

# Smart mirror

## Introducere

**Autor:** Dumitrache Adrian-George

**Grupa:** 334CC

Proiectul consta intr-o **oglanda smart** ce afiseaza diverse informatii prin intermediul unui **ecran LCD**. Aceste informatii includ: **ora** actuala, **vremea** de afara si **umiditatea** camerei, cat si **sfaturi** bazate pe acestea (e.g. ai nevoie de o jacheta daca iesi din casa, umiditatea camerei tale este prea mica pentru temperatura de afara). Informatia afisata se poate cicla prin intermediul unui **buton**.

Dispozitivul este perfect pentru diminetile pe fuga in care iti faci rutina de par/machiaj si ai nevoie sa nu pierzi din vedere ora actuala, cat si daca ai nevoie sa iei decizii rapide legate de vestimentatie.

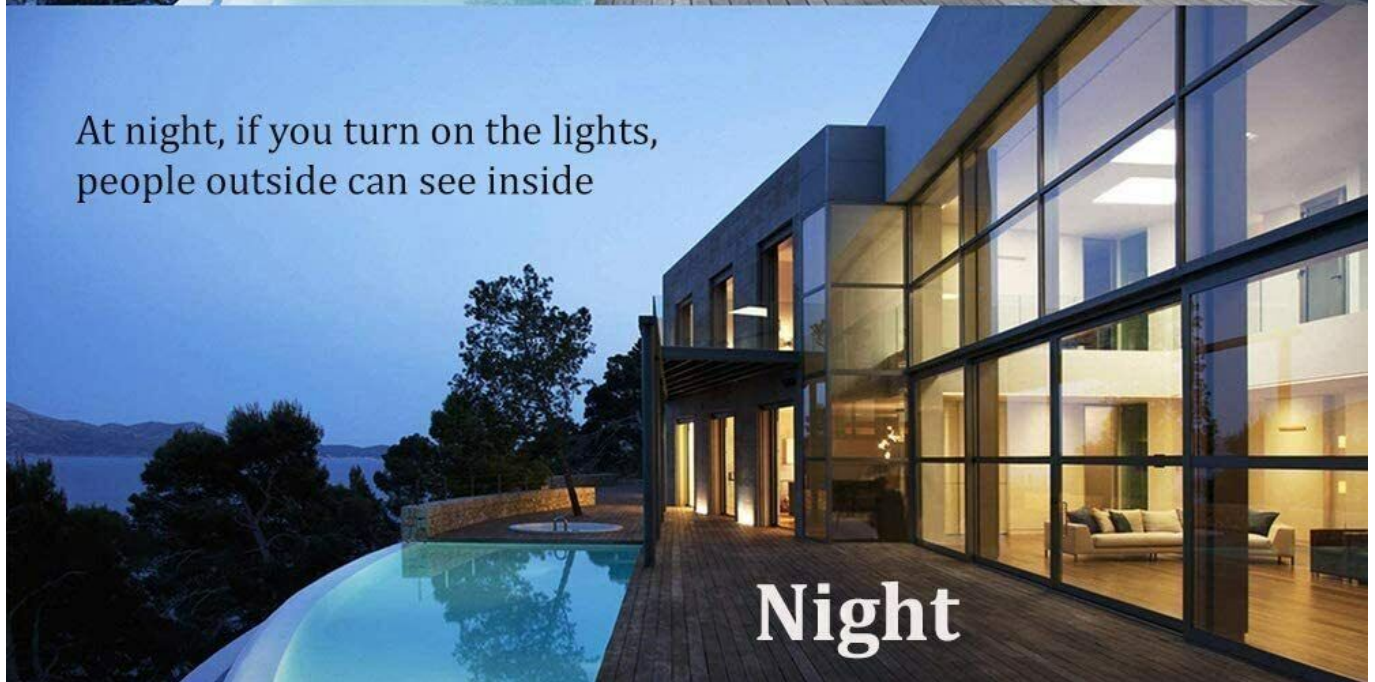
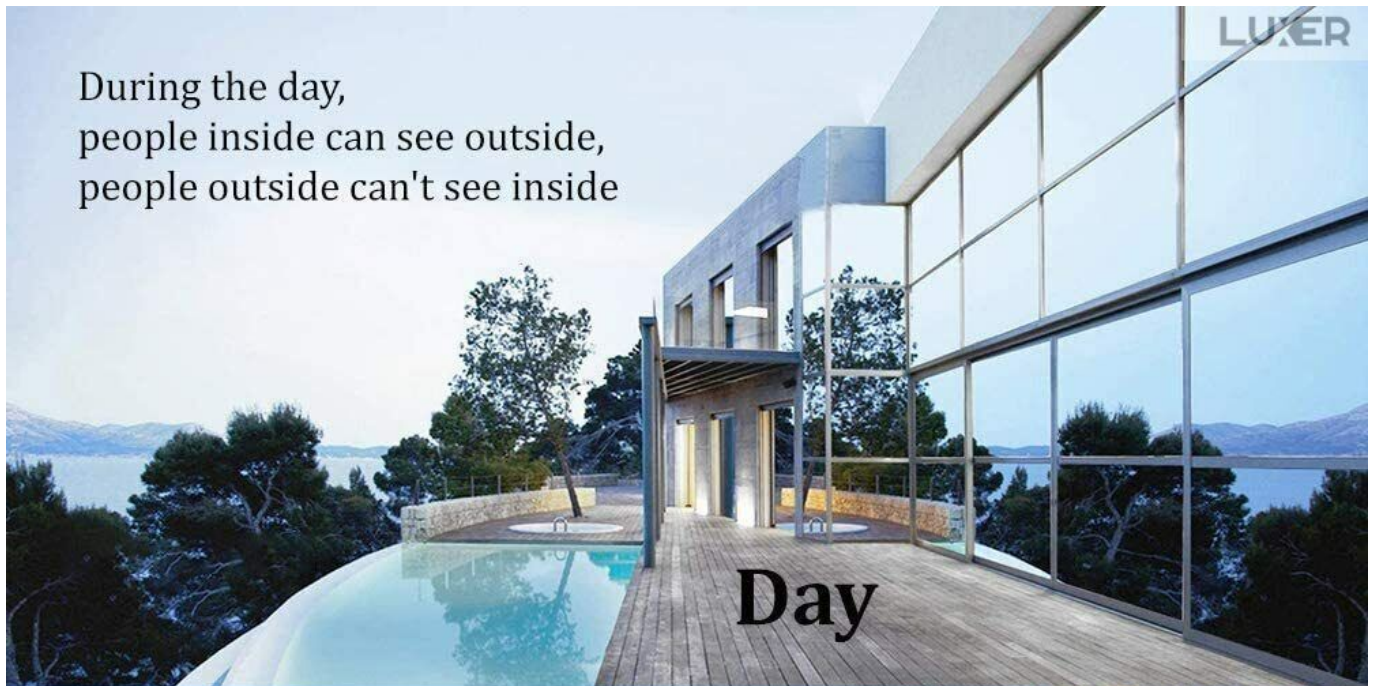
Ideea de baza a plecat de la configuratiile vechi de Rainmeter, un program pentru customizarea Desktop-ului cu diverse widget-uri pentru informatii/shortcut-uri, care practic serveau aceeasi functionalitate: informatii utile intr-un loc foarte convenabil pe care il vezi des.

## Descriere generală

La baza, avem un **microcontroler cu Wifi** ce preia informatii prin **cereri de API** pentru vreme si citeste valori de la hardware clock si senzorul de umiditate. Aceste informatii sunt afisate prin display-ul LCD, informatia afisata poate fi schimbata prin apasarea unui buton.



Display-ul se afla in spatele unei **bucati de sticla** pe care este aplicata **folie cu efect one way mirror** ce permite lumina sa treaca prin spatele ei si pastreaza efectul de oglinda in fata. Fenomenul poate fi observat cel mai bine prin imagine:



## Hardware Design

Am ales o placuta de dezvoltare cu un microcontroller ESP8266 ce are Wifi deja integrat, un display cat mai puternic care sa poata sa produca destula lumina cat sa treaca prin oglinda si componente standard pentru umiditate si ceas.

## Lista componente

Nume	Cantitate	Link
------	-----------	------

Placuta WEMOS D1 R2 WIFI	1	<a href="http://cleste.ro">cleste.ro</a>
2.8" SPI LCD Module with ILI9341 Controller	1	<a href="http://optimusdigital.ro">optimusdigital.ro</a>
Senzor Umiditate DHT22 AM302	1	<a href="http://optimusdigital.ro">optimusdigital.ro</a>
DS3231 Real-time Clock Module	1	<a href="http://optimusdigital.ro">optimusdigital.ro</a>
Buton tactil	1	<a href="http://cleste.ro">cleste.ro</a>
Rezistente	1	<a href="http://cleste.ro">cleste.ro</a>
Breadboard 140 puncte	1	<a href="http://cleste.ro">cleste.ro</a>
Set 140 fire jumper	1	<a href="http://cleste.ro">cleste.ro</a>
Film One Way Mirror	1	<a href="http://emag.ro">emag.ro</a>
Rama fotografie	1	am deja

## Hardware

Am legat si testat toate componentele (cu exceptia ceasului pentru care nu am baterie, dar care pare sa raspunda la functiile de biblioteca otherwise). Singurul lucru ce va fi schimbat in schema finala este display-ul, ce nu va mai fi legat la breadboard direct (n-am avut fire female-male).

Legaturile sunt destul de standard:

- buton si senzor DHT legate la pini digitali ce nu sunt folositi pentru SPI/I2C/whatever deci D3 si D4.
- display legat la MOSI/D7 si SCK/D5 pentru SPI + SCL/D1 si SDA/D2 pentru I2C (trebuie neaparat legat la pini cu aceste protocoale)
- ceas legat la cealalta intrare de I2C (pentru ca ceasul comunica prin I2C)



Oglinda in sine:

## Schematic

Pentru o vizualizare mai simpla a legarii componentelor:




## Software Design

Am abuzat de faptul ca placuta mea este compatibila cu Arduino IDE din deplin. Astfel, folosesc **Arduino IDE** si urmatoarele biblioteci pentru Arduino:


- **ESP8266** pentru Wifi si cereri HTTP
- **WiFi Manager** pentru a servi o pagina web ce ajuta la setup-ul oglinzii
- **Arduino JSON** pentru manipulare date primite de la API-uri
- **AdaFruit ILI9341** pentru display si **AdaFruit GFX** pentru a scrie la display

- **Simple DHT** pentru comunicare cu senzorul de umiditate
- **DS3231** pentru RTC
- **Clock** pentru a pastra evidenta timpului dupa citirea initiala de la RTC (mai simplu decat sa citim de o mie de ori de la ceas)

Arhitectura de baza a software-ului este strans legata de ce trebuie afisat la ecran, astfel avem urmatoare **masina de stari**: 

In starea **loading screen** se realizeaza:

- setupul tuturor obiectelor din diversele biblioteci folosite
- conectarea la Wifi
- citirea unor valori initiale de la senzori

Prima data cand oglinda este folosita, aceasta are nevoie de un mic setup din partea utilizatorului, mai exact trebuie configurata **retea WiFi** la care se va conecta oglinda. Pentru acest lucru oglinda va deschide un access point numit **SmartMirrorAP** la care se pot care utilizatorul trebuie sa se conecteze de pe telefon/calculator, in aceasta retea este servita o pagina web la adresa IP **192.168.4.1** in care utilizatorul poate alege retea dorita: 

Dupa setup-ul initial, oglinda se va conecta automat retea aleasa in urma unui restart.

Starea **weather** prezinta informatii despre vreme de pe **OpenWeatherMap** si un sfat pentru o vestimentatie adecvata. Pentru a determina **locatia** despre care sa preluam informatii este folosita **geolocatia IP-ului**. Evident, aici pot aparea erori daca geolocatia IP-ului este una eronata. Spre exemplu, cand conectez oglinda la datele mele mobile am un IP cu geolocatie de Brasov.

Starea **humidity info** prezinta procentul de umiditate al camerei si determina daca acest procent este adecvat vremii de afara sau nu. In plus, exista si o logica interna de loading deoarece citirile de la senzorul de umiditate sunt lente si inconsistente, fiind posibil sa dureze peste 10 secunde de la pornire pana se face o citire cu succes.

Starea **time and date** prezinta ora si data actuala, acestea se actualizeaza la fiecare minut.

Am abordat urmatoarele subiecte de laborator:

- **intreruperi**
- **SPI** pentru display
- **I2C** pentru RTC
- **GPIO** pentru buton

## Rezultate Obținute

Demo software: <https://youtu.be/XXP2upEUx5Y>

Demo produs final: <https://youtu.be/awK3mPZTxlS>

## Concluzii

A fost foarte distractiv si cred ca o sa folosesc aceasta oglinda in continuare. Ar merge extinsa si mai mult, eventual cu o metoda mai smart de a schimba intre ecrane (ma gandeam la un senzor de gesturi), double buffering la ecran si mai multe ecrane.

## Download

[Cod sursa](#)

## Jurnal

- 12 mai pe la ora 12: mi-au venit toate piesele, optimus digital mi-a trimis display-ul gresit, aia e ne descurcam
- 12 mai pe la ora 23: am legat si am testat tot mai putin RTC-ul, pentru care nu am deocamdata o baterie CR2032 (ma voi duce in piata sa cumpar probabil)
- 13 mai: am scris documentatia hardware, am actualizat putin schematic-ul pentru ca n-am legat lucrurile cum am crezut ca o sa le leg si am facut si un symbol dragut pentru placuta
- 23 mai: am scris si documentat ideea de baza de software, ramane de lipit totul la oglinda si eventual imbunatatit codul pana la PM fair :)
- 24 mai: am atasat totul la oglinda si am facut niste imbunatatiri la partea de software (e.g: access point pentru configurare)

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

## Resurse hardware

- [Wemos D1 R2 datasheet](#)
- [Tutorial conectare display la ESP8266](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/adrian.dumitrache02>



Last update: **2024/05/24 21:09**