

**Технологично училище  
“Електронни системи”**

**Увод във вградените микрокомпютърни  
системи**



**Безжичен MIDI Bluetooth контролер**

**Изготвил: Александър Димитров, 10г**

# 1. Описание на проекта

## Основна идея:

Проектът представлява MIDI контролер, който комуникира безжично чрез Bluetooth MIDI протокол с компютър, таблет или смартфон. Контролерът ще има няколко бутона, потенциометри (slider and knob), които ще изпращат MIDI сигнали към музикален софтуер като Ableton Live, FL Studio, GarageBand и др.

## Целева аудитория:

- Музиканти и продуценти, търсещи мобилни и безжични решения •
- Хоби ентусиасти, които искат да изградят собствен MIDI контролер •
- DJ-и, които искат персонализиран интерфейс

# 2. Функционалност

## Основни функции:

- Изпращане на MIDI съобщения (Note On/Off, Control Change) чрез Bluetooth
  - Note On/Off - за всеки бутон
  - Control Change - за потенциометрите
- Множество бутона (за стартиране на лупове и/или барабани)
- Потенциометри за контрол на параметри (filter cutoff, сила на звука, ефекти и др.)
- LED индикация за статус на свързване
- Захранване чрез батерия или USB

## Решавани проблеми / нужди:

- Избяване от кабели при използване на MIDI контролери
- Персонализиран и достъпен контролер
- Компактно и мобилно решение за живи изпълнения

# 3. Използван хардуер

## Компоненти:

- **ESP32 DevKit v1** – за Bluetooth и микроконтрол
- **Бутони** – за контрол на ноти
- **Потенциометри (10k)** – за контрол на параметри

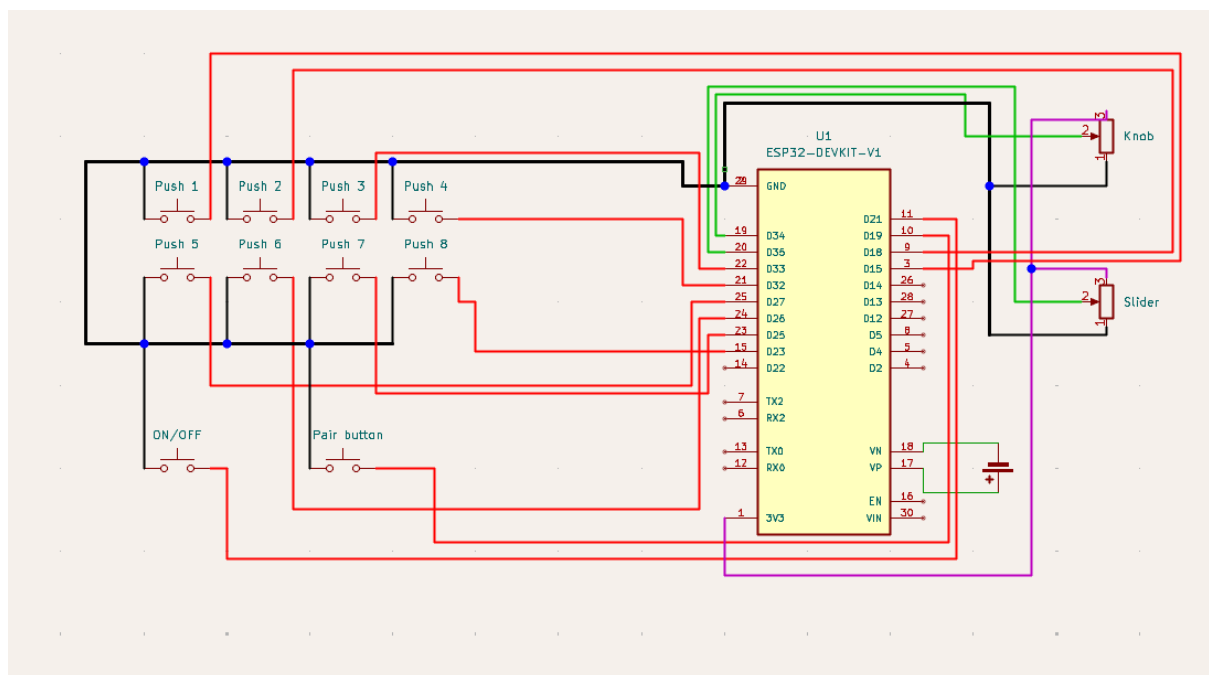
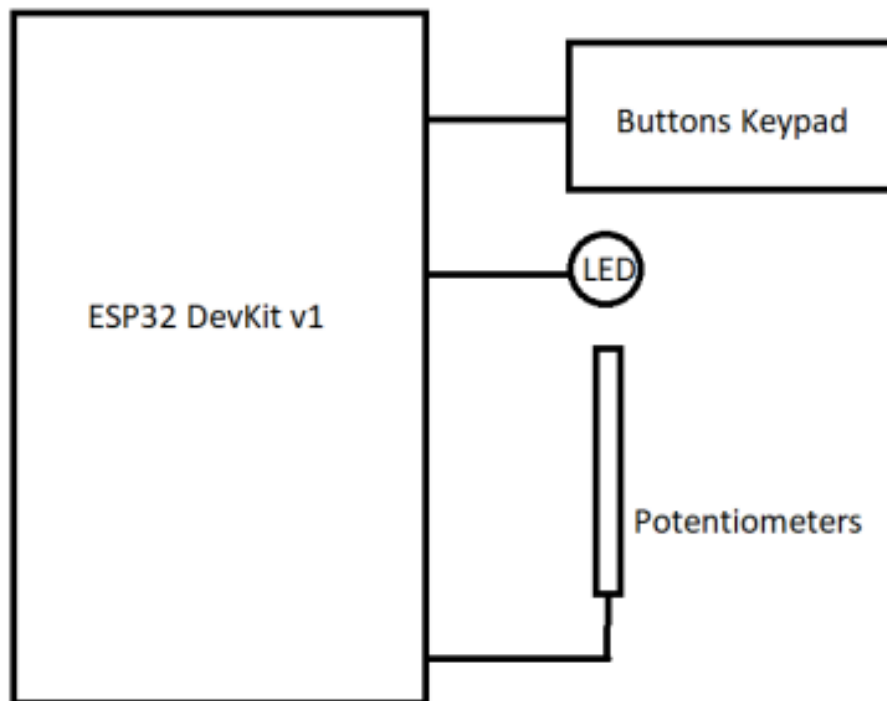
- Съпротивления, кабели, прототипна платка
- Батерия Li-ion 3.7V + заряден модул
- LED диоди – индикация на статус

Списък с материали и доставчици:

Компонент	Количество	Доставчик
ESP32 DevKit v1	1	Ardboard
Бутони	8	Elimex, AliExpress
Потенциометри (10k)	2 (слайдер, врътка)	Elimex
Прототипна платка	1	Elimex
Батерия + зарядно	1	Elimex / Comet
Кабели, резистори	много	Comet

## 4. Схематика

- Бутоните и потенциометрите са свързани към GPIO пиновете на ESP32
- Bluetooth MIDI се реализира чрез софтуерна библиотека за ESP32
- LED показва статуса на връзката или активност



## 5. Сопс код (screenshot-и):

```
1  #include <BLEMIDI_Transport.h>
2  #include <hardware/BLEMIDI_ESP32.h>
3
4  BLEMIDI_CREATE_DEFAULT_INSTANCE()
5
6  const int buttonPins[8] = {15, 18, 33, 32, 27, 26, 25, 23};
7  const int noteNumbers[8] = {60, 62, 64, 65, 67, 69, 71, 72};
8
9  const int potPinPitch = 34;
10 const int potPinVelocity = 35;
11
12 int lastButtonStates[8] = {HIGH, HIGH, HIGH, HIGH, HIGH, HIGH, HIGH, HIGH};
13
14 void setup() {
15     for (int i = 0; i < 8; i++) {
16         pinMode(buttonPins[i], INPUT_PULLUP);
17     }
18
19     pinMode(LED_BUILTIN, OUTPUT);
20     digitalWrite(LED_BUILTIN, LOW);
21
22     BLEMIDI.setHandleConnected(onConnected);
23     BLEMIDI.setHandleDisconnected(onDisconnected);
24
25     MIDI.begin();
26 }
27
```

```
28 void loop() {
29     int velocity = map(analogRead(potPinVelocity), 0, 4095, 0, 127);
30
31     //center (8192), max (16383)
32     int pitchValue = analogRead(potPinPitch);
33     int pitchBend = map(pitchValue, 0, 1300, 0, 8190);
34     MIDI.sendPitchBend(pitchBend, 1);
35
36     for (int i = 0; i < 8; i++) {
37         int currentState = digitalRead(buttonPins[i]);
38
39         if (lastButtonStates[i] == HIGH && currentState == LOW) {
40             // Button pressed
41             MIDI.sendNoteOn(noteNumbers[i], velocity, 1);
42         } else if (lastButtonStates[i] == LOW && currentState == HIGH) {
43             // Button released
44             MIDI.sendNoteOff(noteNumbers[i], 0, 1);
45         }
46
47         lastButtonStates[i] = currentState;
48     }
49
50     delay(5); //debounce
51 }
52
```

```
53 void onConnected() {
54     digitalWrite(LED_BUILTIN, HIGH);
55 }
56
57 void onDisconnected() {
58     digitalWrite(LED_BUILTIN, LOW);
59 }
60
```

## 6. Демо:

### Общ принцип:

Този MIDI контролер е изграден върху ESP32 и комуникира с музикален софтуер(Koala Sampler, SynprezFM) чрез BLE MIDI. Той има:

- 8 бутона, които изпращат различни MIDI ноти
- 2 потенциометъра, които контролират различни стойности на софтуера: velocity и pitch bend

### Хардуер:

- **Бутони:** Свързани към обща маса и пинове 15, 18, 33, 32, 27, 26, 25, 23. Когато се натисне бутон, ESP32 изпраща midi команда "Note On" към избраната midi нота. Когато се пусне се изпраща командата "Note Off".
- **Потенциометър 1 (knob):** Свързан към пин 34. Той работи като pitch bend. Когато е в долната си позиция, тонът е в неутрално положение. Когато се завърти нагоре, тонът се огъва към по-висока височина на тона.

- **Потенциометър 2 (slider):** Свързан към пин 35.

Той определя velocity - колко “силно” да звучи натиснатата нота.

## **BLE MIDI:**

Контролерът използва библиотеката BLEMIDI\_ESP32, за да комуникира с устройства през Bluetooth.