

MK14ESP8266keys

A Web server to program the MK14 computer

This provides an ESP8266 based web server which can “program” the MK14 using “hex” files loaded into the ESP8266 file store, and it’s all done on the web.

Setup

Having built the mk14keyscp board you will need to program the ESP8266 module.

The main sketch file, MK14ESP8266keys.ino, must be in a folder called MK14ESP8266keys and have the other files in the same folder. These are to split out the code and hopefully make it easier to read and modify.

File name	Description
MK14ESP8266keys.ino	The initial code file
htmlcode.ino	the basic HTML code
htmlcodeprocess.ino	the more complex html code
mk14hexread.ino	decodes the data in a hex file
mk14keys.ino	sends “keys” to the MK14
wificode.ino	handles the setup and html pages for the wifi connections
wifidata.h	Some wifi related data
zimage.ino	Handles sending the favicon.ico

Libraries

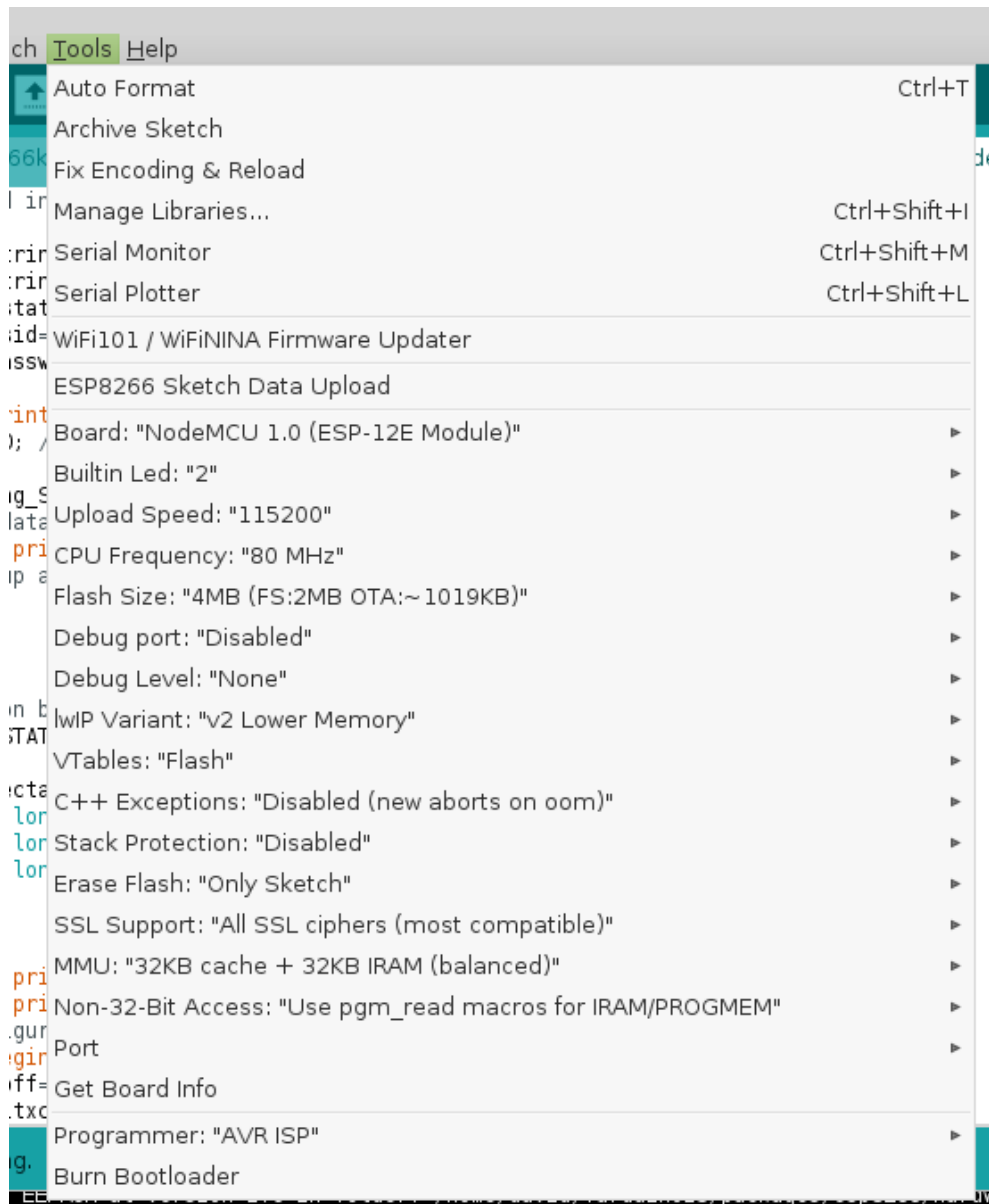
There are a number of libraries needed by the sketch, and you will need to install these before it can be built. The easiest way is to use the Tools → Manage Libraries option in the Arduino IDE.

- <ESP8266WiFi.h> provides the wifi access
- <ESP8266WebServer.h> provides the actual webserver wrapper
- <ESP8266HTTPClient.h> provides functions to handle data from browsers
- <ESP8266mDNS.h> this one allows you to access the system by name !!!
- <EEPROM.h> provides access the EEPROM to store SSID and password
- <FS.h> provides access to the SPIFFS file system
- <DoubleResetDetect.h> provides the code to check for double resets

Don't think I've missed any but let me know if I have.

The sketch should be loaded into the Arduino IDE and the correct board selected in the Tools menu.

Here is a copy of the options I used when programming my ESP8266 LOLIN v3.



You then need to put the ESP8266 into flash mode. Some development boards do this automatically. I found that when running under Windows it seemed to work, where as when running under linux (Debian 9) it did not, and I needed to press the PRG button.

The other issue I had under Linux was that the standard drivers for the CH340 USB to TTL chip used on the development board did not work. I downloaded the drivers, CH341SER_LINUX.ZIP, from http://www.wch-ic.com/downloads/CH341SER_LINUX_ZIP.html and followed the instructions to compile and then load it. My current issue is that it needs to be loaded after each reboot. Luckily reboots don't happen very often – I must fix the issue. I think if you are using a more modern linux build the problem may have been solved.

Once you have it loaded then I would suggest you select the “format the file store option” to setup the SPIFFS flash file system. If you had used one of the other file options before it maynot be compatible with SPIFFS.

There is a jumper attached to Arduino A7 and the ESP8266 A0 which can be used to tell the firmware which version of the MK14 OS it is talking to. Add jumper to use the original OS that starts with ---- --, leave open for the later versions.

When running the sketch it outputs debug information to the serial port at 115200 baud. Among other things this will show the IP address of the ESP8266. You can set a debug variable to true and a lot more info will be output from the MK14 part of the code.

You can then connect using that IP address, for the very first time you will need to connect to the WAP SSID: ESP8266MK14key, password mynetwork and then load <http://192.168.8.1> in the browser.

From there you use the system, or you can go to the set SSID page and set the SSID and password. Once this has been set you can reboot the ESP8266 and if all is well it will connect to your wifi. Be careful not to “double” boot the ESP8266 else it will revert to WAP mode.

There is a video of the basic operations at <https://www.youtube.com/watch?v=Oq6K6SKH3e0> maybe I’ll get some more details documented later.

The webpages will display the version of the MK14 OS it will be programming for. This is obtained by looking at the state of the jumper setting on the PCB.

MK14 version 1 of monitor ---- --	
Programs available	
<input type="radio"/> message.hex	displays a message
<input type="radio"/> LIFE.HEX	1 dimension life
<input type="radio"/> SiriusHardware.hex	displays SLOthIE
<input type="radio"/> ONEARM.HEX	one arm bandit
<input type="button" value="Program MK14"/> <input type="button" value="Delete File"/>	
File contents	
file contents will appear here	
Send keystrokes to MK14	
upload a new file format the file store Set SSID and password	

Main web page

MK14 version 2 of monitor 0000 00	
Programs available	
<input checked="" type="radio"/> message.hex	displays a message
<input type="radio"/> LIFE.HEX	1 dimension life
<input type="radio"/> SiriusHardware.hex	displays SLOthIE
<input type="radio"/> ONEARM.HEX	one arm bandit
<input type="button" value="Program MK14"/> <input type="button" value="Delete File"/>	
File contents	
<pre>:000F2000D1 This displays a message on the screen :100F1F0000C40D35C40031C40F36C4CA32C4C0C8B2 :100F2F00F0C40701C280C980C4FF027094F588E114 :080F3F009CEFC6FF94E790DF70 :100FA0000879796D40377739403E3F6E406D7740AC :100FB000063E7F4079373071406E38383F1F407742 :100FC000406D30403940713F406D4079796D403718 :020FD00077396F :02FFFE000F20D2 :00000001FF</pre>	
Send keystrokes to MK14	
upload a new file format the file store Set SSID and password	

Main page showing program contents

For now you can

- Load .hex files into the ESP8266 memory
- Send them to the MK14 via the “mk14keyscp PCB”
- Delete any of the hex files
- Recover any deleted files
- and format the file system.

There is a keystrokes page that allows you to send keystrokes to the MK14, either by entering them in a text box, or by pressing the buttons.

Send keystrokes to the MK14

enter the keystrokes to send to the MK14
Use 0 to 9, A to F,
G=GO, M=MEM , T=TERM, Z=ABORT, R=Reset

Send keystrokes

or click on the keys below

Science of Cambridge

go	mem	abort	a
7	8	9	b
4	5	6	c
1	2	3	d
Term	0	f	e

MK 14

reset

version 2 of monitor 0000 00

usage:
Z 0 F 2 0 M C 4 M 0 7 M 0 7 M 3 F Z 0 F 2 0 G

There is also a /dir option, no link on main page, which will list all the files in the SPIFFS file system.

Stay safe
David Allday
November 2021