# Physical computing mit Scratch und scratchClient – Anfänger

Raspberry Pi, Arduino, Scratch und scratchClient steuern Servos, LEDs und mehr

Hans de Jong & Gerhard Hepp Pi And More 10 1/2 Stuttgart – 24 February 2018

### Part 1: Introduktion







- Wenn die Präsentation durcheinander ist ...
  - Sie benötigen die Calibri Schriftart auf Ihrem Raspberry Pi (Standard unter Windows, aber nicht auf Raspberry Pi).
  - Sehen Sie sich daher die .pdf-Version an, in der die zu installierende Schriftart nicht benötigt wird.

# Workshop-Organisation



- Begrüßung und Einführung Präsentation (5 min).
- Danach wird jeder in seinem eigenen Tempo arbeiten.
- Wählen Sie mit Ihrem "Arbeitsplatzpartner", welche Themen Sie bearbeiten möchten.
  - Es gibt mehr Material, als in 110
     Minuten abgedeckt werden kann.
- Sprache: Deutsch und Englisch
- Am Ende kopieren Sie das von Ihnen erstellte Material auf Ihren USB-Stick (wenn Sie möchten)
- Aufbrechen und aufräumen (5 min)

- Die wichtigsten Schritte:
  - 1. Holen Sie sich eine funktionierende Hardware- und scratchClient- Konfiguration mit einem scratchClient-Konfigurationstool.
  - 2. Stellen Sie Komponenten auf die Platine und testen Sie Ihr Setup.
  - 3. Schreibe etwas Code in Scratch
  - 4. Fügen Sie weitere Hardware hinzu und aktualisieren Sie die Konfigurationsdatei.
  - 5. Und wiederhole das.

#### Ziele



- Am Ende des heutigen Tages sollten Sie in der Lage sein, einige dieser Dinge zu tun:
  - Reproduzieren Sie das Setup zu Hause (vorausgesetzt, Sie haben die Hardware)
    - Sie können alle Dateien über https://github.com/hansdejongehv/scratchClient-Tutorials abrufen
    - Oder gehen Sie zu www.github.com und suchen Sie nach scratchClient
  - Verstehen (abhängig davon, wie weit Sie kommen und wie tief Sie tauchen)
    - Digitalausgang (z. B. Beleuchtung einer LED)
    - Digitaler Eingang (z. B. Tastenerkennung)
    - Analogeingang (z. B. von einem Potentiometer)
    - Pulsbreitenmodulation (PWM)
      - Zum Dimmen von LEDs
      - Zur Steuerung von Servos
      - Um einen Summer zu hören
  - Verstehen Sie, wofür alle Widerstände sind
  - In der Lage sein, scratchClient zu konfigurieren und auszuführen
  - Programm Scratch zur Steuerung der physikalischen Ein- und Ausgabe
  - Überwachung der Ein- und Ausgänge

#### Habe Spaß!

#### Keine Ziele



- Es ist kein Ziel, ein komplettes nützliches Spiel oder ein anderes Programm zu erstellen.
- Sie können dies mit Ihrer eigenen Kreativität zu Hause tun, da Sie jetzt wissen, wie Sie mithilfe von scratchClient mehrere Hardwareteile von Scratch aus steuern können.

# Beispiel für was mit Scratch und scratchClient erstellt werden kann



https://www.youtube.com/watch?v=Qo1gnXNzhqE

#### Versionen von Scratch



- scratchClient kann damit arbeiten
  - Scratch 1.4
  - Der (für Raspberry Pi) neue Scratch 2
- Dieser Workshop ist für Scratch 2 geschrieben
  - Scratch 2 auf Raspberry Pi hat einige Bugs, aber wir werden es umgehen.
  - Wenn Sie sehen möchten, wie man es für Scratch 1.4 macht, siehe das Ende der Präsentation in Anhang A.

## Nur ein paar Regeln heute

scratchClient

- Setzen Sie immer einen Widerstand in Reihe mit den Komponenten, wenn angegeben
  - Wenn Sie denken, dass es keine Notwendigkeit gibt, dann sagen Sie es uns bitte und wir werden es erklären was der Grund ist.
- Beim Ändern der Verdrahtung
  - 1. Lösen Sie das USB-Kabel vom Arduino Nano
  - 2. Schalten Sie die 9V-Stromversorgung aus

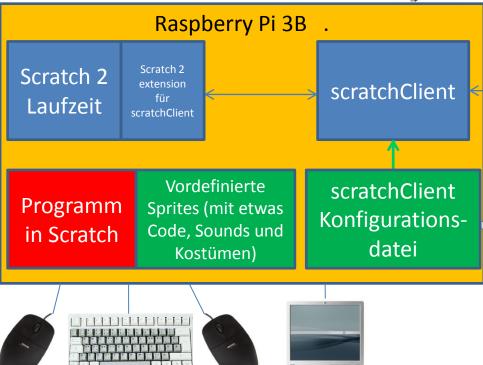


- 3. Überprüfen Sie, überprüfen Sie nochmals und überprüfen Sie erneut, ob die Verkabelung korrekt ist. Sie können Bauteile bei falscher Verdrahtung sprengen!
- 4. Stellen Sie sicher, dass Sie **beide** (4-Augen-Prinzip) davon überzeugt sind, dass die Verkabelung in Ordnung ist bevor Sie das Gerät wieder anschließen und wieder einschalten.
- 5. Nach dem Ändern einer Konfigurationsdatei: starte scratchClient neu
- Wenn etwas kaputt geht oder beschädigt wird: wir haben etwas Ersatzmaterial
  - Bitte legen Sie nichts zurück in die Schachtel.



# Die Einrichtung §



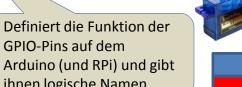






scratchClient Skizze

**Arduino Nano** 



GPIO-Pins auf dem

ihnen logische Namen.



Standardeinrichtung

Von den Studenten programmiert

Vorbereitet vom Lehrer oder Experten





- Der Workshop ist für alle da, vom Anfänger bis zum Experten.
- Nicht genug Zeit, um alles zu tun, also wählen Sie, was Sie tun möchten.
- Gelbe Folien haben Hintergrundinformationen und Sie können sie überspringen, wenn Sie möchten oder später schauen
- Empfohlen
  - Alle: Lernen Sie, scratchClient zu konfigurieren, indem Sie LEDs (digital out) und Button (digital in)
  - Alles: Probieren Sie es in Scratch aus
  - Danach: Wählen Sie weitere Themen aus separaten Dateien
    - Zwischenstufe (Material nur auf Englisch)
    - Fortgeschrittenes Niveau (Material nur auf Englisch)
    - Expertenlevel (Material nur auf Englisch)

# Teil 2: Kennenlernen der Komponenten

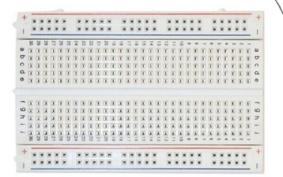
#### Steckbrett

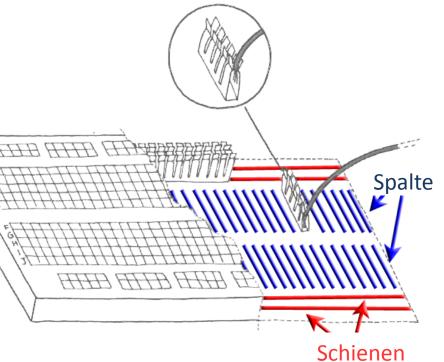


 Wird verwendet, um schnell elektronische Schaltkreise aufzubauen

 Beachten Sie die 2 Schienen für + (VCC) und - (GND)

 Beachten Sie bei jeder Spalte (vertikal) die 2 Balken mit je 5 miteinander verbundenen Löchern.





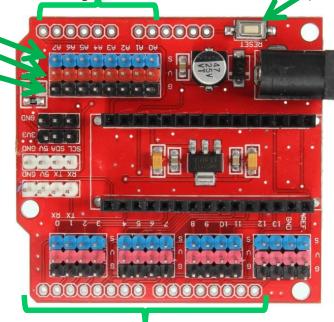
# Mit Blick auf die Arduino Nano

scratchClient

Pro GPIO Signal 3
Steckerleisten:
S (blau = Signal)
V (rot = VCC = +)
G (schwarz = GND = -)
(Sehr praktisch für z.B.
Servos anschließen)

Adapterplatine

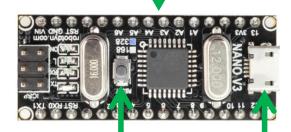
Analoge Porten (die meisten können auch als digitale Porten verwendet werden)



Reset Taster

9V Stromansluß

Arduino Nano mit 328P Prozessor

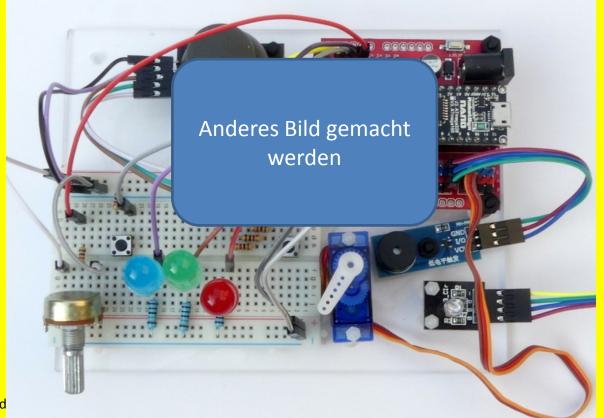


Micro USB Port

**Reset Taster** 

#### Wie das letzte Brett aussieht ...

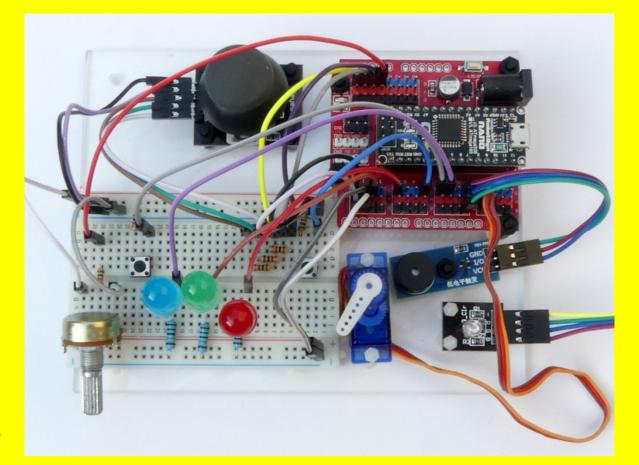




... am Ende der Einsteigerkurs

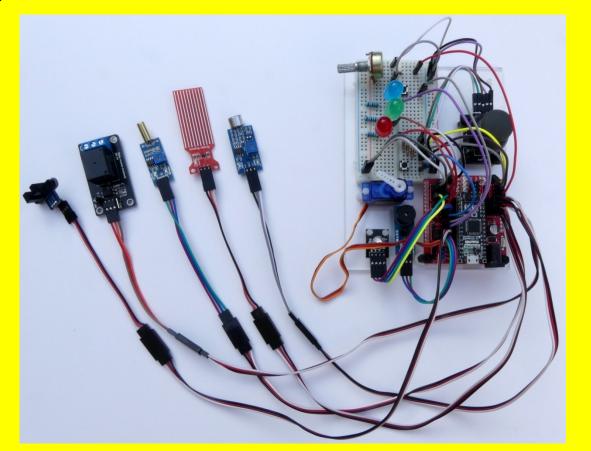
#### ... am Ende der Zwischenstufe ...





# ... und am Ende des fortgeschrittenen Niveaus





# Teil 3: Laden der Skizze in den Arduino (wahlweise)

# Vorbereitung für die Programmierung des Arduino Nano

- Der Arduino muss ein Programm ausführen (in Arduino-Begriffen: Skizze), damit er mit dem Raspberry Pi kommunizieren und die Nachrichten von scratchClient verstehen kann.
- Wir müssen damit anfangen.
  - Dies ist jedoch bereits geschehen, sodass Sie die nächste Folie überspringen können, es sei denn, Sie möchten es selbst ausprobieren.

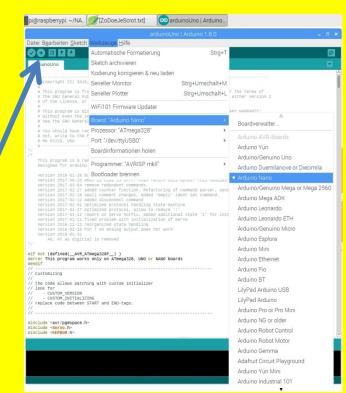
#### Hochladen von scratchClient auf den Arduino



 Navigieren Sie zur ScratchClient-Skizze für Arduino Uno in / home / pi / scratchClient / arduino / arduinoUno



- Doppelklicken Sie, um die Arduino IDE zu öffnen
- Klicken Sie auf Werkzeuge und stellen Sie sicher, dass diese festgelegt sind:
  - Board: Arduino Nano
  - Prozessor: Atmega328
  - Port: Der Port, an dem der Arduino Nano angeschlossen ist (normalerweise /dev/ttyUSB0)
- Klicken Sie auf die Schaltfläche Hochladen.
- Warten Sie, bis der Upload abgeschlossen ist (ohne Fehler).



# Teil 4: Definieren der Konfiguration

Gib die Anschlüsse Namen und definiere den Zweck der Anschlüsse



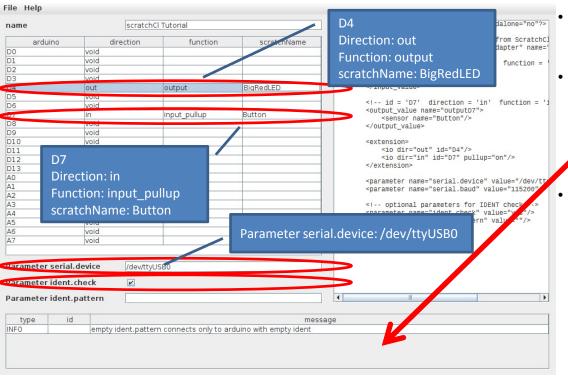


- Wir haben eine leere Konfigurationsdatei auf dem Desktop abgelegt: ScratchClient Tutorial.scl
- Klicken Sie mit der rechten Maustaste auf die Datei und wählen Sie scratchClient Edit



#### Definieren Sie die erste Konfigurationsdatei





- **Doppel**klicken Sie auf eine Zelle, um ein Dropdown-Menü zu erhalten
  - Zuerst für die Richtung, dann für die Funktion.
- Achte darauf, dass Sie alle Anslüsse Namen geben, wenn Sie etwas anderes wählen als void in direction.
  - Stellen Sie also sicher, dass Sie nicht speichern, wenn Sie immer noch rote Ränder um Zellen haben. scratchClient kann mit einer solchen Konfigurationsdatei nicht starten.
- Das Tool überprüft falsche Konfigurationen. Beispiele:
  - Die Anslüsse 0, 1 und 13 können überhaupt nicht verwendet werden
  - Analog ist nur auf A0 bis A7 verfügbar
  - Die Anslüsse A6 und A7 können nur für analoge Eingänge verwendet werden
  - Anslüsse 9 und 10 können nicht für PWM verwendet werden, wenn ein Ansluß für Servo konfiguriert ist (siehe später)

# Speichern Sie die Konfigurationsdatei

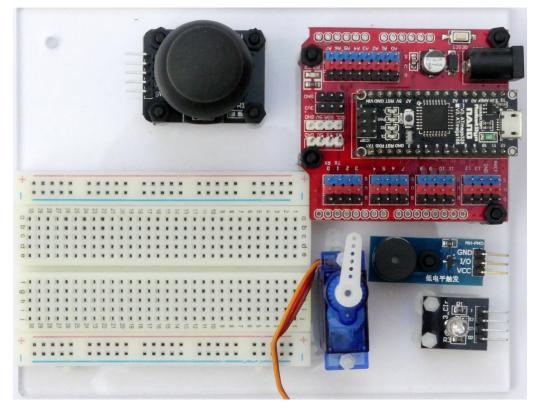


- Speichern Sie die Datei (Stgr-S oder File → Save).
- Warnung: Schließen der Datei ohne speichern verlieren Sie alle Änderungen.
- Lassen Sie das Werkzeug für die nächsten Übungen geöffnet.

# Teil 5: Verdrahten Sie die Platine und starten Sie das erste Setup





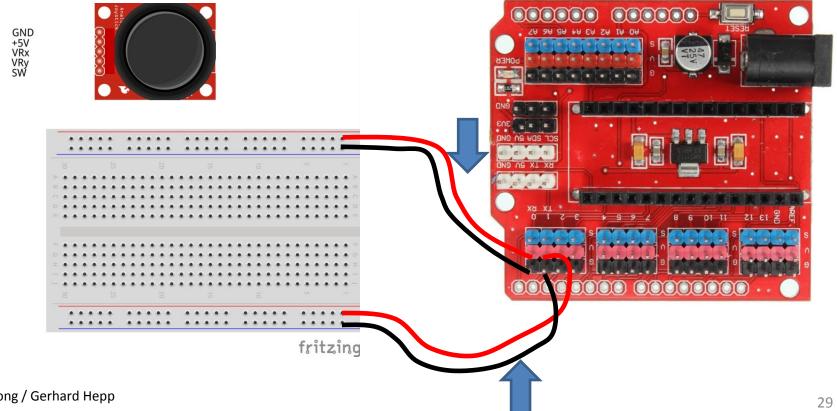


# Verwenden Sie kurze Drähte und verwenden Sie die angegebenen Löcher

- Es gibt einige kurze Drähte (10 cm) und einige längere (15 cm)
- Verwenden Sie den kürzesten, den Sie können
  - Holen Sie sich ein weniger chaotisches Setup
  - Sie könnten sonst später keine langen Drähte mehr haben
- Ignorieren Sie die Drahtfarben.
- Sie können jedoch prinzipiell an verschiedenen Stellen auf dem Steckbrett aufbauen ...
  - ... bitte benutzen Sie die angegebenen Spalten, um zu vermeiden, dass im späteren Teil des Workshops der Platz auf dem Steckbrett knapp wird.

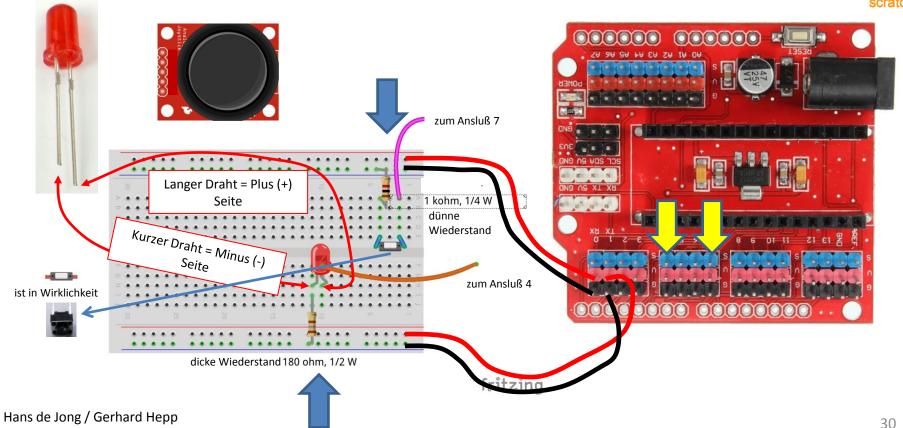
#### Schließen Sie die Stromkabel an





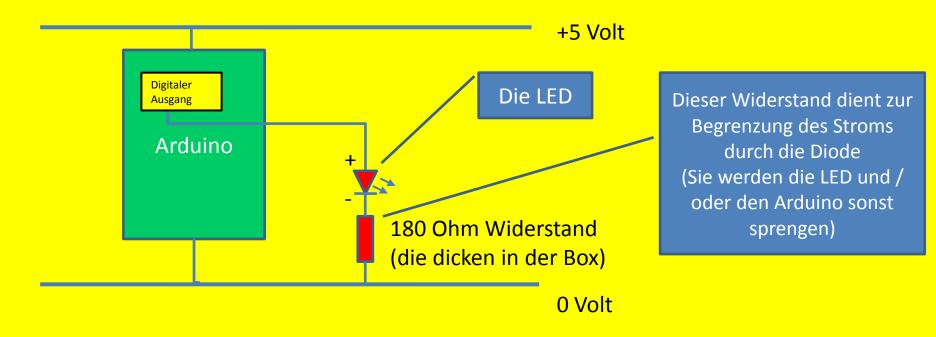
#### Setzen Sie die rote LED und den Knopf ein





# Warum setzt man einen Widerstand in Reihe mit der LED?





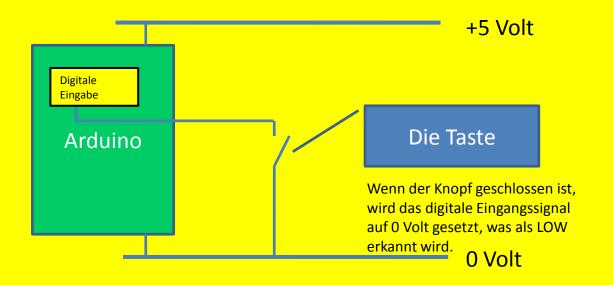
# Was benötigt Arduino an einem digitalen Eingangspin?

scratchClient

- Ein digitaler Eingang Pin von Arduino muss entweder erhalten
  - O-Volt-Eingang (tatsächlich gilt 0 bis 1,5 Volt als LOW-Eingangssignal)
  - 5-Volt-Eingang (tatsächlich werden 2,5 bis 5 Volt als HIGH-Eingangssignal betrachtet)
- Wenn es etwas zwischen 1,5 Volt und 2,5 Volt bekommt, wird die Interpretation nicht stabil sein (könnte LOW oder HIGH sein).
- Wenn kein Signal empfangen wird, ist die Interpretation des Signals nicht stabil (könnte LOW oder HIGH sein).



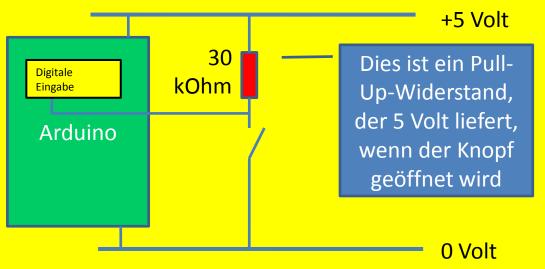




Aber was wird der Arduino feststellen, wenn der Knopf geöffnet ist?

## Einen HOHEN Wert erzeugen



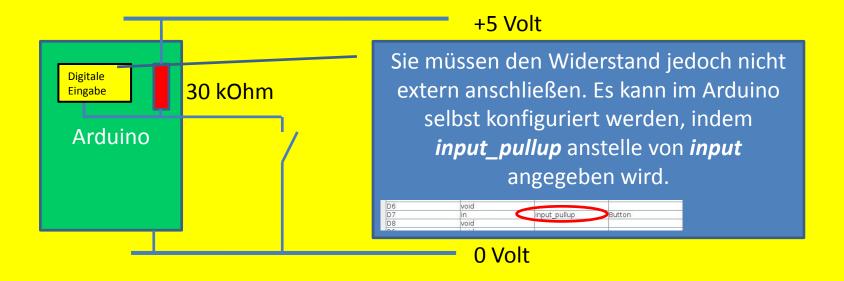


Wenn die Taste gedrückt wird, verbindet es den Eingang des Arduino mit 0 Volt. Es wird ein kleiner Strom durch den Widerstand fließen, aber der Eingang wird 0 Volt sein.

Wenn der Schalter geöffnet ist, zieht der Widerstand den Eingang auf 5 Volt hoch. Ein sehr kleiner Strom fließt durch den Widerstand, ist aber klein genug, dass die Spannung am digitalen Eingang sehr nahe bei 5 Volt = HOCH liegt.

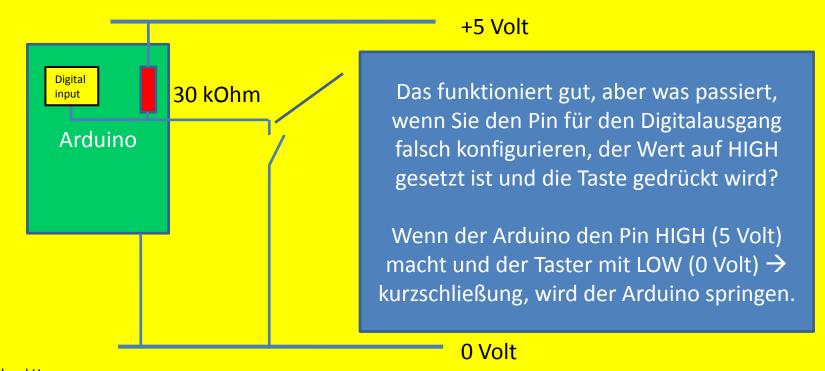
# Erzeugen eines HIGH-Wertes ohne externen Widerstand





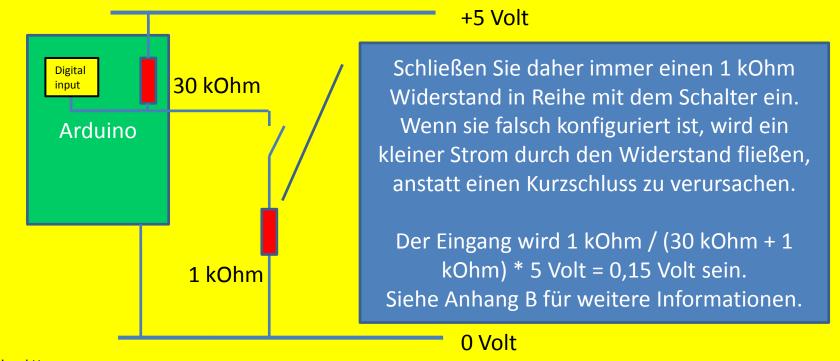
# Gefahr bei Fehlkonfiguration





#### Widerstand in Reihe, um Schäden bei Fehlkonfiguration zu vermeiden









 Bitte überprüfen Sie jetzt beide von Ihnen, dass die Verkabelung korrekt ist.





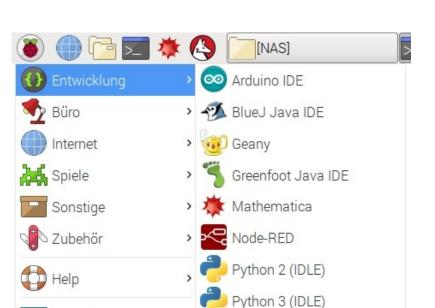
- Verbinden Sie den 9-Volt-Stecker mit der Platine und schalten Sie das Gerät ein
- Verbinden Sie den USB-Anschluss mit dem Arduino
- Doppelklicken Sie scratchClient Tutorial.scl auf dem Desktop, um scratchClient mit der gerade aktualisierten Konfigurationsdatei zu starten

• Es wird auch ein Browserfenster geöffnet, in dem Sie Variablen sehen können

- Später in der Präsentation erläutert
- Im Terminalfenster sehen Sie Beschwerden, dass scratchClient keine Verbindung zu Scratch hat

• Was logisch ist, weil Scratch noch nicht gestartet wurde.

#### Start Scratch 2



Scratch

π))) Sonic Pi

Scratch 2



Hans de Jong / Gerhard Hepp

instellungen

Shutdown...

Run...

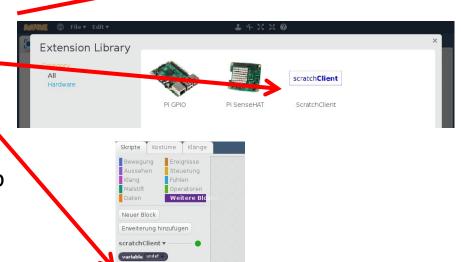
### Starten Sie Scratch 2 und erhalten SiescratchClient Blöcke

Skripte Kostüme Klänge

Bewegung Ereignisse
Aussehen Steuerung
Klang Fühlen
Malstift Operatoren
Daten Weitere Blo

Erweiterung hinzufügen

- Klicken Sie auf Weitere Blöcke
- Klicken Sie auf Erweiterung hinzufügen
- Wählen Sie scratchClient
- Jetzt werden die zusätzlichen Blöcke von scratchClient eingeschlossen.
- Warten Sie 10 Sekunden und schauen Sie, ob die blaue LED am Arduino Nano langsam statt schnell blinkt. Ist dies der Fall, ist die Verbindung hergestellt.



#### Erstellen Sie das Scratch-Programm



- Erstellen Sie dieses Programm in Scratch und probieren Sie es aus (klicken Sie auf die Blöcke oder klicken Sie auf die grüne Flagge über dem Animationsfenster).
- Digitaleingang (Taste):
  - 0 = gedrückt
  - 1 = nicht gedrückt
- Digitalausgang (LED)
  - 0 = aus
  - 1 = ein
- Analysieren Sie, wie das Programm funktioniert.

```
Wenn angeklickt
wiederhole fortlaufend
send name BigRedLED value variable Button
```

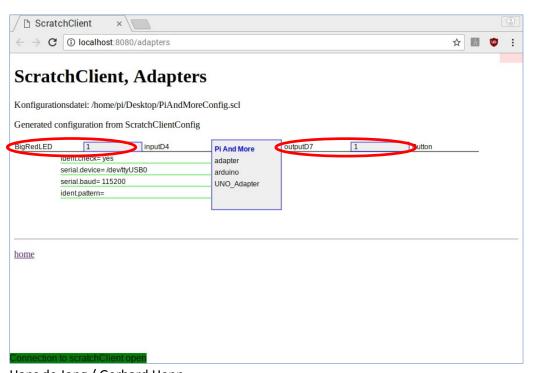
### Funktioniert es? (siehe nächste Folie für Hilfe)



- Wenn die LED am Arduino langsam blinkt, nur dann ist die Konfiguration heruntergeladen und nur dann funktioniert scratchClient.
  - Es kann 10 Sekunden dauern, bis sowohl scratchClient als auch die ScratchClient-Erweiterungen in Scratch 2 geladen sind, bevor dies geschieht.

## Sie können die Werte überwachen, die ausgetauscht werden





- Zusammen mit dem scratchClient wird der Browser mit der URL localhost: 8080 / adapter geöffnet
- Beachten Sie die Eingabe- und Ausgabeanweisungen:
  - Die Ausgabe eines Adapters ist eine Eingabe für Scratch
  - Die Ausgabe von Scratch ist eine Eingabe für die Adapter.
  - Daher scheinen die Eingabe und Ausgabe der Namen umgekehrt zu sein, was in der Konfigurationsdatei enthalten ist.
  - Daher beziehen Sie sich am besten auf die Variablennamen.
- Sie werden sehen, dass Werte nur angezeigt werden, nachdem sie sich geändert haben (andernfalls wird ein Fragezeichen (?) Angezeigt).
- Nicht für den Moment, aber Sie können auf das Feld klicken und einen Wert eingeben, der dann in die entsprechende Richtung gesendet wird.



scratchClient

- Nehmen Sie eine Änderung am Programm vor, so dass die LED beim Drücken der Taste leuchtet.
- Es gibt (mindestens) zwei
   Möglichkeiten
- Sie benötigen die hier abgebildeten Blöcke



Blöcke für Methode 1



Blöcke für Methode 2

Hans de Jong / Gerhard Hepp

BIOCKE TUT IVIETTIOUE 2

45

# Was ist, wenn es nicht funktioniert?



- Überprüfen Sie, ob mehrere Scratch-Instanzen geöffnet sind
  - scratchClient kann immer nur mit einem Scratch arbeiten (unabhängig davon, ob es sich um Scratch 1.4 oder Scratch 2 handelt).
- Überprüfen Sie, ob die blaue LED am Arduino Nano langsam blinkt.
- Manchmal, besonders nach dem Neustart, wenn alles in Ordnung scheint, kann das Trennen und erneute Verbinden des Arduino Nano helfen.
- Versuchen Sie, die Variablen zu überwachen, siehe vorher.





- Speichern Sie Ihr Scratch-Programm regelmäßig. Andernfalls wird es beim Ausschalten verloren gehen.
  - Ein Stromausfall kann leicht passieren, da Sie Kabel ziehen und den Stromanschluss des Raspberry Pi beeinträchtigen können.
- Hier kommen einige aktuelle Scratch 2 Anomalien auf RPi
  - Sie sollten keine Leerzeichen in Dateinamen verwenden (wenn Sie dies tun, werden sie entfernt)
  - Scratch 2 vergisst den Ordner, in dem die Datei von (Desktop) geöffnet wurde. Speichern Sie es einfach in / home / pi, wie Scratch vorschlägt
  - Scratch 2 speichert es mit einer Erweiterung .sbx unabhängig von der Angabe.
- Also einmal nach dem Einschalten
  - Doppelklicken Sie auf dem Desktop auf CleanUpScratchFiles.sh
  - Sie können das Fenster minimieren, aber lassen Sie es für immer laufen
  - Dies überwacht permanent den Ordner /home/pi, und
    - Verschiebt alle Dateien mit .sbx auf den Desktop als .sb2-Dateien
    - Alle Leerzeichen in Dateinamen werden entfernt
    - Wenn sich bereits eine solche Datei auf dem Desktop befindet, wird sie in den Ordner ScratchProgramsBackup auf dem Desktop verschoben



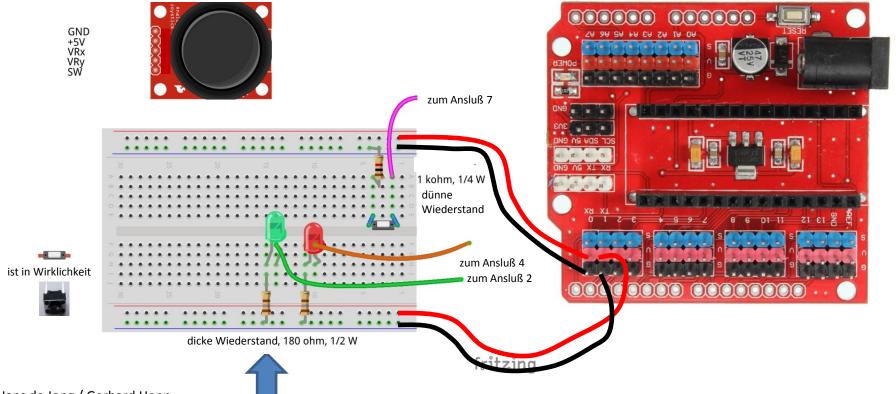
CleanUpScrat chFiles.sh



# Teil 6: Hinzufügen der großen grünen LED

#### Hinzufügen der grünen LED









- Überprüfen Sie die korrekte Verkabelung
- Schalten Sie die 9V-Stromversorgung ein
- Schließen Sie das USB-Kabel erneut an

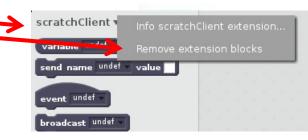
## Aktualisieren Sie die Konfigurationsdatei und starten Sie scratchClient neu



- Benutze das Config-Tool (welches noch geöffnet sein sollte)
- Definieren Sie einen Ausgang (direction: out, function: output) an Ansluß 2 und nennen Sie ihn BigGreenLED
- Speichern Sie die Konfigurationsdatei (und lassen Sie das Werkzeug geöffnet)
- Doppelklicke scratchClient Tutorial.scl erneut auf dem Desktop
- Dies wird die vorherige Instanz von scratchClient stoppen und mit der aktualisierten Konfiguration neu starten.



- Schmeissen Sie die scratchClient Erweiterung weg
- Füge die Erweiterung nochmals hinzu
  - Sie werden sehen in der Weitere Blöcke Abschnitt daß Sie wählen können um Werten nach BigGreenLED zu senden



#### Scratch aktualisieren



- Fügen Sie Code hinzu, der das tut:
  - Drücken Sie die Taste: Die LED leuchtet
  - Noch einmal drücken: LED

erlischt

Sie benötigen diese
 Programmelemente.

```
warte bis variable Button = 0

wiederhole fortlaufend

warte bis variable Button = 0

send name BigGreenLED value 0 warte bis variable Button = 1

send name BigGreenLED value 1 warte bis variable Button = 1
```

#### Dies ist das Ende des Anfängerniveau

- Sie haben Digital Out (LED) und Digital In (eine Taste) funktionieren
- Sie wissen, wie man mit scratchClient und Scratch 2 arbeitet
- Sie wissen, wie Sie scratchClient konfigurieren
- Wenn Sie wollen und Zeit haben
  - Sehen Sie sich einige der gelben Folien an, wenn Sie sie früher übersprungen haben
  - Machen Sie weiter mit einer der nächsten Ebenen
    - Diese befinden sich in verschiedenen Dateien
- Andernfalls,
  - Dateien auf Ihren USB-Stick kopieren und aufräumen, siehe weiter

#### Übersicht der nächsten Niveaus



- Zwischenniveau (Material nur auf Englisch)
  - Analogeingang: Potentiometer
  - Pulsweitenmodulation (wegen Abwesentheit eines analogen Ausgangs)
  - Dimmen einer LED
  - Ein Servo steuern
  - Einen Summer steuern
- Fortgeschrittenes Niveau (Material nur auf Englisch)
- Expertenniveau (Material nur auf Englisch)

54

# Teil 10: Nimm Ihre Arbeit mit nach Hause

#### Wollen Sie Ihre Arbeit nach Hause mitnehmen?



- Wenn Sie Ihren eigenen USB-Stick mitgebracht haben, schließen Sie ihn an und kopieren Sie die Datei scratchClient Tutorial.scl auf den Desktop sowie alle auf dem Desktop erstellten .sb2-Dateien
- Der Rest des Materials, können Sie von www.github.com herunterladen, suchen Sie nach scratchClient
- Nehmen Sie den Flyer mit, um sich zu merken, wo das Material auf GitHub zu finden ist.

# Teil 11: Zusammenfassung & Mitnehmen

#### Takeaways des Anfänger-Workshops



- Mit scratchClient definieren Sie:
  - Funktion jedes Anschluß
  - Symbolischer Name für jeden konfigurierten Anschluß
- scratchClientConfig ist das Tool zum Einrichten der Konfiguration
- Starten Sie scratchClient neu, nachdem Sie die Konfiguration geändert haben
- Setzen Sie einen Widerstand in Reihe mit LEDs
- Setzen Sie einen Widerstand in Reihe mit Schaltern
- Konfigurieren Sie einen Pull-up-Widerstand, wenn das Eingangssignal zwischen 0 Volt und offen liegt und nicht zwischen 0 Volt und 3 bis 5 Volt.
- In Scratch 2 verwenden Sie den Erweiterungsblock für scratchClient, um Blöcke zu erhalten, die Sie für die Interaktion mit scratchClient verwenden können
- Sie können den Wert aller Anschlüße vom Browser aus überwachen
- scratchClient kann viel mehr ...

- Funktionen, die eine Anschluß auf Arduino haben kann:
  - Digitaleingang
  - Digitalausgang
  - Analoger Eingang \*
  - Kein analoger Ausgang
  - Pulsweitenmodulation als Alternative \*
    - Zum Modulieren der Helligkeit einer LED
    - Zum Steuern eines Servos
    - Zum Steuern eines Summers
  - Es gibt einige mehr, siehe den fortgeschrittenen Workshop
  - Sie können Pull-up-Widerstände an Digitaleingang konfigurieren

<sup>\*</sup> Siehe Zwischenniveau

#### Teil 12: Aufräumen / Abbau





- Trennen Sie die Platine von USB und schalten Sie das Gerät aus
- Bitte entfernen Sie alle Komponenten und Kabel vom **Steckbrett**
- Entfernen Sie alle Drähte von der Arduino-Platine.
- Lass die Drähte auf der 3-farbigen LED (Sie haben das nicht benutzt)
- Lassen Sie die Kabel am Summer
- Wenn etwas kaputt ist, bitte
  - Werfen Sie es weg oder reichen Sie es ein (wenn es unklar ist)
  - Schreiben Sie eine Notiz im Schachtel, dass es fehlt
  - Legen Sie nichts was kaputt ist zurück in die Schachtel
- Rasberry Pi absließen
- Lassen Sie uns wissen, was Sie über diesen Workshop gedacht haben, jetzt mündlich oder später per E-Mail
  - hans.piam@hanselma.nl
  - heppg@web.de

## Anhang A Bei Verwendung von Scratch 1.4

#### Die Einrichtung







Programm

in Scratch

Raspberry Pi 2B or 3B

Remote Sensor Protocol über Sockel 42001 Globale Variablen Sendungen

Vordefinierte Sprites (mit etwas Code, Sounds und Kostümen)

scratchClient

scratchClient Konfigurationsdatei

> Definiert die Funktion der GPIO-Pins auf dem Arduino (und RPi) und gibt ihnen logische Namen.

scratchClient Skizze

**Arduino Nano** 





Von den Studenten programmiert

Vorbereitet vom Lehrer oder **Experten** 

Hans de Jong / Gerhard Hepp

62

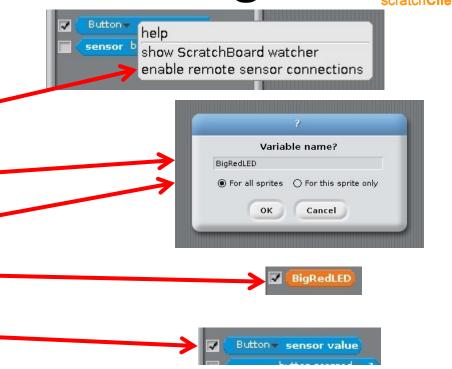




 Nichts neues, Sie können die gleiche Konfigurationsdatei für Scratch 1.4 wie für Scratch 2 verwenden.

Erstellen Sie das Scratch-Programm

- Start Scratch → Entwicklung → Scratch
- Unterstützung für Netzwerksensoren einschalten (Rechtsklick auf Wert von Sensor)
- Erstellen Sie die Variable BigRedLED, die allen Sprites zur Verfügung steht
- Machen Sie die Variable sichtbar (kreuzen Sie das Kästchen an)
- Machen Sie den Button-Sensor sichtbar
- Speichern Sie die Datei auf dem Desktop.







- Stellen Sie dieses Programm auf "Scratch", wodurch die rote LED leuchtet, wenn die Taste losgelassen wird.
- Testen Sie, ob es funktioniert.

```
when clicked

forever

set BigRedLED to Button sensor value
```

#### Anhang B Grundlegende Elektronik





# Ende des Anfängerworkshops