

EINLEITUNG

Der AnalyzerDude Version 1.0 ist ein Gerät um elektronische Verbindungen zu prüfen und zu analysieren. Dieses Manual beschreibt die Funktionsweisen und die Bedienung.
Im inneren des AnalyzerDude befindet sich ein ATmega328p auf dem eine Firmware läuft. Im Folgenden wird die Benutzung und die Entwicklung beschrieben.

HARDWARE

Die Hardware stellt eine Referenzimplementierung dar. Sie ist mit zwei analogen Eingängen, 2 direkten digitalen Eingängen und 4 galvanisch getrennten Eingängen ausgestattet.
Auf der Frontseite befinden sich 4 Druckknöpfe für die Eingabe und Steuerung der Firmware. Ein Reset Button, zwei steuerbaren LED's, sowie einer Betriebsanzeige. Das Display stellt ein Menu und aktuelle Werte dar.

Die Software muss der jeweiligen Hardwareimplementierung angepasst werden. Die Referenzimplementierung ist für den ATmega328p entwickelt worden.

SOFTWARE

Die Firmware zum AnalyzerDude (vers. 1.0) ist freeware. Sie kann ohne Einschränkungen genutzt werden. Für die jeweils verwendeten Hardwarekomponenten wird im Folgenden beschrieben welche Anpassungen vorgenommen werden müssen.

Die Software ist in C geschrieben und kann mit avr-gcc compiliert werden. Unit-tests sind mit dem CU Framework verfasst worden. (<http://cu.danfis.cz>)

LIBRARIES

Folgende third-party libraries wurden verwendet:

NAME	AUTOR	VERWENDUNG
lcd	Peter Fleury	C include file for the HD44780U LCD library (lcd.c)
uart	Peter Fleury	Interrupt UART library with receive/transmit circular buffers

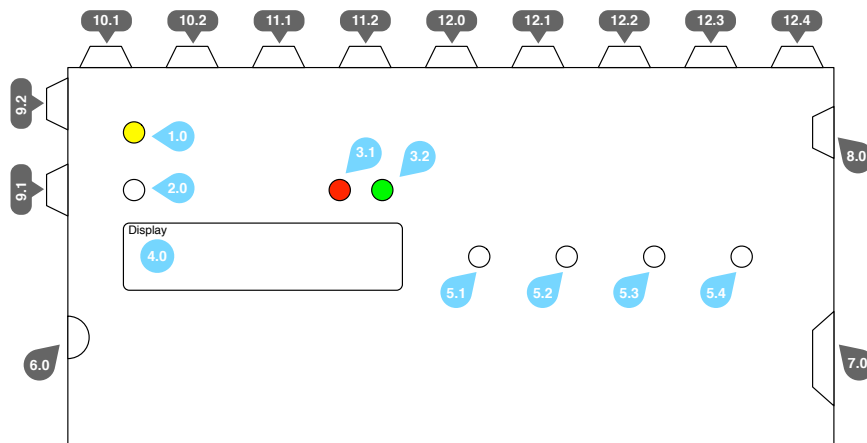
IN BETRIEBNAHME

BETRIEBSSPANNUNG

Der AnalyzerDude läuft mit einer Betriebsspannung von 5V. Über den externen Stromanschluss kann das Device mit einer Spannung von 5V - 9V betrieben werden.

Neben dem Hauptanschluss für die Betriebsspannung kann der AnalyzerDude ebenso über die ISP Schnittstelle betrieben werden. Die über USB kommende 5V Versorgungsspannung ist das Device voll Funktionsfähig.

Der UART Schnittstelle stehen ebenfalls Versorgungspins zur Verfügung.



1.0 - Power Indicator. Steht das Gerät unter Spannung leuchtet der Power-Indicator leicht orange. Die LED muss sichtbar aber leicht leuchten. Strahlt das Led unangenehm hell kann zu viel Spannung auf dem Gerät sein.

2.0 - Reset Button - Mit dem Reset Button kann das Gerät neu gestartet werden. Nach betätigung des Reset Buttons startet das Gerät erneut. Es ist der Welcome-Screen zu sehen, bevor der erste Menüpunkt angezeigt wird.

3.1 - Bestätigungs-Indicator. Programmierbares, rotes LED. Dieser Indikator wird verwendet um eine Bestätigung anzuzeigen. Leuchtet das rote LED, so befindet sich das gerät entweder in einem Zustand der Bearbeitung, oder hat bestätigt eine Benutzereingabe durch kurzes aufleuchten.

3.2 -Run-Indicator. Programmierbares, grünes. Dieser Indikator LED wird verwendet um die Geschäftigkeit des Gerätes anzuzeigen. Das LED wird im normal Betrieb einmal pro Loop umgeschaltet. Es sollte ein deutlich sichtbares grünes Licht sichtbar sein. Fängt das Licht an zu flackern, so liegt eine Operatin vor, die erheblich lange Zeit benötigt. Siehe Trubleshhoting

4.0 - Display. Das Dispaly stellt das Menu und den Aktuellen Betriebsmodus dar.

5.1 - Bleuer funktions Button - Funktionsbutton I. Dieser Button kann programmiert werden. Im Normalbetrieb lässtr sich mittels dieses Biuttons durch das Menu blättern.

5.2 - Grüner funktion Button - Funktionsbutton II. DieserButton kann pogrammiert werden. Im Normalbetrieb lässt sich mittels dieses Buttons ein Menüpunkt auswählen.

5.3 - gelberr funktion Button - Funktionsbutton III. DieserButton kann pogrammiert werden. Im Normalbetrieb hat dieser Button keine Funktion

5.4 - Weißer funktion Button - Funktionsbutton IV. DieserButton kann pogrammiert werden. Im Normalbetrieb bricht dieser Button ein ausführendes Programm ab und führt zurück zum Menu.

6.0 - Power. Spannungsversorgung 5V - 9V.

7.0 - In System Programmer. Über diese ISP-Schnittstelle lässt sich das Gerät neu Programmieren. Firmwareupdates können eingespielt werden.

8.0 - External Communication Port. Über den ECP lässt sich Spannung (5V) in das Gerät ein und ausführen. Die Reset Funktionalität ist über einen Pin herausgeführt. Desweiteren befindet sich eine 2-Wire UART Schnittstelle auf dem Port. (Siehe Kommunikation)

9.1 - V-, (GND) 0V Spannungs Heranführung für die Versorgung der Ports 10.x und 11.x

9.2 - V++ 5V Spannungs Heranführung für die Versorgung der Ports 10.x und 11.x

10.1. Analog Input I - Dieser Port stellt eine Spannung von 0 V- 5V in 1024 Werten dar. Dieser ist direkt mit dem ADC verbunden. Eine Überspannung größer als 5V zerstört das Gerät.

10.2. Analog Input II - Dieser Port stellt eine Spannung von 0 V- 5V in 1024 Werten dar. Dieser ist direkt mit dem ADC verbunden. Eine Überspannung größer als 5V zerstört das Gerät.

11.1 Digital Input I - Dieser Digitale Eingang ist direkt mit dem Mikrokontroller verbunden. Anliegende Spannungen zwischen 0V und 5V können als 0 und 1 ausgegeben werden. Eine Überspannung größer als 5V zerstört das Gerät. Dieser Port ist schneller als Port 12 und sollte für zeitkritische Anwendungen verwendet werden.

11.2 Digital Input II - Dieser Digitale Eingang ist direkt mit dem Mikrokontroller verbunden. Anliegende Spannungen zwischen 0V und 5V können als 0 und 1 ausgegeben werden. Eine Überspannung größer als 5V zerstört das Gerät. Dieser Port ist schneller als Port 12 und sollte für zeitkritische Anwendungen verwendet werden.

12.0 Ground Port für den Port 12. (Pin 12.1 - 12.4). Der digitale Port 12 ist ein mit Überspannungsschutz ausgestatteter, galvanisch getrennter digitaler Eingang in das Gerät. Wird eine externe Spannungsquelle durch diesen Port überwacht, so ist Pin 12.0 mit der Masse des zu analysierenden Gerätes zu verbinden.

12.1 Galvanisch getrennter digitaler Input Pin I mit Überspannungsschutz (getestet bis 16V).

12.2 Galvanisch getrennter digitaler Input Pin II. mit Überspannungsschutz (getestet bis 16V).

12.3 Galvanisch getrennter digitaler Input Pin III. mit Überspannungsschutz (getestet bis 16V).

12.4 Galvanisch getrennter digitaler Input Pin VI. mit Überspannungsschutz (getestet bis 16V).

MENÜFÜHRUNG

Startet der AnalyzerDude so wird nach kurzem WelcomeScreen der erste Menüpunkt angezeigt. Das Menü lässt sich einfach erweitern um neue Funktionen hinzuzufügen (siehe menu.h). Die ausgelieferte Version der Software bietet folgendes Grundsetup an Menüpunkten und Untermenüs.

EBENE/ MENÜPUNKT	DISPLAY	BESCHREIBUNG
1	Say Hello	Test Applikation und Beispielanwendung als Blaupause für eigene Funktionsroutinen.
2	Count ext ports	Zählt die steigenden Flanken an den 4 externen galvanisch getrennten Pins 12.1 - 12.4
3	Show DAC values	Stellt den Wert der beiden analogen Eingänge 10.1, 10.2 dar in einer Auflösung von 0-1023 dar
4	Options:	Optionsmenü.
4.1	Send data via UART	Lässt den Benutzer wählen ob die gemessenen Werte auch über die UART-Schnittstelle (8.0) gesendet werden sollen.

Um das Menü zu steuern sind die Buttons wie folgt belegt:

BLAU: zum nächsten Menüpunkt gehen.

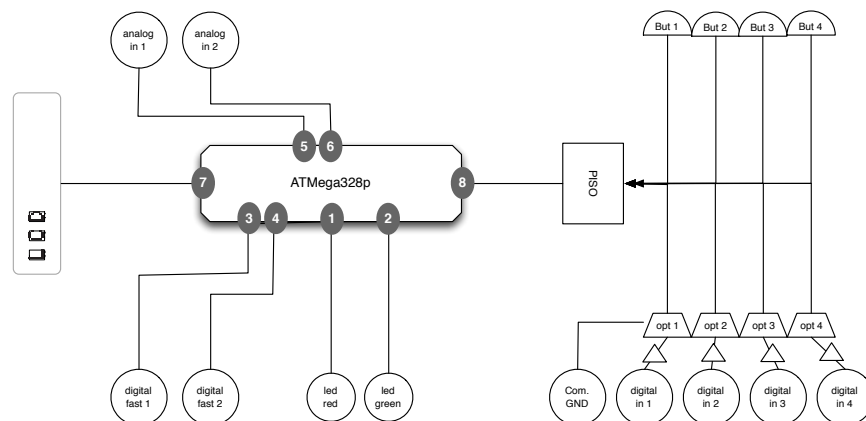
GRÜN: Menüpunkt aufrufen. Applikation starten oder ins Untermenü gehen.

GELB: [nicht belegt]

Weiß: laufende Applikation oder Untermenü Verlassen.

SCHEMATA

Die Schematik des AnalyzerDude zeigt die Grundkonfiguration auf und wie die Komponenten zusammenspielen. Wird die Betriebssoftware für angepasste Konfigurationen verwendet, so müssen entsprechende Konfigurationen angepasst werden.



1: Port B, Pin 0. Setzen des Data Direction Registers in der Datei deviceconfig.h in der Sektion **control led configuration** möglich. Das LED ist als LED_RED definiert. (siehe lights.h)

2: Port B, Pin 1. Setzen des Data Direction Registers in der Datei deviceconfig.h in der Sektion **control led configuration** möglich. Das LED ist als LED_GREEN definiert. (siehe lights.h)

3: Port D, Pin 7. Der digitale Eingang ist direkt mit der MCU verbunden. Die Pinkonfiguration kann in der Datei deviceconfig.h in der Sektion **define fast digital inputs** möglich.

4: Port B, Pin 2. Der digitale Eingang ist direkt mit der MCU verbunden.

5: Port C, Pin 4. Der analoge Eingang ist direkt mit der MCU verbunden.

6: Port C, Pin 5. Der analoge Eingang ist direkt mit der MCU verbunden.

7: Das SNT-Display ist über 4 Datenleitungen auf Port C, Pin 0-3 verbunden. Reset liegt auf Port D, Pin 4. Wenn verbunden, dann liegt der Read/Write Anschluss auf Port D, Pin 5. Enable, sofern nicht fest auf Vcc verbunden ist auf Port D, Pin 5 zu finden. Die Konfiguration kann in der Datei lcd.h in der Sektion **Definitions for 4-bit IO mode** geändert werden.

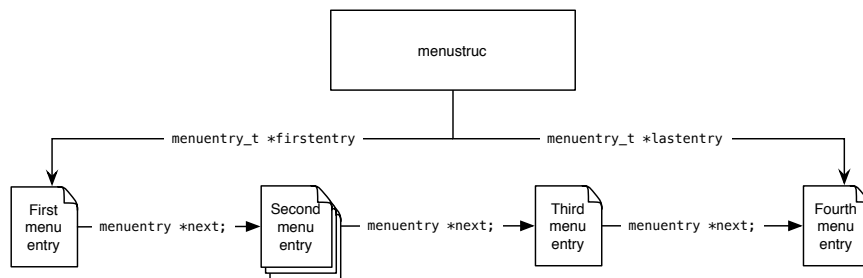
8:

MENU.H

Das Menü wird in der main.c erzeugt. Die Datei menu.h um Unterordner /ui stellt methoden zur Verfügung um Menüpunkte zu erstellen und durch das Menü zu blättern.

menustruc

Das menustruc hält das Menu. im Struc befinden sich Zeiger auf den ersten und den letzten Menüeintrag. Über das menustruc lässt sich immer ein Menüeintrag finden.

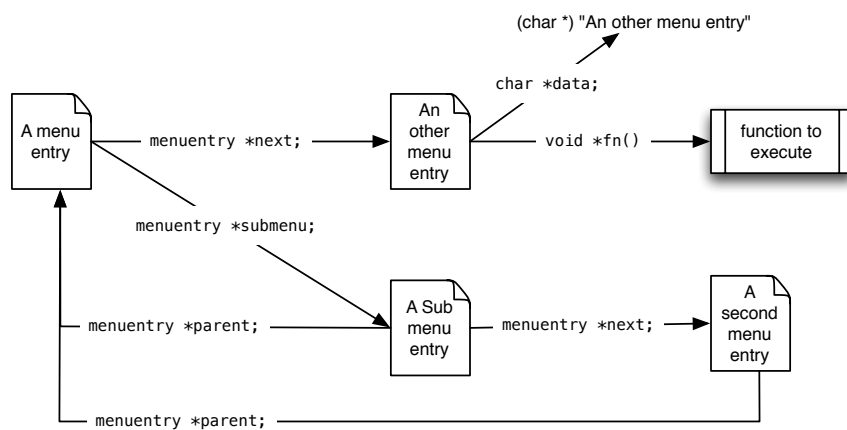


Die einzelnen Menüpunkte sind über den Zeiger "next" verbunden. Es ist möglich vom ersten Menüpunkt iterativ durch alle Punkte zu blättern, nicht jedoch von Punkt 1 direkt zu Punkt 3 zu springen.

menuentry

Ein Einzelner Menüpunkt ist im Struct `menuentry` beschreiben.

```
typedef struct menuentry {  
    struct menuentry *next;    // pointer to the next entry  
    struct menuentry *parent;  // pointer to the parent entry  
    struct menuentry *submenu; // submenu of this item  
    char *data;                // the raw data string to show  
    void (*fn)();              // function to execute at startup  
} menuentry_t;
```



Ein Menüpunkt verweist immer auf den nächsten Menüpunkt über den Zeiger "next". Gibt es zu einem Menüpunkt ein Untermenü, so ist diesen über den Zeiger "submenu" zu erreichen. Die funktion `fn()` sollte in diesem Fall auf `&dummy()` zeigen.

Jeder Untermenüpunkt kennt über den Zeiger "parent" den aufrufenden Hauptpunkt um auf diesen bei Abbruch des Submenüs zurückzukehren.

Bevor das Menü benutzt wird, sollte es initialisiert werden:

//Initiaialisieren des Menüs

```
void ui_menu_init(void);
```

Um das Menü zu benutzen stehen Methoden bereit, welche Einfluss auf den sich aktuell im Speicher /Display befindenen Menüeintrag haben:

Scahltet auf den nächsten Menüeintrag:

```
void ui_menu_next(void);
```

Am Ende eines Menüs wird beim aufruf von ui_menu_next wieder mit dem ersten begonnen. Das Menü kreist fortwährend.

Der Aufruf ui_menu_has_submenu ist erfolgreich, wenn der aktuelle Eintrag ein Untermenü besitzt.

```
bool ui_menu_has_submenu(void);
```

Ist ein Submenu vorhanden, so lässt sich mit ui_menu_enter_submenu in die tiefergelegene Ebene springen.

```
void ui_menu_enter_submenu(void);
```

Das Untermenü lässt sich mit ui_menu_leave_submenu wieder verlassen. Der aktuelle Menüpunkt ist nach dem Aufruf der Punkt von dem eingesprungen wurde.

```
void ui_menu_leave_submenu(void);
```

Jede Page oder Sprung operation setzt den aktuellen Menüpunkt, aktualisiert jedoch nicht das Display. Um den aktuellen Menüpunkt darzustellen wird ui_menu_show aufgerufen.

```
void ui_menu_show(void);
```

Jedem Menüpunkt ist beim Anlegen eine Funktion mitgegeben. Um die Funktion hinter dem Menüpunkt zu starten wird ui_menu_run aufgerufen. Um die aktuell laufende Funktion zu beenden muss eine Implementation erfolgen. Der AnalyzerDude 1.0 regelt die Ausführung in der Main-Routine

```
void ui_menu_run(void);
```

Ein Menüpunkt muss nach dem Initialisieren des Menüs mit einem darzustellene Namen und einer Funktion die bei Selektion aufzurufen ist anzulegen.

```
menuentry_t *ui_menu_add(char*, void*);
```

das Anlegen eines Untermenüs unterscheidet sich darin, das der hauptknoten als Parameter mitgegeben werden muss.

```
menuentry_t *ui_menu_add_sub(menuentry_t *parent,  
                             char*, void*);
```