Kasino

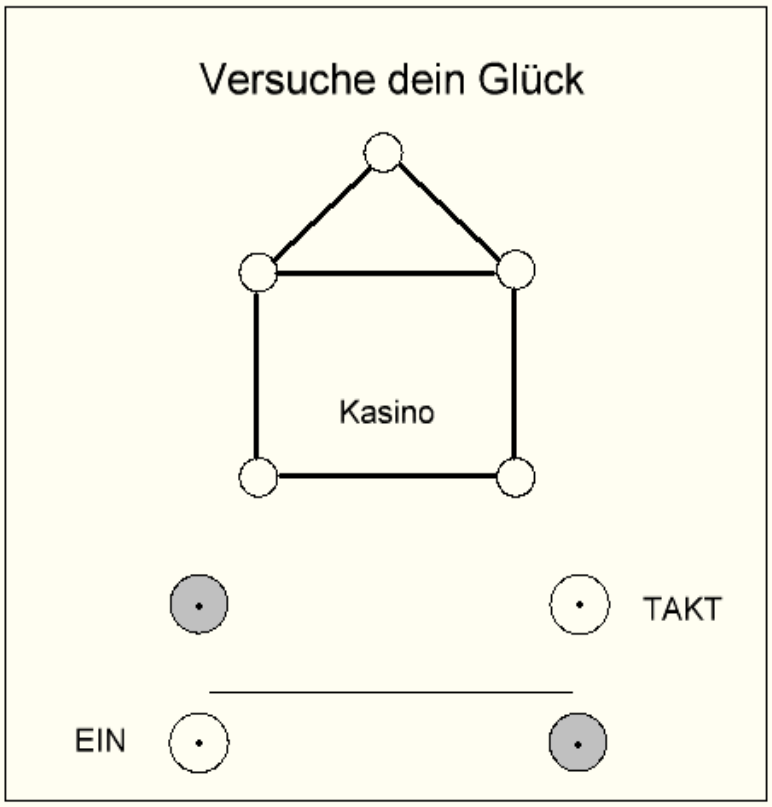
Ich stand vor der Herausforderung, mit geringsten Kosten eine Elektronik anzubieten, die für die Schüler ein interessantes Objekt sein sollte. Dabei wurde ebenso gewünscht, dass einige Lötverbindungen selbst ausgeführt werden sollten. Denn das „Elektronik-Löten“ macht den Kindern großen Spaß.

Lösung

Die Mikroprozessoren Atmega: Attiny45 oder Attiny 13A sind im Versandhandel schon für 0,70 € zu erhalten. Die Lötverbindungen sind wegen der hohen Integrationsdichte des Chips übersichtlich und nicht zu umfangreich. Eine frühere Ausführung mit CMOS-Bausteinen war aufwendiger und wesentlich teurer als der Einsatz eines Mikroprozessors.

Alle Teile inkl. Mikroprozessor kosten < 3 €.

Das Spiel

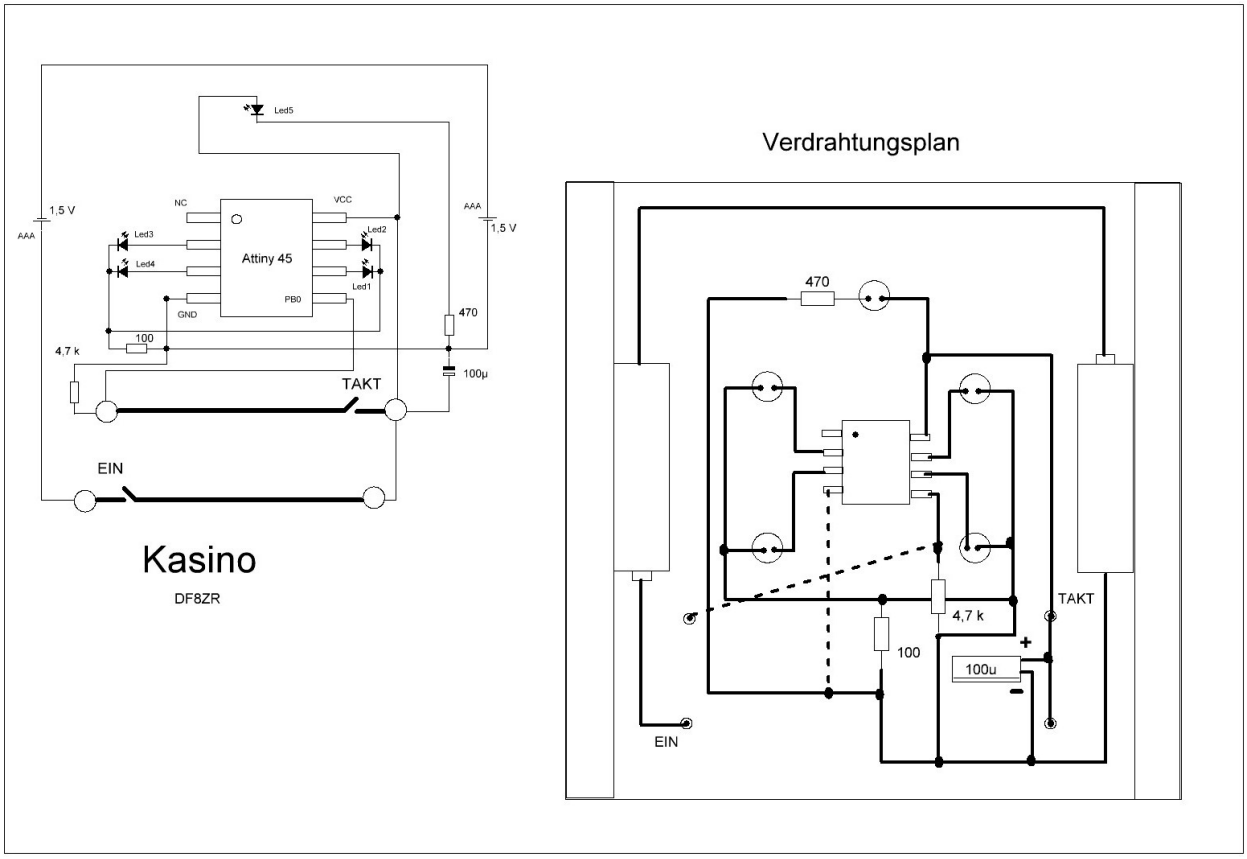


Der Spieler muss anhaltend die Betriebstaste drücken, denn dadurch wird die Schaltung mit Strom versorgt. Da der Mikroprozessor bereits ab einer Versorgung

von 1,8 V arbeitet, war es naheliegend, hier zwei einfache Stabzellen zu verwenden. Die LEDs haben ja auch eine Flussspannung von mehr als 1,6 V. Zwei Zellen vom Typ AAA bieten hinreichend Kapazität für eine Spieldauer von einigen Stunden. Zur Einsparung von Kosten werden die Zellen einfach mit Heißkleber an den Innseiten der Holzleisten befestigt. Die Anschlüsse werden dauerhaft angelötet. Dazu werden sie zuvor mit Schmirgelleinen etwas aufgeraut. Die meisten Leitungen werden mit Blankdraht verlegt. Einige Drähte kann man isoliert ausführen.

Die oberste LED am Haus leuchtet solange, wie man die Betriebstaste gedrückt hält. Nur durch geschicktes Betätigen des Tasters TAKT gelingt es, alle LEDs dauerhaft zum Leuchten anzuregen. Nicht jeder Spieler schafft das sofort. Meistens muss man viele Male den TAKT-Taster schließen und öffnen.

Schaltung und Software



Die Software ist(in Bascom) so geschrieben, dass nur 4 Ports als Outputs benutzt werden. Der Port PB0 ist als Input geschaltet. PB5 ist der RESET und wird hier nicht als I/O verwendet, weil ich sonst mit meinen bescheidenen Mitteln die Chips nicht flashen kann. Die Fuse-Bits bleiben also unverändert. Wer die Möglichkeit hat, einen

Bootloader zu flashen, der könnte auch PB5 verwenden. Allerdings bleibt man dann im Unklaren, ob das Spiel eingeschaltet ist, weil ein Zustand ohne Leuchten aller LEDs nicht vermieden wird.

```
 1
 2
 3
 4
 5
 6 $regfile = "attiny13.dat"
 7 $Crystal=1000000
 8 $hwstack=16
 9 $swstack=16
10 $framesize=32
11
12
13 'Es handelt sich um ein Geschicklichkeitsspiel!
14
15
16 'Zum Spiel:
17
18 'Der Spieler muss anhaltend den EIN-Taster geschlossen halten: So lange leuchtet
19 'die obere LED, der Mikroprozessor wird mit Strom versorgt.
20 'Dann muss er den TAKT-Taster in dem Augeblick lösen, wenn alle LEDs leuchten.
21 'Das kann gelingen, setzt aber eine schnelle Reaktion  voraus, denn
22 'das Debouncing dauert 25 ms. Während dieser Zeit nimmt der Pozessor
23 'keine Befehle entgegen, falls man den TAKT-Taster zwischdurch öffnet.
24 'Drückt man also innerhalb dieser Zeit den Taster, so bleibt das ohne Wirkung.
25 'Erst durch geschicktes Betätigen des Tasters gelingt es, alle LEDs
26 'gleichzeitig zum Leuchten zu bringen
27
28
29
30
31
32
33
34
```

```
35 '$regfile = "attiny13.dat"
36 '$Crystal=4000000
37 '$hwstack = 16
38 '$swstack=16
39 '$framesize = 16
40
41 '$crystal = 1000000
42
43
44  'es handelt sich um einen ATmega13a, aber er kompiliert nur mit ATtiny13!
45
46  ' aber so funktioniert das Kasino auch!!!
47
48 Config PINB.0 = Input
49
50   T Alias PINB.0                                          'Taster
51
52
53
```

```
54
55 Config PINB.1 = Output
56 Config PINB.2 = Output
57 Config PINB.3 = Output
58 Config PINB.4 = Output
59 'Config Pinb.5 = Output
60 'der PB5 ist der RESET. Habe ihn nicht über Fusebits deaktiviert,
61 'weil ich sonst mit meinen Mitteln den Chip nicht flashen konnte
62
```

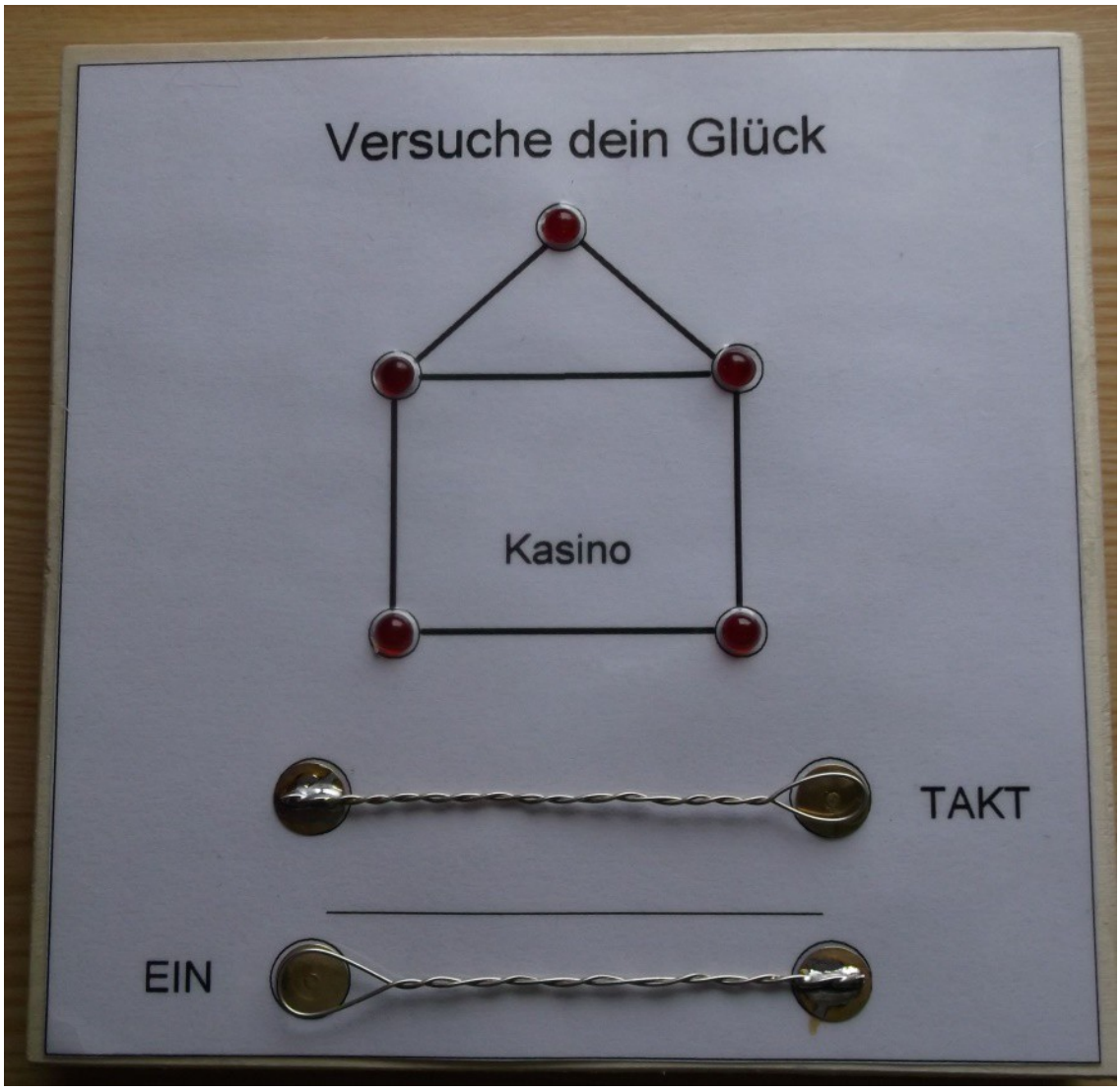
```
63
64 Led1 Alias PINB.1
65 Led2 Alias PINB.2
66 Led3 Alias PINB.3
67 Led4 Alias PINB.4
68 'Led5 Alias Pinb.5
69 'Die Led5 ist die Betriebsanziege, die immer leuchtet, solange
70 'man den Betriebstaster drückt
71
72
73 Do
74
75 Anfang:
76
77 Led1 = 0
78 Led2 = 0
79 Led3 = 0
80 Led4 = 0
81
82
83
84
85 Do
86
87 Debounce T , 1 , Ontaster
88  'reagiert nur   auf die Flanke von 0 -> 1
89  'wartet auf das Schließen des Tasters TAKT
90
91
92
93 Loop
94
95 '_____________________________________________________
96
```

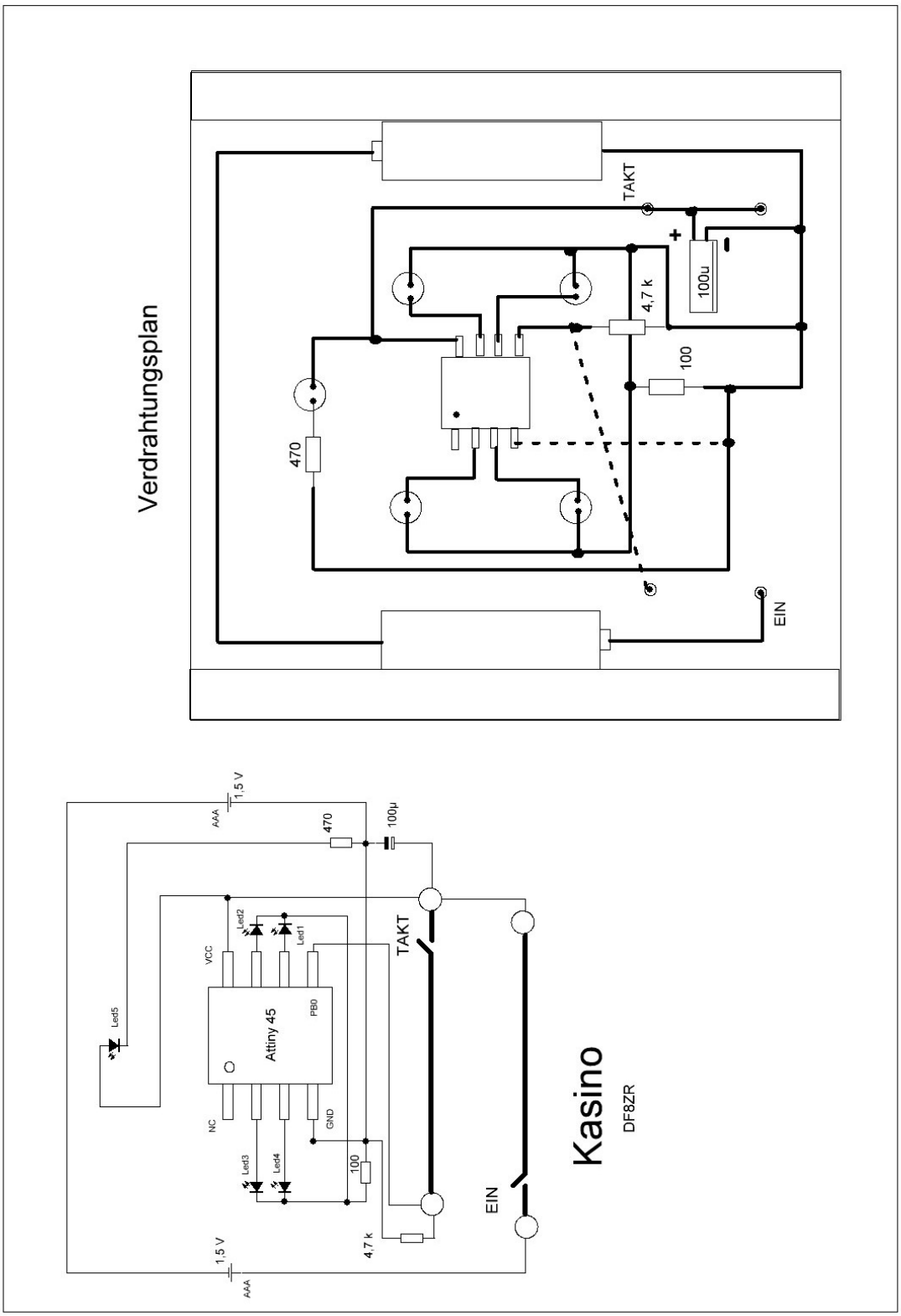
```
97
98 Ontaster:
99
100
101
102 Led1 = 1
103 Waitms 100
104
105
106 If T = 0 Then Goto Anfang
107    'T ist 0, wenn Taster losgelassen wurde
108
109 Led2 = 1
110 Waitms 100
111
112
113 If T = 0 Then Goto Anfang
114
115 Led3 = 1
116 Waitms 100
117
118
119 If T = 0 Then Goto Anfang
120
121 Led4 = 1
122 Waitms 50
123
124 'hier muss der taster TAKT geöffnet werden, wenn man alle LEDs
125 'leuchten lassen will
126
127 Led1 = 0
128
129
130
131
132 Return
```

```
133 '_____________________________________________________
134
135
136 Loop
137
138 'Ende des Programms
139 End
```

Fotos

Prototyp:





DF8ZR; im Februar 2015

Kosten

32 Bretter 13,5 x 13,5 cm; 4 mm dick = 7,99 € Sperrholz Pappel

+ 6 Stck Leisten 1m x 12 x 12 mm = 9,54 € -> 17,53 € Holz -> 0,55 € / Brett

---

Holz : 0,55 €/ Schüler

+ 2 Stck AAA Batterien = 0,40 €/ Brett

+ 0,70 €/ Mikroprozessor

+ Heiß-Kleber 0, 10 /Brett

+ Vorlage 0,05 / Brett

+ Widerstände a' 0,10€/Stck = 0,30 €

+ 1 Kondensator 100u/16V 0,04 €/Brett

+ Reißzwecken 0,04 €/Brett

+ Draht blank 10m/2,10 € --> 0,15 €/Schüler

+ Draht isoliert 0,1 €/Schüler

+ Lötzinn 0,5 m ca. 0,1 €/Brett

+ IC-Fassung 0,04 €/Stck

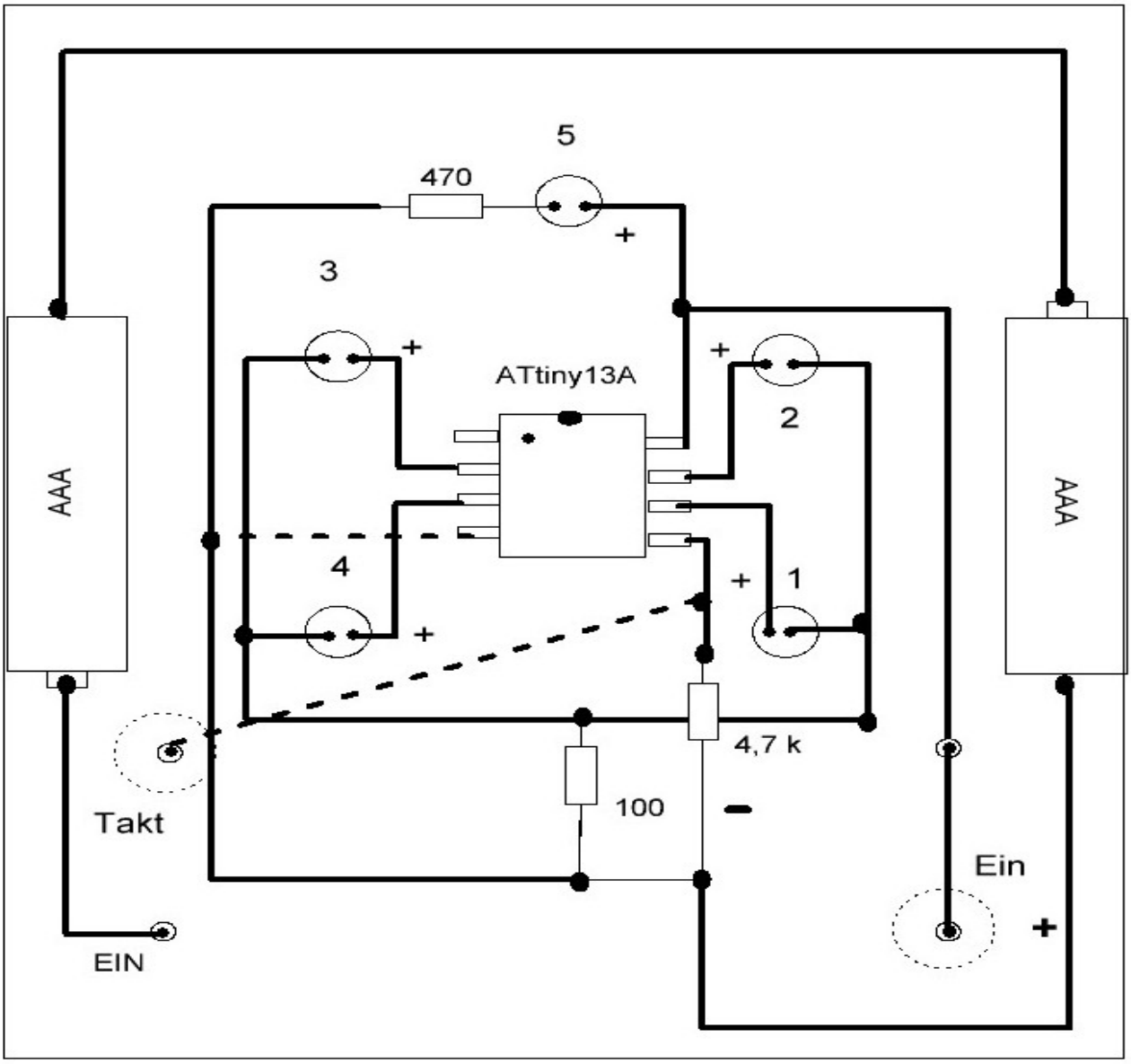
+ 5 Stck LED rot 0,45 €/Brett

+ zerstörte Teile austauschen; Werkzeugeinsatz(Bohrer,Schmirgelpapier, Uhu)

2,92 -> **3,00 €/Schüler für Kasino**

Kasino verdrahten

Ihr bekommt von mir das Brett mit den beiden angeklebten Leisten. Die Löcher für die Leuchtdioden sind bereits gebohrt. Ihr müsst noch die Vorlage aufkleben, danach die vier Reißzwecken durchdrücken. Später stellen wir dann gemeinsam die beiden verdrillten Blankdrähte für die Taster her. Ich muss nach dem Aufkleben der Vorlage noch die „Löcher“ für die fünf Leuchtdioden aus dem Papier ausstanzen. Doch nun zum Wichtigsten:



Die gestrichelt gezeichneten „Linien“ sind isolierte Drähte, die dick gedruckten Blankdrähte!

Diesen Verdrahtungsplan habe ich bereits auf die Rückseite des Brettes geklebt. Die fetten Punkte an den Drähten bedeuten Lötverbindungen, die ihr herstellen sollt. Die Anschlussdrähte der Leuchtdioden habe ich dünn gezeichnet. Es ist wichtig, dass der lange Draht einer LED dort herauskommt, wo ein Pluszeichen ist. Da fließt nämlich der Strom hinein. Vertauscht ihr die Anschlussdrähte, dann wird die LED nie leuchten. Also steckt bitte die Leuchtdioden so hinein, dass die Anschlussdrähte richtig liegen.

Die Batterien werden von mir mit Heißkleber befestigt. Ich werde die kritischen Anschlüsse daran verzinnen und vielleicht auch kurze Drähte anlöten, sodass es für euch etwas einfacher ist, die Verbindungen herzustellen. Den isolierten Draht, der die beiden Batterien in Reihe schaltet, lötet ihr zuletzt an. Erst wenn die Schaltung fast fertig ist und eine Überprüfung mir zeigt, dass keine wesentlichen Fehler zu erwarten sind, setzen wir die Batterien ein. Der Grund ist, dass während des Aufbaus der Schaltung ohne Batterien keine Fehlerströme fließen können, die die Dioden zerstören würden. Deshalb kommt auch ganz zuletzt erst der Chip in die Fassung. So ein Mikroprozessor ist ein vergleichsweise teures Bauteil. Schon ein falscher Anschluss kann ihn zerstören.

Die Batterien halten einige Stunden. Wer die Schaltung schützen möchte, kann einen Deckel aus Pappe an die Leisten kleben. So können auch andere Spieler nicht versehentlich in die Verdrahtung fassen und Schäden anrichten.

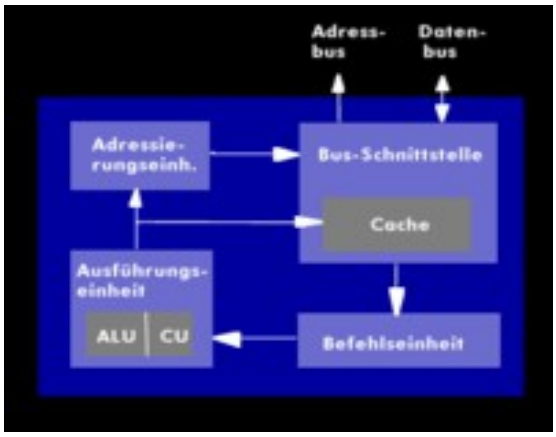
Insgesamt werden wir etwa vier bis fünf Stunden mit dieser Bastelarbeit zubringen. Beim Basteln ist eure Geschicklichkeit gefordert. Es kommt nicht darauf an schnell zu sein! Geht mit Zuversicht die Sache an und konzentriert euch bei der Arbeit. Ihr übt dabei Drähte korrekt zu biegen und vor allem das Elektronik-Löten und dürft stolz auf euer Werk sein. Es macht bestimmt auch immer wieder Spaß, damit zu spielen.

Man kann das Programm auch als Script(Scratch) ausprobieren:
https://drive.google.com/drive/folders/1H7pfZ0ZSuWKnRrRZoO0xXQm8KREfrUp1
Hier heißt es “hexenhaus”

Was ist ein Mikroprozessor?

Diese Frage habt ihr euch sicherlich auch schon gestellt. Und ja, er ist ein moderner integrierter Schaltkreis, auch Chip genannt, mit dem man komplizierte elektronische

Vorgänge höchst wirkungsvoll ablaufen lassen kann. Er ist tatsächlich ein elektronischer Rechner in kompakter Ausführung, ein Wunder moderner Technik. Allerdings mit begrenzten Eigenschaften, die nicht alles ermöglichen, was ein Personal Computer kann. Aber wir können mit diesem Schaltkreis viele kleine Steuerungen, Überwachungen oder sogar Spiele verwirklichen, ohne viel Aufwand treiben zu müssen, wie er bei der gleichen Aufgabe von einfachen Schaltkreisen aus den Logik-Familien sonst erforderlich wäre. Durch die hohe Integration der internen Elemente ist ein Mikroprozessor ein kleiner Tausendsassa in der Welt der Elektronik.



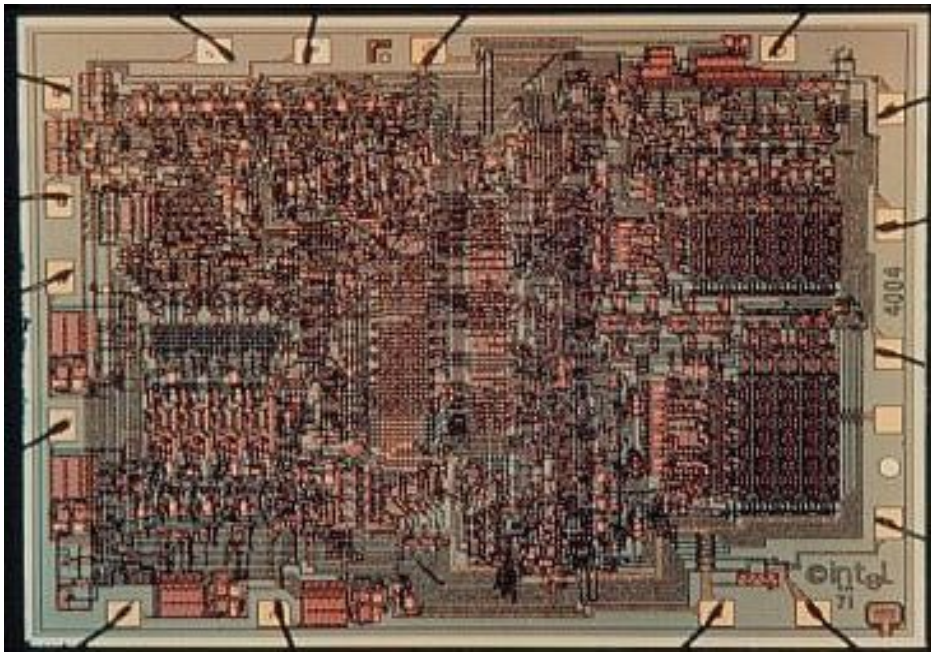
aus „itwissen.info“

Das Bild zeigt die innere Aufteilung in Funktionsblöcke. Es stellt den eigentlichen Kern dar, die zentrale Recheneinheit. Dieser wird auch CPU = „central processing unit“ genannt. Die CPU gehorcht der Befehlseinheit und sie hat einen Speicher(Cache) für die Zwischenablage von Ergebnissen. Nach außen verständigt sich die CPU über einen Adressbus und einen Datenbus. Der Adressbus schaltet die in einem großen Datenspeicher befindlichen Befehle durch. Der Datenbus stellt Daten zur Verfügung, mit denen etwas gemacht werden soll und er hat auch eine Verbindung zum Massenspeicher. Die Daten können an verschiedene Schnittstellen geleitet werden und kommen so z.B. als auszudruckende Zeichen auf das externe Gerät oder den Bildschirm. Der Datenbus nimmt auch die Zeichen von der Tastatur an und leitet sie an die Recheneinheit weiter. In unserem Fall erkennt er, dass wir den Taster für den TAKT schließen oder öffnen.

Die Vorgänge werden synchron ausgeführt. Dabei gehorchen alle internen Einheiten einem gemeinsamen Takt. So weiß die CPU stets, dass die gerade anstehenden Daten im richtigen Augenblick einer Speichstelle korrekt zugeordnet und zu verarbeiten sind. Die Bearbeitung der Daten erfolgt oft in mehreren Schritten vom Takt

bestimmt. Er regelt auch die synchrone Ablage in den großen Massenspeicher(Festplatte).

Ein Mikroprozessor hat keine Festplatte. Unser ATtiny13A hat gerade mal nur 1024 Speicherstellen, die als Halbleiterzellen auf dem Chip integriert sind. Für unsere kleine Schaltung sind die wenigen Speichstellen aber mehr als ausreichend. In einem speziellen Speicher liegen die Daten des Programms, die Befehle. Er behält sie auch dann, wenn keine Batterie den Chip mit Strom versorgt. Diese Daten sind wie bei Computern üblich als Nullen und Einsen elektrisch gespeichert. Die Reihenfolge dieser Muster von „Zahlen“ ist das eigentliche Programm, das von der CPU verstanden wird. Und auf dem Bildschirm werden die Computerzahlen wieder zu Buchstaben und Zeichen, die wir Menschen verstehen. Das alles macht der Mikroprozessor in unserem Kasino eine Million Mal in der Sekunde. Denn so schnell ist der interne Takt. Mit einem externen Element, das eine Schwingung mit konstanter Frequenz machen kann, einem sog. Quarz, kann man die Schnelligkeit auf das Vierfache steigern. Brauchen wir aber nicht, denn der Wechsel des Aufleuchtens der LEDs im „Kasino“ geschieht in einer Zehntelsekunde und wir haben damit schon die Grenzen unserer persönlichen Reaktionsfähigkeit erreicht. Es gelingt deshalb nicht immer, schon nach kurzer Zeit alle Dioden gemeinsamen zum Leuchten zu bringen.



Ca 2000 Transistoren bildeten den ersten Mikroprozessor

Auf einem Plättchen(Target) aus Silizium wurden einige Tausend Transistoren hergestellt und mit Leiterbahnen verbunden. Letztere sind heute manchmal nur 15 Nanometer breit. Ein Grippevirus ist ca. 50 nm dick. Ohne besondere Maschinen wäre das alles nicht möglich. Und die Anschlüsse, die nach außen an die Pins führen, werden innerhalb des Kunststoffgehäuses mit ganz dünnen Drähten gemacht, die an bestimmte Stellen des Chips angeschweißt sind. Im nächsten Bild(cczwei.de) ist so ein „Target“ zu sehen. Die Einzelheiten wurden stark vergrößert fotografiert.

Am Rand des Chips sind die etwas “dickeren“ Drähte zu erkennen. Ganz rechts unten ist die Struktur eines Speicherbereiches mit dem typischen Muster auszumachen. Wir sehen aber auch viele dünne Verbindungslinien. Es handelt sich um einen vergleichsweise sehr einfachen Mikroprozessor aus der Anfangszeit der Entwicklung dieser Bauelemente.

Die Programmierung habe ich in einer sog. höheren Programmiersprache gemacht. Sie nennt sich „Bascom“ und ist von der sehr bewährten Computersprache BASIC abgeleitet. Mit einem weiteren Programm werden die von mir in Zahlen und Buchstaben(Wörtern) notierten Befehle in die Computerzahlen umgesetzt. Man nennt so ein Umsetzprogramm Compiler. Auch der Compiler ist letztlich wieder nur als Folge von Computerzahlen für meinen PC verständlich. Sein Ergebnis ist dann die Reihenfolge derjenigen Computerzahlen, die unser Mikroprozessor versteht. Diese Befehle sind in einer Datei abgelegt und wurden von meinem PC auf den Befehlsspeicher des Mikroprozessors dauerhaft übertragen. Dort werden sie aktiv, wenn wir den EIN-Taster drücken. In einer Schleife fragt der Mikroprozessor ständig ab, ob du die TAKT-Taste betätigst oder wieder löst. Er zieht daraus nach einer bestimmten Vorgabe(Algorithmus) und von mir vorgegebenen Logik seine Schlüsse und lässt die Dioden leuchten oder nicht.

Soweit soll‘s genug sein. Wenn du Programmierer werden willst, mache dich im Netz schlau. Es ist heute so einfach, das Programmieren zu erlernen. Einen Mikroprozessor kann man für 0,70 € kaufen. Bascom gibt es als Download kostenlos.

DF8ZR und Elektronika; im März 2015