

Analoge Ausgangskarte mit 4 Kanälen und 10 Bit Auflösung (I2HAA)



I2C-Analog Output card 4 Channel 0-10V 10 Bit

Allgemeines

Die Analog-Out Karte verfügt über insgesamt 4 D/A Wandler mit einer Auflösung von 10 Bit. Die Analog-Out Karte ist komplett Register gesteuert. Dies bedeutet es gibt keine Befehle! Einzig der Inhalt der Register steuert das Verhalten der Karte. Alle Daten werden in les- und schreibbaren Registern gespeichert.

Hardware

Gesteuert werden alle Vorgänge auf der Karte von einem Mikrokontroller (PIC 18F13K22). Dieser stellt die Verbindung zwischen dem I2C-Bus und den D/A-Wandlern her. Er sorgt dafür dass die Daten aus den Registern an die DA-Wandler MCP4812 gesendet werden. Die Ausgangsspannung lässt sich mit einer Breite von 10 Bit gegen eine interne Referenzspannung von 2,048 Volt einstellen. Die nachgeschalteten Operationsverstärker sorgen für eine Ausgangsspannung im Bereich von 0 – 10 Volt. Der minimale Wert liegt bei 0x0000 (dec. 0) der Maximale 0x03FF (dec. 1023). Der maximale Wert entspricht einer Spannung von 2,048 Volt am Ausgang des MCP4812 bzw. 10,24 Volt am Ausgang der Operationsverstärker.

I2C-Adresse

Wie bei allen Karten kann man die I2C-Adresse innerhalb eines gewissen Rahmens wählen. I2C-Adressen sind immer als 7+1 Bit Adressen aufgebaut. Das letzte Bit entscheidet darüber ob der Master lesen oder schreiben will. Auf gerade I2C-Adressen wird immer geschrieben – von ungeraden Adressen wird immer gelesen. Somit ergibt sich das auf der Adresse 0xB0 (176 dec.) Daten auf die Karte geschrieben und auf der Adresse 0xB1 (177 dec.) Daten von der Karte gelesen werden Mittels Jumper 3 lässt sich die Basisadresse der Karte einstellen.

Wird der Jumper in die Position OFF gestellt, so ist die Basisadresse 0xB0. Daraus ergeben sich folgende Adressen:

176 dez = B0 hex

178 dez = B2 hex

180 dez = B4 hex

182 dez = B6 hex

184 dez = B8 hex

186 dez = BA hex

188 dez = BC hex

190 dez = BE hex

In der Position ON ist die Basisadresse 0xD0. Daraus ergeben sich folgende Adressen:

208 dez = B0 hex

210 dez = D2 hex

212 dez = D4 hex

214 dez = D6 hex

216 dez = D8 hex

218 dez = DA hex

220 dez = DC hex

222 dez = DE hex

Datenübertragung

Bei der Datenübertragung können beliebig viele Bytes gesendet oder empfangen werden. Dabei gelten vier Regeln.

- die Anzahl der Bytes muss immer ungerade sein (1, 3, 5 usw.)
- es muss mindestens ein Byte übertragen werden
- alle Registerwerte werden mit dem LSB voran gesendet
- alle überflüssigen Schreib- oder Leseoperationen werden ignoriert.

Im einfachsten Fall schreibt man immer 3 Bytes zur Analogkarte.

```
Byte 1      Zeiger = 0

Byte 2      Analogwert 0 Low Byte

Byte 3      Analogwert 0 High Byte
```

Würde man weiter 2 Bytes übertragen landen diese im Analog-Kanal 1

```
Byte 1      Zeiger = 0

Byte 2      Analogwert 0 Low Byte

Byte 3      Analogwert 0 High Byte

Byte 4      Analogwert 1 Low Byte

Byte 5      Analogwert 1 High Byte
```

oder direkt nur auf Kanal 3

```
Byte 1      Zeiger = 3

Byte 2      Analogwert 3 Low Byte

Byte 3      Analogwert 3 High Byte
```

Beispiel für Arduino

```
/*
=====
Test I2HAA board
=====
*/

#include <Wire.h>
#define I2C_AOUT_ADDR 176 >> 1 // I2HAA-Adresse als 7 Bit (Alle Jumper unten)

int AW_CH0=0; // Analogwert Kanal 1
int AW_CH1=0; // Analogwert Kanal 2
int AW_CH2=0; // Analogwert Kanal 3
int AW_CH3=0; // Analogwert Kanal 4
byte LBy=0; // Low-Byte
byte HBy=0; // High-Byte

void setup() {
  Serial.begin(9600); // Serielle Schnittstelle konfigurieren
```

```

Wire.begin(); // I2C-Pins definieren
}

void loop() {

    AW_CH0 = 250; // Spannung am Ausgang 1 z.B. 2,5V
    AW_CH1 = 500; // Spannung am Ausgang 2 z.B. 5V
    AW_CH2 = 750; // Spannung am Ausgang 3 z.B. 7,5V
    AW_CH3 = 1000; // Spannung am Ausgang 4 z.B. 10V

    // ===== Kanal 0 schreiben =====
    HBy = AW_CH0 / 256; //HIGH-Byte berechnen
    LBy = AW_CH0 - HBy * 256; //LOW-Byte berechnen

    Wire.beginTransmission(I2C_AOUT_ADDR); // Start Übertragung zur ANALOG-OUT Karte
    Wire.write(0); // Kanal schreiben
    Wire.write(LBy); // LOW-Byte schreiben
    Wire.write(HBy); // HIGH-Byte schreiben
    Wire.endTransmission(); // Ende
    delay(20);

    // ===== Kanal 1 schreiben =====
    HBy = AW_CH1 / 256; //HIGH-Byte berechnen
    LBy = AW_CH1 - HBy * 256; //LOW-Byte berechnen

    Wire.beginTransmission(I2C_AOUT_ADDR); // Start Übertragung zur ANALOG-OUT Karte
    Wire.write(1); // Kanal schreiben
    Wire.write(LBy); // LOW-Byte schreiben
    Wire.write(HBy); // HIGH-Byte schreiben
    Wire.endTransmission(); // Ende
    delay(20);

    // ===== Kanal 2 schreiben =====
    HBy = AW_CH2 / 256; //HIGH-Byte berechnen
    LBy = AW_CH2 - HBy * 256; //LOW-Byte berechnen

    Wire.beginTransmission(I2C_AOUT_ADDR); // Start Übertragung zur ANALOG-OUT Karte
    Wire.write(2); // Kanal schreiben
    Wire.write(LBy); // LOW-Byte schreiben
    Wire.write(HBy); // HIGH-Byte schreiben
    Wire.endTransmission(); // Ende
    delay(20);

    // ===== Kanal 3 schreiben =====
    HBy = AW_CH3 / 256; //HIGH-Byte berechnen
    LBy = AW_CH3 - HBy * 256; //LOW-Byte berechnen

    Wire.beginTransmission(I2C_AOUT_ADDR); // Start Übertragung zur ANALOG-OUT Karte
    Wire.write(3); // Kanal schreiben
    Wire.write(LBy); // LOW-Byte schreiben
    Wire.write(HBy); // HIGH-Byte schreiben
    Wire.endTransmission(); // Ende
    delay(20);
}

```