

MIKROKONTROLLER & I²C BUS

by AS

www.makerconnect.de

<https://www.makerconnect.de/resource>

makerconnect.de

Drehgeber mit RGB LED

Drehgeber /
Encoder



Copyright

Sofern nicht anders angegeben, stehen die Inhalte dieser Dokumentation unter einer „Creative Commons - Namensnennung-NichtKommerziell-Weitergabe unter gleichen Bedingungen 3.0 DE Lizenz“



Sicherheitshinweise

Lesen Sie diese Gebrauchsanleitung, bevor Sie diesen Bausatz in Betrieb nehmen und bewahren Sie diese an einem für alle Benutzer jederzeit zugänglichen Platz auf. Bei Schäden, die durch Nichtbeachtung dieser Bedienungsanleitung verursacht werden, erlischt die Gewährleistung / Garantie. Für Folgeschäden übernehmen wir keine Haftung! Bei allen Geräten, die zu ihrem Betrieb eine elektrische Spannung benötigen, müssen die gültigen VDE-Vorschriften beachtet werden. Besonders relevant sind für diesen Bausatz die VDE-Richtlinien VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860. Bitte beachten Sie auch nachfolgende Sicherheitshinweise:

- Nehmen Sie diesen Bausatz nur dann in Betrieb, wenn er zuvor berührungssicher in ein Gehäuse eingebaut wurde. Erst danach darf dieser an eine Spannungsversorgung angeschlossen werden.
- Lassen Sie Geräte, die mit einer Versorgungsspannung größer als 24 V- betrieben werden, nur durch eine fachkundige Person anschließen.
- In Schulen, Ausbildungseinrichtungen, Hobby- und Selbsthilfewerkstätten ist das Betreiben dieser Baugruppe durch geschultes Personal verantwortlich zu überwachen.
- In einer Umgebung in der brennbare Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können, darf diese Baugruppe nicht betrieben werden.
- Im Falle einer Reparatur dieser Baugruppe, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen. Eine Reparatur des Gerätes darf nur von fachkundigen Personen durchgeführt werden.
- Spannungsführende Teile an dieser Baugruppe dürfen nur dann berührt werden (gilt auch für Werkzeuge, Messinstrumente o.ä.), wenn sichergestellt ist, dass die Baugruppe von der Versorgungsspannung getrennt wurde und elektrische Ladungen, die in den in der Baugruppe befindlichen Bauteilen gespeichert sind, vorher entladen wurden.
- Sind Messungen bei geöffnetem Gehäuse unumgänglich, muss ein Trenntrafo zur Spannungsversorgung verwendet werden
- Spannungsführende Kabel oder Leitungen, mit denen die Baugruppe verbunden ist, müssen immer auf Isolationsfehler oder Bruchstellen kontrolliert werden. Bei einem Fehler muss das Gerät unverzüglich ausser Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- Es ist auf die genaue Einhaltung der genannten Kenndaten der Baugruppe und der in der Baugruppe verwendeten Bauteile zu achten. Gehen diese aus der beiliegenden Beschreibung nicht hervor, so ist eine fachkundige Person hinzuzuziehen

Bestimmungsgemäße Verwendung

- Auf keinen Fall darf 230 V~ Netzspannung angeschlossen werden. Es besteht dann Lebensgefahr!
- Dieser Bausatz ist nur zum Einsatz unter Lern- und Laborbedingungen konzipiert worden. Er ist nicht geeignet, reale Steuerungsaufgaben jeglicher Art zu übernehmen. Ein anderer Einsatz als angegeben ist nicht zulässig!
- Der Bausatz ist nur für den Gebrauch in trockenen und sauberen Räumen bestimmt.
- Wird dieser Bausatz nicht bestimmungsgemäß eingesetzt kann er beschädigt werden, was mit Gefahren, wie z.B. Kurzschluss, Brand, elektrischer Schlag etc. verbunden ist. Der Bausatz darf nicht geändert bzw. umgebaut werden!
- Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller, sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und /oder Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.
- Der Autor dieses Tutorials übernimmt keine Haftung für Schäden. Die Nutzung der Hard- und Software erfolgt auf eigenes Risiko.

Drehgeber / Encoder

Für die Bedienung an meinem Prozessor verwende ich einen

Rotary Encoder - Illuminated (RGB)

(So steht die Bezeichnung beim Verkäufer)

Einen genauen Typ oder Hersteller ist mir nicht bekannt. Nach der Schrift im Datenblatt, kommt er aus China.

Drehkodierer (Rotary Encoders) sind ähnlich im Einsatz wie Potentiometer, im Unterschied zu Potentiometern aber arbeiten die Encoder mit voller Drehung und ohne Beschränkung. Die Ausgabe erfolgt jedoch in Gray Codes welche Ausschluss über das Ausmass und die Richtung der Drehung geben können. Dieser Encoder ist speziell cool wegen der integrierten RGB LED (gemeinsame Anode). Ebenso verfügt der Encoder über einen Druckschalter.

(Angaben des Verkäufers Bointec)



Encoder mit RGB LED auf P98 und Wannenstecker

Funktionen:

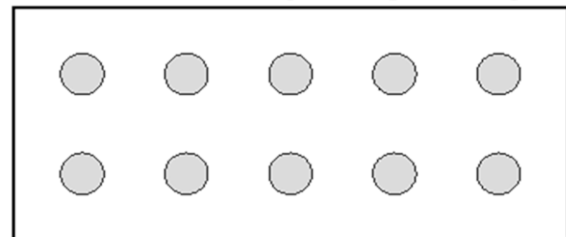
- Schalter Weg: 0.5mm
- Durchmesser Achse: 6.0mm
- Achsen Länge: 18mm
- 90° Montage
- RGB LED
- Druckschalter

Die Belegung des Wannensteckers auf der Platine P98 habe ich noch mal extra dargestellt. Es können auch andere Typen angeschlossen werden. Dazu muss im Programm die Angabe der Pins und Ports geändert werden.

Die genaue Funktion eines Encoders ist im Netz auf diesen Seiten dargestellt.

Belegung Wannenstecker

2 4 6 8 10

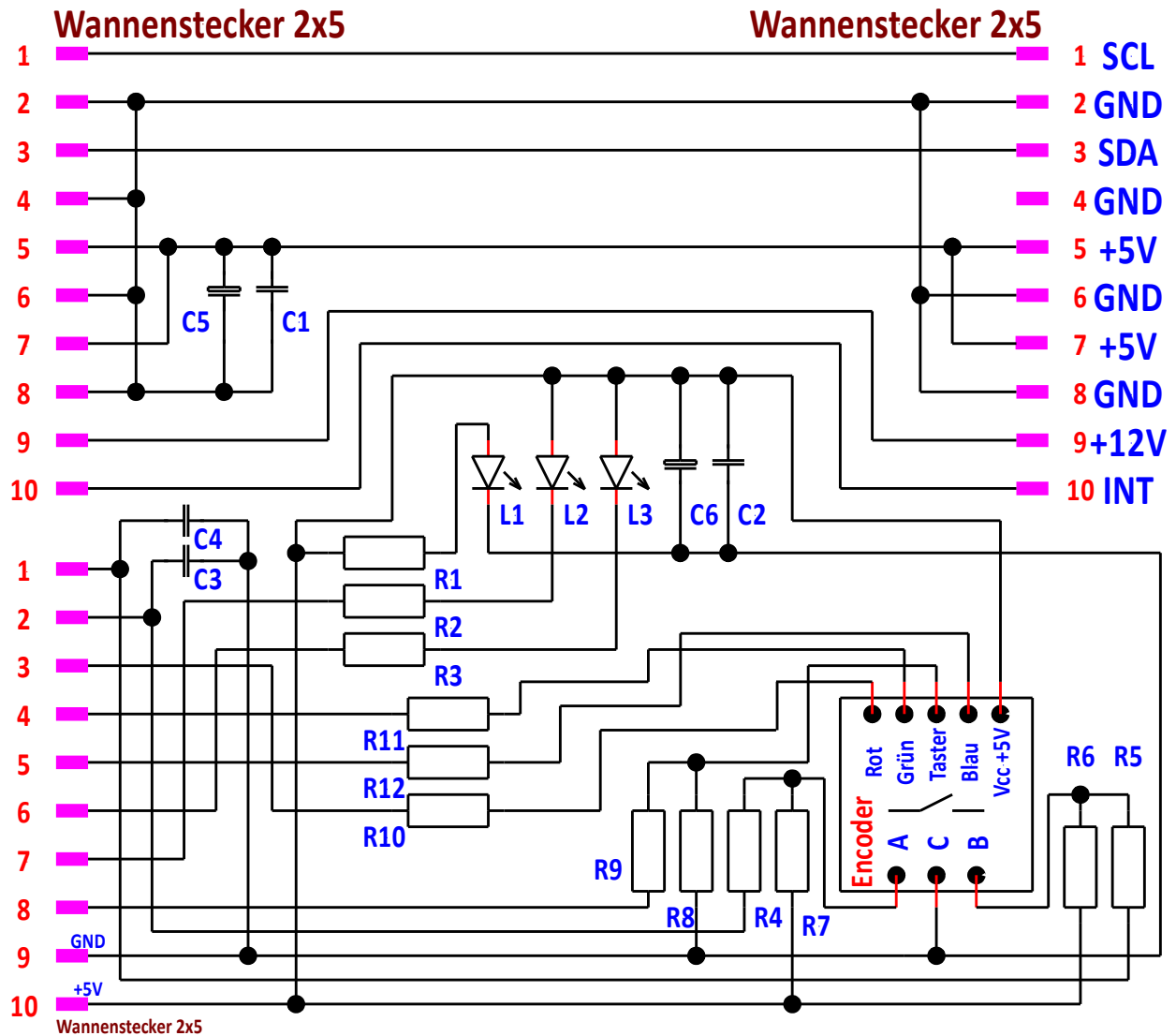


1 3 5 7 9

- | | |
|---------------|-------------------|
| 1 - Encoder B | 6 - LED 3 |
| 2 - Encoder A | 7 - LED 2 |
| 3 - LED Rot | 8 - Taster |
| 4 - LED Grün | 9 - GND (Masse) |
| 5 - LED Blau | 10 - Vcc (+5V) |

(Ansicht von oben)

<https://www.mikrocontroller.net/articles/Drehgeber>



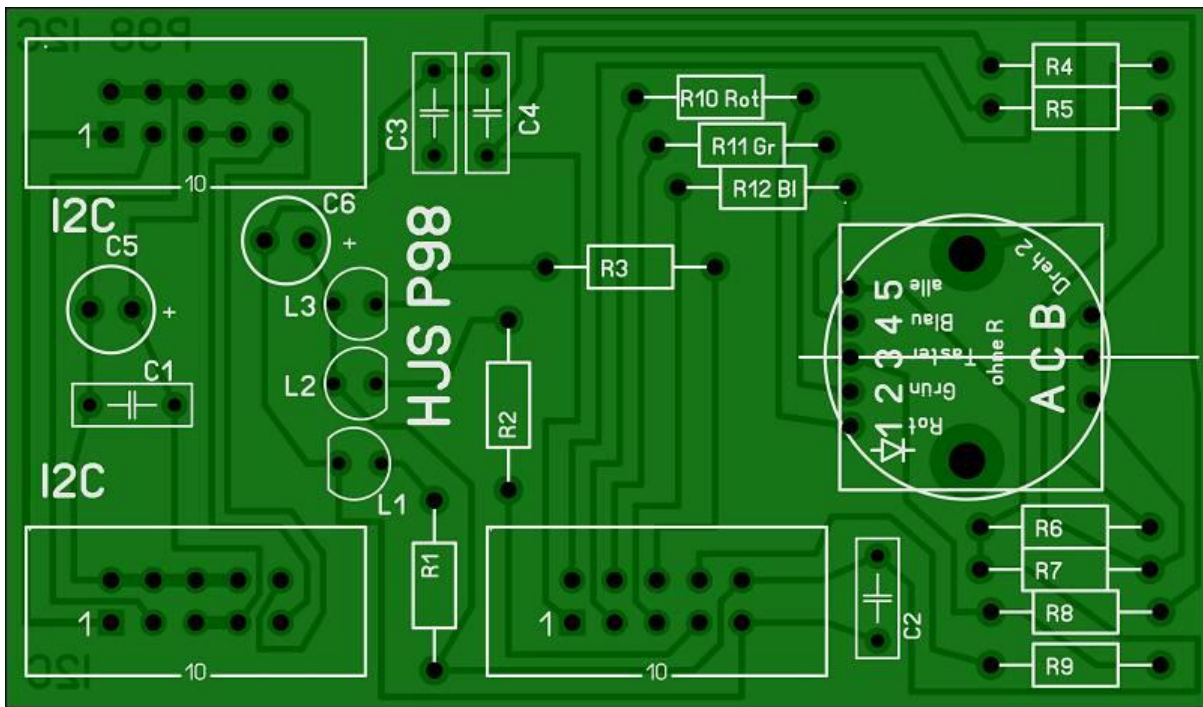
Schaltbild P98 Encoder

Bauteile:

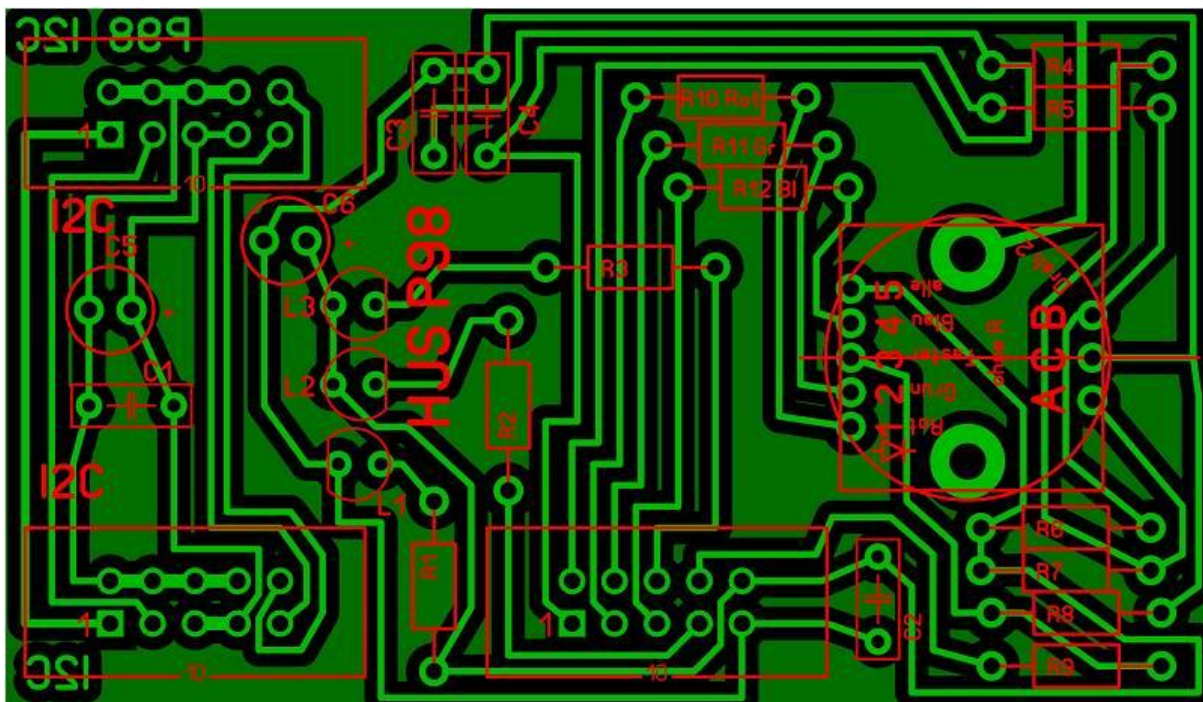
L1 - L3 - LED, 3 oder 5mm, 20 mA
R1 - R3 - Widerstand 220 Ohm
R4 - R7 - Widerstand 10 kOhm
R9 - Widerstand 220 Ohm
R11 - Widerstand 100 Ohm (Grün)
C1, C2 - Kondensator 100 nF
C5, C6 - Elko 100/16
Halterung - 2 teilig

1 x Platine ca. 72 x 42 mm (P98)
3 x Wannenstecker 2x5 (Port, I²C)
R8 - Widerstand 10 kOhm
R10 - Widerstand 180 Ohm (Rot)
R12 - Widerstand 100 Ohm (Blau)
C3, C4 - Kondensator 10 nF
 Encoder mit LED RGB

Im Datenblatt des Herstellers sind die Daten für die internen LED angegeben. Die erforderlichen Widerstände muss man sich dann ausrechnen. Wird ein anderer Encoder verwendet sind die Widerstände entsprechend anzupassen.
 Der Taster im Encoder liegt ebenfalls auf Vcc +5V.



Bestückungsseite P98 Encoder mit Aufdruck



Leiterzugseite P98 Encoder

Als Beispiel habe ich wieder ein Programm geschrieben. Es nutzt die Auswertung von Peter Danegger. Den Timer habe ich anders eingesetzt. Als Auswertung schalte ich die LEDs im Encoder einzeln oder zusammen. Die Auswertung kann man jederzeit seinem eigenen Programm anpassen, z.B. als Einstellung, Bedienung eines Menüs oder als Einknopfbedienung. Die Funktion kann man aus den Kommentarzeilen entnehmen.

```
// ATB_Enc_Prg_1.c * Created: 10.12.2016 21:51:28 Author : AS
// Auswertung Drehgeber Peter Dannegger
// ATmega 1284p auf Board 1, 16 MHz, Drehgeber auf P98 auf Port A, NT2,

#define F_CPU 16000000 // CPU clock in Hz
#include "avr/io.h"
#include <avr/interrupt.h>

int8_t dreh = 8; // Angabe der max Stellungen
int8_t val = 0; // Startwert der Drehung

#define PHASE_A (PINA & 1<<PA0) // Eingang Encoder A P98
#define PHASE_B (PINA & 1<<PA1) // Eingang Encoder B P98

volatile int8_t enc_delta; // -128 ... 127
static int8_t last;

void encode_init( void )
{
    int8_t new;
    new = 0;
    if( PHASE_A ) new = 3;
    if( PHASE_B ) new ^= 1; // convert gray to binary
    last = new; // power on state
    enc_delta = 0;
}

ISR( TIMERO_COMPA_vect ) // 1ms for manual movement
{
    int8_t new, diff;
    new = 0;
    if( PHASE_A ) new = 3;
    if( PHASE_B ) new ^= 1; // convert gray to binary
    diff = last - new; // difference last - new
    if( diff & 1 )
    { // bit 0 = value (1)
        last = new; // store new as next last
        enc_delta += (diff & 2) - 1; // bit 1 = direction (+/-)
    }
}

int8_t encode_read1( void ) // einzel Schritt Encoder
{
    int8_t val;
    cli();
    val = enc_delta;
    enc_delta = 0;
    sei();
    return val;
}

int8_t encode_read2( void ) // zwei Schritt Encoder
{
    int8_t val;
    cli();
```

```

    val = enc_delta;
    enc_delta = val & 1;
    sei();
    return val >> 1;
}

int8_t encode_read4( void )           // vier Schritt Encoder
{
    int8_t val;
    cli();
    val = enc_delta;
    enc_delta = val & 3;
    sei();
    return val >> 2;
}

void timer_init()                     // Timer 1ms
{
    TCCR0A = (1<<WGM01);
    TCCR0B = (1<<CS01)|(1<<CS00);
    OCR0A = 249;
    TIMSK0 |= (1<<OCIE0A);
}

int main( void )                      // Hauptschleife
{
    timer_init();
    DDRA |= 0b11111100 ;              // LED auf Ausgang
    PORTA = 0b01111100;               // LED auf aus bei 1
    encode_init();
    sei();
    while(1)
    {
        val += encode_read4();         // Auswahl 1,2 oder 4 Schritt Encoder
        if (val>dreh)                  // Begrenzung oben
        {
            val=0;
        }
        if (val<0)                     // Begrenzung unten
        {
            val=dreh;
        }
        // Beginn Auswertung
        if (val==1)                     // LED blau
        {
            PORTA &= ~(1<<PA4);        // LED Blau ein
        }
        else
        {
            PORTA |= (1<<PA4);          // LED Blau aus
        }
        if (val==2)                     // LED grün
        {
            PORTA &= ~(1<<PA3);        // LED grün ein
        }
    }
}

```



```
}
else
{
    PORTA |= (1<<PA3);           // LED grün aus
}
if (val==3)                      // LED rot
{
    PORTA &= ~(1<<PA2);          // LED rot ein
}
else
{
    PORTA |= (1<<PA2);           // LED rot aus
}
if (val==5)
{
    PORTA &= ~(1<<PA3);          // LED grün ein
    PORTA &= ~(1<<PA4);          // LED blau ein
}
if (val==6)
{
    PORTA &= ~(1<<PA2);          // LED rot ein
    PORTA &= ~(1<<PA4);          // LED blau ein
}
if (val==7)
{
    PORTA &= ~(1<<PA2);          // LED rot ein
    PORTA &= ~(1<<PA3);          // LED grün ein
}
if (val==8)
{
    PORTA &= ~(1<<PA2);          // LED rot ein
    PORTA &= ~(1<<PA3);          // LED grün ein
    PORTA &= ~(1<<PA4);          // LED blau ein
}
}
```

Es wurden von Peter 3 verschiedene Encoder vorgesehen. Die Auswahl kann durch

`val += encode_read4();`

erfolgen. Dazu einfach die **4** in **1** oder **2** austauschen.

Dabei muss die Funktion des Encoders getestet werden.

Einige Teile des Textes wurden zur besseren Übersicht **farblich** gestaltet.

Die Nutzung erfolgt auf eigenes Risiko.

Ich wünsche viel Spaß beim Bauen und programmieren.

Achim

myroboter@web.de