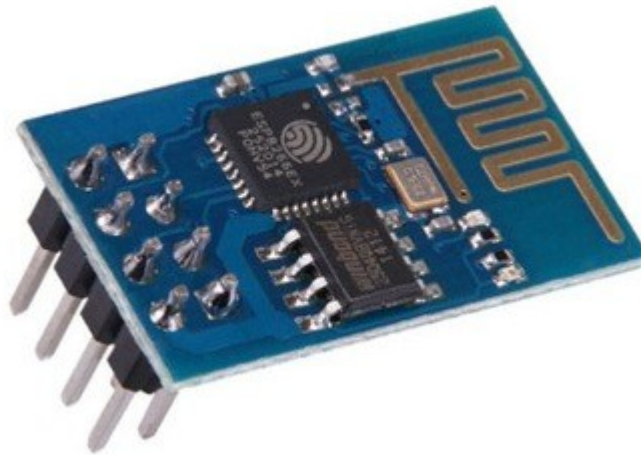


ESP8266

Fernabfrage der Raumtemperatur

Als Bastler kann man an diesem tollen Baustein nicht vorübergehen. Mit dem preiswerten Teil habe ich mir einen funktionsfähigen Melder gebaut, den ich z.B. im Urlaub über das Handy ansprechen kann.



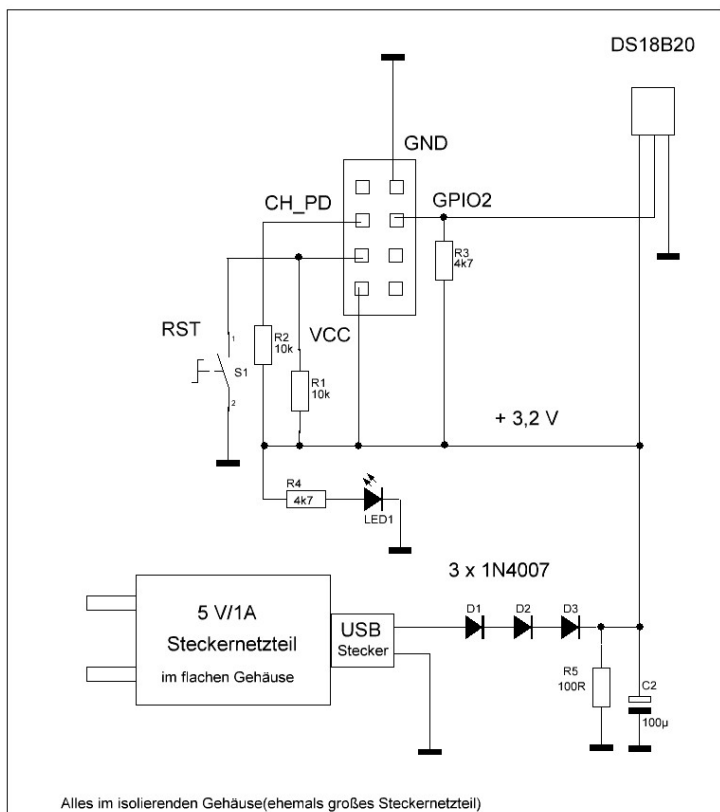
Vorgeschichte

Im letzten Frühjahr waren die Außentemperaturen noch so niedrig, dass unsere Gasheizung auch im unbewohnten Haus während unserer Abwesenheit durchlaufen musste. Allerdings wusste ich, dass das Wiederanspringen unsicher war. Deshalb hatte ich mir einen professionellen Temperaturfühler zugelegt, den man vom Internet aus ansprechen konnte. Und tatsächlich kam er zum Einsatz. Von Holland aus kontrollierte ich täglich die Raumtemperatur in unserem Wohnzimmer. Und als diese einen niedrigen Wert erreichte, rief ich meinen Nachbarn an. Da seine Frau unsere Katze versorgte, hatte sie unseren Haustürschlüssel. Meinen Nachbarn bat ich, mit dem Schalter die Heizung kurz vom Netz zu trennen und danach das Wiederanlaufen zu beobachten. So sprang sie erfolgreich wieder an und wir konnten sicher sein, in ein geheiztes Haus zurückzukehren. Der Kauf des Webmelders für ca. 50 EUR hatte sich fast gelohnt. Jetzt aber wollte ich unbedingt selbst so eine Schaltung realisieren.

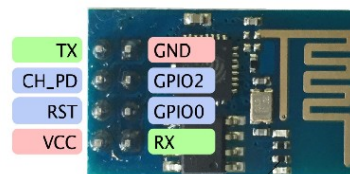
Die Ausführung

Im folgen Bild erkennt man die einfache Schaltung. Ich hatte noch ein Steckernetzteil für 5V. Solche Teile im flachen Gehäuse werden auch für das Laden der Handys von Samsung mit 3V Ausgangsspannung angeboten. Um nun hier von 5V auf 3V Vcc zu kommen, legte ich drei 1N4007 in Reihe. Der Belastungswiderstand ist wichtig, damit am Chip keine Überschreitung geschieht. Er sollte auch so angebracht werden, dass seine Verlustwärme nicht auf den Fühler trifft. Im Betrieb kann der ESP schon mal über 300 mA ziehen. Ein zusätzlicher Elko stützt die Vcc.

Web-Server für Temperatur-Abfrage über das WWW



ESP 8266-01



nach der Betätigung des RST-Tasters zeigt das hellrote Leuchten der Diode die Betriebsbereitschaft an

DF8ZR

RST

Es hat sich gezeigt, dass der RESET des Chips dauerhaft auf Vcc liegen sollte. Für den Start des gespeicherten Programm muss man allerdings einmal auf den Taster drücken. Wenn die LED auf dem Board des ESPs hell leuchtet, ist das ein Zeichen der Betriebsbereitschaft. Aber sicher kann man erst dann sein, wenn man die Abfrage mit einem Handy kontrolliert hat.

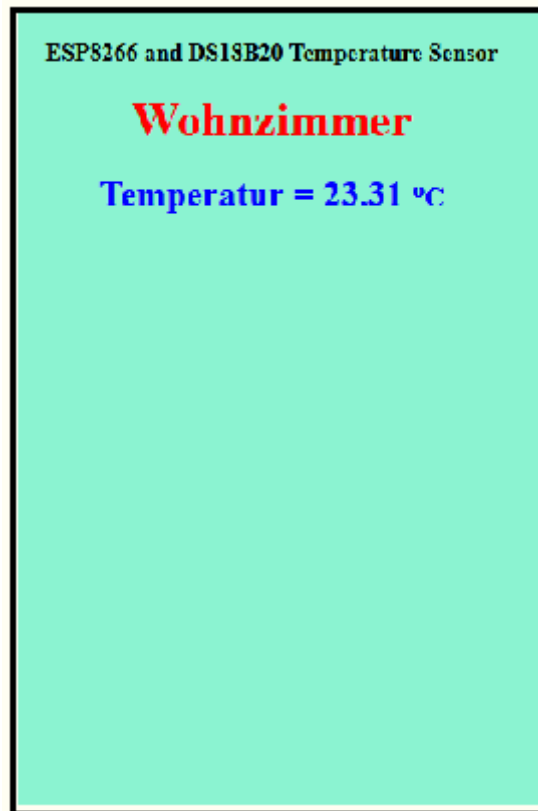
Wer hier meint, dass nach einem Stromausfall die Sache nicht mehr klappt, kann ja eine elektronische Resetschaltung entwerfen, um einen solchen Fall abzusichern. Mir genügt aber vorerst diese Lösung.

Software

Hier habe ich im Netz eine Vorlage gefunden. Leicht abgeändert und mit weiteren Kommentaren versehen, stelle ich sie nebenan zum Download bereit. Die Datei muss noch angepasst werden. Den Namen des eigenen WLANs und das dazugehörige Passwort und den Port auf der Fritzbox oder dem jeweiligen Router muss man eintragen, bevor man kompiliert.

Auf dem Handy

Auf meinem Handy Samsung S3 Neo erscheint die Abfrage so:



Die IDE

Ich habe die IDE vom Arduino eingesetzt. Es ist nicht ganz einfach, alle Bibliotheken zu finden. GitHub usw. helfen hier. Weil aber ggf. ein Server außer Betrieb ist, findet man ESP8266WiFi nochmal bei GitHub. Habe sie erst nach sehr langem Suchen entdeckt und in meine Version Arduino 1.8.3 installiert.

Breadboard

Es ist zu empfehlen, im ersten Schritt ein Breadboard einzusetzen. Mit zwei Stabzellen AA habe ich die Vcc mit 3V bereitgestellt. Man wird aber bemerken, dass die nach wenigen Stunden bereits verbraucht sind. Deshalb ist für den dauernden Betrieb unbedingt ein Netzteil vorzusehen. Wenn man alles berührungssicher in ein Kunststoffgehäuse bringt, sind auch die VDE-Vorschriften erfüllt. Ich verwendete ein Gehäuse von einem ausgedienten Steckernetzteil. Aber auch solche von alten Funksteckdosen sind gut geeignet.

Flashen

Es kann sein, dass man nicht gleich den Erfolg hat. Ein moderner Converter für USB nach FTDI hat einen Umschalter auf 3V für den Eingang des ESPs. Damit ging die Sache wenigstens elektrisch flott vonstatten. Allerdings können noch Mängel bei den Bibliotheken sein, die das Kompilieren in die Länge ziehen. Hier gilt es, nicht

aufzugeben! Mit Geduld erreicht man sein Ziel. Und es sei noch bemerkt, dass viele Vorschläge im Netz unvollständig sind und Mängel haben, weshalb sie nicht einfach zum Erfolg führen. Die hohe Zeit der Anwendungen war 2014 und endete 2015. Aktuell findet man selbst in den Sachbüchern nur Ansätze für die Programmierung, leider keine kompletten Lösungen!

DynDNS

Für den Abruf über das Internet benötigt man einen sog. DNS-Server. Der ermittelt die aktuelle IP des Routers. Dies wird z.B. von T-Online um Mitternacht täglich geändert. Im Browser des Handys trägt man dann die IP des DNS-Servers ein und schließt nach einem Doppelpunkt mit der Nummer des freigeschalteten Ports:

z.B. `http:xxxxxxxxx.yyy:81`

Dadurch ist der Socket für den ESP eindeutig adressiert und kann weltweit angesprochen werden. Die Übertragung der Temperatur wird über das html-Protokoll abgewickelt. Im Browser des Handys richte man dazu ein Lesezeichen ein und braucht dann nicht jedesmal die URL einzutippen.

Wie man sich eine DynDNS-IP beschafft, wird im Netz vielfach beschrieben.

Raspberry-PI

Mein Freund Volker hat über diese Funktion hinaus noch andere Möglichkeiten mit dem Rpi programmiert. Hier ein Link auf seine Seite:

<http://dl4fj.mo00.com>

Er wird sich demnächst auch mit dem ESP8266 befassen und ihn als Webserver einsetzen.

Viel Spaß damit!

DF8ZR; im Juli 2017