

Ionuț BRÎNZAN (78466) - Sistem de monitorizare si udare automata a plantelor

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Proiectul va fi o implementare a unui sistem de monitorizare si udare automata a plantelor de apartament/sera.

Ideea de la care am pornit a fost aceea ca eu uit mereu sa ud plantele, iar in situatia mea cred ca sunt si alte persoane.

Scopul acestui proiect este acela de a usura munca oamenilor care nu au suficient timp liber pentru a se ocupa de aceste plante.

Descriere generală

Sistemul va monitoriza plantele prin intermediul unor senzori de: temperatura (temperatura mediului ambiant), umiditate (umiditatea solului), lumina (intensitatea luminii).

Planta va fi udata in cazul in care aceste valori nu corespund cu cele nominale si doar in cazul in care planta nu va fi afectata , la udarea acesteia se va tine cont de umiditate, timpul de udare fiind dat de cat de repede ajunge umiditatea la valoarea maxima permisa pentru planta.

Datele legate de temperatura, umiditate, intensitatea luminii vor putea fi vizualizate pe telefon. Comunicarea cu telefonul va fi facuta printr-un modul bluetooth astfel incat datele sistemului vor putea fi vizualizate de la o distanta apreciabila iar parametrii pentru udare vor putea fi modificati prin aceasta aplicatie(care sunt parametrii nominali pentru planta, limite, etc).

Aplicatia pe telefon dispune de urmatoarele functii:

- poate vizualiza datele oferite de senzori.
- poate modifica parametrii de udare.
- poate dezactiva modul de udare automata a plantei (trecerea pe modul manual).
- poate uda planta cat timp tine apasat pe un anumit buton.
- poate activa sistemul de udare cand doreste.

In cazul in care utilizatorul nu are telefonul cu el, acesta poate observa local datele prin intermediul LCD-ului.

Udarea plantei se va face cu o pompa de apa care va fi comandata printr-un releu de 5V.

Schema bloc:



Hardware Design

Nume Produs	Cantitate
Placă de bază și componentele obligatorii	1
Condensator electrolitic 10V 16uF	1
Stabilizator de tensiune 3.3V LP2950ACZ	1
Mini Pompă de Apă Submersibilă 5V	1
Modul cu Senzor de umiditate a solului(analog)	1
Modul Senzor de Lumină Ambientală TEMT6000(analog)	1
Modul Senzor de Temperatură DHT-11(digital)	1
Modul Bluetooth Master Slave HC-05	1
LCD Nokia 5110	1
Releu comandat în 5V	1
Alimentator telefon 5V 900mA	1
Fire de legatura mama-tata	2
Tub termocontractabil 2.5mm 1m	1
Fire de legatura mama-mama	inf

Schema electrica:



Observatii hardware:

- Pentru alimentarea pompei am folosit un alimentator mai vechi de telefon de 5V si 900mA.
- Toate modulele au fost legate conform schemei.
- LCD-ul este alimentat la 3.3V.
- Ceea ce nu se vede in schema este ca am separat sistemul pompei de restul proiectului pentru ca folosesc un releu iar in viitor daca este nevoie pot folosi o pompa la un voltaj mai mare doar prin schimbarea pompei si a alimentarii aferente. (intre releu si pompa folosesc o mufa si pot desprinde pompa de restul proiectului, este util pentru transport si pentru modificari ulterioare ce pot aparea in viitor).

Software Design

Folosire senzori

Pentru senzorul de temperatura (digital) folosesc biblioteca dht11 si functia **get_temp** in care folosesc functia dht11_temperature pentru a afla valoarea temperaturii date de senzor dupa care scriu valoarea respectiva intr-un buffer pentru afisarea pe LCD.

Pentru senzorul de umiditate a solului folosesc functia **get_humid** in care setez parametrii ADC pentru citirea umiditatii (ma intereseaza niste valori foarte precise si cu interval de valori mic reprezentand procentul de umiditate).

Pentru senzorul de lumina folosesc functia **get_light** in care setez parametrii ADC pentru citirea luminii (nu ma intereseaza precizia foarte mult, dar am nevoie de un interval mai mare de valori).

Pentru senzorii analogici am folosit niste functii din laboratoare pe care le-am grupat intr-o biblioteca numita adc.

Folosire LCD

Pentru a afisa datele pe LCD folosesc biblioteca nokia5110 si functiile urmatoare create pentru o intelegere mai buna a codului:

- `prepare_lcd` care pregateste lcd-ul pentru o noua iteratie de afisare.
- `write_temp`, `write_light`, `write_humid` care afiseaza valorile date de senzori pe LCD (acestea sunt stocate in niste buffere).
- `show_updates` care ne trimite o notificare pe LCD atunci cand un parametru de udare a fost modificat.

Folosire Bluetooth

Pentru bluetooth am folosit biblioteca usart (`USART0_init`, `USART0_print`).

Trimiterea datelor am facut-o cu ajutorul `USART0_print` cu care trimit un string catre telefon, string care este alcatuit din 4 parti si 3 separatoare. Exemplu structura: "**temp|humid|light|mode**". Valorile temp, humid, light sunt cele date de catre senzori si sunt separate prin "|" iar ultima valoare mode informeaza telefonul despre modul pe care microcontrolerul stie ca este setata udarea plantei (automata: tine cont de parametrii de udare, manuala: modul de udare automata este oprit).

Primirea datelor de la telefon este facuta cu ajutorul rutinelor de tratarea a intreruperilor. Pentru a activa aceste rutine am setat pinul RXCIE0 al UCSR0B si am creat urmatoarea rutina:

ISR(USART0_RX_vect) { }

Rutina este activata in momentul in care primesc un caracter pe USART. In functie de caracterul primit voi executa anumite operatii. Operatiile executate de ATMEGA324 sunt urmatoarele:

- daca caracterul este **a**, utilizatorul doreste sa ude planta folosind sistemul de udare automata, dar fara sa tina cont daca conditiile de udare sunt indeplinite.
- daca caracterul este **b**, planta va fi udata cat timp utilizatorul tine apasat pe buton.
- daca caracterul este **c**, utilizatorul a luat degetul de pe buton-ul de la punctul anterior iar planta nu va fi udata.
- daca caracterul este **d**, utilizatorul doreste sa modifice valoarea minima a temperaturii care este acceptata pentru udarea plantei. String-ul primit de microcontroller pe USART va fi urmatorul: **d val_tempd** iar modul de parsare este urmatorul: daca caracterul primit este d atunci cat timp urmatoarele valori primite sunt diferite de d voi concatena la valoarea temperaturii ceea ce primesc. Cand primesc d ma opresc.
- daca caracterul este **e, f, g, h** procedez asemanator ca la caracterul **d** doar ca vor fi modificati parametrii de: valoare maxima a temperaturii, valoare minima a umiditatii, valoare maxima a umiditatii, valoare minima a luminii.
- daca caracterul este **i** schimb modul de functionare existent (auto/manual).

Aplicatie Android

Pentru a crea aplicatia Android am folosit App Inventor 2 al celor de la MIT. Aplicatia are urmatoarele functionalitati:

- Afisare valori temperatura, umiditate, lumina citite de catre microcontroler.
- Editare parametrii de udare.
- Udare cat timp tii apasat pe buton.
- Activare udare planta fara a tine cont de alti parametrii de udare in afara de umiditate.
- Schimbare mod de udare: automata/manuala.
- Conectare bluetooth.

Logica program

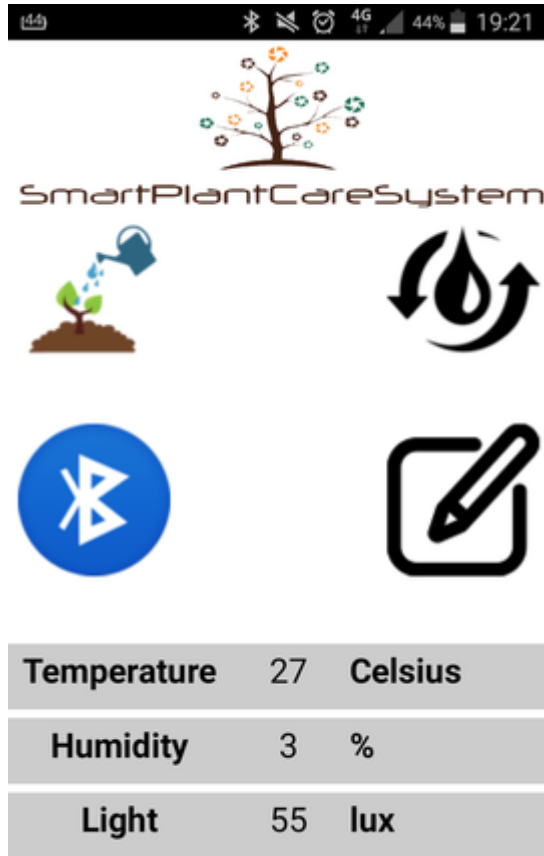
Logica programului este data de functia **main** si este urmatoarea:

- Initializez LCD, USART, ADC si setez pinul corespunzator pentru releu ca iesire.
- Activez intreruperile si setez pinul RXCIE0.
- Intr-un while execut urmatoarele operatii: pregatesc lcd-ul pentru o noua iteratie, citesc datele de pe senzorii de temperatura, umiditate si lumina si le afisez. Daca modul de udare este automata verific daca parametrii sunt corespunzatori iar in cazul in care nu sunt ud planta. Trimit datele catre telefon prin bluetooth. Daca utilizatorul a modificat parametrii de udare prin bluetooth afisam pe lcd modificarea facuta. ISR-ul pentru receive se ocupa de tot ce inseamna receptie prin bluetooth pentru ca programul sa nu aiba un comportament blocant.

Rezultate Obținute

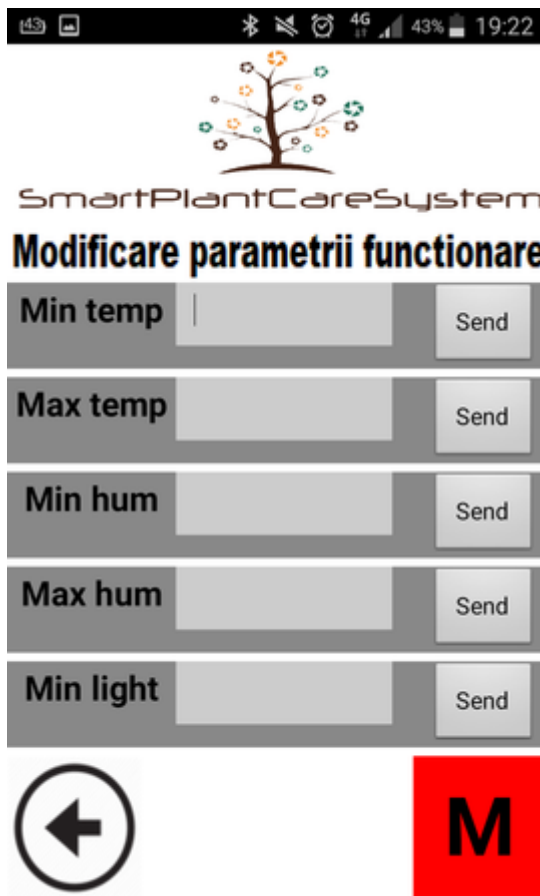
Proiectul a fost realizat cu succes.

Aplicatia Android de pe telefon arata astfel:



Copyright @ Brinzan Ionut | PM 2018

Ecran principal:



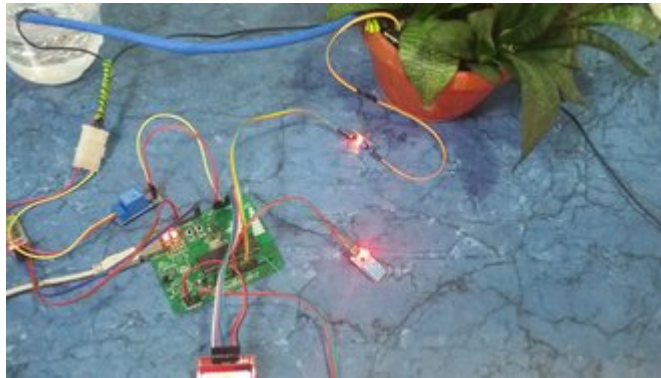
Copyright @ Brinzan Ionut | PM 2018

Ecran editare:

Poze cu proiectul:



Afisaj LCD:



Proiectul in general:

Concluzii

- Am creat un proiect functional pe care il pot folosi in casa.
- Ca sa il fac portabil mai imi trebuie un incarcator de telefon cu care sa alimentez microcontrolerul si senzorii.
- Proiectul se poate extinde foarte usor.

Download

Schema

[schema_finala.sch](#)

Arhiva

[336cc_brinzanionut_proiect.zip](#)

Arhiva contine:

- Fisierele .c si .h
- Makefile
- README
- Aplicatia android .apk (trebuie doar instalata pe telefon)

Jurnal

21 Mai 2018: Finalizare proiect

23 Mai 2018: Finalizare documentatie

Bibliografie/Resurse

Resurse software

- <https://www.avrfreaks.net/forum/tut-soft-using-usart-interrupt-driven-serial-comms?page=all>
- <http://cs.curs.pub.ro/wiki/pm>
- <https://github.com/LittleBuster/avr-nokia5110>
- <http://davidegironi.blogspot.ro/2012/12/reading-temperature-and-humidity-on-avr.html#.WwMmdkiFNEY>
- <http://ai2.appinventor.mit.edu/#>

Resurse hardware

- http://cs.curs.pub.ro/wiki/pm/_media/doc8272.pdf
- <https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- <https://www.sparkfun.com/datasheets/Sensors/Imaging/TEMT6000.pdf>
- <http://www.carobot.cc/how-to/hc-05-guide/>
- <https://blog.robofun.ro/2017/04/04/arduino-uda-ghivecele-cu-flori/>
- Documentația în format [PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/vcorneci/ionutb>



Last update: **2021/04/14 15:07**