

# Baby Monitor



## Dispozitiv de monitorizare pentru bebelusi:

- atentioneaza parintii atunci cand nu sunt in camera si plange bebelusul
- functioneaza ca o lampa de veghe pe parcursul noptii
- reda o animatie cu scopul de a capta atentia copilului si a-l face sa se opreasca din plans
- vine in ajutorul familiilor cu nou nascuti pentru a usura monitorizarea in timpul noptii
- dispozitivul este un sprijin pentru parinti, care pot dormi mai linistiti stiind ca vor fi anuntati de acesta in cazul in care copilul se trezeste

## Descriere generală

Folosind un senzor de zgomot, dispozitivul sesizeaza plansetele copilului si porneste un avertisment sonor pentru parinti. Totodata, are incorporat un senzor de lumina care aprinde o matrice de leduri la lasarea intunericului. Pe un ecran lcd va rula o animatie ce are ca scop captarea atentiei bebelusului.

Schema bloc :



## Hardware Design

### Componente :

- ecran LCD;
- matrice de led-uri;
- placuta Arduino ATmega 2500;
- modul buzzer activ;
- modul cu fotorezistor;
- modul senzor de zgomot

## Conectarea componentelor :

- LCD-ul este conectat la placuta Arduino folosind pini digitali din partea dreapta
- matricea de LED-uri foloseste 4 pini PWM si se alimenteaza la pinul 5V
- senzorii utilizeaza pini analogici
- buzzerele utilizeaza pini digitali

## Schematica electrica:

Realizata folosind **Autodesk Fusion**



[Pentru o intelegere mai buna a modului in care sunt conectate componentele, atasez un close-up al placutei:](#)



## Software Design

### Rationamentul implementarii:

Mediul de dezvoltare folosit a fost **Visual Studio Code**, folosind extensia **PlatformIO**. Impreuna cu etapele implementarii sunt incluse fragmente de cod pentru o intelegere mai usoara a rationamentului.

Am inceput prin a construi **animatia redata pe lcd** : dupa ce am ales imaginile, le-am convertit in siruri de octeti folosind [image2cpp](#). Acestea sunt stocate si redade din memoria flash a microcontroller-ului, folosind functii specifice bibliotecii TFT\_HX8357. **Urmatorul fragment de cod este inclus in metoda loop():**

```
if (millis() - lastImageChangeTime >= 4000) {
  // Schimba imaginea la urmatoarea in slideshow
  switch (currentImage) {
    case 1:
      tft.drawBitmap(0, 0, image1, IMAGE_WIDTH, IMAGE_HEIGHT, TFT_CYAN);
      currentImage = 2;
      delay(2000);
      tft.fillScreen(TFT_DARKCYAN);
      break;
    case 2:
      tft.drawBitmap(0, 0, image2, IMAGE_WIDTH, IMAGE_HEIGHT, TFT_CYAN);
      currentImage = 1; // Resetăm contorul pentru a începe slideshow-ul
  }
}
```

```

        delay(2000);
        tft.fillScreen(TFT_DARKCYAN);
        break;
    }
    lastImageChangeTime = millis(); // Actualizăm momentul ultimei schimbări
    de imagine
}

```

Mai apoi, am configurat lampa de veghe. Am ales sa fac aceasta caracteristica putin mai interactiva, asa ca, atunci cand fotorezistorul nu mai sesizeaza lumina, pe matricea de LED-uri se reda o animatie cu o inima care bate. Pentru asta folosesc functiile :

- displayHeartFrame(const byte\* frame) - aprinde doar ledurile care construiesc un frame
- animatedHearts() - realizeaza schimbarea de cadre consecutive

```

// un exemplu de frame
const byte heart3[8] = {
    B00000000,
    B01100110,
    B11111111,
    B11111111,
    B01111110,
    B00111100,
    B00011000,
    B00000000
};

void displayHeartFrame(const byte* frame) {
    for (int row = 0; row < 8; row++) {
        lc.setRow(0, row, frame[row]);
    }
}

void animateHearts() {
    for (int i = 0; i < 5; i++) {
        displayHeartFrame(heartFrames[i]);
        delay(100); // Ajustăm viteza animației
    }
}

```

In metoda loop() pornim LED-urile in functie de valorile semnalate de fotorezistor :

```

// Dacă nivelul de lumină este sub pragul stabilit, activăm matricea de
LED-uri
if (lightLevel < LIGHT_THRESHOLD) {
    animateHearts();
} else {
    lc.clearDisplay(0); // Stingem toate LED-urile
}

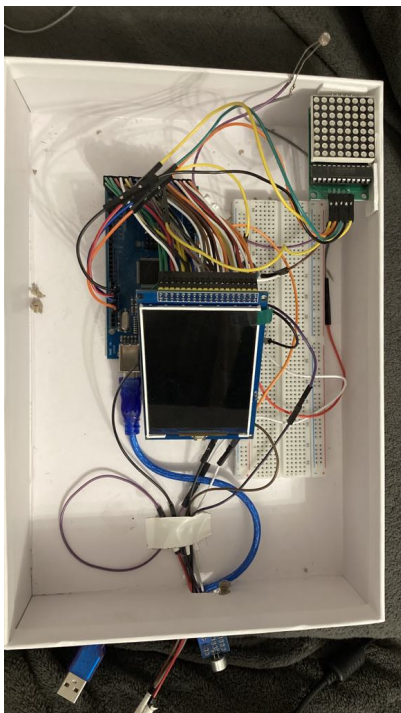
```

Ultima si cea mai greu de testat etapa o reprezinta  **alarma pentru parinti** . Cea mai solicitanta parte

a fost configurarea buzzer-ului astfel incat sa produca un sunet controlat atunci cand microfonul sesizeaza zgomote ce trec de pragul stabilit. Astfel, **in metoda loop()** configuram buzzer-ul sa cante o melodie simpla ce formeaza un semnal scurt de avertizare pentru parinti ca bebelusul a inceput sa planga:

```
// Dacă nivelul de zgomot este peste pragul stabilit, activăm buzzer-ul
if (soundLevel > NOISE_THRESHOLD) {
  buzzerEndTime = millis() + 1 * 1000; // Activăm buzzer-ul
  // Redăm o melodie simplă
  for (int thisNote = 0; thisNote < 8; thisNote++) {
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(BUZZER_PIN, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(BUZZER_PIN); // Oprește tonul curent
  }
}
```

## Rezultate Obținute



Pentru un aspect mai organizat si potrivit pentru copii, am ales sa introduc piesele intr-o cutie. Inainte de a o inchide complet, am fotografiat asezarea componentelor in interior; Atasez si o fotografie cu proiectul atunci cand fotorezistorul nu sesizeaza lumina ambientala si aprinde matricea de LED-uri.

Pentru a surprinde intreaga functionalitate am incarcat un video in care este surprinsa reactia dispozitivului la zgomot si la lumina [https://youtu.be/NcxkkqK\\_NjU](https://youtu.be/NcxkkqK_NjU)

## Concluzii

Cred ca am invatat multe pe parcursul realizarii acestui proiect si ca este mai satisfacator sa lucrezi la un cod si sa poti vedea o materializare a functionalitatilor sau disfunctionalitatilor sale.

## Download

Aici puteti regasi arhiva ce contine codul pe care l-am creat:

[baby\\_monitor.zip](#)

## Bibliografie/Resurse

Pentru realizarea acestui proiect am folosit urmatoarele biblioteci :

- [LedControl.h](#)
- [TFT\\_HX8357.h](#)

In structurarea si implementarea acestuia, am urmarit indicatiile din urmatoarele laboratoare :

- <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab5-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab4-2023-2024>

Acestea m-au ajutat sa folosesc LCD-ul cu protocolul SPI, sa calibrez senzorii si matricea de LED-uri si sa ma folosesc de buzzer-ul activ.

## Github Repository:

[https://github.com/catalinabogdan/baby\\_monitor/tree/main](https://github.com/catalinabogdan/baby_monitor/tree/main)



[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/fgul/elena.bogdan2912>



Last update: **2024/05/26 22:15**