

```

/* Glockengeläut mit Arduino
*
* Bim... Bam.. Bim...Bam... aber nur optisch. Drei Glocken-Umrisse so aneinander
* gereiht, dass bei einem Knightrider-Lauflicheffekt der Eindruck entsteht, dass
* die Glocken sich bewegen. Alles eine Frage des Timings.
*
* Als Zielprozessor habe ich einen ATTiny 85-20 gewählt, weil er schön klein und
* kompakt ist. Das Timing herauszufinden (Prinzip 'Versuch und Fehler') geht
* mit Arduino aber einfacher. Deswegen dieses kleine Programm. Für den ATiny wird
* es dann später in BasCOM umgesetzt.
*
* Wenn die Energieversorgung zu den Glocken auf 180mA begrenzt wird, ist der
* Wechsel von einer Glocke zur nächsten etwas flüssiger.
*/

#define G1 2 // Glocke 1; links oder rechts
#define G2 3 // Glocke 2; mitte
#define G3 4 // Glocke 3; rechts oder links

#define GDelay1 500 // Verzögerung (ms) in Links- oder Rechts-Position
#define GDelay2 40 // Verzögerung (ms) in der Mitte

// Ports definieren
void setup() {
  // Alle Digital-Pins werden Ausgang
  pinMode(G1, OUTPUT);

```

```
pinMode(G2, OUTPUT);
pinMode(G3, OUTPUT);

// Alle Pins aus
digitalWrite(G1, LOW);
digitalWrite(G2, LOW);
digitalWrite(G3, LOW);
}

// Glocken der Reihe nach mit 'Überlappung' ein- und ausschalten
void loop() {
  digitalWrite(G1, HIGH);
  delay(GDelay2);
  digitalWrite(G2, LOW);
  delay(GDelay1);

  digitalWrite(G2, HIGH);
  delay(GDelay2);
  digitalWrite(G1, LOW);
  delay(GDelay2);

  digitalWrite(G3, HIGH);
  delay(GDelay2);
  digitalWrite(G2, LOW);
  delay(GDelay1);
```

```
digitalWrite(G2, HIGH);  
delay(GDelay2);  
digitalWrite(G3, LOW);  
delay(GDelay2);  
  
}
```