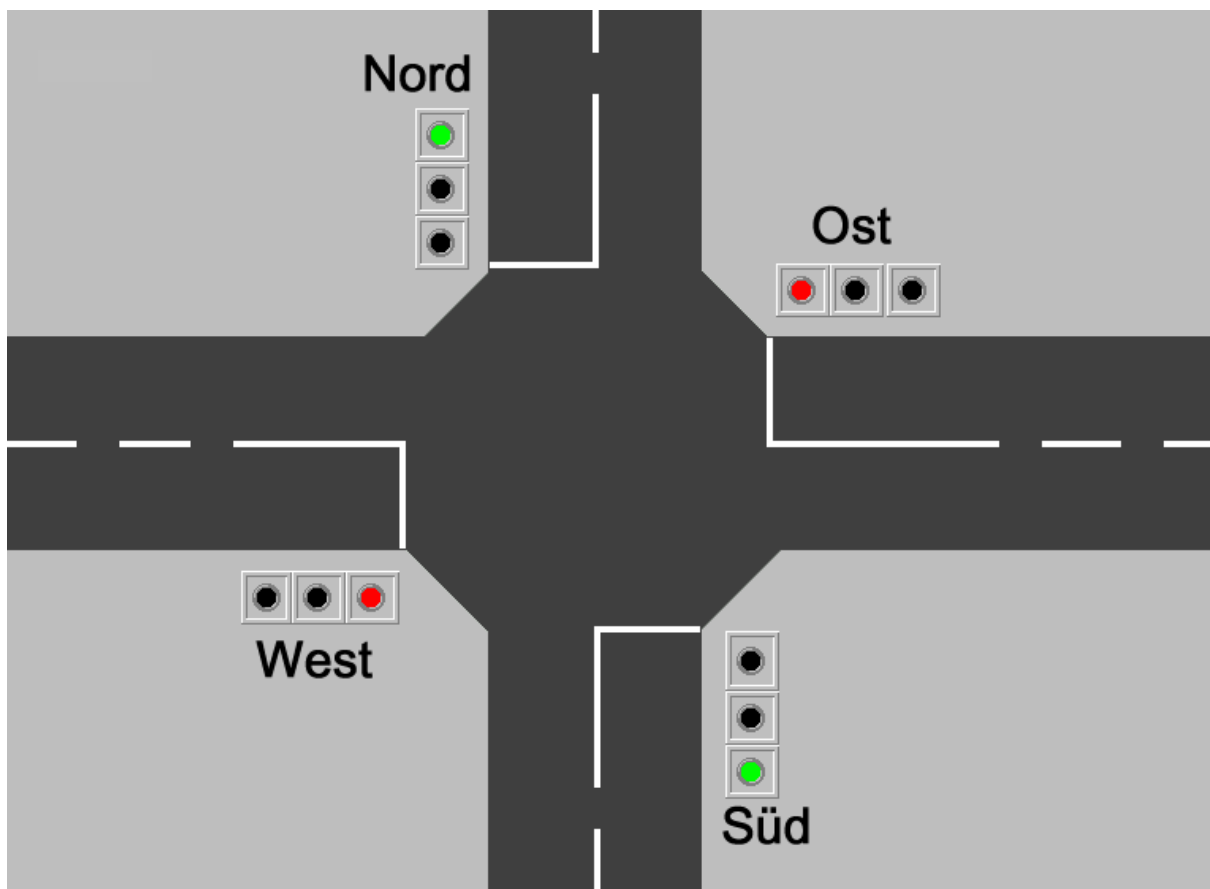


## Eine Modell-Ampel und drei Wege zur Realisierung

Gunther Zielosko

### 1. Sinn und Zweck

Falls Sie eine Modellbahnanlage besitzen, ein Stadtmodell mit Legosteinen aufbauen wollen oder einfach nur Kinder bzw. Enkelkinder haben, die gern mit Autos spielen, es gibt Anlässe genug, irgendwo eine automatisch funktionierende Ampelanlage aufzubauen. Eine einzelne Ampel ist dabei noch keine allzu große elektronische Herausforderung, eine komplette Anlage mit 4 gesteuerten Ampeln schon eher. Wir werden in dieser Applikation verschiedene Wege aufzeigen, eine solche Ampelanlage zu realisieren. Allerdings wird nur eine davon mit dem Tiger arbeiten. Wir werden für die einzelnen Signal-Lampen farbige LED's verwenden. Dabei wird die Ampelanlage so konzipiert, dass sie mit jedem der drei vorgeschlagenen Steuerungssysteme sofort zusammen arbeiten kann. Wie die Anlage mechanisch aufgebaut wird, überlassen wir jedem Einzelnen. Sie können je nach Geschick und Erfordernis kleine oder große, „echte“ Ampeln mit Mast oder nur in ein Brett eingelassene LED's verwenden.



*Bild 1 unsere virtuelle Ampelanlage auf dem PC*

Auf der Ampelseite sehen wir zunächst 4 Ampeln mit je drei LED's (rot, gelb, grün). Alle LED's erhalten einen Vorwiderstand von 50 bis 500 - je nach LED-Typ, gewünschter Helligkeit und Treibereigenschaften. Moderne LED's sind bei 5...10 mA bereits so hell, dass es für die Ampel-Anwendung reichen sollte. Da die LED's in den verschiedenen Farben auch unterschiedliche „Brennspannungen“ haben, sollten Sie die Vorwiderstände so dimensionieren, dass alle LED's mit dem gleichen Strom (z.B. 5mA) betrieben werden, wenn Sie an die Steuerleitung z.B. 5V anlegen, wahrscheinlich ist Ausprobieren die beste Lösung.

Um später die Übersicht zu behalten, werden wir diesen LED's unverwechselbare Namen geben, einfach nach Himmelsrichtung und Farben sortiert:

Rot\_Nord                    die rote LED im Norden (oben)  
Gelb\_Nord                usw.  
Grün\_Nord

Rot\_West                   die rote LED im Westen (links)  
Gelb\_West                usw.  
Grün\_West

Rot\_Ost                    die rote LED im Osten (rechts)  
Gelb\_Ost                usw.  
Grün\_Ost

Rot\_Süd                    die rote LED im Süden (unten)  
Gelb\_Süd                usw.  
Grün\_Süd

Da unsere Ampel für verschiedene Ansteuerungen verwendbar sein soll, werden wir die Steuerleitungen für jede einzelne LED zunächst alle an einen Pfostensteckverbinder führen. Da aber in der Regel die Nord- und Südampele bzw. die Ost- und Westampele das gleiche Signalspiel haben, können wir die 12 Steuerleitungen ggf. auf 6 reduzieren. Im Falle der PC-Steuerung ermöglicht dies z.B. die Steuerung mit den 8 Datenleitungen der Parallel-Schnittstelle.

Bild 2 zeigt die Verdrahtung der 12 LED's, wie sie für alle Ansteuerungsvarianten gleichermaßen benutzt werden kann.

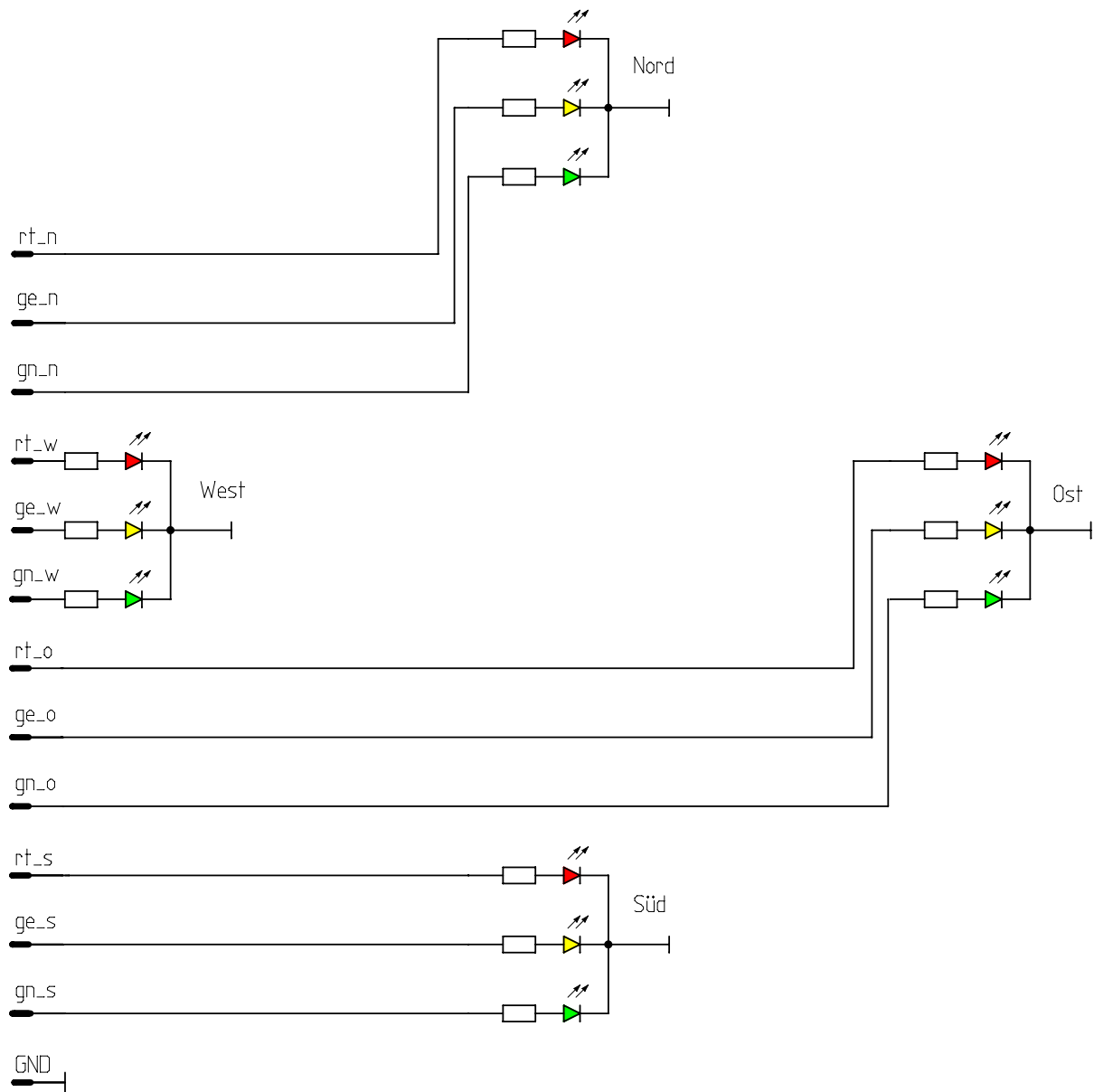


Bild 2 die Verdrahtung der 12 Ampel-LED's

## 2. Die erste Variante – Steuerung durch den PC

Die folgende Lösung erfordert steuerungstechnisch den geringsten Aufwand. Ein PC mit Druckerschnittstelle ist die Steuerzentrale. Ein Kabel verbindet die Modell-Ampelanlage mit dem PC. Der Clou an unserer Software-Lösung ist, dass neben der Steuerung der Modellampel eine simulierte Ampelanlage synchron auf dem Bildschirm mitläuft.

PC's mit Druckerschnittstelle (oder so genannte Parallelschnittstelle) besitzen normalerweise eine 25-polige SUB-D-Buchse.

Für unsere Anwendung darf diese nicht mit einer eventuell auch vorhandenen 25-poligen seriellen Schnittstelle verwechselt werden!



Bild 3 25-poliger SUB-D-Stecker mit Pinbelegung

Pin	Bezeichnung	In/Out	Funktion
1	/STROBE	Out	Strobe (Startimpuls)
2	D0	Out	Datenbit 0
3	D1	Out	Datenbit 1
4	D2	Out	Datenbit 2
5	D3	Out	Datenbit 3
6	D4	Out	Datenbit 4
7	D5	Out	Datenbit 5
8	D6	Out	Datenbit 6
9	D7	Out	Datenbit 7
10	/ACK	In	Acknowledge
11	BUSY	In	Busy
12	PE	In	Paper end
13	SELECT	In	Select in
14	/AUTOFD	Out	Autofeed
15	/ERROR	In	Error
16	/INIT	Out	Initialisierung und Reset
17	/SELECT	Out	Select out
18	GND		Masse
19	GND		Masse
20	GND		Masse
21	GND		Masse
22	GND		Masse
23	GND		Masse
24	GND		Masse
25	GND		Masse

Tabelle 1 Standardbelegung der Parallel-Schnittstelle

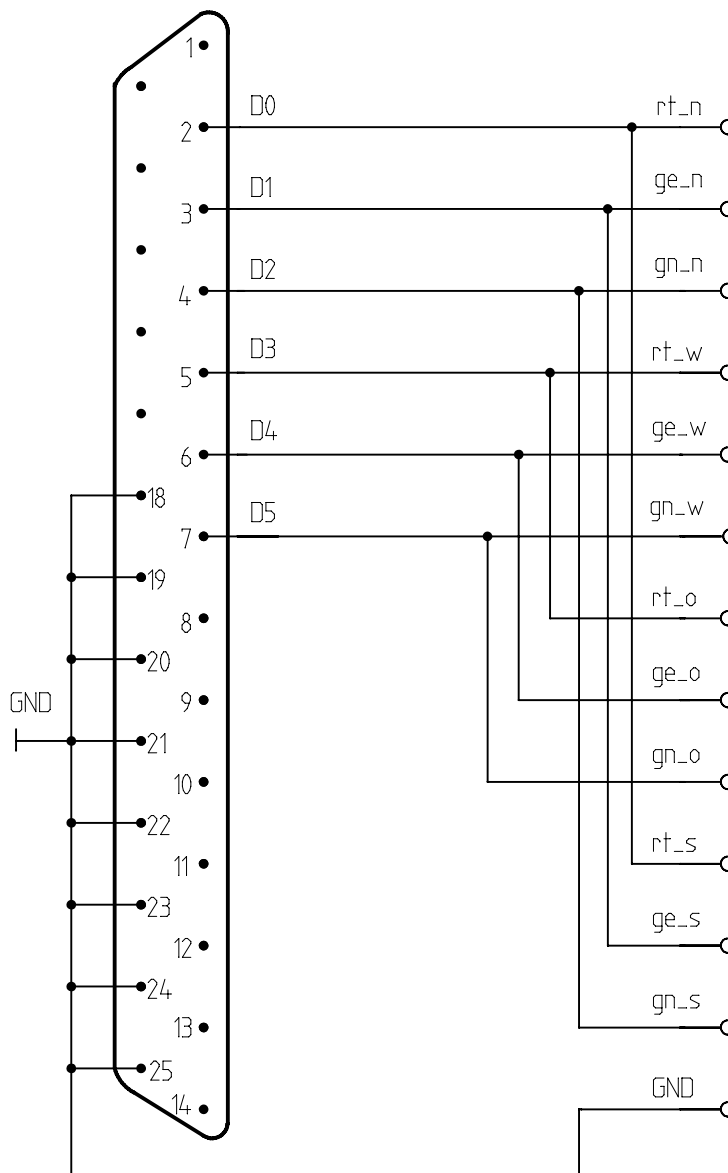
Im Laufe der Zeit wurde die Parallel-Schnittstelle mit immer mehr Funktionen ausgestattet, so wurden z.B. die Data-Pins zu bidirektionalen I/O-Pins (funktionieren auch als Eingang). Für unsere Anwendung ist dies jedoch ohne Belang.

An diese Buchse wird ein Kabel mit 25-poligem SUB-D-Stecker wie folgt angeschlossen:

zum Parallelport

25-pol. SUB-D-Stecker

Frontansicht



*Bild 4 Anschlusskabel für die Parallel-Schnittstelle des PC*

25-pol. SUB-D-Stecker Pin	Funktion	Ampel-LED's
2	D0	Rot_Nord + Rot_Süd
3	D1	Gelb_Nord + Gelb_Süd
4	D2	Grün_Nord + Grün_Süd
5	D3	Rot_Ost + Rot_West
6	D4	Gelb_Ost + Gelb_West
7	D5	Grün_Ost + Grün_West
18...25	GND	GND

*Tabelle 2 Anschluss an die Ampelanlage*

Da die Treiber der Parallel-Schnittstelle robust genug ausgelegt sind, kann eine Leitung ohne Probleme 2 LED's treiben (z.B. Rot\_Nord + Rot\_Süd). Das macht insbesondere dadurch Sinn, weil hier nicht 12 reguläre Datenbits, sondern nur 8 zur Verfügung stehen.

Tabelle 3 zeigt das sich ergebende Signalspiel bei der Parallelport-Ansteuerung der Ampel.

Takt	Grün Ost+ West	Gelb Ost+ West	Rot Ost+ West	Grün Nord+ Süd	Gelb Nord+ Süd	Rot Nord+ Süd	Zeit [Sek]	OUT (dez.)
	D5	D4	D3	D2	D1	D0		
1			Red			Red	3	9
2			Red		Yellow	Red	3	11
3			Red	Green			20	12
4			Red		Yellow		3	10
5			Red			Red	3	9
6		Yellow	Red			Red	3	25
7	Green					Red	20	33
8		Yellow				Red	3	17
1(W)			Red			Red	3	9
2(W)			Red		Yellow	Red	3	11
3(W)			Red	Green			20	12
4(W)			Red		Yellow		3	10
5(W)			Red			Red	3	9
6(W)		Yellow	Red			Red	3	25
7(W)	Green					Red	20	33
8(W)		Yellow				Red	3	17

*Tabelle 3 Signalspiel der Ampel mit Parallelport-Ansteuerung (grau markiert = Wiederholung)*

Für die gezeigte Ampelkonfiguration gibt es natürlich auch ein PC-Programm AMPEL01.ZIP, das wieder als Testpoint-Runtime-Version mitgeliefert wird. Weitere Informationen zu diesen Testpoint-Runtime-Versionen finden Sie im Applikationsbericht 031 „Applikationen mit Testpoint“. Entpacken Sie die angehängte ZIP-Datei und führen Sie „SETUP.EXE“ aus. Das Programm AMPEL01.TST, das sich dann unter Start → Programme

---

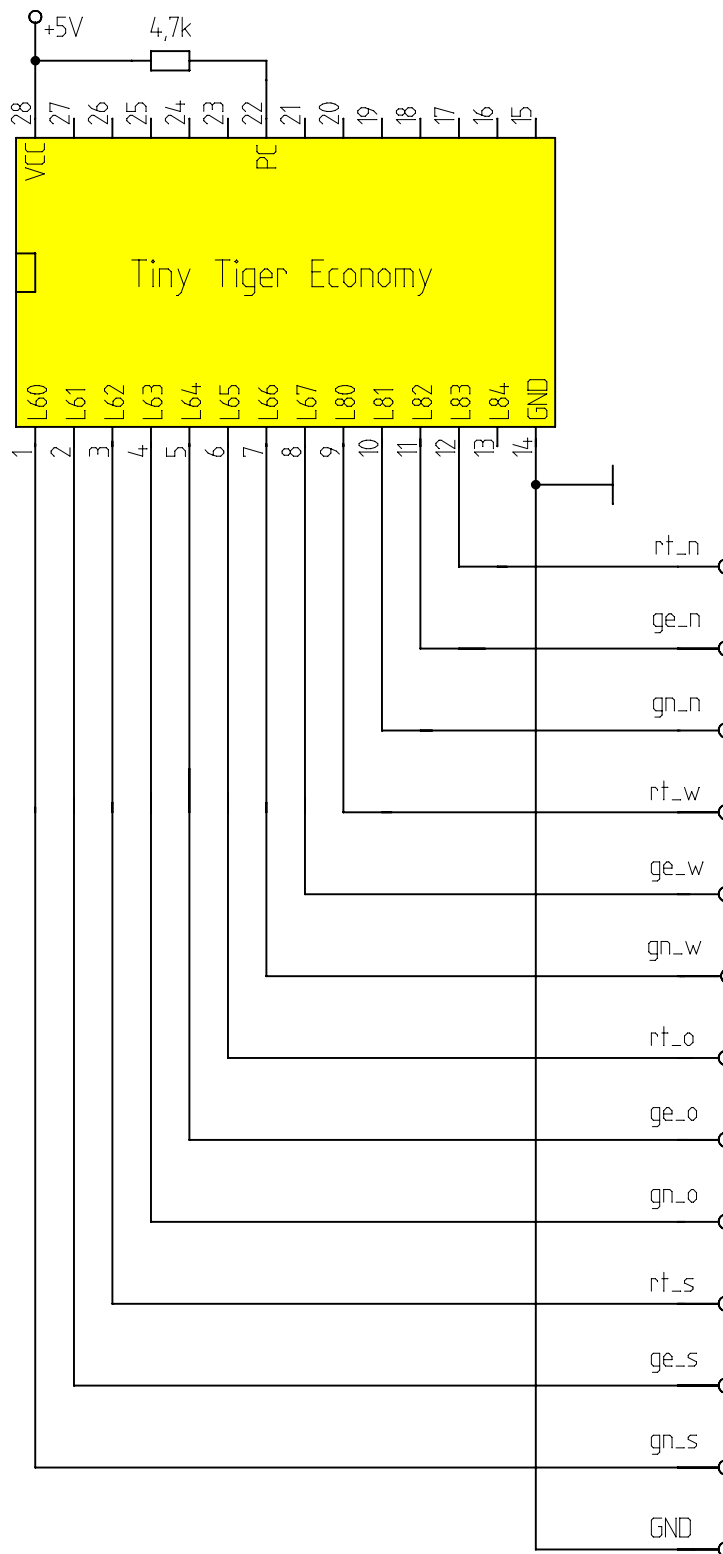
auf Ihrem Windows PC befindet, steuert eine LED-Modellampel wie beschrieben und zeigt diese Ampel auch gleichzeitig auf dem Computer-Bildschirm. Sie können also auch ohne Strippenziehen und Löten sofort losspielen...

### **3. Die zweite Variante – Steuerung über einen Tiny-Tiger E**

Eigentlich ist ein solch leistungsfähiger Steuer-Computer viel zu schade für eine simple Ampelsteuerung. Die Mitglieder der Tiger-Gemeinde dürften aber in der Regel mal einen Econo-Tiger™ der einfachsten Ausführung übrig haben. Und vielleicht kann der Tiger in der Modellbahnanlage noch weitere Aufgaben übernehmen...

Die Anbindung an die bestehende Ampelanlage ist einfach. Wir werden die 12 Steuerleitungen direkt an vorhandene Portleitungen anschließen (L60...L67, L80...L83). Der Rest ist ein simples Programm (AMPEL01.TIG), mit dem das Signalspiel über OUT-Befehle zeitgesteuert realisiert wird. Eine Zusammenfassung und Erläuterung der Signalspiele finden Sie in Tabelle 4. Die grau markierten Phasen (Zeilen) sind bereits Wiederholungen der vorher stehenden Zeilen. Wer möchte, kann die Leucht- und Dunkelzeiten der Ampel nach seinen Wünschen im Programm einfach verändern.

Der Tiny-Tiger E muss natürlich mit einer stabilisierten Betriebsspannung von +5V versorgt werden. Dem Anwender steht es frei, dies mit einem geeigneten Netzteil oder mit einer Batterielösung gemäß Applikationsbericht Nr. 081 zu tun.



*Bild 5 So schließen wir den Economy-Tiger an das Ampelmodell an*



Takt	Rot Nord	Gelb Nord	Grün Nord	Rot West	Gelb West	Grün West	Rot Ost	Gelb Ost	Grün Ost	Rot Süd	Gelb Süd	Grün Süd	Zeit [Sek.]	Port 8 (dez.)	Port 6 (dez.)
	L83	L82	L81	L80	L67	L66	L65	L64	L63	L62	L61	L60		OUT	OUT
1	Red			Red			Red			Red			3	9	36
2	Red	Yellow		Red			Red			Red	Yellow		3	13	38
3			Green	Red			Red					Green	20	3	33
4		Yellow		Red			Red				Yellow		3	5	34
5	Red			Red			Red			Red			3	9	36
6	Red			Red	Yellow		Red	Yellow		Red			3	9	180
7	Red					Green			Green				20	8	76
8	Red				Yellow			Yellow					3	8	148
1(w)	Red	Gray	Gray	Red	Gray	Gray	Red	Gray	Gray	Red	Gray	Gray	3	9	36
2(w)	Red	Yellow	Gray	Red	Gray	Gray	Red	Gray	Gray	Red	Yellow	Gray	3	13	38
3(w)	Gray	Gray	Green	Red	Gray	Gray	Red	Gray	Gray	Gray	Gray	Green	20	3	33
4(w)	Gray	Yellow	Gray	Red	Gray	Gray	Red	Gray	Gray	Gray	Yellow	Gray	3	5	34
5(w)	Red	Gray	Gray	Red	Gray	Gray	Red	Gray	Gray	Red	Gray	Gray	3	9	36
6(w)	Red	Gray	Gray	Red	Yellow	Gray	Red	Yellow	Gray	Red	Gray	Gray	3	9	180
7(w)	Red	Gray	Gray	Red	Gray	Green	Red	Gray	Green	Red	Gray	Gray	20	8	76
8(w)	Red	Gray	Gray	Red	Yellow	Gray	Red	Yellow	Gray	Red	Gray	Gray	3	8	148

Tabelle 4 Signalspiel bei der Tiger-Version (grau markiert = Wiederholung)

#### 4. Die dritte Variante – reine Hardware

Die beiden bisher vorgestellten Steuervarianten haben jeweils bestimmte Nachteile. Die PC-Variante braucht eben immer einen PC – im Kinderzimmer kann das schwierig werden. Die Variante mit dem Tiger erfordert eine stabile 5V – Versorgung, entweder aus einem Netzteil oder aus einer nicht ganz einfach zu realisierenden Batterie-Stromversorgung. Die folgende Variante ist da anspruchsloser. Sie wird mit einfachen CMOS-Gatterschaltungen der 4000-er Serie aufgebaut, die von etwa 3V bis 18V arbeiten und dabei bis auf die LED's kaum Strom brauchen. Eigentlich ideal für Spielzeuganwendungen mit Batterie oder die Modellbahn. Ein Nachteil bei dieser Variante – der Hardware-Aufwand ist erheblich größer. Dennoch sind es nur drei Gatter-Schaltkreise, ein paar Transistoren, Dioden und Widerstände. Auch Kompromisse bezüglich der Zeitabläufe müssen wir eingehen. Bei den beiden Software-Varianten (PC und BASIC-Tiger) sind wir völlig frei in der Gestaltung der Leucht- und Pausenzeiten der einzelnen LED's. Eine reine Gatterschaltung ist da leider nicht so flexibel.

Fangen wir bei dem Ablaufschema einer solchen Ampel an. Wir stellen uns 10 Takte vor, in denen das Ampelspiel ablaufen soll. Im Gegensatz zur den beiden Vorgängervarianten, bei denen wir jede Phase beliebig lang und unabhängig von den anderen Phasen gestalten konnten, sind hier die Takte alle gleich lang. Längere Leuchtphasen müssen also durch mehrere gleichartige Takte realisiert werden. Eine Bedingung ist wieder, dass die LED's der Nord- und Süd-Ampeln sowie die der Ost- und Westampeln jeweils das gleiche Signalspiel haben. Dann kann eine Ampelsteuerung mit 10 gleichen Takten z.B. so aussehen:

Takt	Rot Nord/ Süd	Gelb Nord/ Süd	Grün Nord/ Süd	Rot Ost/ West	Gelb Ost/ West	Grün Ost/ West
0	Rot					Grün
1	Rot				Gelb	
2	Rot	Gelb		Rot		
3			Grün	Rot		
4			Grün	Rot		
5			Grün	Rot		
6		Gelb		Rot		
7	Rot			Rot	Gelb	
8	Rot					Grün
9	Rot					Grün

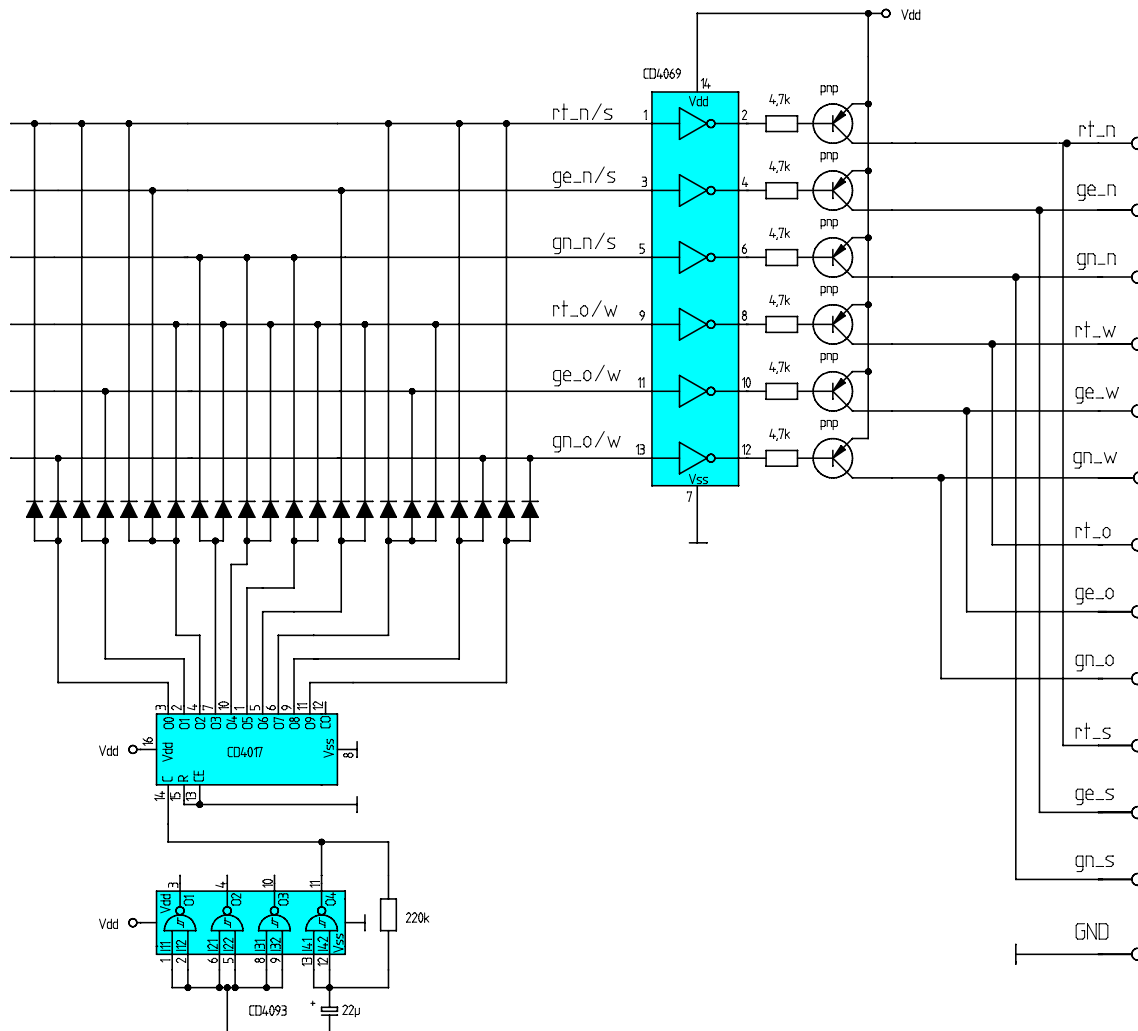
*Tabelle 5 Ablaufschema einer einfachen Ampelsteuerung mit 10 Takten*

Zur Realisierung brauchen wir zunächst einen einfachen CMOS-Taktgenerator für einen sehr langsamen Takt. Ausreichend gut geeignet ist ein RC-Multivibrator mit einem CD4093 (Nand-Gatter mit Schmitt-Trigger-Eingängen). Damit die Taktzeit in der Größenordnung Sekunden liegt, sind hohe Kapazitäten erforderlich. Die Folge sind dann extrem langsam ansteigende Eingangssignale. Die Schmitt-Trigger-Eingänge sorgen trotzdem für ein schlagartiges Umschalten der Ausgänge auch bei sehr langsamen Eingangsfanken.

Als nächstes Glied wird ein Zähler gebraucht, der mindestens bis 10 zählen kann. Der CD4017 ist ein dekadischer Zähler mit 10 dekodierten Ausgängen, d.h. jeder Zählerzustand aktiviert einen Ausgang (Ausgang wird High). Andere Zähler wären prinzipiell auch geeignet, der Dekodieraufwand wäre aber höher.

Jetzt müssen den zehn Takten die in der Tabelle 3 dargestellten Ampelsignale zugeordnet werden. Das machen wir mit einer simplen Dioden-Logik (kleine Si-Dioden). Wenn also Takt 0 aktiv ist, müssen über zwei Dioden die LED-Treiber für die LED's Rot\_Nord/Süd und Grün\_Ost/West angesteuert werden. Weil wir uns bei der Ampelverdrahtung in der Variante PC-Parallelport bereits festgelegt haben, behalten wir die Zuordnungen und Pegel zum Betrieb der LED's auch hier bei. Das hat zur Folge, dass die Treiberschaltung für die jeweils beiden zusammengehörigen LED's (z.B Rot\_Nord und Rot\_Süd) noch einen Inverter CD4069 sowie eine pnp-Transistorstufe benötigen.

Bild 6 zeigt nun die komplette Schaltung:



*Bild 6 Schaltung der reinen Hardware-Version*

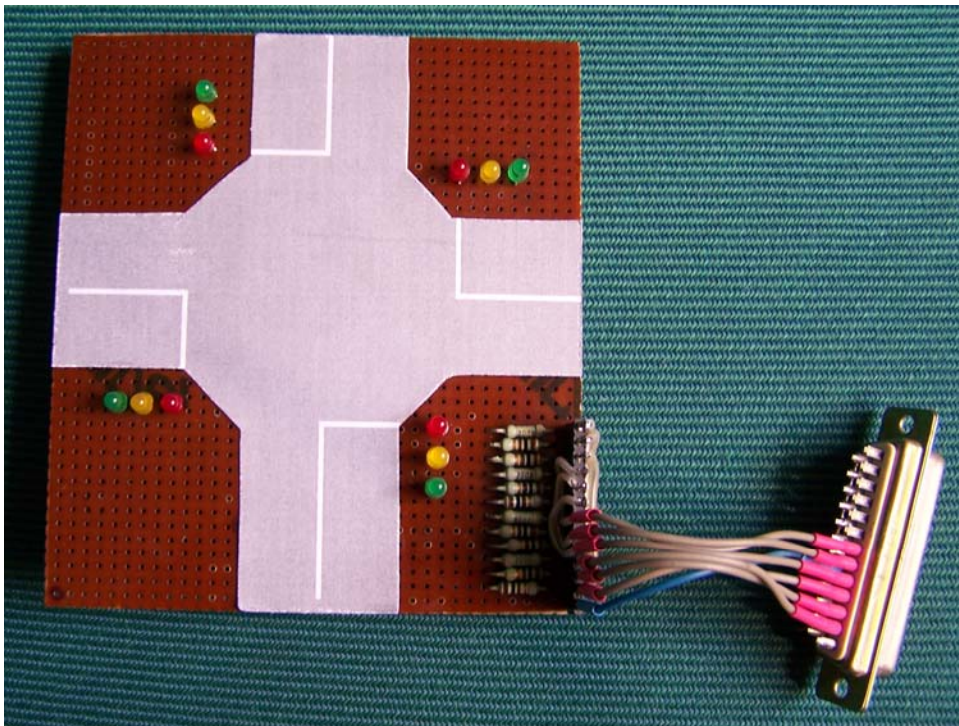
Die Betriebsspannung Vdd für diese Steuerung mit CMOS-Gattern kann zwischen etwa 4V und 15V (Gleichspannung) liegen und muss nicht besonders stabilisiert sein. Zu beachten ist bei dem möglichen großen Betriebsspannungsbereich, dass die Vorwiderstände für die 12 Ampel-LED's an die gewählte Spannung angepasst werden müssen.

## 5. Fazit

Ein jeder kann nun entscheiden, welche Variante zu seiner Ampel am besten passt:

- Die PC-Steuerung kommt mit einem 7-poligen Kabel an der Parallelschnittstelle aus und benötigt keine zusätzliche Stromversorgung, ist aber immer auf einen vorhandenen und betriebenen PC angewiesen.

- Die Steuerung mit dem kleinen Tiny-Tiger E ist autark, kann eigenen Bedürfnissen besser angepasst werden, fällt aber auch teurer aus und benötigt eine etwas aufwändige Stromversorgung.
- Die reine Hardware-Steuerung benötigt einigen Aufwand beim Verdrahten und Löten, eventuell eine spezielle Platine und eine ganze Menge Einzelbauteile. Sie hat aber den Vorteil, dass sie mit jeder Batterie im Bereich 4...15V problemlos und stromsparend arbeitet.



*Bild 7 ein Laboraufbau für die Ampelkreuzung – hier mit Ansteuerung durch Parallelport*

In jedem Fall ist eine wirklichkeitsnah funktionierende Ampel für jede Modellbahn oder für andere Anlagen ein echtes Highlight. Auch das spielerische Heranführen unserer Kinder an den Straßenverkehr dürfte mit einer Ampelkreuzung im Kinderzimmer wesentlich interessanter sein als die übliche Theorie. Dieser Applikationsbericht soll auch ein Denkanstoß für eigene kreative Ideen sein, z.B. kann die Anlage noch um Fußgängerampeln oder LED's für Abbiegespuren erweitert werden – interessant für die Nachbildung echter Kreuzungen in Städtmodellen usw.

Viel Spaß beim Spielen...