

Arduino Mega Vorbereitung für das GSF WildBugChilGru LabView PST Programm.

1. Installiere die Arduino IDE **arduino-1.8.1-windows.exe** aus:

..\WildBugChilGru-master\Arduino

Hinweis: Alternativ kann auch die aktuelle Version von www.arduino.cc heruntergeladen werden.

2. Öffne den Sketch **MEGA_PSTfreq_BME280_v1.ino** aus:

..\WildBugChilGru-master\ArduinoMEGA_PSTfreq_BME280_v1

3. Du solltest nun das sehen:

```
//-----
// Frequency Measure for a Motorcycle Testbench
// ICP4 and ICP5 are used for external signals. Based on Arduino Mega
// Rising edges will be detected
// Send commands with Newline \n
// Andreas Benz andreasnbenz@gmail.com
// SEND e for BME280 data
// SEND m for starting measurement
// MAX31855 library integrated
// Set thermo = true for max31855 data transfer
//-----OUTPUT over Serial-----
//sendCycle;Messfrequenz;Frequenz Kanal1(Pin49);Frequenz Kanal2(Pin48);Max31855 Thermocouple read
//-----
//Debug Output Pin38   PORTD ^= (1 << PD7);
//Built 1.01 02.03.2017
//-----

//-----SETUP-----
uint32_t baud = 115200;
int comlevel = 1;           // increase for lower data transfer frequenz 1 approx. 45Hz
int ringsize = 50;          // size of Ringbuffer
uint16_t thermocycle = 25;
bool thermo = false;        // MAX31855 Data (Thermoelement) send yes or no
int p_cor= 500;             // Correction for BME280 Pressure Measurement in Pa
int t_cor= 0;
int h_cor= 0;
//-----MAX31855-----

#define MAXCS 53             // CS Pin fpr MAX31855

volatile uint32_t StartTime4[50];    // ICR4-Wert bei 1.High-Flanke speichern
volatile uint32_t EndTime4[50];      // ICR4-Wert bei 2.High-Flanke speicher
volatile int ovlCAPTst4[50];
volatile int ovlCAPTend4[50];
volatile int Messung4[50];
volatile int ErsteFlanke4 = 0;
volatile int ovlTIM4 = 0;

volatile uint32_t StartTime5[50];    // ICR5-Wert bei 1.High-Flanke speichern
volatile uint32_t EndTime5[50];      // ICR5-Wert bei 2.High-Flanke speicher
volatile int ovlCAPTst5[50];
volatile int ovlCAPTend5[50];
volatile int Messung5[50];
volatile int ErsteFlanke5 = 0;
volatile int ovlTIM5 = 0;
```

4. Nun müssen mehrere Setup Parameter eingegeben werden:

- a. `uint32_t baud = 115200` Hier muss die Übertragungsgeschwindigkeit der Seriellen Kommunikation zwischen Arduino und PC festgelegt werden. Wichtig ist das LabView den gleichen Wert benutzt.
- b. Auswertung des MAX31855 Thermoelementmodul:
`bool thermo = false;` Auswertung aus
`bool thermo = true;` Auswertung aktiv
- c. BME280 Korrekturwerte
`int p_cor = 500;` Druck Korrektur in Pa
`int t_cor = 0;` Temperatur Korrektur in °C
`int h_cor = 0;` Feuchte Korrektur in %

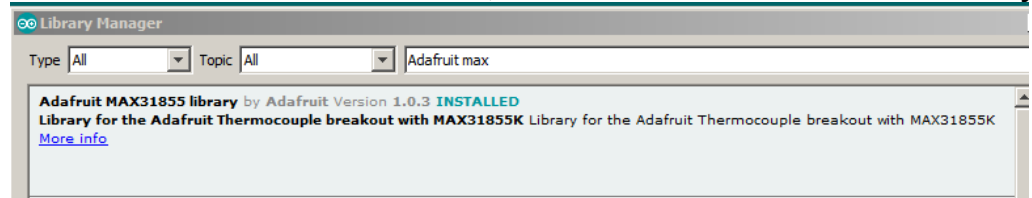
Trage hier die gewünschten Parameter ein.

Hinweis: Es sollten für den Start nur die Werte für die BME Korrektur angepasst werden. Erst wenn ein sicherer Programmablauf etabliert ist, sollte die Auswertung des MAX31855 benutzt werden. Des Weiteren befindet sich die Thermoelemente Auswertung noch in der Testphase.

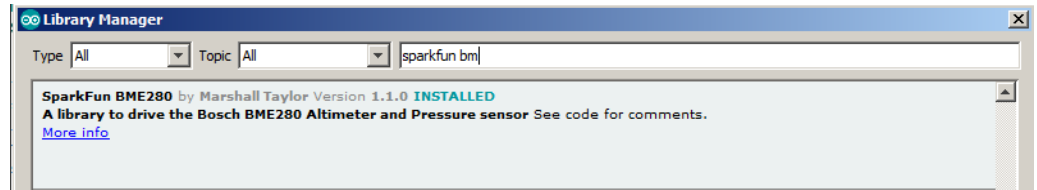
5. Hinzufügen der benötigten Libraries.

a. Variante a: **empfohlen**, Internetverbindung erforderlich:

- i. Navigiere in der Arduino IDE in der Menüleiste auf *Sketch --> Bibliothek einbinden --> Bibliotheken verwalten...*
- ii. Gebe in die Suche **Adafruit max** ein, installiere die **Adafruit MAX31855 library**



- iii. Gebe in die Suche **Sparkfun bme** ein, installiere die **SparkFun BME280** Bibliothek.



b. Variante b: keine Internetverbindung:

- i. Navigiere in der Arduino IDE in der Menüleiste auf *Sketch --> Bibliothek einbinden --> ZIP-Bibliothek einbinden...*
- ii. Binde beide im Ordner **..WildBugChilGru-master\ArduinoLibrary** befindlichen ZIP-Dateien ein.

6. Schließe nun den **Arduino MEGA R3** an einen USB Port des Computers an.

Es sollte der bekannte Windows Treiber Installation Hinweis erscheinen und die Arduino FTDI Treiber installiert werden.

7. Gehe in der Menüleiste auf *Werkzeug* --> *Port* und wähle den Port aus an dem der Arduino Mega angeschlossen ist. Hinter dem Port Name sollte **Arduino Mega 2560** stehen.

Achte darauf dass in *Werkzeug* --> *Board* der Arduino Mega ausgewählt ist.

8. Nun klicke auf Hochladen 

9. Der Sketch sollte kompilieren und anschließend auf den Arduino übertragen werden.

10. Test: Öffne den *Seriellen Monitor* in der Menüleiste *Werkzeuge*

- a. Stelle die Kommunikationsgeschwindigkeit auf *115 200 baud* ein und den Lineend auf *Newline*.

- b. gebe in das Eingabefeld ein **e** ein und drücke die Entertaste.

Es sollten nun die Klimadaten des BME280 im Ausgabefenster erscheinen. (Falls das **WildGruChil Shield** auf dem Arduino installiert ist und der BME280 sich auf dem Shield befindet).

- c. gebe in das Eingabefeld ein **m** ein und drücke die Entertaste.

Die Messung sollte beginnen.

Mit dem erneuten senden eines **m** sollte die Ausgabe beendet werden.

Der Arduino ist nun bereit für den Betrieb des Prüfstandes.

Falls etwas nicht funktioniert:

DON'T PANIC

In besonders schweren Fällen von Ratlosigkeit melde dich unter:

0171 / 206 2145

Ich werde versuchen zu helfen.

Andreas