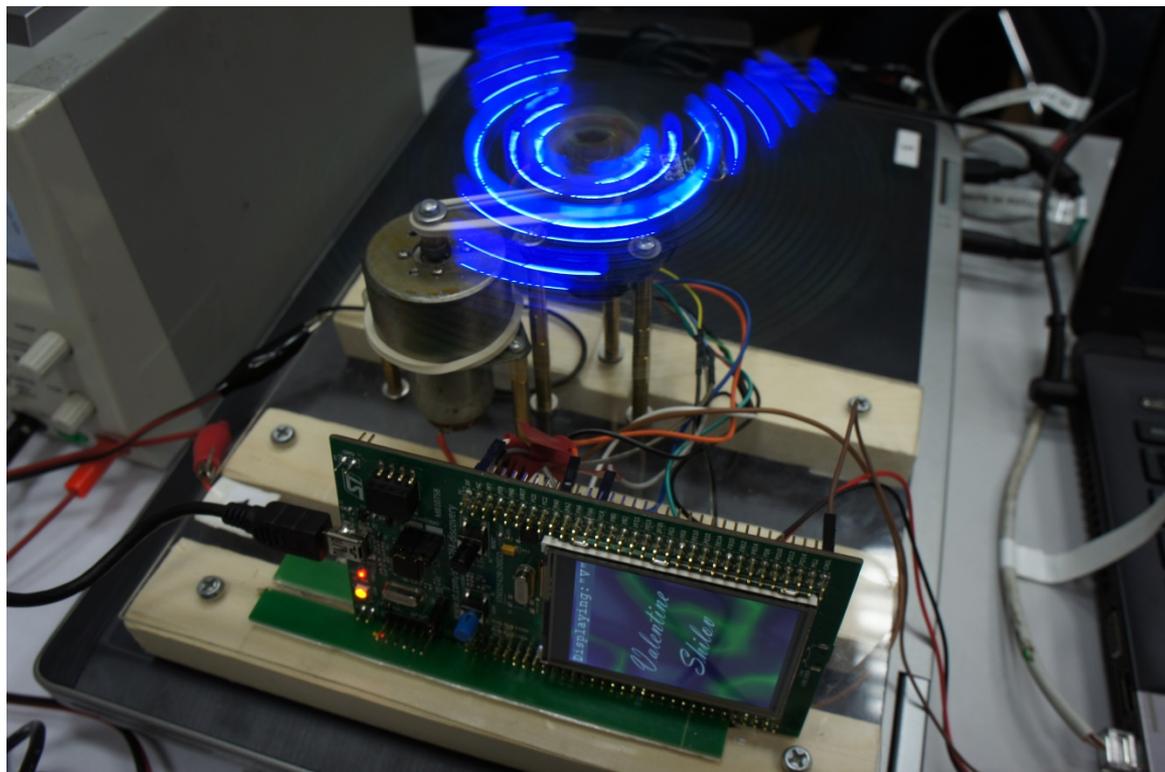


# Stroboscopic display

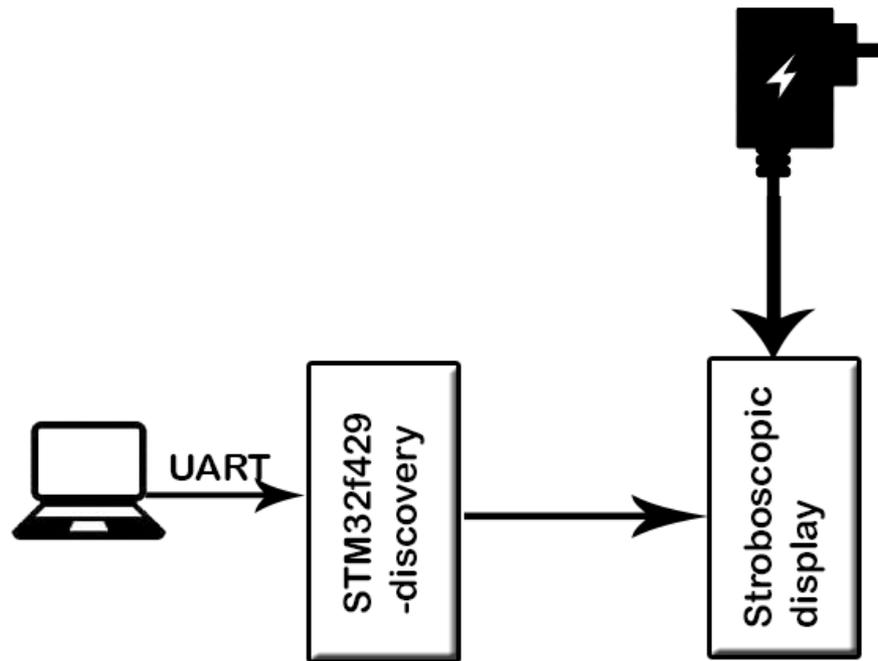


Шилов Валентин

Научный руководитель Николай Петров

# Цель проекта

Отображение на стробоскопическом дисплее символов, передаваемых с компьютера по COM-порту.

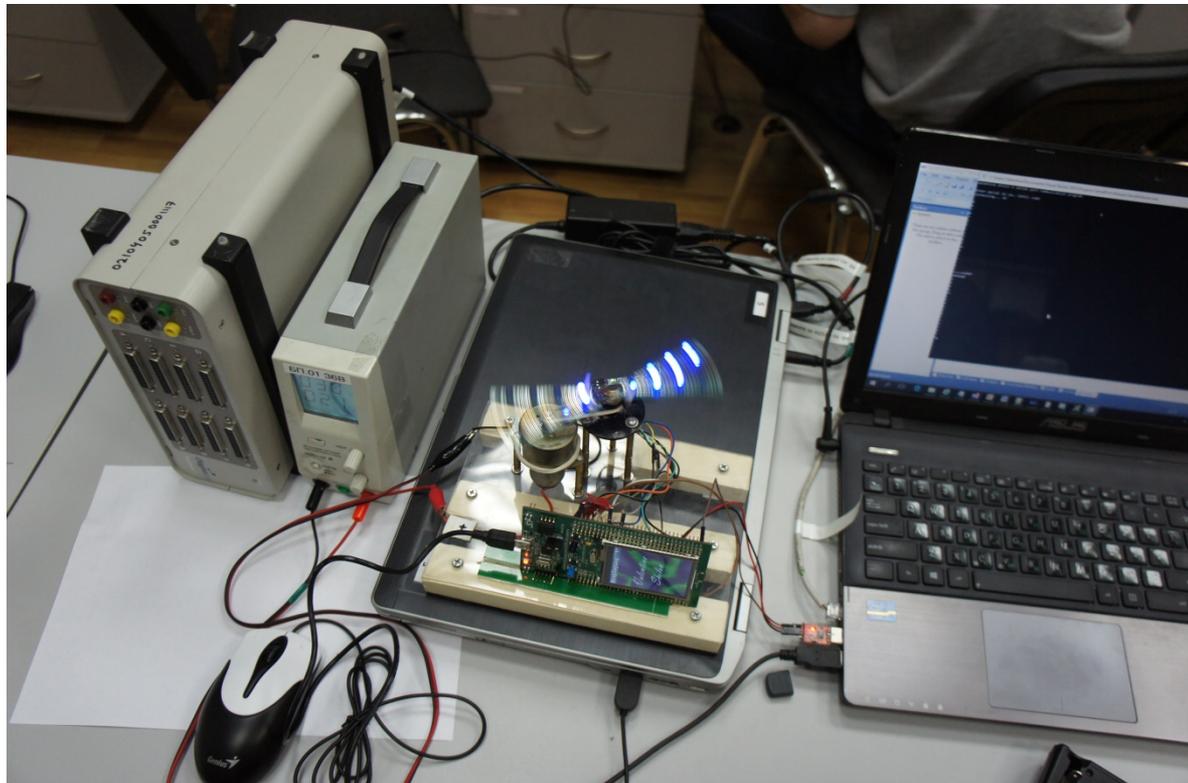


# Оборудование

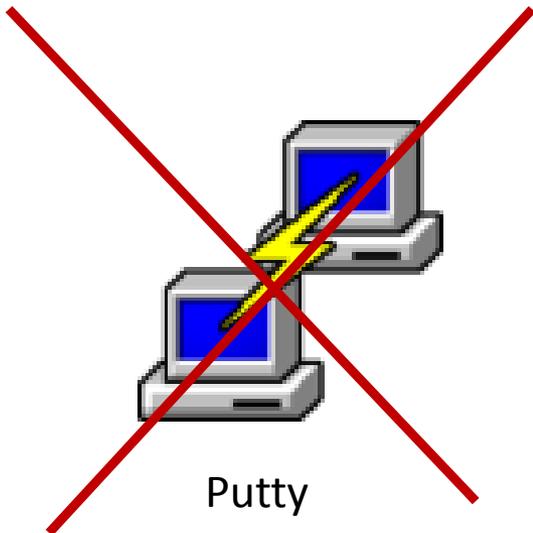
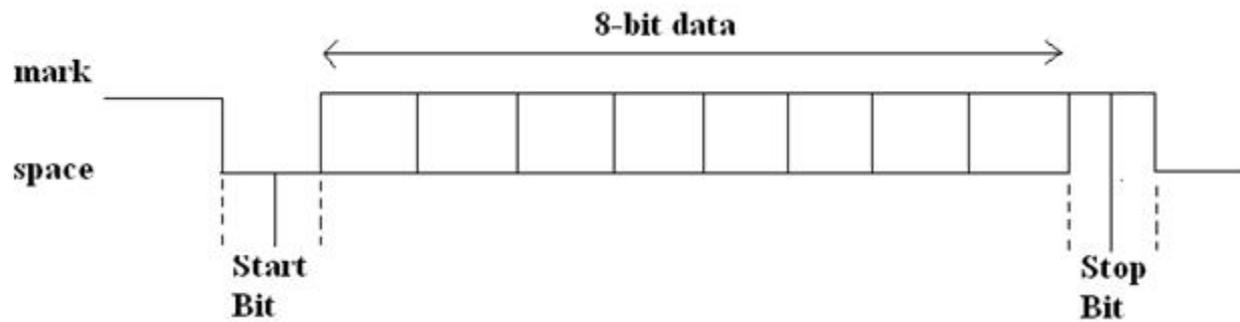
- STM32F429ZI-Discovery
- DC-мотор
- Контроллер геркона
- Блок питания
- 8 светодиодов, сдвиговый регистр
- Адаптер USB->UART

# Стробоскопический эффект

- Частота вращения  $>10\text{Гц}$
- Есть зависимость от яркости



# Получение символов. UART (Universal asynchronous receiver/transmitter)

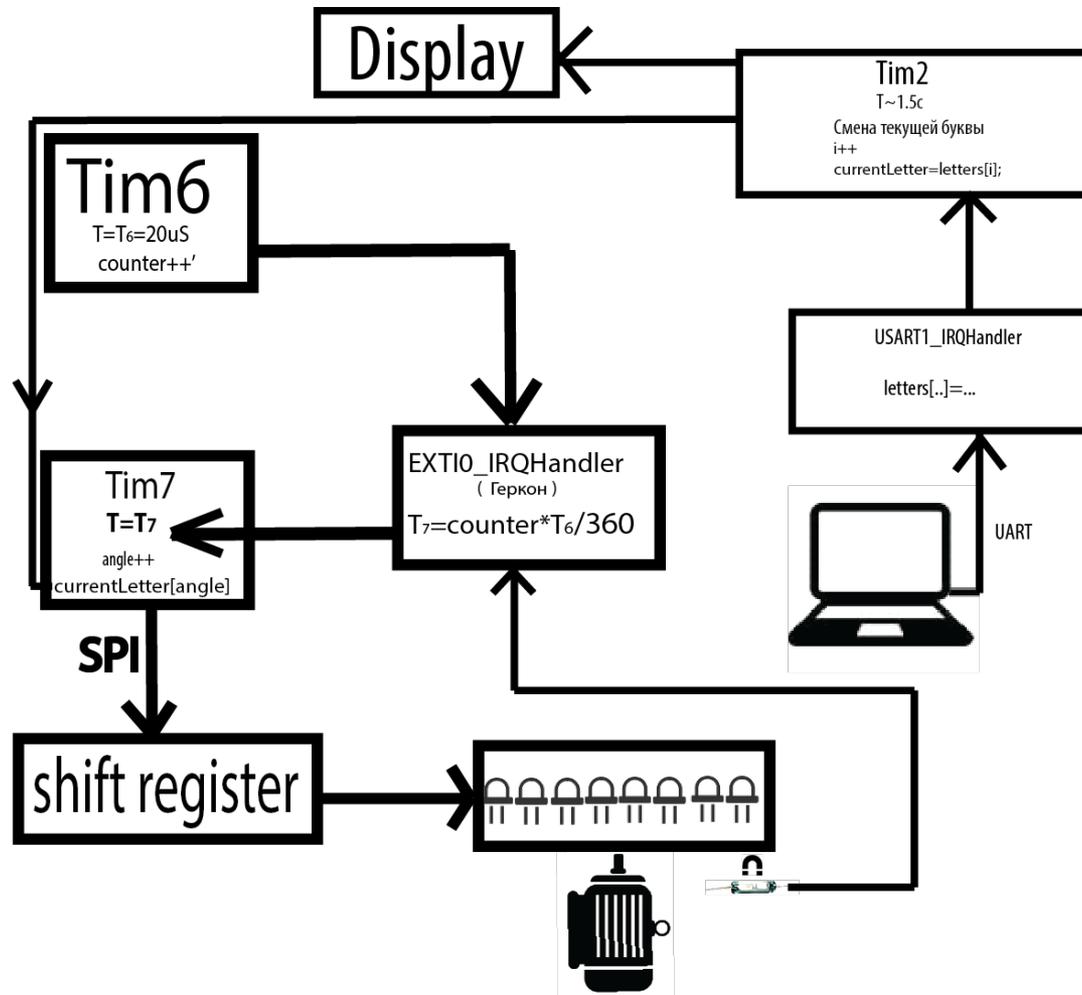


Putty

```
C:\Users\Valentine\Documents\Visual Studio 2010\Projects\SerialPort\Release\SerialArduino.exe
Valentine Shilov's serial port communication program for strobe display
Enter Serial Id (ex. COM3)> COM3
Connecting... OK

Hello!!! Enter symbols in capital letters!!! Supported charsets: English ,Russian
HELLO
```

# Отображение на стробоскопическом дисплее



# Технические подробности:

- Каждый символ для отображения на дисплее хранится в виде массива из 360 байт в том виде, в котором отправляется на сдвиговый регистр. Вся таблица символов представляет из себя массив из (кол-во символов)\*360 байт.

```
//filename=letters_small.txt scaler=1,000000
//symbol table: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ()!<>+ -=~`' ; , . / $ # * & % ^ "
uint8_t letter_0[] = {
//0 -> 65 -> A
  6, 0, 4, 36, 32, 38, 6, 38, 36, 36, 36, 38, 38, 4, 38, 36, 6, 36, 38, 36, 4, 44, 36, 46, 40, 12, 44, 1
//1 -> 66 -> B
  6, 128, 64, 132, 104, 224, 14, 206, 110, 238, 108, 238, 238, 204, 238, 238, 108, 236, 110, 236, 206, 238, 2
//2 -> 67 -> C
  0, 128, 64, 128, 64, 192, 0, 192, 64, 192, 64, 192, 64, 192, 192, 192, 192, 192, 64, 192, 192, 192, 200, 2
//3 -> 68 -> D
  12, 128, 72, 132, 72, 192, 12, 200, 76, 204, 76, 204, 204, 204, 200, 204, 76, 200, 76, 204, 200, 204, 204,
//4 -> 69 -> E
  6, 0, 64, 4, 96, 96, 6, 70, 102, 102, 100, 102, 102, 68, 102, 102, 96, 96, 98, 96, 66, 98, 96, 98, 224,
//5 -> 70 -> F
  6, 0, 64, 4, 96, 96, 6, 70, 102, 102, 100, 102, 102, 68, 102, 102, 96, 96, 98, 96, 66, 98, 96, 98, 96, 6
//6 -> 71 -> G
  14, 136, 76, 140, 72, 206, 206, 206, 206, 206, 204, 206, 206, 204, 206, 206, 206, 204, 206, 204, 206, 204,
//7 -> 72 -> H
```

# Преимущества такой схемы

Перед отправкой по SPI не нужно производить никаких вычислений – высокая скорость

Быстрый поиск нужного символа

## Недостатки:

Требуется большой объём памяти для хранения всей таблицы

# Программа для перевода символов

Получение картинки символа (с помощью Windows API)

Масштабирование

Переход в полярную систему координат

Вычисление значения, которое нужно передать на сдвиговый регистр, чтобы зажечь необходимые для данного угла диоды

Полярная система координат:

$$x = r \cdot \cos(\varphi);$$

$$y = r \cdot \sin(\varphi);$$

Переход:

$$r = \sqrt{x^2 + y^2};$$

$$\varphi = \text{atan2}(y, x);$$

# Вывод произвольной картинки на дисплей микроконтроллера

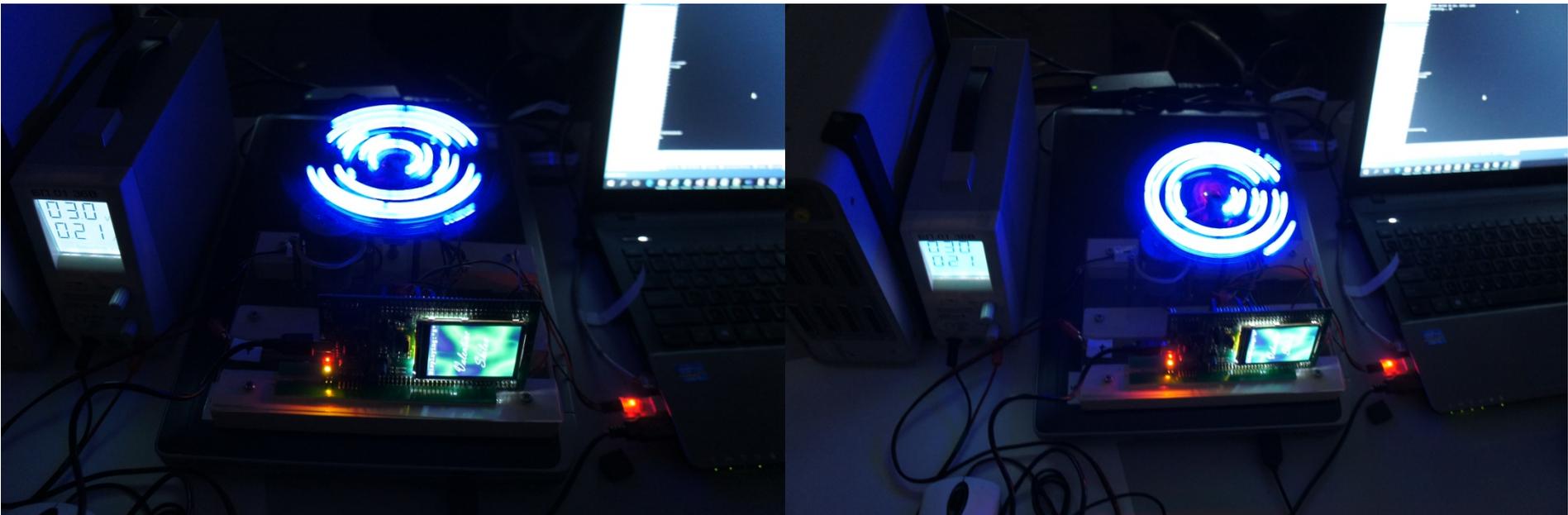
1. Экспорт изображения в формат bmp (**RGB565**)
2. Преобразование в массив байт и размещение в коде программы
3. Для размещения во флеш-памяти используется директива `#pragma location=`
4. Вывод на дисплей при помощи функции `BSP_LCD_DrawBitmap`

```
#pragma location=0x08104000
const uint8_t image[153672] =
{
  0x42, 0x4D, 0x48, 0x58, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x46, 0x00, 0x00, 0x00, 0x38, 0x00,
  0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x40, 0x01, 0x00, 0x00, 0x01, 0x00, 0x10, 0x00, 0x03, 0x00,
  0x00, 0x00, 0x02, 0x58, 0x02, 0x00, 0x12, 0x0B, 0x00, 0x00, 0x12, 0x0B, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0x00, 0x00, 0xE0, 0x07, 0x00, 0x00, 0x1F, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x61, 0x09, 0x61, 0x09, 0x61, 0x09, 0x61, 0x09, 0x61, 0x09, 0x81, 0x09,
  0x81, 0x09, 0x81, 0x09, 0x80, 0x09, 0xA1, 0x09, 0xA1, 0x09, 0xA1, 0x09, 0xA1, 0x09, 0xC1, 0x09, 0xC1, 0x09,
  0xC1, 0x09, 0xE1, 0x09, 0xE1, 0x09, 0x01, 0x0A, 0x01, 0x0A, 0x21, 0x0A, 0x21, 0x0A, 0x41, 0x0A, 0x41,
  0x61, 0x0A, 0x61, 0x0A, 0x81, 0x0A, 0xA1, 0x0A, 0xA1, 0x0A, 0x12, 0xC1, 0x0A, 0xE1, 0x0A, 0xE2, 0x12,
  0x02, 0x13, 0x02, 0x13, 0x22, 0x13, 0x22, 0x13, 0x22, 0x13, 0x22, 0x13, 0x22, 0x13, 0x22, 0x13, 0x42, 0x13,
  0x42, 0x13, 0x42, 0x13, 0x42, 0x13, 0x62, 0x13, 0x62, 0x13, 0x62, 0x13, 0x62, 0x13, 0x62, 0x13,
  0x82, 0x13, 0x82, 0x13, 0x82, 0x13, 0x82, 0x13, 0xA2, 0x13, 0xA2, 0x13, 0xA2, 0x13, 0xA2, 0x13, 0xC2, 0x13,
  0xC2, 0x13, 0xC2, 0x13, 0xE2, 0x13, 0xE2, 0x13, 0x02, 0x14, 0x22, 0x14, 0x22, 0x14, 0x22, 0x14, 0x42, 0x14,
  0x62, 0x14, 0x82, 0x14, 0xA3, 0x1C, 0xC3, 0x1C, 0xE3, 0x1C, 0x03, 0x1D, 0x23, 0x1D, 0x43, 0x1D,
  0x63, 0x1D, 0xA3, 0x1D, 0xC3, 0x1D, 0xA3, 0x1D, 0x83, 0x1D, 0x63, 0x1D, 0x43, 0x1D, 0x23, 0x1D,
  0x03, 0x1D, 0x03, 0x1D, 0xE3, 0x1C, 0xC3, 0x1C, 0xA3, 0x1C, 0x83, 0x14, 0x62, 0x14, 0x42, 0x14,
  0x22, 0x14, 0x02, 0x14, 0xE2, 0x13, 0xC2, 0x13, 0xA2, 0x13, 0xA2, 0x13, 0x82, 0x13, 0x62, 0x13,
  0x62, 0x13, 0x42, 0x13, 0x22, 0x13, 0x22, 0x13, 0x02, 0x13, 0x02, 0x13, 0xE2, 0x12, 0x12, 0xE2, 0x12,
  0xE2, 0x12, 0x0A, 0xC2, 0x0A, 0xA1, 0x0A, 0xA2, 0x0A, 0xA2, 0x0A, 0xA2, 0x12, 0xA1, 0x0A,
  0xA1, 0x0A, 0xA1, 0x0A, 0xA1, 0x0A, 0xA1, 0x0A, 0x81, 0x0A, 0x81, 0x0A, 0x81, 0x0A, 0x81, 0x0A,
  0x81, 0x0A, 0x61, 0x0A, 0x61, 0x0A, 0x61, 0x0A, 0x61, 0x0A, 0x41, 0x12, 0x41, 0x0A,
  0x41, 0x0A, 0x41, 0x0A, 0x41, 0x0A, 0x41, 0x0A, 0x21, 0x0A, 0x21, 0x0A, 0x21, 0x0A, 0x01, 0x0A,
  0x01, 0x0A, 0x01, 0x0A, 0xE1, 0x09, 0xE1, 0x09, 0xE1, 0x09, 0xE1, 0x09, 0xC1, 0x09, 0xC1, 0x09,
```



# Результат

- Реализована отрисовка всех букв английского и русского алфавитов на стробоскопическом дисплее
- Последовательный вывод слов



# Перспективы

- Увеличение количества светодиодов для получения непрерывного изображения
- Отображение анимированных изображений
- Вывод на стробоскопический дисплей изображений, рисуемых в реальном времени на сенсорном экране

Спасибо за внимание!

