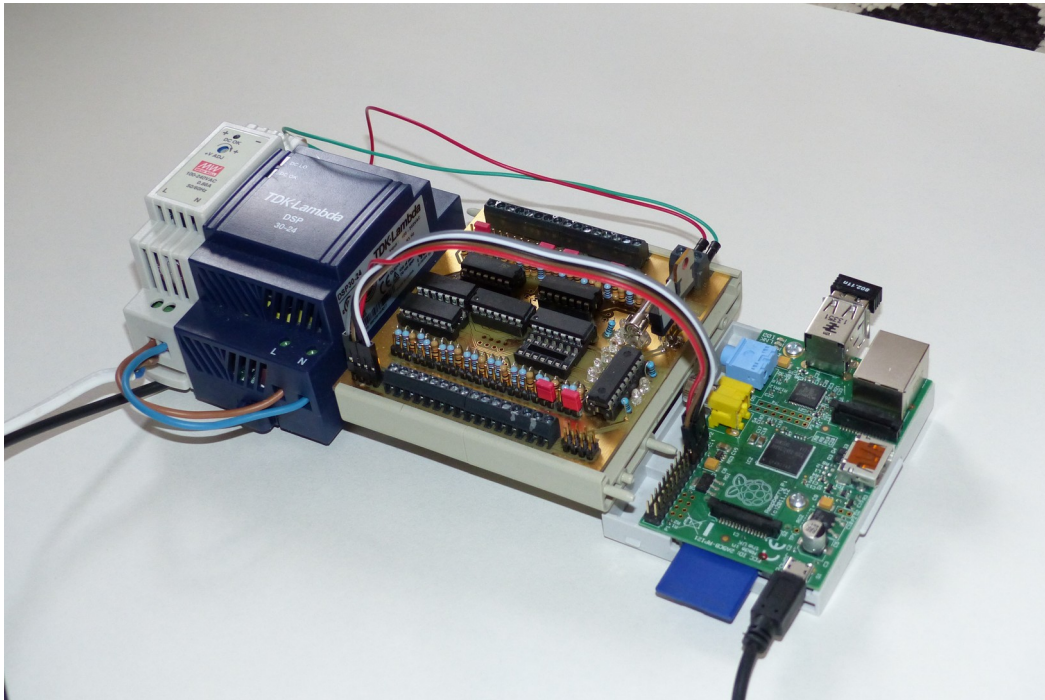


Open Source Expansion Board For Raspberry Pi (will probably work for any device with SPI bus as well)



Features:

Open Source Layout

8 Analog Input 0-10V or 0-24V

8 Analog Output 0-0V or 0-24 V

Connection via SPI Bus to the Raspberry Pi

Fits into a Rail Cap Housing „Wieland Bamberg WEB1001“ 93 mm

Inhaltsverzeichnis:

1. Allgemein / Haftungsausschluss / Lizenz
2. Bestimmungsgemäße Verwendung
3. Technische Daten
4. Zusammenbau
5. Konfiguration
6. Stückliste
7. Schaltplan
8. Anschlußbeispiel Seriell
9. Anschlußbeispiel Parallel
10. Codebeispiel
11. Datenblätter

1. Allgemein / Haftungsausschluss / Lizenz

Betriebsbedingungen:

- Der Betrieb darf nur mit der dafür vorgesehenen Anschlussspannung erfolgen
- Die zulässige Umgebungstemperatur beträgt zwischen 0 und 40°C
- Die Umgebung muss staubfrei und trocken sein.
- Die Bildung von Kondenswasser ist zu vermeiden, bzw. eine Inbetriebnahme ist nur nach vollständiger Trocknung möglich.
- Baugruppen und Bauteile gehören nicht in Kinderhände
- Die Baugruppen dürfen nur unter Aufsicht einer fachkundigen erwachsenen Person in Betrieb genommen werden.
- In gewerblichen Einrichtungen sind die Unfallverhütungsvorschriften des Verbandes der gewerblichen Berufsgenossenschaften für elektrische Anlagen und Betriebsmittel zu beachten
- In Schulen und Ausbildungseinrichtungen ist das Experimentieren und Betreiben der Baugruppe von geeignetem, fachkundigem Personal zu überwachen
- Die Baugruppe darf nicht zur Steuerung von sicherheitsrelevanten Einrichtungen genutzt werden.
- Die Baugruppe darf nicht in der Nähe von brennbaren Stäuben oder Gasen betrieben werden.

2. Bestimmungsgemäße Verwendung:

Die Baugruppe dient zum Anlegen von Spannungen, bzw. zum Messen von analogen Spannungen in Hobby, Modellbau, Prototypenbau. Ein anderer Einsatz als vorgegeben ist nicht zulässig.

Sicherheitshinweis:

Beim Umgang mit elektronischen Bauteilen sind die gültigen VDE – Vorschriften zu beachten. Dies sind unter anderem VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860.

Lizenz:

Diese Schaltung und das Platinenlayout unterliegen der General Public Licence (GPL).

Die Schaltung und das Platinenlayout darf beliebig vervielfältigt, kopiert, nachgebaut und optimiert werden.

Die Schaltung, das Platinenlayout oder fertige Platinen dürfen nicht weiterverkauft werden. Ein Weiterverkauf ist lediglich als Einbau in ein Modell / Prototyp / Maschine gestattet.

3. Technische Daten

Versorgungsspannung	24 - 30 V
Stromaufnahme	Max 1 A
Analoge Eingänge	0-10V / 0-24V Auflösung 10 bit 5 V möglich mit geändertem Spannungsteiler
Analoge Ausgänge	0-10V / 0-24V Auflösung 10 bit (8 oder 12 möglich bei Verwendung des entsprechenden DA – Wandlers) 5 V möglich mit geändertem Spannungsteiler Max Strom: 70mA

4. Zusammenbau:

Tip:

Je nach Einsatzzweck können Komponenten weggelassen werden.

- Wird nur ein Board benutzt, kann AND1 weggelassen werden und in der SPI Konfiguration die Brücke für direkte Verbindung CS0 gesetzt werden.
- Soll nur ein Signallevel 0-10V oder 0-24 V benutzt werden, muß auch nur der entsprechende Widerstandsarray eingesetzt werden (10k für 10V, 2,7k für 24).
- Soll der Analog Ausgang für 0-5V benutzt werden, können die Widerstände der Operationsverstärker entsprechend geändert werden.
- Für den Analogeingang muß der Spannungsteiler nicht bestückt werden, um 0-5V Eingangsspannung zu erreichen.
- Über die 5V Out klemme kann der RPI versorgt werden. Dies ist leider aber nicht bei allen Modellen möglich, da der RPI von unterschiedlichen Herstellern gebaut wird, und diese Varianten unterschiedlich viel Strom verbrauchen. Auch scheint es einen Unterschied zwischen Versorgung über die GPIO – Stecker und den USB – Anschluß zu geben, sowie zwischen Modell 1 und Modell 2.

Vorgehensweise Montage:

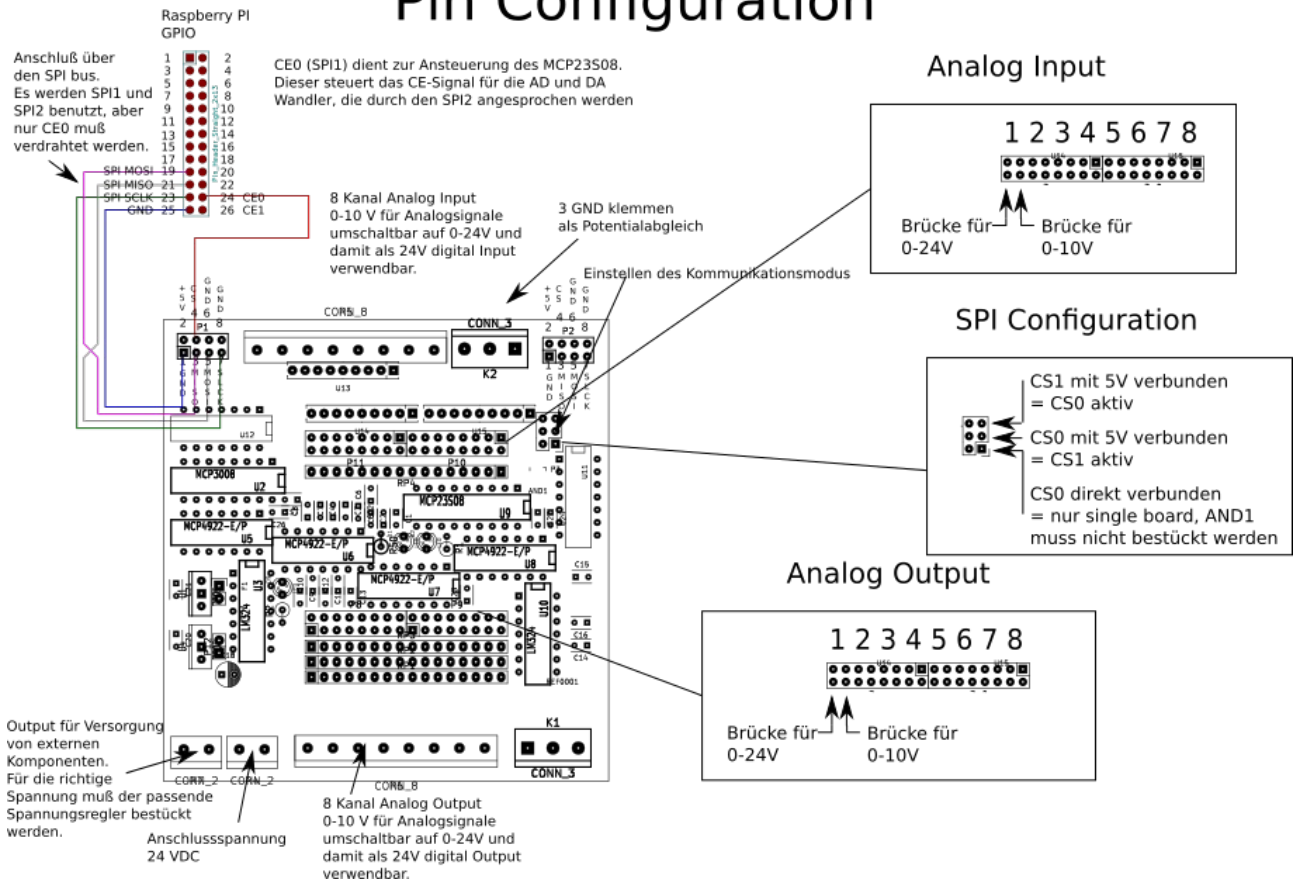
- Zuerst die flachen Bauteile bestücken, und dann die höheren. Es empfiehlt sich, mit dem AND1 – Baustein anzufangen (falls dieser benötigt und bestückt werden soll).
- Es muß auf eine gute Leitfähigkeit der gelöteten Pins zur Platine geachtet werden.

Inbetriebnahme:

- Platine mit 24 V verbinden. Rote Power LED muß leuchten
- Platine mit Raspberry Pi SPI Bus verbinden
- Brücken einsetzen, um die Eingänge und Ausgänge für 0-10V oder 0-24 zu konfigurieren
- Beispielprogramme ausführen
- Ausgangsspannung messen und durch Anlegen einer Spannung die Eingangsspannung anzeigen lassen.
- Falls die Platine nicht auf Anhieb funktioniert, sollte zunächst die Funktion des 23S08 überprüft werden. Hierzu den Chip -Select für einen Baustein ansteuern und mit Voltmeter die Veränderung des Pegels messen. Funktioniert dies nicht, mit einem Ohmmeter die Verbindung von Raspi zu den entsprechenden Pins an dem Baustein messen.

4. Konfiguration:

Pin Configuration



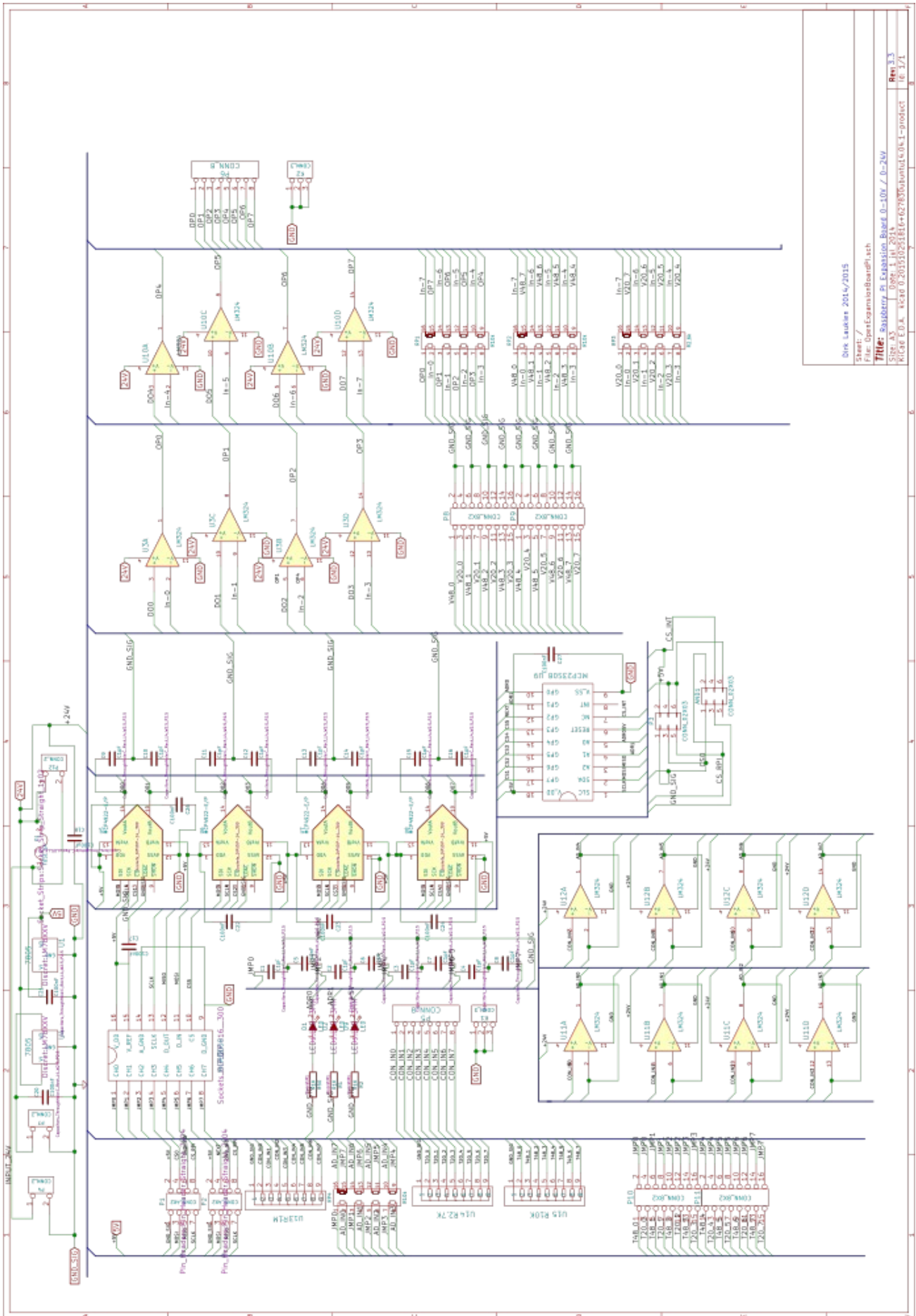
Anmerkung SPI Konfiguration:

- Wenn keine Brücke gesetzt ist, kann CS0 mit CS0 vom Raspberry Pi Verbunden werden, und CS1 mit dem GPIO, um mehrere Boards parallel zu betreiben. Der CS1 dient dann als „Board Select“.
- Wird nur ein einziges Board benutzt, muß der AND – Baustein nicht bestückt werden. Die Brücke für „CS0 direkt verbinden“ muß dann gesetzt werden.
- Ist der AND -Baustein bestückt und es soll nur ein Board benutzt werden, kann der CS1 mit 5 V durch die entsprechende Brücke gesetzt werden. Bei mehreren Boards wird beim Ersten Board auch diese Brücke gesetzt.

5. Stückliste

Id	Bezeichner	Gehäuse	Stückzahl	Bezeichnung	Bemerkung
1	R50,R1,R2	R1	3	R1K	Widerstand 1kOhm
2	U1,U4	LM78XXV	2		7805 Festspannungsregler 5V
3	U2	DIP-16_300	1	MCP3008	AD Wandler 8 Kanal
4	U5,U6,U7,U8	DIP-14_300	4	MCP4922-E/P	DA Wandler 2 Kanal
5	U9	DIP-18_300	1	MCP23S08	Dig Output
6	P1,P2	Pin_Header_Straight_2x04	2	CONN_4X2	Pin Connector
7	K1,K2	bornier3	2	CONN_3	Klemme 3 Kontakte
8	P4,P7	Reihenklemme_2_508	2	CONN_2	Klemme 2 Kontakte
9	P5,P6	Reihenklemme_8_508	2	CONN_8	Klemme 8 Kontakte
10	U3,U10,U11,U12	DIP-14_300	4	LM324	Operationsverstärker
11	P8,P9,P10,P11	Pin_Header_Straight_2x08	4	CONN_8X2	Pin Connector
12	RP1,RP2,RP4	Pin_Header_Straight_1x16	3	R10k	Widerstandsarray 10kOhm
13	RP3	Pin_Header_Straight_1x16	1	R2,6k	Widerstandsarray 2,7kOhm
14	D1,D2,D9	LED-3MM	3	LED	LED
15	P12	Pin_Header_Straight_1x02	1	CONN_2	Pin Connector
16	F1	Socket_Strip_Straight_1x02	1	FUSE_1A	Sicherung 1A
17	P3	Pin_Header_Straight_2x03	1	CONN_02X03	Pin Connector
18	U13	SIP9_Housing	1	R1M	Widerstandsarray 1M
19	U14	SIP9_Housing	1	R2,7K	Widerstandsarray 2,7kOhm
20	U15	SIP9_Housing	1	R10K	Widerstandsarray 10kOhm
21	AND1	TSOT-6-MK06A_Handsoldering	1	CONN_02X03	AND – Baustein
22	C1,C2,C3,C4,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16	C_Rect_L4_W2.5_P2.5	14	C1pF	Kondensator 1pF
23	C5	C_Rect_L4_W2.5_P2.5	1	C10nF	Kondensator 10nF
24	C6	C_Rect_L4_W2.5_P2.5	1	C1F	Kondensator 1F
25	C17,C22,C23,C24,C26,C27,C20,C21	C_Rect_L4_W2.5_P2.5	8	C100nF	Kondensator 100nF
26	C18	C_Radial_D5_L6_P2.5	1	C100uF	Kondensator 100uF

6. Schaltplan

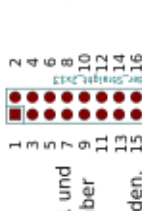


7. Anschlußbeispiel SPI Seriell

Serial Connection

Raspberry Pi
GPIO

Anschluß über den SPI bus. Es werden SPI1 und SPI2 benutzt, aber nur CE0 muß verdrahtet werden.



Über die Pin-Brücken wird konfiguriert, ob die Kanäle als 0-10V oder 0-24V input betrieben werden.

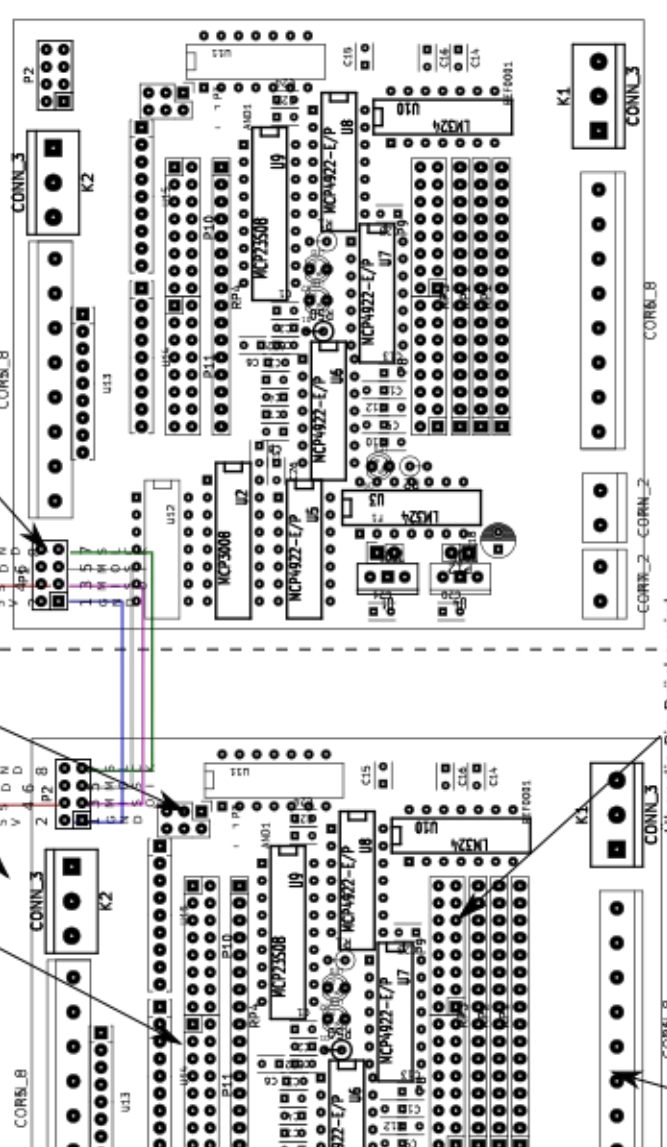
8 Kanal Analog Input 0-10 V für Analogsignale umschaltbar auf 0-24V und damit als 24V digital input verwendbar.

3 GND klemmen als Potentialabgleich

Einstellen des Kommunikationsmodus

Option: mehrere In/Out - Karten lassen sich hintereinanderschalten

Beliebig viele Platinen können über den SPI-Bus dazugeschaltet werden. Über die Adressierung des MCP23S08 können die Platinen angesprochen werden.



Die Sicherung muss nicht bestückt werden. Sie kann mit P12 gebrückt werden.

Output für Versorgung von externen Komponenten. Für die richtige Spannung muß der passende Spannungsregler bestückt werden.

Anschlussspannung 24 VDC

Über die Pin-Brücke wird konfiguriert, ob 0-10V oder 0-24V Ausgangssignal benutzt wird. Jeder Kanal ist individuell konfigurierbar.

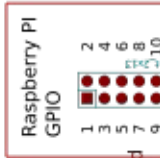
8 Kanal Analog Output 0-10 V für Analogsignale umschaltbar auf 0-24V und damit als 24V digital Output verwendbar.

Sheet: **Model A "SPIDER"**
 File: OpenExpansionBoardPi.kicad_pcb
 Title: Open Source Expansion Board **2014 / 2015 Dirk Laukien**
 Size: A4 Date: **01.05.2015**
 Kicad E.O.A. kicad 0.201502061101+54062Tubuntu14.04.1-product
 Rev: **1**
 Id: 1/1

8. Anschlußbeispiel SPI Parallel:

Prallel Connection

Anschluß über den SPI bus.
 Es werden SPI1 und SPI2 benutzt, aber nur CEO muß verdrahtet werden.



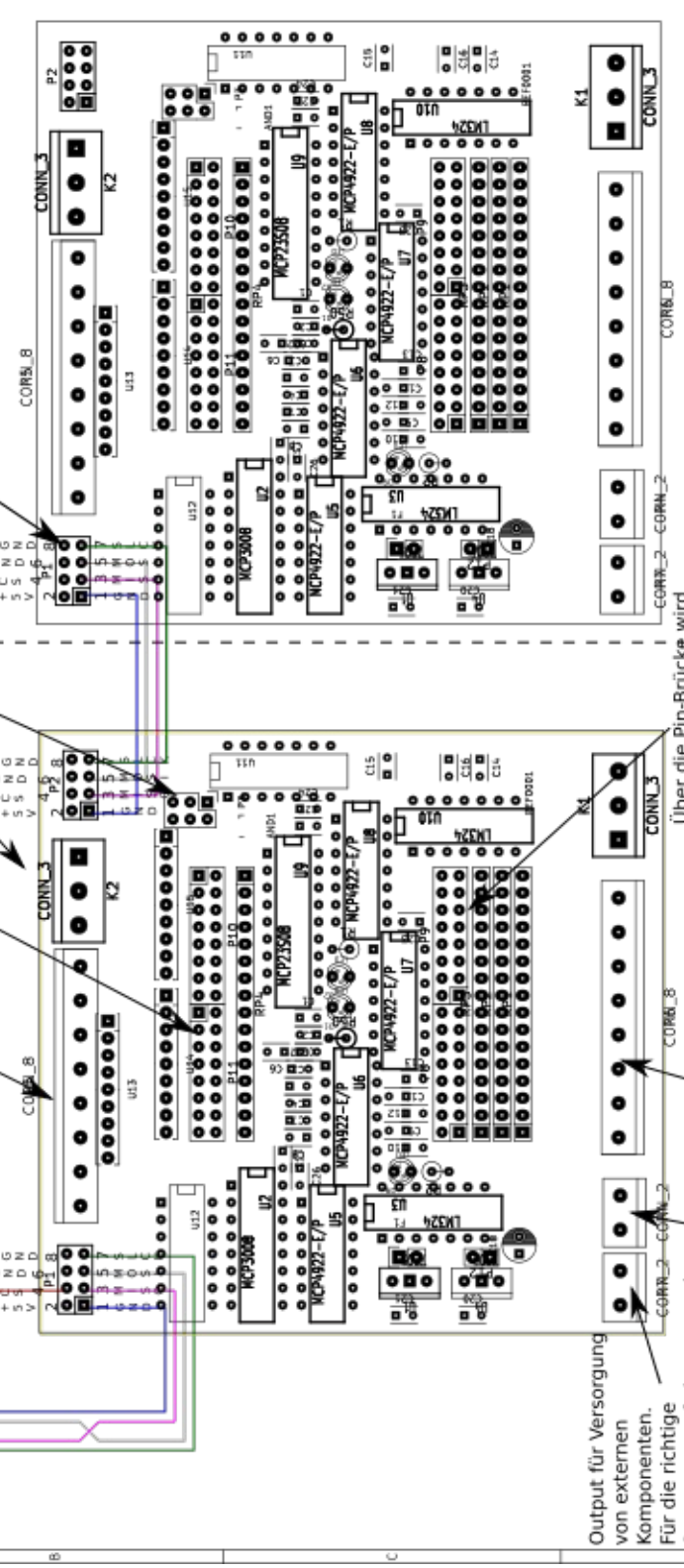
CEO (SPI1) dient zur Ansteuerung des MCP23S08. Dieser steuert das CE-Signal für die AD und DA Wandler, die durch den SPI2 angesprochen werden

Über die Pin-Brücken wird konfiguriert, ob die Kanäle als 0-10V oder 0-24V input betrieben werden.

Option: mehrere In/Out - Karten lassen sich hintereinanderschalten

8 Kanal Analog Input 0-10 V für Analogsignale umschaltbar auf 0-24V und damit als 24V digital input verwendbar.
 Einstellen des Kommunikationsmodus
 3 GND klemmen als Potentialabgleich

Es können so viele Expansion Boards angeschlossen wie GPIO vorhanden sind.



Output für Versorgung von externen Komponenten. Für die richtige Spannung muß der passende Spannungsregler bestückt werden.
 Anschlußspannung 24 VDC

Über die Pin-Brücke wird konfiguriert, ob 0-10V oder 0-24V Ausgangssignal benutzt wird. Jeder Kanal ist individuell konfigurierbar.

8 Kanal Analog Output 0-10 V für Analogsignale umschaltbar auf 0-24V und damit als 24V digital Output verwendbar.

9. Codebeispiel

Unten ein Codebeispiel in C. Auf der Website www.Openexpansionboard.de sind Codebeispiele in C, Python und Gambas enthalten. Ausserdem befindet sich dort ein Codebeispiel für den Betrieb von mehreren in Reihe geschalteten Boards.

Codebeispiel C

```
#include <stdint.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <getopt.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/types.h>
#include <linux/spi/spidev.h>

#define ARRAY_SIZE(array) sizeof(array)/sizeof(array[0])

static const char *device = "/dev/spidev0.0";
static const char *device2 = "/dev/spidev0.1";
static uint8_t mode = SPI_MODE_0;
static uint8_t bits = 8;
static uint32_t speed = 1000000;
//static uint16_t delay = 0;
static uint8_t CS_AOUT1 = 128;
static uint8_t CS_AOUT2 = 64;
static uint8_t CH_AOUT1 = 0;
static uint8_t CH_AOUT2 = 1;
static uint8_t CS_AOUT3 = 32;
static uint8_t CS_AOUT4 = 16;
static uint8_t CS_IN1 = 8;
static uint8_t CH_IN1 = 0;
static uint8_t CH_IN2 = 1;
static uint8_t CH_IN3 = 2;
static uint8_t CH_IN4 = 3;
static uint8_t CH_IN5 = 4;
static uint8_t CH_IN6 = 5;
static uint8_t CH_IN7 = 6;
static uint8_t CH_IN8 = 7;
static uint8_t CS      = 11;
int HIGH = 1;
int LOW = 0;
int Max_Volt_Value = 4095;
int Max_Volt = 10;
int Board = 0;
int fd = 0;
int fdr=0;
static uint16_t delay=0;

static void chipselect(int Board, int8_t Channel, int8_t Value)
// Chipselect wir durch den Baustein 23S08 vorgenommen. Bitte mit der Doku des Bausteins vergleichen
{
    uint8_t outwert=0;
    uint8_t wr_bufh[]={0x40, 0x09, 252};
    uint8_t wr_bufl[]={0x40, 0x09, (254)-(Channel)};
    if (Value == 1)
    {
        wr_bufl[2] = 252;
    }
    unsigned char buf[3];
    int ret;
    struct spi_ioc_transfer xfer;
```

```

struct spi_ioc_transfer tr = {
    .tx_buf = (unsigned long)wr_buf1,
    .rx_buf = (unsigned long)buf,
    .len = ARRAY_SIZE(wr_buf1),
    .delay_usecs = delay,
    .speed_hz = speed,
    .bits_per_word = bits,
};

ret = ioctl(fd, SPI_IOC_MESSAGE(1), &tr);
}

static float GetVolt(int Board, int8_t Chip, int8_t Channel, int Maxvolt, int Maxvoltvalue)
// Die analoge Spannung wird durch den MCP3008 gemessen.
{
    float returnval;
    int retval;
    int resultat,ret;
    char c;
    unsigned char buf[3];
    uint8_t wr_buf[]={1,(8+Channel)<<4,0};
    struct spi_ioc_transfer tr = {
        .tx_buf = (unsigned long)wr_buf,
        .rx_buf = (unsigned long)buf,
        .len = ARRAY_SIZE(wr_buf),
        .delay_usecs = delay,
        .speed_hz = speed,
        .bits_per_word = bits,
    };

    chipselect(Board,Chip,0);
    ret = ioctl(fd, SPI_IOC_MESSAGE(1), &tr);
    chipselect(Board,Chip,1);
    retval= (((buf[1] & 3) << 8) + buf[2]);
    returnval = (float)retval;
    returnval = returnval*Maxvolt/Maxvoltvalue;
    return returnval;
}

static void SetVolt(int Board, int8_t Chip, int8_t Channel, float Value, int Maxvolt, int Maxvoltvalue)
// Die Ausgangsspannung wird durch die 4 MCP4922 vorgegeben.
{
    int Outwert = 0;
    int ret = 0;
    unsigned char buf[3];
    Outwert = (int)(Value * Maxvoltvalue/Maxvolt);

    if (Outwert >=Maxvoltvalue)
    {
        Outwert = Maxvoltvalue;
    }
    uint8_t wr_buf[]={{(Channel<<7) | (3<<4)}+(Outwert>>8),Outwert & 255};
    struct spi_ioc_transfer tr = {
        .tx_buf = (unsigned long)wr_buf,
        .rx_buf = (unsigned long)buf,
        .len = ARRAY_SIZE(wr_buf),
        .delay_usecs = delay,
        .speed_hz = speed,
        .bits_per_word = bits,
    };

    chipselect(Board,Chip,0);

    ret = ioctl(fd, SPI_IOC_MESSAGE(1), &tr);
    chipselect(Board,Chip,1);
}

```

```

int main(int argc, char **argv)
{
    int i;
    int ret = 0;
    float output = 0;
    uint8_t wr_buf[]={0x40, 0x00, 0x00};
    float readval;

    fd = open(device, O_WRONLY);
    fdr = open(device2, O_RDONLY);
    if (fd<=0) {
        printf("Device %s not found\n", device);
        exit(1);
    }
    write(fd,wr_buf,ARRAY_SIZE(wr_buf)); //to init the AD ports. Later I use ioctl instead of write
command.

    while(1)
    {
        output=output + 0.01;
        if (output >= 10)
        {
            output = 0;
        }
        SetVolt(Board,CS_AOUT1,CH_AOUT1,output,Max_Volt, Max_Volt_Value);
        readval = GetVolt(Board,CS_IN1,CH_IN1,24, 1024);
        printf("value: %.2f\n\n",readval);
    }
    close(fd);
    return ret;
}

```