

Programación placas “Lora MakerAlc”

Al añadir un “nuevo proyecto” en PlatformIO le asignamos un nombre y seleccionamos la placa “ATmega4808”.

Es una placa genérica por lo que tendremos que indicar en el PlatformIO.ini las especificaciones particulares de la placa. En principio para la placa que vamos a usar en el maker son 3.

; Velocidad del reloj. (20 MHz para 5V o 8 MHz para 3V)

board_build.f_cpu=8000000L

; Tipo de reloj que lleva la placa (interno/externo)

board_hardware.oscillator=internal

; nº de pines (Hay varios modelos)

board_build.variant=28pin-standard

Añadiendo estas tres líneas ya podemos programar la placa.

Esta placa no tiene puerto USB por lo que el programa no podremos cargarlo directamente con un cable USB. Para cargarlo utilizaremos una placa diseñada para cargar los programas por el protocolo “UPDI” (características de la placa en la segunda parte del documento).

Es **MUY IMPORTANTE** que leáis como utilizar esta placa antes de conectarla ya que de lo contrario podéis quemar algún circuito de la placa Lora.

No disponemos de las modificaciones que hay que incluir en PlatformIO para poder subir el programa a través de la placa “Serial UPDI” por lo que utilizaremos un programa específico para esto. El programa es el Pymcuprog (mas adelante se explica como instalarlo).

Para que PlatformIO utilice Pymcuprog para la carga del programa en vez de hacerlo el mismo tenemos que añadir en el Platformio.ini la siguiente línea.

; Para subir el programa a la placa.

upload_command=pymcuprog write -t uart -u COM3 -d atmega4808 -v info --erase --filename \$SOURCE

(los parámetros se explican mas adelante)

Una vez añadidas estas 4 líneas al fichero PlatformIO.ini podremos programar la placa y cargar el programa utilizando la placa “Serial UPDI” y teniendo el Pymcuprog instalado.

Instalación de Pymcuprog

Para instalar Pymcuprog solo tenemos que teclear en la ventana de comandos la siguiente instrucción “pip install pymcuprog”. Este comando hace una llamada a un programa de Python que se encarga de hacer la instalación automática sin que tenga que intervenir el usuario.

El proceso de instalación puede devolver un mensaje de aviso que indique que se ha instalado el paquete de pymcuprog pero que no tenemos en la variable de entorno “PATH” el directorio “Scripts” de Python. La instalación devuelve la ruta completa que tiene el siguiente aspecto

`%LOCALAPPDATA%\Packages\{Python}\LocalCache\local-packages\{Python}\Scripts`

Donde pone {Python} se refiere al nombre del directorio, pero este varia en función de la versión instalada. Es fácilmente reconocible porque el nombre incluye la palabra “Python” y el numero de versión.

Añadiremos la ruta completa a la variable de entorno PATH. En mi caso he añadido “C:\Users\RGV13\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.7_qbz5n2kfra8p0\LocalCache\local-packages\Python37\Scripts”

Una vez modificada la variable de entorno reiniciamos el ordenador y probamos.

- Parámetros utilizados.

`pymcuprog write -t uart -u COM3 -d atmega4808 -v info --erase --filename $SOURCE`

Comando:

`write` → Indica que vamos a “escribir” en la placa.

Parámetro/Valor:

`-t` → “Tipo” de placa a la que nos conectamos / `uart` → Específico para SerialUPDI

`-u` → Puerto a utilizar / `COM3` (o el que corresponda)¹

`-d` → Dispositivo al que se accede / `atmega4808`

`--erase` → Borrar la memoria del dispositivo antes de escribir.

`--filename` → Fichero a subir al dispositivo / `$SOURCE` (Variable que contiene el nombre del fichero)

`-v` → Nivel de detalle de la información / `info` (Opcional. Si da algún error al subir el fichero cambiar “info” por “debug” para tener un mayor detalle del error y en que momento se ha producido)

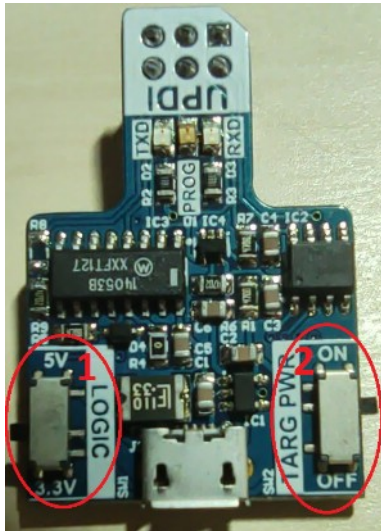
En el siguiente enlace está la información de todos los parámetros que utiliza pymcuprog

<https://github.com/microchip-pic-avr-tools/pymcuprog/blob/main/help.md>

1.- Si no sabemos en que puerto está conectada la placa lo mejor es utilizar el comando “pio device list” en la consola de PlatformIO para que nos muestre el nombre del puerto.

Configuración placa “SerialUPDI”

La placa “Serial UPDI” tiene 3 switches que permiten configurarla de distintas formas para que se adapte al tipo de placa que estemos utilizando.



- 1.- Selección de voltaje (5 o 3,3). Para la placa que vamos a utilizar 3,3V.
 - 2.- Alimentación de la placa. (On / Off). Si está en Off la placa recibe alimentación. Si está en On la placa debe tener su propia alimentación. (Independientemente de que este switch este en On o en Off la selección del voltaje debe ser el correcta ya que es el mismo que se utiliza para las señales lógicas)
 - 3.- Modo comunicación (UART / AUTO / UPDI). En nuestro caso seleccionar UPDI (Este modo nos permite cargar el programa pero no utilizar el “Monitor serial” para comunicarnos con la placa)
- Para poder utilizar el monitor serial para comunicarnos con la placa debemos cambiar el switch a la posición AUTO después de cargar el programa.

La forma de conectar el cargador SerialUPDI a la placa LoRa es la siguiente:

A close-up photograph of the LoRa module. The green PCB features the 'Taller IoT LoRa 480B' and 'MAKERALC' branding. The pin headers are labeled with pins A0 through A7, GND, and U. The 'UPDI' pin is specifically highlighted with a red circle and an arrow. The blue PCB on the right shows various components including a USB connector, a 'TARG PWR' switch, and a 'LOGIC' section with an 'XKF1127' component.

Si miramos el cargador por la parte de abajo veremos que tambien tiene una patilla con el texto “UPDI”. Ya tenemos identificadas las patillas, solo hay que conectarlas.

