

# WFH Employee Monitoring system

**Nume:** Cheresdi Doru

**Grupa:** 331CA

## Introducere

Proiectul este un sistem de monitorizare a angajatilor WFH si de asigurare a ergonomiei la birou. Sistemul asigura faptul ca angajatul se afla la birou in timpul programului, ca isi ia pauzele de la lucru si ca isi ajusteaza modul de lucru pentru si asigura ergonomia la birou. Munca WFH a devenit din ce in ce mai raspandita in ultimii ani si un astfel de sistem ar fi de folos companiilor care opereaza in acest mod, cat si angajatilor.

## Descriere generală

System de monitorizare a angajatilor WFH si asigurare a ergonomiei muncii la birou. Sistemul suporta setarea programului de lucru care este verificat cu un senzor RTC, iar verificarea ca angajatul se afla la birou se face cu un senzor de distanta cu ultrasunete la un anumit interval de timp.

Setarea programului se poate face dupa introducerea parolei de la PC, microcontrollerul va comunica cu acesta prin interfata seriala. Parola are rolul de a securiza sistemul astfel incat angajatul sa nu poata schimba programul de lucru fara aprobarea angajatorului. Daca comunicarea cu PC-ul nu este disponibila, in modul DEV se poate sari peste introducerea parolei apasand butonul 2. Setarea programului de lucru se face prin 2 moduri de configurare:

- prin potentiometru, tensiunea fiind tradusa intr-o ora intre 8-20
- prin senzorul de distanta si pozitionarea mainii la o distanta care se traduce in ora dorita

Alegerea modului de configurare a programului (potentiometru sau senzor de distanta) se va face in modul urmatoar: se contorizeaza timpul folosind modulul RTC si daca in 10 secunde se apasa butonul 2 modul de configurare va fi cel bazat pe potentiometru, altfel se va folosi cel cu senzorul de distanta.

Se asigura ergonomia la birou in modul urmatoar:

- Se verifica ca angajatul nu este prea apropiat de birou. Daca distanta fata de birou a angajatului este prea mica, atunci se trimite un warning la PC si se porneste un buzzer pentru a atentiona angajatul.
- Se verifica lumina din incapare folosind un senzor de lumina si daca se detecteaza intuneric, angajatul este atentionat sa porneasca dark mode pe pc pentru ergonomie.
- Se verifica ca angajatul ia pauze la un anumit interval de timp. Daca trece acel interval si angajatul

nu a luat o pauza(s-a ridicat de la birou), atunci se porneste o alarma care poate fi oprita fie daca se ridica de la birou sau daca apasa pe butonul 2 de snooze. Se contorizeaza numarul de snooze-uri pe care angajatul le-a dat.

- Se verifica ca angajatul a fost la birou pentru cel putin 1/4 din program. Daca lipseste pentru mai mult de 3/4 din program, atunci sistemul intra in modul WORKER\_FAILED.

Toate avertismentele sunt inregistrate si afisate la sfarsitul programului in raportul de lucru. Datele sunt prelucrate si un raport este trimis la PC pentru a fi afisate.

Pentru demo, sistemul va avea modul: DEV. Pentru productie se va implementa modul PROD, diferenta fiind intervalele de timp. In modul DEV aceasta sunt mult mai scurte pentru a se putea testa mai usor proiectul, iar in modul PROD intervalele de timp vor fi cele corespunzatoare unui program de lucru normal.



## Hardware Design

Componentele folosite sunt urmatoarele

- Arduino uno
- senzor de distanta cu ultrasunete
- fotorezistor
- modul RTC
- ecran LCD 2x16
- 3 butoane
- potentiometru
- modul I2C pentru LCD
- rezistente
- buzzer pentru alarma
- fire

Schema electrica: 



## Software Design

Mediu de dezvoltare: Arduino IDE

Programul va functiona folosind o masina de stari cu urmatoarele stari:

- STATE\_RUNNING\_NORMAL - programul ruleaza in modul normal, se esantioneaza date la fiecare 10 min (distanta fata de monitor, daca se afla la pc, senzorul de lumina) si se verifica daca userul intra in modul de configurare
- STATE\_WAIT\_PASS - microcontroller asteapta primirea parolei pe interfata seriala pentru a incepe

- configurarea programului de lucru
- STATE\_CONFIG\_CHOOSE\_1 - se asteapta apasarea butonului pentru a verifica care mod de configurare este ales pentru ora de inceput
  - STATE\_CONFIG\_CHOOSE\_2 - se asteapta apasarea butonului pentru a verifica care mod de configurare este ales pentru ora de sfarsit
  - STATE\_CONFIG\_CHOOSE\_POT - se accepta input pentru o ora de program folosind potentiometrul
  - STATE\_CONFIG\_DISTANCE - se accepta input pentru o ora de program folosind senzorul de distanta
  - STATE\_WARNING\_NOT\_AT\_DESK - utilizatorul nu se afla la birou
  - STATE\_WARNING\_TURN\_ON\_LIGHT\_MODE, STATE\_WARNING\_DISTANCE\_TOO\_SMALL si STATE\_WARNING\_TAKE\_A\_BREAK- userul va fi atentionat ca nu respecta o regula de ergonomie (distanta fata de monitor, lumina si darkmode, daca a luat pauze de 10 min la 60 min de lucru etc) prin pornirea unui buzzer. Userul poate forta microcontroller sa treaca inapoi in starea RUNNING\_NORMAL prin apasarea butonului de snooze
  - STATE\_WORKER\_FAILED - userul a lipsit prea mult timp din program(mai mult de 3/4 din program)

Funcția manageState() se ocupa de gestionarea starilor si tranzitiilor pe baza inputurilor primite, fie de la butoane, RTC, senzor de lumina sau senzorul de distanta. Atunci cand programul de lucru se incheie, microcontroller va trimite date pe interfata seriala despre variabilele esantionate (timpul petrecut la birou, de cate ori a dat snooze, distanta medie fata de birou) Displayul afiseaza mesaje in functie de starea in care se afla, ora sau eroarea aparuta.

Biblioteci:

- Wire pentru a conecta LCD-ul si modulul RTC pe aceeasi magistrala I2C
- LiquidCrystal\_I2C pentru LCD
- RTCLib si DS3231 (pentru RTC).

Codul se afla in repository-ul de github:

<https://github.com/DoruCheresdi/Proiect-PM/blob/main/Proiect-PM.ino>

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

## Rezultate Obținute

Am reusit sa realizez ce mi-am propus, un sistem de monitorizare al angajatului in regim WFH si de asigurare a ergonomiei la birou.

## Concluzii

Proiectul a fost unul interesant de creat mai ales ca are si aplicabilitate practica. Probleme intampinate au fost mai mult la nivel hardware, dar am gasit solutii software pentru a le mitiga.

## Download

<https://github.com/DoruCheresdi/Proiect-PM/blob/main/Proiect-PM.ino>

Arhiva cu codul: [cheresdidoruproiectpm.zip](#)

## Jurnal

- 04.05.2023 → scrierea documentatiei
- 11.05.2023 → verificarea componentelor
- 18.05.2023 → creat montaj si scris schelet cod
- 19.05.2023 → continuat de scris cod
- 21.05.2023 → terminat de scris cod
- 29.05.2023 → polishing software

## Bibliografie/Resurse

[https://adafruit.github.io/RTClib/html/class\\_date\\_time.html](https://adafruit.github.io/RTClib/html/class_date_time.html)

[https://adafruit.github.io/RTClib/html/class\\_time\\_span.html](https://adafruit.github.io/RTClib/html/class_time_span.html)

<https://github.com/NorthernWidget/DS3231>

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

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/gpatru/doru.cheresdi>



Last update: **2023/05/29 17:26**