

ATtiny2313 RC Lichtsteuerung Standard

(Rev B)

V. PIPPAN

<http://www.vpippan.at>

30. August 2014



- Bitcoin: 1N6R7RQzLumNDhwkUoS5wYkJgkGMYxxYBb
- Flattr: flattr.com/t/1164314



Dieses Werk ist unter der Creative-Commons-Lizenz vom Typ Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Österreich lizenziert. Um eine Kopie dieser Lizenz einzusehen, besuchen Sie <http://creativecommons.org/licenses/by-sa/3.0/at/> oder schreiben Sie einen Brief an Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

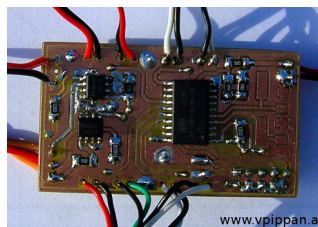
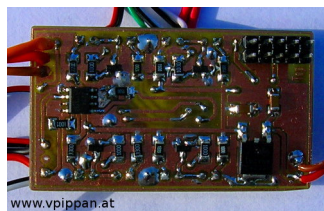
Inhaltsverzeichnis

1. Allgemeines	6
2. Hardwarebeschreibung	7
2.1. Allgemeines	7
2.2. Rev A	8
2.2.1. Technische Daten	8
2.3. Rev B	9
2.4. Mögliche Änderungen für eine neue Version	9
2.5. Fertigungsunterlagen	10
3. Messungen am Conrad Modul	10
3.1. Landelichter	10
3.2. Blinklichter Rot	11
3.3. Blinklichter Weiß	12
3.4. Positionslichter	12
4. Berechnungen	13
4.1. Annahmen	13
4.2. Gleichungen für einen unbelasteten Spannungsteiler	13
4.3. Notwendiger R_x zum Schalten des Komparators (Akku leer Erkennung)	14
4.4. Spannungen U_x am Komparator (Überprüfen der max. Komparator Eingangsspannung und Schaltschwelle	15
4.5. Akkuspannung bei der Komparator Schaltschwelle	15
4.6. Akku Abschaltspannungen unter Berücksichtigung der Widerstandstoleranzen und Temperaturdrift der internen Referenzspannung	15
4.7. Strom I durch den Komparator Spannungsteiler bei vollem Akku	16
4.8. Maximaler Strom durch den Spannungsregler	16
4.9. Maximale Verlustleistung in den Schalt Transistoren	18
4.10. Maximale Verlustleistung in den LED Vorwiderständen	18

4.11. Vorwiderstände für die Optokoppler	18
4.11.1. Optokoppler Eingang	18
4.11.2. Optokoppler Ausgang	20
4.12. To-Do	21
5. Messungen	22
5.1. Landelichter	22
5.2. Blinklichter Rot	23
5.3. Blinklichter Weiß	23
5.4. Positionslichter	24
5.5. Stromaufnahme der gesamten Schaltung (I_{ges})	24
5.5.1. Alle LEDs eingeschalten	24
5.5.2. Nur Positionslichter, ohne Landelichter	24
5.5.3. alle LEDs ausgeschalten	25
5.5.4. Akku leer Abschaltung, μC im Schlafmodus	25
5.6. U_{bat} bei Akku leer Abschaltung (alle LEDs aus, μC im Schlafmodus)	25
5.7. To-Do	25
6. Softwarebeschreibung	25
6.1. Allgemeines	25
6.2. Eigenschaften der Software	26
6.3. Softwaredokumentation	26
7. Links	27
8. Versionsgeschichte	27
8.1. Software Standard (20120612)	27
8.2. Hardware Standard (Rev A – 20130624)	27
8.3. Hardware Standard (Rev B – 20130809)	28
8.4. Software Standard (20130902)	28
8.5. Dokumentation Standard (20131112)	28
8.6. Dokumentation Standard (20131212)	28
8.7. Dokumentation Standard (20140614)	28
8.8. Dokumentation Standard (20140830)	29

9. Gewährleistungsausschluß	29
10. Haftungsbegrenzung	29
11. Interpretation von 9 und 10	30
12. Dokumentation To-Do	30
A. Fertigungsunterlagen	30
A.1. Einkaufs Liste	30
A.1.1. Optionale Bauteile	30
A.1.2. Zusätzlich benötigte Teile	33
A.2. Bauteil Liste	34
A.3. Weitere Unterlagen	37
B. Softwaredokumentation	42

1. Allgemeines



Beim Conrad sah ich ein fertiges [Lichtsystem](#) und dachte mir: gute Sache, schnell eingebaut, nicht viel Arbeit und funktioniert sofort.

Nach ein paar Flügen mußte ich dann feststellen, daß es doch nett wäre, die Lichter über den Sender Ein- und Auszuschalten. Außerdem sollte die Stromversorgung vom Empfänger Akku getrennt, über einen kleinen LiPo Akku erfolgen. Somit war auch noch ein Spannungsregler notwendig.

Also warum nicht gleich das Modul durch einen Eigenbau ersetzen, der alles nötige auf einer Platine hat?

Falls größeres Interesse an dieser Schaltung besteht, bin ich gerne bereit mehrere Platinen fertigen zu lassen und die Schaltung als Bausatz abzugeben.

Für Fragen, Anregungen, Erfahrungsaustausch, Probleme, Beschwerden, etc. oder um über das Projekt zu diskutieren, schreiben Sie doch in mein [Forum](#) oder eine E-Mail (webmaster@vpippan.at) direkt an mich.

Die Downloads für die Hardware und Software zu diesem Projekt finden Sie auf der Projektseite: http://www.vpippan.at/index.php?pid=Elektronik_Projekte_RcLichtsteuerung.

2. Hardwarebeschreibung

2.1. Allgemeines

Als μC dient ein ATtiny2313 der für diesen Zweck mehr als genug Pins zur Verfügung stellt. Da diese Schaltung keine besonderen Timing Anforderungen hat, verzichte ich auf einen externen Quarz (Gewicht und Platz). Der interne Taktgenerator ist hier ausreichend. Die LEDs werden dann vom μC über MOSFETs geschaltet. Zusätzlich überwacht der μC die Spannung des Akkus, damit ein zu tiefes entladen des LiPo vermieden wird. Bei zu geringer Akku Spannung wird die Beleuchtung einfach abgeschaltet.

Da Sicherheit in der Luftfahrt immer an erster Stelle steht, ist das gesamte Beleuchtungssystem über Optokoppler vom Empfänger getrennt und verfügt über eine eigene Stromversorgung. Die Vorwiderstände an den Optokopplern (R17, R20, R21) sind für eine max. Spannung von 6 V ausgelegt. Bei Verwendung von höheren Versorgungsspannungen des Empfängers im Modell sind diese Widerstände zu ändern!

Optional enthält die Schaltung eine Schnittstelle zum Jeti Duplex System. Die dafür notwendige Schaltung ist bereits auf der Platine integriert und kann bei Bedarf bestückt werden. Diese Schnittstelle ist allerdings als experimentell zu betrachten, da ich mit dem Jeti System noch keinerlei Erfahrung habe und selber erst alles nötige herausfinden muß! Natürlich kann die Schaltung auch ohne das Jeti System verwendet werden.

Die Schaltung kann auch als Schaltmodul für DC Verbraucher verwendet werden. Dazu müssen nur die LED Vorwiderstände (R3 bis R10) überbrückt werden. Zu Beachten ist dabei jedoch, daß die max. möglichen Ströme (Spannungsregler, MOSFET's und Leiterbahnbreiten) nicht überschritten werden. Beispiel-

berechnungen sind in diesem Dokument unter Punkt 4 zu finden. Durch die Verwendung eines μC ist es auch möglich, verschiedene Schaltabläufe (automatisch ablaufend) über den Sender zu aktivieren. Zur Steuerung eines Ablaufes muß nur der μC entsprechend programmiert werden.

Um den Platzbedarf und das Gewicht möglichst gering zu halten, ist die gesamte Schaltung in SMD Technik aufgebaut. Für einen der üblichen PCB Fertiger sollte die Platine eigentlich kein Problem darstellen. Die verwendeten Leiterbahnbreiten (0,5 mm) und Leiterbahnabstände (0,4 mm) sind selbst mit meiner primitiven Ausrüstung sicher zu beherrschen. Mit etwas Erfahrung stellen auch die Lötarbeiten (trotz der SMD Bauweise) kein Problem dar. Es ist lediglich darauf zu achten sauber zu arbeiten. Sehr hilfreich zur Kontrolle ist eine Lupe. Einzig die Optokoppler stellten sich als schwierig zu löten heraus, aber auch dies ist ohne Kurzschlüsse zu schaffen.

Wie auf den Bildern 1 zu sehen können die Anschlusskabel direkt angelötet werden, oder die Platine wird mit Stiftleisten als Steckverbindung aufgebaut. Als Gehäuse für die fertige Platine dient ein Stück Schrumpfschlauch.

2.2. Rev A

2.2.1. Technische Daten

- Abmessungen der Platine: 52 mm x 30 mm
- Masse: 10 g (ohne Akku, LEDs und Kabel)
- Leiterbahnbreiten 0,5 mm und 0,7 mm mm (für Versorgungsleitungen) / Leiterbahnabstände 0,4 mm
- Durchkontaktierung außen 1,4 mm und Bohrung 0,8 mm
- Ein Eingang für das Empfängersignal, über Optokoppler getrennt (3 V bis 6 V)
- Vier Ausgangskanäle, zwei Anschlüsse pro Kanal
- Ein Ausgang für Jeti Telemetrie, über Optokoppler getrennt (3 V bis 6 V)

- Eigene Spannungsversorgung zur Trennung vom Empfängerstromkreis (5,7 V bis 10 V)
- Überwachung der Akkuspannung

2.3. Rev B

Im Vergleich zu Rev A hat sich nur wenig geändert. Der 10 polige ISP Stecker ist durch einen 6 poligen ersetzt worden. Weiters wurde die Spannungsversorgung des μC mit einem Stützkondensator und einer Filterspule versehen, um störendes Rauschen (verursacht durch den μC) besser zu unterbinden. Die Beschaltung der Optokoppler (für Datenübertragung zum Jeti System) war in Rev A falsch. Dies fiel bis jetzt nicht auf, da ich noch keine Zeit hatte damit zu experimentieren. Nun sollte es stimmen und funktionieren. Allerdings ist es noch nicht getestet!

2.4. Mögliche Änderungen für eine neue Version

- kleinere Stecker für Lichtanschlüsse und ISP
- Version ohne Jeti Unterstützung, da hier kaum notwendig
- Beim Transistor für Positionslichter Platz für einen dritten Widerstand und Anschluss schaffen (z. B. für ein weißes Licht)
- 2. Empfängersignal Eingang (z. B. Für das Fahrwerksignal → Automatische Ansteuerung vom Landelicht)
- Schaltung mit 0805 Bauteilen und kleineren Leiterbahnbreiten neu Layouten
- Zusätzliche Ausgangskanäle
- Verpolungsschutz und Sicherung bei der Spannungsversorgung
- μC Spannungsversorgung verbessern → so wie beim GPS Logger (LC Filter)

- Filter an Lichtausgängen um Einstreuungen durch lange LED Leitungen zu verhindern
- Statt Spannungsteiler ein Trimpoti für bessere Einstellung der Akkuüberwachung

2.5. Fertigungsunterlagen

Alle benötigten Informationen zum Aufbau des Moduls finden sich im Anhang A.

3. Messungen am Conrad Modul

Gemessen wurden U_{led} und I_{led} an jeweils einer einzelnen LED. Mit bekannter U_{bat} läßt sich daraus der Vorwiderstand berechnen.

$$R = \frac{U_{bat} - U_{led}}{I_{led}} \quad (1)$$

U_{bat} wird mit 5 V als konstant angenommen. Die Widerstände für die LEDs wurden etwas kleiner gewählt, da die Conrad Schaltung mit bis zu 6 V betrieben werden kann \Rightarrow höhere $I_{led} \Rightarrow$ bei meiner Schaltung ist aber $U_{bat} = 5 \text{ V}$ fix.

3.1. Landelichter

Folgende Werte wurden gemessen:

- $U_{led} = 3,15 \text{ V}$
- $I_{led} = 8,8 \text{ mA}$

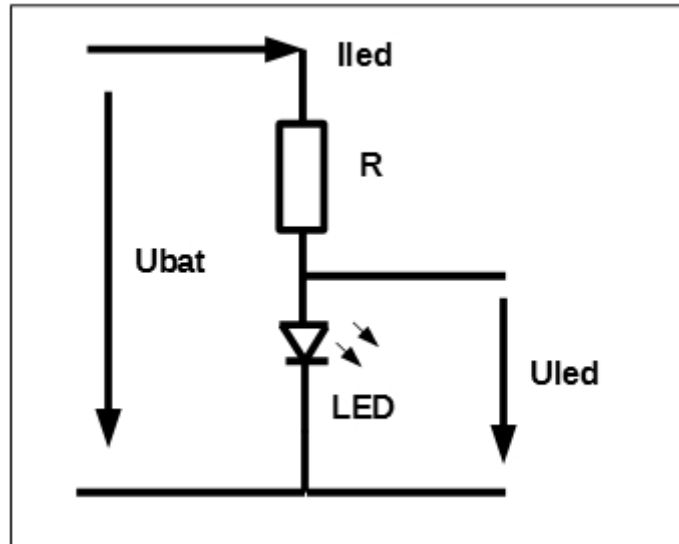


Abbildung 1: Messungen am Conrad Modul

Aus Gleichung 1 ergibt sich ein Vorwiderstand für die LED von $R = 210,23 \Omega$. Daher wird ein Widerstand aus der E12 Reihe mit $R = 180 \Omega$ gewählt.

Der Strom durch den Schalttransistor ergibt sich somit aus $2 \cdot I_{led}$ zu $I_{Transistor} = 17,6 \text{ mA}$.

3.2. Blinklichter Rot

Folgende Werte wurden gemessen:

- $U_{led} = 1,9 \text{ V}$
- $I_{led} = 8,8 \text{ mA}$

Aus Gleichung 1 ergibt sich ein Vorwiderstand für die LED von $R = 352,27 \Omega$. Daher wird ein Widerstand aus der E12 Reihe mit $R = 330 \Omega$ gewählt.

Der Strom durch den Schalttransistor ergibt sich somit aus $2 \cdot I_{\text{led}}$ zu $I_{\text{Transistor}} = 17,6 \text{ mA}$.

3.3. Blinklichter Weiß

Folgende Werte wurden gemessen:

- $U_{\text{led}} = 3 \text{ V}$
- $I_{\text{led}} = 12,45 \text{ mA}$

Aus Gleichung 1 ergibt sich ein Vorwiderstand für die LED von $R = 160,64 \Omega$. Daher wird ein Widerstand aus der E12 Reihe mit $R = 150 \Omega$ gewählt.

Der Strom durch den Schalttransistor ergibt sich somit aus $2 \cdot I_{\text{led}}$ zu $I_{\text{Transistor}} = 24,9 \text{ mA}$.

3.4. Positionslichter

Folgende Werte wurden gemessen:

- $U_{\text{led-Grün}} = 3,15 \text{ V}$
- $I_{\text{led-Grün}} = 12,7 \text{ mA}$
- $U_{\text{led-Rot}} = 2 \text{ V}$
- $I_{\text{led-Rot}} = 12,3 \text{ mA}$

Aus Gleichung 1 ergibt sich ein Vorwiderstand für die LED von $R_{\text{Rot}} = 243,9 \Omega$ und $R_{\text{Grün}} = 149,61 \Omega$. Daher wird ein Widerstand aus der E12 Reihe mit $R_{\text{Rot}} = 220 \Omega$ und $R_{\text{Grün}} = 120 \Omega$ gewählt.

Der Strom durch den Schalttransistor ergibt sich somit aus $I_{\text{led-Rot}} + I_{\text{led-Grün}}$ zu $I_{\text{Transistor}} = 25 \text{ mA}$.

Die gesamte Stromaufnahme aller LEDs beträgt dann $I_{\text{led-Gesamt}} = 85,1 \text{ mA}$.

4. Berechnungen

Die hier gezeigten Gleichung und Berechnungen sollen auch als Beispiel dienen, um die Schaltung einfach an eigene Bedürfnisse anzupassen.

4.1. Annahmen

- LiPo Akku voll: 8,4 V (4,2 V pro Zelle)
- LiPo Akku leer: 6,4 V (3,2 V pro Zelle)
- Verwendung der internen Referenz für den analog Komparator: 1,11 V bei 5 V Versorgung und 25 °C (laut Datenblatt)

Die Spannung der internen Referenz schwankt mit der Versorgungsspannung und über die Temperatur. Dies wird hier für die Rechnung aber vernachlässigt. Die interne Referenz wird verwendet um Bauteile und Platz einzusparen, auch wenn diese Lösung nicht optimal ist.

4.2. Gleichungen für einen unbelasteten Spannungsteiler

$$U = \frac{R_1 + R_x}{R_x} \cdot U_x \quad (2)$$

$$R_x = \frac{U_x \cdot R_1}{U - U_x} \quad (3)$$

$$U_x = \frac{R_x}{R_1 + R_x} \cdot U \quad (4)$$

Gleichungen 3 und 4 erhält man einfach durch Umformen aus Gleichung 2.

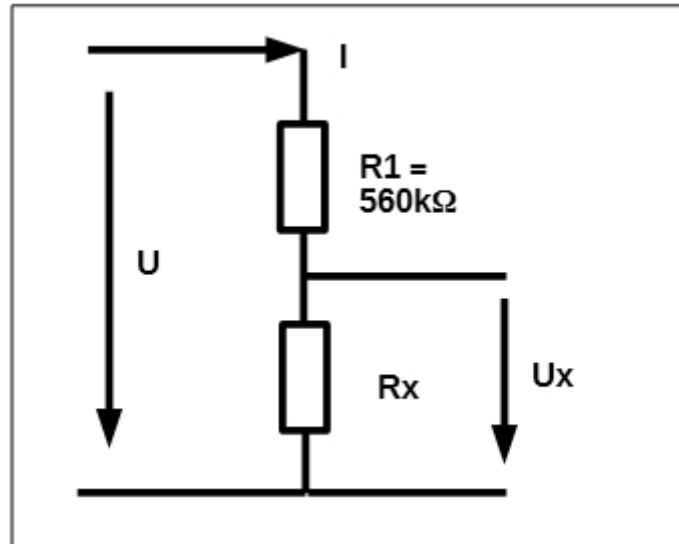


Abbildung 2: Unbelasteter Spannungsteiler

4.3. Notwendiger R_x zum Schalten des Komparators (Akku leer Erkennung)

Die Referenzspannung des Komparators beträgt $1,11\text{ V}$. Um einen leeren Akku zu erkennen muß R_x daher so gewählt werden, daß bei leerem Akku ($U = 6,4\text{ V}$) die Spannung $U_x = 1,11\text{ V}$ erreicht oder unterschritten wird. Aus der Gleichung 3 erhält man mit diesen Werten einen Widerstand $R_x = 117,504\text{ k}\Omega$.

Somit wird $R_x = 115\text{ k}\Omega$ gewählt (da dieser Wert in der E12 Normreihe nicht vorkommt, werden 2 Widerstände mit $100\text{ k}\Omega$ und $15\text{ k}\Omega$ in Reihe geschaltet).

4.4. Spannungen U_x am Komparator (Überprüfen der max. Komparator Eingangsspannung und Schaltschwelle)

Zur Berechnung der Spannungen $U_{x,voll}$ und $U_{x,leer}$ wird die Gleichung 4 verwendet.

Bei voll geladenem Akku ($U = 8,4 \text{ V}$) ergibt sich die Spannung am Komparator-eingang $U_{x,voll} = 1,431 \text{ V}$. Die max. Eingangsspannung (5 V) vom Komparator wird also nicht überschritten.

Bei leerem Akku ($U = 6,4 \text{ V}$) errechnet sich die Spannung $U_{x,leer} = 1,09 \text{ V}$. Der Komparator schaltet also bei leerem Akku sicher um.

4.5. Akkuspannung bei der Komparator Schaltschwelle

Der Komparator schaltet bei einer Spannung $U_x = 1,11 \text{ V}$ (= gleich seiner Referenzspannung) an seinem Eingang um. Aus Gleichung 2 ergibt sich in diesem Fall eine Akkuspannung von $U = 6,515 \text{ V}$. Was eine Spannung von $3,257 \text{ V}$ pro Zelle entspricht.

Der Komparator schaltet also bereits um, wenn der Akku noch nicht ganz leer ist. Da bei dieser Rechnung die Widerstandstoleranzen und Schwankungen der Referenz aber nicht berücksichtigt sind, ist ein etwas früheres umschalten durchaus gewünscht.

4.6. Akku Abschaltspannungen unter Berücksichtigung der Widerstandstoleranzen und Temperaturdrift der internen Referenzspannung

Die Spannung U am Akku wird wieder durch die Gleichung 2 berechnet. Die Änderungen der Akkuspannung, verursacht durch Widerstandstoleranzen und Temperaturdrift der Referenz werden mit der Formel zur Fehlerfortpflanzung berechnet. Siehe dazu auch: [Wikipedia - Fehlerfortpflanzung](#).

Es ergeben sich die folgenden Spannungen:

- $U_{\max} = 6,652 \text{ V}$ (3,312 V pro Zelle) mit Widerstandstoleranzen 1 % und 50°C
- $U_{\min} = 6,319 \text{ V}$ (3,159 V pro Zelle) mit Widerstandstoleranzen 1 % und -10°C

Bei den obigen Zahlen handelt es sich um die Grenzwerte. Die Akkuspannung bei der ein Akku als leer erkannt wird liegt also irgendwo zwischen diesen beiden Werten. Somit ist sichergestellt, daß auch bei abweichenden Widerstandswerten (innerhalb der Toleranz) und Temperaturdrift der Referenzspannung (im Bereich -10°C bis 50°C) der Akku sicher vor Tiefentladung geschützt wird. Für die detaillierte Berechnung siehe die Maxima Datei.

4.7. Strom I durch den Komparator Spannungsteiler bei vollem Akku

Mit der Formel $I = \frac{U}{R_1 + R_x}$ Strom durch den Komparatorspannungsteiler bestimmt. Es ergibt sich ein Wert von $I = 12,444 \mu\text{A}$.

Dies bedeutet, daß dem Akku innerhalb von 24 Stunden 0,299 mA h entnommen werden (zusätzlich zum Stromverbrauch des μC im Schlafmodus). Für einen Flugtag kein Problem, jedoch sollte der Akku nicht dauerhaft mit der Schaltung verbunden bleiben!

4.8. Maximaler Strom durch den Spannungsregler

Zur Berechnung des maximal möglichen Stromes durch den Spannungsregler finden sich im Datenblatt die folgenden Formeln:

$$P_{D,max} = (V_{ein,max} - V_{aus,min}) \cdot I_{aus,max} + V_{ein,max} \cdot I_{GND} \quad (5)$$

$$R_{\vartheta,JA} = \frac{150^{\circ}\text{C} - T_A}{P_D} \quad (6)$$

Folgende Werte werden angenommen:

- $V_{\text{ein,max}} = 8,4 \text{ V}$ (maximale Akkuspannung)
- $V_{\text{aus,min}} = 4,755 \text{ V}$ (aus dem Datenblatt)
- $I_{\text{aus,max}} = 0,5 \text{ A}$
- $I_{\text{GND}} = 0,02 \text{ A}$ (aus Datenblatt)
- $T_A = 35^{\circ}\text{C}$ (angenommene Umgebungstemperatur am Einbauort der Schaltung)

Mit diesen Werten erhält man aus der Gleichung 5: $P_D = 2 \text{ W}$ und aus Gleichung 6: $R_{\vartheta,JA} = 57,5 \frac{^{\circ}\text{C}}{\text{W}}$.

Um den Spannungsregler innerhalb der zulässigen Temperaturen zu halten, muß der berechnete Wert von $R_{\vartheta,JA}$ größer als $R_{\vartheta,JA} = 60 \frac{^{\circ}\text{C}}{\text{W}}$ für das Gehäuse (diesen Wert bekommt man aus dem Datenblatt) sein!

Wie man sieht ist das bei meiner Schaltung knapp nicht der Fall. Allerdings ist der Spannungsregler mit dem GND Pin ebenfalls an der Platine angelötet und wird daher über diesen Pfad zusätzlich gekühlt (die Platine wirkt wie ein Kühlkörper). Aus diesem Grund sollte die Schaltung bei den oben gegebenen Bedingungen noch problemlos funktionieren.

Für die Berechnung habe ich die Maximal und Minimal Werte aus dem Datenblatt verwendet. Führt man die Berechnung mit den typischen Werten durch, dann sieht das Ergebnis besser aus, es enthält aber keine Sicherheiten mehr.

Will man die Schaltung bei höheren Eingangsspannungen $V_{\text{ein,max}}$ und/oder höheren Umgebungstemperaturen T_A mit dem maximalen Strom $I_{\text{aus,max}} = 0,5 \text{ A}$ betreiben, dann ist am Spannungsregler unbedingt ein Kühlkörper zu montieren! Die dafür benötigte Berechnung ist im Datenblatt gegeben.

4.9. Maximale Verlustleistung in den Schalt Transistoren

Für die Berechnung der Leistung wird folgende Formel verwendet:

$$P = I^2 \cdot R = \frac{U^2}{R} = U \cdot I \quad (7)$$

Werte aus dem Datenblatt:

- $P_{\text{tot}} = 0,83 \text{ W}$ bei einer Temperatur von 25°C am Lötunkt
- $R_{\text{DS,on}} = 1050 \text{ m}\Omega$ bei $V_{\text{GS}} = 5 \text{ V}$, $I_{\text{DS}} = 100 \text{ mA}$ und $T_{\text{J}} = 25^\circ\text{C}$ (kleinere Ströme sind im Datenblatt nicht mehr eingezeichnet, daher wird mit diesem Wert gerechnet)

4.10. Maximale Verlustleistung in den LED Vorwiderständen

Die verwendeten Widerstände sind für eine maximale Verlustleistung von 250 mW geeignet.

4.11. Vorwiderstände für die Optokoppler

4.11.1. Optokoppler Eingang

Laut Auskunft der Firma Jeti dürfen am Empfänger Data Ausgang maximal 5 mA ($= I_{\text{led}}$) entnommen werden. Die Auslegung der Vorwiderstände für die Optokoppler erfolgt mit der Annahme einer Empfängerversorgung von max. $U_{\text{bat}} = 6 \text{ V}$.

Zur Berechnung von Optokopplern siehe auch: Mikrocontroller.net - Optokoppler.

Aus dem Datenblatt (Fig. 2) erhält man für $I_{\text{led}} = 5 \text{ mA}$ eine Spannung $U_{\text{led}} = 1,07 \text{ V}$ bei 25°C .

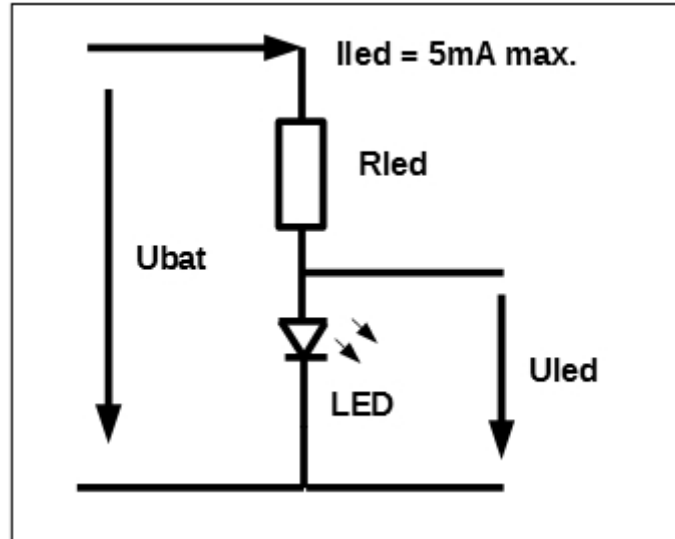


Abbildung 3: Spannungsteiler für Optokoppler

Mit der Formel

$$R_{led} = \frac{U_{bat} - U_{led}}{I_{led}} \quad (8)$$

läßt sich der benötigte Vorwiderstand zur $R_{led} = 986 \Omega$ berechnen. Es wird der nächst größere Widerstand aus der E12 Normreihe gewählt. Daraus folgt $R_{led} = 1 \text{ k}\Omega$.

Aus der Gleichung

$$I_{led} = \frac{U_{bat} - U_{led}}{R_{led}} \quad (9)$$

kann nun der Strom durch die LED mit dem gewählten Widerstand berechnet werden. Es ergeben sich die folgenden Werte:

- $I_{\text{led}} = 4,93 \text{ mA}$ für $U_{\text{bat}} = 6 \text{ V}$ und mit $U_{\text{led}} = 1,07 \text{ V}$
- $I_{\text{led}} = 1,95 \text{ mA}$ für $U_{\text{bat}} = 3 \text{ V}$ und mit $U_{\text{led}} = 1,05 \text{ V}$

Am Optokoppler darf $I_{\text{led}} = 60 \text{ mA}$ (für DC) nicht überschritten werden, der begrenzende Faktor sind hier allerdings die 5 mA vom Jeti Empfänger. Wie aus obigen Ergebnissen zu sehen, werden diese 5 mA auch bei max. Spannung der Empfängerbatterie nicht überschritten.

4.11.2. Optokoppler Ausgang

Um den Eingang als HIGH zu erkennen, müssen am Eingangspin des μC mindestens $0,6 \text{ V}$ anliegen.

Der dafür benötigte Widerstand lässt sich über die Formel

$$R_{\text{aus}} = \frac{U_{R,\text{aus}}}{CTR \cdot I_{\text{led}}} \quad (10)$$

berechnen. Mit dem minimalen $I_{\text{led}} = 1,95 \text{ mA}$ und einem CTR von 5 für den gewählten Optokoppler erhält man aus der Gleichung 10: $R_{\text{aus}} = 61,54 \Omega$.

Um sicher über der Schaltschwelle des μC zu liegen (Widerstandstoleranzen, Spg. Schwankungen, etc.) wird ein Sicherheitsfaktor von 4 gewählt.

Damit ergibt sich $R_{\text{aus}} = 246,16 \Omega$. Aus der E12 Normreihe wird $R_{\text{aus}} = 220 \Omega$ gewählt.

Die Spannung am Widerstand R_{aus} und damit am Eingang des μC kann über die Formel

$$U_{R,\text{aus}} = R_{\text{aus}} \cdot CTR \cdot I_{\text{led}} \quad (11)$$

berechnet werden. Es ergeben sich die folgenden Werte:

- $U_{R,aus} = 2,14 \text{ V}$ für $I_{led} = 1,95 \text{ mA}$
- $U_{R,aus} = 5 \text{ V}$ für $I_{led} = 4,93 \text{ mA}$ (die Versorgungsspannung von 5 V kann nicht überschritten werden)

Der Strom $I_{R,aus}$ durch den Widerstand R_{aus} wird über die Formel

$$I_{R,aus} = CTR \cdot I_{led} \quad (12)$$

oder über die Formel

$$I_{R,aus} = \frac{5 \text{ V}}{R_{aus}} \quad (13)$$

berechnet, falls man aus Gleichung 11 das Ergebniss $U_{R,aus} > 5 \text{ V}$ erhält.

Für $I_{R,aus}$ erhält man also die folgenden Werte:

- $I_{R,aus} = 9,75 \text{ mA}$ für $I_{led} = 1,95 \text{ mA}$
- $I_{R,aus} = 22,72 \text{ mA}$ für $I_{led} = 4,93 \text{ mA}$

Der max. Kollektorstrom (für DC) am Ausgang des Optokoppler beträgt $I_{C,max} = 50 \text{ mA}$ (laut Datenblatt). Somit sind wir auch am Ausgang des Optokopplers auf der sicheren Seite.

4.12. To-Do

- Was passiert bei Temp. Änderung mit Strom am Optokoppler Eingang (wird er zu groß)?
- Optokoppler so wie verbaut OK → Wegen Jeti Protokoll?
- Maximal möglichen Strom der Mosfets berechnen
- LED Vorwiderstände so berechnen, daß alle LED gleich hell sind
- Berechnungen der Vorwiderstände für andere LED Typen einfügen

- Berechnung der Caps. für Quarz → siehe Atmel App Note
- Verwendung des Analog Komparator → siehe Atmel App Note

5. Messungen

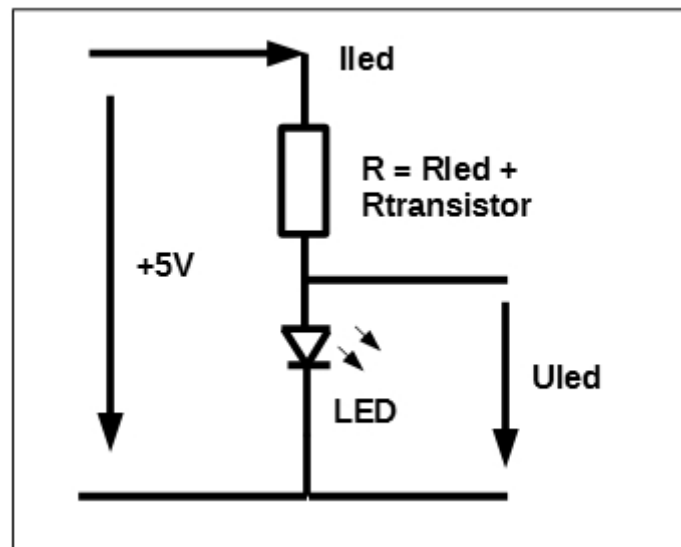


Abbildung 4: Messungen am fertigen Modul

Gemessen wurden U_{led} und I_{led} an jeweils einer einzelnen LED.

5.1. Landelichter

Folgende Werte wurden gemessen:

- $U_{\text{led}} = 3,15 \text{ V}$
- $I_{\text{led}} = 9,98 \text{ mA}$

Der Strom durch den Schalttransistor ergibt sich somit aus $2 \cdot I_{\text{led}}$ zu $I_{\text{Transistor}} = 19,96 \text{ mA}$.

5.2. Blinklichter Rot

Folgende Werte wurden gemessen:

- $U_{\text{led}} = 2,02 \text{ V}$
- $I_{\text{led}} = 9 \text{ mA}$

Der Strom durch den Schalttransistor ergibt sich somit aus $2 \cdot I_{\text{led}}$ zu $I_{\text{Transistor}} = 18 \text{ mA}$.

5.3. Blinklichter Weiß

Folgende Werte wurden gemessen:

- $U_{\text{led}} = 3,12 \text{ V}$
- $I_{\text{led}} = 11 \text{ mA}$

Der Strom durch den Schalttransistor ergibt sich somit aus $2 \cdot I_{\text{led}}$ zu $I_{\text{Transistor}} = 22 \text{ mA}$. Die EIN Zeit der Lichter ist 134 ms und die AUS Zeit beträgt 130 ms (diese Werte gelten für die Blinkfolge meiner Software. Die Zeiten können in der Software geändert werden).

5.4. Positionslichter

Folgende Werte wurden gemessen:

- $U_{\text{led-Grün}} = 3,07 \text{ V}$
- $I_{\text{led-Grün}} = 15,22 \text{ mA}$
- $U_{\text{led-Rot}} = 1,99 \text{ V}$
- $I_{\text{led-Rot}} = 13,29 \text{ mA}$

Der Strom durch den Schalttransistor ergibt sich somit aus $I_{\text{led-Rot}} + I_{\text{led-Grün}}$ zu $I_{\text{Transistor}} = 28,51 \text{ mA}$.

Die gesamte Stromaufnahme aller LEDs beträgt dann $I_{\text{led-Gesamt}} = 88,47 \text{ mA}$.

5.5. Stromaufnahme der gesamten Schaltung (I_{ges})

5.5.1. Alle LEDs eingeschalten

- 52 mA minimal
- 110 mA maximal
- 79 mA effektiv

5.5.2. Nur Positionslichter, ohne Landelichter

- 30 mA minimal
- 88 mA maximal
- 56 mA effektiv

5.5.3. alle LEDs ausgeschalten

- 3,6 mA minimal
- 32 mA maximal (Strom über den Optokoppler Ausgang während der Eingangsimpuls auf HIGH ist)
- 8,2 mA effektiv

5.5.4. Akku leer Abschaltung, μC im Schlafmodus**5.6. U_{bat} bei Akku leer Abschaltung (alle LEDs aus, μC im Schlafmodus)****5.7. To-Do**

- Messen von $U_{\text{Transistor}}$
- Berechnen von $R_{\text{Transistor}}$ und Vergleich mit Wert aus Datenblatt
- Berechnen Transistor Verlustleistung
- Signale nach Optokoppler?
- Ripple auf Spannungsversorgung? \rightarrow Weil keine Caps bei μC
- Stromaufnahme der Schaltung im Normalbetrieb / im Schlafmodus

6. Softwarebeschreibung**6.1. Allgemeines**

Die Software ist in C geschrieben und steht unter der GPLv3. Wie bei der Hardware bereits erwähnt läuft die Software auf einem Attiny2313.

Die aktuelle Software ist geeignet für die Hardware Versionen Rev. A und Rev. B.

6.2. Eigenschaften der Software

- Erkennung des Empfänger Eingangssignals über einen eigenen Timer
- Im Sourcecode einstellbare Schaltzeitpunkte
- Verwendung des Watchdog Timers um zu erkennen ob ein Empfänger Eingangssignal vorhanden ist
- Wenn kein Empfängersignal vorhanden: Abschalten der Beleuchtung und warten auf ein neues Empfängersignal
- Überwachung der Akkuspannung mit einem Komparator
- Bei Akku Unterspannung: Abschalten der Beleuchtung und in den Schlafmodus gehen
- Erzeugen der Blinkfrequenz über einen eigenen Timer
- Im Sourcecode konfigurierbare Blinkfolge und Blinkfrequenz
- Schlafmodus (zum Energiesparen) wenn nichts zu berechnen ist

6.3. Softwaredokumentation

Eine vollständige Dokumentation der Software findet sich im Anhang B oder als [HTML Version](#) auf meiner Website.

7. Links

- [Beschreibung \(Englisch\) vom Jeti Protokoll](#)
- [Positionslicht \(Wikipedia\)](#)
- [Vorschriften über die von Luftfahrzeugen zu führenden Lichter \(Anlage 1 zur Luftverkehrs-Ordnung \(zu §§ 17 und 19 Abs. 7 LuftVO\)\) \(gesetz-im-internet.de\)](#)
- [Licht an! Was bedeutet die Beleuchtung am Flugzeug? \(aviation-friends-hamburg-forum.de\)](#)
- [Flugzeugbeleuchtung \(wiki.flightgear.org\)](#)
- [Schnelleinführung in die Modellflug Beleuchtung \(optotronix.de\)](#)
- [Beleuchtung \(aviation4u.de\)](#)

8. Versionsgeschichte

8.1. Software Standard (20120612)

- Ursprünglich veröffentlichte Version

8.2. Hardware Standard (Rev A – 20130624)

- Ursprünglich veröffentlichte Version

8.3. Hardware Standard (Rev B – 20130809)

- Fehler bei der Jeti Optokoppler Beschaltung korrigiert
- ISP Stecker von 10 polig auf 6 polig geändert
- Stützkondensator und Filterspule zur Spannungsversorgung des μC hinzugefügt

8.4. Software Standard (20130902)

- Blinken, Akku Spannungsüberwachung und Erkennung ob Empfängersignal vorhanden hinzugefügt

8.5. Dokumentation Standard (20131112)

- Berechnung für Spannungsregler hinzugefügt, maximale Eingangsspannung im Schaltplan geändert

8.6. Dokumentation Standard (20131212)

- Datum zur Hardware Revision hinzugefügt
- Bemaßungen von Layer Comments User auf Layer Drawings User verschoben

8.7. Dokumentation Standard (20140614)

- Neuer Projektname und daher Dateien entsprechend umbenannt und geändert
- Projektversionierung nochmals geändert
- Dokumentation mit \LaTeX geschrieben anstatt eines Spreadsheets

8.8. Dokumentation Standard (20140830)

- Fehler in Tabellenbeschriftung korrigiert

9. Gewährleistungsausschluß

Es besteht keinerlei Gewährleistung für das Projekt, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Urheberrechtsinhaber und/oder Dritte das Projekt so zur Verfügung, „wie es ist“, ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich – aber nicht begrenzt auf – die implizite Gewährleistung der Marktreife oder der Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Projekts liegt bei Ihnen. Sollte sich das Projekt als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.

10. Haftungsbegrenzung

In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Urheberrechtsinhaber oder irgendein Dritter, der das Projekt wie oben erlaubt modifiziert oder übertragen hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher allgemeiner oder spezieller Schäden, Schäden durch Seiteneffekte (Nebenwirkungen) oder Folgeschäden, die aus der Benutzung des Projekts oder der Unbenutzbarkeit des Projekts folgen (einschließlich – aber nicht beschränkt auf – Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder anderen getragen werden müssen, oder dem Unvermögen des Projekts, mit irgendeinem anderen Projekt zusammenzuarbeiten), selbst wenn ein Urheberrechtsinhaber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.

11. Interpretation von 9 und 10

Sollten der o.a. Gewährleistungsausschluß und die o.a. Haftungsbegrenzung aufgrund ihrer Bedingungen gemäß lokalem Recht unwirksam sein, sollen Bewertungsgerichte dasjenige lokale Recht anwenden, das einer absoluten Aufhebung jeglicher zivilen Haftung in Zusammenhang mit dem Projekt am nächsten kommt, es sei denn, dem Projekt lag eine entgeltliche Garantieerklärung oder Haftungsübernahme bei.

12. Dokumentation To-Do

- Bedienungsanleitung (Entwicklungsumgebung aufsetzen für Software und Hardware, µC Firmware installieren, Anschluss des Moduls, etc.)
- Schriftart vom V. Pippa Schriftzug überall ändern (Kapitälchen und kursiv)

A. Fertigungsunterlagen

A.1. Einkaufs Liste

A.1.1. Optionale Bauteile

Falls die Schaltung mit der optionalen Jeti Duplex Kommunikation aufgebaut werden soll, dann gelten in der folgenden Tabelle (Spalte Anzahl) die Werte in Klammern.

Alle berechneten Preise sind inklusive der Option. Wenn also auf die optionale Jeti Duplex Kommunikation verzichtet wird, dann sinkt der Preis für das Modul.

Bauteil	Wert	Hersteller	Farnell Best. Nr.	Herstellerbezeichnung	Stückpreis/€	Anzahl	Preis/€
Kapazität	10 µF	Multicomp	1759434	MCCA000554	0,057	1	0,057
Kapazität	4,7 µF	Multicomp	1759444	MCCA000563	0,042	1	0,042
Kapazität	100 nF	Multicomp	1759167	MCCA000296	0,01	1	0,01
Kapazität	22 pF	Multicomp	9406441	MCCA000481	0,066	0 / (2)	0,132
Induktivität	10 µH	Taiyo Yuden	2112893	LB2012T100M	0,084	1	0,084
Spg. Regler	NCP5501DT50G	ON Semiconductor	1703369	NCP5501DT50G	0,714	1	0,714
µC	ATTINY2313A-SU	Atmel	1841617	ATTINY2313A-SU	2,67	1	2,67
Quarz	8 MHz					0 / (1)	0
Optokoppler	VO23AT	Vishay	1870802	VO23AT	0,625	1 / (3)	1,875
Widerstand	560 kΩ	Welwyn	1100264	WCR1206-560KFI	0,021	1	0,021
Widerstand	100 kΩ	Multicomp	9335773	MC 0.125W 1206 1 % 100K	0,036	1	0,036
Widerstand	180 Ω	Welwyn	1100171	WCR1206-180RFI	0,019	2	0,038
Widerstand	330 Ω	Vishay Draloric	1470008	CRCW1206330RFKEA	0,027	2	0,054
Widerstand	220 Ω	Welwyn	2078997	ASC1206-220RFT5	0,023	2 / (3)	0,069
Widerstand	120 Ω	Welwyn	1100167	WCR1206-120RFI	0,032	1	0,032
Widerstand	150 Ω	Welwyn	1100169	WCR1206-150RFI	0,013	2	0,026
Widerstand	100 Ω	Vishay Draloric	1469977	CRCW1206100RFKEA	0,032	4	0,128
Widerstand	10 kΩ	Vishay Draloric	1469970	CRCW120610K0FKEA	0,027	1	0,027
Widerstand	1 kΩ	Vishay Draloric	1469965	CRCW12061K00FKEA	0,027	1 / (2)	0,054
Widerstand	15 kΩ	Welwyn	1100223	WCR1206-15KFI	0,032	1	0,032
Widerstand	820 Ω	Vishay Draloric	1470029	CRCW1206820RFKEA	0,027	0 / (1)	0,027
MOSFET	2N7002E	NXF	1894721	2N7002E	0,036	4	0,144
Stiftleiste							0
PCB		V. PIPPAN		RCLS_S_PCB_RevB	10	1	10

Der Gesamtpreis mit allen notwendigen und optionalen Bauteilen ergibt sich somit zu 16,272€.

Falls die Bauteile nicht bereits vorhanden sind und die Mindestbestellmengen bestellt werden müssen, dann ergeben sich die folgenden Preise:

Bauteil	Wert	Min. Bestellmenge	Preis (Mindestbestellung)/€
Kapazität	10 μ F	10	0,57
Kapazität	4,7 μ F	10	0,42
Kapazität	22 pF	10	0,66
Kapazität	100 nF	100	1
Induktivität	10 μ H	1	0,084
Spg. Regler	NCP5501DT50G	1	0,714
μ C	ATTINY2313A-SU	1	2,67
Optokoppler	VO223AT	1	1,875
Quarz	8 MHz		0
Widerstand	560 k Ω	10	0,21
Widerstand	100 k Ω	50	1,8
Widerstand	180 Ω	10	0,19
Widerstand	330 Ω	50	1,35
Widerstand	220 Ω	100	2,3
Widerstand	120 Ω	10	0,32
Widerstand	150 Ω	10	0,13
Widerstand	100 Ω	50	1,6
Widerstand	10 k Ω	50	1,35
Widerstand	1 k Ω	50	1,35
Widerstand	15 k Ω	10	0,32
Widerstand	820 Ω	50	1,35
MOSFET	2N7002E	1	0,144
Stiftleiste			0
PCB		1	10

Womit sich der Gesamtpreis zu 30,407€ für alle Bauteile ergibt.

Bestellnummern, Mindestbestellmengen und Stückpreise stammen von [Farnell Österreich](#).

Mit den Fertigungsunterlagen zum Herunterladen können Sie entweder die Platine selber ätzen oder von einem Auftragsfertiger herstellen lassen.

Zusätzlich können Sie auch eine Platine bei mir bestellen. Diese wird dann von mir geätzt und gebohrt.

Falls Sie nicht über die Hardware verfügen um den µC selber zu programmieren, dann besteht natürlich auch die Möglichkeit, diesen bereits programmiert bei mir zu kaufen.

Wenn Ihnen die Mindestbestellmengen zu groß sind, z. B. weil Sie nicht so viele Bauteile benötigen, dann melden Sie sich einfach bei mir. Falls ich das gewünschte Bauteil noch auf Lager habe verkaufe ich es Ihnen gerne!

A.1.2. Zusätzlich benötigte Teile

- Die Kabel und LEDs vom [Conrad Lichtmodul](#), ca. 23€
- Kabel, Stecker, Akku, etc.

oder

- 1 x [LED Grün](#), ca. 0,80€ pro Stück
- 3 x [LED Rot](#), ca. 0,40€ pro Stück
- 2 x [LED Weiß](#) für Positionslichter, ca. 1,30€ pro Stück
- [LED Weiß](#) für Landelichter, ca. 1,80€ pro Stück
- Kabel, Stecker, Akku, etc.

Achtung! Bei Verwendung der hier vorgeschlagenen oder anderer LEDs, sind die Werte der Vorwiderstände neu zu berechnen! Die Werte der Vorwiderstände in den Tabellen gelten nur bei der Verwendung des Conrad Lichtmoduls!

A.2. Bauteil Liste

Referenz	Wert	Footprint	Toleranz	Hersteller	Herstellerbezeichnung
C1	10 µF	SM1206	-20 % / 80 %	Multicomp	MCCA000554
C2	4,7 µF	SM1206	-20 % / 80 %	Multicomp	MCCA000563
C5	100 nF	SM0805	-20 % / 80 %	Multicomp	MCCA000296
L1	10 µH	SM0805	±20 %	Taiyo Yuden	LB2012T100M
Con_1	Batt.	PIN_ARRAY_2X1			
Con_2	RX-IN	PIN_ARRAY_3X1			
Con_3	Landelicht-1	PIN_ARRAY_2X1			
Con_4	Landelicht-2	PIN_ARRAY_2X1			
Con_5	POS-ROT-1	PIN_ARRAY_2X1			
Con_6	POS-ROT-2	PIN_ARRAY_2X1			
Con_7	NAV-ROT	PIN_ARRAY_2X1			
Con_8	NAV-GRÜN	PIN_ARRAY_2X1			
Con_9	POS-WEISS-1	PIN_ARRAY_2X1			
Con_10	POS-WEISS-2	PIN_ARRAY_2X1			
IC1	NCP5501DT50G	DPAK2_Regul		ON Semiconductor	NCP5501DT50G
IC2	ATTINY2313A-SU	Atmel_20S		Atmel	ATTINY2313A-SU
ISP1	AVR-ISP-10	PIN_ARRAY_3x2			
OK1	VO223AT	Vishay_VO223AT		Vishay	VO223AT
R1	560 kΩ	SM1206	±1 %	Welwyn	WCR1206-560KFI
R2	100 kΩ	SM1206	±1 %	Multicomp	MC 0.125W 1206 1 % 100K
R3	180 Ω	SM1206	±1 %	Welwyn	WCR1206-180RFI
R4	180 Ω	SM1206	±1 %	Welwyn	WCR1206-180RFI
R5	330 Ω	SM1206	±1 %	Vishay Draloric	CRCW1206330RFFKEA
R6	330 Ω	SM1206	±1 %	Vishay Draloric	CRCW1206330RFFKEA
R7	220 Ω	SM1206	±1 %	Welwyn	ASC1206-220RFT5
R8	120 Ω	SM1206	±1 %	Welwyn	WCR1206-120RFI
R9	150 Ω	SM1206	±1 %	Welwyn	WCR1206-150RFI
R10	150 Ω	SM1206	±1 %	Welwyn	WCR1206-150RFI
R11	100 Ω	SM1206	±1 %	Vishay Draloric	CRCW1206100RFFKEA
R12	100 Ω	SM1206	±1 %	Vishay Draloric	CRCW1206100RFFKEA
R13	100 Ω	SM1206	±1 %	Vishay Draloric	CRCW1206100RFFKEA
R14	100 Ω	SM1206	±1 %	Vishay Draloric	CRCW1206100RFFKEA
R15	10 kΩ	SM1206	±1 %	Vishay Draloric	CRCW120610K0FFKEA
R16	220 Ω	SM1206	±1 %	Welwyn	ASC1206-220RFT5

Referenz	Wert	Footprint	Toleranz	Hersteller	Herstellerbezeichnung
R17	1 k Ω	SM1206	$\pm 1\%$	Vishay Draloric	CRCW12061K00FKEA
R21	15 k Ω	SM1206	$\pm 1\%$	Welwyn	WCR1206-15KFI
T1	2N7002E	SOT23GDS		NXF	2N7002E
T2	2N7002E	SOT23GDS		NXF	2N7002E
T3	2N7002E	SOT23GDS		NXF	2N7002E
T4	2N7002E	SOT23GDS		NXF	2N7002E
PCB				V. PIPPAN	RCLS_S_PCB_RevB
C3 (optional)	22 pF	SM1206	$\pm 5\%$	Multicomp	MCCA000481
C4 (optional)	22 pF	SM1206	$\pm 5\%$	Multicomp	MCCA000481
Q1 (optional)	8 MHz	Q_49U3HMS			
OK2 (optional)	VO223AT	Vishay_VO223AT		Vishay	VO223AT
OK3 (optional)	VO223AT	Vishay_VO223AT		Vishay	VO223AT
R18 (optional)	820 Ω	SM1206	$\pm 1\%$	Vishay Draloric	CRCW1206820RFKEA
R19 (optional)	220 Ω	SM1206	$\pm 1\%$	Welwyn	ASC1206-220RFT5
R20 (optional)	1 k Ω	SM1206	$\pm 1\%$	Vishay Draloric	CRCW12061K00FKEA
Con_11 (optional)	Jeti-Data	PIN_ARRAY_3X1			

A.3. Weitere Unterlagen

- Schaltplan
- PCB Größe und Bohrplan
- PCB Belichtungsvorlage
- PCB Bestückungsplan

2S LiPo oder
max. 8,4V
min. 5,7V

Querstrom über Spannungsteiler
für Messung der Akkuspannung
beträgt ca. 12,44µA
Teilung der Akkuspannung gibt
einen Wert zwischen 1,431V
(Akku voll) und 1,09V
(Akku leer) am Komparator

Pro Transistor max. 200mA
Insgesamt (wenn alle
Transistoren eingeschalten)
max. 400mA

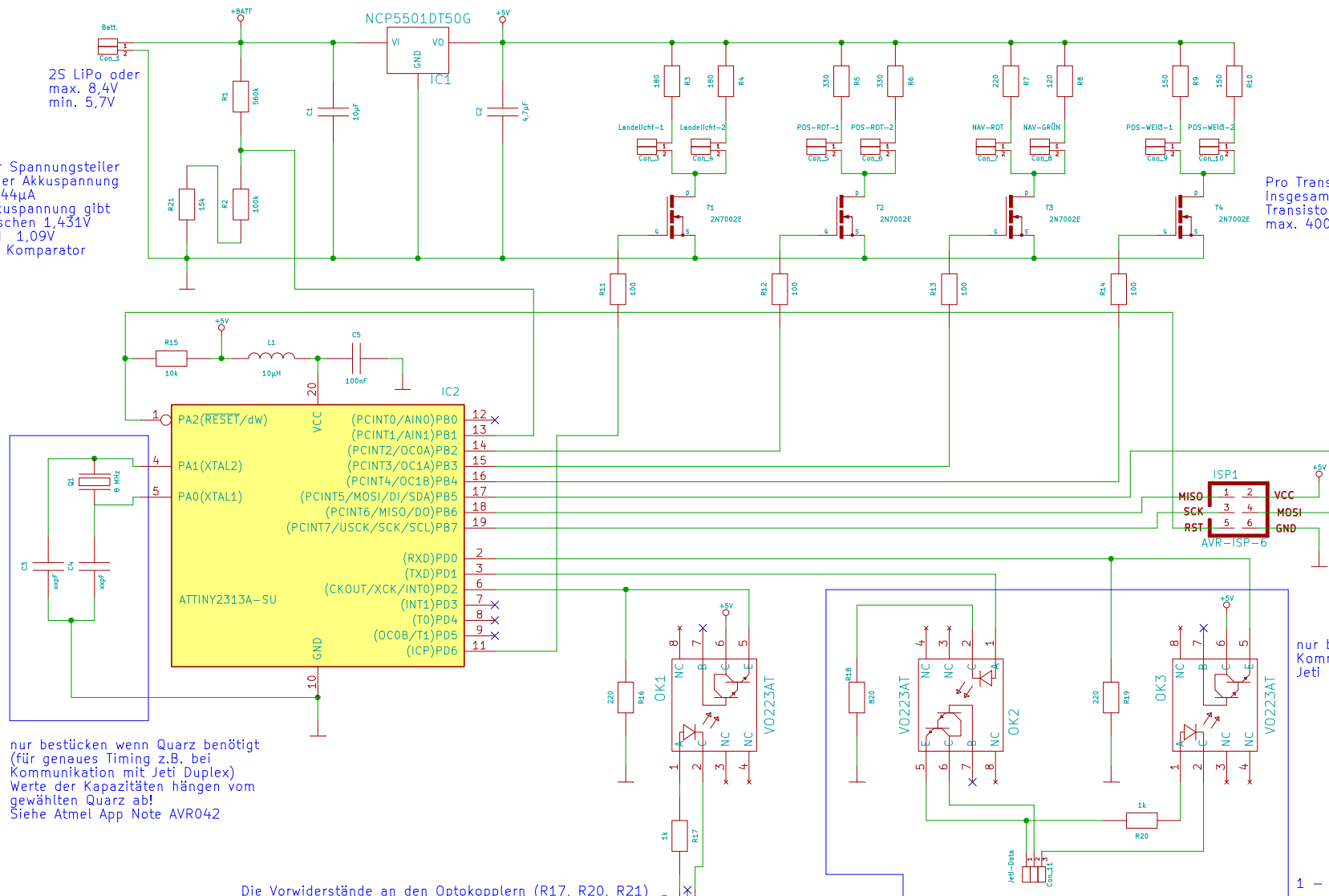
nur bestücken wenn Quarz benötigt
(für genaues Timing z.B. bei
Kommunikation mit Jeti Duplex)
Werte der Kapazitäten hängen vom
gewählten Quarz ab!
Siehe Atmel App Note AVR042

Die Vorwiderstände an den Optokopplern (R17, R20, R21)
sind für eine max. Spannung von 6V ausgelegt. Bei
Verwendung von höheren Versorgungsspannungen
des Empfängers im Modell sind diese Widerstände zu
ändern!

1 - Data (max. 6V / min. 3V)
2 - VCC (not used)
3 - GND

Wenn kein LiPo als Versorgung verwendet wird,
und die Versorgungssp. <= 6,8 V ist, müssen
entweder die Widerstände des Spg. Teilers oder
die Firmware geändert werden.
Sonst wird der Akku als leer erkannt und die
Software schaltet alles ab!

- LEDs über Spannungsregler versorgt -> Helligkeit bleibt konst. und ändert sich nicht mit der Batteriespannung
- Widerstände für LEDs etwas kleiner gewählt da die original Schaltung mit bis zu 6V betrieben werden kann
- Weitere Infos siehe im Tabellendokument



nur bestücken für
Kommunikation mit
Jeti Duplex System

1 - Jeti Data
(max. 6V / min. 3V)
2 - VCC
(max. 6V / min. 3V)
3 - GND

Lizenz: CC-BY-SA

V. Pippa (www.vpippa.at)

File: attiny2313_rc_lichtsteuerung_standard.sch

Sheet: /

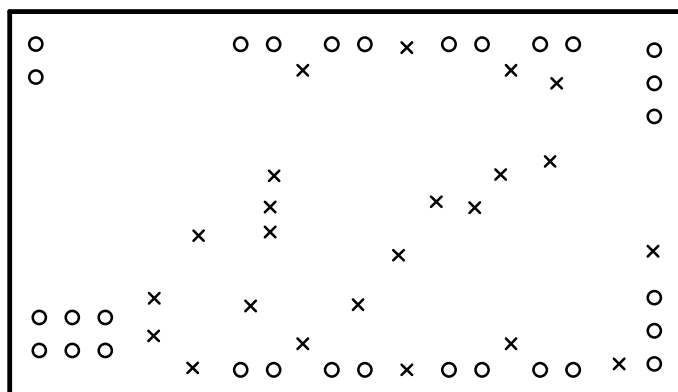
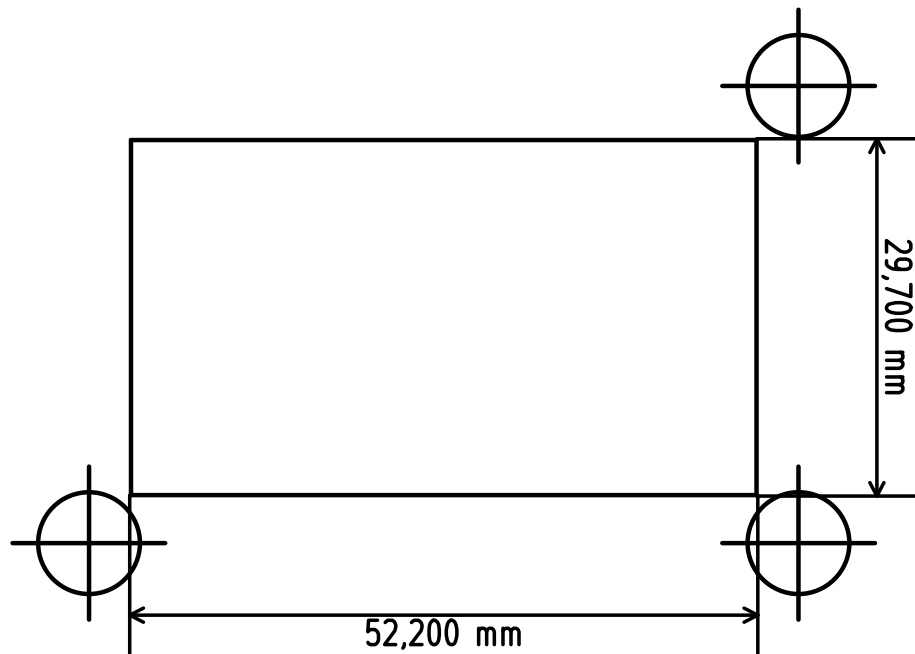
Title: ATtiny2313 RC Lichtsteuerung Standard

Size: A4 Date: 24 mar 2014

KiCad E.D.A. eschema (2013-07-07 BZR 4022)-stable

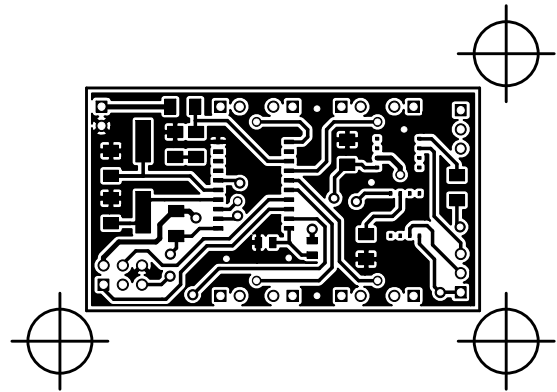
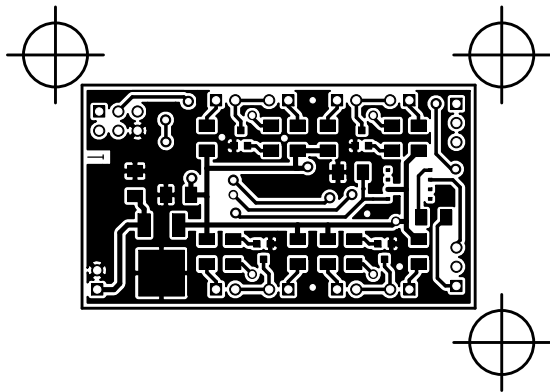
Rev: B

Id: 1/1



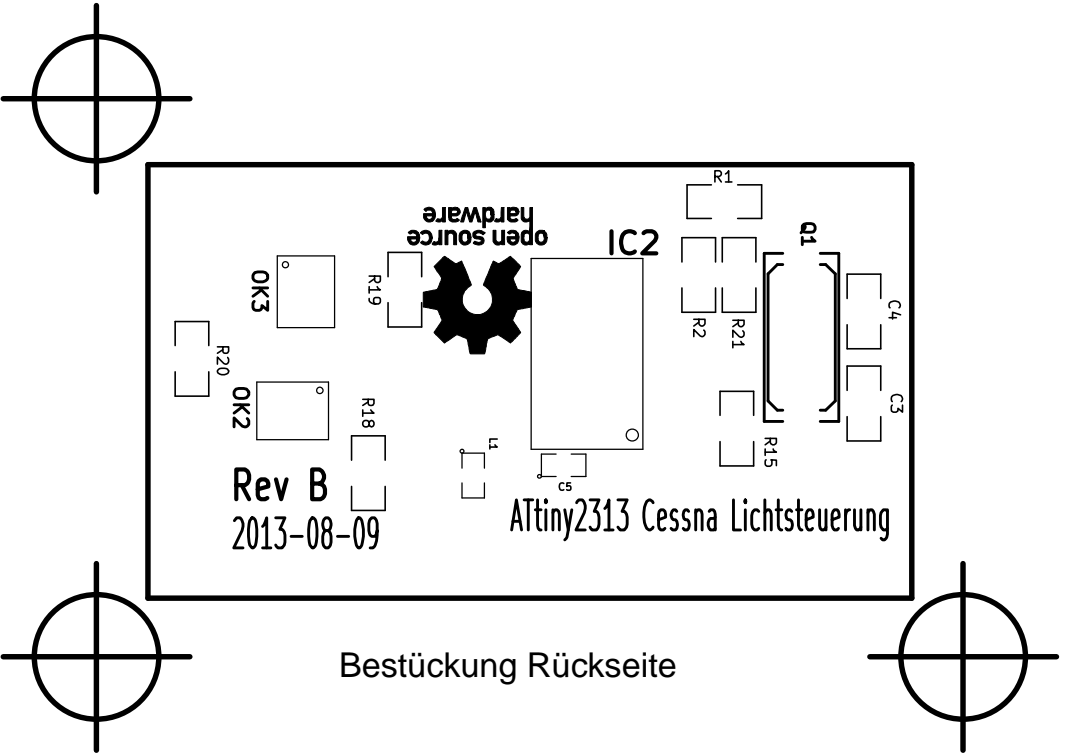
Drill Map:

x 0.80mm / 0.032" (23 holes)
o 1.02mm / 0.040" (30 holes)

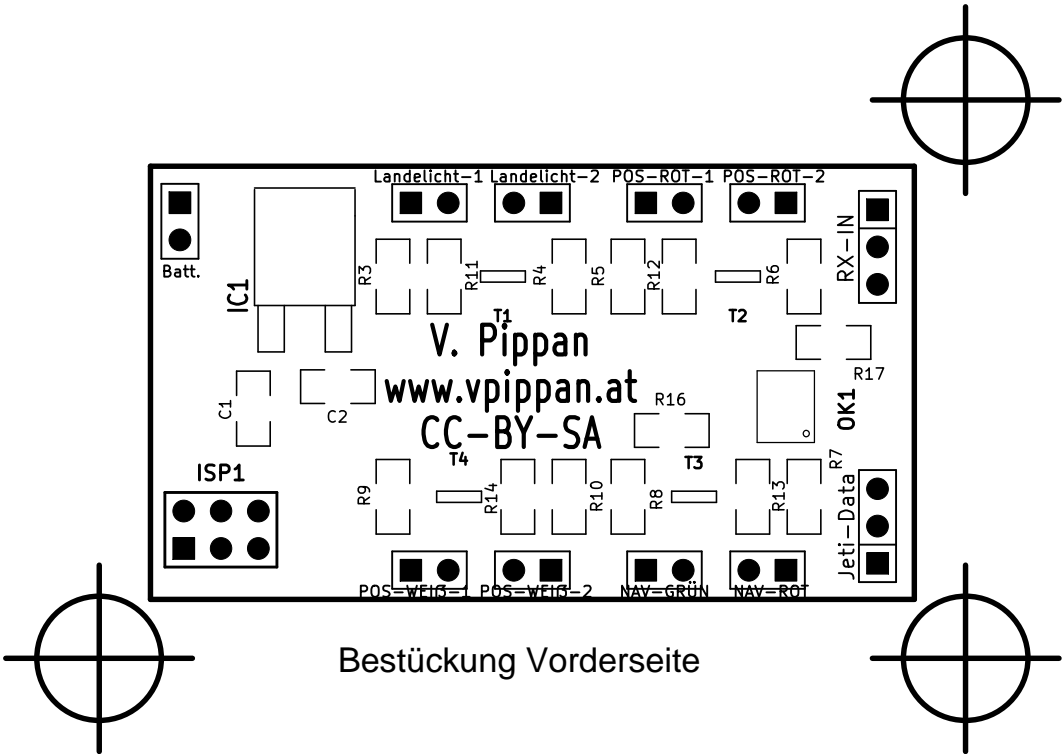


Top -> gespiegelt & von oben betrachtet

Bottom -> nicht gespiegelt & von oben betrachtet



Bestückung Rückseite



Bestückung Vorderseite

B. Softwaredokumentation

ATtiny2313RCLichtsteuerungStandard
2013-09-02

Erzeugt von Doxygen 1.8.7

Sam Jun 14 2014 17:28:47

Inhaltsverzeichnis

1	Ausstehende Aufgaben	1
2	Datei-Verzeichnis	3
2.1	Auflistung der Dateien	3
3	Datei-Dokumentation	5
3.1	attiny2313_rc_lichtsteuerung_standard.c-Dateireferenz	5
3.1.1	Ausführliche Beschreibung	6
3.1.2	Makro-Dokumentation	7
3.1.2.1	F_CPU	7
3.1.3	Dokumentation der Funktionen	7
3.1.3.1	ISR	7
3.1.3.2	main	7
3.1.4	Variablen-Dokumentation	8
3.1.4.1	batt_full	8
3.1.4.2	blktm_oflow	8
3.1.4.3	in0_puls_len	8
3.1.4.4	no_sig	8
3.2	batt_komp.c-Dateireferenz	8
3.2.1	Ausführliche Beschreibung	9
3.2.2	Dokumentation der Funktionen	9
3.2.2.1	init_batt_komp	9
3.2.2.2	ISR	9
3.3	batt_komp.h-Dateireferenz	10
3.3.1	Ausführliche Beschreibung	10
3.3.2	Dokumentation der Funktionen	11
3.3.2.1	init_batt_komp	11
3.3.3	Variablen-Dokumentation	11
3.3.3.1	batt_full	11
3.4	blink_timer.c-Dateireferenz	11
3.4.1	Ausführliche Beschreibung	12
3.4.2	Dokumentation der Funktionen	13

3.4.2.1	init_blink_timer	13
3.4.2.2	ISR	13
3.5	blink_timer.h-Dateireferenz	13
3.5.1	Ausführliche Beschreibung	14
3.5.2	Dokumentation der Funktionen	14
3.5.2.1	init_blink_timer	14
3.5.3	Variablen-Dokumentation	14
3.5.3.1	blktm_oflow	14
3.6	input_timer.c-Dateireferenz	14
3.6.1	Ausführliche Beschreibung	15
3.6.2	Dokumentation der Funktionen	16
3.6.2.1	init_input_timer	16
3.6.2.2	ISR	16
3.7	input_timer.h-Dateireferenz	17
3.7.1	Ausführliche Beschreibung	18
3.7.2	Dokumentation der Funktionen	18
3.7.2.1	init_input_timer	18
3.7.3	Variablen-Dokumentation	18
3.7.3.1	in0_puls_len	18

Kapitel 1

Ausstehende Aufgaben

Datei [attiny2313_rc_lichtsteuerung_standard.c](#)

Umschalten der Zustände aus der Hauptschleife entfernen und in eine eigene Funktion packen -> diese Funktion wird nur ca. alle 17 ms benötigt (nach Erkennung eines Eingangsimpulses oder nach Wachtdog Überlauf) und muß nicht in der Hauptschleife andauernd überprüft werden -> uC länger im Schlafmodus? -> weniger Stromverbrauch

- Volatile -> ISR Optimierung bei Zugriff auf `blktm_oflow` in [blink_timer.c](#) -> siehe mikrocontroller.net
- Umschalten von Ein- / Aus und umgekehrt erst nach mehreren gleichen Eingangsimpuls längen
- alle nicht benötigten uC Teile abschalten
- Abspeichern der Einstellungen (Schaltpunkte, Blinkfolgen, etc.) im EEPROM -> weniger Speicherverbrauch im Flash, nicht hardcoded -> dann möglich umstellen über Jeti Protokoll
- Ausgabe des Schaltzustandes über das Jeti Protokoll
- Ändern der Schaltzeitpunkte, Blinkfrequenz, etc. über das Jeti Protokoll
- Messen von PWM Tastverhältnisse siehe Atmel App Note

Kapitel 2

Datei-Verzeichnis

2.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

attiny2313_rc_lichtsteuerung_standard.c	
Hauptfile des Projekts ATtiny2313 RC Lichtsteuerung Standard	5
batt_komp.c	
Komparator zur Batterie Spannungsüberwachung	8
batt_komp.h	
Include File für batt_komp.c	10
blink_timer.c	
Timing für das blinken der Lichter	11
blink_timer.h	
Include File für blink_timer.c	13
input_timer.c	
Erfassung der Impulslänge des Empfängersignals mit dem Timer0	14
input_timer.h	
Include File für input_timer.c	17

Kapitel 3

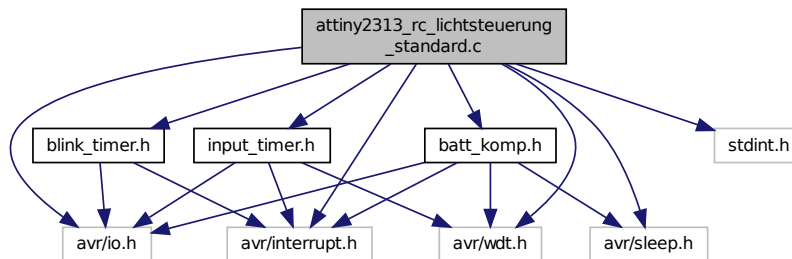
Datei-Dokumentation

3.1 attiny2313_rc_lichtsteuerung_standard.c-Dateireferenz

Hauptfile des Projekts ATtiny2313 RC Lichtsteuerung Standard.

```
#include <avr/io.h>
#include <stdint.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
#include <avr/wdt.h>
#include "input_timer.h"
#include "blink_timer.h"
#include "batt_komp.h"
```

Include-Abhängigkeitsdiagramm für attiny2313_rc_lichtsteuerung_standard.c:



Makrodefinitionen

- `#define F_CPU 1000000UL`

Funktionen

- `int main ()`
Hauptfunktion.
- `ISR (WDT_OVERFLOW_vect)`
Interrupt Service Routine für Watchdog Timer.

Variablen

- volatile uint8_t `in0_puls_len` = 0
*Zählerstand vom Input Timer (Timer0) = Pulslänge des Eingangs 1 -> Zählerstand * 64us = Pulslänge in ms.*
- volatile uint8_t `blktm_oflow` = 0
Anzahl der Überläufe vom Blink Timer (Timer 1) -> Ein Überlauf entspricht 0,065s.
- volatile uint8_t `batt_full` = 1
Batteriestatus -> solange = 1 wird die Hauptschleife ausgeführt, wenn = 0 wird alles abgeschaltet und der uC in den Schlafmodus versetzt.
- volatile uint8_t `no_sig` = 0
Status Empfänger Eingangssignal -> 0 = Signal vorhanden, 1 = kein Signal.

3.1.1 Ausführliche Beschreibung

Hauptfile des Projekts ATtiny2313 RC Lichtsteuerung Standard.

Autor

V. Pippan (webmaster@vpippan.at)

Datum

2013-08-30

Version

20130830

Copyright 2013, 2014 V. Pippan (webmaster@vpippan.at)

This file is part of ATtiny2313 RC Lichtsteuerung Standard.

ATtiny2313 RC Lichtsteuerung Standard is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

ATtiny2313 RC Lichtsteuerung Standard is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with ATtiny2313 RC Lichtsteuerung Standard. If not, see www.gnu.org/licenses.

Noch zu erledigen

- Umschalten der Zustände aus der Hauptschleife entfernen und in eine eigene Funktion packen -> diese Funktion wird nur ca. alle 17 ms benötigt (nach Erkennung eines Eingangsimpulses oder nach Watchdog Überlauf) und muß nicht in der Hauptschleife andauernd überprüft werden -> uC länger im Schlafmodus? -> weniger Stromverbrauch
- Volatile -> ISR Optimierung bei Zugriff auf `blktm_oflow` in [blink_timer.c](#) -> siehe mikrocontroller.net
- Umschalten von Ein- / Aus und umgekehrt erst nach mehreren gleichen Eingangsimpuls längen
- alle nicht benötigten uC Teile abschalten

- Abspeichern der Einstellungen (Schaltpunkte, Blinkfolgen, etc.) im EEPROM -> weniger Speicherverbrauch im Flash, nicht hardcoded -> dann möglich umstellen über Jeti Protokoll
- Ausgabe des Schaltzustandes über das Jeti Protokoll
- Ändern der Schaltzeitpunkte, Blinkfrequenz, etc. über das Jeti Protokoll
- Messen von PWM Tastverhältnisse siehe Atmel App Note

3.1.2 Makro-Dokumentation

3.1.2.1 #define F_CPU 1000000UL

3.1.3 Dokumentation der Funktionen

3.1.3.1 ISR (WDT_OVERFLOW_vect)

Interrupt Service Routine für Watchdog Timer.

Ein Überlauf des Watchdog Timer löst diesen Interrupt aus. Wenn für längere Zeit kein Empfänger Eingangssignal anliegt wird der Watchdog Timer nicht zurückgesetzt. Dies führt zu einem Überlauf des Watchdog Timers und innerhalb dieser Routine wird die Variable no_sig auf 1 gesetzt. Dies führt zum Abschalten der Beleuchtung, falls diese vorher eingeschalten war.

3.1.3.2 int main ()

Hauptfunktion.

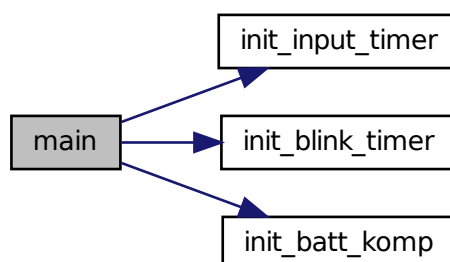
Die Hauptfunktion enthält die Initialisierungen beim Programmstart und die Hauptschleife des Programms. Wird die Hauptschleife abgebrochen (= Batterie leer) werden die Lichter ausgeschaltet, alle Interrupts deaktiviert, die Timer gestoppt und der uC in den Power Down Schlafmodus versetzt. Aus diesem Zustand kann der uC nur durch einen Reset (Anstecken eines geladenen Akkus) wieder aktiviert werden! < Zählerstand bei dem Navigationslichter einschalten sollen (bei meinem Servotester -> 12, normal 21)

< Zählerstand bei dem Landelichter einschalten sollen (bei meinem Servotester -> 19, normal 26)

Hauptschleife

Die Hauptschleife wird ausgeführt solange die Variable batt_full = 1 ist. batt_full = 0 bedeutet einen leeren Akku und die Hauptschleife wird beendet.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



3.1.4 Variablen-Dokumentation

3.1.4.1 `volatile uint8_t batt_full = 1`

Batteriestatus -> solange = 1 wird die Hauptschleife ausgeführt, wenn = 0 wird alles abgeschaltet und der uC in den Schlafmodus versetzt.

3.1.4.2 `volatile uint8_t blktm_oflow = 0`

Anzahl der Überläufe vom Blink Timer (Timer 1) -> Ein Überlauf entspricht 0,065s.

3.1.4.3 `volatile uint8_t in0_puls_len = 0`

Zählerstand vom Input Timer (Timer0) = Pulslänge des Eingangs 1 -> Zählerstand * 64us = Pulslänge in ms.

3.1.4.4 `volatile uint8_t no_sig = 0`

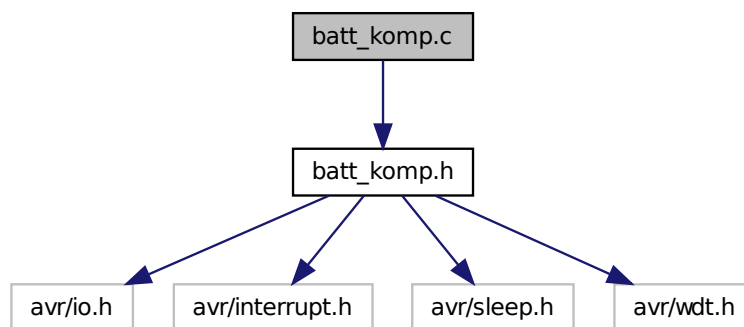
Status Empfänger Eingangssignal -> 0 = Signal vorhanden, 1 = kein Signal.

3.2 batt_komp.c-Dateireferenz

Komparator zur Batterie Spannungsüberwachung.

```
#include "batt_komp.h"
```

Include-Abhängigkeitsdiagramm für batt_komp.c:



Funktionen

- void `init_batt_komp()`
Komparator zur Akkuüberwachung einstellen.
- `ISR (ANA_COMP_vect)`
Interrupt Service Routine für den Analog Komparator.

3.2.1 Ausführliche Beschreibung

Komparator zur Batterie Spannungsüberwachung.

Autor

V. Pippa (webmaster@vpippa.at)

Datum

2013-08-30

Version

20130830

Copyright 2013, 2014 V. Pippa (webmaster@vpippa.at)

This file is part of ATtiny2313 RC Lichtsteuerung Standard.

ATtiny2313 RC Lichtsteuerung Standard is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

ATtiny2313 RC Lichtsteuerung Standard is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with ATtiny2313 RC Lichtsteuerung Standard. If not, see www.gnu.org/licenses.

Als Vergleichsspannung für den Komparator wird die interne Referenzspannung des uC verwendet. Diese beträgt typ. 1,1V und schwankt über die Temperatur.

Die hohe Akkuspannung wird über einen Spannungsteiler so geteilt, daß ein Vergleich mit der Referenzspannung möglich wird. Allerdings ändert sich diese heruntergeteilte Spannung, aufgrund des Teilungsfaktors, nur mehr in einem kleinen Bereich.

Durch eine Berechnung mit Hilfe der Fehlerfortpflanzung (siehe Maxima Datei) habe ich sichergestellt, daß trotz störender Einflüsse (Widerstandstoleranzen, Temperaturänderung) der Akku sicher als leer erkannt wird. Ein Berechnungsbeispiel findet sich im Tabellendokument welches auch die Bauteilliste etc. enthält.

3.2.2 Dokumentation der Funktionen

3.2.2.1 void init_batt_komp ()

Komparator zur Akkuüberwachung einstellen.

- Digital Input Buffer für Pins AIN0 und AIN1 deaktivieren -> spart Strom
- Interrupt an positiver Komparator Flanke auslösen, interne Referenzspannung an den Komparator Eingang AIN0 schalten, Komparator Interrupt aktivieren

3.2.2.2 ISR (ANA_COMP_vect)

Interrupt Service Routine für den Analog Komparator.

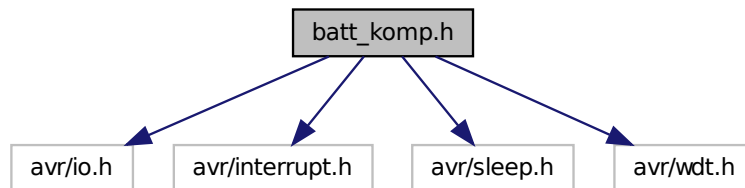
Wenn die Akkuspannung unter einen vorgegebenen Wert fällt, erzeugt der Analoge Komparator eine positive Flanke an seinem Ausgang und dieser Interrupt wird ausgeführt. Innerhalb dieser Routine wird batt_full = 0 gesetzt und somit die Hauptschleife des Programms beendet, alles ausgeschaltet und der uC in den Schlafmodus versetzt.

3.3 batt_komp.h-Dateireferenz

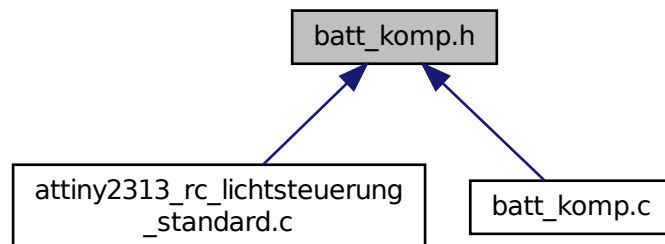
Include File für [batt_komp.c](#).

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
#include <avr/wdt.h>
```

Include-Abhängigkeitsdiagramm für batt_komp.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Funktionen

- void [init_batt_komp](#) ()
Komparator zur Akkuüberwachung einstellen.

Variablen

- volatile uint8_t [batt_full](#)
Batteriestatus -> solange = 1 wird die Hauptschleife ausgeführt, wenn = 0 wird alles abgeschaltet und der uC in den Schlafmodus versetzt.

3.3.1 Ausführliche Beschreibung

Include File für [batt_komp.c](#).

Autor

V. Pippa (webmaster@vpippa.at)

Datum

2013-08-30

Version

20130830

3.3.2 Dokumentation der Funktionen

3.3.2.1 void init_batt_komp ()

Komparator zur Akkuüberwachung einstellen.

- Digital Input Buffer für Pins AIN0 und AIN1 deaktivieren -> spart Strom
- Interrupt an positiver Komparator Flanke auslösen, interne Referenzspannung an den Komparator Eingang AIN0 schalten, Komparator Interrupt aktivieren

3.3.3 Variablen-Dokumentation

3.3.3.1 volatile uint8_t batt_full

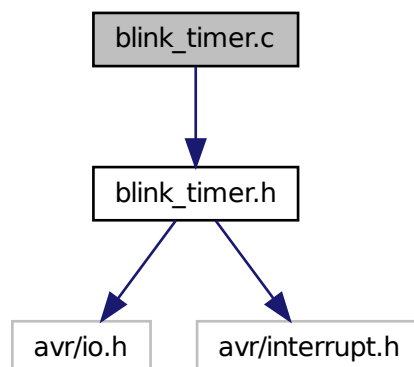
Batteriestatus -> solange = 1 wird die Hauptschleife ausgeführt, wenn = 0 wird alles abgeschaltet und der uC in den Schlafmodus versetzt.

3.4 blink_timer.c-Dateireferenz

Timing für das blinken der Lichter.

```
#include "blink_timer.h"
```

Include-Abhängigkeitsdiagramm für blink_timer.c:



Funktionen

- void `init_blink_timer()`
- `ISR (TIMER1_OVF_vect)`

Interrupt Service Routine für Timer 1 (Blink Timer) Überlauf.

3.4.1 Ausführliche Beschreibung

Timing für das blinken der Lichter.

Autor

V. Pippan (webmaster@vpippan.at)

Datum

2013-09-02

Version

20130902

Copyright 2013, 2014 V. Pippan (webmaster@vpippan.at)

This file is part of ATtiny2313 RC Lichtsteuerung Standard.

ATtiny2313 RC Lichtsteuerung Standard is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

ATtiny2313 RC Lichtsteuerung Standard is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with ATtiny2313 RC Lichtsteuerung Standard. If not, see www.gnu.org/licenses.

Der 16 Bit Timer wird zum erzeugen der Blinksignale benutzt. Bei einer CPU Frequenz von 1 MHz und ohne Vorteiler benötigt der Timer 0,065 s ($(1/f_{cpu}) * 2^{16}$) bis zum Überlauf. Beim Timer Überlauf wird ein Interrupt ausgelöst und die Variable `blktm_oflow` um 1 erhöht. Diese Variable bestimmt somit die verstrichene Zeit (Anzahl Timerüberläufe * 0.065 s). Daraus wird dann der aktuelle Status innerhalb der Blinkfolge abgeleitet und die Lichter entsprechend ein oder ausgeschalten.

Durch das zählen der Timerüberläufe ist die kürzest mögliche Zeitperiode = 0,065 s. Dies bedeutet, eine Pause oder die Zeit vom Ein- bis zum Ausschalten (blitzen) eines Ausganges kann nicht kürzer als 0,065 s sein. Für eine originalgetreue Darstellung eines Positionsblitzes sollte dies aber ausreichend kurz sein.

Erklärung zum erstellen eigener Blinkfolgen:

Am besten zeichnet man sich die gewünschte Blinkfolge auf ein Blatt Papier, beginnend mit der Pause nach Ende der vorherigen Blinkfolge. Die Folge unterteilt man dann in 0,065 s lange Abschnitte. Je mehr Abschnitte, desto länger dauert die Folge (desto länger sind die Lichter eingeschalten). Die Abschnitte werden durchnummeriert und anhand der vorher gezeichneten Folge sieht man, ob beim entsprechenden Abschnitt (= Zustand) ein Licht ein oder auszuschalten ist.

Nun braucht man nur mehr bei den case Anweisungen in der ISR die Nummer des Abschnitts und den Befehl zum schalten des Lichtes eintragen. Beim Ausschalten des letzten Lichtes (letzter Zustand = nullter Zustand) noch darauf achten, die Variable `blktm_oflow` = 0 zu setzen. Beim nächsten Timerüberlauf sind dann wieder 0,065 Sekunden vergangen, wir befinden uns im Zustand 1 und die Blinkfolge beginnt von vorne. Zur Verdeutlichung ein Bild meiner Blinkfolge:

3.4.2 Dokumentation der Funktionen

3.4.2.1 void init_blink_timer ()

Überlauf Interrupt für Timer 1 (Blink Timer) einschalten

3.4.2.2 ISR (TIMER1_OVF_vect)

Interrupt Service Routine für Timer 1 (Blink Timer) Überlauf.

Dieser Interrupt wird bei jedem Überlauf von Timer 1 (Blink Timer) aufgerufen. Innerhalb der Routine wird die Variable blktm_oflow um eins erhöht. Durch die switch Anweisung wird der aktuelle Wert von blktm_oflow überprüft und entsprechend die Lichter ein oder ausgeschaltet. Dadurch entsteht die Blinksequenz.

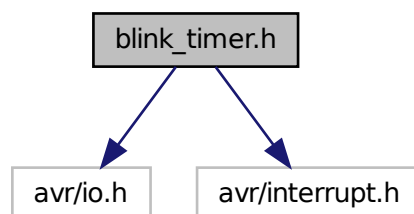
3.5 blink_timer.h-Dateireferenz

Include File für [blink_timer.c](#).

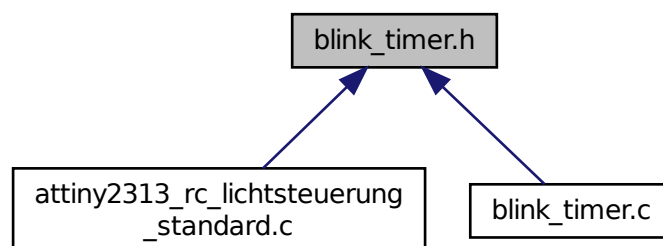
```
#include <avr/io.h>
```

```
#include <avr/interrupt.h>
```

Include-Abhängigkeitsdiagramm für blink_timer.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Funktionen

- void `init_blink_timer` ()

Variablen

- volatile uint8_t `blktm_oflow`

Anzahl der Überläufe vom Blink Timer (Timer 1) -> Ein Überlauf entspricht 0,065s.

3.5.1 Ausführliche Beschreibung

Include File für `blink_timer.c`.

Autor

V. Pippa (webmaster@vpippa.at)

Datum

2013-09-02

Version

20130902

3.5.2 Dokumentation der Funktionen

3.5.2.1 void `init_blink_timer` ()

Überlauf Interrupt für Timer 1 (Blink Timer) einschalten

3.5.3 Variablen-Dokumentation

3.5.3.1 volatile uint8_t `blktm_oflow`

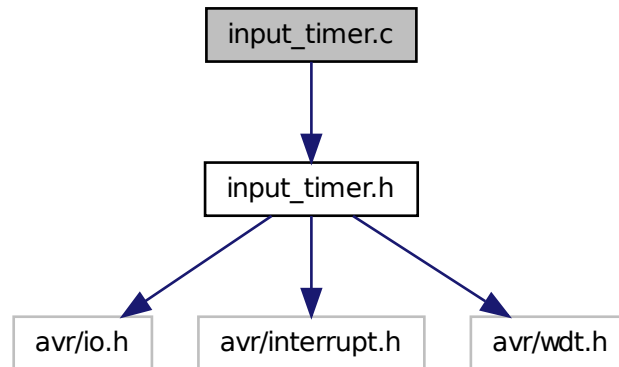
Anzahl der Überläufe vom Blink Timer (Timer 1) -> Ein Überlauf entspricht 0,065s.

3.6 `input_timer.c`-Dateireferenz

Erfassung der Impulslänge des Empfängersignals mit dem Timer0.

```
#include "input_timer.h"
```

Include-Abhängigkeitsdiagramm für input_timer.c:



Funktionen

- void `init_input_timer()`
Timer0 zur Messung des Empfängersignals einstellen.
- `ISR` (`INT0_vect`)
Interrupt Service Routine für externen Interrupt INT0.

3.6.1 Ausführliche Beschreibung

Erfassung der Impulslänge des Empfängersignals mit dem Timer0.

Autor

V. Pippan (webmaster@vpippan.at)

Datum

2013-09-02

Version

20130902

Copyright 2013, 2014 V. Pippan (webmaster@vpippan.at)

This file is part of ATtiny2313 RC Lichtsteuerung Standard.

ATtiny2313 RC Lichtsteuerung Standard is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

ATtiny2313 RC Lichtsteuerung Standard is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with ATtiny2313 RC Lichtsteuerung Standard. If not, see www.gnu.org/licenses.

Das RC Eingangssignal hat eine Periodendauer von ca. 20ms $\rightarrow f = 50\text{Hz}$. Die Länge des positiven Pulses schwankt zwischen minimal 0,75ms und maximal 2,25ms, die Standard Länge liegt zwischen minimal 1ms und maximal 2ms, wobei die Neutralstellung bei 1,5ms liegt.

Der uC läuft mit 1MHz = Periodendauer von 1us. Mit $f_{\text{CPU}} / 64$ erhält man eine Periodendauer von 64us für den Timer Takt. D.h. eine Erhöhung des Timer Zählerstandes um 1 bedeutet eine Verlängerung des RC-Empfänger Pulses um 64us. Der 8bit Timer zählt bis 2^8 (256) \rightarrow Eine Periodendauer von 64us * 256 ergibt 16,38ms bis der Timerüberlauf erreicht wird.

Mit dieser Timergeschwindigkeit kann man ohne Timerüberlauf bis zur max. Pulsdauer des RC-Empfängersignals (2,25ms) zählen. Bei minimaler Pulsdauer ergibt sich damit ein Zählerstand von 12, bei max. Pulsdauer ein Zählerstand von 35 (bei Standard Länge der Pulse min. 16 und max. 31), die Neutralstellung liegt bei einem Zählerstand von 23. Das Eingangssignal kann dabei auf 64us genau, dies entspricht 23 (15 bei Standard Pulsängen) Schritten, aufgelöst werden. Was für diesen Zweck hier genau genug sein sollte. Anmerkung: Durch Verwendung des 16bit Timers kann das RC Eingangssignal wesentlich feiner aufgelöst werden, was für eine einfache Schaltanwendung wie hier aber nicht notwendig ist.

Umrechnung von gegebenem Schaltzeitpunkt auf den Zählerstand:

RC-Empfängersignal Schaltzeitpunkt (ms) / $(64\text{us} * 10^{-3})$ = Zählerstand bei dem geschaltet wird

Erkennung eines nicht vorhandenen RC-Empfängersignals mit dem Watchdog Timer:

Der Watchdog Timer wird nach jedem korrekt erkanntem Eingangsimpuls (negative Flanke) zurückgesetzt. Wenn nach einer negativen Flanke kein weiterer Eingangsimpuls (keine positive Flanke) mehr kommt, dann verbleibt das Programm einfach so lange im aktuellen Zustand, bis der Watchdog anspricht und das Programm in den Standard Zustand (gesamte Beleuchtung aus) versetzt.

Der Fall, daß das Eingangssignal nach der positiven Flanke unterbrochen wird (z.B.: durch abziehen des Steckers, ausschalten der Empfängerstromversorgung) spielt hier keine Rolle. Sobald am Optokoppler nichts mehr anliegt, zieht der Widerstand R16 den Pin auf Masse, was eine fallende Flanke erzeugt. Somit entsteht auch in diesem Fall ein korrekt erkannter (wenn auch zu kurzer) Eingangsimpuls.

3.6.2 Dokumentation der Funktionen

3.6.2.1 void init_input_timer ()

Timer0 zur Messung des Empfängersignals einstellen.

- Einstellen von Pin 6 (PD2) als Quelle für externen Interrupt
- externen Interrupt aktivieren und konfigurieren

3.6.2.2 ISR (INT0_vect)

Interrupt Service Routine für externen Interrupt INT0.

Diese ISR dient der Erfassung der Pulslänge vom RC-Empfängersignal. Sie wird bei jeder Flanke des Eingangssignals aufgerufen. Das umschalten auf welche Flanke reagiert werden soll erfolgt dabei innerhalb dieser Routine.

Nach einer positiven Flanke (Eingang ist HIGH):

- Timer 0 (Input Timer) starten
- Umschalten auf Interrupt Auslösung nach negativer Flanke

Nach einer negativen Flanke (Eingang ist LOW):

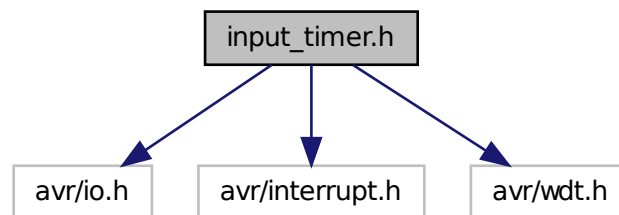
- Timer 0 (Input Timer) stoppen
- Auslesen des Timer Wertes
- Timer Zählerstand auf 0 zurücksetzen
- Umschalten auf Interrupt Auslösung nach positiver Flanke
- Watchdog Timer zurücksetzen

3.7 input_timer.h-Dateireferenz

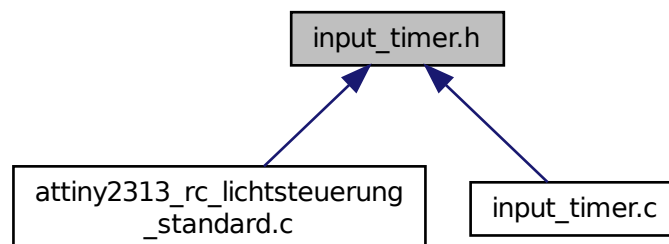
Include File für [input_timer.c](#).

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>
```

Include-Abhängigkeitsdiagramm für input_timer.h:



Dieser Graph zeigt, welche Datei direkt oder indirekt diese Datei enthält:



Funktionen

- void `init_input_timer()`
Timer0 zur Messung des Empfängersignals einstellen.

Variablen

- volatile uint8_t `in0_puls_len`
*Zählerstand vom Input Timer (Timer0) = Pulslänge des Eingangs 1 -> Zählerstand * 64us = Pulslänge in ms.*

3.7.1 Ausführliche Beschreibung

Include File für `input_timer.c`.

Autor

V. Pippa (webmaster@vpippa.at)

Datum

2013-06-26

Version

20130626

3.7.2 Dokumentation der Funktionen

3.7.2.1 void `init_input_timer()`

Timer0 zur Messung des Empfängersignals einstellen.

- Einstellen von Pin 6 (PD2) als Quelle für externen Interrupt
- externen Interrupt aktivieren und konfigurieren

3.7.3 Variablen-Dokumentation

3.7.3.1 volatile uint8_t `in0_puls_len`

Zählerstand vom Input Timer (Timer0) = Pulslänge des Eingangs 1 -> Zählerstand * 64us = Pulslänge in ms.