

## Funktion Installation und C#- Source für CH341A Chip

Bei ebay erhältlich (mit Stichwort „i2c usb“ suchen):



USB to I2C/IIC TWI SMBUS master Converter for ADC,Decoder,Program  
24x EEPROM 5v

**EUR 7,95**

Aus China

Sofort-Kaufen






Kostenloser Versand

59 verkauft

Wird völlig ohne irgendwelche Doku oder Software geliefert, dafür billig. Aber man findet im Internet alles Nötige. Der benutzte Chip ist nach Öffnen des Gehäuses schnell identifiziert: CH341A.

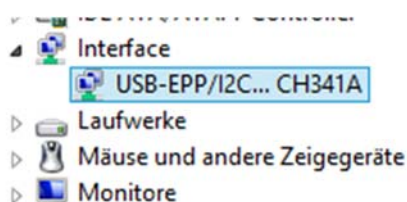
### Installation des USB- I2C- Treibers

Es müssen die richtigen Treiber vorliegen. Im Ordner CH341\_EPP\_MEM\_I2C\_IO findet man folgende Dateien

-  CH341DLL.DLL
-  CH341W64.SYS
-  CH341WDM.CAT
-  CH341WDM.INF
-  CH341WDM.SYS

Also USB- CHIP in USB einstecken und Treiber aktualisieren:

Ergebnis im Gerätemanager von Windows:



### Betrieb des USB- I2C- Chips mit C++ DLL in C#

Notwendige DLL: CH341DLL.DLL

Dort die Unterprogramme (Aus Dependency Walker)



Und eventuell noch `CH341StreamI2C`

Die Parameter für den Aufruf kann man den Header Dateien aus CH341dll.h entnehmen

```
HANDLE WINAPI CH341OpenDevice(ULONG iIndex );
VOID WINAPI CH341CloseDevice(ULONG iIndex );
BOOL WINAPI CH341ReadI2C( ULONG iIndex, UCHAR iDevice, UCHAR iAddr, PCHAR oByte );
BOOL WINAPI CH341WriteI2C( ULONG iIndex, UCHAR iDevice, UCHAR iAddr, UCHAR iByte );

BOOL WINAPI CH341StreamI2C( ULONG iIndex, ULONG iWriteLength, PVOID
iWriteBuffer, ULONG iReadLength, PVOID oReadBuffer );
```

## Jetzt in C#:

Deklaration:

```
[DllImport("ch341dll.DLL",CallingConvention=CallingConvention.Winapi)]
public extern static long CH341OpenDevice(int i);

[DllImport("ch341dll.DLL",CallingConvention=CallingConvention.Winapi)]
public extern static void CH341CloseDevice(int iIndex);
```

Die CallingConvention klappte bei WIN8.1 /64 nur mit WinApi oder StdCall.

Aufruf:

```
if (CH341OpenDevice(0) > 0)
{
    label12.Text = "Chip is open";

    CH341CloseDevice(0);
}
```

Schreiben mit Write ist schlecht, da jeder Schreibvorgang 10 ms dauert.

Besser schreiben mit Stream:

Deklaration:

```
[DllImport("ch341dll.DLL", CallingConvention = CallingConvention.StdCall)]
public extern static bool CH341StreamI2C(int iIndex, int wlen, ref byte WBuf,
int rlen, ref byte RBuf);
```

Achtung, `ref` zeigt nicht auf das Array, funktioniert nicht, sondern auf das erste Element des Arrays.

Beispiel: Schreiben eines Wertes auf den DAC mit Adresse C4.

```
byte[] WriteBuf = new byte[10];
byte[] ReadBuf = new byte[10];
WriteBuf[0] = 196; // xC4 auf 8 Bit
WriteBuf[1] = 3;
WriteBuf[2] = 255;
CH341StreamI2C(0, 3, ref WriteBuf[0], 0, ref ReadBuf[0]);
```

Oder Init des ADC und Auslesen des Abtastwertes

```

byte[] WriteBuf = new byte[10];
byte[] ReadBuf = new byte[10];

WriteBuf[0] = Convert.ToByte("92", 16);
WriteBuf[1] = Convert.ToByte("01", 16);
WriteBuf[2] = Convert.ToByte("42", 16);
WriteBuf[3] = Convert.ToByte("e3", 16);
CH341StreamI2C(0, 4, ref WriteBuf[0], 0, ref ReadBuf[0]);

```

## 2. Init- Schritt

```

// Adc auf Conversion 1 * vorher
WriteBuf[0] = Convert.ToByte("92", 16);
WriteBuf[1] = Convert.ToByte("00", 16);
CH341StreamI2C(0, 2, ref WriteBuf[0], 0, ref ReadBuf[0]);

```

## Auslesen

```

WriteBuf[0] = Convert.ToByte("93", 16);
WriteBuf[1] = Convert.ToByte("02", 16);
CH341StreamI2C(0, 2, ref WriteBuf[0], 2, ref ReadBuf[0]);
textBox1.Text=ReadBuf[0].ToString("x2")+ " " +
readBuf[1].ToString("x2")+ "\r\n";
high = (sbyte)ReadBuf[0];
mvolt = high * 32 + ReadBuf[1] / 32;
labADCOut.Text = (mvolt/1000).ToString("f3") + " V";

```

Damit geht auch ein 1 ms- Sample and Hold

Die Taktrate des I2C- Bus kann man so verstellen:

```

// *****
// Serielle Datenübertragung (I2C, RS232, SPI)
// *****
// CH341SetStream() konfiguriert I2C und SPI
// Bit 1-0: I2C speed 00= low speed /20KHz
// 01= standard /100KHz
// 10= fast /400KHz
// 11= high speed /750KHz           plus 0x60 , das ist wichtig.

```

Also:

```

[DllImport("ch341dll.DLL", CallingConvention = CallingConvention.Winapi)]
public extern static long CH341SetStream(int iIndex, int iMode);

```

und dann mit radioButtons:

```

if (rB100k.Checked) CH341SetStream(0, 0x61);
if (rB20k.Checked) CH341SetStream(0, 0x60);
if (rB40k.Checked) CH341SetStream(0, 0x62);
if (rB750k.Checked) CH341SetStream(0, 0x63);

```