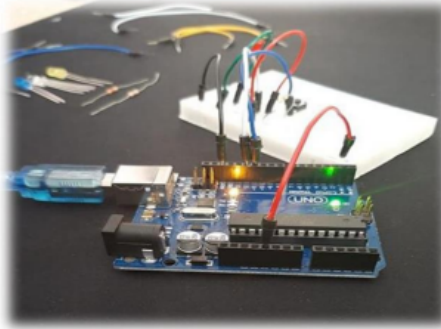
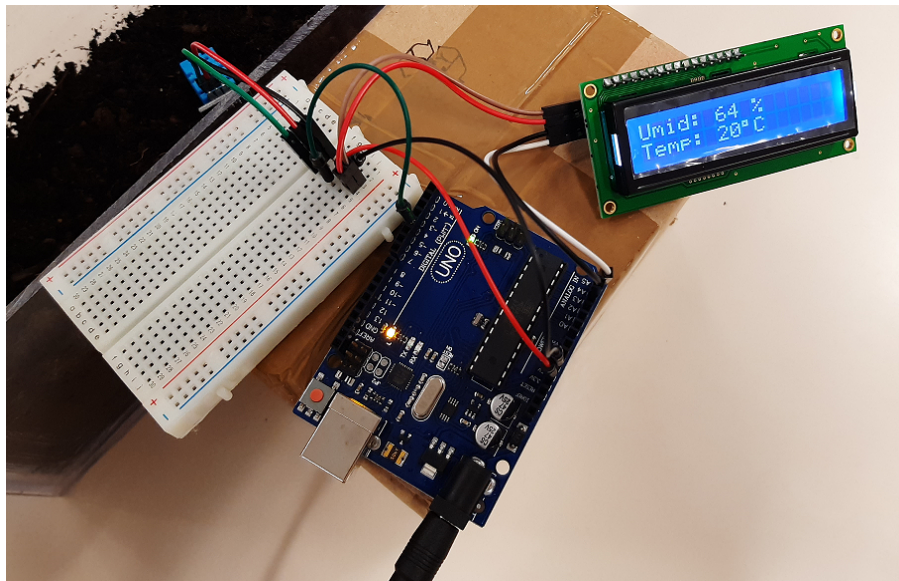


## STAZIONE 2 LABORATORIO DI INFORMATICA

Inizialmente abbiamo approfondito alcuni aspetti applicativi della piattaforma Arduino (hardware e software) utilizzando dei kit forniti dalla scuola.



Dopo vari approfondimenti ci siamo posti l'obiettivo di ricavare i dati relativi all'umidità e alla temperatura nel terreno.



L'hardware è stato programmato tramite uno "Sketch"

```

File Modify Sketch Showment Auto
//LCD
//Compatible with the Arduino IDE 1.0
//Library version: 1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define LCD_ROWS 16
#define LCD_COLS 20

// LCD I2C address
LiquidCrystal_I2C lcd(0x27, 20, 4); // set the LCD address to 0x27 for a 16 char and 2 line display

void setup()
{
  lcd.begin(); // initialize the LCD
  lcd.backlight();

  lcd.createChar(0, bell);
  lcd.createChar(1, note);
  lcd.createChar(2, clock);
  lcd.createChar(3, heart);
  lcd.createChar(4, dumb);
  lcd.createChar(5, obese);
  lcd.createChar(6, crown);
  lcd.createChar(7, retarow);
  lcd.home();

  lcd.print("Hello world...");
  lcd.setCursor(0, 1);
  lcd.print("I ");
  lcd.printByInt(1);
  lcd.print(" Arduino");
  lcd.setCursor(0, 1);
  displayPWCode();
}
  
```

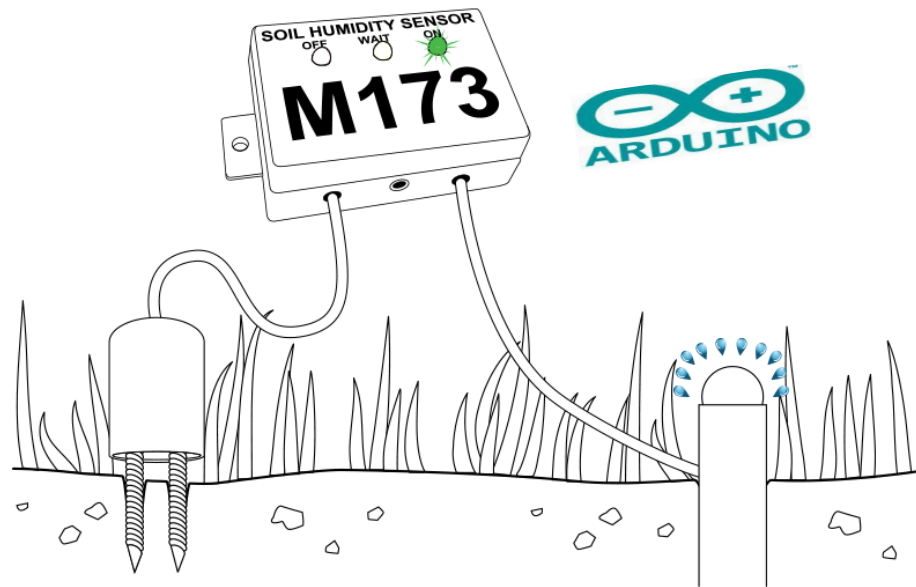
```

File Modify Sketch Showment Auto
// FILE: dht11_test1.pde
// PURPOSE: DHT11 library test sketch for Arduino
//
// Function: dht11.h
// dht11 DHT;
// #define DHT11_PIN 4

void setup() {
  Serial.begin(9600);
  Serial.println("DHT TEST PROGRAM");
  Serial.println("LIBRARY VERSION: 1.1");
  Serial.println(DHT11_LIB_VERSION);
  Serial.println();
  Serial.println("Type, status, humidity (%), temperature (C)");
}

void loop() {
  int dht;
  Serial.println("DHT11, V");
  dht = DHT.read(DHT11_PIN); // READ DATA
  switch (dht) {
    case DHT11_OK:
      Serial.println("OK,V");
      break;
    case DHT11_ERROR_CHECKSUM:
      Serial.println("Checksum error,V");
      break;
    case DHT11_ERROR_TIMEOUT:
      Serial.println("Time out error,V");
      break;
    default:
      Serial.println("Unknown error,V");
      break;
  }
  // DISPLAY DATA
  Serial.println(DHT.humidity,1);
  Serial.println(" ",1);
  Serial.println(DHT.temperature,1);
  Serial.println();
  delay(1000);
}
  
```

L'automatismo così impiegato innaffia le piante solo in prossimità dei valori minimi di umidità, consentendo un importante risparmio delle risorse idriche e una maggiore salvaguardia della salute delle piante.



Questo è il gruppo che ha lavorato alla programmazione di Arduino e allo sviluppo del progetto attraverso hardware e software

