



# Das Internet der Dinge

## Einfach einsteigen mit dem Pretzelboard



Infos zum Maker-Kit

im ELV-Web-Shop

#1467

Das Internet der Dinge, kurz IoT (Internet of Things) ist für die meisten Menschen ein abstraktes „Ding“, unter dem man sich nicht immer etwas Konkretes vorstellen kann. Tatsächlich bietet uns die Industrie bereits zahlreiche Alltagslösungen, die das IoT repräsentieren – von der per Smartphone gesteuerten Leuchte bis zum drahtlosen Fitness-Armband. Wesentliches technisches Kennzeichen ist meist die drahtlose Einbindung in ein Netzwerk – nicht ganz trivial für den Selbstbauer. Mit dem Maker-Kit „Internet of Things“ bietet der Franzis-Verlag nun in bewährter Weise eine Experimentierplattform an, die den IoT-Einstieg per WLAN besonders einfach macht.

### Kleines Board mit Potenzial

Wesentlicher Inhalt des Maker-Kits – Autor ist Fabian Kainka – ist das Pretzelboard, auch als NanoESP bekannt. Dazu kommen wie immer ein ausführliches 100-seitiges Begleitbuch, ein Zugangscode zu den zugehörigen Beispielprogrammen sowie neben zwei Steckbrettern die notwendigen Bauelemente für die insgesamt 26 Experimente des Kits.

Das Pretzelboard (Bild 1) ist eine Kombination aus einem Arduino Nano und dem weit verbreiteten WLAN-Modul ESP8266 [1]. Für beide Teilbaugruppen finden sich ISP-Anschlüsse, sodass erfahrene Programmierer nicht unbedingt die Arduino-IDE nutzen müssen. Die Spannungsversorgung erfolgt per USB, kann aber auch aus anderen Spannungsquellen wie Batterien, Akkus (7–12 V) oder einer 5-V-USB-Powerbank bzw. einem USB-Netzteil erfolgen. Das ist natürlich bei den meisten Einsätzen des Boards sinnvoll, etwa als Sensor, Stand-alone-WLAN-Webserver oder Access-Point.

Als Mikrocontroller kommt der ATmega 328 mit 32-KB-Flash, 2-KB-SRAM und 1-KB-EEPROM zum Einsatz, der USB-Kommunikation dient der CH340G. Dieser bedingt die Installation eines speziellen Treibers.

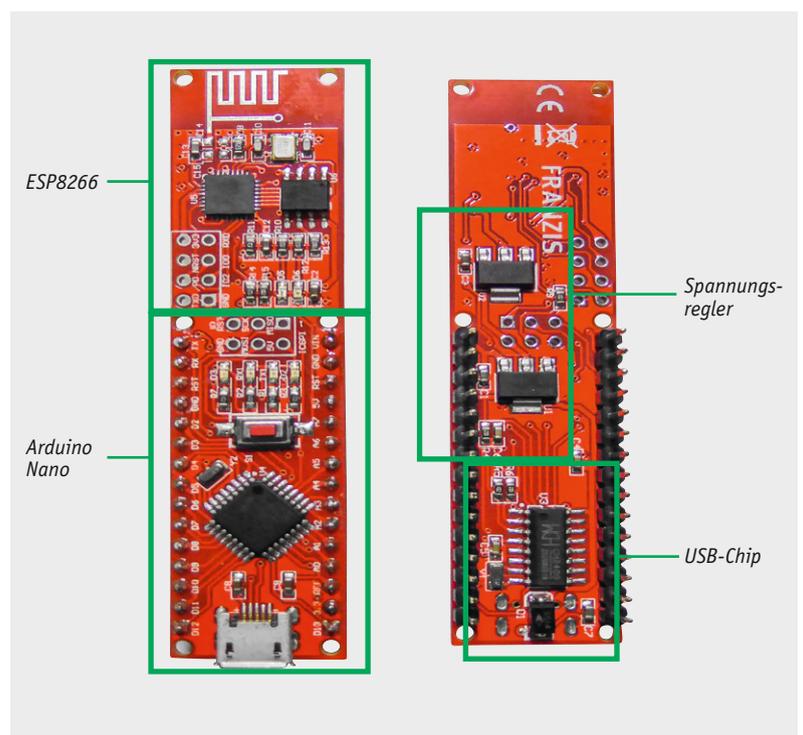


Bild 1: Die Baugruppen des Pretzelboards in der Übersicht



Pin	Port	Special	PWM	PWM	Special	Port	Pin
D13	PB5	L-LED	SCK			PB4	D12
3.3 V		(max 50 mA)				PB3	D11
AREF						PB2	D10
A0/D14	PC0					PB1	D9
A1/D15	PC1					PB0	D8
A2/D16	PC2					PD7	D7
A3/D17	PC3					PD6	D6
A4/D18	PC4	SDA				PD5	D5
A5/D19	PC5	SCL				PD4	D4
A6		(only analog in)				PD3	D3
A7		(only analog in)				PD2	D2
5 V							<b>GND</b>
Reset	PC6					PC6	Reset
<b>GND</b>							
Vin		(max 1 A, 1.2V)					

LEDs:	Free to Use
D3: L-LED (connected to Pin D13)	
D2: Power LED (ATmega328)	
TX: TXD (ATmega328)	
RX: RXD (ATmega328)	
D6: Power LED (ESP8266)	
D5: TXD (ESP8266)	

Notes:	Power Pins (GND)
Max per Pin 40mA	
Absolute max all Pins: 100mA	
A6/A7: Only analog in	

Special Pins
1) ATmega328p
2) Esp8266
3) SPI-Flash
4) Reset-Button
5) ISP ATmega328p
7) ESP8266 Connector

Bild 2: Das Pinout des Pretzelboards

Der ESP8266 ist mit einem 4-MBit-Flash ausgestattet, verfügt bereits über einen integrierten TCP/IP-Stack und kommuniziert über die UART-Schnittstelle mit dem ATmega328. Damit muss sich der Nutzer nicht mit der Programmierung von Netzwerk-Protokollen, den Schichten des TCP/IP-Referenzmodells und deren Abarbeitung im Stapel (Stack) etc. beschäftigen, der Nano-Prozessor übergibt bzw. empfängt die Daten aus dem WLAN einfach per seriellem Protokoll, um den Rest kümmert sich die Firmware des ESP8266.

Bild 2 zeigt das Pinout des Pretzelboards, hier sehen wir auch die Spannungsversorgungsanschlüsse für den autarken Betrieb. Für die Bereitstellung der 5 V für den ATmega 328 und der 3,3 V für den ESP8266 sind zwei Spannungsregler auf dem Board vorhanden, eine einfache Pegel-Shifter-Schaltung sorgt für die Pegelumsetzung zwischen den beiden Bausteinen.

Damit hat man, was die Hardware betrifft, alles in der Hand, um einen kompletten kleinen Web-Server für das WLAN zu bauen. Mehr – meist eher noch weniger – steckt auch in den WLAN-Leuchten, den WLAN-Wettersensoren und anderen IoT-Geräten nicht, meist wird sogar nur der ESP8266 allein eingesetzt und das Anwendungsprogramm allein in diesen integriert.

Da wir einen Arduino vor uns haben, liegt die Programmierung über die Arduino-IDE natürlich nahe, und so hat der Autor auch den gesamten Kurs darauf aufgebaut; alle Beispiele werden hierüber programmiert.

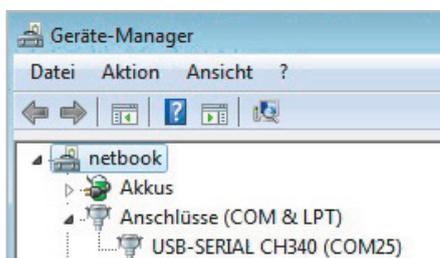


Bild 3: So erscheint der ordnungsgemäß installierte USB-Treiber.

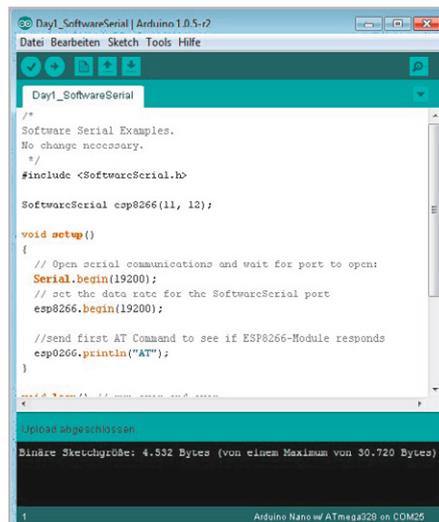


Bild 4: Die gesamte Programmierung des Lehrgangs erfolgt einfach per Arduino-IDE.

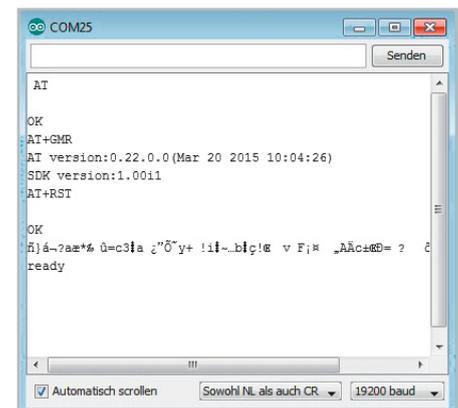


Bild 5: Per seriellem Monitor kann man sofort die AT-Befehle zur Kommunikation mit dem ESP8266 testen.

Weiterführende Beispiele und ein Forum zum Pretzelboard findet man auf der Homepage des Autors [2].

## Schritt für Schritt ins WLAN

Also, Board auf das Steckbrett aufstecken, Buch aufschlagen, und los! Hat die Treiberinstallation geklappt, findet sich das Board im Gerätemanager mit einer virtuellen seriellen Schnittstelle wieder (Bild 3). Startet man nun die Arduino-IDE (Bild 4 zeigt sie schon mit auf den ATmega geladenem erstem Programm), muss man nur noch den Prozessor (ATmega328) und das Board (Nano) einstellen, unten in der IDE erscheint dann das Board an der richtigen Schnittstelle. Damit sind schon alle Vorbereitungen erledigt. Was sagt der ESP8266? Die ganz einfache Variante ist das allgegenwärtige Smartphone, das uns einen funktionierenden WLAN-Access-Point mit einem offenen Netzwerk namens NanoESP anzeigt, die Grundeinstellung des WLAN-Moduls (siehe Titelbild).

Sofort geht es an die substanziellen Experimente. Der Autor geht systematisch vor und führt zunächst in die grundsätzliche Kommunikation mit dem ESP8266 mittels AT-Kommandos ein. Mit einem einfachen Kommunikationsprogramm (Bild 4) und dem seriellen Monitor der Arduino-IDE (Bild 5) kann man sofort die wichtigsten AT-Kommandos testen. Eine komplette AT-Kommandotabelle findet sich am Ende des Begleitbuchs.

Sofort geht es weiter an die Konfiguration über WLAN-AT-Befehle. Diese sollte man, auch wenn der Kurs ein automatisches Konfigurationsprogramm anbietet, einmal ausprobieren, um die Konfiguration auch zu verstehen, zudem erlernt man hier die Grundlagen der sicheren Kommunikation.

Mit einem ersten Anwendungsprogramm, einem WLAN-Sniffer, der andere aktive WLANs in der Nähe akustisch detektiert, wird das erste (Kennenlern-)Kapitel abgeschlossen.

## Erste Kommunikation

Im zweiten Kapitel folgt das detaillierte Heranführen an die Kommunikation in einem Netzwerk, sprich PC und Pretzelboard. Hier werden Themen wie IP, Ports und das später zur Anwendung kommende User Datagram Protocol (UDP) besprochen und mit Trainingslektionen unterfüttert. Dieses Kapitel gipfelt dann in den ersten Anwendungen – wir können vom PC aus eine Last (hier eine LED) fernschalten bzw. Textnachrichten auf ein Display schicken.

## Das Pretzelboard als TCP-Client

Hier geht es weiter hinein in die Netzwerkkommunikation, das Pretzelboard wird als normaler Netzwerk-Teilnehmer (Client) eingerichtet, der über das Protokoll TCP mit anderen Teilnehmern kommuniziert und z. B. den Inhalt von Webseiten abfragen kann. Klas-



sisches Beispiel ist die Internetuhr, hier wird mit dem Beispielprogramm eine Webseite kontaktiert, auf der Zeitdaten in unterschiedlicher Form und für unterschiedliche Zeitzonen abgerufen werden können. Dafür nötige Bibliotheken sind bereits in die Arduino-Sketches eingebunden bzw. aus den Standardbibliotheken, z. B. für die LCD-Anzeige, der IDE ladbar.

Seite für Seite werden die Anwendungen interessanter, so kann man Wetterseiten abfragen und das zu erwartende Wetter per Display und von Weitem erkennbarer RGB-LED visualisieren oder das Mediacenter „Kodi“ vom Pretzelboard aus über Taster steuern. Da es ja freisteht, die Arduino-Programme selbst zu schreiben bzw. die Beispiele zu modifizieren, kann man sich auch die ersten Heimautomatisierungsprogramme selbst schreiben, z. B. die Uhr als Wecker einsetzen, externe Lasten über eine Relaischaltstufe zu bestimmten Zeiten oder, nach Einbindung des mitgelieferten Fototransistors, nach Helligkeit schalten. Da beginnt allmählich das Internet der Dinge Gestalt anzunehmen. Schnell entsteht bereits hier der Gedanke, etwa ein eigenes WLAN für eine Kommunikation zwischen mehreren ESP-Boards aufzubauen, um z. B. Fernsteuerungsaufgaben zu realisieren. Dazu muss man nicht mehrere Pretzelboards kaufen, eine preiswerte Kombination aus Arduino Nano und einem ESP8266-Modul mit TCP/IP-Stack inklusive Pegelwandler und Stromversorgung kann man sich bei günstigem Einkauf für unter 10 Euro zusammenstellen. Ganz minimalistisch geht es mit der Kombination des äußerst preiswerten Arduino ProMini, der keine feste USB-Schnittstelle besitzt, und einem ESP8266-Modul zu. Der Arduino wird mithilfe eines temporär angeschlossenen USB-Moduls, wie des UM2102 aus dem ELV-Programm (Bild 6), programmiert, dann die ganze Applikation an ihrem Arbeitsort, etwa als Temperatursensor, installiert, und schon hat man einen minimalistischen IoT-Sensor.

### Vom Client zum Web-Server ...

... ist es am Pretzelboard nur ein weiteres Beispielprogramm und einige AT-Befehle weit. Dann hat man einen TCP-Server installiert, den man über einen beliebigen Browser im Netzwerk aufrufen kann. Sofort geht es weiter im Programm, und es entsteht ein autonomer, batteriebetriebener Webserver, der sich per Browser ansteuern lässt und dort eine LED schaltet. Nach dieser einfachen Applikation, bei der man im Browser noch quasi im Quelltext des Programms arbeitet, folgt nun die komfortablere Version mit einer im Webserver hinterlegten HTML-Seite, die das Schalten der LED über Schaltflächen erlaubt (Bild 7). Nach einem kurzen Crashkurs in HTML sind wir dann schon in der Lage, weitere Aktionen zu starten, Sensoren einzusetzen und zu visualisieren und die GPIOs des ATmega fernzusteuern (Bild 8). Den Abschluss des Kapitels bildet dann die Königsdisziplin – wir lernen, wie man aus dem Internet via Router auf das Pretzelboard zugreift.

### IoT total mit ThingSpeak

Zum Abschluss geht es komplett ins Internet. Sicher haben Sie schon etwas von IFTTT [3] gehört. Der Dienst erlaubt z. B. die Verknüpfung von Ereignissen mit Reaktionen über das Internet nach dem Wenn-dann-Prinzip. Ein nicht ganz unähnlicher Ansatz ist ThingSpeak [4], dem sich

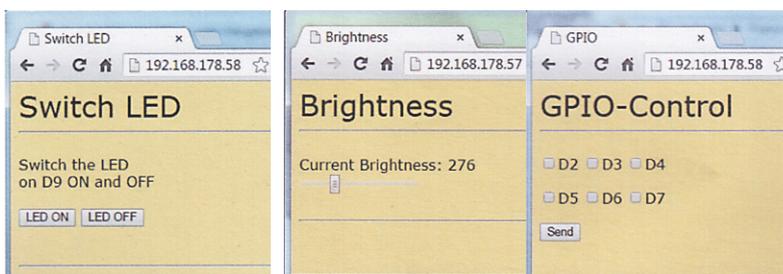


Bild 7: Webserver mit HTML-Bedienfeld – auch das kann das Pretzelboard

Bild 8: Weitere HTML-Beispiele für Fernkontrolle und Fernsteuern via Webserver

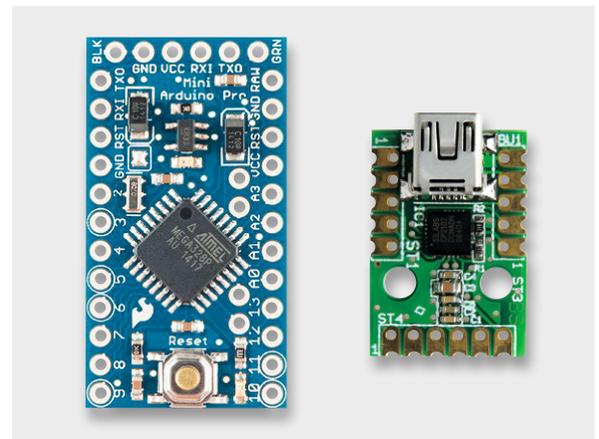


Bild 6: Für Minimal-Applikationen mit dem Arduino Pro Mini (links) als USB-Schnittstelle einsetzbar: der UM2102 von ELV (rechts, Best.-Nr. CI-09 18 59).

das letzte Kapitel des Begleitbuchs widmet (Bild 9). Meldet man sich hier an, steht eine Onlineplattform zur Verfügung, über die man Daten visualisieren, Ereignisse und Daten auf Webseiten abfragen und damit Aktionen auslösen, oder z. B. eine Twitter-Meldung bei einem Einbruchversuch am Standort des Webservers absetzen lassen kann. Spätestens hier fängt die absolut praktische Nutzung im IoT an – und dank des kurzweiligen, systematischen Leitfadens ist man nun fit für etwas Eigenes! **ELV**



### Weitere Infos:

- [1] Forum, Wiki zum ESP8266: [www.esp8266.com](http://www.esp8266.com)
- [2] Pretzelboard: [www.pretzelboard.cc](http://www.pretzelboard.cc)
- [3] IFTTT: <https://ifttt.com>
- [4] ThingSpeak: <https://thingspeak.com>

FRANZIS Maker Kit Internet of Things  
Best.-Nr. CI-12 21 85, € 79,95

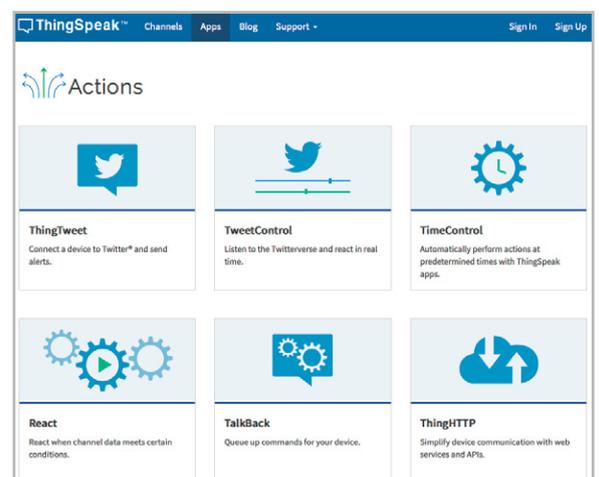


Bild 9: Erweitert die Einsatzmöglichkeiten des Pretzelboards erheblich – IoT-Plattform ThingSpeak