

NWT	Klasse 10	Microcontroller 2
Der Arduino bekommt Sinne – Messen mit dem Digitaleingang		

Bisher hat der Arduino immer nur etwas GETAN. Er hat die Anweisungen ausgeführt, die du ihm gegeben hast. Wenn Menschen sich so verhalten würden, wäre die Welt eintönig – wir müssten nur tun was man uns sagt und wären gleichzeitig taub, blind und gefühllos.

Abhilfe: Alle Ports, die du bisher als AUSGÄNGE kennengelernt hast, lassen sich auch als EINGÄNGE benutzen. Statt **digitalWrite()** und **analogWrite()** wird dazu der Begriff **digitalRead()** oder **analogRead()** benutzt.

Du baust jetzt mithilfe des Arduino ein Batterietestgerät, indem du das Potenzial des +-Anschlusses misst.

```
int Wahrnehmung;
void setup()
{
  pinMode(13, INPUT);
}
void loop()
{
  Wahrnehmung = digitalRead(13);
  delay(500);
}
```

Setzt den Modus für Port 13 auf „Eingang“

Der Arduino liest das Potenzial, das am Port 13 anliegt, aus: Alles, was über ca. 1,4 Volt ist, wird vom Arduino als „an“, „HIGH“ bzw. „1“ wahrgenommen, Potenziale unter 1,4 V spürt er als „aus“, „LOW“ bzw. „0“. Der Wert (0 oder 1) wird dann an die Variable Wahrnehmung übergeben.

Aufgabe 1: Stecke ein Kabel in Port 13. Übernimm das Programm von oben. Lass dir zusätzlich den Wert für *Wahrnehmung* auf dem seriellen Monitor ausgeben.
Stecke nun das andere Ende des Kabels nacheinander in GND, in 3,3V und in 5V.

Infoblatt:

Nochmal zur Klärung: Digitaleingänge

Eingänge werden dazu benutzt, um mit Hilfe von Signalen von z.B. Schaltern, Tastern, Lichtschranken, Wassermeldern, Temperatursensoren,... Programmabläufe zu beeinflussen: wenn der Schalter gedrückt wird / die Temperatur 25°C übersteigt /... dann soll der µC einen bestimmten Programmblock ablaufen lassen.

Alle digitalen Ports des µC lassen sich entweder als Ausgänge (z.B. `pinMode(13, OUTPUT);`) oder als Eingänge (z.B. `pinMode(13, INPUT);`) konfigurieren. Eingang bedeutet dabei allerdings nicht, dass Strom in den µC hinein fließt: ein Eingang ist ein Messeingang für Potenziale. Er lässt keine bzw. nur eine vernachlässigbar kleine Stromstärke hineinfließen und misst die bzgl. des Nullniveaus (Ground) anliegende Spannung.

Wird z.B. Port 13 als Eingang (Input) verwendet, misst der µC fortlaufend das Potential, das an Port 13 anliegt. Abfragen lässt sich dieser Wert mit dem Befehl `digitalRead(Port);`. Zum Beispiel liefert der Befehl `digitalRead(13)` den Wert 0 („low“), wenn das Potential an Port 13 unter 1,4 V liegt, und 1 („high“), wenn das Potential an Port 13 über 1,4 V liegt.

So ein Mikrocontrollereingang, der nur zwei Zustände (0 oder 1) unterscheidet, heißt **Digitaleingang**.

Wenn du die Spannung deiner Batterie genau bestimmen möchtest, hilft dir die Tatsache, dass die analogen Ports A0 bis A5 nicht nur zwei, sondern 1024 verschiedene Potenzialwerte (0 bis 1023) einlesen können. Der Befehl dafür lautet: `analogRead(analoger Port);`

Je nachdem, welches Potenzial an den Ports anliegt, wird ein Wert zwischen 0 und 1023 ausgegeben. Die Schrittweite („Auflösung“) beträgt dabei $5V : 1023 \approx 0,0049V \approx 0,005 V$. Damit ergibt sich die nebenstehende Tabelle.

Potenzial am Port	analoger Wert
0 V	0
$0,0049 V \approx 0,005 V$	1
$0,0098 V \approx 0,01 V$	2
...	...
1,0 V	205
...	...
3,3 V	675
...	...
5,0 V	1023