
Moderne Technik – Mikrocontroller

Software und low power

Onkobu Tanaake

2017-03-11T16:42:00

Manchmal gibt es kleine Probleme, die nach einer großen Lösung schreien. Es gibt eine Vielzahl von Nachtlichtern: mit Bewegungsmelder, mit Bärenaufdruck, mit Zeitschalter und natürlich per Smartphone steuerbar. Als Netzwerk- und Medientechniker erwächst daraus der Trieb, es mir selbst zu machen. Die Eckdaten sind schnell zusammengetragen:

- Kein Netzanschluss, Sicherheit geht vor
- LEDs als effiziente Lichtquelle
- Bewegungsmelder mit geringer Stromaufnahme
- wiederaufladbar/ mit Batteriebetrieb
- nur bei Dunkelheit aktiv

Da auch Akkus zum Einsatz kommen sollen und die Plattform flexibel bleiben soll, muss eine variable Spannungsüberwachung eingebaut werden. Eine Zeitschaltautomatik soll die Beleuchtung nach einstellbarer Zeit wieder abschalten. Da es nicht ständig am Netz hängt, muss der Stromverbrauch sehr niedrig sein, um eine lange Laufzeit zu gewährleisten.

Die gesamte Steuerung wird in einem Mikrocontroller ungebracht. Er kann programmiert werden, verfügt über flexible Ein- und Ausgänge, sowie zusätzliche Peripherie. Dazu gehört bspw. eine PWM-Steuerung (Puls-Weiten-Modulation), um bspw. die Helligkeit regeln zu können. Weiters verfügen Mikrocontroller über Analog- Digital-Wandler, um die Spannungsmessung realisieren zu können. Und Mikrocontroller können in verschiedene Standby-Modi versetzt werden, um sie bspw. per Steuersignal wieder aufwecken zu können. Während sie schlafen, verbrauchen sie nur wenige Mikroampere.

Mein Arbeitspferd für solche Zwecke ist der ATTiny44 aus dem Hause Atmel (mittlerweile eine Microchip-Firma und damit keine echte Konkurrenz mehr zu PIC-Controllern). Er hat ausreichend Eingänge, Timer und Peripherie. Seine Aufgaben sind:

- Schlafen, bis der Bewegungsmelder auslöst
- Unterschreiten einer Spannungsschwelle anzeigen
- LED-Konstantstromquelle (Boost-Konverter) steuern
- Helligkeitssensor einbeziehen, nur bei Dunkelheit aktiv werden

Als Alternative könnte statt des Mikrocontrollers eine kleine Gruppe von Logik-Gattern verwendet werden. Ein CD4060 wäre der Zeitgeber. Ein Flip-Flop steuert die Lichtquelle und ein Helligkeitssensor blockiert ggf. das Flip-Flop. Der Helligkeitssensor würde an das Flip-Flop angeschlossen. Solange dieses nicht vom Helligkeitssensor in den RESET-Zustand gezwungen würde, setzt der Bewegungsmelder das Flip-Flop. Der CD4060 liefert los und würde bis zu einem Zielwert zählen. Solange der nicht erreicht ist, würden die LEDs leuchten. Ist er erreicht, wird das Flip-Flop getriggert und setzt sich quasi zurück – Anfangszustand.

Diese Schaltung braucht wenigstens 3 CMOS-Gatter: Zähler, Flip-Flop, Logik-Kombination »nicht Helligkeitssensor und Bewegungsmelder«. Als SOIC- Gehäuse mit jeweils 14-16 Beinen braucht das die dreifache Fläche als ein ATTiny44. Der Stromverbrauch ist ebenfalls höher. Ein CD4060 mit RC-Oszillator zieht einige Milliampere, aufgrund des Linearbetriebes. Eine Quarz-Version ist denkbar, braucht aber mehr Fläche und ist schlechter regelbar. Der RC-Oszillator kann mit einem Potentiometer in weiten Grenzen geregelt werden. Da die Schaltung im Wesentlichen ausgeschaltet ist, ein Mikrocontroller also hauptsächlich schläft, ist der Ruhestromverbrauch dieses einzelnen Bauteils geringer als die Menge der Logik-Gatter, die alle etwa 10-15 μA ziehen würden (ATTiny44 im tiefsten Schlafmodus nur 16 μA).

Aber es hat auch seine Nachteile, einen Mikrocontroller einzusetzen:

- Programmierumgebung/ Programmiergerät notwendig
- höherer Preis (ca. 1€) als einzelne Gatter (jeweils 10-30 Cent)
- aufwändigere Fehlersuche (Programmierfehler, Schaltungsfehler, Laufzeitfehler)

Damit endet der erste Teil. Im zweiten geht es um das Layout.