

Session 8 – Hub & Breadboard - Ampelanlage

Heute beschäftigen wir uns mit dem TI-Innovator Hub und dem Breadboard und bauen eine Ampelanlage auf.

Dabei 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!

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 eine Ampelanlage aufzubauen und zu programmieren!

FAQs:

- Frage: Wie kopiere ich Dateien und Programme und nenne 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 Touchpad 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 eingefuegt werden soll.
 - <ctrl><v> drücken.
 - Man kann natuerlich 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. Den Vibrationsmotor mit output 1 verbinden [<menu>-8-8-B]:
Send "CONNECT VIB.MOTOR 1 OUT 1"
4. Den Vibrationsmotor ein oder ausschalten oder mit 2Hz „blinken“ lassen
[<menu>-8-1-B]:
Send "CONNECT VIB.MOTOR 1 ON"
Send "CONNECT VIB.MOTOR 1 OFF"
Send "CONNECT VIB.MOTOR 1 ON BLINK 2"
5. Einen Digitalausgang verbinden [<menu>-8-8-F]:
Send "CONNECT DIGITAL.OUT x BB y"
6. Einen Digitaleingang verbinden [<menu>-8-9-E]:
Send "CONNECT DIGITAL.IN x BB y"
7. Einen Digitalausgang auf HIGH oder LOW setzen [<menu>-8-1-I]:
Send "SET DIGITAL.OUT x HIGH/LOW"
8. Einen Digitaleingang lesen [<menu>-8-2-E], [<menu>-8-5]:
Send "READ DIGITAL.IN 1"
Get var

Aufgaben:

1. Die Ampelteile auf dem Breadboard befestigen



- a. 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.

Such Dir die passenden Teile aus Deiner MINT Box, wie auf dem Bild hier ersichtlich. Die gelbe LED ist nicht Teil der MINT Box, du erhältst sie von Deinem Betreuer.

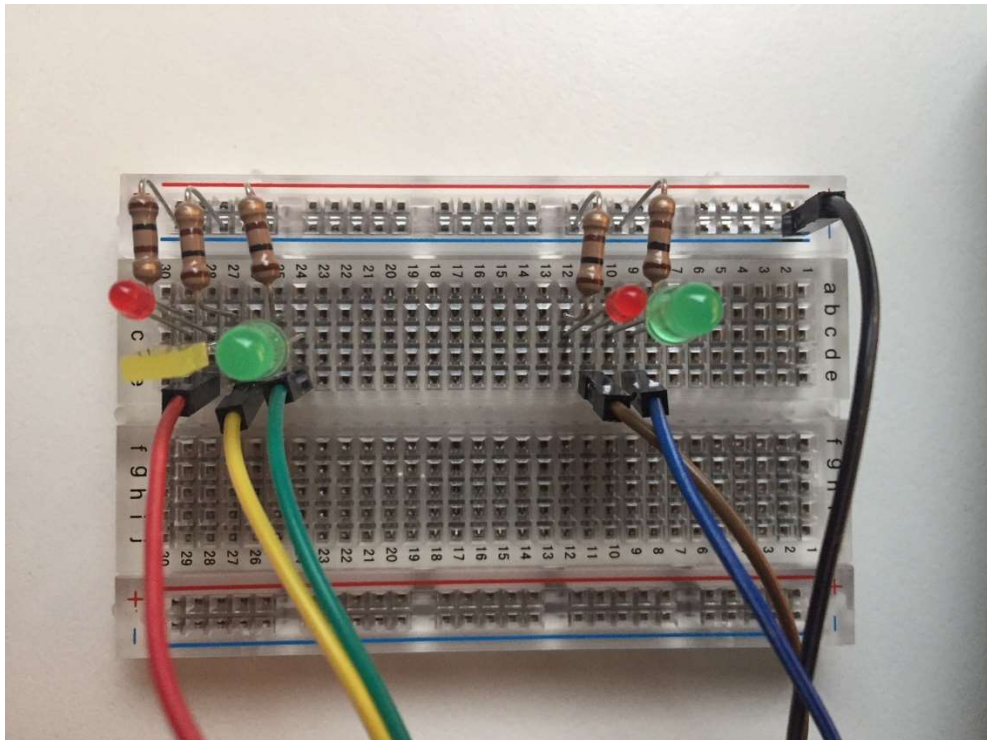
Wir nehmen einen 100 Ohm Vorwiderstand aus der Tüte, die mit STEMEE/AC/RES/A usw bedruckt ist. Dein Betreuer erklärt Dir, wie man den Farbcode der Widerstände liest und welche die 100 Ohm Vorwiderstände sind.

Für jede LED brauchen wir ein Anschlusskabel, mit dem wir die LED mit dem

entsprechenden BreadBoard Anschluss verbinden. Damit Du nicht durcheinander kommst, nimm bitte folgende Farben für die einzelnen Anschlussdrähte:

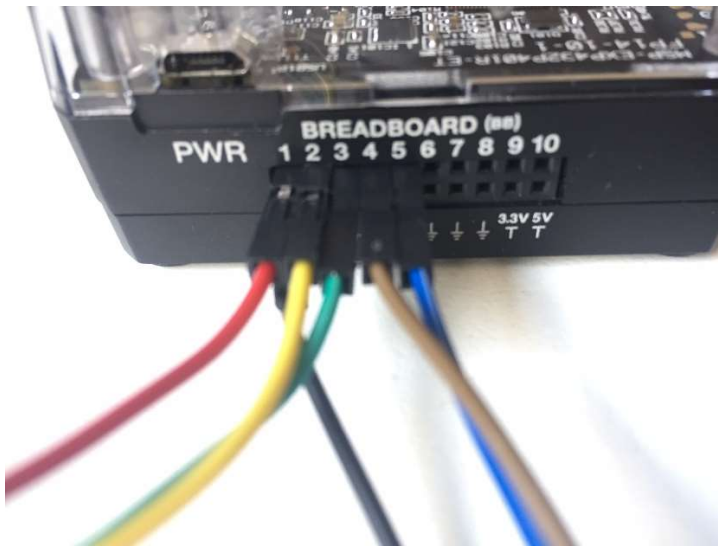
- Ampel-Rot (BB1): rotes Kabel
- Ampel-Gelb (BB2): gelbes Kabel
- Ampel-Grün (BB3): grünes Kabel
- Fußgänger-Rot (BB4): braunes Kabel
- Fussgänger-Grün (BB5): blaues Kabel
- Minus/Masse: schwarzes Kabel

b. Lege das Breadboard vor Dir hin, wie auf dem Bild hier zu sehen. Lass Dir die Funktionsweise des Breadboards von Deinem Betreuer erklären. Stecke jede LED (rot, gelb, grün) der Ampel mit ihrem längeren Anschluss (Anode) nach rechts in einer Reihe nebeneinander in das Breadboard. Hier im Bild habe ich die Reihe c genommen.



c. Mit ein bisschen Abstand, stecke die zwei LEDs (rot, grün) der Fußgängerampel ebenfalls in einer Reihe in das Board, mit der Anode nach rechts.

- d. Verbinde bitte jede Anode der Leuchtdiode mit dem entsprechenden Anschluss am Hub, wie oben unter (a) zugeordnet.
- e. An jede Kathode wird nun ein Vorwiderstand verbunden (hier im Bild Reihe b) und das andere Ende des Vorwiderstands wird in die mit einem blauen Minus (-) gekennzeichnete Reihe gesteckt.
- f. Zum Schluss steckst Du das schwarze Kabel in die blaue (Minus) Reihe und verbindest es mit einem der Minus (Masse) Anschlüsse des Hubs unterhalb der BB Anschlüsse, wie auf dem Bild hier zu sehen.



Lass noch mal einen Betreuer über Dein Breadboard schauen, bevor Du ein Programm erstellst, um Deine Hardware zu testen.

2. Das Breadboard mit den Ampel-LEDs testen (AmpelTest)

- a. Erstelle ein Programm, bei dem nacheinander jede der 5 LEDs an und wieder ausgeht (AmpelTest1).

Verwendete Befehle:

<Send CONNECT LED x BB x> Die LEDs mit dem Hub verbinden.

<For ,, EndFor>, damit man nicht so viel tippen muss.

<Send SET LED eval(i) ON/OFF> Die LEDs ein oder ausschalten

<Wait x> Pause, bevor die nächste LED an- oder ausgeht.

- b. Speicher das Programm von (a) unter AmpelTest1 ab und nutze das Programm als Basis für AmpelTest2. Gehe wie folgt vor:
 - i. Im AmpelTest1 Programm die Taste <doc> drücken, dann 1 (für Datei), 5 (für speichern unter) und dann AmpelTest2 als Dateinamen eingeben und <ok> anklicken.
 - ii. Du solltest dann als Dateinamen oben in der Mitte des Bildschirms <AmpelTest2> sehen.
 - iii. Dann im Programm-Editor Fenster noch den Programmnamen ändern: <menu> drücken, dann 1 (für Aktionen), 6 (für Umbenennen) und auch dort ampeltest2 als neuen Programmnamen eingeben.
 - iv. Erst jetzt solltest Du dein neuen Programmcode ändern, hast Dir aber eine Menge Tipparbeit gespart.
- c. Ändere das AmpelTest2 Programm so ab, dass die fünf LEDs der Reihe nach angehen (1 Sekunde Pause zwischendurch), bis schließlich alle LEDs leuchten.

3. Ampelsteuerungen

Hier erstellst Du zunächst einzelne Programme für die Sequenz ‚Ampel schaltet von Rot nach Grün‘, ‚Ampel schaltet von Grün nach Rot‘, ‚Fußgängerampel von Rot nach Grün‘, Fußgängerampel von ‚Grün nach Rot‘, sowie ‚Ampel aus‘.

- a. Erstelle ein neues Dokument unter dem Namen <Ampel3>
- b. Im Dokument, füge ein neues Programm Editor Fenster ein mit dem Namen <AmpelRotGrün>. Dort erstelle ein Programm, dass zunächst die rote LED1 leuchten lässt, dann die rote und gelbe zusammen und nach einer Pause nur noch die grüne LED3, so, wie bei einer richtigen Verkehrsampel. Benutze nach jeder Änderung den <wait> Befehl, um die Ampel wie in der Wirklichkeit von Rot, über Rot-Gelb, nach Grün schalten zu lassen. Spiele mit den <wait> Zeiten, bis die Abfolge so aussieht, wie bei einer richtigen Verkehrsampel.
- c. Erstelle im selben Dokument ein weiteres Programm (mit dem <ctr>doc Befehl), dass Du <AmpelGrünRot> nennst. Dieses Programmfenster programmierst Du dann so, dass zunächst die grüne LED leuchtet, dann nur die gelbe und nach einer Pause nur noch die rote LED.
- d. Und nochmal brauchen wir ein neues Programmfenster, nenne das neue Fenster <FussGrünRot>. Editiere das Programmfenster so, dass zunächst die grüne Fußgänger-LED (LED4) leuchtet, dann die rote (LED5).
- e. Noch ein neues Programmfenster und nenne das neue Fenster <FussRotGrün>. Editiere das Programmfenster so, dass zunächst die rote LED leuchtet, dann die grüne.
- f. Zum Schluss noch ein neues Programmfenster und nenne das neue Fenster <Ampelaus>. Editiere das Programmfenster so, dass alle 5 LEDs ausgeschaltet werden.
- g. Du solltest nun fünf Programmfenster in Deiner Datei haben (AmpelRotGrün, AmpelGrünRot, FussGrünRot, FussRotGrün, AmpelAus). Nun erstelle ein weiteres Programmfenster mit dem Namen <Ampel3>.
- h. In dem <Ampel3> Programmfenster verbindest Du zunächst wieder die fünf LEDs über die <Send CONNECT LED x BB x> Befehle. Danach rufst Du Deine Unterprogramme auf, um eine komplette Ampelsteuerung einer Fußgängerampel zu bauen. Die Namen Deiner Unterprogramme siehst Du nach Drücken der <var> Taste. Du kannst sie dort im Fenster

auswählen und musst sie nicht immer wieder neu eintippen.

Hier die Programmschritte im <Ampel3> Programm:

- i. Alle fünf LEDs mit dem Hub verbinden
- ii. Verkehrsampel ist grün (LED3 ist an) und Fußgängerampel ist rot (LED4 ist an). Alle anderen LEDs sind aus.
- iii. Nach einer von Dir gewählten Zeit (<wait> Kommando) AmpelGrünRot() aufrufen (die Verkehrsampel schaltet von Grün nach Rot).
- iv. 1 Sekunde oder länger warten.
- v. FussRotGrün() aufrufen (die Fußgänger haben Grün)
- vi. 10 Sekunden warten
- vii. FussGrünRot() aufrufen (die Fußgänger haben Rot)
- viii. Nach 1 Sekunde Wartezeit AmpelRotGrün aufrufen (der Verkehr bekommt wieder Grün)
- ix. Programmierere noch eine Schleife zu Beginn und am Ende des Programms, so dass sich das Ampel-Programm immer wiederholt, bis Du die Taste „esc“ auf dem Taschenrechner drückst. Nachdem die <esc> Taste gedrückt wurde, schaltest Du alle LEDs aus mit dem ampelaus() Unterprogramm. Dein Programm sieht dann so aus:

```
Prgm
Connect LEDs (LED1 bis LED5)
t:= "x"
While t ≠ "esc"
  hier steht Dein eigentliches Programm...
...
t:=getKey()
EndWhile
EndPrgm
```

- i. Speicher Dir dieses Programm gut ab, Du brauchst es später nochmal!

Hinweis: Durch die vielen <wait> Kommandos kann es eine Weile dauern, bis Dein Programm merkt, dass Du die <esc> Taste gedrückt hast. Du erkennst am Uhr-Symbol auf dem Rechner, dass Dein Programm noch <wait> Kommandos abarbeitet.

4. Fußgängerampel für Blinde

Hier wollen wir unsere Fußgängerampel mit akustischen Signalen ausstatten. Blinde können so hören oder fühlen, ob die Fußgängerampel gerade Rot oder Grün ist. Ein Dauerton bedeutet: die Ampel ist grün. Ein unterbrochener Ton bedeutet: die Ampel ist rot.

- a. Kopiere Dir Dein Programm-Dokument unter einem neuen Namen (z.B. Ampel4) und ändere auch den Namen Deines Hauptprogramms in der Datei von Ampel3 auf Ampel4.
- b. Nimm den Vibrationsmotor aus der Packung der MINT Box und verbinde ihn mit der OUT1 Buchse des Hubs.
- c. Nimm den Zusatzakku aus der MINT Box und schließe ihn an die PWR Buchse vom Hub an. Ohne den Zusatzakku wird der Vibrationsmotor nicht laufen und Du erhältst einen runtime error, wenn Du dein Programm ablaufen lässt.
- d. Schalte den Zusatzakku ein, indem Du kurz auf den Einschalter im silbernen Feld des Zusatzakkus drückst. Im Hub muss die Zusatzakku-LED leuchten (siehe Bild).
- e. Ergänze Dein Programm von Aufgabe 3 mit folgenden Vibrationsmotor-Befehlen:
 - Füge eine `<Send "CONNECT VIB.MOTOR 1 TO OUT 1">` Zeile direkt nach den `CONNECT LED...` Zeilen ein.
 - Füge eine `<Send "SET VIB.MOTOR 1 ON BLINK 2">` direkt hinter dem Befehl ein, wo Du Deine Fußgängerampel auf Rot schaltest.
 - Füge eine `<Send "SET VIB.MOTOR 1 ON">` Zeile direkt hinter dem Befehl ein, wo Du Deine Fußgängerampel auf Grün schaltest.
 - Schalte den Vibrationsmotor am Ende Deines Programms aus, indem Du den Befehl `<Send "SET VIB.MOTOR 1 OFF">` direkt hinter dem `ampelaus()` Kommando einfügst.

5. Fussgänger-Bedarfsampel

Nun wollen wir unsere Ampel so programmieren, dass der Straßenverkehr immer Grün hat, solange kein Fußgänger die Straße überqueren will. Wir nutzen dazu die Taste 5 auf dem Taschenrechner und quittieren die Grünanforderung mit einem Text auf dem Display des Rechners.

- a. Editiere Dein Ampelprogramm von Aufgabe 4 mit folgenden Kommandozeilen:
Hinter den Befehlen, wo Verkehrsampel grün leuchtet und die Fußgängerampel rot, fügst Du folgende Zeilen ein:
If t = 5 then
 DispAt 1, "Bitte warten, Grün kommt!"
 Wait 5
 Hier steht Dein Programm von Aufgabe 4...
 .
 .
 .
Füge über der t:=getKey() Zeile vor der EndWhile Zeile folgende Zeilen ein:
EndIf
 DispAt 1, "Taste 5 drücken, um Grün anzufordern!"
- b. Editiere Dein Programm so, dass nach Drücken der Taste ,5' die Verkehrsampel nur einmal von Grün auf Rot geht, nach einer Wartezeit die Fußgängerampel dann von Rot auf Grün und, nachdem die Fußgänger genug Zeit für die Überquerung der Straße gehabt haben, die Fußgängerampel wieder auf Rot und die Verkehrsampel wieder auf Grün geht und so bleibt, bis Du wieder die Taste ,5' auf dem Rechner drückst.

6. Kreuzungsampel mit zwei Hubs

Zuletzt wollen wir noch eine komplette Verkehrskreuzungsampel aufbauen. Hierzu suche Dir bitte einen Partner mit einem zweiten Taschenrechner, Hub und Breadboard.

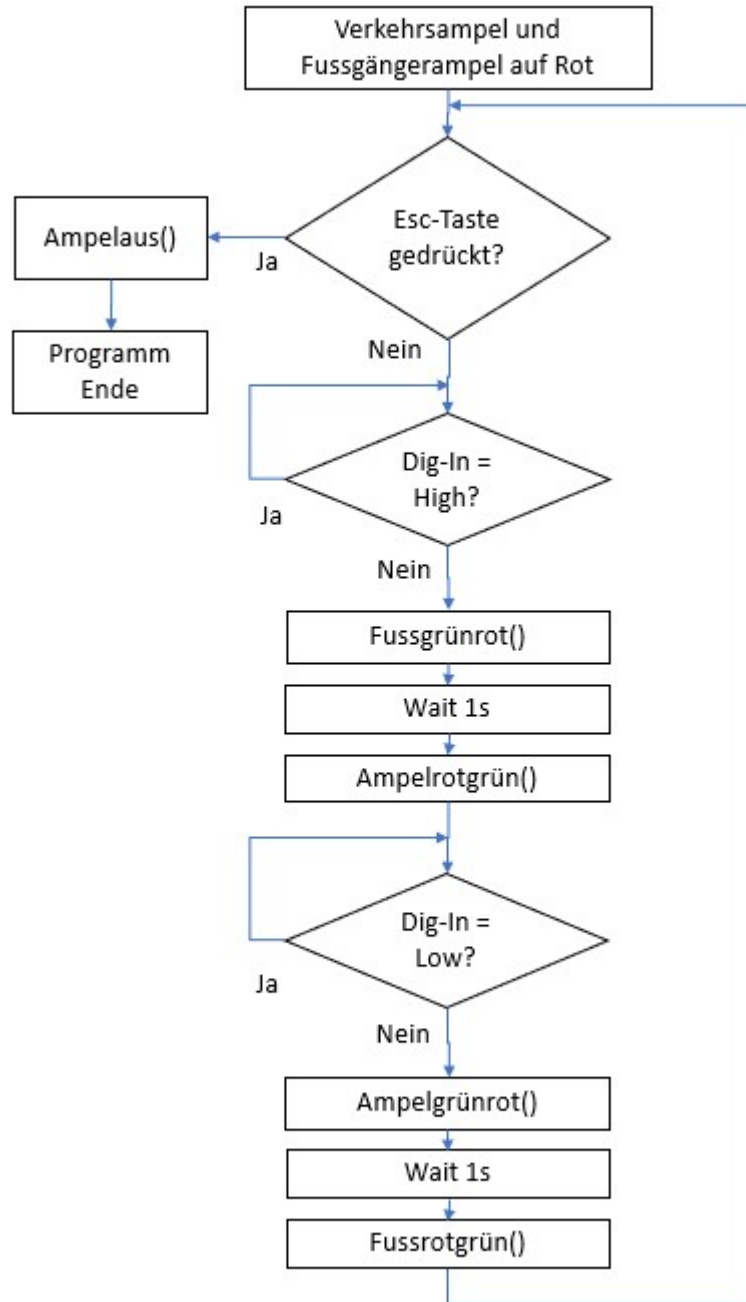
- a. Zuerst legt Ihr fest, welcher der zwei Hubs der Lehrer sein soll und welcher Hub der Schüler. Der Lehrer gibt den Ampeltakt vor und sagt dem Schüler-Hub, wann er seine Ampel auf Rot bzw auf Grün schalten soll.
- b. Die Programme des Lehrer-Hubs und des Schüler-Hubs müssen unterschiedlich sein, pass also bitte genau auf, ob Du den Lehrer-Hub oder den Schüler-Hub in dieser Aufgabe übernommen hast!!
- c. Beide, Lehrer-Hub und Schüler-Hub stecken den Vibrationsmotor ab und auch den Zusatzakku. Beides brauchen wir für diese Aufgabe nicht mehr.
- d. Alle Lehrer Hub Besitzer kopieren ihre Datei <Ampel3> auf eine neue Datei mit dem Namen <Ampel6L>. Alle Schüler Hub Besitzer kopieren ihre Datei <Ampel3> auf <Ampel6S>.
- e. Änderungen im Lehrer-Ampel Programm (Ampel6L):
 - i. Direkt nach dem Verbinden der 5 LEDs Füge folgendes Kommando ein:
Send "CONNECT DIGITAL.OUT 1 BB 7"
 - ii. Suche die Stelle in Deinem Programm innerhalb der <While> Schleife, wo Du die Ampel zum ersten Mal auf Grün schaltest. Füge direkt danach folgendes Kommando ein:
Send "SET DIGITAL.OUT 1 HIGH"
Dieser Befehl sagt dem Schüler, dass die Lehrerstrasse grün hat und seine Straße folglich rot haben muss.
 - iii. Suche die Stelle in Deinem Programm, wo Du die Verkehrsampel von Grün auf Rot schalten lässt. Genau hinter dieser Stelle fügst Du folgende Zeile ein:
Send "SET DIGITAL.OUT 1 LOW"
 - iv. Suche die Stelle im Programm, wo Du die Fußgängerampel nach 10 Sekunden Grün wieder auf Rot schaltest.
Genau hinter dieser Stelle fügst Du folgende Zeilen ein:

Send "SET DIGITAL.OUT 1 HIGH"

Wait 4

- v. Zum Testen Deines Programms stecke die Leitung von LED4 (Fußgängerampel-Rot) vorübergehend in den Pin BB7 vom Hub. Gleich nach Start des Programms sollte LED4 rot leuchten, Deine Ampel sollte Grün zeigen und nach einer von Dir eingestellten Zeit auf Rot gehen. Nachdem die Ampel auf Rot gegangen ist, sollte LED4 ausgehen und aus bleiben, so lange Du der Schüler-Ampel freie Fahrt programmiert hast. Danach sollte LED4 wieder angehen und nach der von Dir programmierten Wartezeit sollte Deine Ampel wieder auf Grün gehen. Der ganze Vorgang wiederholt sich, bis Du die <esc> Taste am Rechner gedrückt hast.
- vi. Wenn Alles gut funktioniert, stecke die Leitung von LED4 wieder in die Buchse BB4 Deines Hubs. Mache dann weiter mit Schritt f.
- f. Änderungen im Schüler-Ampel Programm (Ampel6S):
 - i. Direkt nach dem Verbinden der 5 LEDs Füge folgendes Kommando ein:
Send "CONNECT DIGITAL.IN 1 to IN 7 PULLUP"
 - ii. Direkt nach der While Schleife fügst Du folgende Kommandos ein:
Send "READ DIGITAL.IN 1"
Get(lehrer)
Dort fragen wir die Leitung des Lehrer-Hubs ab.

- iii. Ersetze innerhalb der While Schleife den kompletten Code mit neuem Code nach diesem Fluss-Diagramm:



- iv. Zum Testen des Schüler Programms nimm Dir bitte ein Kabel aus der MINT Box und stecke das eine Ende in die Buchse BB7 Deines Hubs. Starte Dein Programm (Ampel6S). Direkt nach dem Start

sollte Deine Verkehrsampel auf Rot stehen und Deine Fußgängerampel auf Grün.

- v. Stecke nun das andere Ende Deines Kabels in die Minus(-) Buchse direkt unter der Buchse BB7. Jetzt sollte Deine Fußgängerampel von Grün nach Rot gehen, dann Deine Ampel von Rot nach Grün und Alles so bleiben, bis Du das Kabel wieder von der Minus(-) Buchse abziehst. Deine Ampel sollte dann wieder von Grün nach Rot wechseln, die Fußgängerampel von Rot nach Grün, so lange, bis Du das Kabel wieder in die Minus(-) Buchse steckst. Drücke die <esc> Taste am Rechner, um Dein Programm zu beenden. Vielleicht musst Du das BB7 Kabel nochmal ab- und wieder anstecken, bis Dein Programm wirklich beendet ist. Mache dann weiter mit Schritt f.
- g. Nachdem nun das Lehrer- und das Schüler-Programm erfolgreich getestet wurden, können die beiden Hubs miteinander verbunden werden.
- h. Zuerst die blauen Minus (-) Reihen der beiden Hubs mit einem Kabel.
- i. Als nächstes verbindest Du die beiden BB7 Buchsen der beiden Hubs mit dem Kabel von dem Programmtest des Schülerprogramms.
- j. Nun starten der Lehrer und der Schüler ihre Programme. Und fertig ist die vollautomatische Ampelkreuzung!

7. Aufräumen!

- a. Bitte steckt den Vibrationsmotor wieder in seine Tüte im untersten Fach der MINT Box.
- b. Das Kabel vom Zusatzakku landet im mittleren Fach der MINT Box.
- c. Der Zusatzakku kommt in das oberste Fach der MINT Box.
- d. Die Widerstände braucht ihr nicht wieder gerade biegen, sie kommen so in die Widerstandstüte im untersten Fach der MINT Box.
- e. Die LEDs kommen in die LED Tüte im untersten Fach der MINT Box.
- f. Die Kabel kommen in die große Kabeltüte und die Tüte zusammen mit dem Breadboard und dem Hub in das oberste Fach der MINT Box.
- g. Die Taschenrechner kommen in den silbernen Alukoffer.

Vielen Dank fürs Aufräumen!!

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