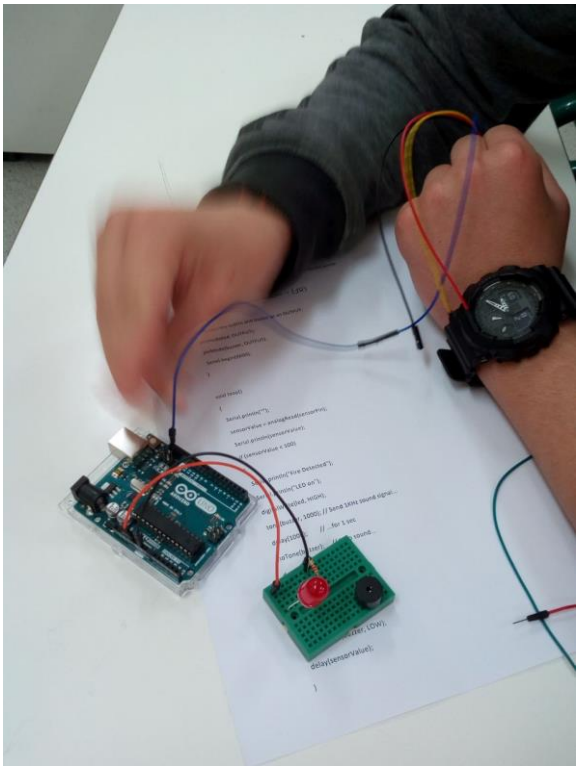
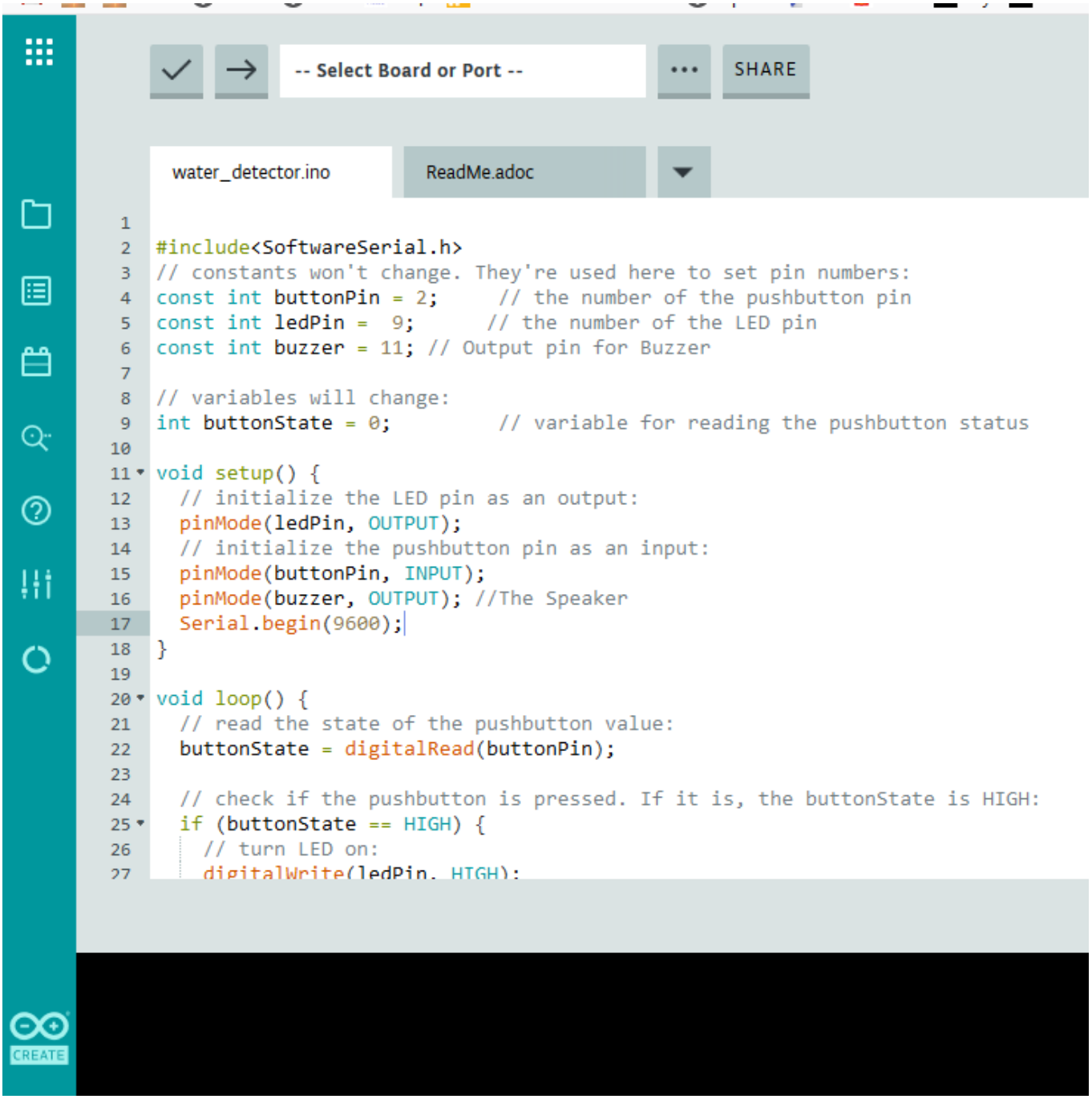
We are the makers – IoT Learning Scenario

1. Title of the Scenario	***Detecting amount of remaining water in an improvised watering system***
2. Target group	This scenario can be fit with ages: 12-15 years old
3. Duration	This scenario can be implemented in the classroom in 3 sessions (2-3 hours each)
4. Learning needs which are covered through the exercise	- Understanding the value of preserving plants and flowers during summer without over-watering them, - Highlighting traditional and modern methods of watering plants, - Understanding basic Arduino theory (modules, add-ons, platform, programming language, etc.) - Understanding how sensors operate
5. Expected learning outcomes	- Realizing the importance of controlling the amount of water - Building basic Arduino constructions - Effectively using Snap for basic projects - Basic Arduino programming (code) - Effectively using and programming with sensors
6. Methodologies	**Lesson 1: Welcome session** - Team formation - Small Introduction/Presentation: Preserving plants and flowers during summer while saving water, Presentation of the project objectives, setting the team goals, elaborating on the final outcome/result - Arduino: First familiarization **Lesson 2:** - Arduino Construction (boards, sensors, etc.) - Snap 4 Arduino: Commands, compilation, execution - Arduino code: a set of commands are introduced, and explanation is provided **Lesson 3:** - Programming towards task implementation (Snap4Arduino, code). It is worth noting that half-baked solutions are also used in order to smoothly engage students in programming with Snap4Arduino
7. Place / Environment	Computer Lab
8. Tools / Materials / Resources	Projector, Audio system, Arduino kits, sensors

9. Step by step description of the activity / content	**Lesson 1** 1. Small team formation activity – team bonding 2. Demonstration of short videos about improvised watering systems (immerse students in the context of the activity and provide them with basic information). 3. Presentation of the steps that will be followed towards project objectives achievement 4. Introduction to Arduino – short demonstration (through video and/or real time demostration) **Lesson 2** 1. Construction of Arduino in teams (boards/sensors attachment, etc.) 2. Demonstration of Snap4Arduino – easy to start with tasks for familiarization purposes (blinking LED, etc.) 3. Demonstration of Arduino coding platform – easy to start with programming tasks for familiarization purposes **Lesson 3** 1. Snap4Arduino and/or coding platform to implement the project (watering plants during summer) 2. Testing the solutions 3. Discussion – conclusions Is this project related to real life? Does it address real risks?
10. Feedback	Lesson 1: Through discussion, the teacher decides whether the students have realized the importance of preserving plants and saving water, especially during summer. Lesson 2: The amount of the small projects' success (construction and programming) Lesson 3: Focus on the contribution of each team towards project completion
11. Assessment & Evaluation	Lesson 1: A short questionnaire is delivered for students to fill in. The questionnaire focuses on the topic of the project and aims at exploring students' perceptions on problems related to watering procedures. **Lesson 2:** Focus groups are organised in order to explore how each team worked towards the final goal, the team dynamics and the way the tasks were carried out and failures were encountered **Lesson 3:** The final project is evaluated from technical perspective and conceptual. It is interesting to see what type of tools the students used and mixed, how complex solutions they implemented, whether the project scenario was extended, whether ideas for optimal solutions were put forward. The evaluation is based on ongoing observations during the implementation of the project and review of final outcome (by the teacher).






```
#include<SoftwareSerial.h>
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  9;      // the number of the LED pin
const int buzzer = 11; // Output pin for Buzzer

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  pinMode(buzzer, OUTPUT); //The Speaker
  Serial.begin(9600);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
```