

# Pulse Sense

## Introducere

Pulse Sense este un sistem de monitorizare a sănătății, conceput pentru a putea măsura și analiza ritmul cardiac și nivelul de saturație al oxigenului (SpO<sub>2</sub>), ținând cont de asemenea inclusiv de temperatura ambientală. Scopul proiectului este de a furniza o soluție accesibilă și precisă pentru monitorizarea continuă a indicatorilor cheie ai sănătății cardiovasculare. Sistemul este conceput să ajute utilizatorii dornici de a înțelege mai bine cum variațiile parametrilor din mediul înconjurător pot influența starea de sănătate.

## Descriere generală

**Arduino ATmega328P** gestionează toate operațiunile și procesează datele primite, fiind fundamental pentru coordonarea activităților întregului sistem, de la colectarea datelor până la afișarea și alertarea utilizatorului.

**Pulsoximetrul MAX30102** este utilizat pentru a măsura ritmul cardiac și nivelul de saturație al oxigenului. Acest senzor joacă un rol fundamental în monitorizarea stării de sănătate. Sistemul poate ajusta alertele în funcție de condițiile de mediu, oferind recomandări mai precise utilizatorului.

**Senzorul de temperatură Adafruit ADT7410** măsoară temperatura ambientală, adăugând un strat suplimentar de date care influențează interpretarea stării de sănătate. Prin aceasta, sistemul poate ajusta alertele în funcție de condițiile de mediu, oferind recomandări mai precise utilizatorului.

**Ecranul OLED** este folosit pentru a prezenta informațiile colectate într-un format accesibil, permițând utilizatorului să urmărească în timp real valorile vitale, cum ar fi ritmul cardiac, nivelul de oxigen și temperatura ambientală.

**Buzzer-ul** este integrat pentru a furniza alerte auditive când valorile monitorizate depășesc anumite praguri setate predefinit.

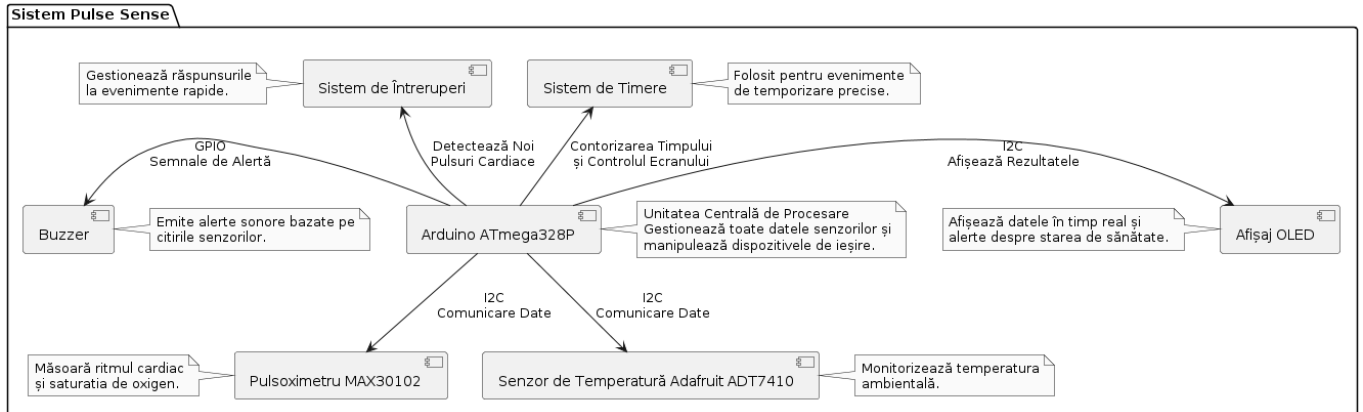
**Sistemul de întreruperi** este configurat pentru a gestiona evenimentele critice, cum ar fi detectarea unui nou puls cardiac. Acesta îi permite plăcuței Arduino să răspundă rapid la schimbări semnificative, asigurând că datele esențiale sunt procesate imediat.

**Sistemul de timere** este folosit pentru a controla evenimentele bazate pe timp, cum ar fi actualizarea periodică a datelor pe afișaj sau controlul duratei de afișare a anumitor ecrane. Timerele ajută la eficientizarea consumului de energie și la îmbunătățirea experienței utilizatorului.

**Interacțiunea** dintre aceste componente este bine orchestrată de microcontrollerul Arduino. Datele de la senzori sunt colectate și transmise prin I2C, procesate de Arduino și apoi afișate pe OLED pentru

a informa utilizatorul. Buzzer-ul este activat de Arduino în situații de alertă, bazându-se pe parametrii predefiniți. Sistemul de întreruperi și cel de timere sprijină operațiunile în timp real și asigură răspunsuri prompte și precise la evenimentele detectate.

Schema bloc a sistemului Pulse Sense este următoarea:

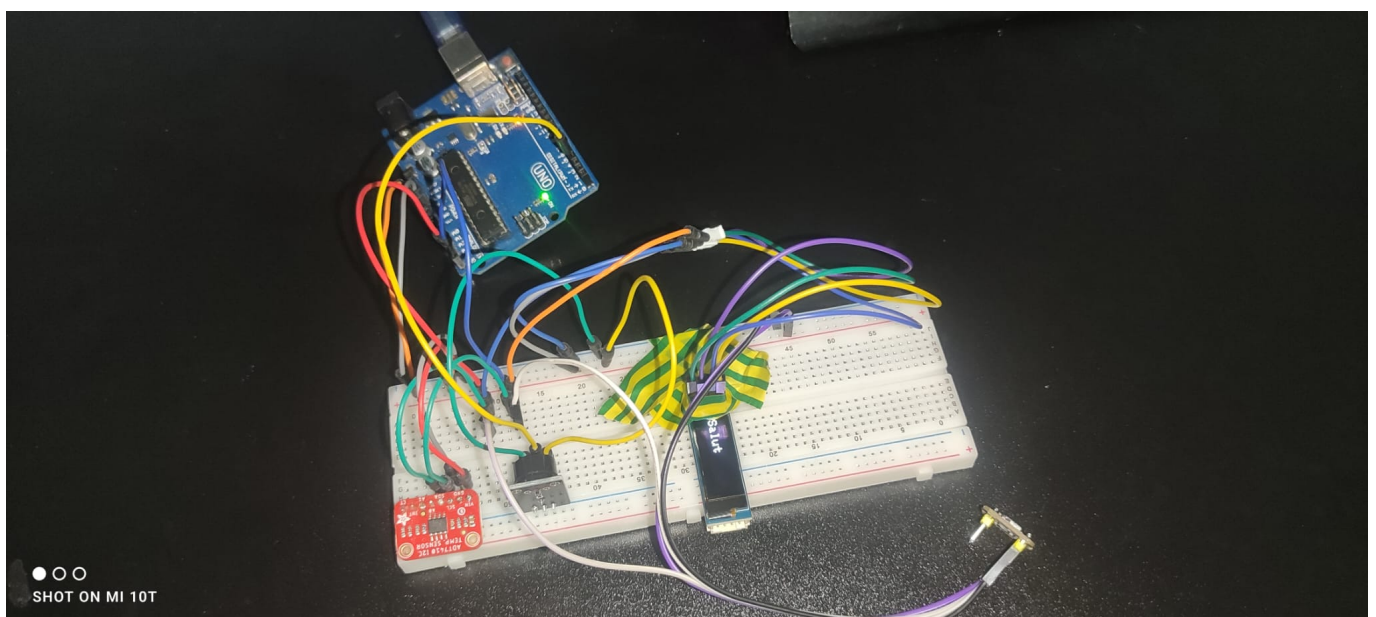


## Hardware Design

Lista de piese:

- Arduino ATmega328P
- Pulsoximetru MAX30102
- Senzor de temperatură Adafruit ADT7410
- Ecran OLED 128x32
- Buzzer
- Breadboard
- Fire conductoare

Schema fizică a circuitului:



## Explicații wiring:

### Arduino ATmega328p

1. este alimentat prin cablul USB

### Pulsoximetru:

```
VCC-> 5V
GND-> GND al plăcuței Arduino
SDA-> A4
SCL-> A5
```

### Buzzer:

```
S(Signal) ->5V
+ -> pin 3
GND-> GND al plăcuței Arduino
```

### Senzorul de temperatură Adafruit ADT7410:

```
Vin->5V
GND->GND
SDA->A4
SCL->A5
```

### Ecran OLED:

```
5V->5V
GND->GND
SDA->A4
SCL->A5
```

### Schema circuitului realizată folosind Fritzing este:



### Schema electrică este:



### Observații și clarificări

Având în vedere faptul ca nu toate componentele sunt disponibile folosirii online, am adaugat in schema circuitului respectiv în schema electrică unele componente similare, folosite doar în cadrul proiectării online.

a)În locul unui ecran OLED 128x32 I2C clasic, am adăugat un ecran OLED din familia Grove.

b)În locul pulsoximetrului MAX30102, am adăugat MAX39192, lăsând pinii care nu sunt necesari nefolosiți.

c)În locul senzorului de temperatură ADT7410 de la Adafruit, am folosit un TMP102.

d)În locul unui buzzer pasiv cu 3 fire, am folosit un buzzer cu 2 fire. Al treilea, care nu putea fi folosit, legându-l totuși la pinul 3 al plăcuței Arduino.

### **Alte observații**

a) Rolul fiecărei componente în parte a fost prezentat cu ajutorul unei diagrame UML în cadrul secțiunii “Descriere generală”.

b) Pinul SDA (Serial Data) este conectat la pinul analogic A4 al plăcuței Arduino. SDA este utilizat pentru comunicarea I2C, un protocol de comunicare serială care permite transferul de date între Arduino și elementele de circuit.

c) Pinul SCL (Serial Clock) este conectat la pinul analogic A5 al plăcuței Arduino. SCL este utilizat pentru a sincroniza transferul de date pe magistrala I2C.

## **Software Design**

### **1. Biblioteci folosite**



Am folosit:

-Adafruit\_GFX pentru ecranul OLED ce functioneaza pe I2C.

-Adafruit\_SSD1306 tot pentru ecranul OLED in sa pentru SSD1306.

-Wire pentru comunicatiile I2C.

-MAX30105 pentru pulsoximetru.

-heartRate pentru a aplica un algoritm de calculare a pulsului.

-Adafruit\_ADT7410 pentru senzorul de temperatura.

### **2. Funcționalități din laboratoare pe care le-am folosit**

a) Protocolul I2C



I2C a fost folosit atât pentru pulsoximetru, cât și pentru ecranul OLED și senzorul de temperatură

**b) Timere**



În acest cod, funcția `millis()` este utilizată pentru a obține timpul curent în milisecunde de la pornirea Arduino-ului. Aceasta este folosită pentru a realiza o temporizare simplă pentru a afișa temperatura timp de 10 secunde de la începutul execuției programului, apoi interfața pentru măsurarea pulsului va apărea.

### c)GPIO



GPIO este folosit aici pentru buzzer, activează buzzer pe pinul 3 și dezactivează buzzer pe pinul 3

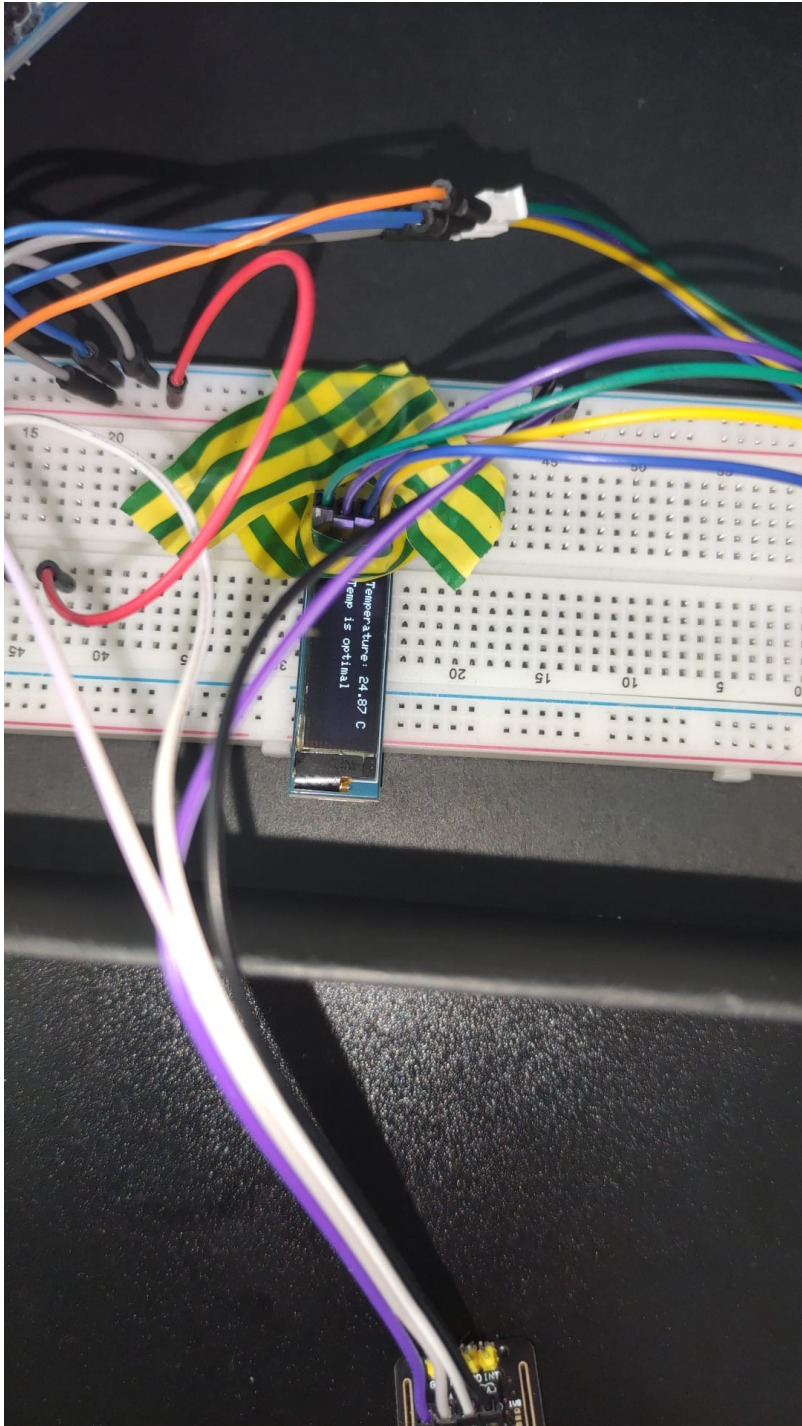
### 3.Codul însoțit de explicații

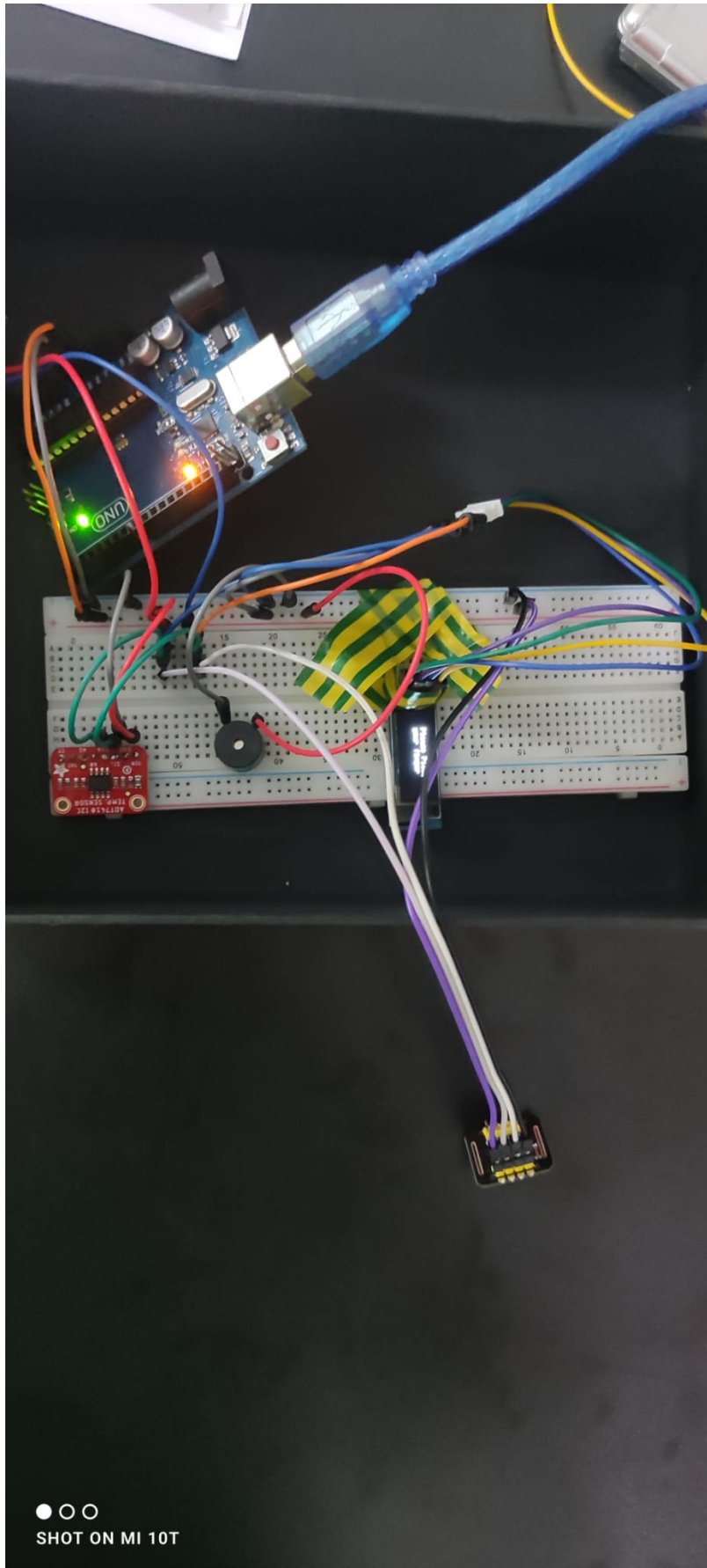


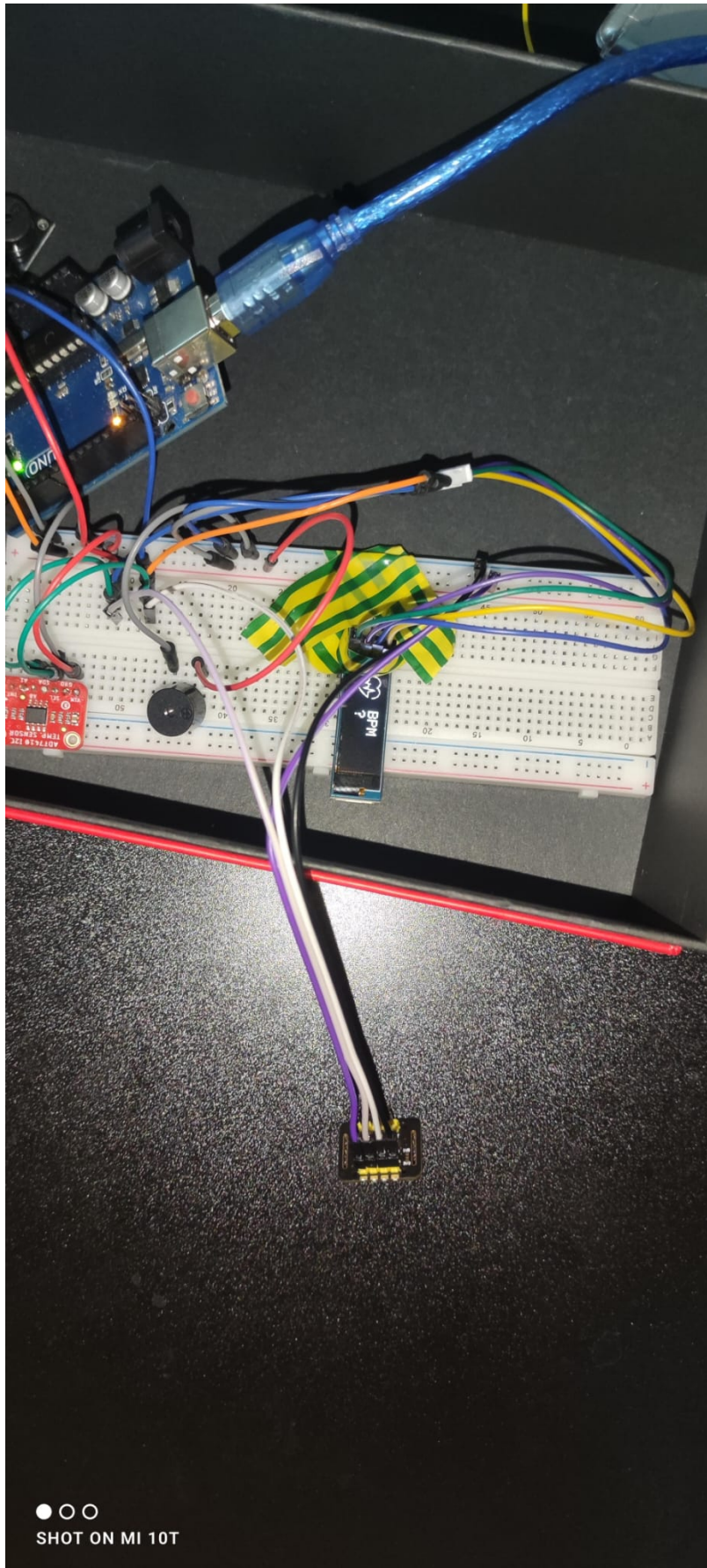
### 4.Calibrare și optimizare

Am calibrat tabloul de pixeli pentru a scoate ca output imaginea dorită, iar la nivelul senzorului de temperatură am calibrat timpul ca să fie vizibil întrucât acel timp de 10 secunde este de fapt din momentul în care Arduino pornește(pierzându-se câteva secunde în timpul pornirii și afișării siglei ecranului OLED).

## Rezultate Obținute







Observație: azi voi înlocui pulsoximetrul cu altul nou , întrucât nu mai măsoară pulsul corect.



## Concluzii

Consider că acest proiect m-a ajutat să aprofundez noțiunile studiate în cadrul laboratoarelor și cursurilor, aducând o nuanță mult mai practică asupra noțiunilor prezentate.

## Download

Arhiva proiectului este: [333aa\\_popa\\_adrian\\_proiect\\_pm\\_1.rar](#)

## Bibliografie/Resurse

[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

<https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30102.pdf>

<https://www.analog.com/media/en/technical-documentation/data-sheets/ADT7410.pdf>

<https://www.instructables.com/Interfacing-Buzzer-to-Arduino/>

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/adrian\\_gabriel.popa](http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/adrian_gabriel.popa)



Last update: **2024/05/26 12:59**