

Light meter

Autor: [Andrei Catalin CLICINSCHI](#)

Introducere

Exponometru care masoara intensitatea luminii si propune combinatii de setari (apertura, shutter speed) in functie de ISO-ul selectat si eventuale filtre ND.

Descriere

Schema bloc



Mod de folosire

1. Se introduce valoarea ISO a senzorului foto (analog sau digital)
2. [Optional] Se introduce valoarea atenuarii filtrului ND folosit
3. Se apasa butonul de masurare
4. Se cicleaza printre setarile propuse pana se gaseste o combinatie adecvata
5. Se face poza folosind setarile alese!

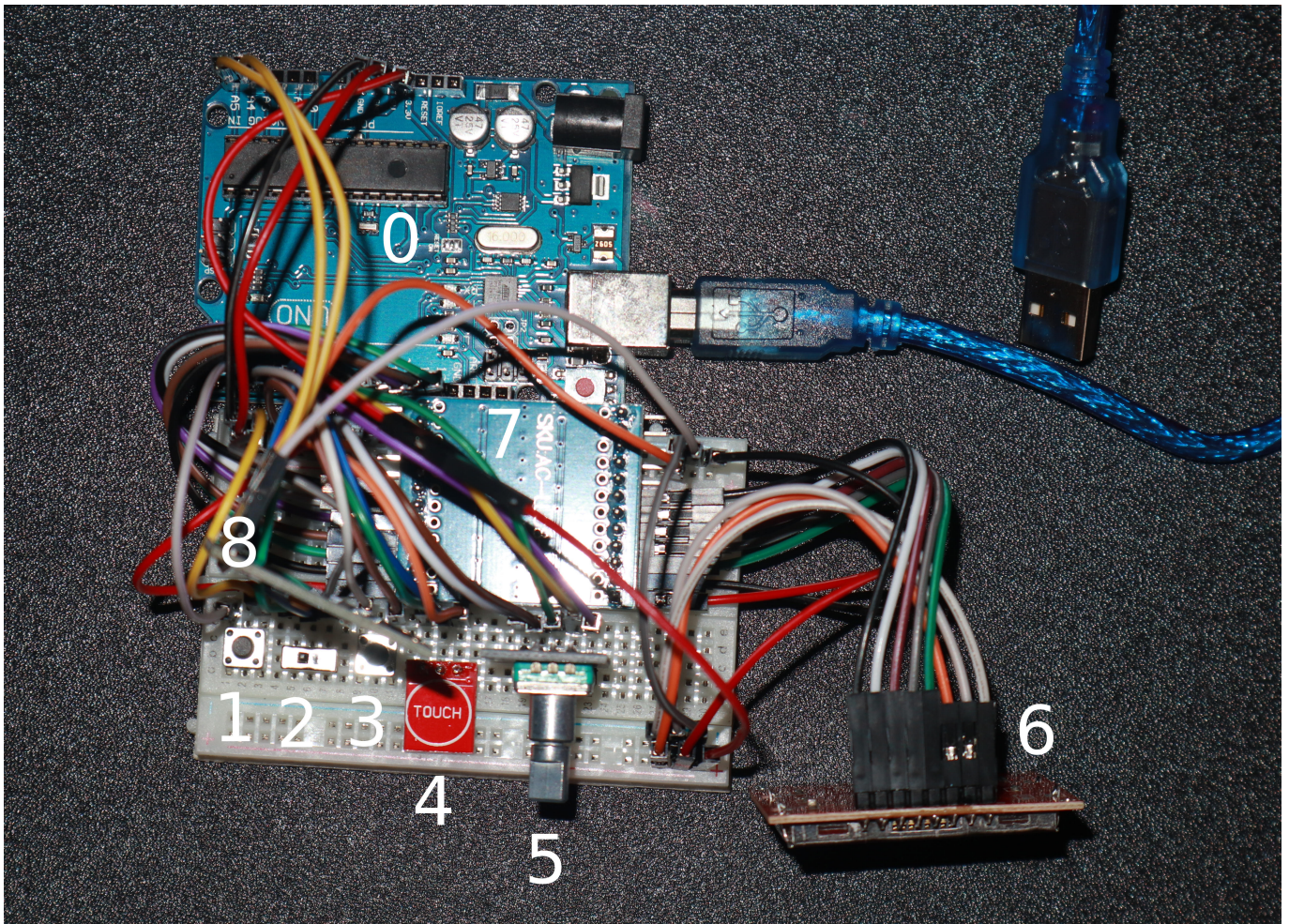
Hardware Design

Piese

- * Arduino Uno R3 ATmega328P : **0**
- * Modul intensitate luminoasa (senzor BH1750) : **8**
- * Ecran Nokia 5110 84×48 (Philips PCD8544) : **6**
- * Modul conversie nivel logic 3.3V - 5V bidirectional cu 8 canale : **7**

- * Butoane simple : **1, 3**
- * Slider switch : **2**
- * Senzor capacitiv TTP223 : **4**
- * Modul encoder rotativ KY-040 : **5**
- * Fire de legatura si breadboard

Schema electrica



Funcțiile butoanelor

1 : Set ISO button

- la prima apasare se activeaza modul de setare ISO
- la a doua apasare se seteaza ISO-ul

2 : Activate ND switch

- in pozitia OFF exponometrul ignora campul ND
- in pozitia ON exponometrul tine cont de campul ND si nu propune setari pana nu este introdusa

valoarea

3 : Set ND button

- daca **2** este OFF, este ignorat
- daca **2** este ON:

- la prima apasare se activeaza modul de setare ND
- la a doua apasare se fixeaza valoarea filtrului ND

4 : Measure button

- la atingere exponometrul masoara si afiseaza EV sau fluxul luminos

5 : Rotary encoder

- Rotire

- in modul de setare ISO, cicleaza printre valorile din vectorul ISO values
- in modul de setare ND, cicleaza printre valorile din vectorul ND values
- daca exista setari afisate, cicleaza printre combinatiile valide de setari corespunzatoare EV-ului masurat

- Apasare

- comuta afisarea intre EV si fluxul luminos masurat in lux

Software Design

Codul este disponibil pe [GitHub](#)

Pentru realizarea proiectului am folosit bibliotecile:

- U8glib.h, pentru afisare pe ecran
- Wire.h si BH1750.h, pentru a lucra cu senzorul de lumina
- math.h pentru cateva functii utile la conversia lux → EV

Ecranul este impartit in 5 regiuni:

- stanga sus → ISO, introdus de utilizator
- dreapta sus → ND filter, introdus de utilizator
- centru → EV_100 sau fluxul luminos, dupa caz, output de la senzorul de lumina
- stanga jos → valoarea aperturii, calculata de arduino
- dreapta jos → valoarea shutter speed-ului, calculata de arduino

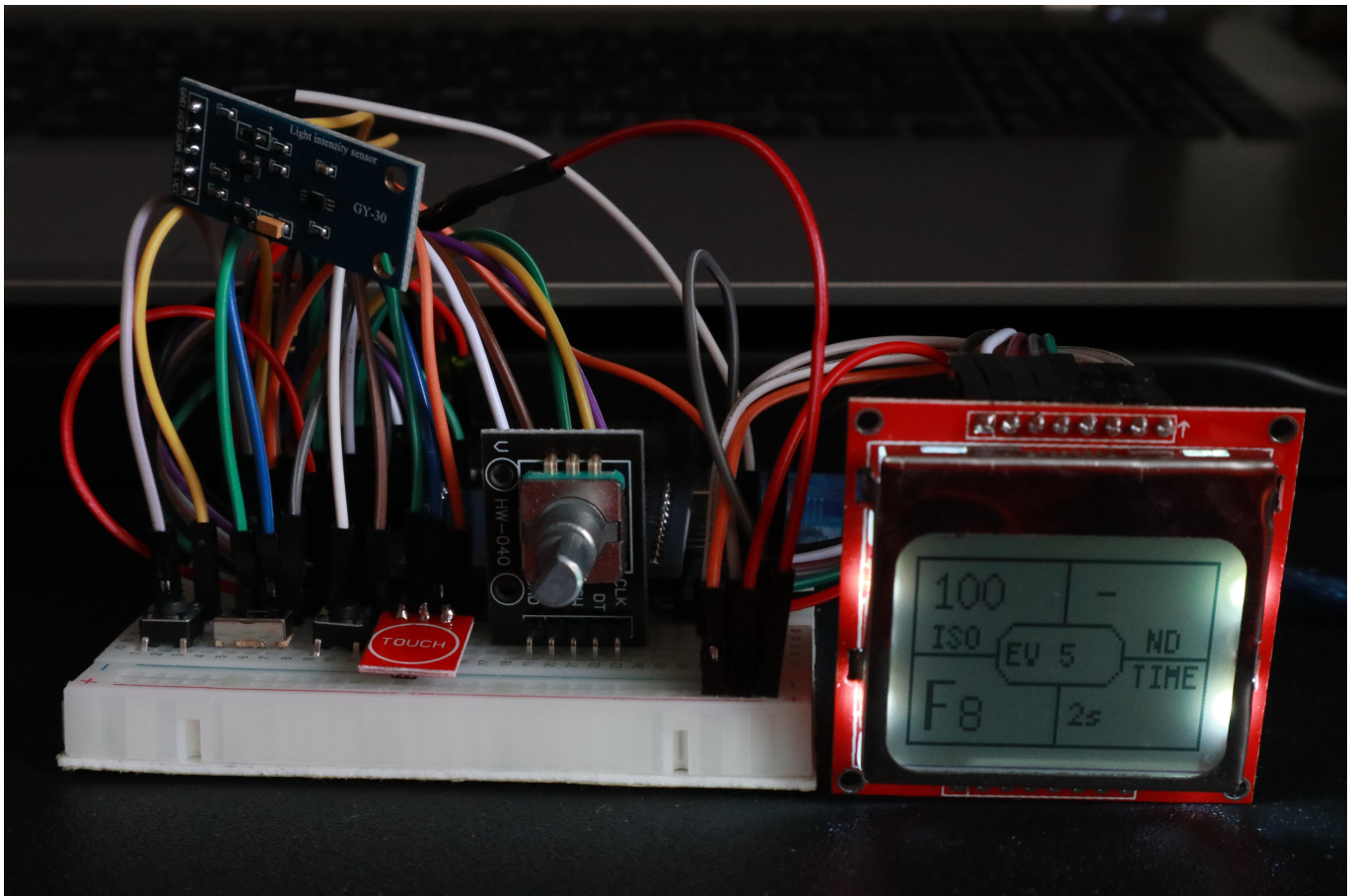
La inceputul programului sunt definite denumiri sugestive pentru utilizarile pinilor, apoi alias-uri pentru starile logice ale butoanelor si switch-ului. Urmeaza apoi variabile care retin starile elementelor hardware de input. Aceste stari (sau tranzitiile lor) determina schimbarea unor variabile de stare asociate exponometrului, nu sunt folosite direct in determinarea comportamentului aparatului. De exemplu, butonul 1, folosit pentru introducerea ISO-ului, are asociat pinul digital definit prin "#define SET_ISO_BUTTON 2", starea "int SET_ISO_BUTTON_state", iar la tranzitia BUTTON_NOT_PRESSED to BUTTON_PRESSED, se face toggle la variabila de stare "bool input_ISO". Aceasta variabila de stare este "false" daca ISO-ul a fost deja selectat (sau inca nu a fost inceputa selectia) si "true" daca exponometrul asteapta ca utilizatorul sa introduca ISO-ul prin encoderul rotativ, afectand comportamentul programului. Pentru a detecta tranzitiile am folosit variabila "int

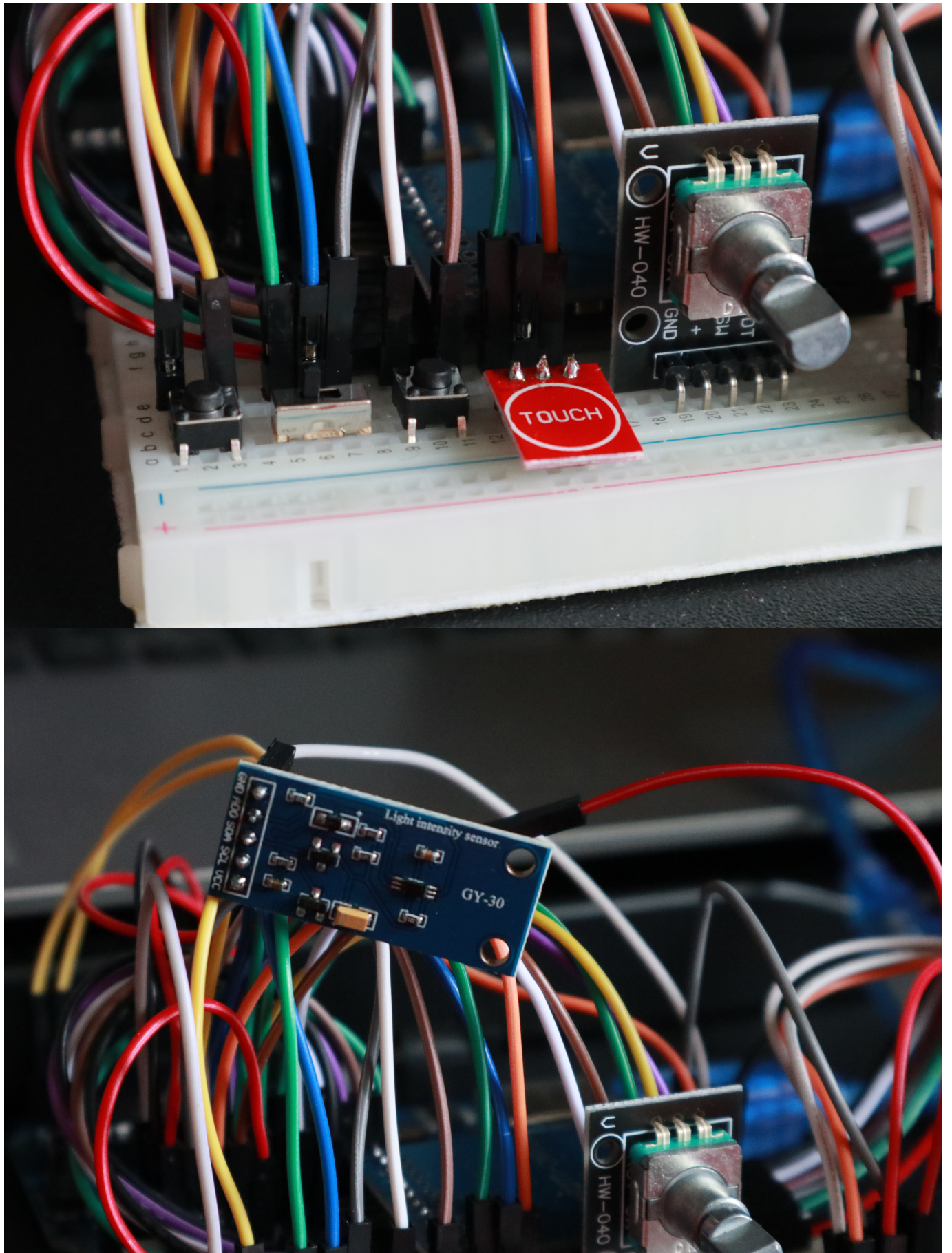
SET_ISO_BUTTON_state_tmp". In mod similar am procedat si cu celelalte butoane hardware, care controleaza variabilele hardware, care la randul lor controleaza variabilele de stare ale exponometrului.

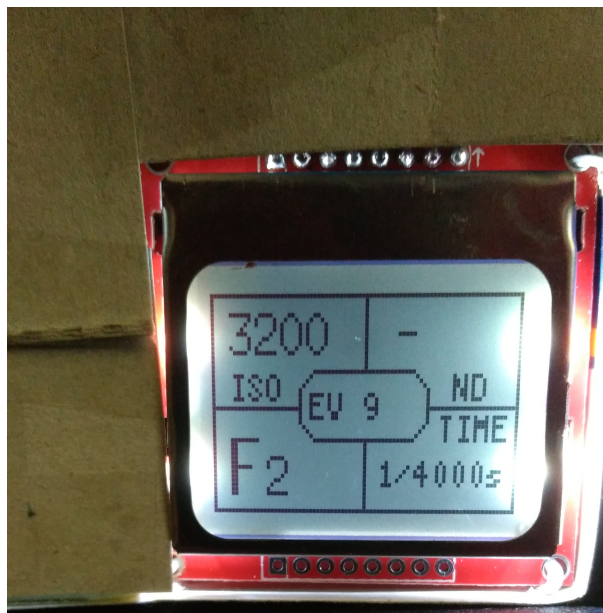
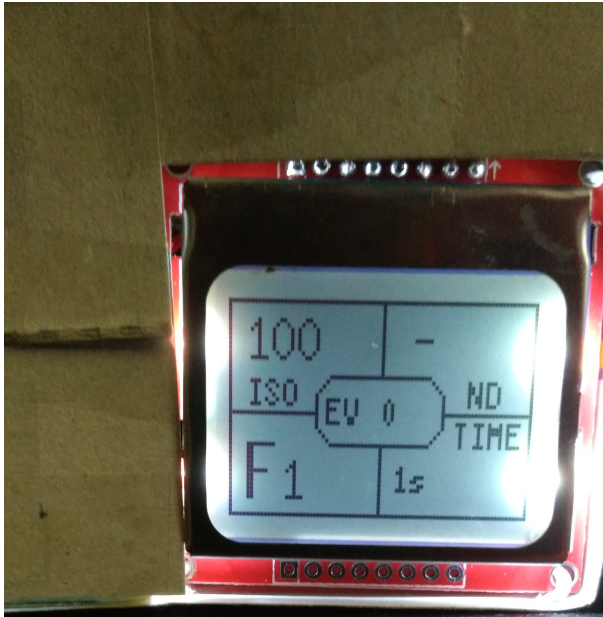
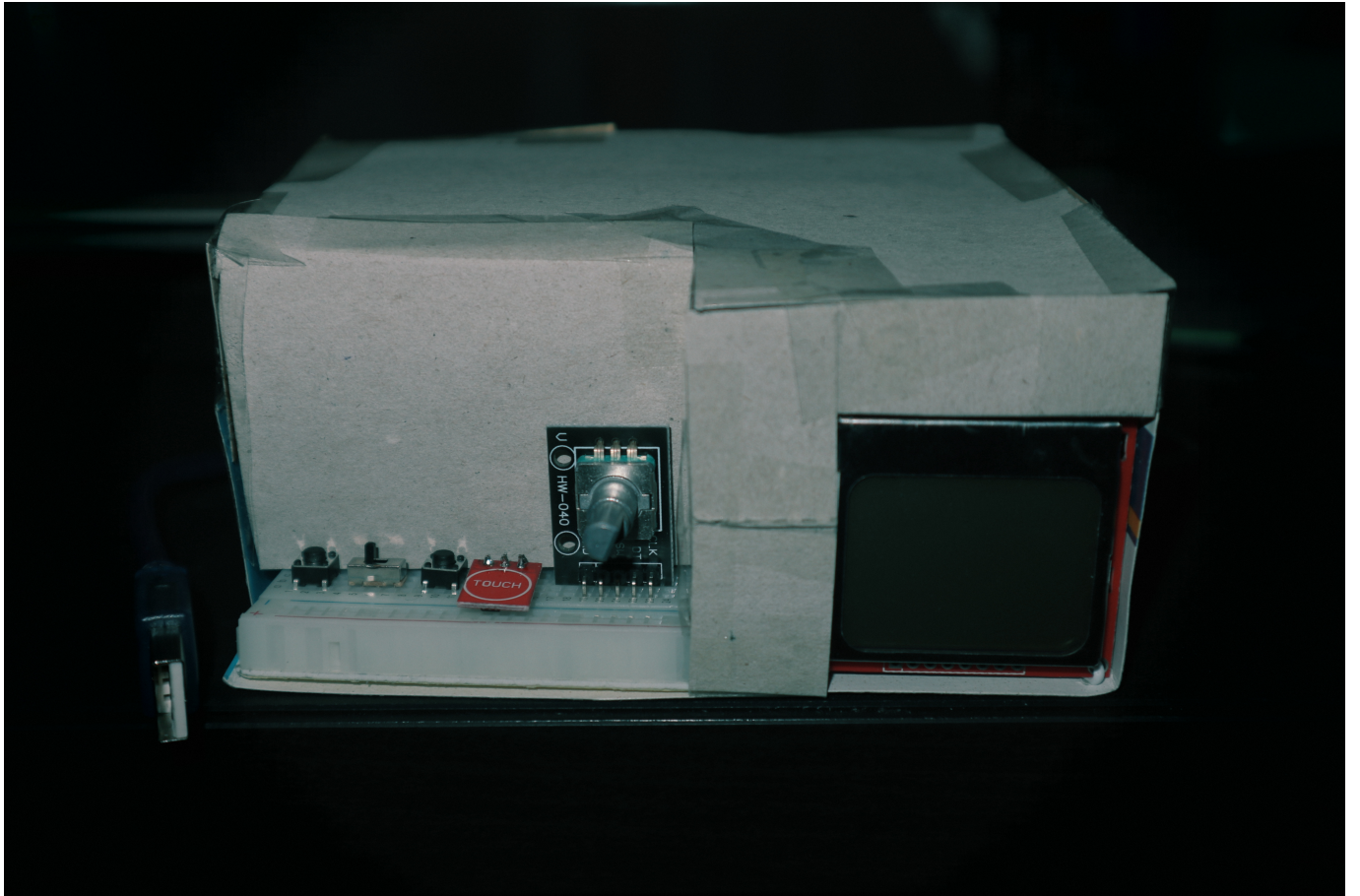
Valorile pentru cele 4 variabile (ISO, ND filter, apertura, shutter speed) sunt memorate in niste vectori, programul avand memorat un index al valorii care trebuie afisata (sau -1 daca trebuie afisat '-'). O modificare a unuia dintre indecsi cu 1 corespunde unui stop de lumina, calculul setarilor fiind foarte usor. Se pleaca de la valoarea cunoscuta $EV_{100} = 0$ (ISO = 100), pentru care apertura = F1 si shutter_speed = 1s, se scrie ecuatia liniara care modeleaza setarile, apoi se incearca pe rand indecsii din vectorul de aperturi si se calculeaza indexul de shutter speed corespunzator. Daca nu se gaseste nicio combinatie valida de indecsi, se afiseaza '-' la ambele setari, altfel se afiseaza prima pereche gasita, corespondenta celei mai mari aperturi posibile si celui mai scurt timp de expunere corespunzator acelei aperturi.

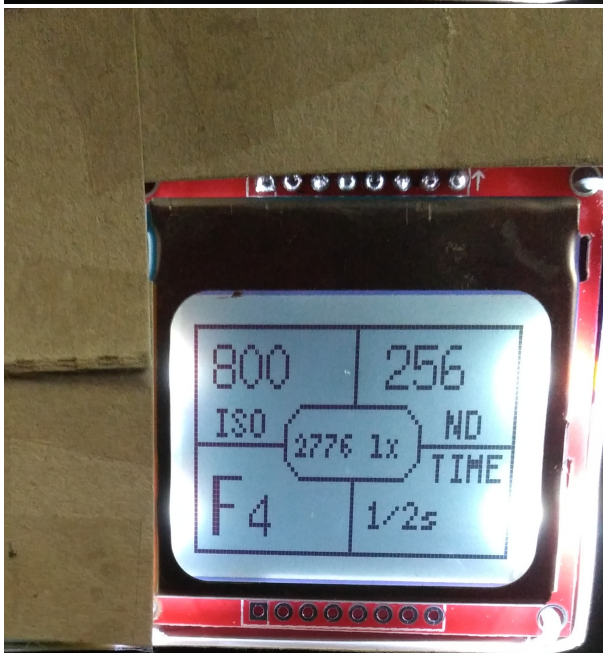
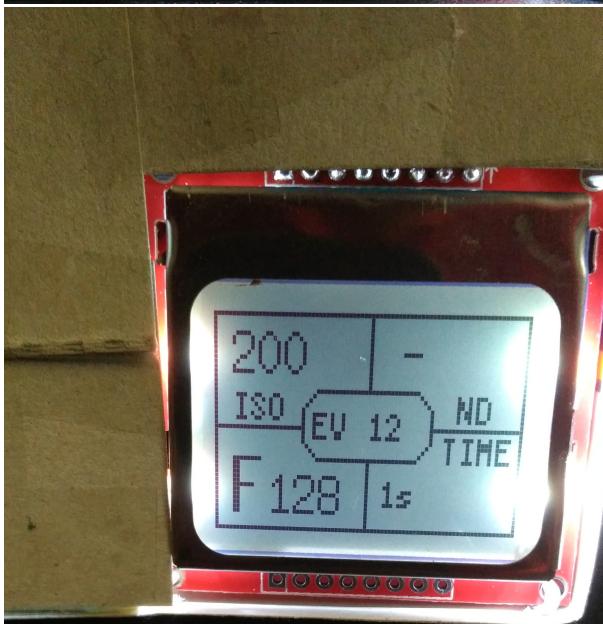
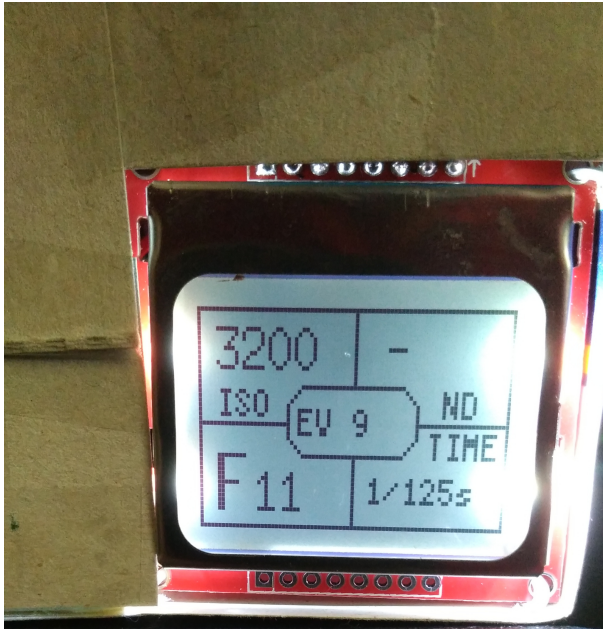
Afisarea se face prima data in interiorul functiei setup(), apoi in loop() de fiecare data cand a fost detectat un input de la utilizator (pentru a optimiza putin consumul de resurse). Starile butoanelor sunt verificate secvential in functia loop(), aplicatia nefiind una de mare viteza nu am avut probleme cu viteza de refresh si nici nu a fost nevoie sa fac debouncing. Pentru modularizarea codului am impartit functia de afisare in partea constanta (layout-ul ecranului) si valorile parametrilor, in functie de indecsi.

Poze









Concluzii

Proiectul a fost destul de usor de implementat, am gasit resurse foarte bune pe internet pentru lucrul cu componentele de care am avut nevoie. Desi nu am facut optimizari deosebite, merge bine si am reusit sa implementez tot ce planuisem de la inceput. Am subestimat cat de multe fire de legatura imi trebuie, asa ca a fost nevoie de o a doua comanda de piese (for reference, pentru un proiect de dimensiunea asta, 30-40 mama-tata si tata-tata si vreo 20 mama-mama ar trebui sa fie ok). Am facut o cutie de carton pentru a proteja componentele, mi-ar fi placut sa fac ceva 3D printed dar nu a mai fost timp.

Am testat exponometrul cu telefonul si cu un mirrorless si valorile obtinute sunt foarte apropiate, in limita de 1EV, diferentele putand aparea din cauza faptului ca senzorul meu este omnidirectional, cuprinde o zona mai mare din scena decat meter-ul TTL (through the lens) de la aparat, din cauza unor diferente la calibrare la conversia lux → EV, din cauza unor diferente subtile de lumina din scena in momentul masurarii, etc.

In schimb, pentru schema electrica, mi s-a parut greu de lucrat cu Eagle, bibliotecile cu componente sunt imprastiate pe multe site-uri si nu corespund intotdeauna cu piesele. De exemplu, nu am reusit sa gasesc schema pentru convertorul de nivel logic (o versiune anterioara de convertor, aveam 2 piese cu cate 2 canale dar ecranul avea nevoie de 5 canale minim, de aceea l-am luat pe asta mare cu 8 canale odata cu cablurile suplimentare) iar schema gasita pentru ecranul PCD8544 avea pinii in alta ordine, asa ca am preferat sa imi desenez singur componentele si legaturile, pe <https://www.diagrams.net/>.

Download

[Download PDF](#)

Bibliografie

Video piese

[Butoane](#)

[Switch](#)

[Buton touch](#)

[Rotary encoder](#)

[Nokia screen 1](#), [Nokia screen 2](#)

[Senzor lumina](#)

Documentatie cod

[U8glib.h](#)
[AVR libc math.h](#)

Info EV

[Conversie lux -> EV](#)
[Conversie EV -> setari](#)
[Filtre ND](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/lightmeter>



Last update: **2021/06/30 18:31**