

Abtasten von Wechselspannungen mit einem AVR Mikrocontroller

In diesem Artikel wird gezeigt wie mit einem AVR Atmega Mikrocontroller Wechselspannungen abgetastet werden können. Dazu wird das myAVR Board MK2 verwendet (Bild 1). Der im Board verwendete Atmega 8 Mikrocontroller wurde durch einen Atmega 328P Mikrocontroller ersetzt, da der Atmega 328P im Gegensatz zum Atmega 8 hardwareseitige Interrupts durch den Timer erlaubt. Als Software zum programmieren des Mikrocontrollers wurde Bascom eingesetzt.

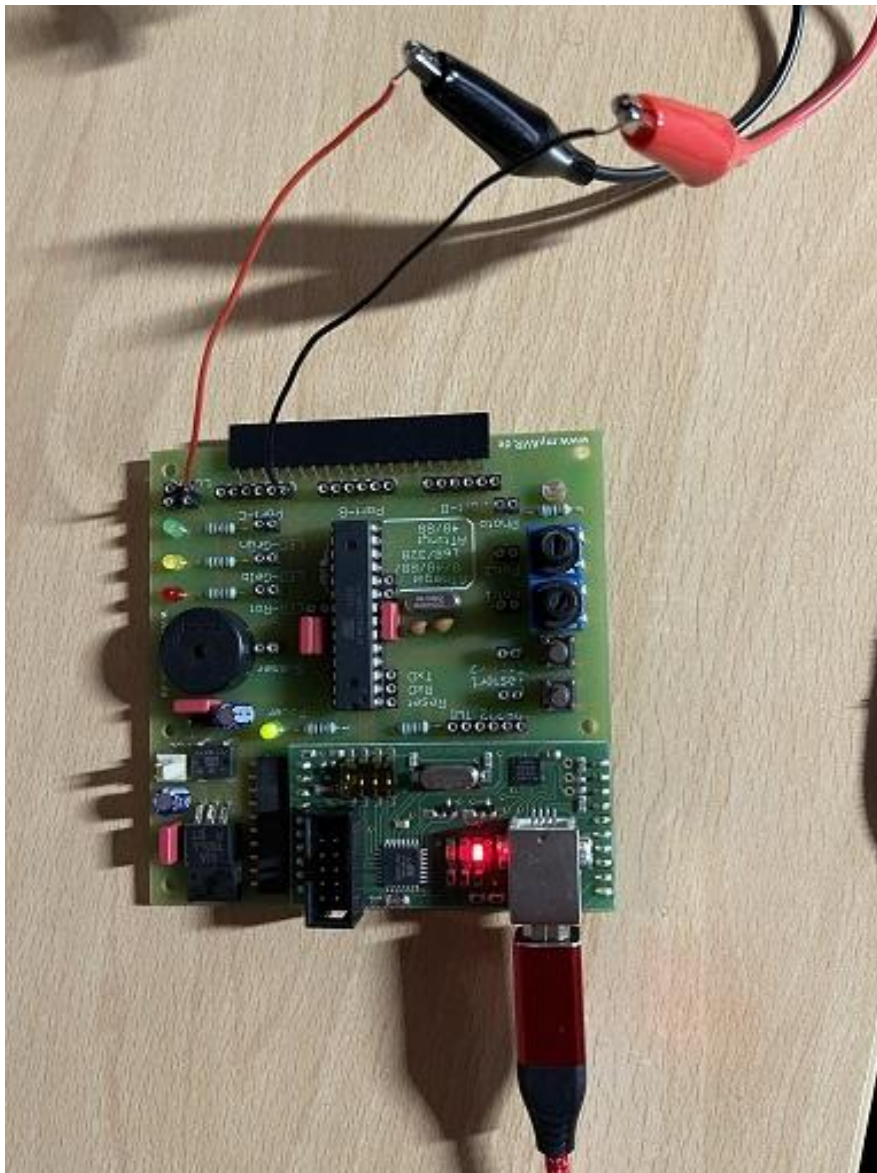


Bild 1: myAVT Board MK2

Ein Funktionsgenerator (Bild 2) wurde an den Pin PC0 (ADC0, Pin23) und an Masse des Atmega 328P angeschlossen.



Bild2: Funktionsgenerator zur Erzeugung der Wechselspannung

Der Offset des Funktionsgenerators wurde so gewählt das keine negative Spannung am ADC Eingang anliegt (Bild 3). Das wurde zunächst mit einem Oszilloskop geprüft (Bild4).

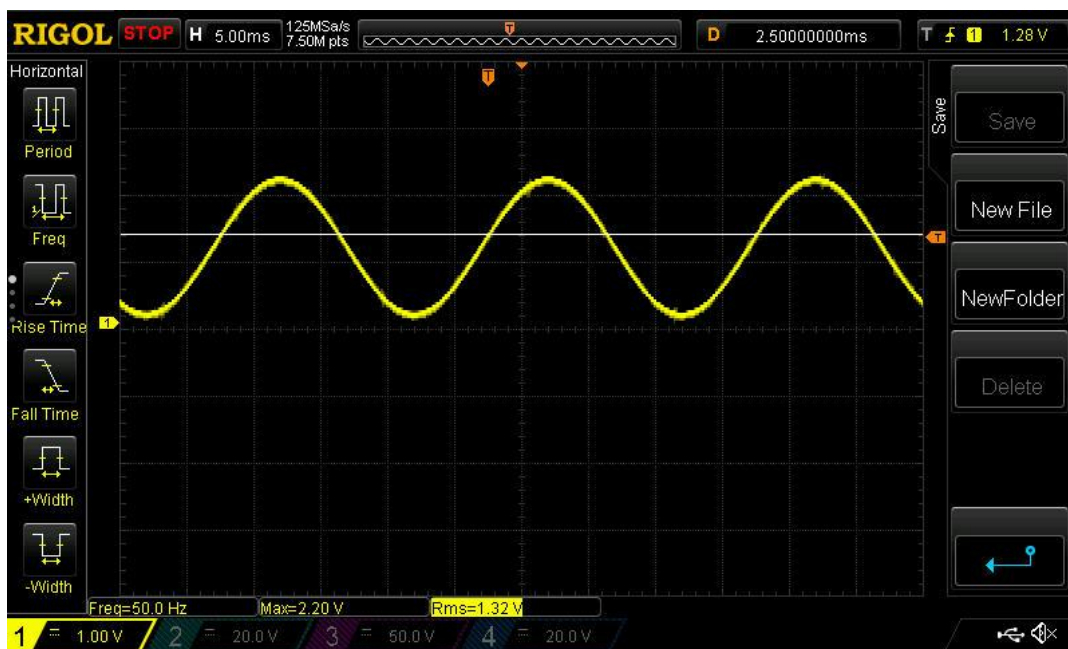


Bild 3: Wechselspannung ohne negativen Anteil

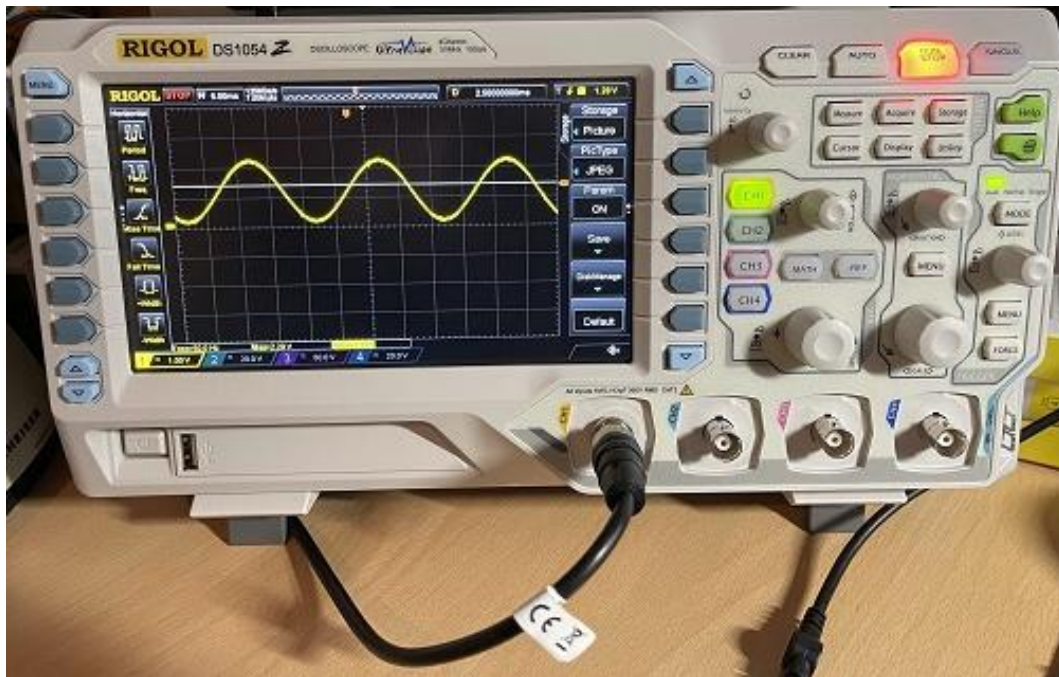


Bild4: Überprüfung des ADC Eingangssignals mit dem Oszilloskop

Das Bascom Programm ist in Listing 1 dargestellt. Als Terminalprogramm zum einlesen der Messwerte in den PC wurde Hterm eingesetzt. Damit Hterm die Werte einlesen kann, muss der Programmierer des MK2 Boards mittels des Programms mySmartUSB-Terminal in den Daten-Modus versetzt werden. Das Terminalprogramm kann die eingelesenen Daten dann als ASCII Datei speichern, welche dann wiederum in Excel importiert werden können. In Bild 5 sind die eingelesenen Messwerte dargestellt. Der Mikrokontroller hat alle $200\mu\text{s}$ einen Messwert erfasst. Es liegen also 100 Messwerte pro Periode vor bei einer hier verwendeten Eingangssignalfrequenz von 50 Hz.

Eine Übersicht der genutzten Register ist als Excel Tabelle verfügbar.

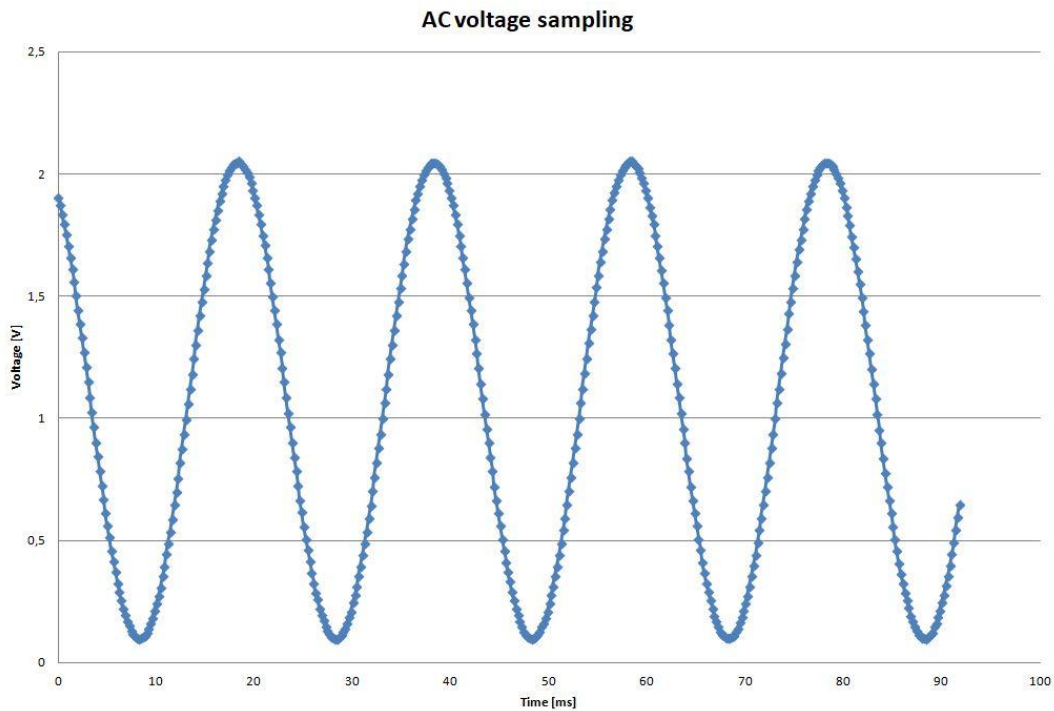


Bild 5: In Excel eingelesene Messwerte

Listing 1:

`$crystal = 16000000` 'Hier wird eine Quarfrequenz von 16 MHz eingestellt

`$regfile = "m328pdef.dat"` 'Die Definitionsdatei für den Atmega328P wird ausgewählt

`$baud = 19200` 'Die Übertragungsgeschwindigkeit zum PC wird auf 19200 Baud eingestellt

`Admux = &B01000000` 'Einstellen des Admux Registers

'Bit 7 (REFS1) und 6 (REFS0) (01) -> Spannungsreferenz für den AD-Wandler einstellen. (01) bedeutet 'AVCC mit externen Kondensator an AREF pin

'Bit 5 (ADLAR) (0) -> Ergebnisse Links- oder Rechtsbündig im ACDL und ADCH Register. (0) bedeutet

'rechtsbündig

'Bit 4 ist reserviert

'Bits 3 bis 0 (MUX3 – MUX0) (0000) -> Auswahl ADC Kanal. (0000) bedeutet ADC0

`Adcsra = &B11101111` 'Einstellen des ADCSRA Registers

' Bit 7 (ADEN) (1)-> AD einschalten

'Bit 6 (ADSC) (1)-> ADC Wandlung starten

'Bit 5 (ADATE) (1)-> Automatischer Trigger ein, Triggerquelle wird gewählt durch ADTS Bits in Register

'ADCSR

'Bit4 (ADIF) (0) -> Wird gesetzt wenn die AD-Wandlung abgeschlossen ist.

'Bit3 (ADIE) (1) -> ADC Interrupt ein

'Bits2 bis 0 (ADPS2-0) -> Stellt den Prescaler des AD-Wandlers ein. Der Prescaler ist so zu wählen,

'das der AD-Wandler Takt zwischen 50 und 200kHz liegt. Hier: (111) entspricht Prescaleer von 128

' Bei 16MHz Chiptakt ergibt sich ein AD-Wandlertakt von $16\text{MHz}/128=125\text{kHz}$.

ADCSR = &B00000101 'Bit6 (ACME) (0) -> Analog Komparator Multiplexer einschalten

'Bit2 bis 0 (101) -> ADC Auto Trigger quelle wählen. Hier (101) Timer/Counter 1 compare match B.

'Die restlichen Bits sind nicht belegt

On Adcc Adcinterrupt 'ADCC = AD Conversion Complete

Dim Array_bits(500) As Word ,Array mit 500 16 bit Werten zur Speicherung der AD Daten

Dim Qadc As Word 'nimmt bis zu 16 bits auf. Hier hin werden die 10 bit Ergebnisse des AD Wandlers 'gespeichert

Dim Xadcl As Byte At Qadc Overlay 'Xadcl wird an die Adresse von Qadc gespeichert

Dim Xadch As Byte At Qadc + 1 Overlay 'Xadch wird an die Adresse von Qadc +1 gespeichert

Dim Value As Single

Dim Adc_complete As Bit

Dim I As Long

Compare1a = 3200 'Compare Wert muss in COMPARE1A eingetragen werden

Config Timer1 = Timer , Prescale = 1 , Clear Timer = 1 , Compare B = Disconnect

'Definition des 16 bit Timers mit einem Prescaler von 1. Der Timer bewirkt alle

' $1/16000000*3200=200\mu\text{s}$ einen Interrupt aus

Enable Compare1b

Enable Interrupts

I = 1

Do

Loop Until Adc_complete = 1 'Schleife läuft solange bis 500 AD-Wandlungen abgeschlossen sind

Disable Interrupts 'Dann werden die Interrupts deaktiviert

Do 'Diese Schleife sorgt mit Ihren Aufrufen für die Berechnung des Messwertes aus den AD-Werten

'und gibt die 500 Messwerte über die serielle Schnittstelle an das Terminalprogramm weiter.

For I = 1 To 500

Qadc = Array_bits(I) '16 bit Wert für Wert wieder in Word Variable schreiben um sie im Unterprogramm anzupassen und auszugeben

Gosub Calculatevalue 'Unterprogramm Aufruf

Gosub Printvalue 'Unterprogramm Aufruf zum aufgeben der Messwerte über die Schnittstelle

Waitms 5

Next I

Loop

Calculatevalue: 'Unterprogramm zur Berechnung der Messwerte in Volt aus den AD-Werten

Value = Qadc * 4.73

Value = Value / 1023

Return

Printvalue: 'Unterprogramm zur Ausgabe der Messwerte über die serielle Schnittstelle

Print Value

Print " "

Return

Adcinterrupt: 'Unterprogramm welches vom Befehle „On Adcc Adcinterrupt“ aufgerufen wird

'und die AD-Werte in Variablen Speichert

Incr I

Xadcl = Adcl ' ADCL und ADCH sind Register die das Ergebnis der AD-Wandlung enthalten

Xadch = Adch

Array_bits(I) = Qadc 'Speicherung des aktuellen ADC Wertes in ein 500 Werte Array wobei jeder Wert ein 16 Bit (Word) Wert ist

If I = 500 Then Adc_complete = 1 'Wenn 500 Werte eingelesen sind dann aufhören

Return