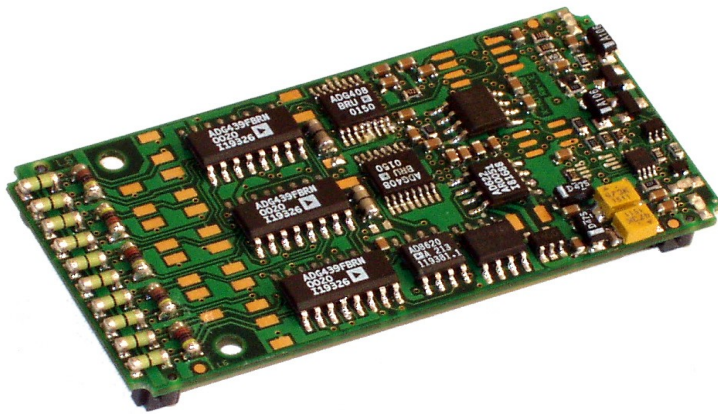


X-AD14-20

X-AD12-16

20 analoge Eingänge, bis 14 Bit Auflösung
und max. 3 Msps Abtastrate



10.5. X-AD14-20 und X-AD12-16

Inhaltsverzeichnis

10.5. X-AD14-20 und X-AD12-16.....	1
10.5.1. Beschreibung.....	2
10.5.2. Modul-Device-Treiber.....	3
10.5.2.1. Installation.....	3
10.5.2.2. Kanaleigenschaftsstruktur CPS_XAD1420.....	3
10.5.2.3. Analoge Eingänge (Differenz)	4
10.5.2.4. Analoge Eingänge (massebezogen)	5
10.5.2.5. High-Speed-Messung eines Einzelkanals (massebezogen/Differenz). .	7
10.5.2.6. Messung mit Modulen mit FIFO (ab F/N F5).....	9
10.5.2.7. Benutzung eines Triggersignals (ab F/N F5).....	12
10.5.2.8. Hardwareformate.....	13
1.1.1. Anschlusspins des Moduls (bezogen auf den Modul-Stecker A).....	14
10.5.3. Besondere Eigenschaften (Angaben gelten für Standardversion Versi- on X-AD14-20/F).....	16

10.5.1. Beschreibung

Diese Module stellen bis zu 20 externe analoge Eingänge zur Verfügung. Jeder der Eingänge kann als Masse-bezogener Kanal oder mit einem anderen Eingang zusammen als Differenzkanal gemessen werden. Die Umschaltung kann je Kanal per Software vorgenommen werden. Maximal sind also 20 Masse-bezogene oder 10 Differenz-Kanäle möglich. Außerdem kann jeder der Kanäle per Software auf 8 Eingangsspannungsbereiche (von ± 10 Volt bis ± 250 mV) eingestellt werden. Eine Ausnahme hiervon macht das Low-Cost Modul X-AD12-16/L, das nur zwei Eingangsspannungsbereiche zur Verfügung stellt. Die verschiedenen Module unterscheiden sich durch die Auflösung und die max. Wandlungsrate (0,4 Msps bis 3 Msps).

Folgende Bestückungsvarianten sind lieferbar:

Typ	Sub-typ	Modul-Version	Ein-gänge	Auflö-sung	max. Abtastrate	Bemerkung
36	1	X-AD14-20/F	20	14	2,2 Msps	Temperatur
36	2	X-AD14-20/M	20	14	0,8 Msps	Low-Power Mode, Temperatur
36	3	X-AD14-20/S	20	14	0,4 Msps	Low-Power Mode, Temperatur
36	7	X-AD12-16/L	16	12	0,8 Msps	Eingangsbereich $\pm 2,5$ V und ± 10 V
36	8	X-AD12-16/5	16	12	0,8 Msps	Eingangsbereich $\pm 2,5$ V und ± 5 V

Einige dieser Module bieten einige Sonderfunktionen, z.B. kann die on-board Temperatur zwischen -40°C und $+125^{\circ}\text{C}$ gemessen werden. Für Eich- und Abgleichzwecke kann außerdem die Offsetspannung an mehreren Stellen der analogen Verstärkerkette, z.B. des Differenzverstärkers und des A/D-Wandlers gemessen werden. Zur Bestimmung der Settle-Time bei Kanalumschaltung stehen zwei Kanäle mit $+10$ Volt und -10 Volt zur Verfügung. Der on-board Settle-Timer erspart umständliche Programmierung. Er ist programmierbar von 0 bis 65536ns in Schritten von 1ns. Zu beachten ist, dass bei anderen Bestückungen für die größten Eingangsbereiche die Settle-Time auch vom Quell-Widerstand der zu messenden Spannungsquelle abhängig

ist und berücksichtigt werden muss.

Bestückungsvarianten größter Eingangsbereich

Es gibt das Modul auch als Bestückungsvarianten mit 0..20 mA Eingängen oder mit einem max. Eingangsbereich größer ± 10 Volt, z.B. ± 50 Volt oder ± 100 Volt. Auf Wunsch sind auch andere Bereiche je Kanal möglich.

Größter Eingangsbereich	Modul-Version
± 10 Volt	/x/10 (Standardversion)
± 20 Volt	/x/20
± 50 Volt	/x/50
± 100 Volt	/x/100
0..20 mA	/x/i

10.5.2. Modul-Device-Treiber

10.5.2.1. Installation

Der Modul-Device-Treiber für das OsX hat die Programmnummer 8024h und den Dateinamen mxad1420.exe. Der Modul-Device-Treiber für Windows hat den Namen mxad1420.sys. Dies gilt für alle Subtypen des Moduls. Diese Beschreibung gilt ab der MDD Version 1.F.001.

Die Installation aus einem PC-Programm (z.B. für Steckplatz 1, Layer 0):

Error = max_load_mdd (hModul, 1, 0, 0, 0x8024, NULL, &hMDD);

Befehl in einer INS-Datei (z.B. für Steckplatz 1, Layer 0):

MAXLOADMDD slot=1 layer=0 progno=8024

10.5.2.2. Kanaleigenschaftsstruktur CPS_XAD1420

Die CPS für das Modul hat den Namen CPS_XAD1420 bzw. CPS_XAD1420_A.

CPS_XAD1420_A wurde gegenüber CPS_XAD1420 um die Elemente usHWOversampling und usSWOversampling erweitert.

10.5.2.3. Analoge Eingänge (Differenz)

Das Modul bietet 10 bzw. 8 (X-AD12-16) analoge Differenzeingänge, die über folgende CPS angesprochen werden können:

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_AIN_DIFF</i>	Kanal zu einem analogen Differenzeingang
<i>.usIndexFirst</i>	X-AD14-20: 0 ... 9 X-AD12-16: 0 ... 7	Nummer des ersten Eingangs
<i>.usIndexLast</i>	X-AD14-20: 0 ... 9 X-AD12-16: 0 ... 7	Nummer des letzten Eingangs
<i>.usReadMode</i>	<i>IO_MODE_DIRECT</i>	Direkter Lesezugriff
<i>.usFlags</i>	0 <i>_CP_EXCLUSIVE</i> <i>_CP_UNCORRECTED</i> <i>_CP_HW_FORMAT</i> <i>_CP_FORCE_BLOCK</i>	Die Daten werden im normierten, korrigierten Datenformat zurückgeliefert. Der Zugriff ist nicht exklusiv. Wie bei 0, nur dass der Zugriff exklusiv erfolgt. Die Werte werden keiner internen Korrektur unterzogen. In der momentanen Version des Treibers muss dieses Flag gesetzt werden, sofern das Flag <i>_CP_HW_FORMAT</i> nicht gesetzt ist. Es werden direkt die vom Wandler gelieferten Rohwerte zurückgegeben (das Flag <i>_CP_UNCORRECTED</i> hat keine Auswirkungen). Dieses Flag legt fest, dass die Rückgabewerte mit der Funktion <i>max_read_channel_block</i> gelesen werden müssen, auch wenn <i>usIndexFirst</i> == <i>usIndexLast</i> ist.
<i>.usRange</i>	<i>RANGE_BIP_10V</i> <i>RANGE_BIP_5V</i> <i>RANGE_BIP_2V5</i> <i>RANGE_BIP_2V</i> <i>RANGE_BIP_1V25</i> <i>RANGE_BIP_1V</i> <i>RANGE_BIP_500MV</i> <i>RANGE_BIP_250MV</i>	Bipolar ±10 V (nicht bei X-AD12-16/5) Bipolar ±5 V (nicht bei X-AD12-16/L) Bipolar ±2,5 V Bipolar ±2 V (nicht bei X-AD12-16) Bipolar ±1,25 V (nicht bei X-AD12-16) Bipolar ±1 V (nicht bei X-AD12-16) Bipolar ±500 mV (nicht bei X-AD12-16) Bipolar ±250 mV (nicht bei X-AD12-16)
<i>.usSettleTime</i>	0 ... 65535	Settle-Time in Vielfachen von 1ns.
<i>.usHWoversampling</i>	1, 2, 4, 8, 16, 32	Erst ab Modul Rev. F verfügbar! Der Wert gibt an, über wieviele Messungen eine Mittelung durchgeführt werden soll. Bei einem Wert =1 wird der Kanal nur einmal gemessen. Die Messungen und Mittelungen erfolgen direkt per Hardware.

Strukturelement	Werte	Bedeutung
<i>.usSWOversampling</i>	1 ... 65535	Erst ab Modul Rev. D verfügbar! Der Wert gibt an, über wieviele Messungen der MDD eine Mittelung durchführen soll. Bei einem Wert =1 wird der Kanal nur einmal gemessen.

Eingabedienst

Wenn das Flag *_CP_HW_FORMAT* gesetzt ist, ist der Datentyp des Kanals *DATA_SHORT*:

- **max_read_channel_short** (Einzelkanal)
- **max_read_channel_block** (Mehrkanal)

Wenn das Flag *_CP_HW_FORMAT* nicht gesetzt ist, ist der Datentyp des Kanals *DATA_LONG*:

- **max_read_channel_long** (Einzelkanal)
- **max_read_channel_block** (Mehrkanal)

Wenn das Flag *_CP_FORCE_BLOCK* und das Flag *_CP_HW_FORMAT* gesetzt sind, erfolgt der Zugriff mit:

- **max_read_channel_block** (Mehrkanal)

Wenn das Flag *_CP_FORCE_BLOCK* gesetzt ist und das Flag *_CP_HW_FORMAT* nicht gesetzt ist, erfolgt der Zugriff mit:

- **max_read_channel_block** (Mehrkanal)

10.5.2.4. Analoge Eingänge (massebezogen)

Das Modul bietet 20 bzw. 16 (X-AD12-16) analoge Eingänge (massebezogen), auf die über folgende CPS zugegriffen werden kann:

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_AIN_SE</i>	Kanal zu einem analogen massebezogenen Eingang
<i>.usIndexFirst</i>	X-AD14-20: 0 ... 19 X-AD12-16: 0 ... 7, 10 ... 17	Nummer des ersten Eingangs
<i>.usIndexLast</i>	X-AD14-20: 0 ... 19 X-AD12-16: 0 ... 7, 10 ... 17	Nummer des letzten Eingangs

Strukturelement	Werte	Bedeutung
<i>.usReadMode</i>	<i>IO_MODE_DIRECT</i>	Direkter Lesezugriff
<i>.usFlags</i>	0 <i>_CP_EXCLUSIVE</i> <i>_CP_UNCORRECTED</i> <i>_CP_HW_FORMAT</i> <i>_CP_FORCE_BLOCK</i>	Die Daten werden im normierten, korrigierten Datenformat zurückgeliefert. Der Zugriff ist nicht exklusiv. Wie bei 0, nur dass der Zugriff exklusiv erfolgt. Die Werte werden keiner internen Korrektur unterzogen. In der momentanen Version des Treibers muss dieses Flag gesetzt werden, sofern das Flag <i>_CP_HW_FORMAT</i> nicht gesetzt ist. Es werden direkt die vom Wandler gelieferten Rohwerte zurückgegeben (das Flag <i>_CP_UNCORRECTED</i> hat keine Auswirkungen). Dieses Flag legt fest, dass die Rückgabewerte mit der Funktion <i>max_read_channel_block</i> gelesen werden müssen, auch wenn <i>usIndexFirst == usIndexLast</i> ist.
<i>.usRange</i>	<i>RANGE_BIP_10V</i> <i>RANGE_BIP_5V</i> <i>RANGE_BIP_2V5</i> <i>RANGE_BIP_2V</i> <i>RANGE_BIP_1V25</i> <i>RANGE_BIP_1V</i> <i>RANGE_BIP_500MV</i> <i>RANGE_BIP_250MV</i>	Bipolar ±10 V (nicht bei X-AD12-16/5) Bipolar ±5 V (nicht bei X-AD12-16/L) Bipolar ±2,5 V Bipolar ±2 V (nicht bei X-AD12-16) Bipolar ±1,25 V (nicht bei X-AD12-16) Bipolar ±1 V (nicht bei X-AD12-16) Bipolar ±500 mV (nicht bei X-AD12-16) Bipolar ±250 mV (nicht bei X-AD12-16)
<i>.usSettleTime</i>	0 ... 65535	Settle-Time in Vielfachen von 1ns
<i>.usHWOversampling</i>	1, 2, 4, 8, 16, 32	Erst ab Modul Rev. F verfügbar! Der Wert gibt an, über wieviele Messungen eine Mittelung durchgeführt werden soll. Bei einem Wert =1 wird der Kanal nur einmal gemessen. Die Messungen und Mittelungen erfolgen direkt per Hardware.
<i>.usSWOversampling</i>	1 ... 65535	Erst ab Modul Rev. D verfügbar! Der Wert gibt an, über wieviele Messungen der MDD eine Mittelung durchführen soll. Bei einem Wert =1 wird der Kanal nur einmal gemessen.

Eingabedienst

Wenn das Flag *_CP_HW_FORMAT* gesetzt ist, ist der Datentyp des Kanals *DATA_SHORT*:

- **max_read_channel_short** (Einzelkanal)

- **max_read_channel_block** (Mehrkanal)

Wenn das Flag *_CP_HW_FORMAT* nicht gesetzt ist, ist der Datentyp des Kanals *DATA_LONG*:

- **max_read_channel_long** (Einzelkanal)
- **max_read_channel_block** (Mehrkanal)

Wenn das Flag *_CP_FORCE_BLOCK* und das Flag *_CP_HW_FORMAT* gesetzt sind, erfolgt der Zugriff mit:

- **max_read_channel_block** (Mehrkanal)

Wenn das Flag *_CP_FORCE_BLOCK* gesetzt ist und das Flag *_CP_HW_FORMAT* nicht gesetzt ist, erfolgt der Zugriff mit:

- **max_read_channel_block** (Mehrkanal)

10.5.2.5. High-Speed-Messung eines Einzelkanals (massebezogen/Differenz)

Unter Verwendung der folgenden CPS kann eine High-Speed-Messung eines Einzelkanals durchgeführt werden:

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_AIN_SE</i> <i>DEVICE_AIN_DIFF</i>	Kanal zu einem analogen Differenz- oder massebezogenen Eingang
<i>.usIndexFirst</i>	siehe Analoge Eingänge	Nummer des ersten Eingangs
<i>.usIndexLast</i>	<i>.usIndexFirst</i>	Nummer des letzten Eingangs
<i>.usReadMode</i>	<i>IO_MODE_DIRECT</i> <i>IO_MODE_MULTIPLE</i>	Direkter Lesezugriff Der Treiber liefert mehrere Messwerte des Kanals, die im Treiber nacheinander erfasst werden.
<i>.usFlags</i>	<i>_CP_HIGH_SPEED</i>	High-Speed-Messung
<i>.usRange</i>	siehe Analoge Eingänge	Messbereich
<i>.usSettleTime</i>	0	Reservierter Parameter

Anmerkungen

Mit diesem Mode ist die schnellstmögliche Abtastrate des Moduls erreichbar. In diesem Mode liefert der Kanal ohne Umrechnung die vom Wandler gelesenen Rohwerte (siehe 9.2.2.6.) zurück. Vor Beginn und nach dem Ende von Messungen muss mit Hilfe von Sonderdiensten die Messung „verriegelt“ werden.

Eingabedienst

Der Datentyp des Kanals ist `DATA_SHORT`. Ist `.usReadMode = IO_MODE_DIRECT`, erfolgt der Zugriff mit:

- `max_read_channel_short`

Ist `.usReadMode = IO_MODE_MULTIPLE` erfolgt der Zugriff mit:

- `max_read_channel_block`

Sonderdienst

- `max_channel_control`, Steuerbefehle `CMD_HIGH_SPEED_START`, `CMD_HIGH_SPEED_STOP`: Zu Beginn einer Messung muss der Steuerbefehl `CMD_HIGH_SPEED_START` gesendet werden. Danach können die Messwerte gelesen werden. Am Ende der Messung muss der Steuerbefehl `CMD_HIGH_SPEED_STOP` gesendet werden. Zwischen dem Aufruf `CMD_HIGH_SPEED_START` und `CMD_HIGH_SPEED_STOP` kann nicht mit einem anderen MDD-Kanal auf dieses Modul zugegriffen werden! Dem Dienst werden keine Daten übergeben.

Beispiel:

```
// Kanal öffnen
Error = max_open_channel(hMdd, sizeof(rcAin), &rcAin, NULL, NULL, &hAin);

// Am Anfang der Messung das Steuerkommando CMD_HIGH_SPEED_START senden
// Das Modul ist jetzt für andere MDD-Kanäle gesperrt
max_channel_control(hAin, CMD_HIGH_SPEED_START, 0, NULL);

// Jetzt können High-Speed Messungen des Kanal durchgeführt werden
// Wenn IO_MODE_MULTIPLE nicht gesetzt ist
max_read_channel_short(hAin, &sData);
...
// Wenn IO_MODE_MULTIPLE gesetzt ist
// der Treiber soll 100 Messwerte a 2 Byte (short) liefern
ulSize = 100 * 2;
max_read_channel_block(hAin, &ulSize, (void*)asData);
...
// Am Ende der Messung das Steuerkommando CMD_HIGH_SPEED_STOP senden
// Das Modul ist jetzt für andere MDD-Kanäle wieder freigegeben
max_channel_control(hAin, CMD_HIGH_SPEED_STOP, 0, NULL);
```


10.5.2.6. Messung mit Modulen mit FIFO (ab F/N F5)

Unter Verwendung der CPS CPS_XAD1420_FIFO kann der FIFO-Modus des Moduls benutzt werden

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_AIN_SE</i> <i>DEVICE_AIN_DIFF</i>	Kanal zu einem analogen Differenz- oder massebezogenen Eingang
<i>.usIndexFirst</i>	siehe Analoge Eingänge	Nummer des ersten Eingangs
<i>.usIndexLast</i>	siehe Analoge Eingänge	Nummer des letzten Eingangs
<i>.usReadMode</i>	<i>IO_MODE_RAM</i>	Messwerte werden per IO geholt und im Puffer des MDD abgelegt.
	<i>IO_MODE_DMA</i>	Messwerte werden per DMA geholt und im Puffer des MDD abgelegt (nur mit X-MAX-400 möglich)
<i>.usFlags</i>	0	Die Daten werden im normierten, korrigierten Datenformat zurückgeliefert. Der Zugriff ist nicht exklusiv.
	<i>_CP_UNCORRECTED</i>	Die Werte werden keiner internen Korrektur unterzogen.
	<i>_CP_HW_FORMAT</i>	Es werden direkt die vom Wandler gelieferten Rohwerte zurückgegeben (das Flag <i>_CP_UNCORRECTED</i> hat keine Auswirkungen).
<i>.usRange</i>	siehe Analoge Eingänge	Messbereich
<i>.usSettleTime</i>	0 ... 65535	Settle-Time in Vielfachen von 1ns
<i>.ulSampleCount</i>	1..4294967294	Anzahl der aufgenommenen Messwerte, bei der die Messung automatisch beendet wird.
	<i>XAD1420_FIFO_CONTINUOUS</i>	Die Messung läuft kontinuierlich.
<i>.ulCallbackEvent</i>	0	Keine Benachrichtigung
	<i>XAD1420_EVENT_DATA_READY</i>	Das Modul hat neue Daten vorliegen. Bei <i>IO_MODE_RAM</i> liegen mit diesem Signal die Daten auch schon im MDD vor. Bei <i>IO_MODE_DMA</i> wird nun ein DMA-Transfer gestartet.
	<i>XAD1420_EVENT_BUFFER_FULL</i>	Der Puffer des Moduls oder des MDD ist überlaufen und die Messung wurde beendet.
	<i>XAD1420_EVENT_MEASUREMENT_END</i>	Falls eine bestimmte Anzahl von Messungen angefordert wurde, wird hiermit signalisiert, dass die Messung abgeschlossen wurde.
	<i>XAD1420_EVENT_DMA_READY</i>	Messdaten wurden per DMA übertragen und liegen im MDD vor.
	<i>XAD1420_EVENT_DMA_ERROR</i>	Ein DMA-Transfer wurde mit Fehler abgebrochen.

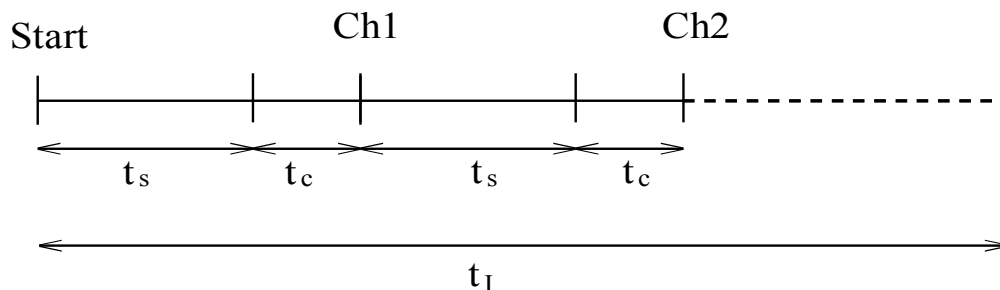
Strukturelement	Werte	Bedeutung
<code>.ulSampleInterval</code>	0 1..1970000	Aufeinanderfolgende Datenblöcke werden so schnell wie möglich erfasst Blocklänge in ns

Anmerkungen

Nach dem Öffnen des Kanals werden zunächst noch keine Wandlungen durchgeführt. Erst nach dem Aufruf des Sonderdienstes `max_channel_control` mit dem Kommando `CMD_START` werden Wandlungen durchgeführt und die Messdaten gepuffert. Die Messung kann jederzeit mit `CMD_STOP` abgebrochen werden.

Die Daten werden in diesem Modus immer per `max_read_channel_block` abgeholt. Der MDD gibt so viele Daten zurück, wie in den angegebenen Puffer passen, mindestens jedoch einen Datenblock der Größe $(\text{usIndexLast} - \text{usIndexFirst} + 1) * \text{sizeof}(\text{LONG})$ bzw. $(\text{usIndexLast} - \text{usIndexFirst} + 1) * \text{sizeof}(\text{SHORT})$. Ein solcher Datenblock enthält die Messwerte der Kanäle `usIndexFirst` bis `usIndexLast` einer Wandlung. Der Lesedienst gibt als Fehlercode `ERR_OK` zurück falls Daten vorlagen und `ERR_DATA_NOT_READY` falls der Puffer im MDD leer ist. D.h. `ERR_DATA_NOT_READY` zeigt keinen schweren Fehler an, sondern lediglich dass momentan keine Messdaten vorliegen. `ERR_DATA_NOT_READY` muss aber trotzdem mit `max_clear_error` quittiert werden. Um den Datendurchsatz zu maximieren sollte bei `max_open_channel` eine Callback-Funktion angemeldet werden, in der die Daten abgeholt werden.

Die folgende Grafik verdeutlicht die Timing-Parameter `ulSettleTime` (t_s) und `ulSampleInterval` (t_I) für eine Messung von Kanal 1 und 2. Die Zeit t_c ist die Wandlungszeit des ADC und beträgt 400ns. Falls $(t_c + t_s) * \text{Kanalzahl}$ kleiner als t_I ist, ergibt sich eine Totzeit am Ende eines Datenblockes bevor die Wandlung wiederholt wird.



Beispiel:

```
ULONG g_ulCallbackcount;
ULONG g_ulSampleCount;
```

```

MAX_CALLBACK ain_callback(MAXCHLHND hChannel, ULONG ulParam, ULONG ulSize, void *pData)
{
    LONG alAin[2];
    MAX_ERROR Error;
    ULONG ulSize_local, ulCount, *pulEvent=(ULONG*)pData;

    Error = ERR_OK;

    ulCount = 0;
    g_ulCallbackcount++;

    if (*pulEvent & XAD1420_EVENT_MEASURE_END)
    {
        printf("XAD1420_EVENT_MEASURE_END!\n");
    }

    if (*pulEvent & XAD1420_EVENT_BUFFER_FULL)
    {
        printf("XAD1420_EVENT_BUFFER_FULL!\n");
        return;
    }
    else if (*pulEvent & XAD1420_EVENT_DATA_READY)
    {
        printf("XAD1420_EVENT_DATA_READY\n");
        while (Error == ERR_OK)
        {
            {
                ulSize_local = 2*sizeof(LONG);
                alAin[0] = 0;
                alAin[1] = 0;
                Error = max_read_channel_block(hChannel, &ulSize_local, (void*)alAin);
                if (Error == ERR_OK)
                {
                    g_ulSampleCount++;
                    printf("%d|d: %0.3f\t%0.3f  [%d]\n",
                        g_ulCallbackcount, ulCount,
                        alAin[0]/1000000.0, alAin[1]/1000000.0, g_ulSampleCount);
                    ulCount++;
                }
            }
            if (Error == ERR_DATA_NOT_READY)
                max_clear_error();
        }
        if ((Error != ERR_OK) && (Error != ERR_DATA_NOT_READY))
        {
            printf("Error: %xh\n", Error);
        }
    }
}

...

rcAin.usDevice = DEVICE_AIN_SE;
rcAin.usIndexFirst = 1;
rcAin.usIndexLast = 2;
rcAin.usReadMode = IO_MODE_RAM;
rcAin.usFlags = 0;
rcAin.usRange = RANGE_BIP_10V;
rcAin.usSettleTime = 1000;
rcAin.ulSampleCount = 4000; //Kanal 1-2 wird jeweils 4000 mal gemessen
rcAin.ulCallbackEvent = XAD1420_EVENT_DATA_READY | XAD1420_EVENT_BUFFER_FULL |
XAD1420_EVENT_MEASURE_END;
rcAin.ulSampleInterval = 1500000; //1,5ms

// Kanal öffnen
Error = max_open_channel(hMdd, sizeof(rcAin), &rcAin, ain_callback, NULL, &hAin);

// Am Anfang der Messung das Steuerkommando CMD_START senden
max_channel_control(hAin, CMD_START, 0, NULL);

...

```

10.5.2.7. Benutzung eines Triggersignals (ab F/N F5)

Unter Verwendung der CPS CPS_XAD1420_FIFO kann für zwei Kanäle eine Triggerschwelle und eine zugehörige Aktion konfiguriert werden (z.B. Messung starten)

Strukturelement	Werte	Bedeutung
<i>.usDevice</i>	<i>DEVICE_TRIGGER</i>	Kanal zu einem Triggerkanal
<i>.usIndexFirst</i>	<i>0..1</i>	Nummer des Triggerkanals
<i>.usIndexLast</i>	<i>.usIndexFirst</i>	Nummer des Triggerkanals
<i>.usReadMode</i>	<i>0</i>	Keine Bedeutung
<i>.usFlags</i>	<i>0</i>	Keine Bedeutung
<i>.usRange</i>	siehe Analoge Eingänge	Messbereich des verwendeten Eingangs
<i>.usSettleTime</i>	<i>0 ... 65535</i>	Settle-Time in Vielfachen von 1ns
<i>.ulCallbackEvent</i>	<i>0</i>	Keine Benachrichtigung
	<i>XAD1420_EVENT_TRIGGER1</i>	Triggerkanal 1 (<i>usIndexFirst=0</i>) wurde ausgelöst
	<i>XAD1420_EVENT_TRIGGER2</i>	Triggerkanal 2 (<i>usIndexFirst=1</i>) wurde ausgelöst
<i>.lTriggerLevel</i>		Schaltschwelle des Triggers im normierten Datenformat ($1=1\mu\text{V}$)
<i>.ulTriggerEvent</i>	<i>TRIGGER_LOW_LEVEL</i>	Der Trigger soll bei Unterschreitung der Schaltschwelle auslösen
	<i>TRIGGER_HIGH_LEVEL</i>	Der Trigger soll bei Überschreitung der Schaltschwelle auslösen
	<i>TRIGGER_NEG_EDGE</i>	Der Trigger soll bei unter die Schaltschwelle fallender Flanke auslösen
	<i>TRIGGER_POS_EDGE</i>	Der Trigger soll bei über die Schaltschwelle steigender Flanke auslösen
<i>.ulTriggerPurpose</i>	<i>CMD_START</i>	Eine bereits konfigurierte Messung soll durch den Trigger gestartet werden
	<i>CMD_STOP</i>	Eine bereits konfigurierte Messung soll durch den Trigger beendet werden
<i>.usPreTriggerSamples</i>		Anzahl der Messdaten, die von der Zeit vor einem Triggerereignis gespeichert werden (max. 5120 Einzelwerte). Z.Z. noch nicht unterstützt
<i>.usTriggerChannel</i>	<i>0..10 (SE) bzw. 0..20 (DIFF)</i>	Index des analogen Eingangs, der als Triggerkanal dienen soll, siehe Analoge Eingänge
<i>.usTriggerDevice</i>	<i>DEVICE_AIN_SE o. DEVICE_AIN_DIFF</i>	siehe Analoge Eingänge

Anmerkungen

Nach dem Öffnen des Kanals ist der Trigger noch nicht aktiviert. Dies geschieht mit dem Kommando `max_trigger_channel`. Soll der Trigger eine Messung starten, so muss `max_trigger_channel` vor dem Starten des eigentlichen Messkanals aufgerufen werden.

Beispiel:

```

MAX_CALLBACK trigger_callback(MAXCHLHND hChannel, ULONG ulParam, ULONG ulSize, void *pData)
{
    ULONG *pulEvent=(ULONG*)pData;

    if (*pulEvent & XAD1420_EVENT_TRIGGER1)
    {
        printf("XAD1420_EVENT_TRIGGER1!\n");
        return;
    }

    if (*pulEvent & XAD1420_EVENT_TRIGGER2)
    {
        printf("XAD1420_EVENT_TRIGGER2!\n");
        return;
    }
}

...

// open trigger
cps.usDevice = DEVICE_TRIGGER;
cps.usReadMode = 0;
cps.usIndexFirst = 0;
cps.usIndexLast = 0;
cps.usRange = RANGE_BIP_10V;
cps.usSettleTime = 1000;
cps.ulCallbackEvent = XAD1420_EVENT_TRIGGER1;
cps.lTriggerLevel = 1000000; //1V
cps.ulTriggerEvent = TRIGGER_LOW_LEVEL;
cps.ulTriggerPurpose = CMD_START;
cps.usPreTriggerSamples = 0;
cps.usTriggerChannel = 2;
cps.usTriggerDevice = DEVICE_AIN_SE;

Error = max_open_channel(g_hMdd, sizeof(cps), (void*)&cps, trigger_callback, 0, &hTrigger);

Error = max_trigger_channel(hTrigger);

// starte DACQ
Error = max_channel_control(hAin, CMD_START, 0, NULL);

```

10.5.2.8. Hardwareformate

Bereich	Rohwert	Spannungswert/Stromwert
Alle Spannungsbereiche (<i>RANGE_BIP...</i>)	-8192...+8191	Neg. Vollausschlag...pos. Vollausschlag
<i>RANGE_20MA</i>	0...+8191	0...20mA

Die 12-Bit-Varianten liefern ebenfalls 14 Bit, wobei die Genauigkeit auf 12 Bit beschränkt ist.

1.1.1. Anschlusspins des Moduls (bezogen auf den Modul-Stecker A)

Kanal	SE-Kanal Eingangs- Pin	SE-Kanal GND-Pin	Differenz-Kanal: Pin für +Ein- gang	Differenz-Kanal: Pin für -Eingang	Differenz-Ka- nal: Pins für GND
AIN-0	1	2	1	3	2, 4
AIN-10	3	4	-	-	-
AIN-1	5	6	5	7	6, 8
AIN-11	7	8	-	-	-
AIN-2	9	10	9	11	10, 12
AIN-12	11	12	-	-	-
AIN-3	13	14	13	15	14, 16
AIN-13	15	16	-	-	-
AIN-4	17	18	17	19	18, 20
AIN-14	19	20	-	-	-
AIN-5	21	22	21	23	22, 24
AIN-15	23	24	-	-	-
AIN-6	25	26	25	27	26, 28
AIN-16	27	28	-	-	-
AIN-7	29	30	29	31	30, 32
AIN-17	31	32	-	-	-
AIN-8	33 ¹	34	33 ¹	35 ¹	34, 36
AIN-18	35 ¹	36	-	-	-
AIN-9	37 ¹	38	37 ¹	39 ¹	38, 40
AIN-19	39 ¹	40	-	-	-

¹ bei X-AD12-16 sind diese Pins not connected

10.5.3. Besondere Eigenschaften

(Angaben gelten für Standardversion Version X-AD14-20/F)

Parameter	Wert	Einheit
Anzahl externe Analogeingänge je Kanal wählbar per Software als Masse-bezogener Eingang (SE) oder je 2 Eingänge als Differenzeingang, auch gemischter Betrieb möglich	20	
Auflösung , max.	14	Bit
Eingangsbereiche ¹ (Standardbestückung) ±10V, ±5V, ±2,5V, (±2,0V), ±1,25V, (±1V), ±500mV, ±250mV Mögliche max. Eingangsbereiche (Bestückungsvarianten) z.B.: 0 .. 20 mA, 4 .. 20 mA, ±250 V, ±100 V, ±50 V	6 a.A.	
Wandlungszeit	450	ns
Überspannungsfestigkeit der Eingänge (Power on oder off)	-37 .. +52	V
Settle-Timer , einstellbar von 30 ns bis 60 µs	on-board	
Sonderkanäle (on-board Temperatur, Offset, ±10V)	8	
Temperaturmessung für automatischen Abgleich, Bereich Genauigkeit bei 25°C /-40°C bis +125°C Nichtlinearität Ausgangsspannung bei 0°C (typ.) Kennliniensteilheit (typ.) Langzeitstabilität, bei 125°C für 1000 Stunden (typ.)	-40 .. +125 ±3/±4 ±0,8 +424 +6,25 ±0,2	°C °C °C mV mV/°C °C
Temperatur-Bereich (Betrieb)	-40 .. +85	°C
Abmessungen	29x58x8	mm
Gewicht	9,75	g

¹ Anmerkung: Die in (Klammern) angegebenen Bereiche stehen ebenfalls zur Verfügung, die anderen Bereiche bieten aber eine höhere Genauigkeit.

Stromaufnahme max. Wandlungsrate / Low-Power Mode

Modul-Revisionen A, B, C und D

3,3V	160/150	mA
12V	45/25	mA
-12V	50/30	mA

Modul ab Rev. E:

3,3 V	65/tbd	mA
12 V	36/tbd	mA
-12 V	34/tbd	mA
