

Humboldt-Universität zu Berlin
Kultur-, Sozial- und Bildungswissenschaftliche Fakultät
Institut für Musikwissenschaft und Medienwissenschaft
Seminar: *Pong Studies*
Dozent: Dr. Stefan Höltgen
Sommersemester 2018

Hausarbeit:

Avoid missing Ball (from the Pong-Bot)

Untersuchung der zeitlichen Prozesse von „*Pong*“

Thomas Fecker
thomasfecker@web.de
Matrikelnr.: 590029

Inhaltsverzeichnis

1.	EINLEITUNG	3
2.	DIE ERSTEN SCHRITTE ZUM ZEITLICHEN PROZESS VON PONG	5
2.1	DAS BEWEGTE BILD VON PONG	5
2.2	UNTERSUCHUNG VON ELEKTRONISCHEN MEDIEN UND ZEITLICHEN SIGNALEN.....	7
3.	DIE ZEITLICHKEIT DER PONG-SCHALTUNG	9
3.1	DER TAKT.....	9
3.2	ERZEUGUNG UND STEUERUNG DER PADDLES.....	11
3.3	ERZEUGUNG DES BALLS.....	13
4.	PROJEKT „PONG-BOT“	14
4.2	PROGRAMMABLAUF DES PONG-BOTS.....	15
4.3	SYNCHRONISATION UND POSITIONSERMITTLUNG	17
4.4	ERFASSUNG DER BALLPOSITION	18
4.5	ERZEUGUNG DES STEUERSIGNALS FÜR DIE PADDLES.....	19
5.	IT’S NOT A BUG, IT’S PLAYABILITY!	19
6.	AUSWERTUNG	21
7.	LITERATURVERZEICHNIS.....	23
8.	ABBILDUNGSVERZEICHNIS	24

Anhang:	Kommentierter Programmcode
	Programmablaufplan
	Schaltpläne: CLK Platine
	Ballerzeugung
	Erzeugung Paddles
	Eigenständigkeitserklärung
	Prüfungsprotokoll

1. Einleitung

Mit der Erfindung des ersten „Pong“-Automaten setzte *Atari* in den 1970er Jahren den Grundstein für die heutige Videospielindustrie. Auch wenn „Pong“ nicht das erste Videospiel war, es gab bereits Videospiele wie „*Tennis for Two*“, die „*Odyssey*“-Konsole von *Magnavox* oder „*ComputerSpace*“, setzte es sich doch mit seinem einfachen Spielprinzip gegenüber anderen Videospielen dieser Zeit durch.¹ Aus diesem Grund führt an „Pong“, wenn man sich mit der Geschichte der Videospiele beschäftigt, kein Weg vorbei. Dies zeigt die Fülle an Literatur, die zum diesem Spiel verfasst wurde.

Dass über „Pong“ jedoch noch nicht alles gesagt und geschrieben wurde, ist das Ergebnis des Seminars *Pong-Studies*, welches Grundlage für diese Hausarbeit ist. Während des Projekts „Pong-Sound“, das Inhalt des Seminars war, entstand die Idee, sich „Pong“ aus einem technischen Blickwinkel zu nähern. Ziel soll es sein, die Zeitlichkeit bzw. die unsichtbaren (zeitlichen) Prozesse der Innenlogik² von „Pong“ zu ergründen. Um die zeitlichen Prozesse der „Pong“-Maschine zu untersuchen, wäre es eine Möglichkeit diese nachzubauen und selbst in Funktion zu setzen.³ Dies würde im Fall der „Pong“-Maschine jedoch einen sehr hohen Aufwand bedeuten. Auf der anderen Seite wäre eine rein theoretische Analyse des Schaltplans ebenso wenig förderlich, denn Beschreibungen des Schaltplans existieren bereits⁴ und zeigen nicht zwangsläufig, was in der Schaltung *eigentlich* geschieht⁵. Aus diesem Grund sollen die

1 Vgl. Claus Pias, *Computer Spiel Welten. Computerspiele*, München: sequenzia, 2002, S. 105.

2 Vgl. Ingo Köster, „Mediale Maßverhältnisse in Raum und Zeit. Ein Versuch der Systematisierung“, in: *Medien in Raum und Zeit*, hrsg. von Kai Schubert, Bielefeld: transcript, 2009. S. 23-45, hier S. 31.

3 Vgl. Wolfgang Ernst, *Gleichursprünglichkeit: Zeitwesen und Zeitgegebenheit von Medien. Vom photographischen zum televisionären punctum*, hrsg. von Wolfgang Ernst, Friedrich Kittler, Berlin: Kulturverlag Kadmos, 2012, S. 195-201, hier S. 198.

4 Diese sind für diese Hausarbeit jedoch weniger relevant, da hier vor allem Bezug auf die Schaltung des Pong-Nachbaus genommen wird. Bei diesem sind im Gegensatz zur Originalschaltung einige Schaltungsteile verändert worden. Aus diesem Grund wären solche Beschreibungen für diese Hausarbeit eher unbrauchbar.

5 Vgl. Wolfgang Ernst, *Gleichursprünglichkeit: Zeitwesen und Zeitgegebenheit von Medien. Der springende Punkt (die Legende vom diskreten Bildelement)*, hrsg. von Wolfgang Ernst, Friedrich Kittler, Berlin: Kulturverlag Kadmos, 2012, S. 233-226, hier S. 223.

zeitlichen Prozesse der „Pong“-Maschine anhand eines neuen Projekts untersucht werden.

Das Projekt „Pong-Bot“ ist aus dem Umstand entstanden, dass „Pong“ auf dem Arcade-Automaten lediglich im Zweispielermodus verfügbar ist. Die Paddles werden von den Spielern⁶ durch zwei Potentiometer direkt gesteuert. Der Automat ist nicht dafür konzipiert, allein zu spielen. Am Ende dieser Hausarbeit soll ein solcher *Single-Player-Modus* verfügbar sein.

Bevor der „Pong-Bot“ wirklich entstehen kann, gilt es zunächst die offensichtlichsten zeitlichen Funktionen der Bilderzeugung zu betrachten. Im Anschluss kann die Analyse der „Pong“-Schaltung erfolgen, denn es muss klar sein wie Pong im Ganzen zustande kommt und an welchen Stellen Eingriffe in die Funktion erfolgen können. Zwar werden in dieser Hausarbeit nur diejenigen Schaltungsteile bearbeitet, welche für den „Pong-Bot“ relevant sind, jedoch muss in der Vorarbeit zuerst das große Ganze betrachtet werden, damit die benötigten Prozesse der Schaltung verstanden und richtig genutzt werden können. Dies bildet die Basis für das gesamte Projekt. Anhand der Funktion lässt sich die Arbeitsweise des „Pong-Bot“ entwickeln.

Da es sich bei „Pong“ nicht um ein programmiertes (also softwarebasiertes) Spiel handelt, muss der „Pong“-Bot ebenfalls sein eigenes Stück Hardware besitzen. Verwendet werden soll ein programmierbarer Mikrocontroller, welcher in die Funktion von „Pong“ eingebunden wird. Das Ergebnis der gesamten Hausarbeit wird der „Pong-Bot“ sein, der die Ballposition auf dem Spielfeld erfasst und das ihm zugewiesene Paddle in die korrekte Position bewegt. Dies jedoch so, dass er für seinen menschlichen Gegenspieler noch besiegbar ist.

6 Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Sämtliche Personenbezeichnungen gelten gleichwohl für beiderlei Geschlecht.

2. Die ersten Schritte zum zeitlichen Prozess von Pong

2.1 Das bewegte Bild von Pong

Wie schon bei „Computer Space“ setzt Nolan Bushnell, der Mitbegründer der Firma *Atari*, bei den „Pong“-Automaten auf eine festverdrahtete Hardware und die Ausgabe über einen Schwarzweiß-Fernseher.⁷ Bei der festverdrahteten Hardware handelt es sich fast ausschließlich um integrierte Schaltkreise der TTL-Baureihe.⁸ Diese Art der elektronischen Bauteile arbeitet mit diskreten Signalen, also mit den binären Zuständen *high* und *low*.⁹ Um die Zeitlichkeit dieser elektronischen Schaltung zu untersuchen, ist es ratsam, zunächst das Bild an sich bzw. den Bildaufbau des Schwarzweiß-Fernsehers zu betrachten. Denn durch die Wahl des Ausgabegeräts wird die Funktion und Zeitlichkeit der gesamten Schaltung vorgegeben. Sie muss an die technischen Gegebenheiten und Eigenschaften des Ausgabegeräts angepasst sein.

Auch wenn das Bild beim Spielen von „Pong“ als fließend wahrgenommen wird, verbergen sich dahinter diskrete und zeitliche Prozesse. Das Bild wird mittels einer Kathodenstrahlröhre zeilenweise übereinander geschichtet erzeugt. Der Kathodenstrahl beginnt mit dem Bildaufbau in der linken oberen Ecke des Bildschirms, wandert bis zum rechten Ende des Bildschirms um danach eine neue Zeile zu beginnen. Dieser Vorgang wiederholt sich so lange bis der Kathodenstrahl das rechte untere Rand des Bildschirms erreicht hat und ein Bild vollständig erzeugt ist. Dieser Vorgang geschieht mit einer Frequenz von annähernd 60Hz. Durch die zeilenweise Bildgenerierung wird hier schon ersichtlich, dass es sich nicht um einen kontinuierlichen, sondern um einen diskontinuierlichen Prozess der Bildgenerierung handelt. Denn: „Zieht der Kathodenstrahl beim Bildschreiben auch kontinuierliche und amplitudenmoduliert seine Zei-

7 Claus Pias, *Computer Spiel Welten. Computerspiele*, München: sequenzia, 2002, S. 105-118, hier S. 112.

8 Stefan Hölting, „Logik. Schaltalgebra“, in: *Medientechnisches Wissen. Band 1: Logik, Informations- und Speichertheorie*, hrsg. von Stefan Hölting, Berlin/Boston: Walter de Gruyter GmbH, 2018 (Band 1), S. 78-121, hier S. 88.

9 Bei der TTL-Bauweise betragen die Spannungspegel: high > 2V, low < 0,8V.

len, gibt es zwei diskontinuierliche Momente, den des Zeilensprungs und des Bildwechsels [...]“.¹⁰ Der Bildwechsel und der Zeilensprung sind beim Bildaufbau die entscheidenden Faktoren, die den zeitlichen Rahmen eines Bildes festlegen. Während der Bildwechsel mit einer Frequenz von 60Hz vollzogen wird, muss die Frequenz des Zeilensprungs wesentlich höher sein, damit die übereinandergeschichteten Zeilen ein vollständiges Bild ergeben können (Abb. 1).

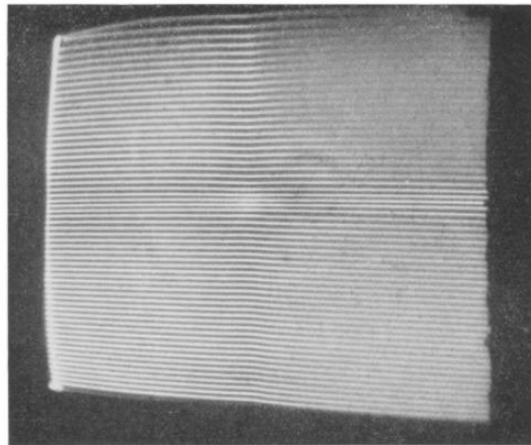


Abb. 1: Abbildung der Zeilen erzeugt durch einen Kathodenstrahl, URL: <https://link.springer.com/article/10.1007/BF01497902>

Die Zeit, in der bzw. die Frequenz, mit der der Kathodenstrahl die Zeilen schreibt – die Zeilensprungfrequenz – lässt sich bei „Pong“ sehr leicht ermitteln. Denn wenn man das Bild bei Pong genauer betrachtet, fällt auf, dass es nicht aus einer kontinuierlichen Linie aus Licht besteht, sondern sich – als ein statisches Element betrachtet – aus insgesamt 119210 einzelnen Bildpunkten zusammensetzt; 119210 Zeitpunkte zwischen zwei Bildwechseln, an denen der Kathodenstrahl ein- oder ausgeschaltet wird. Die Anzahl der Bildpunkte ergibt sich aus der Zeilenanzahl von 262 Bildzeilen, welche jeweils 455 einzelne Bildpunkte enthalten. Daraus lässt sich die horizontale

10 Wolfgang Ernst, *Gleichursprünglichkeit: Zeitwesen und Zeitgegebenheit von Medien. Kairoische Momente: Zeilenumbruch und Bildwechsel*, hrsg. von Wolfgang Ernst, Friedrich Kittler, Berlin: Kulturverlag Kadmos, 2012, S. 227 -233, hier S. 227.

Bildfrequenz – also die Zeit, in der eine Zeile vollständig geschrieben ist – mit annähernd 16kHz ermitteln.¹¹ Für den einzelnen Bildpunkt bedeutet dies wiederum, dass dieser mit einer Frequenz von 7MHz geschrieben werden muss. Die Betrachtung des Bildes als statisches Element ist dabei nur ein Hilfsmittel zur Erläuterung des Bildaufbaus, denn das sichtbare Bild des Kathodenstrahls steht niemals still, sondern ist ein – erst mit Abschalten des Bildschirms endender – Kreislauf, wodurch das statisch erscheinende Bild permanent in Bewegung ist.

2.2 Untersuchung von elektronischen Medien und zeitlichen Signalen

Um die Funktion einer elektronischen Schaltung vollständig zu verstehen, genügt es oft nicht, lediglich den Schaltplan zu analysieren. Die theoretische Erfassung der Funktion ist dabei nur ein Teil, denn jede Schaltung hat durch Bauteil- oder Verarbeitungstoleranzen immer ihre speziellen Eigenheiten. Es gilt also, durch Messung herauszufinden, was wirklich in der Schaltung geschieht¹² und welche Prozesse nicht nur theoretisch, sondern als darstellbares elektronisches Signal vorhanden ist. Zur Erfassung elektronischer Signale stehen unterschiedlichste (Mess-)Methoden zur Verfügung. Da es sich bei einer TTL-Schaltung wie „Pong“ fast ausschließlich um sehr schnelle zeitliche digitale Signale handelt, muss das Messgerät auch diese zeitlichen Signale darstellen bzw. sichtbar machen können. Die Prozesse müssen durch Messgeräte in einer Weise dargestellt werden, die unseren Sehgewohnheiten entspricht.¹³ Die beiden gängigsten Messgeräte zur Erfassung zeitlicher Signale sind der Logik-Analysator und das Oszilloskop. Beide Messgeräte haben bei der Signalerfassung Vor- und Nachteile. Beim Messen mit einem Logik-Analysator lassen sich viele Signale gleichzeitig, über einen längeren Zeitraum hinweg erfassen und darstellen. Dies ist

11 Die Korrektheit der Pixelzahl und der daraus resultierenden horizontalen Bildfrequenz wird im Abschnitt zur Zeitlichkeit der TTL-Schaltung ermittelt und bestätigt.

12 Wolfgang Ernst, *Gleichursprünglichkeit: Zeitwesen und Zeitgegebenheit von Medien. Der springende Punkt (die Legende vom diskreten Bildelement)*, hrsg. von Wolfgang Ernst, Friedrich Kittler, Berlin: Kulturverlag Kadmos, 2012, S. 223 -227, hier S. 223.

13 Florian Hars, „Physikgeschichte. Hundert Jahre Braunsche Röhre“, in: *Physikalische Blätter*, URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/phbl.19980541117>, Stand 23.11.2018.

ein großer Vorteil gegenüber dem Oszilloskop, denn bei diesem werden oft nur zwei Signale gleichzeitig erfasst und stehen nur für einen kurzen Augenblick sichtbar zur Verfügung.¹⁴ Der Vorteil eines Oszilloskops ist jedoch, dass die Signale viel präziser dargestellt werden können. Während bei einem Logik-Analysator ein ‚perfekter‘ rechteckiger Signalverlauf angezeigt wird, stellt das Oszilloskop den Signalverlauf, wie er wirklich ist, dar.¹⁵ Deshalb eignet sich das Oszilloskop für die Analyse der „Pong“-Schaltung wesentlich besser als der Logik-Analysator.¹⁶

Stellt man sich die Frage, wie eine elektronische Schaltung – wie die „Pong“-Schaltung – funktioniert, lässt man oft die Funktion des Messgerätes, mit dem eine Schaltung analysiert wird, außen vor. Denn gerade bei der Betrachtung von „Pong“ ist es interessant die Analyse mit einem Oszilloskop durchzuführen. Hier werden die zeitlichen Signale einer Schaltung analysiert, die zur Ansteuerung einer Kathodenstrahlröhre genutzt werden. Diese werden mit dem Oszilloskop erfasst, das ebenfalls eine Kathodenstrahlröhre für die Darstellung des zu messenden Signals verwendet.¹⁷ Der Unterschied liegt jedoch in der Art und Weise, wie der Kathodenstrahl gelenkt wird. Denn während, wie in *Abschnitt 2.1* erläutert, das Bild eines Schwarzweiß-Fernsehers zeilenweise übereinander geschichtet erzeugt wird, wird beim Oszilloskop der Kathodenstrahl mithilfe einer Sägezahnspannung gleichmäßig in waagrechter Richtung von links nach rechts über den Bildschirm geführt und nur in X- und Y-Richtung abgelenkt.¹⁸ Das hier wesentlich kontinuierlicher erscheinende Bild des Oszilloskops, besitzt ebenfalls ein diskontinuierliches Moment. Die Linie, welche der Kathodenstrahl

14 Oszilloskope mit mehr als zwei Kanälen sind zwar erhältlich, jedoch ist das klassische Zweikanal-Oszilloskop die gängigste Ausführung, da es weniger kostenintensiv in der Produktion ist.

15 Stefan Hölting, „Logik. Schaltalgebra“, in: *Medientechnisches Wissen. Band 1: Logik, Informations- und Speichertheorie*, hrsg. von Stefan Hölting, Berlin/Boston: Walter de Gruyter GmbH, 2018 (Band 1), S. 78-121, hier S. 117.

16 Es ist wichtig bei der Verbindung zweier unterschiedlicher Systeme die exakten Signalverläufe zu erkennen. Da die Taktflanken nicht in einem 90 Grad Winkel verlaufen, ist es wichtig die Zeit zu erkennen in denen sie zwischen zwei unterschiedlichen Zuständen wechseln.

17 Digitale Speicheroszilloskope werden hier nicht betrachtet. Diese arbeiten in der Darstellung zwar gleich, haben jedoch intern wesentlich andere Prozesse. Der größte Unterschied besteht darin, dass die elektronischen Signale in digitale Werte gewandelt und zwischengespeichert werden.

18 Vgl. Herbert Bernstein, *Messen mit dem Oszilloskop. Aufbau des analogen Oszilloskops*, Wiesbaden: Springer Fachmedien Wiesbaden, 2016 (2. Auflage), S. 4.

erzeugt, wird zwar nicht mehr in einzelne Bildpunkte geteilt, aber der Kathodenstrahl muss nach Vollendung einer Linie zum Anfang, an den linken Bildschirmrand, zurückspringen. Diese Diskontinuität haben das Oszilloskop und der Schwarzweiß-Fernseher gemeinsam. Das Oszilloskop ermöglicht dadurch einen anderen, viel tieferen Blick in die elektronischen Prozesse von „Pong“, welche für den Spieler verborgen bleiben. „Pong“ wird durch das Oszilloskop auf einer tieferen Ebene neu dargestellt.

3. Die Zeitlichkeit der Pong-Schaltung

Betrachtet man den Schaltplan von „Pong“, fällt das Fehlen eines Speichers auf, auf dem das Spiel hinterlegt ist.¹⁹ Dies bedeutet, dass alle Elemente des Spiels in Echtzeit erzeugt werden müssen, genau wie das Bild ständig in Bewegung ist. Die statisch erscheinenden Elemente des Spiels – das Netz, die Score-Anzeige, die Form der Paddles und des Balls – sind in der Zeitlichkeit der Schaltung gespeichert und entstehen im Moment, wenn der Kathodenstrahl über den Bildschirm wandert. Dies macht den „Pong“-Automaten, wie bereits das Spiel „Computer Space“, zu einem Computer, der nur den Zweck ein bestimmtes Spiel – in dem Fall „Pong“ – zu spielen, erfüllen kann.²⁰

3.1 Der Takt

Zentrales Element für alle zeitlichen Abläufe der TTL-Schaltung ist der Haupt-Takt – auch *Main-Clock* oder *CLK* genannt. Damit alle Elemente des Spiels korrekt dargestellt und gesteuert werden können, muss dieser wesentlich höher sein als die 60Hz des Bildwechsels und auch wesentlich höher sein als die Zeilensprungfrequenz von 16kHz. Es werden zusätzlich für die einzelnen Schaltungsteile unterschiedliche Takte mit unterschiedlichen Taktgeschwindigkeiten benötigt. Jeder Schaltungsteil benötigt seinen eigenen Takt, der wiederum vom Haupttakt abhängig ist, damit alle Schaltungsteile synchron miteinander arbeiten. Durch Teilung des Haupttaktes mithilfe von Zählerbausteinen, werden die benötigten Einzeltakte in „Pong“ erzeugt.

¹⁹ Ausgenommen sind 1-Bit-Speicher in Form von Flip-Flops.

²⁰ Claus Pias Claus, *Computer Spiel Welten. Computerspiele*, München: sequenzia, 2002, S. 105-118, hier S. 111.

Der Haupttakt wird bei „Pong“ mit Hilfe eines Quarzoszillators erzeugt.²¹ Die 14MHz Schwingung des Quarzoszillators wird durch die äußere Beschaltung zu einem Rechtecksignal umgewandelt (Abb. 2). Durch die Wandlung entsteht ein Rechtecksignal (*CLK*) mit einer Frequenz von 7 MHz. Dies entspricht der Frequenz, mit der die einzelnen Pixel durch den Kathodenstrahl geschrieben werden.²²

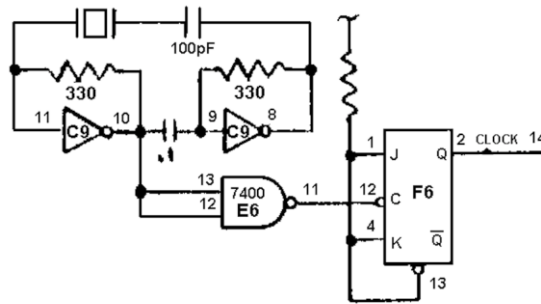


Abb. 2: Erzeugung des Main-Clock (CLK) mithilfe eines Quarz-Oszillator,
URL: <https://8bit-museum.de/scans/atari/pong8.gif>, Stand:02.01.2018.

Durch den CLK werden die beiden 4-Bit-Counter (2A und 2B) mit einem Zählertakt versorgt. Die beiden Counter erzeugen das Taktsignal für den Zeilensprung. Die Counter werden bei Überschreiten des Zählwerts (454) über das NAND-Gatter 2C zurückgesetzt. Dies bestätigt die bereits erwähnte Anzahl von 455 Bildpunkten für eine Zeile. Die horizontale Bildfrequenz bzw. die Zeilenfrequenz lässt sich nun rechnerisch überprüfen:

$$\frac{1}{\frac{1}{7 * 10^6 Hz} * 455} = 15384 Hz$$

21 Die Takterzeugung ist in der Dokumentation des „Pong“-Nachbaus nicht dokumentiert. Aus diesem Grund wird auf den Schaltplan der Original-Schaltung zurückgegriffen.

22 Die hier abgebildete Schaltung stammt aus dem Originalschaltplan, welcher im Internet frei zugänglich ist. Alle weiteren Beschreibungen beziehen sich auf den Schaltplan des „Pong“-Nachbaus. Dieser befindet sich im Anhang dieser Arbeit.

Mit Rücksetzen der beiden Counter 2A und 2B wird der Counter für die vertikale Bildposition um einen Wert inkrementiert. Der Counter für die vertikale Bildposition besteht aus den beiden 4-Bit-Countern 4A und 4B und werden mit überschreiten des Zählwertes 261 über das NAND-Gatter 5C zurückgesetzt. Die vertikale Bildfrequenz bzw. die Frequenz des Bildwechsels kann nun ebenfalls berechnet werden:

$$\frac{1}{\frac{1}{15384Hz} * 262} = 58,72Hz$$

3.2 Erzeugung und Steuerung der Paddles

Erzeugung und Steuerung der Paddles sind eng miteinander verbunden. Die Erzeugung bzw. Steuerung der beiden Paddles ist identisch, weshalb die Erzeugung nur für eines der Paddle erläutert werden muss. Die Funktion lässt sich analog auf das zweite Paddle übertragen.

Um die Funktion dieses Schaltungsteils zu analysieren, muss die Funktion des Bildaufbaus nochmals vor Augen geführt werden. Denn durch den zeilenweisen Bildaufbau ist es nicht möglich, dass das Paddle als ein kompletter Block aus Bildpunkten erzeugt und verschoben werden kann. Was hier durch den Spieler mit einer Drehbewegung des Potentiometers eingestellt wird, sind lediglich Zeitfenster innerhalb der Bilderzeugung, an dem der Kathodenstrahl eingeschaltet wird. Die Verbindung zwischen dem analogen Potentiometer und der digitalen Welt der TTL-Bausteine ist der *NE555* Zeitbaustein.²³

Im Grundzustand, wenn das Paddle nicht sichtbar ist, ist der Ausgang des *NE555* auf *high*. Dies führt dazu, dass der Counter 4B, welcher für die Erzeugung des Paddle-Signals verantwortlich ist, dauerhaft zurückgesetzt wird und somit das Paddle nicht sichtbar ist. Das Taktsignal /256V – welches durch den vertikalen Bildcounter auf der

²³ Bei dem *NE555* handelt es sich um ein aktives elektronisches Bauelement. Der *NE555* wird hauptsächlich für Zeitsteuerungen oder auch als Taktgenerator verwendet. Die Funktionsweise dieses Bauelements wird durch die äußere Verschaltung definiert. Durch die flexiblen Einsatzmöglichkeiten ist der *NE555* ein weitverbreitetes Bauelement.

Clock-Platine erzeugt wird – und die Einstellung des Potentiometers erzeugt ein pulswidenmoduliertes Signal am Ausgang des NE555. Je nach Position des Potentiometers wird die Dauer des Low-Pegels für die Position des Paddles erweitert oder verkleinert. Je kürzer dieser Low-Impuls ist, desto höher wird das Paddle auf dem Bildschirm sichtbar.

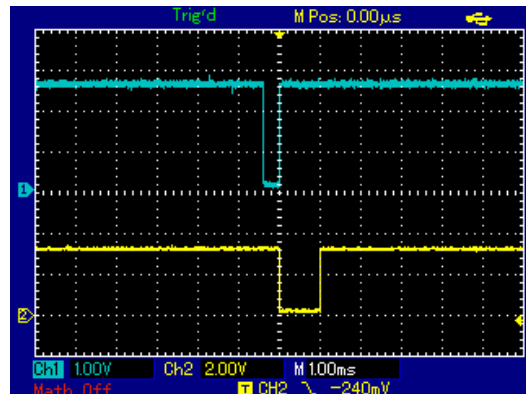


Abb. 3: Paddle-Signal (blau) bei Paddle-Position "oben" am Ausgang des NE555

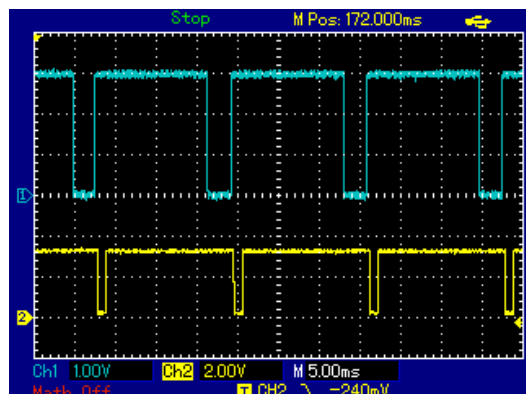
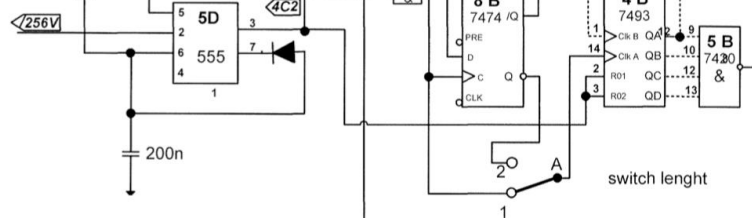


Abb. 4: Paddle-Signal (blau) bei Paddle-Position "mitte" am Ausgang des NE555

Wird der Ausgang des NE555 *low*, beginnt der Counter 4B zu Zählen. Dieser wird bei dem Zählerwert 15 wieder zurückgesetzt. Das Paddle ist also 15 Bildpunkte lang. Der Nachbau des Pong-Automaten besitzt an dieser Stelle eine Modifikation: Durch das *Flip-Flop 8B* kann die Länge der Paddles vergrößert werden.

Die Breite des Paddles wird von den Bausteinen 3A und 5A festgelegt. Werden die Signale 128H und 4H *high*, so wird der invertierende Ausgang des Flip-Flops A3 *low*.



Wie bereits bei der Erzeugung der Paddles, sind die Erzeugung und die Steuerung des Balls eng miteinander verbunden. Das Ballsignal wird aus zwei Signalsträngen aus der horizontalen und vertikalen Bilderzeugung, sowie der Signale der *Ball-motion-control* erzeugt. Für diese Hausarbeit ist die *Ball-motion-control* nicht relevant. Der Ball wird nur anhand seiner Position auf dem Bildschirm erfasst und nicht anhand seiner Fluggeschwindigkeit, seiner Flugrichtung oder seines Flugwinkels. Der Ball kann in zwei Bereiche aufgeteilt werden: den horizontalen Ball-Anteil und den vertikalen Ball-Anteil. Die Größe des Balls wird mithilfe von zwei 4-Bit-Counter realisiert.

Das Signal /H-BALL repräsentiert die horizontale Position des Balls. Wenn dieses *low* ist, ist der Ball sichtbar. Das horizontale Ball-Signal bzw. die horizontale Position, an der sich der Ball befindet, wird durch die beiden 4-Bit-Counter 2C/3C und das Flip-Flop 7D festgelegt. Durch die Verschaltung dieser drei Bausteine entsteht ein 9-Bit-Counter, welcher durch das Signal der Ball-Geschwindigkeit (*MOVE*) mit einem bestimmten Wert – abhängig der Ballgeschwindigkeit – vorgeladen wird. Wichtig für diese Hausarbeit ist lediglich, dass das Ballsignal für genau vier Takte des CLK, also 4 Pixel auf dem Bildschirm aktiv ist. Dies bei den Counterwerten von 508 bis 511.

Das Signal */VBALL* repräsentiert die vertikale Position des Balls. Wie das Signal */H-Ball* werden zwei 4-Bit-Counter zur Erzeugung des Signals verwendet. Die Counter *4C* und *4D* werden jedoch nicht durch den Haupttakt CLK getaktet, sondern über das */H-Sync*-Signal. Beeinflusst wird die vertikale Position durch einen Decoder, welcher die Steilheit der Flugbahn verändert.²⁴ Zur Erfassung des Ballsignals ist lediglich der Ausgang des Schaltungsteils, das */V-BALL*-Signal, von Relevanz. Dieses wird für vier Zustände (243, 247, 251, 255) des Counters *low*. Durch die Kombination beider Signale entsteht das */BALL*-Signal, welches auf diese Weise den Ball aus insgesamt acht Bildpunkte erzeugt.

4. Projekt „Pong-Bot“

Anders als die „Pong“-Hardware, soll der Pong-Bots nicht aus einer festverdrahteten TTL-Schaltung bestehen. Ziel ist es vielmehr zwei unterschiedliche Arten von Technik und Abläufen miteinander zu verbinden. Aus diesem Grund wird der „Pong-Bot“ als Software entwickelt, welche mithilfe eines Mikrocontrollers ausgeführt und verarbeitet wird. Die Plattform für den „Pong-Bot“ bildet das *STM42F4-Discovery-Kit* der Firma *STMicroelectronics*. Diese vorwiegend für Lehr- und Entwicklungszwecke eingesetzte Platine besitzt einen 32-Bit-ARM-Cortex-M4-Prozessor und diverse Peripherie- und Anschlussmöglichkeiten, wodurch diese Platine für eine Vielzahl an Anwendungen geeignet ist. Das Programm ist in der Programmiersprache C geschrieben und die für das *STM42F4-Discovery-Kit* entwickelte Standard-Library kommt darin zum Einsatz. Die Verwendung der Standard-Library führt zu einer einfacheren und übersichtlicheren Darstellung und Entwicklung des Programmcodes. Beispielsweise müssen einzelne Bits am GPIO-Eingang nicht durch logische Operatoren mit einem der GPIO-Registern verglichen bzw. abgefragt werden, sondern es reicht ein Aufruf einer

²⁴ Diese Signale werden an dieser Stelle nicht behandelt. Es ist lediglich wichtig, dass hier Werte für den Counter vorgeladen werden können, um die Flugbahn des Balls zu beeinflussen.

Funktion `(GPIO_ReadInputDataBit(GPIOx, GPIO_Pin_x) == x)` zum Abfragen eines bestimmten Zustandes an einem GPIO-Pin.²⁵

Dies erleichtert die Programmierung des Mikrocontrollers, hat jedoch in größeren und aufwändigeren Programmen eine eher schlechte Performanz, denn es werden viele für das Programm nicht benötigte Operationen ausgeführt. Für den „Pong-Bot“ ist dies jedoch von geringerer Bedeutung, da das Programm für diesen weniger aufwändig ist und in Verbindung mit der TTL-Schaltung keine hohe Performanz benötigt.

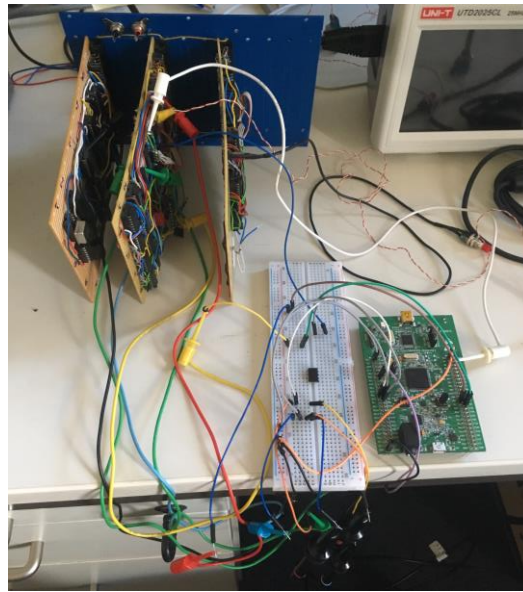


Abb. 6: Testaufbau des „Pong-Bot“.

4.2 Programmablauf des Pong-Bots

Vor der eigentlichen Programmierung des „Pong-Bot“ musste zunächst ein grober Programmablauf erstellt werden. Dieser wurde im Laufe der Entwicklungsarbeit angepasst und erweitert. Die Schwierigkeit lag allgemein darin, zwei unterschiedlich arbeitende Systeme – der Signalfluss der TTL-Schaltung und das C-Programm mit einem

²⁵ Dies ist nur ein Beispiel für eine Erleichterung des Programmierens mit C. Der wesentlichste Vorteil entsteht bei der Initialisierung des Mikrocontrollers, denn die einzelnen Register, welche aktiviert werden müssen, müssen nicht mehr im sehr umfangreichen Handbuch nachgeschlagen werden.

zeilenweisen Abarbeiten des Programmcodes – in Einklang zu bekommen. Aus diesem Grund mussten die Funktion und der Ablauf der Pong-Hardware im Vorfeld analysiert werden. Die „Pong“-Hardware besitzt in ihrem Ablauf keinen definierten Anfangs- oder Endzustand, sondern befindet sich, genau wie das Bild, welches sie erzeugt, in einem permanenten Fluss. Auch waren zu große Eingriffe in die Schaltung von Pong eher unerwünscht. Da die einzelnen Schaltungsteile eng und in zeitlicher Abhängigkeit miteinander arbeiten, konnte nur oberflächlich in die Schaltung eingegriffen werden, damit das zeitliche Gefüge der Schaltung nicht unterbrochen bzw. gestört wird. Abb. 6 zeigt den finalen Programmablauf, welcher im Laufe der Entwicklung entstanden ist. Dieser Programmablauf besitzt bereits die Korrektur einiger Probleme, wie beispielsweise das Problem der Synchronisation der beiden Systeme, auf die im nächsten Abschnitt näher eingegangen wird.

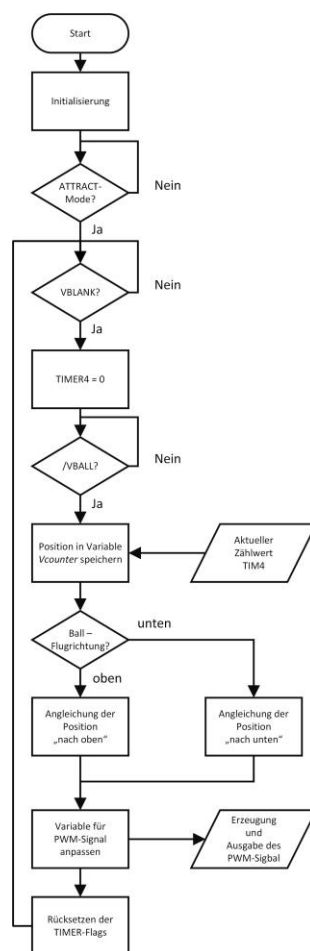


Abb. 7: Vollständiger Programmablaufplan des Pong-Bots

4.3 Synchronisation und Positionsermittlung

Damit beide Systeme miteinander interagieren können, müssen sie synchronisiert werden. Sie benötigen einen gemeinsamen Bezugspunkt, um die Prozesse aneinander anzupassen. Jedoch arbeiten beide Systeme mit unterschiedlichen Abläufen. Eine Möglichkeit sie zu synchronisieren besteht darin, einen gemeinsamen Takt für beide zu verwenden, wobei eines der Systeme im wortwörtlichen Sinne den ‚Takt vorgibt‘. Im hier vorliegenden Projekt würde dies bedeuten, dass man entweder den Takt der TTL-Schaltung an den Takt des Mikrocontrollers oder den Takt des Mikrocontrollers auf den Takt der TTL-Schaltung anpasst. Beide Möglichkeiten haben jedoch weitreichende Auswirkungen auf das jeweils angepasste System.

Wird der Takt der TTL-Schaltung an den des Mikrocontrollers angepasst, dann wird der zeitliche Ablauf der Schaltung zerstört. Die TTL-Schaltung ist so entworfen, dass sie das Spiel abhängig vom Haupttakt (*CLK* mit 7 MHz) erzeugt. Es kann also kein Eingriff in den Takt der TTL-Schaltung erfolgen. Eine Anpassung des Mikrocontroller-Taktes ist aber ebenfalls nicht möglich, da dieser nur in einem bestimmten Rahmen veränderbar ist. Zwar besteht die Möglichkeit, einen externen Takt in den Mikrocontroller einzuspeisen und zu verwenden. Dieser wird aber im Mikrocontroller intern beschleunigt, um auf die für diesen nötige Taktgeschwindigkeit zu kommen.²⁶ Um diese Probleme zu umgehen, wurde mithilfe des Mikrocontrollers intern eine Art ‚virtueller‘ Counter-Takt erzeugt, welcher permanent durch die Abfrage des */VBLANK*-Signals abgeglichen und korrigiert wird. Zur Angleichung des Counters mit dem */VBLANK*-Signal, wird der *GPIO_Pin_0* des GPIO-Port-B permanent in einer Schleife abgefragt.²⁷ Dies hat den Vorteil, dass kein größerer Eingriff in beide Systeme erfolgen werden musste, damit eine Synchronisation zustande kommt. Der interne Takt im Mikrocontroller wird durch einen Timer – durch den Timer 4 – erzeugt. Dieser

26 Auf dem Development-Board ist ein 8 Mhz-Quarz verbaut, mit dem der Mikrocontroller intern den Takt von 168 MHz erzeugt.

27 Programmcode, Zeile 39

wird mit 16 KHz initialisiert und inkrementiert mit dieser Geschwindigkeit eine interne Variable.²⁸ Diese Taktfrequenz entspricht ungefähr der Geschwindigkeit des Zeilenwechsels des Bildschirms (*Abschnitt 2.1*).

Liest man den Wert der Variable zu einem bestimmten Zeitpunkt aus, dann erhält man die Zeile bzw. die Zeilennummer, in welcher sich der Kathodenstrahl befindet, als numerischer Wert. Dieser Wert ist zwar nicht exakt, aber ausreichend für die Ermittlung der aktuellen Position des Kathodenstrahls bzw. der aktuellen Position, in der sich die TTL-Schaltung im Bildaufbau befindet.

4.4 Erfassung der Ballposition

Durch die Ermittlung der aktuell im Aufbau befindlichen Bildzeile lässt sich Ballposition feststellen. Da der Ball lediglich ein Signal in der Schaltung ist, kann durch Abfrage des Signalzustands und der aktuellen Bildzeile, die Position des Balls ermittelt werden. Hierdurch ergibt sich für den Ablauf der Positionserfassung des Balls folgender Ablauf: Mit dem *GPIOB_Pin_0* wird die Taktflanke des */VBLANK*-Signals (Bildwechsel) erfasst und der Timer 4 zurückgesetzt.²⁹ Der Timer startet den Zählvorgang so lange bis das */VBALL*-Signal an *GPIO_Pin_1* erscheint.³⁰ Wird das */VBALL*-Signal erfasst, so wird der aktuelle Zählwert des Timer 4 in die Variable *Vcounter* gespeichert.³¹ Damit ist in der Variable *Vcounter* die Zeilenzahl gespeichert, in der sich der Ball derzeit befindet. Die Variable kann nun in einem nächsten Schritt weiterverarbeitet und zur Steuerung des Paddle ausgegeben werden.³²

28 Programmcode, Zeile 121 -133.

29 Programmcode, Zeile 45.

30 Programmcode, Zeile 52.

31 Programmcode, Zeile 54.

32 Die Anpassungen, welche an der Variablen für die aktuelle Ballposition vorgenommen wurden, entstanden durch den Test direkt am Gerät. Da die TTL-Signale des Nachbaus gewissen Toleranzen unterliegen mussten diese Anpassungen im Live-Betrieb vorgenommen werden und waren zunächst nicht Teil des geplanten Programms.

4.5 Erzeugung des Steuersignals für die Paddles

Um so wenig wie möglich in die Schaltung von Pong eingreifen zu müssen, war im ursprünglichen Programmmentwurf die Steuerung der Paddles mithilfe eines *Digital-Analog-Wandlers* geplant. Das Potentiometer, welches vom Spieler bedient wird, sollte dadurch komplett ersetzt werden. Diese Lösung stellte sich jedoch als nicht durchführbar heraus, da der Eingang des *NE555* Zeitbausteins durch einen vorgeschalteten Kondensator mit einem nichtlinearen Spannungswert arbeitet. Aus diesem Grund wird in der finalen Version des „Pong-Bots“ das pulswidenmodulierte Signal des *NE555* vollständig durch den Mikrocontroller erzeugt.

Da der Wertebereich des pulswidenmodulierten Signals in *Abschnitt 3.2* bereits ermittelt wurde, kann dieses im Mikrocontroller erzeugt und anstelle des *NE555* in die „Pong“-Schaltung eingespeist werden. Um mithilfe eines Mikrocontrollers ein pulswidenmoduliertes Signal zu erzeugen, wird ein weiterer Timer benötigt. Verwendet wird der Timer 2, welcher in einer weiteren Unterfunktion initialisiert wird.³³ Der Timer wird so initialisiert, dass die Variable *Vcounter* nur minimal durch das Programm verändert werden muss. Die Ausgabe des Wertes erfolgt über die Funktion *TIM_SetCompare1(TIM2, Vcounter - 30)* und wird mit jedem neuen Wert, also mit jeder neu ermittelten Ballposition angepasst bzw. ausgegeben.³⁴

5. It's not a bug, it's playability!

Folgt man dem Ablauf der Bilderzeugung und dem Programmablauf, dann kommt man zu dem Schluss, dass hier der ‚perfekte‘ „Pong“-Spieler entwickelt wurde. Denn in der Theorie bedeutet dieser Programmablauf, dass der Ball durch das Paddle immer getroffen wird und somit der „Pong-Bot“ unbesiegbar ist. Dies ist jedoch nicht

³³ Programmcode, Zeile 135 –163.

³⁴ Programmcode, Zeile 68.

der Fall, auch wenn bei dem, in dieser Hausarbeit entwickelten, „Pong-Bot“ die Besiegbarkeit keine berechnete oder programmierte ist.³⁵ Vielmehr resultiert sie aus der Nutzung bestimmter Fehler, welche sich im Zusammenspiel zwischen TTL-Schaltung und Mikrocontroller ergeben. Dies sind einmal die Synchronisationsfehler, Wartezeiten des Mikrocontrollers und unvorhergesehene oder fehlerhafte Signale.

Die größte Abweichung entsteht bei der Synchronisation zwischen Pong-Schaltung und Mikrocontroller. Die Frequenz, mit der die Bildzeile geschrieben wird, ist nie exakt 16 kHz. Des Weiteren wird die Zeit zwischen zwei Bildwechseln durch den „Pong-Bot“ nicht berücksichtigt. Dies bedeutet, die erfasste Position, welche in der Variable *Vcounter* gespeichert ist, entspricht nie der tatsächlichen Position des Kathodenstrahls und damit auch nicht der aktuellen Ballposition. Trotz der permanenten Anpassung durch das */VBLANK*-Signal wird nie ein Zustand der absoluten Synchronisation erreicht. Die Abweichungen lassen sich durch diese unsaubere Synchronisation und trotz permanenter Anpassung nicht vermeiden.

Eine weitere Fehlerquelle sind die Wartezeiten des Mikrocontrollers. Die beiden Funktionen *delay_100us(1)*³⁶ werden zweimal im Programmcode aufgerufen und dienen dazu, die Taktflanken eines Signals genau zu erfassen. Da die Signale der „Pong“-Schaltung nicht immer die gleiche Flankensteilheit besitzen, muss erst geprüft werden, ob sich der Signalzustand wirklich geändert hat und nicht zwischen zwei Zuständen schwingt. Es wird deshalb oft durch eine kurze Wartezeit geprüft. Das Signal erreicht einen bestimmten Signalzustand und wenn dieser Signalzustand nach einer bestimmten Zeit immer noch vorhanden ist, kann davon ausgegangen werden, dass das erwartete Signal wirklich erreicht ist. Jedoch lassen sich so nicht alle fehlerhaften Signale herausfiltern, denn die Wartezeit kann nicht beliebig lang eingesetzt

35 Ein Beispiel für eine programmierte Besiegbarkeit ist der „Pong“-Gegner der *Atari VCS*. Bei jedem Bildwechsel werden die Ball- und Paddleposition mit einem Offset von ± 2 Bildzeilen angepasst. Die Anpassung des Paddles wird alle acht Bildwechsel ausgesetzt. So entsteht eine Abweichung, welche visuell kaum wahrnehmbar ist, aber ausreicht, um den „Pong“-Gegner zu besiegen.

36 Der Aufruf der Funktion *delay_100us(1)* bedeutet jedoch nicht, dass der Mikrocontroller an dieser Stelle des Programms 100 Mikrosekunden wartet, sondern dass er in das Unterprogramm *delay_100us(1)* springt und mit einer vorher festgelegten Geschwindigkeit (Hier 16,8 kHz) auf 1 zählt. Dieser Wert lässt sich jedoch frei definieren. Hier im Programm beträgt die Wartezeit ca. $0,06 \cdot 10^{-3}$ Sekunden.

werden. Wird sie zu lange, wird die Position des Balls kaum noch oder gar nicht mehr ermittelt.

Durch diese ‚Fehler‘ des Zusammenspiels zwischen „Pong“-Hardware und Mikrocontroller, entstehen Abweichungen, welche den „Pong“-Bot erst besiegbar machen. Das bemerkenswerte ist dabei, dass diese Fehler keinem Muster folgen. Da hier die Abweichung völlig willkürlich und in eingeschränktem Maße auftreten, lässt sich hierdurch der Pong-Bot als guten Gegner einsetzen um mit ihm Pong zu spielen.

6. Auswertung

Für diese Arbeit wurde ein großer Teil der Zeitlichkeit von Pong und dessen Hardware erarbeitet. Die hier dokumentierten und bearbeiteten Schaltungsteile bilden jedoch nur einen kleinen Teil der gesamten Schaltung. Sie sind jedoch die Grundlage der kompletten Zeitlichkeit der Schaltung. Vor allem der Takt (*CLK*) ist hier das zentrale Element zur Strukturierung und Koordination aller zeitlichen Prozesse in „Pong“. Während der Beschäftigung mit dieser Schaltung, ist immer wieder klar geworden, wie wichtig es ist, sich nicht nur mit dem Schaltplan auseinander zu setzen, sondern auch durch Messung nachzuvollziehen, was tatsächlich geschieht. Nur durch Kombination von theoretischer Erörterung anhand des Schaltplans und direktem Messen und Nachvollziehen der Signalverläufe wurde die Funktion der Schaltung in dieser kurzen Zeit erkenntlich. Dadurch wurde auch erst die bemerkenswerte Leistung der *Atari*-Mitarbeiter wirklich sichtbar und nachvollziehbar. Denn das komplexe zeitliche Zusammenspiel jedes einzelnen Schaltungsteils ist hier entscheidend; das Spiel entsteht erst in seinen zeitlichen Prozessen, welche bereits mit dem Anlegen der Versorgungsspannung beginnen und permanent im Hintergrund aktiv sind.

Das Ergebnis dieser Hausarbeit ist nicht nur ein tieferes Verständnis für die Abläufe im Inneren der „Pong“-Schaltung – welche dem Spieler völlig verborgen bleiben – sondern ein spielbarer „Pong-Bot“. Dadurch ist die (alte) „Pong“-Hardware nicht nur ein historisches Stück Technik, sondern wurde zu etwas, mit dem man in der Ge-

genwart arbeitet, etwas Neues dazu erfindet und neue Erkenntnisse durch diese Arbeit gewinnt – beispielsweise wie Fehler und Ungenauigkeiten eines Programmcode darüber entscheiden, ob ein „Pong“-Gegner besiegbar ist oder nicht.

Ein nächster Schritt nach dieser Hausarbeit wird die feste Einbindung des „Pong-Bot“ in den Nachbau von „Pong“. Eine Platine, welche sich in die Hardware von „Pong“ einfügt und so vollständig zu einem Teil des „Pong“-Nachbaus wird. Interessant wäre auch die Entwicklung eines „Pong-Bot“ in TTL-Technik, um noch näher an der Hardware von „Pong“ zu arbeiten. Dies wirft jedoch wiederum neue Probleme auf. Die nützlichen Fehler des bereits bestehenden „Pong-Bots“ müssten dann auf eine andere Art und Weise erzeugt werden, damit der „Pong-Bot“ wirklich besiegbar und damit spielbar ist. Dies wäre jedoch eine äußerst interessante Herausforderung um sich nochmals mit „Pong“, der „Pong“-Hardware und der Entwicklung eines „Pong“-Projekts auseinanderzusetzen.

Abschließend ist zu erwähnen, dass der größte Gewinn dieser Hausarbeit die Erkenntnisse sind, welche während der Entwicklung des „Pong“-Bots entstanden: die direkte Arbeit mit der Hardware, die sich als wesentlich effizienter herausgestellt hat als nur die theoretische Arbeit mithilfe eines Schaltplans. Dies ist ein sehr guter Grund alte Hardware zu konservieren oder nachzubauen, damit nicht nur das Wissen über die Funktion erhalten bleibt, sondern auch neue Projekte wie der „Pong-Bot“ entstehen können.

Das Ergebnis des fertigen „Pong-Bots“ steht als Video auf YouTube bereit. Das linke Paddle wird durch den „Pong-Bot“ gesteuert. Das rechte Paddle wird durch den Spieler (mich) bedient:

Video 1: <https://www.youtube.com/watch?v=YLKv72n0AKA>

Video 2: <https://www.youtube.com/watch?v=HkFuW1QrcT4>

7. Literaturverzeichnis

- Bernstein Herbert, *Messen mit dem Oszilloskop*, Wiesbaden: Springer Fachmedien Wiesbaden, 2016 (2. Auflage).
- Ernst Wolfgang, *Gleichursprünglichkeit: Zeitwesen und Zeitgegebenheit von Medien. Elektronische Zeitbilder*, hrsg. von Wolfgang Ernst, Friedrich Kittler, Berlin: Kulturverlag Kadmos, 2012.
- Hars Florian, „Physikgeschichte. Hundert Jahre Braunsche Röhre“, in: *Physikalische Blätter*, URL: <https://online-library.wiley.com/doi/abs/10.1002/phbl.19980541117>, Stand 23.11.2018.
- Hudec E., „Die Braunsche Röhre, insbesondere für Fernseh Zwecke“, in: *Naturwissenschaften*, URL: <https://link.springer.com/article/10.1007/BF01497902>, Stand: 14.12.2018.
- Höltgen Stefan, „Logik. Schaltalgebra“, in: *Medientechnisches Wissen. Band 1: Logik, Informations- und Speichertheorie*, hrsg. von Stefan Höltgen, Berlin/Boston: Walter de Gruyter GmbH, 2018 (Band 1), S. 78-121.
- Köster Ingo, „Mediale Maßverhältnisse in Raum und Zeit. Ein Versuch der Systematisierung“, in: *Medien in Raum und Zeit*, hrsg. von Kai Schubert, Bielefeld: transcript, 2009. S. 23 - 45.
- Nake Wolfgang, *Pong-Nachbau*. Dokumentation wurde mit der Hardware des Pong-Nachbaus für das Seminar „Pong-Studies“ bereitgestellt. Berlin: 2018.
- Pias Claus, *Computer Spiel Welten*, München: sequenzia, 2002.
- STMicroelectronics, „Reference manual. STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs“, hochgeladen von STMicroelectronics, Juni 2018, URL: https://www.st.com/content/ccc/resource/technical/document/reference_manual/3d/6d/5a/66/b4/99/40/d4/DM00031020.pdf/files/DM00031020.pdf/jcr:content/translations/en.DM00031020.pdf, Stand: 10.08.2018.
- STMicroelectronics, „User Manual. Discovery kit with STM32F407VG MCU“, hochgeladen von STMicroelectronics, Juni 2018, URL: https://www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf, Stand: 10.08.2018.
- Urbanski Klaus, Woitowitz Roland, *Digitaltechnik: Ein Lehr- und Übungsbuch*, Berlin: Springer, 2007 (5. Auflage).

8. Abbildungsverzeichnis

- Abb. 1: Abbildung der Zeilen, erzeugt durch einen Kathodenstrahl. URL: <https://link.springer.com/article/10.1007/BF01497902> (Abrufdatum: 01.10.2018).
- Abb. 2: Erzeugung des Main-Clock (CLK) mithilfe eines Quarz-Oszillators, URL: <https://8bit-museum.de/scans/atari/pong8.gif> (Abrufdatum: 02.01.2018).
- Abb. 3: Paddle-Signal (blau) bei Paddle-Position "oben" am Ausgang des NE555. Quelle: T. Fecker.
- Abb. 4: Paddle-Signal (blau) bei Paddle-Position "mitte" am Ausgang des NE555. Quelle: T. Fecker.
- Abb. 5: Erzeugung der eines Paddles aus dem Schaltplan des Pong-Nachbaus.
- Abb. 6: Testaufbau des Pong-Bot. Quelle: T. Fecker.
- Abb. 7: Vollständiger Programmablaufplan des Pong-Bots. Quelle: T. Fecker.