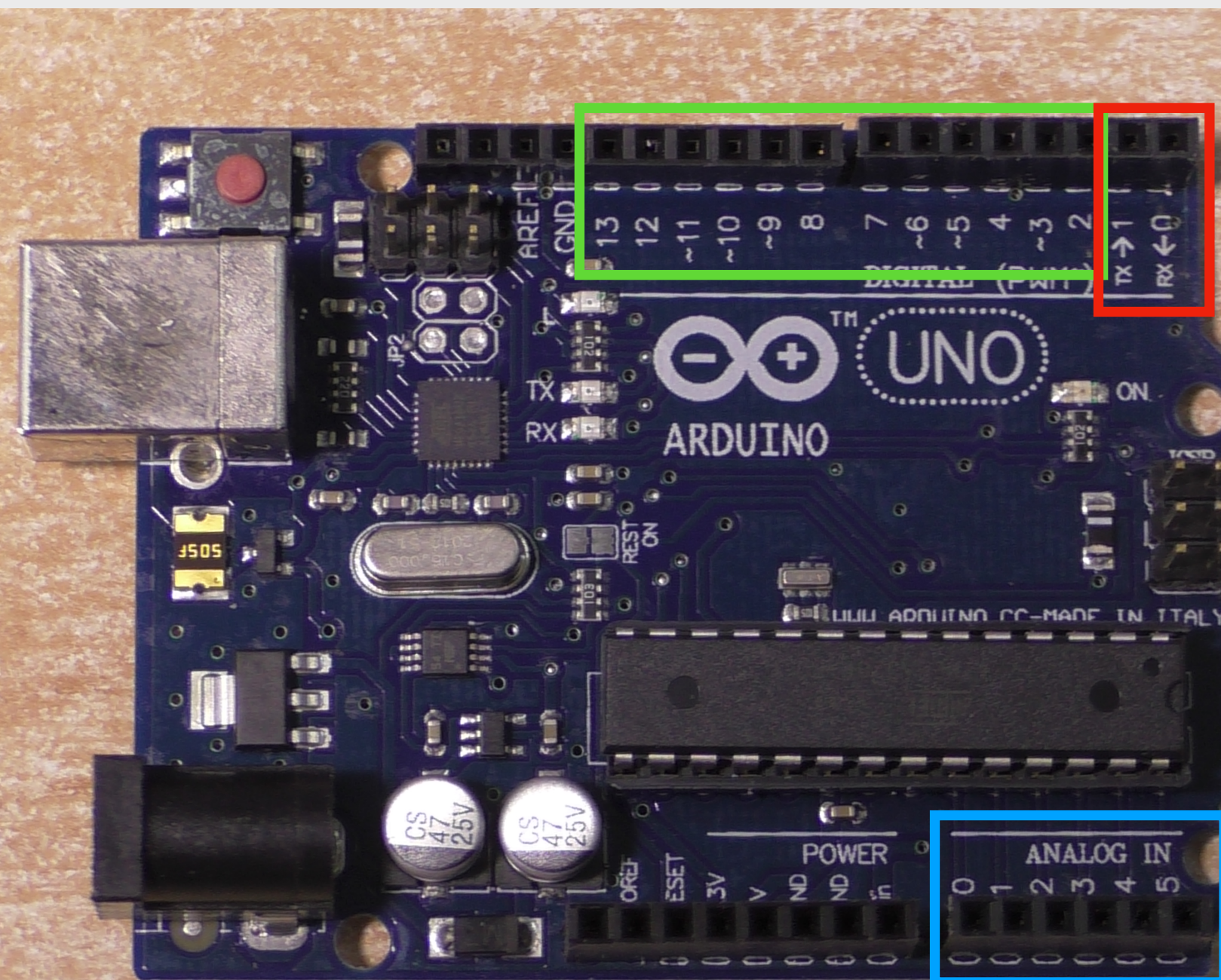


# Digitale Ausgänge

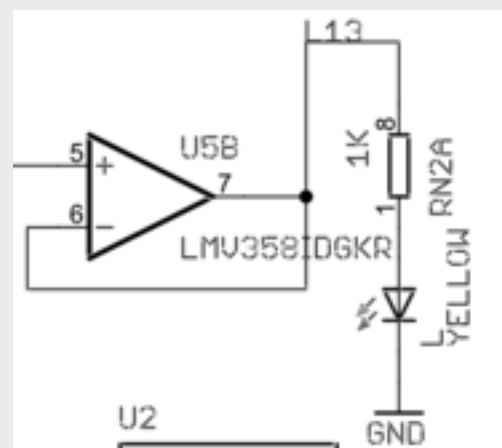


## Pin 0 und 1

Diese Pins werden zur seriellen Kommunikation und für die USB - Schnittstelle verwendet. Wenn du nicht genau weisst, was tu tust, dann verwende sie nicht!

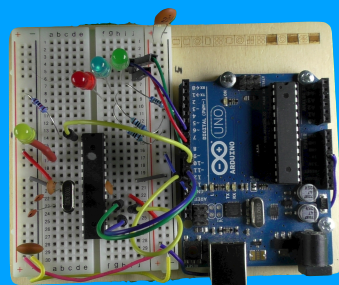
## Pin 2 bis 13

Das sind die normalerweise verwendeten Pins. An Pin 13 ist die interne LED angeschlossen. Diese belastet aber den Ausgang nicht.

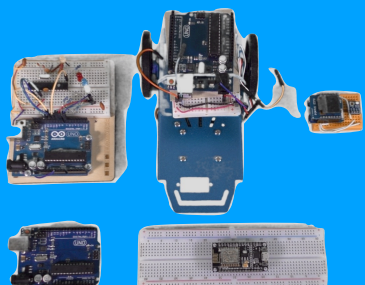


## Pin A0 bis A5

Das sind eigentlich analoge Eingänge. Wenn du sie nicht brauchst, dann kannst du sie auch digital verwenden.



Der schnelle einfache Einstieg in Arduino & Co.  
Digitale Ausgänge





# Die Befehle

Vorbereitung: `pinMode(pinNummer, OUTPUT);`

Ohne diesen Befehl arbeitet der Pin als Eingang. Daher werden üblicherweise in `setup()` alle Ausgabepins mit diesem Befehl vorbereitet. Dies funktioniert auch mit A0 bis A5.

5 V ausgeben: `digitalWrite(pinNummer, HIGH);`

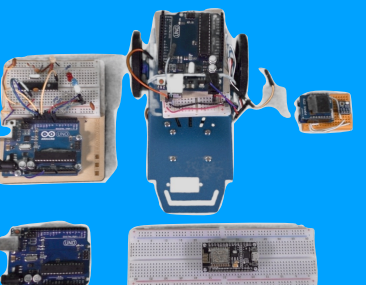
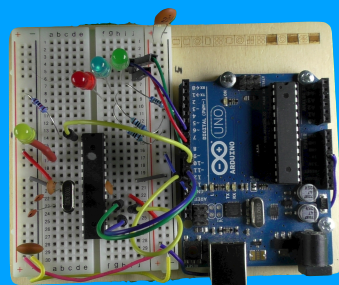
Der Ausgang wird auf 5V (beziehungsweise die Betriebsspannung des Prozessors) gesetzt. Es muss beachtet werden, dass beim Arduino UNO der maximale Strom nur 20 mA betragen darf! Bei anderen Prozessoren kann dieser Wert abweichen.

0 V ausgeben: `digitalWrite(pinNummer, LOW);`

Der Ausgang wird auf 0 V, beziehungsweise Ground (GND) gesetzt. Es muss beachtet werden, dass beim Arduino UNO der maximale Strom nur 20 mA betragen darf! Bei anderen Prozessoren kann dieser Wert abweichen.

Status abfragen: `digitalRead(pinNummer);`

Es kann abgefragt werden, ob der Pin zuletzt auf HIGH oder LOW gesetzt wurde. ACHTUNG: Diese Abfrage funktioniert bei den meisten Arduino - Varianten. Es gibt aber Prozessoren, die hier keinen korrekten Wert zurückgeben.



# Der Arduino spricht zu uns

Das ist nicht ganz richtig. Er kann selbstverständlich nicht sprechen. Aber er kann uns etwas schreiben, das wir im seriellen Monitor der Arduino IDE lesen können.

Vorbereitung: `Serial.begin(9600);`

Die serielle Schnittstelle wird vorbereitet. Dabei müssen wir die Geschwindigkeit angeben.

Etwas schreiben: `Serial.print("Hallo"); Serial.print(5); Serial.print(pinNummer);`

Diese Befehle würden den text, die Zahl und den Inhalt von pinNummer direkt hintereinander ausgeben. Wenn pinNummer 13 wäre, würde Hallo513 ausgegeben.

Auf einer neuen Zeile schreiben:

`Serial.println("Hallo"); Serial.println(5); Serial.println(pinNummer);`

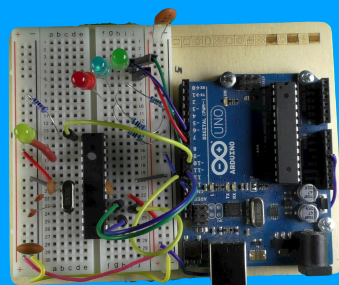
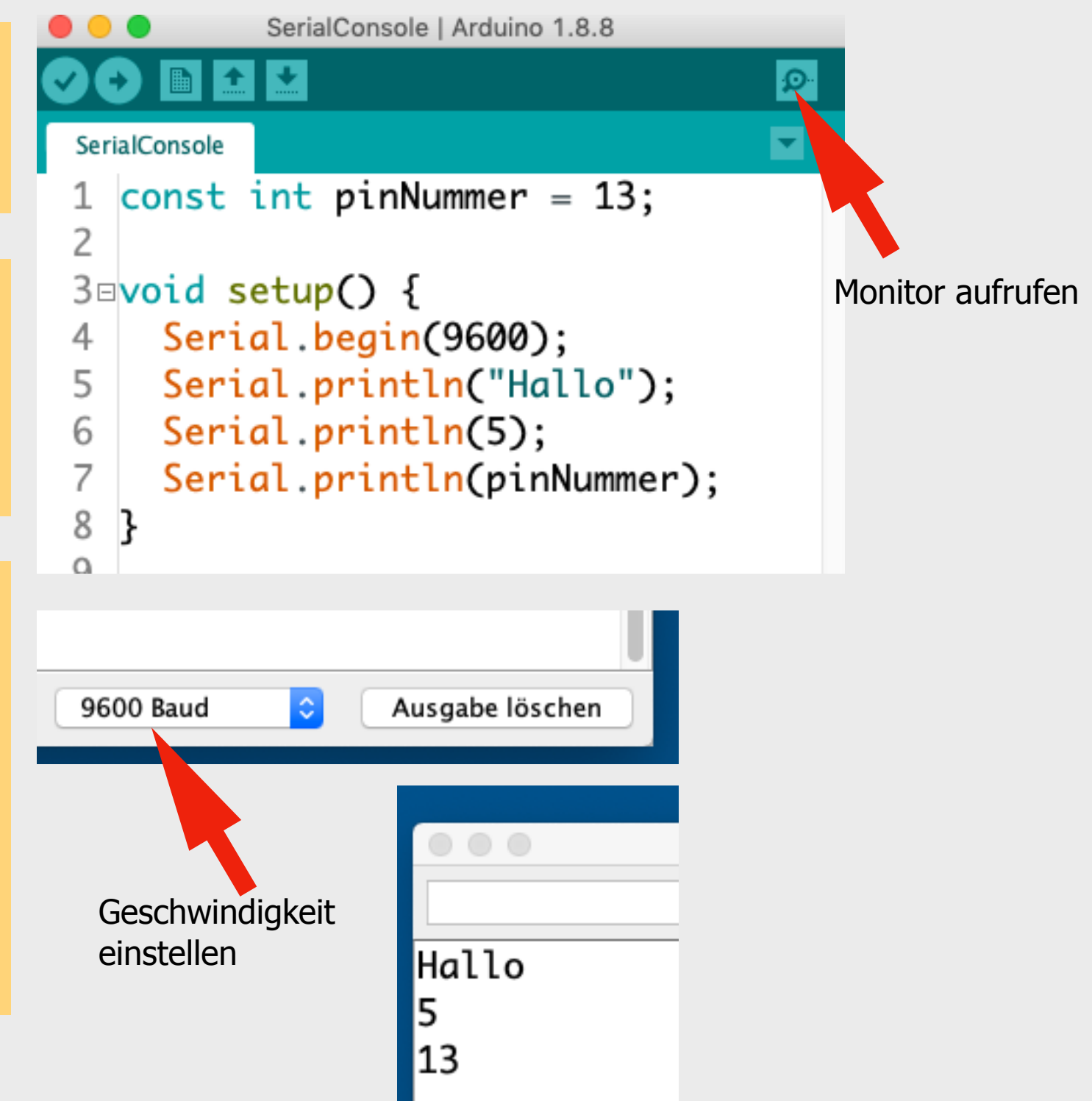
Das gäbe dann:

Hallo

5

13

Serial kann noch viel mehr. Damit werden wir uns aber später befassen.



Der schnelle einfache Einstieg in Arduino & Co.  
Digitale Ausgabe

