1 Einleitung

1.1 Messen und Vergleichen in der Leichtathletik

Im Sport und auf Wettkämpfen ist die oberste Maxime der faire Vergleich von Leistungen. Im Wettkampf sind nicht nur die Sportler und Athleten gefordert sich innerhalb der Wettkampfregeln zu bewegen, sondern vor allem auch die Schiedsrichter und Wettkampfrichter. Gerade die Wettkampfrichter in der Leichtathletik müssen nicht nur dafür Sorge tragen, dass der Wettkampf unter den gleichen vorgeschriebenen Bedingungen für jeden Athleten abläuft, sondern müssen es oft verstehen äußerst exakt Leistungen zu messen. Um einen fairen Leistungsvergleich zu gewährleisten muss das Messsystem gewissen Anforderungen entsprechen. Solange es sich nur um die Messung von z.B. Weitsprung handelt, reicht ein Maßband und ein das menschliche Auge vollkommen aus, um präzise Aussagen über die gesprungene Weite abzugeben. Schwieriger gestaltet sich die Zeitmessung bei Wettkämpfen. Dabei muss der Mensch unweigerlich technische Hilfsmittel verwenden, um präzise Aussagen über die benötigte Zeit machen zu können. Die weltbesten Athleten liegen nur hundertstel Sekunden auseinander. Hier stößt der Mensch als Zeitmesser endgültig an seine Grenzen. Um Zeiten exakt zu messen kann man heute auf Halbleitertechnik zurückgreifen, die eine wesentlich niedrigere Reaktionszeit als der Zeitnehmer Mensch vorweist. Auch im Schulsport stellt sich die Problematik des exakten Leistungsvergleichs. Daraus folgt die Notwendigkeit, eine Lösung zu entwickeln, die auch im Schulsport einen gerechten Wettkampfvergleich und gerechte Notengebung garantieren kann.

1.2 Zielsetzung des Messapparats

Um dieses Anliegen zu verwirklichen wird zur Aufgabe gestellt ein Gerät zu entwickeln, das den Anforderungen einer möglichst exakten Zeitmessung gewachsen ist. Die Zielsetzung dieser Aufgabe besteht im Wesentlichen darin, die Zeitmessung bis zu einer Genauigkeit von 1/100-stel Sekunden zwischen einem Startsignal und dem

Überqueren der Zielgeraden zu automatisieren. Neben dieser Grundvoraussetzung sollte ebenso eine einfache Bedienung zum Einsatz kommen und eine Möglichkeit bestehen die Messergebnisse auf einen Computer übertragen zu können.

2 Grundlagen

2.1 Installierung auf dem Sportplatz

Ein solcher Messapparat muss natürlich, bevor er Verwendung finden kann, auf der Laufstrecke installiert werden. Hierbei ist zu beachten, dass der Läufer keinesfalls durch mechanische Einwirkung oder Kabel gestört werden darf. Deshalb sollte der Messautomat neben der Laufbahn installiert werden und seine Arbeit verrichten können, ohne den Läufer dabei zu stören. Da viele Läufe in der Leichtathletik auf Rundkursen durchgeführt werden ist auch im Bezug auf die Stromversorgung ein mobiler Einsatz ohne 230V Kabel wünschenswert.

2.2 IR-Lichtschranke

Um die Zeit beim Zieleinlauf genauest feststellen zu können eignet sich im besonderen Maße die Verwendung einer Lichtschranke. Sie kann, ohne den Lauf zu behindern, das Überqueren der Ziellinie, das eine Unterbrechung zwischen Sender und Empfänger mit sich bringt, schnell und zuverlässig detektieren. Im Regelfall arbeiten Lichtschranken mit Lasern oder mit Licht im Infrarot (IR)-Bereich. Die Verwendung von „weißem Licht“ ist nicht praktikabel, weil die Störanfälligkeit durch Umgebungslicht sehr groß wäre. Da Laser noch sehr schwer beschaffbar sind und zudem teuer, wurde für eine Lösung mit einer IR-Lichtschranke entschieden. Der wesentliche Nachteil von IR-Lichtschranken besteht in der geringeren Reichweite gegenüber der Laser -Variante. Da die Breite der Laufbahn aber nicht mehr als ca. 12 Meter mit 8 Bahnen beträgt, ist dies ohne Bedeutung.

2.2.1 Aufbau eines IR-Senders

Für Lichtschranken eignen sich IREDs (**i**nfra**r**ed **e**mitting **d**iode), die häufig in handelsüblichen Fernbedienungen für Unterhaltungselektronik verbaut werden.

Abbildung 1: LD274, Siemens (vgl.Datenblatt:LD274; Siemens)

Diese IREDs sind verwandt mit Leuchtdioden (LED) mit dem Unterschied, dass für IREDs ein besonderes Halbleitermaterial, Galliumarsenid (GaAs) zur Verwendung kommt, da dieses besonders Licht im Infrarot-Bereich abstrahlt (vgl. Datenblatt: LD274, Siemens; S.1). Ein Beispiel ist die IRED „LD174“ von Siemens (siehe Abb. 1). Den unerwünschten Anteil des ausgestrahlten Spektrums absorbiert größtenteils ein Filter, der zugleich das Gehäuse der IRED bildet. So ist es möglich, dass eine IRED fast nur Licht mit 950nm abgibt. Somit sendet die IRED Licht im Bereich der Wellenlänge von etwa 950nm aus (siehe Abb. 2). Da keine exakte Wellenlänge wie bei einem Laser ausgesandt wird ist der Empfang von Signalen, die von einer IRED ausgehen, noch relativ störanfällig gegen Fremdlicht. Deshalb ist es notwendig das Signal später noch mit einer Trägerfrequenz zu modellieren. Durch die Modulation (Ein- und Ausschalten der IRED in einer bestimmten Frequenz) kann der Empfänger das Fremdlicht und gegebenenfalls fremde IR-Sender heraus filtern. Die Trägerfrequenzen bewegen sich hier im kHz-Bereich und sind vom jeweiligen Empfänger abhängig. Wie auch LEDs sind IREDs spannungsgesteuert. Je größer die Spannung ist, welche an der IRED abfällt, um so größer ist der Betrag der Stromstärke, die durch den Halbleiter fließt (vgl. Datenblatt: LD274, Siemens; „Forward current“ S.4). Umso mehr Strom durch die Diode fließt, desto heller leuchtet sie (vgl. Datenblatt: LD274, Siemens; „Radiant intensity“ S.4). Hierbei muss man auf den zulässigen Maximalstrom achten (vgl. Datenblatt: LD274, Siemens; „Permissible

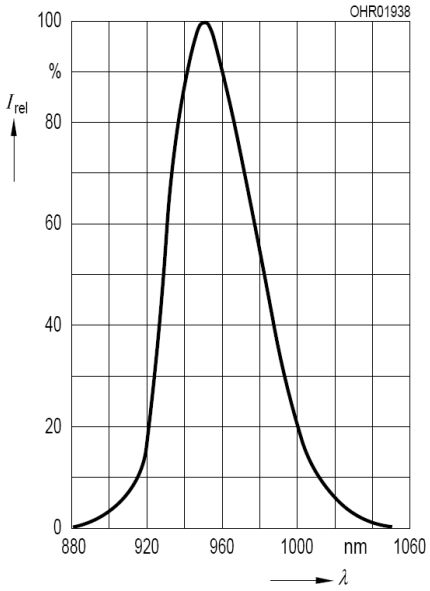

Abbildung 2: Spektrumsverteilung IRED des Typs „LD274“ von Siemens(vgl. Datenblatt: LD274, Siemens; S. 4)

pulse handling capability“ S.4). Die Strahlungsstärke lässt sich zudem beliebig vergrößern, indem man mehrere einzelne IREDs parallel oder in Reihe schaltet. Bei der Ansteuerung müssen diese Parameter eingehalten werden, da sonst die Dioden aufgrund des zu hohen Stroms und die damit folgende Wärmeentwicklung zur Zerstörung führen.

2.2.2 Aufbau eines IR-Empfängers

Das passende Gegenstück für die IREDs, der IR-Empfänger, ist etwas komplexer im Aufbau. Die Empfänger für IR-Signale wie zum Beispiel die Tsop17-Baureihe von Vishay oder die SFH5110-Reihe von Osram/Infineon zeichnen sich zwar nicht durch schnelle Verarbeitung der Signale aus, dafür aber durch äußerst gute Unempfindlichkeit gegen Störlicht. Die Empfänger sind integrierte Schaltkreise, die eine Fotodiode mit Vorverstärker, automatische Verstärkungsregelung, Bandpassfilter, Demodulator und eine Ausgangsstufe in einem Gehäuse vereinen (siehe Abb. 3).

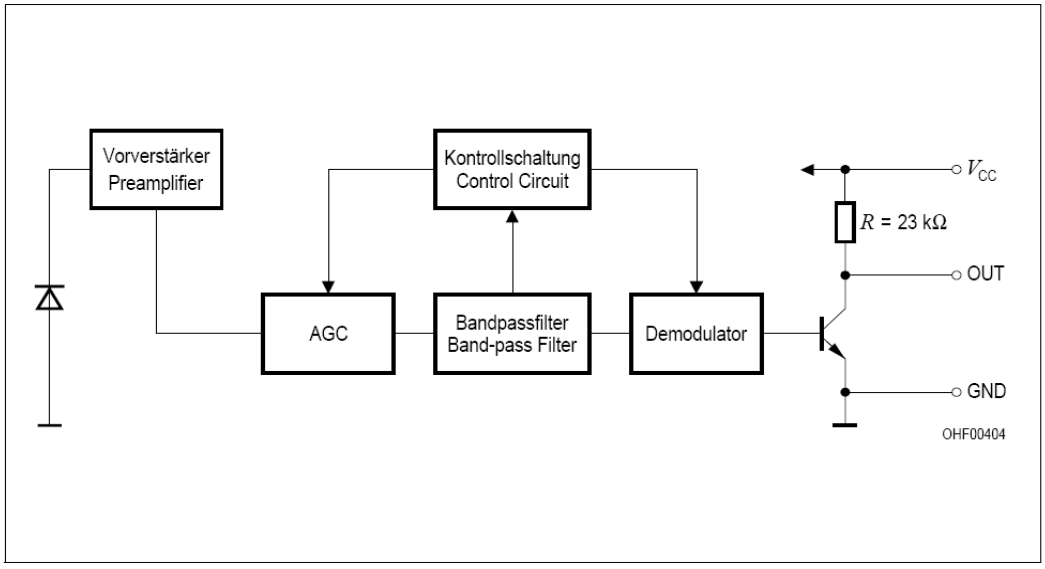

Abbildung 3: Blockschaltbild SFH 511x, Siemens (vgl. Application Note SFH 511x von Osram/ Infineon; S.3)

Die Fotodiode mit Vorverstärker wandelt die IR-Strahlung in Strom um. Anschließend wird dieser zur Weiterverarbeitung verstärkt. Daraufhin fließt ein Strom durch die automatische Verstärkungsregelung (AGC = Automatic Gain Control). Die AGC regelt die Verstärkung des Signals in Abhängigkeit vom Umgebungslicht. Bei absoluter Dunkelheit können die größten Reichweiten erreicht werden, da es kein Störlicht gibt,

welches aus dem Signal gefiltert werden muss. Das Bandpassfilter sorgt dafür, dass nur Signale, welche die entsprechende Trägerfrequenz besitzen an den Demodulator weiter geführt werden. Der Demodulator sorgt schließlich dafür, dass das modulierte Signal im kHz-Bereich demoduliert wird und nach Passieren der Ausgangsstufe am Output für die Auswertung bereit steht. Durch den verwendeten Schaltverstärker (Transistor) kann der Ausgang gleich mit einem Mikrocontroller verbunden werden. (vgl. Application Note SFH 511x von Osram/Infineon; S.3 ff). Der Schaltkreis und die Fotodiode sind in einem Kunststoff eingegossen, der zugleich als Filter für Tageslicht verwendet wird, um die Fremdlichtstörungen, welche an die Fotodiode gelangen, so gering wie möglich zu halten. Der Kunststoff lässt nur 10% des Lichtanteils $\lambda < 830nm$ durch. Der Kunststofffilter ist ausgelegt mit IREDs, die mit 950nm Licht aussenden, zu arbeiten (vgl. Application Note SFH 511x von Osram/ Infineon; S.5) Des weiteren muss bei den IR-Empfängern darauf geachtet werden, dass man sie mit einem Signal ansteuert, welches den Spezifikationen entspricht. Eine gute Auflistung, welche Kriterien die Datenwörter neben der Trägerfrequenz erfüllen müssen, findet man unter „Suitable Data Format“ auf Seite 3 des Tsop17 Datenblatts von Vishay. Dies gilt gleichermaßen für den SFH 511x. Zusätzlich wird empfohlen gegen Störungen, ausgehend von der Versorgungsspannung, dem Baustein einen RC-Filter vor zu schalten (vgl. Application Note SFH 511x von Osram/ Infineon; S.7).

2.3 Timer IC NE555

Der Timer IC NE555, welcher später Verwendung finden soll, um die Modulationen am IR-Signal zu erzeugen und zu überwachen, ist ein häufig verwendetes IC im Bereich von Frequenz-Modulation und Überwachung. Sehr flexibel einsetzbar ist der NE555 und erfreut sich auf Grund dessen und des günstigen Anschaffungspreises größter Beliebtheit. Nicht zuletzt durch viele „Applications Notes“ und Standardschaltungen ist er ein sehr gern gewähltes Timer IC. Das Aufgabengebiet erstreckt sich über viele Anwendungen mit und um Frequenzen.

Im Inneren des NE555 fügt sich ein FlipFlop (Bistabile Kippstufe), zwei Komparatoren („Vergleicher“), ein Transistor und eine Ausgangsstufe zu einer

funktionellen Gruppe zusammen (siehe Abb. 4 und Tabelle 1). Drei Widerstände dienen als Spannungsteiler. An dem ersten Komparator liegen somit 1/3 VDD am negativen Eingang an, während an dem zweiten Komparator 2/3 VDD am positiven Eingang anliegen. Beide Ausgänge der Komparatoren sind mit dem bistabilen FlipFlop verbunden. Am invertierten Ausgang des FlipFlops ist ein Ausgangspuffer geschaltet. Dieser ermöglicht hohe Ströme am Ausgang des ICs (bis 200mA). Zudem ist am FlipFlop noch ein Transistor mit Vorwiderstand angebracht für den Entlade-Pin des NE555.

Für die einzelnen Zustände an den Anschlüssen des ICs ergeben sich die in Tabelle 2 dargestellten Beziehungen. „As previously established“ bedeutet, das der zuletzt eingetretene Zustand erhalten bleibt.

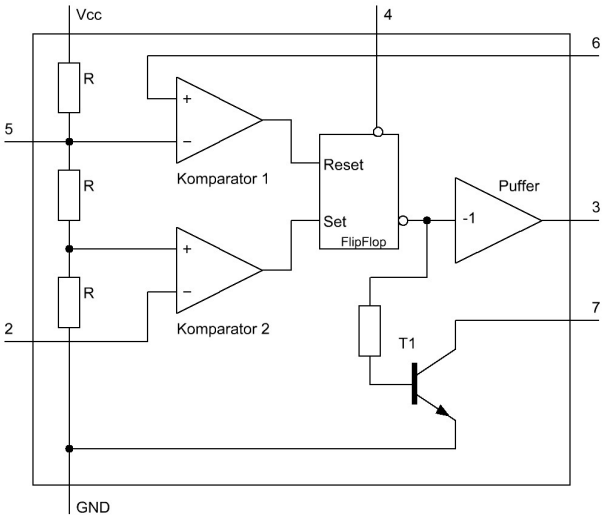

Abbildung 4: Functional Block Diagram (vgl. Datenblatt NE555, Texas Instruments; S.2)

Pin	***Bezeichnung***
GND	Masse
2	Trigger
3	Output (Ausgang)
4	Reset
5	Control (Steuerung)
6	Threshold (Rücksetzen Indirekt)
7	Discharge (Entladung)
VDD	Versorgungsspannung

Tabelle 1: Pin-Belegung NE555 (vgl. Datenblatt NE555; Texas Instruments; S.1)

Reset	**Trigger Voltage**	**Threshold Voltage**	**Output**	**Discharge**
„low“	Irrelevant	Irrelevant	„low“	On
„high“	< 1/3 VDD	Irrelevant	„high“	Off
„high“	> 1/3 VDD	> 2/3 VDD	„low“	On
„high“	> 1/3 VDD	< 2/3 VDD	As previously established	

Tabelle 2: "Function Table" (vgl. Datenblatt NE555, Texas Instruments; S.2)

2.3.1 Astabiler Multivibrator

Der NE555 kann durch nur wenige externe Bauteile zu einem Astabilen Multivibrator werden. Am Ausgang des Timers entsteht somit ein Rechtecksignal. Später wird der Astabile Multivibrator für die Modulation des IR-Signals eingesetzt. Die Schaltung findet sich im Datenblatt des NE555 von Texas Instruments, S. 10ff.

Im Folgenden soll gezeigt werden wie der periodische Zyklus zu Stande kommt. Dafür werden nur die relevanten PINs des ICs betrachtet. Es wird davon ausgegangen, dass die Versorgungsspannung anliegt und Reset auf „high“ liegt, damit der NE555 überhaupt seine Aufgabe erfüllen kann. Man nimmt an, der Kondensator C1 ist bis auf knapp 1/3 VDD aufgeladen. Der interne Transistor, welcher mit seinem Kollektor den Discharge-PIN des NE555 bildet, wird vereinfacht als Schalter dargestellt (siehe Abb. 5). Da der Ausgang aufgrund der anliegenden Spannung an Trigger und Threshold den Zustand „high“ besitzt, ist der Discharge geöffnet. Der Kondensator C1 lädt sich nun über R1 und R2 auf. Erreicht die Spannung an C1 einen Wert von 2/3 VDD, ändert sich der Zustand am Ausgang von „high“ auf „low“ (siehe Abb. 6). Dies hat zur Folge, dass der Schalter geschlossen wird. Nun entlädt sich C1 über R2 bis die Spannung an Trigger und Threshold wieder den kritischen Wert von 1/3 VDD erhält. Der

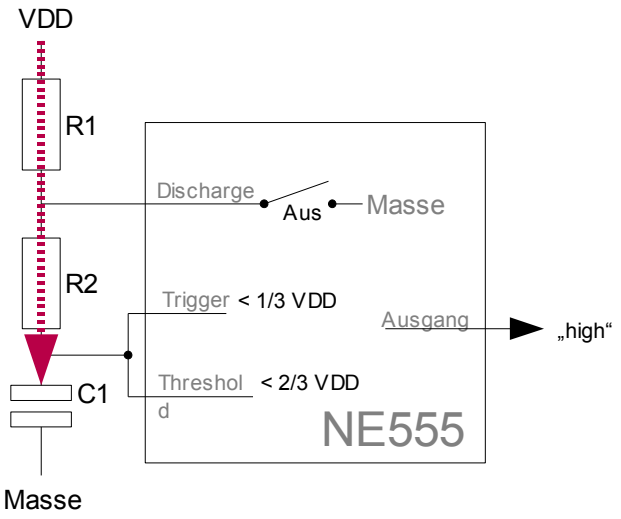

Abbildung 5: Funktionsschema NE555 Nr. 1

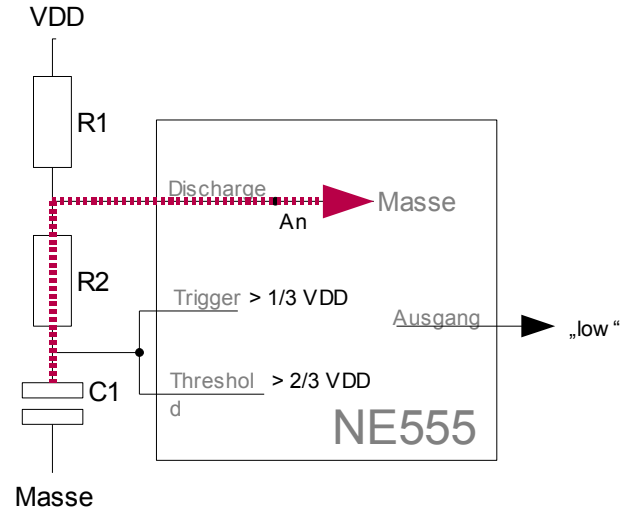

Abbildung 6: Funktionsschema NE555 Nr. 2

Vorgang beginnt wieder neu. Die einzelnen Pulsweiten von „high“ und „low“ lassen sich durch C1, R1 und R2 fast beliebig bestimmen (siehe Formel 1 und 2).

$$Formel\,1: \quad t_{high} = 0{,}693 * (R1 + R2) * C1$$

$$Formel\,2: \quad t_{low} = 0{,}693 * R2 * C1$$

vgl. Datenblatt: NE555, Texas Instruments; S.11

2.3.2 Pulse-Missing-Detektor

Für den Empfänger der Lichtschranke ist es wichtig ein unterbrochenes Signal detektieren zu können. Hierfür wird wieder der NE555 eingesetzt als Pulse-Missing-Detektor. Die Schaltung erkennt zuverlässig Störungen, z.B. einen zu langen Pulse oder einen ungewöhnlich langen zeitlichen Abstand zwischen zwei folgenden Pulsen. Das Inputsignal, an dem die zu überwachende Pulsfolge anliegt, ist mit Trigger verbunden und außerdem mit der Basis eines PNP Transistors (T1) (schließt Stromkreis, wenn an seiner Basis „low“ anliegt). In den nächsten drei Abbildungen soll klar werden wie die Schaltung Abnormalitäten in der Pulsfolge erkennt.

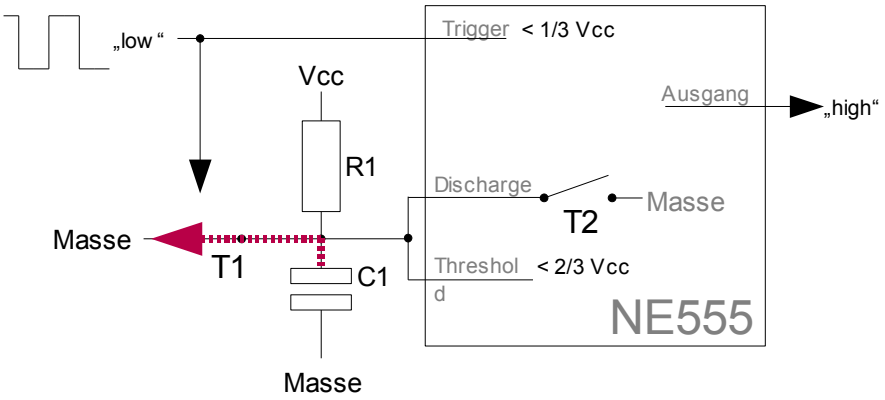

Abbildung 7: Funktionsschema NE555 Nr. 3

In Abbildung 7 liegt gerade ein „low“ an. Dieses hat zur Folge, dass T1 den Stromkreis schließt, C1 wird entladen. Somit liegen Threshold und Trigger unter einem Potential von <1/3 VDD, der Ausgang des NE555 ist somit „high“ und T2 offen.
Nun liegt in Abbildung 8 „high“ an, T1 öffnet und C1 wird über R1 geladen. R1 und C1

müssen so bestimmt werden, dass der „high“ Puls nicht länger ist als die Zeit, in der der Kondensator über R1 auf 2/3 VDD geladen wird. Wird dieser zulässige Zeitraum eingehalten, entlädt sich der Kondensator wieder durch eine „low“ Pulsphase wie in Abbildung 7 gezeigt.

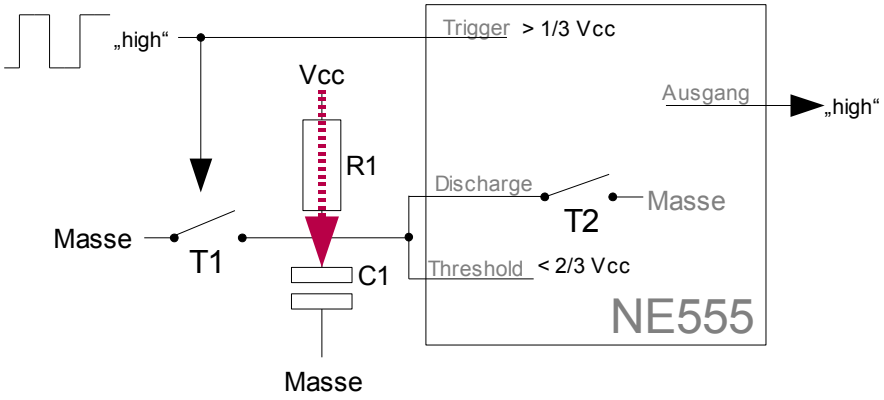

Abbildung 8: Funktionsschema NE555 Nr. 4

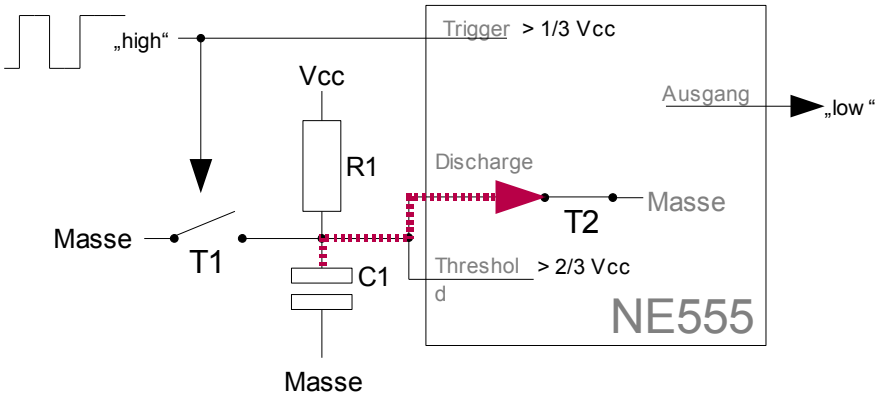

Abbildung 9: Funktionsschema NE555 Nr. 5

Wenn der Zeitintervall überschritten wird liegt an Trigger und Threshold ein Potential von >2/3 VDD an. Dies hat zur Folge, dass Schalter T2 geschlossen wird und sich der Kondensator entladen kann. Zudem fällt der Ausgang des NE555 auf „low“ und gibt somit ein eindeutiges Signal, dass die Pulsfolge in ihrer Regelmäßigkeit unterbrochen wurde. Der Zeitraum wird über C1 und R1 bestimmt (siehe Formel 3)

$$Formel\ 3: \quad t = 1{,}1 * R1 * C1$$

vgl. Datenblatt: NE555, Texas Instruments; S.9

2.3.3 Monostabile Operation

Nachdem der NE555 schon vertraut sein sollte, soll zuletzt noch ein Überblick über die Monostabile Schaltung gegeben werden. Die Monostabile Schaltung ist ähnlich wie die zuletzt beschriebene Schaltung aufgebaut, jedoch ohne einen externen PNP Transistor (T1 in Abb. 9). Nachdem Trigger von „high“ auf „low“ gezogen wird, wird eine einmalige Sequenz eingeleitet, bei der C1 über R1 auf 2/3VDD geladen wird (siehe Formel 3). Während dieser Ladezeit ist der Ausgang „high“, danach fällt der Ausgang wieder auf „low“ zurück und C1 wird über Discharge entladen. Im Monostabilen Betrieb eignet sich der Timer als Entpreller für Tasten. Der NE555 wird die Ansteuerung der Sende-Diode übernehmen und am IR- Empfänger auch seine Arbeit als „Missing-Puls-Detektor“ verrichten und zuletzt noch das Startsignal entprellen. Somit ist er ein wichtiger Partner an der Seite des eingesetzten Mikrocontrollers.

2.4 Mikrocontroller

Während der NE555 nur im Bereich der Frequenz Schaltung eingesetzt werden konnte, verlangte die moderne Steuertechnik nach Bausteinen, die universell und dynamisch angepasst werden können. Die Antwort lieferte die Entwicklung mit Mikrocontroller, welche in den 70er Jahren ihren Siegeszug begonnen. Mikrocontroller sind kleine integrierte Schaltungen auf Halbleiterbasis, welche möglichst viel Prephari wie z.B logische Ein- und Ausgänge, Timer, Zähler, Komparatoren usw. in einem kompakten Gehäuse vereinen. Mikrocontroller haben zudem einen integrierten Speicher, welcher vom Entwickler programmiert werden kann. Die Programmierung des Mikrocontrollers macht es möglich alle Ein- und Ausgänge mehr oder weniger abhängig vom verwendeten Mikrocontroller frei zu programmieren. Mit der integrierten Prozessoreinheit kann ein Mikrocontroller auch mathematische Verknüpfungen bearbeiten. Ein Controller stellt somit einen kleinen Computer dar. Seine Fähigkeiten hängen vom jeweiligen Typ und von der zum Einsatz gebrachten Programmiersprache ab (vgl. Programming an Customizing the AVR Micro-controller, § 1.1).

2.4.1 AVR-Mikrocontroller

Der Halbleiterhersteller Atmel drängt mit der AVR µC-Familie auf den Markt, einer ganzen Reihe von leistungsstarken 8 Bit Controllern. Die kompakten Schaltungen von Atmel sind besonders leistungsstark aufgrund ihres guten Verhältnisses zwischen aufgenommener Leistung und „Millionen Befehle pro Sekunde“ (MIPS/mW). Eine weitere Besonderheit der AVR Controller ist die gute Abstimmung mit C-Compilern (C ist eine der grundlegenden Programmiersprachen), da bei der Entwicklung schon ein großes Augenmerk darauf gerichtet wurde. Alle AVR-Mikrocontroller werden in Atmels CMOS Technologie gefertigt. Ein On-Chip In-System programmierbares Flash Memory sorgt dafür, dass der µC programmiert werden kann, selbst wenn er sich in der Schaltung befindet. Somit fällt ein lästiges Ein- und Ausbauen des Chips beim Programmieren weg. Den Entwicklern steht eine große Palette von verschiedenen Typen zur Verfügung. Das Feld der AVRs trennt sich in die „ATmega-“ und „ATiny-“ Typen. Die Bezeichnungen sind auf die unterschiedliche Ausstattung zurück zu führen. Während der ATiny-Controller nur wenige Anschlüsse besitzt, kann die Reihe der ATmega-µC große Speicher und zudem Ausstattung für den Highend Bereich im Feld der 8 Bit Controller aufweisen (vgl. AVR Risc Mikrocontroller, S.13).

2.4.2 Programmierung

Ein solcher Mikrocontroller kann erst sinnvolle Arbeit verrichten wenn er fachgerecht programmiert wird. Die einfachste Art einen µC zu programmieren führt über das Hardware SPI (Serial Peripheral Interface). Bevor man das Programm in den µC schreiben kann muss es in Assembler oder C geschrieben und anschließend durch einen Compiler in eine HEX-Datei transformiert werden. Die HEX-Datei kann dann in den µC übertragen werden. Hierzu muss er sich im Reset-Mode befinden (Reset „low“). Dann können die Daten über MISO (Master In Slave Out), MOSI (Master Out Slave In) und SCK (Serial Clock) in den µC übertragen werden. Die Programmiersoftware „Bascom“, die später eingesetzt wird, arbeitete mit der Hochsprache „Basic“ und enthält auch gleich einen Compiler und ein Programmierprogramm (CD-Menü >> Bascom

Demo). Eine rundum sorglose Lösung. Es gibt viele verschiedene Programmieradapter für den PC - sowohl für die Parallel-, Seriell- oder USB-Schnittstelle. Die Entscheidung ist Geschmackssache. Allerdings sollte darauf geachtet werden, dass die Programmier-software den Adapter unterstützt. Abbildung 10 stellt eine kleine Übersicht da. So kann die gesamte Peripherie des µC programmiert werden, wie zum Beispiel die Ein- und Ausgänge. Einen Überblick verschafft Abbildung 10.

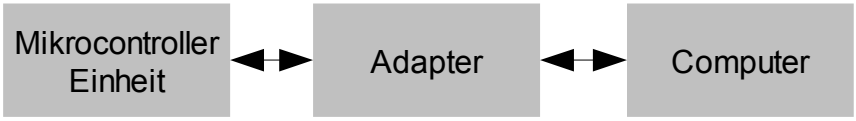

Abbildung 10: Verbindung zwischen PC und Mikrocontroller über Adapter

2.4.3 I/O Ports

Die wichtigste Aufgabe eines µC ist es digitale Zustände, d.h. eine logische „1“ oder „0“ bereitstellen zu können, sowie das Einlesen von logischen Zuständen. Dies kann der µC fast mit all seinen PINs, vorausgesetzt ist die richtige Konfiguration der einzelnen PORTs (PORT = in der Regel 8 frei konfigurierbare PINs des µC). Die PORTs und deren einzelne PINs können unabhängig voneinander durch die internen Steuerregister DDRx (Data Direction Register) und PORTx (Port Latch Register) konfiguriert werden. Das „x“ steht für den jeweiligen Port, meist durch a,b,c... abgekürzt. Beide Register können beschrieben und gelesen werden. So entstehen 4 mögliche Zustände. Siehe Tabelle 3 (vgl. AVR Risc Mikrocontroller, S.102 ff).

DDRx	PORTx	I/O	Pull-Up	Bemerkung
0	0	Eingang	Nein	Tri-State (hochohmig)
0	1	Eingang	Ja	Liefert Strom, wenn nach „low“ gezogen
1	0	Ausgang	Nein	Aktiver Ausgang („low“)
1	1	Ausgang	Nein	Aktiver Ausgang („high“)

Tabelle 3: vgl. AVR Risc Mikrocontroller, S.102

2.4.4 Interrupts

Des weiteren haben verschiedene PINs die besondere Funktion eines externen Interrupts, wobei der µC auch einige interne Interrupts besitzt. Ein solcher Interrupt kann den Programmablauf, ausgelöst durch ein eingetretenes Ereignis, unterbrechen. Nachdem die ISR (Interrupt Service Routine) ausgeführt wurde wird das zuvor unterbrochene Programm des µC weiter ausgeführt. Die ISR ist nichts anderes als ein kleines Unterprogramm, mit dem der µC auf ein Ereignis reagiert (z.B ein Tastendruck). Die Interrupts funktionieren mit Hilfe von Vektoren, die auf eine Speicherstelle im µC zeigen. An dieser Speicherstelle wird die auszuführende ISR abgelegt. Eine Übersicht über die internen und externen Interrupts und deren Hierachie ist in jedem Datenblatt eines µC vermerkt (vgl. Datenblatt: AtMega 8, Atmel; S.46 Tabelle 18).

2.4.5 Timer

Neben Interrupts besitzen µC auch ein oder mehrere Timer. Dieser ermöglicht dem µC einen Zeitraum exakt zu bestimmen. Die AVRs besitzen je nach Typ zwei unterschiedliche Timer, 8 und 16 Bit Timer. Einen so genannten „Prescaler“ sind beiden gemeinsam vor geschaltet. Der Mikrocontroller wird mit einer Taktfrequenz betrieben, welche entweder durch den internen RC-Oscillator oder durch ein externes Quarz erzeugt wird. Der Prescaler teilt diese Taktfrequenz durch 1, 8, 64, 256 oder 1024. Der Vorteil liegt darin, dass durch die verschieden mögliche Teilung der Taktfrequenz des µC sich später für den Timer größerer Spielräume ergeben. Die maximale Periodendauer ergibt sich aus der Taktfrequenz, dem Prescaler und dem verwendeten Timer (8 oder 16 BIT). Berechnet wird sie durch Formel 4.

$$Formel\,4 \quad : T = 2^N \times \frac{Prescale}{Taktfrequenz}$$

mit N = 8 oder 16 (Bit-Zahl des Timers)
Prescale = 1, 8, 64, 256, 1024

vgl. AVR Risc Mikrocontroller, S.146

Der 8 BIT Timer (Timer0) funktioniert nach dem einfachsten Prinzip. Er besitzt ein Register (8BIT => 256 Zustände). Bei jedem Takt, den der Timer durch den Prescaler bekommt wird das Register inkremediert. Hat das Register den maximalen wird von 255 (dezimal) angenommen läuft es über und verursacht einen Timer-Interrupt. Das Register wird zurückgesetzt und der Vorgang beginnt von vorn.
Der 16 BIT Timer bietet hier einige Möglichkeiten als PWM und Counter, wobei nur auf die Timer Funktion eingegangen werden soll. Wie der Name sagt, besitzt der Timer1 ein 16 BIT Register, das bei jedem vom Prescaler gegebenen Takt inkremediert wird. Die Besonderheit liegt nun darin, dass Timer1 nicht bei einem Überlauf des Registers einen Interrupt auslöst. Timer1 besitzt ein 16 BIT „Output Compare Register“. Dieses Register wird ständig mit dem aktuellen Zählerstand des Timers verglichen. Sind beide Register gleich, wird ein Interrupt ausgelöst und der Timer1 auf 0 zurückgesetzt. Der Vorgang beginnt von neuem. Dies hat den Vorteil, dass Timer1 so justiert werden kann durch das Output Compare Register. Der Wert für das Register berechnet sich nach Formel 5 (vgl. AVR Risc Mikrocontroller, § 4.2, 6.4, 6.4.1).

$$Formel5: \quad OutputCompare = \frac{f_{osc}}{Prescaler} * T_{soll} \quad \text{mit} \quad \text{Prescale} = 1, 8, 64, 256, 1024$$

vgl. AVR Risc Mikrocontroller, S. 147

Die Genauigkeit des Timers ist somit nur vom Taktgeber abhängig, da der Controller in Echtzeit auf den Interrupt reagiert. Als Taktgeber wird ein Standartquarz, das in der Regel eine Toleranz von +/-50ppm (Parts per Million) nicht überschreitet, verwendet. Die genaue Toleranz ist vom jeweiligen Hersteller abhängig. Daraus ergibt sich eine maximale Abweichung innerhalb von 20 Sekunden von 1 Millisekunde. Allgemeines über Quarze kann im Buch „The Crystal Cookbook“ von Bernd Neubig & Wolfgang Briese nachgelesen werden.

2.4.6 UART

Um die mit dem Timer gemessenen Zeiten später übertragen zu können, benötigt man

das UART (Universal Asynchronous Receiver/Transmitter). Durch das UART kann der µC mit anderer Hardware kommunizieren. Das UART besteht aus 2 Datenleitungen und einer beiden ICs gemeinsamen Bezugsmasse. Eine Leitung ist für das Senden (TxD für Transmit Data) von Daten zuständig, die andere für das Empfangen (RxD für Receive Data). Das Übertragungsprotokoll ist beim UART sehr einfach gehalten. Am Anfang der Übertragung wird ein Start-BIT gesendet um den Beginn der Übertragung zu signalisieren. Danach folgen die eigentlichen Datenbits. Die Übertragung endet mit dem Stop-BIT, welches das Ende der Übertragung dem Empfänger signalisiert. Im ruhenden Zustand ist die Leitung „high“, die abfallende Flanke leitet das Startbit ein. Wichtig bei der Übertragung ist, dass der Empfänger eine 16-fach höhere Abtastrate für das ankommende Signal hat als die eigentliche Übertragungsgeschwindigkeit (auch Baudrate genannt). Erst wenn mindestens 2 „Samples“ den gleichen Wert haben gilt die Übertragung des einzelnen BITs als erfolgreich. Diese doppelte Absicherung und die 16-fache Abtastung beim Empfangen ist unumgänglich für die Sicherheit der Übertragung der UART-Schnittstelle. In Abb. 11 wird der Ablauf deutlich (vgl. AVR Risc Mikrocontroller, S.111 ff).

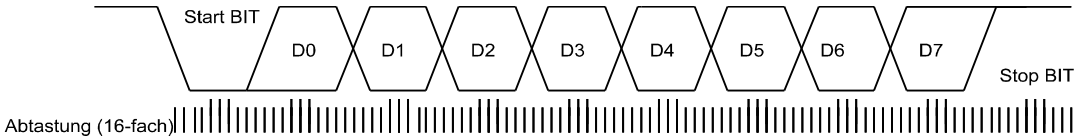

Abbildung 11: Übtertragungs Diagramm (vgl. AVR Risc Mikrocontroller, S.113)

2.5 Charakter LCD Anzeige

Um die gemessenen Zeiten auch ablesen zu können wird an den µC eine LCD Anzeige angeschlossen. Solche Anzeigen werden unter anderem oft in Getränkeautomaten eingesetzt. Sie zeichnen sich aus durch einen günstigen Anschaffungspreis und die einfache Ansteuerung. Die Display-Module bestehen aus einem Segment Treiber (Steuerchip) und der dazu gehörigen Anzeige, die mit 5x7 Pixel Feldern pro Charakter (Buchstabe) ausgestattet ist. Optional werden Hintergrundbeleuchtungen eingebaut um eine bessere Lesbarkeit zu ermöglichen.

In diesem Projekt soll das LCD Display mit der Modellbezeichnung „EA DIP082“ verwendet werden. Es bietet neben seinem sehr kompakten Design und Kompatibilität zum ASCII Zeichensatz (Standardtabelle für Buchstaben) die Möglichkeit es im 4 oder 8 BIT Modus (4 oder 8 Leitungen) anzusteuern. Da der Steuerchip KS0066 auch kompatibel zu anderen Display-Größen ist muss der Chip, bevor es das 2x8 Charakter Display richtig ansteuern kann, zuerst initalisiert werden. Bei der Initalisierung werden alle nötigen Informationen, die zur Steuerung des Displays benötigt werden einmalig in den KS0066 geladen. Neben den Datenleitungen, über die später Befehle und Charakters übertragen werden benötigt das Display noch bis zu drei Steuerleitungen. Wird die Enable (abgekürzt mit „E“)-Steuerleitung „high“, wird das angelegte Datenwort an den Datenleitungen gelesen, ist E „low“ wird kein Datenwort gelesen. Die zweite wichtige Steuerleitung hat die Bezeichnung „Register Select“ (kurz: RS). Ist RS „low“, wird das empfangene Datenwort als Befehl interpretiert. Dies ist hauptsächlich der Fall bei der Initalisierung des Moduls oder wenn später Einstellungen geändert werden. Ist RS „high“, wird das Datenwort als Charakter interpretiert und anschließend auf dem Display angezeigt. Der dritte und letzte Pin zur Ansteuerung ist der „Read/Write“ (kurz: R/W). Er wird nicht unbedingt benötigt. Ist er „high“, können zuvor in das LCD geschriebene Daten wieder ausgelesen werden. Wenn R/W „low“ ist werden Datenwörter in das Modul geschrieben. R/W kann auf Masse gelegt werden wenn man keine Daten aus dem Display lesen will.

Im Diagramm (Abb. 12) soll ein Überblick über den generellen Verlauf der Ansteuerung gegeben werden. Als Quelle dient das Datenblatt des KS0066 und des LCD Moduls EA DIPS082. Dort findet man unter anderem auch umfassendere und detaillierte Informationen über das Display.

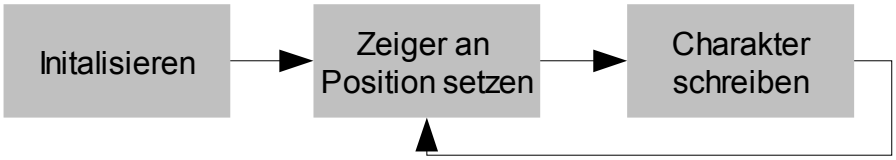

Abbildung 12: Ansteuerung LCD

2.5 USB Konverter FT232R

Damit die gemessenen Zeiten nicht nur auf dem LCD angezeigt werden können wird durch einen USB Serial Konverter, wie der FT232R von „FTDI Chip“, die Kommunikation zu einem PC möglich. Der FT232R ist auf der einen Seite mit dem Mikrocontroller durch das UART verbunden und auf der anderen Seite mit dem PC durch die USB (Universal Serial Bus) verbunden.
Der Aufbau einer solchen Verbindung ist im Datenblatt des FT232R genau beschreiben. Bei der Entwicklung der Schaltung wird der Schaltplan aus dem Datenblatt (vgl. Datenblatt: FT232R, FTDI Chip; „Self Powered Configuration“ S.20) übernommen.
Später kann der Chip noch mittels eines mitgelieferten Programms von FTDI Chip programmiert werden mit „persönlichen Daten“, z.B einen Namen für das USB Gerät (CD-Menü >> FT_Mprog). Außerdem können Einstellungen für eventuell angeschlossene Kontrolllampen hinterlegt werden. Bei diesem Projekt beschränkt man sich auf die Namengebung der Lichtschranke als „StoppUhrUSB“.

3 Schaltung und Konstruktion

3.1 Lichtschranke

Als erstes soll die Schaltung für die Lichtschranke entworfen werden. Hierbei werden die Erkenntnisse, Fakten und Diagramme aus Punkt 2.2 und 2.3 herangezogen. Bei dem ersten Test mit der Lichtschranke stellte sich heraus, dass der Tsop1740 besser mit etwas längeren Pausen zwischen den IR-Signalen zurecht kam als mit gleich langen Periodendauern von Signal und Pause. Des weiteren stellte sich heraus, dass vor Sender und Empfänger eine Abdunklung benötigt wird, um zusätzlich vor Sonnenlicht zu schützen. Auch der Einsatz von mehreren IREDs zeigte sich als sehr hilfreich. Hingegen den Angaben in den Datenblättern reichte die Energie einer IRED nicht für eine fehlerfreie Funktion der Schranke aus. Der Einsatz von Polarisationsfiltern und Reflektoren erwies sich auch als eine Sackgasse. Der Gebrauch von Linsen wirkte sich

allerdings positiv hinsichtlich der Reichweite aus. Deshalb wurden sie auch später in die Konstruktion aufgenommen.

3.1.1 Sendemodul

Das Sendemodul arbeitet in einem NE556, welcher dem NE555 gleich ist mit dem einzigen Unterschied, dass 2 NE555 in einem Gehäuse verbaut sind.

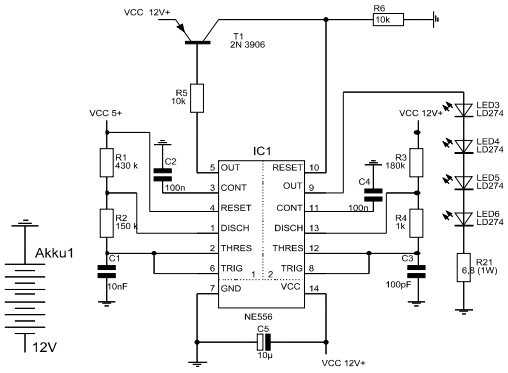

Abbildung 13: Schaltplan des Sendermoduls

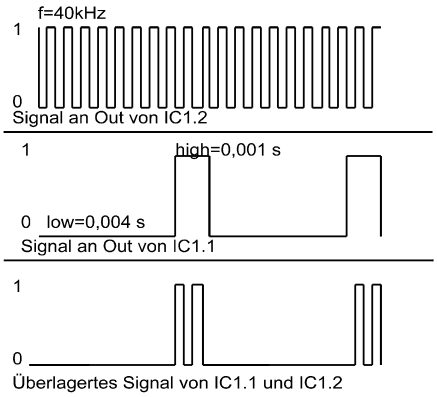

Abbildung 14: Erzeugte Frequenzen

Mittels IC1.2 (siehe Abb.13) wird das benötigte 40kHz Signal für den Tsop1740 erzeugt. Hierbei wird an den IC R3, R4 und C3 wie in Punkt 2.3.1 angepasst. Zu erwähnen ist, dass t_{high} etwas größer ist als t_{low} . R25 darf nicht 0 sein. Der geringe Unterschied kann allerdings vernachlässigt werden. Der Reset des IC1.2 ist invertiert verbunden mit dem Ausgang des IC1.1. Die Invertierung des Signals erfolgt mittels T1, R5 und R6. IC1.1, R3, R4 und C3 erzeugen wiederum eine Frequenz, mit der IC1.2 gesteuert wird. Die Periodendauern sind in Abb.14 vermerkt. C2, C4 und C5 stabilisieren die Spannung. IRED3-6 senden das IR-Signal aus, R21 begrenzt dessen Strom. Akku1 wird für die mobile Spannungsversorgung benötigt. Als Akku wird ein Blei-Akku verwendet da er bei einer konstanten Ladespannung von 13,8 Volt geladen werden kann ohne zu gasen. Daher ist er besonders einfach in der Wartung und zuverlässig.

3.1.1 Start- und Empfangsmodul

Als nächstes wird das Start- und Empfangsmodul entwickelt. Die Schaltung (siehe Abb. 15) hat einen ähnlichen Umfang wie die des Sendemoduls. IC2.1, T2, C6 P1, R7, bilden einen Puls-Missing-Detektor. Mittels dem Potimeter P1 können Feinabstimmungen vorgenommen werden. Der Input des Detektors (Basis von T2) ist mit der empfohlenen Standartschaltung (R10, R19 und C10) des Tsop 1740 verbunden, der die IR-Signale vom Sender empfängt. Der Ausgang des IC6.1 betreibt LED4, R26 begrenzt den Strom. Das Interrupt-Signal für den Mikrocontroller wird zwischen Out von IC2.1 und R16 abgegriffen.

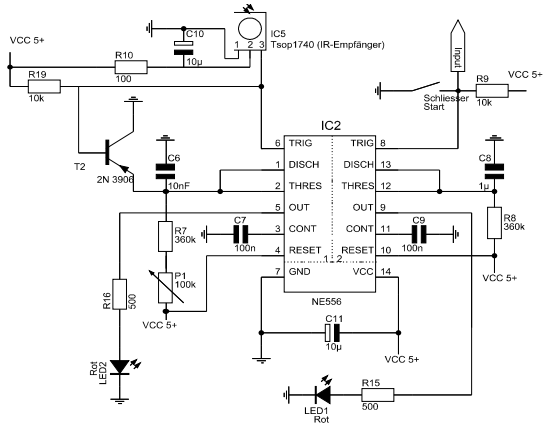

Abbildung 15: Schaltplan des Start- und Empfangsmodul

IC2.2 übernimmt das Erzeugen einer sauberen Flanke für das Startsignal. Der IC, C8 und R8 bilden eine Monostabile Schaltung. Getriggert wird die Schaltung über einen Schließer und R9, der wiederum den Strom begrenzt. Der Start wird durch LED1 signalisiert. R15 begrenzt den Strom, der durch LED1 fließt. Zwischen R15 und Out von IC2.2 wird der Interrupt abgegriffen. C11, C7 und C9 stabilisieren die Spannung für IC2.

3.1.2 Mikrocontroller, LCD, Spannungsregler und FT232R

Als letztes bleiben noch Mikrocontroller, LCD, Spannungsregler und der FT232R und ihre Schaltung übrig (siehe Abb. 16). Der Spannungsregler „µA7805“ hält die Spannung konstant bei 5 Volt, versorgt durch Akku2. C18, C19, C20 und C21 benötigt der Spannungsregler, um schnell Stromschwankungen ausgleichen zu können.
IC4, R17, R18, C16 und C17 bilden die „Self Powered Configuration“ des FT232R (vgl. Datenblatt: FT232R, FTDI Chip; „Self Powered Configuration“ S.20).
Der AtMega8 (IC3) wird mit seinem Reset zwischen C15 und R14 angeschlossen.

Durch C12 ist er gegen eventuelle Spannungsschwankungen geschützt. R14 begrenzt den Strom, wenn Reset auf „low“ gezogen wird. Damit der angeschlossene Oszillator störungsfrei arbeiten kann benötigt er ebenfalls 2 Kondensatoren C13 und C14. An PD5 (Port D, BIT Nr.5) wird ein kleiner Peeper verbunden, über PD8, PD7 und PB0 werden Tasten angeschlossen. Als Strombegrenzer (Pull-Up Widerstände) kommen R11-R13 zum Einsatz.

Zuletzt wird das LCD mit dem Controller verbunden. Das LCD arbeitet im 4 BIT Modus. Alle nicht benötigten Leitungen werden mit Masse verbunden. P2 und R20 bilden einen Spannungsteiler. Durch P3 kann später der Kontrast bestimmt werden.

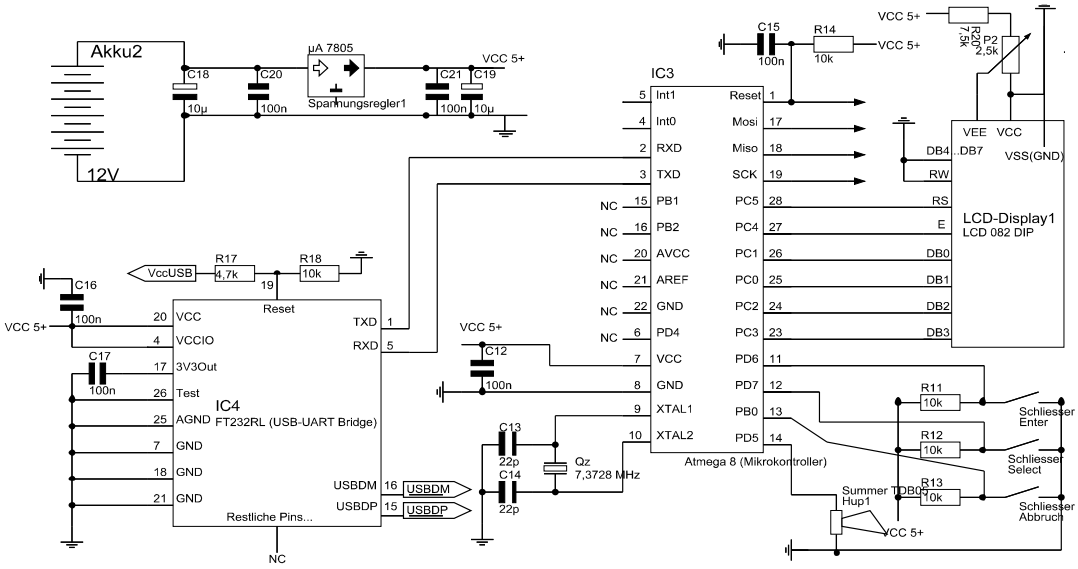

Abbildung 16 : Schaltung des Mikrocontrollers, LCD, Spannungsregler und FT232R

3.3 Gesamtschaltung

Fügt man nun die einzelnen Teile zusammen erhält man die fertige Schaltung für das Messsystem. Im Schaltplan (CD-Menü >> Schaltplan) ist die Gesamtschaltung und die Belegung der Steckverbindungen für USB, Startpistole und Stromversorgung vermerkt.

3.4 Platine und Gehäuse

Aufgrund der komplexen Schaltung war es nötig mehrere Platinen für den Schaltkreis zu erstellen. Alle Dateien und Vorlagen befinden sich auf der CD-Rom unter „Platine“. Die Platinen wurden selbst durch ein Ätzverfahren im eigenen Labor hergestellt.
Da das Gerät im Freien bei Veranstaltungen eingesetzt werden soll und im Schulbetrieb verwendet werden soll, empfiehlt sich ein robustes Alugehäuse. Dieses wurde nach Planzeichnung von einer in Landshut ansässigen Firma (P.Walzock GmbH) angefertigt. Der Konstruktionsplan ist auf der CD-Rom unter „Gehäuse“ abgelegt.

4 Erläuterung zur Programmierung und Steuerung

4.1 Mikrocontroller Programmablauf

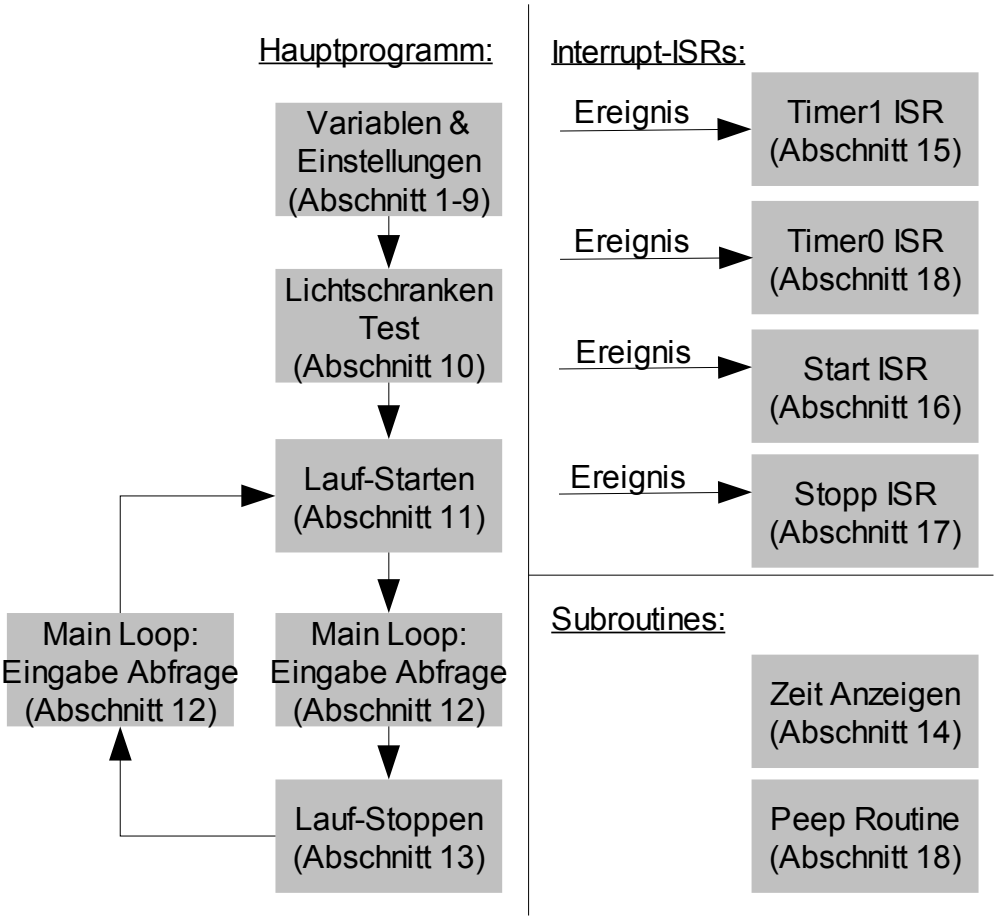

Abbildung 17: Programmablauf des Mikrocontroller

Als letzte Aufgabe bei der Realisierung des Messsystems bleibt die Programmierung des Mikrocontrollers. Eine Übersicht über den Programmablauf gibt Abb. 17.

Das gesamte Programm wurde mit Bascom geschrieben. Als Anleitung diente das Manual von Bascom, das neben der Demo von Bascom frei zum Download zur Verfügung steht (CD-Menü >> Bascom Demo).

Timer0 ISR spielt eine untergeordnete Rolle. Läuft Timer0 (8BIT Timer) über wird der Peep-Ton ausgeschaltet, der zuvor durch ein Ereignis eingeschaltet wurde (Start- bzw. StopISR).

Timer1 (16BIT Timer) übernimmt die eigentliche Zeitmessung in Millisekunden. Bei jedem ISR Aufruf wird eine Variable um 1 erhöht und anschließend in Minuten, Sekunden und Millisekunden dargestellt. Der Timer wurde zuvor in Abschnitt 6 für eine Periodendauer von 1 Millisekunde konfiguriert.

In der StartISR, die zuerst durch Abschnitt 11 freigegeben werden muss, wird die Timer1ISR aktiviert und die StoppISR. ausgelöst wird sie durch das Drücken des Starttasters.

Wird nun die Lichtschranke durchbrochen, wird die Stopp ISR aufgerufen, die die verstrichene Zeit abspeichert und sie gleichzeitig über das UART an den PC sendet.

Im Abschnitt 1-9 werden Variablen definiert und Konfigurationen vorgenommen. Dies ist selbsterklärend und im Quellcode selbst vermerkt, der in (CD-Menü >> Quellcode) C zu finden ist.

Der Lichtschranken -Test dient zum Einstellen der Lichtschranke. Zur Überprüfung mit Enter beendet man die Testphase.

In Abschnitt11 (Lauf Starten) werden alle Variablen initalisiert und die StartISR „Enable“ geschaltet.

In Abschnitt12 (Main Loop) werden in einer Schleife die Eingabetasten abgefragt und das UART und gegebenenfalls die Routinen aufgerufen.

Mit der Entertaste kommt man zu Abschnitt13 (Lauf Stoppen). Hier werden alle Interrupts deaktiviert und falls Zeiten gemessen wurden diese durch Aufrufen der Subroutine „Zeiten Anzeigen“ auf dem Display ausgegeben.

Mit erneuter Betätigung der Entertaste gelangt man über Abschnitt12 wieder zu Abschnitt11.

Durch Pressen der Zeit-Taste werden die Zeiten chronologisch angezeigt, falls

vorhanden.

Die Peep Routine gibt für 30ms einen Peepston aus und wird immer zur Bestätigung eines Ereignisses aufrufen.

Die Ansteuerung über die USB Schnittstelle ist in Tabelle 4 dargestellt.

Befehl	**Ereignis**	**Rückmeldung**
„R“	Lauf Stoppen	„{S}“
„S“	Lauf Starten	„{B}“
„A“	Schranke Aktivieren	„{A}“
„D“	Schranke Deaktivieren	„{D}“
-	Startsignal Erfolgt	„{L}“
-	Keine Zeiten	„{K}“

Tabelle 4: Steurbefehle und Rückmeldungen über USB

4.2 USB Ansteuerung

Das Ansprechen der Treiber für die USB UART Bridge FT232R erfolgt über die objekt-orientierte Programmiersprache Visual Basic 6 (Grundkenntnisse durch „Visual Basic 6, echt einfach“ von Natascha Nicol und Ralf Albrecht). Es wurde ein kleines Programm erstellt „StoppUhrUSB“ für das Ansteuern des Messsystems, dessen Funktionsweise kurz erläutert werden soll (CD-Menü >> Quellcode).

Beispiele zur Ansteuerungen des Chips und die benötigten Module, Treiber und Konfigurationssoftware sind auf der Internetseite des Herstellers zum Download bereit gestellt (http://www.ftdichip.com/). Die im folgenden aufgelisteten Funktionen wurden einem Beispiel von FTDI Chips entnommen (CD-Menü >> Beispiel_FT_Chip).

Durch die Funktion „FT_OpenEx“ (siehe Tabelle 5) wird nach dem Chip in einer Endlosschleife gesucht. Gibt die Funktion über „ftstatus“ eine positive Rückmeldung werden anschließend die Parameter für die Verbindung über das UART mit dem Mikrocontroller gesetzt (siehe Tabelle 6 und 7). Die Datenübertragung kann nun beginnen.

Nachdem die Verbindung hergestellt ist wird in einer Endlosschleife abgerufen wie viele Zeichen in den Speicher des FT232R schon vom Mikrocontroller geschrieben

worden sind (siehe Tabelle 8). Stimmt der zurückgegebene Wert mit 3 überein wird die Zeichenkette zum Weiterverarbeiten abgeholt (siehe Tabelle 9) und an eine Prozedur zum Auswerten weitergereicht.

Funktion:	
ftstatus = FT_OpenEx(NAME, FT_OPEN_BY_DESCRIPTION, lngHandle)	
Wert:	**Parameter Beschreibung:**
Name	Eingespeicherter Name des Chips
FT_OPEN_BY_DESCRIPTION	FT232 über Namen identifizieren
lngHandle	Zahlenwert für Zugriff auf FT232R

Tabelle 5: FT_OpenEX

Funktion:	
ftstatus = FT_SetBaudRate(lngHandle, 115200)	
Wert:	**Parameter Beschreibung:**
115200	Baudrate (Verbindungsgeschwindigkeit)
lngHandle	Zahlenwert für Zugriff auf FT232R

Tabelle 6: FT_SetBaudRate

Funktion:	
ftstatus = FT_SetDataCharacteristics(lngHandle, FT_BITS_8, FT_STOP_BITS_1, FT_PARITY_NONE)	
Wert:	**Parameter Beschreibung:**
FT_BITS_8	Anzahl der BITS
FT_STOP_BITS_1	Art des Stop-BITs
FT_PARITY_NONE	Nicht invetiert
lngHandle	Zahlenwert für Zugriff auf FT232R

Tabelle 7: FT_SetDataCharacteristics

Funktion:	
ftstatus = FT_GetQueueStatus(lngHandle, Bytes_waiting)	
Wert:	**Parameter Beschreibung:**
Bytes_waiting	Anzahl gespeicherter Bytes im FT232R
lngHandle	Zahlenwert für Zugriff auf FT232R

Tabelle 8: FT_GetQueueStatus

Funktion:	
ftstatus = FT_Read(lngHandle, String_Empfang, 3, Gelesen)	
Wert:	**Parameter Beschreibung:**
String_Empfangen	Empfangener String
3	Anzahl der zu lesenden Zeichen
Gelesen	Anzahl der gelesenen Zeichen
lngHandle	Zahlenwert für Zugriff auf FT232R

Tabelle 9: FT_Read

Um Daten senden zu können ist lediglich nach Herstellung der Verbindung eine Funktion notwendig (siehe Tabelle 10)

Funktion:	
ftstatus = FT_Write(lngHandle, String_Schreiben 1, Written)	
Wert:	**Parameter Beschreibung:**
String_Schreiben	Zu sendender String
3	Anzahl der zu sendenden Zeichen
Written	Anzahl der gesendeten Zeichen
lngHandle	Zahlenwert für Zugriff auf FT232R

Tabelle 10: FT_Write

Der Programmverlauf von „StoppUhrUSB“ ist sehr einfach gehalten und wird in Abb. 18 als Blockschema dargestellt.

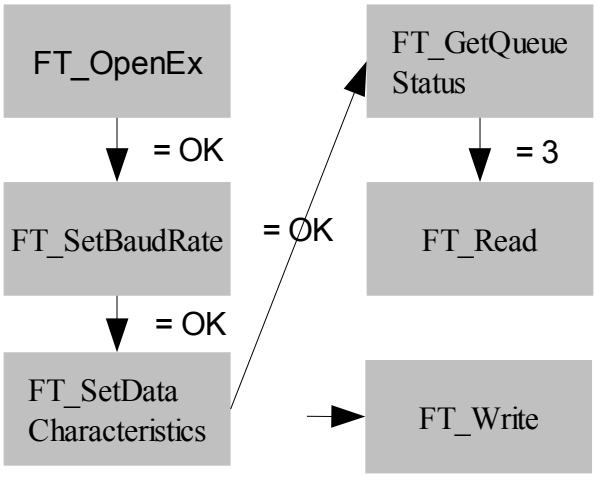

Abbildung 18: Programmverlauf von "StoppUhrUSB"

5 Fazit

Nach Erproben und Testen der Hardware und der Programme im Schulsport konnte sich das System als zuverlässig beweisen. Die Bedienung wurde durchwegs als einfach empfunden, ebenfalls die Installation der Lichtschranke. Auch wenn es viel Zeit, Mühe und Geld kostete den Apparat zu entwickeln, trägt er erheblich zur Entlastung bei Wettkämpfen bei und garantiert einen exakten und fehlerfreien Leistungsvergleich. Somit hat sich die Mühe gelohnt.

6 Anhang A: CD-Rom Inhaltsverzeichnis

1. Installation von StopUhrUSB
   - -Treiber
   - -PC-Software
2. Facharbeit
   - -Gliederung
   - -Ausführung
   - -Datenblätter
   - -Schaltplan
   - -Stückliste
   - -Platinen
   - -Gehäuse
   - -Quellcodes
   - -Photos
   - -BascomDemo
   - -FT_Mprog
   - -Beispiel_FT_Chip
   - -CD Inhalt

Info

Falls das CD-Menü nach dem einlegen der CD nicht automatisch öffnet kann es auch per Hand aufgerufen werden:

Pfad: CD-Rom\StoppUhrUSB_CD_Menu.exe

7 Literatur

Titel: AVR Risc Mikrocontroller 2.Auflage
Verfasser: Claus Kühnel
Verlag: Kühnel
Erscheinungsjahr: 2004
Ort: Altendorf
ISBN: 3907857046

Titel: Programming and customizing the AVR microcontroller
Verfasser: Dhananjay V. Gadre
Verlag: McGraw-Hill
Erscheinungsjahr: 2001
Ort: USA
ISBN: 007134666X

Titel: The Crystal Cookbook (in German)
Verfasser: Bernd Neubig & Wolfgang Briese
Verlag: Franzis´
Erscheinungsjahr: 1997
Ort: Deutschland
ISBN: 3772358535

Titel: Visual Basic 6 echt einfach
Verfasser: Natascha Nicol und Ralf Albrecht
Verlag: Franzis´
Erscheinungsjahr: 2000
Ort: Deutschland
ISBN: 3772374158

Application Note:

Titel: Empfängerbaustein für Infrarot (IR)-Fernsteuerungen
Verfasser: Thomas Richter and Karl Leahy
Verlag: OSRAM
Erscheinungsjahr: 1999
URL: http://fl.hw.cz/data_ic/sfh5110_applicat.pdf
Aufrufdatum: 2.5.2006

Datenblätter:

Titel: Atmega8 (Datenblatt)
Verfasser: Atmel
Erscheinungsjahr: 2006
URL: http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf
Aufrufdatum: 2.5.2006

Titel: EA DIP S082 (Datenblatt)
Verfasser: Electronic Assembly
Erscheinungsjahr: -
URL: http://www.lcd-module.de/deu/pdf/doma/dips082.pdf
Aufrufdatum: 2.5.2006

Titel: FT232R USB UART I.C. (Datenblatt)
Verfasser: FTDI Chip
Erscheinungsjahr: 2005
URL: http://www.ftdichip.com/Documents/DataSheets/DS_FT232R_v104.pdf
Aufrufdatum: 2.5.2006

Titel: KS0066U (Datenblatt)
Verfasser: Samsung Electronic
Erscheinungsjahr: -
URL: http://www.tranzistoare.ro/datasheets2/55/553815_1.pdf
Aufrufdatum: 2.5.2006

Titel: LD 274 (Datenblatt)
Verfasser: Osram
Erscheinungsjahr: -
URL:
http://www.reichelt.de/?SID=27LPnjrKwQARsAAEmxYZseff644fa383c0b38cbec031e9fca056c;ACTION=7;LA=6;OPEN=1;INDEX=0;FILENAME=A500%252FLD274%2523OSR.pdf
Aufrufdatum: 2.5.2006

Titel: NE555 (Datenblatt)
Verfasser: Texas Instruments
Erscheinungsjahr: 1998
URL: http://www.ortodoxism.ro/datasheets/texasinstruments/ne555.pdf
Aufrufdatum: 2.5.2006

Titel: Tsop17.. (Datenblatt)
Verfasser: Vishay Telefunken
Erscheinungsjahr: -
URL:
http://www.reichelt.de/?SID=27LPnjrKwQARsAAEmxYZseff644fa383c0b38cbec031e9fca056c;ACTION=7;LA=6;OPEN=1;INDEX=0;FILENAME=A500%252FTSOP1733_TSOP1737_TSOP1740%2523VIS.pdf
Aufrufdatum: 2.5.2006

Erklärung:

Ich erkläre hiermit, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe.

_____,den_____ _____

Ort Datum Unterschrift