

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный
университет»



УТВЕРЖДАЮ

Декан ЭТФ

_____ А.С. Гудим

« ____ » _____ 2019 г.

СОГЛАСОВАНО

Заведующий кафедрой ПЭ

_____ Д.А. Киба

« ____ » _____ 2019 г.

«Умное зеркало»

Комплект конструкторской документации

Руководитель СКБ

Ю.С. Иванов

Подпись/дата

Ответственный исполнитель

Д.М. Грабарь

Подпись/дата

Комсомольск-на-Амуре 2019

Карточка проекта

Название	«Умное зеркало»
Тип проекта	<u>В рамках конкурса</u> (инициативный, по заказу, в рамках конкурса, учебная работа, другое)
Исполнители	<u>Грабарь Д.М. – 8ИНБ-1</u> ответственный исполнитель Балагурова А.Е. – 8ИНБ-1 Буровская О.М. – 8ИНБ-1
Срок реализации	<u>09.02.2019-10.04.2019</u> Месяц, год

Использованные материалы и компоненты

Наименование	Количество, шт.
Raspberry Pi 3B+	1
LED-лента	1
Тактовая кнопка	1
ЖК-дисплей	1
Камера Raspberry Pi	1
ИК-датчик	2
Резистор 10 кОм	1

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»



ЗАДАНИЕ
на разработку

Выдано студентам:

Грабарь Д.М. - 8ИНб-1, Балагурова А.Е. – 8ИНб-1, Буровская О.М. – 8ИНб-1

Название проекта:

«Умное зеркало» _____

Назначение:

Необычное зеркало, которое способно отображать время, информацию о погоде в регионе и новостях. _____

Область использования:

Изделие может применяться дома _____

Функциональное описание устройства

При приближении человека к зеркалу срабатывает камера, распознающая лицо человека при помощи специального файла, в котором расположены правила определения лица помещенного в микрокомпьютер, Raspberry Pi 3В используется библиотека «OpenCV». Подсветка включается только при обнаружении лица камерой, которая передает информацию на микрокомпьютер Raspberry Pi3 В и после посылается на светодиодную ленту. Помимо этого на поверхность зеркала выводятся время, данные о погоде и новости из интернета, который берутся из сервисов «ЯндексПогода», «GoogleНовости» и из самого микрокомпьютера. Также имеется кнопка для отключения светодиодной ленты зеркала. _____

Техническое описание устройства: _____

Зеркало управляется МК-типа Raspberry Pi 3B+. В корпусе изделия находится ЖК-дисплей, который подключен к микрокомпьютеру. Помимо этого на внешней стороне рамки зеркала находится светодиодная лента, которая управляется распознающей камерой, а также тактовая кнопка. При нажатии на кнопку отключается работа светодиодной ленты и камеры. При повторном нажатии на кнопку камера и светодиодная лента начинают работать. При появлении человека в области распознавания камера засекает его присутствие, включается подсветка зеркала. Время, данные о погоде и новости отображаются на поверхности зеркала постоянно. _____

Требования:

Зеркало должно быть безопасным, надежным, эстетичным и обеспечивать удобство использования. _____

Перечень графического материала:

1. Принципиальная схема изделия _____

2. Блок-схемы работы изделия _____

3. Чертежи изделия _____

4. Внешний вид изделия _____

Руководитель проекта

Ю.С. Иванов

Подпись/дата

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный
университет»



ПАСПОРТ

«Умное зеркало»

Руководитель СКБ

Ю.С. Иванов

Подпись/дата

Ответственный исполнитель

Д.М. Грабарь

Подпись/дата

Комсомольск-на-Амуре 2019

Содержание

1	Общие положения	3
1.1	Наименование изделия	3
1.2	Наименования документов, на основании которых ведется проектирование системы.....	3
1.3	Перечень организаций, участвующих в разработке системы	4
1.4	Сведения об использованных при проектировании нормативно-технических документах	4
2	Назначение и принцип действия	6
2.1	Назначение изделия	6
2.2	Области использования изделия	6
2.3	Принцип действия.....	6
3	Состав изделия и комплектность.....	7
4	Технические характеристики	8
4.1	Основные технические характеристики умного зеркала.....	8
5	Устройство и описание работы изделия.....	9
5.1	Устройство изделия	9
5.2	Описание работы изделия	10
6	Условия эксплуатации	11
6.1	Правила и особенности размещения изделия	11
6.2	Меры безопасности.....	12
6.3	Правила хранения и транспортирования.....	12
	ПРИЛОЖЕНИЕ А	13

					СКБЭТФ.2.ИП.010000ПП	Лист
						2
Изм.	Лист.	№ документа	Подп.	Дата.		

1 Общие положения

Настоящий паспорт является документом, предназначенным для ознакомления с основными техническими характеристиками, устройством, правилами установки и эксплуатации устройства «Умное зеркало» (далее «изделие»).

Паспорт входит в комплект поставки изделия. Прежде, чем пользоваться изделием, внимательно изучите правила обращения и порядок работы с ним. В связи с постоянной работой по усовершенствованию изделия, повышающей его надежность и улучшающей условия эксплуатации, в конструкцию могут быть внесены изменения, не отраженные в данном издании.

1.1 Наименование изделия

Полное наименование системы – «Умное зеркало» (УЗ).

1.2 Наименования документов, на основании которых ведется проектирование системы

Создание УЗ осуществляется на основании требований и положений следующих документов:

- задание на разработку.

					СКБЭТФ.2.ИП.010000ПП	Лист
						3
Изм.	Лист.	№ документа	Подп.	Дата.		

1.3 Перечень организаций, участвующих в разработке системы

Заказчиком создания АПК УЗ является Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет» (далее заказчик), находящийся по адресу: 681013, Хабаровский край, г. Комсомольск-на-Амуре, Ленина пр-кт., д. 27.

Исполнителями работ по созданию АПК УЗ являются Конструкторы студенческого конструкторского бюро электротехнического факультета (далее СКБ ЭТФ), студенты группы 8ИНб-1, Грабарь Даниил Михайлович, Балагурова Анастасия Евгеньевна, Буровская Ольга Максимовна.

1.4 Сведения об использованных при проектировании нормативно-технических документах

При проектировании использованы следующие нормативно-технические документы:

ГОСТ 2.001-2013. Единая система конструкторской документации. Общие положения.

ГОСТ 2.102-2013. Единая система конструкторской документации. Виды и комплектность конструкторских документов.

ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.

ГОСТ 2.610-2006. Единая система конструкторской документации. Правила выполнения эксплуатационных документов.

ГОСТ 2.004-88. Единая система конструкторской документации. Общие требования к выполнению конструкторских технологических документов на печатающих и графических устройствах вывода ЭВМ.

ГОСТ 2.051-2006. Единая система конструкторской документации. Электронные документы. Общие положения.

					СКБЭТФ.2.ИП.010000ПП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		4

ГОСТ 2.052-2006. Единая система конструкторской документации.
Электронная модель изделия. Общие положения.

ГОСТ 2.601-2013. Единая система конструкторской документации.
Эксплуатационные документы.

					СКБЭТФ.2.ИП.010000ПП	Лист
						5
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		

2 Назначение и принцип действия

2.1 Назначение изделия

Умное зеркало – необычное зеркало, которое способно отображать время, информацию о погоде в регионе и новостях, а также зажигать светодиодную ленту при обнаружении человека.

В состав изделия входят: зеркало, блок питания.

2.2 Области использования изделия

Изделие может применяться в любом помещении где требуется зеркало, торговые центры, кафе, туалетные комнаты, дома и д.р.

2.3 Принцип действия

При приближении человека к зеркалу срабатывает камера, распознающая лицо человека. Включается подсветка, и на поверхность зеркала выводятся время, данные о погоде и новости из интернета. Также имеется кнопка для отключения подсветки зеркала.

					СКБЭТФ.2.ИП.010000ПП	Лист
						6
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		

3 Состав изделия и комплектность

В комплект поставки входит:

- Зеркало.
- Блок питания.
- Паспорт.

					СКБЭТФ.2.ИП.010000ПП	Лист
						7
Изм.	Лист.	№ документа	Подп.	Дата.		

4 Технические характеристики

4.1 Основные технические характеристики умного зеркала

Основные технические характеристики умного зеркала приведены в таблице 1.

Таблица 1 – Технические характеристики блока мишеней

Наименование параметра	Значение
Тип сенсора	Камера
Интерфейсы	Micro USB
Питание, В	12, 5
Область считывания, м	1-5
Тип индикации	ЖК дисплей
Кнопка выключения	нет
Длина шнура питания, м	3
Габариты, мм	480*325*100
Масса нетто, кг	3

5 Устройство и описание работы изделия

5.1 Устройство изделия

Корпус изделия – это деревянная рамка со встроенным зеркалом, дисплеем и микрокомпьютером.

Управление зеркалом выполняется микрокомпьютером Raspberry PI 3В. Зеркало имеет кнопку включения и выключения светодиодной ленты. Помимо этого присутствует камера, фиксирующая лицо находящегося перед ним человека, зажигается светодиодная лента. Напряжение питания светодиодной ленты 12 В, напряжение питания Raspberry PI 3В осуществляется подачей питания 5 В от сети. Принципиальная схема зеркала представлена в Приложении А.

					СКБЭТФ.2.ИП.010000ПП	Лист
						9
Изм.	Лист.	№ документа	Подп.	Дата.		

5.2 Описание работы изделия

Перед началом использования изделия необходимо установить его на стену на высоте человеческих глаз. При установке зеркала необходимо защитить его от попадания воды.

Подключить питание от сети переменного тока через адаптер. Расстояние от зеркала до человека должно быть не менее 1 метра.

Камера направлена в сторону человека. В Raspberry PI 3B располагается файл с каскадами лица, программа анализирует их и на основе этого определяет, в какой охваченной камерой области находится человек. Помимо камеры в корпус зеркала встроен ЖК-дисплей, который показывает дату, время, погоду и последние новости. Данные о погоде берутся с сервиса «ЯндексПогода», новостях с сервиса «GoogleНовости», дату и время не посредственно с микрокомпьютера. Все данные обновляются с промежутком в 10 мин. Блок-схемы работы управляющих программ приведены в Приложении А.

					СКБЭТФ.2.ИП.010000ПП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		10

6 Условия эксплуатации

Изделие выпускается в климатическом исполнении УХЛ 4.2 по ГОСТ 15150-69 и предназначен для использования в стационарных условиях в закрытых помещениях при соответствующих климатических условиях:

- интервал температур от +10 до +35 °С;
- относительная влажность воздуха до 80 % при температуре +25 °С;
- высота над уровнем моря не более 2000 м;
- атмосферное давление от 86,6 до 106 кПа (от 650 до 800 мм рт. ст.).

В помещении, где используется изделие не должно возникать условий для конденсации влаги (выпадения росы). Изделие является электронным прибором, требующим бережного обращения.

Для обеспечения безотказной работы, сохранения точности и его сбережения необходимо соблюдать следующие правила:

- изучить паспорт, прежде чем приступить к работе с изделием;
- предохранять изделие от ударов и повреждений;
- не прикасаться руками к оптическим элементам;
- не допускать самостоятельную разборку изделия.

6.1 Правила и особенности размещения изделия

Для распознавания лиц зеркало должно находиться на расстоянии 1-5м от человека. Допускается эксплуатация изделия в условиях нормальной и ограниченной освещенности. Изделие должно быть расположено на расстоянии не менее 1 м от нагревательных приборов.

ВНИМАНИЕ! При эксплуатации изделия запрещается проводить самостоятельно какие-то либо работы по извлечению и установке внутренних компонентов изделия.

					СКБЭТФ.2.ИП.010000ПП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		11

6.2 Меры безопасности

Необходимо соблюдать требования техники безопасности и следующие меры предосторожности:

- не оставлять изделие включенным без наблюдения;
- после транспортировки в холодное время года изделие необходимо выдержать при комнатной температуре не менее двух часов;
- внутренние осмотры и ремонт изделия должны производиться только квалифицированными специалистами;
- не устанавливайте изделие на неустойчивой подставке, стойке или ненадежном кронштейне.

6.3 Правила хранения и транспортирования

Транспортирование изделия в упакованном виде может производиться железнодорожным, автомобильным (в закрытых транспортных средствах), воздушным, речным и морским видами транспорта в соответствии с правилами перевозок грузов, действующих на транспорт данного вида. Условия транспортирования изделия по части воздействия климатических факторов должны соответствовать группе 5 по ГОСТ 15150.

После транспортирования изделие должно быть выдержано не менее 2 часов в транспортной таре при температуре $20\pm 5^{\circ}\text{C}$ и относительной влажности воздуха не более 80%.

Распакованное изделие должно храниться в отапливаемом и вентилируемом чистом помещении при температуре от $+5$ до $+40^{\circ}\text{C}$ и относительной влажности воздуха не более 60 %. При температуре ниже 25°C допускается увеличение относительной влажности до 80 %. Воздух в помещении не должен содержать примесей, вызывающих коррозию металлов, налеты на поверхностях оптических деталей.

					СКБЭТФ.2.ИП.010000ПП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		12

ПРИЛОЖЕНИЕ А

(обязательное)

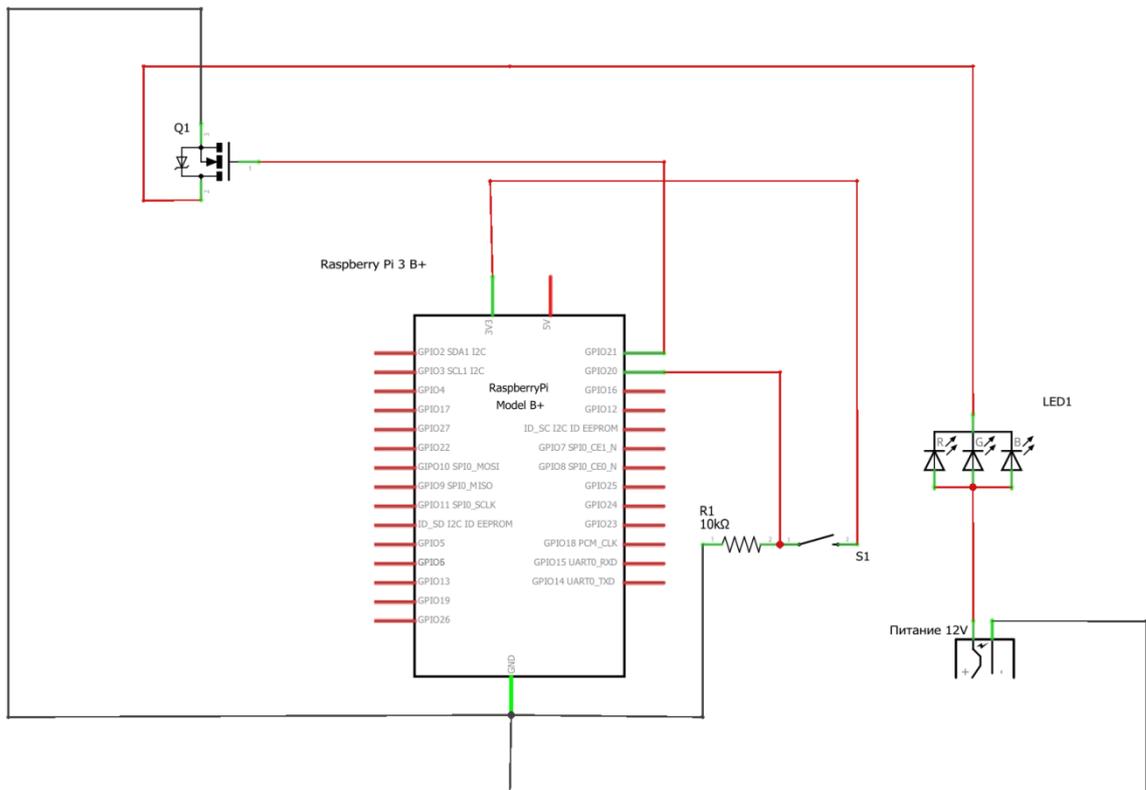


Рисунок А1 – Принципиальная схема электрической цепи

Изм.	Лист.	№ документа	Подп.	Дата.

СКБЭТФ.2.ИП.01000033

Лист

13

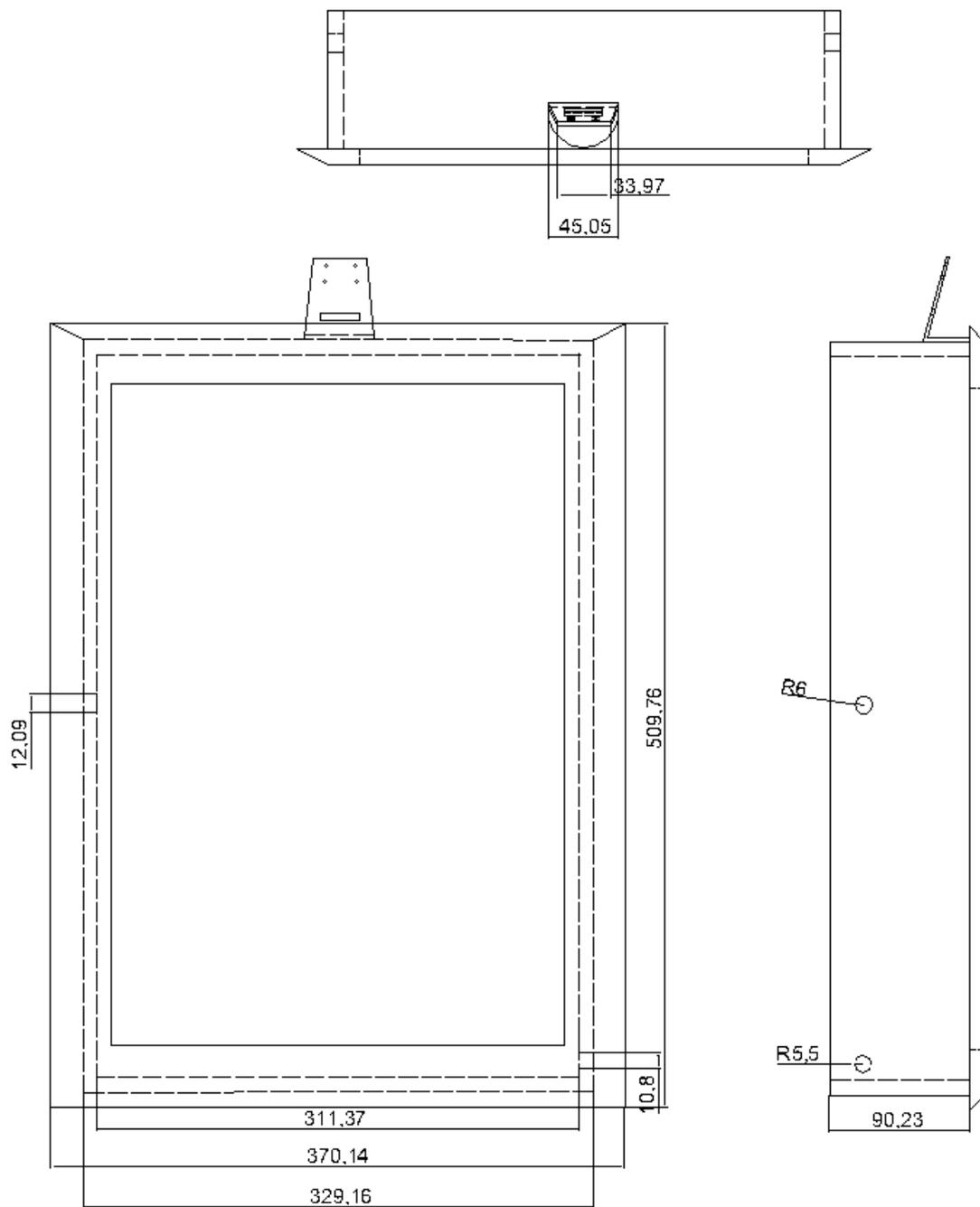


Рисунок А2 – Принципиальная схема изделия

					СКБЭТФ.2.ИП.010000Э0	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		15

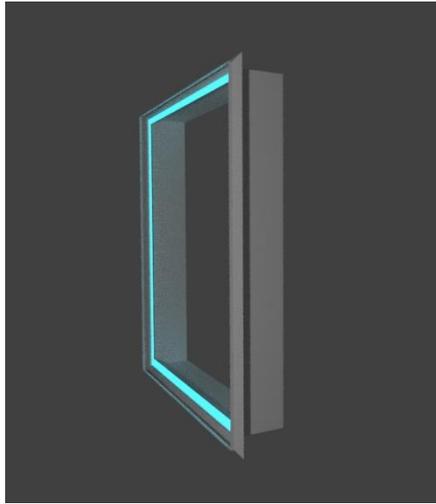


Рисунок А3 – Трехмерная модель изделия



Рисунок А4 – Внешний вид изделия без подсветки

					СКБЭТФ.2.ИП.010000ИЛ	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		16

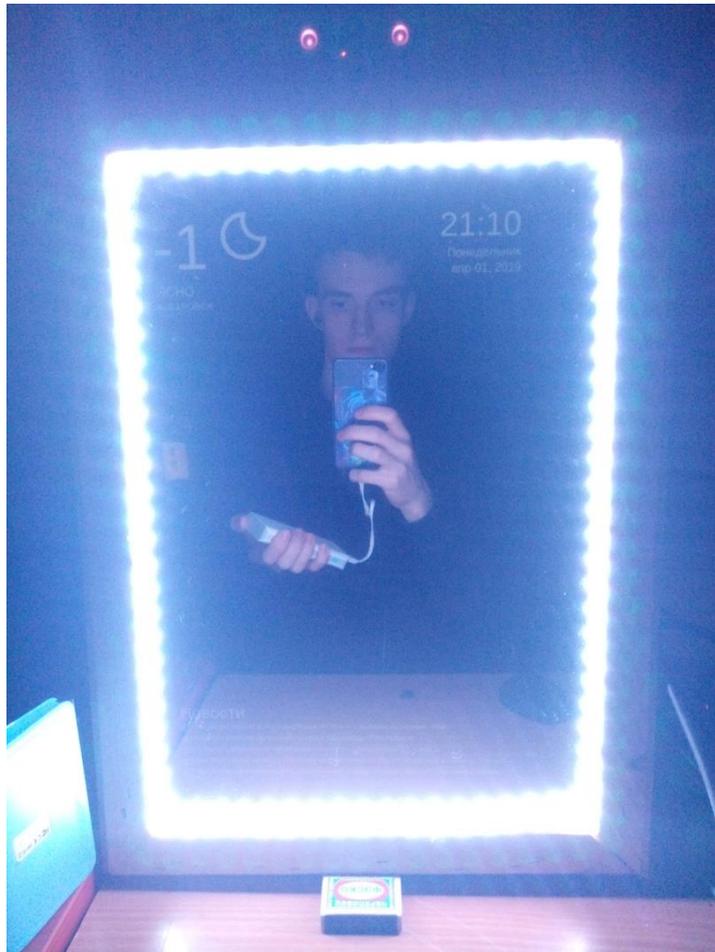


Рисунок А5 – Внешний вид изделия с подсветкой



Рисунок А6 – Внешний вид изделия с задней стороны

					СКБЭТФ.2.ИП.010000ИЛ	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		17

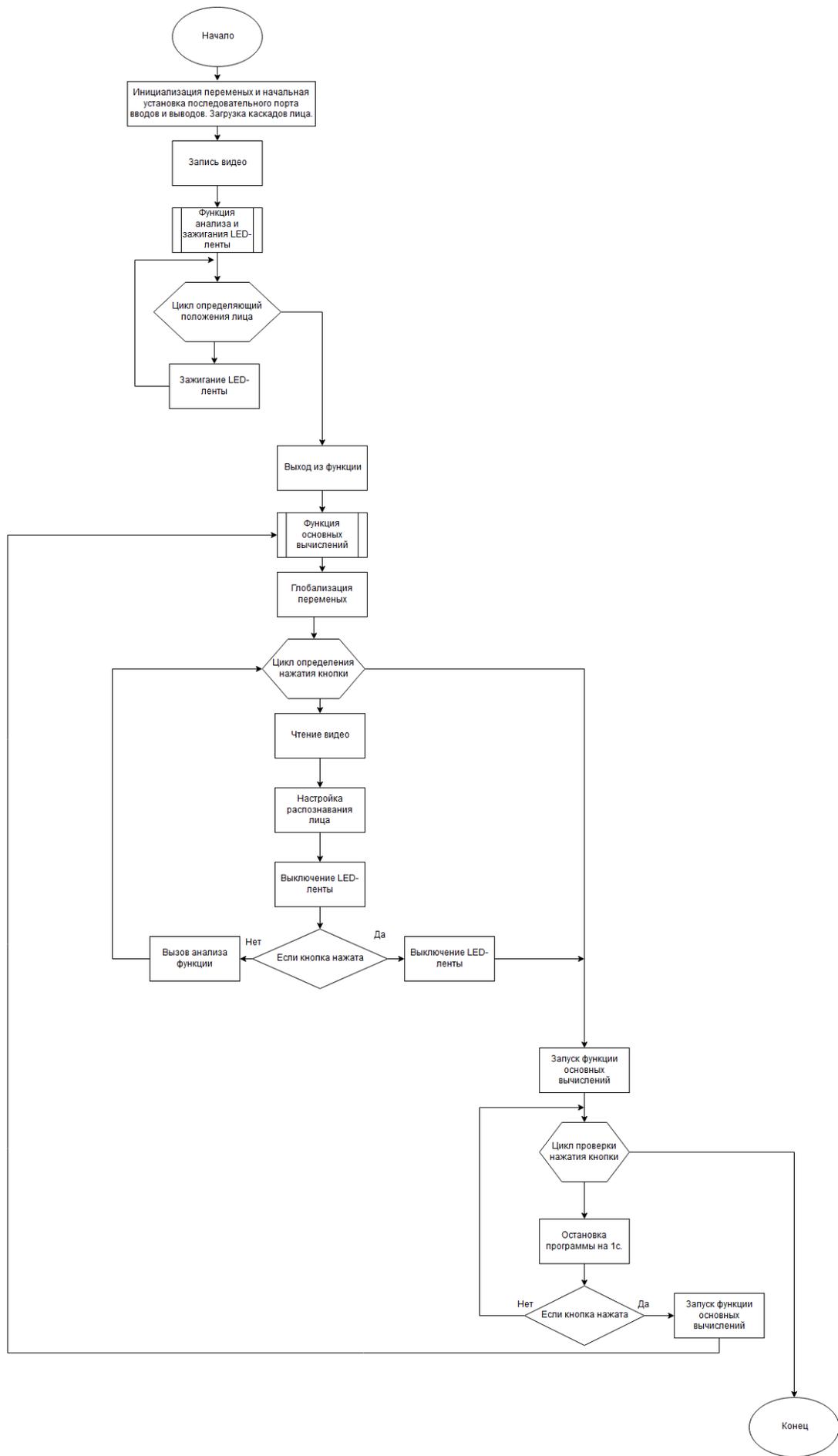


Рисунок А7 – Блок-схема управляющей программы определения лица

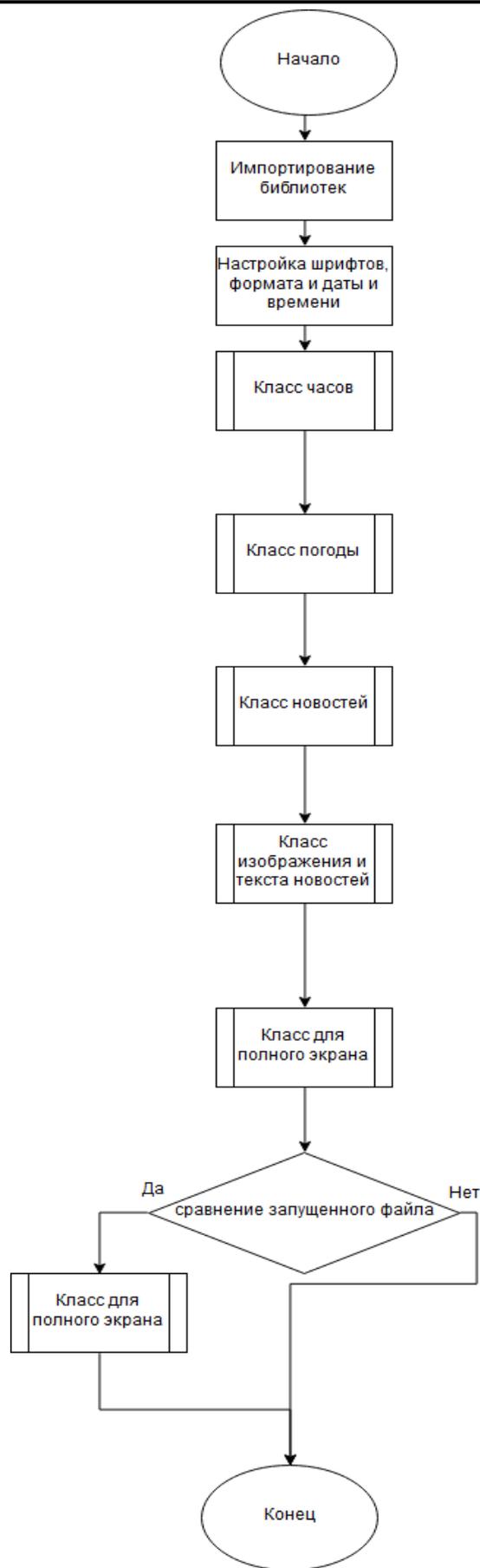


Рисунок А8 – Блок-схема управляющей программы зеркала

Листинг управляющей программы определения лица

```
import numpy as np
import cv2
import RPi.GPIO as GPIO
import time

white_led = 21 #пин ленты
key = 20 #пин кнопки

# Настройка GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(white_led, GPIO.OUT)
GPIO.setup(key, GPIO.IN)
GPIO.output(white_led, GPIO.LOW)

faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')# Файл с
характеристиками определения лица

cap = cv2.VideoCapture(0)#Запись видео
cap.set(3,640) # ширина
cap.set(4,480) # высота

# анализ лица и зажигания свет. ленты
def analyz():
    for (x,y,w,h) in faces:
        GPIO.output(white_led, GPIO.HIGH)

def main():
    global white_led, key, cap, faces
    while True:
        ret, img = cap.read() #Чтение видео
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        #Настройка распознавания лица
        faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor=1.2,
```

					СКБЭТФ.2.ИП.010000ЛП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		20

```

minNeighbors=5
,
minSize=(20, 20),
)

GPIO.output(white_led, GPIO.LOW)#

#Проверка на нажатие кнопки. Если кнопка нажата, то выходим из цикла иначе запустить
анализ лица
if GPIO.input(key) == True:
#print("key down")
GPIO.output(white_led, GPIO.LOW)
break
else:
analyz()

main()#Запуск основы проекта
#Проверка на возвращение на определение лица
while True:
time.sleep(1)
if GPIO.input(key) == True:
#print("yes")
main()

```

					СКБЭТФ.2.ИП.010000ЛП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		21

Листинг управляющей программы зеркала

```
from tkinter import *
import locale
import threading
import time
import requests
import traceback
import feedparser

from PIL import Image, ImageTk
from contextlib import contextmanager
from bs4 import BeautifulSoup

LOCALE_LOCK = threading.Lock()

ui_locale = " # например 'fr_FR' для французского, " по умолчанию
date_format = "%b %d, %Y" # проверьте Python Doc для strftime () для
xlarge_text_size = 94
large_text_size = 48
medium_text_size = 28
small_text_size = 18

@contextmanager
def setlocale(name): #функция доказательства потока для работы с локалью
    with LOCALE_LOCK:
        saved = locale.setlocale(locale.LC_ALL)
        try:
            yield locale.setlocale(locale.LC_ALL, name)
        finally:
            locale.setlocale(locale.LC_ALL, saved)

class Clock(Frame):
    def __init__(self, parent, *args, **kwargs):
        Frame.__init__(self, parent, bg='black')
        # инициализировать метку времени
        self.time1 = "
```

					СКБЭТФ.2.ИП.010000ЛП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		22

```

self.timeLbl = Label(self, font=('Helvetica', large_text_size), fg="white", bg="black")
self.timeLbl.pack(side=TOP, anchor=E)

# инициализировать день недели
self.day_of_week1 = ""

self.dayOWLbl = Label(self, text=self.day_of_week1, font=('Helvetica', small_text_size),
fg="white", bg="black")
self.dayOWLbl.pack(side=TOP, anchor=E)

# инициализировать метку даты
self.date1 = ""

self.dateLbl = Label(self, text=self.date1, font=('Helvetica', small_text_size), fg="white",
bg="black")
self.dateLbl.pack(side=TOP, anchor=E)

self.tick()

def tick(self):
global time2
with setlocale(ui_locale):
time2 = time.strftime('%H:%M') #часы 24 формат

day_of_week2 = time.strftime('%A')
date2 = time.strftime(date_format)

# если временная строка изменилась, обновите ее
if time2 != self.time1:
self.time1 = time2
self.timeLbl.config(text=time2)

if day_of_week2 != self.day_of_week1:
self.day_of_week1 = day_of_week2
self.dayOWLbl.config(text=day_of_week2)

if date2 != self.date1:
self.date1 = date2
self.dateLbl.config(text=date2)

# Вазывает себя каждые 200 миллисекунд
# обновить отображение времени по мере необходимости
# может использовать > 200 мс, но отображение становится прерывистым
self.timeLbl.after(200, self.tick)

class Weather(Frame):
def __init__(self, parent, *args, **kwargs):

```

					СКБЭТФ.2.ИП.010000ЛП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		23

```

Frame.__init__(self, parent, bg='black')
self.temperature = ""
self.location = ""
self.currently = ""
self.icon = ""
self.degreeFrm = Frame(self, bg='black')
self.degreeFrm.pack(side=TOP, anchor=W)
self.temperatureLbl = Label(self.degreeFrm, font=('Helvetica', xlarge_text_size), fg="white",
bg="black")
self.temperatureLbl.pack(side=LEFT, anchor=N)
self.iconLbl = Label(self.degreeFrm, bg="black")
self.iconLbl.pack(side=LEFT, anchor=N, padx=20)
self.currentlyLbl = Label(self, font=('Helvetica 20'), fg="white", bg="black")
self.currentlyLbl.pack(side=TOP, anchor=W)
self.locationLbl = Label(self, font=('Helvetica 14'), fg="white", bg="black")
self.locationLbl.pack(side=TOP, anchor=W)
self.get_weather()

def get_weather(self):
r = requests.get('https://export.yandex.ru/bar/reginfo.xml?region=76') #Парсим сайт погоды,
чтобы поменять регион подставить вместо "76" свой id Города
soup = BeautifulSoup(r.text, 'lxml') #'Парсим сайт погоды'...
#Берем температуру
temperature2 = 'temperature'
find_temperature = soup.find(temperature2)
self.temperature = find_temperature.text
#Место расположения
location2 = 'title'
find_location = soup.find(location2)
self.location = find_location.text
#Текущая погода
currently2 = 'weather_type'
find_currently = soup.find(currently2)
self.currently = find_currently.text
#Вывод на экран
self.temperatureLbl.config(text=self.temperature)

```

					СКБЭТФ.2.ИП.010000ЛП	Лист
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		24

```

self.currentlyLbl.config(text=self.currently.upper())
self.locationLbl.config(text=self.location.upper())

if self.currently == 'ясно':
if time2 < '19:30' and time2 > '08:00':
image = Image.open('img/Sun.png')
image = image.resize((100, 100), Image.ANTIALIAS)
image = image.convert('RGB')
photo = ImageTk.PhotoImage(image)
self.iconLbl.config(image=photo)
self.iconLbl.image = photo
else:
image = Image.open('img/Moon.png')
image = image.resize((100, 100), Image.ANTIALIAS)
image = image.convert('RGB')
photo = ImageTk.PhotoImage(image)
self.iconLbl.config(image=photo)
self.iconLbl.image = photo

elif self.currently == 'облачно':
image = Image.open('img/Cloud.png')
image = image.resize((100, 100), Image.ANTIALIAS)
image = image.convert('RGB')
photo = ImageTk.PhotoImage(image)
self.iconLbl.config(image=photo)
self.iconLbl.image = photo

elif self.currently == 'переменная облачность' or self.currently == 'облачно с прояснениями':
if time2 < '19:30' and time2 > '08:00':
image = Image.open('img/PartlySunny.png')
image = image.resize((100, 100), Image.ANTIALIAS)
image = image.convert('RGB')
photo = ImageTk.PhotoImage(image)
self.iconLbl.config(image=photo)
self.iconLbl.image = photo
else:
image = Image.open('img/PartlyMoon.png')

```

					СКБЭТФ.2.ИП.010000ЛП	Лист
<i>Изм.</i>	<i>Лист.</i>	<i>№ документа</i>	<i>Подп.</i>	<i>Дата.</i>		25

```

image = image.resize((100, 100), Image.ANTIALIAS)
image = image.convert('RGB')
photo = ImageTk.PhotoImage(image)
self.iconLbl.config(image=photo)
self.iconLbl.image = photo

elif self.currently == 'дождь':
image = Image.open('img/Rain.png')
image = image.resize((100, 100), Image.ANTIALIAS)
image = image.convert('RGB')
photo = ImageTk.PhotoImage(image)
self.iconLbl.config(image=photo)
self.iconLbl.image = photo

elif self.currently == 'снег' or self.currently == 'облачно, небольшой снег' or self.currently ==
'облачно, снег' :
image = Image.open('img/Snow.png')
image = image.resize((100, 100), Image.ANTIALIAS)
image = image.convert('RGB')
photo = ImageTk.PhotoImage(image)
self.iconLbl.config(image=photo)
self.iconLbl.image = photo

self.after(1000, self.get_weather)

class News(Frame):
def __init__(self, parent, *args, **kwargs):
Frame.__init__(self, parent, *args, **kwargs)
self.config(bg='black')
self.title = 'Новости' # Надпись "Новости"
self.newsLbl = Label(self, text=self.title, font=('Helvetica', medium_text_size), fg="white",
bg="black")
self.newsLbl.pack(side=TOP, anchor=W)
self.headlinesContainer = Frame(self, bg="black")
self.headlinesContainer.pack(side=TOP)
self.get_headlines()

def get_headlines(self):

```

					СКБЭТФ.2.ИП.010000ЛП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		26

```

try:
for widget in self.headlinesContainer.winfo_children():
widget.destroy()
headlines_url = "https://news.google.com/rss?hl=ru&gl=RU&ceid=RU:ru"
feed = feedparser.parse(headlines_url)
for post in feed.entries[0:5]:
headline = NewsHeadline(self.headlinesContainer, post.title)
headline.pack(side=TOP, anchor=W)
except Exception as e:
traceback.print_exc()

self.newsLbl = Label(self, text='Отсутствует подключение к интернету', font=('Helvetica 40'),
fg="white", bg="black")
self.newsLbl.pack(side=TOP, anchor=W)

self.after(600000, self.get_headlines)

class NewsHeadline(Frame):
def __init__(self, parent, event_name=""):
Frame.__init__(self, parent, bg='black')

image = Image.open("img/Newspaper.png")
image = image.resize((25, 25), Image.ANTIALIAS)
image = image.convert('RGB')
photo = ImageTk.PhotoImage(image)

self.iconLbl = Label(self, bg='black', image=photo)
self.iconLbl.image = photo
self.iconLbl.pack(side=LEFT, anchor=N)

self.eventName = event_name
self.eventNameLbl = Label(self, text=self.eventName, font=('Helvetica', small_text_size),
fg="white", bg="black")
self.eventNameLbl.pack(side=LEFT, anchor=N)

class FullscreenWindow:

def __init__(self):
self.tk = Tk()
self.tk.configure(background='black')

```

					СКБЭТФ.2.ИП.010000ЛП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		27

```

self.topFrame = Frame(self.tk, background = 'black')
self.bottomFrame = Frame(self.tk, background = 'black')
self.topFrame.pack(side = TOP, fill=BOTH, expand = YES)
self.bottomFrame.pack(side = BOTTOM, fill=BOTH, expand = YES)
self.state = False
self.tk.bind("<Return>", self.toggle_fullscreen)
self.tk.bind("<Escape>", self.end_fullscreen)
# часы
self.clock = Clock(self.topFrame)
self.clock.pack(side=RIGHT, anchor=N, padx=100, pady=60)
# погода
self.weather = Weather(self.topFrame)
self.weather.pack(side=LEFT, anchor=N, padx=100, pady=60)
# новости
self.news = News(self.bottomFrame)
self.news.pack(side=LEFT, anchor=S, padx=100, pady=60)

def toggle_fullscreen(self, event=None):
self.state = not self.state
self.tk.attributes("-fullscreen", self.state)
return "break"

def end_fullscreen(self, event=None):
self.state = False
self.tk.attributes("-fullscreen", False)
return "break"

if __name__ == '__main__':
w = FullscreenWindow()
w.tk.mainloop()

```

					СКБЭТФ.2.ИП.010000ЛП	Лист
Изм.	Лист.	№ документа	Подп.	Дата.		28

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный
университет»

УТВЕРЖДАЮ

Декан ЭТФ

_____ А.С. Гудим

« ____ » _____ 2019 г.

СОГЛАСОВАНО

Заведующий кафедрой ПЭ

_____ Д.А. Киба

« ____ » _____ 2019 г.

**АКТ о приемке в эксплуатацию
«Умное зеркало»**

г. Комсомольск-на-Амуре

« ____ » _____ 2019 г.

Комиссия в составе представителей:

заказчика Ю.С. Иванов – руководитель СКБ ЭТФ, Д.А. Киба – Заведующий кафедрой ПЭ,

исполнителя Д.М.Грабарь – 8ИНБ-1, А.Е.Балагурова – 8ИНБ-1, О.М.Буровская – 8ИНБ-1.

составила акт о нижеследующем:

«Исполнитель» передает аппаратно-программный комплекс «Умное зеркало», в составе:

Оборудование, в составе:

- Зеркало;
- Блок питания.

Программное обеспечение, в том числе:

- Рабочие программы управления изделием.

Эксплуатационная документация:

- Паспорт изделия

Аппаратно-программный комплекс «Умное зеркало» прошел опытную эксплуатацию с « » _____ по « » _____ 2019г. и признан годным к эксплуатации. Были протестированы все режимы функционирования, отказы системы, а также аварийные отключения по вине системы не наблюдались.

Руководитель СКБ

Ответственный исполнитель

_____/ Ю.С. Иванов /

_____/ Д.М. Грабарь/

