**Pimung**

1. **Informații generice despre proiect**

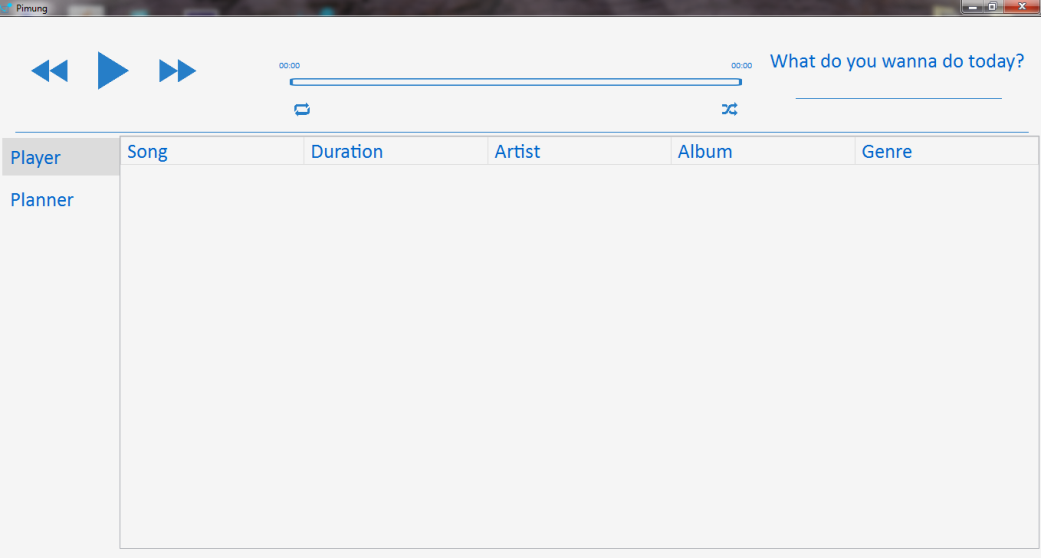
Pimung este un player muzical şi un planificator al activităţilor zilnice. Acum poţi controla muzica prin bătăi în masă (ciocăniri). Mai mult, ai opţiunea de a trimite muzica prin reţeaua locală la o placuţă Raspberry Pi la care sunt conectate boxele. Astfel poţi asculta melodiile la boxele sau la sistemul audio din colţul camerii în timp ce stai în pat cu laptop-ul. La Raspberry Pi mai este conectată şi o matrice de leduri pe care se arată frecvenţele melodiei în timp real.

Pimung te şi ajută să fii mai productiv. Introduci care este scopul tău principal pentru ziua respectivă, iar el îţi reaminteşte mereu in acea zi ce trebuie să faci. Apăsând pe butonul ,,Planner” îţi poţi alcătui un to do list offline. Dacă cumva nu eşti destul de motivat pentru a face ceea ce ţi-ai propus, apasă pe butonul de sub întrebarea ,*Need some motivation?* şi vei citi un citat motivaţional.

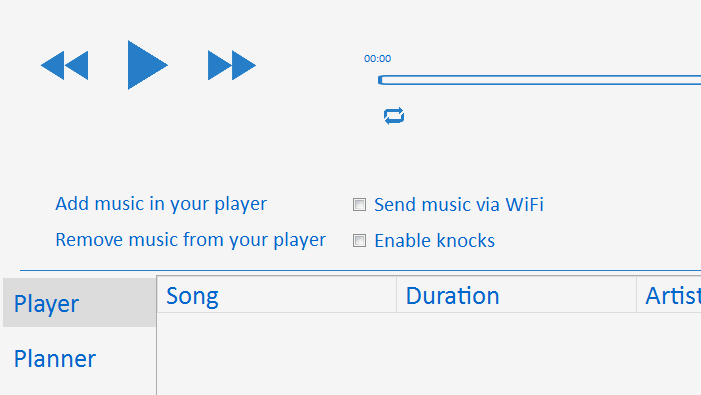
1. **Prezentarea aplicaţiei**

Proiectul este alcătuit din trei părţi: o aplicaţie pentru Windows, un circuit făcut cu placa Arduino Uno şi un server pe Raspberry Pi. La Raspberry mai e conectată şi o matrice de leduri.

Când **aplicaţia pentru Windows** este deschisă pentru prima dată, ea arată în felul următor:



Pentru a adăuga muzică în librărie, se deschide meniul aplicației. Pentru aceasta se mută cursorul deasupra liniei albastre ce desparte butoanele de play, forward şi backward de cuvintele Song, Duration, Artist etc şi se trage în jos astfel:

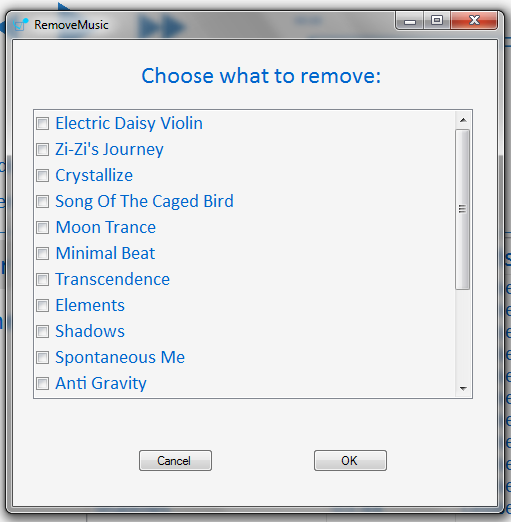


Se apasă pe ,,Add music in your player” pentru a adăuga melodii. Sunt acceptate numai fişiere cu extensia .mp3 sau .wav. După adăugare se trage înapoi de linia albastră pentru a ascunde meniul. Apăsând pe cuvintele ,,Song”, ,,Duration“ etc se pot ordona melodiile adăugate în ordine alfabetică, după album, gen muzical, durată sau artist.



Pentru a asculta o melodie se dă dublu-click pe ea. Butoanele de *forward*, *backward*, *replay* şi *shuffle* sunt active doar în timp ce muzica este redată. Când o melodie se termină iar butoanele de *shuffle* sau *replay* nu sunt selectate, începe următoarea melodie.

Pentru a scoate melodii din librărie se apasă pe butonul *Remove music from your player*. După apăsare se deschide un dialog în care se selectează melodiile ce se vor scoase din librărie.

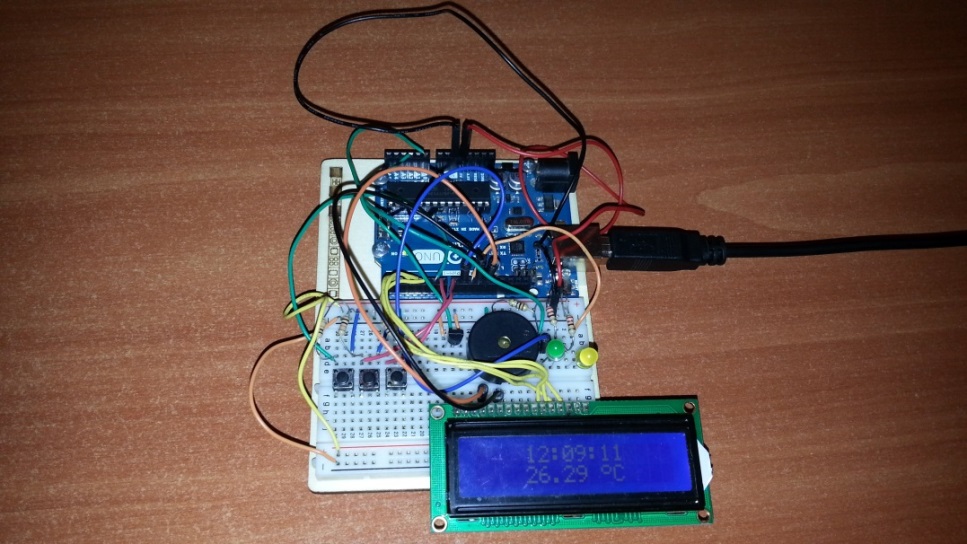


În partea dreapta sus este întrebarea: ,,What do you wanna do today?”. Pentru a răspunde se dă click pe linia de sub întrebare. Răspunsul nu trebuie să conţină mai mult de 23 de caractere. După ce răspunsul este scris se apasă ENTER. Acum răspunsul are următorul format: ,,Today, <<TEXT INTRODUS>>!”. Pentru a schimba textul introdus se apasă pe X-ul din dreapta răspunsului. Acesta este reţinut doar pentru ziua respectivă, iar dacă aplicaţia este deschisă în altă zi, întrebarea va apărea din nou.

Dacă se dă click pe butonul *Planner* ce se află în meniul din stânga, tabelul cu melodii va dispărea şi se va afişa o listă de sarcini şi întrebarea *Need some motivation?*. Dacă se apasă pe butonul de sub întrebare pe care scrie *Press me!!,* va apărea un citat motivaţional. Dacă se apasă încă o dată, citatul se va schimba.

Pentru a adăuga sarcini de lucru în lista de sarcini se scrie sarcina, apoi se apasă ENTER sau se dă click pe butonul de lângă. Pentru a şterge sarcinile rezolvate se apasă pe butonul *Delete checked items*.

**Circuitul făcut cu placa Arduino:**



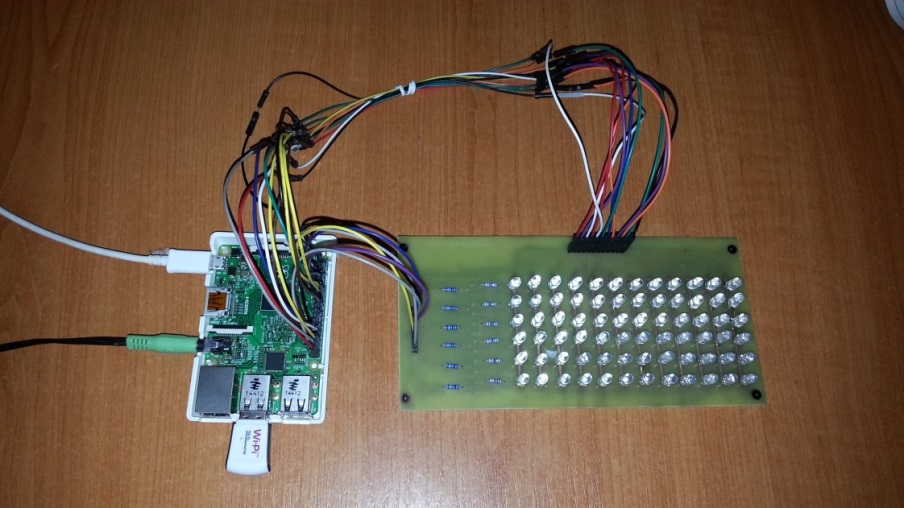
Acesta conţine un ecran LCD pe care sunt afişate ceasul şi temperatura.

Primul buton din stănga se numeşte buton de control, al doilea buton de ore şi al treilea de minute. Ţinând apăsat pe butonul de control şi pe butonul de ore ceasul va arăta cu o oră mai mult. Ţinând apăsat pe butonul de control şi pe butonul de minute ceasul va arăta cu un minut mai mult. Astfel se reglează ceasul.

Pentru a face legătura dintre placa Arduino Uno şi aplicaţia pentru Windows, in timp ce se ţin apăsate butoanele de ore si minute, se bifează căsuţa ,,Enable Knocks” din meniul aplicaţiei pentru Windows. După apăsare va apărea o căsuţa de dialog în aplicaţie care va indica dacă s-a reușit comunicarea serială dintre Arduino și aplicația pentru Windows.

Placa Arduino detectează vibrațiile și în funcție de aceastea trimite diferite comenzi calculatorului. Când Arduino detectează 2 vibrații consecutive el trimite calculatorului comanda de play/stop, dacă detectează 3 vibrații trimite comanda de forward iar pentru 3 vibrații consecutive trimite comanda de backward.

**Plăcuța Raspberry Pi împreuna cu matricea de leduri:**



Pentru a rula serverul trebuie mai întâi să fii logat. Apoi se executată comanda ,,sudo ./server”. Acum serverul incepe sa ruleze.

Pentru a te conecta la server trebuie din meniul aplicației bifată căsuța ,,Send music via WiFi”. După ce se bifează, se deschide un dialog în care trebuie introdusă adresa IP a server-ului. După ce se introduce adresa IP utilizatorul va fi anunțat dacă conexiunea a reușit sau nu.

Acum serverul se utilizează în mod normal dar muzica este trimisă la Raspberry Pi și nu se mai aude în difuzoarele laptop-ului sau calculatorului. Pe matricea de leduri apare spectograma melodiei folosind fast fourier transform.

1. **Proiectarea arhitecturală**

În realizarea aplicației pentru Windows am folosit limbajul de programare C# și am lucrat cu Visual Studio 2013. Pentru redarea muzicii am folosit Windows Media Player Library.

Programul pentru placa Arduino Uno a fost scris în Arduino IDE. Circuitul contine: 3 butoane, un piezo (pentru detectarea vibrațiilor), un senzor de temperatură TMP36, un ecran LCD și două leduri.

Serverul de pe Raspberry Pi 2 a fost realizat în limbajul C. Pentru redarea muzicii am folosit ALSA (Advanced Linux Sound Architecture). Matricea de leduri conține 72 de leduri, 6 tranzistori si 12 rezistețe.

1. **De ce am ales aceste tehnologii?**

* Aplicația pentru Windows am ales să o fac în **C#**, pentru că C#-ul e un limbaj de programare care oferă foarte multe posibilități, se lucrează ușor cu el, are o comunitate mare în spate, IDE-ul Visual Studio oferă o interfață ușor de folosit și foarte puternică în același timp și pentru că poți face multe cu puțin.
* Am ales să folosesc **Arduino** Uno pentru că este ușor de remodificat circuitul, este ieftin, limbajul de programare nu este complicat, are pini analogici și este popular.
* Am ales **Raspberry** Pi 2 pentru server și pentru matricea de leduri pentru că are putere mare de procesare, se conectează ușor la internet și pentru că are mulți pini GPIO.

1. **Ghid de instalare**

Se deschide folder-ul ce conține fișierele de instalat. Se dă dublu-click pe setup.exe. Va apărea un dialog și se apasă pe butonul *Install*. Aplicația este instalată. Pentru a rula server-ul trebuie să te loghezi pe Raspberry Pi. Apoi se intră în folder-ul *server* prin comanda *cd server* și se scrie comanda *make run*. Acum server-ul funcționază. Placa Arduino are nevoie doar de curent pentru a funcționa.

1. **Elemente distinctive ale aplicației Pimung**

* Niciun alt player muzical nu mai oferă posibilitatea de a schimba sau opri muzica prin intermediul vibrațiilor.
* Spre deosebire de alte aplicații, cu Pimung poți trimite muzica prin rețea, astfel încât nu mai trebuie să ții boxele conectate mereu la calculator.
* Aplicația permite aflarea temperaturii din cameră cu ușurință.

1. **Resurse externe**

* În fișierul *server.c* am folosit funcția *void* ***four1****(double data[], int nn, int isign)* pe care am luat-o de pe [*www.nr.com*](http://www.nr.com)dar licența nu este specificată.
* Pentru trimiterea muzicii prin rețeaua locală printr-un stream am folosit librăria open source **Naudio.**
* Pentru redarea muzicii pe Raspberry am folosit **ALSA** (Advanced Linux Sound Architecture). Pentru a o folosi am urmat tutorialul: *http://equalarea.com/paul/alsa-audio.html*
* Pentru redarea muzicii în player am folosit **Windows Media Player Library.**
* Schema și design-ul matricii de leduri au fost făcute de mine, dar cablajul a fost realizat de un inginer electronist. Toate piesele de pe clabaj au fost lipite de mine.

1. **Opinie personală despre proiect**

Proiectul mi-a permis să îmi lărgesc cunoștiințele de programare și l-am privit ca pe o provocare. Nu intenționez deocamdată să îl transform într-un proiect comercial. L-am făcut pentru mine, pentru a asculta muzică întru-un mod diferit dar se adresează tuturor persoanelor cărora le place muzica și își doresc o modalitate mai ușoară de a o controla, dar în același timp își doresc să fie productive.