

Министерство образования Пензенской области  
МБОУ финансово-экономический лицей № 29 г. Пензы

**Проект**

для участия в V открытом региональном конкурсе  
исследовательских и проектных работ школьников  
«Высший пилотаж - Пенза» 2023  
на тему:

**«Screen Light Sync»**

Выполнил:  
Стулков Лукьян

Научный руководитель:  
Трофимов Юрий Александрович

Пенза, 2022

## Содержание

Введение.....	3
1. Теоретическая часть.....	4
1.1. Особенности пользования компьютером при недостаточной освещенности. ....	4
1.2. Сравнение возможных аналогов .....	4
1.3. Программно-аппаратная платформа Arduino .....	5
1.4. Язык программирования Python .....	6
1.5. Использование Serial-порта для передачи информации.....	7
2. Практическая часть .....	9
2.1. Схема проекта Screen Light Sync.....	9
2.2. Основные компоненты Screen Light Sync .....	9
2.3. Программная часть с использованием Python .....	10
2.4. Программная часть с использованием Arduino .....	11
2.5. Тестирование Screen Light Sync .....	12
2.6. Оценка себестоимости проекта.....	12
Заключение .....	13
Список литературы .....	14
Приложение А .....	15
Приложение В.....	17

## Введение

Большое число пользователей проводят много времени за компьютером в темное время суток при выключенном свете. При продолжительной работе за компьютером в темноте повышается нагрузка на глаза. Это связано с тем, что человек не может быстро приспособиться к переменному свету: вокруг человека темнота, а впереди яркий свет. Это приводит к переутомлению глазного яблока.

Устройство Screen Light Sync позволит снизить нагрузку на глаза и адаптировать освещение для комфортного пользования компьютером.

### Цель работы

Создание устройства для подсветки пространства при использовании компьютера с возможностью автоматической регулировки яркости.

### Объект исследования

Возможность синхронизации яркости экрана с подсветкой пространства при использовании компьютера.

### Предмет исследования

Прототип устройства Screen Light Sync.

### Задачи

- 1) Изучить особенности пользования компьютером в темное время суток.
- 2) Разработать схему проекта.
- 3) Разработать аппаратную часть проекта.
- 4) Разработать программную часть
- 5) Создать прототип устройства Screen Light Sync
- 6) Провести тестирование прототипа
- 7) Выполнить экономическое обоснование проекта.

В процессе работы над проектом были рассмотрены:

- программно-аппаратная платформа Arduino (модификация Seeduino Xiao);
- среда разработки Arduino IDE;
- язык Python и библиотеки tkinter, serial, numpy, pyautogui.

### Практическая значимость

Устройство Screen Light Sync позволит свести к минимуму нагрузку для глаз при длительной работе за компьютером при недостаточном освещении.

### Целевая аудитория

Устройство Screen Light Sync может использоваться в домашних условиях, в образовательных учреждениях и офисах.

**Ожидаемыми результатами** проекта являются создание макета устройства Screen Light Sync и его практическое применение при выполнении повседневных задач.

**В перспективе планируется** создать полноценную модель устройства, расширить функционал устройства и наладить массовое производство.

## 1. Теоретическая часть

### 1.1. Особенности пользования компьютером при недостаточной освещенности.

Частая смена яркости картинки в тёмной комнате заставляет наши глаза постоянно подстраиваться. В темноте зрачок расширяется, при свете, наоборот, сужается – учитывая высокую динамичность картинки на экране монитора, получаем постоянное изменение размеров зрачка. Как результат – слезящиеся красные глаза после длительного пользования компьютером.

Усталость глаз от просмотра в темноте можно значительно снизить, уменьшив контраст между экраном и окружающей обстановкой. Это можно сделать двумя способами:

- отрегулировав яркость экрана;
- используя внешнюю подсветку.

Но каждый раз регулировать яркость экрана при изменении освещения не очень удобно и на это требуется много времени. Конечно, существуют устройства, например, ТВ-приемники, с автоматическим режимом изменения яркости экрана в зависимости от освещённости. Данный режим позволяет получать изображение оптимальной яркости, минимально нагружающее глаза и при дневном свете, и в полной темноте. В данном случае изменяется яркость экрана, но есть и проблемная сторона: датчик света может быть физически чем-нибудь закрыт, и устройство, считая, что в помещении темно, будет неправильно регулировать яркость экрана.

Из-за этих проблем лучше и эффективнее изменять внешнее освещение в зависимости от яркости экрана. Поэтому в устройстве Screen Light Sync есть специальный автоматический режим: яркость подсветки адаптируется под яркость экрана.

### 1.2. Сравнение возможных аналогов

1) Умный светильник YEELIGHT LED Screen Light Bar Pro (рис. 1)



Рисунок 1. Умный светильник YEELIGHT LED Screen Light Bar Pro

Стоимость: 6 500 р.

Особенности: есть пульт управления, беспроводной тип соединения с компьютером, провод питания.

**Сравнение.** Устройство Screen Light Sync выглядит минималистичнее, следовательно, меньше весит. У Screen Light Sync нет пульта, управление осуществляется с помощью приложения.

2) Светодиодная лампа Baseus (рис. 2)



Рисунок 2. Светодиодная лампа Baseus

Стоимость: 4 300 р.

Особенности: нет пульта - управление осуществляется кнопками на самой лампе.

**Сравнение.** Устройство Screen Light Sync не имеет кнопок на корпусе, потому что:

- 1) имеет минималистичный дизайн;
- 2) нет возможности поломки кнопок;
- 3) более высокий уровень доступности стоимости устройства.

### 1.3. Программно-аппаратная платформа Arduino

Arduino – это инструмент для проектирования электронных устройств более плотно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры, которые фактически не выходят за рамки виртуальности. Данный микроконтроллер применяется для создания электронных устройств с возможностью приема сигналов от различных цифровых и аналоговых датчиков, которые могут быть подключены к нему, и управления различными исполнительными устройствами. Проекты устройств, основанные на Arduino, могут работать самостоятельно или взаимодействовать с программным обеспечением на компьютере (напр.: Flash, Processing, MaxMSP).

Платы Arduino относительно дешевы по сравнению с другими платформами, их программное обеспечение работает под ОС Windows, Macintosh OSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows. Среда подходит как для начинающих пользователей, так и для опытных и основывается на среде программирования Processing.

Сфера использования данной платформы на современном этапе практически безгранична. С помощью неё можно спроектировать множество различных систем, которые смогут помочь человеку оптимизировать затраты природных ресурсов и уменьшить стоимость потребляемых услуг. Так на базе микроконтроллеров работает система «умный дом», автоматические вентиляторы и светофоры, миниметеостанции, квадрокоптеры и т.д. Спектр применения широк.

Основные преимущества платформы Arduino: открытая схема оборудования, открытый код программы, простая и удобная среда программирования, возможность функционирования на различных видах систем, программирование, подключение и питание может осуществляться одним USB-кабелем, приемлемая цена оборудования.

В нашем проекте будет использоваться самая маленькая Arduino-плата из всех, что представлены на рынке. При этом функциональность ее на высоте, у нее много возможностей и интерфейсов, которые так нужны разработчикам (рис. 3).



Рисунок 3. Плата Seeeduino Xiao

#### 1.4. Язык программирования Python

Язык программирования Python является, пожалуй, одним из самых популярных. Используется в веб-разработке, DataScience, системах автоматизации, приложениях – сферы его применения поистине безграничны. Но главное преимущество Питона – это его низкий порог входа. Иными словами, обучиться программированию на этом языке может практически каждый.

Python обладает простым и понятным синтаксисом. Даже далекому от программирования человеку будет несложно разобраться в строках кода.

Интерпретируемость — одно из главных преимуществ “питона”. Напомним, что интерпретируемым называется тот язык, код на котором не нужно компилировать, а можно сразу запускать.

Запуск кода на Python возможен на любом ПК, интерпретаторы для него есть везде. А это значит, что разработчик может сразу видеть результат. Однако, будучи интерпретируемым языком, Python уступает по скорости компилируемым языкам (поскольку не работает непосредственно на компьютере).

Интерпретируемость сильно упрощает тестирование и перемещение с платформы на платформу небольших блоков кода. Совместимость с большинством существующих ОС делает Python универсальным языком программирования.

Он прекрасно подходит для тех, кто только начинает программировать. Данный высокоуровневый язык позволяет разработчику сосредоточиться на сути и целях, а не конкретных шагах и процедурах. По этой причине писать скрипты на Python оказывается быстрее, чем на многих других языках. Его сходство с естественным языком (английским) сильно облегчает его изучение, если сравнивать Python с другими языками программирования. Разработчику легко запомнить синтаксис “питона” и начать читать код.

Помимо небольших прикладных скриптов, на Python можно писать и крупные коммерческие приложения, думая только о том, как обеспечить им необходимый функционал (все мелкие задачи решаются “под капотом”, и программисту не нужно беспокоиться о них). Надежность делает язык программирования Python очень популярным.

Ключевым фактором при его выборе является гибкость этого языка программирования и его соответствие парадигме объектно-ориентированного программирования. Поэтому его активно применяют в Data Science и машинном обучении.

Преимущества языка программирования Python:

- Простота изучения. Язык программирования Python легко выучить с нуля, поскольку он ясен, логичен и читабелен. Сложный синтаксис и запутанные правила — это не про “питон”. Если вы владеете английским, то у вас всё получится.
- Высокая востребованность языка. В 2020 г. в рейтинге IEEE Spectrum Python занял первое место. Он не только широко распространён, но и применим в разных видах программных продуктов. Заработки Python-программистов очень неплохи, особенно в веб-разработке, машинном обучении и Data Science.
- Использование Python в технологиях искусственного интеллекта и машинного обучения. С помощью Python удобно производить сложные расчеты, поэтому он и пригодился в машинном обучении. А библиотеки, отвечающие за нейронные системы, пришлись очень кстати в ИИ.
- На языке программирования Python можно написать программу для ПК, мобильное и веб-приложение (сегодня в вебе и мобильной разработке очень много таких продуктов). У языка Python имеется множество расширений для создания микро- и макропроектов.
- Открытый исходный код и бесплатная доступность языка на официальном сайта. Кроме того, каждый может внести свой вклад в развитие Python и добавить новые фичи в него.
- Язык независим от ОС и с одинаковой эффективностью работает на главных платформах — Linux, Windows, macOS.

Недостатки Python:

- Медленная работа. “Питон” плохо подходит для создания высокопроизводительных приложений, лучше сразу выбрать другой язык.
- Крайне мало возможностей для процессорных оптимизаций: их сводит на нет особая модель памяти языка.

Для нашего проекта данные недостатки являются несущественными, поэтому язык Python был выбран в качестве основного для написания приложения для управления устройством Screen Light Sync.

## 1.5. Использование Serial-порта для передачи информации

Serial port (серийный порт, последовательный порт или COM-порт -, communications port) - это последовательный интерфейс с двойной направленностью. Вся информация по этому порту передается с шагом равным одному биту, т.е. данные передаются бит за битом, в отличие от параллельного порта. Несмотря на то, что в некоторых других интерфейсах как, например, в Ethernet, FireWire и USB, применяется последовательный обмен данными, название «последовательный порт» закрепилось за Serial-портом.

Данный порт, в сравнении с другими "последовательными" технологиями, обладает отличительной особенностью: в нем отсутствует какое-либо временное требование между 2 байтами. Временные требования существуют только между битами одного байта. Величина, обратная временной паузе между битами одного байта, носит название «baud rate» (скорость передачи). Кроме того, в данной технологии нет такого понятия, как "пакет". Другие технологии "последовательной" передачи данных (X.25, USB или Ethernet), используют "пакеты", также в них существуют и жесткие временные требования между битами одного пакета.

Ранее последовательный порт применялся для подключения терминала, позднее его использовали для подключения модема или мыши. В настоящее время его применяют в качестве средства соединения с источниками бесперебойного питания, а также в качестве средства связи с аппаратными средствами разработки встраиваемых вычислительных систем, спутниковыми ресиверами, кассовыми аппаратами и приборами систем безопасности.

Несмотря на морально устаревший формат, Serial-порт еще нередко используется в промышленном и узкоспециальном оборудовании, а также на некоторых современных компьютерах. Последовательный порт активно вытесняется интерфейсом USB и FireWire.

Однако имеются специальные стандарты эмуляции последовательного порта над USB и над Bluetooth. Программная эмуляция порта широко используется и по сей день. Так, практически все мобильные телефоны сегодня эмулируют внутри себя COM-порт и модем, с целью реализации доступа компьютера к сети Интернет.

Главное преимущество данной технологии состоит в простоте подключения.

Главными недостатками данного порта являются его низкая скорость, большие размеры разъемов, а также высокие требования к времени отклика операционной системы. Также, в данном стандарте наблюдается высокое количество прерываний (одно прерывание на каждые 8 байт).

Для нашего проекта основным каналом связи между устройством Screen Light Sync и компьютером является именно Serial-порт т.к.:

- Arduino эмулирует именно Serial-порт;
- он прост в реализации и программировании;
- его недостатки несущественны для проекта.



## 2. Практическая часть

### 2.1. Схема проекта Screen Light Sync

По проводу, подключенному к компьютеру и к лампе, идет питание и информация о яркости экрана к плате Arduino, а она, в соответствии с полученной информацией о яркости экрана, регулирует напряжение светодиодной ленты в устройстве.

### 2.2. Основные компоненты Screen Light Sync

1. Каркас (рис. 4) (он сделан тонким, из специального алюминиевого короба, чтобы не загромождать пространство для работы за компьютером).



Рисунок 4. Каркас устройства

2. USB-провод (рис. 5) (по нему идет напряжение и информация о яркости экрана).



Рисунок 5. USB-кабель

3. Плата Arduino (рис. 6) (она считывает информацию о яркости экрана через Serial-порт и регулирует напряжение светодиодной ленты в устройстве Screen Light Sync).

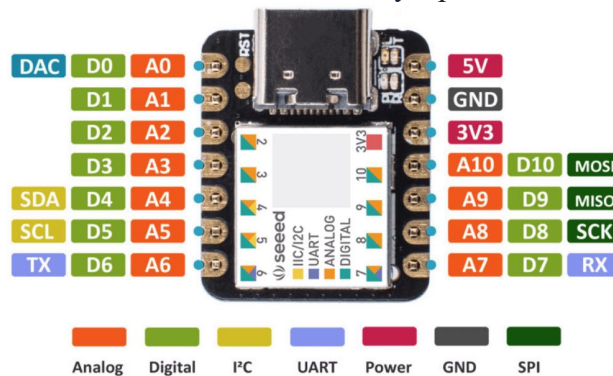


Рисунок 6. Плата Arduino

4. Светодиодная лента с рабочим напряжением 5v (рис. 7) (она получает определенное напряжение с платы Arduino).



Рисунок 7. Светодиодная лента

### 2.3. Программная часть с использованием Python

В программе есть два режима: автоматический и ручной (рис. 8).

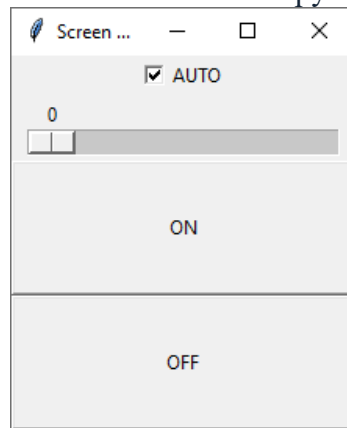


Рисунок 8. Интерфейс приложения (автоматический режим)

В ручном режиме можно выставить любую яркость света от 0 до 255 (в соответствии со значениями, используемыми в Arduino) (рис. 9).

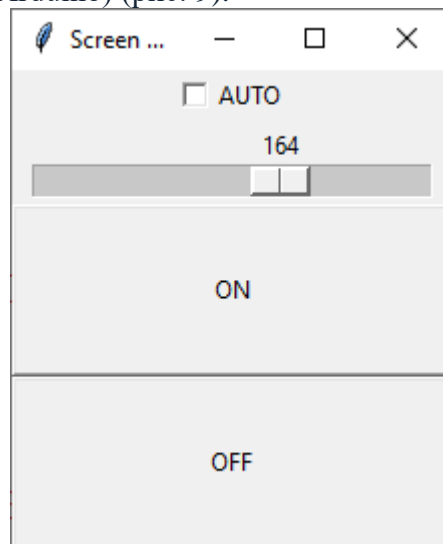
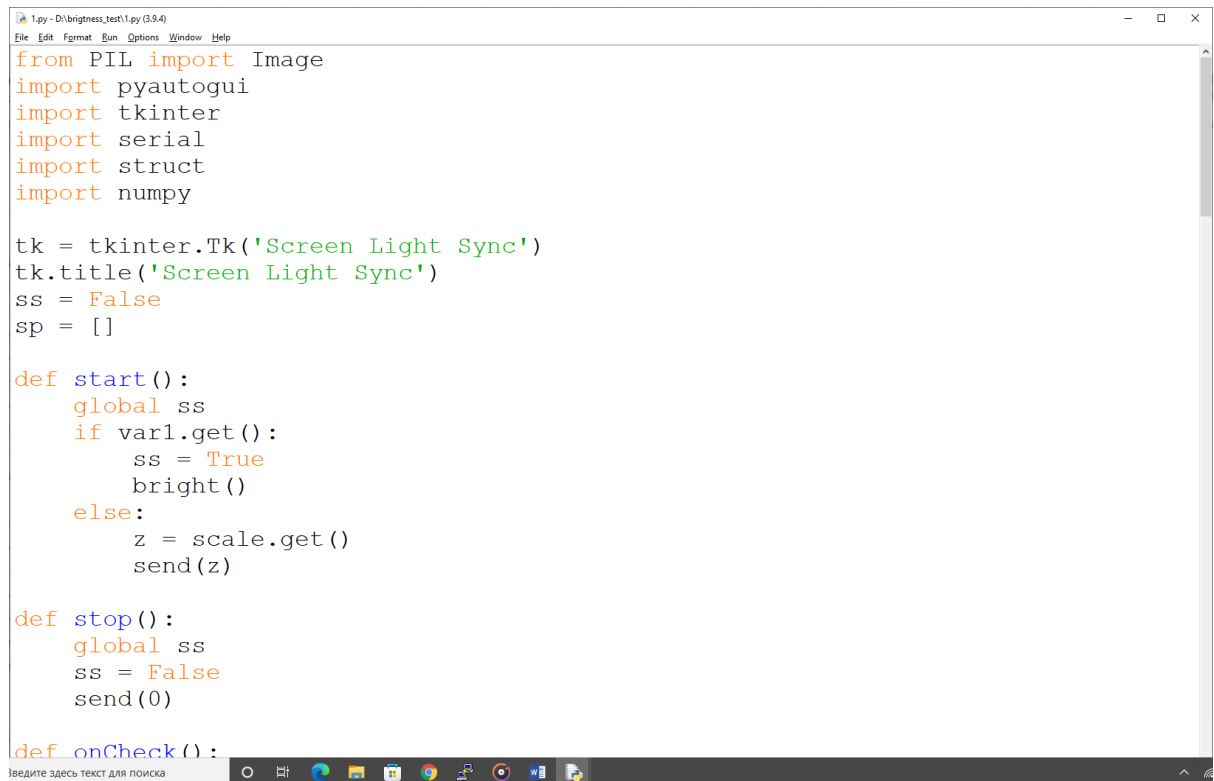


Рисунок 9. Интерфейс приложения (ручной режим)

В автоматическом режиме программа работает следующим образом: делается снимок экрана, по специальной формуле рассчитывается яркость в диапазоне от 0 до 255, и это значение отправляется через Serial-порт в плату Arduino (рис. 10).



```

1.py - D:\brightness_test1.py (3.9.4)
File Edit Format Run Options Window Help
from PIL import Image
import pyautogui
import tkinter
import serial
import struct
import numpy

tk = tkinter.Tk('Screen Light Sync')
tk.title('Screen Light Sync')
ss = False
sp = []

def start():
    global ss
    if var1.get():
        ss = True
        bright()
    else:
        z = scale.get()
        send(z)

def stop():
    global ss
    ss = False
    send(0)

def onCheck():

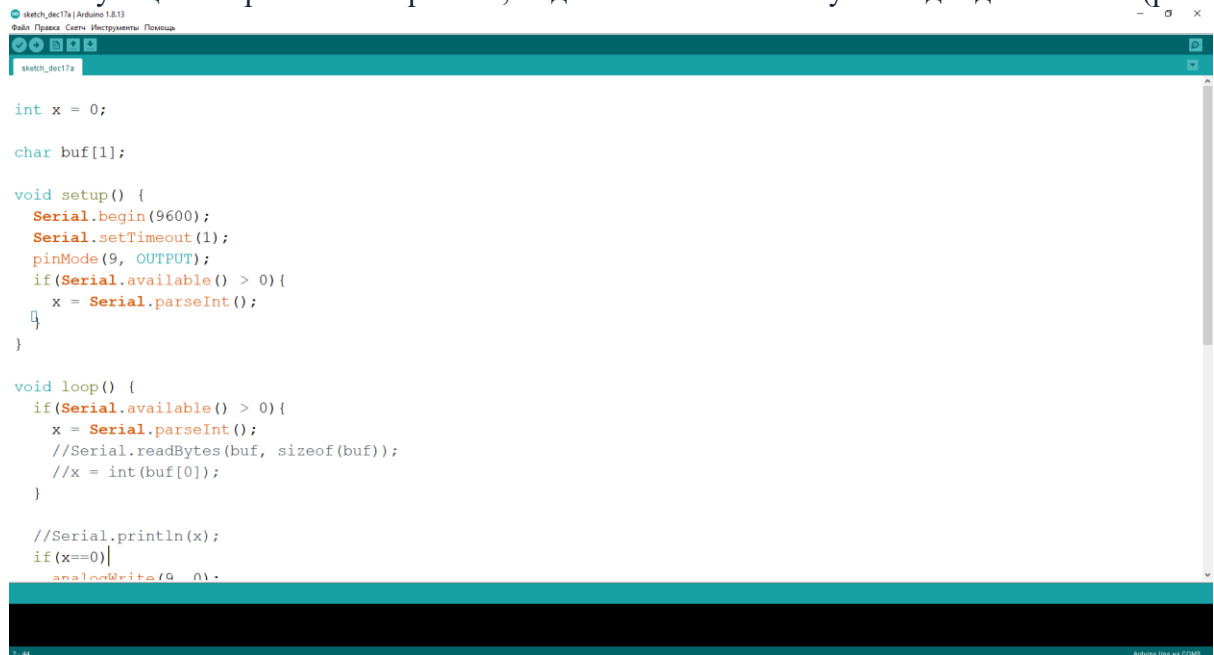
```

Рисунок 10. Фрагмент программы на Python

Листинг программы на Python приведен в приложении А.

## 2.4. Программная часть с использованием Arduino

Arduino считывает значение, посылаемое из программы на Python и подает соответствующее напряжение на разъем, подключенный к плюсу светодиодной ленты (рис. 11).



```

sketch_dec7a | Arduino 1.8.13
Файл Правка Скетч Инструменты Помощь
sketch_dec7a

int x = 0;

char buf[1];

void setup() {
    Serial.begin(9600);
    Serial.setTimeout(1);
    pinMode(9, OUTPUT);
    if (Serial.available() > 0) {
        x = Serial.parseInt();
    }
}

void loop() {
    if (Serial.available() > 0) {
        x = Serial.parseInt();
        //Serial.readBytes(buf, sizeof(buf));
        //x = int(buf[0]);
    }

    //Serial.println(x);
    if (x==0) {
        analogWrite(9, 0);
    }
}

```

Рисунок 11. Фрагмент программы на Arduino

Листинг программы для Arduino приведен в приложении В.

## 2.5. Тестирование Screen Light Sync

Программа работает быстро, значение яркости экрана передается со скоростью меньше половины секунды, задержка минимальная и не бросается в глаза.

Интерфейс программы управления интуитивно понятен, подключение и настройка устройства не требует каких-то специальных навыков и знаний.

В ходе испытаний была доказана работоспособность устройства Screen Light Sync и возможность его постоянного использования и массового производства (рис. 12).

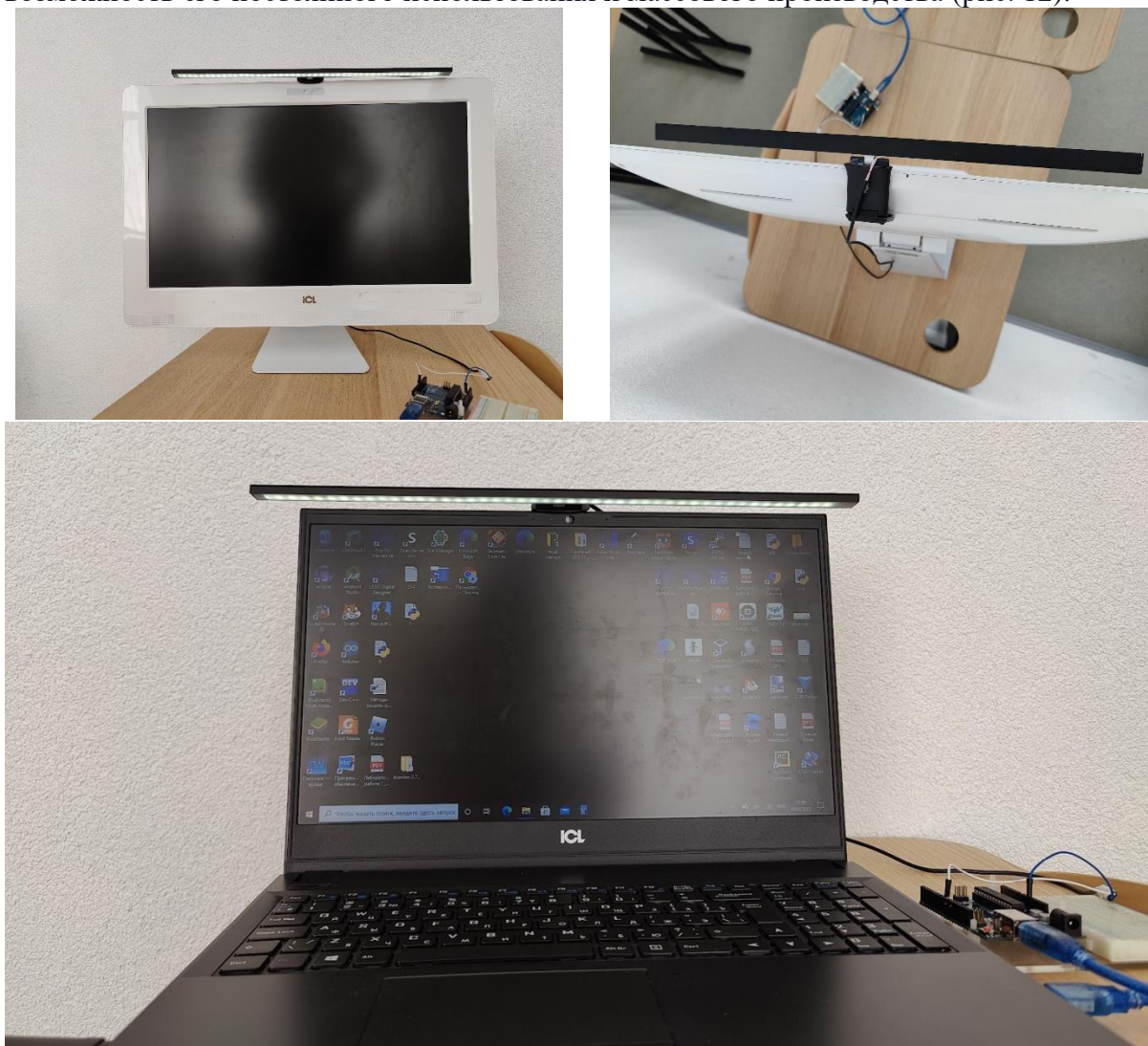


Рисунок 12. Пример тестирования устройства

## 2.6. Оценка себестоимости проекта

Учитывая стоимость всех комплектующих, итоговая себестоимость проекта не превышает 1500 руб. Если установить стоимость продажи устройства в районе 3000 – 3500 руб. (а это намного ниже, чем у аналогов), возможно получение реальной выгоды более чем 100%.

## Заключение

Пользование компьютером в темном помещении может негативно сказаться на зрении, так как в фокус берется лишь свет от монитора. Но и свет от обычной настольной лампы без регулирования мощности не всегда позволяет правильно подобрать режим для комфортной работы.

Поэтому было разработано устройство Screen Light Sync, которое адаптирует освещенность рабочего пространства под яркость экрана. Кроме автоматического предусмотрен также ручной режим управления яркостью подсветки, что делает устройство универсальным в применении.

Перспективы проекта:

- создание приложения для управления яркостью подсветки со смартфона;
- создания пульта управления для регулирования яркости подсветки в ручном режиме (поворотная шайба);
- замена кабеля USB на беспроводное соединения (bluetooth или WiFi).

### Список литературы

1. Почему нельзя долго смотреть телевизор в темноте: [Электронный ресурс]. URL: <https://proglazki.ru/interesnoe/pochemu-nelzya-dolgo-smotret-televizor-v-temnote/>. (Дата обращения: 12.10.2022)
2. Самые маленькие Arduino для ваших мини-проектов + примеры самих проектов: [Электронный ресурс]. URL: <https://habr.com/ru/company/selectel/blog/529998/>. (Дата обращения: 17.10.2022)
3. Язык программирования Python: сферы применения, методы и этапы изучения: [Электронный ресурс]. URL: <https://gb.ru/blog/yazyk-programmirovaniya-python/>. (Дата обращения: 10.11.2022)
4. СОМ-порт (последовательный порт): [Электронный ресурс]. URL: <https://www.alterbit.ru/glossary168.html>. (Дата обращения: 15.11.2022)

## Листинг программы на Python

```
from PIL import Image
import pyautogui
import tkinter
import serial
import struct
import numpy

tk = tkinter.Tk()
tk.title('Screen Light Sync')
ss = False
sp = []

def start():
    global ss
    if var1.get():
        ss = True
        bright()
    else:
        z = scale.get()
        send(z)

def stop():
    global ss
    ss = False
    send(0)

def onCheck():
    global ss
    if var1.get():
        ss = True
        bright()
        scale['state']=tkinter.DISABLED

    else:
        ss = False
        scale['state']=tkinter.NORMAL

def onScale(val):
    z = scale.get()
    send(z)

var1 = tkinter.BooleanVar()
var1.set(1)
ch1 = tkinter.Checkbutton(tk, text="AUTO", variable=var1, onvalue=1, offvalue=0,
command=onCheck)
ch1.pack()

scale = tkinter.Scale(tk, length=200, from_=0, to_=255, command=onScale,
orient="horizontal", state=tkinter.DISABLED)
scale.set(50)
scale.pack()

b1 = tkinter.Button(text='ON', width=30, height=5, command=start)
b1.pack()

b2 = tkinter.Button(text='OFF', width=30, height=5, command=stop)
b2.pack()
```

```

port = serial.Serial(port='COM3', baudrate=9600)
print(port)

def send(w):
    if 0 <= w <= 25:
        z = 0
    elif 26 <= w <= 50:
        z = 1
    elif 51 <= w <= 75:
        z = 2
    elif 76 <= w <= 100:
        z = 3
    elif 101 <= w <= 125:
        z = 4
    elif 126 <= w <= 150:
        z = 5
    elif 151 <= w <= 175:
        z = 6
    elif 176 <= w <= 200:
        z = 7
    elif 201 <= w <= 225:
        z = 8
    elif 226 <= w <= 255:
        z = 9
    if sp:
        if z != sp[0]:
            if port:
                port.write(bytes(str(z), 'utf-8'))
                print(bytes(str(z), 'utf-8'))
            else:
                print('port error')
            sp[0] = z
    else:
        sp.append(z)
        if port:
            port.write(bytes(str(z), 'utf-8'))
            print(bytes(str(z), 'utf-8'))
        else:
            print('port error')

def bright():
    screen = pyautogui.screenshot('l111.png')
    im = Image.open('l111.png')
    x, y = im.size
    ar = numpy.asarray(im)
    ar = ar.reshape(x * y, 3)
    #print(ar)

    ar1 = ar[:, 0] * 0.3
    ar2 = ar[:, 1] * 0.59
    ar3 = ar[:, 2] * 0.11
    s = (ar1.sum() + ar2.sum() + ar3.sum())

    w = int(s / (x * y))
    send(w)

    if ss:
        tk.after(25, bright)

tk.mainloop()

```



**Листинг программы для Arduino**

```
int x = 0;

char buf[1];

void setup() {
  Serial.begin(9600);
  Serial.setTimeout(1);
  pinMode(9, OUTPUT);
  if(Serial.available() > 0){
    x = Serial.parseInt();
  }
}

void loop() {
  if(Serial.available() > 0){
    x = Serial.parseInt();
    //Serial.readBytes(buf, sizeof(buf));
    //x = int(buf[0]);
  }

  //Serial.println(x);
  if(x==0)
    analogWrite(9, 0);
  else if(x == 1)
    analogWrite(9, 50);
  else if(x == 2)
    analogWrite(9, 75);
  else if(x == 3)
    analogWrite(9, 100);
  else if(x == 4)
    analogWrite(9, 125);
  else if(x == 5)
    analogWrite(9, 150);
  else if(x == 6)
    analogWrite(9, 175);
  else if(x == 7)
    analogWrite(9, 200);
  else if(x == 8)
    analogWrite(9, 225);
  else if(x == 9)
    analogWrite(9, 500);
  // delay(100);
}
```

**Рецензия научного руководителя  
на исследовательскую работу  
«Screen Light Sync»  
(автор: Стулков Лукьян)**

**V открытого регионального конкурса исследовательских и  
проектных работ школьников «Высший пилотаж - Пенза» 2023**

Проблема регулирования яркости экрана является актуальной для многих современных устройств. Где-то ее научились решать достаточно эффективно (например, в смартфонах), а где-то решений нет до сих пор (или эти решения сильно повышают стоимость устройства). Особенно остро эта проблема возникает в темное время суток.

В своей работе автор предлагает простое, надежное и эффективное решение проблемы автоматической регулировки яркости экрана любого монитора. Основой устройства является светодиодная лампа собственного дизайна с универсальным креплением. Устройство подключается к компьютеру через порт USB, что обеспечивает питание и передачу управляющих сигналов. Непосредственное управление устройством осуществляет плата Arduino, встроенная в лампу. Она принимает управляющие сигналы от специально разработанной программы, позволяющей использовать устройство как в ручном, так и в автоматическом режиме. В программе для ПК реализован уникальный алгоритм анализа яркости экрана.

Отличительными особенностями устройства Screen Light Sync (по сравнению с аналогами) являются: низкая стоимость, универсальность применения (ПК, моноблоки, ноутбуки), удобство использования (минимум кабелей), автоматический и ручной режим работы.

Практическая значимость проекта велика, так как устройство Screen Light Sync позволит обеспечить более комфортную и безопасную работу за компьютером.

Научный руководитель



Ю.А. Трофимов