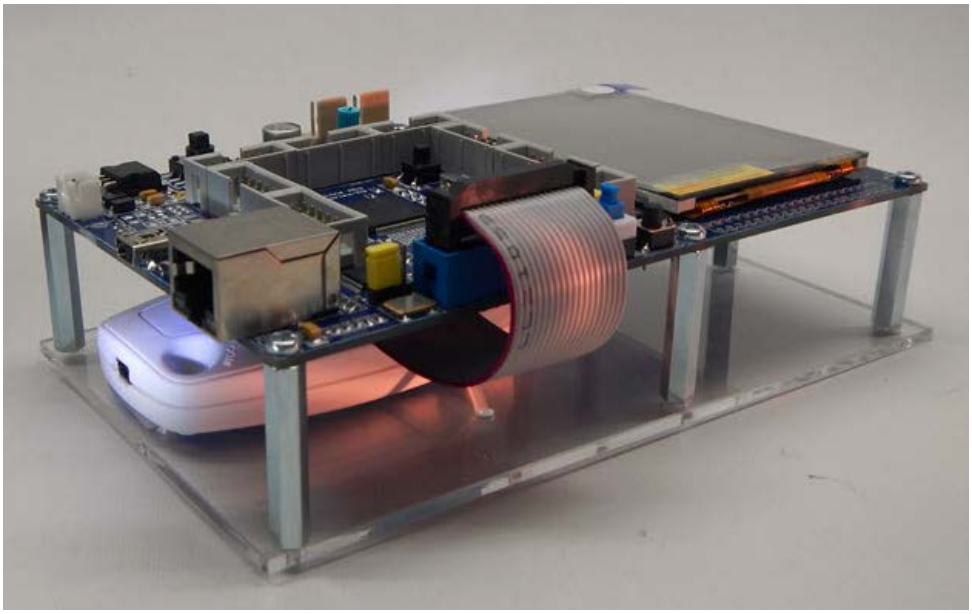
MCB32 APP

Logik Analyse am SPI Bus des Displays



MCB32 - Embedded Programmierung

Version: 1.01A

Bitte beachten. Diese Unterlagen können ohne Vorankündigung jederzeit angepasst, verbessert und erweitert werden. Wir bitten Sie Wünsche und auch Fehler zu melden. (info@mcb32.ch)

1 Inhaltsverzeichnis

2 Entwicklungsumgebung KEIL aufsetzen

Wir arbeiten mit der IDE der Firma Keil. Damit können wir die Programme bis zu 32kB Programmcode schreiben. Das genügt für die ersten Schritte während der Ausbildung.

Keil uVision5

IDE: Integrated Development Environment von http://www2.keil.com/mdk5/install

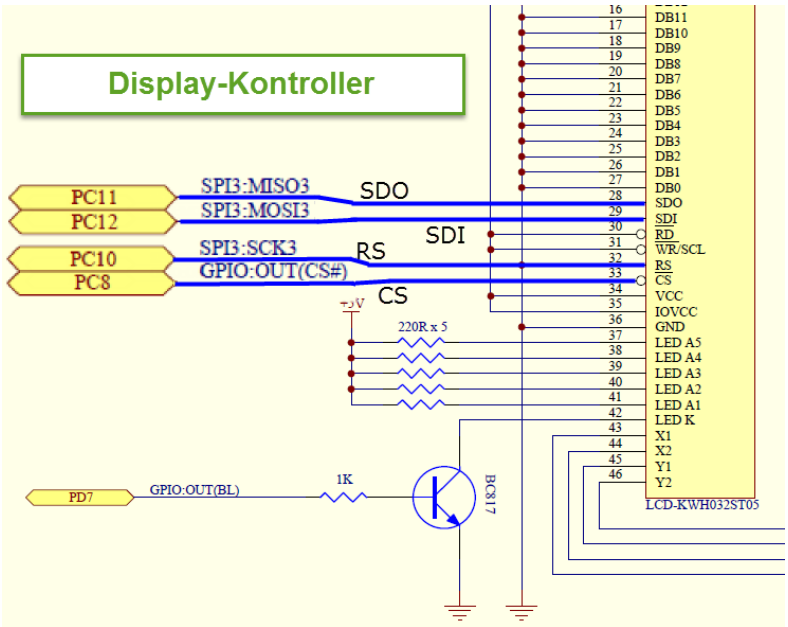
ARM: Advanced RISC Machines

2.1 Neues Projekt einrichten mit Keil µVision 5

1. Schritt: Ein Rumpfprogramm in Keil einrichten (siehe P0toP1.c Muster)

2. Schritt: Display Schema Analyse

Aus dem Schema lesen wir folgende Pins:
PC 8: CS
PC10: RS (SCK3)
PC11: MISO (SDO)
PC12: MOSI (SDI -> Daten zu Display)

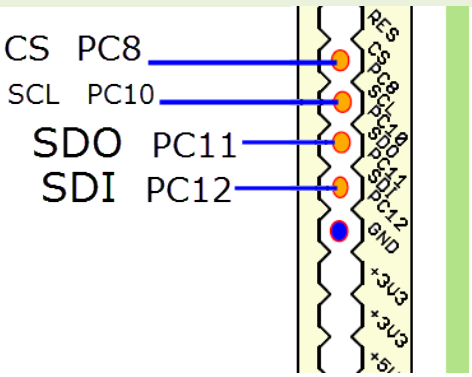


3. Schritt: Display Board Anschlüsse

Logik-Analyser wird wie folgt angeschlossen:

Kanal 0: SPI-MOSI
Kanal 1: SPI-MISO (nicht genutzt)
Kanal 2: SPI-Clock (pos Flanke)
Kanal 3: SPI-CS-Enable (LowAk)

Weitere Einstellungen:
- MSB First
- Daten gültig bei ansteigender Flanke
- ACHTUNG: Display ILI9341 wird mit 9 Bit angesteuert. Das heisst wir erwarten 9Bit Daten pro Transfer.



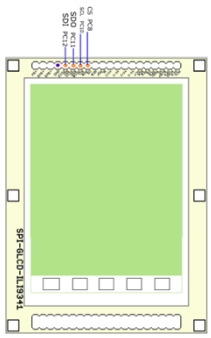
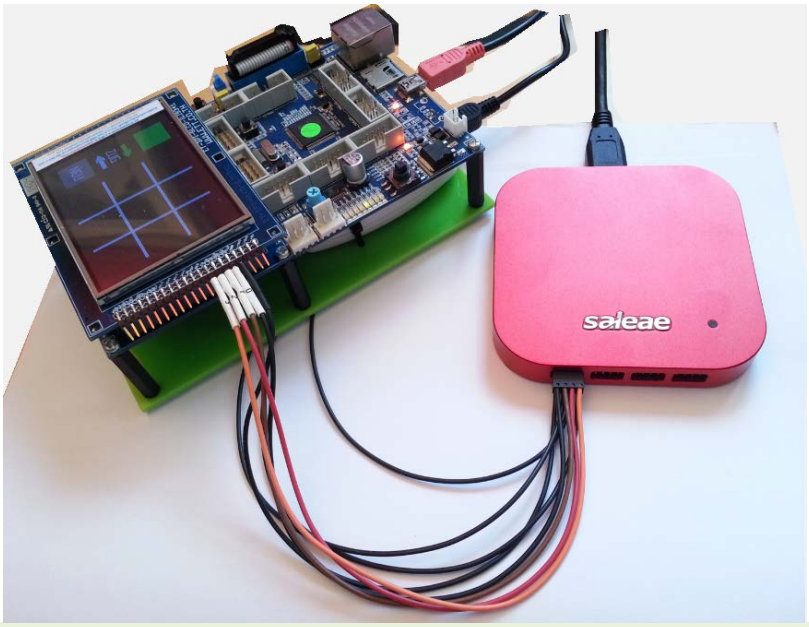


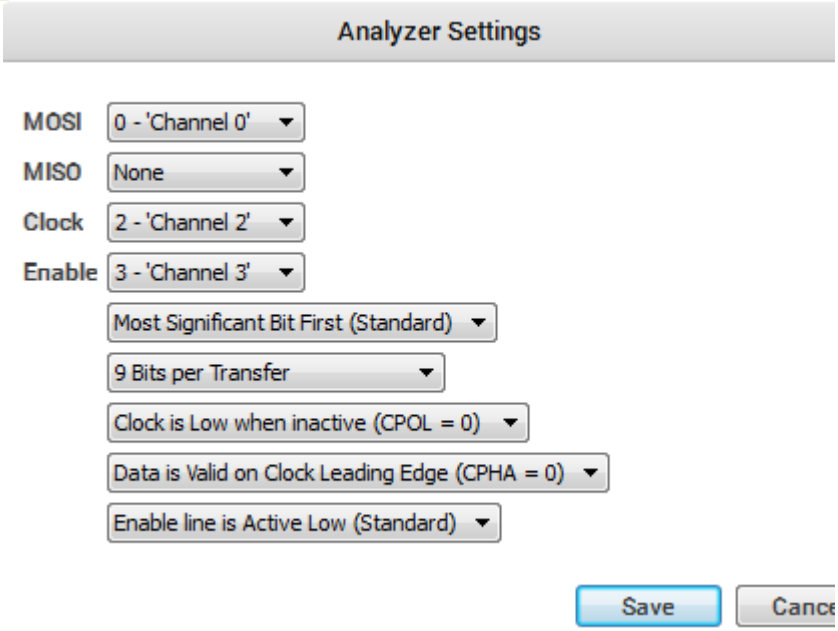
Bild zeigt Display von oben wo an der Stirnseite alle Signale herausgeführt sind.

Bild mit angeschlossenem Logik-Analyser.



4. Schritt: Logik Analyser Setup

Nebenstehend das Setup des Saleae Logik Analysers.



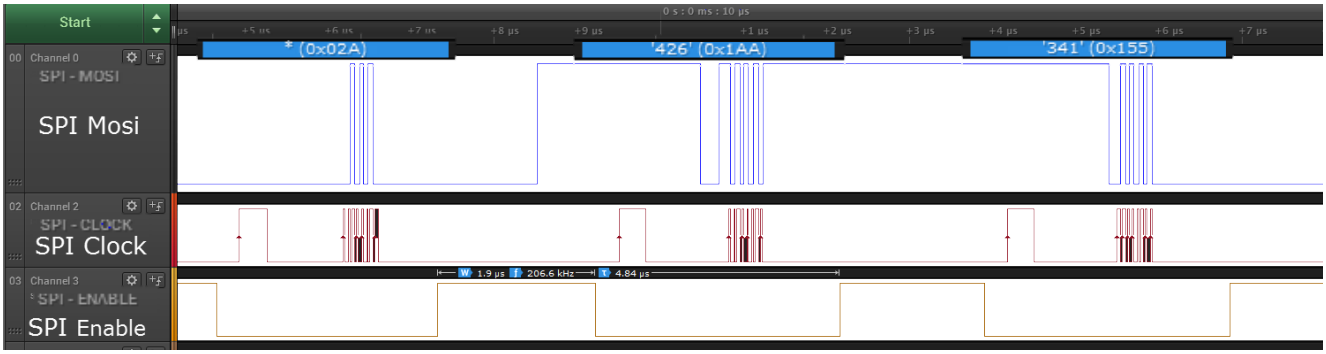
5. Schritt: Testprogramm schreiben

Im Main Loop wird regelmässige ein Kommando und die dazugehörigen Daten gesandt.

Dies erlaubt die Daten auf dem SPI zu verfolgen.
Im nebenstehenden Programm ist die Idee umgesetzt.

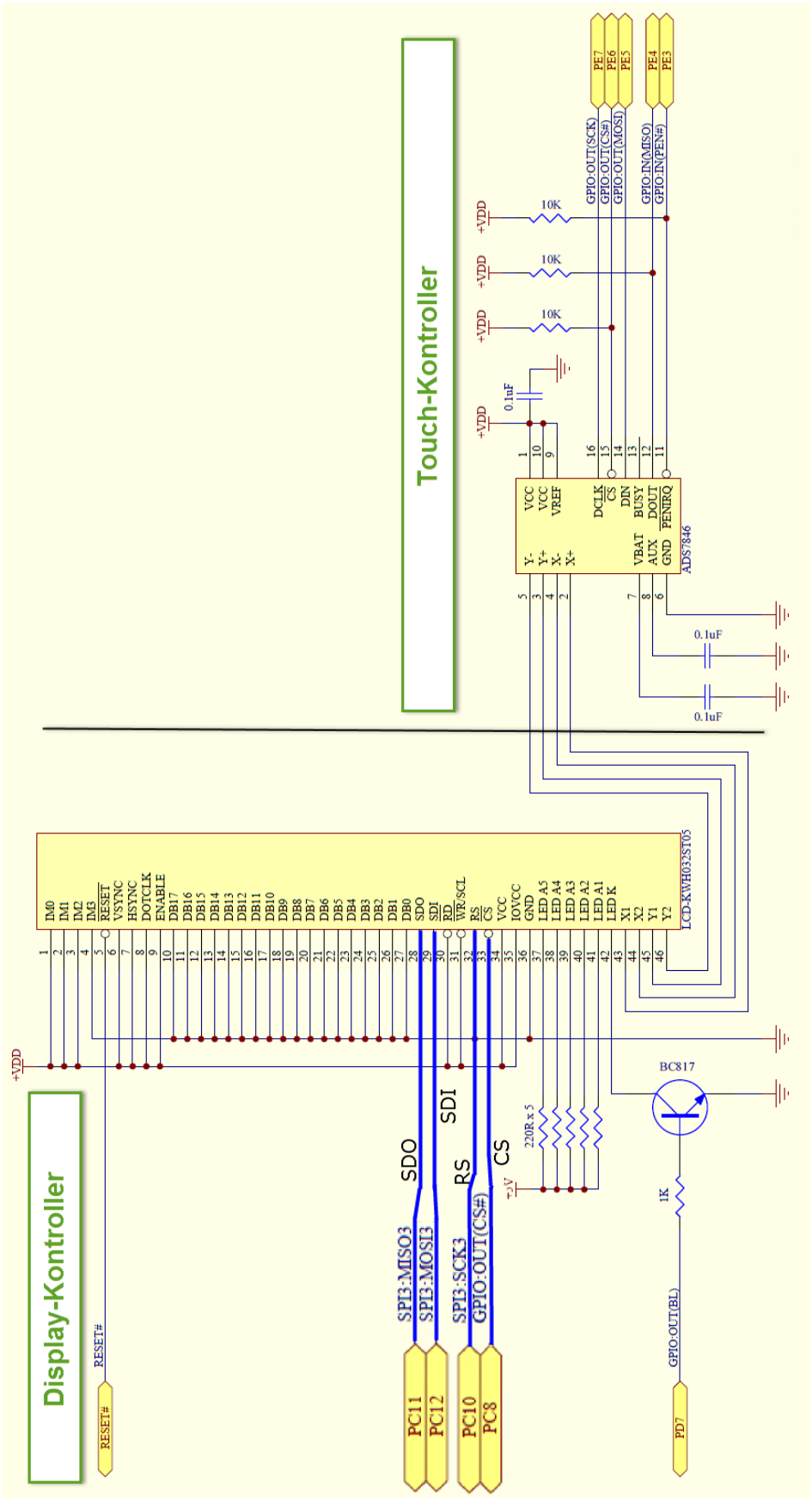
```
while ( 1 )
{
  P1=P0;
  GLCD_Write_Command(0x2A);   // Schreibe Kommando an Display
  GLCD_Write_Data16(0xAA55);  // Sende die Daten zum Kommando
}
```

6. Analyse starten und Daten aufzeichnen



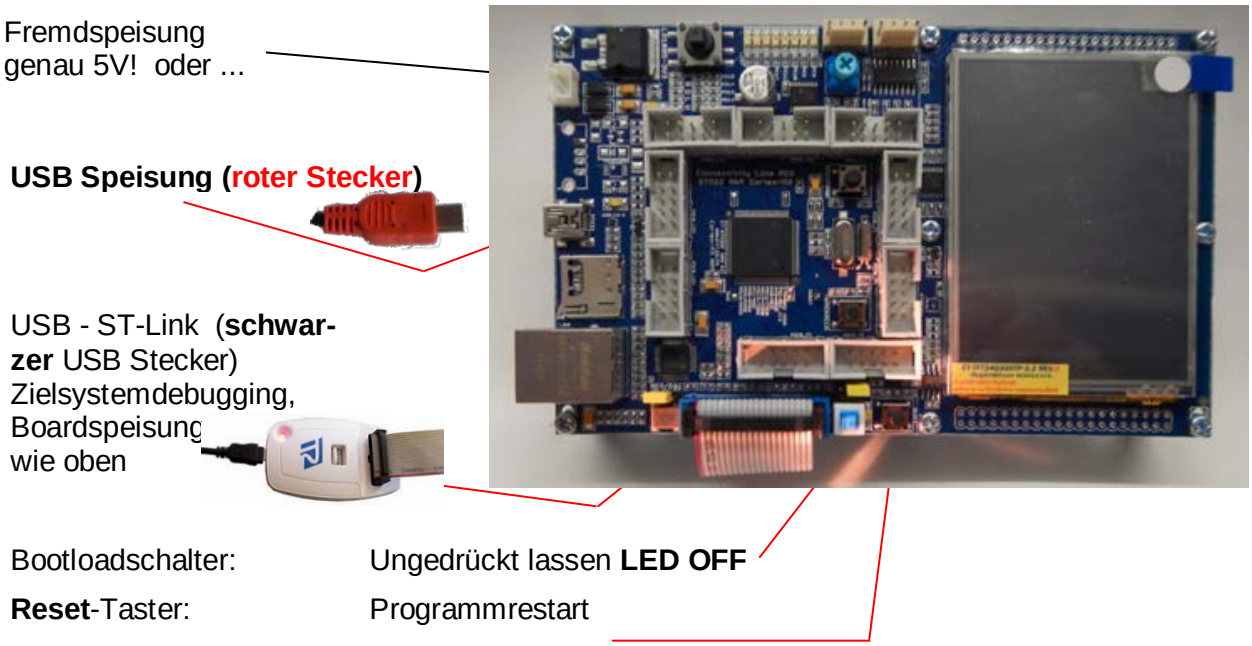
Man sieht schön wie das Programm umgesetzt wird. Das Kommando zuerst und dann die Daten.

3 Anhang Schema

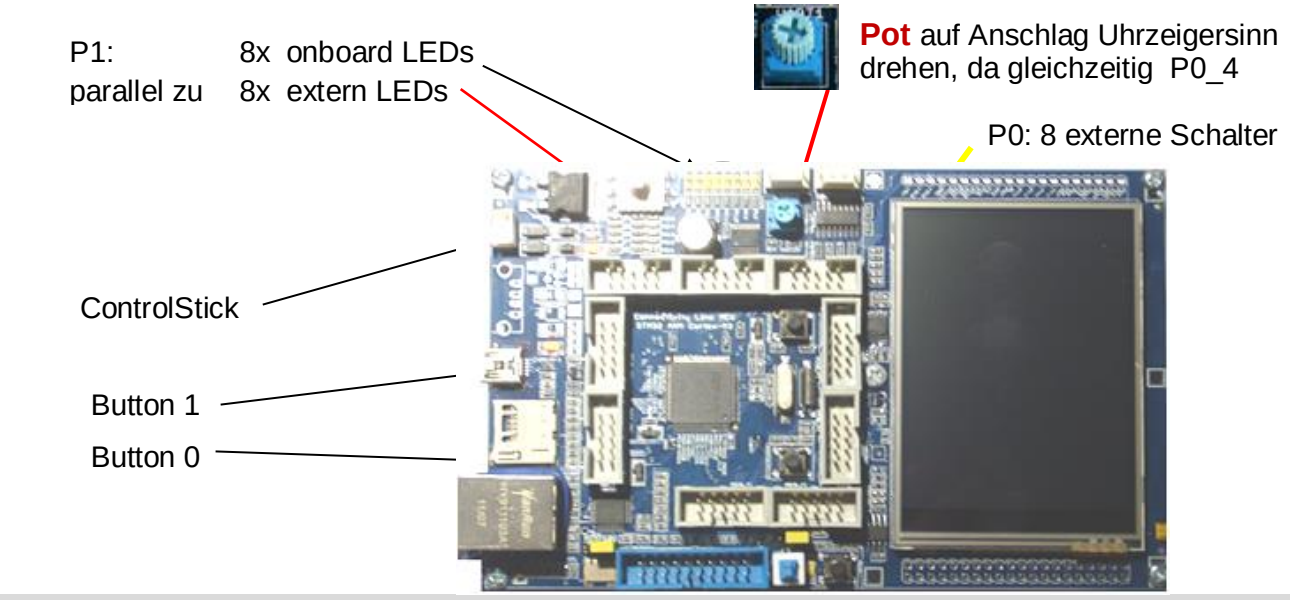


4 Anhang Anschlüsse am µC-Board MCB32

Entwicklungsanschlüsse

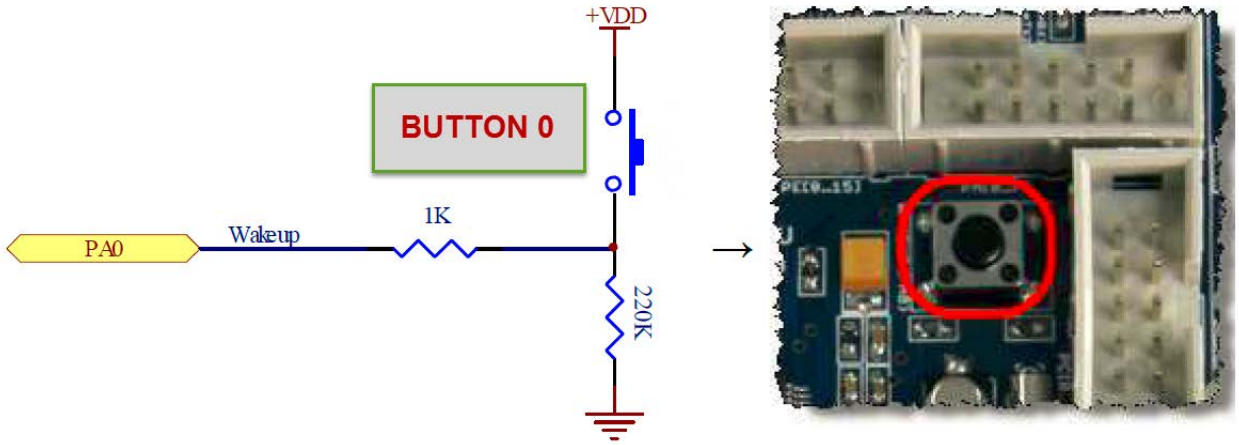


Digitale Ein- und Ausgaben am µC-Board MCB32. **ACHTUNG** mit Potentiometer (Pot)

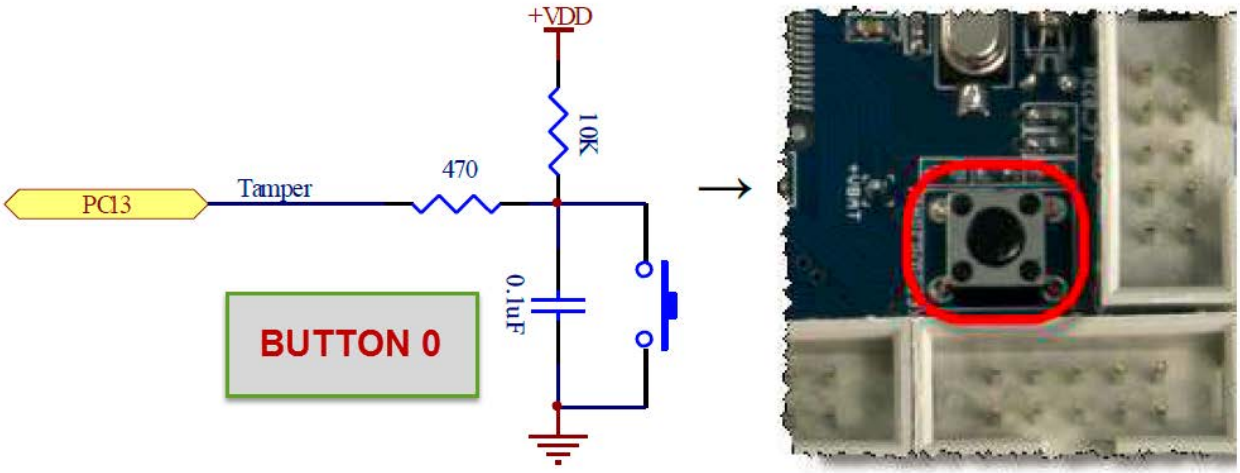



```
// In TouchP0P1.h definierte Pin-Bezeichnungen PA_0 .. PD_11, ohne Bezeichner wie Button .. !

char   Button0        = PA_0;          // Bitwert 1/0, aktiv low, prellt wenig
char   Button1        = PC_13;

char   Stick          = PD_High;       // als Byte 0xF8 open, aktiv low, alle entprellt
char   StickSelect    = PD_15          // Bitwert   1/0; Bytewert   0x80
char   StickDown      = PD_14;         //           1/0;            0x40
char   StickLeft      = PD_13;         //           1/0;            0x20
char   StickUp        = PD_12;         //           1/0;            0x10
char   StickRight     = PD_11;         //           1/0;            0x08
```

Button 0 (PA 0)

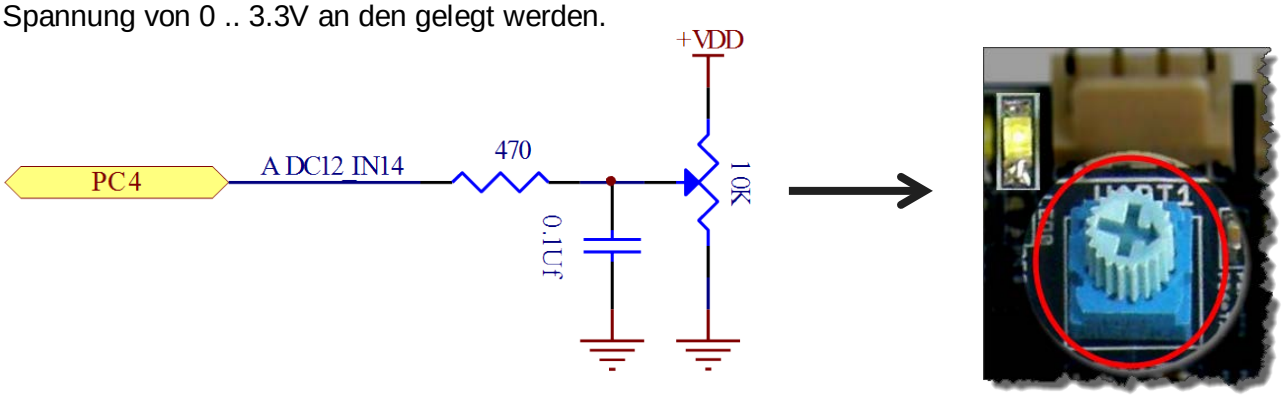


Button 1 (PC 13)



Potentiometer (PC 4) // resp. P0_4 (Library)

Wenn der Port PC4 als Analog-Input (AD-Wandler) geschaltet ist kann mit dem Potentiometer eine Spannung von 0 .. 3.3V an den gelegt werden.



5 Anhang: Referenzen

[1] Robert Weber / Berufsschulzentrum Uster / Projektvorlagen (div)
Danke Robi für die tollen Vorlagen.

[2] E. Schellenberg, E. Frei / TBZ. Programmieren im Fach HST

[3] Unterlagen MCB32 / Cityline / E. Malacarne

[4] Link zu Wikipedia. http://de.wikipedia.org/wiki/ARM_Cortex-M3 (1.9.14)

[5] D. Schoch ; TBZ ; « C-Grundlagen.doc 29.5.2001 » ; Neu erstellt

[6] E. Frei ; TBZ ; Erweiterungen in [2] « C-Grundlagen.doc 29.5.2001 «

[7] Dirk Louis ; C++ « Programmieren mit Beispielen », MT-Verlag

[8] Link zu Wikipedia: http://de.wikipedia.org/wiki/Einerkomplement

6 Anhang Literaturverzeichnis und Links

[1] R. Jesse, Arm Cortex M3 Mikrocontroller. Einstieg und Praxis, 1 Hrsg., www.mitp.de, Hrsg., Heidelberg: Hütigh Jehle Rehm GmbH, 2014.

[2] J. Yiu, The definitive Guide to ARM Cortex-M3 and M4 Processors, 3 Hrsg., Bd. 1, Elsevier, Hrsg., Oxford: Elsevier, 2014.

7 Anhang Wichtige Dokumente

Die folgende Liste zeigt auf die wichtigsten Dokumente welche im WEB zu finden sind. Beim Suchen lassen sich noch viele nützliche Links finden.

- Datenblatt (STM32F107VC) Beschreibung des konkreten Chips für Pinbelegung etc.
- Reference Manual (STM32F107VC) (>1000Seiten in Englisch)
  Ausführliche Beschreibung der Module einer Familie. Unter Umständen sind nicht alle Module im eingesetzten Chip vorhanden – siehe Datenblatt.
- Programming Manual (Cortex-M3)
  Enthält beispielsweise Informationen zum Interrupt Controller (NVIC).
- Standard Peripheral Library (STM32F10x)
  Im Gegensatz zu anderen MCUs sollen die Register der STM32 nicht direkt angesprochen werden. Dafür dienen die Funktionen der Standard Peripheral Library.
  Sie ist auf http://www.st.com/ zusammen mit einer Dokumentation (Datei: stm32f10x_stdperiph_lib_um.chm) herunterladbar.