

ESP AsyncPrinter

here is how to use the [ESPAsyncTCP](#)'s AsyncPrinter to print debug messages to a tcp socket at our development machine when the serial port is used for other stuff. However, i had to learn that this method where we keep the socket connection open and just write messages to it will interfere with other libraries needing a tcp connection, so i am not so sure if this really was a good solution. however, i will still keep the code here just in case.. this can of course also be used for other stuff. In my case it caused troubles with using the AsyncMQTT library in parallel.

first we need to install the library:

```
cd <arduino directory>/portable/sketchbook/libraries/  
git clone https://github.com/me-no-dev/ESPAsyncTCP.git
```

for my development machine to receive those messages i need to open a socket with netcat in listening mode. this is for linux but i am sure there are tools like that in windows too, i just don't know them. so on your development machine run:

```
nc -l 3333
```

and leave the terminal open.

now we could integrate it as is into our code as you can see in the example [I posted on Github](#), but the problem with AsyncPrinter is, that it blocks forever waiting for a connection. This means, that our sketch won't make it past the setup process if our debug host is not online. that's not what we want. I therefore modified the AsyncPrinter source slightly by removing these two lines from both AsyncPrinter::connect() functions:

```
while(_client->state() < 4)  
    delay(1);
```

let's add AsyncPrinter and a debug() function which handles sending debug messages if we have a connection to a debug host:

```
#include <ESP8266WiFi.h>           //ESP8266 Core WiFi Library (you most  
likely already have this in your sketch)  
#include <DNSServer.h>            //Local DNS Server used for redirecting  
all requests to the configuration portal  
#include <ESP8266WebServer.h>     //Local WebServer used to serve the  
configuration portal  
#include <WiFiManager.h>          //https://github.com/tzapu/WiFiManager  
WiFi Configuration Magic  
#include <AsyncPrinter.h>  
  
bool enableOnlineDebugging=true;  
const char *debuggerHost="192.168.168.48";  
  
AsyncPrinter ap;  
WiFiManager wifiManager;
```

```
bool debugOnline=false;
bool firstround=true;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  wifiManager.autoConnect();
  wifiManager.startWebPortal();
  ap.connect(debuggerHost,3333);
  if(enableOnlineDebugging){
    //block for maximum 5 seconds trying to reach the debugger
    for ( int i = 0; i < 10 ; i++){
      if(ap.connected()){
        debugOnline=true;
        ap.println("setup done");
        break;
      } else {
        delay(500);
      }
    }
  }
}

void debug(const char *message){
  if(debugOnline){
    ap.println(message);
  } else {
    Serial.println("debugger is not online");
  }
}

void loop() {
  // put your main code here, to run repeatedly:
  wifiManager.process();
  if(firstround){
    debug("hello world");
    firstround=false;
  }
}
```

you should now see "setup done" and "hello world" on your terminal with nc running.

From:

<http://wiki.psuter.ch/> - pswiki

Permanent link:

http://wiki.psuter.ch/doku.php?id=esp_asyncprinter

Last update: **16.03.2018 05:42**

