

# MIKROKONTROLLER & I<sup>2</sup>C BUS

by AS

[www.makerconnect.de](http://www.makerconnect.de)

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

makerconnect.de

AT 42 QT 1070  
Eine „Bedienung“ mal anders  
( Ohne Tasten )

Q - Touch



## 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.



## Q Touch mit dem AT 42 QT 1070

### Eine „Bedienung“ mal anders ( ohne Tasten )

Für die Eingabe von Informationen in Zusammenhang mit einem Display oder Einstellung von Werten sind Taster, Drehgeber oder Potentiometer nicht immer die schönste oder auch beste Lösung. Entweder gibt es Probleme mit der Platzierung, der Größe oder die kleinen Spalten machen Probleme mit der Feuchtigkeit der Umgebung.

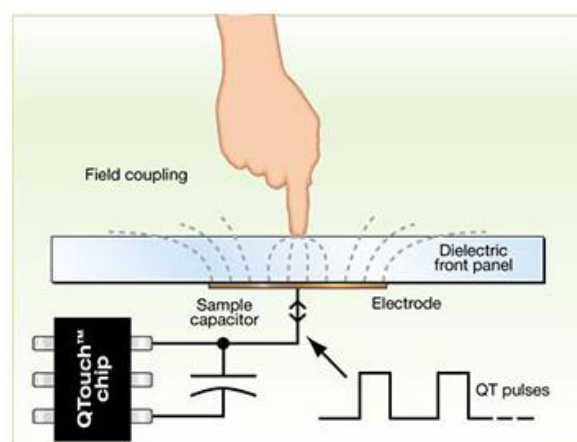
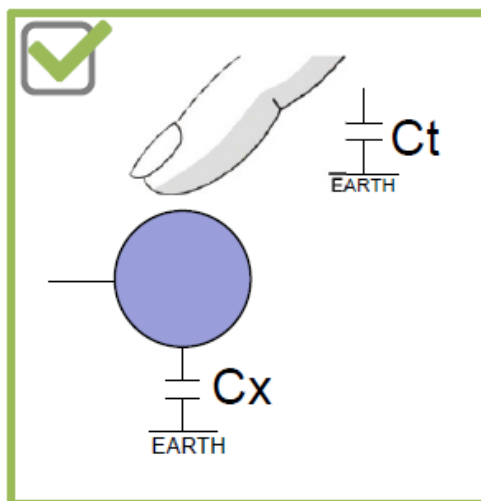
Möchte man eine optisch ansprechende Lösung realisieren, so sind Touch Sensoren bestens für solch eine Bedienung geeignet. Diese können hinter den verschiedensten Frontplatten, wie Aluminium, Plexiglas oder Fliesen angebracht werden und detektieren ganz unsichtbar die Berührung.

Die beschriebene Methode ermittelt die Berührung nicht über eine Kapazitätsmessung, sondern wird über eine Charge-Transfer-Messung detektiert.

Modul Q-Touch mit dem  
AT42QT1070 und 7 Keys  
( Berührungsflächen )  
Platine P 92

Die Platine P92 verwende ich in  
meinem Modularen System zum  
Anschluss im I<sup>2</sup>C Bus.

Doch zunächst wollen wir uns die  
Bauart und Funktion genauer  
ansehen.



Der IC vom Typ AT42QT1070 detektiert die Berührung unseres Fingers mit der „Platine“. Auf der Oberseite befindet sich eine Angabe des Ortes bzw. der Größe unserer Sensorfläche. Auf der Unterseite der Platine befindet sich der eigentliche Sensor.

## Funktionsprinzip

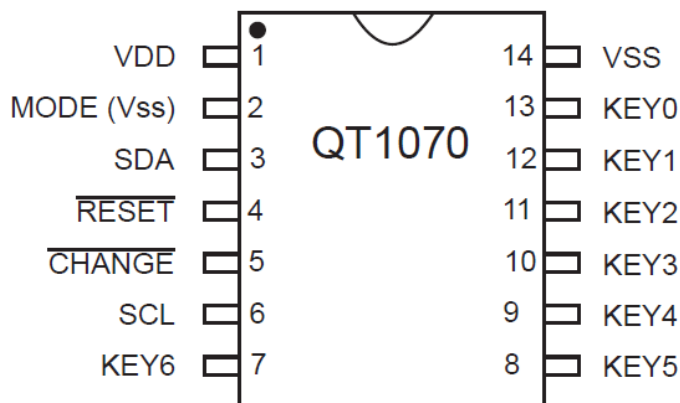
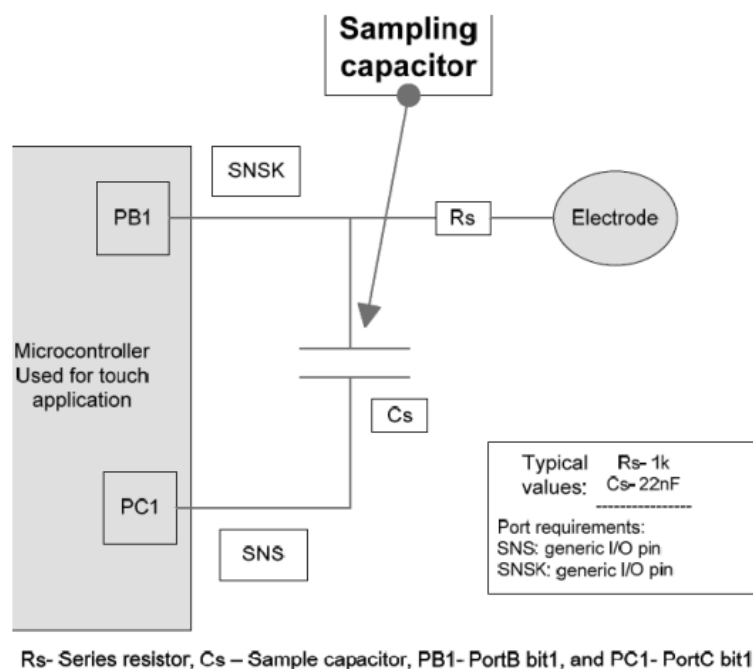
Es gibt 2 Kondensatoren die als Speicher für Ladungen dienen, den Kondensator  $C_s$  als Speicher und die "Touch" Elektrode als Kondensator gegen Erde.

Am Anfang werden beide entladen, um die Kondensatoren in einen definierten Zustand zu bringen. Das bedeutet, dass PB1 und PC1 als Ausgänge und auf LOW geschaltet werden. Etwas warten damit Entladung auch vollständig erfolgt. Danach werden in einer Schleife folgende Punkt abgearbeitet:

1. Der Touch Kondensator wird über PB1 aufgeladen (nur wenige pf). Dabei ist der Port PC1 ist als Eingang geschaltet, damit in den  $C_s$  kein Strom fließen kann
2. Pin PB1 wird als Eingang konfiguriert und PC1 als Ausgang mit 0V. Jetzt wandert die Ladung aus dem Touch Kondensatorin  $C_s$  (welcher viel grösser ist und sich nur gering auflädt)
3. Wenn die Spannung am  $C_s$  ausreicht um vom AVR als 1 gelesen zu werden, wird die Schleife beendet und durch geführte Schleifenanzahl ausgegeben. Der Zählerstand ist ein Maß für die benötigten Ladezyklen. Wenn nicht beginnt die Schleife von vorne.

Der Endstand des Zählers verändert sich, wenn man mit dem Finger den Touchsensor berührt.

Sozusagen wird durch den kleineren Touch-Kondensator schrittweise der große Kondensator  $C_s$  aufgeladen. Die Anzahl der benötigten Ladezyklen hängt von der "Größe" des Touch Kondensator ab. Und die wiederum von dessen Umgebung (Finger).



Sehen wir uns als nächste IC selber an. Er wird in einem SO14 Gehäuse angeboten. Als DIP gibt es ihn nicht.

**Achtung SMD !**

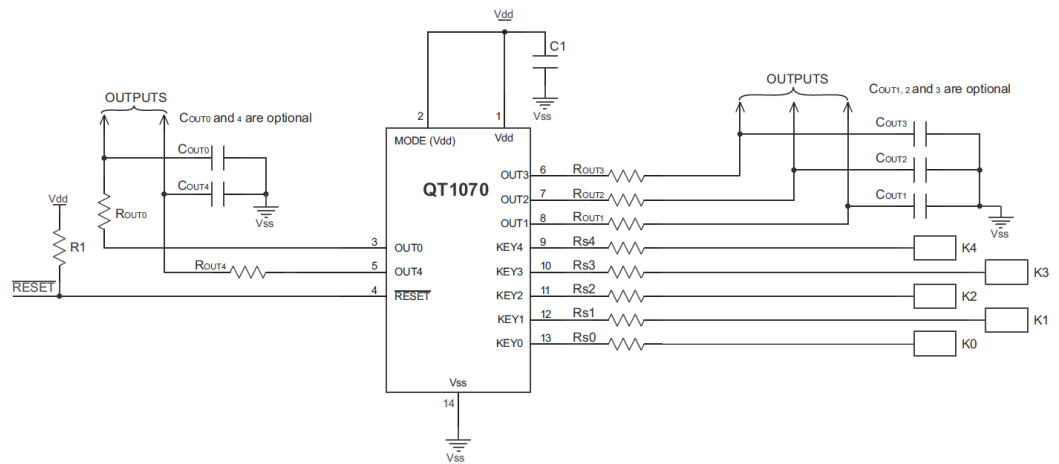
Bitte die Bauart genau beachten.

## Verwendung:

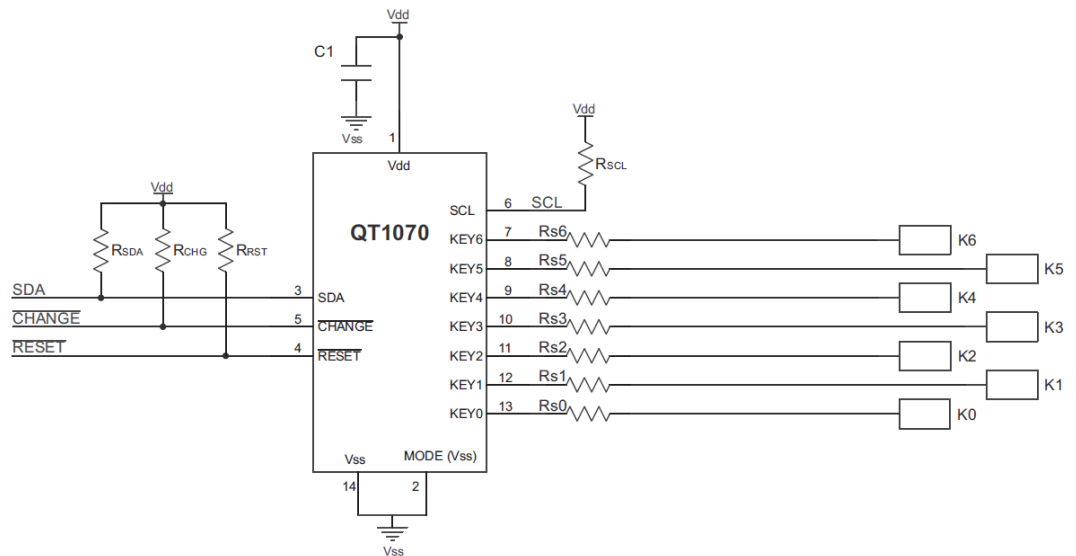
Der AT42QT1070 kann in zwei verschiedenen Modi betrieben werden:

- Standalone Mode - autarker Betrieb von 5 Keys und 5 Ausgängen zum Anschluss von z.B. LEDs
- Comms Mode - bis zu 7 Keys (Tasten), Auswertung über den I<sup>2</sup>C Bus (unser Modul)

## Standalone Mode mit 5 Keys und 5 Ausgängen



## Comms Mode mit 7 Keys und I<sup>2</sup>C Bus



## Zuordnung der Register zu den Angaben

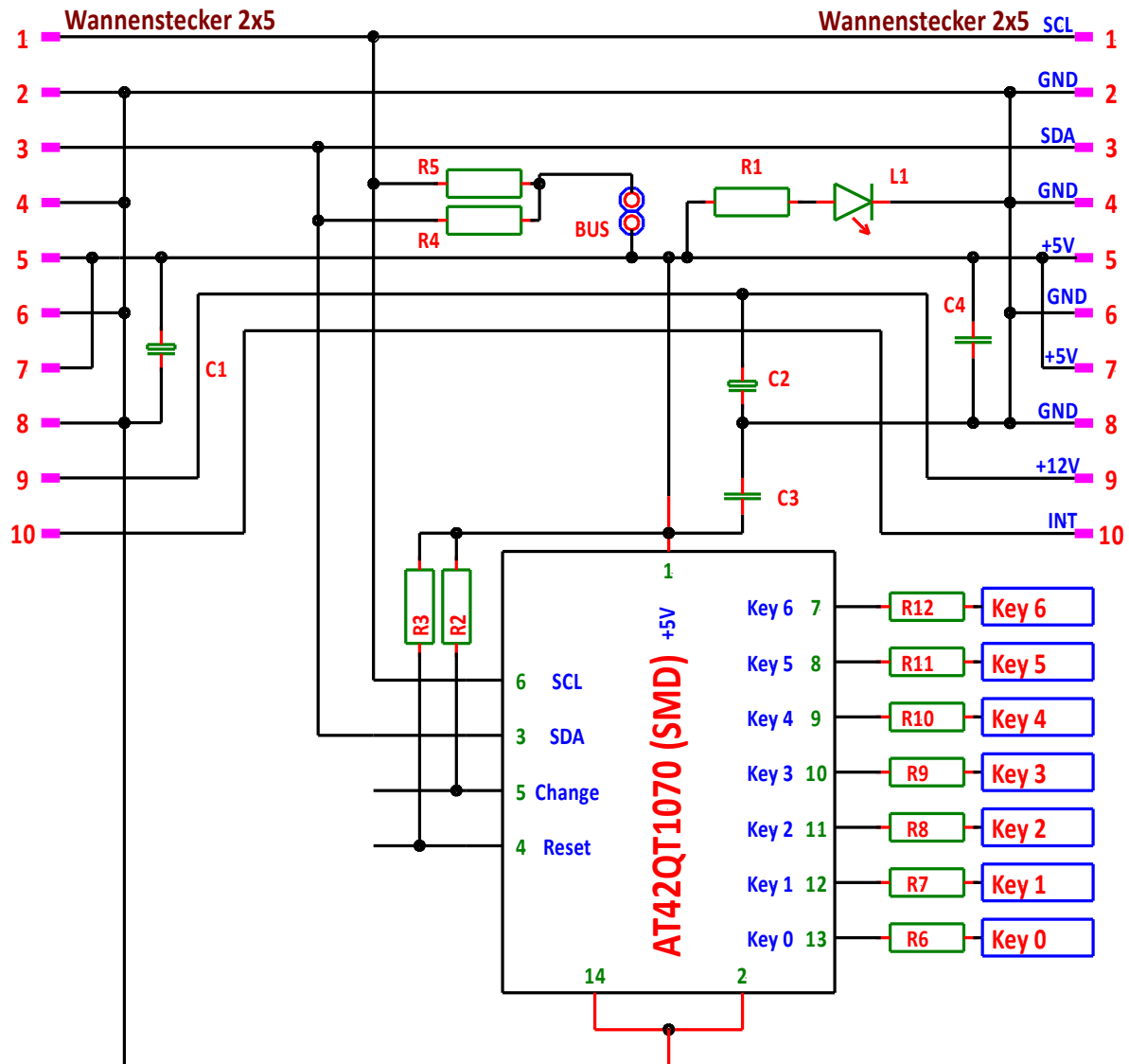
3	Key status	Reserved	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1	Key 0	R
4 – 5	Key signal 0	Key signal 0 (MSByte) – Key signal 0 (LSByte)								R
6 – 7	Key signal 1	Key signal 1 (MSByte) – Key signal 1 (LSByte)								R
8 – 9	Key signal 2	Key signal 2 (MSByte) – Key signal 2 (LSByte)								R
10 – 11	Key signal 3	Key signal 3 (MSByte) – Key signal 3 (LSByte)								R
12 – 13	Key signal 4	Key signal 4 (MSByte) – Key signal 4 (LSByte)								R
14 – 15	Key signal 5	Key signal 5 (MSByte) – Key signal 5 (LSByte)								R
16 – 17	Key signal 6	Key signal 6 (MSByte) – Key signal 6 (LSByte)								R

Als letztes habe ich eine Kopie aus dem Datenblatt des Herstellers eingefügt. Es werden die einzelnen Register angegeben. Es sind weitere Einstellungen möglich, auf die ich nicht weiter eingehe. Wir wollen uns nur mit dem Comms Mode beschäftigen. In der Schaltung ist ein kleiner Fehler. Am Pin 6 liegt SCL an und wird über den Widerstand  $R_{SCL}$  nach +5V gelegt. Dieser Pin muss ebenfalls nach aussen geführt werden.

## Technische Angaben

- Betriebsspannung von 1,8V bis 5,5V
- I<sup>2</sup>C Bus max. 400 kHz
- Bus Adresse (nach Hersteller) 0x1B - verwendet 0x36

Von verschiedenen Herstellern werden unterschiedliche Standalone Module angeboten.



Q Touch Modul mit dem AT42QT1070

## Bauteile QTouch Platine P92

**L1** - LED, 20 mA, 3 oder 5 mm

**R2,4,5,6-12** - Widerstand 10 KOhm

**C1,2** - Elko 100/16

**2** x Wannenstecker 2x5, RM 2,54

**1** x Jumper

**1** x AT 42 QT 1070 ( **SMD** )

**R1** - Widerstand 220 Ohm

**R3** - Widerstand 67 kOhm

**C3,4** - Kondensator 100 nF

**1** x Steckerleisten 2-polig

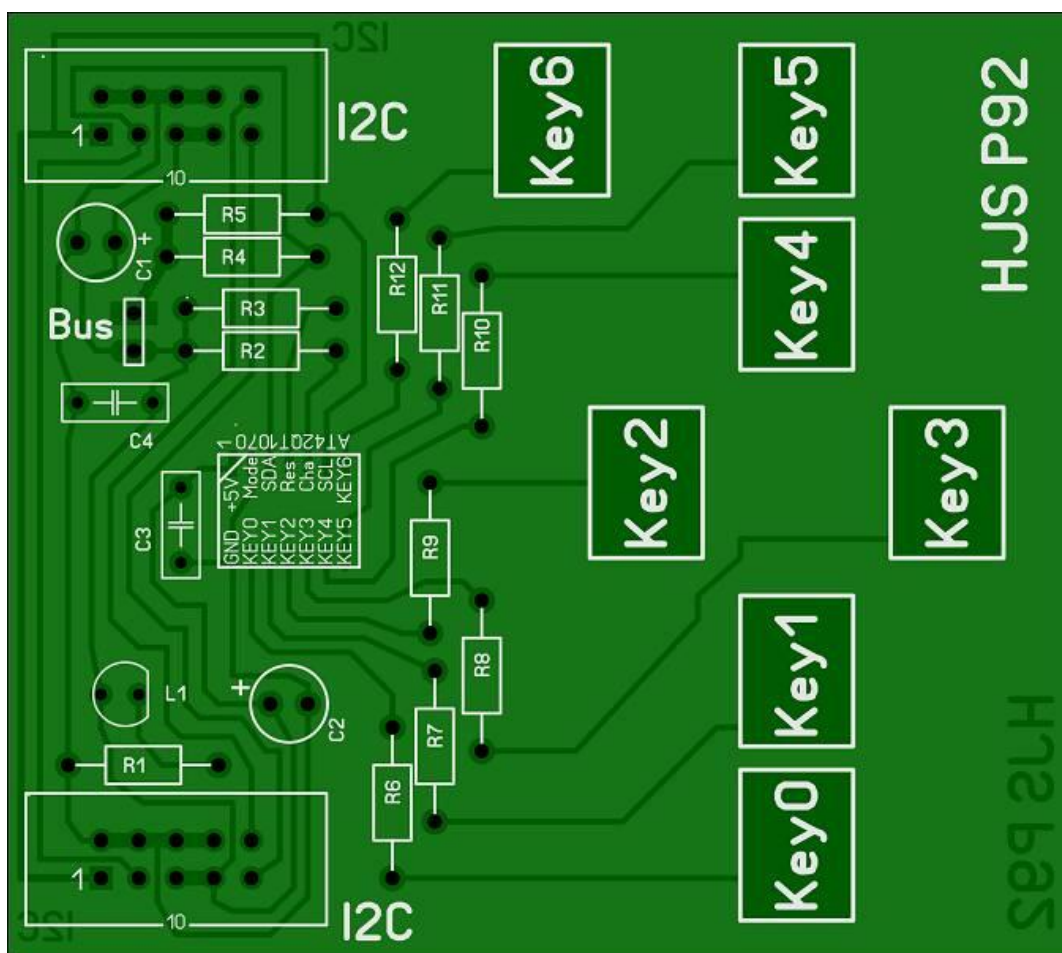
**1** x Platine P92 ( 72 x 64 mm )

Die Pins 4 und 5 werden nur mit Widerständen an +5V angeschlossen.

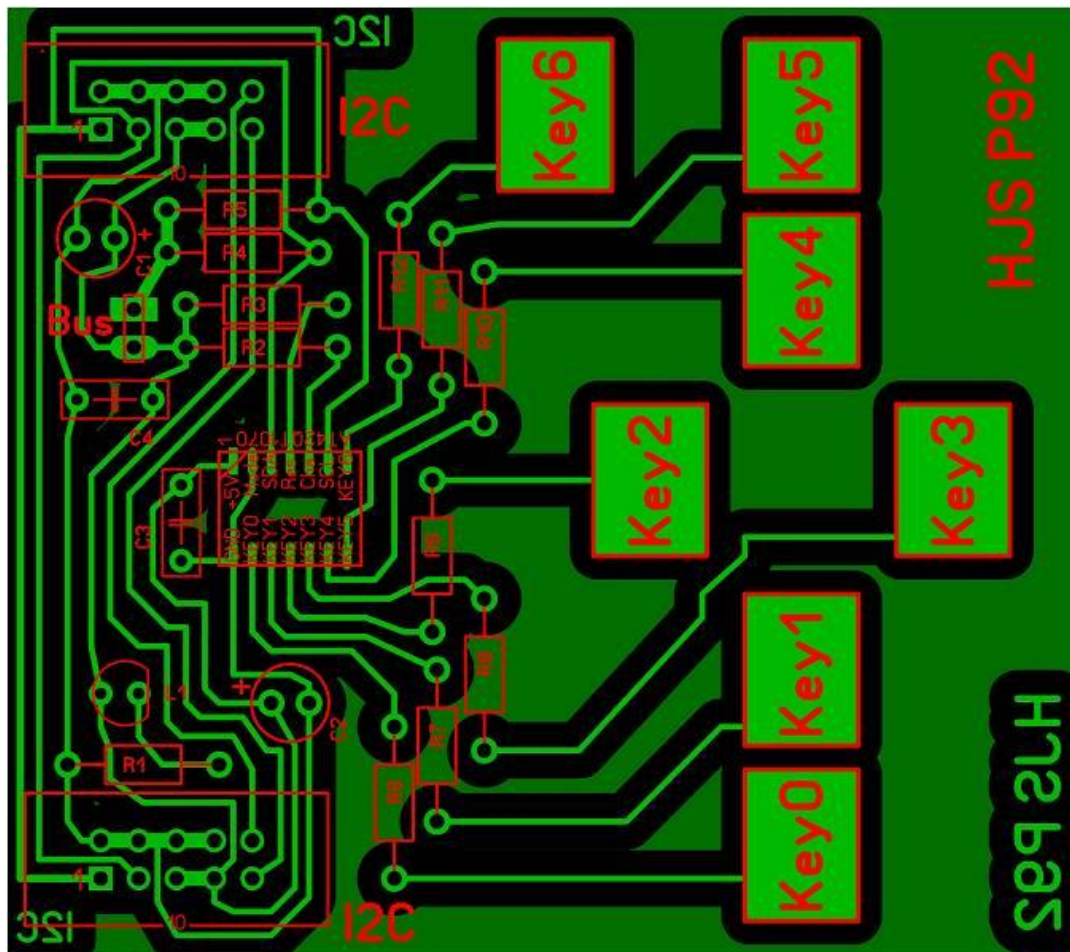
Die Belegung der Steckbuchsen entspricht dem RN Standard.

Die Grösse und Lage der Keys bitte der Platinen Zeichnung entnehmen.

Platine in der Fotoansicht



Platine in der Durchsicht





```

/* ATB_Touch_Prg_1.c Created: 27.03.2016 17:37:48 Author : AS */
// Hardware D2 - Display I2C, P92 - AT42QT1070
// P30 - Board 1 mit AT1284p mit 16MHz, NT2 mit 5/12V 3A

#include <stdbool.h>
#include <avr/pgmspace.h>
#include "main.h"
#include <util/delay.h>
#include "i2clcd.h"
#include "i2cmaster.h"
#include "avr/io.h"
#include "util/delay.h"
#include "avr/interrupt.h"
#include <stdlib.h>
int8_t d;
int8_t dt;
uint8_t ret; // Variable für I2C Kommunik.
char Buffer[20]; // Variable für LCD Anzeige

void anzeige()
{
    lcd_printlc(1,2,"Touch + Display"); // Zeile 1
    lcd_printlc(2,3,"Touch Modul 1"); // Zeile 2
    lcd_printlc(3,2,"mit AT42QT1070"); // Zeile 3
    lcd_printlc(4,2,"(by achim 2016)"); // Zeile 4
}

void erfassung()
{
    ret = i2c_start(0x36); // Start
    if (ret == 0)
    {
        i2c_start( 0x36 ); // Addressiere Device
        i2c_write( 0x03 ); // Setze lesezugriff ori 0x03
        // bei 0x02 ändert von 0 auf 1 bei jeder Taste - Detection Status
        // bei 0x03 gibt jede Taset einen Bit Wert aus - Key Status
        // bei 0x05 Anzeige Empfindlichkeit Key 0
        // bei 0x07 Anzeige Empfindlichkeit Key 1
        // bei 0x09 Anzeige Empfindlichkeit Key 2
        // bei 0x0B Anzeige Empfindlichkeit Key 3
        // bei 0x0D Anzeige Empfindlichkeit Key 4
        // bei 0x0F Anzeige Empfindlichkeit Key 5
        // bei 0x11 Anzeige Empfindlichkeit Key 6
        i2c_start( 0x36 | 0x01 ); // Starte Lesezugriff
        d=i2c_readNak(); // Wert auf d übertragen
        i2c_stop();
    }
    else // Fehlererkennung
    {
        // Wenn AT kein OK sendet
        lcd_command(LCD_CLEAR); // Leere Display
        _delay_ms(2); // Warte 2ms
        lcd_printlc(1,1,"Fehlermeldung:"); // "Lesevorgang"
        lcd_printlc(3,4,"AT42QT1070"); // "Nicht OK (NOK)"
    }
}

```



```

        lcd_printlc(4,2,"nicht OK");           // "Lesevorgang"
        _delay_ms(2000);                       // Warte 2ms
    }
}

int main(void)
{
    cli();                                     // Interrupts deaktiviert
    i2c_init();                               // Starte I2C Bus
    lcd_init();                               // Starte I2CLCD
    lcd_light(0);                             // 0=Licht an, 1=aus
    // Display Befehle
    lcd_command(LCD_DISPLAYON | LCD_CURSOROFF | LCD_BLINKINGOFF);
    lcd_command(LCD_CLEAR);                   // Leere Display
    _delay_ms(2);                             // Warte 2ms
    anzeige();
    _delay_ms(5000);                           // Warte 5000ms auf Start
    lcd_command(LCD_CLEAR);                   // Leere Display
    _delay_ms(2);                             // Warte 2ms
    while(1)
    {
        erfassung();                         // Unterprg Touch auslesen
        lcd_command(LCD_CLEAR);               // Leere Display
        _delay_ms(2);                         // Warte 2ms
        itoa(d, Buffer, 10 );                 // Umrechnung Werte
        lcd_printlc(3,14,Buffer);             // Anzeige Werte

        lcd_printlc(1,1,"Anzeige Keys:");
        lcd_printlc(3,1,"Key Wert:");
        lcd_printlc(4,1,"Taste Key:");

        if (d==1)                             // Auswertung
        { dt=0; }
        if (d==2)
        { dt=1; }
        if (d==4)
        { dt=2; }
        if (d==8)
        { dt=3; }
        if (d==16)
        { dt=4; }
        if (d==32)
        { dt=5; }
        if (d==64)
        { dt=6; }
        if ((d==1)||(d==2)||(d==4)||(d==8)||(d==16)||(d==32)||(d==64))
        {
            itoa(dt, Buffer, 10 );
            lcd_printlc(4,13,Buffer);
        }
        else
        {
            lcd_printlc(4,13,"---");
        }
    }
}

```

```
}  
  _delay_ms(100);  
}  
}
```

Ansicht des fertigen Moduls  
Q Touch mit der Platine P92



Ansicht des fertigen Moduls  
Q-Touch mit den Keys 0 - 6.  
Die genaue Grösse und  
Anordnung bitte der Platinen  
Zeichnung entnehmen

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](mailto:myroboter@web.de)