

Scenariu de învățare 'We are the makers - IoT': Agricultură inteligentă cu un robot IoT pentru plante

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

Această lucrare a fost dezvoltată și testată în școală cu circa 18 elevi cu vârste cuprinse între 13 și 17 ani, în anul școlar 2018/2019. Ea reflectă experiența avută cu numeroase meandre și unele eșecuri. Deoarece domeniul IoT este unul foarte complex, materialele de predare trebuie alese cu mare grijă. Această lucrare se dorește o recomandare, un punct de plecare.

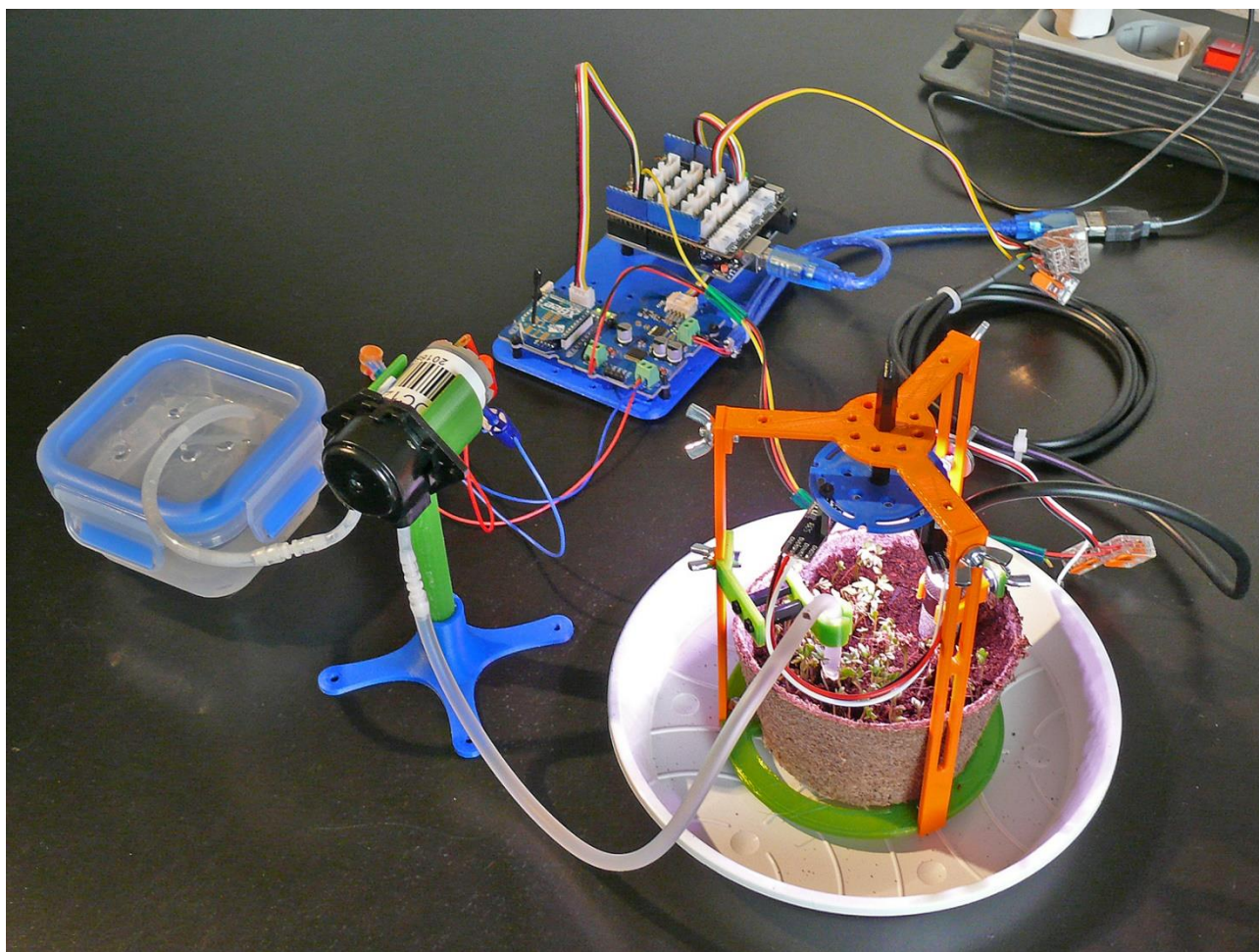


Figura 1: Prototipul unui robot IoT pentru creșterea plantelor

1. Titlu	Cum să crești plante cu ajutorul unui robot IoT pentru plante
2. Grup țintă	14 - 17 ani
3. Durată	Minim 5 săptămâni cu câte 2*45 min - lecții pe săptămână: în total circa 6-8 ore.
4. Nevoile de învățare	<ul style="list-style-type: none"> ▪ Interacțiunea dintre componentele electronice și vietăți (în acest caz: plante) ▪ Monitorizarea și afectarea parametrilor biologici ▪ Lanțul de comunicare al dispozitivelor IoT ▪ Principiile senzorilor și actuatorilor ▪ Diferite principii de măsurare a umidității din sol ▪ Principiile iluminării cu LED a plantelor în creștere ▪ Reglarea fină a parametrilor mașinii pentru optimizarea cultivării plantelor ▪ Principiile rețelelor de comunicații fără fir ▪ Construcția și imprimarea 3D a unui mediu robotizat
5. Rezultatele învățării	<ul style="list-style-type: none"> ▪ Cum funcționează un sistem IoT? ▪ Unde sunt posibilitățile și limitările sistemelor IoT? ▪ Ce componente - hard și software - sunt esențiale pentru construirea unui dispozitiv IoT? ▪ Cum se construiesc regulile de biomonitorizare și influențare a viețuitoarelor?
6. Metodologie	În acest scenariu, elevii vor construi și programa singuri un dispozitiv de creștere a plantelor complet interactiv, de la zero. Elevii vor construi, de asemenea, o aplicație pentru controlul la distanță al robotului IoT pentru plante.
7. Locație / Mediu	<ul style="list-style-type: none"> ▪ un laborator cu un set de piese și componente electronice; ▪ fiecare grup de elevi trebuie să aibă un calculator sau laptop cu drepturi de administrator pentru instalarea diferitelor pachete software; ▪ Proiectoare pentru prezentarea pașilor și prezentarea lucrărilor elevilor; ▪ fiecare elev trebuie să țină un jurnal de laborator

8. Instrumente / Materiale / Resurse

Imprimante 3D

Sunt necesare circa 3-4 imprimante 3D întrucât elevii își vor tipări singuri roboții IoT pentru plante.

Componente 3d tipărite:

Ca punct de plecare, toate părțile necesare sunt date în format .stl și ca fișiere Autodesk Fusion 360.

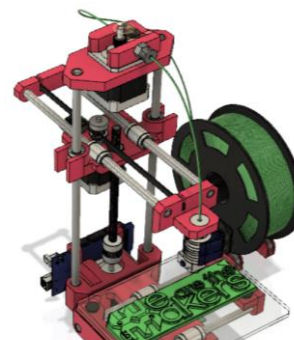


Figura 2: Simbolul unui imprimante 3D

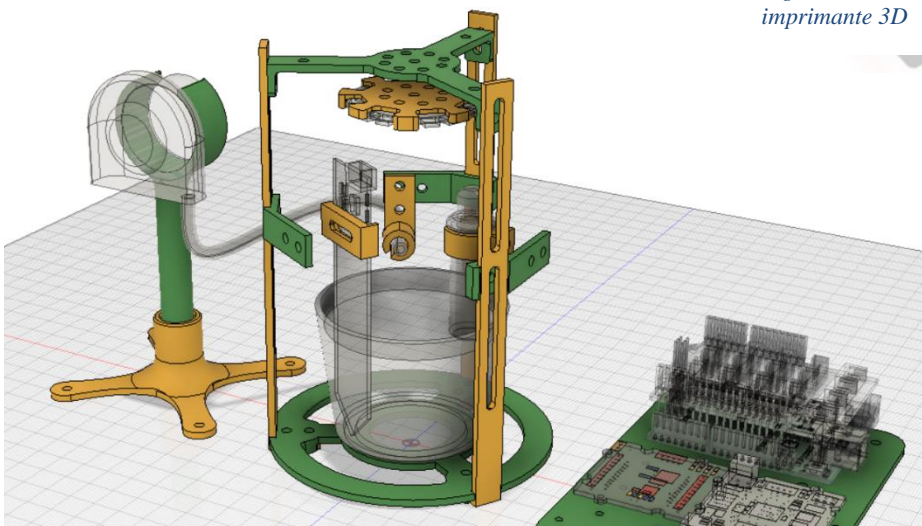


Figura 3: Privire de ansamblu asupra datelor CAD

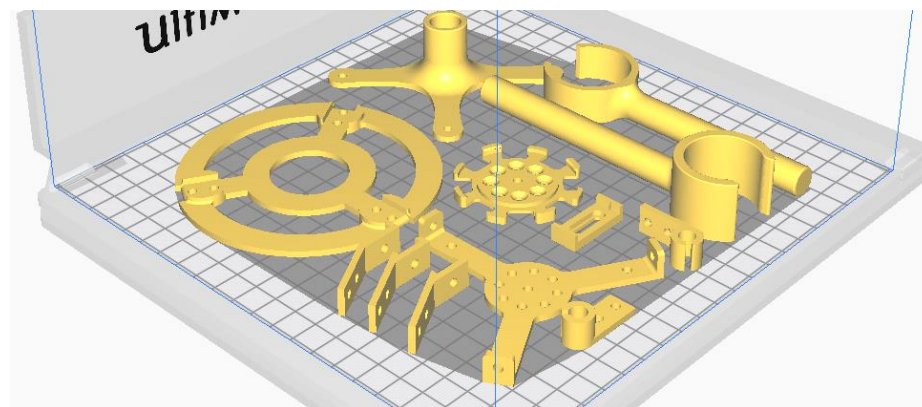
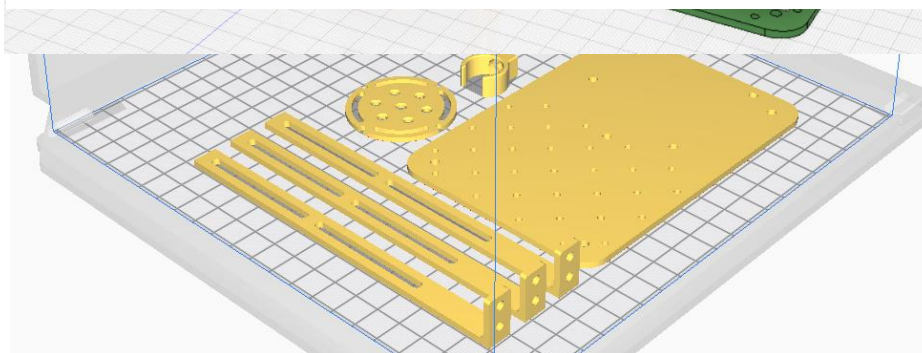
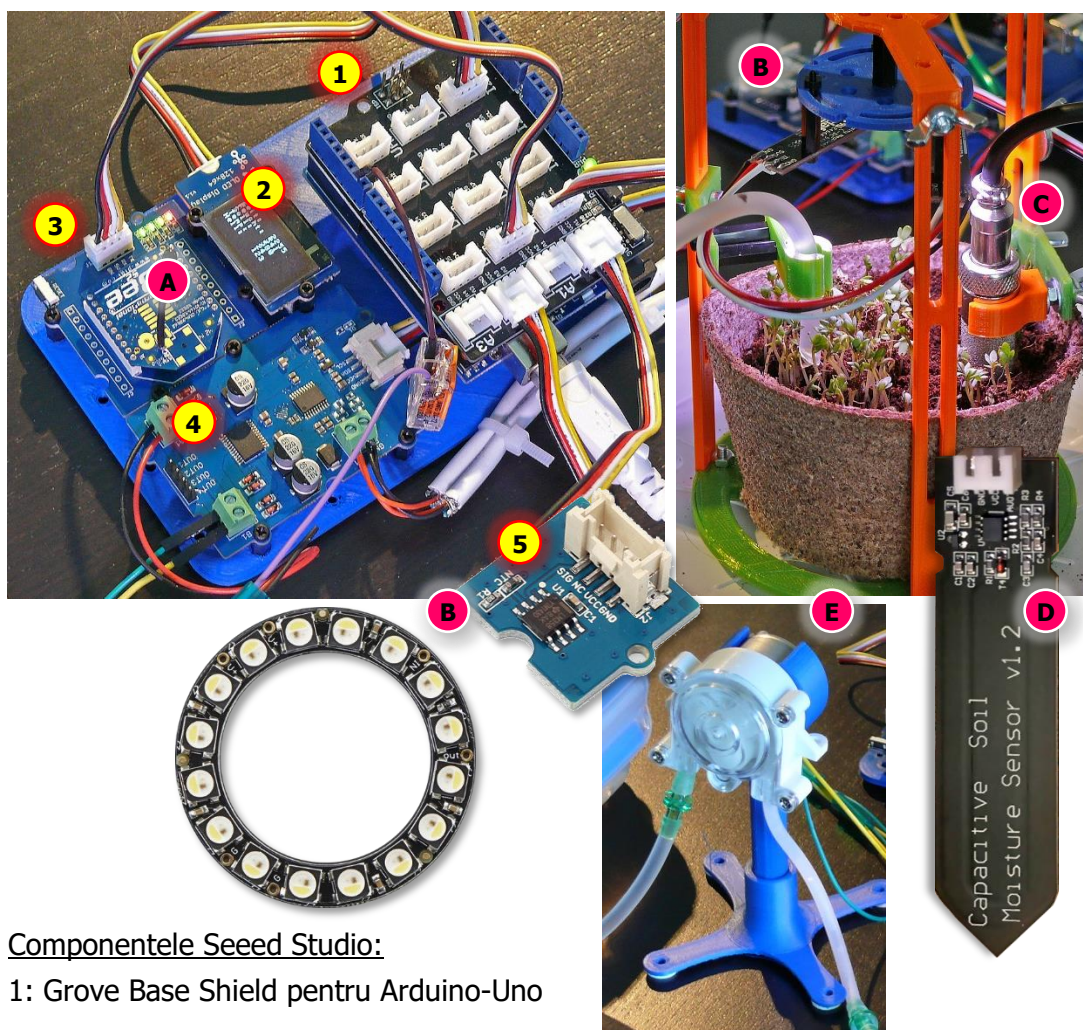


Figura 4: setul complet a datelor 3D într-o imprimantă 3D de dimensiune 20cm x 20cm

Componente electronice:

În această lucrare, recomandăm sistemul Grove deoarece este ușor de utilizat (http://wiki.seeedstudio.com/Grove_System/). Toate componentele de bază cu excepția XBees, senzorilor de umiditate și LED-urilor aparțin sistemului Grove standard:



Componentele Seeed Studio:

- 1: Grove Base Shield pentru Arduino-Uno
- 2: Grove OLED 128x64
- 3: Mufă Grove Bee
- 4: Grove - I2C Motor Driver (TB6612FNG)
- 5: Senzor de temperatură Grove v1.2

Senzori și actuatoare obișnuite:

- 1: Arduino Uno (sau echivalent)
- A: XBee seria 2C sau seria 2
- B: Adafruit Neopixel-ring cu 16 RGBW (NU RGB!) la 4500 K (lumină caldă)
- C: Senzor de temperatură și umiditate SHT20 sau
- D: Senzor capacitiv analog de umiditate a solului
- E: Pompă peristaltică cu motor 6V DC

Diverse componente:

- tub de silicon 5-6 mm (pentru acvarii)
- Adaptor pentru conectarea tuburilor de silicon la pompa peristaltică
- Șuruburi M3 și piulițe fluture
- Distanțiere din nylon M3 (distanțiere Hex)
- Distanțiere din nylon M2 pentru Grove (au găuri de 2mm)
- Fire Grove
- Piulițe WAGO
- Fire Jumper
- Vase de pepinieră cu diametrul de 8 cm
- Sursă de alimentare USB de 2-2.5A maxim
- Adaptor XBee USB (ex. <https://www.waveshare.com/xbee-usb-adapter.htm>)



Figura 5: Distanțiere din nylon

Plante:

Potrivite pentru realizarea experimentelor la școală sunt plantele cu creștere rapidă numite microplante ("microgreens"); nu sunt toxice și pot fi consumate:

- Creson
- Fasole Mung
- Gulie roz
- Vlăstari de trifoi roșu
- Vlăstari de broccoli
- Valerianella locusta (salată de câmp "Vit")



Figure 6: Garden cress

Calculatoare cu următoarele softuri preinstalate:

- Autodesk Fusion 360 (sau oricare alt soft de modelare 3D, ex. Wings3D)
- Softul de slicing CURA,
- Conexiune la internet pentru descărcarea de librării
- Arduino IDE
- Processing IDE
- XCTU Software pentru configurarea XBees

Librării Arduino pentru componente:

Unele componente precum senzorul de umiditate SHT20 sau motorul necesită librării pentru Arduino IDE pentru a funcționa corespunzător.

Modalitatea de importare a unei librării este prezentată la adresa:

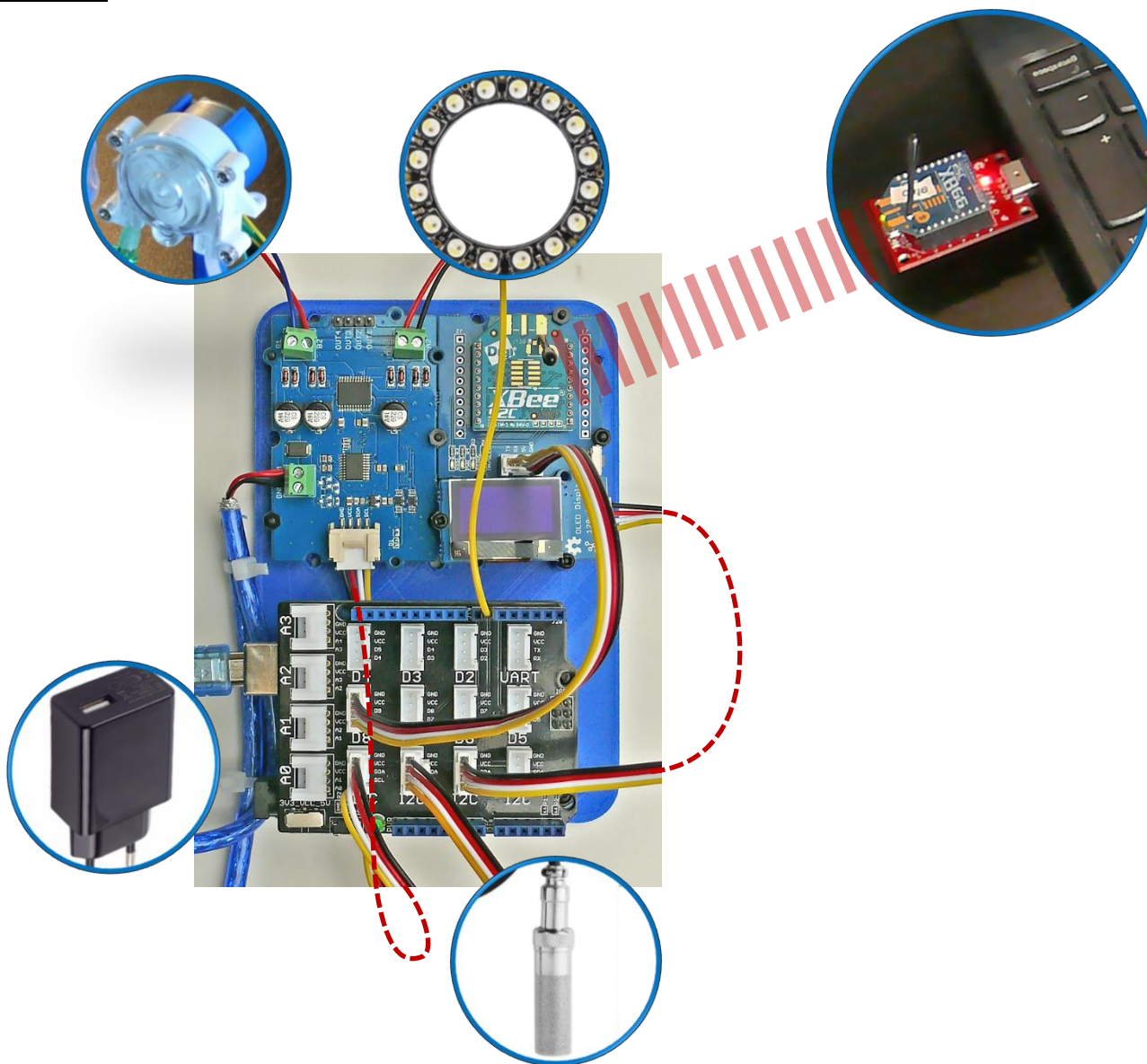
<https://www.arduino.cc/en/Guide/Libraries>

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

O serie de coduri speciale pentru senzorul de temperatură pot fi găsite aici:

http://wiki.seeedstudio.com/Grove-Temperature_Sensor_V1.2/

Conexiuni:



8b Teorie despre lumina cu LED-uri pentru stimularea creșterii plantelor și măsurarea umidității solului

Lumina cu LED-uri pentru stimularea creșterii plantelor

Folosirea luminii cu LED-uri pentru stimularea creșterii plantelor are la bază teoria PAR - "photosynthetically active radiation" (RO. radiație activă fotosintetic): Plantele folosesc fotoni de lumină pentru reacții chimice de sintetizare a zahărului din dioxid de carbon; aceste reacții chimice au loc folosind pigmenți de clorofilă în interiorul cloroplastelor fiecărei celule a plantei.

Atunci când clorofila este iradiată cu lumină naturală, absoarbe lumina roșie și albastră. Culoarea verde nu este absorbită direct pentru procesul de fotosinteză. Prin urmare plantele sunt verzi.

Lumina LED pentru stimularea creșterii plantelor trebuie să furnizeze, în special lumină albastră și roșie din spectrul de absorbție al clorofilei. De aceea se folosesc părțile R și B a LED-ului RGBW de mare putere. Partea verde a LED-ului nu este utilizată.

Dar o plantă folosește și alte părți din spectrul continuu de lumină. Procesul de fotosinteză este unul complex și reprezintă un domeniu de cercetare. Pe scurt: lumina verde pătrunde mai adânc în plantă și face ca procesul de fotosinteză să fie mai eficient, deoarece influențează rata de absorbție a clorofilei. Prin urmare, e necesară și o cantitate mică din spectrul verde spre galben.

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

În plus, creșterea plantelor este influențată de hormonii plantelor, care de asemenea reacționează la lumina naturală și au nevoie în special de un spectru de lumină continuu. Spre exemplu, fitocromii reacționează la lumina infraroșie. https://en.wikipedia.org/wiki/Plant_hormone
<https://en.wikipedia.org/wiki/Phytochrome>

În consecință, o lumină optimă pentru stimularea creșterii plantelor nu trebuie să se limiteze la părțile roșu și albastru ale spectrului, ci trebuie să includă și partea albă a LED-ului care produce un spectru continuu alb-cald care să influențeze sistemele secundare ale procesului de fotosinteză și hormonii plantelor. Prin urmare, folosim LED-ul 4500K-RGBW de la Adafruit Industries.

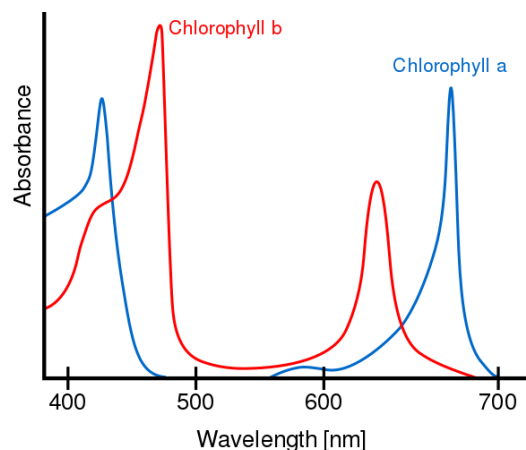


Figura 7: Spectrul de absorbție a pigmenților de clorofilă a și b.
https://en.wikipedia.org/wiki/Chlorophyll_b

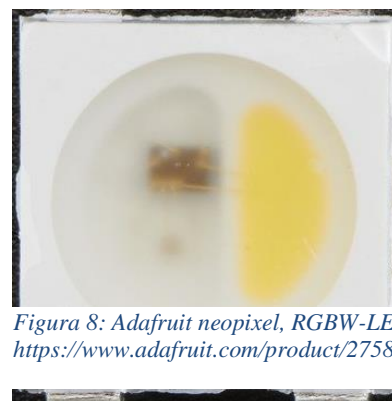


Figura 8: Adafruit neopixel, RGBW-LED,
<https://www.adafruit.com/product/2758>

Detectarea umidității din sol

De obicei umiditatea este măsurată ca procent a umidității aerului. Prin urmare, este necesar un senzor de temperatură și umiditate. Umiditatea poate fi măsurată în moduri diferite și una dintre cele mai frecvente metode este măsurarea capacitivă. Senzorul este un condensator a cărui capacitate este modificată odată cu absorbția / desorbția apei.

Capacitatea C a unui condensator depinde de placa A , de distanța dintre plăci d și de mediul dielectric dintre două plăci de metal cu o permitivitate constantă dată ϵ_R :

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

În timp ce distanța dintre plăci și dimensiunea plăcilor nu pot fi modificate de umiditate, constanta de permitivitate poate. De obicei permitivitatea unui compus dat este comparată cu permitivitatea vidului perfect și astfel se numește permitivitate relativă. În tabelul următor sunt prezentate câteva valori importante ale permitivității relative:

Mediu	Permitivitatea relativă
Vid	1
Aer	1.0006
Apă	80
Sol uscat mineral	5

Ca o consecință, capacitatea solului crește cu cât conține mai multă apă. Solul umed are o permitivitate semnificativ mai mare decât cel uscat. Uzual se măsoară conținutul volumetric de apă a solului, notat SWC (en. "volumetric soil water content"):

$$SWC = \frac{\text{Volumul de apă}}{\text{Volumul de sol}}$$

Un circuit electronic pentru măsurarea acestor schimbări a capacității e construit ca un circuit RC. În funcție de capacitatea sa, un circuit RC are o caracteristică a constantei de timp ce poate fi măsurată de un microcontroler. Cu cât e mai mare capacitatea, cu atât e mai lungă constanta de timp. În concluzie, umiditatea e măsurată astfel:

creșterea umidității $\xrightarrow{\text{conduce la}}$ creșterea capacității $\xrightarrow{\text{conduce la}}$ creșterea constantei de timp

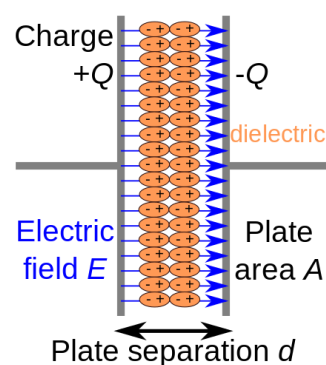


Figura 9: Schema unui condensator.
<https://en.wikipedia.org/wiki/Capacitor>

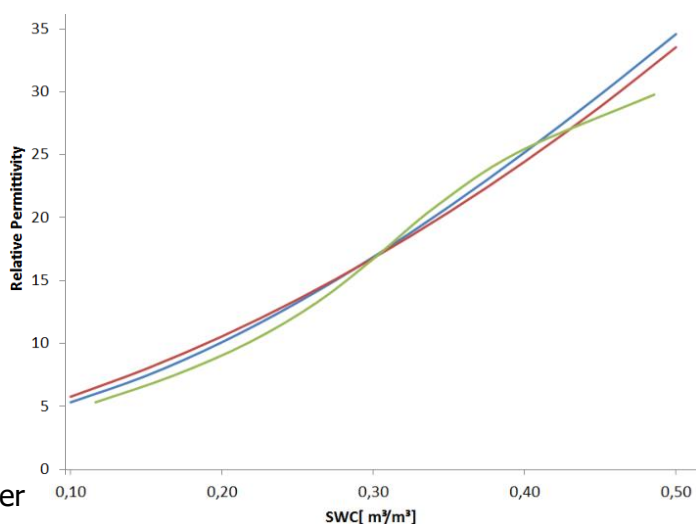




Figura 10: Porretta, Bianchi "Profilul permitivității relative și a conductivității electrice de la modele de pământ nesaturat cu conținut de apă", ANNALS OF GEOPHYSICS, 59, 3, 2016, G0320

Comparație între SHT20 și senzorul de umiditate a solului:

SHT 20	Senzor de umiditate a solului
	
Măsoară umiditatea aerului din interiorul carcasei sale impermeabile și simultan temperatura	Măsoară direct permitivitatea relativă a solului. Este necesară o măsurare suplimentară a temperaturii.
Comunică prin I2C cu microcontrolerul Arduino	Produce un semnal analogic care trebuie digitalizat cu convertorul analog-digital a microcontrolerului Arduino
Nu trebuie să fie complet introdus în sol, deoarece e necesar un flux de aer	Trebuie păstrat cât mai adânc în interiorul solului
Costă circa 20 Euro	Costă circa 5 Euro
ATENȚIE: Deoarece acest senzor are ca ieșire umiditatea relativă, valorile de măsurare reflectă condițiile solului. De cele mai multe ori, senzorul produce valori peste 100% deoarece aerul din interiorul carcasei impermeabile e saturat de umiditatea solului. De aceea trebuie utilizate valorile brute pe 16 biți care nu sunt convertite în valori fără sens ale umidității aerului.	ATENȚIE: Deoarece acest senzor măsoară permitivitatea relativă a solului, e absolut esențial ca senzorul să fie complet introdus în sol. În plus, valorile de la senzor sunt afectate, deoarece odată cu udarea, volumul solului se modifică și, prin urmare, suprafața de acoperire a senzorului. În plus, creșterea plantelor modifică, de asemenea valorile permitivității.

8c Exemplu de cod sursă Arduino

```
#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: Processing App



Figura 11: Captură din interfața aplicației

De obicei dispozitivele IoT au o interfață care le permite utilizatorului să monitorizeze și să controleze de la distanță dispozitivul conectat.

O modalitate (relativ) simplă de a obține acest lucru cu elevii este de a citi datele seriale de la XBee și de a afișa valorile pe o interfață grafică cu utilizatorul.

În plus, Arduino IDE și Processing IDE sunt strâns legate una cu alta.

Pentru butonul "GIESSEN" și butonul circular "SOLL" (care înseamnă valoarea optimă pentru umiditatea solului) este utilizată librăria "ControlP5" ce poate fi instalată cu ușurință din Processing IDE.

Pentru controlP5:

<https://code.google.com/archive/p/controlp5/downloads>

8c. Exemplu de cod sursă Processing

```
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.

Descrierea activității

Lecțiile 1 & 2 (90min):

Elevilor li se va prezenta despre IoT prin exemple: roboți de aspirare cu aplicații la distanță, stații meteo bazate pe web, urmărirea activității cu aplicații de comunicare și ferme agricole inteligente. Elevii trebuie să examineze modul în care funcționează aceste dispozitive și ce componente sunt necesare: un sistem bazat pe microcontroler controlează și coordonează senzorii și actuatoarele atașate. În plus comunică și coordonează alte sisteme similare, de obicei prin rețele de comunicare fără fir. Părți necesare: senzori, actuatoare, dispozitive de comunicare. Trebuie să se discute despre posibilități, amenințări și limitări: când se merită folosirea de IoT-uri și când nu.

Lecțiile 3&4 (90 min)

Elevii ar trebui să stabilească componentele pentru monitorizarea și optimizarea creșterii plantelor. Apoi ei pot începe să construiască un dispozitiv, de la zero, folosind componentele date. La sfârșit, pot fi predate tehnici de programare de bază care să le permită elevilor să creeze propriile lor programe.

Lecțiile 5&6 (90 min)

Se predă teorie cu privire la modalitățile de măsurare a umidității solului și iluminarea plantelor. Elevii pot face măsurători ale conținutului volumetric de apă a solului (SWC) cu ajutorul senzorilor pentru a realiza curbele de calibrare. Elevii trebuie să compare rezultatele lor experimentale cu ale celorlalți pentru a observa că fiecare grup de elevi a obținut seturi de valori măsurate care diferă semnificativ unele de altele.

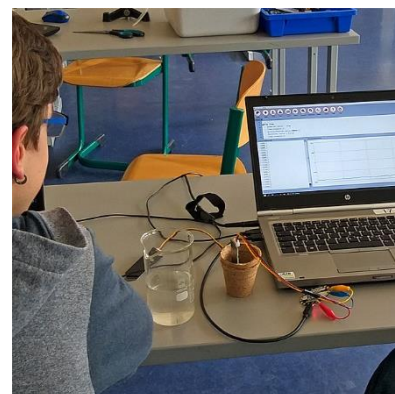


Figura 12: experimente cu senzorii de umiditate

Lecțiile 7&8 (90 min):

Elevii încep programarea și controlul pompelor peristaltice cu driverul de motor. Profesorul trebuie să-i învețe pe elevi teoria H-Bridges și, în plus, despre comunicarea I2C dintre componentele electronice. Elevii trebuie să măsoare comunicarea I2C cu osciloscopul. Profesorul prezintă conceptul de PWM (Pulse width modulation). Elevii pot măsura comunicarea dintre Arduino și LED-urile Neopixel cu ajutorul osciloscopelor.

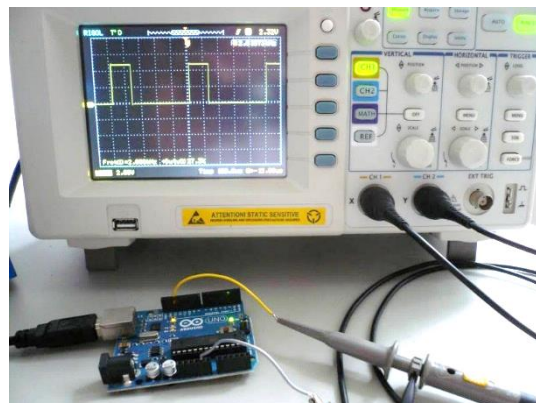


Figura 13: Măsurări PWM cu osciloscopul

Lecțiile 9&10 (90 min):

Construirea unei rețele de comunicație: modulele XBee și comunicarea serială (UART). Elevii trebuie să folosească soft XCTU ca punct de plecare pentru experimentele lor privind comunicarea în rețelele wireless.

ATENȚIE: XBees ar trebui preconfigurate în pereche de profesor, altfel se va pierde mult timp pentru învățarea modului de lucru cu multe tipuri de configurări. În esență, perechile XBee sunt definite folosind trei valori bazate pe utilizator diferite, care sunt marcate mai jos cu roșu.

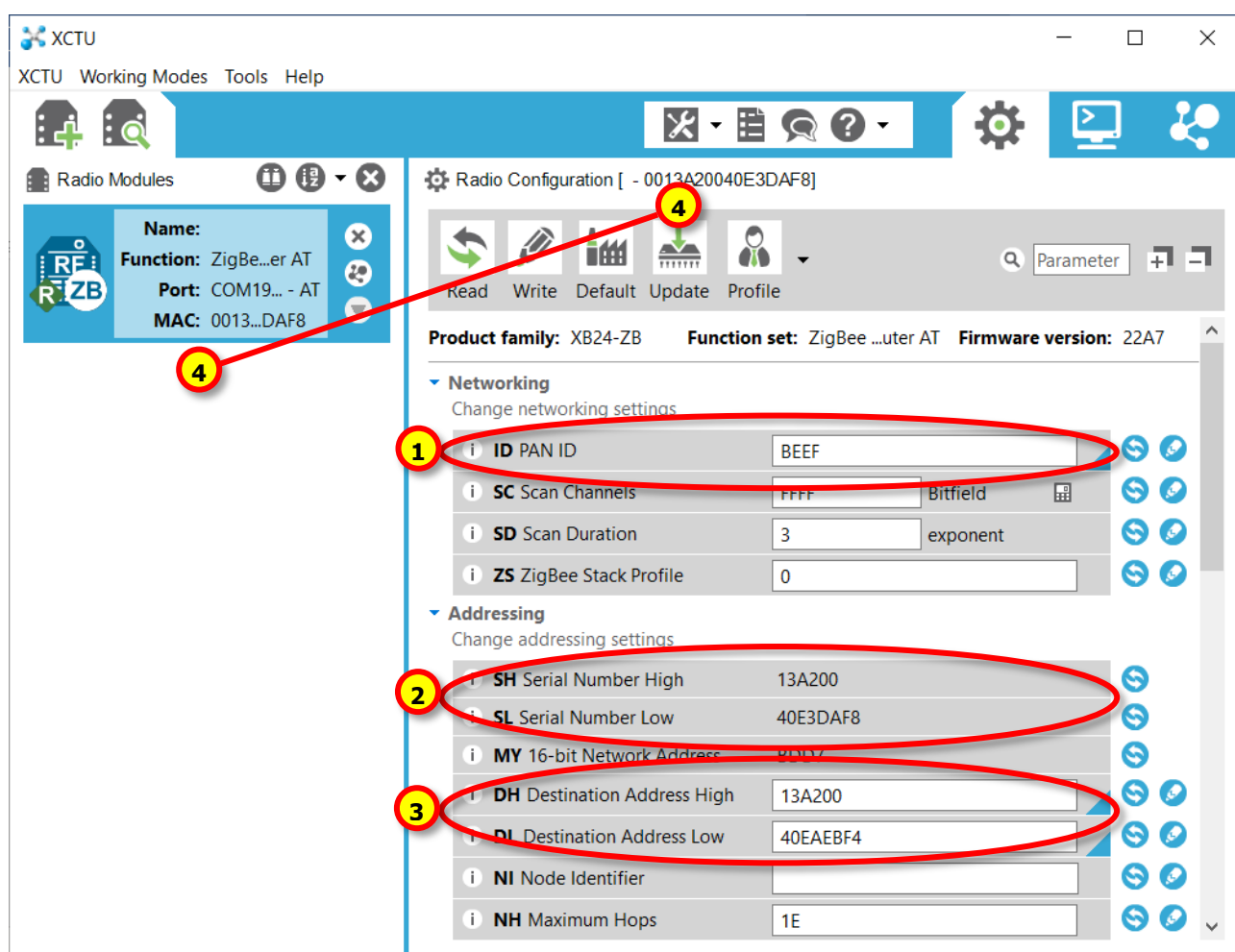
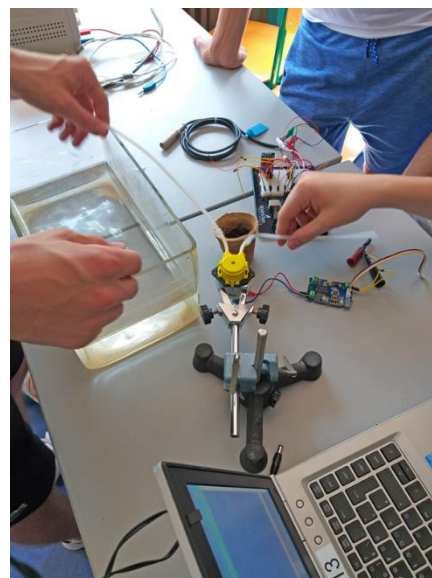
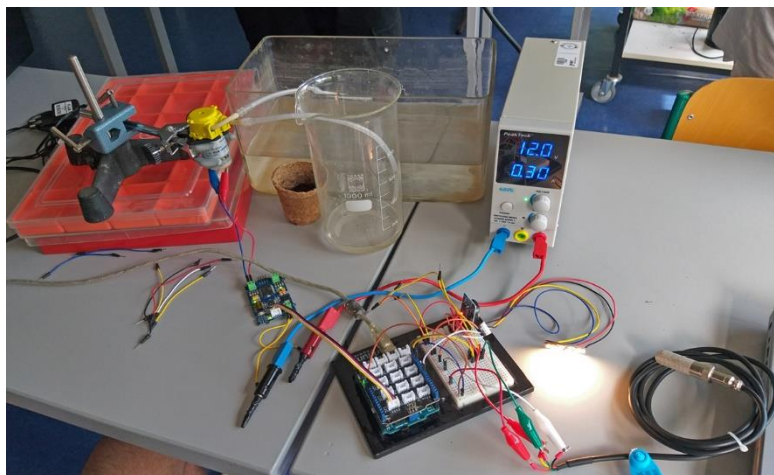


Figura 14: Captură după interfața cu utilizatorul XCTU

- 1: PAN ID trebuie să fie la fel pentru ambele XBees. Se folosesc simboluri hexazecimale, eg. așa numitul "Hexspeak": "BEEF", "CAFE", "F00D", "AFFE", etc..
2. Valoarea *Serial Number High* trebuie să fie copiată de la un XBee ...
3. ...la celălalt.
4. Un XBee trebuie să fie configurat 'Coordinator' iar celălalt 'Router'. Atenție: În unele materiale găsiți că al doilea ar trebui configurat 'Endpoint'. Acest lucru ar putea genera o serie de probleme, deoarece XBees configurate 'Endpoint' intră în stand-by pentru a minimiza consumul de energie. Apoi, XBees poate fi conectat la postul de comunicație UART a plăcii Arduino.

Lecțiile 11 până la final (270 min):

Programare liberă! Și recoltare fericită 😊



<p>10. Feedback</p>	<p>La sfârșitul lecției, elevii ar trebui să aibă cunoștințe solide cu privire la modul în care funcționează principiile IoT și modul în care dispozitivele conectate la Internet comunică între ele. În carul lecțiilor elevii au experimentat șansele și limitările tehnologiei actuale. De asemenea, au fost îndrumați privind aspectele de bază din electronică și informatică. În plus, au fost predate aspectele biologice ale creșterii platelor.</p>
<p>11. Evaluare</p>	<p>Elevii țin fiecare câte un jurnal de laborator ce poate fi verificat de către profesor. Elevii pot prezenta, de asemenea, rezultatele experimentelor realizate de către ei. În plus, la sfârșitul lecțiilor se va da un test standard de verificare.</p>