

Nachtrag zum Projekt GW60 und ESP8266 (Wemos D1 mini)

Ich habe inzwischen drei GW60 umgerüstet und in Betrieb. Für mich ist allerdings nicht ganz zufriedenstellend, dass die Impulsauswertung über den integrierten Reed-Kontakt nur solange korrekt funktioniert, solange die im GW60 integrierte Timer-Funktion **nicht** aktiviert ist. Ist Diese aktiviert und die LED unter dem Uhrensymbol des GW60 leuchtet, liegen bei geschlossenem Reed-Kontakt die PWM-Impulse der LED-Ansteuerung auf der Impulsleitung und der Zähler im Wemos zählt diese dann natürlich mit. D.h. das Zählergebnis gibt daher nicht die wirkliche Anzahl der Umdrehungen wieder, die der Rollladen dann braucht um hoch oder runter zu fahren.

➔ Abhilfe... Einbau eines separaten Impulsgebers.

Dazu habe ich, wie auch auf den Bildern in dieser Dokumentation zu sehen ist, einen Hall-Sensor auf ein Stück Lochrasterplatine gelötet und das Ganze mit doppelseitigem Klebeband auf der Rückseite der Steuerplatine befestigt. Die Steuerplatine hab ich noch etwas ausgefräst, damit der Hall-Sensor mehr Platz findet. Die +5Volt und die Masse habe ich direkt von der Steuerplatine abgegriffen. Die Impulsleitung geht wie gehabt über die Opto-Kopplerplatine zum Wemos. Durch Einbau dieses zusätzlichen Sensor´s können auf der Opto-Kopplerplatine die drei 2,2 KOhm Widerstände, der 0,1 uF Kondensator sowie der Transistor entfallen. Die Diode kann ebenfalls entfernt und durch eine Drahtbrücke ersetzt werden, wenn der 1 kOhm Widerstand und die Mini-LED auf der Sensorplatine nicht verbaut werden. Werden diese optionalen Bauelemente allerdings genutzt oder wird ein Arduino-Hall-Sensor-Modul verwendet, bleibt die Diode drin, muss aber umgedreht werden.

Auch am Sketch für den Wemos habe ich etwas „rumgebastelt“.

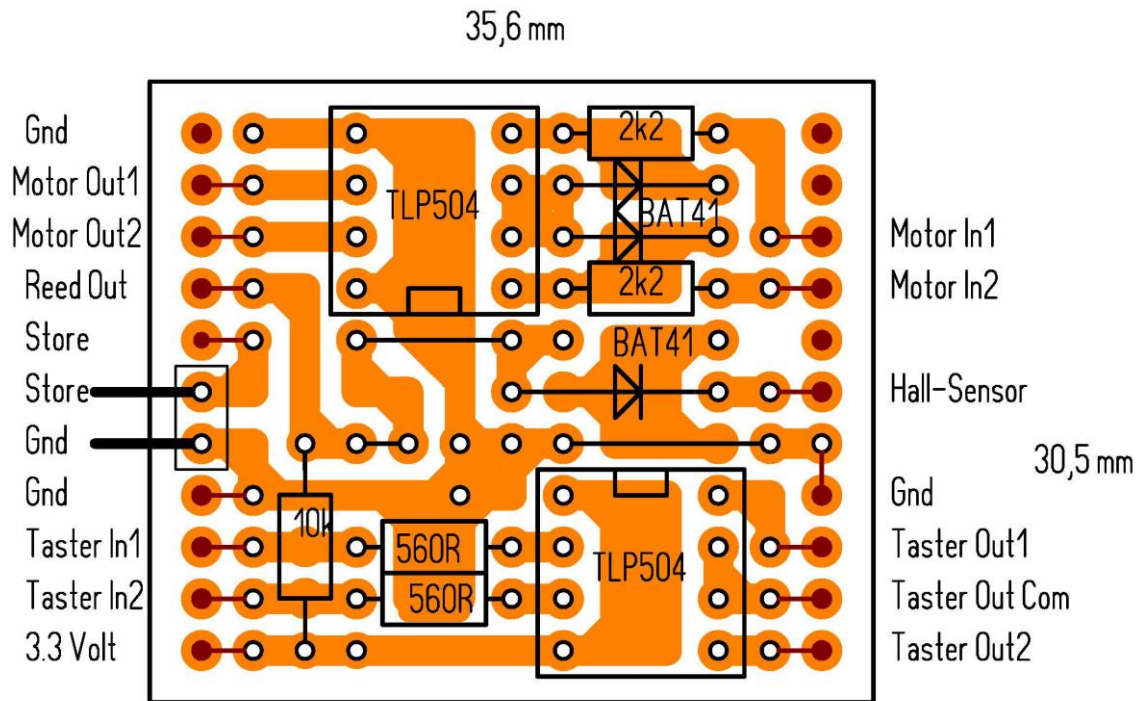
Es gibt jetzt zwei Sketch-Varianten für den Wemos D1 mini.

In der ersten Variante kann die Netzwerkadresse vom Router über DHCP zugewiesen, aber auch fest im Sketch angegeben werden. Hier sind die MQTT-Topics z.B. für Fhem, „EG_Buero/Up“ oder „EG_Buero/Down“ usw. und die ClientID durch die Software fest zugewiesen und müssen daher im Sketch für jeden Wemos individuell angepasst werden. Ist daher relativ ortsgebunden, da es wenig Sinn macht einen GW60 der z.B im Schlafzimmer verbaut ist, mit „EG_Buero/xxx“ anzusprechen. Vorteil ist allerdings, dass der Wemos über jede ihm zugewiesenen IP unter diesen Topics erreichbar ist.

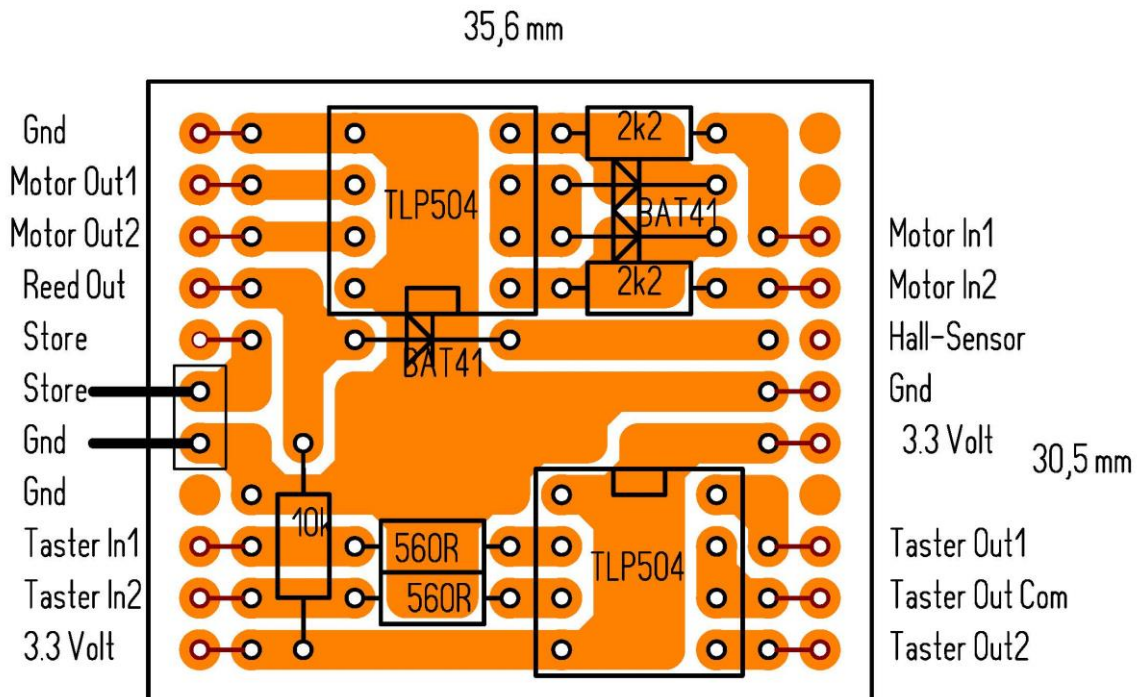
Bei der zweiten Variante werden dem Wemos über den Sketch feste Werte für die Variablen **NAME** und **WIP** zugewiesen. Aus diesen werden dann die ClientID und die entsprechenden MQTT-Topics generiert. Aus z.B. „String NAME = WEMOS_“ und „int WIP = 145“ (hier: 145 = letzte Dreiergruppen der IP-Adresse) wird die ClientID „WEMOS_145“ generiert. Die MQTT-Topics heißen dann entsprechend „WEMOS_145/Up“ oder „WEMOS_145/Down“ usw. .

Es muss also nur die IP-Adresse des WEMOS bekannt sein um die MQTT-Topics in z.B. Fhem zuordnen zu können und somit ist der Einsatz auch relativ ortsungebunden, da die örtliche Zuordnung allein von Fhem festgelegt wird. Es reicht daher im Allgemeinen auch aus, dem Sketch einmal mitzuteilen, wie denn sein Wemos D1 mini heißt, auf dem er seinen Dienst verrichten soll.

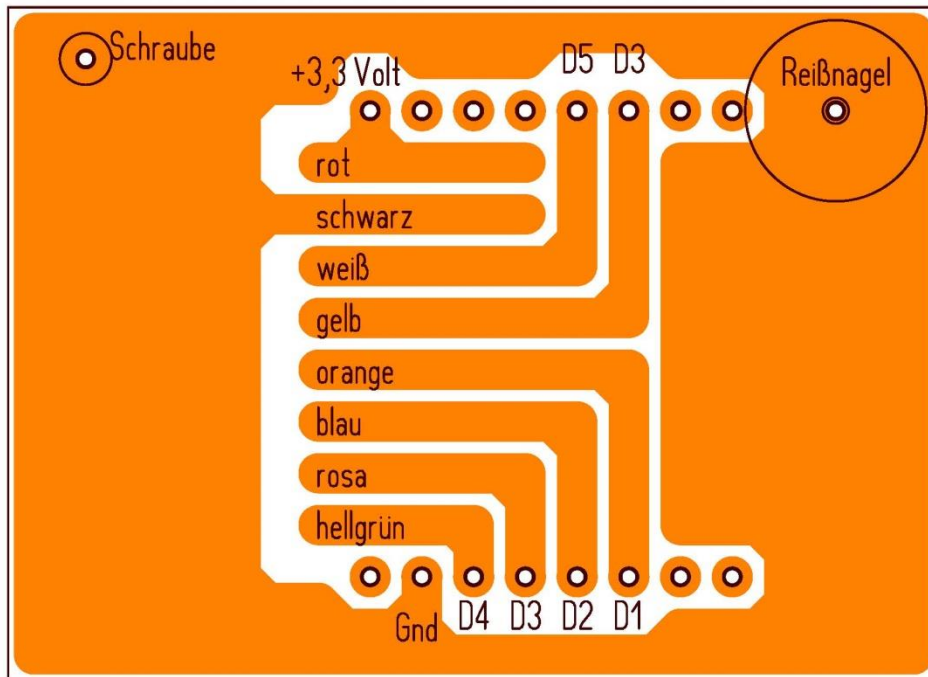
Neue Bestückung der alten Platine:



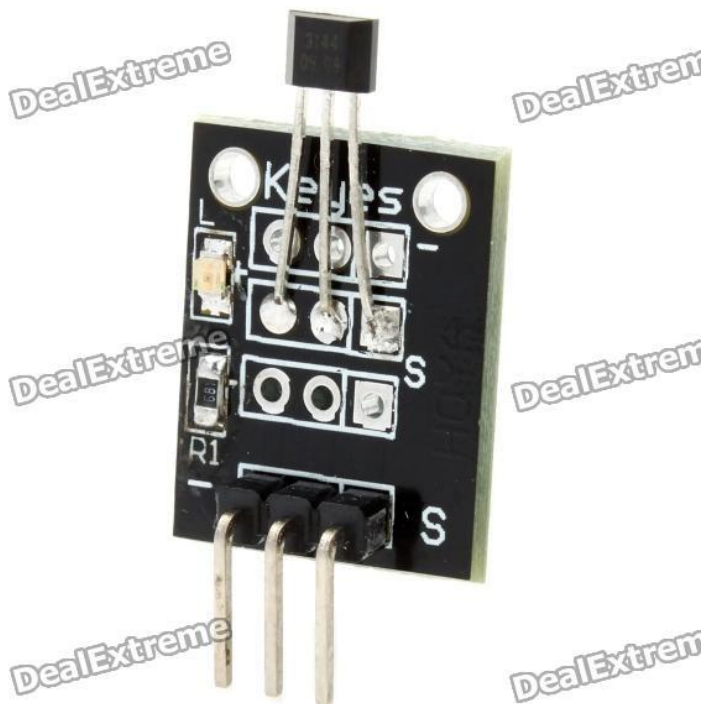
... und die neue Platine:



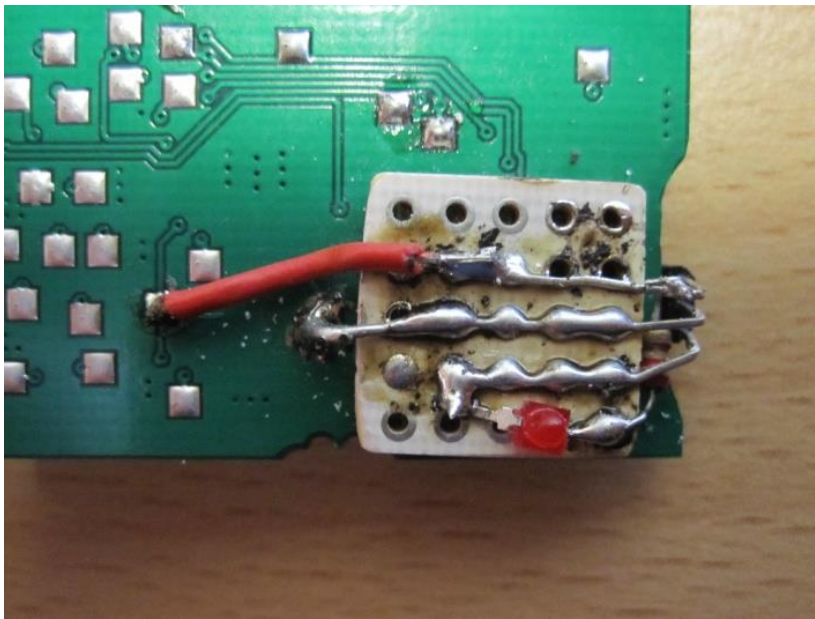
...und die neue Befestigungsplatine:



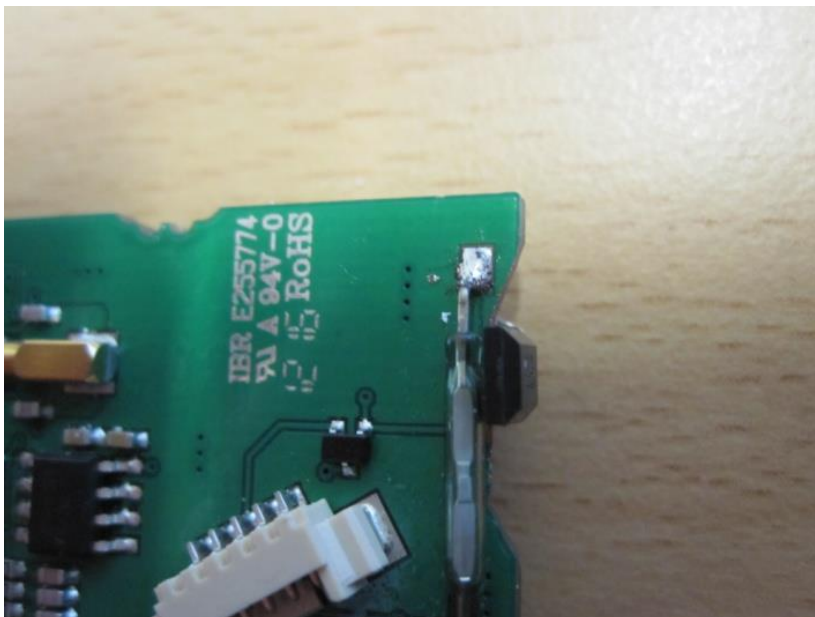
...ein Sensor-Fertigmodul



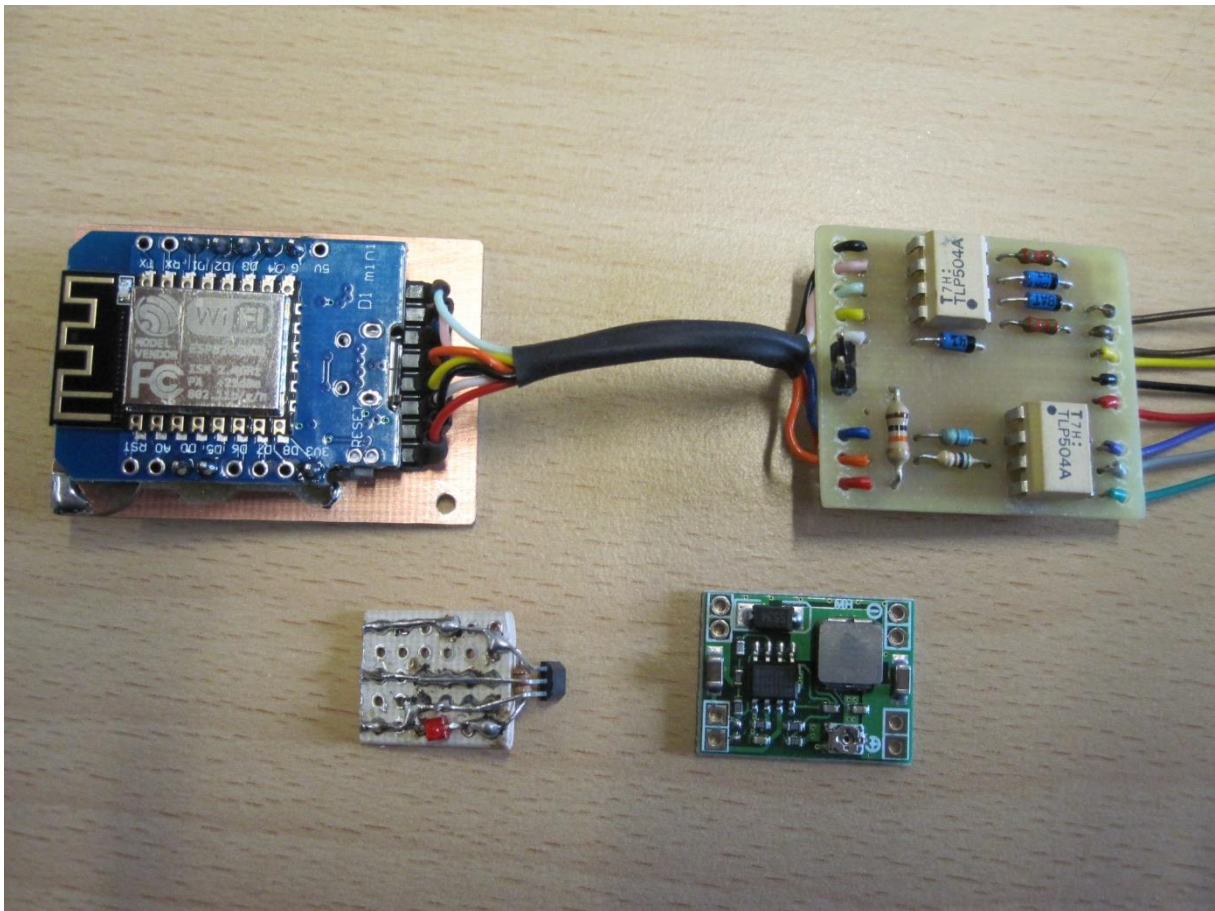
... der Sensor Eigenbau auf der GW60 Steuerplatine:



...und von der anderen Seite:



... und hier nochmal die für den Umbau erforderlichen Teile:



... der Eigenbau-Sensor und die Befestigungsplatine für den Wemos werden auf der Lötseite bestückt und wie zu sehen ist, hab ich dem Wemos auch noch eine Steckverbindung spendiert. Somit kann er ohne löten ausgetauscht werden.