

# Dual Bitmuster Generator

## Anforderungen:

Für die Ansteuerung von zwei Motoren müssen mit einer MCU die Bitmuster generiert werden. Die Motorendrehzahl ist proportional zur Frequenz dieser Bitmuster. Die beiden Drehzahlen sollen unabhängig voneinander durch die beiden Potentiometer P0 und P1 eingestellt werden können.

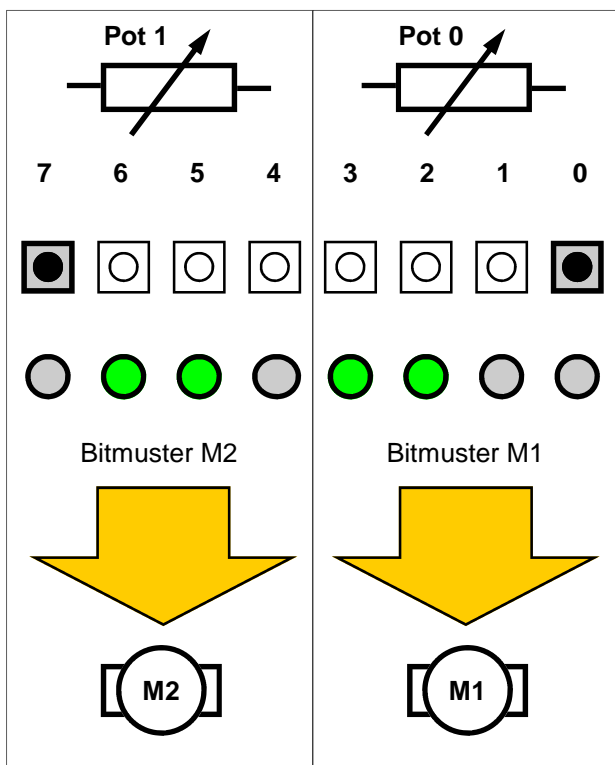
Die Ausgabe der Bitmuster erfolgt über das gleiche 8-Bit-Port und wird auf der LED-Zeile dargestellt. Das untere Nibble steuert den Motor M1,

das obere Nibble steuert den Motor M2.

Mit dem Potentiometer P0 wird die Frequenz des Bitmusters für den Motor M1 eingestellt, mit dem Potentiometer P1 die Frequenz des Bitmusters für den Motor M2.

Solange die Taste T0 gedrückt ist, soll der Motor M1 stoppen. Solange die Taste T7 gedrückt ist, soll der Motor M2 stoppen. Die Motoren stehen still, wenn das Bitmuster nicht mehr ändert. Die übrigen Tasten dürfen auf die Funktion des Systems keinen Einfluss haben.

## Prinzip des Bitmustergenerators, Bitmuster M1 und M2, Portbelegungen:



Schritt	Bitmuster M2	Bitmuster M1
1	0 1 1 0	1 1 0 0
2	1 0 0 1	0 0 1 1
3	0 1 1 0	1 1 0 0
4	1 0 0 1	0 0 1 1
5	0 1 1 0	1 1 0 0
6	1 0 0 1	0 0 1 1
usw.	usw.	usw.

### Schrittdauereinstellung des Bitmusters:

Mit den Potentiometern kann die Zeit T eingestellt werden, während der ein einzelnes Bitmuster am Portausgang aktiv ist.

Pot 0 (M1)	T = 100ms bis 1.35s	(+10%)
Pot 1 (M2)	T = 50ms bis 680ms	(+10%)

## Lösungshinweis:

Dieses Problem wird am einfachsten gelöst, indem die Endlosschleife des Programms in regelmässigen Zeitabständen ausgeführt wird. Bei professionellen Anwendungen wird dazu ein Interrupt benutzt. Sie können in Ihre Endlosschleife einen Delay einbauen. Dieser Delay sollte wesentlich länger sein als die Programmlaufzeit jedoch mindestens so kurz wie das kleinste benötigte Zeitintervall. Durch das Zählen der Anzahl Durchläufe können Sie so beliebige Zeitintervalle generieren.

# Dual Bitmuster Generator

MCT Prüfungsvorbereitung / Serie B

© by B.Wamistrer, Bern 2002

## Anforderungen:

Für die Ansteuerung von zwei Motoren müssen mit einer MCU die Bitmuster generiert werden. Die Motorendrehzahl ist proportional zur Frequenz dieser Bitmuster. Die beiden Drehzahlen sollen unabhängig voneinander durch die beiden Potentiometer P0 und P1 eingestellt werden können.

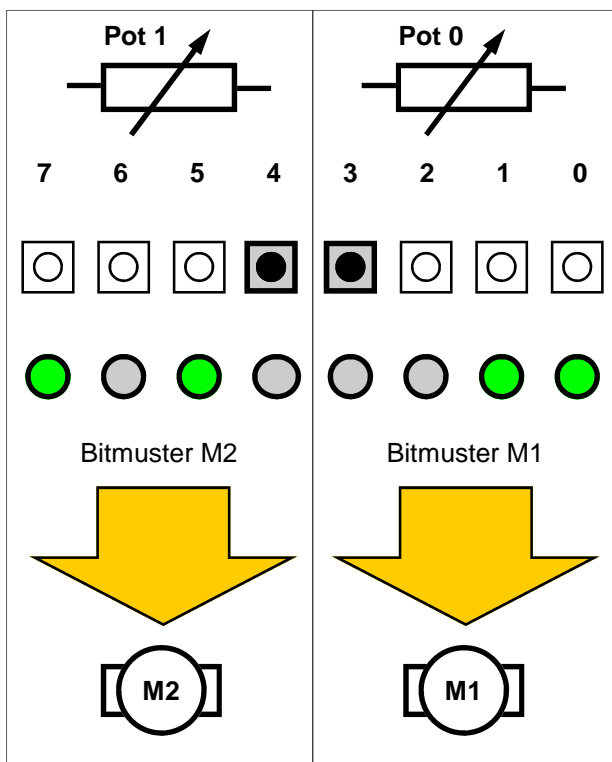
Die Ausgabe der Bitmuster erfolgt über das gleiche 8-Bit-Port und wird auf der LED-Zeile dargestellt. Das untere Nibble steuert den Motor M1,

das obere Nibble steuert den Motor M2.

Mit dem Potentiometer P0 wird die Frequenz des Bitmusters für den Motor M1 eingestellt, mit dem Potentiometer P1 die Frequenz des Bitmusters für den Motor M2.

Solange die Taste T3 gedrückt ist, soll der Motor M1 stoppen. Solange die Taste T4 gedrückt ist, soll der Motor M2 stoppen. Die Motoren stehen still, wenn das Bitmuster nicht mehr ändert. Die übrigen Tasten dürfen auf die Funktion des Systems keinen Einfluss haben.

## Prinzip des Bitmustergenerators, Bitmuster M1 und M2, Portbelegungen:



Bitmuster M2	Schritt	Bitmuster M1
0 1 0 1	1	1 1 0 0
1 0 1 0	2	0 0 1 1
0 1 0 1	3	1 1 0 0
1 0 1 0	4	0 0 1 1
0 1 0 1	5	1 1 0 0
1 0 1 0	6	0 0 1 1
usw.	usw.	usw.

### Schrittdauereinstellung des Bitmusters:

Mit den Potentiometern kann die Zeit T eingestellt werden, während der ein einzelnes Bitmuster am Portausgang aktiv ist.

Pot 0 (M1) T = 100ms bis 1.35s (+-10%)  
 Pot 1 (M2) T = 50ms bis 680ms (+-10%)

## Lösungshinweis:

Dieses Problem wird am einfachsten gelöst, indem die Endlosschleife des Programms in regelmässigen Zeitabständen ausgeführt wird. Bei professionellen Anwendungen wird dazu ein Interrupt benutzt. Sie können in Ihre Endlosschleife einen Delay einbauen. Dieser Delay sollte wesentlich länger sein als die Programmlaufzeit jedoch mindestens so kurz wie das kleinste benötigte Zeitintervall. Durch das Zählen der Anzahl Durchläufe können Sie so beliebige Zeitintervalle generieren.