

Experiment 22: Einer wird gewinnen

Das nächste Projekt mit Digitallogik führt uns tiefer in das Prinzip der Rückkopplung ein. Hierbei wird das Ausgangssignal zurück an den Eingang geführt – um ihn in diesem Fall zu blockieren. Es ist ein kleines Projekt, aber ziemlich raffiniert, und die Ideen dahinter werden dir für später von Nutzen sein.

Was du brauchst

- Steckboard, Schaltdraht, Seitenschneider, Abisolierzange, Multimeter
- 9-Volt-Stromversorgung (Batterie oder Netzteil)
- 74HC32-Logikchip (1)
- 555-Timer (2)
- Umschalter, einpolig als Schiebeschalter (2)
- Taster (2)
- Widerstände: 220 Ω (1), 2,2 k Ω (1), 10 k Ω (3)
- Kondensatoren: 0,01 μ F (2), 0,1 μ F (1), 0,33 μ F (1)
- LM7805-Spannungsregler (1)
- Standard-LEDs (2)
- Niedrigstrom-LED (1)

Das Ziel

Bei Quizsendungen wie *Jeopardy!* kommt es darauf an, welcher Teilnehmer am schnellsten antwortet. Wer zuerst seinen Antwortknopf drückt, schließt automatisch die anderen Teilnehmer aus, sodass ihre Knöpfe unwirksam werden. Wie können wir eine Schaltung bauen, die genau das macht?

Wenn du online suchst, findest du einige Elektronik-Websites, auf denen andere Leute Schaltungen vorgeschlagen haben, die auf diese Weise funktionieren, aber dort fehlen immer einige Funktionen, die meiner Meinung nach nötig sind. Mein Ansatz hier ist sowohl einfacher als auch raffinierter. Er ist einfacher, weil nur sehr wenige Chips gebraucht werden, andererseits aber auch raffinierter, weil er eine »Quizmaster-Steuerung« hat, die ein realistischeres Spiel ermöglicht.

Ich schlage erst einige Ideen für eine Version mit zwei Spielern vor. Sobald ich dieses Konzept fertig entwickelt habe, zeige ich noch, wie man es auf vier oder noch mehr Spieler erweitern könnte.

Ein Konzeptexperiment

Ich will zeigen, wie so ein Projekt von einer Idee zur fertigen Version heranreift. Wenn ich die Schritte für die Entwicklung einer Schaltung durchgehe, hoffe ich, dass ich dich dazu inspiriere, in Zukunft deine eigenen Ideen zu entwickeln. Dies ist viel mehr wert, als nur das nachzubauen, was jemand anderes entwickelt hat.

Schauen wir uns erst einmal das Grundkonzept an: Zwei Teilnehmer haben zwei Taster, und derjenige, der zuerst drückt, schließt den anderen aus.

Manchmal hilft es mir, ein solches Vorhaben anhand einer Skizze zu veranschaulichen. Also fange ich damit an. In Abbildung 4-112 geht das Signal von jedem Taster durch eine gedachte Komponente, die ich »Tastersperre« nenne und die vom Taster der anderen Person aktiviert wird. Ich bin noch nicht ganz sicher, was die Tastersperre sein oder wie sie funktionieren wird, jedenfalls wird sie vom ersten Spieler aktiviert und sperrt den anderen Spieler.

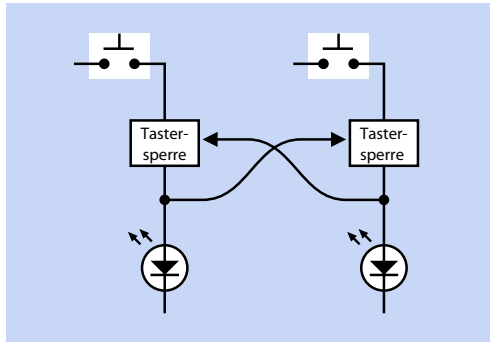


Abbildung 4-112.

Das Grundkonzept: Wer zuerst »am Drücker ist«, blockiert den anderen Spieler.

Wenn ich mir das so ansehe, fällt mir gleich ein Problem auf. Will ich das Quiz auf drei Teilnehmer erweitern, wird es kompliziert, weil jeder Spieler die »Tastersperre« von zwei Gegnern aktivieren muss. Und bei vier Spielern muss jeder Spieler die »Tastersperre« von drei Gegnern aktivieren. Die Anzahl der Verbindungen ist nicht mehr handhabbar. Abbildung 4-113 zeigt dies.

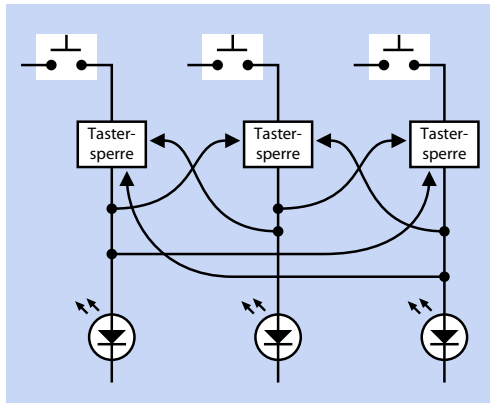


Abbildung 4-113.

Erhöht man die Anzahl der Teilnehmer von zwei auf drei, ist die doppelte Anzahl von Verbindungen erforderlich.

Angesichts der Komplexität sollten wir nach einem besseren Ansatz suchen.

Es gibt noch ein Problem. Nachdem ein Spieler den Taster losgelassen hat, werden die Taster der anderen Spieler wieder freigeschaltet. Wie in den Experimenten 15, 19 und 21 brauche ich also eine Verriegelung – ein Flipflop,

im Englischen auch als »Latch« bezeichnet. Es hält das Signal vom Taster des ersten Spielers und blockiert die anderen Spieler weiterhin, selbst nachdem der erste Spieler seinen Taster losgelassen hat.

Das klingt jetzt noch komplizierter. Aber Moment mal. Wenn der Taster des gewinnenden Spielers ein Flipflop triggert, hält das Flipflop jetzt die gewinnende Schaltung unter Strom, und der Taster des Gewinners ist nicht mehr relevant. Das Flipflop kann also *alle* Taster blockieren. Das vereinfacht die Sache stark. Ich kann das als Abfolge von Ereignissen zusammenfassen:

1. Der erste Spieler drückt seinen Taster.
2. Sein Signal wird verriegelt.
3. Das verriegelte Signal wird zurückgekoppelt und blockiert alle Taster.

Die neue Skizze in Abbildung 4-114 zeigt dies. Jetzt ist der Aufbau modular und lässt sich auf nahezu jede Anzahl von Spielern erweitern, indem man einfach weitere Module hinzufügt, ohne die Komplexität zu erhöhen.

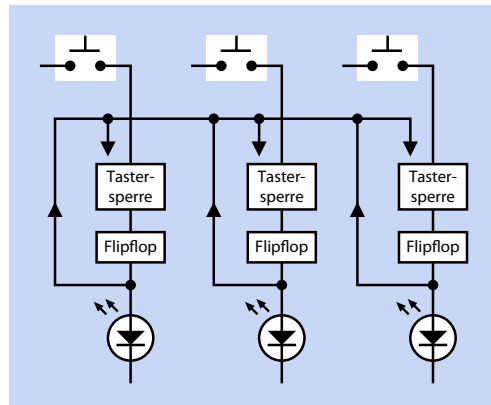


Abbildung 4-114.
Jedes Flipflop blockiert nun alle Taster.

Es fehlt aber noch etwas Wichtiges: ein Rückstellschalter, der das System wieder in den Ausgangszustand versetzt, nachdem die Spieler Zeit hatten, ihre Taster zu drücken und zu sehen, wer gewonnen hat. Außerdem brauche ich eine Möglichkeit, die Spieler davon abzuhalten, ihre Taster zu früh zu drücken, bevor der Quizmaster die Frage fertig vorgelesen hat. Vielleicht kann ich diese Funktion mit nur einem Schalter umsetzen, den nur der Quizmaster bedienen kann.

Abbildung 4-115 zeigt diese Version. In der Ruhestellung kann der Schalter des Quizmasters das System zurücksetzen und den Strom von den Tastern abschalten. In der Spielstellung hält der Schalter das System nicht mehr im Rücksetzmodus und schaltet den Strom an die Taster. Ich gehe nun wieder zurück zur Version mit nur zwei Spielern, um alles möglichst einfach zu machen, doch das Konzept ist immer noch leicht erweiterbar.

Jetzt muss ich mich mit einem Logikproblem im Schaltbild auseinandersetzen. So wie ich es gezeichnet habe, ist alles miteinander verbunden. Die Pfeile zeigen zwar die Richtung der Signale an, doch ich weiß nicht, wie ich die Signale eigentlich daran hindern könnte, die falsche Richtung einzuschlagen. Wenn ich keine entsprechenden Maßnahmen treffe, lässt das Signal von einem beliebigen Spieler beide LEDs leuchten. Wie kann ich das verhindern?

Ich könnte mit Dioden in den nach »oben« gehenden Drähten verhindern, dass der Strom nach unten fließt. Doch ich habe eine elegantere Lösung: Ich füge ein OR-Gatter ein, weil die Eingänge eines OR-Gatters elektrisch voneinander getrennt sind. Abbildung 4-116 zeigt dies.

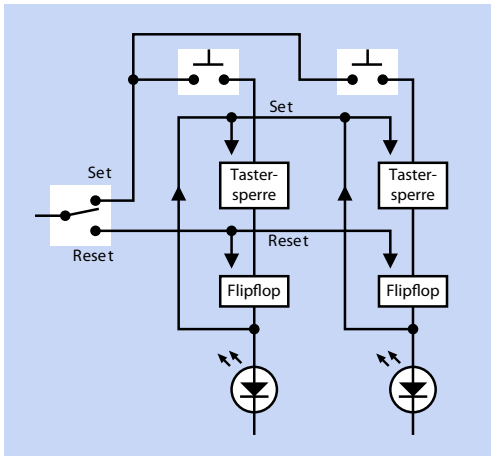


Abbildung 4-115.
Die Quizmaster-Steuerung wurde nun hinzugefügt.

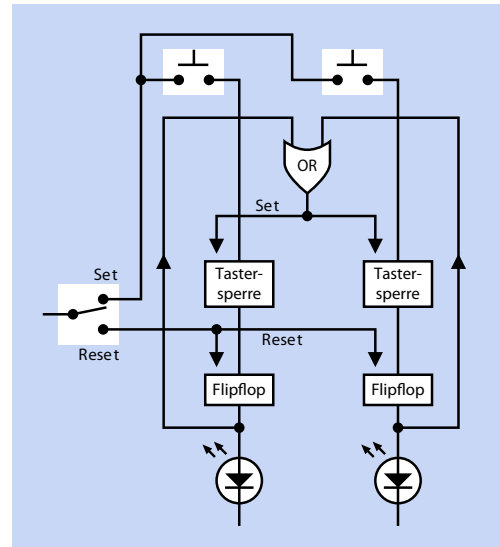


Abbildung 4-116.
Ein OR-Gatter isoliert die Schaltung des einen Spielers von der des anderen.

Das einfache OR-Gatter besitzt nur zwei logische Eingänge. Hält mich das davon ab, mehr Spieler hinzuzufügen? Nein, weil es auch OR-Gatter mit drei, vier oder acht Eingängen gibt. Wenn einer von ihnen auf High-Pegel liegt, ist der Ausgang High. Bei weniger als acht Spielern kannst du die nicht verwendeten Eingänge auf Masse legen und ignorieren.

Nun bekomme ich eine klarere Vorstellung, wie die von mir so genannte »Taster Sperre« praktisch aussehen soll. Ich glaube, ich sollte noch ein Logikgatter verwenden. Das Prinzip bei diesem Gatter sollte folgendes sein: »Wenn es nur ein Eingangssignal von einem Taster gibt, lasse ich es durch. Wenn es aber weitere Eingangsimpulse gibt, lasse ich sie nicht durch.«

Bevor ich aber die Gatter auswähle, muss ich entscheiden, wie das Flipflop aussehen soll. Ich kann ein handelsübliches Flipflop nehmen, das einschaltet, wenn es ein Signal erhält, und das ausschaltet, wenn das nächste Signal kommt. Doch Flipflop-Chips bieten noch mehr Features, als ich für eine derartig einfache Schaltung brauche. Deshalb benutze ich wieder 555-Timer im bistabilen Modus. Sie brauchen nur sehr wenige Verbindungen, funktionieren

sehr einfach und liefern ausreichend Strom, um helle LEDs anzusteuern. Das einzige Problem dabei ist, dass 555-Timer im bistabilen Modus Folgendes benötigen:

- Einen High/Low-Übergang am Eingang, um einen High-Ausgang zu liefern
- Ein Low am Reset-Eingang, um einen Low-Ausgang zu liefern

Nun gut, dann muss der Taster jedes Spielers einen High/Low-Impuls anstelle eines Low/High-Impulses liefern. Damit wären die Anforderungen der Timer erfüllt.

In Abbildung 4-117 ist nun endlich der vereinfachte Schaltplan zu sehen. Ich wollte die Pins der 555-Timer in ihrer tatsächlichen Anordnung zeigen, sodass ich die Bauelemente etwas anders platzieren musste, um die Anzahl der Leitungskreuzungen gering zu halten. Doch logisch gesehen ist es genau dasselbe Grundkonzept.

Der Platz hat nicht gereicht, um mit Plus- und Minussymbolen zu zeigen, welche Timer-Pins in jedem Zustand gehalten werden, sodass ein roter Kreis bedeutet, dass das Pin auf High-Pegel liegt, und ein blauer Kreis für einen Low-Pegel steht. Schwarze Kreise bedeuten, dass sich der Zustand des Pins ändern kann. Weiße Kreise zeigen an, dass die Zustände dieser Pins nicht wichtig sind und die Pins unbeschaltet bleiben können.

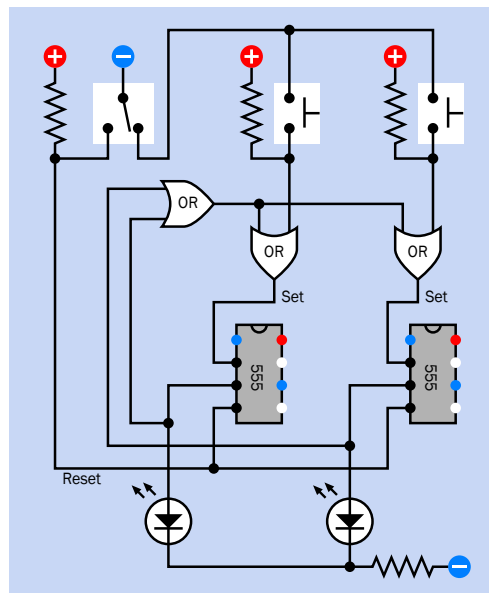


Abbildung 4-117.

Ein vorläufiges Logikdiagramm. Blaue Pins an den Timern werden auf Low gehalten, rote Pins auf High und weiße Pins sind nicht von Bedeutung.

Bevor du jetzt versuchst, es nachzubauen, gehe erst noch einmal alles theoretisch durch. Das ist der letzte Schritt, um sicherzugehen, dass es keine Fehler gibt. Denke dabei daran, dass der 555-Timer ein *negatives* Signal am Trigger-

Eingang benötigt, um den Ausgang auf *High* zu schalten. Wenn also irgend ein Spieler einen Taster drückt, muss der Taster einen negativen »Stromfluss« durch die Schaltung erzeugen.

Da dies ein wenig der Intuition widerspricht, habe ich in Abbildung 4-118, Abbildung 4-119, Abbildung 4-120 und Abbildung 4-121 in vier Schritten veranschaulicht, wie es funktioniert.

In Schritt 1 befindet sich der Quizmaster-Schalter im Reset-Modus. Der Low-Pegel an den Reset-Eingängen der Timer zwingt deren Ausgänge auf Low und die LEDs bleiben dunkel. Außerdem führen die Timer-Ausgänge zu den Eingängen des OR1-Gatters. Da beide Eingänge Low sind, hat auch der Ausgang Low-Pegel. Dieser liegt an OR2 und OR3, bleibt aber wirkungslos, weil jeweils ein Eingang dieser OR-Gatter auf High liegt, was durch die Pullup-Widerstände parallel zu den Tastern sichergestellt ist. Denn wenn mindestens ein Eingang eines OR-Gatters auf High liegt, schaltet der Ausgang auf High. Und solange der Trigger-Eingang eines Timers im bistabilen Modus High bleibt, wird der Timer nicht ausgelöst. Die Schaltung ist also stabil.

In Schritt 2 hat der Quizmaster eine Frage gestellt und seinen Schalter nach rechts umgelegt, um die Taster der Spieler mit der (negativen) Betriebsspannung zu verbinden. Allerdings hat noch kein Spieler geantwortet, sodass die Pullup-Widerstände die Schaltung in einem stabilen Zustand halten mit Low an den Ausgängen der Timer.

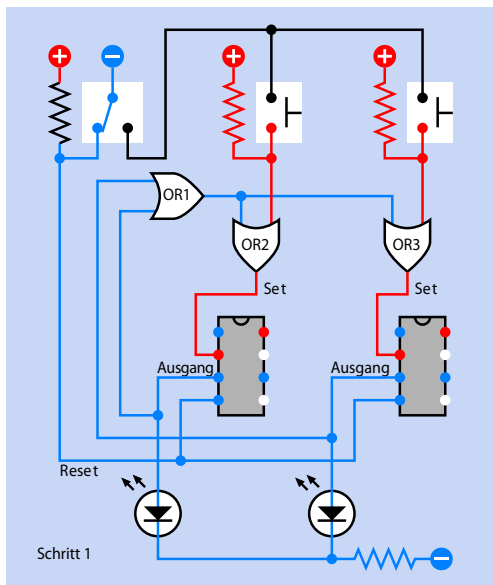


Abbildung 4-118.
Schritt 1 der Schaltungsvisualisierung: Reset-Modus

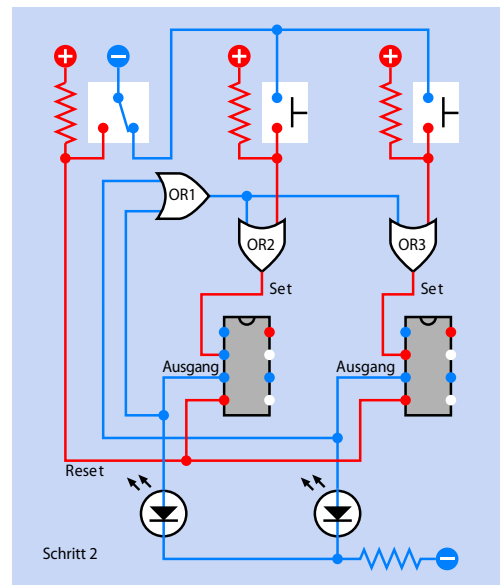


Abbildung 4-119.
Schritt 2 der Schaltungsvisualisierung: Die Taster der Spieler sind aktiv, aber noch hat keiner der Spieler einen Taster gedrückt.

In Schritt 3 hat Spieler 1 den linken Taster gedrückt. Dadurch gelangt Low-Pegel an OR2, und da nun an beiden Eingängen von OR2 Low-Pegel liegt, geht der Ausgang dieses Gatters auf Low. Dieser High/Low-Übergang löst den linken Trigger aus. Doch Bauelemente reagieren nicht gänzlich verzögerungsfrei, und der Timer hat das Signal noch nicht verarbeitet.

In Schritt 4, nur einige Mikrosekunden später, hat der Timer das High/Low-Eingangssignal verarbeitet und den Ausgang auf High geschaltet, sodass die LED leuchtet. Der auf High liegende Ausgang wird außerdem zum Eingang von OR1 zurückgeführt. Da nun ein Eingang von OR1 auf High liegt, schaltet der Ausgang des Gatters auf High. Dieser Pegel gelangt zu den Eingängen von OR2 und OR3, sodass deren Ausgänge auf High schalten. Im Ergebnis liegen nun die Trigger-Eingänge beider Timer auf High-Pegel. Jede Tasterbetätigung durch irgendeinen Spieler bleibt nun wirkungslos, weil OR1 fortwährend High-Pegel liefert.

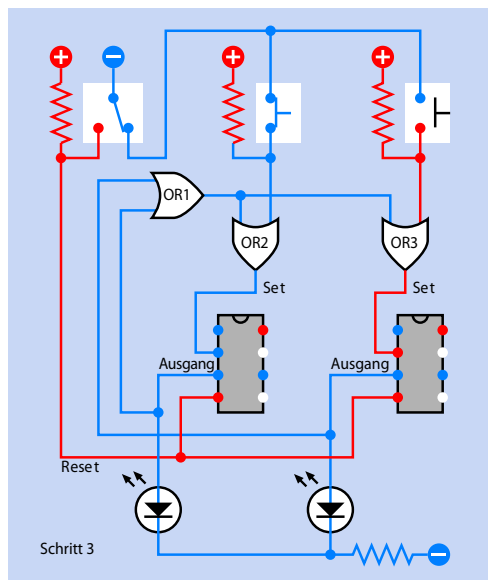


Abbildung 4-120.
Schritt 3 der Schaltungsvisualisierung: Der linke Spieler hat einen Taster gedrückt, doch der 555-Timer hat noch nicht reagiert.

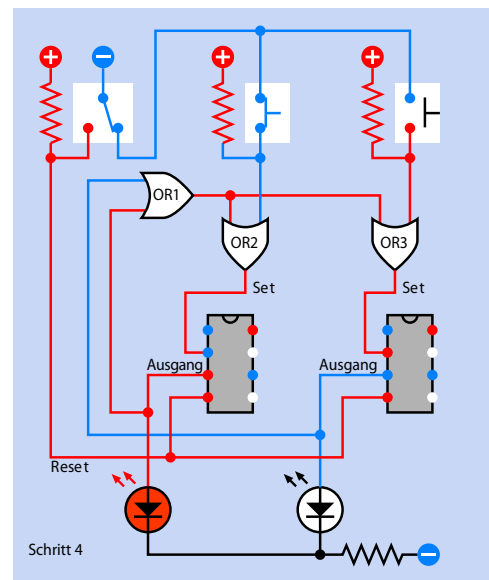


Abbildung 4-121.
Schritt 4 der Schaltungsvisualisierung: Die Aktion des linken Spielers hat sich ihren Weg durch die Schaltung gebahnt und blockiert nun den rechten Spieler.

- Wenn ein 555-Timer im Flipflop-Modus arbeitet, kippt ein Low-Pegel am Trigger-Eingang den Ausgang auf High und der Ausgang bleibt auf High, selbst wenn der Trigger-Eingang wieder auf High geht.
- Der Ausgang des 555-Timers lässt sich nur dann wieder auf Low zurücksetzen, wenn der Reset-Eingang auf Low gezogen wird. Das passiert nur, wenn der Spielmeister seinen Schalter in den Reset-Modus umlegt.

Es gibt allerdings eine Situation, die dieses schöne Szenario trüben kann. Was passiert, wenn beide Spieler ihre Taster absolut gleichzeitig drücken? In der Welt der digitalen Elektronik ist dies allerdings höchst unwahrscheinlich. Doch falls es dennoch irgendwie passiert, sollten beide Timer reagieren und beide LEDs leuchten auf, was ein Unentschieden anzeigt.

In der TV-Show Jeopardy! ist nie ein Unentschieden zu sehen. Absolut nie! Ich frage mich, ob das Elektroniksystem der Show ein gleichzeitiges Antworten der beiden Spieler registriert. Vielleicht ist auch eine Zufallskomponente eingebaut, die einen der beiden Spieler auswählt. Natürlich ist das nur eine Spekulation.

Um zu demonstrieren, wie sich eine Zwei-Spieler-Schaltung für mehrere Spieler erweitern lässt, habe ich in Abbildung 4-122 einen vereinfachten Schaltplan für eine Drei-Spieler-Schaltung angegeben. Die Schaltung ließe sich nach oben hin beliebig erweitern, nur begrenzt durch die Anzahl der verfügbaren Eingänge am Gatter OR1.

Der Aufbau auf dem Steckboard

In Abbildung 4-123 habe ich den Schaltplan mit einem realen OR-Chip neu gezeichnet, wobei das Layout der Anordnung auf dem Steckboard möglichst nahe kommt, damit du diese Schaltung leicht aufbauen kannst. Abbildung 4-124 zeigt eine Steckboard-Version und die Werte der Bauelemente sind in Abbildung 4-125 angegeben.

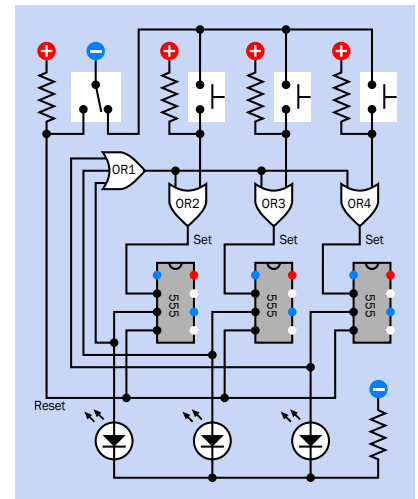


Abbildung 4-122.
Die Schaltung lässt sich leicht für mehr Spieler erweitern.

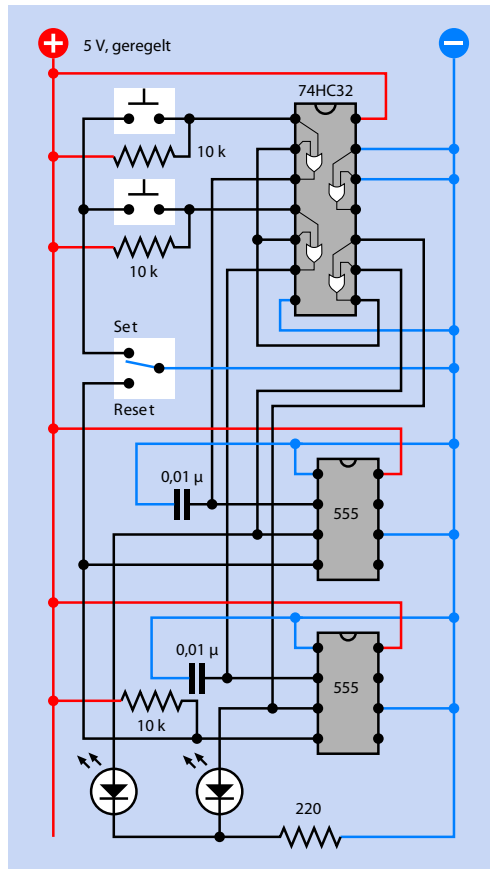


Abbildung 4-123.
Der Schaltplan für die Zwei-Spieler-Version wurde hier neu gezeichnet und verwendet einen Chip, der vier OR-Gatter mit je zwei Eingängen enthält.

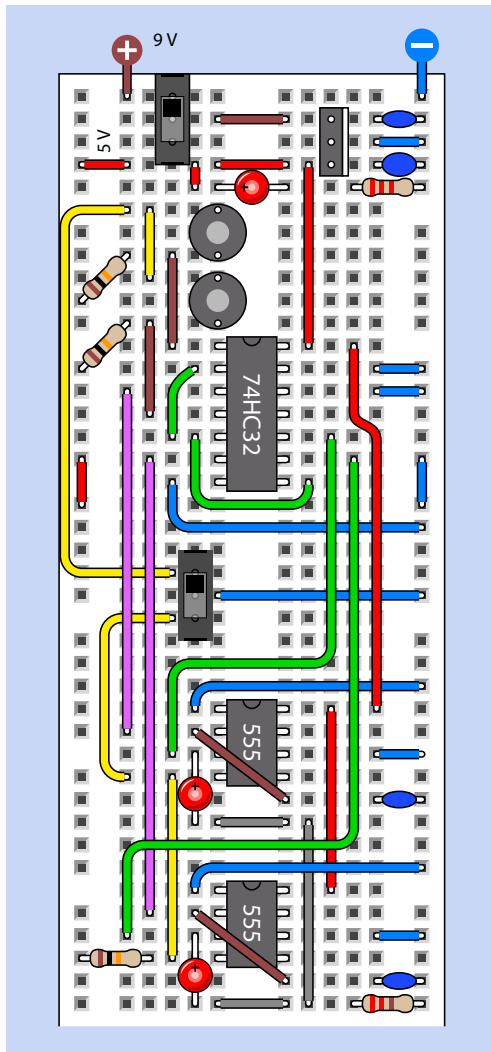


Abbildung 4-124.
Das Steckboard-Layout, das dem Schaltplan entspricht

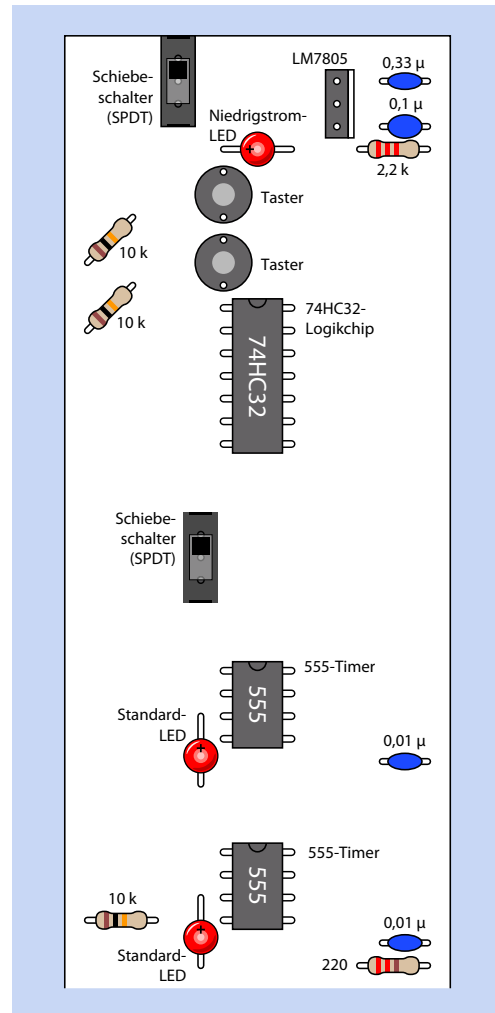


Abbildung 4-125.
Bauelementewerte für den Steckboard-Aufbau

Da ich ausschließlich OR-Gatter als Logikgatter verwendet habe und nur drei Gatter erforderlich sind, genügt ein einziger Logikchip: der 74HC32, der vier OR-Gatter mit je zwei Eingängen enthält. (Die Eingänge des vierten Gatters habe ich auf Masse gelegt.) Die beiden OR-Gatter auf der linken Chip-Seite übernehmen dieselben Funktionen wie OR2 und OR3 in meinem vereinfachten Schaltplan, und das OR-Gatter rechts unten fungiert als OR1, an dessen Eingängen die Ausgänge (Pin 3) der beiden 555-Timer liegen. Wenn du alle Bauelemente beisammen hast, solltest du diese Schaltung recht schnell aufbauen und testen können.

Vielleicht ist dir aufgefallen, dass ich zwischen Pin 2 jedes 555-Timers (dem Eingang) und Masse einen 0,01- μ F-Kondensator hinzugefügt habe. Warum? Als ich die Schaltung ohne die Kondensatoren getestet hatte, wurden manchmal einer oder beide 555-Timer allein durch das Umlegen des Quizmaster-Schalters ausgelöst, ohne dass jemand einen Taster gedrückt hatte.

Das hat mich erst einmal verwirrt. Wie wurden die Timer getriggert, ohne dass irgendjemand etwas gemacht hat? Vielleicht reagierten sie auf ein »Prellen« des Quizmaster-Schalters – d.h. auf winzige und sehr schnelle Schwingungen in den Kontakten, wenn der Schalter umgelegt wird. Jedenfalls hat sich das Problem mit den kleinen Kondensatoren lösen lassen. Zwar können sie auch die Reaktionszeit des 555-Timers marginal verlängern, doch nicht so sehr, dass sie den langsameren menschlichen Reflexen in die Quere kommen.

Was die Taster angeht, ist es egal, ob sie »prellen«, weil sich jeder Timer bereits beim allerersten Impuls selbst verriegelt und alle folgenden Unterbrechungen ignoriert.

Wenn du die Schaltung aufbaust, kannst du damit experimentieren, die 0,01- μ F-Kondensatoren wegzulassen, und den Quizmaster-Schalter Dutzende Male hin- und herschalten. Möglicherweise siehst du einige »Falschmeldungen«. Mehr zum Schalterprellen und wie man es unterdrückt, erkläre ich im nächsten Experiment.

Erweiterungen

Wenn du die Schaltung auf dem Steckboard aufgebaut hast und eine dauerhafte Version herstellen willst, solltest du die Schaltung gleich so erweitern, dass mindestens vier Spieler teilnehmen können. Hierfür ist ein OR-Gatter mit vier Eingängen erforderlich. Die naheliegende Wahl ist der 74HC4078, der acht Eingänge besitzt. Verbinde einfach die nicht benutzten Eingänge mit Masse.

Falls du schon einige 74HC32-Chips hast und nicht extra einen 74HC4078 besorgen willst, kannst du auch drei Gatter in einem einzelnen 74HC32 so verknüpfen, dass sie wie ein OR mit vier Eingängen funktionieren. Sieh dir dazu das einfache Logikdiagramm in Abbildung 4-126 an, das drei OR-Gatter zeigt, und denke daran, dass der Ausgang eines OR-Gatters auf High geht, wenn mindestens ein Eingang auf High liegt.

Und wenn du schon einmal darüber nachdenkst, versuche doch einmal herauszufinden, wie du drei AND-Gatter mit je zwei Eingängen als Ersatz für ein AND-Gatter mit vier Eingängen zusammenschalten kannst.

Für ein Spiel mit vier Teilnehmern brauchst du natürlich zwei zusätzliche 555-Timer, zwei LEDs und zwei Taster.

Was den Schaltplan für ein Spiel mit vier Spielern angeht – ich überlasse es dir, ihn zu zeichnen. Skizziere zunächst eine vereinfachte Version, die lediglich die Logiksymbole zeigt. Wandle dies dann in einen Steckboard-Aufbau um (was der schwierige Teil ist). Noch ein Vorschlag: Meiner Meinung nach bist du für den Anfang mit Stift, Papier und Radiergummi schneller als mit der Software für Schaltungen oder Grafikdesign.

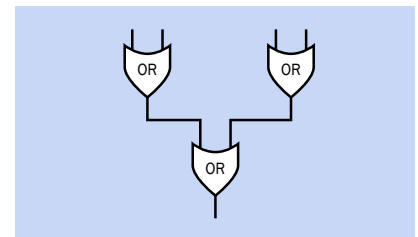


Abbildung 4-126.

Drei OR-Gatter mit je zwei Eingängen können ein OR-Gatter mit vier Eingängen nachbilden.