

TEMA - PROIECT ELECTRONICA DIGITALA

Bănescu Ema-Ioana, Grupa 321CB
Facultatea de Automatica si Calculatoare

PARTEA A: Descriere generala:

In cadrul acestui proiect am realizat un sistem digital de achizitie si afisare a datelor de mediu, gandindu-ma ca aplicatie practica la un sistem de irigare a gazonului, utilizand o placa Arduino UNO, un senzor de umiditate a solului, un foto-rezistor (LDR), un ecran LCD 16x2 (varianta cu interfata I2C), un LED de stare, pentru a indica starea pompei de apa, un motor DC, pentru a actiona pompa si un buton prin care se poate inchide, respectiv porni intregul sistem.

Ideea proiectului:

Am dorit sa creez un sistem care sa inlocuiasca munca noastra de a uda gazonul, intrucat este mult mai eficient si mai usor ca un sistem electronic sa se ocupe de acest lucru. Asa ca, pentru a automatiza, am dorit sa concep un circuit care sa ude gazonul, doar daca umiditatea solului este foarte mica, tinand cont si de luminozitatea de afara, precum si daca sistemul este pornit/oprit de catre cel care il utilizeaza.

Pentru ca gazonul sa creasca in conditii optime, umiditatea solului trebuie sa fie de 60 – 80%. Astfel, daca umiditatea scade sub acest prag, va porni sistemul de irigare.

Puteam aborda problema in urmtorul fel: daca umiditatea e mai mica de 70%, sa pornesc pompa de irigare si daca umiditatea e mai mare de 70% sa o opresc, insa am considerat aceasta abordare destul de problematica.

Sa presupunem urmtorul scenariu: Initial, umiditatea e mai mica de 70%, deci pompa va porni pana cand umiditatea ajunge mai mare decat 70% (de exemplu 70.01%), dupa care se va opri. Dupa ce se opreste, umiditatea scade pana la 69.99% si iar porneste, apoi umiditatea iar creste si pompa se va opri. Datorita faptului ca 69.99% e foarte aproape de 70.01%, pompa se va porni si opri extrem de des.

Cum am rezolvat asta?

- Stabilind un interval in care umiditatea solului poate jongla, pentru ca pompa sa nu faca porniri si opriri dese. Acesta va fi chiar 60 – 80 %.

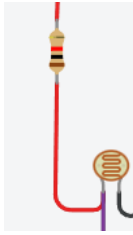


1. Daca umiditatea e mai mica decat, 60 va trebui sa pornim pompa.
2. Daca umiditatea e mai mare de 80 vom opri pompa.

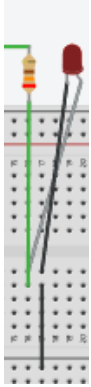
In acest mod, pompa nu va mai face porniri si opriri foarte dese. Umiditatea va creste pana la 80%, pompa se opreste si se porneste doar cand scade pana la 60%.

In plus, am vrut ca pompa sa nu se porneasca in timpul zilei, deoarece irigarea e de mult mai mare calitate daca se face mai spre seara/in timpul noptii, pentru ca patrunde mai multa apa in sol si mai putina se evapora. De aceea, am introdus in sistem si un fotorezistor (a carui rezistenta variaza in functie de intensitatea luminii care cade pe el. Cand este expus la lumina, rezistenta scade, iar cand este in intuneric, rezistenta creste.) Aceste doua conditii de umiditate si luminozitate, s-au reflectat in cod in sectiunile in care am comparat umiditatea cu 60%, 80 %,respectiv lumina cu 50.000 lucsi (lx) (adica vreau sa porneasca cand am o luminozitate scazuta afara). Motorul nu va porni daca lumina nu este mai mica de 50.000 lucsi (lx), indiferent de umiditatea solului. Maparea la mine in cod este 0

Senzor de umiditate (pentru a prelua umiditatea solului)



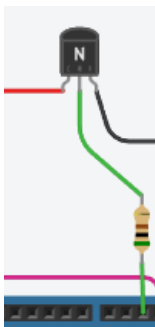
Fotorezistenta, pentru a prelua informatii despre procentajul de luminozitate al mediului.



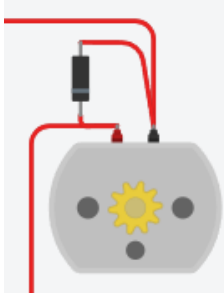
Un led si o rezistenta conectate in serie, pentru a indica daca pompa de irigare este pornita/oprita.



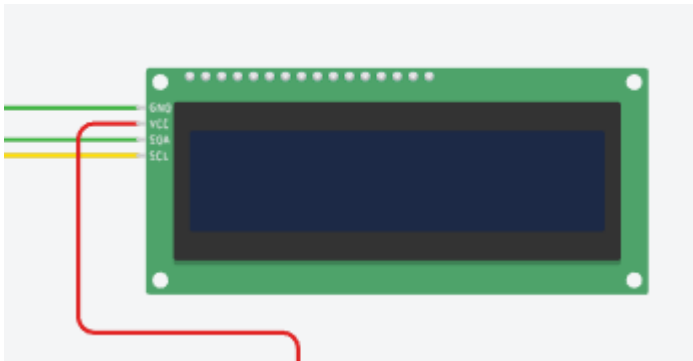
Un buton, prin care utilizatorul poate inchide sistemul, pentru a nu mai porni pompa de irigare pe baza informatiilor preluate de la senzori.



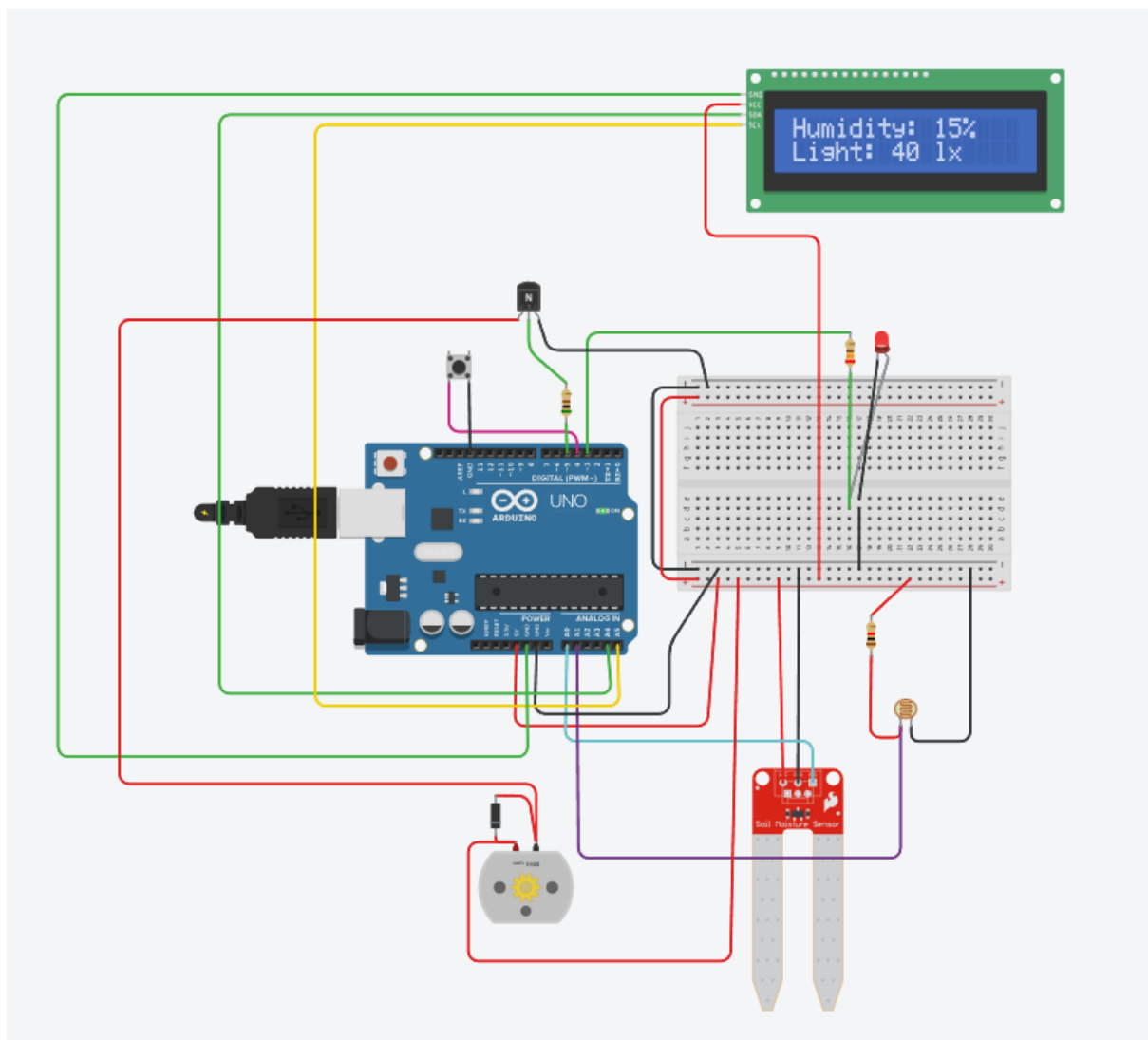
Un tranzistor NPN.



Un Motor DC si o dioda. Explicatii despre schema si design, in sectiunea aferenta lui, de mai jos.



Un afisaj LCD, pentru a afisa informatii despre informatiile preluate de la senzori.



PARTEA B: Parametrii aferenti componentelor:

1. Microcontroller-ul de pe placa Arduino UNO

-> **Model:** ATmega328P

-> **Tensiune de alimentare (VCC):** 1,8 V - 5,5 V (pe placa, este alimentat cu 5 V stabilizat)

-> **Frecventa nominala de lucru:** 16 MHz (cristal de cuarț extern)

28.3 DC Characteristics

$T_A = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$, $V_{CC} = 2.7\text{V}$ to 5.5V (unless otherwise noted)

Parameter	Condition	Symbol	Min.	Typ. ⁽²⁾	Max.	Units
Power supply current ⁽¹⁾	Active 4MHz, $V_{CC} = 3\text{V}$	I_{CC}		1.5	2.4	mA
	Active 8MHz, $V_{CC} = 5\text{V}$			5.2	10	mA
	Active 16MHz, $V_{CC} = 5\text{V}$			9.2	14	mA
	Idle 4MHz, $V_{CC} = 3\text{V}$			0.25	0.6	mA
	Idle 8MHz, $V_{CC} = 5\text{V}$			1.0	1.6	mA
	Idle 16MHz, $V_{CC} = 5\text{V}$			1.9	2.8	mA
Power-down mode ⁽³⁾	WDT enabled, $V_{CC} = 3\text{V}$				44	μA
	WDT enabled, $V_{CC} = 5\text{V}$				66	μA
	WDT disabled, $V_{CC} = 3\text{V}$				40	μA
	WDT disabled, $V_{CC} = 5\text{V}$				60	μA

-> **Curent consumat in mod "Active"** (16 MHz, $V_{CC} = 5\text{V}$):

- $ICC = 9,2 \text{ mA} \text{ -- } 14 \text{ mA}$ (valoare tipica $\approx 12 \text{ mA}$)

->**Curent consumat in mod "Idle"** (16 MHz, VCC = 5 V):

- $ICC = 1,9 \text{ mA} \text{ -- } 2,8 \text{ mA}$ (valoare tipica $\approx 2,4 \text{ mA}$)

->**Curent consumat in modul "Power-down":**

- $ICC = 40 \text{ }\mu\text{A} \text{ -- } 60 \text{ }\mu\text{A}$

->**Limite de tensiune pe pinii I/O:** 0 -- VCC (maxim 5 V)

->**Curent maxim per pin digital:** 20 mA (cu limita totala pe grupurile de porturi)

->**Datasheet:** https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

2. Arduino Uno

->**Sursa de 5 V:**

De pe USB (5 V \pm 5 %) sau de la regulatorul liniar de pe jack/VIN (7 - 12 V input => 5 V)

->**Frecventa de ceas:**

Cristal de cuarț 16 MHz => semnalul de ceas (CLK) intern pentru ATmega328P

->**Conexiune MCU (microcontrollerul) , alimentarea pe placa:**

ATmega328P primește 5 V (VCC) de la regulator, iar pinii AVCC si AREF sunt conectati tot la 5 V pentru ADC (cu condensatori pentru filtru)

GND comun = 0 V

->**Fisa tehnica:**

<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

3. Senzor de umiditate a solului (rezistiv si analogic)

-> **Model:** FSEN0114

-> **Tensiune de alimentare:** 3,3V -- 5V(in cazul meu e alimentat la 5V)

-> **Curent de functionare:** 35 mA

-> **Tensiune de iesire analogica:** 0 V - 4,2 V

->**Intervalele de citire:**

0 - 300 = sol foarte uscat

300 - 700 = sol umed

700 - 950 = sub apa

->Fisa tehnica: <https://www.datasheethub.com/fc-28-soil-moisture-sensor-module/>

Am consultat datasheetul de pe site-ul de mai sus, dand butonul download -> modelul SEN0114.

chrome-extension://efaidnbmnnnibpcajpcgclefindmkaj/https://www.datasheethub.com/wp-content/uploads/2022/08/SEN0114_Web.pdf

4. Fotorezistor GL5528

-> **Rezistenta in intuneric (< 0,1 lux):** $\geq 1 \text{ M ohmi}$

-> **Rezistenta la lumina (> 10 lux \approx 25 grade Celcius):** Intre 8 kOhmi – 20 kOhmi

-> **Tensiunea de alimentare:** 5V (divizor de tensiune)

-> **Tensiunea maxima de alimentare:** 150V

-> **Putere disipata maxima:** 100 mW

-> **Spectru de raspuns:** 540 nm

-> **Curent maxim estimat:** daca LDR scade la 8 kohmi (caz de lumina foarte puternica), atunci curentul prin rezistor va fi aproximativ: $(5 \text{ V}) / (1 \text{ kohmi} + 8 \text{ k ohmi}) \approx 0,56 \text{ mA}$.

OBS: In situatia extrema in care LDR ar ajunge la 0 ohmi (ipoteza teoretica), s-ar consuma 5 mA ($5 \text{ V} / 1 \text{ k ohmi}$), iar in intuneric: LDR $\approx 1 \text{ Mohmi} \Rightarrow$ curentul prin resistor $\approx 0 \text{ mA}$.

->**Range-ul de temperatura la care poate opera:** 30 -- 70 grade celsius

OBS: Rezistenta de 1kOhm pe care am conectat-o in serie cu fotorezistorul asigura un divizor stabil pentru citirea analogica.

->Fisa tehnica: <https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/SEN-09088.pdf>

5. Afisaj LCD 16x2 cu interfata I2C (PCF8574)

-> **Tensiune de alimentare:** 5V

-> **Controller HD44780 (fara backlight):** 0,7 mA -- 2mA

-> **Curent de functionare:** aproximativ 1 mA (fara backlight)

-> **Curent backlight (un LED alb):** $\approx 20 \text{ mA}$

-> **Interfata:** I2C (cu adresa implicita: 0x27):

In schema tipica a modulelor LCD 16x2 cu I2C, expander-ul folosit este PCF8574 care preia comunicatia de pe magistrala I2C (SDA/SCL) si se ocupa cu liniile de date (D4 - D7), RS si E ale controller-ului HD44780, precum si semnalul de backlight. **Tensiunea de**

alimentare este de: 5 V (mergand direct cu LCD si ATmega328P) si **curentul consumat** de 0,1 mA pana la 1 mA (in functie de cate linii sunt HIGH).

-> **Fisa tehnica:** https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf
https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615

Linkul de mai sus, pentru a stii cum duc conexiunile catre ce pini de pe placuta arduino.

6. Buton de apasare (Push-button de 6 * 6 mm)

-> **Tip:** Normally Open

-> **Curent consumat (pull-up intern):** aproximativ 50 mA 24VDC

-> **Rezistenta contact:** $\leq 0,1$ ohmi

-> **Consum propriu:** neglijabil cand nu e apasat, iar la apasare face doar contact intern, foarte low.

-> **Fisa tehnica:**

https://components101.com/sites/default/files/component_datasheet/Push-Button.pdf

7. Motorul DC / Pompa de apa QR30B. Prin motorul DC, eu am simulat o pompa de apa, dar neavand o astfel de componenta dedicata in tinkercad, am folosit motorul DC. Mai jos am enumerat cateva dintre specificatiile unei pompe de apa, pe care am vrut sa o „simulez” in sistemul meu.

-> **Tensiune de alimentare:** 5 V DC

-> **Puterea:** 2.3W

-> **Curent de functionare:** aproximativ 416 mA , la 5V. (functioneaza numai noaptea, daca este cazul, conform parametrilor si conditiilor stabilite)

-> **Debit:** 220 L/h

-> **Varf pornire:** pana la 500 mA -- 600mA, pentru cateva milisecunde

-> **Dimensiuni fizice:** 54 mm (lungime) * 37 mm (latime) * 42 mm (inaltime)

-> **Grad de protectie:** IP68 (protectie totala impotriva prafului si apei adanci)

-> **Tip motor:** Motor brushless cu ax ceramic (pentru rulara silentioasa si o durata lunga de viata > 20 000 – 30 000 ore)

-> **Temperatura de functionare:** 0 grade Celsius - 100 grade Celsius (este rezistent la apa calda pana la 100 grade C)

-> **Zgomot** \approx 40 dB

-> **Fisa tehnica:**

<https://www.newegg.com/p/2YK-005A-007M3?srltid=AfmBOoq9ihSSFCm4lg4OSQkEbZrRIU5iPaasRIN50F2gwVCQWJDV9IGx>

8. LED rosu 5 mm (WW05A3SRP4-N) + rezistenta conectata, de 230 ohmi

->LED rosu:

V_f (tensiune directa): min. 2.0 V, tipic 2,4 V => 20 mA

Curent nominal I_F : 20 mA (max luminozitate)

Unghi de emisie: 15°

->Rezistenta serie (230 ohmi):

Calcul curent: $(5\text{ V} - 2,2\text{ V}) / 230\text{ ohmi} \approx 12\text{ mA}$ (consider $V_f = 2.2\text{ V}$, media dintre capete)

Putere disipata: $\approx (0,012\text{ A})^2 \times 230\text{ ohmi} \approx 0,033\text{ W}$ (sub 0,25 W)

Rezistenta limitatoare am conectat-o in serie cu LED-ul.

-> Fisa tehnica: <https://descargas.cetronic.es/WW05A3SRP4-N%20.pdf>

9. Tranzistor NPN P2N2222A + rezistor baza 500 ohmi

->Tranzistor P2N2222A (TO-92):

-> V_{CEO} (colector-emitor) max: 40 V

-> I_C continuu max: 600 mA

-> V_{CE} (saturatie): 0,2 V - 0,3 V (la $I_C = 150\text{ mA}$, $I_B = 15\text{ mA}$)

->Putere disipata P_D : 625 mW (@25 °C)

->Rezistor baza ($R_4 = 500\text{ ohmi}$):

-> $V_{BE} \approx 0,7\text{ V} \Rightarrow$ tensiunea pe R_4 : $5\text{ V} - 0,7\text{ V} = 4,3\text{ V} \Rightarrow I_B \approx 4,3\text{ V} / 500\text{ ohmi} \approx 8,6\text{ mA}$

Functia lui: comanda deschiderea/inchiderea pompei -> a motorului DC.

Fisa tehnica: <https://www.onsemi.com/pdf/datasheet/p2n2222a-d.pdf>

10. Dioda flyback (1N4001)

Tip: 1N4001 (dioda redresor siliciu)

Este montata invers pe motor pentru protectie la spike-uri inductive atunci cand motorul se opreste/porneste brusc, sa nu risc sa se arda.

- > **Tensiune inversa maxima (V_R):** 50 V
- > **Curent direct continuu (I_F):** 1 A (max)
- > **Curent varf (I_{FSM}):** 30 A (puls 8,3 ms)
- > **Cadere directa (V_F):** $\approx 0,7\text{ V} - 1,1\text{ V}$ (in functie de curent)
- > **Curent invers (I_R):** $\leq 5\text{ }\mu\text{A}$ la 50 V
- > **Rolul in schema:** ofera o cale de revenire a curentului cand tranzistorul se opreste, protejandu-l de tensiuni negative inductive.

Fisa tehnica: <https://www.vishay.com/docs/88503/1n4001.pdf>

11. Rezistoare pasive aditionale

- > **R1 (230 ohmi, LED):** $1/4\text{ W} \Rightarrow$ disip. maxima $\approx 0,033\text{ W}$ la 12 mA
 - > **R2 (1 kohmi, divizor cu LDR):** $1/4\text{ W} \Rightarrow$ disip. max $\approx 0,025\text{ W}$ (5 mA)
 - > **R3 (500 ohmi, baza tranzistor):** $1/4\text{ W} \Rightarrow$ disip. maxima $\approx 0,037\text{ W}$ (8,6 mA)
-

Mai jos explic succint conexiunile pe care le-am realizat, practic, in tinkercad, pentru motorul DC, tranzistor si acea dioda dintre bornele tranzistorului, pentru ca aici mi- a luat putin timp pana cand mi-am dat seama cum sa realizez niste conexiuni corecte, a.i motorul sa functioneze corect. In rest, totul a fost straight-forward, intrucat si pentru afisajul LCD am gasit un tutorial pe youtube in care am vazut cum ar trebui sa il conectez la placuta. Linkul acestuia este cel de mai jos:

<https://www.youtube.com/watch?v=vJ-dvjeMJXI&t=1114s>

(de la minutul 17:40); iar pentru senzorii folositi, a fost destul de explicit din descrierile lor din tinkercad.

Pentru tranzistorul NPN: Colectorul e conectat la unul dintre firele motorului, emitorul conectat la GND, iar baza conectata la pinul digital 5 al placii Arduino, printr-un rezistor de 500 de ohmi. Astfel, placuta va putea controla tranzistorul, deci si motorul.

Pentru motorul DC: Un fir l-am conectat la 5V, iar pe celalalt la colectorul tranzistorului, pentru ca „pompa” sa se inchida/deschida in functie de starea tranzistorului.

Dioda de protectie: Am pus-o pentru ca mi s-a parut ca placuta arduino si tranzistorul ar trebui protejate atunci cand motorul se opreste brusc. Astfel ca am legat anodul(-) diodei la colectorul tranzistorului si catodul(+) la 5V de pe breadboard. In acest fel, dioda va absorbi varfurile de tensiune produse de motor cand se opreste.

Daca in cod programul va ajunge in punctul in care se va scrie HIGH pe pinul 5, atunci va trece curent prin baza, deci tranzistorul conduce, a.i pompa va porni. In caz contrar, cand ii va asigna starea LOW, tranzistorul se inchide, deci si pompa se opreste.

Partea C: Consumul total al sistemului:

Ideea: tensiunea de alimentare este $V = 5\text{ V}$. Pentru fiecare componenta, memorez curentul care trece prin aceasta, apoi calculez puterea aferenta: $P = U \cdot I$.

Voi calcula pt scenariul in care pompa este oprita (motorul nu trage curent), de asemenea pentru cel in care pompa este pornita (motorul trage curent, iar tranzistorul de comanda este in saturatie)

In fiecare scenariu, insumez curentii tuturor componentelor active si calculez curentul total => apoi, puterea totala.

Nr.Crt	Componenta	V	I (mA)	Observatii si conditii
1	ATmega328P (μC de pe Arduino UNO)	5 V	12	Curent tipic in „Active” (16 MHz, 5 V)
2	Senzor umiditate sol (SEN0114)	5 V	35	Spec. producator: 35 mA la 5 V
3	Divizor LDR (fotorezistor GL5528 + rezistor 1 kohmi)	5 V	0,56	In lumina puternica: $R_{LDR} \sim 8\text{ kohmi} \Rightarrow I \sim 0,56\text{ mA}$
4	LCD 16×2 cu interfata I2C (PCF8574 + HD44780)	5 V	21	1 mA (controller) + 20 mA (backlight).
5	LED rosu + rezistor 230 ohmi	5 V	12	Calcul: $(5\text{ V} - 2,2\text{ V}) / 230\text{ ohmi} \sim 12\text{ mA}$
6	Rezistor la baza (500 ohmi) conectat la tranzistorul P2N2222A	5 V	8,6 (doar cand pompa e pornita)	Daca tranzistorul conduce: $I_b \sim (5 - 0,7) / 500 \sim 8,6\text{ mA}$
7	Motor DC / pompa apa QR30B	5 V	416 (continuu) peak 500–600	Curent continuu: 416 mA Varf la pornire: intre 500 – 600 mA (cateva ms)
8	Buton (pull-up intern Arduino)	5 V	~ 0	Pull-up intern => consum neglijabil la 5 V

1) Pompa oprita:

In aceasta stare, motorul nu trage curent si rezistorul de baza nu consuma (tranzistorul P2N2222A este in blocare)

1. μC ATmega328P: $I_1 = 12\text{ mA}$ si $P_1 = 5\text{ V} \cdot 12\text{ mA} = 60\text{ mW}$
2. Senzorul umiditate sol: $I_2 = 35\text{ mA}$ si $P_2 = 5\text{ V} \cdot 35\text{ mA} = 175\text{ mW}$
3. Divizor LDR + 1 kohmi : $I_3 \sim 0,56\text{ mA}$ si $P_3 = 5\text{ V} \cdot 0,56\text{ mA} = 2,8\text{ mW}$
4. LCD 16×2 (I2C): $I_4 = 21\text{ mA}$ si $P_4 = 5\text{ V} \cdot 21\text{ mA} = 105\text{ mW}$

5. LED-ul rosu conectat cu rezistenta de 230 ohmi:

$$I_5 = 12 \text{ mA} \quad \text{si} \quad P_5 = 5V * 12 \text{ mA} = 60\text{mW}$$

6. Rezistorul la baza (500 ohmi): $I_6=0 \text{ mA}$ (tranzistorul blocat) si $P_6=0 \text{ mW}$

7. Motor DC / pompa: $I_7=0 \text{ mA}$ si $P_7 = 0\text{mW}$.

8. Butonul (pull-up intern): $I_8 \approx 0 \text{ mA}$ si $P_8 = 0 \text{ mW}$.

Deci, in situatia cu pompa oprita, suma curentilor este: $12 + 35 + 0,56 + 21 + 12 \approx 80,56 \text{ mA}$

Deci, puterea totala este: $5V * 80,56 \text{ mA} = 402,8 \text{ mW}$ (deci aproximativ $0,40 \text{ W}$).

2) Pompa pornita:

In acest scenariu motorul trage curent (416 mA continuu) si tranzistorul este saturat (rezistorul de la baza consuma $8,6 \text{ mA}$)

1. μC ATmega328P: $I_1 = 12 \text{ mA}$ si $P_1=5V * 12 \text{ mA} = 60 \text{ mW}$

2. Senzorul umiditate sol: $I_2 = 35 \text{ mA}$ si $P_2=5V * 35 \text{ mA} = 175 \text{ mW}$

3. Divizor LDR + 1 kohmi : $I_3 \approx 0,56 \text{ mA}$ si $P_3=5V * 0,56 \text{ mA} = 2,8 \text{ mW}$

4. LCD 16×2 (I2C): $I_4 = 21 \text{ mA}$ si $P_4 = 5V * 21\text{mA} = 105\text{mW}$

5. LED-ul rosu conectat cu rezistenta de 230 ohmi:

$$I_5 = 12 \text{ mA} \quad \text{si} \quad P_5 = 5V * 12 \text{ mA} = 60\text{mW}$$

6. Rezistorul la baza tranzistorului (500 ohmi): $I_6 = 8,6\text{mA}$, $P_6 = 5V * 8,6\text{mA} \approx 43 \text{ mW}$

7. Motorul DC / pompa: $I_7 = 416 \text{ mA}$, $P_t = 5V * 416 \text{ mA} = 2,08 \text{ W}$

8. Butonul: $I_8 \approx 0\text{mA}$ si $P_8 = 0 \text{ mW}$

Deci, in situatia cu pompa pornita, suma curentilor este: $12 + 35 + 0,56 + 21 + 12 + 8,6 + 416 + 0 = 505,16 \text{ mA}$.

Deci, puterea totala este: $5V * 505,16 \text{ mA} = 2,5258 \text{ W}$ (deci aproximativ $2,53 \text{ W}$).

Obs: In primele milisecunde, motorul poate trage un varf de aproximativ $500 - 600 \text{ mA}$, deci daca as include un astfel de varf, atunci:

Curentul maxim ar fi 500 mA (motor) + $12 + 35 + 0,56 + 21 + 12 + 8,6 \approx 589,16 \text{ mA}$.

Deci puterea maxima ar fi: $P_{\text{peak}} = 5V * 0,589 \text{ A} \approx 2,945 \text{ W}$ (aproximativ $2,95 \text{ W}$). Dar acest varf va dura doar cateva milisecunde, dupa care consumul va scadea la valoarea 416 mA + restul componentelor

In scenariile de mai sus, nu am mai luat in calcul si consumul intern al placii Arduino UNO (cu regulatorul de 5 V , ledurile, si conversorul USB-serial), dar daca ar fi sa adaug si acest lucru, ar fi la curent: pt pompa oprita: $80,56 \text{ mA} + 60\text{mA}$ adica $140,56 \text{ mA}$ deci puterea aprox. $0,70\text{W}$, iar in cazul cu pompa pornita: $505,16 \text{ mA} + 60 \text{ mA} = 565,16 \text{ mA}$ deci aproximativ $2,83 \text{ W}$.

PARTEA D: Autonomie

Pentru a obține o tensiune de 5V pornind de la acumulatori de 1,2 V e nevoie de un număr de $(5V / 1,2V)$ acumulatori, adică 5. Fiecare acumulator are 2500 mAh, deci pachetul în serie va avea tot 2500 mAh. Fiecare are nominal 1,2 V, deci cinci în serie dau $5 \times 1,2 V = 6 V$. Cum eu am nevoie doar de 5V, atunci va trebui să folosesc un regulator pentru a obține 5V din 6V.

Dar presupun că acel regulator nu face pierderi de tensiune (și tensiunea va rămâne constantă). Dacă alimentez sistemul în starea în care pompa e oprită (cu consumul de aproximativ 80,6 mA), atunci autonomia teoretică este de $2,5 \text{ Ah} / 0,0805 \text{ A}$ deci aproximativ 31 ore. Pe de altă parte, dacă pompa funcționează continuu (consumul total undeva la 505,2 mA) atunci autonomia va scădea la $2,5 \text{ Ah} / 0,5052 \text{ A}$ deci 4,95 ore, aprox. 5.

PARTEA E: Măsuri pe care le-aș putea introduce pentru a reduce consumul energetic

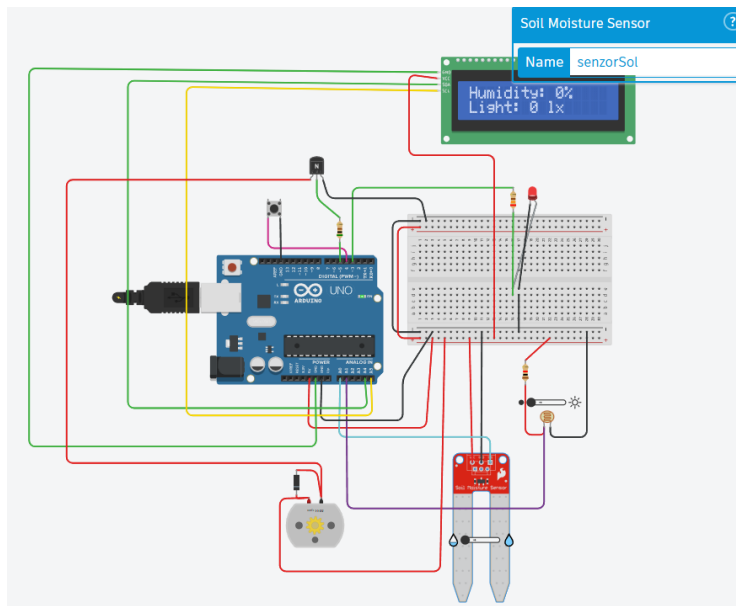
Backlight-ul LCD-ului ar putea fi limitat doar la afișările relevante, adică aș putea să îl aprind pentru 2-3 secunde, atunci când am nevoie să actualizez datele, apoi să îl opresc. Astfel, în repaus, senzorii ar putea fi verificați, dar backlight-ul să rămână oprit.

De asemenea, aș fi putut înlocui BJT-ul cu un MOSFET (N-channel), întrucât tranzistorul pe care l-am ales, cel P2N2222A, consumă cam 8,6 mA doar pe bază atunci când e în saturatie. Un MOSFET nu are curent de "bază", ci doar un curent de gate, care va fi zero, în DC, deci s-ar elimina acei 8,6 mA. Aș fi putut lega sursa MOSFET-ului la GND, drain-ul la borna negativă a pompei și borna pozitivă a pompei direct la +5V. În paralel, o rezistență de aproximativ 100 kohm, de pull-down, ar asigura că MOSFET-ul rămâne oprit atunci când MCU (microcontrollerul) este în reset.

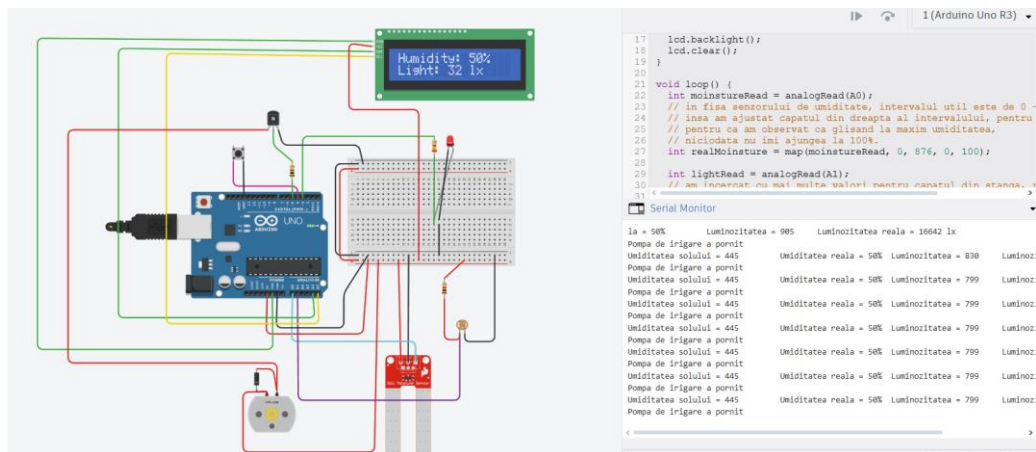
Astfel, avantajul ar fi că economisesc 8,6 mA ori de câte ori pompa e pornită și caderea de tensiune este foarte mică pe MOSFET. Deci pierderile sunt foarte mici (sub 0,1W), comparative cu câteva sute de mW pierdute prin tranzistorul meu P2N2222A conectat cu rezistorul la bază.

PARTEA F: Poze cu sistemul pentru:

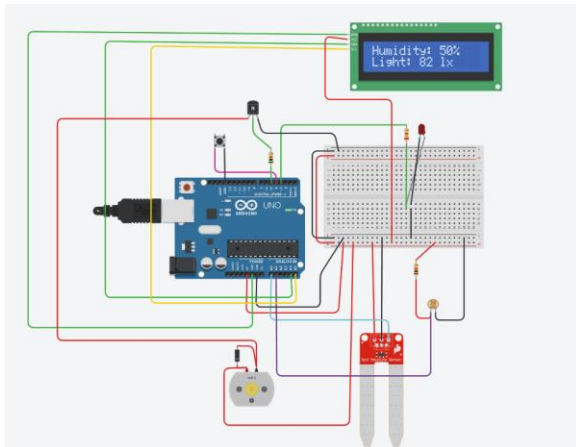
1) Umiditate 0, Luminozitate 0. (luminozitatea scăzută (aproape noapte/noapte) și solul extrem de uscat).



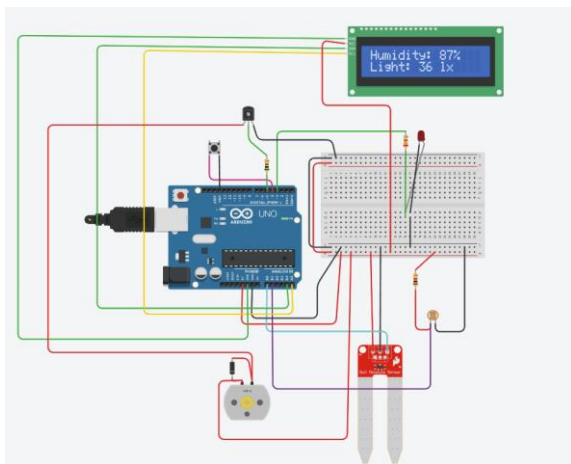
2)Umiditatea 50 la suta, si lumina 32 lux (lx notat in cod) – deci inca sunt in parametrii pentru care pompa sa fie pornita.



3)Umiditatea de 5 la suta, dar inca nu s a facut seara: (se observa cum ledul se opreste, si rotilele motorului s-au oprit.)



4)Umiditatea deja optima, de 87 la suta, luminozitatea este cea la care sistemul poate porni, dar avand deja umiditate optima solul, nu mai porneste motorul, led-ul ramane stins.



PARTEA G: Codul implementat: (pentru lcd am folosit biblioteca si API-ul din LiquidCrystal

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

bool statusSistem;

void setup() {
  pinMode(5, OUTPUT); // Pompa
  pinMode(3, OUTPUT); // LED-ul
  pinMode(A0, INPUT); // senzorul de umiditate
  pinMode(A1, INPUT); // fotorezistor-ul
  pinMode(4, INPUT_PULLUP); // butonul

  Serial.begin(9600);
  statusSistem = true;
  lcd.init();
  lcd.backlight();
  lcd.clear();
}

void loop() {
  int moistureRead = analogRead(A0);
  // in fisa senzorului de umiditate, intervalul util este de 0 - 950
  // insa am ajustat capatul din dreapta al intervalului, pentru solul complet ud,
  // pentru ca am observat ca glisand la maxim umiditatea,
  // niciodata nu imi ajungea la 100%.
  int realMoisture = map(moistureRead, 0, 876, 0, 100);
```

```

28
29 int lightRead = analogRead(A1);
30 // am incercat cu mai multe valori pentru capatul din stanga, repsectiv dreapta,
31 // iar cele de mai jos sunt cele mai potrivite pe care le-am gasit.
32 // astfel ca, in intuneric total, valoarea citita direct de pe pin, de 344,
33 // va deveni pragul pentru 0 lx, iar 1017, pentru valoarea maxima lx, setata la
34 // 100000 lx.
35 // am scazut din 100000 lx, intrucat am vrut sa inversez scalarea mapului.
36 // a.i. intunericul sa fie reprezentat printr-o valoare mare, iar lumina puternica,
37 // printr-una foarte mica (0).
38 long realLight = 100000 - map(lightRead, 344, 1017, 0, 100000);
39
40 if (digitalRead(4) == 0) {
41     statusSistem = !statusSistem;
42     delay(300);
43 }
44
45 Serial.print("Umiditatea solului = ");
46 Serial.print(moistureRead);
47 Serial.print("\t Umiditatea reala = ");
48 Serial.print(realMoisture);
49 Serial.print("\t Luminozitatea = ");
50 Serial.print(lightRead);
51 Serial.print("\t Luminozitatea reala = ");
52 Serial.print(realLight);
53 Serial.println(" lx");
54
55 lcd.clear();
56 lcd.setCursor(0, 0);
57 Serial.print("Umiditate: ");

```

Mai jos, am impartit realLight la 1000 pentru ca am vrut valori usor de interpretat de ochiul uman, intre 0 si 100, pentru luminozitate. De fiecare data cand afisez pe lcd, sterg ce afiseaza in momentul respectiv de timp, setez cursorul in coltul din stanga sus apoi afisez cat e umiditatea, si apoi imi setez cursorul in coltul din stanga jos (pe a doua linie a lcd-ului) si afisez cat e acum luminozitatea, in lucsi – scalata intre 0 si 100.)

```

53 Serial.println(" lx");
54
55 lcd.clear();
56 lcd.setCursor(0, 0);
57 lcd.print("Humidity: ");
58 lcd.print(realMoisture);
59 lcd.print("%");
60
61 lcd.setCursor(0, 1);
62 lcd.print("Light: ");
63 lcd.print(realLight / 1000);
64 lcd.print(" lx");
65
66 if (realMoisture < 60 && realLight < 50000 && statusSistem) {
67     digitalWrite(3, HIGH); // aprind LED-ul
68     digitalWrite(5, HIGH); // pornesc motorul (pompa)
69     Serial.println("Pompa de irigare a pornit");
70 }
71
72 if ((realMoisture > 80 || realLight >= 50000) && statusSistem) {
73     digitalWrite(3, LOW); // sting LED-ul
74     digitalWrite(5, LOW); // opresc pompa
75     Serial.println("Sistemul va lua o pauza!");
76 }
77 if (!statusSistem) {
78     digitalWrite(3, LOW); // opresc LED-ul.
79     digitalWrite(5, LOW); // setez pe LOW pinul pentru pompa,
80                             // adica o opresc
81     Serial.println("Sistemul s-a oprit de catre proprietar.");
82 }


```



```

61  lcd.setCursor(0, 1);
62  lcd.print("Light: ");
63  lcd.print(realLight / 1000);
64  lcd.print(" lx");
65
66  if (realMoinsture < 60 && realLight < 50000 && statusSistem) {
67      digitalWrite(3, HIGH); // aprind LED-ul
68      digitalWrite(5, HIGH); // pornesc motorul(pompa)
69      Serial.println("Pompa de irigare a pornit");
70  }
71
72  if ((realMoinsture > 80 || realLight >= 50000) && statusSistem) {
73      digitalWrite(3, LOW); // sting LED-ul
74      digitalWrite(5, LOW); // opresc pompa
75      Serial.println("Sistemul va lua o pauza!");
76  }
77  if (!statusSistem) {
78      digitalWrite(3, LOW); // opresc LED-ul.
79      digitalWrite(5, LOW); // setez pe LOW pinul pentru pompa,
80                          // adica o opresc
81      Serial.println("Sistemul s-a oprit de catre proprietar.");
82  }
83
84  delay(3000);
85  }
86
87

```

 Serial Monitor

atea reala = 87% Luminozitatea = 771 Luminozitatea reala = 36553 lx
Sistemul va lua o pauza!

Schema bloc:

