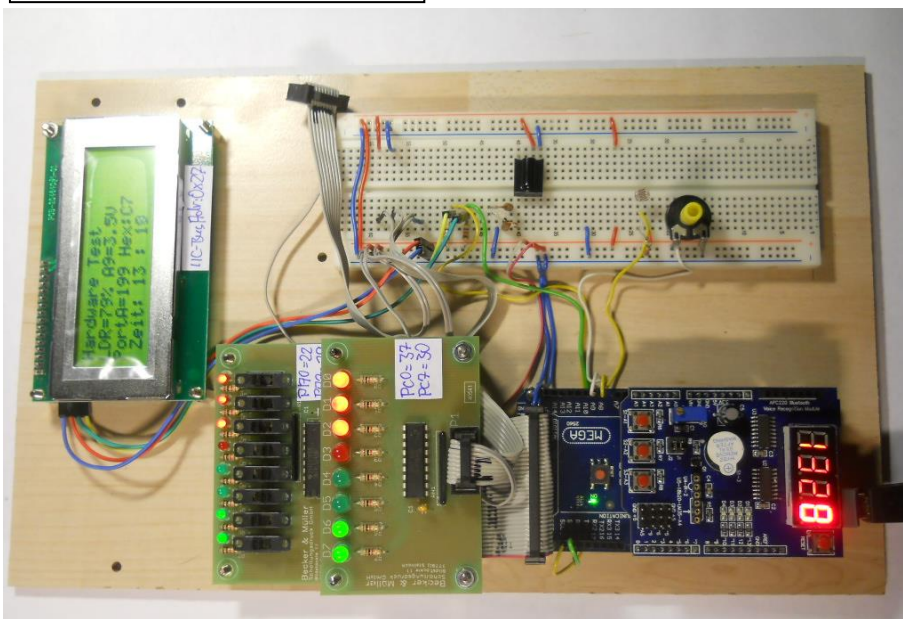


Messbox mit ArduinoMEGA und „Profilab*“

Benutztes Experimentierboard.
Grundlage ein Arduino-MEGA-
Board.
Bezugsquelle im Text.



*Profilab ist eine kostenpflichtige
Visualisierungssoftware.

Profilab Expert 4.0 - [n:\arduino_messen_jfbma16\messbox_anal_digi_ein_aus\messboxanal_digital_ein_aus.prj]

Datei Bearbeiten Frontplatte Konfiguration Optionen Fenster Registrierung ?

Empfangen/Senden von Daten ueber die serielle Schnittstelle mit Stringketten vom/zum uC Arduino

oben ser. Schnittstelle Nr. einstellen mit Maus nach oben fahren Baud: 19200

Empfangen 10bit und 8bit vom uC

Analog 0

Volt 0000,0

0 5

MSB 2.7 LSB 2.0

Senden Analogwert 8bit

000 analWert

Senden Digitalwert 8 bit

000 digitWert

Länge Übertragung (String) für die Messwerte: 00 Länge Stri

"Messbox" mit ARDUINO MEGA

Hier wird eine Lösung einer „USB-Messbox“ gezeigt

Aufgaben: Diese Oberfläche dem Bedarf entsprechend verändern

Hardware ein ARDUINO System

Software der Messwerterfassung: Profilab4.0 eine grafische Programmieroberfläche.

Die folgend gezeigte Lösung ist voll funktionsfähig:

Digital ein; Digital aus Analog ein; Analog aus (Sichtweise ARDUINO-MEGA µC)

Software:

The image displays the 'Front panel' software interface and a detailed hardware block diagram. The software interface (top) shows data reception and transmission controls, including a voltage gauge, digital bit displays, and a serial communication section. The hardware diagram (bottom) illustrates the internal components of the 'Messbox', including a microcontroller, A/D and D/A converters, and various digital and analog components.

Software Interface Details:

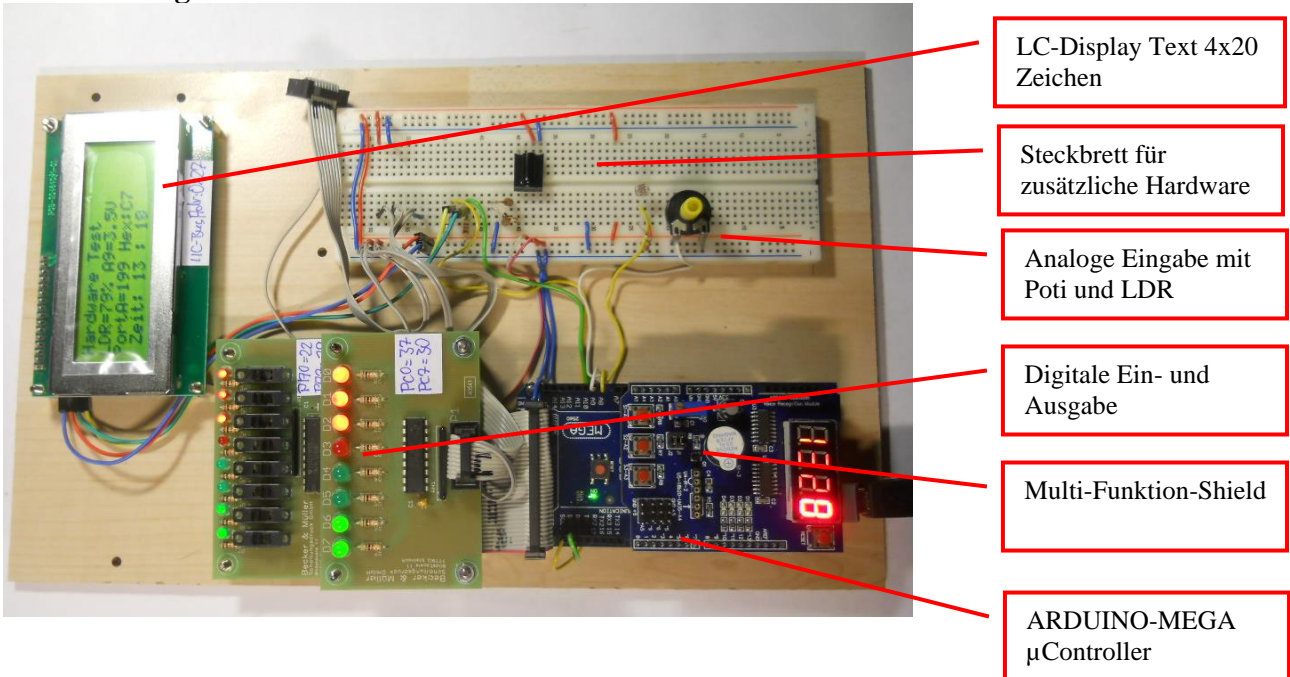
- Empfangen/Senden von Daten ueber die serielle Schnittstelle mit Stringketten vom/zum µC Arduino**
- Empfangen 10bit und 8bit vom µC**: Analog 0 (0000,0 Volt), Digital 8bit (00 Hex, 000).
- Senden Analogwert 8bit**: 109 analWert.
- Senden Digitalwert 8 bit**: 020 digitWert.
- Buttons: "00 Länge Stri", "00 Hex", "000", "109 analWert", "020 digitWert".
- Text: "Messbox" mit ARDUINO MEGA, "Hinweis: Start Programm dauert einige Sec!", "c Neumaier Mai16 Gew.-Techn Schule Offenburg".

Hardware Block Diagram Details:

- COM Request**: Serial communication interface with parameters like 19200 Baud, 1000 Hz, and 13 synchronization characters.
- Logikbauteile -> AD/DA Wandler**: Includes A/D (ADC1) and D/A (DAC1) converters.
- Frontplattenelemente -> Anzeige**: Displays for analog and digital values.
- Frontplattenelemente -> Bedienung digital**: Digital control elements like buttons and switches.
- Logikbauteile -> AD/DA Wandler**: Additional logic components for data conversion.
- Formel E0(51*4)**: Formula for converting 10-bit values to 0-5V.
- Formel E(255/8bit)**: Formula for converting 8-bit digital values to 0-255 bits.

Notwendiges Übungsboard mit ARDUINO MEGA

Beschreibung



LC-Display Text 4x20 Zeichen

Steckbrett für zusätzliche Hardware

Analoge Eingabe mit Poti und LDR

Digitale Ein- und Ausgabe

Multi-Funktion-Shield

ARDUINO-MEGA µController

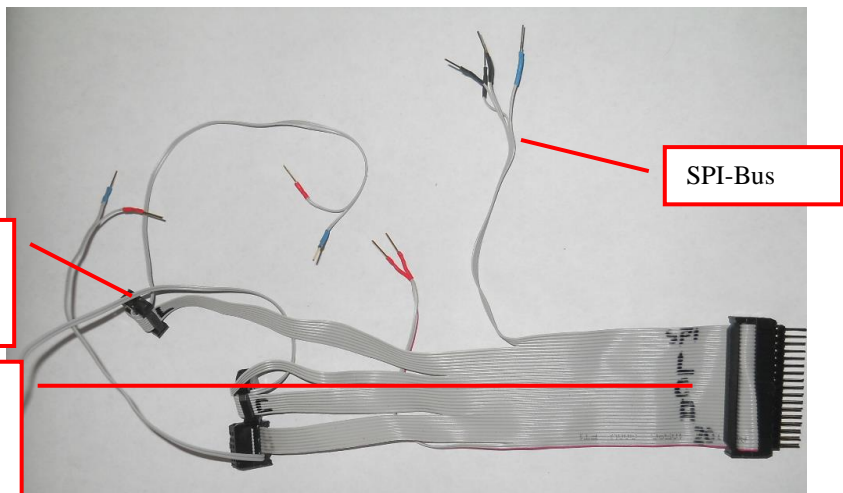
Aufgebaut auf einer Trägerplatte (Laminatboden) Maße: 18 x 29,7 cm -> Höhe DIN A4
Hardware:

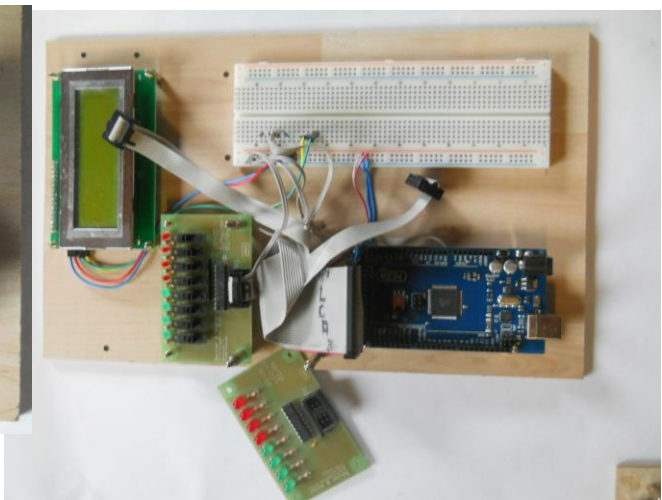
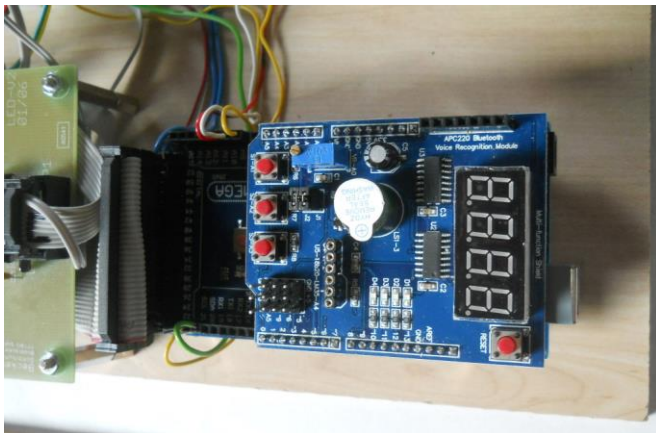
1. ARDUINO-MEGA µC mit USB -Chip CH340. Inzwischen sind die Treiber online gut verfügbar.
2. Multi-Funktion Shield mit Taster, LEDs, 4x7Segment Anzeige und Adapter für Sensoren und Aktoren.
3. LC Display Textanzeige, 4 Zeilen mit jeweils 20 Zeichen Das Display hat einen IIC-Bus Adapter und wird somit über den IIC-Bus angesteuert. Die Adresse lautet 0x27.
4. Digitale Ein- und Ausgabe:
Die Ports A und C werden benutzt:
Port A für die digitale Eingabe über 8xSchalter
Port C für die digitale Ausgabe über 8x LEDs
Weiterhin steht Port L ebenfalls zur Verfügung über einen 10poligen Wannenstecker. Pin 1 0Volt, Pin2 +5V, Pin 2 bis 10 sind die 8 Datenleitungen.

Die Steckerleiste des Arduino MEGA wird mit einem speziell angefertigten Adapterkabel beschaltet:

10 polig Wannenstecker 1:
GND 2: +5V 3..10: P0
...P7

34 polig Wannenstecker mit eingesteckten speziellen Stiftleisten (länger als normale Stiftleisten)

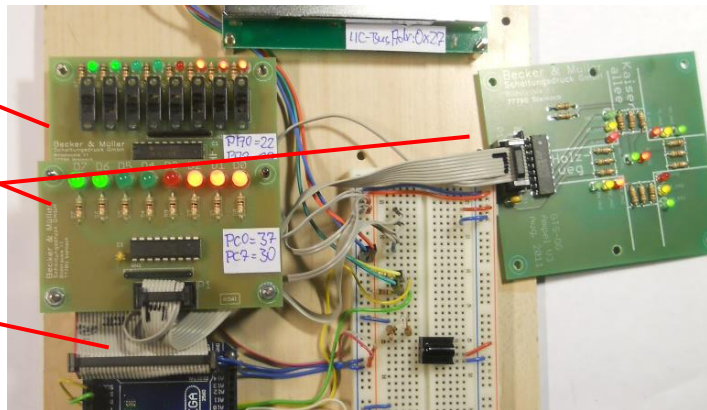




Details Aufbau:

Multi-Funktion-Shield aufgesteckt.

- PORT_A: 8xSchalter-Platine
- PORT_C: 8xLED-Platine
- PORT_L: hier zusätzlich eine Ampel- Übungsplatine
- 34 polig Wannenstecker mit Flachbandkabel: +5V; PORT_A; PORT_C; PORT_L; SPI-Bus



Intentionen für den Aufbau dieser Übungsplatine:

- Fest montierte Hardware (8xSchalter; 8xLEDs) und Zugriffe auf 3PORTS (A, C, L) auf der MEGA-Steckleiste.
- Zugriff auf die Busse: SPI und IIC. Diese Leitungen stehen auf dem Steckbrett zur Verfügung.
- LC-Display: Wichtiger Bestandteil eines jeden uC-Systems. Hier ein großes Display mit 4x20 Zeichen. Verdrahtung erfolgt sehr einfach über den IIC-Bus.
- Steckbrett ist ein „Muss“ für ein Arduino-System. Es kann dadurch ganz einfach verschiedene Hardware angeschlossen werden, wie z.B. Sensoren, Sender und Empfänger im Bereich Funk, WIFI, Blue-Tooth, Potentiometer, H-Brücken, usw.
- Die Startersets sind zwar günstig, aber eine unübersichtliche Anzahl einzelner kleiner Bauteile. Hier herrscht eine Grundordnung. Mann kann, wenn nötig, dieses Übungsboard leicht erweitern. Funktioniert dann die getestete Schaltung, kann man immer noch ein kleines NANO-Board verwenden für seine spezielle Schaltung.
- Warum ein (teures) MEGA-Board und kein UNO? Wenn man ein Shield aufsteckt, stehen beim UNO-Board oft keine freien Pins mehr zur Verfügung.
- Ein Shield soll problemlos aufgesteckt werden können. Hier habe ich mich für ein Multi-Funktion-Shield entschieden, dass einige hilfreiche Hardware bietet, wie Taster, 7Seg-Anzeige, LEDs und Adapter für Sensoren und Bluetooth.

Ich stelle hier meine Ideen und Erfahrung zur Verfügung als ehemaliger Lehrer. Dazu gibt es eine CD mit Beispielen für den Unterricht. Lassen Sie sich da überraschen!

| | | |
|--|---|-------------|
| Gerhard Neumaier Dipl. Gewerbelehrer StD i.R. | Arduino μController | Datum:..... |
| | Messbox mit Profilab | Name: |

Kostenzusammenstellung:

Quelle Hardware in Deutschland: <https://www.roboter-bausatz.de/diy-elektronik/mainboards/arduino/>
Günstige Quelle ohne Zollformalitäten!

1. Arduino-Mega:

| Bauteil | Preis in € | Bemerkung |
|---|--------------|--------------------------------|
| Arduino MEGA2560 R3 | 11,00 | |
| Multi-Funktion-Shield | 5,00 | |
| Steckbrett | 2,50 | |
| LCDisplay4x20mit IIC | 6,00 | |
| Platine 8xSchalter | 8,50 | Handgefertigt |
| Platine 8xLED | 6,00 | Handgefertigt |
| Flachbandkabel mit Stecker, Stiftleiste | 8,00 | Handgefertigt |
| 14x 10mm Abstandbozen | 3,00 | |
| 4x 20mmAbstandbolzen | 1,20 | |
| Muttern 3mm | 0,20 | |
| Steckkabel | 1,00 | |
| Potj, LDR, Widerstände | 1,00 | |
| Trägerbrett Laminat | 5,60 | Handgefertigt, gesägt, gebohrt |
| Zusammenbau | 5,00 | |
| Summe: | 63,00 | |

Infos: g.neumaier@yahoo.de

Infos, Quellen:

<https://www.arduino-tutorial.de/> Einstieg in das Thema

- <https://www.arduinoforum.de/arduino-Thread-Arduino-MEGA-2560-R3> Vergleich UNO -MEGA +Informationen wie

Pinbelegung, Schaltplan

<https://www.arduino.cc/en/Hacking/PinMapping2560> Pinbezeichnung am IC

<https://core-electronics.com.au/tutorials/compare-arduino-boards.html#mega> PIN_OUTs aller ARDUINO Typen

Meine Seiten:

<https://elektrofortbildung.wordpress.com/meine-homepages/> Hier sind die Links auf meine Seiten

<https://egselektronik.wordpress.com/ucontroller/> μ Controller auch ARDUINO

http://www.elektroniker-geraete-systeme.de/u_controller/default.html μ Controller auf der EGS Seite

| | | |
|--|-----------------------------|-------------|
| Gerhard Neumaier Dipl. Gewerbelehrer StD i.R. | Arduino µController | Datum:..... |
| | Messbox mit Profilab | Name: |

```

/*****Quellcode ARDUINO MEGA*****/
| CLASS:      Arduino MEGA2560
| Compiler:   Arduino 1.5.8| PROGRAM aktuell: MessboxAnalDigitalein_aus.ino
| PROGRAM:    Analog_1x_8xDig_nach_RS232__2x8bitvonRS232.ino  ->ino = Arduino code
| AUTHOR:     Gerhard Neumaier
| DATE:       Okt. 2012
| DESCRIPTION: Analogwert A0 und 8xpins vom PORT_A ueber die ser.-Schnittstelle uebertragen
8bit empfangen und als PWM-Signal am Portpin 10 ausgeben
8bit empfangen und als Digitalert am PORT_C ausgeben
| REQUIREMENTS: Arduino MEGA 8 1x Poti an A0, an PORT_A 8xSchalter  Datenuebertragung
ueber USB mit COM-Schnittstelle
| NOTES:      Ein komplettes Programm, keine Aenderungen notwendig
| Aufgaben:   Geschwindigkeit (BAUD) aendern, Ausgabewerte aendern,
!!Analogwert A1 hinzufuegen          !! weiteren 8bit Wert empfangen
| REVISION HISTORY:  Mai 2016 G. Neumaier
*****/
/*----- PROGRAM CONSTANTS-----*/
#define leds PORTC //PortC mit leds definieren
/*----- VARIABLE DECLARATIONS GLOBAL-----*/
int PortpinA7 = 29; //PortpinA7 wird der Pin29 ARDUINO-MEGA Board zugeordnet
int PortpinA6 = 28;
int PortpinA5 = 27;
int PortpinA4 = 26;
int PortpinA3 = 25;
int PortpinA2 = 24;
int PortpinA1 = 23;
int PortpinA0 = 22; //PortpinA0 wird der Pin22 ARDUINOBoard zugeordnet

unsigned char digitalValue;

int sensorValue;
//Variable zum empfangen ueber RS232:
  unsigned int value;
  unsigned char ch, pwmsignal_1;
  int ledPin = 10; // LED Connected to digital pin 10
/*----- FUNCTION PROTOTYPES-----*/
void setup()
{
  // Pin 22 PORTA0, 23 PORTA1, 24 PA2, 25 PA3, 26 PA4, 27 PA5, 28 PA6, 29 PA7
  // Die Pins 29 bis 22 werden als Eingang definiert:
  pinMode(PortpinA7, INPUT);
  pinMode(PortpinA6, INPUT);
  pinMode(PortpinA5, INPUT);
  pinMode(PortpinA4, INPUT);
  pinMode(PortpinA3, INPUT);
  pinMode(PortpinA2, INPUT);
  pinMode(PortpinA1, INPUT);
  pinMode(PortpinA0, INPUT);

  // Die Pins 37 bis 30 (PORT_C)werden als Ausgang definiert
  DDRC = B11111111; // DDRC Datenrichtungsregister PORTC 1= Ausgang 0=Eingang

  Serial.begin(19200); // Einstellung BAUDrate Fest: 8bit Daten, keine Paritaet
}
/*----- FUNCTION PROTOTYPES-----*/
unsigned char schalter8_in (void); //Bekanntgabe der Funktion schalter8_in

void loop()
{
  sensorValue = analogRead(A0); // read the input on analog pin 0
  digitalValue = schalter8_in(); //PortA einlesen an Variable uebergeben
  //schalter8_in() ist eine selbstgeschriebene Funktion die
  //8 bit zu einem Byte zusammenfasst. siehe unterhalb loop
  //digitalValue=129; //Test
  // Ausgabe an serielle Schnittstelle als Zeichenkette
  // Serial.print()gibt jede Stelle des Zahlenwertes als ASCII-Zeichen an:
  //Bsp.: Serial.print(130) sendet in Hex 31(0) 33(3) 30(0)
  Serial.print(sensorValue); //Wert wird als 1,2,3 oder 4stellige Dezimalzahl gesendet
  Serial.write(13); //Sendet ein Byte, hier 13, ist das Steuerzeichen fuer neuen String
  delay(2); // delay in between reads for stability 2msec
}

```

| | | |
|--|---|-------------|
| Gerhard Neumaier Dipl. Gewerbelehrer StD i.R. | Arduino μController | Datum:..... |
| | Messbox mit Profilab | Name: |

```

Serial.print(digitalValue); //Wert wird als 1,2,3 oder 4stellige Dezimalzahl gesendet
Serial.write(13); //Sendet ein Byte, hier 13, ist das Steuerzeichen für neuen String
Serial.write(10); //Sendet ein Byte, hier 10, ist das Steuerzeichen für neue zeile
    //!!!!Das Steuerzeichen 10 beendet die Stringkette, Datensatz abgeschlossen
delay(10); // delay in between reads for stability

//Zeichen empfangen:
if( Serial.available()>0)
{
    ch = Serial.read();
    if( isDigit(ch) )// ASCII-Zeichen zwischen 0 bis 9? ->Vorhandene Funktion
    {
        //value = (value * 10) + (ch - '0'); // Ja, neuer Wert value berechnen
        value = (value * 10) + (ch - 48); // Ja, neuer Wert value berechnen
        //Hinweis: ASCIIZeichen 0 = 48 ASCIIZeichen 1 = 49 usw
    }
    if (ch == 97) // Newline-Zeichen? 97 kleines a ASCII Tabelle!
    {
        pwmsignal_1 = value; // Empfangener Wert an leds uebergeben
        value = 0; //Empfangener Wert wieder loeschen
            //leds=value nur bis max 255 sinnvoll
            //value bis max 65535
        // delay(2); // Verzoeigerung 2msec zur Sicherheit
    }

    if (ch == 98) // Newline-Zeichen? 98 kleines b ASCII Tabelle!
    {
        leds= value; // Empfangener Wert an leds uebergeben PORTC oben definiert
        value = 0; //Empfangener Wert wieder loeschen
            //leds=value nur bis max 255 sinnvoll
            //value bis max 65535
        // delay(2); // Verzoeigerung 2msec zur Sicherheit
    }
} //Ende if (Serial.aviable)
    analogWrite(ledPin, pwmsignal_1);
} //Ende loop

//-----
//Funktion zum einlesen 8bit an den Pins 22 bis 29
//die Funktion gibt diese Werte als ein Byte zurück:
unsigned char schalter8_in (void)
{
    unsigned char Schalter8; //lokale Variable
    boolean x0, x1, x2, x3, x4, x5, x6, x7; //lokale Variable
    x7 = digitalRead(PortpinA7);
    x6 = digitalRead(PortpinA6);
    x5 = digitalRead(PortpinA5);
    x4 = digitalRead(PortpinA4);

    x3 = digitalRead(PortpinA3);
    x2 = digitalRead(PortpinA2);
    x1 = digitalRead(PortpinA1);
    x0 = digitalRead(PortpinA0);
    //Einzelne Pins zu einem Byte zusammenrechnen:
    Schalter8 = (x7*128 + x6*64 + x5 *32 + x4*16 + x3*8 + x2*4 + x1*2 + x0*1);
    return Schalter8; //Rückgabewert
}

```

Aufgaben:

1. Die Messbox mit drei Analogen Eingängen erweitern. Dazu muss die Software vom ARDUINO-System
 - Analysiert und verstanden werden
 - Danach gezielt erweitert werden.
2. Ein digitaler Sensor an den ARDUINO anbinden z.B. Temperatur Feuchte DHT11 bzw DHT22 und die Messwerte an Profilab senden.
3. Schnelle Impulse messen (Frequenzzähler)