

Analog-Digital-Wandler (Arduino)

- Thema
- Microchip Data Sheet
- Ziel
- Sketch
- Messung
- ADC-Prinzip
- Komponenten & Schritte
- Komparator
- Wägeverfahren
- Annäherungsverfahren
- Animation
- 4bit-Fehler
- 4bit vs. 10bit
- Referenzspannung
- Zusammenfassung
- Quellen

Thema

Frage	Wie rechnet der ADC eines ATmega328?
ADC	Analog-to-Digital Converter (Analog-Digital-Umsetzer)
Problem	Wie kann für analoge (physikalische Werte) Werte, eine binäre Darstellung gefunden werden.
Idee	Auf der "analogen Seite" den Messwert mit einem "Vergleichswert" testen.
Lösungsansatz	Auf der "digitalen Seite" einen Vergleichswert aus einer "binären Variablen" bilden. Aus dem binären Vergleichswert einen analogen Vergleichswert generieren.
	Auf der analogen Seite einen Vergleich von Vergleichswert und Messwert durchführen. Das Ergebnis des Vergleiches, als „HIGH“ oder „LOW“, an die digitale Seite zurückgeben.
	Auf der digitalen Seite einen weiteren Vergleichswert erzeugen.
	Den Vorgang für alle Stellen des binären Vergleichswertes ausführen.
Quelle	https://de.wikipedia.org/wiki/Analog-Digital-Umsetzer

Microchip Data Sheet

Aus Kapitel 24.2

The Atmega ... features a **10-bit successive approximation** ADC.

Schrittweise (aufeinanderfolgend) Annäherung.

10 Bits

0 0 | 0 0 0 0 | 0 0 0 0 (ergibt dezimal 0 bis 1023), also 1024 Stufen

Folgerungen

Das Arduino Board verfügt über einen 10 Bit Analog-Digital-Umsetzer.

Eingangsspannungen zwischen 0 bis 5 Volt werden auf einen Bereich von 0 bis 1023 abgebildet.

Die Auflösung liegt dann bei $5 \text{ Volt} / 1024 = 4,9 \text{ mV}$ pro Stufe.

Der Einlesevorgang für einen analogen Spannungswert liegt bei $100 \mu\text{s}$ ($0,0001 \text{ s}$); pro Sekunde lassen sich 10.000 Messungen durchführen

Quelle:

<https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf>

<https://www.rahner-edu.de/mikrocontroller/arduino-uno/spannungen-messen/>

Ziel

Messwert V_{IN} in Volt
überführen in einen digitalen (binären) Wert

Aus analog

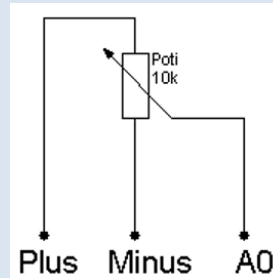
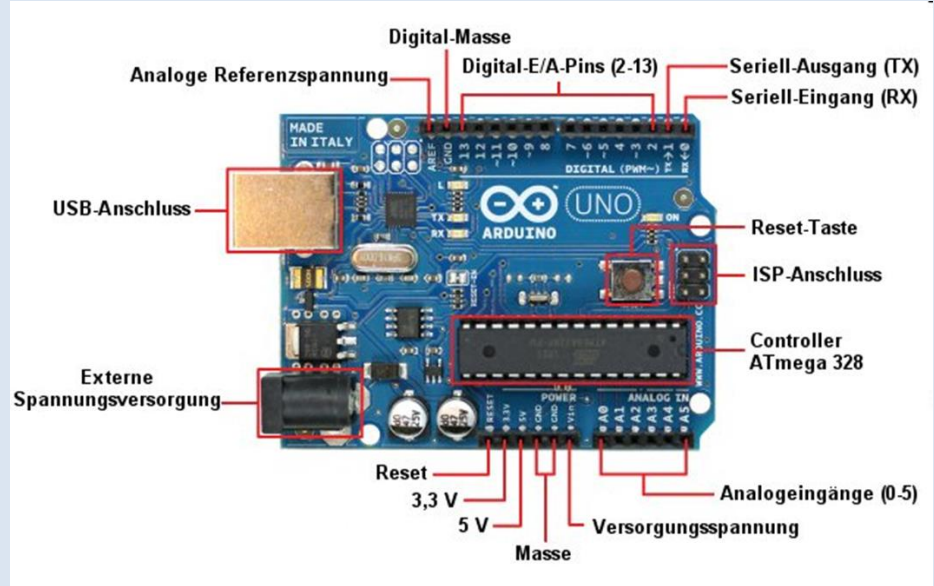
3 Volt

soll digital werden

0001001100

mit digitaler Anzeige (Monitor)

3,000 V



Sketch

```
int analogPin = A0;           // select the input pin for the potentiometer
int sensorValue = 0;          // digital values from 0 to 1023
double vAREF = 4.98;          // Arduino-Referencevoltage set to External
double sensorValueInVolt = 0.0; // sensorValueInVolt = sensorValue / 1024 * vAREF

void setup() {
  Serial.begin(9600);
  analogReference(EXTERNAL);    // DEFAULT (5 V) INTERNAL (1.1 V)
}

void loop() {
  // read the value from the sensor; values range from 0 to 1023
  sensorValue = analogRead(analogPin);

  // recalculate sensorValue to a value in Volt
  sensorValueInVolt = double(sensorValue)/1024 * vAREF;

  // Sending values to monitor
  Serial.print(sensorValue,DEC);
  Serial.print("    ");
  Serial.println(sensorValueInVolt,DEC);
  delay(500);
}
```

Messung

Externe Referenzspannung $v_{AREF} = 4.98$
7805 Voltage regulator



Spannung am Poti Realer Wert von Multimeter
3,14 Volt



Wert vom Arduino-Analog-Eingang `sensorValue = analogRead(analogPin);`
 `sensorValue = 646`

COM3	
647	3.1465430259
646	3.1416797637

10bit-successive 1024 Stufen (0 bis 1023)
approximation register 0 0 | 0 0 0 0 | 0 0 0 0

SAR Ergebnis 1 0 | 1 0 0 0 | 0 1 1 0 entspricht dezimal **646**

Umrechnung in Volt `sensorValueInVolt = double(sensorValue)/1024 * vAREF;`

Arduino ADC $3,1218 = \mathbf{646} / 1024 * 4.98$
ADC Wert **3,1418**

ADC-Prinzip

Block diagram

DAC = digital-to-analog converter

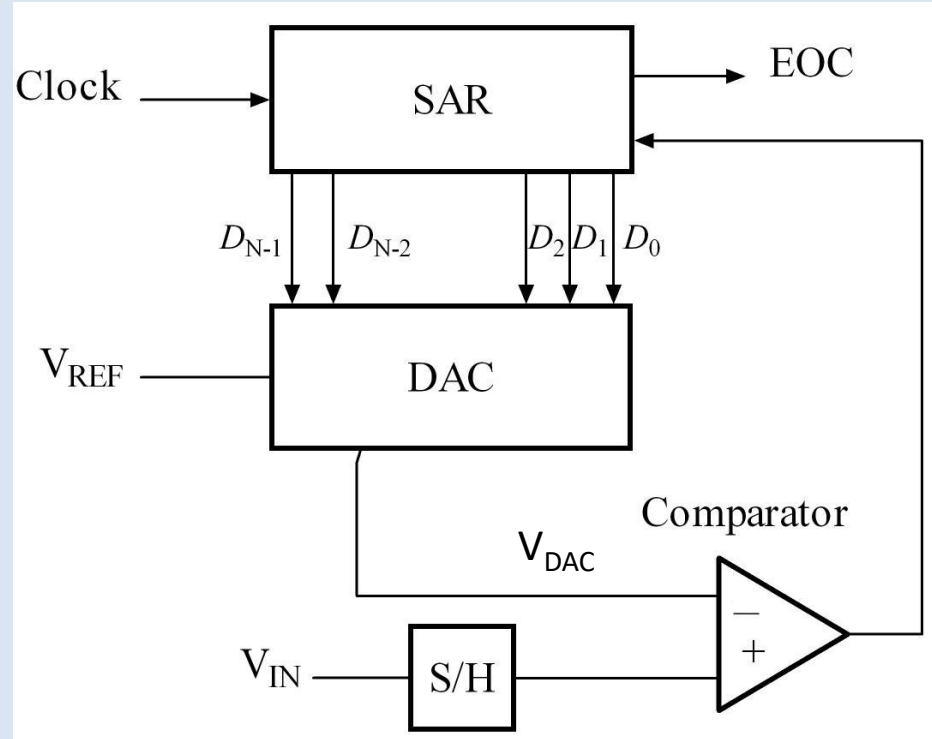
EOC = end of conversion

SAR = successive approximation register

S/H = sample and hold circuit

V_{IN} = input voltage

V_{REF} = reference voltage (5 V)



https://en.wikipedia.org/wiki/Successive-approximation_ADC#/media/File:SA_ADC_block_diagram.png

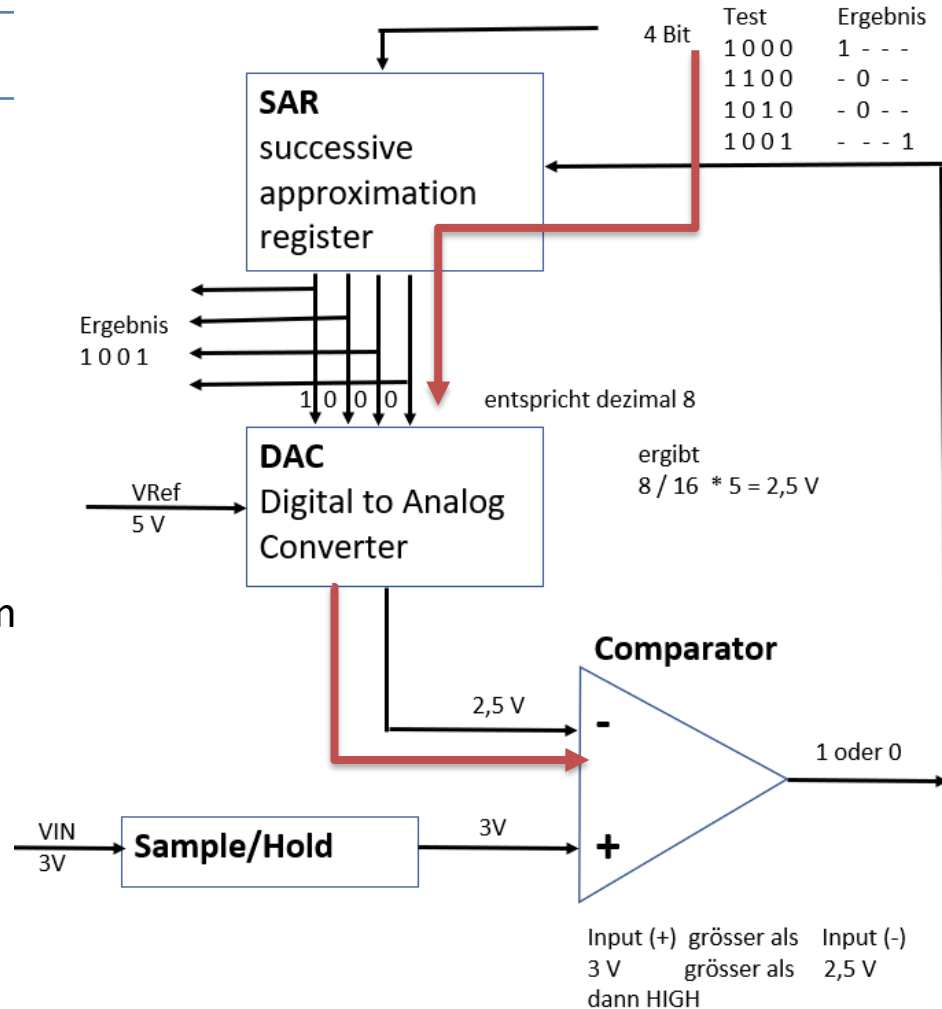
Komponenten & Schritte

Komponenten:

- SAR Register
- Digital-Analog Wandler DAC
- Komparator
- Referenzspannungsquelle

Schritte:

- SAR erzeugt Test-Bitmuster
- DAC setzt Bitmuster in Spannung um
- Sample/Hold hält Testwert fest
- Komparator vergleicht Spannungen



Komparator

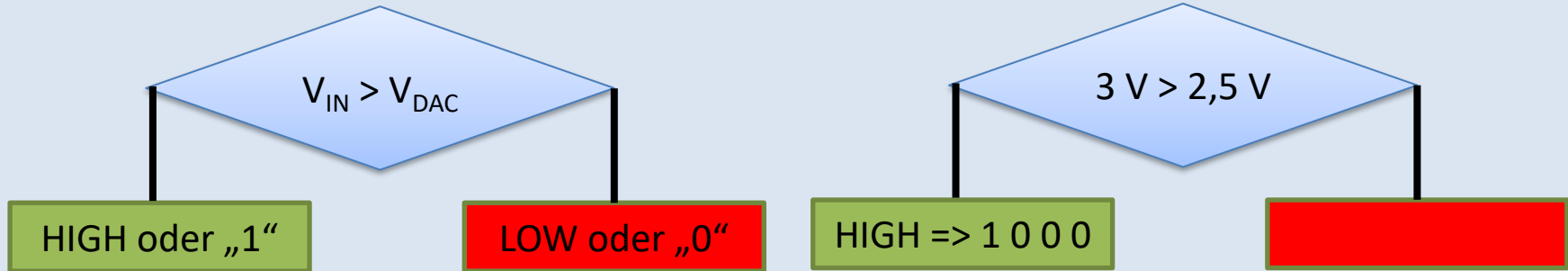
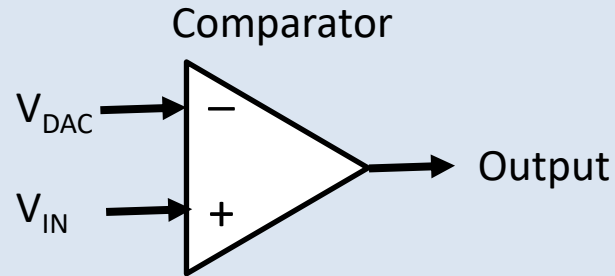
(analoge Werte vergleichen)

V_{DAC} Vergleichswert vom DAC (2,5 V)

V_{IN} Messwert (3 V)

$V_{DAC} > V_{IN} \Rightarrow \text{Output} = \text{HIGH oder „1“}$

$V_{DAC} < V_{IN} \Rightarrow \text{Output} = \text{LOW oder „0“}$



Achtung: Hier als Beispiel, ein 4-Bitregister angenommen!

Quelle: [https://de.wikipedia.org/wiki/Komparator_\(Analogtechnik\)](https://de.wikipedia.org/wiki/Komparator_(Analogtechnik))

Wägeverfahren

1

4-Bitregister

Stufen = 16

$V_{REF} = 5\text{ V}$

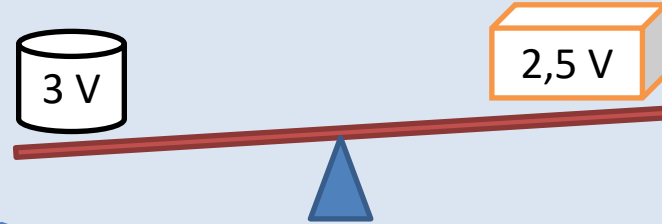
$V_{IN} = 3\text{ V}$

$$\text{Test } V = \text{SARWert} / \text{Stufen} * V_{ref}$$

SARWert Startwert MSB

1 0 0 0

$$1\ 0\ 0\ 0 \Rightarrow 8$$
$$8 / 16 * 5 = 2,5\text{ V}$$



Das 1 Bit
bleibt stehen

$$3\text{ V} > 2,5\text{ V}$$

HIGH \Rightarrow 1 0 0 0

Wägeverfahren

2

4-Bitregister

Stufen = 16

$V_{REF} = 5\text{ V}$

$V_{IN} = 3\text{ V}$

Test $V = \text{SARWert} / \text{Stufen} * V_{ref}$

SARWert Folgewert

1 1 0 0

1 1 0 0 \Rightarrow 12

12 / 16 * 5 = 3,75 V

3 V

3,75 V

3 V > 3,75 V

Das 2 Bit
wird entfernt

LOW \Rightarrow 1 0 0 0

Wägeverfahren

3

4-Bitregister

Stufen = 16

$V_{REF} = 5\text{ V}$

$V_{IN} = 3\text{ V}$

Test $V = \text{SARWert} / \text{Stufen} * V_{ref}$

SARWert Folgewert

1 0 1 0

1 0 1 0 $\Rightarrow 10$

$8 / 16 * 5 = 3,125\text{ V}$

3 V

3,125 V

3 V > 3,125 V

Das 3 Bit
wird entfernt

LOW $\Rightarrow 1\ 0\ 0\ 0$

Wägeverfahren

4

4-Bitregister

Stufen = 16

$V_{REF} = 5\text{ V}$

$V_{IN} = 3\text{ V}$

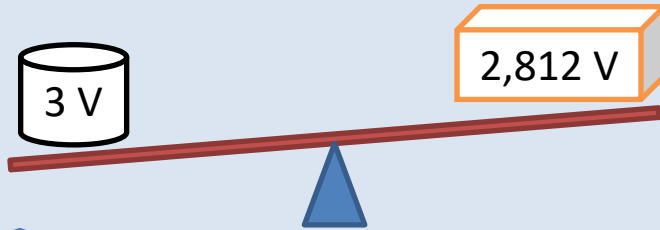
Test $V = \text{SARWert} / \text{Stufen} * V_{ref}$

SARWert Folgewert

1 0 0 1

1 0 0 1 \Rightarrow 9

$9 / 16 * 5 = 2,812\text{ V}$



Das 4 Bit
bleibt stehen

$3\text{ V} > 2,812\text{ V}$

HIGH \Rightarrow 1 0 0 1

Wägeverfahren

5

4-Bitregister

Stufen = 16

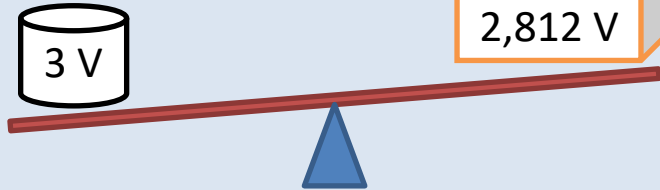
$V_{REF} = 5\text{ V}$

$V_{IN} = 3\text{ V}$

$$\text{Test } V = \text{SARWert} / \text{Stufen} * V_{ref}$$

SARWert Ergebnis

1 0 0 1



Kontrollrechnung

$$\begin{aligned} 1\ 0\ 0\ 1 &\Rightarrow 9 \\ 9 / 16 * 5 &= 2,812\text{ V} \end{aligned}$$

Fehler

$$\begin{aligned} 3,0 &\quad \text{Soll} \\ - 2,8125 &\quad \text{Ist} \\ = 0,1875 &\quad \text{Abweichung} \end{aligned}$$

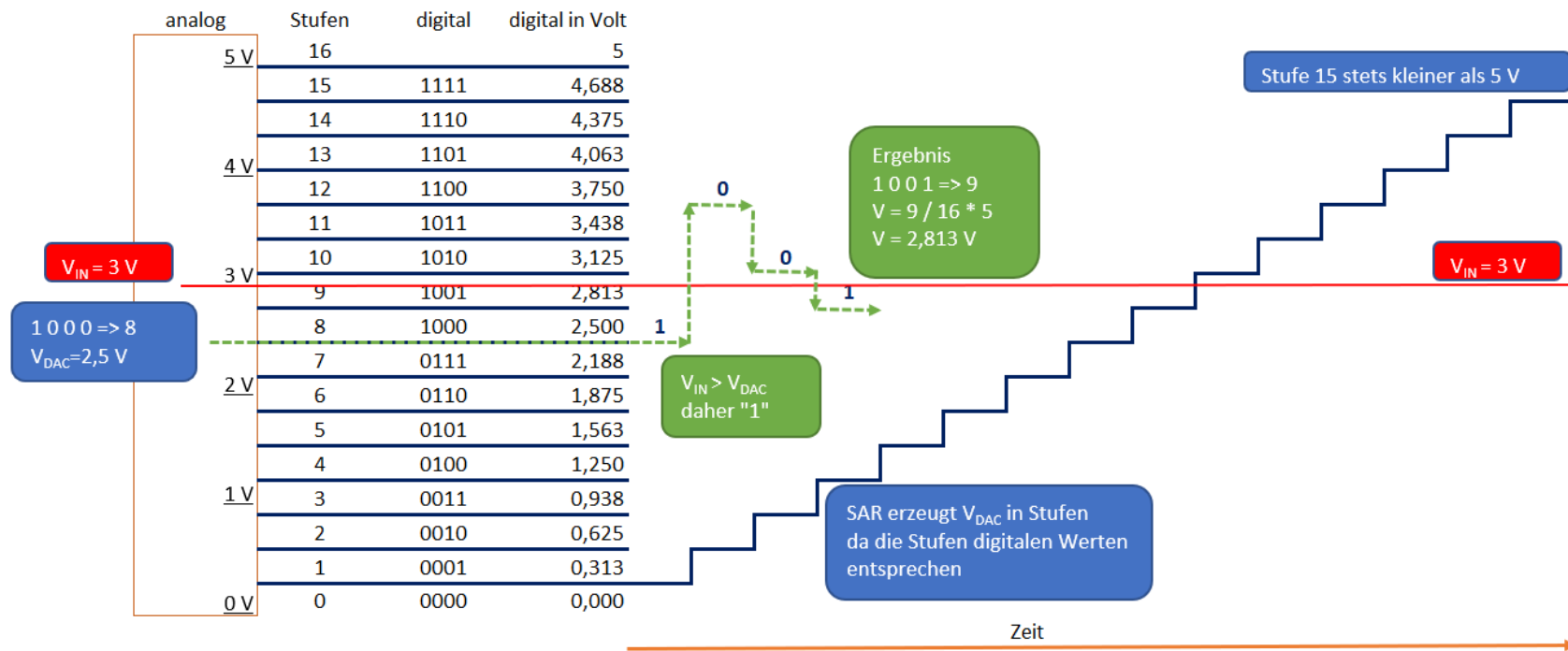
Abhilfe: 10-Bitregister des ATmega328

Annäherungsverfahren

Vref	5	V
Register	4	bit
Stufen	16	
Schrittweite	0,3125	V

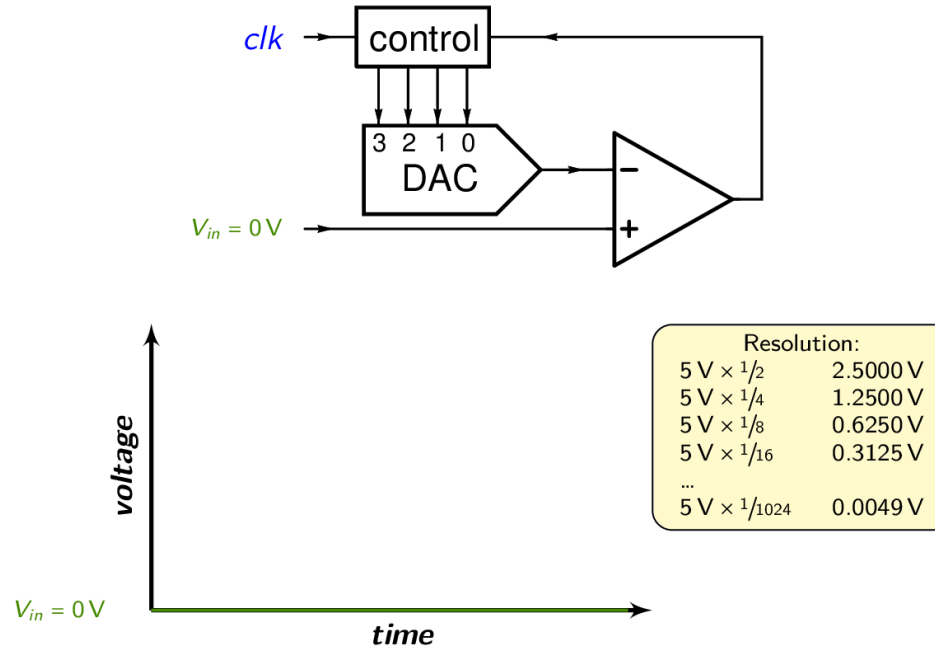
4-Bitregister haben 16 mögliche Werte

$0,3125 = 5 / 16$



Animation

Successive Approximation – example of a 4-bit ADC



[https://upload.wikimedia.org/wikipedia/commons/b/b2/4-bit Successive Approximation DAC.gif](https://upload.wikimedia.org/wikipedia/commons/b/b2/4-bit_Successive_Approximation_DAC.gif)

4bit-Fehler

Stufen 16 4bit-Register

Vref

VIN

4bit-Reg.	Dezimal	Test V	Comparator	Kontrolle
1000	8	2,5	0	Umrechnung binär nach dezimal
100	4	1,25	0	3
110	2	0,625	1	entspricht in Volt
111	3	0,9375	1	0,9375 V

VIN

4bit-Reg.	Dezimal	Test V	Comparator	Kontrolle
1000	8	2,5	1	Umrechnung binär nach dezimal
100	12	3,75	0	9
110	10	3,125	0	entspricht in Volt
111	9	2,8125	1	2,8125 V

VIN

4bit-Reg.	Dezimal	Test V	Comparator	Kontrolle
1000	8	2,5	1	Umrechnung binär nach dezimal
100	12	3,75	1	15
110	14	4,375	1	entspricht in Volt
111	15	4,6875	1	4,6875 V

4bit vs. 10bit

Stufen 16 4bit-Register

Vref 4,8 V

VIN 3,13 V

4bit-Reg.	Dezimal	Test V	Comparator	Kontrolle
1000	8	2,4	1	Umrechnung binär nach dezimal
100	12	3,6	0	10
110	10	3	1	entspricht in Volt
111	11	3,3	0	3 V

Stufen 1024 10bit-Register

Vref 4,8 V

VIN 3,13 V

10bit-Reg.	Dezimal	Test V	Comparator	Umrechnung binär nach dezimal
10	512	512	2,4	1
9	256	768	3,6	0
8	128	640	3	1
7	64	704	3,3	0
6	32	672	3,15	0
5	16	656	3,075	1
4	8	664	3,1125	1
3	4	668	3,13125	0
2	2	666	3,121875	1
1	1	669	3,1359375	0

Referenzspannung

ATmega328

Voltage reference selection	Volt	analogReference()
AVCC (gleich VCC)	< 5 V (USB)	DEFAULT
External reference AREF	z.B. 5 V	EXTERNAL
Internal reference	1,1 V	INTERNAL

AREF-PIN Ein Voltmeter an AREF gibt aktuelle Referenzspannung aus!

Einfluss auf Messwerte	5 V Referenzspannung	1,1 V Referenzspannung
Auflösung	5 V / 1024 = 4,9 mV	1,1 V / 1024 = 1,1 mV
Max. Messwert	5 V	1,1 V

Analog Grenzwert 5 V (grössere Werte führen zur Beschädigung)

Quelle: <https://www.youtube.com/watch?v=E8GqHvOK4DI&t=61s>

Zusammenfassung

Das SAR-Verfahren liefert hinreichend genaue digitalisierte Werte.

Die digitalisierten Werte hängen ab von

- der Auflösung
- und der Referenzspannung.

Die schrittweise Annäherung (successive approximation) führt stets zu einem Verlust von Genauigkeit (Information).

Der digitale Wert liegt stets unterhalb des zu messenden Wertes.

Um bessere Ergebnisse zu erreichen muss die Auflösung erhöht werden. Im Falle des Arduino muss man dann einen externen ADC, z.B. mit 12bit verwenden.

Die Anzahl der Messungen pro Sekunde ist begrenzt, da das SAR-Verfahren für die schrittweise Annäherung eine gewisse Zeit beansprucht.

Quellen

YouTube

[Electronic Basics #27: ADC \(Analog to Digital Converter\) – YouTube](#)
[Successive Approximation ADC Explained – YouTube](#)
[Comparators: The Building Blocks of Analog to Digital Converters \(ADC\) – YouTube](#)
[Electronic Basics #10: Digital to Analog Converter \(DAC\) - YouTube](#)

[Der Analog-Digital Wandler \(tu-chemnitz.de\)](#)

[Successive-approximation ADC – Wikipedia](#)
[Analog-Digital-Umsetzer – Wikipedia](#)
[Sukzessiv-Approximation ADC - gaz.wiki](#)

[Understanding SAR ADCs: Their Architecture and Comparison with Other ADCs - Tutorial - Maxim \(maximintegrated.com\)](#)

[Arduino-Programmierung Analogeingang \(netzmafia.de\)](#)

[SAR-Verfahren :: successive approximation register \(SAR\) :: ITWissen.info](#)