Drehgeber KY040

Für etwas mehr als einen Euro kann man inzwischen Drehgeber wie in Abb. 1 erstehen. Der Drehgeber wird mit 5 V betrieben und besitzt Anschlüsse im üblichen Rastermaß; dadurch lässt er sich direkt an unsere Attiny-Platine anschließen.

Abbildung 1

Beim Drehen erzeugt der Drehgeber an den Ausgängen CLK und DT Signale, aus denen man sowohl die Anzahl der "Raster"-Schritte als auch die Drehrichtung ermitteln kann. In Abb. 2 ist das Timing-Diagramm für die Drehrichtung "links" (von oben gesehen gegen den Uhrzeigersinn) wiedergegeben. Der dargestellte Abschnitt weist vier Phasen auf; er entspricht der Drehung um 2 Raster. Bei weiterer Drehung wiederholt sich dieser Zyklus. Bei einer Drehung nach rechts werden die einzelnen Phasen in umgekehrter Richtung durchlaufen.

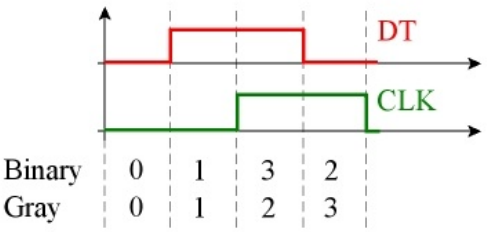

Abbildung 2

Um die Drehbewegung mit unserem Mikrocontroller zu erfassen, schließen wir DT an PortB.0 und CLK an PortB.1 an. Diese zwei Bit können wir zu einer 2-Bit-Zahl zusammenfassen, welche die jeweilige Phase des Drehgebers beschreibt. Ein Zyklus lässt sich bei einer Linksdrehung somit durch die Zahlenfolge 0 → 1 → 3 → 2 beschreiben. In Abb. 2 sind sie in der Zeile "Binary" angegeben.

Die Auswertung der Signale kann in unterschiedlicher Weise erfolgen. Wesentlich ist dabei aber immer, dass die ***Änderung*** der Phase betrachtet wird. Nur so kann die Drehrichtung bestimmt werden.

Eine erste Möglichkeit besteht darin, dass die Phasenwerte vor und nach dem Wechsel zu einer 4-Bit-Zahl zusammengefasst werden; dabei belegen die neue Phase die niederwertigen Bits und die alte Phase die höherwertigen Bits. Ein Phasenwechsel-Wert von 11 = 8 + 3 bedeutet demnach ein Wechsel von Phase 2 nach Phase 3. Rechnerisch gesehen gibt es 16 verschiedene Kombinationen. Davon sind allerdings nicht alle möglich; z. B. gibt es keinen Wechsel von 0 nach 3.

Den Phasenwechsel-Werten können wir nun jeweils eine Drehrichtung zuordnen; diese halten wir in einem Array fest. Leider beginnt bei BASCOM die Indizierung nicht mit 0, sondern mit 1; deswegen müssen wir alle Phasenwechselwerte um 1 erhöhen:

```
Kodierung(1) = 0           'keine Veränderung
Kodierung(2) = 1           'rechts
Kodierung(3) = 255         'links
Kodierung(4) = 128         'nicht möglich
Kodierung(5) = 255
Kodierung(6) = 0
Kodierung(7) = 128
Kodierung(8) = 1
Kodierung(9) = 1           'ab jetzt gespiegelt
Kodierung(10) = 128
Kodierung(11) = 0
Kodierung(12) = 255
Kodierung(13) = 128
Kodierung(14) = 255
Kodierung(15) = 1
Kodierung(16) = 0
```

Damit ist die Programmierung nun recht einfach. Das folgende Programm gibt auf dem Terminal die aktuelle Position aus. Alle benutzten Variablen sind vom Typ Byte. Daraus folgt: Bei der Berechnung des aktuellen Zählerstands bedeutet die Addition von 255 nichts anderes als eine Subtraktion von 1.

```
Zaehler = 0
Letzte_phase = Pinb And &B00000011
Do
  Phase = Pinb And &B00000011
  Phasenwechsel = 4 * Letzte_phase
  Phasenwechsel = Phasenwechsel + Phase
  Phasenwechsel = Phasenwechsel + 1
  Zaehler = Zaehler + Kodierung(phasenwechsel)
  If Letzte_phase <> Phase Then Printbin Zaehler
  Letzte_phase = Phase
Loop
```

Eine zweite Möglichkeit soll noch vorgestellt werden. Sie ist etwas sparsamer mit dem Speicherplatz, aber vielleicht nicht so naheliegend. Die wesentliche Idee besteht darin, die Phasenwerte anders zu kodieren: Zur Beschreibung der Phasen benutzen wir nicht mehr die Binary-Werte aus Abb. 2, sondern Gray-Codes:

0 = 00(G)
1 = 01(G)
2 = 11(G)
3 = 10(G)

Der entscheidende Vorteil ist nun, dass sich innerhalb eines Zyklus bei einer Linksdrehung eine um 1 aufsteigende Folge ergibt und bei einer Rechtsdrehung eine um 1 absteigende Folge. Die

Drehrichtung lässt sich nunmehr sehr leicht aus der Differenz aufeinander folgender Phasenwerte bestimmen. Genaueres kann man dem folgenden Programmcode entnehmen. Leider kennt BASCOM die Verknüpfung XOR nicht als Bit-Operation; deswegen ist die auskommentierte Kodierung nicht möglich.

```
Zaehler = 0

'Gray-code
'Letzte_phase.0 = Pinb.0 Xor Pinb.1
'Letzte_phase.1 = Pinb.1

Letzte_phase = Pinb And &B00000011
Letzte_phase = Letzte_phase Xor Letzte_phase.1

Do
  Phase = Pinb And &B00000011
  Phase = Phase Xor Phase.1
  Differenz = Phase - Letzte_phase
  If Differenz.0 = 1 Then           'Änderung
    If Differenz.1 = 1 Then         'Richtung
      Incr Zaehler
    Else
      Decr Zaehler
    End If
    Printbin Zaehler
  End If
  Letzte_phase = Phase
Loop
```