



Факултет Електроника и Автоматика  
Катедра Компютърни Системи и Технологии

Дипломна Работа на тема:  
“Проектиране на РСВ с използване на AVR/PIC асемблер”

Дипломант:  
Веселин Станчев  
ФН: 614872  
спец. КСТ

Научен Ръководител:  
д-рант инж. Стефан Лишев

Пловдив 2023 г.

# Съдържание

Списък с използвани термини.....	6
Списък на фигурите.....	6
Увод.....	7
Предпоставки за темата на дипломната работа.....	8
Проектирани AVR/PIC печатни платки (Приложение 1).....	8
Използван софтуер за проектиране: Fritzing.....	8
Реализирана дипломна работа за бакалавърска степен: Статична библиотека реализирана чрез GNU Assembler за ATtiny328P/RP2040.....	8
Асемблери според instruction set архитектурата.....	8
Цел на дипломната работа:.....	9
1. Обзор на съществуващи печатни платки с AVR/PIC базирани микроконтролери.....	10
I. Предметна област.....	11
II. Класификации на реализираните платки.....	11
IV. Резултати и обобщаване на резултатите.....	16
2. Изисквания към проекта.....	17
Вследствие на направения обзор на съществуващи печатни платки, и поставената цел на проекта, биват поставени следните изисквания:.....	17
Изисквания към проектирането на печатната платка:.....	17
Изискванията към проектирането на печатната платка ще бъдат изпълнени в шеста глава....	17
Възможности за реализация на платката.....	18
Възможни протоколи за използване.....	18
3. Анализ на микроконтролерите които ще бъдат използвани за печатната платка.....	19
Анализ на Attiny85.....	20
Анализ на PIC10F320.....	22
5. Софтуерни инструменти за проектиране на платката и програмиране.....	27
GNU MAKE.....	28
GIT.....	29
6. Проектиране на печатната платка.....	35
Хардуерни компоненти.....	36
7. Програмиране на микроконтролерите.....	38
Проектирани AVR/PIC печатни платки използвайки свободно -достъпни схеми и програма за проектиране Fritzing.....	39
1. Attiny 85 Traffic Lights.....	39

## Списък с използвани термини

Термин	Описание
ISA-Instruction Set Architecture	Сет от достъпни инструкции на процесора
RISC-Reduced Instruction Set	Редуциран сет от инструкции ,като по-сложните операции биват изпълнени от основните инструкции
GNU Assembler	Свободен асемблер част от GNU проекта. Наличен в GNU Binutils пакета
AVR Assembler	Свободен асемблер. Поддържа множество AVR RISC базирани микроконтролери
PIC Assembler	Асемблер за PIC базирани микроконтролери

## Списък на фигурите

Фигура	Описание
0.1	Показани са асемблерите за всеки вид архитектура (ISA)
3.1	Пинове на ATTiny85
3.2	Блокова схема на ATTiny85
3.3	Пинове на PIC10F320
3.4	Блокова схема на PIC10F320
6.1	Блокова схема на печатната платка
6.2	Символ ATTiny85
6.3	Footprint ATTiny85
6.4	Символ PIC10F320
6.5	Footprint PIC10F320
6.6	Схема на платката
6.7	Layout на платката

## Увод

С развитието на езиците от високо ниво като напр. C++, Python, Ruby се развиват и Assembler-ните езици като напр. NASM, GNU Assembler.

Assembler-ните езици са изключително важни при програмирането на:

- микропроцесори
- микро контролери
- вградени устройства (embedded devices) изградени на основата на микро контролерите
- операционни системи от тип Real Time.
- Софтуерно осигуряване в авиацията

За различните видове микроконтролери съществуват различни асемблери.

За AVR-базираните микроконтролери като напр.:

- ATtiny85
- Atmega 16 A
- rp2040

За AVR-базираните микроконтролери съществуват следните свободни асемблери:

- GNU Assembler
- AVR Assembler
- RISC-V Assembler

За PIC-базираните микроконтролери като напр.:

- pic16f84a
- pic16f877a
- pic10f320

За PIC-базираните микроконтролери съществуват следните асемблери:

- PIC Assembler

## Предпоставки за темата на дипломната работа

За предварителната подготовка при избор на темата на настоящата дипломна работа са налице следните предпоставки:

Проектирани AVR/PIC печатни платки (Приложение 1).

Използван софтуер за проектиране: Fritzing

Реализирана дипломна работа за бакалавърска степен: Статична библиотека реализирана чрез GNU Assembler за ATtiny328P/RP2040

## Асемблери според instruction set архитектурата

За различните видове instruction set architectures (ISA) съществуват различни асемблери. За x86\_64 архитектурата може да бъде използван Nerwide assembler. За ARM архитектурата може да бъде използван GNU Assembler. За PIC архитектурата може да бъде използван PIC Assembler.

Това е указано в таблицата по-долу:

Сет от инструкции (ISA)	Асемблер
x86_64	Nerwide assembler
ARM	GNU Assembler, AVR Assembler
RISC-V	RISC-V Assembler
PIC RISC	PIC Assembler

Фиг. 0.1

## **Цел на дипломната работа:**

Да бъде проектирана печатна платка (PCB) с използване на AVR-базиран микроконтролер и PIC-базиран микроконтролер с цел комуникация между двата микроконтролера на асемблерно ниво. Платката ще може да бъде използвана за учебни и университетски цели. Проектът е изцяло open-source с цел по-доброто бъдещо развитие.

## **Причини за изпълнение на поставената цел:**

Използване на два свободни асемблера  
Използване на микроконтролери с малки размери с цел преносимост на печатната платка  
Реализиране на софтуерната част за печатната платка като самостоятелна софтуерна библиотека  
намаляване на големината на elf файла чрез използване на асемблер  
Изцяло практическа насоченост на проекта

## **Вследствие на поставената цел трябва да бъдат изпълнени следните задачи:**

Обзор на съществуващи печатни платки с AVR/PIC базирани микроконтролери  
Изисквания към проекта  
Анализ на микроконтролерите които ще бъдат използвани за печатната платка  
Анализ на асемблерите които ще бъдат използвани  
Софтуерни инструменти за проектиране на платката и програмиране  
Проектиране на печатната платка  
Програмиране на микроконтролерите  
Реализиране на кода като библиотека  
Постигнати резултати. Бъдещо развитие

## **Глави**

### **Увод**

- I. Обзор на съществуващи печатни платки с AVR/PIC базирани микроконтролери
- II. Изисквания към проекта
- III. Анализ на микроконтролерите които ще бъдат използвани за печатната платка
- IV. Анализ на асемблерите които ще бъдат използвани
- V. Софтуерни инструменти за проектиране на платката и програмиране
- VI. Проектиране на печатната платка
- VII. Програмиране на микроконтролерите
- VIII. Реализиране на кода като библиотека
- IX. Постигнати резултати. Бъдещо развитие

### **Приложения**

### **Заклучение**

## **1. Обзор на съществуващи печатни платки с AVR/PIC базирани микроконтролери**

### **Въведение**

В настоящата глава ще бъдат разгледани печатни платки (PCB), базирани върху AVR/PIC микроконтролери.

Съдържанието на главата се състои от следните подглави:

**Предметна област**

**Класификации на реализираните платки**

**Анализ на платките**

**Резултати и обобщаване на резултатите**

**Заклучение**

**Източници**

За всяка печатна платка ще бъдат описани:

**обща характеристика на платката  
микроконтролерите които поддържа  
интерфейсите които поддържа  
асемблерите на които може да бъде програмирани микроконтролерите**

Печатните платки които ще бъдат анализирани са:

**ANAVI MACRO PAD 2**

**ANAVI LIGHT CONTROLLER**

**PIC16F84A**

**PIC16F877A**

**Ключови думи:** PCB, MCU, AVR/PIC, Assembly language

## I. Предметна област

Използването на свободни асемблери е изключително важно за разбирането на компютърните архитектури. Свободните асемблери са с отворен лиценз и могат да бъдат използвани за университетски цели. Могат да бъдат проектирани печатни платки, чрез използването на свободен софтуер като например:

***Fritzing***

***LibrePCB***

Реализираните печатни платки може да бъдат програмирани на съвместим с микроконтролерите им асемблер. За програмирането на различните микроконтролери се използват различни асемблери в зависимост от техния сет от поддържани инструкции (ISA) и целта на проекта, който се постига чрез тях. Няколко вида асемблери са разгледани в [Таблица 0.1].

Ще бъдат разгледани възможностите на различни реализирани печатни платки.

## II. Класификации на реализираните платки

### Класификация спрямо архитектурата (ISA) на микроконтролерите

#### RISC базирани

**ANAVI MACRO PAD 2**- базирана на ATTiny 85 – AVR RISC архитектура

**ANAVI LIGHT CONTROLLER** - базирана на ATTiny 85 – RISC архитектура

#### PIC базирани

**PIC16F84A**

**PIC16F877A**



### **III. Анализ на платките**

#### **ANAVI Macro Pad 2**

##### **Обща характеристика на платката**

ANAVI Macro Pad 2 предоставя възможност за програмиране на 2 основни механични бутони които могат да бъдат програмирани чрез Python-базиран фърмуер. Бутоните могат да използвани за стартирането на различни софтуерни приложения

##### **Описание на микроконтролерите които поддържа:**

*Използван е ATTiny85*

##### **Описание на интерфейсите които поддържа:**

**Описание на асемблерите на които могат да бъдат програмирани микроконтролерите:**

**Асемблерите трябва да бъдат съвместими с AVR RISC архитектурата.**

**Такива са:**

- ☐ GNU Assembler
- ☐ AVR Assembler

## **ANAVI Light Controller**

**обща характеристика на платката**

**описание на микроконтролерите които поддържа**

**описание на интерфейсите които поддържа**

**описание на асемблерите на които могат да бъдат програмирани микроконтролерите**

## **PIC16F84A**

**обща характеристика на платката**

**описание на микроконтролерите които поддържа**

**описание на интерфейсите които поддържа**

**описание на асемблерите на които могат да бъдат програмирани микроконтролерите**

## **PIC16F877A**

**обща характеристика на платката**

**описание на микроконтролерите които поддържа**

**описание на интерфейсите които поддържа**

**описание на асемблерите на които могат да бъде програмирани микроконтролерите**

#### IV. Резултати и обобщаване на резултатите

Анализът на платките е обобщен чрез следната таблица:

Име	Поддържа микроконтролерите	Поддържа интерфейсите	Поддържа асемблерите
<b>ANAVI Macro Pad 2 ANAVI Light Controller</b>	ATTiny85    PIC16F84A PIC16F877A	ISP USB	AVR Assembler

#### V. ЗАКЛЮЧЕНИЕ

Микроконтролери могат да бъдат програмирани на асемблер с цел ефективно използване на паметта им. Могат да бъдат използвани за различни проекти в сферата на компютърните архитектури. В следващата глава ще бъде направен нализ на микроконтролерите които ще бъдат използвани за реализацията на печатна платка. При програмирането на платката ще бъде използван асемблер.

## **2. Изисквания към проекта**

Вследствие на направения обзор на съществуващи печатни платки, и поставената цел на проекта, биват поставени следните изисквания:

### **Изисквания към проектирането на печатната платка:**

- ☐ трябва да бъдат избрани AVR RISC-базиран микроконтролер както и PIC-базиран микроконтролер, между които да бъде извършена комуникацията
- ☐ трябва да бъдат избрани интерфейси с които ще разполага печатната платка
- ☐ трябва да бъде избран софтуер за проектирането на печатната платка. Възможен софтуер може да бъде:
  - ☐ LibrePCB
  - ☐ Fritzing
- ☐ трябва да бъде създадена блокова схема на печатната платка
- ☐ трябва да бъдат избрани подходящи допълнителни хардуерни компоненти
- ☐ с помощта на избрания софтуер трябва да бъдат създадени:
  - ☐ символи на хардуерните компоненти (ако не са налични в основната библиотека с компоненти)
  - ☐ footprints на хардуерните компоненти (ако не са налични в основната библиотека с компоненти)
  - ☐ схема на печатната платка
  - ☐ трябва да бъде избрано разположение на хардуерните компоненти, размерите на печатната платка да бъде малък
- ☐ трябва да бъде редуциран фоновия шум на платката

Изискванията към проектирането на печатната платка ще бъдат изпълнени в шеста глава.

## **Възможности за реализация на платката**

Съществуват различни възможности за комуникация между два микроконтролера:

- Между 2 AVR RISC - базирани микроконтролери

Могат да бъдат програмирани чрез:

AVR Assembler

GNU Assembler

- Между 2 PIC - базирани микроконтролери

Могат да бъдат програмирани чрез:

PIC Assembler

- Между AVR и PIC - базиран микроконтролер

Могат да бъдат програмирани чрез:

PIC Assembler

AVR Assembler

## **Възможни протоколи за използване**

I2C

UART

UEXT

### **Изисквания към софтуерното осигуряване на печатната платка:**

- ☐ трябва да бъдат избрани асемблери съвместими с архитектурата на микроконтролерите
- ☐ трябва да бъде избран протокол за комуникация спрямо когото да бъде написан кодът
- ☐ кодът за микроконтролерите трябва да бъде реализиран чрез функции
- ☐ функциите трябва да бъдат съобразени със избрания протокол за комуникация между микроконтролерите
- ☐ трябва да не се допуска излишно обръщение към регистрите
- ☐ трябва да бъде изложена логиката на компилацията
- ☐ трябва да бъде показана компилацията
- ☐ трябва кодът да бъде реализиран и като софтуерна библиотека поради:
  - ☐ бъдещото развитие на проекта
  - ☐ преносимост

Изискванията към програмирането на печатната платка ще бъдат изпълнени в седма глава.



### **3. Анализ на микроконтролерите които ще бъдат използвани за печатната платка**

В предишната глава бяха разгледани съществуващи печатни платки за които могат да бъдат използвани асемблер за avr или pic assembler.

За реализирането на печатната платка е нужно да бъдат използвани микроконтролери, които могат да бъдат програмирани чрез:

**AVR Assembler**

**PIC Assembler**

Избраните микроконтролери за реализация на проекта са :

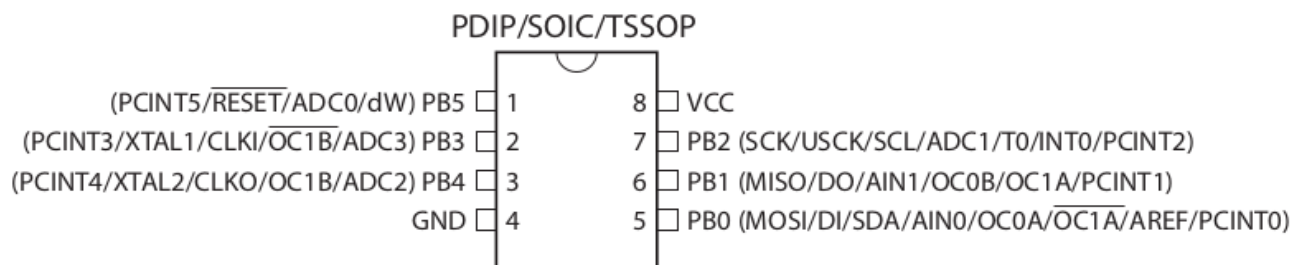
**Attiny85**

**PIC10F320**

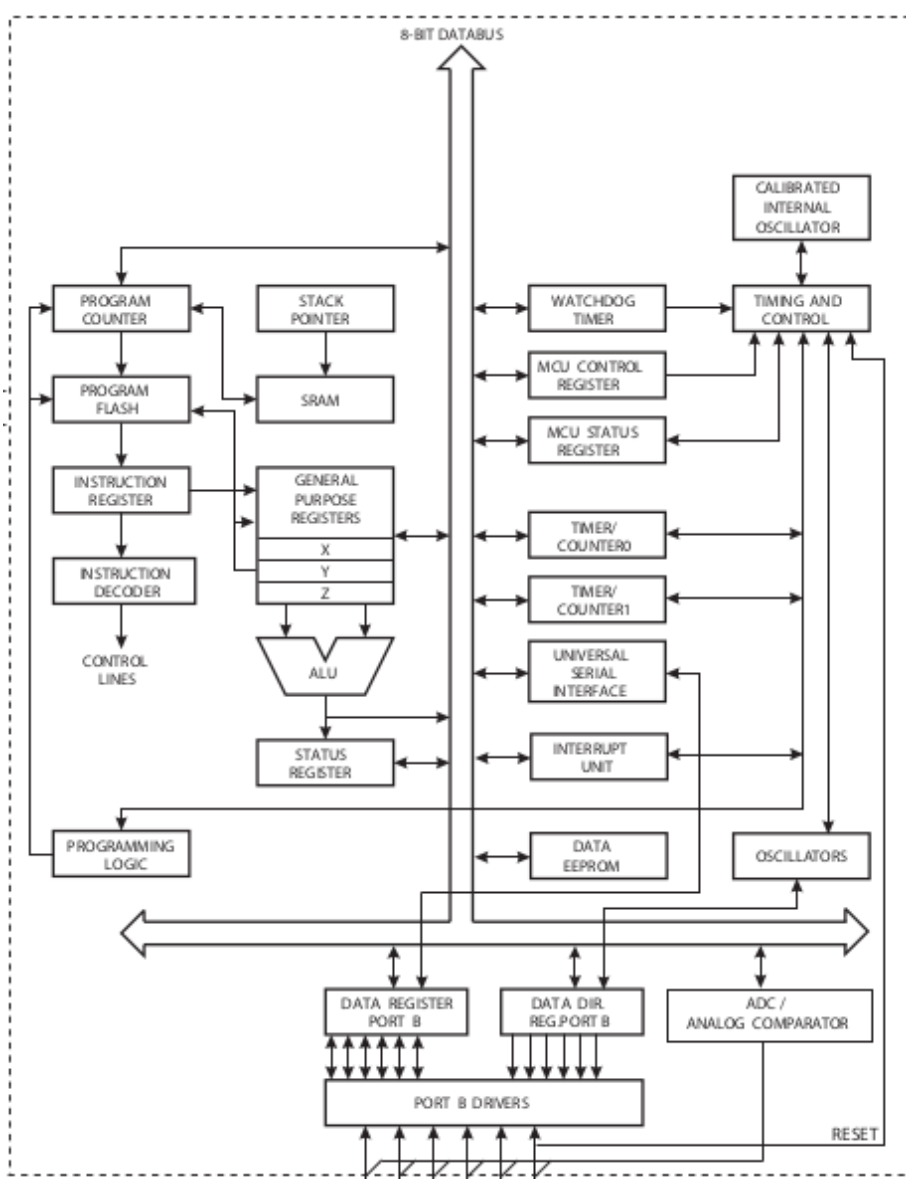
#### **Анализ на Attiny85**

Attiny85 е 8-битов микроконтролер, базиран върху AVR-RISC архитектура.

Следните пинове са достъпни за използване:



Фиг. 3.1



Фиг. 3.2

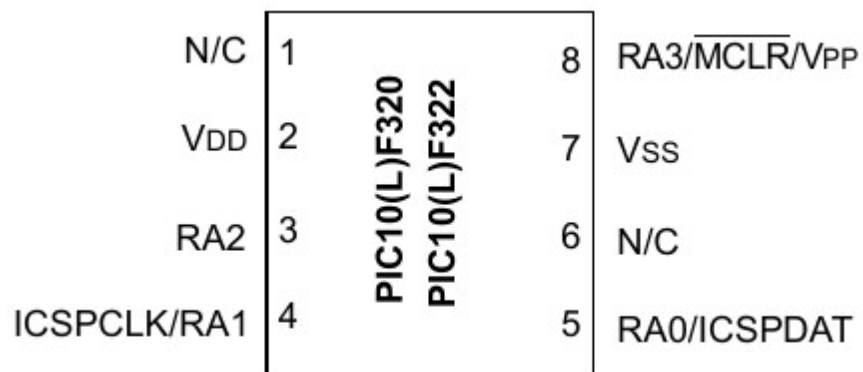
Поддържа 32 регистъра с общо предназначение. За програмирането му могат да бъдат използвани следните асемблери:

**GNU Assembler**

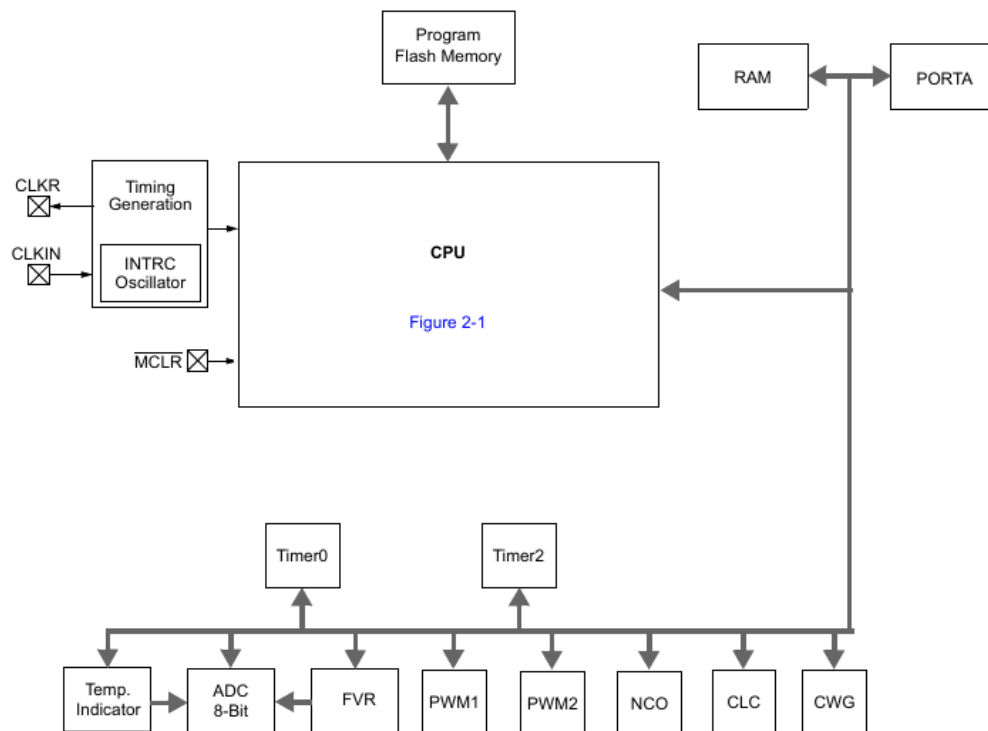
**AVR Assembler**

## Анализ на PIC10F320

Следните пинове са достъпни за използване:



Фиг. 3.3



Фиг. 3.4



#### **4. Анализ на асемблерите които ще бъдат използвани**







## 5. Софтуерни инструменти за проектиране на платката и програмиране



За реализиране на проекта е избран свободен сет от инструменти с отворен код или са част от GNU проекта. Сетът от инструменти е с терминален интерфейс с цел по-удобна работа.

За реализацията на софтуерната част на проекта са избрани следните инструменти:

- GNU Make
- Git
- avra
- gputils
- avr-gcc
- avrdude
- Vim

За реализацията на хардуерната част на проекта е избран следният инструмент:

- Fritzing

## GNU MAKE

**GNU Make** е система за автоматизация на компилацията или тестването на програма. Част от GNU проекта.

Може да бъде използван за множество програмни езици, като например:

C  
C++  
Assembler  
BASH

Може да бъде използвана не само за автоматизация на програмирането, но и за автоматизация на различни услуги.

Всеки makefile се състои от следните компоненти:

**цели (targets)**  
**файлове, нужни за постигане на целта**  
**команда или поредица от команди които се изпълняват в рамките на целта**

*Пример за използването ѝ е изложен по-долу:*

asmcomp: program.

## GIT

Git е разпределена система за контрол на версиите с отворен код, която може да бъде използвана за проследяване на промените за всеки тип файл. Използван е за проследяване на файловете в настоящия проект. Пример за използването ѝ е изложен по-долу:

`git clone https://github.com/vrstanchev/Degrees.git` – клониране на отдалеченото хранилище

`cd Degrees` – достъпване на хранилището

`git add .` - добавяне на файловете на проекта към хранилището

`git commit -m "Master Degree"` - потвърждаване на направените промени във файловете

`git push origin main` - Изпращане на направените промени към отдалеченото хранилище

**AVRA**

## GPUTILS

**AVR-GCC**

**AVRDUDE**





## 6. Проектиране на печатната платка

**Изпълнение на изискванията към проектирането на печатната платка:**

За реализацията на платката са избрани микроконтролерите:

**ATTiny85**

**PIC10F320**

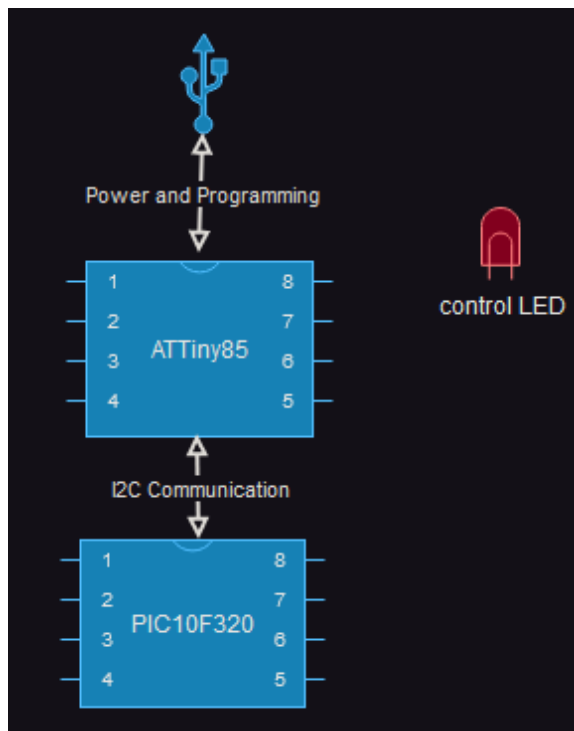
Интерфейси с които ще разполага печатната платка:

USB

Избран софтуер за проектирането на печатната платки:

LibrePCB

Блокова схема на печатната платка

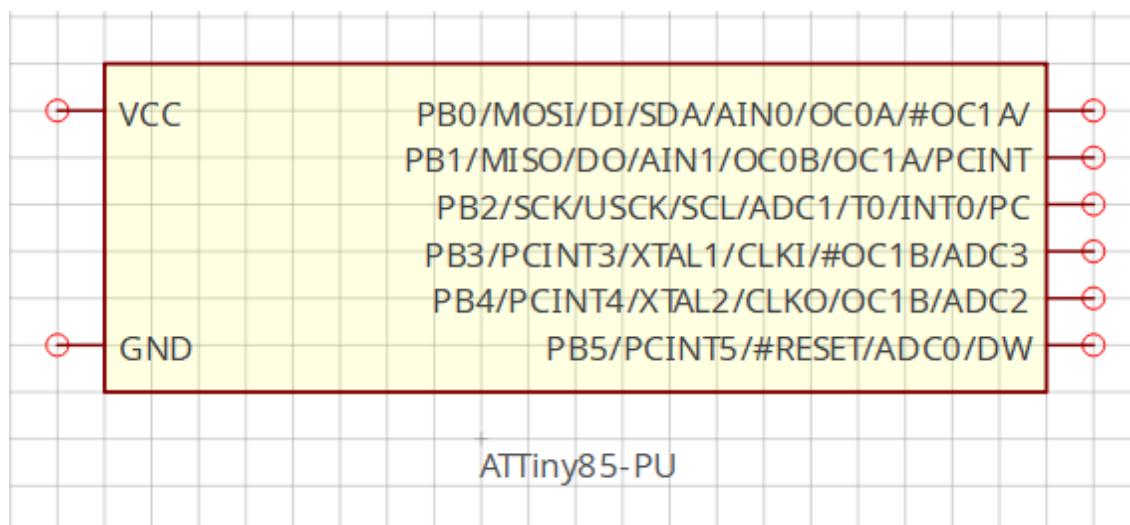


Фиг .6.1

## Хардуерни компоненти

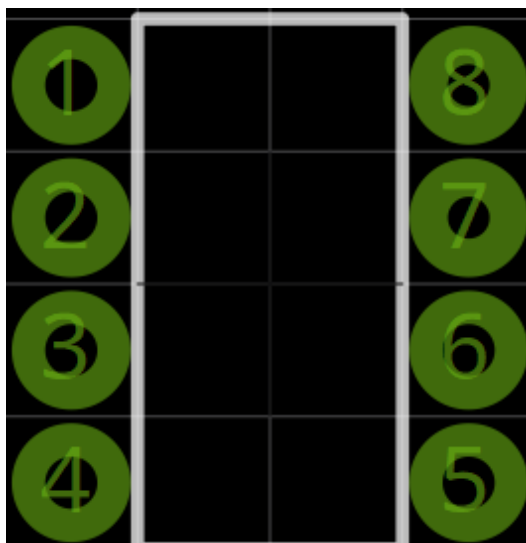
Компонент Attiny85

Символ Attiny85



Фиг .6.2

Footprint Attiny85



Фиг. 6.3

трябва да бъдат избрани подходящи допълнителни хардуерни компоненти

символи на хардуерните компоненти (ако не са налични в основната библиотека с компоненти)

footprints на хардуерните компоненти (ако не са налични в основната библиотека с компоненти)

схема на печатната платка

трябва да бъде избрано разположение на хардуерните компоненти, размерите на печатната платка да бъде малък

трябва да бъде редуциран фоновия шум на платката

## 7. Програмиране на микроконтролерите

**Изпълнение на изисквания към софтуерното осигуряване на печатната платка:**

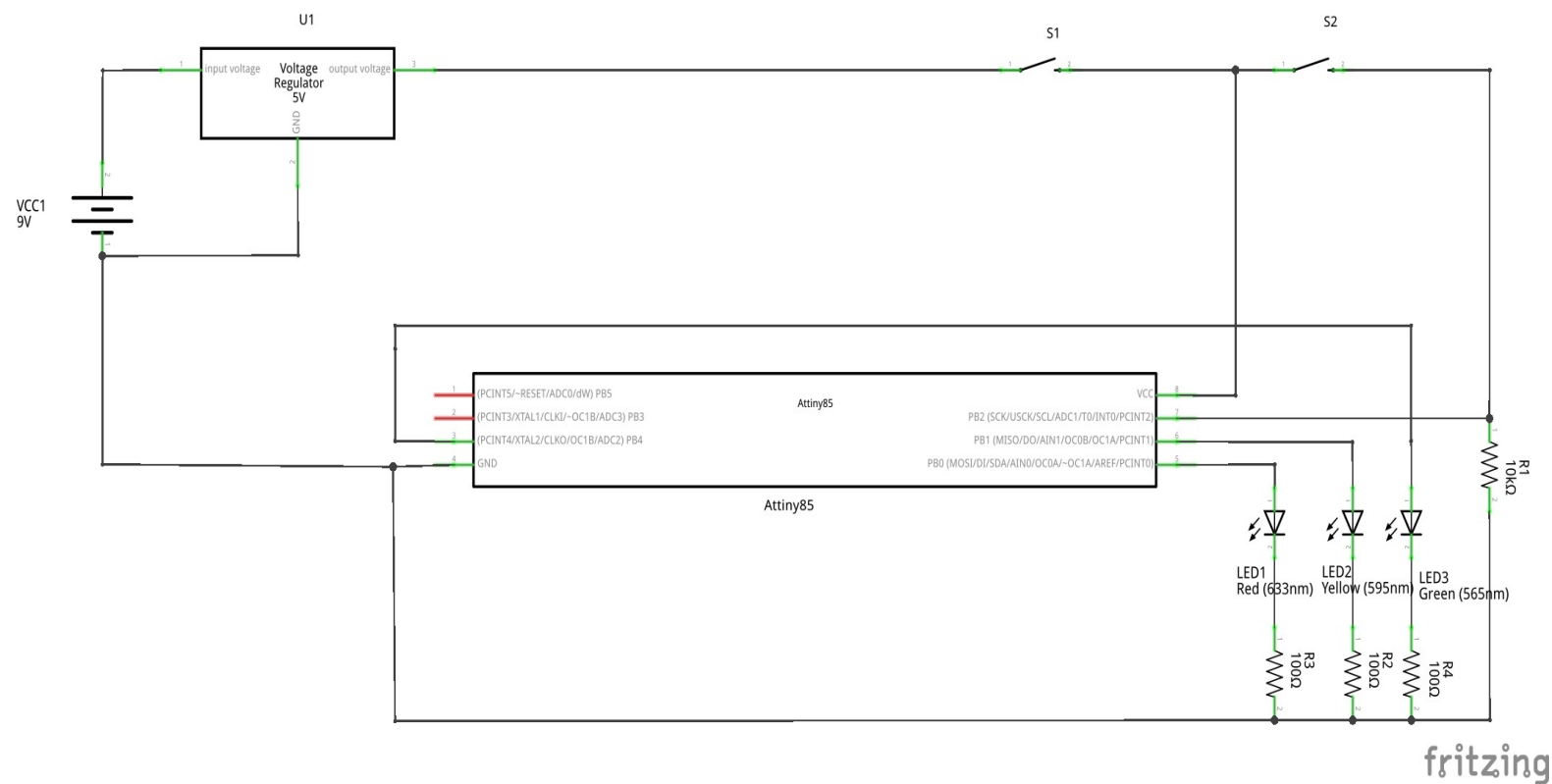
- ☐ Избрани са следните асемблери:
  - ☐ AVR Assembler
  - ☐ PIC Assembler
- ☐ Избран е протокол за комуникация I2C
- ☐ кодът за ATTiny85 ще бъде във функцията att\_func()
- ☐ кодът за PIC10F320 ще бъде във функцията pic\_func()
- ☐ функциите ще бъдат съобразени с I2C
- ☐ трябва да не се допуска излишно обръщение към регистрите
- ☐ трябва да бъде изложена логиката на компилацията
- ☐ трябва да бъде показана компилацията
- ☐ трябва кодът да бъде реализиран и като софтуерна библиотека поради:
  - ☐ бъдещото развитие на проекта
  - ☐ преносимост

## Приложение 1:

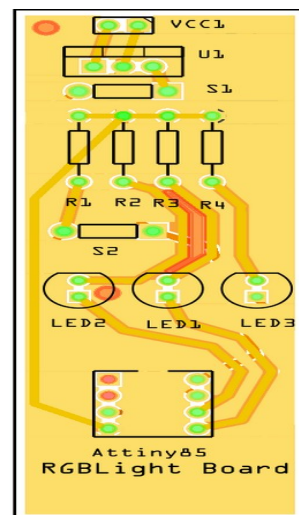
Проектирани AVR/PIC печатни платки използвайки свободно -достъпни схеми и програма за проектиране Fritzing

### 1. Attiny 85 Traffic Lights

#### □ Схема Attiny 85 Traffic Lights



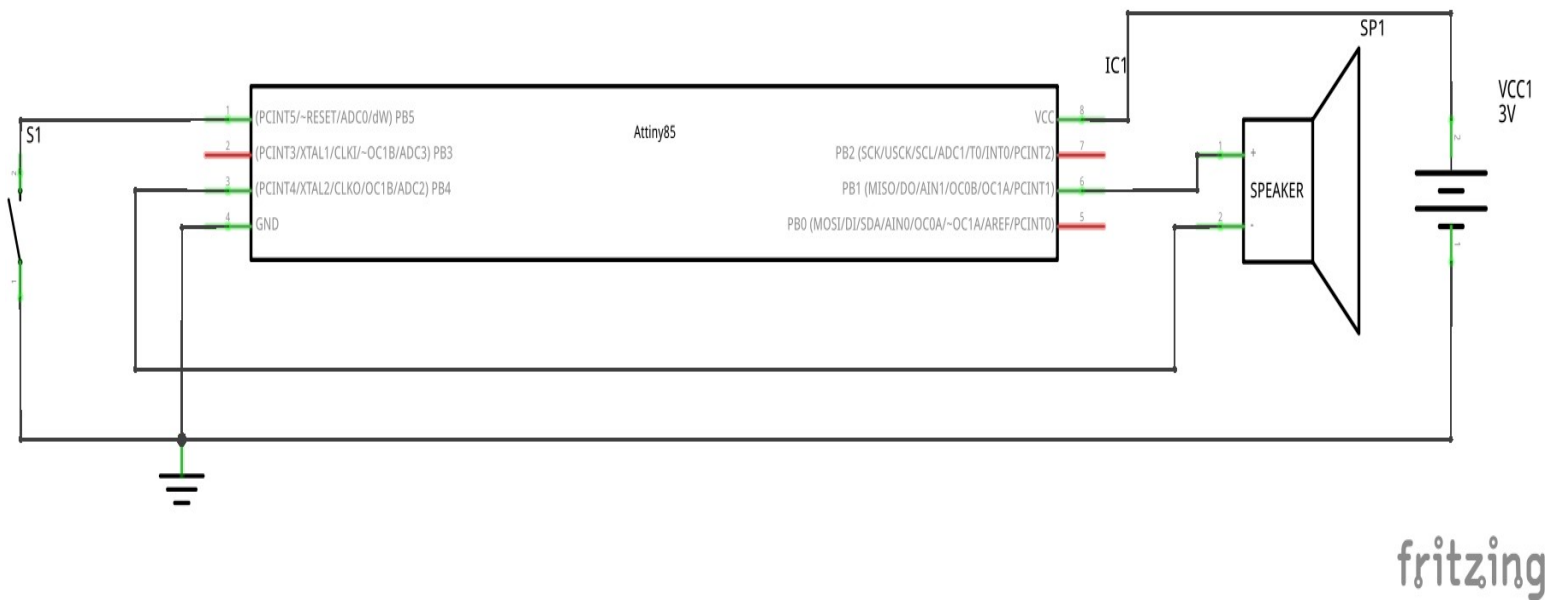
#### □ Преглед на PCB Attiny 85 Traffic Lights



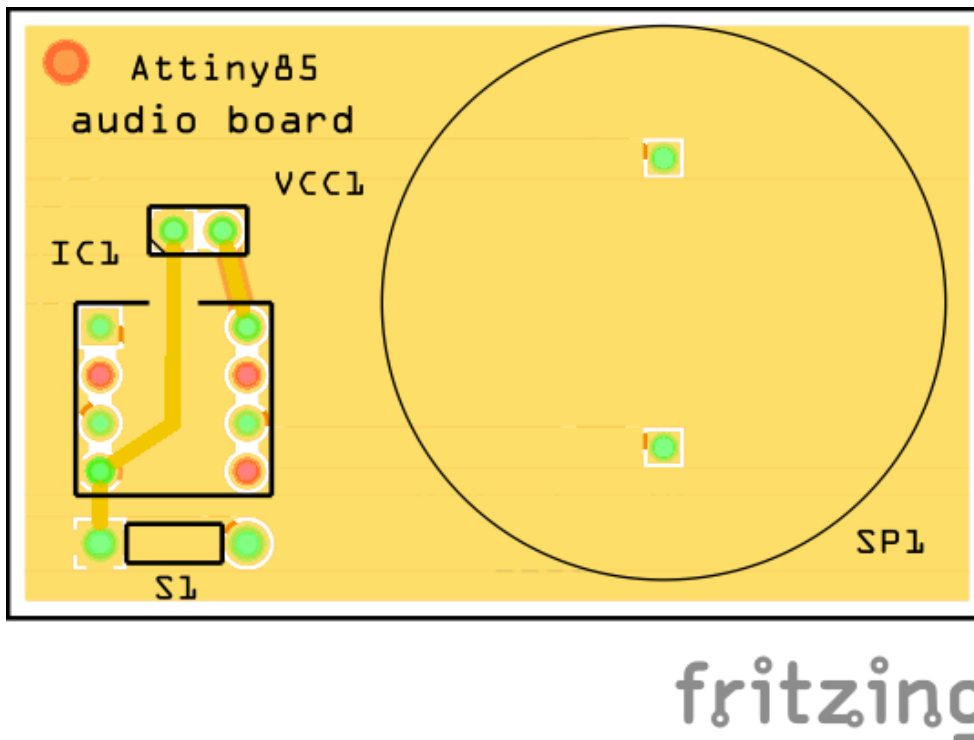
## **2. Attiny 85 Audio Player**

### **□ Cxema Attiny 85 Audio Player**





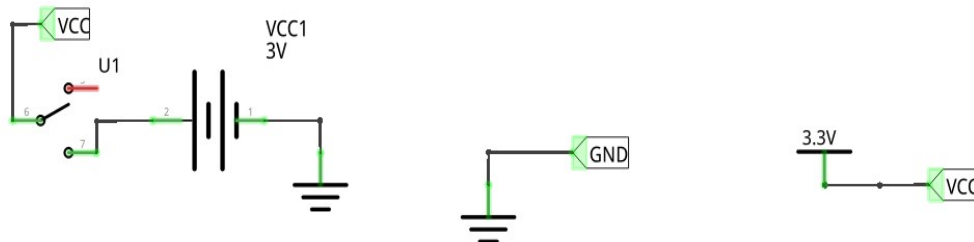
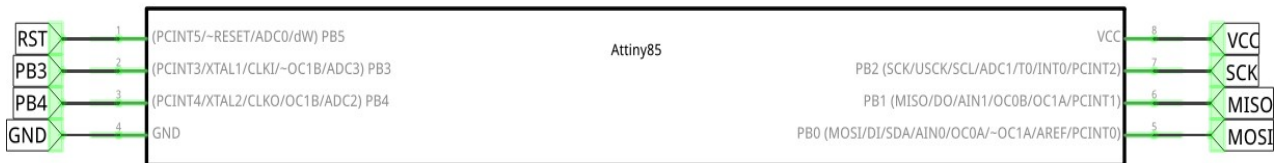
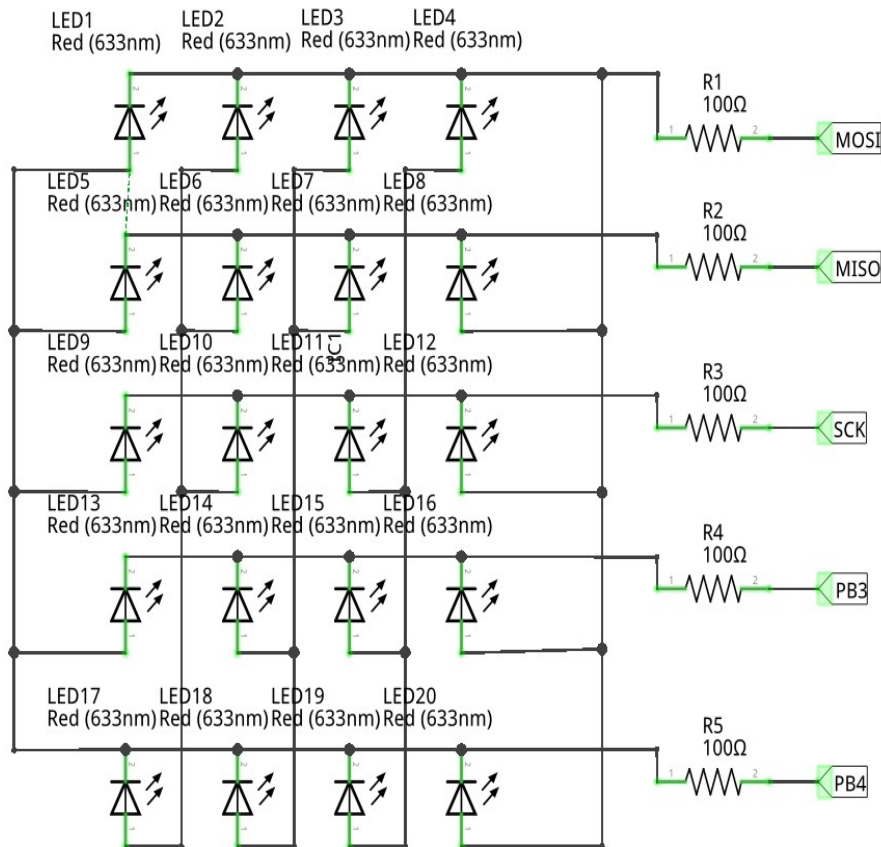
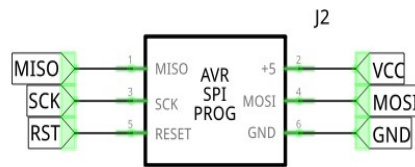
- Преглед на PCB Attiny 85 Audio Player



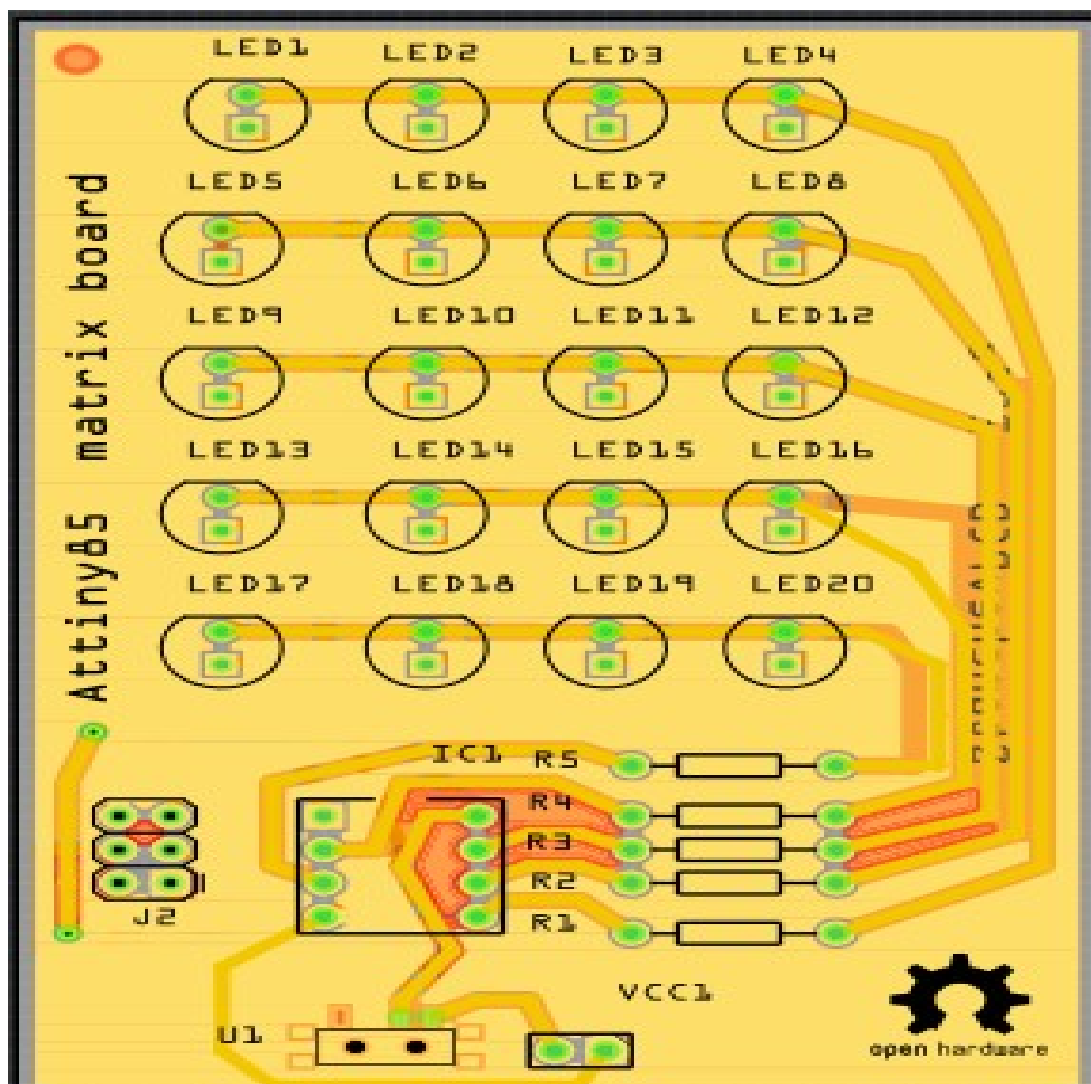
### 3. Attiny 85 LED matrix

- Cхема Attiny 85 LED matrix



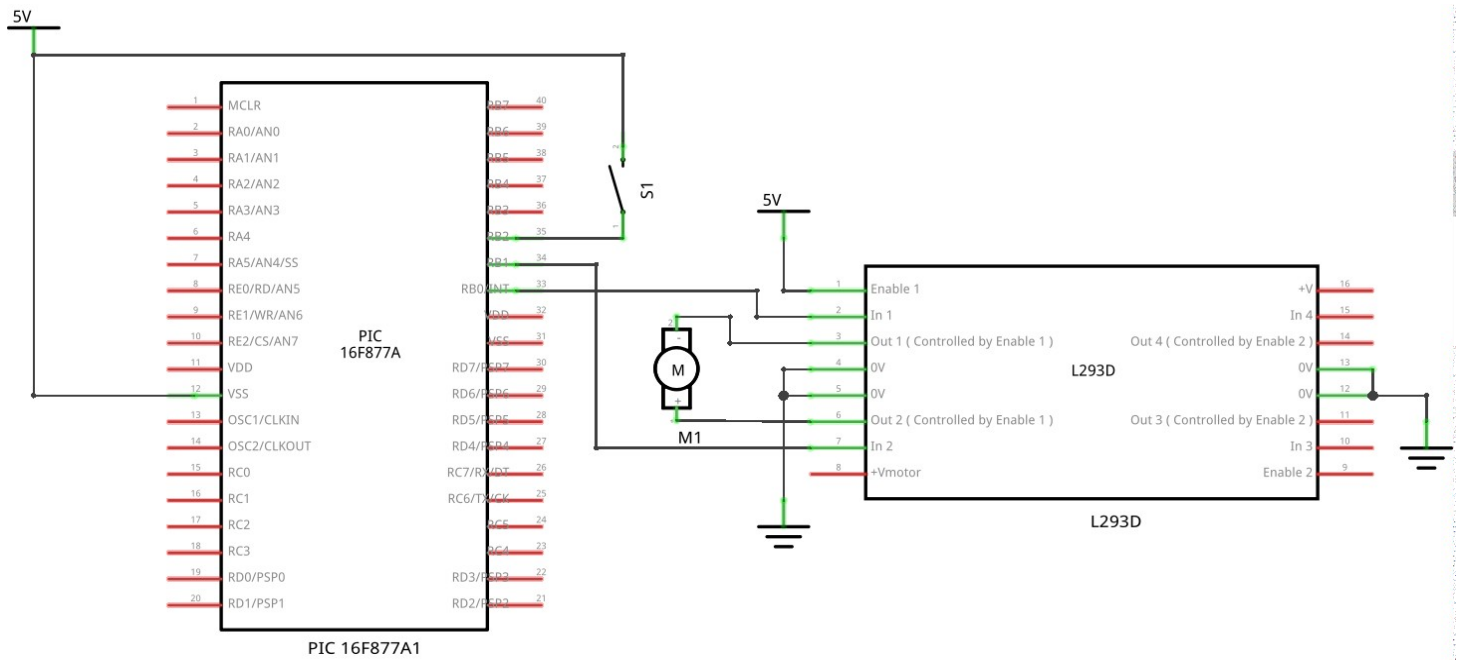


Преглед на PCB Attiny 85  
LED matrix



### 3. pic16f877a Motor

□ Cхема pic16f877a Motor



fritzing

## **Приложение 2**

### **Исходен код на проекта**

Источници:

[1]={ [1.1],[1.2],[1.3],[1.4]}

[1.1] **\_ANAVI Macro Pad 2 DataSheet**

[1.2] **\_ANAVI Light Controller DataSheet**

[1.3] **avr64dd32 DataSheet**

[1.4] **rp2040 board DataSheet**