

Table of contents

1 Serial to Websocket converter

2

1 Serial to Websocket converter

RFLink gateway

RFLink RF gateway의 치명적 단점은 Serial 통신만 지원한다는 것이다. Arduino Mega의 USB포트를 이용하면 USB to Serial converter를 통해, 혹은 Arduino Mega의 Serial 포트를 통해 통신해야한다. Wireless를 지향하는 내 관점에서 이것은 치명적이었다. 할 수 없이 뭔가를 해야 했다.

Serial to Websocket

RF 신호가 올 때만 메시지를 보내는 방식 때문에 뭔가 비동기적인 방식을 사용해야 했다. MQTT 말고 다른 걸 해보고 싶어 웹소켓을 골랐다. ESP8266 보드에 웹소켓 라이브러리를 넣어서, serial 포트에서 데이터가 들어오면 웹소켓으로 broadcast 해주면 될 듯 했다. 문제는 ESP8266을 한번도, 아니 arduino 조차도 한번도 제대로 해본적이 없다는 점이었다. 그래서 초보에게 쉬운 보드를 찾아보았다. Witty Cloud Development board를 알리익스프레스에서 USD 3.45짜리 2개를 구입했다. USB to serial 컨버터 내장에, 5v를 3.3v로 변환도 해주니 ESP8266에 처음 맡겨 놓는 나에게 적당해 보였다. 이 때까지만 해도, 1 ~ 2 시간이면 되겠지 하는 자만심에 빠져 있었다.

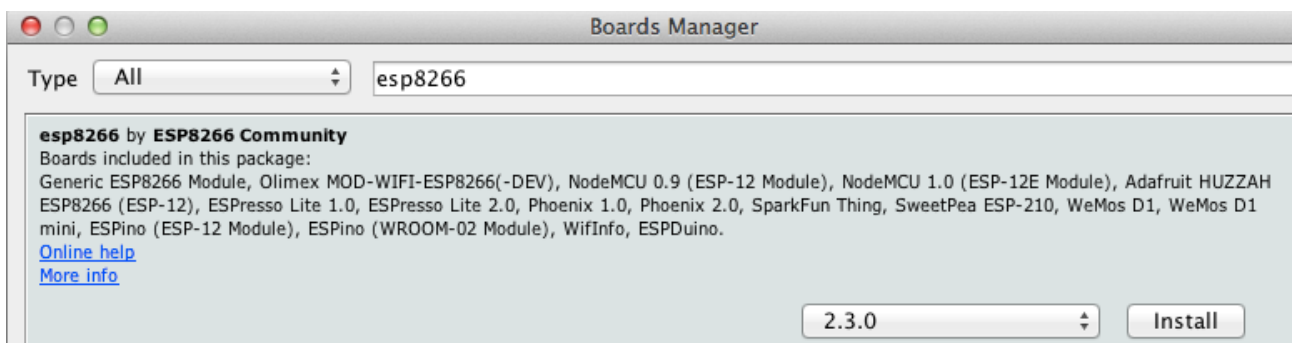
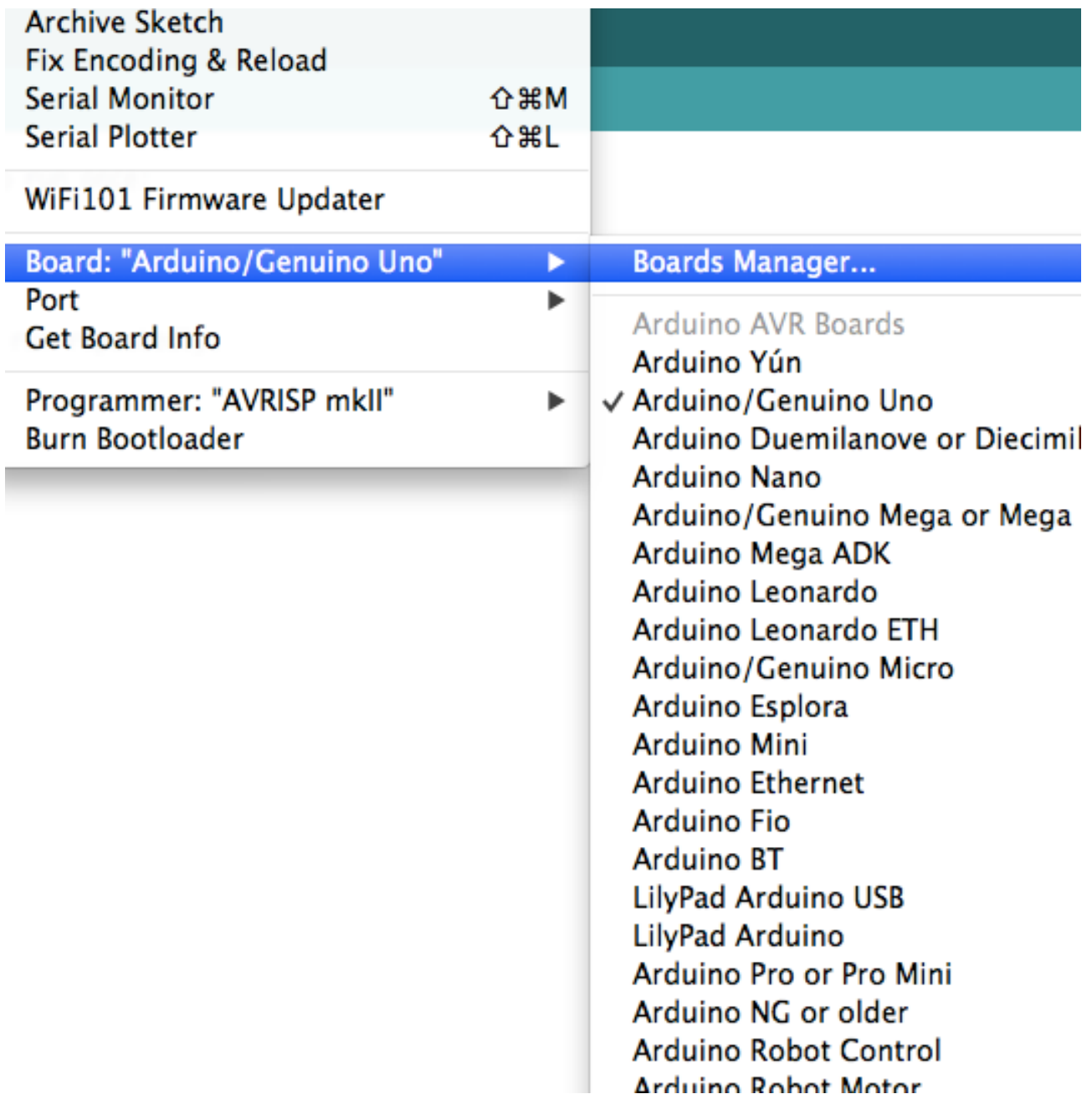
삼질의 시작 CH340G USB칩

안 그래도 알리익스프레스에서 구한 저렴한 arduino mega 2560 때문에 머리가 아픈 상황이었다. osx 10.9.5 상태에서 아무리 해도 드라이버가 인식되지 않는 것(윈도우에서는 드라이버 설치시 잘 동작해서 RFLink 펌웨어를 올릴 수 있었음)이었는데, 아차차 Witty Cloud Development board도 같은 칩을 사용한다는 걸 미리 체크하지 못한 것이었다.

현재 시점에서 CH340G의 osx용 드라이버는 V1.4인데, osx 10.9.5 에서는 이걸 깔면 안된다. V1.3을 깔아야 하는데, 이외에도 usb 케이블 길이에 따른 이슈도 있었다. 가지고 있던 스마트폰 충전용 케이블을 사용했더니 부팅 후 자동으로 종료되는 문제가 발생해서 여러번 시도해보았다. 2시간여를 끄꿍대다가 혹시나 싶어, 가장 짧은 케이블을 꽂고 했더니 그제서야 인식이 되었다. 일단 드라이버가 제대로 설치된 이후로는 arduino mega 2560도 긴 usb 케이블을 가지고서도 잘 인식이 되었다.

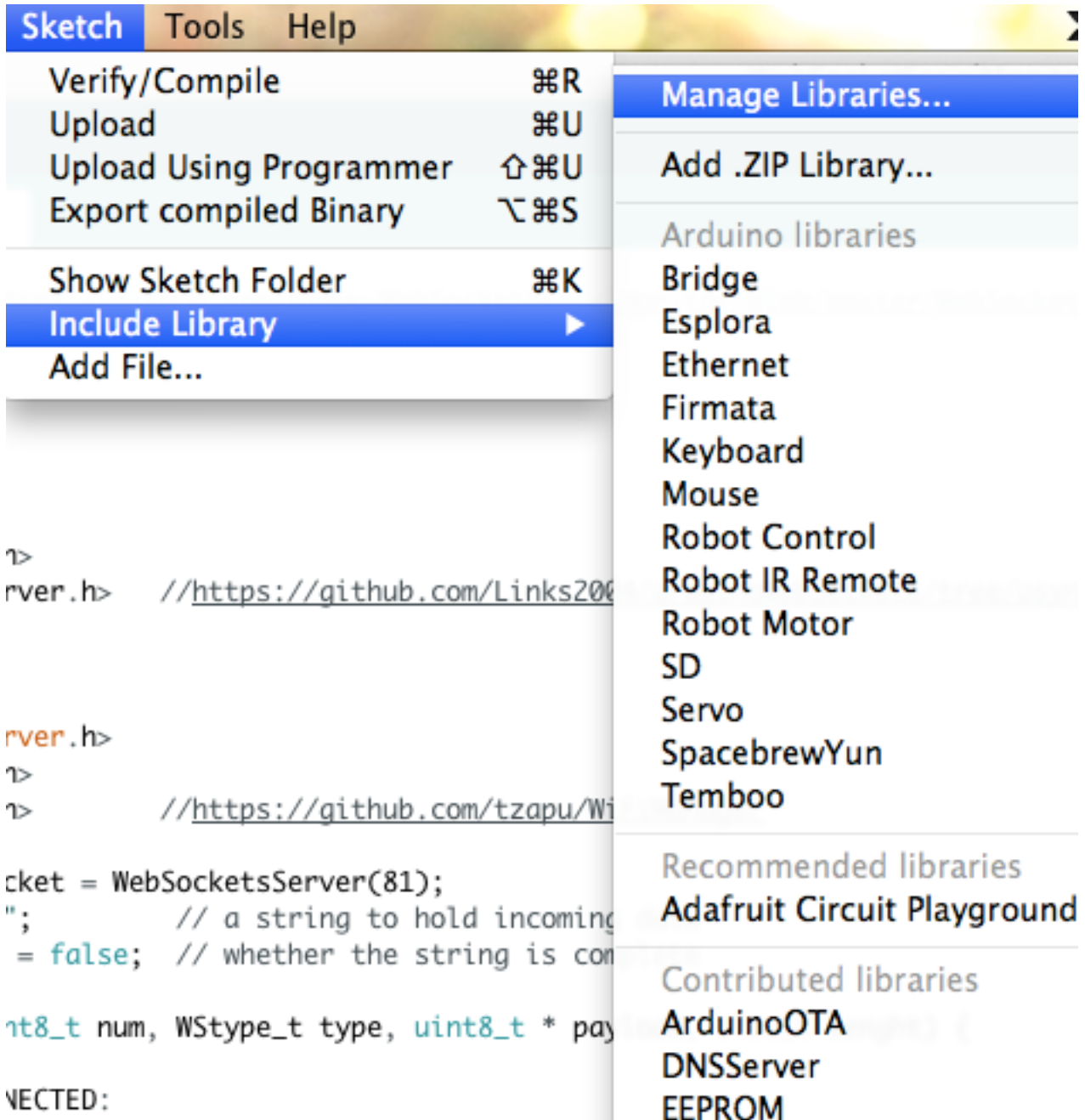
Arduino IDE로 esp8266 개발 환경 설정 및 라이브러리 설치

- Arduino IDE 다운로드 후 설치
- ESP8266용 Boards Manager 설치
 - Preference = Settings에서 Additional Boards Manager URL에 http://arduino.esp8266.com/stable/package_esp8266com_index.json 입력
 - Tool - Board - Boards Manager 클릭 후 esp8266 입력 후 해당 목록 클릭해서 버전 2.3 선택 후 설치하면 됨



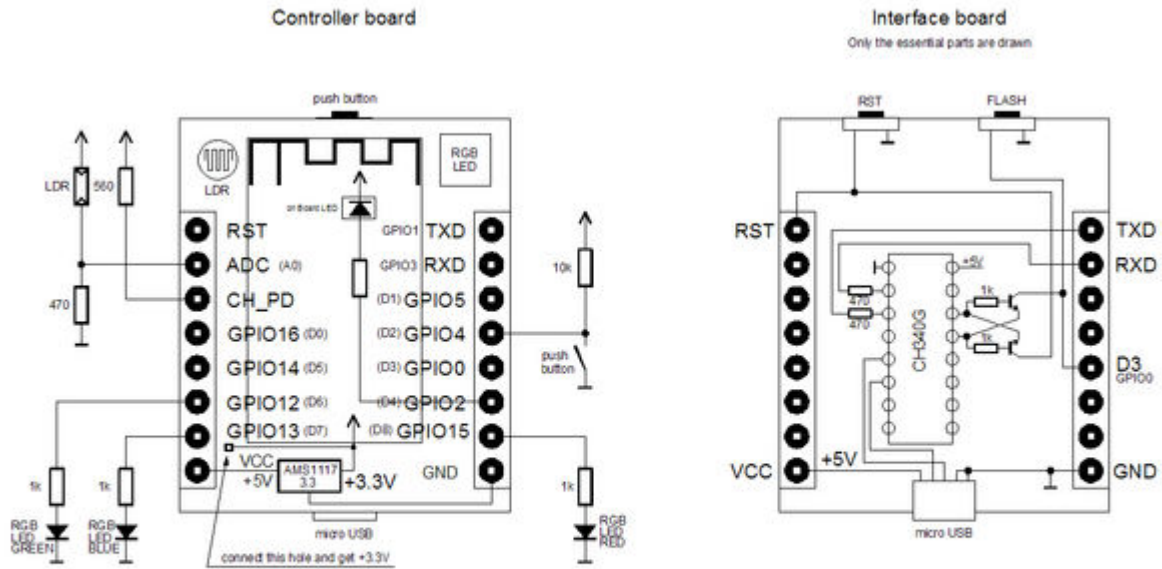
- Arduino 재시작
- Witty cloud 보드의 아래쪽 보드(ESP8266MOD칩이 없는 보드)에 달린 usb를 pc에 연결
- Tool - Board
 - WeMos D1 R2 & mini를 선택
 - Port에서 /dev/cu.wchusbserial410 선택
 - Upload Speed에서 57600 선택
- 필요한 추가 라이브러리 - ArduinoWebsockets하고 WifiManager 설치
 - Sketch - Include Library - Manage Libraries.. 클릭

- * websockets 검색 후 나오는 목록 선택 후 2.X.X 선택 후 설치
- * wifimanager 검색 후 나오는 항목 설치



Websocket for RFLink gateway

- github에서 [Serial to Websocket 예제](#)를 찾은 후 ip대신 이름으로 접근하게 해주는 mDNS 및 OTA 펌웨어 업데이트가 가능하도록 간단히 수정했다. 최종 소스는 [여기서](#) 다운 받을 수 있다.
- Verify 혹은 Upload 실행 후 후 보드의 Flash 버튼 눌러서 펌웨어 업로드하면 되는 데 빌드가 끝날 때쯤해서 Flash버튼을 누르고 있다가 업로드가 시작되면 버튼에서 손을 떼도 된다.
 - 아무리 해도 안되는 경우에는 usb 케이블을 뺀 후 Flash 버튼을 누른 채로 usb 케이블을 연결 후 Upload해서 업로드가 시작된 후 버튼에서 손을 떼면 된다.
- 이제 Witty cloud 보드에서 하단 보드를 빼고 상단보드만 아두이노에 연결하면 된다. esp8266 자체는 3.3V 기반이지만, Witty cloud 보드는 5V를 3.3V로 바꿔서 사용하므로 VCC에 5V를 넣어줘야 한다는 것이다. 혹시 보드상의 esp8266에 직접 연결시에는 3.3V를 넣어줘야 한다.
 - 아두이노 RX --> 상단 보드 TX
 - 아두이노 TX --> 상단 보드 RX
 - 아두이노 5V --> 상단 보드 VCC
 - 아두이노 GND --> 상단 보드 GND



witty_cloud_module

- Chrome extension Simple WebSocket Client 설치해서 ws://rflink.local:81 입력(소스에 입력한 명칭을 사용) 후 open 버튼 눌러서 RFLink gateway에서 메시지를 받을 수 있는지 확인해 본다.
- 웹브라우저로 <http://rflink.local/update> (소스에 입력한 명칭을 사용) 접속해서 펌웨어 업로드도 가능하다.