

Digitaltechnik

Laborübung Schrittmotorsteuerung

Prof. Dr. Helmut Neemann
e-mail: helmut.neemann@mosbach.dhbw.de

16. Juni 2020



- 1 CPLD ATF1504
- 2 Laborplatinen
- 3 Schaltungsentwurf
- 4 Das Flashen auf den Chip
- 5 Schrittmotoren
- 6 Steuerung
- 7 Der verwendete Motor
- 8 Aufgaben



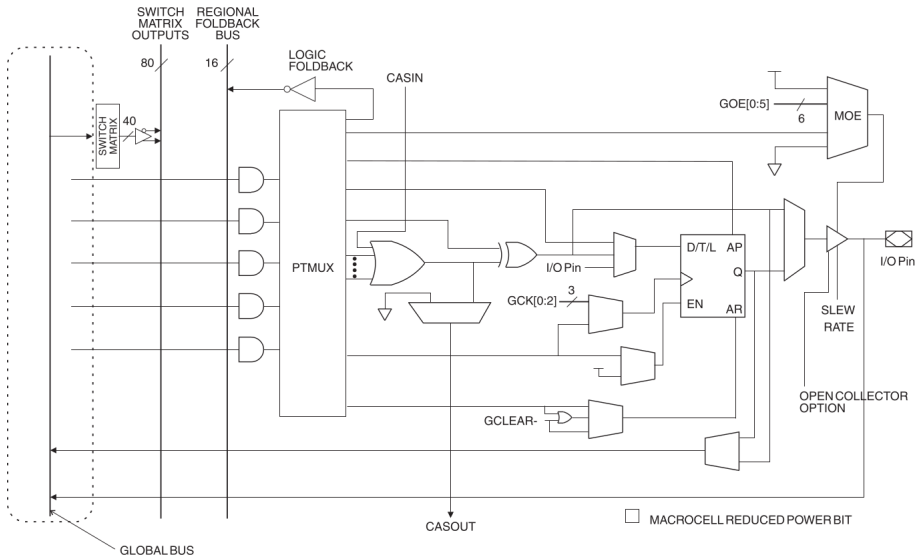
- 1 CPLD ATF1504
- 2 Laborplatinen
- 3 Schaltungsentwurf
- 4 Das Flashen auf den Chip
- 5 Schrittmotoren
- 6 Steuerung
- 7 Der verwendete Motor
- 8 Aufgaben



- Der ATF1504 ist ein CPLD mit 64 Makrozellen:
 - ▶ Jede Makrozelle enthält ein D-Flip-Flop, kann aber auch rein kombinatorisch genutzt werden.
 - ▶ Jede Makrozelle kann maximal 5 Produktterme verknüpfen.
 - ▶ Werden mehr Produktterme benötigt, werden mehrere Makrozellen kaskadiert.
- Es gibt 32 IO Pins:
 - ▶ Ein Teil der Pins wird für die Clock und das JTAG Interface verwendet.
 - ▶ Es stehen damit noch 24 Pins zur freien Verfügung.
 - ▶ Jeder Pin kann als Eingang oder Ausgang konfiguriert werden.
- 5V Versorgungsspannung
- Verfügbar im PLCC-Gehäuse (Sockelbar)
- unterstützt ISP (In System Programming)

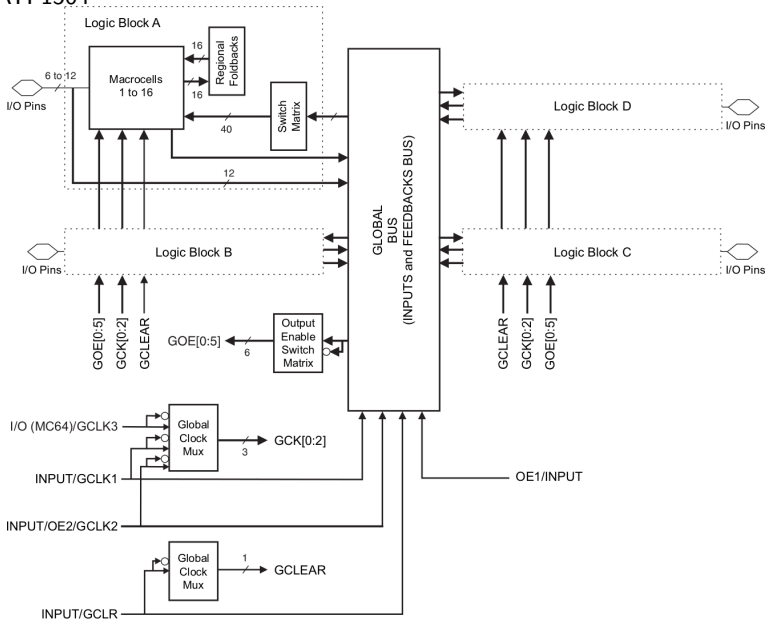
ATF1504 Makrozelle

CPLD ATF1504



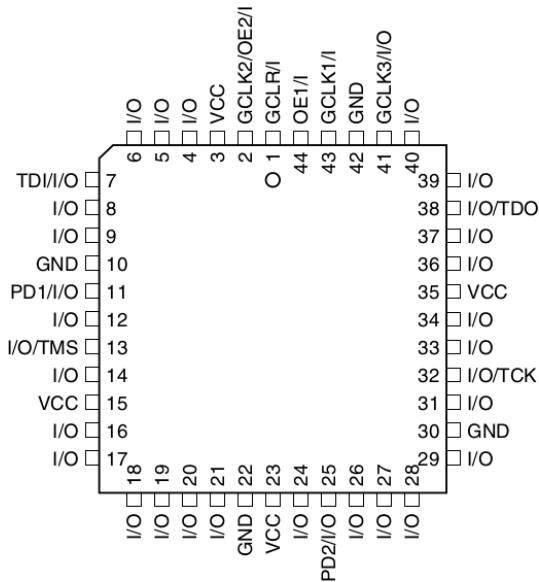
ATF1504 Blockdiagramm

CPLD ATF1504



ATF1504 Pinout

CPLD ATF1504



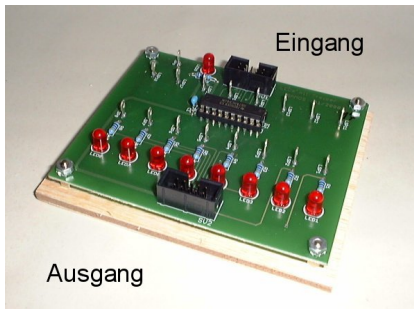
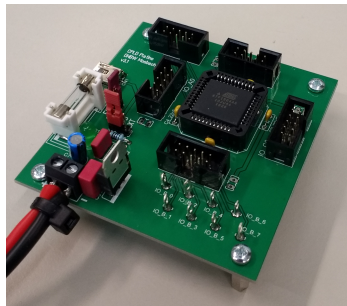
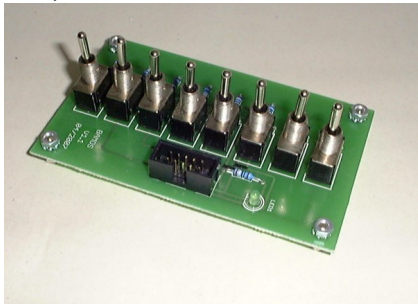


- 1 CPLD ATF1504
- 2 Laborplatinen**
- 3 Schaltungsentwurf
- 4 Das Flashen auf den Chip
- 5 Schrittmotoren
- 6 Steuerung
- 7 Der verwendete Motor
- 8 Aufgaben

Laborplatten



Laborplatten



Laborplatine CPLD

Laborplatinen



JTAG

Takt

USB

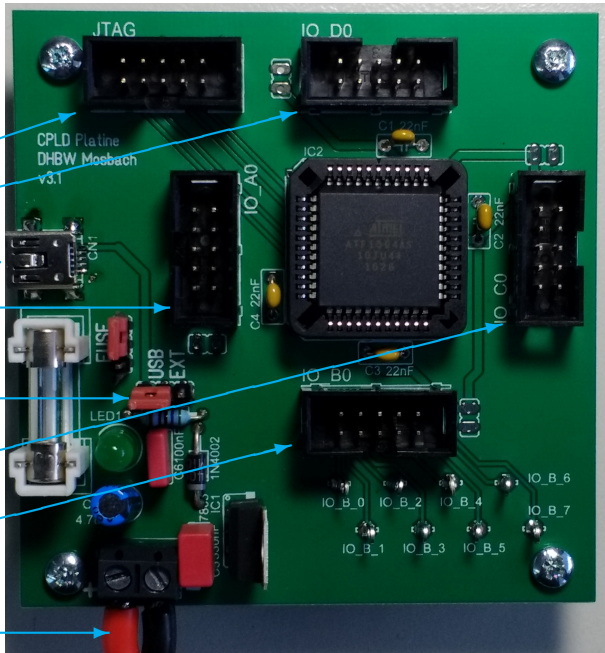
Port A

Power JP

Port C

Port B

Power



ATF1504 Pinzuordnung

Laborplatinen



Port	Pin	Port	Pin	Port	Pin
A0	4	B0	16	C0	26
A1	5	B1	17	C1	27
A2	6	B2	18	C2	28
A3	8	B3	19	C3	29
A4	9	B4	20	C4	31
A5	11	B5	21	C5	33
A6	12	B6	24	C6	34
A7	14	B7	25	C7	36



- Die Platine sollte nicht aus dem USB-Port eines Rechners versorgt werden, um Beschädigungen durch Kurzschlüsse zu vermeiden.
 - ▶ Schon ein Kugelschreiber, der unter die Platine gerät, könnte einen Kurzschluss verursachen.
- Es sollte entweder ein handelsübliches USB-Ladegerät (5 V) verwendet werden:
 - ▶ Den Power Jumper (Power JP) auf „USB“ setzen.
 - ▶ Ladegerät am USB-Port anschließen.
- Oder es wird ein Labornetzteil verwendet:
 - ▶ Den Power Jumper (Power JP) auf „EXT“ setzen.
 - ▶ Auf der Platine ist ein einfacher Linearregler aufgebaut: Die Versorgungsspannung des Netzteils sollte daher bei 6.5 V liegen, um dessen Spannungsabfall auszugleichen.



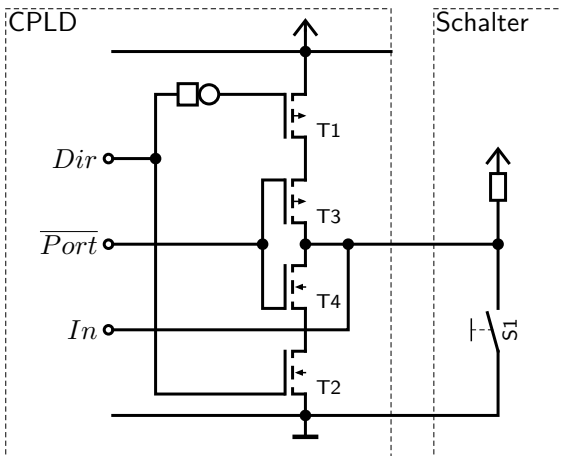
Vorsicht!

- Wenn ein CPLD-Pin als Ausgang konfiguriert wurde, dürfen an diesen Pin nur noch Eingänge angeschlossen werden!
 - ▶ Dies ist z.B. die Eingangsseite eines LED-Boards.
 - ▶ Wird an einen solchen Pin ein zweiter Ausgang angeschlossen, wie z.B. ein Schalter oder die Ausgangsseite eines LED-Boards, könnte das CPLD zerstört werden!
- Niemals dürfen mehrere Ausgänge miteinander verbunden werden!
- Es ist also Vorsicht geboten, wenn die Schalter oder die Ausgangsseite der LED-Boards angeschlossen werden.
 - ▶ Überzeugen Sie sich davon, dass die entsprechenden CPLD-Pins wirklich als Eingang konfiguriert sind!

Tri-State-Ausgang mit angeschlossenem Schalter



Laborplatten



- Wenn der Schalter S1 geschlossen ist, werden die beiden oberen Transistoren (T1, T3) zerstört sobald diese auf „leitend“ geschaltet werden.
- Dies ist der Fall, wenn der Pin als Ausgang konfiguriert ist (T1) und auf High geschaltet wird (T3)!



- 1 CPLD ATF1504
- 2 Laborplatinen
- 3 Schaltungsentwurf**
- 4 Das Flashen auf den Chip
- 5 Schrittmotoren
- 6 Steuerung
- 7 Der verwendete Motor
- 8 Aufgaben



- Um eine Wahrheitstabelle in einem CPLD zu realisieren, muss zunächst eine Schaltung erstellt werden.
- Dies kann jedoch automatisiert geschehen:
 - ▶ Man wählt den Menüpunkt Analyse→Synthese.
 - ▶ Dann legen man fest, wie viele Eingangs- und Ausgangsvariablen benötigt werden.
 - ▶ Jetzt ist die Wahrheitstabelle einzugeben.
 - ▶ Über den Menüpunkt Erzeugen→Schaltung, kann aus der Tabelle eine Schaltung generiert werden.
 - ▶ Die so erzeugte Schaltung sollte man speichern und dann die CPLD-Pinnummern der Ein- und Ausgänge festlegen.
- Auch wenn die Vereinfachung hier automatisch erfolgt: Bitte beachten Sie, dass ein sicherer Umgang mit KV-Tafeln für die Klausur unerlässlich ist.



- Zunächst ist im Simulator die Schaltung zu entwerfen (siehe vorherige Folie).
 - ▶ Es dürfen alle kombinatorischen Gatter verwendet werden.
 - ▶ Es können jedoch **nur** die vorhandenen D- oder JK-Flip-Flops genutzt werden.
 - ★ Ein Flip-Flop aufgebaut z.B. aus zwei NOR-Gattern ist **nicht** verwendbar!
 - ▶ Bei Ein- und Ausgängen ist die gewünschte Pinnummer anzugeben.
- Mit dem Menüpunkt Analyse→Analyse kann die Schaltung analysiert werden.
 - ▶ Es öffnet sich die Tabellenansicht mit der ermittelten Wahrheitstabelle.
 - ▶ Von hier aus lässt sich die erforderliche JEDEC-Datei erzeugen (Siehe Abbildung).
- Es wird der ATMEL-Fitter gestartet und dessen Ausgabe wird in einem eigenen Fenster angezeigt (Siehe Abbildung).
 - ▶ Daher muss →WinCUPL installiert sein.
 - ▶ Wenn WinCUPL nicht im Standard-Ordner installiert wurde, muss der Ordner `Fitters` in den Digital-Einstellungen gesetzt werden.

Erzeugung der JEDEC-Datei

Schaltungsentwurf



Software interface showing a logic circuit design and a table of truth values.

The circuit diagram shows three inputs: B0 (16), B1 (17), and B2 (18). B0 and B1 are connected to an AND gate (&). The output of the AND gate and B2 are connected to an OR gate (≥ 1). The output of the OR gate is C0 (26).

The truth table below shows the relationship between the inputs and the output C0:

File	New	Columns	Settings	Generate	KV-Table
B0	B1			Schaltung	
0	0			Schaltung mit JK Flipflops	
0	0			Schaltung mit Gattern mit zwei Eingängen	
0	1			Schaltung mit NAnd-Gattern	
0	1			Bausteine	
0	1				1
1	0				0
1	0				1
1	1				1
1	1				1

Equation: $C0 = (B0 \wedge B1) \vee B2$

Component selection path: GAL16v8 > GAL22v10 > ATF150x > ATF1502 > ATF1504 > PLCC44 > CUPL > ATF1508 > TQFP44 > TT2 JEDEC

Erzeugt eine dem Berkeley Logic Interchange Format (BLIF) ähnliche Beschreibung der Logik. Danach wird der Atmel-Fitter gestartet, um daraus die JEDEC Datei zu erzeugen.

Ausgabe des Fitters

Schaltungsentwurf



Software interface showing a logic circuit design and a fit message.

The circuit diagram shows three input variables: B_0 (16), B_1 (17), and B_2 (18). B_0 and B_1 are connected to an AND gate (&). The output of the AND gate and B_2 are connected to an OR gate (≥ 1). The output of the OR gate is connected to a circular output symbol labeled C_0 (26).

The fit message window displays the following information:

Meldung vom externen Fitter

Atmel ATF1504AS Fitter Version 1.8.7.8 (02-05-03)
Copyright 1999,2000 Atmel Corporation

Fitter_Pass 1, Preassign = KEEP, LOGIC_DOUBLING : OFF
...
Design fits successfully

Buttons:

Equation: $C_0 = (B_0 \cdot B_1) \vee B_2$

File	New	Columns	Set	Create	KV-Table	
B0		B1		B2		C0
0		0		0		0

ATMISP7

Schaltungsentwurf



The screenshot shows the ATMISP7 software window. The title bar reads "ATMISP". The menu bar includes "File", "Edit", "View", "Options", and "Help". The toolbar contains icons for file operations. The main interface is divided into several sections:

- Device Chain Hierarchy:** A tree view showing the path "C:\daten\daten\Temp\Digital\z.chn" > "Chip1" > "Device type: ATF1504AS". Other details include "Jtag instruction: Program/Verify", "Inst width: 10", "IDCODE: yes", and "Jedec file: C:\daten\daten\Temp\Digital\z.jed".
- Hardware Setting:** A table with three columns: "# of Dev." (value: 1), "Port Setting" (dropdown: USB), and "Cable Type" (dropdown: ATDH1150USB).
- SVF Setting:** Includes an "SVF File Name" field with a "Browse" button, a "Write SVF file" checkbox (unchecked), a "Use state reset" checkbox (checked), an "SVF Version" dropdown (value: Revision D), and a "TCK period (us)" field (value: 1).
- Log Output:** A text area at the bottom left containing the following text:

```
... executing JTAG instructions...please wait...  
log: erasing device #1.  
log: programming device #1.  
log: verifying device #1: ... OK ...  
log: JTAG instructions finished successfully.
```
- Footer:** The Atmel logo is on the left. On the right, there are "Run" and "Exit" buttons. The Windows taskbar at the bottom shows "Ready" and "NUM" keys.

Vereinfachte Erzeugung der JEDEC-Datei

Schaltungsentwurf



Für den jeweils zuletzt ausgewählten Chip kann direkt aus dem „Erzeugen“-Menü erneut eine JEDEC-Datei erzeugt werden.

The screenshot shows a logic design environment. At the top, a logic diagram is displayed on a grid background. It features three inputs: B0 (17), B2 (18), and C0 (26). B0 and B2 are connected to a square logic block. The output of this block is connected to another square logic block, which is then connected to the C0 output. A blue line highlights the connection between the two logic blocks.

Below the diagram, a window titled 'Tabelle' is open. It contains a truth table with columns 'Datei', 'Neu', 'Spalten', 'Setzen', 'Erzeugen', and 'KV-Tafel'. The table has 16 rows, with the first row being the header and the following 15 rows containing binary values (0 and 1). The last row of the table contains the logical expression $C0 = (B0 \wedge B1) \vee B2$.

A context menu is open over the 'Erzeugen' column of the table. The menu items are: 'Schaltung', 'Schaltung mit JK Flipflops', 'Schaltung mit Gattern mit zwei Eingängen', 'Schaltung mit NAnd-Gattern', 'Bausteine', and 'ATF150x → ATF1504 → PLCC44 → TT2, JEDEC'. The 'ATF150x → ATF1504 → PLCC44 → TT2, JEDEC' option is highlighted.

A tooltip is visible next to the highlighted menu item, containing the text: 'Erzeugt eine dem Berkeley Logic Interchange Format (BLIF) ähnliche Beschreibung der Logik. Danach wird der Atmel-Fitter gestartet, um daraus die JEDEC Datei zu erzeugen.'

Pins einfacher zuordnen

Schaltungsentwurf



Diese Schaltung lässt sich in einem zweiten Fenster öffnen, um die Pins daraus zu kopieren:

Pinzuordnung des verwendeten ATF1504, PLCC44
Sie können die Pins in Ihre Schaltung kopieren.

A7 (14)	<input type="checkbox"/>	B7 (25)	<input type="checkbox"/>	C7 (36)	<input type="checkbox"/>
A6 (12)	<input type="checkbox"/>	B6 (24)	<input type="checkbox"/>	C6 (34)	<input type="checkbox"/>
A5 (11)	<input type="checkbox"/>	B5 (21)	<input type="checkbox"/>	C5 (33)	<input type="checkbox"/>
A4 (9)	<input type="checkbox"/>	B4 (20)	<input type="checkbox"/>	C4 (31)	<input type="checkbox"/>
A3 (8)	<input type="checkbox"/>	B3 (19)	<input type="checkbox"/>	C3 (29)	<input type="checkbox"/>
A2 (6)	<input type="checkbox"/>	B2 (18)	<input type="checkbox"/>	C2 (28)	<input type="checkbox"/>
A1 (5)	<input type="checkbox"/>	B1 (17)	<input type="checkbox"/>	C1 (27)	<input type="checkbox"/>
A0 (4)	<input type="checkbox"/>	B0 (16)	<input type="checkbox"/>	C0 (26)	<input type="checkbox"/>

<input type="radio"/>	A7 (14)	<input type="radio"/>	B7 (25)	<input type="radio"/>	C7 (36)
<input type="radio"/>	A6 (12)	<input type="radio"/>	B6 (24)	<input type="radio"/>	C6 (34)
<input type="radio"/>	A5 (11)	<input type="radio"/>	B5 (21)	<input type="radio"/>	C5 (33)
<input type="radio"/>	A4 (9)	<input type="radio"/>	B4 (20)	<input type="radio"/>	C4 (31)
<input type="radio"/>	A3 (8)	<input type="radio"/>	B3 (19)	<input type="radio"/>	C3 (29)
<input type="radio"/>	A2 (6)	<input type="radio"/>	B2 (18)	<input type="radio"/>	C2 (28)
<input type="radio"/>	A1 (5)	<input type="radio"/>	B1 (17)	<input type="radio"/>	C1 (27)
<input type="radio"/>	A0 (4)	<input type="radio"/>	B0 (16)	<input type="radio"/>	C0 (26)



- 1 CPLD ATF1504
- 2 Laborplatinen
- 3 Schaltungsentwurf
- 4 Das Flashen auf den Chip**
- 5 Schrittmotoren
- 6 Steuerung
- 7 Der verwendete Motor
- 8 Aufgaben



- Programmiert wird der Chip mit dem Programmer ATDH1150USB-K
 - ▶ Dieser wird per USB an den PC angeschlossen.
 - ▶ Mit der beiliegenden Flachbandleitung wird der Programmer mit der JTAG-Schnittstelle des CPLD-Boards verbunden.
- Der Simulator erzeugt unter anderem eine Datei mit der Endung „*.chn“.
- Diese kann mit der Software →ATMISP7 geöffnet werden.
 - ▶ Beim Start von ATMISP7 muss der Programmer bereits eingesteckt sein.
- Mit dem Button „Run“ wird der Programmierprozess gestartet.
- Sofort nach dessen Abschluss ist die Schaltung in Betrieb.

ATDH1150USB-K

Das Flashen auf den Chip





- 1 CPLD ATF1504
- 2 Laborplatinen
- 3 Schaltungsentwurf
- 4 Das Flashen auf den Chip
- 5 Schrittmotoren**
- 6 Steuerung
- 7 Der verwendete Motor
- 8 Aufgaben



- Schrittmotoren sind bürstenlose Gleichstrommotoren
- Eine volle Umdrehung in eine Reihe von Schritten aufgeteilt.
- Die Motorposition kann gesteuert werden, indem eine entsprechende Anzahl von Schritten erzeugt wird.
- Es ist kein Positionsregler erforderlich (open-loop controller).
- Es gibt verschiedene Möglichkeiten, wie ein solcher Motor aufgebaut werden kann:
 - ▶ Reluktanz Schrittmotor
 - ▶ Permanent Magnet Schrittmotor
 - ▶ Hybrid Schrittmotor

Reluktanz Schrittmotor

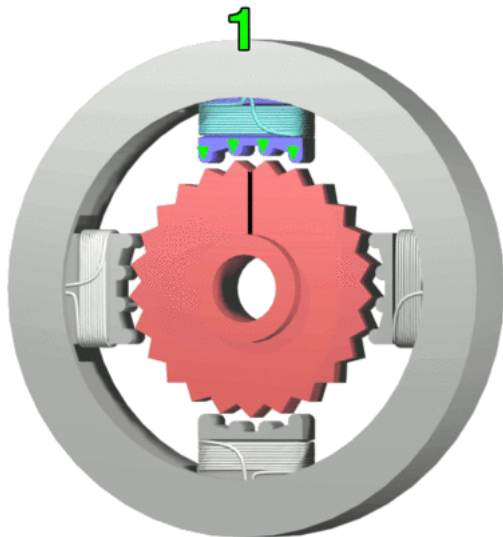
Schrittmotoren



- Es werden keine Permanentmagnete verwendet.
- Statt dessen wird ein weichmagnetischer Rotor eingesetzt.
- Der Motor versucht den magnetischen Widerstand des magnetischen Kreises zu reduzieren
- So bewegt er sich in die Position des geringsten magnetischen Widerstandes.
- Durch Verändern des Spulenstromes kann diese Position verschoben werden.
- Der Motor folgt der Position des geringsten magnetischen Widerstandes.
- Nachteile
 - ▶ Es gibt kein Haltemoment mehr wenn der Strom abgeschaltet wird!

Reluktanz Schrittmotor

Schrittmotoren





- 1 CPLD ATF1504
- 2 Laborplatinen
- 3 Schaltungsentwurf
- 4 Das Flashen auf den Chip
- 5 Schrittmotoren
- 6 Steuerung**
- 7 Der verwendete Motor
- 8 Aufgaben



Es gibt drei Möglichkeiten, wie der Motor angesteuert werden kann:

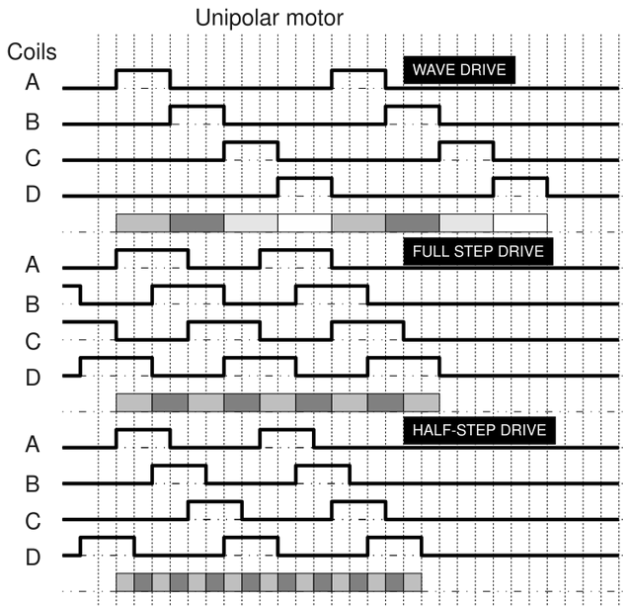
Wave drive Es ist immer nur eine Spule eingeschaltet. Da das Feld dann vergleichsweise klein ist, ist auch das Moment relativ klein.

Vollschrittverf. Dieses Verfahren wird am häufigsten eingesetzt. Es werden immer zwei Spulen eingeschaltet. Das Moment ist daher größer als beim Wave drive. Die Schrittzahl für eine Umdrehung entspricht der des Wave-Dive.

Halbschrittverf. Es wird immer abwechselnd eine und zwei Spulen eingeschaltet. Daher hat der Motor etwas geringeres Drehmoment als im Vollschrittverfahren. Jedoch wird die Anzahl der Schritte, welche für eine Umdrehung erforderlich sind, verdoppelt.

Schritt-Sequenz

Steuerung

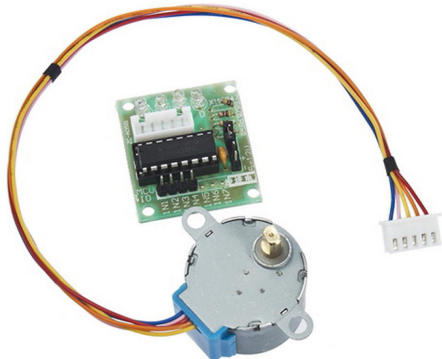




- 1 CPLD ATF1504
- 2 Laborplatinen
- 3 Schaltungsentwurf
- 4 Das Flashen auf den Chip
- 5 Schrittmotoren
- 6 Steuerung
- 7 Der verwendete Motor**
- 8 Aufgaben

28BYJ-48 – 5V Stepper

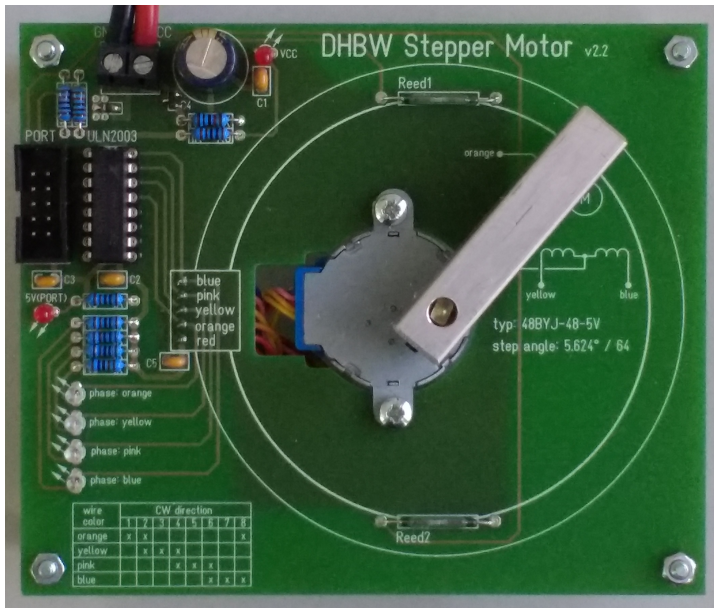
Der verwendete Motor



- Unipolare Ansteuerung
- 32 Schritte im Vollschrittverfahren
- 64 Schritte im Halbschrittverfahren
- 1/63.68 Getriebe um die Geschwindigkeit zu reduzieren
- So sind $63.68 \cdot 64 = 4075.8$ Halbschritte erforderlich für eine Umdrehung.
- Kommt mit Treiberplatine mit einem Darlington-Array und vier LEDs, je eine pro Phase.

Stepper-Platine

Der verwendete Motor

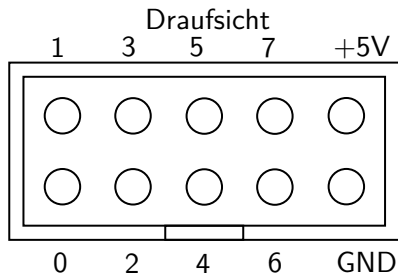


Pinout Laborstecker

Der verwendete Motor



Pinbelegung des verwendeten 10-poligen Steckers:



1: Phase 0

3: Phase 1

5: Phase 2

7: Phase 3

0: Schalter A

2: Schalter B



- 1 CPLD ATF1504
- 2 Laborplatinen
- 3 Schaltungsentwurf
- 4 Das Flashen auf den Chip
- 5 Schrittmotoren
- 6 Steuerung
- 7 Der verwendete Motor
- 8 Aufgaben**



Aufgabe 1

- Implementieren Sie eine Steuerung für einen Schrittmotor.
- Der Motor soll im Halbschrittverfahren angesteuert werden.
- Über einen Eingang D soll die Drehrichtung angegeben werden können.
- Mit jedem Taktimpuls soll der Motor einen Halbschritt ausführen.
- Definieren Sie die erforderliche Logik.
 - ▶ Zeichnen Sie dazu zunächst ein Zustandsdiagramm!
- Weisen Sie die korrekte Funktion in der Digital-Simulation mit entsprechenden Testfällen nach.
 - ▶ Am einfachsten ist es, zuerst die Testfälle anzulegen!
- Steuern Sie einen 28BYJ-48 Schrittmotor an.



Aufgabe 2

- Die Schaltung soll mit einem zusätzlichem Zustandsautomaten erweitert werden.
- Gegeben seien zwei Endlagenschalter S_1 und S_2
- Wird der Schalter S_1 betätigt, soll in Richtung S_2 gefahren werden.
 - ▶ Auch wenn der Schalter die Endlage S_1 nicht mehr erkennt, soll der Motor weiterfahren, bis er die andere Endlage S_2 erreicht.
- Wird die Endlage S_2 erreicht soll der Motor wieder zurückfahren.
 - ▶ Auch hier soll der Motor weiterfahren, auch wenn die Endlage S_2 nicht mehr erkannt wird.
 - ▶ Erst bei der Endlage S_1 soll wiederum ein Richtungswechsel erfolgen.
- Achtung
 - ▶ Die Schalter sind Low-Aktiv
 - ▶ Sie geben eine Null aus, wenn der Motorarm erkannt wird, ansonsten eine Eins.