

interpréteur BASIC pour ESP32 et ESP8266

Programmation avec Annex WiFi RDS



Peter Neufeld (Allemagne)

Annex WiFi RDS (= *rapid development suite*) est un environnement conçu pour utiliser le langage de programmation BASIC sur des modules ESP8266. Il existe une version ESP32. Ceux qui ont programmé les µC à 8 bits d'Atmel/Microchip avec AVR-BASIC ont maintenant accès à la classe de performance 32 bits.

Puisque l'ESP32 est un ESP8266 en plus puissant, la version du BASIC pour l'ESP8266 devrait se changer facilement en une version fonctionnellement identique pour l'ESP32. Au moment de rédiger ces lignes, elle est encore en phase bêta, mais elle sera définitive en mars.

Si vous aimez développer de petits projets avec connexion réseau, vous n'aurez bientôt plus à vous battre avec l'IDE Arduino ni le C/C++. BASIC est plus simple, et gratuit pour un usage privé non commercial. Ça mérite d'être essayé ?

Annex BASIC

Annex WiFi RDS [1] est basée sur le concept original d'ESPbasic [2], auquel a coopéré Francesco Ceccarella a.k.a. *chiccioch*. Toutefois, Annex n'est pas seulement une variante BASIC de plus, mais une suite entièrement révisée qui offre des fonctions améliorées, une plus grande fiabilité et une utilisation optimisée des capacités des µC Espressif.

Et la documentation est à la hauteur.

La gamme de fonctions de l'IDE, y compris l'interprète BASIC,

Caractéristiques

Fonctions d'Annex WiFi RDS:

- IDE intégré via un serveur web, à utiliser avec un navigateur web (optimisé pour Chrome et Firefox)
- Interprète BASIC avec variables à virgule flottante (double précision) et variables de type chaîne, tableaux multidimensionnels (float et chaîne) et sous-programmes
- Présentation syntaxique avec aide contextuelle
- Serveur web et de fichiers programmable
- Prend en charge les mises à jour de microprogrammes OTA (via WLAN)
- Prend en charge les événements asynchrones (interruptions, minuteries, accès au web, UDP, etc.)
- Points d'arrêt, exécution immédiate des commandes, affichage des variables, étape unique
- Accès à toutes les broches d'E/S, 1-Wire, SPI, I²C, PWM, Servo, NeoPixel, USART
- Traitement des erreurs, chien de garde
- TCP (HTTP GET et POST)
- Communication UDP
- Envoi de courriers électroniques via le serveur SMTP SSL
- Communication AJAX, ESP-NOW, MQTT et FTP (client)
- Algorithmes de fusion IMU/AHRS 6 et 9 DOF (*Madgwick* et *Mahony*)
- Contrôleur PID (4 canaux)
- Suite d'utilitaires sous Windows (Annexe Toolkit) avec :
 - Flasheur, gestionnaire de fichiers, sauvegarde/restauration, convertisseur HTML, moniteur de port série, serveur de mise à jour OTA, console UDP et outil d'analyse IP.

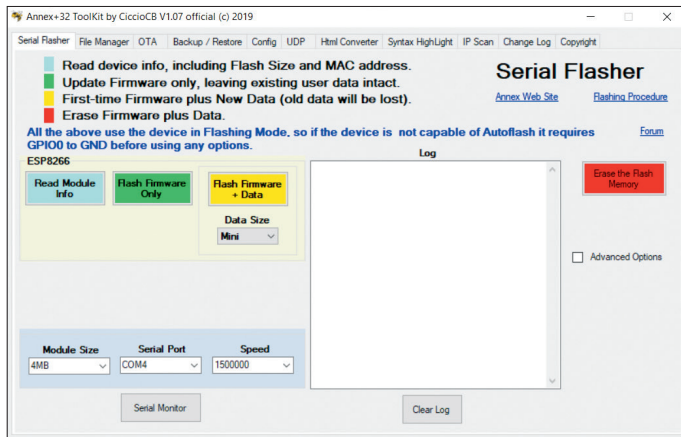


Figure 1 : L'outil Annex Serial Flasher pour le transfert du firmware vers un module ESP.

est étonnante (voir l'encadré **Fonctions**). Le logiciel complet se trouve dans le µC et fonctionne dans le navigateur web, qui affiche le site intégré. Soit le module ESP fournit un point d'accès WLAN et un serveur web sous l'adresse IP 192.168.4.1, soit vous connectez le module à votre WAN via DHCP (statique) et notez l'adresse IP reçue du routeur ou configurée à cet endroit. Grâce à ce dernier, l'ESP peut également synchroniser l'heure et la date avec un serveur de temps sur l'internet sans horloge matérielle supplémentaire.

IDE

Contrairement aux approches basées sur les compilateurs pour les µC ESP tels que l'EDI Arduino, le micrologiciel de l'annexe contient un ensemble fixe, non dépendant de la configuration, de fonctions intégrées et de pilotes pour les périphériques communs (encadré **Support matériel**). C'est à la fois une bénédiction et une malédiction : d'une part, dans le module il y a déjà tout pour adresser le matériel courant et exécuter immédiatement tout code de script utilisant ce matériel avec un support linguistique complet – d'autre part, ces fonctions consomment également de la mémoire dont vous ne disposez donc pas pour vos propres programmes.

Il y a ce qu'il faut pour communiquer avec des appareils

Installation

L'installation et la mise en service dans votre propre WLAN sont décrites en détail [4]. En voici un bref résumé :

- Le microprogramme et tout autre fichier sont transférés à un module ESP via l'interface série ou USB et sa mémoire peut être effacée au préalable.
- Connectez-vous en mode point d'accès (état de livraison) du module ESP ou intégrez mieux le module dans votre propre WLAN en tant que client WLAN.
- Définir les configurations de base telles que les paramètres du réseau local, les mots de passe, le fuseau horaire/STD, le programme Autostart BASIC, le serveur OTA, etc.
- Tout est maintenant prêt pour le chargement, l'édition, le débogage et l'exécution de programmes BASIC par navigateur.

externes au niveau du protocole à partir de vos propres sous-programmes BASIC. Il est également possible d'adresser avec votre propre code les périphériques non répertoriés ici via SPI ou I²C.

Annex WiFi RDS n'est disponible qu'en anglais. Le développeur est actif et répond rapidement aux commentaires sur le forum du projet [1]. Sur le site du projet [3], l'IDE pour les systèmes Windows disponible sous forme d'archive ZIP peut être décompressé n'importe où. L'application incluse [ANNEX-toolkit.exe](#) permet l'installation de l'interprète BASIC sur les modules ESP. Des conseils pour l'installation sont donnés dans l'encadré **Installation**.

Une fois installés sur le µC ESP le microprogramme comprenant l'interprète et l'IDE ainsi que des exemples optionnels, et que le µC est accessible via WLAN, tout peut être fait entièrement sur le PC dans la fenêtre du navigateur qui a chargé le site de l'IDE dans le µC. Le logiciel est optimisé pour Firefox et Chrome. Il est également possible de communiquer sur l'interface série en utilisant l'Annex Toolkit pendant le développement, ce qui est utile pour le dépannage. Parallèlement à la sortie des données via l'interface série, il existe également une sortie vers une zone de l'IDE appelée Wlog, qui permet le

Support matériel

Les dispositifs, actionneurs et capteurs suivants sont directement pris en charge par des commandes et des fonctions spécifiques :

- DHT11, DHT21 et DHT22 : capteurs de température/humidité
- DS18B20 : capteur de température.
- BME280 : capteur de température, d'humidité et de pression atmosphérique
- APDS9960 : capteur de distance, de lumière (intensité et couleur) et de gestes
- BNO055 : capteur d'orientation absolue
- HC-SR04 : Capteur à ultrasons (mesure de la distance)
- RTC DS1307 et DS3231 : Module d'horloge
- PCA9685 : Module PWM/SERVO
- LCD avec HD44780 via I²C ; 1/2/4 lignes de 16/20 caractères chacune

- LCD avec ST7920 avec 128 × 64 pixels monochrome
- Afficheur graphique OLED avec SSD1306 ou SH1106 ; 128 × 64 pixels monochrome
- LCD graphique avec ILI9341 ; 320 × 240 pixels en couleur 16 bits
- Afficheur à 7 segments avec TM1637 ; 4 chiffres
- Afficheur à 7 segments avec TM1638 ; 8 chiffres plus 8 LED et 8 boutons
- Afficheur à 7 segments avec MAX7219 ; 8 chiffres
- Afficheur à matrice de points avec MAX7219 ; 8 × 8 points
- LEDs couleur Neopixel WS2812
- Affichage à matrice de points avec le néopixel WS2812 ; 8 × 8 points
- Interface infrarouge ; bidirectionnelle (protocoles RC communs)

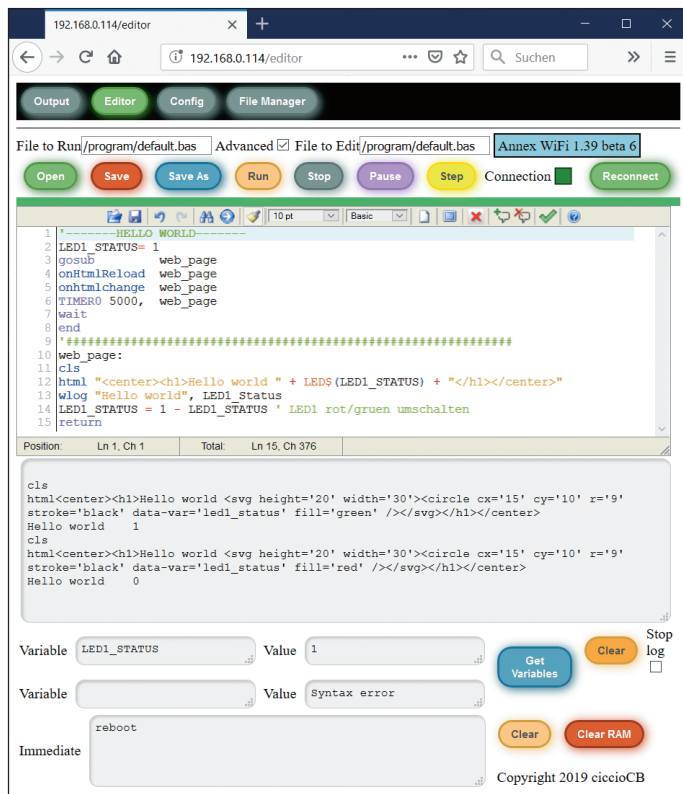


Figure 2 : L'éditeur Annex dans la fenêtre du navigateur avec l'exemple «Hello World».

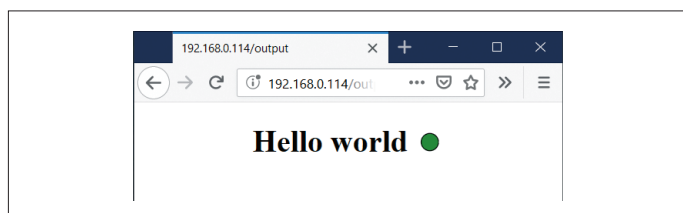


Figure 3 : Exemple «Hello World».

développement, par exemple avec un ordinateur portable sans connexion par câble au module ESP. Il est même possible de travailler avec plusieurs modules en parallèle.

Autres aspects

Pour que l'interpréteur BASIC utilise le moins possible de RAM, le script BASIC est copié de la mémoire SPIFF vers une zone désignée de la mémoire flash pour être exécuté, de sorte que

seuls la liste des lignes de programme, les identificateurs de branche et la liste des sous-programmes définis par l'utilisateur doivent être chargés dans la mémoire limitée du module. Cette méthode est légèrement plus lente que l'exécution directe en RAM, mais elle économise la RAM au profit des variables, ce qui permet d'avoir des programmes relativement longs. Le problème ne se pose pas avec l'ESP32, qui a quatre fois plus de RAM que l'ESP8266.

Une autre considération de performance est qu'un μC ESP doit toujours être capable d'exécuter de multiples activités en arrière-plan (serveur web, serveur de fichiers, etc.). Il a donc besoin de suffisamment de mémoire vive libre pour ces tâches. Les actions menées en parallèle affectent inévitablement la vitesse d'exécution de son propre code.

Selon le développeur, Annex-BASIC est largement compatible avec les variantes PC bien connues GW-BASIC et Visual Basic dont il partage de nombreux concepts, idées et syntaxes. Il existe une compatibilité avec le projet Micromite [5], une variante BASIC pour PIC32. L'interpréteur est basé sur le projet MiniBasic [6] et l'éditeur de texte sur le projet EditArea [7]. La vitesse d'exécution de l'interpréteur ne peut pas rivaliser avec un compilateur. Pour les tâches habituelles, le BASIC interprété est néanmoins bien adapté aux μC ESP, suffisamment rapide, fiable et très facile à utiliser grâce à sa nature de navigateur. Toujours selon le développeur, le BASIC sur un ESP8266 dans le «banc d'essai comparatif BASIC des années 1980» est environ deux à quatre fois plus lent que sous Micromite à l'horloge de 48 MHz. Par rapport à ce qu'offre cet IDE, ce n'est pas une mauvaise performance.

Dans l'article **Sablier avec ESP8266 et Annex WiFi RDS** de la prochaine édition d'Elektor, vous trouverez un exemple pratique instructif basé sur cet IDE attractif. La version pour ESP32 a été équipée de quelques extensions utiles pour ce SoC. (190400-02)

@ WWW.ELEKTOR.FR

→ ESP32 solo
www.elektor.fr/esp32-solo-1-wifi-bt-ble

→ ESP32 DevKitC
www.elektor.fr/esp32-devkitc

Liens

- [1] Annex WiFi RDS : <https://sites.google.com/site/annexwifi/home>
- [2] ESPbasic : www.esp8266basic.com
- [3] Téléchargements : <https://sites.google.com/site/annexwifi/downloads>
- [4] Premiers pas : <https://sites.google.com/site/annexwifi/home/first-steps>
- [5] Micromite : <http://geoffg.net/micromite.html>
- [6] MiniBasic : <https://malcolmmclean.github.io/minibasic/web/MiniBasicHome.html>
- [7] EditArea : www.cdolivet.com/editarea/