



## **Středoškolská technika 2019**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

### **LED Matrix**

**Jakub Marek**

SPŠE V Úžlabině  
V Úžlabině 320, Praha 10

## Úvod

Tento projekt se zrodil z nápadu propagace naší školy pomocí svítícího loga, což splňovalo zařízení, které jsem si chtěl vyrobit už dlouho – LED Matrix. Toto zařízení se skládá ze spousty LED diod, které jsou naskládány do 2D sítě, ty se pak pomocí mikrokontroleru rozsvěčují na požadovanou barvu.



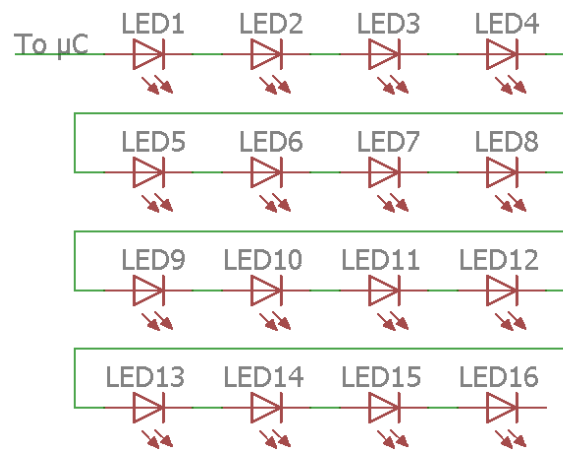
*Obrázek 1 - LED Matrix*

## Použité technologie

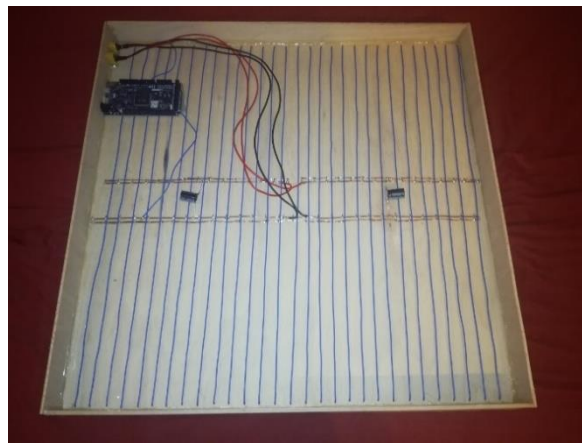
Hlavní řídicí jednotkou displeje je ESP32, ke kterému jsou připojeny programovatelné LED pásky (WS2812B), rozstříhané na pruhy po 30 diodách, viz obrázek 1. Tyto pruhy jsou nalepeny na desku a následně propojeny dráty (napájení je přivedeno ke každému pruhu zvlášť a datový drát vede z konce pruhu na začátek vedlejšího, viz obrázky 2 až 4).

ESP32 již v základu obsahuje řadu modulů, například WiFi nebo Hallův senzor, to je ještě rozšířeno o externí moduly (SD karta, RTC a mikrofon se zesilovačem), které umožňují, aby zařízení mohlo být co nejvíce interaktivní.

Toto vše je pak ovládáno přes wifi terminál z mobilu, nebo z počítače.



Obrázek 2 - Zapojení LED



Obrázek 3 - Vnitřní uspořádání



Obrázek 4 - Adresovatelné RGB LED pásky

## Program

Celý program, který vše ovládá, má okolo 2500 řádků, což vypovídá o tom, že toho displej umí celkem dost. Program obsahuje hlavní smyčku, která slouží jako takové menu, v této smyčce se na displeji zobrazují různé animace (takový spořič obrazovky), přičemž se průběžně kontroluje, zda nějaký uživatel neposlal přes WiFi data. Pokud tomu tak je, dojde k přečtení dat, a pokud tento příkaz zařízení zná, tak přeskočí v programu do jiné sekce, které probíhá, dokud se nevrátíme zpět do menu, viz obrázky 5 a 6 (5 – čtení dat, 6 – rozpoznání příkazu).

```
754 void WiFiRead(){
755     WiFiClient newClient = server.available();
756
757     if (newClient) {
758         for (int i = 0; i < MAX_CLIENTS_NUM; i++) {
759             if (NULL == clients[i]) {
760                 clients[i] = new WiFiClient(newClient);
761                 break;
762             }
763         }
764     }
765
766     for (int i = 0; i < MAX_CLIENTS_NUM; i++) {
767         String stringData = "";
768         if (NULL != clients[i]) {
769             if (clients[i]->available()){
770                 while (clients[i]->available()){
771                     char c = clients[i]->read();
772                     if (clients[i]->available() == 0){
773                         stringData += char(c);
774                         c = '\n';
775                     }
776                     if (c == '\n') {
777                         if (stringData.indexOf("GET") == 0){
778                             WiFiSendSide(i, stringData);
779                         } else {
780                             WiFiApplyCommand(i, stringData);
781                         }
782                         stringData = "";
783                     } else {
784                         stringData += char(c);
785                     }
786                 }
787             } else if (!clients[i]->connected()){
788                 clients[i] = NULL;
789             }
790         }
791     }
792 }
793
```

Obrázek 5 - Čtení dat

```
254 void WiFiApplyCommand(int idUser, String command){
255     if (command.indexOf("/back") == 0 && g_mode != 0){
256         g_mode = 0;
257         displayClear(500);
258     } else if (command.indexOf("/help") == 0){
259         // /help <side_id>
260         int side = (command.length() >= 7) ? command.substring(6, command.length()).toInt() : 1;
261         if (side > (int)ceil(COMMANDS_COUNT / 7.0)){
262             side = 1;
263         }
264         clients[idUser]->print("Commands list ");
265         clients[idUser]->print(side);
266         clients[idUser]->print(" of ");
267         clients[idUser]->println((int)ceil(COMMANDS_COUNT / 7.0));
268         for (int i = 0; i < 7; i++){
269             if (((side - 1) * 7) + i < COMMANDS_COUNT){
270                 clients[idUser]->print(g_commands[ ((side - 1) * 7) + i][1]);
271                 clients[idUser]->print(" - ");
272                 clients[idUser]->println(g_commands[ ((side - 1) * 7) + i][2]);
273             }
274         }
275     } else if (command.indexOf("/game ") == 0){
276         // /game <game_id>
277         if (g_mode == 0){
278             int dataMode = command.substring(6, command.length()).toInt();
279             if (dataMode == 1){
280                 displayClear(500);
281                 g_mode = 100;
282                 sq_snakeGame();
283             } else if (dataMode == 2){
284                 displayClear(500);
285                 g_mode = 101;
286                 ttt_ticTacToe();
287             } else if (dataMode == 3){
288                 displayClear(500);
289                 g_mode = 102;
290                 gol_gameOfLife(true);
291             } else if (dataMode == 4){
292                 displayClear(500);
293                 g_mode = 103;

```

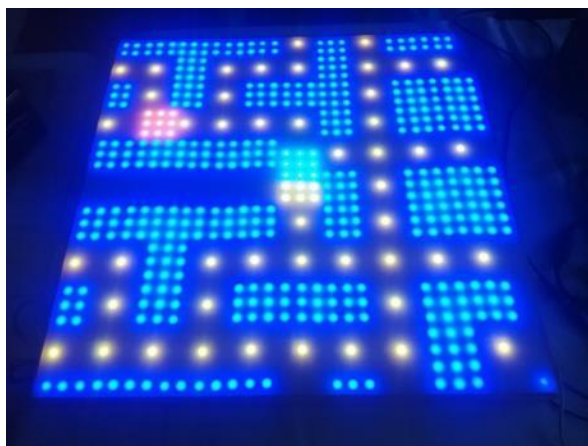
Obrázek 6 - Rozpoznání příkazu

## Použití

Displej má naprogramovaných několik módů – hry, hodiny, stopky, časovač, spektrální analyzátor, galerii a další...

Takže si můžete zavzpomínat na dobu snejka, zkusit utéct před duchy a posbírat všechny puntíky, přeskočit co nejvíce kaktusů nebo proskakovat mezi trubkami, viz obrázky 7 až 11 (reference na snake, pac-man, T-rex z chromu a flappy bird).

Toto zařízení slouží hlavně k ukázce, co vše se dá s LED diodami a mikrokontrolérem vytvořit.



Obrázek 7 - ukázka zobrazení – PacMan



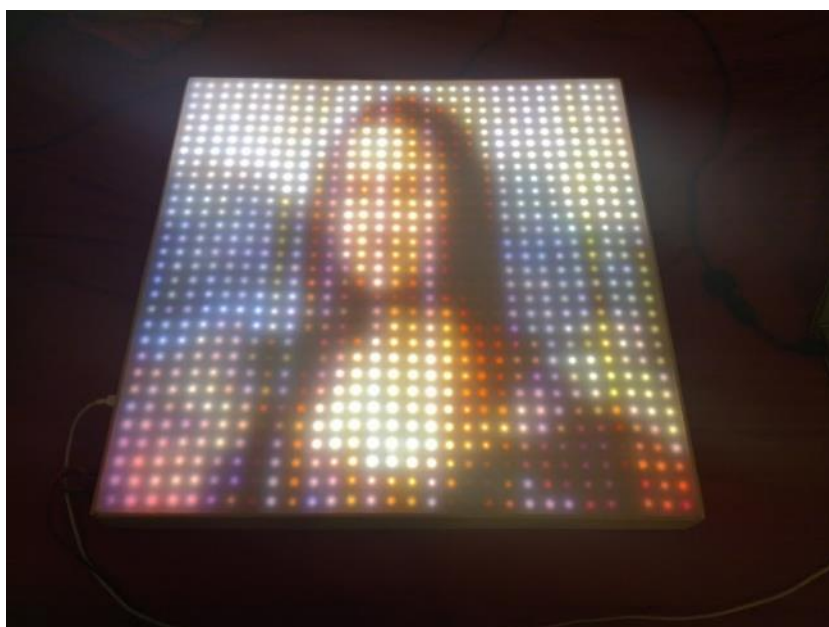
Obrázek 8 - Ukázka zobrazení – Flappy Bird



Obrázek 9 - Ukázka zobrazení - T-Rex



*Obrázek 10 - Ukázka zobrazení - LOGO SPŠE V Úžlabině*



*Obrázek 11 - Ukázka zobrazení - Mona Lisa*

## Závěr

Ač jsem se při konstrukci tohoto zařízení docela nadřel a musel projít nemálo nástrahami, tak tajně doufám, že některé lidi tento projekt osloví a inspiruje je k tvorbě podobných věcí.