

## **'We are the makers - IoT' Learning Scenario: agricoltura intelligente con un IoT-plantrobot**

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

*Il seguente documento è stato sviluppato e testato in un ambiente scolastico con circa 18 studenti di 13-17 anni nell'anno scolastico 2018/2019. Esso riflette l'esperienza con molti meandri e alcuni fallimenti. Poiché il campo dell'internet degli oggetti è complesso, il materiale didattico deve essere scelto con cura. Questo documento dovrebbe essere una raccomandazione, come punto di partenza.*

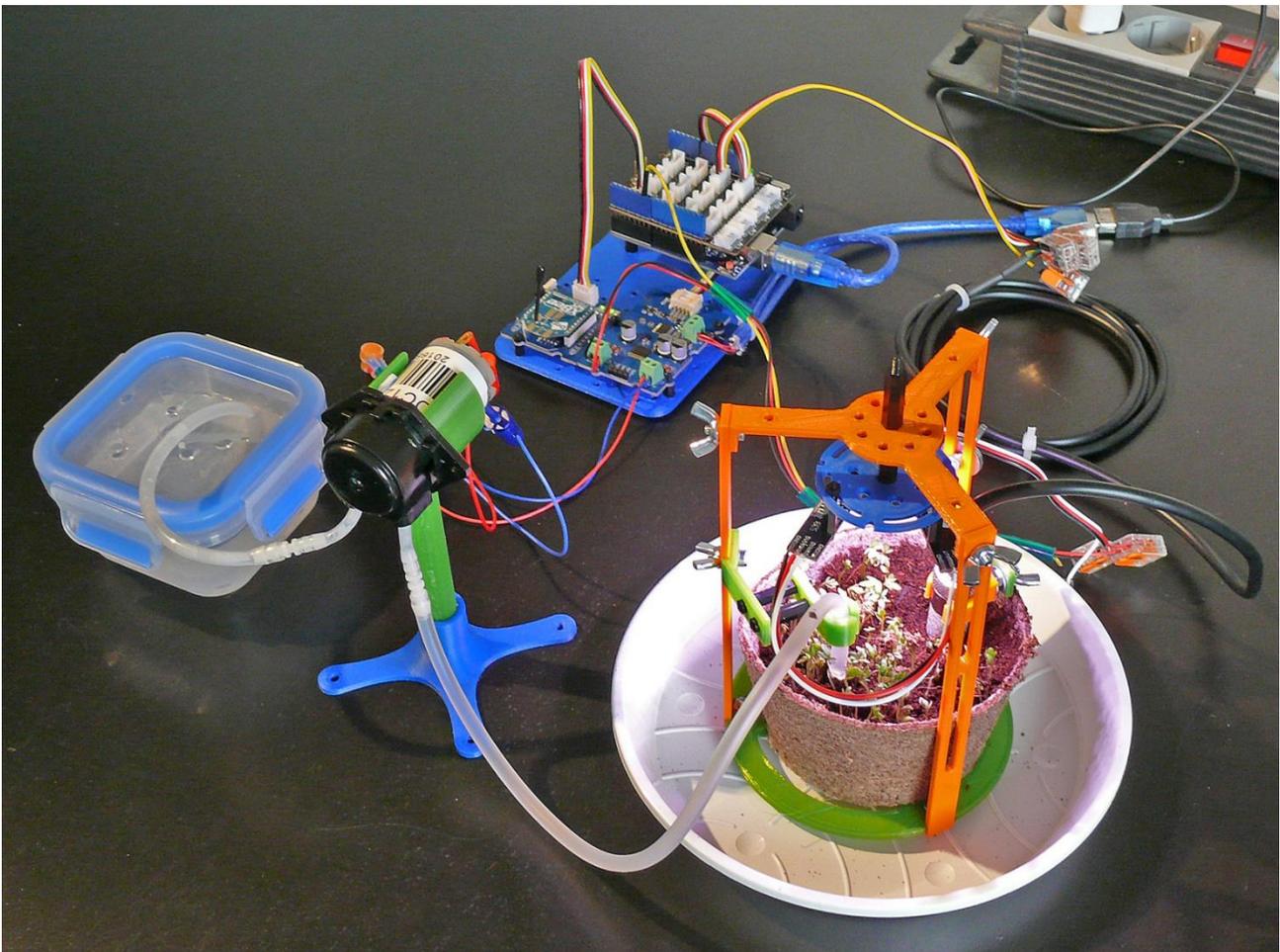


Figura 1: Prototipo di un robot per la coltivazione dell'internet degli oggetti

<b>1. Titolo dello scenario</b>	<b>Imparare a far crescere le piante con l'aiuto di un IoT-plantrobot</b>
<b>2. Gruppo target</b>	14 - 17 anni
<b>3. Durata</b>	Almeno 5 settimane di 2*45min-lessons a settimana: in totale circa 6-8 ore.
<b>4. Esigenze di apprendimento coperte attraverso l'esercizio</b>	<ul style="list-style-type: none"> <li>▪ Interazione tra parti elettroniche e creature (qui: piante)</li> <li>▪ Monitoraggio e influenza dei parametri biologici</li> <li>▪ Catena di comunicazione dei dispositivi IoT</li> <li>▪ Principi di sensori e attori</li> <li>▪ Diversi principi di misurazione dell'umidità nel terreno.</li> <li>▪ Principi dell'illuminazione a LED per la coltivazione delle piante</li> <li>▪ Regolazione fine dei parametri della macchina per ottimizzare la coltivazione delle piante</li> <li>▪ Principi delle reti di comunicazione wireless</li> <li>▪ Costruzione e stampa 3D di un ambiente robotizzato</li> </ul>
<b>5. Risultati attesi dell'apprendimento</b>	<ul style="list-style-type: none"> <li>▪ Come funziona un sistema IoT?</li> <li>▪ Dove sono le possibilità e i limiti dei sistemi IoT?</li> <li>▪ Quali componenti - hardware e software - sono fondamentali per costruire un dispositivo IoT?</li> <li>▪ Come costruire le regole per il biomonitoraggio e per influenzare gli esseri viventi?</li> </ul>
<b>6. Metodologie</b>	In questo scenario gli studenti costruiranno, costruiranno e programmeranno da soli un dispositivo per la coltivazione delle piante completamente interattivo. Gli studenti costruiranno anche un'applicazione per il controllo remoto dell'IoT-plantrobot
<b>7. Posto/ Ambiente</b>	<ul style="list-style-type: none"> <li>▪ un laboratorio con un set di parti e componenti elettronici;</li> <li>▪ ogni gruppo di studenti deve avere un computer o un portatile con privilegi amministrativi per l'installazione di diversi pacchetti software</li> <li>▪ Un proiettore per l'insegnamento delle lezioni e la presentazione dei lavori degli studenti;</li> <li>▪ ogni studente deve tenere un diario di laboratorio</li> </ul>

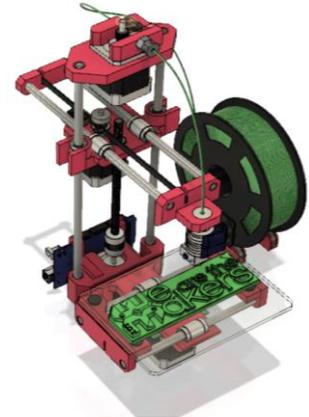
**8.Strumenti/  
Materiali/  
Risorse**

Stampanti-3D

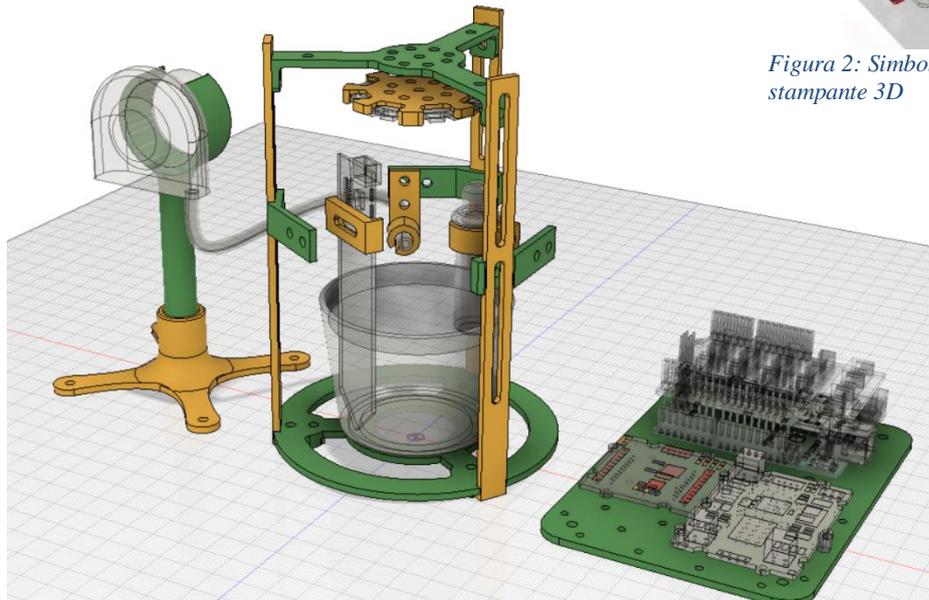
Circa 3-4 stampanti 3D sono necessarie poiché gli studenti stamperanno il loro IoT-plantrobot da se stessi.

Componenti stampati 3d:

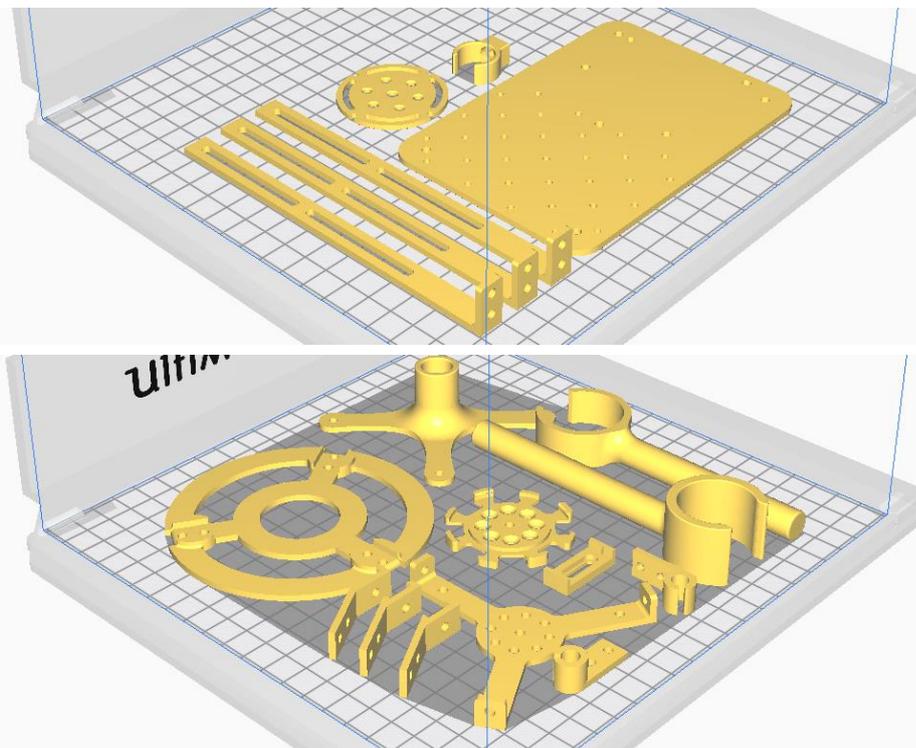
Come punto di partenza tutte le parti necessarie sono fornite in formato .stl e come file Autodesk Fusion 360.



*Figura 2: Simbolo della stampante 3D*



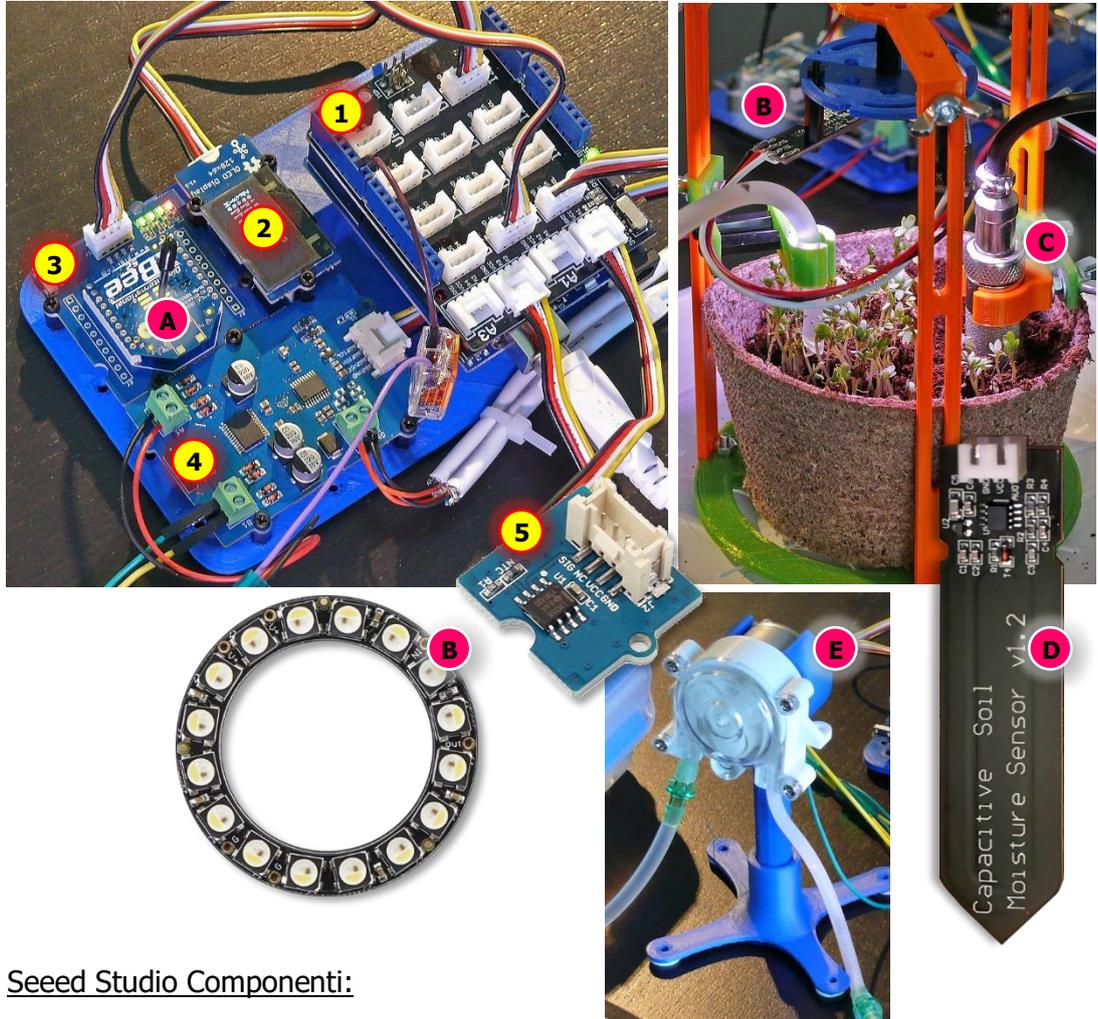
*Figura 3: Panoramica dei dati CAD*



*Figura 1: set completo di dati 3D su una stampante 3D 20cm x 20cm*

Componenti elettroniche:

In questo lavoro, raccomandiamo il sistema seeed grove perché è facile da usare: ([http://wiki.seeedstudio.com/Grove\\_System/](http://wiki.seeedstudio.com/Grove_System/)) Tutti i componenti principali, tranne XBees, sensori di umidità e illuminazione a LED, appartengono allo standard del boschetto:



Seeed Studio Componenti:

- 1: Scudo di base del boschetto per Arduino-Uno
- 2. Boschetto OLED 128x64
- 3. 3. Ape presa Grove Bee
- 4. 5. Grove - Driver motore I2C (TB6612FNG)
- 5. Sensore di temperatura del boschetto v1.2

Sensori e attori regolari:

- 1: Arduino Uno (o equivalente)
- A: XBee Serie 2C o Serie 2
- B: Adafruit Neopixel-ring con 16 RGBW (NON RGB!) a 4500 K (bianco caldo)
- C: SHT20-Sensore di temperatura e umidità in custodia impermeabile o
- D: Sensore di umidità del suolo analogico capacitivo

E: Pompa peristaltica con motore a 6V DC

Parti varie:

- Tubo in silicone da 5-6 mm (per acquari)
- Adattatore per il collegamento di tubi in silicone con pompa peristaltica
- Viti M3 e dadi a farfalla
- Distanziatori M3 in nylon (distanziatore esagonale)
- Distanziatori M2 in nylon (distanziatore esagonale) per Grove (ha fori da 2 mm)
- Fili per boschetti
- Dadi a leva WAGO
- Fili ponte
- Vasi da vivaio con diametro 8cm
- Alimentazione USB con corrente massima 2-2,5A
- Adattatore USB XBee (e.g. <https://www.waveshare.com/xbee-usb-adapter.htm>)



*Figura 5: Distanziatori in nylon*

Piante:

Adatti a fare esperimenti a scuola sono piante a crescita rapida che sono chiamati "Microgreens" / "Microgreen Sprouts"; sono non tossico e mangiabile:

- Crescione da giardino
- Mung Beans
- Ravanello a stelo rosso
- Germogli di trifoglio rosso
- Germogli di broccoli
- Valerianella locusta ("Vit" insalata da campo)



*Figura 6: Crescione da giardino*

computer con il seguente software preinstallato:

Autodesk Fusion 360 (o qualsiasi altro software di modellazione 3D, ad esempio Wings3D)

Software di slicing CURA,

Una connessione internet per scaricare le biblioteche

Arduino IDE

Elaborazione IDE

Software XCTU per la configurazione di XBees

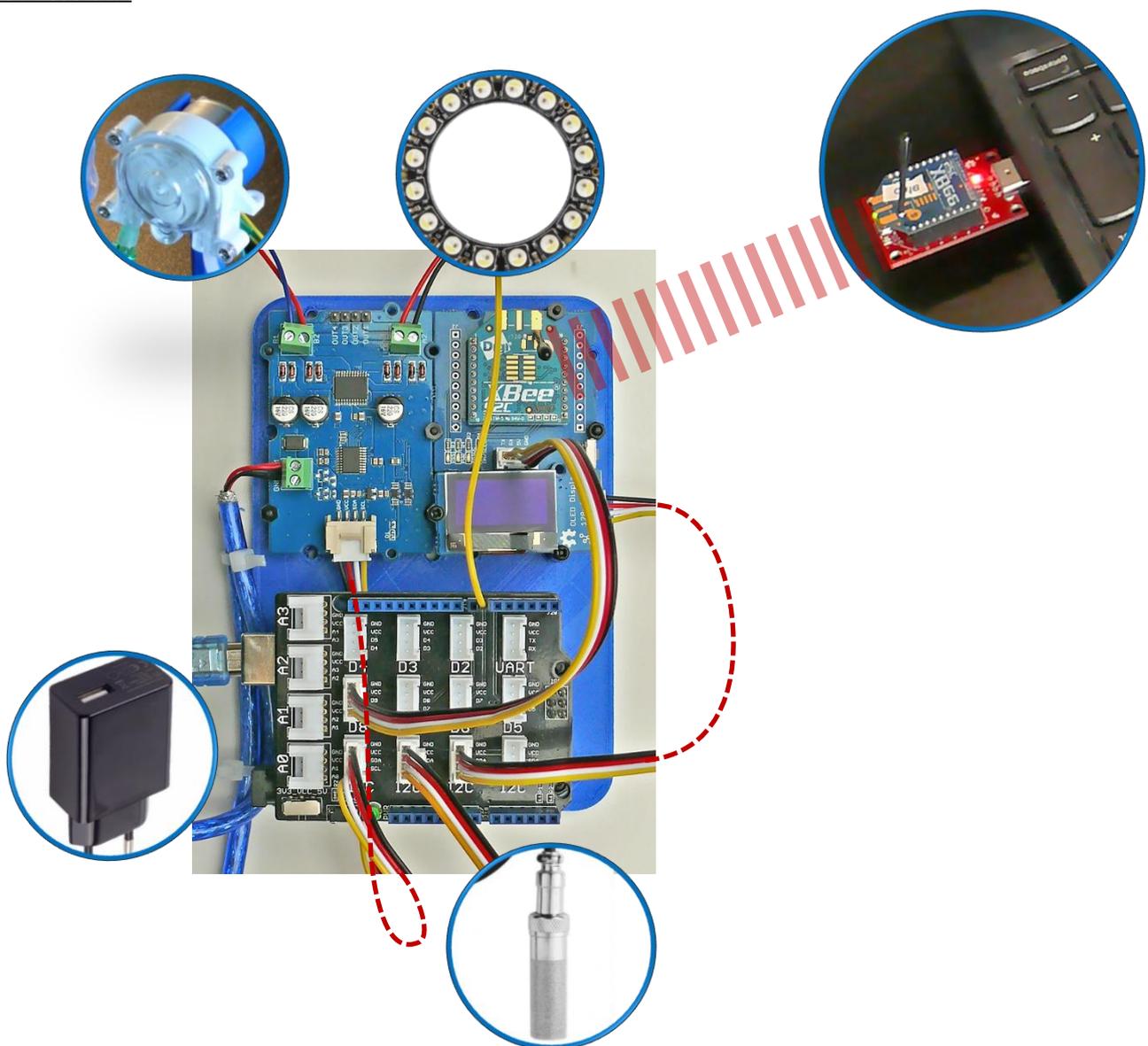
Librerie Arduino per le componenti:

Alcuni component come il sensore di umidità SHT20 o il motore hanno bisogno di librerie per l'Arduino IDE per lavorare in modo opportuno. A questo link è descritto come importare una libreria: <https://www.arduino.cc/en/Guide/Libraries>

- SHT20 lib (DF Robot): [https://codeload.github.com/DFRobot/DFRobot\\_SHT20/zip/master](https://codeload.github.com/DFRobot/DFRobot_SHT20/zip/master)
- OLED lib (Seeed): [https://github.com/Seeed-Studio/OLED\\_Display\\_128X64/archive/master.zip](https://github.com/Seeed-Studio/OLED_Display_128X64/archive/master.zip)
- Motor driver (Seeed): [https://github.com/Seeed-Studio/Grove\\_Motor\\_Driver\\_TB6612FNG](https://github.com/Seeed-Studio/Grove_Motor_Driver_TB6612FNG)
- Neopixel (Adafruit): [https://github.com/adafruit/Adafruit\\_NeoPixel/archive/master.zip](https://github.com/adafruit/Adafruit_NeoPixel/archive/master.zip)

Qui si trovano i codici speciali per il sensore di temperatura:  
[http://wiki.seeedstudio.com/Grove-Temperature\\_Sensor\\_V1.2/](http://wiki.seeedstudio.com/Grove-Temperature_Sensor_V1.2/)

Connessioni:



## A8b Alcune teorie sui fari da coltivazione a LED e sulla misurazione dell'umidità del suolo

### Illuminazione a LED

Alla base dell'utilizzo dei LED per la coltivazione delle piante c'è la teoria del PAR, "radiazione fotosinteticamente attiva": Le piante usano fotoni leggeri per reazioni chimiche per costruire lo zucchero dall'anidride carbonica; queste reazioni chimiche avvengono usando pigmenti clorofillici all'interno dei cloroplasti di ogni cellula della pianta.

La clorofilla, se irradiata con la luce del sole, assorbe la luce rossa e blu. Le parti di colore verde non vengono assorbite direttamente per il processo fotosintetico. Pertanto le piante sono verdi.

La luce LED per la coltivazione delle piante deve fornire principalmente luce blu e rossa dello spettro di assorbimento della clorofilla. Per questo motivo usiamo le parti "R" e "B" di neopixel RGBW ad alta potenza LED. La parte verde del LED non viene utilizzata.

Ma una pianta utilizza anche altre parti della luce solare continua spettro, il processo fotosintetico è più complesso ed è un campo di ricerca attuale. In breve: luce verde si immerge più in profondità in un impianto e rende il processo fotosintetico più efficiente, poiché influisce sul tasso di assorbimento della clorofilla. Pertanto piccolo

sono necessarie quantità di spettro continuo da verde a giallo.

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

Inoltre, la coltivazione delle piante è influenzata dagli ormoni delle piante, che reagiscono anche alla luce del sole e necessitano per lo più di uno spettro solare continuo. Ad esempio, i fitocromi reagiscono all'illuminazione a infrarossi.

[https://en.wikipedia.org/wiki/Plant\\_hormone](https://en.wikipedia.org/wiki/Plant_hormone)<https://en.wikipedia.org/wiki/Phytochrome>

Di conseguenza, una luce di crescita ottimale non deve essere limitata alle parti rosse e blu dello spettro, ma necessita anche di una parte a LED "bianca" che produca uno spettro continuo bianco caldo per influenzare i sistemi fotosintetici secondari e gli ormoni delle piante. Per questo motivo utilizziamo il LED 4500K-RGBW di Adafruit Industries.

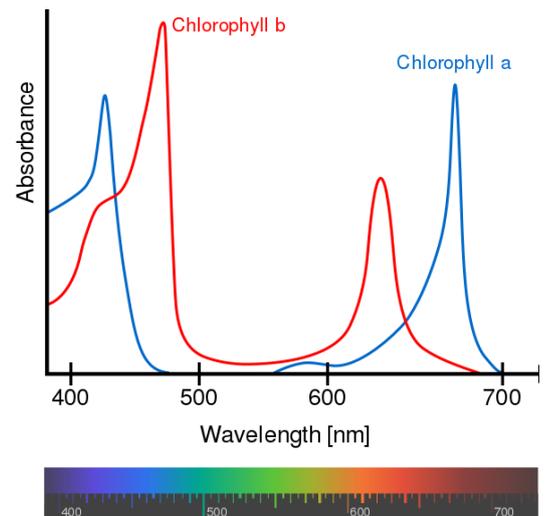


Figura 7: Lo spettro di assorbimento dei pigmenti della clorofilla a e della clorofilla b.  
[https://en.wikipedia.org/wiki/Chlorophyll\\_b](https://en.wikipedia.org/wiki/Chlorophyll_b)



Figura 8: Adafruit neopixel, RGBW-LED,  
<https://www.adafruit.com/product/2758>

## Rilevamento dell'umidità del suolo

Di solito l'umidità viene misurata come percentuale dell'umidità dell'aria. Pertanto è necessario un sensore di temperatura e di umidità. L'umidità stessa può essere misurata in diversi modi e uno dei metodi più comuni è la misura capacitiva. Il sensore stesso è un condensatore la cui capacità viene modificata con l'assorbimento / desorbimento dell'acqua.

La capacità  $C$  di un condensatore dipende dalla zona della piastra  $A$ , dalla distanza tra le piastre  $d$  e il mezzo dielettrico tra due piastre metalliche con una data costante di permittività  $\epsilon_R$ :

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

Mentre la distanza e le dimensioni delle piastre non possono essere modificate con l'umidità, la costante di permittività lo fa. Di solito la permittività di un determinato composto viene confrontato con la permittività del vuoto perfetto e quindi è chiamata "permittività relativa". Ecco alcuni valori importanti per la permittività relativa:

Medio	Permittività relativa
Vuoto	1
Aria	1.0006
Acqua	80
Terreno secco minerale	5

Di conseguenza, più acqua contiene, più aumenta la capacità del suolo. Il suolo acquoso e umido ha una capacità significativamente più elevata rispetto alla sua controparte secca. Di solito si misura come il cosiddetto "contenuto volumetrico di acqua del suolo", SWC. Viene definito come contenuto d'acqua volumetrico:

$$SWC = \frac{\text{Volume of water}}{\text{Volume of soil}}$$

Un circuito elettronico per misurare queste variazioni di capacità è costruito come un circuito RC. A seconda della sua capacità un circuito RC ha una caratteristica costante nel tempo che può essere misurata da un microcontrollore. Più è lunga la capacità, più è lunga la costante di tempo. In sintesi, l'umidità viene misurata in questo modo:

increasing Humidity  $\xrightarrow{\text{leads to}}$  increasing capacitance  $\xrightarrow{\text{leads to}}$  increasing time constant

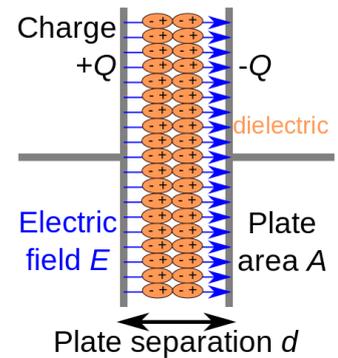


Figura 9: schema di un condensatore.  
<https://en.wikipedia.org/wiki/Capacitor>

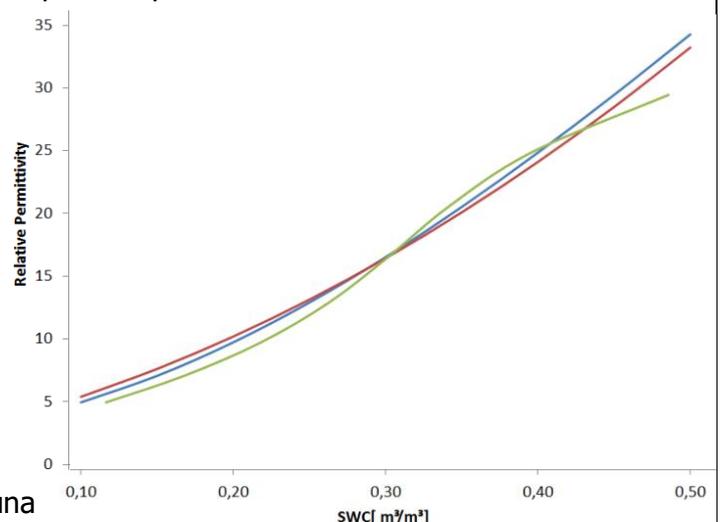


Figura 10: Porretta, Bianchi "Profili di permittività relativa e conducibilità elettrica da modelli di acqua insatura del suolo", ANNALI DI GEOFISICA, 59, 3, 2016, G0320

Confronto SHT20 con custodia e sensore di umidità del suolo:

SHT 20	Sensore di umidità del suolo
	
Misura l'umidità dell'aria all'interno del suo involucro impermeabile e contemporaneamente la temperatura	Misura direttamente la relativa permittività del suolo. È necessaria un'ulteriore misurazione della temperatura.
Comunica via I2C con il microcontrollore Arduino	Produce un segnale analogico che deve essere digitalizzato con l'ADC del microcontrollore Arduino
Non deve essere messo completamente nel terreno, poiché è necessario un flusso d'aria	Deve essere tenuto il più possibile in profondità nel terreno
Costo circa 20 Euro	Costi circa 5 Euro
ATTENZIONE: Poiché questo sensore produce umidità relativa dell'aria in uscita, i valori di misura non hanno alcun senso per le condizioni del suolo. Il più delle volte il sensore produce valori superiori al 100% poiché l'aria all'interno dell'involucro impermeabile è satura di umidità del suolo. È necessario utilizzare i valori grezzi a 16 bit che non vengono convertiti in valori di umidità dell'aria non significativi.	ATTENZIONE: Poiché questo sensore misura la permittività relativa del terreno, è assolutamente indispensabile che il sensore sia in perfetto contatto con il terreno senza strato d'aria tra le due superfici.

## 8c Codice Arduino di esempio

```
#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: Elaborazione dell'applicazione con sourcecode



Figura 11: Schermata dell'interfaccia dell'applicazione

Un tipico dispositivo IoT ha tipicamente un'interfaccia App che permette all'utente di monitorare e controllare a distanza il dispositivo collegato.

Un modo (relativamente) semplice per ottenere questo risultato con gli studenti è quello di usare l'elaborazione con le sue capacità per leggere i dati seriali dall'XBee e disegnare i suoi valori su un'interfaccia utente grafica.

Inoltre, l'IDE di Arduino e l'IDE di elaborazione sono strettamente correlati tra loro poiché l'IDE di Arduino è una 'Forchetta' di elaborazione.

Per il pulsante "GIESSEN" e la manopola circolare "SOLL" (che significa valore ottimale per l'umidità del suolo) si usa la libreria "ControlP5" che può essere facilmente installata dall'interno dell'IDE di elaborazione.

Per il controlloP5 fare riferimento a: <https://code.google.com/archive/p/controlp5/downloads>

## 8c. Elaborazione del codice sorgente come esempio

```
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.**

**Piano delle lezioni**

**Descrizione**

**passo dopo**

**passo**

**dell'attività/contenuto**

**Lezione 1 & 2 (90min):**

Gli studenti saranno introdotti con esempi di IoT: Robot sottovuoto con app remote, stazioni meteorologiche basate su internet, tracciatori di attività con app di comunicazione e, ultimo ma non meno importante, le aziende agricole intelligenti. Gli studenti dovrebbero esaminare come funzionano questi dispositivi e quali componenti sono necessari: un sistema basato su microcontrollore controlla e coordina i sensori e gli attori collegati. Inoltre comunica e coordina con altri sistemi di tipo simile spesso tramite reti di comunicazione wireless. Parti necessarie: Sensori, attori, dispositivi di comunicazione.

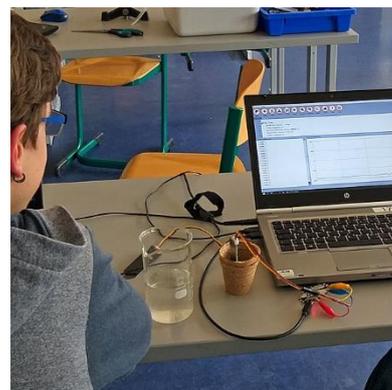
Occorre discutere delle possibilità e delle minacce e anche delle limitazioni: dove ha senso l'Internet degli oggetti e dove non ha senso?

**Lezione 3&4 (90 min)**

Gli studenti dovrebbero pianificare i componenti per il monitoraggio e l'ottimizzazione della coltivazione delle piante. In seguito possono iniziare a costruire una macchina di questo tipo da zero utilizzando le parti fornite. Una volta terminato, si possono insegnare le tecniche di programmazione di base per consentire agli studenti di fare i propri esperimenti.

**Lezioni 5&6 (90 min)**

La teoria della misurazione dell'umidità del suolo e dell'illuminazione delle piante deve essere insegnata. Gli studenti possono effettuare misurazioni di SWC con i loro sensori individuali per realizzare curve di calibrazione. Gli studenti devono confrontare i loro risultati sperimentali per rendersi conto che ogni gruppo di studenti ha i propri valori di misurazione che differiscono in modo significativo.



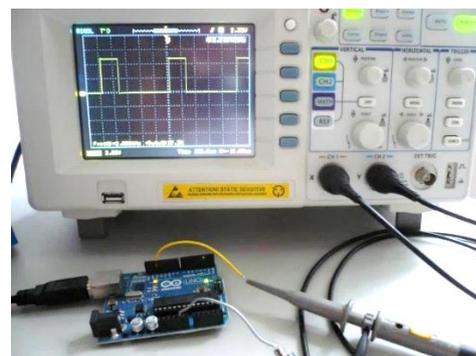
*Figura 12: sperimentazione con i sensori del suolo*

**Lezioni 7&8 (90 min):**

Gli studenti dovrebbero iniziare a programmare e controllare le pompe peristaltiche con il driver del motore. Deve essere insegnata la teoria dei ponti H, e inoltre devono essere spiegate le basi della comunicazione I2C tra i componenti dei dispositivi elettronici.

Gli studenti devono misurare la comunicazione I2C con gli oscilloscopi.

Deve essere introdotto il concetto di PWM (Pulse Width Modulation). Gli studenti possono inoltre misurare la comunicazione tra i LED Arduino e i LED Neopixel tramite Oscilloscopio.



*Figura 13: Misurazione PWM con oscilloscopio*

## Lezioni 9&10 (90 min):

Costruire una rete di comunicazione: Moduli XBee e comunicazione seriale (UART). Gli studenti dovrebbero usare il software XCTU come punto di partenza per i loro esperimenti di comunicazione in reti wireless.

**ATTENZIONE:** XBees dovrebbe essere preconfigurato a coppie dall'insegnante, altrimenti si perde molto tempo imparando a gestire le diverse configurazioni. Fondamentalmente, le XBee-Pairs sono definite usando tre diversi valori basati sull'utente che sono incorniciati qui in rosso:

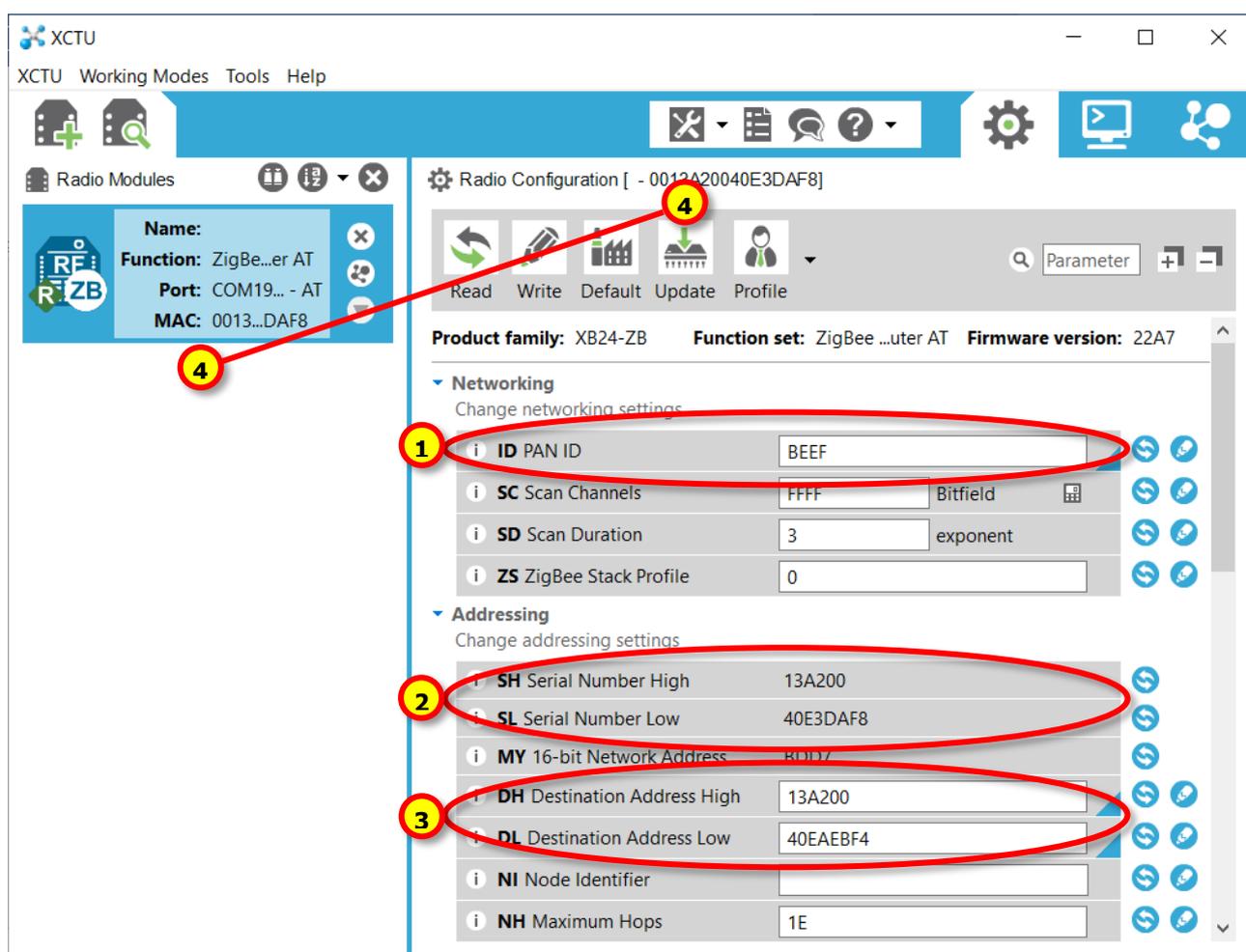


Figura 14: Schermata dell'interfaccia utente XCTU

1: L'ID PAN deve essere lo stesso per entrambi gli XBees. Utilizzare lettere esadecimali, ad esempio il cosiddetto "Hexspeak": "BEEF", "CAFE", "F00D", "AFFE", ecc.

2. Il numero di serie High deve essere copiato da un XBee .....

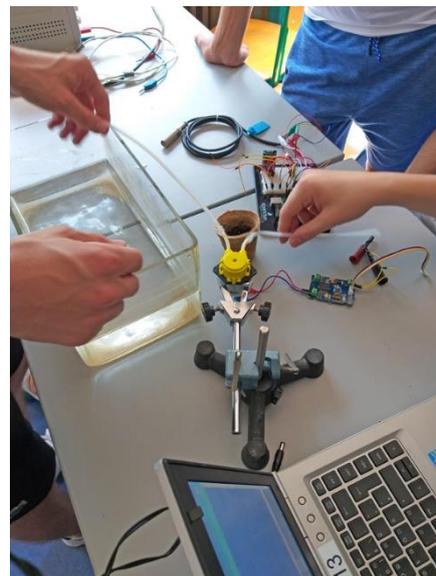
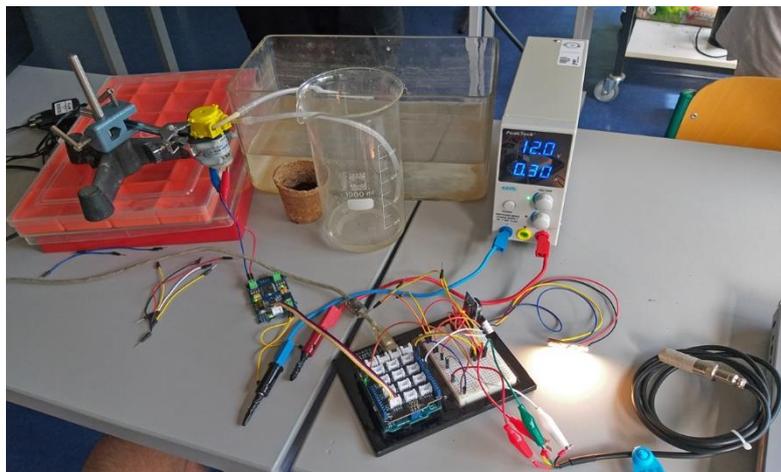
3. ... al secondo.

4. 4. Un XBee deve essere configurato come 'Coordinatore' e l'altro come 'Router'. Attenzione: In alcuni testi si legge che il secondo deve essere configurato come 'Endpoint'. Questo può probabilmente produrre alcuni problemi, dato che gli Endpoint-XBees vanno a dormire per motivi di risparmio energetico.

Dopo di che, XBees può essere messo sulla porta di comunicazione UART dell'arduino.

**Lezione 11 e fine (270 min):**

**Programmazione freestyle! E buon raccolto 😊**



<p><b>10. Feedback</b></p>	<p>Alla fine della lezione, gli studenti dovrebbero avere una conoscenza approfondita di come funzionano i principi dell'Internet degli oggetti e di come le macchine collegate a Internet comunicano. Hanno sperimentato da soli le possibilità e i limiti della tecnologia attuale. Durante la lezione, sono stati insegnati importanti aspetti dell'elettronica, dell'informatica e delle basi della costruzione. Inoltre, sono stati insegnati gli aspetti biologici della coltivazione delle piante.</p>
<p><b>11. Valutazioni</b></p>	<p>Gli studenti tengono il loro diario del lavoro, che può essere rivisto dall'insegnante. Gli studenti possono anche presentare i risultati dei loro esperimenti. Inoltre, alla fine delle lezioni deve essere effettuato un test standard in classe.</p>