargar la batería mientras la placa está en uso, un conjunto de antenas para mejorar la recepción y ranura para tarjeta SD para guardar quizás de configuración.

Aquí hay un resumen de las características de la placa LILYGO TTGO T-A7670G ESP32 (con GPS):

- Voltaje de alimentación: 3.3 V CC o 5 V CC
- Chip ESP32 (dentro MR2049-B) (procesador de doble núcleo de 240 MHz)
- 4 MB de PSRAM
- 16 MB de Flash
- Wi-Fi integrado
- Bluetooth integrado
- Conectividad USB a través: CP2104 o CH9102 (controladores)
- Módulo SIMCOM A7670G incorporado
- Ranura para tarjeta SD para almacenamiento
- Ranura de antena SIM incorporada
- Disponible con o sin GPS
- Pantalla OLED de 0.96 pulgadas (opcional)
  - Interfaz de chips de energía para chip de 4 A 4 V
  - Soporte de batería incorporado 1c 18650
  - Cargador de batería de litio incorporado
- Ranura para tarjeta Micro SD integrada
- Interruptor de encendido/apagado de la batería
- Interruptor de encendido y arranque
- LED de estado

Aquí hay una descripción rápida del tablero:

Se suministra la placa LILIGO TTGO T-A7670G con GPS, también viene con una antena GPS que puedes conectar a la placa de la siguiente manera:

Tambien puedes conectar una ranura microSD:

¿Dónde comprar LILYGO TTGO T-A7670G ESP32?

Consulte el siguiente mapa de precios:

- LILYGO TTGO T-A7670G ESP32 (4G, GPRS y con GPS) - Maker Advisor

Debe seleccionar la versión correcta de la placa para su región:

- A7670E - Europa / Tailandia Australia / America del Sur
- A7670SA - América del Sur / Nueva Zelanda / Australia
- Puedes comprar la tabla LILYGO TTGO T-A7670G aquí

"A7670 Comparar"

	A7670G	A7670E	A7670SA
Product Model Series	LILIGO	LILIGO, TAIWE	
Product	Global	Europe / Thailand Australia	South America / New Zealand / Australia
Frequency Band	2G/3G/4G	4G (FDD-LTE)	4G (TDD-LTE)
Frequency Band	LTE FDD	B1/ B3/ B5/ B7/ B8/ B20	B34/ B38/ B39/ B40/ B41
Frequency Band	LTE TDD	B34/B38/B39/B40/B41	
Frequency Band	GSM	850/900/1800/1900MHz	850/900/1800/1900MHz
Frequency Band	WCDMA	B1/B5/B8	
Frequency Band	CDMA	CDMA2000 1X/EVDO	
Protocols for the control of the module	AT COMMAND / OUTPUT PARA PARSER	USB PORTS	RARE GROUP BUTTON / EXPRESS DISENGAGEMENT

La siguiente tabla muestra las conexiones entre el ESP32 y el chip T-A7670G:

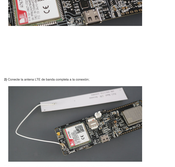
T-A7670G	ESP32
Tosse	GPIO 26
Rx	GPIO 27
Energía	GPIO 4

Para comunicarse con la tarjeta microSD, necesita el protocolo de comunicación SPI. Estas son las GPIO utilizadas:

Tarjeta MicroSD (ranura TF)	ESP32
MOSI	GPIO 23
SCLK	GPIO 18
CS	GPIO 13
MISO	GPIO 19

LILYGO TTGO T-A7670G ESP32 Pinout

La siguiente imagen muestra el pinout de placa T-A7670G ESP32:



Comandos AT

Los comandos AT se utilizan para controlar los módemos, como en el caso del T-A7670G. En el caso del LILIGO, se envían los comandos AT a través del protocolo de comunicación en serie. Luego el módem responde también a través de la comunicación en serie.

Hay cuatro tipos de comandos AT: probar, test, establecer, ejecución. Puede encontrar la lista completa de comandos AT para el T-A7670G en el siguiente enlace:

- Comandos A7670G / A7670E / A7670SA - A7670 AT

Aquí hay algunos comandos AT:

- Comprueba la comunicación con el módem: AT
- Establecer el nivel de brillo del LED: AT+CBLDR=2
- Comprueba el estado del registro del operador: AT+COPS?

Bibliotecas

Con la biblioteca adecuada, el ESP32 se comunica con el chip T-A7670G enviando comandos AT a través de la comunicación en serie. No necesitas una biblioteca, simplemente puedes establecer una comunicación en serie con el módulo y comenzar a enviar comandos AT.

Sin embargo, puede ser más práctico usar una biblioteca. Por ejemplo, la biblioteca TinyGSM sabe que comandos enviar para crear mensajes de texto/para, ATI y lo envuelve en una interfaz sencilla de Arduino Class.

IMPORTANTE: para este tutorial no usaremos la biblioteca oficial TinyGSM de vortmanmaya. Usaremos el tenedor TinyGSM de lewisfw. Si ya tiene instalada la biblioteca TinyGSM by vortmanmaya, debe eliminarla para que esta nueva información

Instalación de la biblioteca TinyGSM fork

La biblioteca TinyGSM fork by lewisfw es más disponible para instalar a través del Administrador de Bibliotecas de Arduino, por lo que necesita copiar los archivos de la biblioteca a la carpeta de bibliotecas de Arduino. Alternativamente, en la IDE de Arduino, puede ir a **Sketch > Incluir Biblioteca > Agregar biblioteca .zip** y seleccione la biblioteca que se acaba descargar y continuación.

- Haga clic para descargar TinyGSM fork (master, .zip)

Preparando la placa LILYGO T-A7670G ESP32

Esta sección está adirlle tenpiero sexo SIM. Necesitas una tarjeta SIM para 4G y GPRS. Para usar 4G y GPRS necesitas una tarjeta SIM para que pueda obtener datos. Esta puede ser costosa en algunos países por lo que podría ser prohibitivo dependiendo de cuánto puedes obtener un plan de datos en su país.

Donde vivimos (Portugal), podemos obtener una tarjeta SIM con un plan de datos, llamadas y SMS por aproximadamente 10 euros/mes. Si tiene una tarjeta SIM de Internet/Internet.com, puede usarla para conectarse a Internet y enviar mensajes de texto a través de Internet. Tambien hay empresas especializadas en tarjetas SIM para proyectos de IoT.

El Fraidmente, si no es el caso también puedes conectarlo a un ordenador, debe instalar el controlador de puerto USB CH9102. Windows y Mac OS.

Detalles de APN

Para conectar su tablero SIM a Internet, necesita tener los **detalles de APN** de su proveedor de planes telefónicos. Necesitas el nombre de dominio, el nombre de usuario y la contraseña.

En mi caso, vamos usando Vodafone Portugal. Si busca la configuración de GPRS APN seguida del nombre del proveedor de su país/ciudad, por lo general, se encuentra. Por ejemplo, APN Vodafone **Portugal**, generalmente puede encontrar en un foro o en su sitio web toda la información que necesita.

Puede ser un poco complicado encontrar los detalles o no utilizar un proveedor conocido. Por lo tanto, es posible que deba ponerse en contacto con ellos directamente.

Programación LILIGO T-A7670G ESP32 T-A7670G

El Fraidmente, si no es el caso también puedes conectarlo a un ordenador, debe instalar el controlador de puerto USB CH9102. Windows y Mac OS.

En mi caso, vamos usando Vodafone Portugal. Si busca la configuración GPRS APN seguida del nombre del proveedor de su país/ciudad, por lo general, se encuentra. Por ejemplo, APN Vodafone **Portugal**, generalmente puede encontrar en un foro o en su sitio web toda la información que necesita.

Puede ser un poco complicado encontrar los detalles o no utilizar un proveedor conocido. Por lo tanto, es posible que deba ponerse en contacto con ellos directamente.

Programación LILIGO T-A7670G ESP32 T-A7670G

Antes de conectar su placa al ordenador, debe seguir los siguientes pasos:

1) Inserte una tarjeta nano SIM.

2) Conecte la antena LTE de banda completa a la conexión.

3) Para programar su placa, debe alternarla utilizando el switch USB tipo C.

Programación del Ciclo HTTPS Parónico ESP32

Este primer tutorial que ejecutaría en la tableta es cuando para comprobar todas las conexiones trabajar el módulo, se conectaría a la red WiFi y hará una actividad HTTPS GET a un sitio web.

1. Copia el siguiente código en la IDE de Arduino:

```
  Serial.println("Enable network failed");
  }
delay(500);

String ipAddress = modem.getLocalIP();
Serial.println("Network IP:", Serial.println(ipAddress);

// Initialize HTTPS
modem.https_begin();

  HTTPSGetRequest();

void loop() {
  // Debug AT
  if (Serial1.available()) {
    Serial.write(Serial1.read());
  }
  if (Serial.available()) {
    Serial1.write(Serial.read());
  }
  delay(1);
}

#ifndef TINY_GSM_FORK_LIBRARY
  #error "The correct library was NOT found. You must install TinyGSM-fo
#endif
```

Ver código en processar

2. Inserta un pin de su tarjeta SIM, si lo tiene. En mi caso, desactivé el pin.

```
#define SIM_PIN ""
```

3. Inserte los detalles de su APN de red en la siguiente línea:

```
#define NETWORK_APN "x"
```

Por ejemplo, ver mi caso:

```
#define NETWORK_APN "net2.vodafone.pt"
```

4. Haga a **Herramientas** > **Placa** y seleccione **Módulo de desarrollo ESP32**
5. Finalmente, suba el código a la tableta.

Ahora al monitor serie a una velocidad de transmisión de 115200. Presione el botón RST a tiempo para reinicializar la placa. Espera algún tiempo hasta que la placa se conecte a la red (en mi caso, puede tardar de 1 a 2 minutos).

Deberías conseguir algo similar en el Arduino IDE Serial Monitor.

```
rst:0x1(POWERON_RESET),boot:0x13(SPI_FAST_FLAH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1184
ho 0 tail 12 room 4
load:0x40078000,len:10396
load:0x40080400,len:6364
entry 0x40080688
Enable network failed
Network IP:, 0.0.0.0
AT+CPIN?
+CPIN: READY
AT+CFUN=1
AT+CGDCONT?
+CGDCONT: 1,"IP","net2.vodafone.pt","0.0.0.0",0,0
AT+CGAUTH=1,0,,
AT+CGACT?
+CGACT: 1,1
AT+CGDATA="PPP",1
+CGDATA: "PPP",1
AT+CGPADDR=1
+CGPADDR: 1,10.151.7.6
```

RANDOM NERD TUTORIALS

Cómo funciona el código

Estamos en estado depuro a la puerta redirecciona del código.

Primero, inserta el pin de la tarjeta SIM, los detalles de APN y la URL de solicitud deseada (solo asegúrese usando una URL de muestra de GitHub que muestra una prueba la conexión).

```
// sec_ID card pin, if any
#define SIM_PIN ""

#define NETWORK_APN "x"

const char *request_url[] = {
  "https://gist.githubusercontent.com/Xu7Sun/aaaaaaaa:7b8d157a41a2477c2
};
```

Luego, necesitamos definir el módulo que está usando. La biblioteca es compatible con muchos módulos diferentes. Para usar el A7670G, incluya la siguiente línea (asegúrate de que para otros módulos TinyGSM):

```
#define LILYGO_T_A7670
```

Incluye el TinyGSM biblioteca de terceros por delante:

```
#include <TinyGsmClient.h>
```

A-7670G Pinout

Las siguientes líneas establecen la velocidad de transmisión y la salida del módulo:

```
#define MODEM_BAUDRATE                (115200)
#define DUMP_AT_COMMANDS              (1)
#define MODEM_TX_PIN                  (26)
#define MODEM_RX_PIN                  (27)
// Keep power when running both modems at the same sequence.
#define BOARD_POWERON_PIN             (4)
#define BOARD_ADC_PIN                 (36)
// Modem is used by default, it must be set to false if you want to use the modem to supply pow
#define BOARD_POWERON_PIN             (12)
#define MODEM_RING_PIN                (33)
#define MODEM_RESET_PIN               (5)
#define BOARD_BAT_ADC_PIN             (35)
#define MODEM_RESET_LEVEL             (1U)
#define BOARD_SOLAR_ADC_PIN           (34)
#define BOARD_LED_PIN                 (13)
#define BOARD_LED_ON_PIN              (HIGH)
#define MODEM_WAKEUP_PIN              (0)
#define SerialAT                      Serial1
#define MODEM_PWRKEY_IDLE_LEVEL       (HIGH)
#define MODEM_LDO_ENABLE_LEVEL        (1U)
```

Comunicación en serie

Inicie la comunicación en serie con el Monitor de Serie para la depuración y el módulo para enviar comandos AT:

```
Serial.begin(115200);
Serial.println("Hi!");
SerialAT.begin(115200, SERIAL_8N1, MODEM_RX_PIN, MODEM_TX_PIN);
```

Alimenta el módem

En el setup() bloque, comprobamos todo el siguiente procedimiento para encender y arrancar el módem:

```
pinMode(BOARD_POWERON_PIN, OUTPUT);
digitalWrite(BOARD_POWERON_PIN, HIGH);

// Set modem reset pin, reset modem
pinMode(MODEM_RESET_PIN, OUTPUT);
digitalWrite(MODEM_RESET_PIN, !MODEM_RESET_LEVEL); delay(100);
digitalWrite(MODEM_RESET_PIN, MODEM_RESET_LEVEL); delay(2600);
digitalWrite(MODEM_RESET_PIN, !MODEM_RESET_LEVEL);

pinMode(BOARD_PWRKEY_PIN, OUTPUT);
digitalWrite(BOARD_PWRKEY_PIN, LOW);
delay(100);
digitalWrite(BOARD_PWRKEY_PIN, HIGH);
delay(1000);
digitalWrite(BOARD_PWRKEY_PIN, LOW);

// Turn on the modem boot pin
Serial.println("Start modem...");
```

Espera hasta que el módulo T-A7670 comience a responder a los comandos AT. Ir que significa que se está ejecutando correctamente.

```
int retry = 0;
while (!modem.testAT(1000)) {
  Serial.print("-");
  if (++retry > 30) {
    digitalWrite(BOARD_PWRKEY_PIN, LOW);
    delay(100);
    digitalWrite(BOARD_PWRKEY_PIN, HIGH);
    delay(1000);
    digitalWrite(BOARD_PWRKEY_PIN, LOW);
    retry = 0;
  }
}

Serial.println();
```

Después, comprueba si la tarjeta SIM está insertada correctamente y desbloqueada con el SIM_PIN si es necesario:

```
// Check if the SIM is inserted
int simRetry = 0;
while (int ret = SIM_READY
  simRetry = modem.getSimStatus();
  Serial.print("SIM ");
  switch (ret) {
  case SIM_ERROR:
    Serial.println("SIM card online");
    break;
  case SIM_LOCKED:
    Serial.println("The SIM card is locked. Please unlock the SIM card
    int pinRet = modem.simUnlock(SIM_PIN);
    break;
  default:
    break;
  }
  delay(1000);
}

Serial.println();
```

Establecer modo de red

Configure el modo de red y el APN para establecer una conexión a Internet:

```
if (!modem.setNetworkMode(MODEM_NETWORK_AUTO)) {
  Serial.println("set network mode failed");
}
String mode = modem.getNetworkMode();
if (!mode.toint() || mode.toint() != 38) {
  Serial.println("mode");
}

bool ipConnRet = modem.setNetworkApn(
  "AT+CGDCONT=1,\"IP\",\"%s\",\"0.0.0.0\",0,0,0\r\n", NETWORK_APN, "\"\"");
Serial.printf("set network apn %s\n", ipConnRet ? "ok" : "fail");

#endif
```

Luego, ejecuta en bucle que comprueba la calidad de la señal y se detengo cuando la placa haya establecido una conexión a Internet exitosa.

```
// Check network registration status and network signal status
int sq;
Serial.print("Wait for the modem to register with the network.");
RegStatus status = REG_NO_RESULT;
while (status == REG_NO_RESULT || status == REG_SEARCHING || status == R
  status = modem.getRegistration();
  switch (status) {
  case REG_UNREGISTERED:
  case REG_SEARCHING:
    sq = modem.getSignalQuality();
    Serial.printf("[%lu] Signal Quality:%d", millis() / 1000, sq);
    delay(300);
    break;
  case REG_DENIED:
    Serial.println("Network registration was rejected, please check if
    return;
  case REG_OK_HOME:
    Serial.println("Online registration success\n");
    break;
  case REG_OK_ROAMING:
    Serial.printf("Network registration successful, currently is roam
    break;
  default:
    Serial.printf("Registration Status:%d", status);
    delay(1000);
    break;
  }
}

Serial.println();

Serial.print("Registration Status:%d", status);
delay(180);
```

Obtener información sobre el módem

Puede ejecutar estos comandos para imprimir la información del módem en el Monitor Serie.

```
String modInfo;
if (modem.getModemInformation(modInfo
  Serial.print("Querying SE system information");
  Serial.println(modInfo);
}

if (!modem.isNetworkActive()) {
  Serial.println("Enable network failed");
}

delay(500);

String ipAddress = modem.getLocalIP();
Serial.print("Network IP:", Serial.println(ipAddress);
```

Solicitud de HTTPS GET

Para hacer una solicitud HTTPS GET a la Solicitud_URL variable definida al principio del boceto, necesita incluir un cliente HTTPS:

```
modem.https_begin();
```

Finalmente, llama a HTTPSGetRequest() para hacer la solicitud HTTPS GET. Esta función imprimirá el código HTTP, el encabezado HTTP y el cuerpo HTTP de la solicitud:

```
  HTTPSGetRequest();
```

HTTPSGetRequest()

Esta función realiza una llamada URL. Luego hace 3 intentos para OBTENER una respuesta HTTPS válida de un sitio web. Imprime algunas información de depuración en el Monitor de Serie, como el código HTTP de una solicitud HTTP y el cuerpo HTTP de la respuesta.

```
void HTTPSGetRequest() {
  Serial.printf("\n\n\t\t\t[ %s ]\n", request_url[ / sizeof(*request_url[0]) - 1
  int rc = 0;
  while (rc == 0) {
    int ret = modem.https_begin();
    Serial.printf("begin:%d\n", ret);
    int tryCount = 60;
    while (--tryCount) {
      int ret = modem.https_set_url(request_url[0]);
      Serial.printf("Failed to request:%d", ret);
      // Debug
      // modem.sendAT("+CSSLCFG=\"sslversion\",0,3");
      int mode = modem.getNetworkMode();
      delay(1000);
      continue;
    }
    int tryCount = 60;
    int ret = modem.https_get();
    if (ret != 1
      Serial.printf("Failed to request:%d", ret);
      continue;
    }
    int tryStatus = 60;
    int status = 0;
    bool isConnect = false;
    while (--tryStatus) {
      Serial.printf("Failed get status | error code :%d\n", status);
      Serial.printf("tryStatus:%d\n", tryStatus);
      delay(1000);
      continue;
    }

    // Get HTTPS header information
    String header = modem.https_header();
    Serial.printf("HTTP Header :%s\n", header.length());
    Serial.printf("header:%s\n", header);
    delay(1000);

    // Get HTTPS response
    String body = modem.https_body();
    Serial.printf("HTTP body :%s\n", body.length());
    Serial.printf("body:%s\n", body);
    delay(3000);
    break;
  }
  Serial.printf("------------------------------------------------------------");
}
```

Consejos de solución de problemas

1. El código no se compila

IMPORTANTE: para este tutorial, no estamos usando la biblioteca oficial TinyGSM de vshymanskyy. Estamos usando el tenedor TinyGSM de teamunix. Si ya tiene instalada la biblioteca TinyGSM by vshymanskyy, debe eliminarla.

Instalación de la biblioteca TinyGSM fork

La biblioteca TinyGSM fork la encontrará en este enlaces como una bifurcación a través del Administrador de Bibliotecas de Arduino, por lo que necesita copiar los archivos de la biblioteca a la carpeta correcta. Si tiene problemas con la instalación de la biblioteca, consulte nuestra guía de solución de problemas **Instalar Bibliotecas** > **Agregar Bibliotecas .zip** y seleccione la biblioteca que acaba de descargar y continuación.

Ahorra hasta un 15 % en Europa

Hertz

- Haga clic aquí para descargar TinyGSM fork (espere .zip)

2. HTTPS GET Request devuelve el código de estado 715

Si estás usando esta biblioteca HTTPS GET y el sitio web devuelve el código de estado 715 en puertos que no sean estándar, es probable que tenga un error conocido en la biblioteca. Para obtener más información sobre este error aquí.

- Error estándar en firmware SIMCOM T-A7670

Tenga en cuenta que algunos sitios web bloquean la conexión ESP32, por lo que puede devolver un error HTTP 0 (HTTP_ERR_CONNECTION) con definido en este propio foro de esta pregunta y también la publicación HTTP GET.

3. No se puede cargar el código, porque el puerto COM no aparece

Para programar su placa, debe alternativamente cuando el cable USB tiene C, si usa el otro puerto USB no le permitirá programar el ESP32.

También necesita instalar en su ordenador el controlador de puerto USB CH9102: Windows y Mac OS.

PANDORA COMPRAR

Terminando

En este tutorial, aprendiste cómo empezar con la placa LILYGO TTGO T-A7670G ESP32, el pinout y sus características principales. Este tutorial también se puede aplicar si está utilizando un ESP32 "normal" conectado a un módulo SIM7670G externo. Este módulo es compatible con 4G, que puede ser muy útil para proyectos de IoT y automatización del hogar. También aprendiste a conectar la placa a Internet y a hacer una solicitud HTTPS GET.

La placa LILYGO TTGO T-A7670G ESP32 también viene con una ranura para tarjetas microSD que puede ser útil para proyectos de registro de datos o para guardar ajustes de configuración. Además, viene con un soporte de batería y un circuito de carga de batería para usar con paneles solares. Por lo tanto, es adecuado para usar en ubicaciones remotas. Sin embargo, aún no he experimentado con el circuito de la batería. Para ver más ejemplos, puedes explorar la página oficial de LILYGO T-A76XX de GitHub.

¿Tienes una de estas tablas? ¿Qué opinas? Hágamos saber en los comentarios a continuación.

También te pueden gustar las siguientes guías:

- ESP32: Hacer solicitudes HTTPS usando la tarjeta SIM - LILYGO T-SIM7000G
- Primeros pasos con LILYGO T-SIM7000G ESP32 (LTE, GPRS y GPS)
- LILYGO T-SIM7000G ESP32: Obtenga datos GPS (latitud, longitud, altitud y más)
- ESP32 Publicar datos en la nube sin Wi-Fi (TTGO T-Call ESP32 SIM800L)

Obtenga más información sobre el ESP32 con nuestros recursos:

- Aprende ESP32 con Arduino IDE
- Construya servidores web con ESP32 y ESP8266
- Proyectos y tutoriales gratuitos de ESP32

Gracias por leer.

HOGAR INTELIGENTE con Raspberry Pi, ESP32, ESP8266 [Libro electrónico]

Aprenda a construir un sistema de automatización del hogar y cubriremos los siguientes temas principales: Node-RED, Node-RED Dashboard, Raspberry Pi, ESP32, ESP8266, MQTT y base de datos InfluxDB **DESCARGAR »**

Recursos recomendados

Construye un sistema de automatización del hogar desde cero » Con Raspberry Pi, ESP8266, Arduino y Node-RED.

Automatización del hogar usando ESP8266 eBook y curso en vídeo » Construir proyectos de IoT y automatización del hogar.

Proyectos paso a paso de Arduino » ¡Construye 25 proyectos de Arduino con nuestro curso, incluso sin experiencia previa!

Qué leer a continuación...

ESP32/ESP8266 PWM con MicroPython - LED atern

ESP32/ESP8266: Lecturas de temperatura y humedad DHT en pantalla OLED

ESP32 HTTP GET y HTTP POST con Arduino IDE (JSON, URL codificada, texto)

¿Te gustó este proyecto? ¡Manténgase actualizado suscribiéndose a nuestro boletín!

Tu dirección de correo electrónico — SUSCRIBIRSE

10 pensamientos sobre "Primeros pasos con LILYGO TTGO T-A7670G ESP32 (4G, GPRS y GPS)"

Ulrik
13 de febrero de 2025 a las 3:00 p. m.

Muy bien, acabo de completar una alarma doméstica para mi hijo con este tablero. Pero lo programé en micropython ya que ardunio es muy aburrido :). Está aislado de Internet, por lo que los comandos se envían por SMS o acceso local a una página web. Después de ajustar muchos detalles con tiempos de espera y traducción del formato de SMS, está funcionando. Así que un TinyGSM sería muy bueno para simplificar esa parte. Tengo un problema cuando hay una caída de energía que generalmente la placa se reinicia, a veces sobrevive al cambio a la batería. ¿Alguien sabe cómo arreglar eso?

responder

Rui Santos
14 de febrero de 2025 a las 14:08

Hola.
Desafortunadamente, no sé cómo solucionar el problema.
No tuvimos tiempo de probar la placa en modo "fuera de la red".

¿Usaste alguna biblioteca para programarlo en MicroPython?
Saludos.

responder

Ulrik
14 de febrero de 2025 a las 14:48

Usé sus instrucciones para la placa SIM anterior para entender cómo funciona la placa y el código arduino en Github que los fabricantes de la placa han publicado y le pedí a chatGPT que lo tradujera a micropython, con mucha depuración y mucho trabajo con los comandos AT, tiempos de espera para las llamadas y el uso del conjunto de caracteres correcto. Creo que TinyGSM se está encargando de esas cosas. Chatgpt me ayuda mucho, especialmente para corregir la sintaxis, pero también hizo muchos problemas ya que no conoce los límites de micropython y que las bibliotecas no son idénticas a las últimas python normales. Creo que si alguien más experimentado en la organización de Python sería fácil hacer una biblioteca. Le pedí a chatGPT que hiciera eso, pero fue un gran trabajo cambiar mi código de trabajo, así que no dí ese paso para este proyecto.

responder

Ulrik
13 de febrero de 2025 a las 3:02 p. m.

Me refería a una biblioteca TinyGSM para micropython

responder

Pierre Pennings
13 de febrero de 2025 a las 7:43 p. m.

¿Cuáles son los beneficios específicos de usar esta placa T-A7670G en lugar de la T-SIM7000G?

responder

Rui Santos
14 de febrero de 2025 a las 2:10 p. m.

Hola.
Tiene 4G.
Saludos.
Rui

responder

Sergio
13 de febrero de 2025 a las 9:45 p. m.

Hola, chicos. Felicidades por tu explicación. Solo me gustaría saber por qué elegiste este tenedor de lib en lugar del original. Tal vez te guste algún problema con eso.

responder

Rui Santos
14 de febrero de 2025 a las 14:12

Hola.
El original no es compatible con este módulo en particular.
Saludos,
Rui

responder

Jorge
14 de febrero de 2025 a las 2:13 a. m.

parece un buen proyecto para construir, necesito terminar el que estoy trabajando ahora mismo y luego podría intentar ver qué hacer al respecto

responder

ALAUDE
14 de febrero de 2025 a las 7:01 a. m.

Hola, pude probar el módulo y funciona muy bien.
¡Excelente trabajo de parto!
La comunicación está establecida, ¿el módulo consume datos de en continuo? Si sí, ¿cuánto?
Gracias.

responder

Publicar un comentario

Sitio web

Recibir avisos de comentarios de seguimiento mediante correo electrónico.

Recibir avisos de nuevas publicaciones por correo electrónico.

Publicar un comentario

Copyright © 2013-2025 · RandomNerdTutorials.com · Todos los derechos reservados

Actualizar la configuración de privacidad

Microsoft Advertising — Titular — Regístrate ›