

## Inbetriebnahme SX-Belegt-Ser-2.10 (ATTiny84 Aufsatzplatine)

1. Bestücken (Quarz, OK4 und R4 werden nicht bestückt !)
2. Mit 10 V Gleichspannung versorgen
3. Stromaufnahme 10..15mA checken
4. Arduino-IDE einrichten und
5. „Bootloader brennen“ wählen (hierbei wird allerdings kein Bootloader gebrannt, sondern nur die „Fuses“ gesetzt, zB 8Mhz interner Oszillator)
6. Programm „SX\_Belegt\_Ser\_84“ auf den Attiny hochladen
7. Seriellen Ausgang checken, es muss „S00000000“ + NewLine ausgegeben werden

(OPTIONAL: Am Besten seriell Timing checken, denn der interne Oszillator ist nicht exakt, so dass das serielle Signal möglicherweise nicht genau 9600 Baud hat – es hat sich aber gezeigt, dass bei VCC von 5V das Timing immer auf +/- 1% stimmt.

- a. Scope an Testpunkte (Optokoppler + Masse) anschliessen
- b. Timing des Seriellen Signals (104µs Pulslänge +/- 1%) checken )

Funktionskontrolle der 8 Eingangskanäle:

1. 22k Widerstand mit +10V und jeweils den 8 Eingängen verbinden
2. Entsprechender serieller Output muss auf "6" (+/-1) wechseln

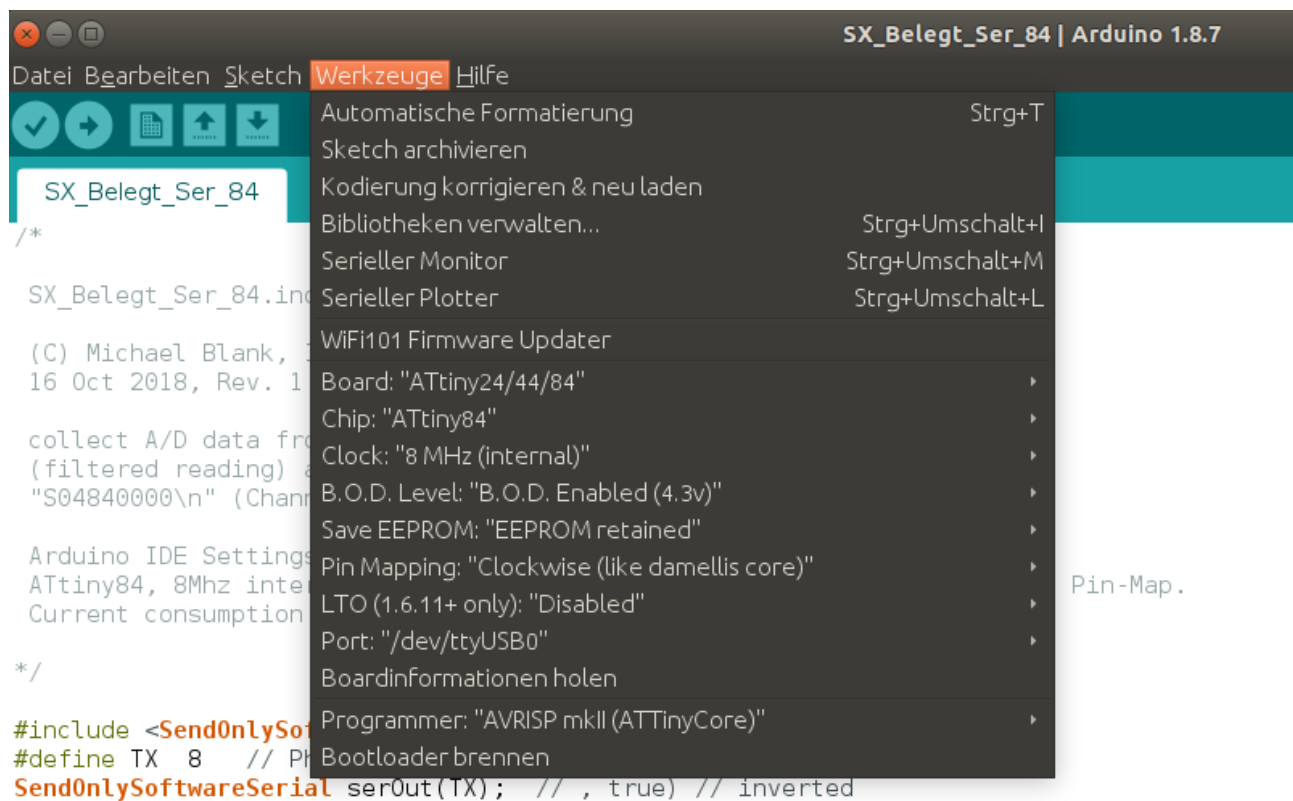
Endkontrolle

3. Auf entsprechend programmierte Basisplatine (mit Arduino Programm „SX\_Belegt\_Ser.ino“ für Arduino Pro Mini - 5V)) aufstecken und im SX-System testen

# Anhang

## Arduino-IDE für ATtiny84 einrichten

1. Board „ATtiny24/44/84“ verwenden, dazu in den IDE Einstellungen als zusätzlichen Boardverwalter [http://drazzy.com/package\\_drazzy.com\\_index.json](http://drazzy.com/package_drazzy.com_index.json) eintragen
2. Board „ATtiny24/44/88“ auswählen und als Programmierer „AVR ISP mkII (ATTiny Core)“ auswählen
3. weitere Einstellungen Chip, Clock, B.O.D, Pin-Mapping siehe Screenshot:



# Arduino Sketch Rev. 1.1

```
/*

SX_Belegt_Ser_84.ino

(C) Michael Blank, IBM Klub BB Modellbahn
16 Oct 2018, Rev. 1.1

collect A/D data from 8 analog inputs with ATtiny84 and sends the
(filtered reading) as ASCII text to main-processor, format
"S04840000\n" (Channel0 first)

Arduino IDE Settings
ATtiny84, 8Mhz internal clock, EEPROM retained, BOD disabled, clockwise Pin-
Map.
Current consumption: ~13mA

*/

#include <SendOnlySoftwareSerial.h>
#define TX 8 // Physical 5 / PB2
SendOnlySoftwareSerial serOut(TX); // , true) // inverted

#define N 8

int inp[N] = { A0, A1, A2, A3, A4, A5, A6, A7 };
uint16_t sum[N] = { 0, 0, 0, 0, 0, 0, 0, 0 };
uint16_t tmp = 0;
uint16_t out = 0;

// the setup function runs once when you power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    //pinMode(LED_BUILTIN, OUTPUT);
    serOut.begin(9600);
    analogReference(INTERNAL1V1);
}

// the loop function runs over and over again forever
// aprox. every 100msec here
void loop() {

    delay(10);

    // collect data for the 8 input channels and filter
    // 22k => output value =6 (5..7)
    // 10k => output value =9
    for (int i=0 ; i < N; i++) {
        delay(10);
```

```

    tmp = 0;
    for (int j = 0; j < 16; j++) {
        tmp += analogRead(inp[i]);    // average
    }
    tmp = tmp / 32;    // effectively dividing by 2
    if (tmp > 12) tmp = 12;    // limit for filter response
    sum[i] = ( 3 * sum[i] + tmp ) / 4;    // LP filter
}

// output to serial port „Sxxxxxxx\n“ (x= 0...9)
serOut.print('S');
for (int i=N-1 ; i >= 0; i--) {
    out = sum[i];
    if (out > 9) {
        serOut.print(9);
    } else {
        serOut.print(out);
    }
}
serOut.print('\n');
}

```