

Session 10 – Hub & Breadboard - Würfel

Heute bauen wir mit dem TI-Innovator Hub und dem Breadboard einen elektronischen Würfel.

Wie auch schon bei der Ampelanlage lernen wir, wie man LEDs an den Hub anschließt und mit ihm zum Leuchten bringt. Wir lernen, Unterprogramme in einem Hauptprogramm aufzurufen, um weniger Code-Zeilen und mehr Übersichtlichkeit in unserem Programm zu haben. Dafür müssen wir üben, wie man Programme auf dem CX kopiert und umbenennt. Schaut Euch mal den Abschnitt FAQs an! Und wir lernen einen neuen Befehl "Zufall" (Random) kennen.

Was brauchen wir?

1. Einen CX Taschenrechner
2. Die MINT Box, die den TI-Innovator Hub und alle Bauteile für das Breadboard enthält
3. Euch, um den elektronischen Würfel aufzubauen und zu programmieren!

FAQs:

- Frage: Wie kopiere ich Dateien und Programme und benenne sie um?

Antwort:

- Im Programm <doc> Taste drücken
 - <1> drücken oder auf <Datei> klicken
 - <5> drücken oder auf <Speichern unter> klicken
 - Neuen Namen in das Dateinamefeld eingeben
 - Einmal <tab> und dann <enter> drücken oder auf <Speichern> klicken. Oben im Display des CX sollte der neue Dateiname stehen.
 - Dann muss noch das Programmfenster neu benannt werden:
 - Im Programmfenster <menu>, dann <1> drücken oder auf <Aktionen> klicken
 - <6> drücken oder auf <Umbenennen> klicken
 - Im markierten Feld den neuen Programmnamen eingeben
 - <tab> und <enter> drücken oder auf <ok> klicken.
- Frage: Wie kopiere ich Code Zeilen im Programmfenster?

Antwort:

- Mit dem Cursor in die zu kopierende Code-Zeile gehen
 - Die <shift> Taste gedrückt halten und die linke oder rechte Cursorstaste des Touchpads wiederholt drücken. Wir sehen, dass die Code Zeile entsprechend markiert wird.
 - Wenn die Code-Zeile komplett markiert (blau) ist, <ctrl><c> drücken, um sie zu kopieren oder <ctrl><x> drücken, um sie auszuschneiden.
 - Mit dem Cursor an die Stelle gehen, wo die Zeile eingefügt werden soll.
 - <ctrl><v> drücken.
 - Man kann natürlich mehrere Codezeilen auf diese Weise selektieren, kopieren, löschen oder einfügen.
- Frage: Wo finde ich die deutschen Umlaute (ö,ä,ü) auf dem CX Rechner?

Antwort:

- <ctrl><Buch> drücken (das ist die Taste unterhalb der Taste) und entsprechendes Symbol auswählen
- Frage: Wo finde ich die ≠,<,> Zeichen?

Antwort:

- <ctrl><=> drücken und entsprechendes Symbol selektieren.

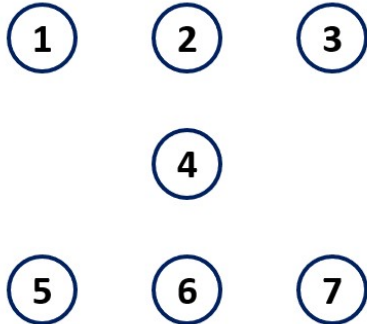
Welche neuen Kommandos verwenden wir?

1. Die LEDs mit dem Hub verbinden [<menu>-8-8-4]:
Send "CONNECT LED x BB x"
2. Die LEDs ein- oder ausschalten [<menu>-8-1-7]:
Send "SET LED x ON/OFF"
3. Die Funktion, um Zufallszahlen zu erzeugen [Buch-Taste unterhalb der del-Taste, auf Reiter 1 klicken, r eingeben]:
randInt(u,o)
Gibt eine ganzzahlige Zufallszahl innerhalb der durch u und o festgelegten Unter- und Obergrenzen aus. Für den Würfel programmieren wir die Funktion so: randInt(1,6)

Aufgaben:

1. Eine Skizze des Würfels anfertigen und beschriften

a. Fertige eine Skizze der Würfelaugen an, so wie hier im Bild:

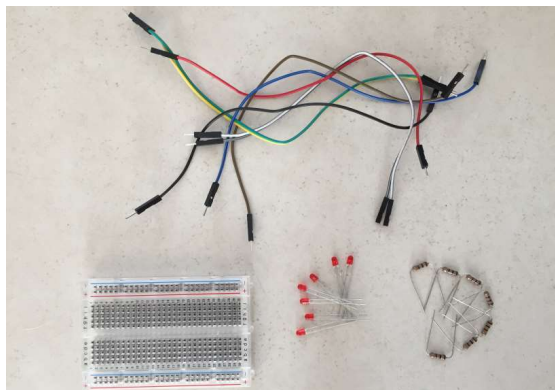


Diese Skizze hilft dir später, die LEDs mit den richtigen Digitalausgängen des Innovator Hubs zu verbinden und bei der Programmierung der verschiedenen Zahlen des Würfels.

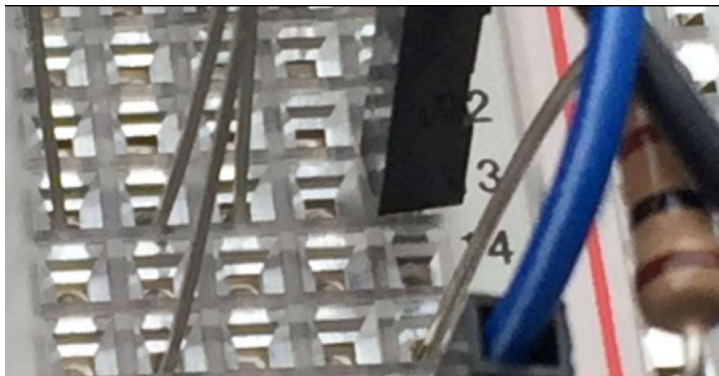
2. Sieben LEDs mit Vorwiderständen auf dem Breadboard befestigen

a. Diese Teile brauchst Du:

- i. Sieben rote LEDs
- ii. Sieben bunte Anschlussleitungen für die LEDs
- iii. Sieben 100 Ohm Vorwiderstände für die LEDs
- iv. Vier schwarze Anschlussleitungen für die Masse-(Minus-) Anschlüsse, mit denen die Kathoden der LEDs verbunden sind
- v. Das Breadboard



- b. Jede LED hat zwei Anschlussdrähte. Der eine Draht ist ein bisschen länger als der andere. Den längeren Draht nennt man die Anode der LED, den kürzeren die Kathode. Wenn man eine Stromquelle (Batterie oder ein Ausgang vom Hub) an die Anode anschließt, fließt Strom von der Anode zur Kathode der LED und die LED leuchtet. Wenn man sie falsch herum anschließt, fließt kein Strom und die LED leuchtet nicht. Fließt ein zu großer Strom durch die LED, dann geht sie kaputt. Deshalb muss man einen sogenannten Vorwiderstand zwischen LED und der Stromquelle einbauen.
- c. Alle Kathoden der LEDs werden miteinander verbunden und dann mit einer gemeinsamen schwarzen Leitung an den Masse-Anschluss der Innovator Hubs angeschlossen.
- d. Da die Anordnung der LEDs auf dem Breadboard auch wie die Augen eines richtigen Würfels aussehen sollen, müssen wir bei der Verdrahtung ein bisschen tricksen. Versucht, möglichst viele Kathoden der LEDs in einer Reihe zu bündeln und diese Reihe dann in einem Minus-Block mit den Kathoden der restlichen LEDs zusammenzuschließen. So spart Ihr Kabel zum Innovator Hub.

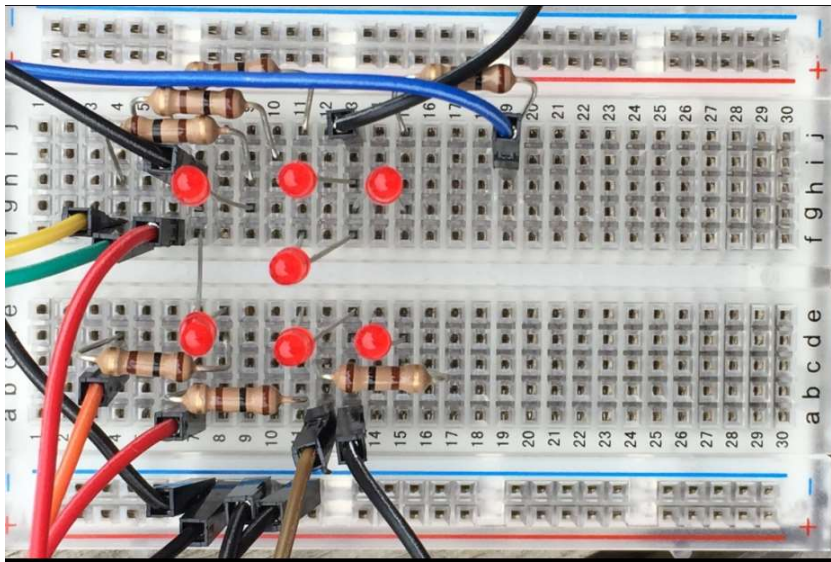


Hier im Beispiel auf dem Bild wurden die Kathoden von LED 2,3 und 4 in Reihe 13 f,g,h des Breadboards gesteckt und mit einer gemeinsamen schwarzen Leitung an den Minus-Block von Reihe 7-12 angeschlossen. Ebenso wurden die Kathoden von LED 1 und 5 in Reihe 7 verbunden und LED 6 und 7 in Reihe 13 d und e gesteckt und diese Reihen auch mit dem Minus-Block von Reihe 7-12 verbunden.

- e. Als nächstes montieren wir die 100 Ohm Vorwiderstände und verbinden sie mit jeweils einer Anode der LEDs. Bei dem anderen Anschluss der

Vorwiderstände muss man genau darauf achten, dass wir nicht aus Versehen schon eine mit Minus verbundene Reihe nehmen. Wir nehmen die 100 Ohm Vorwiderstände aus der Tüte, die mit STEMEE/AC/RES/A usw bedruckt ist. Dein Betreuer erklärt Dir, wie man den Farbcode der Widerstände liest und welche die 100 Ohm Vorwiderstände sind.

- f. Verbinde das noch freie Ende der Vorwiderstände mit einem bunten Kabel mit dem Hub und notiere Dir, welche LED Nummer mit welchem Ausgang des Hubs verbunden wurde.
- g. Zum Schluss richte die LEDs nochmal so aus, dass sie wie die Augen eines Würfels ausgerichtet sind.



- h. Lass noch mal einen Betreuer über Dein Breadboard schauen, bevor Du deine Hardware testest.

3. Das Breadboard mit den Würfel-LEDs testen

- a. Für den Test des LED-Würfels brauchen wir dieses Mal keine Software! Wir verbinden als Erstes die Kathoden aller LEDs über die schwarze Leitung mit einer beliebigen Masse-Buchse des Innovator Hubs. Hier im Bild haben wir die Masse-Buchse unterhalb BB2 benutzt.



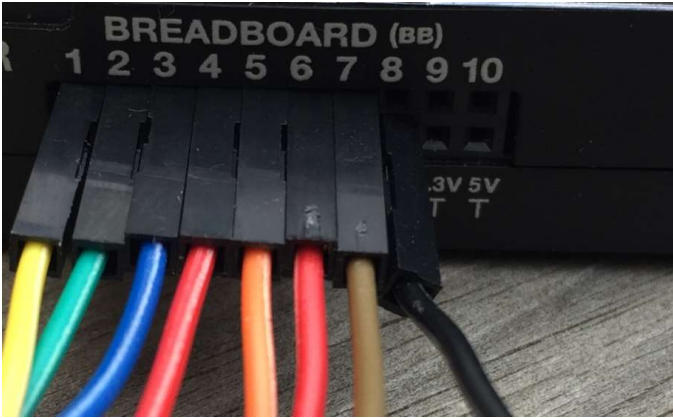
- b. Als Nächstes testen wir jede einzelne LED auf ihre Funktion, indem wir die zugehörige Leitung in die 3.3V Buchse stecken, wie auf dem Bild hier:



Die zugehörige LED sollte leuchten. Ist das nicht der Fall, überprüfe bitte die Verkabelung auf dem Breadboard:

- i. Ist die LED richtig herum angeschlossen? Der kürzere Draht ist die Kathode und er muss mit dem schwarzen Kabel verbunden sein.
 - ii. Ist der Vorwiderstand mit dem längeren Draht der LED verbunden?
 - iii. Ist die bunte Leitung mit dem anderen Ende des Vorwiderstands verbunden?
- c. Wenn die LED leuchtet, steck den Stecker gleich danach in den richtigen BB Anschluss. Schau Dir die Nummerierung der Würfelaußen im Bild unter Aufgabe 1a nochmal an. LED1 sollte mit Buchse BB1 verbunden sein, LED2 mit BB2, usw.

d. Am Schluss sollte Deine Verkabelung ungefähr so aussehen:



4. Ansteuerung des Würfels mit dem Innovator Hub

Als erstes testen wir die Ansteuerung der sieben LEDs mit dem Innovator Hub.

- a. Im Programm werden die sieben LEDs jeweils mit diesem Befehl verbunden: Send "CONNECT LED x BB x"
wobei x die Nummer der LED ist.
- b. Danach schalten wir die LEDs der Reihe nach ein, mit dem Befehl:
Send "SET LED x ON"
Füge hinter jedem Befehl ein <Wait 1> Kommando ein, damit Du sehen kannst, ob sich wirklich eine LED nach der anderen einschaltet.
- c. Zum Schluss schaltest Du alle LEDs der Reihe nach wieder aus, mit Send "SET LED x OFF" und fügst auch dort nach jedem Ausschalten ein Wait 1 Befehl ein.
- d. Schaffst Du es, dieses Programm mit 11 Zeilen Code zu kodieren? Kleiner Tipp: Denk mal an die <For ,, EndFor> und <Eval(x)> Befehle! Damit sparst Du dir eine Menge Tipparbeit!

5. Zufallszahlen erzeugen

Wir machen uns mit der Funktion randInt(a,b) vertraut. Dafür brauchen wir nur den CX CAS Rechner und trennen das USB Kabel, das den Rechner mit dem Hub verbindet.

- a. Erstelle ein Programm (zum Beispiel „Würfel1“), das über die randInt(a,b) Funktion eine zufällige Zahl zwischen 1 und 6 erzeugt und auf dem Display des Taschenrechners anzeigt. Wir rufen randInt(a,b) auf, indem wir den Wert von randInt einer Variablen zuweisen. Die richtige Codezeile sieht so aus:
wurf := randInt(1,6)
- b. Starte das Programm mehrmals und beobachte, ob sich die Zahlen auf dem Display ändern.
- c. Wie oft musstest Du das Programm starten, bis alle Zahlen von 1 bis 6 mindestens einmal angezeigt wurden?
- d. Lösche Dein Ergebnis-Fenster über [menu-1-5 <Protokoll löschen>] .
Erstelle eine Tabelle, in der Du die Häufigkeit der Zahlen 1 bis 6 nach 10 bzw 20 „Würfeln“ einträgst. Vergleiche Deine Tabelle mit Deinen

Nachbarn. Kommt ihr immer zum gleichen Ergebnis?

6. Das Zufallszahlenprogramm für das Breadboard erweitern

- a. Erstelle eine Kopie des Programms aus Aufgabe 5 und speichere es unter einem neuen Namen ab (zum Beispiel Würfel2).
- b. Schließe den Hub wieder an den CAS CX Rechner an.
- c. Nun brauchen wir für jede Würfelzahl ein entsprechendes Würfelaugenprogramm, das die der gewürfelten Zahl entsprechenden LEDs einschaltet. Um uns auch dort Tipparbeit zu sparen, arbeiten wir wie schon bei der Ampelsteuerung mit Unterprogrammen.
- d. Erstelle für jede mögliche Würfelzahl je ein Unterprogramm, das Du <Wurf1>, <Wurf2>,<Wurf3>, usw nennst. In dem Unterprogramm schaltest Du immer nur die LEDs ein, die für die entsprechend gewürfelte Zahl nötig sind.
- e. Zum Schluss erstellst Du noch ein Unterprogramm mit dem Namen <WurfAus>, in dem Du alle LEDs ausschaltest.
- f. Das Hauptprogramm ergänzt Du noch durch folgende Befehle:
 - i. Vor dem randInt(a,b) Befehl müssen die sieben LEDs mit dem Send "CONNECT LED x BB x" Befehl im Hub verbunden werden. Verwende hier wieder eine <For i,1,7 EndFor> Schleife und den Eval(i) Befehl.
 - ii. Lass den <Disp wurf> Befehl aus der Aufgabe 5 im Programm stehen. So kannst Du immer kontrollieren, ob die im Display erschienene Zahl auch richtig als Würfelaugen auf dem Breadboard angezeigt wird. Nach dem <Disp wurf> Befehl erfolgt nun die Übersetzung der gewürfelten Zahl in die entsprechenden Augen des Würfels. Verwende hierfür die <If Then Elseif> Konstruktion und rufe jeweils das richtige Unterprogramm auf, das Deine LEDs einschaltet. Der für die ‚Eins‘ und die ‚Zwei‘ gewürfelte Konstruktion sieht dann so aus:

```
If wurf=1 Then
  wurf1()
Elseif wurf=2 Then
```

wurf2()

usw.

- iii. Vergiss nicht, Deine <If Then ElseIf> Konstruktion am Ende (nach Aufruf von wurf6()) mit einem EndIf abzuschließen.
- g. Starte das Hauptwürfelprogramm (in dem der Befehl randInt(a,b) steht) mit <ctr R>. Mit jedem neuen Start des Programms wird neu gewürfelt und das Ergebnis auf dem Breadboard angezeigt.

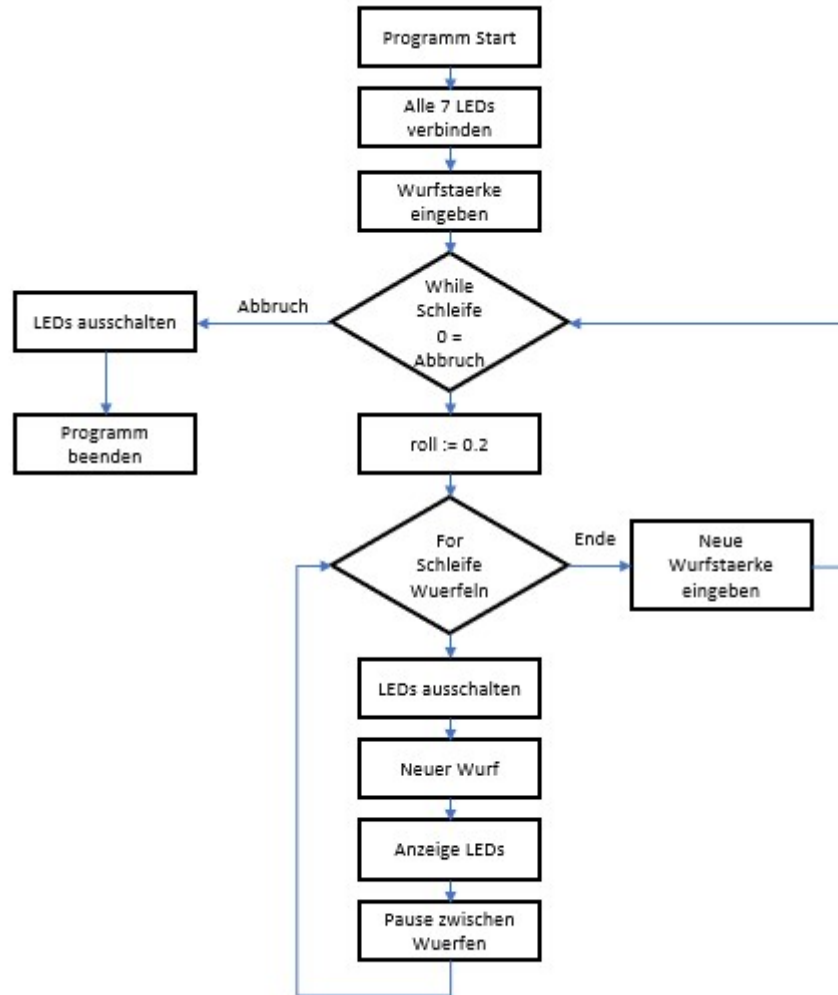
7. Einen rollenden Würfel programmieren

Nun erweitern wir das Programm aus Aufgabe 6 zu einem rollenden Würfel, dessen Rollstärke wir über die Tastatur über die Zahlen 1 bis 9 eingeben können. Bei einer 1 rollt der Würfel nur einmal und bei einer 9 neunmal. Durch Drücken der 0 brechen wir das Programm ab.

- a. Wir kopieren unser Programm aus Aufgabe 6 und vergeben einen neuen Namen, z.B. Würfel3.
- b. Nach dem EndFor Befehl der „Connect LED“ Schleife bauen wir die Abfrage der Würfelstärke mit einem <Request> Befehl ein. Den eingegebenen Wert weisen wir der Variablen „stark“ zu. Die Befehlszeile sieht dann so aus:
Request „Würfelstärke (0=Abbruch):“,stark
- c. Als nächstes bauen wir eine <While EndWhile> Schleife ein, damit das das Würfelprogramm so oft wiederholt wird, bis eine ‚0‘ eingegeben wurde. Dies geschieht durch Einfügen der Zeile „While stark > 0“ direkt nach dem <Request> Befehl. Der <EndWhile> Befehl kommt direkt vor den letzten Befehl <EndPrgm> unseres Programms.
- d. Vor dem <EndWhile> Befehl müssen wir nochmal einen <Request> Befehl einfügen, damit wir das Programm aus der <While EndWhile> Schleife heraus mit der „0“ abbrechen können.
- e. Probier Dein Programm aus. Solange Du eine Zahl grösser „0“ eingibst sollte das Programm einmal würfeln und Dich dann wieder nach der neuen Würfelstärke fragen.
- f. Nun müssen wir dem Würfel das Rollen beibringen. Dafür brauchen wir eine Hilfsvariable „roll“, mit der wir die Pause zwischen dem Rollen

kontrollieren und eine weitere <For ,, EndFor> Schleife, mit der wir die Anzahl des Rollens kontrollieren.

- i. Wir fangen mit einem Anfangswert <roll:= 0.2> an, d.h. zwischen dem ersten und zweiten Rollen des Würfels warten wir 0.2 Sekunden. Wir fügen diese Zeile direkt nach dem <While stark>0> Befehl ein.
- ii. Direkt nach dieser Zeile fügen wir eine weitere <For ,, EndFor> Schleife ein:
For r,1,stark
„stark“ ist unsere Würfelstärke, d.h. wenn wir eine „9“ eingegeben haben, wird neunmal gewürfelt und die Pause zwischen den Würfeln nimmt jedes Mal zu, d.h. der Würfel „rollt aus“.
- iii. Direkt nach dem <EndIf> des Würfels kommt das <EndFor> unserer neuen Schleife.
- iv. Vor diesem EndFor fügen wir noch die Pause zwischen den Würfeln ein:
Wait roll
roll := roll * 1.2
Das bedeutet: nach dem ersten Wurf wartet das Programm 0.2 Sekunden, nach dem zweiten Wurf $0.2 * 1.2 = 0.24$ Sekunden, nach dem dritten $0.24 * 1.2 = 0.288$ Sekunden usw., bis maximal 0.71 Sekunden vor dem 9. Wurf. Du kannst auch gerne mal eine andere Zahl ausprobieren, pass aber auf, dass die Zeit zwischen dem 8. Und 9. Wurf nicht zu groß wird!
- v. Damit der neue Wurf auch immer richtig auf dem Breadboard angezeigt wird, rufen wir nach dem randInt(a,b) Befehl das Unterprogramm <wurfaus()> auf. Damit werden alle LEDs ausgeschaltet, bevor die richtigen LEDs des neuen Wurfs eingeschaltet werden.
- vi. Jetzt ist das Programm recht kompliziert geworden, mit den vielen Schleifen. Damit ihr den Überblick behaltet, hier ein Flussdiagramm des Programms von Aufgabe 7:



8. Aufräumen!

- a. Die Widerstände braucht ihr nicht wieder geradebiegen, sie kommen so in die Widerstandstüte im untersten Fach der MINT Box.
- b. Die LEDs kommen in die LED Tüte im untersten Fach der MINT Box.
- c. Die Kabel kommen in die große Kabeltüte und die Tüte zusammen mit dem Breadboard und dem Hub in das oberste Fach der MINT Box.
- d. Der Innovator Hub kommt in das oberste Fach und das USB Kabel in das mittlere Fach.
- e. Die Taschenrechner kommen in den silbernen Alukoffer.

Vielen Dank fürs Aufräumen!!

Viel Spaß beim Programmieren des TI-Innovator Hubs in der MINT Garage!!