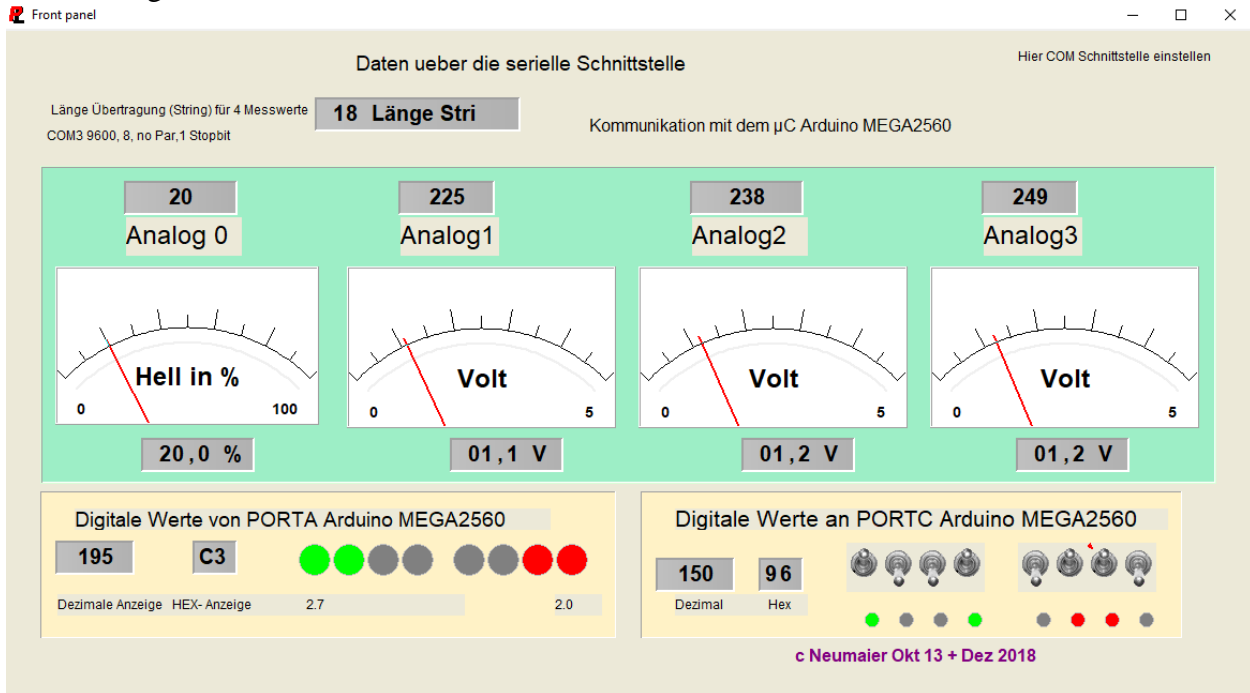


G. Neumaier Dipl.-Gewerbelehrer	Information	Gruppe: Datum:
	ARDUINO µC	

Name:.....

1.2. Programmoberfläche:



2. Quellcode ARDUINO

```

/*****
| CLASS:      Arduino MEGA2560
| Compiler:   Arduino
| PROGRAM:    Anal4x Digi8x_e_aCOM_LCDIIC.ino  ->ino = Arduino Source code
| AUTHOR:     Gerhard Neumaier
| DATE:       Okt. 2013 Dez 2018
| DESCRIPTION: 4Analogports einlesen mit 10bit. A0 LDR 500-10k -1k fest. 3x
Potis
|              8digitale pins einlesen an PortA 8digitale Pins ausgeben an PORTC
|              Analogwerte und Digitalwerte über USB uebertragen und über
virtuelle com-Schnittstelle in PROFILAB visualisieren.

| REQUIREMENTS: Arduino MEGA 8 8xSchalter am PORTA, 8xLEDs am PORTC 3Potis an
A1 bis A3 LDRan A0 Datenübertragung über USB mit COM-Schnittstelle
|              Von der seriellen Schnittstelle kommt ein Byte(8Schalter) an
Port C (8xLED)LC-Display am IIC-Bus mit der Adr: 0x27 IIC-Adapter mit
aufgelötetem Poti
| NOTES:      Ein komplettes Programm, keine Aenderungen notwendig
| Aufgaben:   Geschwindigkeit (BAUD) aendern, Ausgabewerte aendern, Port
aendern
| REVISION HISTORY:
*****/
//----- HEADER h-Dateien einbinden
/*----- ( Import needed libraries )-----*/
#include <Wire.h>

#include <LiquidCrystal_I2C.h> // F Malpartida's NewLiquidCrystal library

/*----Konstante Hardware----*/
#define I2C_ADDR    0x27 // Define I2C Address forPCF8574 the !!PCF8574A 0x20
oder 0x3f !!

```

G. Neumaier Dipl.-Gewerbelehrer	Information	Gruppe: Datum:
	ARDUINO μ C	

Name:.....

```
//Poti aufgelotet ADR mit Loetkolben einstellbar ADR: 0X27: Loetpads offen
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
//D0 bis D3 nicht belegt
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
#define BACKLIGHT_PIN 3

#define LED_OFF 0
#define LED_ON 1

/*-----( Declare objects )-----*/
LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

#define Schalter PINA //PortA mit Schalter definieren
#define leds PORTC //PortC mit leds definieren nicht mit PINC!!

//----- VARIABLE DECLARATIONS GLOBAL
int value;
int sensor0_hell;
float sensor1_volt;
float sensor2_volt;
float sensor3_volt;
//----- FUNCTION PROTOTYPES:
// Keine vorhanden

// Die setup -Routine wird nach dem Reset automatisch 1x aufgerufen
// Die setup-Routine ersetzt hier die Funktion: void init(void)
void setup()
{
    // Die Pins 22 bis 29 werden als Eingang definiert:
    DDRA = B00000000; // DDRA Datenrichtungsregister PORTA 1= Ausgang 0=Eingang
    // Die Pins 37 bis 30 werden als Ausgang definiert
    DDRC = B11111111; // DDRC Datenrichtungsregister PORTC 1= Ausgang 0=Eingang
    // initialize serial communication at 38400 bits per second: BAUD-Rate
    Serial.begin(9600);
    lcd.begin(20, 4); //20 Zeichen pro Zeile, 4Zeilen
    // Switch on the backlight
    lcd.setBacklightPin(BACKLIGHT_PIN,NEGATIVE);
    //lcd.setBacklight(LED_ON);
    // Reset the LC-Display
    lcd.clear();
    delay(1000);
    lcd.home();
}

void loop()
{
    // read the input on analog pin 0:
    int sensorValue0 = analogRead(A8);
    int sensorValue1 = analogRead(A9);
    int sensorValue2 = analogRead(A10);
    int sensorValue3 = analogRead(A11);
}
```

G. Neumaier Dipl.-Gewerbelehrer	Information	Gruppe:
	ARDUINO μ C	Datum:

Name:.....

```
//AnpassenVolt 1023 -0 an 100 -0:
    sensor0_hell = sensorValue0/10; //max 1023/10 = 102
    if (sensor0_hell>100)
    {
        sensor0_hell =100; //Begrenzung auf 100 ->nur bei extremer Helligkeit
    }

sensor1_volt = sensorValue1/204.6; //1023 = 5,00 V !!Punkt statt Komma!!
sensor2_volt = sensorValue2/204.6;
sensor3_volt = sensorValue3/204.6;
//int (Schalter);
int Digital8in = Schalter;

// Ausgabe ueber serielle Schnittstelle
// 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(Digital8in);//Uebergabe Schalter (8bit von PORTA)
Serial.write(10); //Beginn neues Zeichen
delay(1);
Serial.print(sensor0_hell); //0 bis 100 LDR
Serial.write(10); //Einzelne Zeichen im String werden getrennt
//Steuerzeichen 10 ->neue Zeile, Cursor bleibt am Ort
delay(1); // Verzoegerung 1msec zur Sicherheit
Serial.print(sensorValue1);
Serial.write(10);
delay(1); // Verzoegerung 1msec zur Sicherheit
Serial.print(sensorValue2);
Serial.write(10);
delay(1); // Verzoegerung 1msec zur Sicherheit
Serial.print(sensorValue3);
Serial.write(10);
delay(1); // Verzoegerung 1msec zur Sicherheit
Serial.write(13); //!!!Wichtig: Mit dem Steuerzeichen 13 wird ein String
abgeschlossen!!!
// Steuerzeichen 13 ->Wagenrücklauf bei Schreibmaschine
delay(3); // Verzoegerung 3msec zur Sicherheit

    if (Serial.available())
    {
        value = Serial.read(); //liest ein Byte binär
    }
    leds = value; //leds oben als PORTC definiert, dort sind 8xLED
angeschlossen
    delay(5); // Verzoegerung 5msec zur Sicherheit
// (note: line 1 is the second row (Zeile), since counting begins with 0):
lcd.setCursor(0, 0); // set the cursor to column (Spalte) 0, line 0 (Zeile0)
lcd.print("TEMOPOLIS MUSEUM");
lcd.setCursor(0, 1); // set the cursor to column (Spalte) 0, line 0 (Zeile0)
lcd.print("LDR=");
//Alle 3 Möglichkeiten geben Kommazahl aus
// LCDPrintFloat(sensor0_volt); //Aufruf Funktion mit Übergabewert
// lcd.print((float)sensor0_volt, 3); //zeigt den Wert x,xxx an
    lcd.print(sensor0_hell,1); //LDR ->!! zeigt x,x, aber nur wenn sensor0_hell als
float definiert
    lcd.print("%");
    lcd.print(" A9=");
    lcd.print(sensor1_volt,1); //Sensorwert ->!! zeigt x,x
    lcd.print("V");

    lcd.setCursor(0, 2); // set the cursor to column (Spalte) 0, line 1 (Zeile1)
```

G. Neumaier Dipl.-Gewerbelehrer	Information	Gruppe:
	ARDUINO μ C	Datum:

Name:.....

```

lcd.print("A10=");
lcd.print(sensor2_volt,1);
lcd.print("V");
lcd.print(" A11=");
lcd.print(sensor3_volt,1);
lcd.print("V");
    delay(5);
lcd.setCursor(0, 3); // set the cursor to column (Spalte) 0, line 0 (Zeile0)
lcd.print("                ");
lcd.setCursor(0, 3);
lcd.print("Digital C=");
lcd.print(value);
    lcd.setCursor(13, 3);
lcd.print(" A=");
lcd.print(Digital8in);
    delay(200);

}

```