Reduziertes Tastenfeld programmieren
-
Mit zwei Tasten drei Reaktionen auslösen

Dipl.- Ing. Björnstjerne Zindler, M.Sc.

Erstellt: 23. Oktober 2015 – Letzte Revision: 25. Oktober 2015

Inhaltsverzeichnis

Literatur

[001] Keine für vorliegenden Text.

---

1 Einleitung

Vorliegendes Arbeitsblatt ist keine programmiertechnische Neuentdeckung. Für einen erfahrenen Programmierer sind die folgenden Sätze ein „alter Hut“. Sinn ist es jedoch, von Zeit zu Zeit anfallende Redezeit meinerseits zu minimieren, indem ich die Worte auf ein Papier banne und sei es auch nur virtuell. So hat der fragende Mensch dann etwas zu lesen und ich meine Ruhe.

[001] Aufgabe soll sein die Programmierung von n- verschiedenen Reaktionen eines Programmes auf Tastendruck. So ist es jedoch aufwendig für jede Reaktion je eine Taste vorzusehen, gemäß: „Drücke ich Taste 1 passiert Reaktion 1, drücke ich Taste 2 passiert Reaktion 2, drücke . . . “

So wären für 7 Reaktionen eben 7 Tasten notwendig. Bleibt dann noch die Frage, was passiert, wenn mehrere Tasten gleichzeitig gedrückt werden.

Wobei diese Unsicherheit eben die Grundlage des reduzierten Tastenfeldes ist. Lässt man das gleichzeitige Drücken von Tasten zu, muss es auch eine Ruheaktion geben. Sinnvollerweise ist das dann der Fall, wenn keine Taste gedrückt ist. Das ist aber nicht Grundvoraussetzung.

Idee des reduzierten Tastenfeldes ist, dass jeder Betätigungszustand einen eigenen integren Wert am Ausgang liefert. Keine andere gedrückte Tastenkombination darf dann denselben Wert liefern.

Grundlage ist der Binärcode.

Ein einfaches Beispiel mit zwei Tasten folgt und wird weiter ausgebaut bis zum allgemeinen Fall.

2 Beispiele

2.1 2 Tasten mit 4 Aktionen (1 Ruheaktion, 3 Reaktionen)

X_2	X_1	Y =	Y =
0	0	$0 \cdot X_2 + 0 \cdot X_1$	0
0	1	$0 \cdot X_2 + 1 \cdot X_1$	1
1	0	$1 \cdot X_2 + 0 \cdot X_1$	2
1	1	$1 \cdot X_2 + 1 \cdot X_1$	3

$\Rightarrow$

$$X_1 = 1 \qquad X_2 = 2$$

2.2 3 Tasten mit 8 Aktionen (1 Ruheaktion, 7 Reaktionen)

X_3	X_2	X_1	Y =	Y =
0	0	0	$0 \cdot X_3 + 0 \cdot X_2 + 0 \cdot X_1$	0
0	0	1	$0 \cdot X_3 + 0 \cdot X_2 + 1 \cdot X_1$	1
0	1	0	$0 \cdot X_3 + 1 \cdot X_2 + 0 \cdot X_1$	2
0	1	1	$0 \cdot X_3 + 1 \cdot X_2 + 1 \cdot X_1$	3
1	0	0	$1 \cdot X_3 + 0 \cdot X_2 + 0 \cdot X_1$	4
1	0	1	$1 \cdot X_3 + 0 \cdot X_2 + 1 \cdot X_1$	5
1	1	0	$1 \cdot X_3 + 1 \cdot X_2 + 0 \cdot X_1$	6
1	1	1	$1 \cdot X_3 + 1 \cdot X_2 + 1 \cdot X_1$	7

$\Rightarrow$

$X_1 = 1 \qquad X_2 = 2 \qquad X_3 = 4$

2.3 4 Tasten mit 16 Aktionen (1 Ruheaktion, 15 Reaktionen)

X_4	X_3	X_2	X_1	Y =	Y =
0	0	0	0	$0 \cdot X_4 + 0 \cdot X_3 + 0 \cdot X_2 + 0 \cdot X_1$	0
0	0	0	1	$0 \cdot X_4 + 0 \cdot X_3 + 0 \cdot X_2 + 1 \cdot X_1$	1
0	0	1	0	$0 \cdot X_4 + 0 \cdot X_3 + 1 \cdot X_2 + 0 \cdot X_1$	2
0	0	1	1	$0 \cdot X_4 + 0 \cdot X_3 + 1 \cdot X_2 + 1 \cdot X_1$	3
0	1	0	0	$0 \cdot X_4 + 1 \cdot X_3 + 0 \cdot X_2 + 0 \cdot X_1$	4
0	1	0	1	$0 \cdot X_4 + 1 \cdot X_3 + 0 \cdot X_2 + 1 \cdot X_1$	5
0	1	1	0	$0 \cdot X_4 + 1 \cdot X_3 + 1 \cdot X_2 + 0 \cdot X_1$	6
0	1	1	1	$0 \cdot X_4 + 1 \cdot X_3 + 1 \cdot X_2 + 1 \cdot X_1$	7
1	0	0	0	$1 \cdot X_4 + 0 \cdot X_3 + 0 \cdot X_2 + 0 \cdot X_1$	8
1	0	0	1	$1 \cdot X_4 + 0 \cdot X_3 + 0 \cdot X_2 + 1 \cdot X_1$	9
1	0	1	0	$1 \cdot X_4 + 0 \cdot X_3 + 1 \cdot X_2 + 0 \cdot X_1$	10
1	0	1	1	$1 \cdot X_4 + 0 \cdot X_3 + 1 \cdot X_2 + 1 \cdot X_1$	11
1	1	0	0	$1 \cdot X_4 + 1 \cdot X_3 + 0 \cdot X_2 + 0 \cdot X_1$	12
1	1	0	1	$1 \cdot X_4 + 1 \cdot X_3 + 0 \cdot X_2 + 1 \cdot X_1$	13
1	1	1	0	$1 \cdot X_4 + 1 \cdot X_3 + 1 \cdot X_2 + 0 \cdot X_1$	14
1	1	1	1	$1 \cdot X_4 + 1 \cdot X_3 + 1 \cdot X_2 + 1 \cdot X_1$	15

$\Rightarrow$

$X_1 = 1 \qquad X_2 = 2 \qquad X_3 = 4 \qquad X_4 = 8$

2.4 Der allgemeine Fall

Letztendlich kann eine allgemeine Berechnungsgrundlage angeben werden.

Anzahl der Tasten: N

$\Rightarrow$

Anzahl der Aktionen: 2^N

$\Rightarrow$

Anzahl der Ruheaktion: 1

$\Rightarrow$

Anzahl der Reaktionen: $2^N - 1$

$\Rightarrow$

Wichtung der Taste $1 \leq i \leq N$:

$\Rightarrow$

$$X_i = 2^{i-1}$$

Um eine Oktave (12 Tasten) eines polyphonen Synthesizers zu codieren sind daher 4095 Reaktionen zu programmieren.

3 Umkehrschluss

Es kann die Anzahl N der benötigten Tasten ermittelt werden. Ist eine Anzahl von r Reaktionen erwünscht, gilt für N:

$$r = 2^N - 1$$

$\Rightarrow$

$$N \geq \lg_2 (r+1) = \frac{\ln (r+1)}{\ln 2} = \frac{1}{\ln 2} \cdot \ln (r+1) = \ln (r+1)^{\frac{1}{\ln 2}}$$

Wobei [ ] die Rundung auf die nächst höhere integre Zahl darstellt.

r	N	$[N]$
0	0,000000000	0
1	1,000000000	1
2	1,584962501	2
3	2,000000000	2
4	2,321928095	3
5	2,584962501	3
6	2,807354922	3
7	3,000000000	3
8	3,169925002	4
9	3,321928095	4
10	3,459431619	4
11	3,584962501	4
12	3,700439719	4
13	3,807354922	4
14	3,906890596	4
15	4,000000000	4
16	4,087462842	5

4 Programmbeispiel

Wie man das alles programmtechnisch implementieren könnte, zeigt ein kleiner Codeschnipsel für zwei Tasten.

```
//
// Beginn Programmbeispiel
//
void Idle_0(){ . . . } // Definition Methode Ruheaktion 0 (fakultativ )
void Reac_1(){ . . . } // Definition Methode Programmreaktion 1 (obligatorisch)
void Reac_2(){ . . . } // Definition Methode Programmreaktion 2 (obligatorisch)
void Reac_3(){ . . . } // Definition Methode Programmreaktion 3 (obligatorisch)
//
int X_1 = 1; // Definition der Taste X_1 an Eingabepin 1
int X_2 = 2; // Definition der Taste X_2 an Eingabepin 2
//
pinMode(X_1, INPUT); // Definiert X_1 als Eingang
pinMode(X_2, INPUT); // Definiert X_2 als Eingang
//
int Status_X_1 = 0; // Definition Abfrage Taste gedrueckt (1) oder nicht (0)
int Status_X_2 = 0; // Definition Abfrage Taste gedrueckt (1) oder nicht (0)
//
void loop() // Staendige Tastaturabfrage hier als Schleife
{
//
if (digitalRead(X_1) == HIGH){Status_X_1 = 1;} else {Status_X_1 = 0;}
if (digitalRead(X_2) == HIGH){Status_X_2 = 1;} else {Status_X_2 = 0;}
//
switch ( Status_X_2 * 2 + Status_X_1 * 1 )
{
case 0 : // Startet Ruheaktion wenn Y = 0 ergibt
Idle_0();
break ;
case 1 : // Startet Reaktion 1 wenn Y = 1 ergibt
Reac_1();
break ;
case 2 : // Startet Reaktion 2 wenn Y = 2 ergibt
Reac_2();
break ;
case 3 : // Startet Reaktion 3 wenn Y = 3 ergibt
Reac_3();
break ;
}
}
//
// Ende Programmbeispiel
//
```

Wobei in der **switch**- Anweisung der **Y**- Wert des reduzierten Tastenfeldes generiert wird.

LaTeX