Programming the ESP8266 with the Arduino IDE

Originalartikel

Backup

<html> <section id=„things“> <h2 class=„hckuitypographyh2 title-with-anchor“>Things used in this project</h2> <div class=„project-parts view-expanded c7“> <table class=„project-parts-table“><tbody><tr class=„head“><td colspan=„6“> <h3 class=„hckuitypographyh3“>Hardware components</h3> </td> </tr><tr><td class=„part-img“></td> <td> <table><tbody><tr><td class=„hckuitypographybodyL“><a data-ha=„{"eventName":"Clicked link","customProps":{"value":"Everything ESP ESP8266 ESP-01","href":"/esp/products/esp8266-esp-01","type":"part","location":"things"},"clickOpts":{"delayRedirect":true}}“ class=„hckuitypographylinkBlue“ href=„https://www.hackster.io/esp/products/esp8266-esp-01“>Everything ESP ESP8266 ESP-01</td> </tr><tr><td class=„hckuitypographybodyS hckuitypographypebble hckuitypographybreakWord“/> </tr></tbody></table></td> <td class=„hckuitypographybodyL times“>×</td> <td class=„hckuitypographybodyL quantity“>1</td> <td class=„actions“> </td> </tr><tr><td class=„part-img“></td> <td> <table><tbody><tr><td class=„hckuitypographybodyL“><a data-ha=„{"eventName":"Clicked link","customProps":{"value":"Arduino UNO \u0026 Genuino UNO","href":"/arduino/products/arduino-uno-genuino-uno","type":"part","location":"things"},"clickOpts":{"delayRedirect":true}}“ class=„hckuitypographylinkBlue“ href=„https://www.hackster.io/arduino/products/arduino-uno-genuino-uno“>Arduino UNO & Genuino UNO</td> </tr><tr><td class=„hckuitypographybodyS hckuitypographypebble hckuitypographybreakWord“>You could also use a UARTbee or any UART to USB device</td> </tr></tbody></table></td> <td class=„hckuitypographybodyL times“>×</td> <td class=„hckuitypographybodyL quantity“>1</td> <td class=„actions“> </td> </tr><tr><td class=„part-img“></td> <td> <table><tbody><tr><td

class=„hckuitypographybodyL“>Jumper wires (generic)</td> </tr><tr><td class=„hckuitypographybodyS hckuitypographypebble hckuitypographybreakWord“/> </tr></tbody></table></td> <td class=„hckuitypographybodyL times“>×</td> <td class=„hckuitypographybodyL quantity“>1</td> <td class=„actions“> </td> </tr><tr class=„head“><td colspan=„6“> <h3 class=„hckuitypographyh3“>Software apps and online services</h3> </td> </tr><tr><td class=„part-img“></td> <td> <table><tbody><tr><td class=„hckuitypographybodyL“><a data-ha=„{"eventName":"Clicked link","customProps":{"value":"Arduino IDE","href":"/arduino/products/arduino-ide","type":"part","location":"things"},"clickOpts":{"delayRedirect":true}}“ class=„hckuitypographylinkBlue“ href=„https://www.hackster.io/arduino/products/arduino-ide“>Arduino IDE</td> </tr><tr><td class=„hckuitypographybodyS hckuitypographypebble hckuitypographybreakWord“>version 1.6.6 or higher</td> </tr></tbody></table></td> <td colspan=„2“/> <td class=„actions“><svg class=„hckuitypographyicon hckuitypographyicon16“><use xlink:href=„#svg-external-link“/></svg></td> </tr></tbody></table> </section> <section id=„story“> <div class=„project-story collapsible-section collapsed hljs-active hljs-monokai“ itemprop=„text“> The ESP8266 is a low-cost WiFi module built by Espressif Systems. Its popularity has been growing among the hardware community thanks to it's nice features and stability. It can even be easily programmed using your Arduino IDE. In this post, we'll learn how to program your ESP8266 using an UARTbee or Arduino UNO, and the Arduino IDE. <h3 class=„hckuitypographyh3 title-with-anchor“ id=„toc-wire-up-0“>Wire up!</h3> First we'll need to identify the pinout of ESP8266. <div class=„image_carouselcontainerhMJxn image_carouselwrapper102VA image_carouselscrollContainer3mmPE image_carouselimageContainer22WPm image_carouselimageWrapper39AG2“ data-hypernova-key=„ImageCarousel“ data-hypernova-id=„35c15af2-c488-4da9-915c-1709592da0b8“> To set the ESP8266 in programming mode you need to connect its wires like this: ESP8266 UARTbee Arduino UNO <ul class=„hckuitypographybodyL“>
RXD TXD <ul class=„hckuitypographybodyL“>
RX GPIO0 <ul class=„hckuitypographybodyL“>
GND GND <ul class=„hckuitypographybodyL“>
GPIO2 ——- <ul class=„hckuitypographybodyL“>
———- GND <ul class=„hckuitypographybodyL“>
GND GND <ul class=„hckuitypographybodyL“>
VCC 3.3V <ul class=„hckuitypographybodyL“>
3.3V RST <ul class=„hckuitypographybodyL“>
——– ———– <ul class=„hckuitypographybodyL“>
CH_PD 3.3V <ul

class=„hckuitypographybodyL“>
3.3V TXD <ul class=„hckuitypographybodyL“>
RXD TX Note: If you are using an Arduino UNO you will need to set Arduino RST to GND. Please be careful with the VCC of the ESP8266, it works only with a 3.3V supply. <h3 class=„hckuitypographyh3 title-with-anchor“ id=„toc-setup-the-arduino-ide-1“>Setup the Arduino IDE</h3> <ul class=„hckuitypographybodyL“>
Open you IDE and click on „File -> Preferences“. <ul class=„hckuitypographybodyL“>
In „Aditional Boards Manager URLs“ add this line and click on „OK“: http://arduino.esp8266.com/stable/package_esp8266com_index.json <ul class=„hckuitypographybodyL“>
Go to „Tools -> Board -> Boards Manager“, type „ESP8266“ and install it. <ul class=„hckuitypographybodyL“>
Go again to „Tools -> Board“ and select „Generic ESP8266 Module“. <h3 class=„hckuitypographyh3 title-with-anchor“ id=„toc-flash-your-code-2“>Flash your code!</h3> Now you're ready to use your ESP8266 as a stand-alone module without needing an external micro-controller. Check out more ESP8266 resources here to learn how to build IoT products and projects using this amazing module.   Read more</section> <section id=„team“> <h2 class=„hckuitypographyh2 title-with-anchor“>Credits</h2> </section> <section id=„comments“> <h2 class=„hckuitypographyh2 title-with-anchor“>Comments</h2> </section> </html>