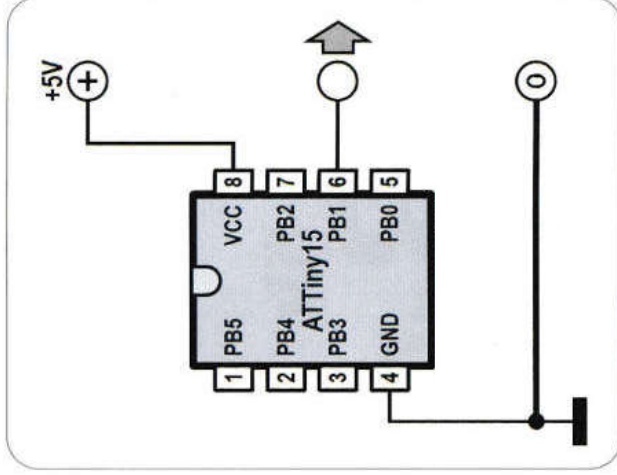




Attiny15 als MHz-Oszillator

Wenn es auf eine absolut genaue Frequenz nicht ankommt, besitzt man mit einem Attiny15 eine schnelle Alternative, die keine zusätzlichen Bauteile benötigt um ein 1-2 MHz Oszillator her zu stellen. Bei einem Preis von rund 1 Euro ist diese Lösung nicht teurer als ein konventioneller Oszillator. Und der Vorteil ist, dass die Frequenz berechenbar ist, unabhängig von Bauteil-Werten oder -Toleranzen. Der interne RC-Oszillator des Controllers ist recht genau auf 1,6 MHz kalibrierbar; der interne Timer 1 lässt sich durch die eingebaute PLL mit 25,6 MHz betreiben. Durch Teilung dieses Taktes kann der Attiny mit einem Minimal-Programm Frequenzen von ca. 50 kHz bis zu 12 MHz auf einem Output-Pin erzeugen, wobei die Differenz zwischen Soll und Ist-Frequenz meist nach oben zunimmt.

Eine sinnvolle Obergrenze ist daher bei ungefähr 2 MHz erreicht; hier kann die Abweichung bereits 15 % betragen. Wie man im Bild erkennen kann, könnte die Schaltung einfacher nicht sein. Nach dem Anlegen der Spannungsversorgung steht das Signal an Pin 6 (PB1) zur Verfügung.



99 % des Programmspeichers frei sind, kann man den kleinen Controller natürlich auch andere Dinge erledigen lassen. Wer es genauer braucht, kann über das Calibration-Byte im OSCCAL-Register (hiermit kann der CPU-Takt in gewissen Grenzen variiert werden) noch einen Abgleich der Output- Frequenz vornehmen. Atmel empfiehlt im Datenblatt, die CPU-Taktfrequenz nicht über 1,75 MHz einzustellen, weil darüber hinaus das Funktionieren der Timer nicht garantiert werden kann.

Statt eines Attiny15 kann alternativ der Nachfolger Attiny45 eingesetzt werden. Dessen Timer 1 kann durch entsprechendes Setzen der CKSEL-Fuses in einen Attiny15-kompatiblen Modus gesetzt werden. Nach Anpassung des Programms lassen sich dann auch höhere (oder genauere) Frequenzen erzeugen, da die PLL des Attiny45 mit 64 MHz arbeitet. <<

Das Beispiel-Programm besteht nur aus 15 Zeilen Assembler-Code. Es ist durch die Kommentare selbsterklärend. Der Code kann von der Elektor-Website heruntergeladen werden [1].

Das Programm initialisiert nur den Timer, der anschließend unabhängig von der CPU das Signal erzeugt. Daher kann die CPU sogar in den Sleep-Modus versetzt werden, um Strom zu sparen. Da noch etwa

WebLink
[1] www.elektormagazine.de/110365