
Praktikum Schaltungstechnik NT8

Erstellung eines Schaltplans für das Projekt "Fun for disabled People (FfdP)"

im WS 2006/07

Durchführung:

Uhl, Michael

Betreuer:

Prof. Dr. Rudolf Lindner

Dipl. Math. (U) Dipl. Ing. (FH) Peter Wiegner

Inhaltsverzeichnis

1	RAHMENBEDINGUNGEN.....	3
2	ÜBERSICHTSPLAN.....	4
3	SPIELBRETT.....	8
4	SPIELBRETTSTEUERUNG.....	10
5	VERBINDUNG PC.....	13
6	PROZESSOR.....	19
7	PLOTTER-TREIBER.....	24
8	PLOTTERSTEUERUNG.....	27
9	ANHANG.....	29
9.1	BAUTEILELISTE.....	29
9.2	NETZLISTE.....	30
10	QUELLENVERZEICHNIS.....	31

1 Rahmenbedingungen

Im Praktikum Schaltungstechnik NT8 wird das Projekt "Fun for disabled People (FfdP)" durchgeführt. Hierzu war es notwendig einen Schaltplan für die Leiterplatte, die für die Steuerung der Spielabläufe zuständig ist, zu erstellen.

Der Schaltplan wurde mit dem Programm Orcad Capture erstellt. Das Programm ermöglicht Schaltpläne hierarchisch nach dem Top-down-Prinzip (engl., von Oben nach Unten) zu erstellen. Es können somit bereits bei der Schaltplanerstellung die einzelnen Schichten (Anfangen von der Anwenderschicht (Top): wie und mit welchen Kabeln muss die gegebene Hardware verbunden werden? - Bis zur untersten Schicht (Bottom): wie werden die Bauelemente auf der Leiterplatte miteinander verdrahtet?) dargestellt werden. Dieses Verfahren dient der besseren Verständlichkeit beim analysieren von Schaltungen, da es das menschliche Schichtendenken unterstützt.

Die vorliegende Dokumentation versucht ebenfalls das Top-down-Prinzip zu verwirklichen und korrespondiert mit dem Schaltplan (schaltplan.opj¹ bzw. schaltplan-rev1.5.pdf²).

2 Übersichtsplan

Die oberste Schicht (Anwenderschicht) wird im Übersichtsplan dargestellt. Darin sind sämtliche Komponenten dargestellt, die aus Sicht des Anwenders nötig sind (PC, Spielbrett, Programmiergerät und Netzteil) und wie er diese miteinander zur verbinden hat.

Der PC, das Programmiergerät und das Netzteil sind dabei vorhandene Komponenten, die bei der Schaltplanerstellung nicht weiter berücksichtigt werden und deshalb keinen hierarchische Block darstellen.

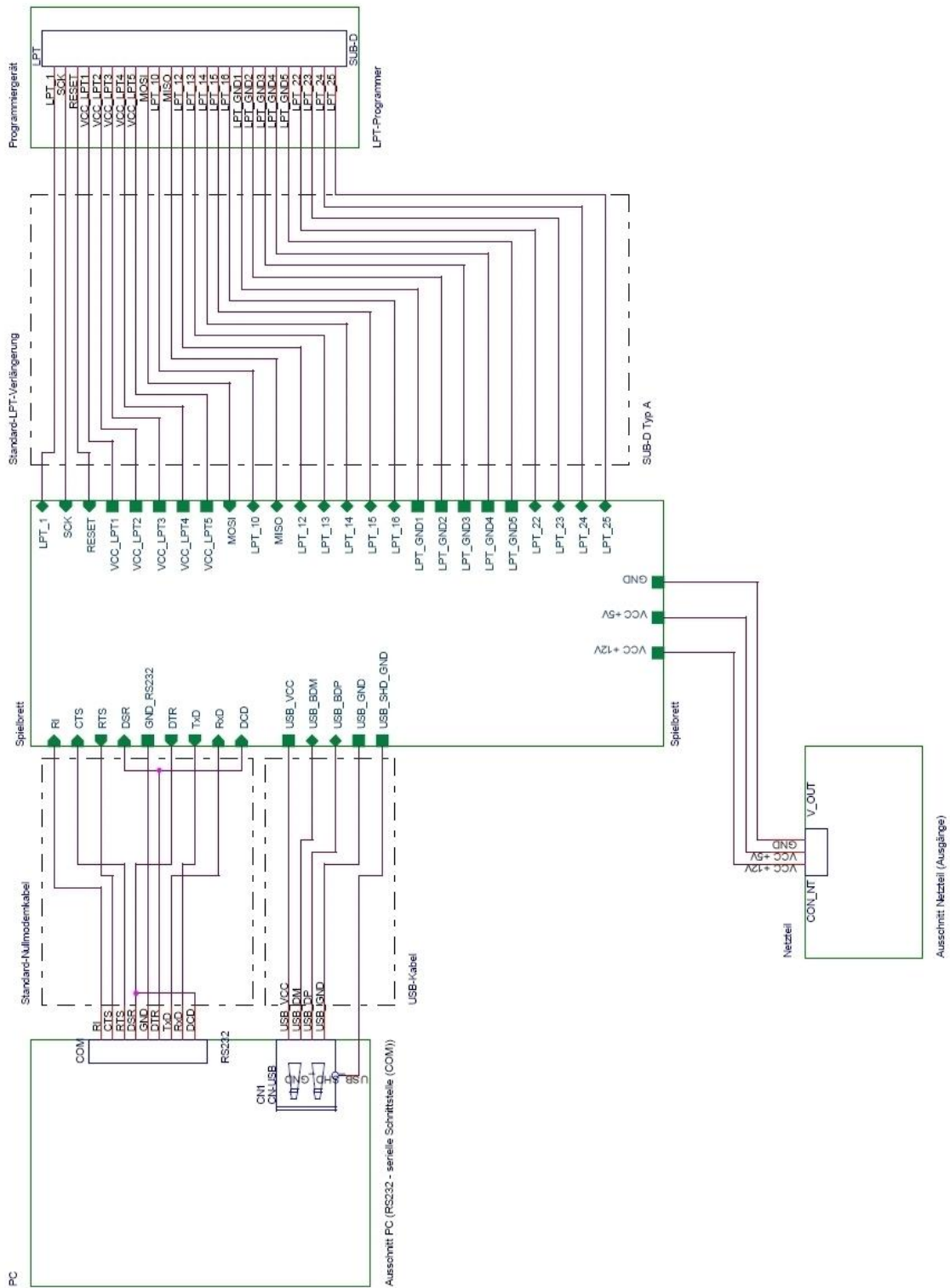
Für den PC wurde die RS232- (COM) und die USB-Schnittstelle dargestellt. In der eigentlichen Spezifikation des Projektes war ausschließlich die RS232-Schnittstelle vorgesehen. Da diese bei neueren Computern jedoch kaum mehr auftritt wurde zusätzlich die USB-Schnittstelle realisiert.

Das Netzteil versorgt die entwickelten Komponenten (enthalten im Block Spielbrett) mit den nötigen Versorgungsspannungen.

Das Programmiergerät wurde im Übersichtsplan als einzelne Komponente dargestellt, da es dem Anwender überlassen bleibt, ob er zur Programmierung der Steuerung ein separates Programmiergerät oder die LPT-Schnittstelle des PCs benutzt. Die Schnittstelle wurde für beide Anwendungsfälle ausgeführt (Standard SUB-D Typ A).

Um die generelle Kostenrelevanz zu berücksichtigen wurden sämtliche externe Schnittstellen durch Standardschnittstellen ausgeführt. Somit können zum einen auf Seiten des PCs die gebräuchlichen Schnittstellen (COM für RS232, USB für USB und LPT für die Programmierung) benutzt werden und zum anderen gängige, und für den Anwender leicht zu beschaffende, Verbindungskabel genutzt werden.

Die Pins der jeweiligen Schnittstellen wurden zur besseren Verständlichkeit im gesamten Schaltplan nach dem jeweiligen Standard (RS232, SUB-D Typ A, USB) bezeichnet.



- Abbildung 1: Übersichtsplan -

Die Pins der RS232-Schnittstelle (bezeichnet als COM-Schnittstelle am PC, verbunden über ein Standard-Nullmodemkabel) sind nach EIA-232 Norm folgendermaßen spezifiziert:

PIN	Bezeichnung	Kommunikationsrichtung	Name
1	DCD	IN	Data Carrier Detected
2	RxD	IN	Receive Data
3	TxD	OUT	Transmit Data
4	DTR	OUT	Data Terminal Ready
5	GND		Ground
6	DSR	IN	Dataset Ready
7	RTS	OUT	Request to Send
8	CTS	IN	Clear to Send
9	RI	IN	Ring Indicator

Zu berücksichtigen ist, das innerhalb des Standard-Nullmodemkabels die RxD- und TxD-Signale miteinander gekreuzt sind.

Das Programmiergerät (bzw. der PC via LPT-Schnittstelle) ist über ein 25-poliges Standard-LPT-Verlängerungskabel mit der Steuerung verbunden. Die LPT-Schnittstelle (spezifiziert in IEEE 1284) definiert eine parallele Schnittstelle zur bidirektionalen Übertragung von Daten zwischen PCs und verschiedenen Peripheriegeräten (offizielle Definition). Nachdem die Definition der Pins in der Spezifikation bereits bewusst zur (individuellen) Auslegung offen gehalten wurde, wurden die erforderlichen Pins für die SPI-Programmierung des ATMEGA16 nach der Pinbelegung des SPI-Programmiersports des ATMEGA16 benannt. Somit ergibt sich für die Programmierschnittstelle folgende Pinbelegung:

PIN	Bezeichnung	Kommunikationsrichtung	Name
1	LPT_1	BI	LPT1 (NC)
2	SCK	IN	SPI Bus Serial Clock
3	RESET	IN	Reset
4	VCC_LPT1		Versorgungsspannung LPT
5	VCC_LPT2		Versorgungsspannung LPT
6	VCC_LPT3		Versorgungsspannung LPT
7	VCC_LPT4		Versorgungsspannung LPT
8	VCC_LPT5		Versorgungsspannung LPT
9	MOSI	IN	SPI Bus Master Output/Slave Input)
10	LPT_10	BI	LPT10 (NC)
11	MISO	BI	SPI Bus Master Input/Slave Output
12	LPT_12	BI	LPT12 (NC)
13	LPT_13	BI	LPT13 (NC)
14	LPT_14	BI	LPT14 (NC)
15	LPT_15	BI	LPT15 (NC)
16	LPT_16	BI	LPT16 (NC)
17	LPT_GND1		Masse LPT
18	LPT_GND2		Masse LPT

19	LPT_GND3	I	Masse LPT
20	LPT_GND4		Masse LPT
21	LPT_GND5		Masse LPT
22	LPT_22	BI	LPT22 (NC)
23	LPT_23	BI	LPT23 (NC)
24	LPT_24	BI	LPT24 (NC)
25	LPT_25	BI	LPT25 (NC)

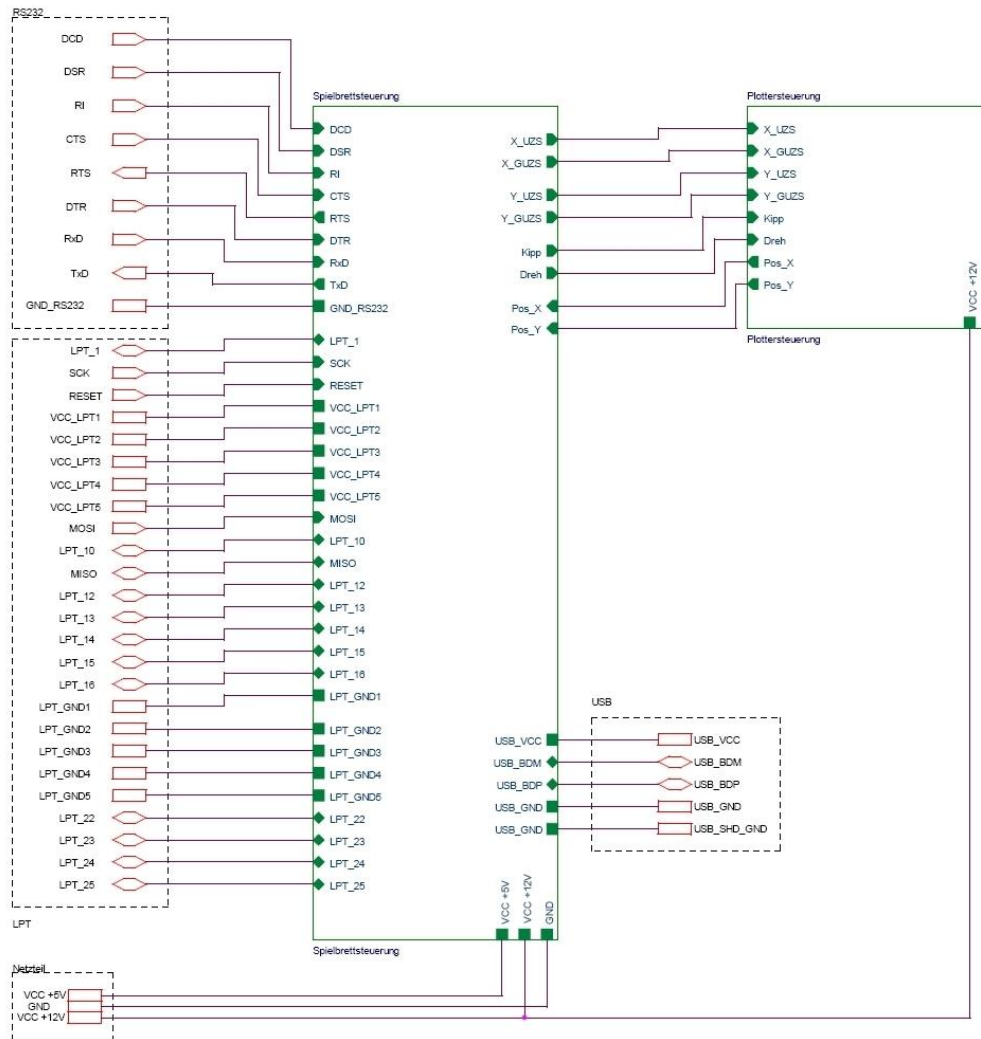
(NC) bezeichnet nicht benötigte (im weiteren Verlauf nicht verbundene, engl. not connectet) Pins. Die Pins 2, 3, 9 11 werden explizit für die SPI-Programmierung des ATMEGA16 benötigt und werden daher bereits in dieser Schicht mit der jeweiligen Bezeichnung genannt. Die Schnittstelle wurde nicht nur aufgrund der bereits erwähnenden Kosten- und Anwenderrelevanz als 25-polige LPT-Schnittstelle ausgeführt, sondern auch unter Berücksichtigung der dadurch bestehenden Möglichkeit der vereinfachten (späteren) Nutzung der JTAG-Schnittstelle des ATMEGA16 (Ermöglichung von Debugging on Chip).

Das gezeichnete Netzteil versorgt die Steuerung mit den Versorgungsspannungen +12VDC und +5VDC und stellt die Masseverbindung her.

3 Spielbrett

Die nächste Schicht (dargestellt durch den hierarchischen Block Spielbrett → Descend Hierarchy (absteigende Hierarchie)) stellt bereits eine Ebene (die oberste) für den Techniker dar. Darin ist bereits eine Unterteilung auf der Leiterplatte nach Funktionsgruppen vorgenommen. Zum einen wird das Spielbrett durch die Spielbrettsteuerung, welche die eigentliche Funktion der Steuerung enthält, dargestellt und zum anderen durch die Plottersteuerung, die die Ausgangssignale für die Motoren des Spielbretts einerseits und Konstantstromquellen für die inkrementelle Wegmessung als Eingangssignale für die Steuerung andererseits enthält.

Wie bereits erwähnt, wurden die Signalbezeichnungen der externen Signale beibehalten.



- Abbildung 2: Spielbrett -

Eingeführt wurden an dieser Stelle die Signale für die funktionale Unterteilung in Spielbrett- und Plottersteuerung:

PIN	Bezeichnung	Kommunikationsrichtung	Name
1	X_UZS	OUT	Signal für die positive Drehrichtung des X-Motors
2	X_GUZS	OUT	Signal für die negative Drehrichtung des X-Motors
3	Y_UZS	OUT	Signal für die positive Drehrichtung des Y-Motors
4	Y_UZS	OUT	Signal für die negative Drehrichtung des Y-Motors
6	Kipp	OUT	Signal für den Servomotor des Kippens der Spielfigur
7	Dreh	OUT	Signal für den Servomotor des Drehens der Spielfigur
8	Pos_X	IN	Signal für die Positionsbestimmung des X-Motors
9	Pos_Y	IN	Signal für die Positionsbestimmung des X-Motors

Die Kommunikationsrichtung bezieht sich auf die Spielbrettsteuerung.

Auf die darunter gelegenen Ebenen Spielbrettsteuerung und Plottersteuerung wird in den Abschnitten 4 bzw. 8 detailliert eingegangen.

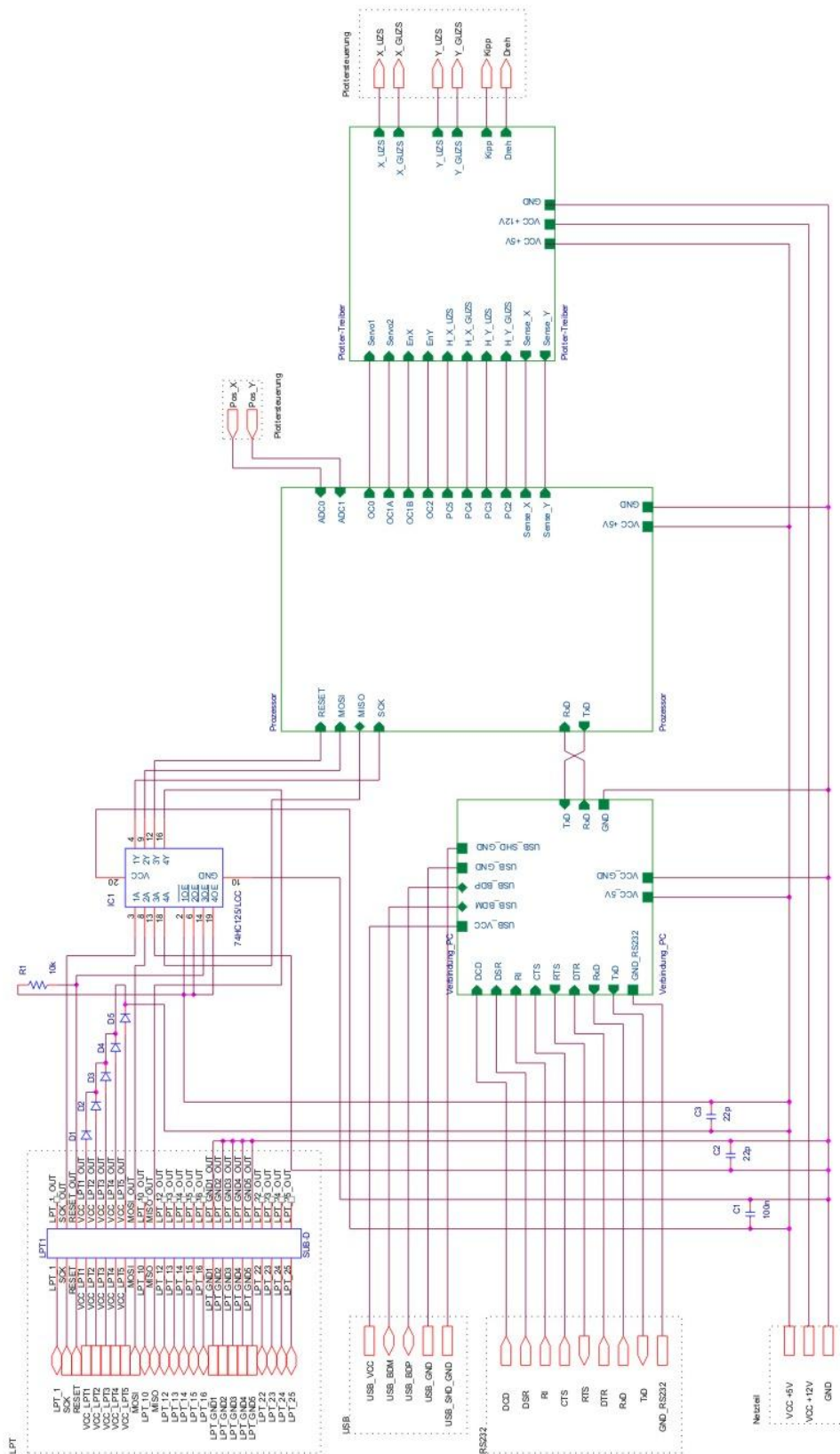
4 Spielbrettsteuerung

Die Ebene Spielbrettsteuerung stellt neben der funktionsgerechten Unterteilung des Funktionsblockes erstmalig die elektrotechnische Realisierung dar.

Die Signale der (externen) LPT-Schnittstelle (Programmierschnittstelle) werden darin über die Sockelleiste LPT1 mit dem IC1 (74HC125/LCC) zum Prozessor verbunden. Der 3-State Quad -Buffer / Line Driver 74HC125/LCC³ dient zur Anpassung der über die parallele Schnittstelle gesendeten Daten an den Eingang des SPI-Ports des ATmega16⁴. Die nicht (in diesem Fall) benötigten Eingänge am IC1 wurden zwecks ihrer Beschreibung auf Masse bzw. auf +5VCC gelegt. Um parasitäre Effekte mit der externen Verbindung zum PC zu vermeiden bzw. um die Spannung am IC zu stabilisieren wurden Koppel- bzw. Abblockkondensatoren verwendet. Um den Strom zu begrenzen wurde zum Pfad des Reset-Eingangs Pin 14 ein Widerstand (R1) parallel geschaltet.

Um die Spannungsversorgung über die LPT-Schnittstelle zur Verfügung zu stellen wurden die VCC_LPTx-Pins mit der Versorgungsspannung +5V verbunden. Um den Schutz vor Überspannungen zu gewährleisten wurden die Freilaufdioden D1-D5 benutzt.

Die Spielbrettsteuerung enthält die funktionalen Blöcken Plotter-Treiber, der die Ausgangssignale für die Plottersteuerung bereitstellt, den Prozessor als Kernstück der Spielbrettsteuerung und den Block „Verbindung_PC“. Dieser empfängt die Signale der RS232- und der USB-Schnittstelle des PCs und gibt diese an den Prozessor weiter.



- Abbildung 3: Spielbrettsteuerung -

Als interne Signale wurden eingeführt zwischen Prozessor und Plottersteuerung:

PIN	Bezeichnung	Kommunikationsrichtung	Name
1	ADC0 / Pos_X	IN	AD-Wandler PIN0 (40)
2	ADC1 / Pos_Y	IN	AD-Wandler PIN1 (39)

Zwischen dem Prozessor und Plotter-Treiber:

PIN	Bezeichnung	Kommunikationsrichtung	Name
3	OC0 / Servo1	OUT	Signal für den Servomotor Kippen
4	OC1A/Servo2	OUT	Signal für den Servomotor Drehen
5	OC1B/EnX	OUT	Signal für die Durchschaltung des Signals für den X-Motor
6	OC2/EnY	OUT	Signal für die Durchschaltung des Signals für den Y-Motor
7	PC5/H_X_UZS	OUT	Signal für die positive Drehrichtung des X-Motors
8	PC4/H_X_GUZS	OUT	Signal für die negative Drehrichtung des X-Motors
9	PC3/H_Y_UZS	OIT	Signal für die positive Drehrichtung des Y-Motors
10	PC2/H_Y_GUZS	OUT	Signal für die negative Drehrichtung des Y-Motors
11	Sense_X	IN	Signal für den Überstromschutz des Motors in X-Richtung
12	Sense_X	IN	Signal für den Überstromschutz des Motors in Y-Richtung

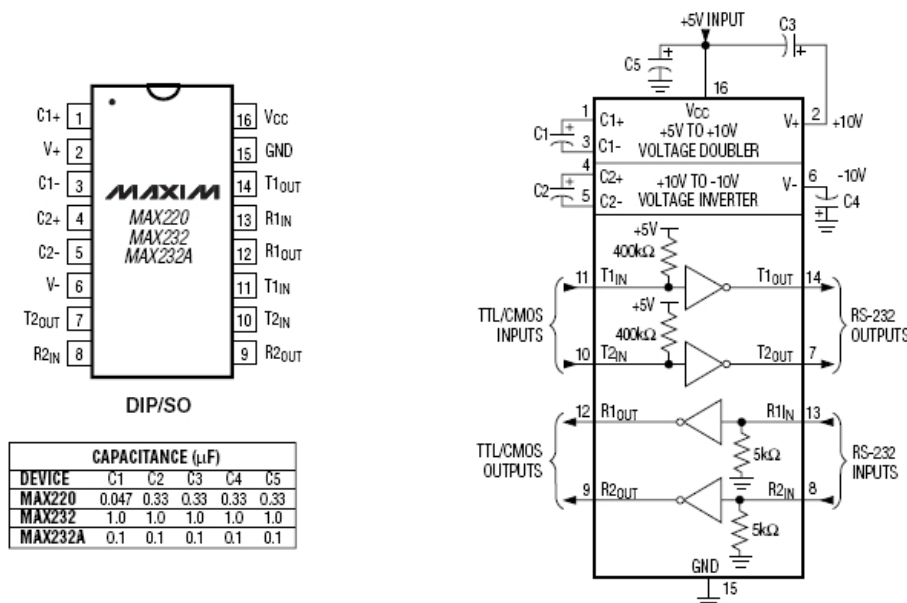
Die Kommunikationsrichtung bezieht sich in diesem Fall auf den Prozessor.

5 Verbindung PC

Der Block „Verbindung_PC“ stellt die funktionelle Schnittstelle zwischen dem PC (RS232, USB) und dem Prozessor (USART) zur Datenkommunikation her. Aufgrund des Umfangs, der durch die gleichzeitige Implementierung von RS232- und USB-Schnittstelle entstand, und der besseren Lesbarkeit wurde die Funktion in einem eigenen hierarchischen Block beschrieben. Des weiteren kann so bereits vor der Layouterstellung der Leiterplatte kenntlich gemacht werden, dass es sich dabei um eine (in sich geschlossene) Funktion handelt, bei der es zu berücksichtigen gilt, die Bauelemente beim späteren Layout möglichst nahe aneinander anzuordnen um lange Leitungswege zu vermeiden (Bauelementeplatzierung nach Funktion).

Als Eingangssignale stehen die bereits beschriebenen Schnittstellensignale der RS232- und USB-Schnittstelle zur Verfügung. Die Ausgangssignale zum Prozessor stellen die Signale TxD, RxD und GND auf der rechten Seite der Zeichnung dar.

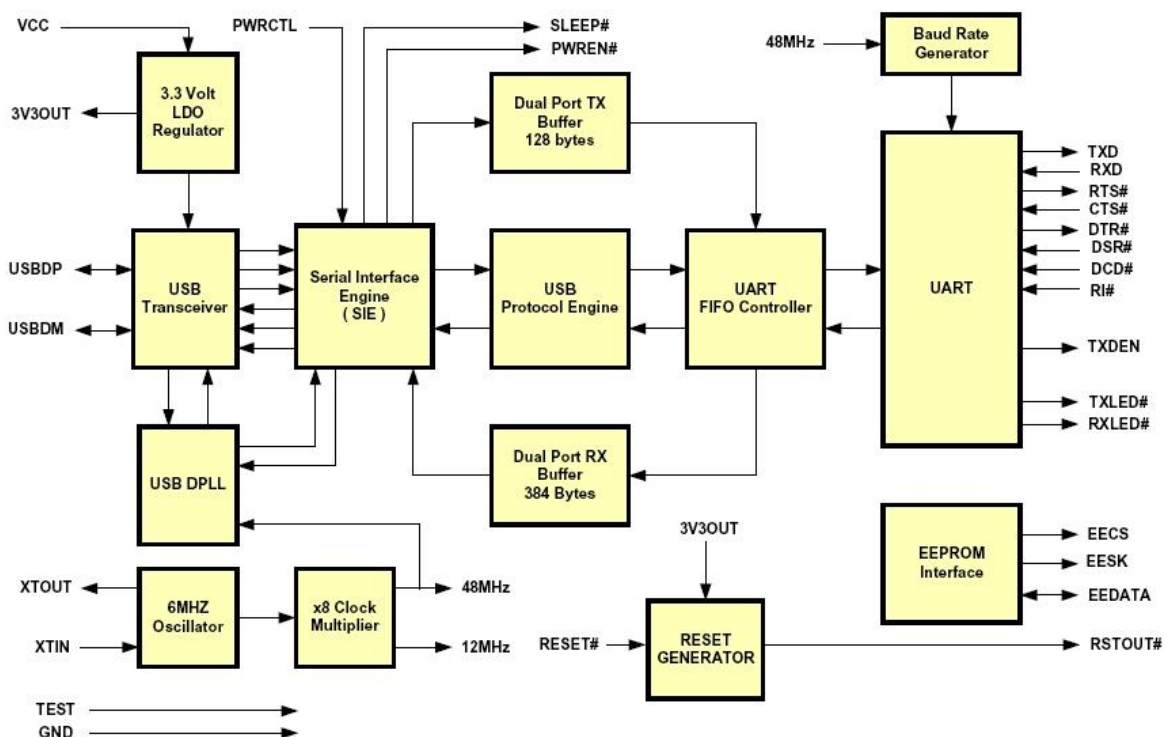
Für eine funktionierende RS232-Datenkommunikation zwischen dem PC und dem Prozessor ATmega16 werden von der standardisierten 9-poligen RS232-Schnittstelle allerdings nur die Signale RxD, TxD und GND benötigt. Die restlichen Signale werden durch die verwendete Sockelleiste (CON9_4 (RS232)) nicht verdrahtet und demnach nicht als (Ausgangs-) Pin nach außen geführt. Da sich die Signalpegel von der RS232-Schnittstelle des PCs unter Umständen sehr stark von den benötigten Pegeln des ATmega16 unterscheiden können (der Low-Pegel muss sich für die USART-Schnittstelle zwischen -0,5V und 1,5V ($0.3 \cdot V_{CC}$) befinden - der High-Pegel zwischen 3,5V ($0.7 \cdot V_{CC}$) und 5,5V ($V_{CC} + 0.5V$)) wurde das IC2 (MAX232) verwendet. Der Multichannel RS232 Driver/Receiver MAX232⁵ dient der Pegelanpassung zwischen der RS232-Schnittstelle des PCs und der USART-Schnittstelle des ATmega16.



- Abbildung 4: Pinbelegung MAX232 -

Abbildung 4 zeigt die Pinbelegung des ICs. Der Spannungsverdoppler (Pin 1, 2, 3) und der Spannungsinverter (Pin 4, 5, 6) für die Versorgungsspannung werden im vorliegenden Fall nicht benutzt und daher nach Datenblatt jeweils über einen Abblockkondensator miteinander verschalten. Das Ausgangssignal TxD vom PC wurde auf den Eingang R1IN (8) - das Eingangssignal RxD zum PC auf den Ausgang T1OUT (14) von IC2 geschaltet. Das (spätere) Signal TxD zum USART-Pin TxD des ATmega16 liefert der Ausgangspin R1OUT(12) - das Signal RxD vom ATmega16 wird mit dem Eingang T1IN (10) des IC2 verschalten.

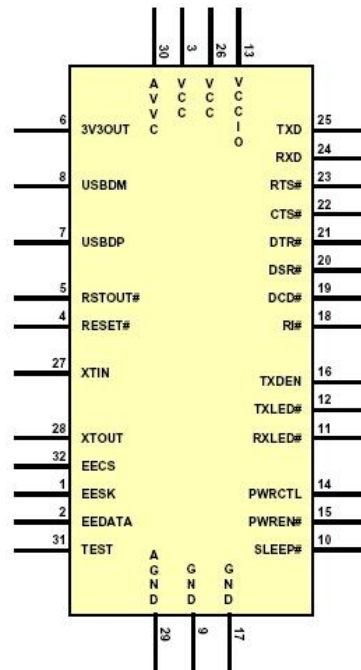
Die Verbindung der USB-Schnittstelle zum PC wird in der Schaltung durch den Stecker CN2 (CN-USB) hergestellt. Da der verwendete ATmega16 Mikrokontroller nicht direkt USB unterstützt, wurde das IC3 (FT232BM)⁶ als „USB zu RS232 Konverter“ eingesetzt.



- Abbildung 5: Blockdiagramm FT232BM -

Wie bereits im Blockdiagramm des IC auffällt, wird aufgrund den hohen Datenraten bei der USB-Kommunikation eine hohe Synchronität zwischen den ein und ausgehenden Daten gefordert. Aus diesem Grund wurde der 6MHz-Quarz Y1 zur Takterzeugung für den USB zu RS232 Konverter eingeführt. Der Quarz ist zwischen XTIN (27) und XTOUT (28) geschaltet und wird parallel mit einem Abblockkondensator (C16) auf Masse gezogen. Die Eingänge der Versorgungsspannungen (AVCC (30), VCC (3, 26), VCC-IO (13) und PWRCTL (14)) und der Masseverbindung (AGND (29), GND (9, 17)) werden über Koppelkondensatorschaltungen (Parallelschaltung aus jeweils einem großen

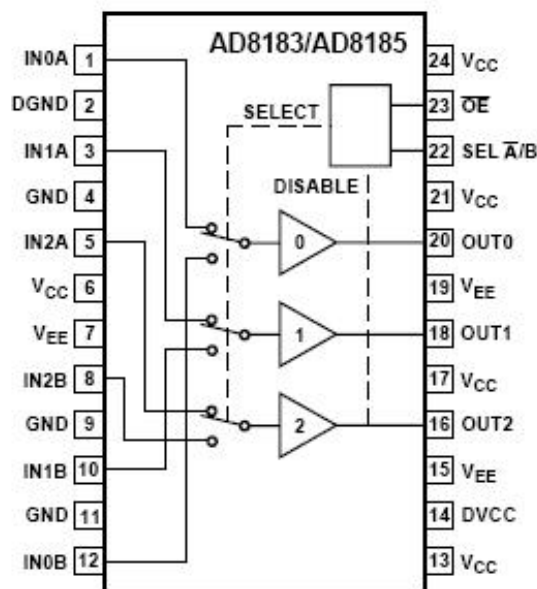
Kondensatorkopf (C12, C17 und C29 - Polung beachten) und zweier 100nF Kondensatoren (zusammen 200nF, leichtere Beschaffbarkeit) zum jeweiligen Potential verbunden.



- Abbildung 6: Pinbelegung FT232BM -

Das Signal 1_USB_OUT stellt die Versorgungsspannung der USB-Schnittstelle und das Reset für das IC3 dar und wird mit Masse verbunden. Als Schutz vor Überspannungen wurde die Freilaufdiode D6 eingeführt. Der Spannungsteiler aus den Widerständen R5 und R6 dient für das Reset-Signal von IC3 (ausgehend von 5V USB_V_{CC} stellt sich somit am Reseteingang eine Spannung von 3V ein). Die Leitungsschirmung (USB_SHD_GND) und die Masse der USB-Verbindung USB_GND werden über den Abblockkondensator C15 mit Masse verbunden. Um die Spannung am Eingang AVCC (30, für den internen „x8-clock-multiplier“ des ICs) besonders vor Störungen zu schützen wurde die Siebschaltung aus R2 und C10 verwendet. Darüber hinaus wurden noch die Eingänge SLEEP (10) und TEST (31) auf Masse gezogen um den „normalen“ Modus ohne externen „USB - RS232 Pegelkonverter“ des IC3 zu erreichen. Das Datensignal USB_BDM (vom PC, 2_USB_OUT an der Buchse) wurde über den Widerstand R3 mit dem USB_BDM-Pin (8) des ICs verbunden. Der 3V3OUT-Pin (darüber werden IC-intern 3,3V bereitgestellt) ist über einem Abblockkondensator (C11) nach Masse verbunden. Das Datensignal USB_BDP (vom PC, 3_USB_OUT an der Buchse) wurde ebenfalls über einen 27 Ohm Widerstand (R4) mit dem USBDP-Pin des ICs (7) verbunden. Da auf den USB-Bus kein Strom vom IC eingespeist werden darf wenn der USB-Host des PCs Daten sendet, ist gleichzeitig mit dem USBDP-Pin noch der Ausgang RSTOUT (5) (interner Reset Generator des ICs) über den Pull-Up-Widerstand R5 verbunden.

Die beiden ICs IC2 (Pegelkonvertierung RS232) und IC3 (USB-RS232 Konverter) senden bzw. empfangen beide die Datensignale TxD bzw. RxD für den ATmega16. Der Mikrokontroller kann jedoch gleichzeitig nur die Signale einer Datenverbindung (RS232 oder USB) über seine USART-Schnittstelle verarbeiten. Da es kaum möglich ist die Auswahl der jeweiligen Datenverbindung über einen mechanischen Mehrfachschalter zu realisieren (Entprellung, Einhaltung der Taktzyklen, Einhaltung der Pegel, usw.) wurde der „Triple 2:1 Multiplexer“ AD8183⁷ (IC4) eingeführt. Um die Auswahl der jeweiligen Datenverbindung zu treffen wurde der Auswahlschalter S1 und als Anzeigeelement hierzu die Leuchtdiode D-USB-Active (leuchtet, falls die USB-Verbindung ausgewählt ist) verwendet.

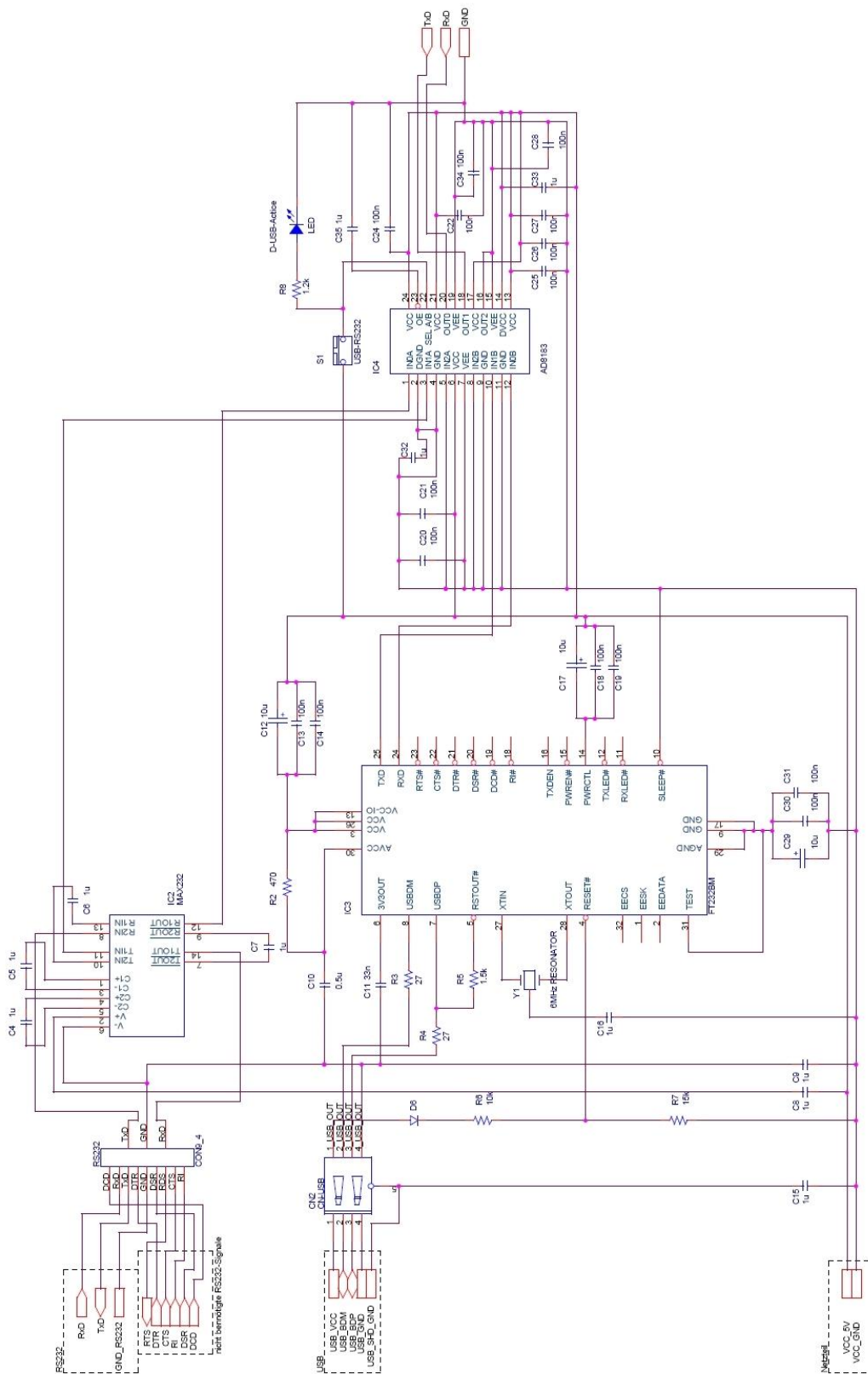


SEL \bar{A}/\bar{B}	\bar{OE}	OUT
0	0	INA
1	0	INB
0	1	High Z
1	1	High Z

- Abbildung 7: Pinbelegung AD8183 -

Der TxD-Ausgang von IC3 wurde mit dem Pin IN1B (10) verbunden - der RxD-Eingang mit dem Pin IN0B (12) des Multiplexers. Die Verbindung des IC2 geschah entsprechend. Allerdings mussten an dieser Stelle RxD und TxD vertauscht werden, da RxD und TxD auf der RS232-Leitung der externen Verbindung gekreuzt sind (TxD → IN0A (1), RxD ← IN1A (3)). Die Pins OUT0 (20) (RxD) und OUT1 (18) stellen die Verbindung zur UART-Schnittstelle des ATmega16 her. Die Steuersignale werden

über die Pins SEL A/B (22) und OE (23) an den Multiplexer herangeführt. Wie in der Wahrheitstabelle in Abbildung 7 zu erkennen ist, muss das Signal OE für die Erfüllung der Funktionalität Low sein. Daher wurde es über C35 auf Masse gelegt. Die Auswahl erfolgt also über das Signal SEL A/B, welches mittels des Schalters S1 bereitgestellt wird. Da der Baustein drei verschiedene Versorgungsspannungen und zwei verschiedene Bezugspotentiale benötigt, mussten neben den „normalen“ Koppelkondensatoren zwischen VCC und GND noch Koppelkondensatoren zwischen GND und DGND (2) (digitale Masse), VEE (7, 15, 19) (beträgt eigentlich -VCC, da jedoch nur binäre Ein- und Ausgangssignale vorliegen wurde VEE ebenfalls mit dem Bezugspotential verbunden) und GND und zwischen DVCC (14) (digitales VCC, hier ebenfalls +5V) und VCC eingefügt werden.



- Abbildung 8: Verbindung PC -

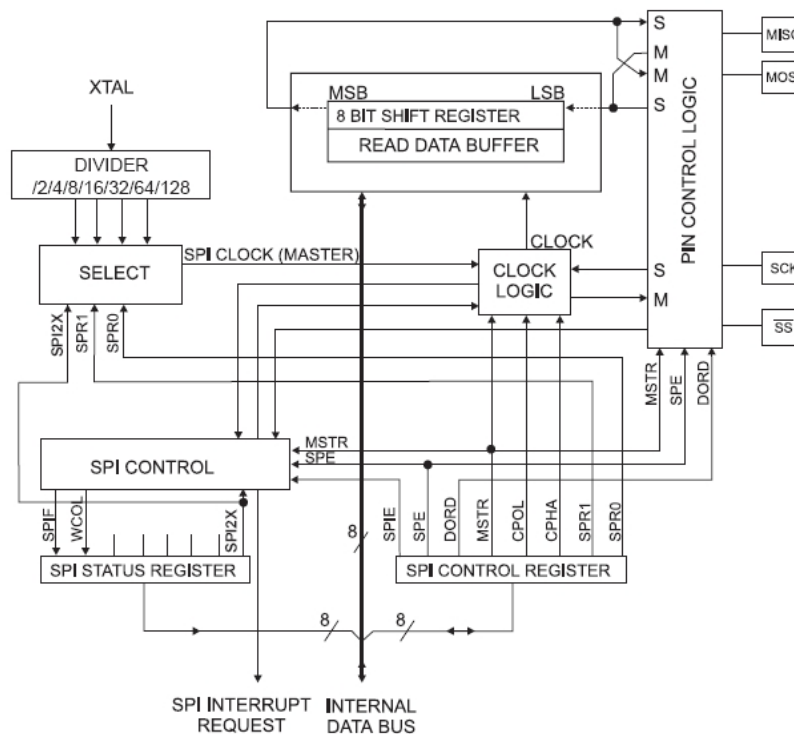
6 Prozessor

Die hierarchische Ebene Prozessor stellt mit dem Mikrocontroller ATmega16 das Kernstück der Steuerung dar.

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

- Abbildung 9: Pinbelegung ATmega16 -

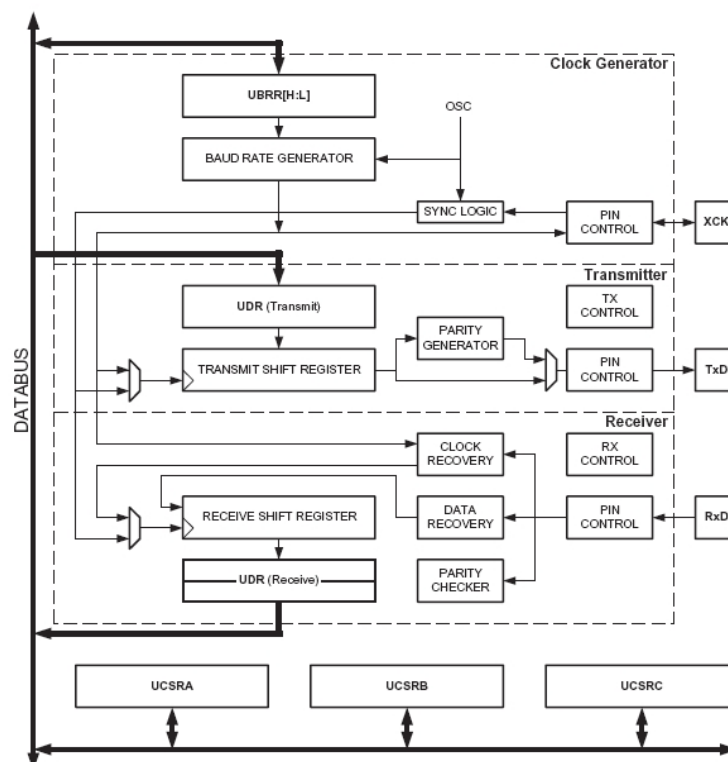
In dieser Ebene werden die Signale der Programmierschnittstelle (RESET, MOSI, MISO und SCK) auf den jeweiligen zur SPI-Schnittstelle gehörenden Pin des Controllers geführt.



- Abbildung 10: Blockdiagramm SPI -

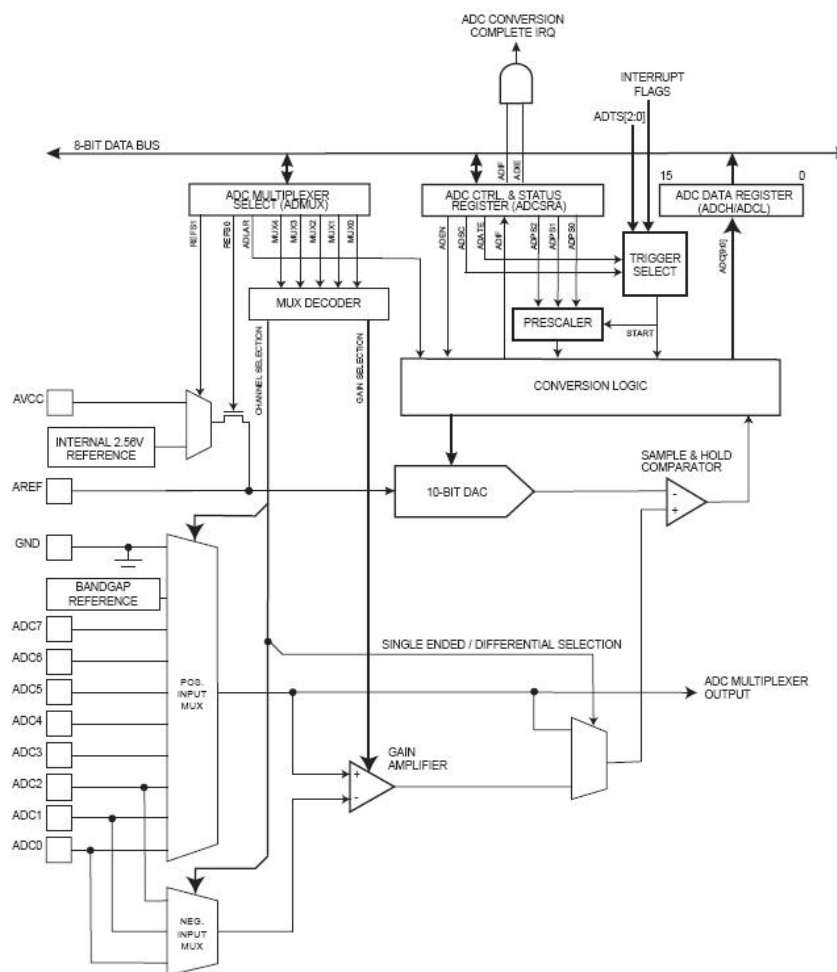
Das Signal SS (5) dient zur Master/Slave Synchronisation die an dieser Stelle nicht benötigt wird. Demnach sind nur die Pins MOSI (6), MISO (7) und SCK (8) verdrahtet. Um das Nebensprechen zwischen den Signalleitungen MISO und SCK zu verhindern, was zur fehlerhaften Programmierung führen kann, wurde zusätzlich eine Siebschaltung mit R16 und C12 realisiert. Während der Programmierung soll das (vorhandene) Programm auf dem Mikrocontroller nicht ausgeführt werden bzw. sofort nach der Programmierung vollständig in den Speicher geladen werden und abgearbeitet werden. Hierzu wird das Reset-Signal der Programmierschnittstelle auf den Reset-Eingang (9) des ATmega16 geführt. Um ein Reset durch den Benutzer zu erlauben wurde mit dem Taster S2 ein Reset-Taster eingeführt. Um den Strom zu begrenzen ist der Widerstand R15 vorgesehen. Zur Entkopplung wurde der Kondensator C42 zwischen Masse und Signalleitung geschaltet.

Um die geforderte Synchronität bei RS232- (USB-) Datenübertragung zu gewährleisten wird der Mikrocontroller nicht, wie standardmäßig ausgeliefert, mit dem internen Oszillator zur Taktgewinnung betrieben, sondern über den externen Quarz Y2. Hiermit wird ein genauere Systemtakt (CLK) erzielt. Somit lässt sich die Baudrate der USART-Schnittstelle für die „Echtzeitanwendung“ besser kalibrieren. Der Quarz ist an die XTAL1 (13) und XTAL2 (12) angeschlossen. Zur Erhöhung der Güte ist der hochohmige Widerstand R17 parallel zum Quarz geschaltet. Zum Abblocken von Störungen auf der Masseleitung sind die Kondensatoren C43 und C44 eingeführt worden. Die Datenverbindung zum PC ist über die Pins RxD (14) und TxD (15) der USART-Schnittstelle realisiert.



- Abbildung 11: Blockdiagramm USART -

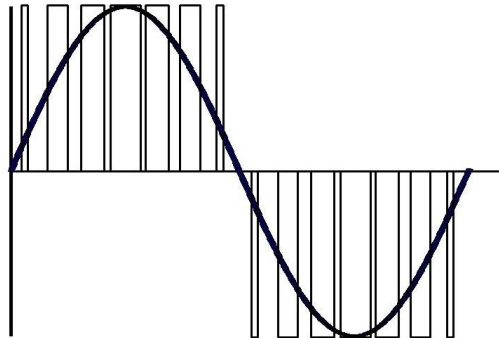
Die auf der rechten Seite des Schaltplans dargestellten Off-Page Connectors stellen zum einen die Schnittstelle zur Plottersteuerung (funktionaler Block) (ADC0, ADC1) und zum anderen die Schnittstelle zur funktionalen Gliederung des Plotter-Treibers (OC0, OC1A, OC1B, OC2, PC5, PC4, PC4, PC2 als Ausgänge und Sense_X, Sense_Y als Eingänge) dar. Die beiden Signale ADC0 und ADC1 von der Plottersteuerung dienen zur inkrementellen Positionsbestimmung der beiden Motoren (X- und Y-Richtung). Als Eingangssignale stehen dem Prozessor hierbei zwei analoge Signale zur Verfügung. Der Mikrocontroller ATmega16 stellt intern bereits einen AD-Wandler bereit. Daher werden die beiden analogen Signale von der Plottersteuerung auf die Eingangspins ADC0 (40) und ADC1 (39) des Mikrocontrollers gelegt.



- Abbildung 12: Blockdiagramm ADC -

Um die Motoren in X- und Y-Richtung steuern zu können werden die Signale OC1B, OC2, PC5, PC4, PC4, PC2 vom ATmega16 verwendet. Die Signale OC1B und OC2 stellen gewöhnliche Digitalsignale dar und dienen dem Plotter-Treiber als Enable-Eingänge. Die Ausgangspins PC5, PC4, PC4 und PC2 stellen PWM-Signale dar. Für

die Ansteuerung der Motoren für die X- und Y-Bewegungsrichtung sind PWM-Signale nötig, da durch die glättende Tiefpasswirkung der Induktivitäten (Motoren) die digitalen Impulse wieder in analogen Schwingungen gewandelt werden.

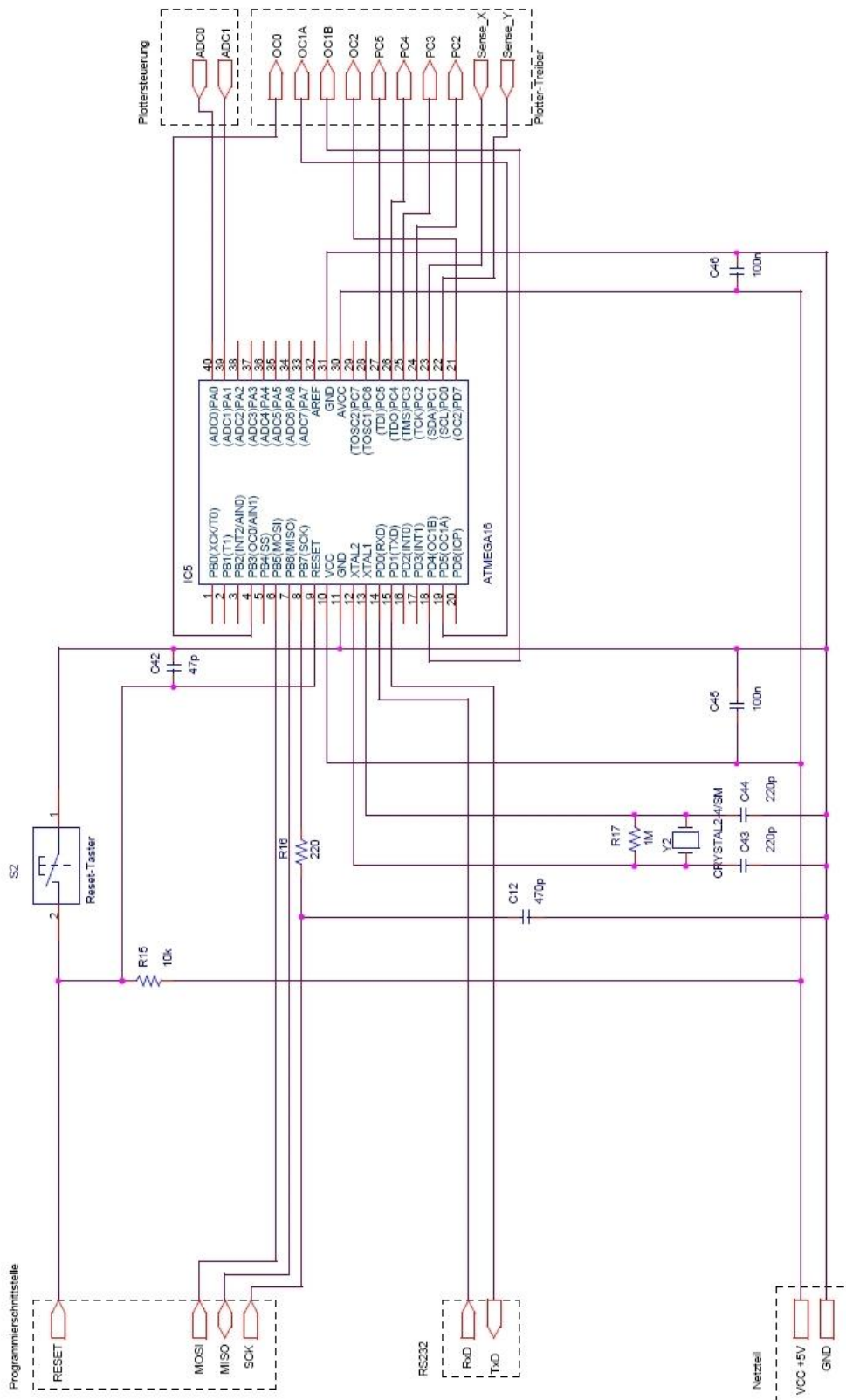


- Abbildung 13: PWM-Signal vor und nach „Glättung“ -

Die Ausgangssignale OC0 und OC1A steuern die für die Kipp- und Drehbewegung zuständigen Servomotoren auf dem Spielbrett. Hierbei handelt es sich wiederum um gewöhnliche Digitalsignale.

Als Eingänge vom Plotter-Treiber stehen dem ATmega16 die Signale Sense_X und Sense_Y zur Verfügung. Diese Signale liefern je nach (Strom-) Belastung des X- und Y-Motors eine bestimmte Spannung, die vom Plotter-Treiber bereitgestellt wird, die vom Mikrocontroller ausgewertet wird und bei zu hoher Belastung den jeweiligen Enable-Ausgang (OC1B, OC2) deaktiviert.

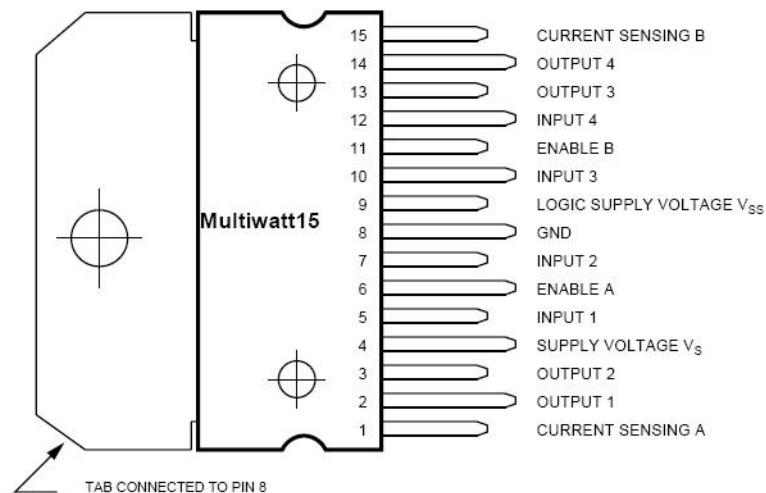
Die Versorgungsspannungen (VCC, AVCC (Versorgungsspannung für den AD-Wandler)) und die Masseverbindung erhält der Mikrocontroller über die intern auf der Leiterplatte bereitgestellten Potentiale (VCC +5V, GND). Zwischen VCC und GND bzw. zwischen AVCC und GND wurden 100nF Koppelkondensatoren (C45, C46) vorgesehen.



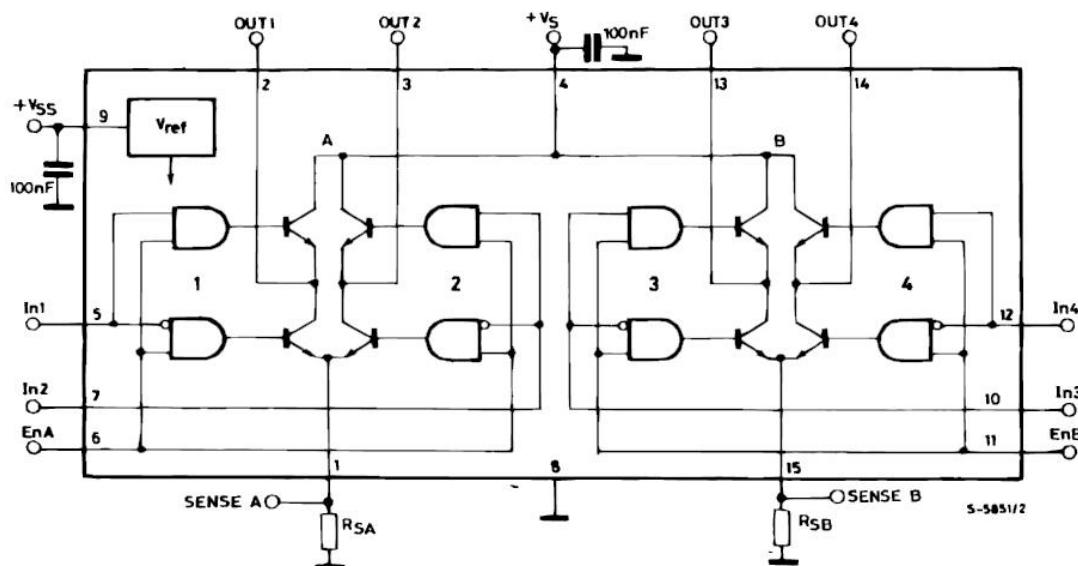
- Abbildung 14: Prozessor -

7 Plotter-Treiber

Der Plotter-Treiber dient zur Bereitstellung der Signale (Spannungen und insbesondere der Ströme) zur Steuerung der Motoren. Herzstück des Motor-Treibers ist das Treiber-IC IC6 (L298)⁸.



- Abbildung 15: Pinbelegung L298 -



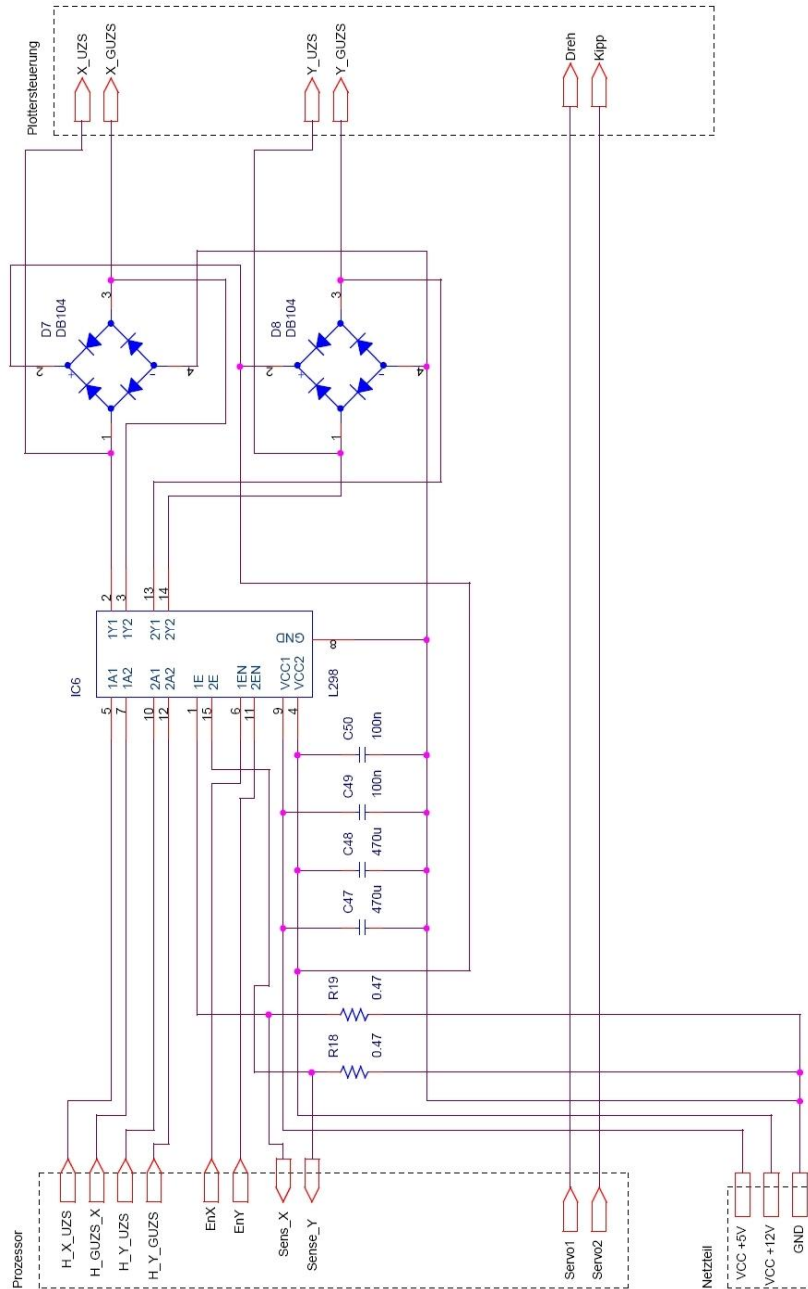
- Abbildung 16: Blockdiagramm L298 -

Anhand der Pinbelegung und des Blockdiagramms wurde das IC6 funktionell verschaltet. Die vom Prozessor erzeugten PWM-Signale H_X_UZS, H_X_GUZS, H_Y_UZS und H_Y_GUZS wurden auf die Eingänge Input 1, 2, 3 und 4 (5, 7, 10, 12) verschaltet. Die Steuersignale EnX und EnY, welche angeben, ob das Signal zum

jeweiligen Motor durchgeschaltet wird, sind mit den Eingängen EnA (6) und EnB (11) verdrahtet. Wie bereits erwähnt, werden die beiden Enable Signale vom Prozessor nur durchgeschaltet, falls die (momentane) Strombelastung für die Motoren nicht zu hoch ist. Damit der Prozessor diese Entscheidung treffen kann, ist es nötig die vom IC6 bereitgestellten Signale SenseA und SenseB abzugreifen, und über die Signalleitungen Sense_X bzw. Sense_Y zum Prozessor zurückzuführen. Hierzu wurden parallel zum Rückführungspfad die beiden Hochlastwiderstände R18 und R19 eingeführt. Diese dienen zur Strombegrenzung.

Als Ausgangssignale für die Plottersteuerung dienen die auf der rechten Seite des Schaltplans dargestellten Signale X_UZS, X_GUZS, Y_UZS, Y_GUZS, Dreh und Kipp. Dabei werden die beiden Eingangssignale Servo1 und Servo2 direkt auf die beiden Ausgangssignale Dreh und Kipp auf die Plottersteuerung durchgeschaltet. Die Signale 1Y1 (X_UZS) (2), 1Y2 (X_GUZS) (3), 2Y2 (Y_UZS) (13) und 2Y2 (Y_GUZS) (14) werden vom Treiber-IC IC6 über die beiden Freilaufdiodeschaltungen D7 und D8 (DB104⁹) an die Plottersteuerung weitergegeben.

Neben der umfangreichen Freilaufdiodeschaltung ist es notwendig das Treiber-IC durch die Kondensatoren C47 bis C50 zu schützen. Direkt an den Pin der jeweiligen Versorgungsspannung (VCC1 (+5V) (9), VCC2 (+12V) (4)) wird parallel zur Masse ein Kondensatortopf (470 μ F, C50, C49 - Polung beachten) platziert und parallel dazu zur Masse ein 100nF Kondensator. Somit ist der jeweilige IC-Eingang von Störungen (vollständig) entkoppelt.



- Abbildung 17: Plotter-Treiber -

8 Plottersteuerung

Die Plottersteuerung dient zum einen der Bereitstellung der Signale für die Motoren (realisiert mit der Sockeleiste CON_PLOTTER) und zum anderen bietet sie mit den Konstantstromquellen LM334¹⁰ (U1, U2) eine Rückführung von den Motoren zum Prozessor für die inkrementelle Wegmessung. Um die Referenzspannung für die Wegmessung zu setzen wurden in Rückkopplungszweigen zwischen V- (3) und R (1) die Widerstände R20 und R21 eingefügt.

CALCULATING R_{SET}

The total current through the LM134 (I_{SET}) is the sum of the current going through the SET resistor (I_R) and the LM134's bias current (I_{BIAS}), as shown in Figure 1.

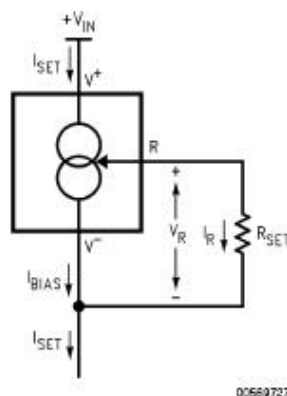


FIGURE 1. Basic Current Source

A graph showing the ratio of these two currents is supplied under **Ratio of I_{SET} to I_{BIAS}** in the Typical Performance Characteristics section. The current flowing through R_{SET} is determined by V_R , which is approximately $214\mu V/K$ ($64\text{ mV}/298\text{ K} - 214\mu V/K$).

$$I_{SET} = I_R + I_{BIAS} = \frac{V_R}{R_{SET}} + I_{BIAS}$$

Since (for a given set current) I_{BIAS} is simply a percentage of I_{SET} , the equation can be rewritten

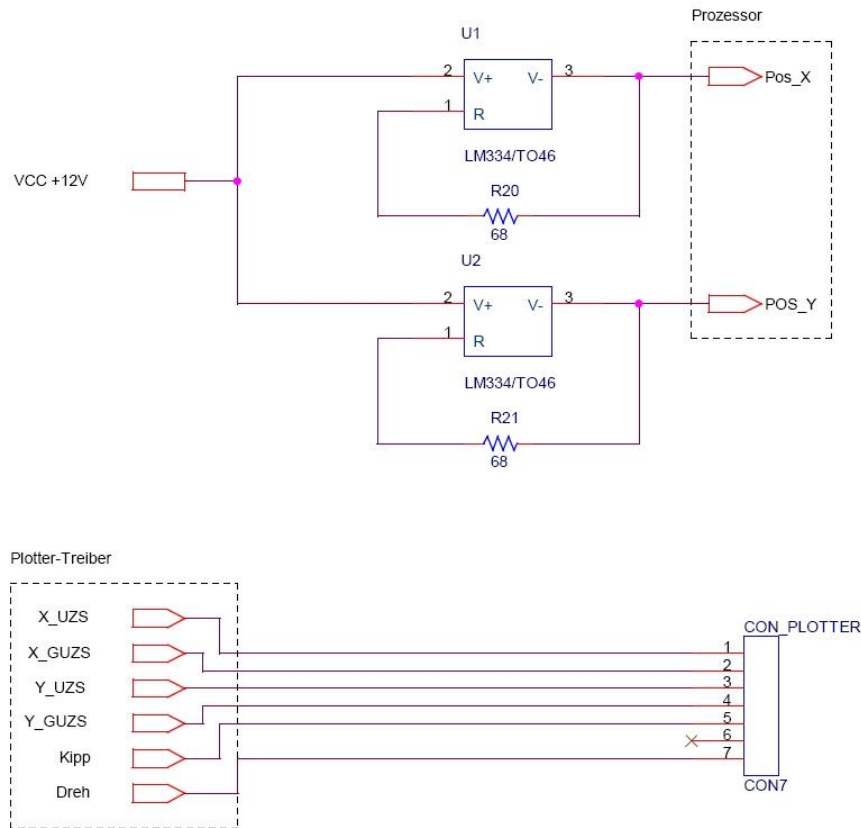
$$I_{SET} = \left(\frac{V_R}{R_{SET}} \right) \left(\frac{n}{n-1} \right)$$

where n is the ratio of I_{SET} to I_{BIAS} as specified in the Electrical Characteristics Section and shown in the graph. Since n is typically 18 for $2\mu A \leq I_{SET} \leq 1\text{ mA}$, the equation can be further simplified to

$$I_{SET} = \left(\frac{V_R}{R_{SET}} \right) (1.059) = \frac{227\ \mu V/^{\circ}K}{R_{SET}}$$

- Abbildung 18: Berechnung der Spannung R_{Set} -

Zur detaillierten Berechnung der Referenzspannung R_{Set} (siehe Auszugsweise Abbildung 18) sei auf das Datenblatt [Quelle 10] verwiesen.



- Abbildung 19: Plottersteuerung-

9 Anhang

9.1 Bauteileliste

Plotter-Treiber Revised: Thursday, December 14, 2006
1 Revision: 1.5

Bill Of Materials December 18, 2006 23:33:59 Page1

Item	Quantity	Reference	Part
1	2	CN2, CN1	CN-USB
2	1	COM	RS232
3	1	CON_NT	V_OUT
4	1	CON_PLOTTER	CON7
5	20	C1, C13, C14, C18, C19, C20, C21, C22, C24, C25, C26, C27, C28, C30, C31, C34, C45, C46, C49, C50	100n
6	2	C2, C3	22p
7	11	C4, C5, C6, C7, C8, C9, C15, C16, C32, C33, C35	
8	1	C10	0.5u
9	1	C11	33n
10	1	C12	470p
11	3	C12, C17, C29	10u
12	1	C42	47p
13	2	C44, C43	220p
14	2	C47, C48	470u
15	1	D-USB-Actice	LED
16	6	D1, D2, D3, D4, D5, D6	d
17	2	D7, D8	DB104
18	1	IC1	74HC125/LCC
19	1	IC2	MAX232
20	1	IC3	FT232BM
21	1	IC4	AD8183
22	1	IC5	ATMEGA16
23	1	IC6	L298
24	2	LPT1, LPT	SUB-D
25	1	RS232	CON9_4
26	3	R1, R6, R15	10k
27	1	R2	470
28	2	R4, R3	27
29	1	R5	1.5k
30	1	R7	15k
31	1	R8	1.2k
32	1	R16	220
33	1	R17	1M
34	2	R19, R18	0.47
35	2	R20, R21	68
36	1	S1	USB-RS232
37	1	S2	Reset-Taster
38	2	U1, U2	LM334/TO46
39	1	Y1	6MHz RESONATOR
40	1	Y2	CRYSTAL2-4/SM

9.2 Netzliste

Netzliste¹¹

10 Quellenverzeichnis

-
- ¹ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/schaltplan.opj>
 - ² <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/schaltplan-rev1.5.pdf>
 - ³ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/datasheets/74hc125.pdf>
 - ⁴ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/datasheets/atmega16.pdf>
 - ⁵ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/datasheets/max232.pdf>
 - ⁶ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/datasheets/ft232bm.pdf>
 - ⁷ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/datasheets/AD8183.pdf>
 - ⁸ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/datasheets/L298.pdf>
 - ⁹ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/datasheets/DB104.pdf>
 - ¹⁰ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/datasheets/L334.pdf>
 - ¹¹ <http://mu21.de/Development/users/.michael/studium/schaltungstechnik/nt8/SCHALTPLAN.EDN>