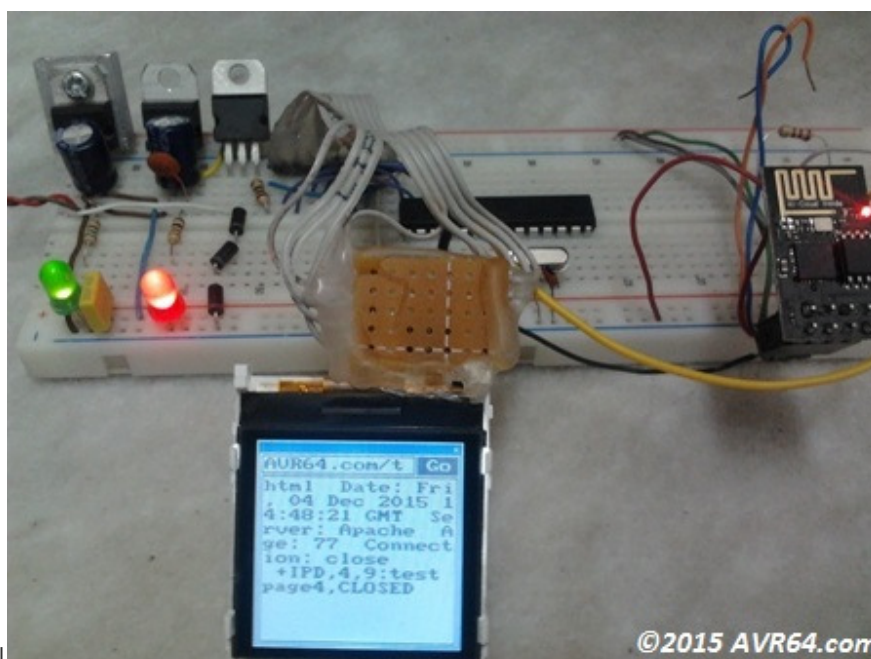


1394/09/14



این پروژه یک مرورگر ساده ی

اینترنت (Internet browser) با استفاده از ماژول Wi-Fi وای فای ESP8266 و میکروکنترلر ATmega8 از خانواده ی AVR می باشد. پروژه ی مرورگر با AVR از نمایشگر رنگی موبایل نوکیا (NOKIA) به شماره ی i6610 بهره می گیرد. البته سایر LCD های موبایل که از چیپ کنترلر pcf8833 استفاده کنند نیز توسط این پروژه راه اندازی می شود. این پروژه از مد Station ماژول ESP8266 استفاده می کند و ماژول در نقش یک Client بسته شده که با پروتکل TCP یک صفحه به آدرس <http://www.avr64.com/t.htm> را باز کرده و محتوای آن را علاوه ی هدرهایی که توسط سرور فرستاده می شود نمایش می دهد. البته این مرور گر فوق العاده ساده بوده و به دلیل حافظه کم میکرو ATmega8 بخش تفسیر Tag های HTML در آن تعبیه نشده و کل محتوای صفحه را بصورت متنی نمایش می دهد. همچنین امکان نمایش عکس و سایر فایل ها نیز وجود ندارد. هدف از انجام این پروژه آموزش کار با ماژول ESP8266 در مد کلاینت و نحوه ی ارتباط با وب سرور (Web server) و نیز نحوه ی کار با LCD های رنگی سریال است.

در این پروژه از یک آرایه ی 500 بایتی برای ذخیره ی اطلاعات دریافتی از Server استفاده شده است. Baud میکرو نیز بر روی 115200 تنظیم شده و با این سرعت با ماژول ESP8266 ارتباط برقرار می کند. در این پروژه و نیز در تمام پروژه هایی که کمی پیچیده تر می شوند گاهی اوقات بخش هایی از برنامه که اصلاً ربطی به هدف اصلی ندارد باعث مصرف حافظه میکرو و نیز صرف وقت زیاد می شود. به طو مثال در این پروژه هدف ما اتصال به اینترنت ونمایش و تفسیر محتواست، ولی چون همانند زبانهای برنامه نویسی سطح بالا مثل ویژول بیسیک، Java و غیره امکاناتی مثل Label، TextBox، TextViwe، EditText، Button و ... وجود ندارد مجبور می شویم که وقت زیادی را برای شبیه سازی این اشیاء صرف کنیم که نتیجه ی آن پر شدن حافظه ی میکرو و دور شدن از هدف اصلی می باشد. در این پروژه برای نمایش اطلاعات دریافتی تابعی به نام Label1 ایجاد کرده ایم که در حدود 250 خط می باشد و کار آن نمایش محتوای دریافت شده از سرور بر روی LCD می باشد. در چنین مواقعی استفاده از بردهای دارای سیستم عامل باعث می شود تا بجای صرف وقت برای نوشتن چنین توابعی از توابع پیش ساخته ی زبان برنامه نویسی

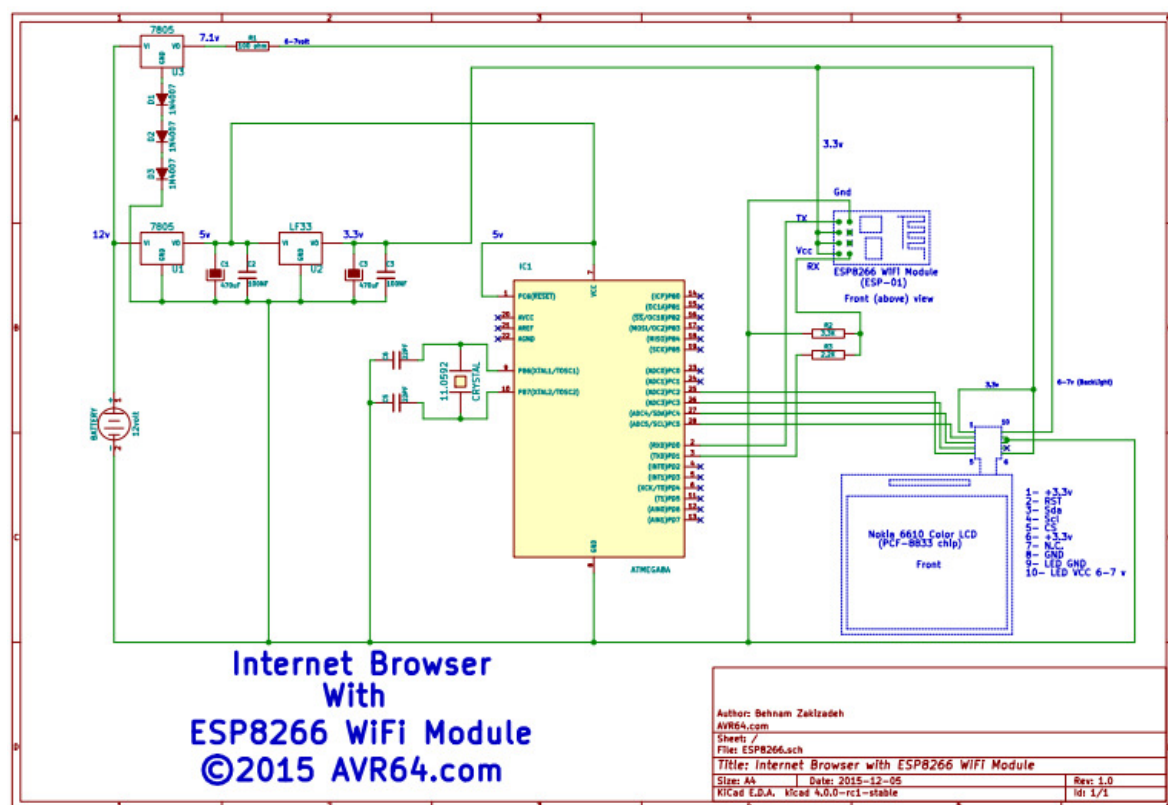
## ESP8266 Internet Browser

سطح بالا استفاده کنیم و فایل اجرایی را روی برد بار کرده و خروجی را مشاهده نماییم. به طور مثال بردهایی با سیستم عامل اندروید (Android) که دارای LCD و شبکه می باشند برای نوشتن چنین برنامه هایی سرعت کار را بالاتر خواهد برد.

در هر صورت این پروژه برای نمایش قدرت میکروکنترلر AVR و نیز آموزش کار با ماژول ESP8266 و LCD رنگی موبایل طراحی شده و با کمی تغییرات می توانید LCD موبایل را حذف کرده و از یک LCD معمولی x162 یا LCD گرافیکی TS12864 برای نمایش صفحه ی Web استفاده کنید. با حذف کتابخانه ی LCD رنگی و یا استفاده از میکرو ی حافظه ی بیشتر مثل ATmega32 و یا حتی ATmega128 می توانید تلای برای تحلیل تگ های HTML بنویسید و یک مرورگر واقعی با AVR بسازید. ممکن است کتابخانه های آماده ای نیز برای تفسیر کدهای HTML وجود داشته باشد که می توانید از آنها برای تکمیل پروژه ی خود استفاده کنید. (احتمالا بایستی عباراتی مثل html2string را جستجو کنید).

### شماتیک:

شماتیک راه اندازی ماژل وای فای ESP8266 به عنوان مرورگر اینترنت



### تحلیل برنامه:

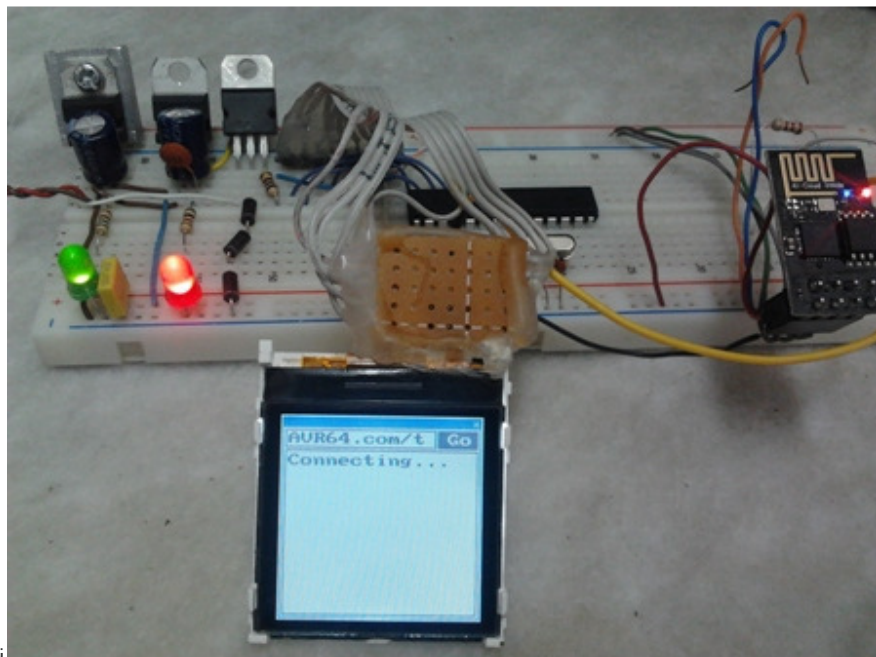
پس از دانلود این برنامه که به زبان بیسیک (BASIC) و با کامپایلر بیسکام یا بسکام (BASCOM-AVR) نوشته شده است مشاهده خواهید کرد که در ابتدای برنامه از دستورات On Urx Recive\_s و Enable Urx Recive\_s استفاده شده است. این دستورات لاینی ما حالت وقفه را برای پورت سریال فعال کرده ایم. وقتی حالت وقفه برای پورت Serial میکرو فعال می شود دیگر نیازی به چک کردن پایه RX میکرو در حلقه نیست و همینکه کاراکتری به این پایه برسد میکرو به طور خودکار به برجسبی که نام آن را به صورت دلخواه Recive\_s گذاشته ایم پرس می کند و کاراکتر دریافتی در آرایه ی 500 عنصری ذخیره می شود. توجه داشته باشید که همیشه در پروژه های سریال که با میکروهای Atmega ی هشت بیتی نوشته می شود اطلاعات موجود در پورت سریال بصورت بایت می باشد و به همین دلیل عناصر آرایه نیز از نوع Byte تعریف شده است. تعداد عناصر هم که 500 در نظر گرفته شده

## ESP8266 Internet Browser

بخاطر محدودیت حافظه SRAM میکرو می باشد و این محدودیت سبب می شود که فقط 500 کاراکتر اول صفحات وب را بتوانیم دریافت کنیم (در اینجا اهمیت RAM بخوبی نمایان می شود).

بخش های بعدی برنامه مربوط به استفاده از کتابخانه ی LCD رنگی نوکیا می باشد که گویا بوده و نیز ترسیم پنجره بر روی LCD و شبیه سازی TextBox با دستور Line و رسم خطوط و نوشته ها با دستور Lcdat. این بخش ها نیازی به توضیح خاصی نداشته و از روی کد پراحتی قابل فهم می باشد.

بخش اصلی برنامه از دستور AT+CWMODE=1 شروع می شود. دستوراتی که با AT شروع می شوند به AT Command مشهور بوده و اکثر ماژول های سریال مثل ماژول بلوتوث، GSM MODEM و Wi-Fi از این دستورات پشتیبانی می کنند. البته فقط بخش AT در آنها مشترک می باشد و هر یک دستورات خاص خود را دارند که در دیتاشیت آنها توضیح داده شده است. در این برنامه نیز با کمک چند AT Command ماژول ESP8266 را پیکرده بندی کرده و برای مد Client آماده می کنیم و سپس SSID و Password مودم Wi-Fi را که در برنامه تعریف کرده ایم برای ماژول ارسال می نماییم تا ماژول، مودم را پیدا کرده و به آن متصل شود. پس از اتصال به مودم، ماژول ما یک آی پی محلی اختصاص از مودم دریافت کرده و می تواند به شبکه جهانی اینترنت متصل شود و با دستوراتی اطلاعات بفرستد یا بگیرد. توجه داشته باشید که مودم های معمولی بیسیم که در منازل و محل کار استفاده می شود عموماً چهار عدد پورت اترنت (Ethernet) و یک آنتن Wi-Fi داشته و به سیم تلفن متصل می شوند. این مودم ها که با نام مودم ADSL شناخته می شوند وقتی روشن می شوند یک IP مثل 188.245.237.142 از شرکت ارائه دهنده خدمات اینترنت دریافت می کنند. سپس مودم به دستگاه هایی که با آن متصل می شود (مثل ماژول وای فای ESP8266) یک آی پی در کلاس C نسبت می دهد که عموماً بصورت 192.168.1.100 می باشد. معمولاً سه عدد اول ثابت بوده و عدد آخر متغیر است. نکته ای که در اینجا حائز اهمیت است اینست که در چنین شبکه هایی هر دستگاه یک آی پی محلی دارد که فقط در همان شبکه شناخته شده است و کل شبکه هم یک آی پی عمومی دارد که موقع جستجوی سایت ها آی پی عمومی مورد استفاده قرار می گیرد. به طور مثال اگر Laptop و موبایل و ماژول ESP8266 را به مودم خانه وصل کنید و وارد سایتی شوید از دید آن سایت یک آی پی از آن سایت بازدید کرده است ولی در شبکه ی محلی و خانگی سه IP محلی مختلف وجود دارد.



تا اینجا ما یک ماژول وای فای داریم که یک IP گرفته و به شبکه ی جهانی اینترنت هم متصل شده است. اما مهمترین بخش کد حدوداً از خطوط 110 تا 120 شروع می شود. در این خطوط که ابتدا و انتهای آن با علامت ستاره کامنت شده است ما یک اتصال TCP به سرور AVR64.com برقرار می کنیم و صفحه ای به آدرس AVR64.com/t.htm را دریافت می کنیم. این بخش از کد در یک حلقه ی تکرار Do...Loop نوشته شده و دائماً تکرار می شود و صفحه ی LCD را رفرش می کند. به همین علت در دستور اول با AT+CIPCLOSE=4 اتصال را بسته ایم چون مثلاً فرض کنیم در دور دوم حلقه هستیم و در دور اول اتصال باز شده و باید در ابتدا اتصال باز شده را ببندیم و مجدداً اتصال برقرار کنیم، پس عملاً در دور اول حلقه این دستور هیچ کاری انجام نمی دهد چون هنوز هیچ اتصالی را باز نکرده

ایم. در خط بعدی با دستور AT+CIPSTART=4 و سایر آرگومانهایی که در جلوی آن است یک اتصال TCP روی کانال شماره 4 (بعداً توضیح داده می شود) به سرور avr64.com و پورت 80 ایجاد می کنیم. ما در این دستور از نام دامین avr64.com استفاده کرده ایم ولی ماژول پس از ارسال این دستور، توسط سرورهای ارائه دهنده ی اینترنت به DNS سرور هدایت می شود و در آنجا یک IP مانند 5.144.130.47 دریافت می کند. همین حالا این آی پی را در مرورگر خود وارد کنید، شاید انتظار داشته باشید که سایت avr64.com را ببینید ولی نه! یک صفحه ی اخطار با نوشته ی SORRY را ملاحظه خواهید کرد. چرا؟ دلیل آن اینست که AVR64.com و درصد بالایی از سایت های اینترنتی دیگر روی سرور های اشتراکی آپلود شده اند و سرور اختصاصی ندارند. شرکت های ارائه دهنده ی خدمات هاست (HOST) معمولاً یک کامپیوتر Server پر قدرت به همراه یک IP فیکس خریداری کرده و هزاران سایت را در حافظه ی آن جای می دهند. به همین دلیل است که ما می توانیم با صرف هزینه بین 10 الی 100 هزار تومان در سال یک هاست داشته باشیم ولی عیب آن اینست که آی پی ما با هزاران سایت دیگر مشترک است. به همین روی وقتی در حال نوشتن یک مرورگر هستیم یا قرار است یک اتصال TCP با یک Web Server برقرار کنیم بایستی پس از متصل شدن به سرور اشتراکی (Shared web hosting service) مشخص کنیم که با کدام دامین موجود در این سرور کار داریم. سه دستور بعدی در سورس ما همین کار را انجام می دهند. البته قبل از ارسال سه دستور بایستی با ارسال دستور AT+CIPSEND=4,44 طول تعداد کاراکترهای آن سه دستور را مشخص کنیم. این بخش فوق العاده مهم بوده و اگر یک کاراکتر کم و زیاد ارسال کنید اتصال پس زده می شود و باید از ابتدا شروع کنید. نکته دیگر هم سرعت بالا در تایپ این دستورات است که به همین دلیل اتصال ماژول به ترمینال کامپیوتر و تایپ دستی ممکن نیست و فقط میکرو می تواند این دستورات را در کسری از ثانیه به سرور ارسال کند ( احتمالاً محدودیت زمانی برای این منظور در پروتکل TCP در نظر گرفته شده که بدانیم Sever اشتراکی به اندازه کافی سرش سلوغ است و نمیتواند منتظر بماند تا یک اتصال ده ها ثانیه طول بکشد و پس از ده ثانیه اتصال را پس می زند). در خط سرش سلوغ AT+CIPSEND=4,44 ما عدد 44 را به عنوان تعداد کاراکترهای سه دستور بعدی ارسال می کنیم (عدد 4 در این دستور و دو دستور قبلی شماره ی اتصال باز شده است. اصولاً در ماژول ESP8266 می توانیم 4 اتصال مختلف را به طور همزمان باز کنیم و این اتصالات با شماره های 1 تا 4 مشخص می شود که ما از کانال 4 استفاده کرده ایم و سایر کانال ها استفاده نشده و تاکنون در آزمایشهایمان بیشتر از یک اتصال همزمان برقرار نکرده ایم). اما عدد 44 چگونه بدست آمده؟ ما در سه خط بعدی سه دستور Print داریم که اولی رشته ای شامل متد دریافت (GET یا POST) و نام صفحه در سرور (t.htm) و شماره پروتکل (HTTP/1.1) می باشد که با احتساب فضای سفید طول آن دقیقاً 19 کاراکتر می شود. اما دو کاراکتر CR و LF که به معنای بازگشت به ابتدای خط و پایان خط می باشند همراه با هر دستور Print ارسال می شود پس ما همیشه طول هر خط را با این 2 کاراکتر جمع می کنیم که می شود 21. خط بعدی نام دامینی است که در سرور اشتراکی با آن کار داریم (www.avr64.com) این خط هم با احتساب سایر کاراکترها که در سورس مشاهده خواهید کرد 21 کاراکتر می شود و اما خط آخر یک دستور Print با یک رشته تهی است. این دستور را معادل دو کاراکتر CR و LF در نظر میگیریم. وجود یک دستور Print خالی سبب ارسال کل درخواست http به سمت web server می شود. پس تا اینجا 21+21+2=44 کاراکتر می شود که در دستور بالایی نوشته ایم. یکی از دلایلی که باید طول کاراکترهای ارسالی را از قبل مشخص کنیم اینست که سرور بداند چه تعداد کاراکتر دریافت خواهد کرد و آیا فضای کافی دارد یا نه و فضا را به درخواست ما اختصاص دهد و چون ممکن است درخواست ما خیلی طولانی باشد و برای ارسال به بسته های کوچکتر شکسته شود و Server باید بداند که تمام بسته های مربوط به درخواست ما را دریافت کرده است یا خیر. خوشبختانه تمام این کارها یعنی تبدیل درخواست به بسته و ارسال مرتب بسته ها و شماره گذاری و کشف خطا و دریافت بسته های نامرتب و بافر کردن آنها و چیدن بسته ها به ترتیب در کنار هم و درخواست مجدد از سرور در صورت بروز خطا و هزار و یک مورد دیگر توسط پشته پروتکل TCP/IP در داخل آی سی ماژول ESP8266 انجام می شود و ما فقط درخواست http می فرستیم و منتظر می مانیم تا پاسخ را بصورت مجموعه ای از بایت ها روی پایه ی RX میکرو دریافت کنیم. در خط بعدی تعداد کاراکترهای دریافت شده از سرور بر روی LCD نشان داده می شود و در نهایت با فراخوانی تابع label1 محتوای دریافت شده از Server که در آرایه 500 عنصری ذخیره شده است بر روی نمایشگر نشان داده می شود.

### تحلیل محتوای دریافت شده از سرور:

پس از اینکه درخواست دریافت یک صفحه از Web Server را برای سرور اینترنتی ارسال کردیم پاسخ زیر برای ماژول برگشت داده می شود:



```
+IPD,4,213:HTTP/1.0 200 OK
Last-Modified: Tue, 01 Dec 2015 16:46:18 GMT
Accept-Ranges: bytes
Content-Length: 9
Content-Type: text/html
Date: Sat, 05 Dec 2015 13:28:42 GMT
Server: Apache
Age: 139
Connection: close
```



```
+IPD,4,9:test page4,CLOSED
```

این خطوط توسط مازول دریافت و به پایه RX میکرو تحویل داده می شود که میکرو تمام کاراکترها را بایت به بایت در بخش از آرایه ی 500 بایتی ذخیره کرده و طی دو مرحله در نمایشگر نشان می دهد. 9 خط اول از پاسخ، یک سری اطلاعات خاص مربوط به سرور و page درخواستی و نوع سرویس دهنده و تاریخ زمان می باشد. در خط اول عبارت IPD,4,213 باینگر آنست که این بسته ی اطلاعاتی در پاسخ اتصال شماره 4 فرستاده شده و شامل 213 بایت یا کاراکتر می باشد (با احتساب کاراکترهای پایان خط که پنهان هستند). کد 200 OK هم یک کد معروف بوده و هنگامی که یک صفحه با موفقیت در Server پیدا شود و به سمت Client ارسال گردد این کد ارسال می شود (مانند کد 404 که در نبود صفحه نمایش داده می شود). بقیه ی اطلاعات در این بسته فعلاً به کار ما نمی آید و به سراغ بسته دوم می رویم. این بسته با IPD,4,9 شروع شده است. یعنی این بسته نیز در پاسخ اتصال شماره 4 بوده و شامل 9 کاراکتر است. معمولاً بسته دوم حاوی اطلاعات صفحه درخواستی یعنی t.htm هست. ما برای سادگی کار در صفحه ی t.htm فقط عبارت test page را نوشته ایم و از هیچ تگ HTML استفاده نکرده ایم. همانطوریکه ملاحظه می کنید عبارت test page در این پیام نوشته شده و بلافاصله پس از آن رشته ی 4,CLOSED آورده شده که به معنای تمام شدن اطلاعات و بستن خودکار اتصال می باشد. اگر صفحه ی درخواستی محتوای بیشتری داشته باشد تعداد بسته های دریافت شده از Server نیز بیشتر خواهد شد و تمام بسته ها ابتدا توسط مازول دریافت و خطایابی شده و بصورت مرتب شده و پشت سر هم چیده شده و سپس به میکرو ارسال می شود.

یکی از توصیه های ما برای آزمایش این پروژه و سایر پروژه های سریال و مازول ESP8266 اینست که حتماً در حین تست، پایه TX مازول را علاوه بر این که به پایه RX میکرو متصل می کنید به پایه RX یک مازول USB به سریال نیز متصل کرده و اطلاعات دریافت شده از مازول را روی یک ترمینال در کامپیوتر مشاهده کنید تا براحتی بتوانید علت کار نکردن میکرو یا کار نکردن مازول را بررسی کنید. برای محیط ترمینال می توانید کامپایلر رایگان آردوینو را دریافت و از ترمینال آن استفاده نمایید. برای مازول هم می توانید از یک برد آردوینو که پایه Reset آن را به Gnd آن متصل کرده اید (تا میکروی آن خاموش شود) به عنوان مازول USB به سریال استفاده نمایید یا از مازول های ارزان قیمت موجود در بازار استفاده کنید. در خصوص برنامه ی Bascom توجه داشته باشید که فیوزبیت را روی کریستال خارجی تنظیم کنید و حتماً از کریستال 11.0592 مگاهرتز استفاده نمایید.



**دانلود Hex + شماتیک + عکس تنظیم فیوزبیت ها + فایل PDF این صفحه**

**\*\* توجه فایل Hex بالا کامپایل شده سورس اصلی بوده و هیچ کم و کاستی ندارد. با این حال برای تست آن بایستی نام مودم ADSL خود را به avr64 تغییر داده و رمز مودم را روی 12345678 قرار دهید. برای انجام تغییرات می توانید فایل سورس اصلی (ESP8266\_Browser.bas) را که به زبان بیسیک و با کامپایلر BASCOM-AVR (بسکام یا بیسکام) نوشته شده از طریق لینک زیر به قیمت 4500 تومان خریداری نمایید. (برای نوشتن این برنامه ها وقت و هزینه ی زیادی صرف می شود، لطفاً سورههای خریداری شده را در اینترنت منتشر نکنید).**

**(توجه مهم: برای جلوگیری از مشکلات احتمالی حتماً در ابتدا Hex دمو و رایگان را دانلود کرده و از مازول پاسخ OK بگیرید و سپس نسبت به خرید سورس اصلی اقدام نمایید).**

برای خرید سورس کامل راه اندازی ماژول ESP8266 به زبان بیسیک و با کامپایلر BASCOM-AVR (بسکام/بیسکام) - قیمت 4500 تومان، ابتدا ایمیل خود را در فیلد زیر وارد نمایید و سپس دکمه خرید را فشار دهید. پس از ورود به سایت زرین پال ابتدا بانک صادر کننده کارت را انتخاب کرده و سپس روی دکمه پرداخت صورتحساب کلیک کنید و در نهایت دکمه انتقال به دروازه بانک را فشار دهید. پس از پرداخت، لینک دانلود به شما نمایش داده می شود. در صورت بروز هرگونه مشکل و یا پشتیبانی در خصوص سورس خریداری شده تماس بگیرید: ✉ ایمیل: [info@avr64.com](mailto:info@avr64.com) )

Avr64.com

[info@avr64.com](mailto:info@avr64.com)

۱۳۹۴/۰۹/۱۴