

1. Modulateblätter

1.1 Touch

Das Touchmodul beinhaltet einen Sensor, welcher auf Kapazitätsänderung, ausgelöst durch eine Fingerberührung, anspricht. Das Modul beinhaltet zudem eine LED, welche frei beschaltet werden kann.

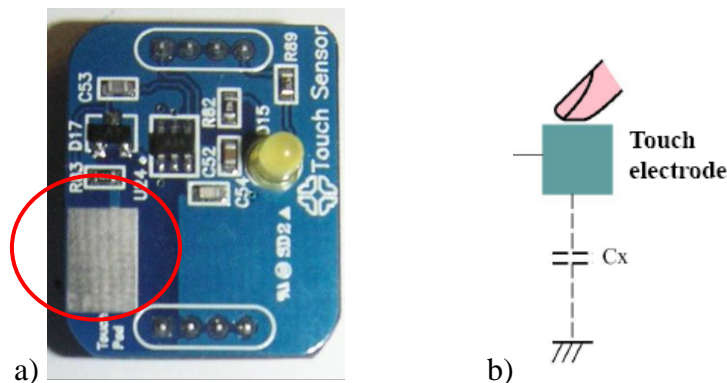


Abbildung 3: Foto (a) des Touchmoduls mit Kennzeichnung der Touch-Elektrode und Auszug (b) aus dem Datenblatt eines Touchsensor-ICs.

Diese Kapazitätsänderung lässt sich mittels des ICs detektieren und in konstante Signalpegel verwandeln. Das TOUCH-Modul hat die in Abbildung 5 dargestellte Pinbelegung, wobei hier die Toucheinheit vereinfacht als Schalter dargestellt wird.

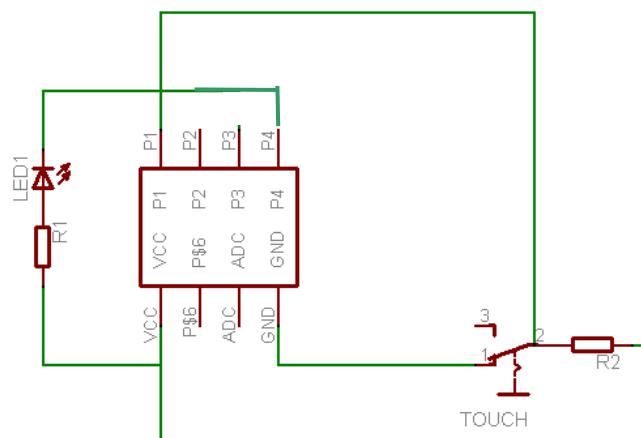


Abbildung 5: Ersatzschaltbild des Touchmoduls

1.2 LED-Modul

Das LED-Modul ist ein optisches Ausgabemodul welches von MA-VIN gesteuert werden kann. Des Weiteren ist für jede LED ein Widerstand vorgesehen, der den Strom für die LED begrenzt.

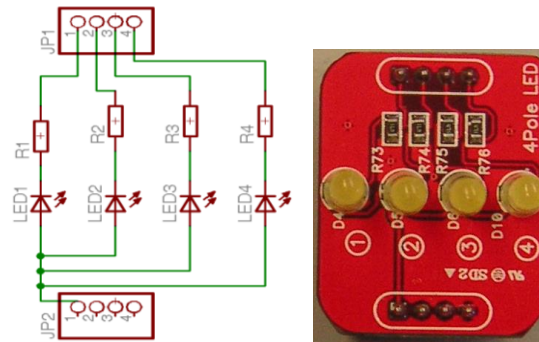


Abbildung 6: Schaltplan und Foto des LED-Moduls

An JP1 sind die I/Os des μ Cs angeschlossen. An JP2 Pin1 ist die Versorgungsspannung angeschlossen. R1-R4 sind die strombegrenzenden Widerstände.

Um die LEDs zu steuern müssen die I/Os an JP1 mit DDR-Register als Ausgang geschaltet werden. Um die LEDs nun einzuschalten, werden die Ausgänge auf Masse gelegt (LED an, da der ganze Strom durch den μ C nach Masse fließen kann).

1.3 Buzzer

Das Buzzer-Modul ist ein akustisches Ausgabemodul. Der piezoelektrische Buzzer wird über 5V Impulse zum Schwingen angeregt, was einen Summertone verursacht. Die Steuerimpulse müssen auf P2 anliegen.

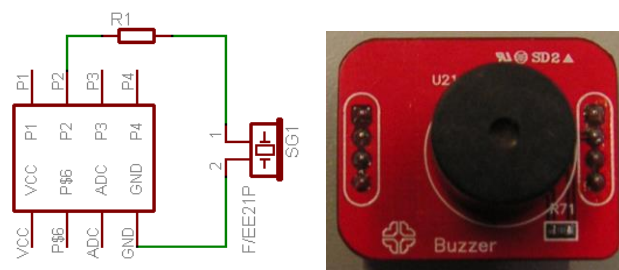


Abbildung 7: Schaltplan und Foto des Buzzers

1.4 Taster-Modul

Auf dem Taster-Modul sind drei Taster aufgebracht. Diese sind als active-low Taster verschaltet. Über einen Pullup-Widerstand (damit bei Tastendruck kein Kurzschluss entsteht) liegt der HIGH-Pegel direkt am μC an. Wird nun ein Taster betätigt wird der Strom über den Taster direkt auf Masse gezogen und ein LOW-Pegel liegt am μC an.

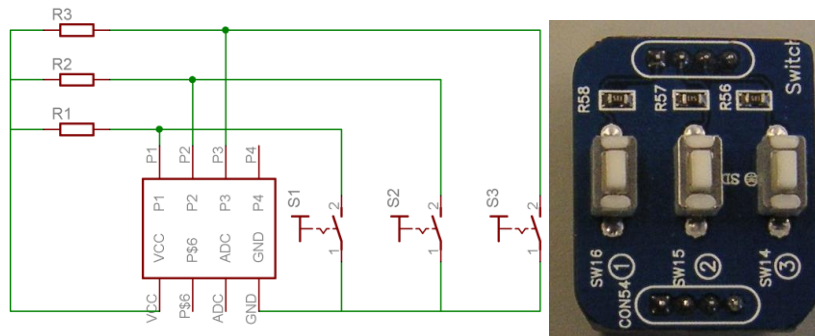


Abb. 8: Schaltplan und Foto des Tastermoduls

1.5 Servomodul

Ein Servo-Antrieb wird über PWM gesteuert. Dazu werden 1-2ms Impulse im Abstand von 20ms gesendet. Die Pulsdauer ist direkt proportional zum Winkel des Motors. Also 1ms-Impuls maximale Linksstellung, 2ms-Impuls maximale Rechtsstellung und 1,5ms Mittelstellung. Der Anschluss des Servos besteht aus den drei Leitungen, Vcc, GND und die Leitung, auf der die Steuerimpulse gesendet werden.

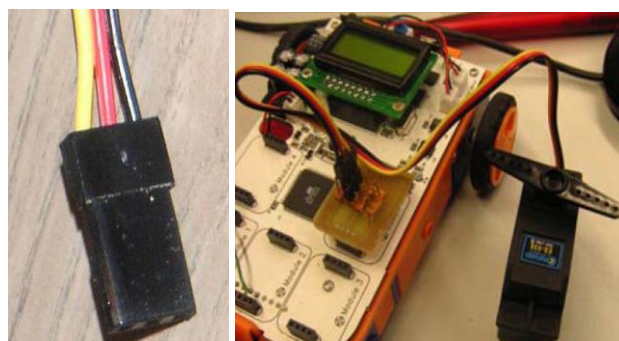


Abbildung 1: Elektrische und logische Anschlüsse des Servomotors

Schwarze Leitung	ist GND
Gelbe Leitung	ist Leitung für Steuerimpulse
Rote Leitung	ist Vcc (ca. 4-5V)

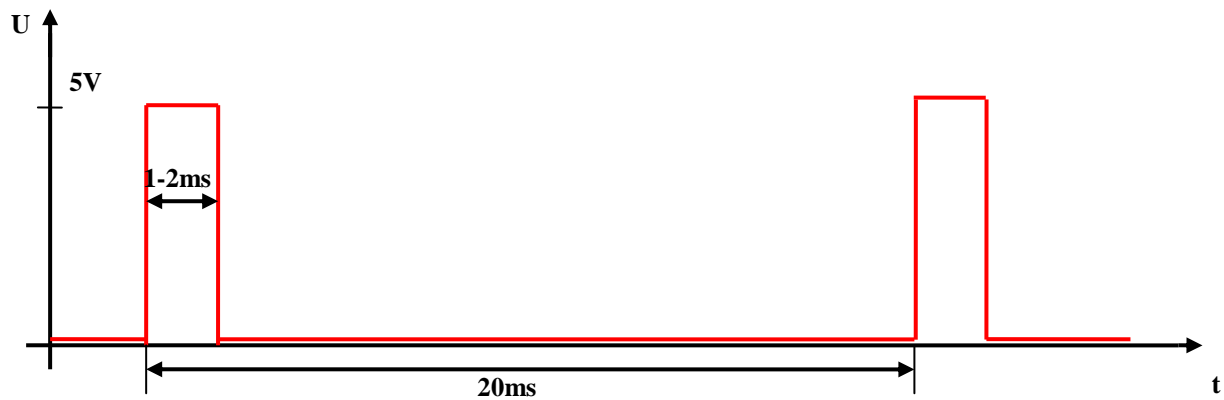


Abbildung 2: Steuerimpulstiming des Servomotors

Die 20ms-Pause hat eine Toleranz von ca. +2ms ohne Beeinträchtigung des Verhaltens des Servos, aber die 1-2ms-Impulse müssen so genau wie möglich sein. D.h. bei einer Steuerung per μC wird hier taktgenau gearbeitet.

1.6 DC-Motoren

Die DC-Motoren des MA-VIN werden über zwei H-Brücken ICs betrieben (BA6289F).

Der μC ist wie folgt mit den Motortreibern verbunden:

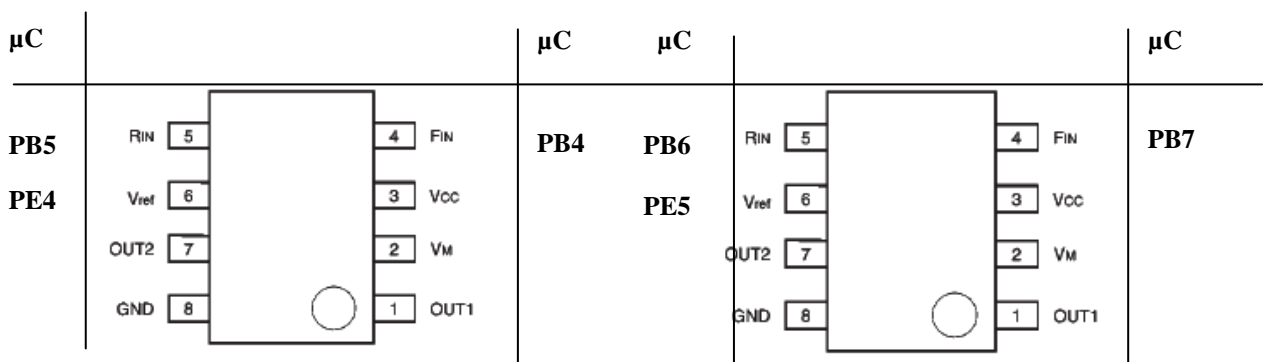


Abbildung 3: Motortreiber mit den referenzierten μC I/Os.

Der Motor ist zwischen OUT1 und OUT2 angeschlossen und wird über VM mit Spannung versorgt.

OUT1 und OUT2 verhalten sich wie folgt, bei $V_{ref} = \text{High}$:

● Input / output truth table

FIN	RIN	OUT2	OUT1	Mode
H	L	L	H	Forward
L	H	H	L	Reverse
H	H	L	L	Brake
L	L	OPEN	OPEN	Standby

Abbildung 4: Wahrheitstabelle, die das Verhalten von OUT1 und OUT2 in Bezug auf die Eingänge wiedergibt.

Die Vorwärts-Rückwärtssteuerung wird durch die Pins PB5/PB4 und beim zweiten IC durch PB6/PB7 wie in der in Abbildung 4 gezeigt übernommen. Die Geschwindigkeitssteuerung erfolgt über Pulsweitenmodulation, wobei die Pulsausgänge PE4/PE5 mit jeweiligem V_{ref} verbunden sind.

Abbildung 5 verdeutlicht den Ablauf einer PWM-Periode. Die Interrupts schalten die Pulse ein und aus. Die Geschwindigkeit ist proportional zum Verhältnis der Länge eines Impulses zur Periodenlänge. Für die Ansteuerung der beiden Motoren werden insgesamt 3 Compare-Matches benötigt. Die x-Achse ist die Zeitachse, die y-Achse ist die Ausgangsspannung (oberes Schaubild) bzw. sind die Zählerstände der Timer-Register (unteres Schaubild).

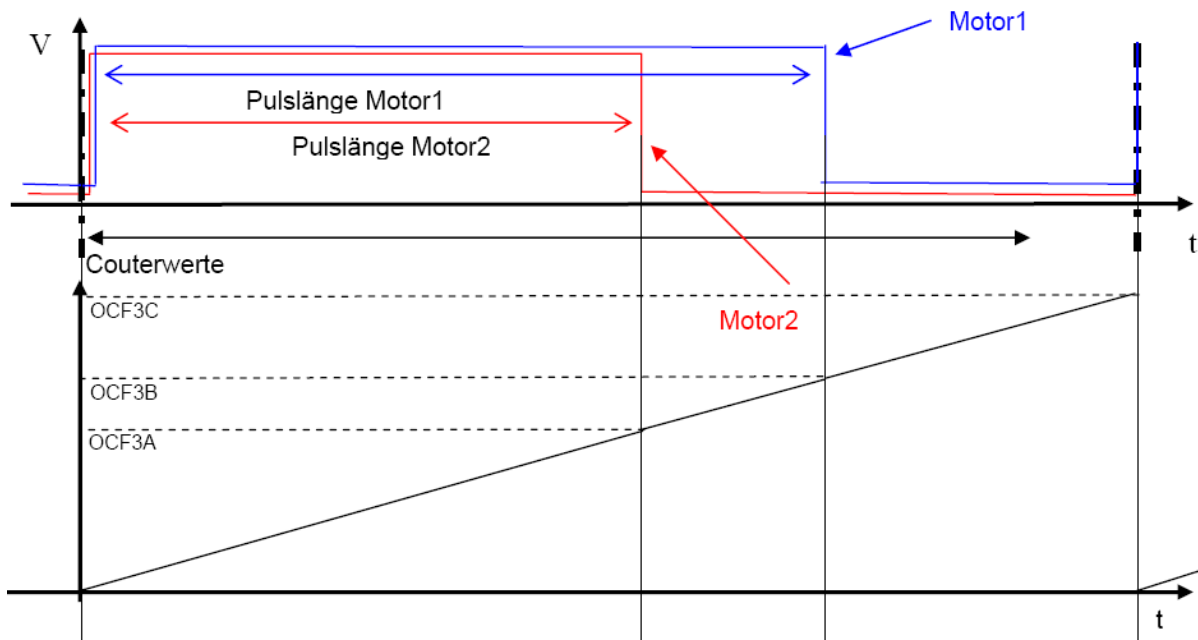


Abbildung 5: Zeit-Spannungsdiagramm der Pulsausgänge:

An OutputCompareMatch OCR3C in Abbildung 5 werden beide Motoren angeschaltet und der Counter auf 0 zurückgesetzt, was an den senkrecht abfallenden Flanke erkannt werden kann. Er legt die Gesamtlänge einer Periode fest. Am OutputCompareMatch OCR3B wird

Motor1 abgeschaltet und an OCR3A Motor 2. Durch das Verhältnis OCR3B/OCR3C kann die Geschwindigkeit des Motors 1 gesteuert werden, durch OCR3A/OCR3C die des Motors 2.

1.7 LDR-Modul

Das LDR-Modul (Abbildung 1) ist ein Sensormodul, welches auf Lichtstärke reagiert. Ein LDR (LightDependentResistor) ist ein lichtabhängiger elektrischer Widerstand. Er wird in einem Spannungsteiler verschaltet. Das Modul liefert eine Spannung in Abhängigkeit von der Lichtstärke zurück.

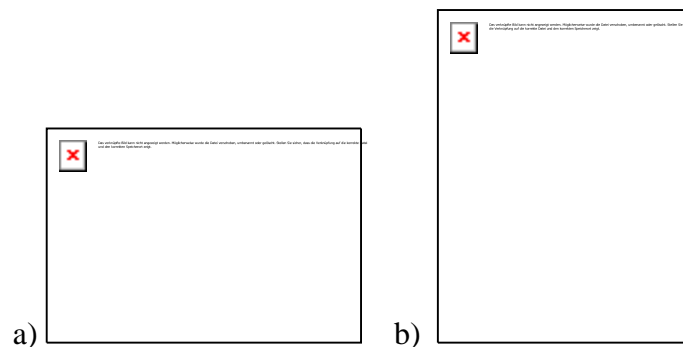
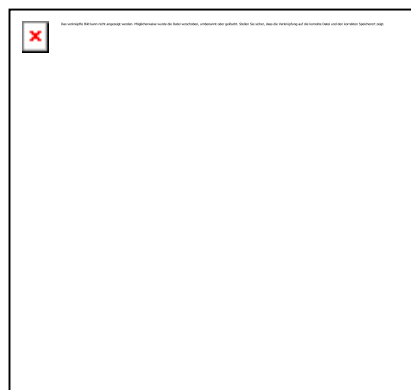


Abbildung 1: Schaltplan (b) und Foto (a) des LDR-Moduls

Die Sensorspannung kann über einen Analog/Digital Converter (ADC) des Prozessors eingelesen und verarbeitet werden.

1.8 Joystick-Modul

Das Joystickmodul besteht aus zwei Teilen: Dem Joystick selbst, welcher über die Potentiometer analoge Spannungen erzeugt, sowie aus der Anschlussplatine für das Beaglebone. Diese beiden Teile werden über ein 6-adriges Kabel verbunden.



1.9 RS-232 USB Modul

Zur Kommunikation zwischen μC und anderen Komponenten kann eine RS-232 Kommunikation verwendet werden. Im Praktikum Echtzeitfähige Systeme und Roboter soll eine solche

Kommunikation mit dem PC implementiert werden. Viele der neuen Rechner besitzen keine serielle COM Schnittstelle mehr, da RS-232 beim Heim-PC durch USB abgelöst wurde. Aber in der Automatisierung und Robotik spielt RS-232 immer noch eine große Rolle. Die Datenübertragung erfolgt durch den UART (Universal Asynchronous Receiver Transmitter). Der UART ist ein Hardware-Baustein, welcher die seriellen RS-232 Daten in ASCII-codierte Zeichen (8-Bit Blöcke) umsetzt. Dieser UART ist auf nahezu allen neueren μ Cs als Hardware integriert und kann zur Kommunikation mit Peripheriegeräten oder auch PCs genutzt werden.

1.9.1 Die RS-232-USB-Verbindung des MA-VIN

Die Praktikumlaptops verfügen über keine RS232 Schnittstelle. Deshalb übernimmt das MA-VIN RS232-Modul die Bereitstellung einer externen RS232-Schnittstelle, die per USB an den Rechner angeschlossen wird. Das RS232-Modul wird automatisch von WindowsXP erkannt und im Geräte Manager als USB Serial PORT angezeigt (Abbildung 1).

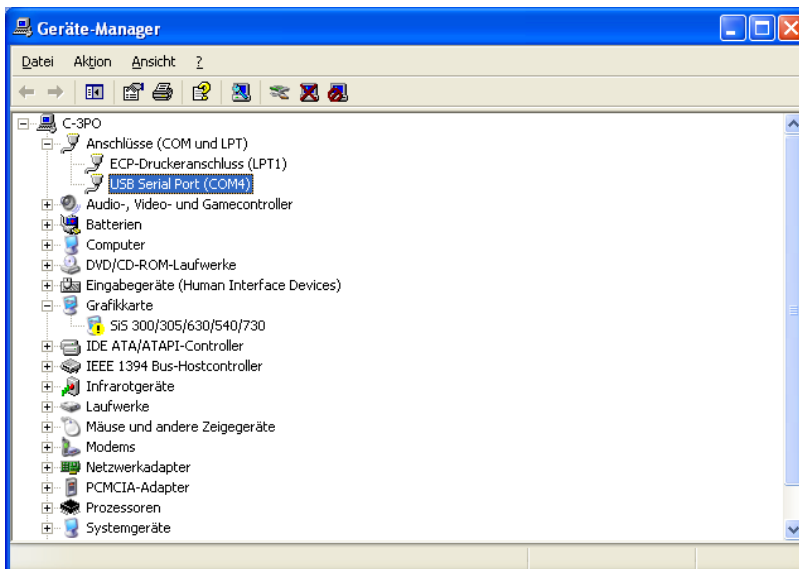


Abbildung 1: Gerätemanager, COM4 wird als serielle Schnittstelle erkannt

Für die Datenübertragung reichen zwei Datenleitungen aus. Über die RXD-Leitung können am Mikrocontroller Daten empfangen, über die TXD-Leitung Daten versendet werden. Das RS232-Modul kann nur am Steckplatz 4 betrieben werden. Die RXD-Leitung liegt dann am Pin **PE0** an, die TXD-Leitung am Pin **PE1**.

1.9.2 Kommunikation über RS-232 mit PC

Um über RS-232 zu kommunizieren beinhaltet jedes Betriebssystem ein Terminalprogramm. Im Falle von WinXP, was auf den Praktikumsrechnern installiert ist, erfolgt ein Verbindungsaufbau über das sogenannte Hyperterminal. Hierzu müssen Verbindungseinstellungen vorgenommen werden:

1. Schnittstelle wählen (Abbildung 2). Wie in Abbildung 1 zu sehen ist die serielle-USB-Schnittstelle automatisch als COM4 deklariert worden. Um diese nutzen zu können muss COM4 auch als Verbindung im Hyperterminal verwendet werden.



Abbildung 2: Hyperterminaleinstellung, Auswahl der genutzten Schnittstelle

2. Nun müssen noch Einstellungen vorgenommen werden, welche das Datenformat, die Übertragungsgeschwindigkeit und Flusssteuerung definieren. Im Praktikum wird eine Datengeschwindigkeit von 9600 Baud genutzt, kein Paritäts-Bit, ein Stopbit, 8 Datenbits und keine Flusssteuerung (Abbildung 3)

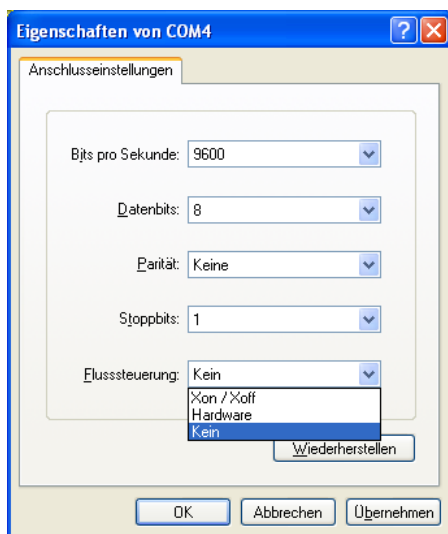


Abbildung 3: Eigenschaften der seriellen Verbindung

Nun werden alle Zeichen, die im Hyperterminal eingegeben werden, über die serielle Schnittstelle COM4 versandt. Das Versenden visualisiert die linke LED auf dem MA-VIN RS-232-Modul.