

'We are the makers - IoT' Scénario d'apprentissage: agriculture intelligente avec un Robot IoT

Auteur: Thomas Jörg, Johannes-Kepler-Gymnasium Weil der Stadt

Le document suivant a été développé et testé dans un environnement scolaire avec env. 18 élèves de 13 à 17 ans lors de l'année scolaire 2018/2019. Il reflète l'expérience de nombreuses tentatives et de quelques échecs. Le domaine de l'IoT étant complexe, le matériel pédagogique doit être choisi avec soin. Ce document est censé être une recommandation, comme point de départ.

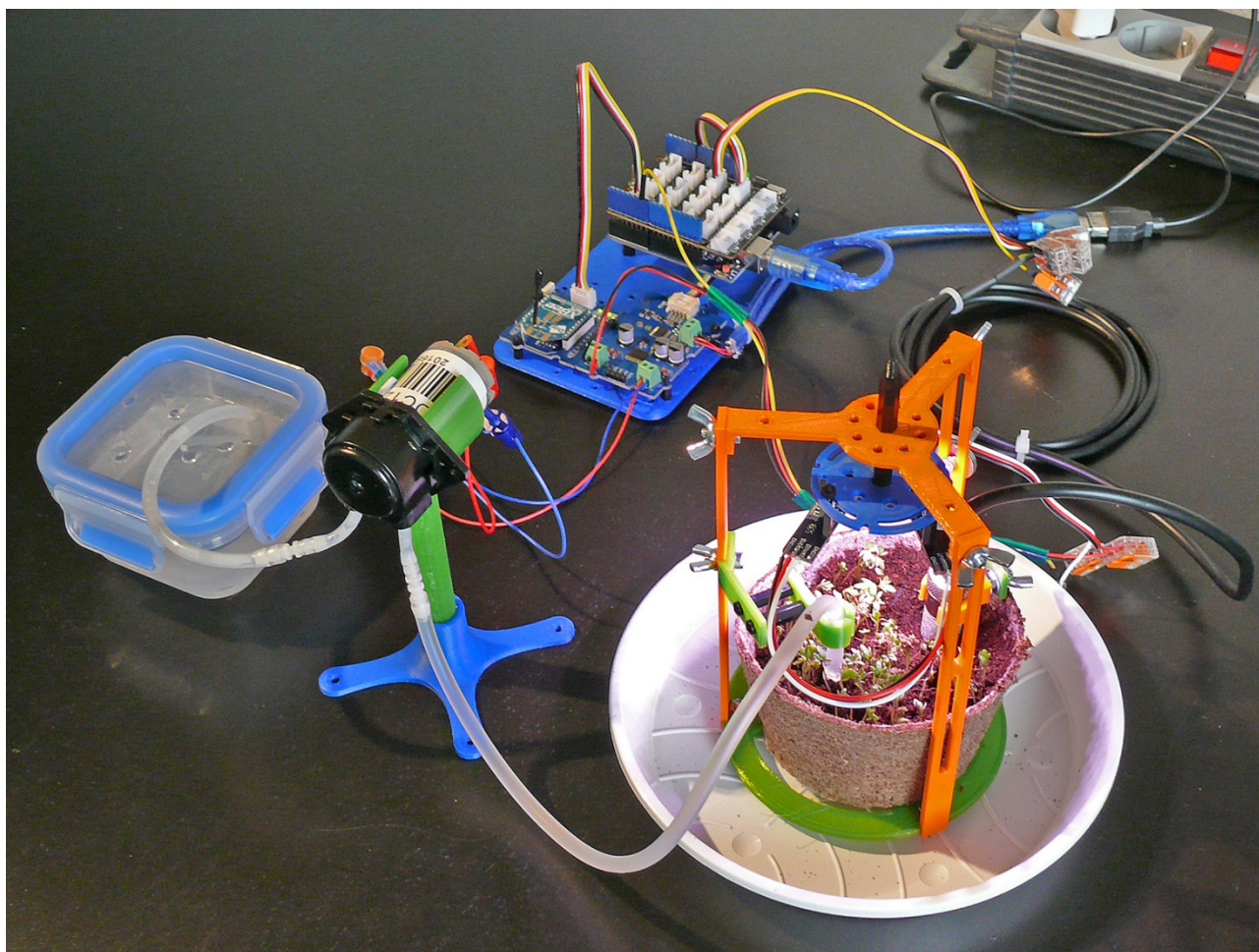


Figure 1: Prototype d'un robot cultivateur IoT

1. Title du Scénario	Apprenez à faire pousser des plantes à l'aide d'un robot IoT
2. Groupe cible	14 - 17 ans
3. Durée	Au minimum 5 semaines à raison de 2 leçons de 45 min par semaine : soit environ 6-8 heures.
4. Besoins couverts par l'activité	<ul style="list-style-type: none"> ▪ Interaction entre les composants électroniques et l'environnement (ici: les plantes) ▪ Surveillance et modification des paramètres biologiques ▪ Chaîne de communication des appareils IoT ▪ Principes des capteurs et acteurs ▪ Différents principes de mesure de l'humidité dans le sol. ▪ Principes d'éclairage LED pour les plantes en croissance ▪ Réglage fin des paramètres de la machine pour optimiser la croissance des plantes ▪ Principes des réseaux de communication sans fil ▪ Construction et impression 3D d'un environnement robotique
5. Résultats attendus	<ul style="list-style-type: none"> ▪ Comment fonctionne un système IoT? ▪ Quelles sont les possibilités et les limites des systèmes IoT? ▪ Quels composants - matériels et logiciels - sont essentiels pour construire un appareil IoT? ▪ Comment construire les règles de la biosurveillance et influencer les créatures vivantes?
6. Méthodologies	Dans ce scénario, les élèves construiront, construiront et programmeront à partir de zéro un dispositif de culture végétale entièrement interactif. Les étudiants construiront également une application pour contrôler à distance l'IoT-plantrobot
7. Lieu	<ul style="list-style-type: none"> ▪ Un laboratoire avec un ensemble de pièces et composants électroniques; ▪ Chaque groupe d'élèves doit avoir un ordinateur ou un ordinateur portable avec des privilèges administratifs pour installer différents logiciels ▪ Un projecteur pour l'enseignement des tutoriels et la présentation des travaux des étudiants; ▪ Chaque étudiant doit tenir un journal de laboratoire

8. Outils / Matériaux / Ressources

Imprimantes 3D

Environ 3-4 imprimantes 3D sont nécessaires car les élèves imprimeront leurs plantrobots IoT eux-mêmes.

Composants imprimés en 3D :

Comme point de départ, toutes les pièces nécessaires sont au format .stl et en tant que fichiers Autodesk Fusion 360.

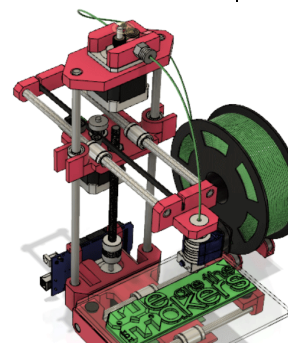


Figure 2 : Imprimante 3D

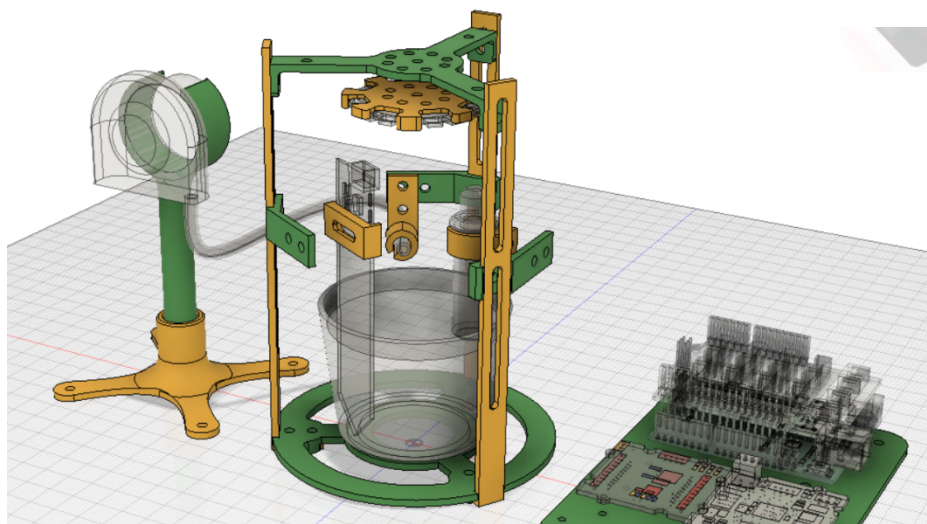


Figure 3: Présentation des données CAO

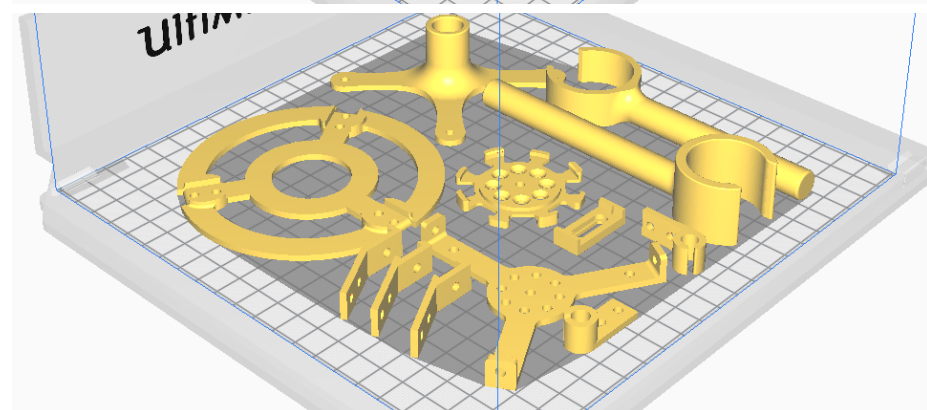
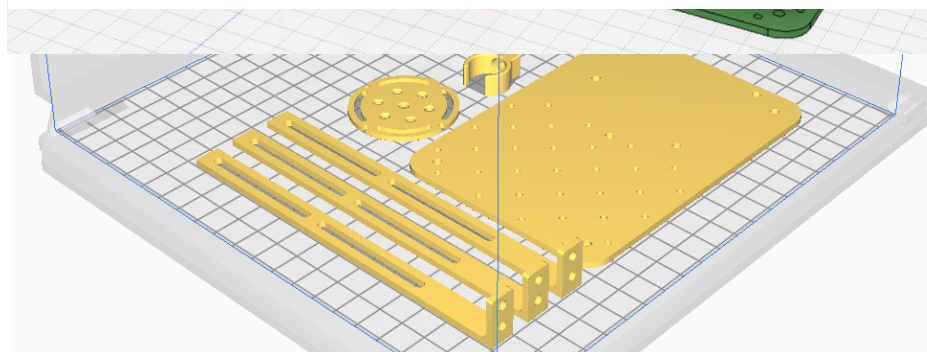
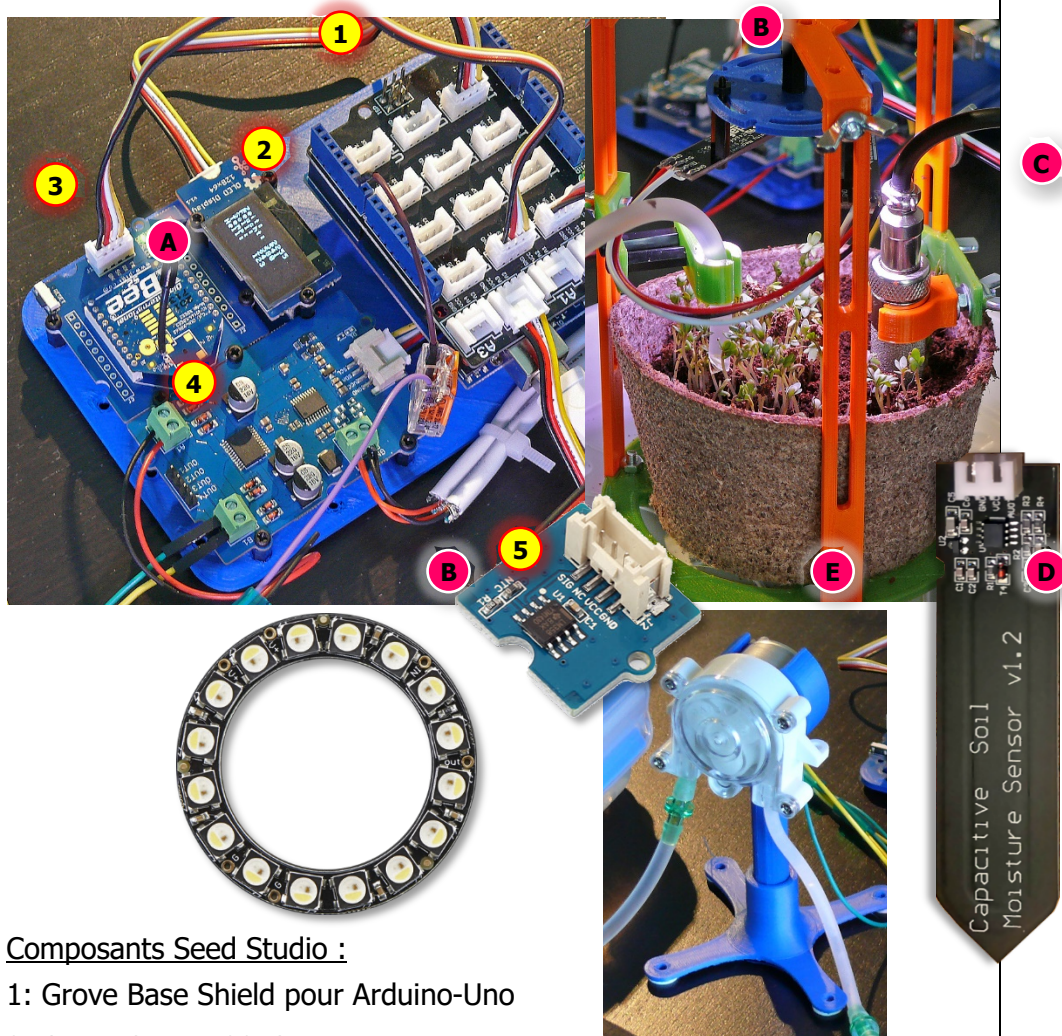


Figure 4: ensemble complet de données 3D sur une imprimante 3D de 20 cm x 20 cm

Composants électroniques:

Dans ce travail, nous recommandons le système Seed de Grove car il est facile à utiliser: (http://wiki.seeedstudio.com/Grove_System/) Tous les composants de base sauf XBees, les capteurs d'humidité et l'éclairage LED appartiennent à la norme de bosquet:



Composants Seed Studio :

- 1: Grove Base Shield pour Arduino-Uno
- 2: Grove OLED 128x64
- 3: Grove Bee socket
- 4: Grove - I2C Motor Driver (TB6612FNG)
- 5: Grove Temperatur Sensor v1.2

Capteurs et actuateurs standards:

- 1: Arduino Uno (ou équivalent)
- A: XBee Series 2C ou Series 2
- B: Anneau Adafruit Neopixel avec 16 RGBW (PAS RGB!) at 4500 K (Blanc chaud)
- C: SHT20- Capteur de température et d'humidité dans un boîtier étanche ou
- D: Capteur d'humidité du sol analogique capacitif
- E: Pompe péristaltique avec moteur 6V DC

Pièces diverses :

- Tubes en silicone de 5-6 mm (pour aquarium)
- Adaptateur pour connecter des tubes en silicone avec une pompe péristaltique
- Vis M3 et écrous papillon
- Entretoises en nylon M3 (entretoise hexagonale)
- Entretoises en nylon M2 (entretoise hexagonale) pour Grove (trous de 2 mm)
- Fils Grove
- Écrous de levier WAGO
- Fils de liaison
- Pépinières de 8 cm de diamètre
- Alimentation USB avec courant maximum de 2-2,5 A
- Adaptateur USB XBee (par exemple <https://www.waveshare.com/xbee-usb-adapter.htm>)



Plantes:

Les plantes à croissance rapide conviennent pour faire des expériences à l'école qui sont appelés «Microgreens» / «Microgreen Sprouts»; elles sont non toxique et comestible:

- Cresson alénois
- Haricots mungo
- Radis à tige rouge
- Germes de trèfle rouge
- Germes de brocoli
- Valerianella locusta



Figure 5: Cresson alénois.

ordinateurs avec les logiciels suivants préinstallés :

- Autodesk Fusion 360 (ou tout autre logiciel de modélisation 3D, par exemple Wings3D)

	<ul style="list-style-type: none">▪ Logiciel de tranchage CURA,▪ Une connexion Internet pour télécharger des bibliothèques▪ Arduino IDE▪ Traitement IDE▪ Logiciel XCTU pour configurer les XBees
--	--

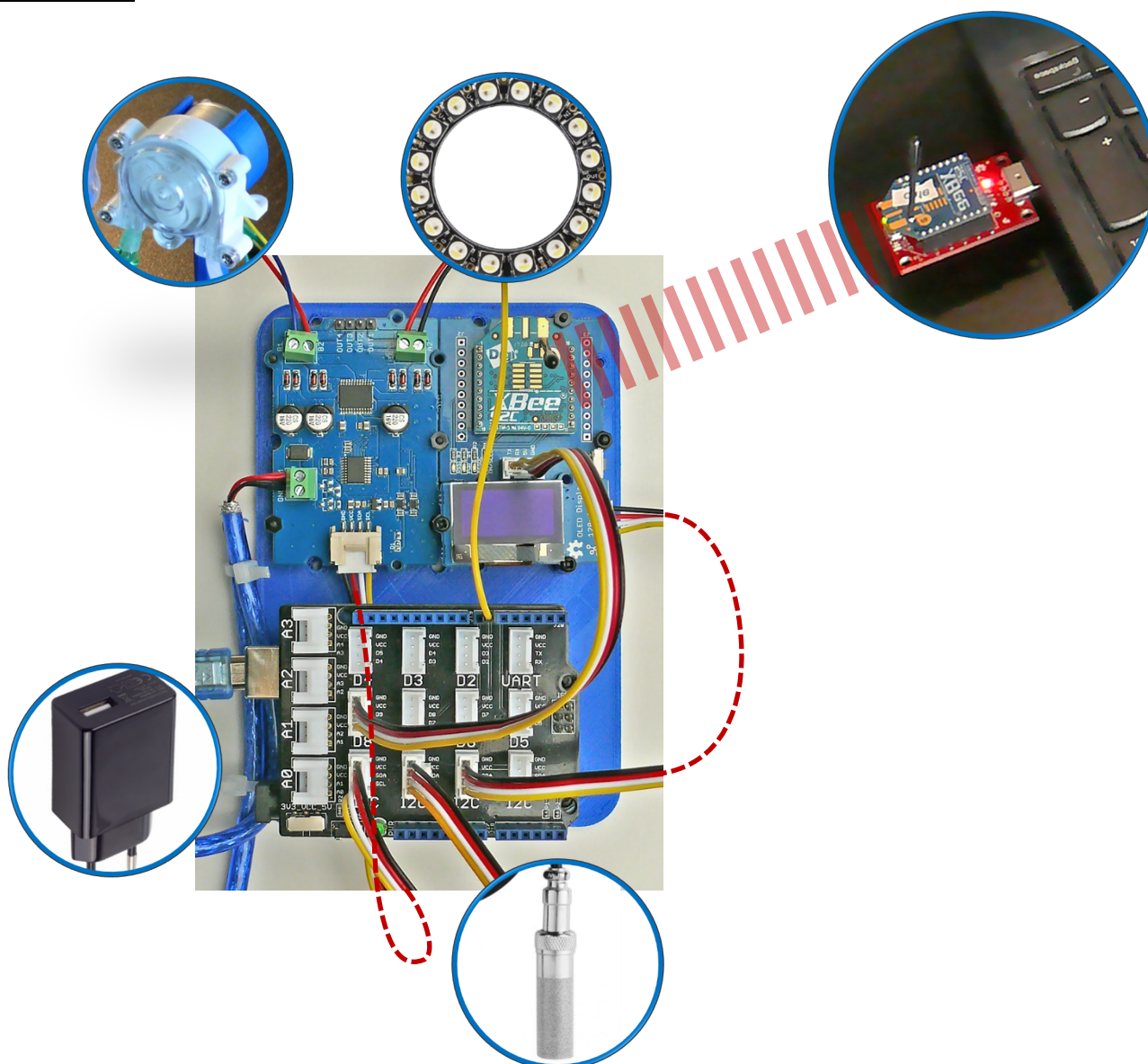
Bibliothèques Arduino pour les composants :

Certains composants comme le capteur d'humidité SHT20 ou le pilote de moteur ont besoin de bibliothèques pour que Arduino IDE fonctionne correctement. La procédure d'importation d'une bibliothèque est décrite ici : <https://www.arduino.cc/en/Guide/Libraries>

SHT20 lib (DF Robot): https://codeload.github.com/DFRobot/DFRobot_SHT20/zip/master
 OLED lib (Seeed): https://github.com/Seeed-Studio/OLED_Display_128X64/archive/master.zip
 Motor driver (Seeed): https://github.com/Seeed-Studio/Grove_Motor_Driver_TB6612FNG
 Neopixel (Adafruit): https://github.com/adafruit/Adafruit_NeoPixel/archive/master.zip

Des extraits de code spéciaux pour le capteur de température peuvent être trouvés ici:
http://wiki.seeedstudio.com/Grove-Temperature_Sensor_V1.2/

Connections:



8b Une théorie de la croissance des LED et de la mesure de l'humidité du sol

Éclairage de croissance LED

L'utilisation de LED pour la culture des plantes repose sur la théorie du PAR, «rayonnement photosynthétiquement actif»: les plantes utilisent des photons lumineux pour des réactions chimiques afin de fabriquer du sucre à partir du dioxyde de carbone; ces réactions chimiques se produisent en utilisant des pigments de chlorophylle à l'intérieur des chloroplastes de chaque cellule végétale.

La chlorophylle, lorsqu'elle est irradiée par la lumière du soleil, absorbe la lumière rouge et bleue. Les parties de couleur verte ne sont pas absorbées directement pour le processus de photosynthèse directement. Les plantes sont donc vertes.

La lumière LED à des fins de culture doit principalement fournir une lumière bleue et rouge du spectre d'absorption de la chlorophylle. C'est pourquoi nous utilisons les parties «R» et «B» du néopixel RGBW haute puissance LED. La partie verte de la LED n'est pas utilisée.

Mais une plante utilise également d'autres parties du spectre continu de la lumière solaire, le processus de photosynthèse est plus complexe et est un domaine de recherche en cours. En bref : la lumière verte plonge plus profondément dans une plante et rend le processus de photosynthèse plus efficace, car elle affecte le taux d'absorbance de la chlorophylle. Donc une petite quantité continue de spectre vert à jaune est nécessaire.

<https://academic.oup.com/pcp/article/50/4/684/1908367>

De plus, la croissance des plantes est affectée par les hormones végétales, qui réagissent également à la lumière du soleil et ont généralement besoin d'un spectre solaire continu. Par exemple, les phytochromes réagissent à l'éclairage infrarouge.

https://en.wikipedia.org/wiki/Plant_hormone

<https://en.wikipedia.org/wiki/Phytochrome>

En conséquence, une lampe de croissance optimale ne doit pas être limitée aux parties rouges et bleues du spectre, mais nécessite également une partie LED «blanche» qui produit un spectre continu blanc chaud pour affecter les systèmes photosynthétiques secondaires et les hormones végétales. C'est pourquoi nous utilisons la LED 4500K-RGBW d'Adafruit Industries.

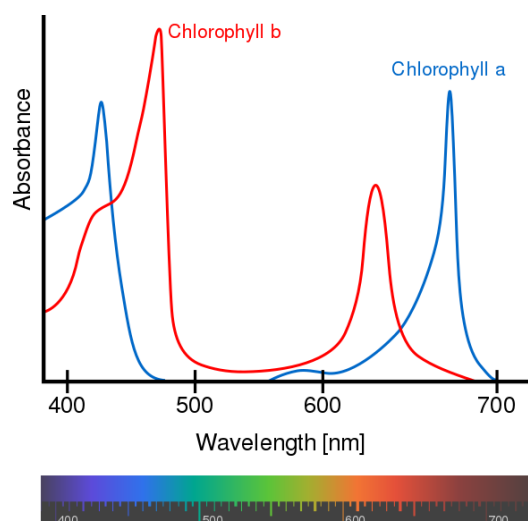


Figure 6: Adafruit neopixel, RGBW-LED,
<https://www.adafruit.com/product/2758>

Détection d'humidité du sol

L'humidité est généralement mesurée en pourcentage de l'humidité de l'air. Par conséquent, un capteur de température et d'humidité est nécessaire. L'humidité elle-même peut être mesurée de différentes manières et l'une des méthodes les plus courantes est la mesure capacitive. Le capteur lui-même est un condensateur dont la capacité est modifiée par l'absorption / la désorption de l'eau.

La capacité C d'un condensateur dépend de la surface de plaque A , de la distance entre les plaques d et du milieu diélectrique entre deux plaques métalliques avec une constante de permittivité donnée ϵ_R :

$$C = \epsilon_R \frac{A}{d}$$

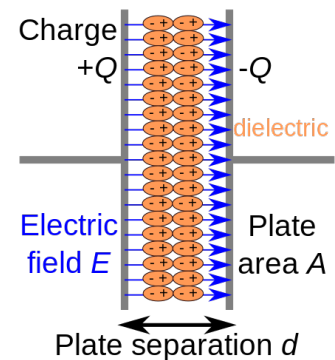


Figure 7: schéma d'un condensateur.
<https://en.wikipedia.org/wiki/Capacitor>

Bien que la distance et la taille des plaques ne puissent pas être modifiées par l'humidité, la constante de permittivité le fait. Habituellement, la permittivité d'un composé donné est comparé à la permittivité du vide parfait et donc est appelée «permittivité relative». Voici quelques valeurs importantes pour la permittivité relative

Milieu	Permittivité relative
Vide	1
Air	1.0006
Eau	80
Sol minéral sec	5

Par conséquent, la capacité du sol augmente à mesure qu'il contient plus d'eau. Les sols aqueux et humides ont une permittivité significativement plus élevée que leurs homologues secs. Habituellement, elle est mesurée en tant que «teneur en eau volumétrique du sol», SWC. Elle est définie comme la teneur en eau volumétrique:

$$SWC = \frac{\text{Volume d'eau}}{\text{Volume de sol}}$$

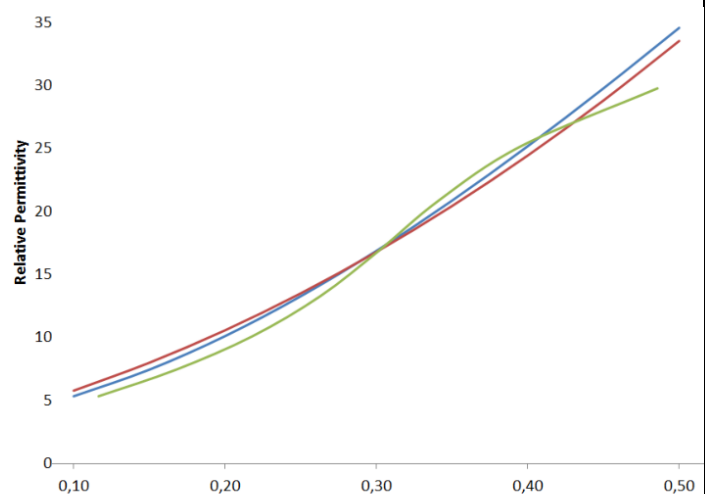




Figure 8: Porretta, Bianchi "Profiles of relative permittivity and electrical conductivity from unsaturated soil water content models", ANNALS OF GEOPHYSICS, 59, 3, 2016, G0320

Un circuit électronique pour mesurer ces changements de capacité est construit comme un circuit RC. En fonction de sa capacité, un circuit RC a une constante de temps caractéristique qui peut être mesurée par un microcontrôleur. Plus la capacité est importante, plus la constante de temps est longue. En résumé, l'humidité est mesurée de cette façon :

L'humidité augmente $\xrightarrow{\text{conduit à}}$ la capacitance augmente $\xrightarrow{\text{conduit à}}$ Constante de temps plus longue

Comparaison du SHT20 avec boîtier et d'un capteur d'humidité du sol:

SHT 20	Capteur d'humidité du sol
	
Mesure l'humidité de l'air à l'intérieur de son boîtier étanche et simultanément la température	Mesure directement la permittivité relative du sol. Une mesure de température supplémentaire est nécessaire.
Communique avec le microcontrôleur Arduino via I2C	Produit un signal analogique qui doit être numérisé avec l'ADC du microcontrôleur Arduino
Ne doit pas être complètement enterré car un flux d'air est nécessaire	Doit être maintenu aussi profondément que possible dans le sol
Coûte environ 20 Euro	Coûte environ 5 Euro
ATTENTION: Étant donné que ce capteur produit une humidité relative de l'air en sortie, les valeurs de mesure n'ont aucun sens pour les conditions du sol. La plupart du temps, le capteur produit des valeurs supérieures à 100% car l'air à l'intérieur du boîtier étanche est saturé d'humidité du sol. Vous devez utiliser les valeurs brutes 16 bits avant conversion en valeurs d'humidité de l'air qui sont sans signification.	ATTENTION: Puisque ce capteur mesure la permittivité relative du sol, il est absolument essentiel que le capteur soit en contact parfait avec le sol sans couche d'air entre les deux surfaces. De plus, les valeurs du capteur dérivent, car avec l'arrosage le volume du sol change et donc la surface de couverture du capteur change également. De plus, les plantes en croissance modifient également les valeurs de permittivité.

Exemple de code source Arduino 8c

```
#include <Adafruit_NeoPixel.h>
#include "Grove_Motor_Driver_TB6612FNG.h"
#include "DFRobot_SHT20.h"
#include <Wire.h>
#include <SoftwareSerial.h>

MotorDriver Energie;
DFRobot_SHT20 sht20;
Adafruit_NeoPixel GrowLED(16,6,NEO_RGBW);
SoftwareSerial Serial_89(8, 9);

unsigned long aktMillis = millis();
unsigned long readMillis = aktMillis;
unsigned long ledMillis = aktMillis;
unsigned long serialMillis = aktMillis;
unsigned long pumpMillis = aktMillis;
unsigned long Feuchtigkeit = 0;
unsigned long FeuchtSoll = 55300;
float Temperatur = 0;
int Giessdauer = 0;
int Helligkeit = 0;

void setup() {
  Wire.begin();
  Serial.begin(9600);
  Serial_89.begin(9600);
  Energie.init();
  GrowLED.begin();
  LEDsetzen();
  sht20.initSHT20();
  delay(100);
  sht20.checkSHT20();
  Energie.dcMotorRun(MOTOR_CHA, 255);
  Energie.dcMotorBrake(MOTOR_CHB);
}

void loop() {
  aktMillis = millis();

  if (aktMillis - serialMillis >= 1000) {
    while (Serial_89.available() > 0) {
      unsigned long test = Serial_89.parseInt();
      if (test > 0 && test < 1000) {
        wasserpumpen(10);
        serialMillis = aktMillis;
      }
      if (test > 1000) {
        FeuchtSoll = test;
      }
    }
  }

  if (aktMillis - ledMillis >= 60000) {
    LEDsetzen();
    ledMillis = aktMillis;
  }

  if (aktMillis - readMillis >= 5000) {
    Feuchtigkeit = sht20.readHumidityRaw();
    Temperatur = sht20.readTemperature();
    Serial_89komm();
    readMillis = aktMillis;
  }

  if (aktMillis - pumpMillis >= 300000) {
    if (Feuchtigkeit < FeuchtSoll) {
      wasserpumpen(5);
      pumpMillis = aktMillis;
    }
  }
}
```

```
void wasserpumpen(int Sekunden) {
  Giessdauer = Giessdauer + Sekunden;
  Serial_89komm();
  Energie.init();
  delay(10);
  Energie.dcMotorBrake(MOTOR_CHA);
  delay(10);
  Energie.dcMotorRun(MOTOR_CHB, -255);
  delay(100);
  Energie.dcMotorRun(MOTOR_CHB, 255);
  delay(Sekunden * 1000);
  Energie.dcMotorBrake(MOTOR_CHB);
  delay(10);
  Energie.dcMotorRun(MOTOR_CHA, 255);
  delay(10);
  LEDsetzen();
}

void LEDsetzen() {
  unsigned long Minuten=(aktMillis%8640)/6;
  int r = 0;
  int g = 0;
  int b = 0;
  int w = 0;
  if (Minuten < 840) {
    if (Minuten < 64) {
      r = Minuten * 4;
      g = 0;
      b = Minuten * 2;
      w = Minuten * 4;
    }
    if (Minuten >= 64 && Minuten <= 776) {
      r = 255 - (Minuten - 64) / 6;
      g = 0;
      b = 128 + (Minuten - 64) / 6;
      w = 255;
    }
    if (Minuten > 776) {
      r = 128 - (Minuten - 776) * 2;
      g = 0;
      b = 255 - (Minuten - 776) * 4;
      w = 255 - (Minuten - 776) * 4;
    }
  }
  Helligkeit = (int)(r + b + w) / 3;
  for (int i = 0; i < 16; i++) {
    GrowLED.setPixelColor(i,
      GrowLED.Color(g, r, b, w));
  }
  GrowLED.show();
}

void Serial_89komm() {
  Serial_89.print("Zeit: ");
  Serial_89.print(aktMillis / 1000);
  Serial_89.print(", fIst: ");
  Serial_89.print(Feuchtigkeit);
  Serial_89.print(", Soll: ");
  Serial_89.print(FeuchtSoll);
  Serial_89.print(", Temp: ");
  Serial_89.print(Temperatur);
  Serial_89.print(", Wass: ");
  Serial_89.print(GesamtGiessdauer);
  Serial_89.print(", Hell: ");
  Serial_89.println(Helligkeit);
}
```


8d: Application de traitement avec code source

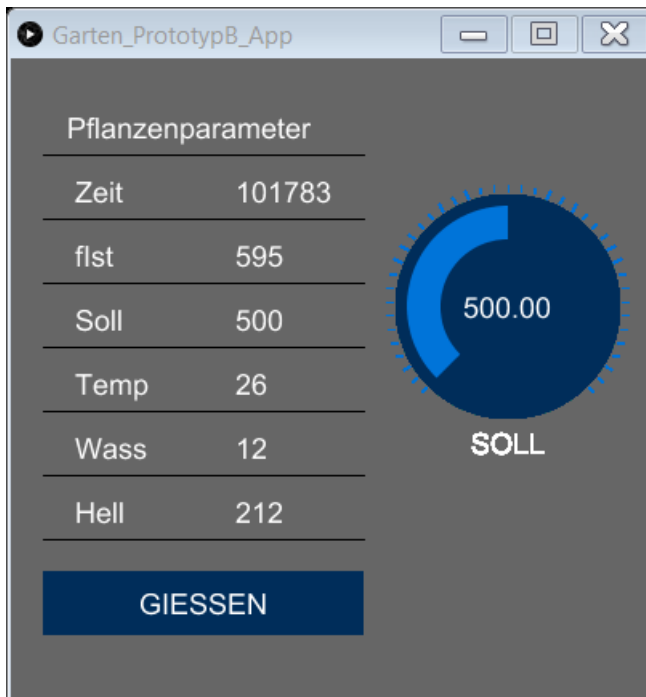


Figure 9: Captre d'écran de l'interface de l'application

Un appareil IoT typique possède généralement une interface d'application qui permet à l'utilisateur de surveiller et de contrôler à distance son appareil connecté.

Une façon (relativement) simple d'y parvenir avec les étudiants consiste à utiliser le traitement avec ses capacités pour lire les données série de l'XBee et dessiner ses valeurs dans une interface utilisateur graphique.

De plus, l'IDE Arduino et l'IDE de traitement sont étroitement liés, car l'IDE Arduino est une «branche» de l'IDE de traitement.

Pour le bouton «GIESSEN» et le bouton circulaire «SOLL» (ce qui signifie une valeur optimale pour l'humidité du sol »), la bibliothèque« ControlP5 »est utilisée et peut être facilement installée à partir de l'IDE de traitement.

Pour controlP5, voir: <https://code.google.com/archive/p/controlp5/downloads>

8c. Code source du traitement comme exemple

```
import controlP5.*;
import processing.serial.*;

Serial arduinoKommunikation;
String payload;
String[] liste;
ControlP5 cp5;
Knob sollFeuchte;
int sollFeuchteWert = 500;
int arduinoSollWert = 0;

void setup() {
  size(400, 400);
  background(102);
  smooth();
  String portName = Serial.list()[0];
  arduinoKommunikation = new Serial(this, portName, 9600);

  cp5 = new ControlP5(this);
  PFont font = createFont("arial", 18);
  textFont(font);
  cp5.setFont(font);

  sollFeuchte = cp5.addKnob("Soll")
    .setRange(300, 700)
    .setValue(sollFeuchteWert)
    .setPosition(240, 85)
    .setRadius(70)
    .setDragDirection(Knob.VERTICAL)
    .setNumberOfTickMarks(40)
    .setTickMarkLength(4)
    .snapToTickMarks(true)
    .onRelease(new CallbackListener() {
      public void controlEvent(CallbackEvent theEvent) {
        sollFeuchteWert= int(theEvent.getController().getValue());
      }
    });
}
```

```

    });

    cp5.addButton("giessen")
        .setValue(0)
        .setPosition(20, 320)
        .setSize(200, 40)
        ;
}

void draw() {
    if (arduinoKommunikation.available() > 0) {
        payload = arduinoKommunikation.readStringUntil('\n');
        if (payload != null) {
            background(102);
            text("Pflanzenparameter", 35, 50);
            line(20, 60, 220, 60);
            liste = split(payload, ",");
            for (int i = 0; i < liste.length; i++) {
                liste[i] = trim(liste[i]);
                String[] Groesse = split(liste[i].trim(), ":");
                text(Groesse[0], 40, 90+40*i);
                int val = parseInt(Groesse[1].trim());
                if (i==2) {
                    arduinoSollWert = val;
                }
                text(int(val), 140, 90+40*i);
                line(20, 100+40*i, 220, 100+40*i);
            }
            if (arduinoSollWert != sollFeuchteWert) {
                arduinoKommunikation.write(str(sollFeuchteWert));
            }
        }
    }
}

public void giessen() {
    if (millis() > 5000) {
        arduinoKommunikation.write(str(2));
    }
}

```

9.

**Description
étape par
étape de
l'activité /
contenu**

Leçons 1 & 2 (90min):

Les étudiants s'initient initiés à l'IoT par des exemples: robots aspirateur avec télécommandes d'application, stations météorologiques basées sur Internet, trackers d'activité avec communication via l'application et enfin, mais non des moindres, des fermes agricoles intelligentes. Les élèves devraient examiner le fonctionnement de ces appareils et les composants nécessaires: un système basé sur un microcontrôleur contrôle et coordonne les capteurs et les actionneurs connectés. De plus, il communique et se coordonne avec d'autres systèmes de type similaire souvent via des réseaux de communication sans fil.

Pièces nécessaires: capteurs, acteurs, appareils de communication.

Les possibilités et les risques doivent être discutées ainsi que les limites: où l'IoT a-t-il un sens et où n'en a-t-il pas?

Leçons 3 et 4 (90 min)

Les élèves doivent planifier les composants pour surveiller et optimiser la croissance des plantes. Ensuite, ils peuvent commencer à construire une telle machine à partir de zéro en utilisant les pièces fournies. Une fois terminé, des techniques de programmation de base peuvent être enseignées pour permettre aux étudiants de faire leurs propres expériences.

Leçons 5 et 6 (90 min)

La théorie de la mesure de l'humidité du sol et de l'éclairage des plantes doit être enseignée. Les élèves peuvent faire des mesures de SWC avec leurs capteurs individuels pour faire des courbes d'étalonnage. Les élèves doivent comparer leurs résultats expérimentaux pour se rendre compte que chaque groupe d'élèves a ses propres valeurs de mesure qui diffèrent considérablement.

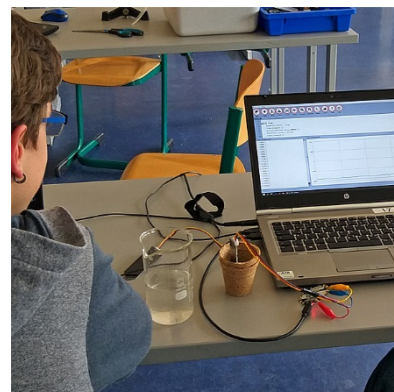


Figure 10: expérimentation avec des capteurs de sol

Leçons 7 et 8 (90 min):

Les élèves devraient commencer à programmer et à contrôler les pompes péristaltiques avec le pilote de moteur. La théorie des ponts en H doit être enseignée et, en outre, les bases de la communication I2C entre les composants des appareils électroniques doivent être expliquées. Les élèves doivent mesurer la communication I2C avec les oscilloscopes. Le concept de PWM (Pulse width modulation) doit être introduit. Les étudiants peuvent en outre mesurer la communication entre les LED Arduino et Neopixel via Oscilloscope.

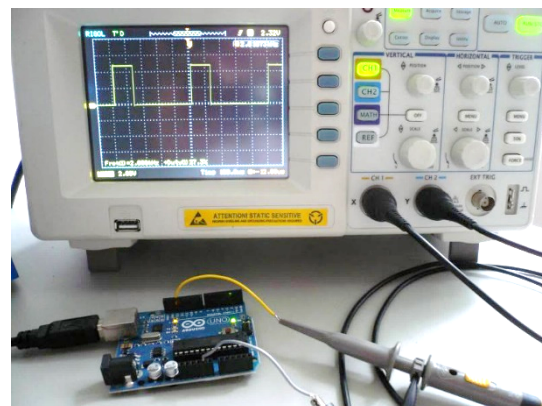


Figure 13: mesure PWM avec oscilloscope

Leçons 9 et 10 (90 min):

Construire un réseau de communication: modules XBee et communication série (UART). Les étudiants doivent utiliser le logiciel XCTU comme point de départ pour leurs expériences de communication dans les réseaux sans fil.

ATTENTION: XBees doit être préconfiguré par paires par l'enseignant, car sinon, beaucoup de temps sera perdu à apprendre à gérer les nombreuses configurations différentes. Fondamentalement, les paires XBee sont définies à l'aide de trois valeurs différentes qui sont encadrées ici en rouge:

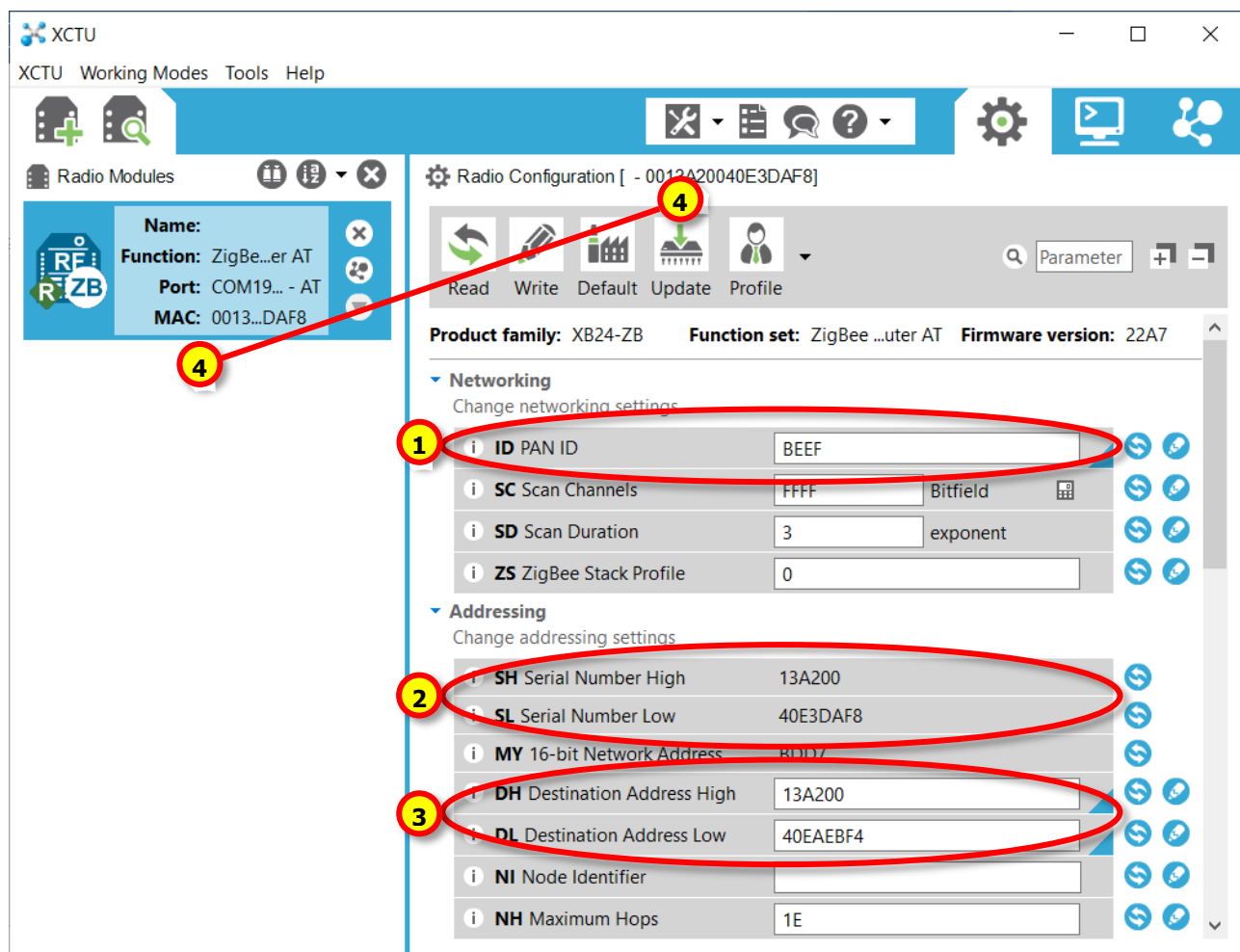


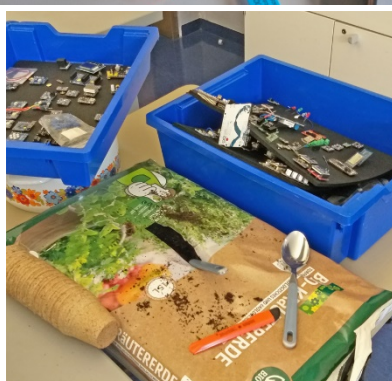
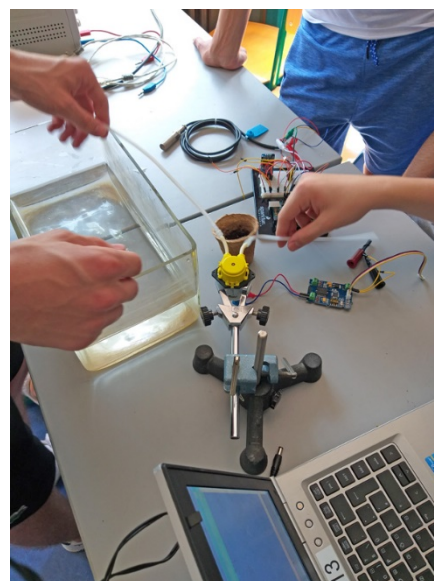
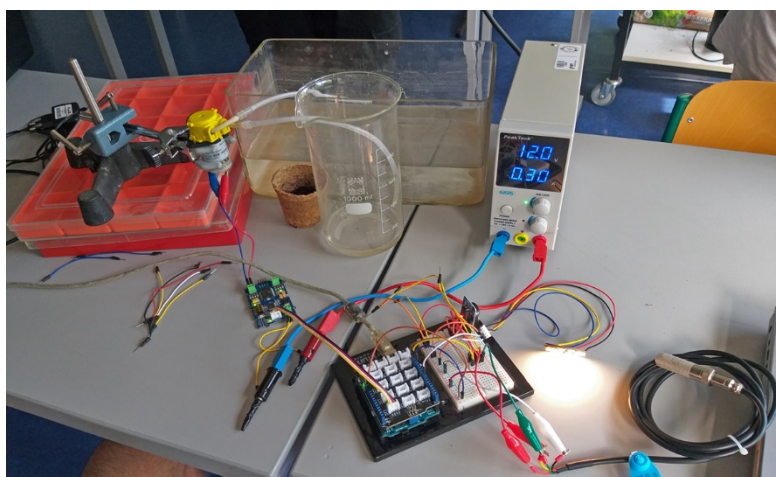
Figure 11: Copie d'écran de l'interface utilisateur de XCTU

- 1: L'ID PAN doit être le même pour les deux XBees. Utilisez des lettres hexadécimales, par ex. soi-disant «Hexspeak»: «BEEF», «CAFE», «F00D», «AFFE», etc.
 2. Le numéro de série élevé doit être copié à partir d'un XBee....
 - 3.... sur le second.
 4. Un XBee doit être configuré en tant que «coordinateur» et l'autre en tant que «routeur».
- Attention: Dans certains textes, vous pouvez lire que le second doit être configuré comme «Endpoint». Cela peut probablement produire certains problèmes, car les endpoint-XBees se mettent en veille pour des raisons d'économie d'énergie.

Après cela, XBees peut être placé sur le port de communication UART de l'arduino.

Leçons 11 pour terminer (270 min):

Programmation freestyle! Et bonne récolte 😊



10. Retour d'information

À la fin de la leçon, les étudiants devraient avoir une connaissance approfondie du fonctionnement des principes de l'IoT et de la façon dont les machines connectées à Internet communiquent. Ils ont expérimenté par eux-mêmes les avantages et les limites de la technologie actuelle. Pendant la leçon, des aspects importants de l'électronique, de l'informatique et des bases de la construction ont été enseignés. De plus, les aspects biologiques de la culture des plantes ont été enseignés.

11. Evaluation

Les étudiants tiennent leur journal du travail, qui peut être relu par l'enseignant. Les étudiants peuvent également présenter les résultats de leurs expériences. De plus, un test standard en classe doit être effectué à la fin des cours.