



Universitatea Tehnică „Gheorghe Asachi” din Iași
Facultatea de Automatică și Calculatoare

LUCRARE DE LICENȚĂ

Domeniul
Calculatoare și tehnologia informației

Specializarea
Calculatoare

ALARMĂ PENTRU AUTOMOBILE

Conducător științific

dr. ing. HULEA Mircea

Absolvent

Maftei Eduard-Ionuț

Sesiunea:
Iulie 2022

Iași, 2022

Cuprins

Introducere.....	1
Capitolul 1. Generalități despre sisteme de alarmă.....	3
1.1. Sisteme.....	3
1.2. Alarmer.....	4
1.3. Microcontroler.....	5
1.3.1. Funcționarea unui microcontroler.....	5
1.3.2. Elementele unui microcontroler.....	6
1.3.3. Caracteristicile microcontrolerului.....	7
1.3.4. Aplicații pentru microcontrolerele.....	7
1.3.5. Microcontrolere vs microprocesoare.....	8
1.4. Protocolul de comunicare I2C.....	8
1.4.1. Funcționarea protocolului de comunicare I2C.....	8
1.4.2. Condițiile de START/STOP.....	9
1.4.3. Condiția de pornire repetată.....	9
1.4.4. Bit de citire/scriere.....	9
1.4.5. Adresare.....	9
1.4.6. Formatul pachetului I2C.....	9
1.4.7. Caracteristici ale protocolului de comunicare I2C.....	9
1.4.8. Avantaje.....	10
1.4.9. Limitări.....	10
1.5. Breadboard.....	10
1.5.1. Tipuri de breadboarduri.....	11
1.5.2. Rânduri, coloane și etichete.....	12
1.5.3. Conectarea găurilor.....	13
1.5.4. Probleme ce pot apărea.....	13
1.5.4.1. Conectarea greșită la rânduri.....	13
1.5.4.2. Încurcarea alimentării cu masa.....	14
1.5.4.3. Conectarea cablurilor și firelor necorespunzător.....	14
1.5.4.4. Punerea componentelor invers.....	15
Capitolul 2. Structura hardware a proiectului.....	17
2.1. Sisteme folosite.....	17
2.2. Componente hardware.....	17
2.3. Funcționalitatea sistemelor.....	17
2.3.1. Sistemul de detecție al mișcării.....	17
2.3.2. Sistemul de detecție al sunetului.....	17
2.3.3. Sistemul de detecție a greutatei.....	18
2.3.4. Sistemul de detecție al accelerațiilor.....	18
2.3.5. Sistemul de avertizare.....	18
2.4. Programarea sistemelor.....	18
2.4.1. Prezentare generală Arduino Uno.....	18
2.4.2. Alimentarea plăcuței.....	19

2.4.3. Comunicația plăcuței.....	20
2.4.4. Configurația pinilor.....	20
2.5. Senzori utilizați.....	22
2.5.1. Senzor ultrasonic.....	22
2.5.2. Modul accelerometru.....	23
2.5.3. Modul de înregistrare a sunetului.....	24
2.5.4. Modul buzzer.....	25
Capitolul 3. Funcționalitatea sistemelor.....	27
3.1. Sistemul de detecție a mișcării.....	27
3.2. Sistemul de detecție a sunetului.....	28
3.3. Sistemul de detecție al accelerației.....	29
3.4. Sistemul de avertizare.....	31
Concluzii.....	32
Bibliografie.....	I
Anexe.....	1

Alarmă pentru automobile

Eduard-Ionuț Maftai

Rezumat

Lucrarea de diplomă reprezintă o sinteză a tuturor cunoștințelor atât teoretice, cât și practice, pe care le-am acumulat pe durata celor patru ani de facultate, dar și prin studiul individual extracurricular. În această lucrare am reușit să demonstrez și să implementez cunoștințe teoretice din domeniul programării microcontrolerelor și crearea unui circuit funcțional care să funcționeze ca o alarmă pentru un automobil.

Lucrarea de licență abordează o temă de actualitate, care este foarte utilă în zilele noastre, având în vedere dezvoltarea de sisteme cât mai complexe care să ne alerteze atunci când automobilul personal a fost implicat într-un accident sau într-o tentativă de furt.

Majoritatea vehiculelor de astăzi sunt fabricate cu alarme auto încorporate, acestea fiind îmbunătățite cu noi funcționalități care să fie în favoarea clientului. Este important ca alarmele să fie îmbunătățite, având în vedere faptul că multe astfel de sisteme de securitate mai vechi nu mai sunt la fel de actuale și pot fi ușor ocolite de către hoții de mașini mai experimentați. Astfel, prin instalarea unui dispozitiv în autovehicul care să alerteze proprietarul când are loc un incident, furtul sau deteriorarea bunului este descurajată.

O să prezint în cadrul acestei lucrări atât modul de proiectare, modul de implementare a resurselor hardware folosite în crearea unui astfel de sistem de alarmă pentru mașină, cât și cum acesta poate fi ulterior îmbunătățit, în vederea înțelegerii funcționalității și importanței acestuia. Voi discuta și despre detaliile tehnice și despre sursele mele de documentație în vederea finalizării lucrării.

Documentația proiectului conține rezultatele experimentale care susțin fundamentele teoretice prezentate în cadrul acesteia. Sper că la final să reușesc să exemplific un model de circuit care poate fi implementat într-un mod cât mai ușor de către oricine și că o să fac înțelesă utilitatea practică a acestuia în societatea contemporană și cum poate fi utilizată în interesul propriu în vederea ușurării vieții cotidiene.

Introducere

Din ce în ce mai multe autoturisme sunt construite cu măsuri de securitate suplimentare pentru a oferi o soluție de protecție împotriva echipamentelor tot mai avansate folosite de către persoane rău intenționate pentru a sparge un autoturism. Pentru a evita orice neplăcere care poate fi cauzată din exterior asupra unui vehicul, sistemele de securitate sunt îmbunătățite și din ce în ce mai performante. Acestea trebuie să fie cât mai folositoare în zilele noastre, având în vedere că suntem obișnuiți ca zilnic să auzim alarme false de la mașini și descurajați din acest punct de vedere să ne asigurăm de fiecare dată când o astfel de alarmă sună, dacă mașina noastră a fost într-adevăr afectată sau nu.

Încă de la început, de la inventarea automobilului, acestea au fost considerate bunuri de preț. Pe măsură ce acestea au devenit din ce în ce mai accesibile pentru populație, a crescut nevoia de a le păstra în siguranță și cu o rezistență ridicată împotriva furturilor.

Alarmele auto au parcurs un drum lung de la început și până astăzi. La început, George Evans și E. B. Birkenbeuel au creat un dispozitiv care a fost special conceput pentru a trimite un semnal în cazul în care cineva neautorizat încerca să miște o mașină. Cu toate acestea, dispozitivul nu a fost la fel de complex ca alarmele pe care le avem astăzi. Era în schimb necesar ca pentru a porni o mașină să fie introdus un cod din trei cifre. Dacă acest cod era introdus greșit, electricitatea era automat deviată către claxon. Prima alarmă de mașină produs în serie a fost creată de către Victor Helman în 1954. Dispozitivul era plasat în torpedoul mașinii și avea fire conectate la un comutator către capotă, uși și portbagaj. Dacă orice parte a mașinii ar fi fost deschisă fără dezactivarea comutatorului principal, situat undeva în exteriorul mașinii, ar fi fost declanșată alarma.

Lucrarea de licență reprezintă un proiect al cărui scop inițial a fost aprofundarea informațiilor despre microcontrolerul Arduino Uno deoarece nu am avut ocazia să lucrez cu acesta în proiectele din timpul facultății. În urma documentării efectuate de către mine, m-am gândit să implementez un sistem de alarmă pentru mașină pentru a înțelege mai bine cum funcționează acesta, având în vedere importanța lor.

În vederea proiectării unui astfel de sistem m-am folosit de mai mulți senzori pe care i-am implementat cu ajutorul fișelor tehnice, dar și cu ajutorul cunoștințelor acumulate în cadrul celor patru ani de facultate. Această documentație nu include doar descriere conceptului de sistem de alarmă, ci și motivația, sursele de documentație și direcții viitoare de dezvoltare a aplicației. În plus, în lucrare se regăsesc toți pașii pe care i-am urmat în detaliu în implementarea alarmei.

Unul dintre punctele pe care doresc să le ating este înțelegerea importanței folosirii unui sistem de alarmă, utilitatea practică în prezent și cum funcționează aceasta. Doresc să creez o lucrare care poate fi văzută ca o viitoare documentație tehnică pentru crearea sistemelor de alarmă pentru mașină care să reprezinte un punct de început pentru viitori dezvoltatori.

Din punct de vedere structural, am încercat pe cât posibil să folosesc principiul modularității în cadrul acestei documentații pentru a o face potrivită în vederea unei dezvoltări ulterioare. Documentația lucrării de diplomă este structurată pe trei capitole, excluzând introducerea și concluziile însoțite de direcțiile viitoare de dezvoltare. În primul capitol am punctat câteva generalități legate de un sistem de alarmă, lucruri care stau la baza înțelegerii unui astfel de concept. În cel de-al doilea capitol am vorbit mai multe despre structura hardware a proiectului și anume, componentele hardware folosite în crearea sistemului fizic. Am detaliat fiecare componentă în parte pentru a ne familiariza cu aceasta și a înțelege modul ei de funcționare în circuitul creat. Ultimul capitol va fi despre implementarea în sine a proiectului și

va conține în mod detaliat pașii care au dus la finalizarea proiectului de diplomă. La finalul acestei documentații se va regăsi secțiunea de concluzii și direcții viitoare de dezvoltare a circuitului pentru a fi cât mai util și în conformitate cu cerințele actuale.

Capitolul 1. Generalități despre sisteme de alarmă

De-a lungul timpului s-au dezvoltat, din necesitate, diferite sisteme care să ne atragă atenția atunci când se întâmplă un pericol. Din dorința ca bunurile noastre să fie păstrate în siguranță în locurile publice, au apărut treptat sisteme din ce în ce mai optimizate de alarmă, în special pentru case și autoturisme. Am ales să cercetez mai amănunțit cum funcționează un astfel de sistem de alarmă pentru mașini și voi prezenta în continuare, în amănunt, principalele concepte în vederea creării unui astfel de sistem, în vederea familiarizării cu termenii utilizați în următoarele capitole.

1.1. Sisteme

În acest subcapitol voi prezenta noțiunile găsite despre sisteme în conformitate cu [1].

Un sistem poate fi gândit ca un set de concepte sau părți care trebuie să lucreze împreună pentru a efectua un anumit proces. Bela Banathy a definit un sistem creat de om ca o entitate compusă din părți care este proiectată și construită de oameni într-un întreg organizat pentru atingerea unui scop specific.

O organizație este un sistem sau o colecție de sisteme. Fiecare loc de muncă dintr-o organizație este utilizat de un sistem pentru a produce sau a susține un produs sau serviciu. Produsul sau serviciul este mijlocul prin care o organizație supraviețuiește sau se întreține. O organizație mare poate avea mai multe sisteme care, în general, sunt împărțite în departamente sau grupuri, în timp ce o companie mai mică poate avea un singur sistem.

Există patru intrări necesare în fiecare sistem pentru a produce un produs sau serviciu:

- Omul: Muncitorii din grup sunt legați printr-o activitate comună;
- Materialul: Produsele brute care intră într-un sistem;
- Tehnologia: Tehnica pentru atingerea unui scop sau scop practic;
- Timpul: Perioada măsurată în timpul căreia începe și se termină o acțiune sau un proces.

Toate sistemele au trei funcții de bază:

- Intrare: Trebuie să intre ceva în sistem, altfel este o sferă misterioasă în care produsele sau serviciile radiază mistic din ea. După cum s-a menționat mai sus, intrările de bază ale unui sistem sunt materialele, oamenii, tehnologia și timpul.
- Proces: Un anumit tip de lucru trebuie realizat în sistem. Această lucrare este tehnologia efectuată care modifică intrarea materialului în ieșirea sistemului.
- Rezultat: Un serviciu sau produs dorit trebuie să fie produs. Dacă nu există ieșire, atunci este o gaură neagră în care intră lucrurile, dar nu iese nimic.

Un proces este o serie planificată de acțiuni în cadrul unui sistem care avansează materiale sau proceduri de la o etapă la alta. Un sistem are de obicei mai multe procese în el. Un proces este ca un mini-sistem prin faptul că există intrări și funcții. Principala diferență este că un sistem produce un produs sau serviciu complet, în timp ce o funcție produce sau susține o parte a produsului sau serviciului. Un exemplu de proces este: muncitorii de asamblare a plăcilor de circuite (oameni) care lipesc piese electronice (materiale) pe plăci de circuite lucrând pe o linie de producție specializată (tehnologie) și completând un număr stabilit de plăci de circuite într-un anumit termen limită (timp). Produsul final (output) este apoi utilizat de alți membri ai echipei de producție în asamblarea unui computer.

Trebuie observat că în exemplul de mai sus avem un furnizor și un client. Clientul poate fi intern sau extern organizației.

1.2. Alarmer

În acest subcapitol voi prezenta noțiunile găsite în conformitate cu [2] și [3].

Un sistem de alarmă este un ansamblu de echipamente electronice interconectate, al cărui scop este de a detecta și semnaliza prezența persoanelor sau evenimentelor nedorite.

În general, există două tipuri de circuite de alarmă. Acestea sunt circuitul de alarmă antiefracție și circuitul de alarmă de incendiu. Deși termenii corecți sunt „nesupravegheat” și „supravegheat”. Circuitele nesupravegheate formează mai multe zone de efracție. De asemenea, rotațiile supravegheate sunt practice în circuitele analogice de incendiu. De aceea folosim cuvintele efracție și incendiu. Există doar două stări într-un circuit de alarmă de efracție: deschis și rezistor de capăt de linie.

Sistemele de alarmă au evoluat din forme rudimentare în cele pe care le cunoaștem astăzi. Accesul la ele de către publicul larg a început după al Doilea Război Mondial, când armata a decis să ofere tehnologia tuturor. Astfel, au luat naștere companii noi producătoare de sisteme de alarmă, fiecare oferind funcții exotice la vremea respectivă. Sistemele de alarmă erau greu de utilizat, nu foarte eficiente ca în ziua de azi și totuși, oamenii aveau încredere în ele.

Chiar dacă în primă fază doar companiile mari aveau acces la ele și își permiteau să împânzească sediile cu sirene și senzori, ceva mai târziu au putut ajunge și la consumatorii finali, cetățeni simpli care doreau și ei să se simtă în siguranță.

Astăzi se găsesc o varietate foarte largă de astfel de sisteme de alarmă atât în magazinele fizice dar și în magazinele online.

Un sistem de alarmă este compus din foarte multe elemente. Sistemul de alarmă trebuie să conțină minim 4 elemente pentru a putea funcționa. În primă fază discutăm despre centrala de alarmă. Aceasta este considerată creierul întregului sistem și în ea se ascunde toată partea de procesare. Este cea care calculează fiecare semnal de la senzori și transmite informația mai departe la sirena.

Centrala de alarmă se montează într-o zonă sigură, protejată și cât mai ascunsă de ochii lumii. Poate să ofere conexiune cablată sau prin wireless, se poate îmbunătăți folosind module speciale și la ea se vor conecta toate celelalte componente. Un exemplu de sistem de alarmă este cel din figura Figura 1.1



Figura 1.1.Exemplu de sistem de alarmă

Senzorii fac și ei parte dintr-un sistem de alarmă, fiind responsabili de detecția

problemelor. Aceștia sunt de mai multe tipuri și forme, iar printre cei mai populari sunt senzorii PIR. Aceștia detectează mișcarea, unii sunt imuni la animale până într-o anumită greutate și se montează de obicei în perete sau în tavan.

Sirena este și ea parte dintr-un sistem de alarmă și are rolul de a emite avertizări acustice sau vizuale în cazul în care un senzor a detectat o problemă. După ce informația a fost analizată de centrala de alarmă, ea trimite un semnal către sirena și îi spune ce avertizare sonoră și acustică trebuie să redea în funcție de senzorul declanșat. Aceasta poate să se afle în interior sau exterior, să funcționeze pe baterii sau cu alimentare directă, poate avea protecții la căldură excesivă sau îngheț și să redea tonalități diferite cu jocuri de lumini în funcție de senzor.

1.3. Microcontroller

Informațiile următoare sunt extrase din sursele [4] și [5].

Microcontrollerul este un circuit integrat, compact, conceput pentru a guverna o anumită operațiune într-un sistem încorporat. Un microcontroller tipic include un procesor, memorie și periferice de intrare/ieșire (I/O) pe un singur cip. Microcontrollerul a fost introdus în industria electronică cu scopul de a ne ușura sarcinile care vin chiar și cu o conexiune la distanță cu automatizare în orice fel.

Denumite uneori controller încorporat sau unitate de microcontroller (MCU), microcontrolerele se găsesc în vehicule, roboți, mașini de birou, dispozitive medicale, automate și aparate electrocasnice, printre alte dispozitive. Ele sunt, în esență, simple computere personale (PC-uri) în miniatură, concepute pentru a controla caracteristicile mici ale unei componente mai mari, fără un sistem de operare front-end (OS) complex. În Figura 1.2 se poate vedea un exemplu de microcontroller.

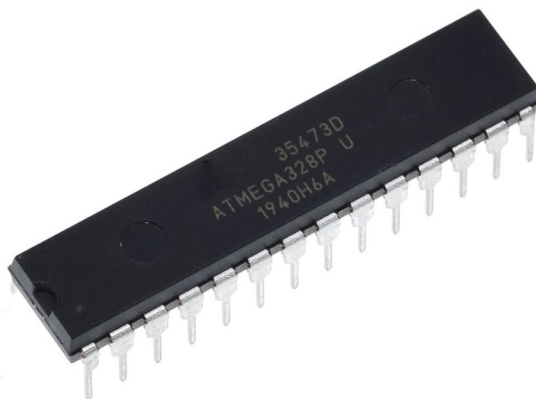


Figura 1.2. Microcontroller

1.3.1. Funcționarea unui microcontroller

Un microcontroller este încorporat în interiorul unui sistem pentru a controla o funcție singulară a unui dispozitiv. Face acest lucru interpretând datele pe care le primește de la perifericele sale folosind procesorul central.

Informațiile temporare pe care le primește microcontrolerul sunt stocate în memoria sa de date, unde procesorul o accesează și folosește instrucțiunile stocate în memoria programului său pentru a descifra și aplica datele primite, apoi își folosește perifericele I/O pentru a comunica și a pune în aplicare acțiunea corespunzătoare.

Microcontrolerele sunt utilizate într-o gamă largă de sisteme și dispozitive. Dispozitivele folosesc adesea mai multe microcontrolere care lucrează împreună în cadrul dispozitivului pentru a-și gestiona sarcinile respective.

De exemplu, o mașină poate avea multe microcontrolere care controlează diferite sisteme individuale, cum ar fi sistemul de frânare, controlul tracțiunii, injecția de combustibil sau controlul suspensiei. Toate microcontrolerele comunică între ele pentru a informa acțiunile corecte. Unele ar putea comunica cu un computer central mai complex din mașină, iar altele ar putea comunica doar cu alte microcontrolere. Ele trimit și primesc date folosind perifericele lor I/O și procesează acele date pentru a-și îndeplini sarcinile desemnate.

1.3.2. Elementele unui microcontroler

Elementele de bază ale unui microcontroler sunt:

- Procesorul (CPU): un procesor poate fi considerat creierul dispozitivului. Acesta procesează și răspunde la diverse instrucțiuni care direcționează funcția microcontrolerului. Aceasta implică efectuarea de operații de bază aritmetice, logice și I/O. De asemenea, efectuează operațiuni de transfer de date, care comunică comenzi altor componente din sistemul încorporat mai mare.
- Memoria: Memoria unui microcontroler este folosită pentru a stoca datele pe care procesorul le primește și le folosește pentru a răspunde la instrucțiunile pe care a fost programat să le execute. Un microcontroler are două tipuri principale de memorie: memoria programului, care stochează informații pe termen lung despre instrucțiunile pe care le execută procesorul, care este nevolatilă, ceea ce înseamnă că păstrează informații în timp fără a avea nevoie de o sursă de alimentare și memoria de date, care este necesară pentru stocarea temporară a datelor în timpul executării instrucțiunilor și este volatilă, ceea ce înseamnă că datele pe care le deține sunt temporare și sunt menținute numai dacă dispozitivul este conectat la o sursă de alimentare.
- Periferice I/O: Dispozitivele de intrare și ieșire sunt interfața procesorului cu lumea exterioară. Porturile de intrare primesc informații și le trimit procesorului sub formă de date binare. Procesorul primește acele date și trimite instrucțiunile necesare către dispozitivele de ieșire care execută sarcini externe microcontrolerului.

În timp ce procesorul, memoria și perifericele I/O sunt elementele definitorii ale microprocesorului, există și alte elemente care sunt frecvent incluse. Termenul de periferice I/O în sine se referă pur și simplu la componentele de suport care interacționează cu memoria și procesorul. Există multe componente de sprijin care pot fi clasificate ca periferice. A avea o anumită manifestare a unui periferic I/O este elementar pentru un microprocesor, deoarece acestea sunt mecanismul prin care este aplicat procesorul.

Alte elemente de sprijin ale unui microcontroler includ:

- Convertor analog-digital (ADC): Un ADC este un circuit care convertește semnalele analogice în semnale digitale. Acesta permite procesorului din centrul microcontrolerului să interfațeze cu dispozitive analogice externe, cum ar fi senzorii.
- Convertor digital-analog (DAC): Un DAC îndeplinește funcția inversă a unui

ADC și permite procesorului din centrul microcontrolerului să comunice semnalele sale de ieșire către componente analogice externe.

- Bus de sistem: Busul de sistem este firul de conexiune care leagă toate componentele microcontrolerului.
- Port serial: Portul serial este un exemplu de port I/O care permite microcontrolerului să se conecteze la componente externe. Are o funcție similară cu un port USB sau paralel, dar diferă prin modul în care schimbă biți.

1.3.3. Caracteristicile microcontrolerului

Procesorul unui microcontroler va varia în funcție de aplicație. Opțiunile variază de la procesoare simple pe 4 biți, 8 biți sau 16 biți până la procesoare mai complexe pe 32 de biți sau 64 de biți. Microcontrolerile pot folosi tipuri de memorie volatile, cum ar fi memoria cu acces aleatoriu (RAM) și tipurile de memorie nevolatilă: acestea includ memoria flash, memoria doar pentru citire programabilă care se șterge (EPROM) și memoria programabilă doar pentru citire (EEPROM) cu ștergere electrică.

În general, microcontrolerile sunt proiectate pentru a fi ușor de utilizat fără componente de calcul suplimentare, deoarece sunt proiectate cu suficientă memorie la bord și oferă pini pentru operațiuni generale I/O, astfel încât să poată interfața direct cu senzorii și alte componente.

Arhitectura microcontrolerului poate fi bazată pe arhitectura Harvard sau arhitectura von Neumann, ambele oferind metode diferite de schimb de date între procesor și memorie. Cu o arhitectură Harvard, magistrala de date și instrucțiunile sunt separate, permițând transferuri simultane. Cu o arhitectură Von Neumann, o magistrală este utilizată atât pentru date, cât și pentru instrucțiuni.

Procesoarele cu microcontroler se pot baza pe calcularea setului de instrucțiuni complexe (CISC) sau a setului de instrucțiuni redus (RISC). CISC are în general aproximativ 80 de instrucțiuni, în timp ce RISC are aproximativ 30, precum și mai multe moduri de adresare, 12-24 în comparație cu RISC 3-5. În timp ce CISC poate fi mai ușor de implementat și are o utilizare mai eficientă a memoriei, poate avea o degradare a performanței din cauza numărului mai mare de cicluri de ceas necesare pentru a executa instrucțiuni. RISC, care pune mai mult accent pe software, oferă adesea performanțe mai bune decât procesoarele CISC, care pun mai mult accent pe hardware, datorită setului său de instrucțiuni simplificat și, prin urmare, simplității sporite a designului, dar din cauza accentului pe care îl pune pe software, software-ul poate fi mai complex. Ce ISC este utilizat variază în funcție de aplicație.

Când au devenit disponibile pentru prima dată, microcontrolerile foloseau doar limbaj de asamblare. Astăzi, limbajul de programare C este o opțiune populară. Alte limbaje comune pentru microprocesoare sunt Python și JavaScript.

MCU-urile dispun de pini de intrare și de ieșire pentru a implementa funcții periferice. Astfel de funcții includ convertoare analog-digitale, controlere cu afișaj cu cristale lichide (LCD), ceas în timp real (RTC), transmițător receptor universal sincron/asincron (USART), temporizatoare, transmițător receptor universal asincron (UART) și magistrală serial universală (conectivitate USB). Senzorii care adună date legate de umiditate și temperatură, printre altele, sunt adesea atașați la microcontrolere.

1.3.4. Aplicații pentru microcontrolerile

Microcontrolerile sunt folosite în mai multe industrii și aplicații, inclusiv în casă și întreprindere, automatizarea clădirilor, producție, robotică, auto, iluminat, energie inteligentă,

automatizare industrială, comunicații și implementări de internet al lucrurilor (IoT).

O aplicație foarte specifică a unui microcontroler este utilizarea acestuia ca procesor de semnal digital. Frecvent, semnalele analogice de intrare vin cu un anumit nivel de zgomot. Zgomotul în acest context înseamnă valori ambigue care nu pot fi traduse cu ușurință în valori digitale standard. Un microcontroler își poate folosi ADC și DAC pentru a converti semnalul analog zgomotos de intrare într-un semnal digital uniform de ieșire.

Cele mai simple microcontrolere facilitează operarea sistemelor electromecanice care se găsesc în articolele uzuale de zi cu zi, cum ar fi cuptoarele, frigiderele, prăjitoarele de pâine, dispozitivele mobile, cheile, sistemele de jocuri video, televizoarele și sistemele de udare a gazonului. Ele sunt, de asemenea, comune în aparatele de birou, cum ar fi fotocopiatoarele, scanere, faxuri și imprimante, precum și contoare inteligente, bancomate și sisteme de securitate.

Microcontrolerele mai sofisticate îndeplinesc funcții critice în aeronave, nave spațiale, nave oceanice, vehicule, sisteme medicale și de susținere a vieții, precum și în roboți. În scenariile medicale, microcontrolerele pot regla operațiunile unei inimi artificiale, rinichi sau alte organe. Ele pot fi, de asemenea, esențiale în funcționarea dispozitivelor protetice.

1.3.5. Microcontrolere vs microprocesoare

Distincția dintre microcontrolere și microprocesoare a devenit mai puțin clară, deoarece densitatea și complexitatea cipurilor au devenit relativ ieftine de fabricat, iar microcontrolerele au integrat astfel mai multe tipuri de funcționalități „generale de computer”. În general, totuși, se poate spune că microcontrolerele funcționează util pe cont propriu, cu o conexiune directă la senzori și actuatori, unde microprocesoarele sunt proiectate pentru a maximiza puterea de calcul pe cip, cu conexiuni interne de magistrală (mai degrabă decât I/O directă) la hardware suport, cum ar fi RAM și porturi seriale. Mai simplu spus, aparatele de cafea folosesc microcontrolere iar computerele desktop folosesc microprocesoare.

Microcontrolerele sunt mai puțin costisitoare și folosesc mai puțină energie decât microprocesoarele. Microprocesoarele nu au RAM încorporată, memorie read-only (ROM) sau alte periferice pe cip, ci mai degrabă se atașează la acestea cu pinii lor. Un microprocesor poate fi considerat inima unui sistem informatic, în timp ce un microcontroler poate fi considerat inima unui sistem încorporat.

1.4. Protocolul de comunicare I2C

În acest subcapitol voi prezenta noțiunile găsite despre sisteme în conformitate cu [6].

I2C înseamnă Inter-Integrated Circuit. Este un protocol de conectare a interfeței magistralei încorporat în dispozitivele pentru comunicații seriale. Acesta a fost proiectat inițial de Philips Semiconductor în 1982. Recent, este un protocol utilizat pe scară largă pentru comunicarea la distanță scurtă. Este, de asemenea, cunoscut sub numele de interfață cu două fire (TWI).

1.4.1. Funcționarea protocolului de comunicare I2C

Folosește doar 2 linii bidirecționale deschise pentru comunicarea de date, numite SDA și SCL. Ambele linii sunt trase sus.

- Date seriale (SDA): transferul datelor are loc prin acest pin;
- Serial Clock (SCL): transportă semnalul ceasului.

I2C operează în 2 moduri:

- modul Master;
- modul Slave.

Fiecare bit de date transferat pe linia SDA este sincronizat printr-un impuls ridicat cu cel scăzut al fiecărui ceas de pe linia SCL.

Conform protocoalelor I2C, linia de date nu se poate modifica atunci când linia de ceas este ridicată, se poate schimba numai atunci când linia de ceas este scăzută. Cele 2 linii sunt de scurgere deschisă, prin urmare este necesar un rezistor de tragere, astfel încât liniile să fie înalte, deoarece dispozitivele de pe magistrala I2C sunt active la nivel scăzut. Datele sunt transmise sub formă de pachete care cuprind 9 biți. Secvența acestor biți sunt:

- Condiția de start – 1 bit;
- Adresa Slave – 8 biți;
- Confirmare – 1 bit.

1.4.2. Condițiile de START/STOP

START și STOP pot fi generate menținând linia SCL ridicată și schimbând nivelul SDA. Pentru a genera condiția START, SDA este schimbat de la mare la scăzut, menținând SCL ridicat. Pentru a genera starea STOP, SDA trece de la scăzut la ridicat, menținând SCL ridicat, așa cum se arată în Figura 1.3.



Figura 1.3. Condiția de START/STOP

1.4.3. Condiția de pornire repetată

Între fiecare pereche de condiții de pornire și oprire, bus-ul este considerat ocupat și niciun master nu poate prelua controlul acestuia. Dacă master-ul încearcă să inițieze un nou transfer și nu dorește să elibereze magistrala înainte de a începe noul transfer, emite o nouă condiție de start. Aceasta se numește condiție de pornire repetată.

1.4.4. Bit de citire/scriere

Un bit de citire/scriere ridicat indică faptul că master-ul trimite datele către slave, în timp ce un bit de citire/scriere scăzut indică faptul că master-ul primește date de la slave.

1.4.5. Adresare

Cadrul de adresă este primul cadru după bitul de pornire. Adresa slave-ului cu care master-ul dorește să comunice este trimisă de master către fiecare slave conectat cu acesta. Slave-ul compară apoi propria sa adresă cu această adresă și trimite ACK.

1.4.6. Formatul pachetului I2C

În protocolul de comunicare I2C, datele sunt transmise sub formă de pachete. Aceste pachete sunt lungi de 9 biți, dintre care primii 8 biți sunt puși în linia SDA, iar al 9-lea bit este rezervat pentru ACK/NACK, adică Acknowledge sau Not Acknowledge de către receptor.

Condiția START plus pachetul de adresă plus încă un pachet de date plus condiția STOP formează împreună un transfer complet de date.

1.4.7. Caracteristici ale protocolului de comunicare I2C

- Protocol de comunicare semi-duplex: comunicarea bidirecțională este posibilă, dar nu simultan;
- Comunicare sincronă: datele sunt transferate sub formă de cadre sau blocuri;
- Poate fi configurat într-o configurație multi-master;
- Întinderea ceasului: ceasul este întins atunci când dispozitivul slave nu este pregătit să accepte mai multe date ținând linia SCL scăzută, dezactivând astfel master-ul să ridice linia ceasului. Master-ul nu va putea ridica linia de ceas deoarece firele sunt cablate AND și așteaptă până când slave-ul eliberează linia SCL pentru a arăta că este gata să transfere următorul bit;
- Arbitraj: protocolul I2C acceptă sistemul de magistrală multi-master, dar mai mult de o magistrală nu poate fi utilizată simultan. SDA și SCL sunt monitorizate de către master. Dacă SDA este găsit ridicat atunci când ar fi trebuit să fie scăzut, se va deduce că un alt master este activ și, prin urmare, oprește transferul de date;
- transmisie în serie: I2C utilizează transmisia în serie pentru transmiterea datelor;
- Folosit pentru comunicare la viteză mică.

1.4.8. Avantaje

- Poate fi configurat în modul multi-master;
- Complexitatea este redusă deoarece folosește doar 2 linii bidirecționale (spre deosebire de comunicația SPI);
- Cost eficient;
- Utilizează caracteristica ACK/NACK datorită căreia are capacități îmbunătățite de tratare a erorilor.

1.4.9. Limitări

- Viteză mică;
- Comunicarea semi-duplex este utilizată în protocolul de comunicare I2C.

1.5. Breadboard

În acest subcapitol voi prezenta noțiunile găsite despre sisteme în conformitate cu [7].

O placă breadboard este folosită pentru a construi și a testa rapid circuite. Placa are multe găuri în care pot fi inserate diferite componente ale circuitelor, precum circuitele integrate și rezistențe. O placă tipică este prezentată în Figura 1.4.

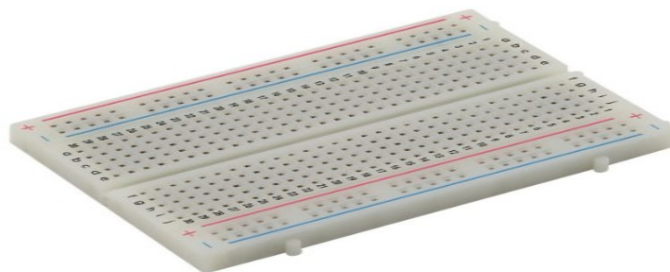


Figura 1.4.breadboard clasic

Cablurile se pot potrivi în placa de breadboard, deoarece interiorul unei plăci de breadboard este alcătuit din rânduri de cleme de metal minuscule. Un exemplu de interior al unui breadboard se poate vedea în Figura 1.5.

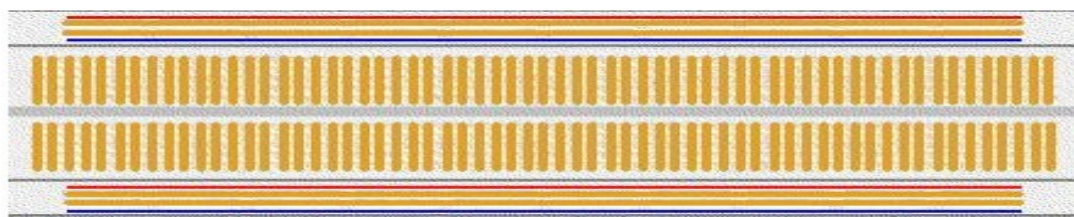


Figura 1.5. Interiorul unui breadboard

Conexiunile nu sunt permanente, așa că este ușor să eliminați o componentă dacă faceți o greșală, sau doar o luați de la capăt și faceți un nou proiect. Acest lucru face ca plăcile să fie excelente pentru începători. Puteți folosi breadboarduri pentru a realiza tot felul de proiecte electronice, de la diferite tipuri de roboți sau un set de tobe electronice, la un detector electronic de ploaie pentru a ajuta la conservarea apei într-o grădină, sau o alarmă auto, în cazul acesta.

În Figura 1.6 este un exemplu de proiect ce se poate face cu ajutorul unui breadboard.

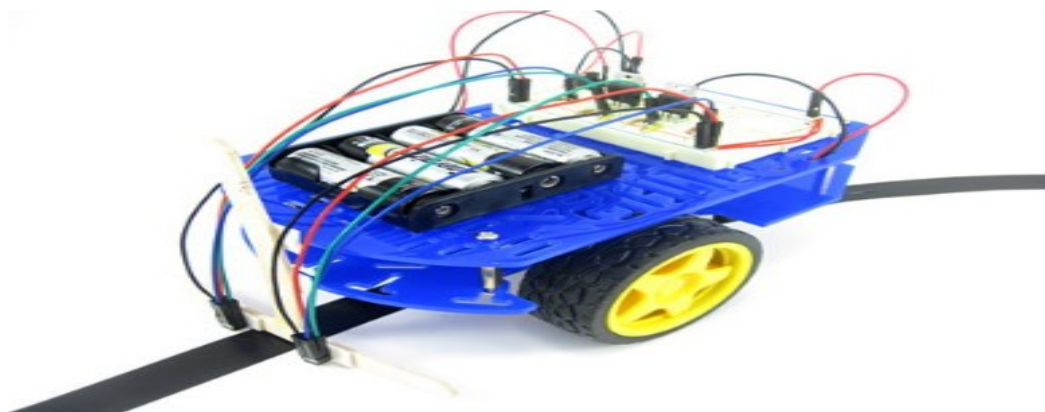


Figura 1.6. Exemplu de proiect făcut cu ajutorul unui breadboard

1.5.1. Tipuri de breadboarduri

Plăcile moderne sunt fabricate din plastic și vin în toate formele, dimensiunile și chiar diferite culori. În timp ce sunt disponibile dimensiuni mai mari și mai mici, cele mai frecvente dimensiuni pe care le veți vedea probabil sunt plăcile „full-size”, „half-size” și „mini”. Majoritatea plăcilor de breadboard vin și cu urechi și creștături pe laterale care vă permit să fixați mai multe plăci împreună. Cu toate acestea, o singură placă de breadboard de jumătate de dimensiune este suficientă pentru multe proiecte de nivel începător.

În Figura 1.7 se pot observa cele mai frecvente 3 versiuni ale plăcii.

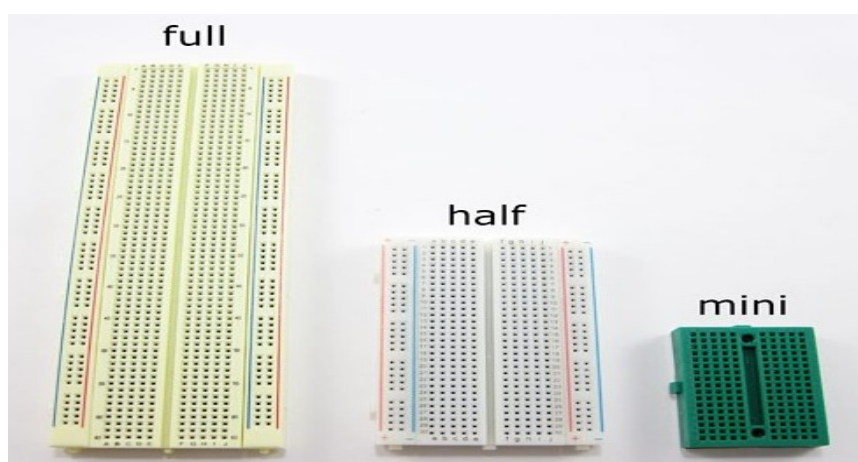


Figura 1.7. Modele de breadboarduri

Din punct de vedere tehnic, aceste plăci sunt numite plăci fără lipire, deoarece nu necesită lipire pentru a face conexiuni. Lipirea este o metodă prin care componentele electronice sunt unite între ele prin topirea unui tip special de metal. Componentele electronice pot fi lipite direct împreună, dar mai frecvent sunt lipite pe plăci de circuite imprimate (PCB). PCB-urile sunt ceea ce veți vedea dacă scoateți capacul multor dispozitive electronice, cum ar fi un computer sau un telefon mobil. Frecvent, inginerii vor folosi panouri fără sudură pentru a prototipa și a testa un circuit înainte de a construi designul final, permanent pe un PCB.

1.5.2. Rânduri, coloane și etichete

Cele mai multe plăci au câteva numere, litere și semne plus și minus scrise pe ele. În timp ce aspectul lor exact poate varia de la o placă la alta, scopul general este întotdeauna același. Aceste etichete vă ajută să localizați anumite găuri pe placa de breadboard, astfel încât să puteți urma instrucțiunile atunci când construiți un circuit. Numerele rândurilor și literele de coloană vă ajută să identificați găurile individuale în panoul, la fel ca celulele dintr-o foaie de calcul. De exemplu, în Figura 1.8 toate găurile evidențiate sunt în „coloana C”, iar în Figura 1.9 toate găurile evidențiate sunt în rândul 12

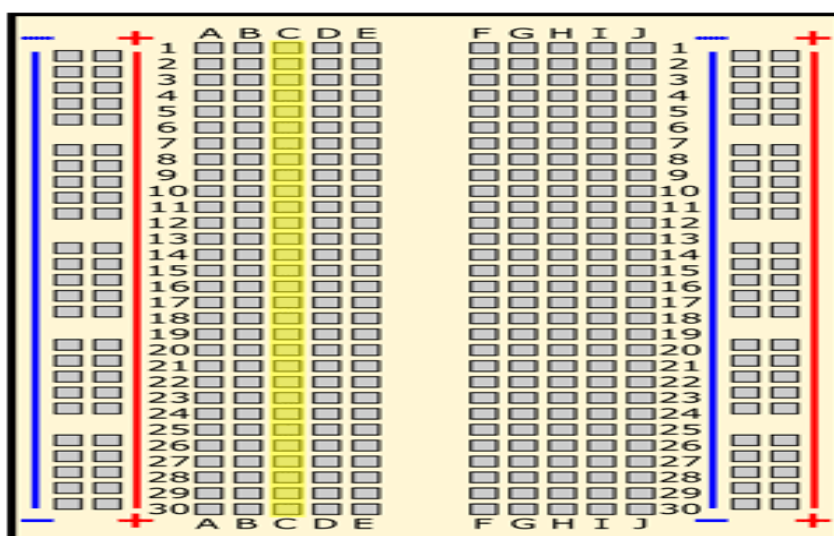


Figura 1.8. Breadboard în care coloana C este evidențiată

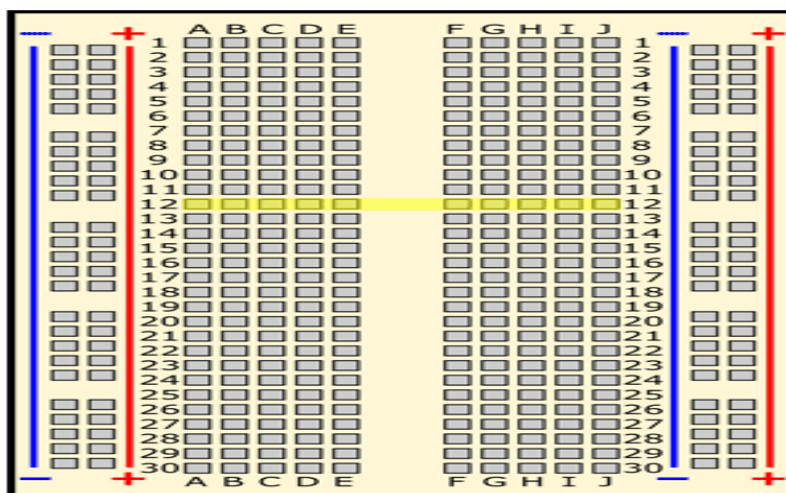


Figura 1.9. Breadboard în care rândul 12 este evidențiat

1.5.3. Conectarea găurilor

Fiecare set de cinci găuri care formează o jumătate de rând (coloanele A–E sau coloanele F–J) este conectat electric. De exemplu, asta înseamnă că orificiul A1 este conectat electric la orificiile B1, C1, D1 și E1. Nu este conectat la orificiul A2, deoarece orificiul respectiv este pe un rând diferit, cu un set separat de cleme metalice. De asemenea, nu este conectat la găurile F1, G1, H1, I1 sau J1, deoarece acestea se află pe cealaltă „jumătate” a plăcii de breadboard, iar clemele nu sunt conectate peste golul din mijloc. Spre deosebire de toate rândurile principale de panouri, care sunt conectate în seturi de cinci găuri, busurile rulează de obicei pe toată lungimea plăcii (dar există unele excepții). În Figura 1.10 se pot vedea ce găuri sunt conectate electric într-o placă tipică de jumătate de dimensiune, evidențiată în linii galbene.

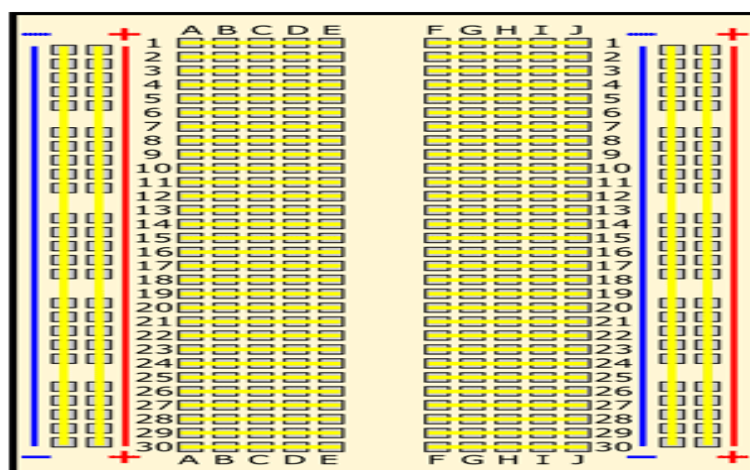


Figura 1.10. Orificii conectate între ele

1.5.4. Probleme ce pot apărea

1.5.4.1. Conectarea greșită la rânduri

O problemă ce apare foarte des este conectarea greșită la rânduri, acestea fiind numerotate foarte mic, se poate greși foarte ușor dacă atenția nu este maximă. Un exemplu de astfel de greșeală putem vedea în Figura 1.11.

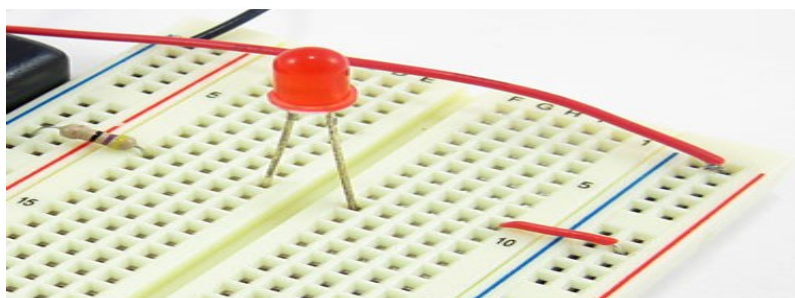


Figura 1.11. Greșeală la încurcarea rândurilor

1.5.4.2. Încurcarea alimentării cu masa

La unele plăci de breadboard alimentarea poate fi în stânga și masa în dreapta, dar la altele poate fi total invers, așadar, similar cu greșirea numerelor de rând, amestecarea magistralelor de alimentare și masă este o altă greșeală comună. Un exemplu de astfel de greșeală putem vedea în Figura 1.12.

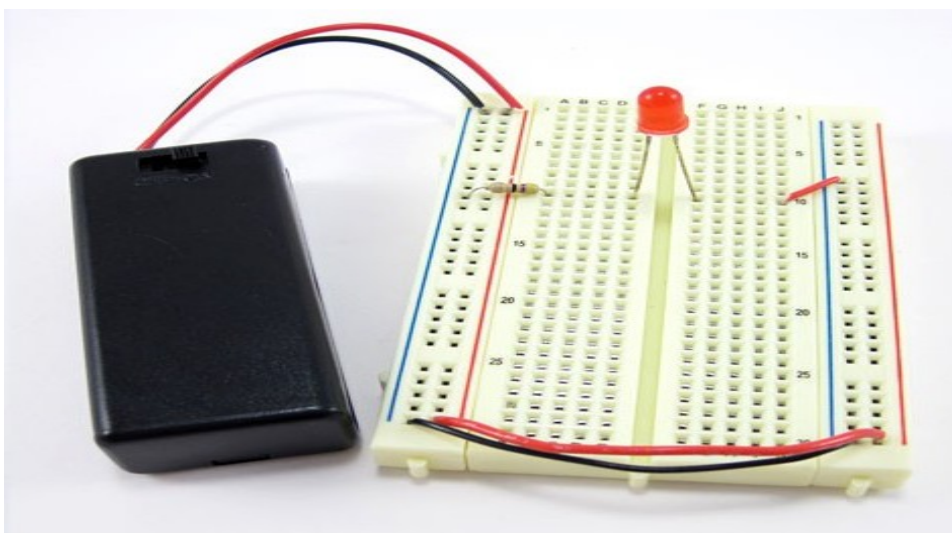


Figura 1.12. Greșeală la încurcarea alimentării cu masa

1.5.4.3. Conectarea cablurilor și firelor necorespunzător

Componentele electronice și firele jumper pot avea toate cabluri de lungimi diferite. Uneori, din greșeală, se vor împinge cablurile doar parțial într-o gaură a plăcii, în loc să le împingă ferm în jos până la capăt (până când nu pot merge mai departe). Acest lucru poate duce la conexiuni slăbite care duc la un comportament ciudat al circuitului, cum ar fi un LED care pâlpâie și se stinge. Un exemplu de astfel de greșeală putem vedea în Figura 1.13.

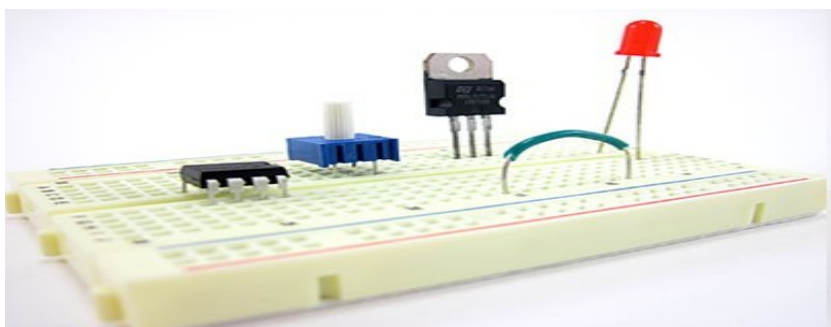


Figura 1.13. Greșeală în împingerea firelor doar parțial

1.5.4.4. Punerea componentelor invers

Pentru unele componente electronice, direcția contează. Unele componente au polaritate, adică au o latură pozitivă și o parte negativă care trebuie conectate corect. Alte componente au mai mulți pini care au toate funcții diferite. Introducerea acestor componente în circuit cu spatele sau cu fața în direcția greșită va împiedica circuitul să funcționeze corect. Dacă circuitul nu funcționează, trebuie să verificăm că sunt introduse corect.

- LED-urile: au o parte pozitivă (numită anod) și o parte negativă (numită catod). Plumbul metalic pentru anod este mai lung decât plumbul pentru catod. Partea catodului are de obicei o margine plată pe partea din plastic a LED-ului;
- Diodele: sunt ca niște supape unidirecționale care lasă electricitatea să curgă doar într-o singură direcție. De obicei, sunt cilindri mici marcați cu o bandă sau o bandă la un capăt (aceasta este direcția spre care poate curge electricitatea);
- Condensatorii: sunt componente care pot stoca sarcina electrică. Condensatorii obișnuiți „disc ceramic” nu sunt polarizați, dar alte câteva tipuri de condensatoare sunt și vor avea de obicei semne săgeți sau minus îndreptate către conductorul negativ;
- Tranzistorii: sunt comutatoare controlate electronic care pot fi folosite pentru a porni și stinge lucruri precum motoare și lumini. Tranzistoarele au în general trei pini. Punerea unui tranzistor într-o placă de breadboard înapoi va inversa ordinea pinilor și va împiedica funcționarea acestuia. Tranzistorii vin în mai multe „pachete” diferite, de obicei un corp din plastic negru cu scriere mică pe o parte.

Capitolul 2. Structura hardware a proiectului

Având în vedere faptul că în capitolul anterior am descris principalele concepte care stau la baza creării acestui proiect pentru a ne familiariza cu acestea, o să detaliez în amănunt detaliile tehnice legate de structura hardware a proiectului. Voi detalia în acest capitol fiecare componentă folosită pentru a înțelege pe viitor importanța acesteia în cadrul sistemului, cât și modul de conectare și utilizare a acestora.

2.1. Sisteme folosite

Pentru realizarea proiectului de licență s-a ales crearea unei alarme auto, începând cu conectarea senzorilor, până la construirea machetei unui autoturism.

În proiect au fost integrate sistemele următoare:

- Sistemul de detecție de mișcare;
- Sistemul de detecție a sunetului;
- Sistemul de detecție a accelerației;
- Sistem de avertizare.

Împreună, toate aceste sisteme combinate vor aduce în viața oamenilor o stare de liniște, aceștia putând sta liniștiți în privința securității mașinilor deținute.

2.2. Componente hardware

Pentru realizarea acestui proiect s-au folosit componentele următoare:

- Laptop;
- O plăcuță Plusivo;
- BreadBoard;
- Senzor ultrasonic;
- Modul accelerometru;
- Modul sunet;
- Buzzer;
- Pistol de lipit;
- Fire mamă-mamă;
- Fire tată-tată.
- Fire mamă-tată;

2.3. Funcționalitatea sistemelor

Sistemele folosite în proiect pot fi folosite atât împreună, cât și separat, dar, în caz de îmbunătățire a alarmei pot fi adăugate nenumărate alte sisteme.

2.3.1. Sistemul de detecție al mișcării

Acest sistem permite detecția mișcărilor în autoturism cu ajutorul senzorului ultrasonic, ce va sta în spatele mașinii, astfel interceptând orice mișcare produsă în interior.

2.3.2. Sistemul de detecție al sunetului

Acest sistem permite detecția sunetului în autoturism, reglând manual sensibilitatea acestuia, permițându-ne astfel să filtrăm doar acele sunete peste o anumită marjă. Acesta este foarte important atunci când cineva dorește să intre prin efracție în mașină, spărgând un geam.

2.3.3. Sistemul de detecție a greutății

Acest sistem permite detecția greutății pe un scaun, dacă cumva până acum intrusul a trecut de celelalte sisteme de siguranță. Atunci când infractorul se așează pe un scaun al automobilului, acesta va înregistra o greutate și va porni alarma.

2.3.4. Sistemul de detecție al accelerațiilor

Acest sistem permite detecția mișcărilor automobilului, pentru ca atunci când mașina este lovită în parcare, sau poate împinsă de cineva, ridicată de autorități, să pornească alarma.

2.3.5. Sistemul de avertizare

Acest sistem va anunța utilizatorul atunci când un senzor este declanșat, astfel va porni o alarmă sonoră în perimetrul autoturismului, pentru a atrage atenția celor din jur și pentru a speria infractorul.

2.4. Programarea sistemelor

Pentru programarea sistemelor am folosit placa de dezvoltare PLUSIVO, care are aceleași caracteristici ca Arduino UNO, amândouă fiind plăci de dezvoltare bazate pe microcontrollerul ATmega328. În continuare voi prezenta informații despre plăcuța Arduino UNO preluate de pe pagina oficială, informații ce pot fi găsite în [8] și niște informații adiționale care pot fi găsite în [4],[9],[10].

2.4.1. Prezentare generală Arduino Uno

Aceasta are 14 pini de intrare/ieșire digitale dintre care 6 pot fi utilizați ca ieșiri PWM, 6 intrări analogice, un rezonator ceramic de 16 MHz, o conexiune USB, o mufă de alimentare, o mufă ICSP și un buton de resetare. Plăcuța conține tot ce este necesar pentru a suporta microcontroller-ul; trebuie conectat la un computer printr-un cablu USB, alimentat cu un adaptor AC-DC sau baterie pentru a începe.

De când a fost lansat pentru prima dată, Arduino Uno a avut un succes uriaș printre pasionații de electronică, de la pasionații începători până la programatorii profesioniști. Este o platformă open-source, ce înseamnă că plăcile și software-ul sunt ușor disponibile și oricine poate modifica și optimiza plăcile pentru o funcționalitate mai bună. Software-ul folosit pentru dispozitivele Arduino se numește IDE (Integrated Development Environment) care este liber de utilizat și necesită unele abilități de bază pentru a-l învăța. Poate fi programat folosind limbajul C și C++.

Această placă vine cu toate caracteristicile necesare pentru a rula controlerul și poate fi conectată direct la computer prin cablu USB care este folosit pentru a transfera codul către controler folosind software-ul IDE (Integrated Development Environment), dezvoltat în principal pentru a programa Arduino. Putem vedea un exemplu de plăcuță în Figura 2.1.

Mediul de dezvoltare integrat (IDE) este un software simplu și ușor de învățat pentru scrierea codurilor Arduino.

Arduino are o flexibilitate uriașă cu care poți face aproape orice îți imaginezi. Poate fi conectat cu ușurință la o varietate de module precum senzori de incendiu, senzori de obstacole, detectoare de prezență, module GPS, module GSM sau orice altceva.

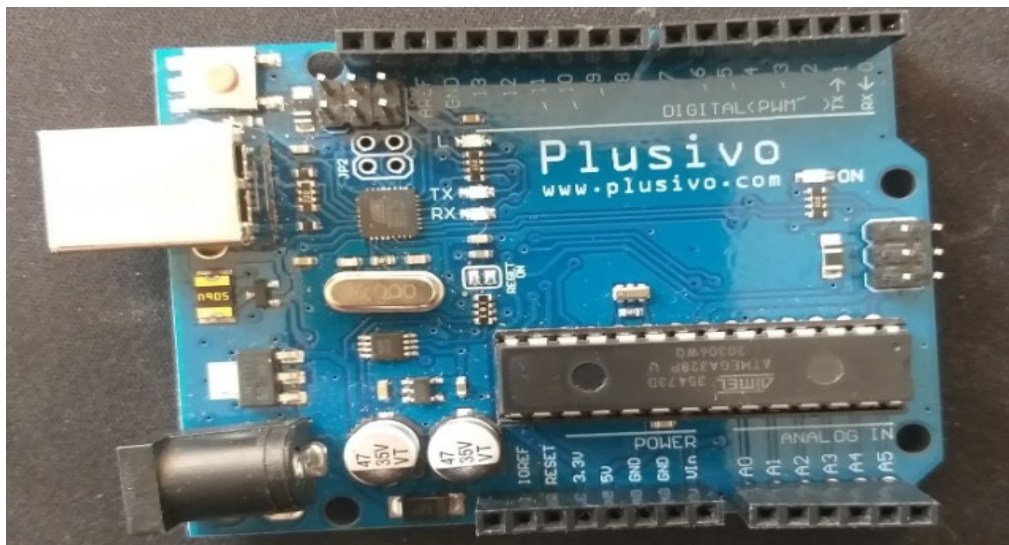


Figura 2.1. Plăcuța Plusivo

IDE este la fel de compatibil cu sistemele Windows, MAC sau Linux, cu toate acestea, este preferabil să se utilizeze Windows. Este o platformă open source unde oricine poate modifica și optimiza placa în funcție de numărul de instrucțiuni și de sarcina pe care dorește să le realizeze. În Tabelul 2.1 se pot găsi informații referitoare la specificațiile unei plăci Arduino.

Tabelul 2.1. Specificații placă Arduino

Microcontroler	ATmega328
Voltajul de operare	5V
Tensiune de intrare recomandată	7-12V
Limitele tensiunii de intrare	6-20V
Pini de intrare analogică	6 (A0-A5)
Pini digitali I/O	14 (din care 6 pot fi folosiți ca PWM)
Curent DC pe pini I/O	40 mA
Curent DC pe pinul de 3.3V	50 mA
Memoria Flash	32 KB din care 0.5 KB utilizați de Bootloader
SRAM	2 KB
EEPROM	1 KB
Frecvența (Clock Speed)	16 MHz

2.4.2. Alimentarea plăcuței

Placa Arduino UNO poate fi alimentată printr-o conexiune USB sau cu o sursă de alimentare externă. Sursa de alimentare este selectată automat.

Alimentarea externă (non-USB) poate proveni fie de la un adaptor AC-DC (de perete) sau de la o baterie. Adaptorul poate fi conectat prin conectarea unei mufe centrale pozitive de 2.1 mm la mufa de alimentare a plăcii. Cablurile de la o baterie pot fi introduse în anteturile pinului

GND și Vin ale conectorului POWER.

Placa poate funcționa pe o sursă externă de la 6 la 20 volți. Cu toate acestea, dacă este furnizat cu mai puțin de 7V, pinul de 5V poate furniza mai puțin de cinci volți și placa poate deveni instabilă. Dacă se utilizează mai mult de 12 V, regulatorul de tensiune se poate supraîncălzi și deteriorează placa. Intervalul recomandat este de la 7 la 12 volți.

Pinii de alimentare sunt următorii:

- Vin: tensiunea de intrare la placa Arduino atunci când folosește o sursă de alimentare externă (spre deosebire de 5 volți de la conexiunea USB sau alte surse de alimentare reglementate). Puteți furniza tensiune prin acest pin sau, dacă furnizați tensiune prin mufa de alimentare, accesați-l prin acest pin;
- 5V: acest pin scoate o tensiune de 5V reglată de la regulatorul de pe placă. Placa poate fi alimentată fie de la mufa de alimentare DC (7-12V), conectorul USB (5V), fie de la pinul VIN al plăcii (7-12V). Alimentarea cu tensiune prin pinii de 5V sau 3,3V ocolește regulatorul și poate deteriorează placa;
- 3V3: O sursă de 3.3 volți generată de către regulatorul de bord. Consumul maxim de curent este de 50 mA;
- GND: Pini de împământare;
- IOREF: acest pin de pe placa Arduino oferă referința de tensiune cu care funcționează microcontrolerul. Un scut configurat corespunzător poate citi tensiunea pinului IOREF și poate selecta sursa de alimentare adecvată sau poate permite traducătorilor de tensiune de pe ieșiri să funcționeze cu 5V sau 3,3V.

2.4.3. Comunicația plăcuței

Arduino poate fi folosit pentru a comunica cu un computer, o altă placă Arduino sau alte microcontrolere. Microcontrolerul ATmega328P oferă comunicație serială UART TTL (5V), care poate fi realizată folosind pinul digital 0 (Rx) și pinul digital 1 (Tx). Un ATmega16U2 de pe placă canalizează această comunicare serială prin USB și apare ca un port de comunicație virtual pentru software-ul de pe computer. Firmware-ul ATmega16U2 folosește driverele standard USB COM și nu este necesar niciun driver extern. Cu toate acestea, pe Windows, este necesar un fișier .inf. Software-ul Arduino include un monitor serial care permite trimiterea de date textuale simple către și de la placa Arduino. Există două LED-uri RX și TX pe placa arduino care vor clipi când datele sunt transmise prin cipul USB-la-serial și conexiunea USB la computer (nu pentru comunicarea în serie pe pinii 0 și 1). O bibliotecă SoftwareSerial permite comunicarea în serie pe oricare dintre pinurile digitale ale plăcuței. ATmega328P acceptă, de asemenea, comunicații I2C (TWI) și SPI. Software-ul Arduino include o bibliotecă Wire pentru a simplifica utilizarea magistralei I2C.

2.4.4. Configurația pinilor

Cei 14 pini de intrare/ieșire digitali pot fi utilizați ca pini de intrare sau de ieșire folosind funcțiile pinMode(), digitalRead() și digitalWrite(). Fiecare pin funcționează la 5 volți și poate furniza sau primi un curent de maxim 40mA și are o rezistență internă de 20-50 KOhmi. Dintre acești 14 pini, unii pini au funcții specifice, așa cum sunt enumerate mai jos:

- Vin: acesta este pinul de tensiune de intrare al plăcii Arduino folosit pentru a furniza alimentarea de la o sursă de alimentare externă;
- 5V: acest pin al plăcii Arduino este folosit ca tensiune de alimentare reglată și este folosit pentru a furniza alimentarea plăcii, precum și a componentelor de la bord;
- 3.3V: acest pin al plăcii este folosit pentru a furniza o sursă de 3.3V care este

2.5. Senzori utilizați

După cum am precizat în unul dintre subcapitolele anterioare, am folosit în implementarea proiectului diferiți senzori și module. Voi prezenta în mod detaliat fiecare dintre conceptele pe care le-am utilizat în vederea funcționării corecte a circuitului final prezentând informații preluate din [11], [12], [13], [14], [15], [16], [17], [18] și [19].

2.5.1. Senzor ultrasonic

Senzorul ultrasonic reprezintă un senzor de detectare a mișcării folosind ultrasunete, numit US-100, fiind capabil să măsoare sau să detecteze un obiect aflat la o distanță de la 2 cm la 450 cm. Acesta funcționează de la 2.4 V la 4.5 V, oferind moduri de ieșire de date digitale și seriale. Senzorul poate fi folosit atât cu microcontrolere de 3.3 V, cât și cu 5 V, dar consumă 2mA atunci când este inactiv.

Senzorul are 5 pini:

- 5V: alimentare 3.3-5 V;
- GND: împământare;
- Tx: este necesar un impuls de 10 microsecunde pentru ca transmițătorul cu ultrasunete să înceapă să funcționeze. Acest pin poate fi folosit și ca transmițător (RX) în comunicațiile seriale;
- Rx: afișează timpul de călătorie a undelor sub formă de impulsuri. Această bază poate fi folosită și ca receptor (RX) în comunicațiile seriale.

Caracteristicile tehnice ale senzorului sunt prezentate în Tabelul 2.2.

Tabelul 2.2. Caracteristici tehnice US-100

Tensiunea de operare (VDC)	2.4 V până la 5 V
Frecvență	40 000 Hz
Consumul mediu curent	2 mA
Distanța de detectare Max	450 cm
Unghi de detectare	15 grade
Diametrul senzorului	16 mm
Dimensiune senzor	45 mm x 20 mm
Greutate	9 g
Temperatura de lucru	20 C – 70 C

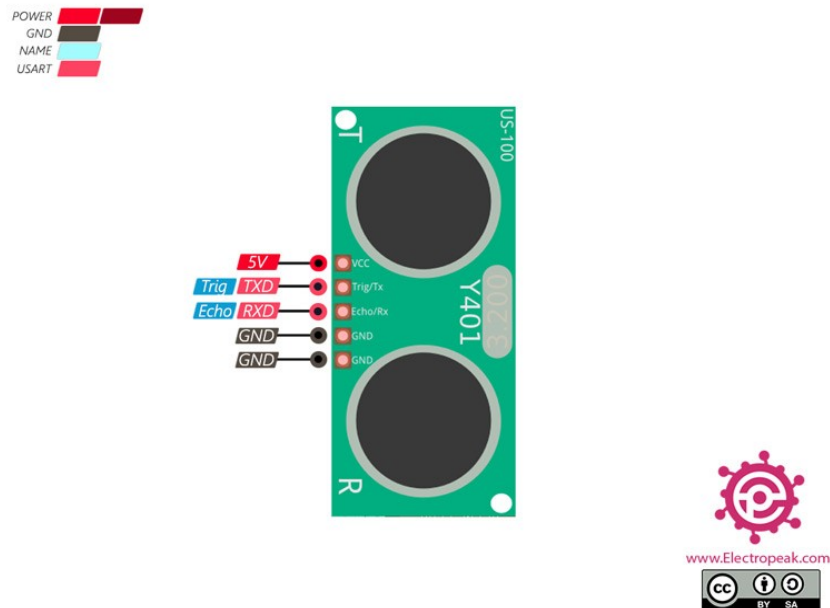


Figura 2.3.Senzorul US-100

Figura 2.3 reprezintă senzorul ultrasonic folosit, punându-se accent pe porturile acestuia.

2.5.2. Modul accelerometru

Modulul accelerometru MPU6050 conține atât un giroscop, cât și un accelerometru cu trei axe. Acesta este foarte precis deoarece conține convertitor 16 biți analog către digital pentru fiecare canal, astfel încât capturează pe toate cele trei axe simultan. Senzorul folosește interfața I2C pentru a comunica cu placa Arduino.

Modulul are 8 pini:

- VCC: alimentare 3-5 V;
- GND: împământare;
- SCL: I2C Clock;
- SDA: I2C data;
- XDA: Auxiliary I2C Data;
- XCL: auxiliary I2C Clock;
- ADO: Adjusting I2C Address;
- INT: întrerupere.

Caracteristicile tehnice ale modulului sunt prezentate în Tabelul 2.3.

Tabelul 2.3.Caracteristici tehnice MPU6050

Tensiune de alimentare	3-5 V
Rază programare giroscop	$\pm (250,500,1000,2000)$ grade/sec (dps)
Rază programare accelerometru	$\pm (2g,4g,8g,16g)$
Ieșire digitală	I2C

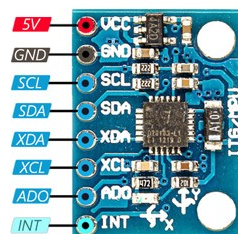


Figura 2.4. Modulul MPU6050

Figura 2.4 reprezintă modulul accelerometru folosit, punându-se accent pe porturile acestuia.

2.5.3. Modul de înregistrare a sunetului

Modulul ce înregistrează sunetul KY-037 constă într-un microfon și un circuit amplificator, iar ieșirea poate fi atât analogică, cât și digitală.

Senzorul are 4 pini:

- Pinul 1: ieșirea analogică. Valoarea sa se schimbă în funcție de intensitatea sunetului primit. Poate fi conectat la pinii analogici Arduino (ADC);
- Pinul 2: alimentare;
- Pinul 3: alimentare;
- Pinul 4: ieșirea digitală. Acționează ca o cheie și se activează atunci când intensitatea sunetului a atins un anumit prag. Pragul de sensibilitate poate fi reglat cu ajutorul potențiometrului de pe senzor.

Caracteristicile tehnice ale senzorului sunt prezentate în Tabelul 2.4.

Tabelul 2.4. Caracteristici tehnice KY-037

Adoptă cipul principal LM393
Microfon electric cu condensator
Caracteristici ieșire: semnal canal unic
Semnal de ieșire la nivel scăzut folosit pentru controlul sunetului
Modul excelent pentru sistem de alarmă sonoră
Tensiune de lucru: DC 4-6 V

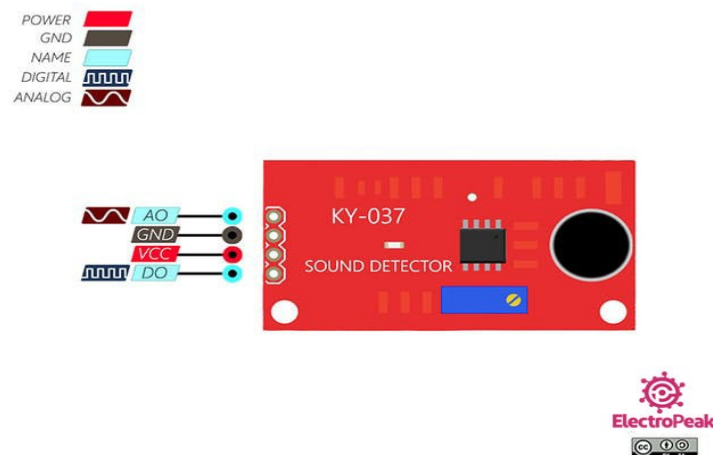


Figura 2.5.Senzorul de sunet KY-037

Figura 2.5 reprezintă modulul de sunet folosit, punându-se accent pe porturile acestuia.

Acest senzor are trei componente funcționale pe placa sa de circuit: unitatea de senzor frontal, care măsoară fizic mediul și îl transmite ca semnal analogic către a doua unitate și amplificatorul. Aceasta amplifică semnalul în funcție de rezistența setată pe potențiometrul rotativ și îl trimite la ieșirea analogică a modulului.

2.5.4. Modul buzzer

Modulul constă într-un buzzer pasiv, care poate genera tonuri între 1.5 și 2.5 kHz prin comutarea și oprirea la frecvențe diferite, fie folosind întârzieri sau PWM. Acesta poate produce o gamă variată de sunete în funcție de frecvența de intrare. În Tabelul 2.5 se pot vedea caracteristicile tehnice ale modulului, iar în Figura 2.6 structura fizică a acestuia.

Tabelul 2.5.Caracteristici tehnice modul buzzer

Tensiune de alimentare	1.5 – 15 V
Gama de generare a tonurilor	1.5 – 2.5 kHz
Dimensiuni	18.5 mm x 15 mm



Figura 2.6.Modul Buzzer pasiv

Pentru controlul acestuia, s-a folosit un semnal PWM (Pulse Width Modulation) precum în Figura 2.7.

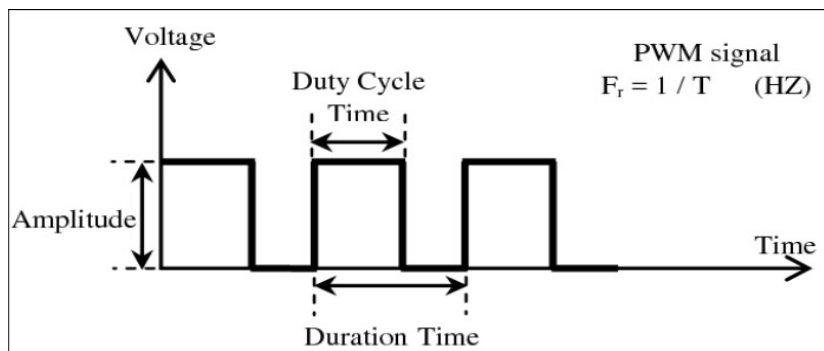


Figura 2.7.PWM

PWM joacă un rol important și critic în multe aplicații și cercetări precum aplicațiile de control al mișcării și conversia și conservarea energiei electrice. Există o mare varietate de tehnici pentru a genera semnale PWM. Unul dintre ele este semnalul PWM bazat pe microcontroler, așa cum este utilizat în microcontrolerul PIC.

Capitolul 3. Funcționalitatea sistemelor

În acest capitol voi prezenta modalitatea de implementare a sistemului de alarmă pe care l-am gândit. Am folosit plăcuța Plusivo pentru a implementa senzorii care mi s-au părut importanți în declanșarea alarmei unei mașini, și anume:

- senzorul ultrasonic US-100;
- modulul de înregistrare a sunetului KY-037;
- modulul accelerometru MPU6050.

Voi descrie în continuare, în mod detaliat, cum au fost aceștia incluși în sistem și voi folosi, pentru fiecare în parte, schema bloc a acestora și o poză în momentul în care aceștia au fost conectați.

3.1. Sistemul de detecție a mișcării

Pentru acest sistem am avut nevoie de:

- un laptop;
- o plăcuță Plusivo;
- un senzor ultrasonic US-100.

Am reprezentat în Figura 3.1 o schemă bloc pentru a putea înțelege mai bine cum am reușit integrarea senzorului în sistem.

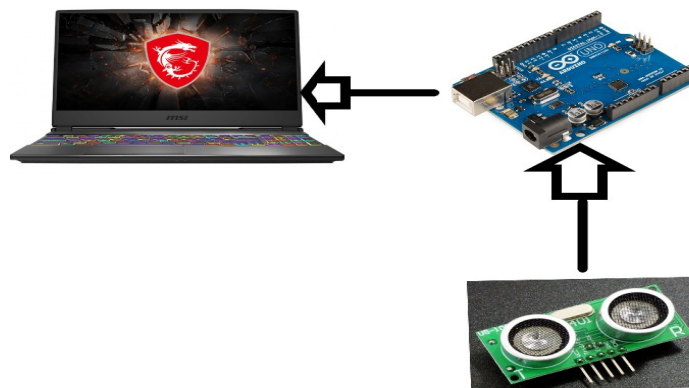


Figura 3.1. Schema bloc a sistemului de detecție a mișcării

Sistemul conține un senzor ultrasonic care se va activa în cazul în care cineva a reușit să intre în mașină, fără a declanșa ceilalți senzori integrați. Scopul acestuia este de a trimite un semnal declanșator sistemului de alarmă în momentul în care detectează o mișcare. Senzorul este capabil să înregistreze mișcarea de la o distanță de minim 10 cm și până la 70 de cm, distanță suficientă pentru ca mișcarea să fie detectată de către senzor în interiorul mașinii. Scopul acestui senzor este de a detecta mișcarea în cazul în care o persoană a reușit să pătrundă în interiorul mașinii, nu de a detecta mișcarea în exteriorul acesteia. În momentul în care acest senzor detectează o mișcare, va trimite un semnal care o să declanșeze sistemul de alarmă, afișând în același timp pe ecran un mesaj de tipul „Alarma a fost activată datorită sistemului de detecție al mișcării!”

În Figura 3.2 senzorul a fost conectat la plăcuță, care la rândul ei este conectată și alimentată de laptop.

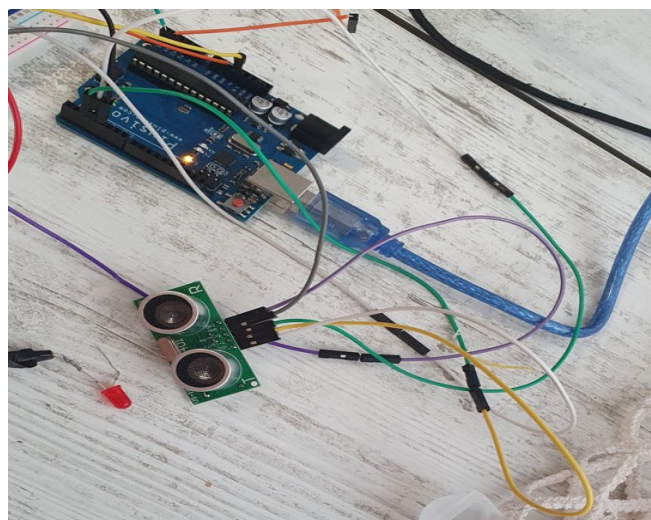


Figura 3.2.Implementarea senzorului US-100 în sistem

3.2. Sistemul de detecție a sunetului

Pentru acest sistem am avut nevoie de:

- un laptop;
- o plăcuță Plusivo;
- un modul de sunet KY-037.

Am reprezentat în Figura 3.3 o schemă bloc pentru a putea înțelege mai bine cum am reușit integrarea modului în sistem.

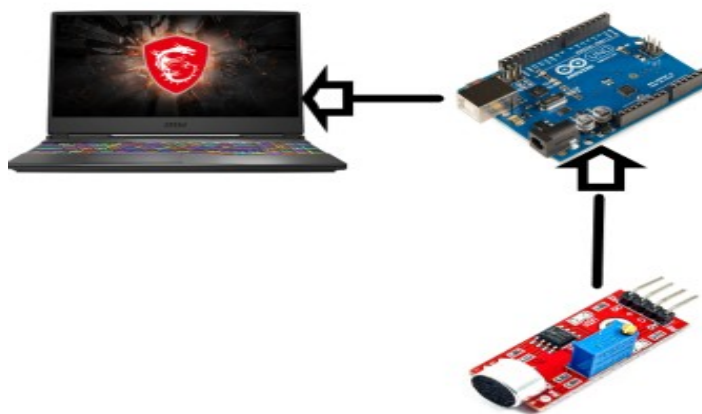


Figura 3.3.Schema bloc a sistemului de detecție a sunetului

Sistemul conține un modul de detecție a sunetului, care, în momentul în care un infractor sparge geamul mașinii acesta va înregistra sunetul, astfel pornind sistemul de alarmă. Modulul va porni alarma doar dacă sunetul înregistrat depășește un prag ales de mine, care poate fi oricând modificat, precum în Figura 3.4.

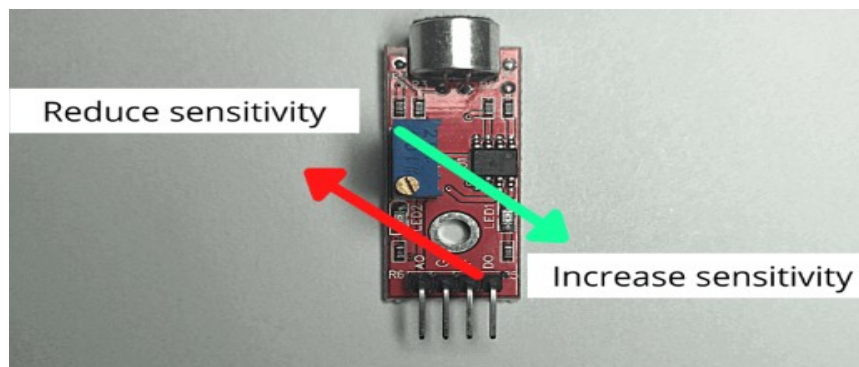


Figura 3.4. Schimbarea sensibilității modulului KY-037

Scopul acestui modul este de a detecta sunetul înăuntrul mașinii, nu de a detecta sunetul în exteriorul acesteia. În momentul în care acest senzor detectează un sunet peste pragul setat, va trimite un semnal care o să declanșeze sistemul de alarmă, afișând în același timp pe ecran un mesaj de tipul „Alarma a fost activată datorită sistemului de detecție al sunetului!”

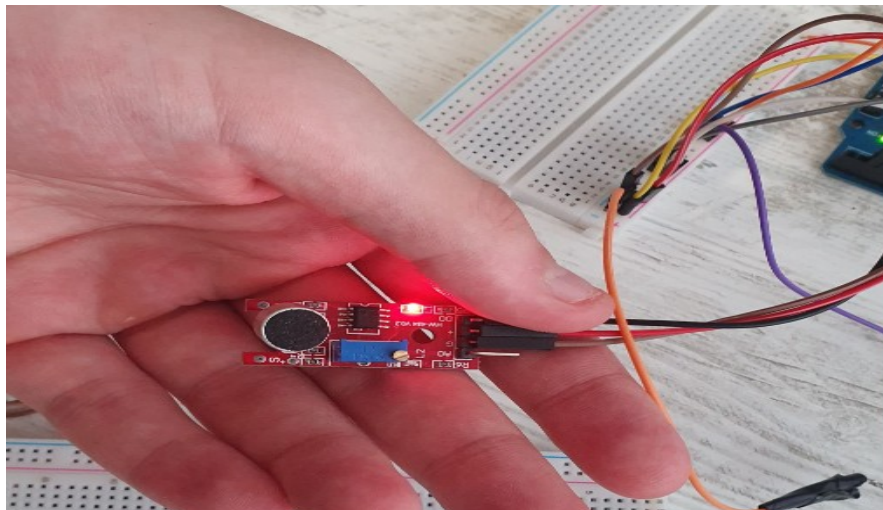


Figura 3.5. Implementarea modulului KY-037 în sistem

În Figura 3.5 modulul a fost conectat la plăcuță, care la rândul ei este conectată și alimentată de laptop.

3.3. Sistemul de detecție al accelerației

Pentru acest sistem am avut nevoie de:

- un laptop;
- o plăcuță Plusivo;
- un modul accelerometru MPU6050.

Am reprezentat în Figura 3.6 o schemă bloc pentru a putea înțelege mai bine cum am reușit integrarea modulului în sistem.

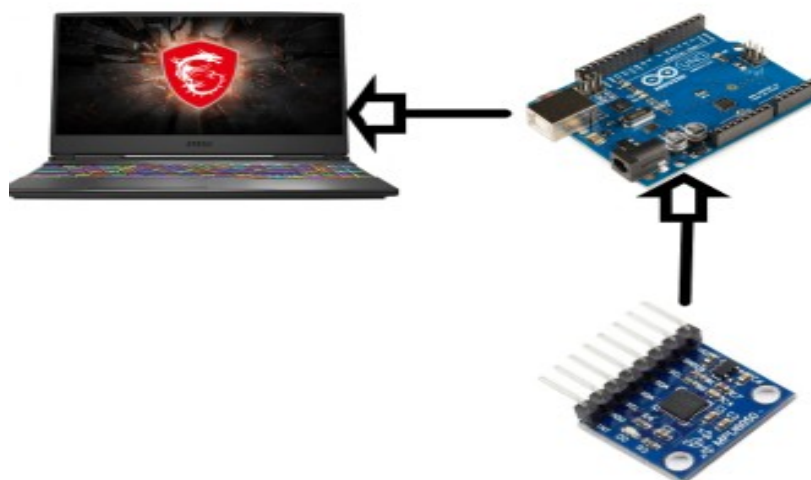


Figura 3.6.Schema bloc a sistemului de detecție a accelerației

Sistemul conține un modul accelerometru și giroscop care se va activa în cazul în care cineva împinge mașina, o lovește, sau o ridică. Scopul acestuia este de a trimite un semnal declanșator sistemului de alarmă în momentul în care detectează o accelerație. Senzorul este capabil să înregistreze accelerația pe 3 axe, astfel încât nu contează în ce direcție se mișcă mașina, orizontal sau vertical, acesta va trimite semnalul când înregistrează o accelerație.

În momentul în care acest senzor detectează o accelerație, va trimite un semnal care o să declanșeze sistemul de alarmă, afișând în același timp pe ecran un mesaj de tipul „Alarma a fost activată datorită sistemului de detecție al accelerației!”

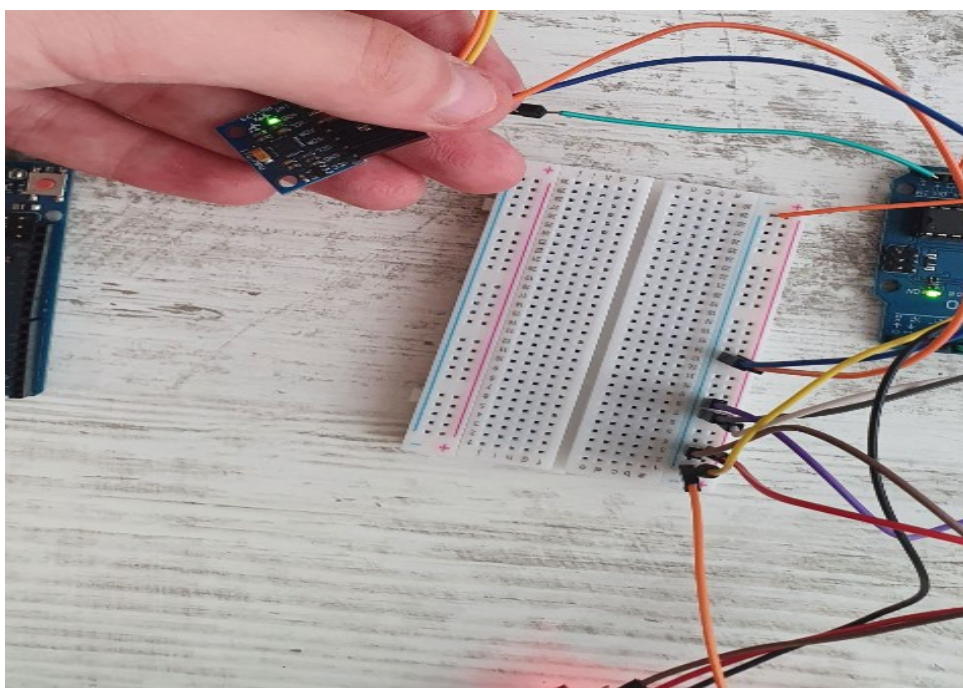


Figura 3.7.Implementarea modului MPU6050 în sistem

În Figura 3.7 modulul a fost conectat la plăcuță, care la rândul ei este conectată și alimentată de laptop.

3.4. Sistemul de avertizare

Pentru acest sistem am avut nevoie de:

- un laptop;
- o plăcuță Plusivo;
- un modul buzzer pasiv.

Am reprezentat în o schemă bloc pentru a putea înțelege mai bine cum am reușit integrarea modului în sistem.



Figura 3.8.Schema bloc a sistemului de avertizare

Sistemul conține un modul buzzer pasiv care, atunci când orice alt sistem din proiect se declanșează, acesta va porni și va face un zgomot care să atragă atenția oamenilor din jur și a proprietarului, dar și să sperie persoana ce a provocat declanșarea.

În momentul în care acest modul detectează o declanșare a altui sistem va porni sunetul, afișând în același timp pe ecran un mesaj de tipul „Alarma sonoră a fost activată!”

Concluzii

Am reușit în acest proiect pentru lucrarea de diplomă să implementez un sistem de alarmă pentru un autoturism, cu ajutorul cunoștințelor acumulate până acum, dar și după o documentare despre acest subiect, pe care am încercat să o sintetizez în capitolele anterioare.

Lucrarea de licență redă o implementare a unor componente hardware în vederea creării unui sistem funcțional, care poate fi dezvoltat ulterior pentru a reprezenta cerințele noi ale clienților și ale pieței care se află în continuă dezvoltare. S-a creat astfel, prin îmbinarea resurselor hardware cu puțin software o alarmă de mașină care poate fi folosită în viața de zi cu zi și care poate fi totodată implementată de către oricine dorește sau are curiozitatea de a încerca. Pot spune că rezultatul final s-a transpus într-un proiect care îndeplinește obiectivele pe care mi le-am propus inițial, atât pe partea de creare a proiectului, cât și pe partea de experiență și dezvoltare personală.

În implementarea sistemului m-am întâlnit cu câteva dificultăți pe care am reușit să le rezolv până la final. Deși inițial îmi propusesem ca proiectul să fie dezvoltat pe o plăcuță RaspberryPi, m-am răzgândit ulterior și am ales plăcuța Arduino Uno pentru a vedea diferențele dintre cele două având în vedere faptul că nu mai lucrasem până acum cu Arduino. Deși în proiectele anterioare am avut de implementat senzori, cei folosiți în lucrarea actuală au reprezentat o mică provocare pentru mine din cauza faptului că nu mai lucrasem cu aceștia, dar nu pot spune că mi-au provocat o dificultate. Una dintre adevăratele provocări a fost conectarea senzorilor pentru a funcționa simultan, lucru pe care l-am rezolvat după o documentare amănunțită în vederea creării unui cod care să permită acest lucru.

În această documentație am încercat să descriu cât mai elaborat modul de implementare a alarmei, documentația pe care am studiat-o în vederea ducerii la bun sfârșit a acesteia pentru a putea ca pe viitor să se aducă îmbunătățiri sistemului. Unul dintre cele mai importante lucruri pentru o dezvoltare viitoare a proiectului este implementarea unui modul GSM.

GSM vine de la Global System for Mobile Communication și este o tehnologie digitală folosită pentru a trimite mesaje sau pentru a suna utilizatorul. Pentru a conecta acesta la plăcuța Arduino Uno ne-ar mai trebui și un Arduino Shield și pentru a funcționa trebuie să avem o cartelă funcțională.

Având în vedere faptul că suntem obișnuiți să auzim alarmele sunând zilnic la diferite mașini, este posibil să nu verificăm tot timpul dacă mașina personală este sau nu afectată în vreun fel. Pentru a ușura munca și a putea fi siguri de ceea ce se întâmplă atunci când autoturism nostru rămâne nesupravegheat într-un loc public ar trebui implementat un modul GSM, pe lângă sistemul nostru de alarmă sonor care sună în mașină care să trimită un mesaj pe telefonul personal atunci când un senzor se declanșează și trimite un semnal către alarmă. Astfel am ști exact ce s-a întâmplat și când a avut loc un accident în care este implicat autoturismul nostru personal.

Tot pentru o direcție viitoare mi s-ar părea important să se implementeze un modul de presiune și greutate pentru a putea, în caz că cineva trece de toate sistemele noastre de alarmă de până acum să înregistrăm atunci când această persoană se așează pe un scaun. Acest modul ar trebui implementat pe toate scaunele.

În același timp, ne-ar mai trebui o cameră foto sau video pentru ca atunci când sistemul de alarmă este activat, aceasta să înregistreze chipul infractorului pentru o urmărire penală în continuare. Am putea lua și exemplul companiei Tesla, ce a implementat un sistem de multe camere video care atunci când cineva se află la o anumită distanță de mașină să pornească toate camerele și să îl înregistreze, salvând toate informațiile atât în computerul de bord al mașinii, cât

și într-un cloud privat, la care doar proprietarul mașinii are acces.

Tot ca o direcție viitoare clientul ar putea cere o aplicație pentru telefon prin care să poate urmări în direct camerele video montate pe mașină, sau să pornească și să oprească sistemul după bunul plac, din propria locuință.

Rezultatul final al lucrării este o alarmă de mașină care îndeplinește obiectivele formulate inițial:

- detectarea sunetului în caz că se sparge un geam;
 - detectarea mișcării în interiorul mașinii;
 - detectarea mișcării fizice a mașinii, prin măsurarea accelerației.
-

Bibliografie

- [1] Autor necunoscut, „*Systems and processes in instructional system design*” [Online], Disponibil la adresa: <http://www.nwlink.com/~donclark/hrd/isd/system.html>, Accesat: 2022.
- [2] Hommer Zhao, „*Alarm circuit: the construction and working principle*” [Online], Disponibil la adresa: <https://www.wellpcb.com/alarm-circuit.html>, Accesat: 2022.
- [3] Mihai Voican, „*Informatii generale despre sistemele de alarma*” [Online], Disponibil la adresa: <https://www.money.ro/informatii-generale-despre-sistemele-de-alarma/>, Accesat: 2022.
- [4] Autor necunoscut, „*What is Arduino Uno?*” [Online], Disponibil la adresa: <https://robu.in/what-is-arduino-uno/>, Accesat: 2022.
- [5] Ben Lutkevich, „*Microcontroller*” [Online], Disponibil la adresa: <https://www.techtarget.com/iotagenda/definition/microcontroller>, Accesat: 2022.
- [6] prernaajitgupta, „*I2C communication protocol*” [Online], Disponibil la adresa: [https://www.geeksforgeeks.org/i2c-communication-protocol/#:~:text=I2C%20stands%20for%20Inter-Integrated,Two%20Wired%20Interface\(TWI\).](https://www.geeksforgeeks.org/i2c-communication-protocol/#:~:text=I2C%20stands%20for%20Inter-Integrated,Two%20Wired%20Interface(TWI).,), Accesat: 2022.
- [7] Autor necunoscut, „*How to use a breadboard for electronics and circuits*” [Online], Disponibil la adresa: <https://www.sciencebuddies.org/science-fair-projects/references/how-to-use-a-breadboard>, Accesat: 2022.
- [8] Autor necunoscut, „*Arduino UNO R3*” [Online], Disponibil la adresa: <https://docs.arduino.cc/hardware/uno-rev3>, Accesat: 2022.
- [9] Tom Joseph, „*Arduino Uno specification*” [Online], Disponibil la adresa: <https://www.tomsonelectronics.com/blogs/news/arduino-uno-specification>, Accesat: 2022.
- [10] Autor necunoscut, „*Arduino Pin Configuration - Complete Guide 2021*” [Online], Disponibil la adresa: <https://robu.in/arduino-pin-configuration/>, Accesat: 2022.
- [11] Mohammadreza Akbari, „*Interfacing US-100 ultrasonic distance sensor with Arduino*” [Online], Disponibil la adresa: <https://electropeak.com/learn/interfacing-us-100-ultrasonic-distance-sensor-with-arduino/>, Accesat: 2022.
- [12] „*US-100 ultrasonic sensor module*”, Cleste.ro, România.
- [13] Autor necunoscut, „*Modul accelerometru și giroscop MPU6050*” [Online], Disponibil la adresa: <https://cleste.ro/modul-accelerometru-i-giroscop-mpu6050.html>, Accesat: 2022.
- [14] Mohammadreza Akbari, „*Interfacing GY-521 MPU6050 3-axis accelerometer gyroscope with Arduino*” [Online], Disponibil la adresa: <https://electropeak.com/learn/interfacing-gy-521-mpu6050-3-axis-accelerometer-gyroscope-with-arduino/>, Accesat: 2022.
- [15] Mohammad Damirchi, „*Interfacing KY-037 sound sensor with Arduino*” [Online], Disponibil la adresa: <https://electropeak.com/learn/interfacing-ky-037-sound-sensor-with-arduino/>, Accesat: 2022.
- [16] „*KY-037 Microphone sensor module (high sensitivity)*”, joy-it, România.
- [17] Mohammad Damirchi, „*Interfacing HX711 load cell amplifier module with Arduino*” [Online], Disponibil la adresa: <https://electropeak.com/learn/interfacing-hx711-load-cell-amplifier-module-with-arduino/>, Accesat: 2022.
- [18] Autor necunoscut, „*Modul cu senzor presiune greutate*” [Online], Disponibil la adresa: <https://cleste.ro/modul-cu-senzor-presiune-greutate.html>, Accesat: 2022.
- [19] Autor necunoscut, „*Modul buzzer pasiv*” [Online], Disponibil la adresa: <https://cleste.ro/modul-buzzer-pasiv.html>, Accesat: 2022.

Anexe.



Figura 1: Fața mașinii

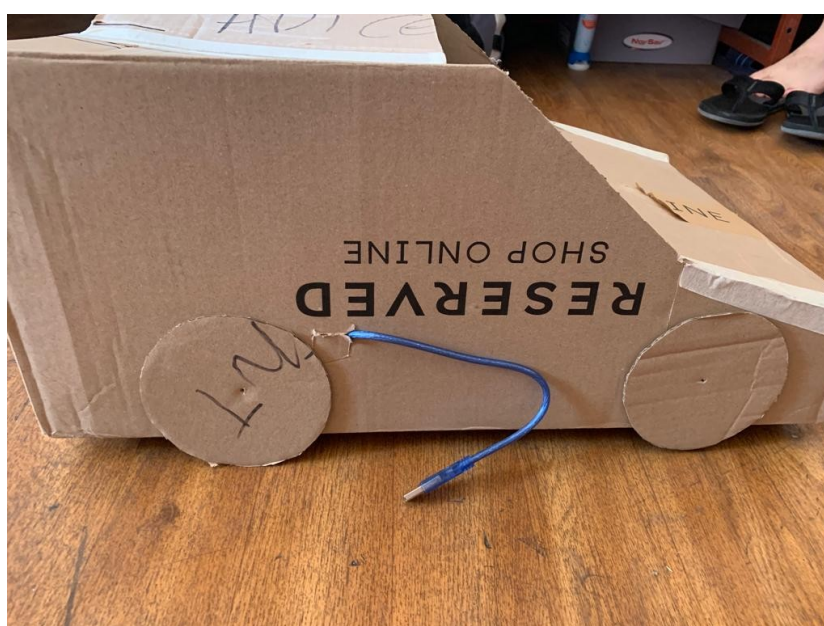


Figura 2: Lateralul mașinii



Figura 3: Spatele mașinii

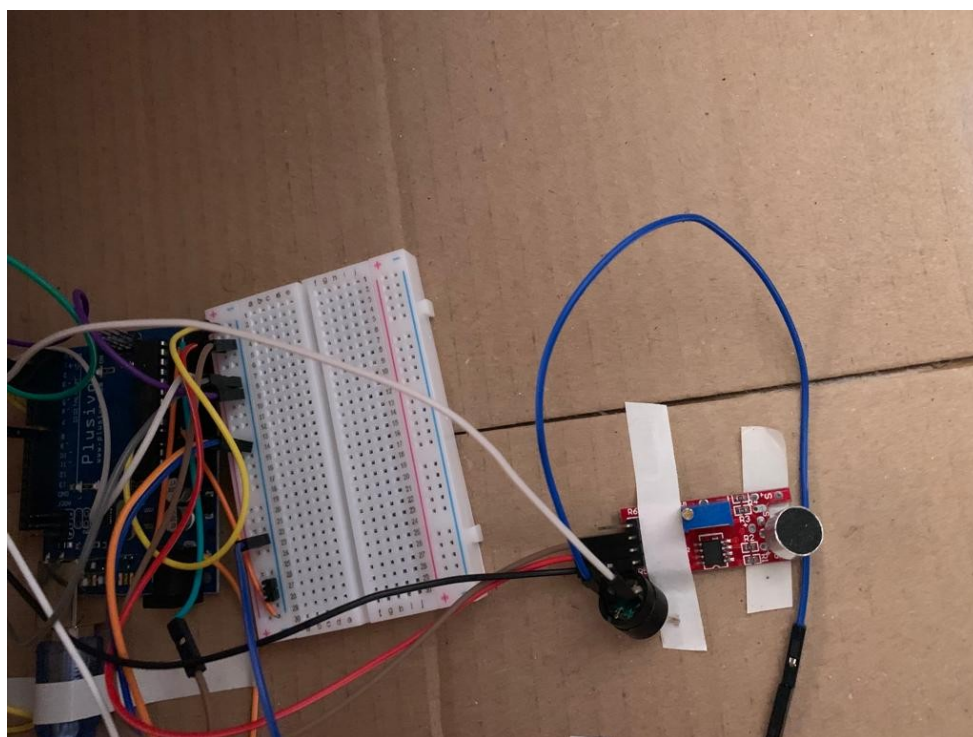


Figura 4: Modulul KY-037 încorporat în sistem

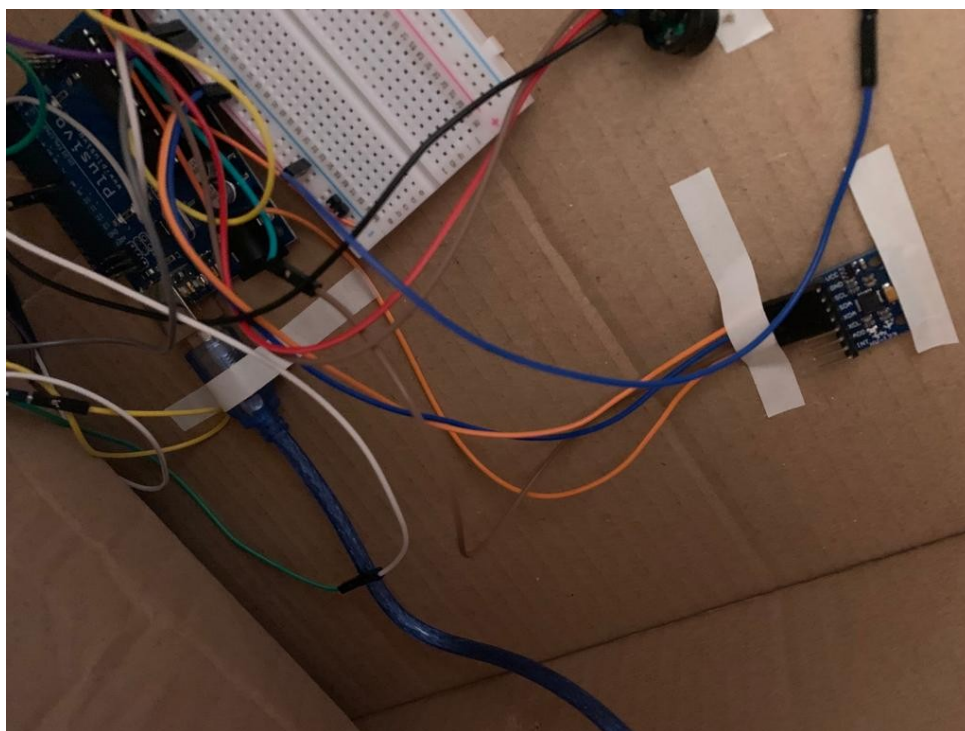


Figura 5: Modulul MPU6050 încorporat în sistem

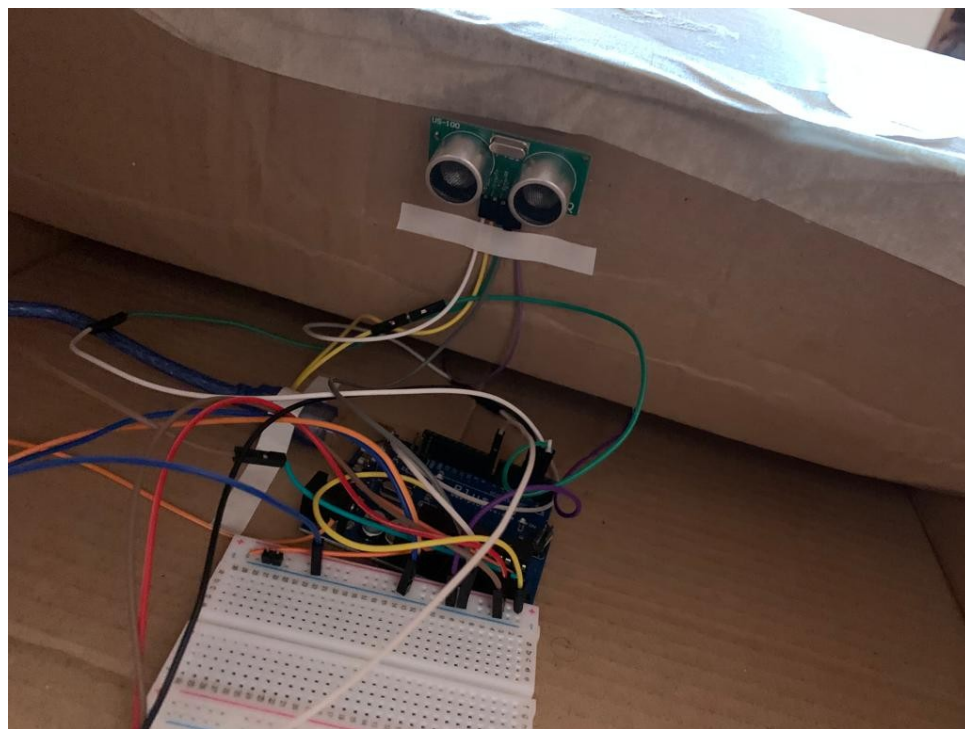


Figura 6: Senzorul US-100 încorporat în sistem

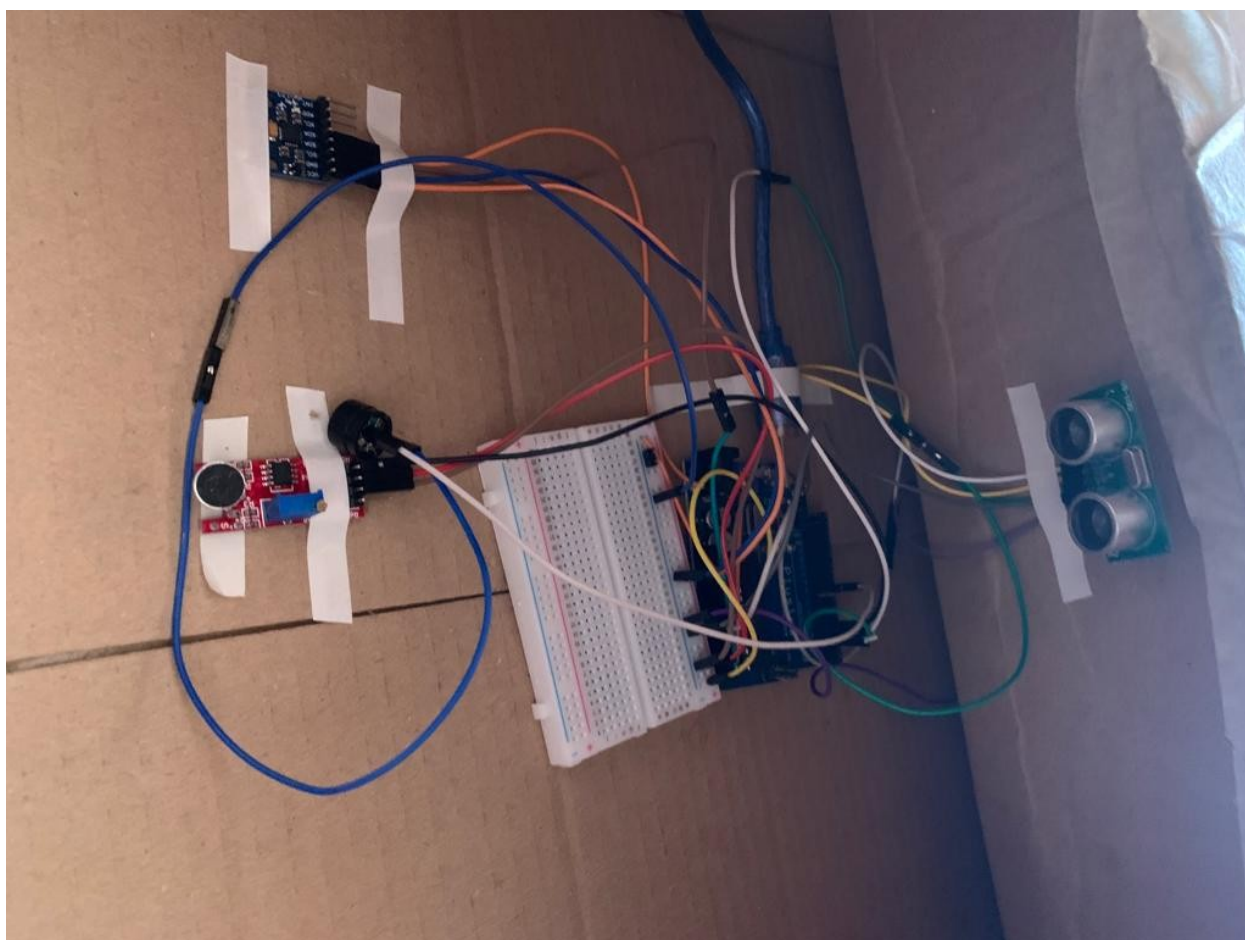


Figura 7: Privire în ansamblu a sistemului