Plant Supervisor

Introducere

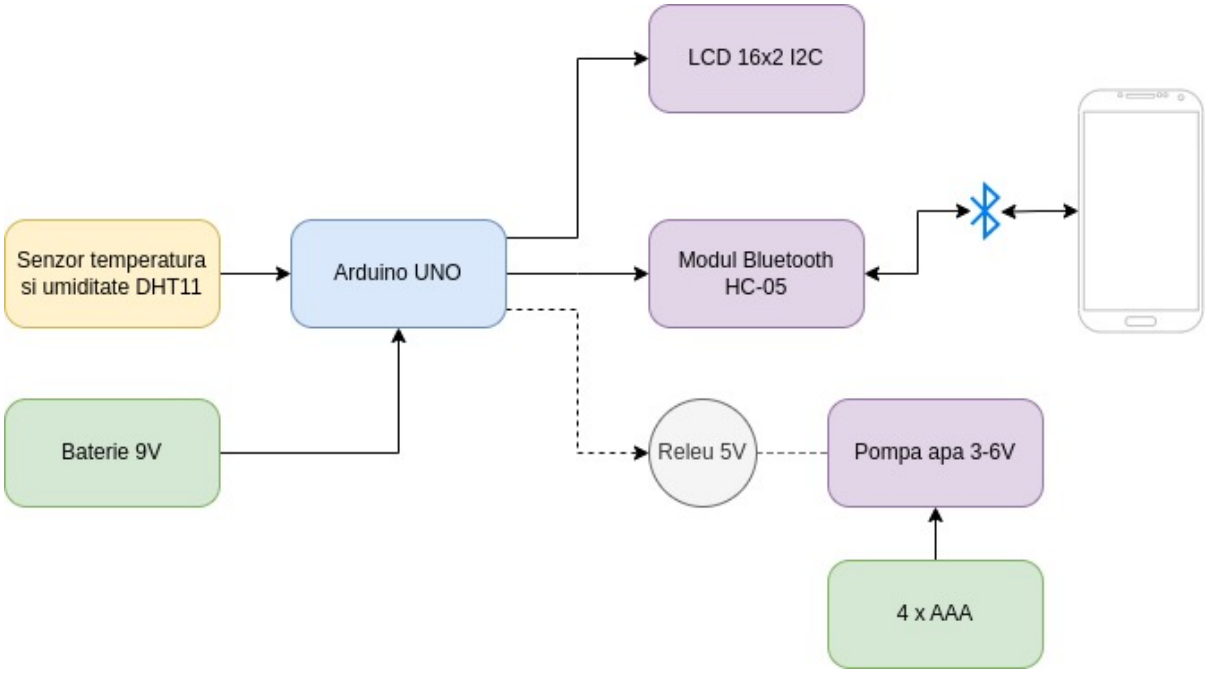
Inteleptii japonezi spun ca intretinerea unui bonsai este un proces dificil si meticulos deoarece el necesita “comunicarea neintrerupta” cu arborele. Plantele fac parte din viata noastra de zi cu zi, ne dau o stare de bine si ne revitalizeaza aerul. Dar cum stim atunci cand o planta are nevoie de ajutorul nostru? De cata apa are nevoie daca afara sunt 21°C? Dar umiditatea este ok? Sa o mut mai la umbra?

Toate aceste intrebari ne fac din ce in ce mai mult sa ne dorim sa intelegem nevoile unei plante asa cum intelegem nevoile celorlalti din jurul nostru. Insa acest lucru poate deveni mult mai usor cu Plant Supervisor.

Descriere generală

Plant Supervisor actioneaza ca traducatorul perfect intre tine si planta ta, ca sa nu iti mai minti vecinii ca orhideea primita cadou s-a ofilit pentru ca “ai fost plecat de acasa”. Plant Supervisor ofera in timp real date despre temperatura si umiditatea din mediul inconjurator atat pe un display incorporat cat si printr-o aplicatie. In plus, sistemul de auto-watering stie sa ofere plantei tale apa atunci cand tu nu ai timp sau esti plecat de acasa.

Schema Block



Hardware Design

Lista Componente
Arduino UNO
LCD Display I2C
Modul Bluetooth HC-05
Senzor temperatura si umiditate DHT11
Pompa de apa submersibila 3-6V
Furtun
Beardboard
Releu un canal 5V
Rezistor 10Ω
Suport 4 baterii AAA + baterii
Baterie 9V
Conector baterie 9V pentru Arduino

Software Design

Pentru implementarea sotfware am folosit urmatoarele biblioteci:

- Senzorul de temperatura si umiditate DHT11
- Ecranul LCD 16x2
- Modulul Bluetooth - builtin SoftwareSerial

Valorile de temperatura si umiditate sunt trimise prin interfata seriala “BTSerial” una dupa cealalta,

separate de caracterul “|”. Timpul de retrimitere a datelor este dat de variabila “transmitInterval” si este setat by default la 3 secunde pentru o experienta de development mai usoara.

Atat pentru transmiterea cand si pentru trimiterea datelor prin bluetooth am folosit functia “milis()” pentru o executie a codului non-blocanta.

```
#include <SoftwareSerial.h>
#include "DHT.h"
#include <LiquidCrystal_I2C.h>

// DHT defines
#define DHT_PIN 7
#define DHT_TYPE DHT11

// RX and TX PINS for BT module
#define RX_PIN 8
#define TX_PIN 9

#define WATER_PIN 10
unsigned long waterTimeStart = 0;
const unsigned long waterInterval = 2000;
bool watering = false;

unsigned long prevTransmitTime = 0;
const unsigned long transmitInterval = 3000;

// Bluetooth module
SoftwareSerial BTSerial(RX_PIN, TX_PIN);

// DHT sensor
DHT dht(DHT_PIN, DHT_TYPE);

// LCD display
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 column and 2
rows

byte gradeChar[8] = {
        0b00111,
        0b00101,
        0b00111,
        0b00000,
        0b00000,
        0b00000,
        0b00000,
        0b00000
};

void lcd_print_hum_tmp(float hum, float tmp) {
  lcd.setCursor(0, 0);
  lcd.print("Hum: ");
  lcd.print(hum);
```

```
  lcd.setCursor(0, 1);
  lcd.print("Tmp: ");
  lcd.print(tmp);
  lcd.write((byte)0);
  lcd.print("C");
}

void print_hum_tmp(float hum, float tmp) {
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.print("   Temperature: ");
  Serial.print(tmp);
  Serial.println("°C");
}

void setup() {
  BTSerial.begin(9600);
  Serial.begin(9600);

  pinMode(RX_PIN, INPUT);
  pinMode(TX_PIN, OUTPUT);

  pinMode(WATER_PIN, OUTPUT);
  digitalWrite(WATER_PIN, LOW);
  watering = false;

  dht.begin();

  lcd.init(); // initialize the lcd
  // lcd.backlight();

  lcd.createChar(0, gradeChar);
}


void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  if(isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read DHT sensor");
    return;
  }

  lcd_print_hum_tmp(humidity, temperature);

  if(BTSerial.available() > 0) {
    byte data = (byte) BTSerial.read();
    if(data == 1) {
      watering = true;
```

```
      waterTimeStart = millis();
      digitalWrite(WATER_PIN, HIGH);
    }
  }

  if (watering && (millis() - waterTimeStart) >= waterInterval) {
    watering = false;
    digitalWrite(WATER_PIN, LOW);
  }

  if(millis() - prevTransmitTime >= transmitInterval) {
    prevTransmitTime = millis();
    BTSerial.print(temperature);
    BTSerial.print("|");
    BTSerial.print(humidity);
  }

}
```

Am setat pinii 8 si 9 ca pini de tip RX si TX folosing SoftwareSerial deoarece acestia servesc la comunicarea dintre Arduino si calculator. Pentru mai multe referinte puteti verifica urmatoarele forumuri:

https://www.quora.com/How-do-you-use-TX-RX-pins-in-Arduino
https://arduino.stackexchange.com/questions/3772/can-tx-and-rx-pins-on-the-uno-be-used-like-regular-digital-pins

Aplicatia mobile

Pentru aplicatia de mobil am folosit MIT App Inventor, un tool de programare mobile ce are la baza limbajul de programare Scratch.

Daca ma judeci ca am folosit programarea cu blockuri, uite niste exemple de Neural Networks facute in Scratch. Enjoy!

https://scratch.mit.edu/studios/529587

In partea din dreapta este codul ce se ocupa de device-ului mobil la modulul de bluetooth si apasarea pe butonul de “Water”.

Atunci cand butonul de udare a plantei este apasat, se trimite catre Arduino un byte number egal cu 1.

In partea din stanga se afla main loop-ul in care se verifica constant daca exista o conexiune

bluetooth sau nu.

- daca exista, se afiseaza mesajele corespunzatoare si se face split dupa caracterul '|' pe datele primite de la modul. Prima valoare este temperatura si a doua este umiditatea.
- daca nu exista, se fac modificarile corespunzatoare de UI pentru ca nu permite user-ului sa apase butonul de “Water” si sa apara un mesaj de averitizare ca nu este conectat prin bluetooth.

Rezultate Obţinute

Un sistem complet de monitorizare si udare efecienta a unei plante.

Concluzii

A fost un proiect foarte interesant pe care l-am finalizat cu succes. Am reusit sa implementez tot ce mi-am propus.

Concluzia principala din acest proiect este faptul ca daca esti cu adevarat motivat poti sa automatizezi multe lucruri, in special cele care tin de mediul in care traim si care fac parte din rutina noastra de zi cu zi.

Proiectul este folosit in prezent de familia mea si il recomanda cu caldura. ❤

Download

- APK file
- plant_supervisor_code.zip

Bibliografie/Resurse

- Adafruit DHT
- LiquidCrystal I2C
- Receive Data from Arduino via HC-05
- Receive Multiple Data from Arduino via HC-05
- Control water pump by Arduino

Export to PDF