Detecting amount of remaining water in an improvised watering system

Project description

This project is about detecting the amount of remaining water in an improvised watering system that can be attached to flower pots and/or plants.

During summer, many owners of plants and flowers in Greece are trying to keep watering them by using improvised, and rather ecological[1], methods and solutions. One of them is the use of common plastic bottles, containing water, that are attached in several ways to plants and/or flower pots (Figure 1). However, the owners have to check (even on a daily basis) the amount of remaining water inside the bottles, and replace it if necessary.

Figure 1 Improvised solutions for watering plants

Therefore, the main idea of the present activity is the creation of an Arduino-based device, equipped with an ultrasonic sensor or with a metallic detector, that will inform the owners about the amount of remaining water inside every bottle. When the bottle is about to get empty, the ultrasonic sensor is triggered (Figure 2) emitting an optical or audio signal to the Arduino-based system, which will instantly inform the owner through a GSM Caller Alarm (Figure 4) or through an Arduino Bluetooth data application. Alternatively, a metallic sensor that detects the level of water (Figure 3), emitting a signal when the level is low, can be applied.

[1] These systems also contribute to saving water through preventing flowers and plants from being over-watered.

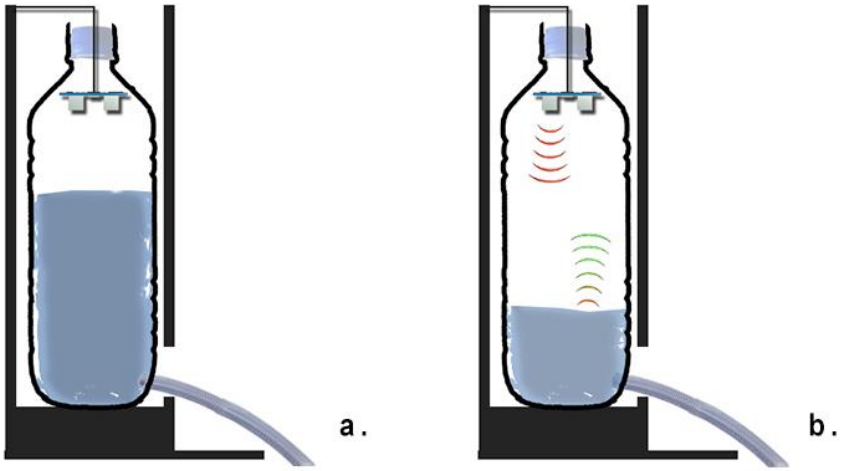


Figure 2 Schematic example of the mechanism with an Ultrasonic sensor

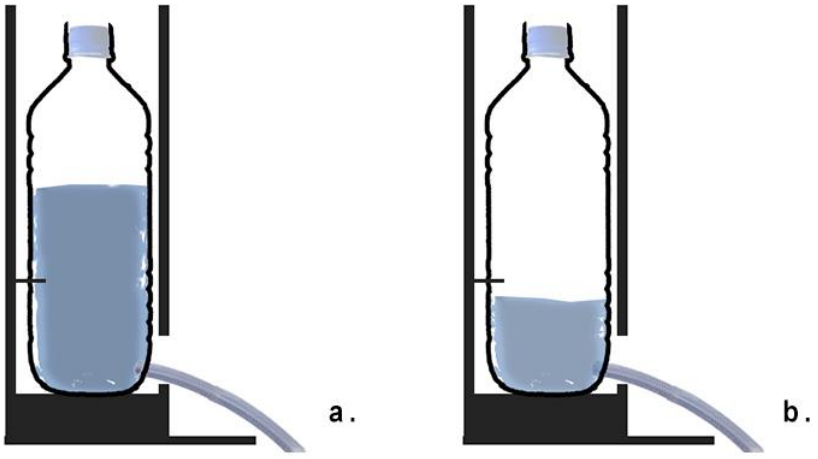
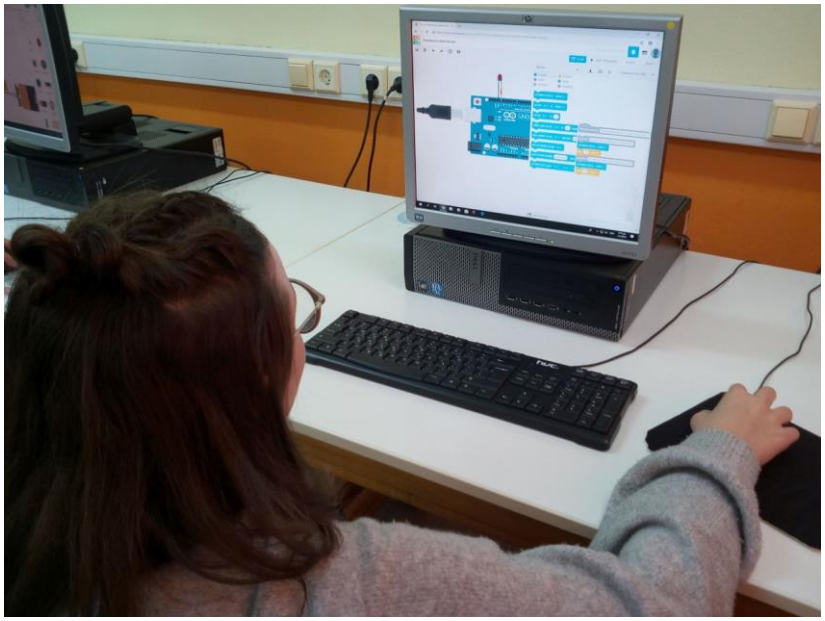
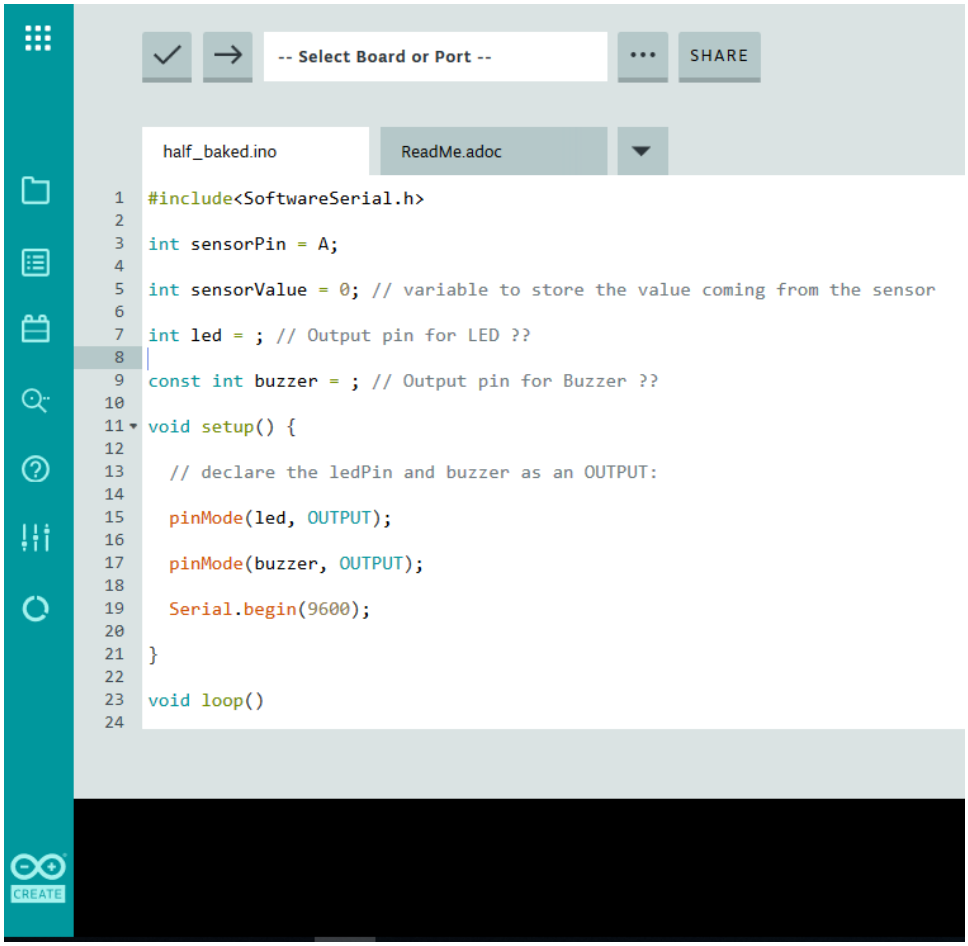


Figure 3 Schematic example of the mechanism with a metallic detector

Both the components (Arduino and GSM) could be powered by solar banks and/or rechargeable batteries.



Number of project: 2017-1-DE03-KA201-035615




```
#include<SoftwareSerial.h>

int sensorPin = A;

int sensorValue = 0; // variable to store the value coming from the sensor

int led = ; // Output pin for LED ??

const int buzzer = ; // Output pin for Buzzer ??

void setup() {

  // declare the ledPin and buzzer as an OUTPUT:

  pinMode(led, OUTPUT);

  pinMode(buzzer, OUTPUT);

  Serial.begin(9600);

}

void loop()
```


```
  // 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);
    // turn buzzer on:
    tone(buzzer, 1000); // Send 1KHz sound signal...
    delay(1000);        // ...for 1 sec
    noTone(buzzer);     // Stop sound...
    delay(1000);
  } else {
    // turn LED & buzzer off:
    digitalWrite(ledPin, LOW);
    digitalWrite(buzzer, LOW);
  }
}
```