ГБОУ Школа № 1538

Кафедра инженерной предпрофессиональной подготовки

БЮДЖЕТНЫЕ ПОРТАТИВНЫЕ ПРИБОРЫ-ВЕРТУШКИ С POV-ДИСПЛЕЕМ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ДЕТЕЙ ПРИ ПЕРЕХОДЕ НЕОСВЕЩЕННЫХ УЧАСТКОВ АВТОМОБИЛЬНОЙ ДОРОГИ

Автор проекта: ученик ГБОУ школы № 1538:
Иванов Михаил Сергеевич, 8Е класс Иванов

Руководитель проекта: доктор технических наук, профессор, почетный работник высшего профессионального образования РФ, педагог дополнительного образования высшей категории ГБОУ школы №1538
Фролов Михаил Ильич

Директор Мухина И.Б.

МОСКВА

2019

ОТЗЫВ

научного руководителя на проектную работу Иванова Михаила Сергеевича «Бюджетные портативные приборы-вертушки с POV-дисплеем для обеспечения безопасности детей при переходе неосвещенных участков автомобильной дороги»

За время работы над проектом Иванов Михаил Сергеевич проанализировал существующие средства по обеспечению безопасности детей при переходе неосвещенных участков автомобильной дороги и по разработке POV-дисплея на основе составленного им списка источников информации.

В ходе работы над проектом Иванов М.С. освоил программирование в среде Arduino IDE, разработал и изготовил недорогие автономные и привлекательные для своих сверстников портативные световые приборы-вертушки с POV-дисплеем для обеспечения безопасности детей в темное время суток при переходе неосвещенных участков автомобильной дороги на основе плат Arduino Uno и Arduino Nano, а также микроконтроллеров ATmega 328P-PU и ATTiny 85-20PU.

Эти оригинальные приборы могут быть также использованы для образовательных целей – изучения стробоскопического эффекта на уроках физики, программирования в среде Arduino IDE различных световых надписей и изображений на уроках информатики и робототехники. Для проведения лабораторных испытаний этого прибора, Иванов М.С. запроектировал и изготовил стенд на основе робототехнического набора Lego Mindstorms Education EV3.

Проведенные Ивановым М.С. полевые испытания прибора-вертушки с POV-дисплеем на неосвещенной трассе в темное время суток показали надежность, а также его видимость с расстояния в 35 м, что соответствует тормозному пути автомобиля, движущегося со скоростью V = 60 км/час.

За время работы над проектом Иванов М.С. проявил самостоятельность, смекалку и умение достигать поставленной перед собой цели.

доктор технических наук, профессор,
почетный работник ВПО РФ М.И. Фролов

20.02.2019 г.

Содержание

Введение 3

1. Анализ существующих решений 4

2. Описание разработанного решения 4

3. Результаты работы 13

Приложение 1. Список источников информации 15

Приложение 2. Дорожная карта проекта 17

Приложение 3. Список использованного оборудования 21

Приложение 4. Скетч отображения на POV-дисплее прибора № 1 слова «АТОМ» 22

Приложение 5. Скетч отображения на POV-дисплее приборов № 2 и № 3 слова «STOP» 26

Приложение 6. Скетч отображения на POV-дисплее прибора № 4 надписи «ШКОЛА 1538» с сердечком 30

Приложение 7. Программа плавного управления мотором EV3 левой и правой кнопками контроллера 32

Введение

Цель проекта

Разработка недорогих автономных и привлекательных для детей портативных световых приборов-вертушек с POV-дисплеем для обеспечения безопасности детей в темное время суток при переходе неосвещенных участков автомобильной дороги, а также для целей образования (информатика, робототехника) и развлечения (установка на колесо велосипеда, самоката, скутера и др.).

Задачи проекта:

1. Проведение анализа рынка светоотражателей для пешеходов.
2. Составление дорожной карты проекта и списка необходимого оборудования.
3. Освоение сред программирования Arduino IDE, EV3 и составление программ для функционирования световых приборов-вертушек с POV-дисплеями испытательного стенда.
4. Сборка и испытание световых приборов с POV-дисплеем на основе плат Arduino Uno, Arduino Nano, а также микроконтроллеров ATmega 328P-PU и ATTiny 85-20PU.
5. 3D-моделирование и сборка испытательного стенда из робототехнического набора Lego Mindstorms EV3 Education.
6. Проведение лабораторных и полевых испытаний светового прибора-вертушки с POV-дисплеем.
7. Анализ полученных результатов.

Обоснование актуальности проекта

В настоящее время отсутствуют удобные и привлекательные для детей средства обеспечения безопасности при переходе неосвещенных участков автомобильной дороги. Вместе с тем, в 2018 г. зафиксировано 15860 ДТП с участием несовершеннолетних при переходе автомобильных дорог [1].

1. Анализ существующих решений

Используемые на текущий момент обязательные для применения светоотражающие средства [2] непривлекательны для детей и имеют ряд недостатков, например: светоотражающий жилет неудобен в носке и хранении; светоотражающую ленту необходимо нашить на одежду; светоотражающие браслеты, брелки и наклейки имеют малую поверхность отражения. При этом в случае плохой видимости (темного времени суток и тумана), а также внезапного отказа внешних осветительных приборов автомобиля, все вышеуказанные средства неэффективны.

2. Описание разработанного решения

Нами разработаны четыре модификации светового прибора-вертушки с POV-дисплеем (POV – Persistance of Vision, англ. – инерция зрения). Список источников информации, дорожная карта проекта и список использованного оборудования представлены в Приложениях 1 – 3.

Прибор № 1 собран из деталей робототехнического набора Lego WeDo Education 1.0, набора Lego Education «Возобновляемые источники энергии» и представляет собой качающуюся балку, приводимую в движение электромотором посредством кривошипно-шатунного механизма [3, 4]. Этот механизм преобразует вращательное движение кривошипа, который одним концом жестко закреплен на валу мотора, в поступательное движение шатуна, шарнирно соединенного с кривошипом. Другой конец шатуна шарнирно соединен с балкой. Нижний конец балки, в свою очередь, шарнирно соединен с плитой-основанием всей этой конструкции.

Электронная часть прибора № 1 с POV-дисплеем (рис. 1) представляет собой вставленные в качающуюся балку 10 светодиодов [5], подсоединенных через резисторы к плате Arduino Uno, которая управляет светодиодами по специально составленной программе-скетчу (см. Приложение 4), написанной в среде программирования Arduino IDE. При раскачивании балки видно световое изображение или текст, запрограммированный в скетче в виде опреде-

ленных последовательностей сигналов светодиодов. Этот эффект называется персистенцией и основан на инерции нашего зрения, когда отдельные последовательные световые вспышки кажутся непрерывными. Например, когда мы смотрим кинофильм, пленка которого состоит из отдельных кадров.

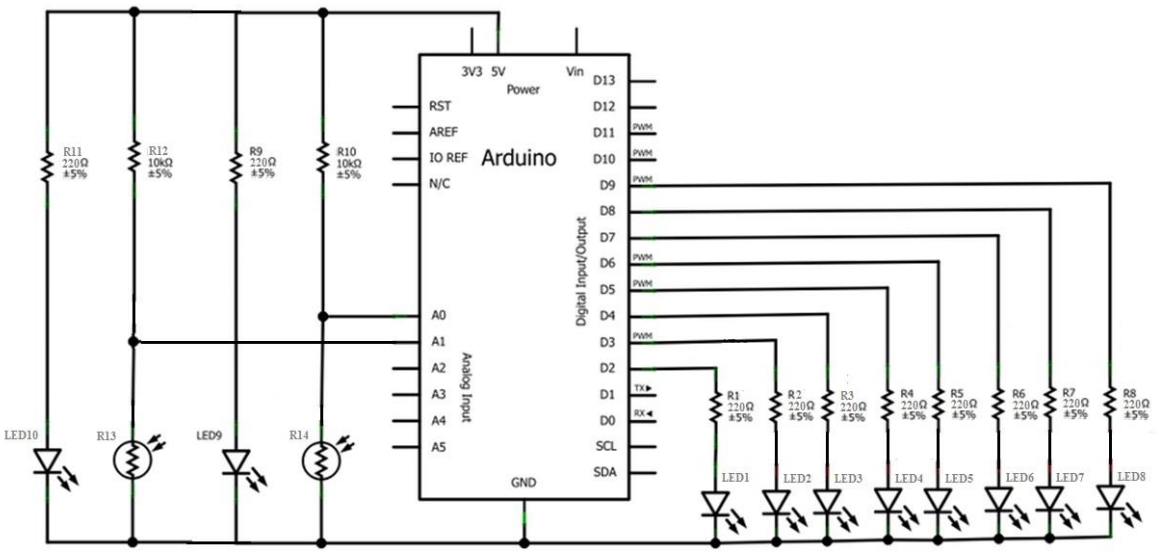


Рис. 1. Принципиальная схема прибора № 1 с POV-дисплеем на основе Arduino Uno

Помимо этого, в электронную часть прибора № 1 входят две оптические пары (оптопары) [5]: светодиод (источник светового сигнала) и фоторезистор – приемник этого сигнала (рис. 1). Оптопары устанавливаются таким образом, чтобы балка оказалась между источником и приемником в крайнем правом и крайнем левом положениях. Тогда при постоянном свечении светодиодов-источников, фоторезисторы фиксируют освещенность, которая меняется в зависимости от того, перекрывает ли балка светодиоды. Изменение освещенности при перекрытии светодиодов фиксирует Arduino Uno и согласно программе переключает вывод символов «слева направо» на «справа налево» и наоборот.

Мною было собрано несколько вариантов прибора № 1. В результате был выбран вариант с массивным основанием (рис. 2), Это увеличило устой-

чивость прибора. С этой же целью и для питания мотора был использован массивный мультиметр-аккумулятор Energy meter из набора Lego Education «Возобновляемые источники энергии».

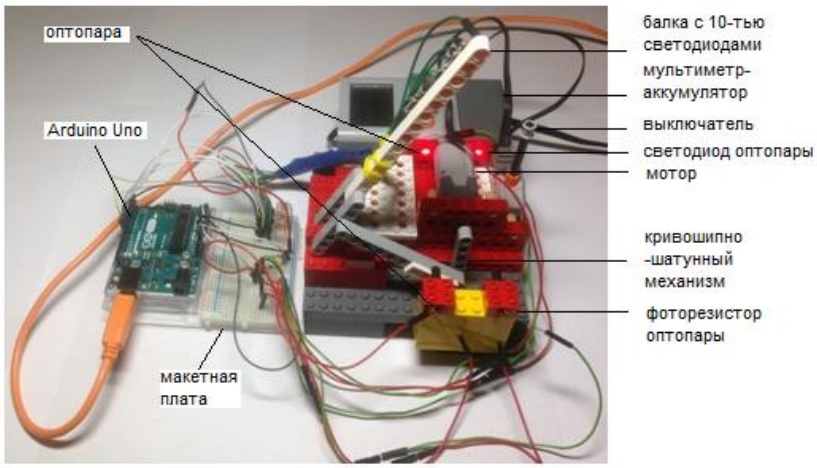


Рис. 2. Прибор №1 с POV-дисплеем на основе Lego, беспаечной макетной платы и платы Arduino Uno

Кроме того, в отличие от [3, 4], я запрограммировал прямой (слева направо) и обратный (справа налево) вывод символов в зависимости от направления движения балки (см. Приложение 4), фиксируемого двумя оптопарами (рис. 1, 2). В результате удалось отобразить на POV-дисплее прибора № 1 слово «АТОМ» (рис. 3) и другие слова.

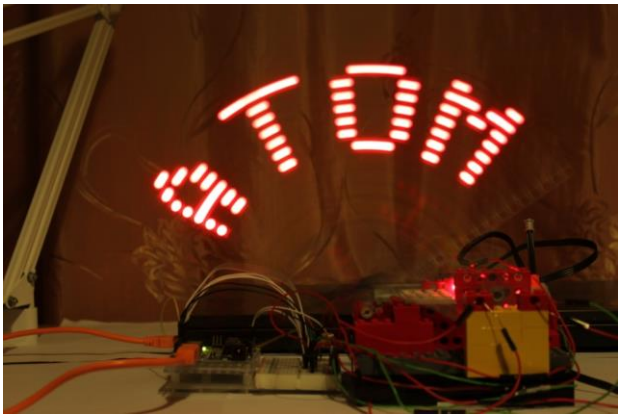


Рис. 3. Результат испытания прибора № 1 с POV-дисплеем

К недостаткам данной конструкции следует отнести ее громоздкость, значительную вибрацию во время работы, необходимость дополнительного крепления проводов, перемещающихся вместе с балкой. При этом в результате замедления движения балки при прохождении «мертвой точки» получаются разные расстояния между буквами и уменьшенная ширина первой буквы (рис. 3). Кроме того, в скетче была использована только латиница.

Для создания более легкого и компактного светового прибора № 2 с POV-дисплеем была использована плата Arduino Nano, беспаечная макетная плата и акселерометр MMA8452, фиксирующий изменение направления движения устройства при размахивании рукой за счет изменения знака ускорения. При этом для большей надежности прибора и четкости изображения при размахивании рукой использовалась светодиодная шкала, а также плоский LiPO-аккумулятор на 3,7V (рис. 4). Схема электрических соединений приведена на рис. 5, скетч – в Приложении 5, а результат – на рис. 6.

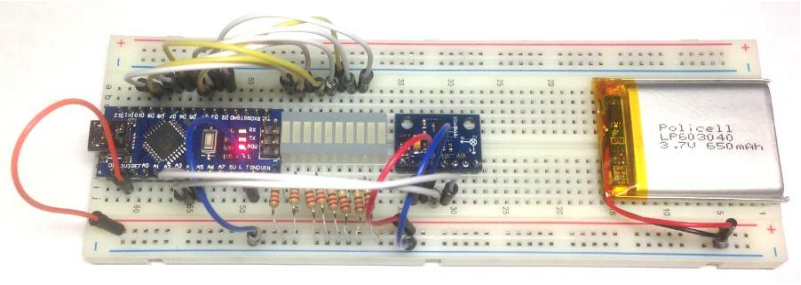


Рис. 4. Прибор № 2 с POV-дисплеем на основе беспаечной макетной платы, платы Arduino Nano, светодиодной шкалы и акселерометра MMA8452

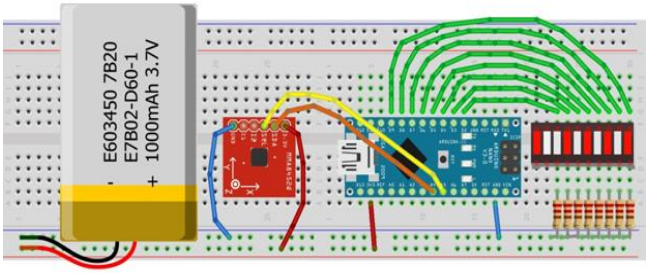


Рис. 5. Схема электрических соединений прибора № 2 с POV-дисплеем на основе Arduino Nano

Рис. 6. Результат испытания прибора № 2 с POV-дисплеем – изображение слова «STOP» в темноте, предназначенное для водителя

Большей компактности устройства можно добиться, перейдя от управляющей платы к микроконтроллеру, что и было реализовано в приборе № 3 с POV-дисплеем на основе микроконтроллера ATmega 328P-PU (рис. 7). Схема электрических соединений приведена на рис. 8, скетч – тот же, что для прибора № 2 (см. Приложение 5).

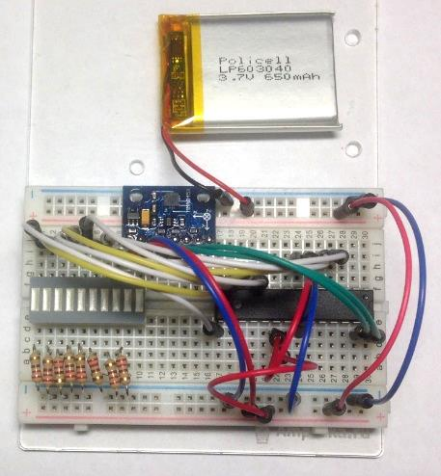


Рис. 7. Прибор № 3 с POV-дисплеем на основе микроконтроллера ATmega 328P-PU

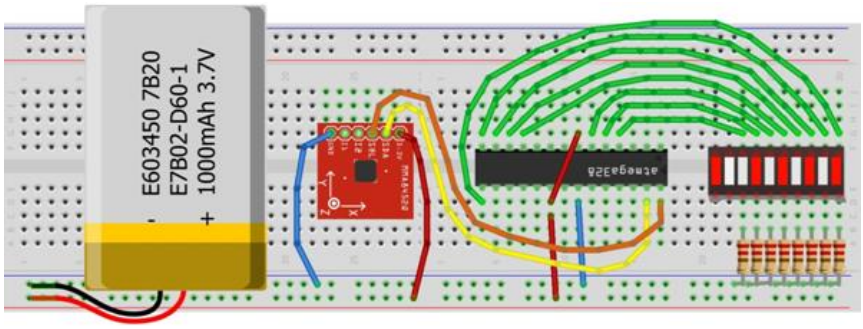


Рис. 8. Схема электрических соединений POV-дисплея на основе ATmega 328P-PU

Стремясь к еще большей компактности и надежности, а также делая шаг к серийному производству, был разработан прибор № 4 (рис. 9, 10) на основе меньшего по размеру микроконтроллера ATtiny85 и печатной платы [7], которая также была уменьшена. Кроме того, было уменьшено количество светодиодов (с 8 до 5) и их размер (с 5 до 3 мм), а также использованы более компактный источник питания CR2032 на 3В и маленький выключатель. В результате после пайки у нас получился легкий и компактный карманный прибор № 4 с POV-дисплеем без соединительных проводов.

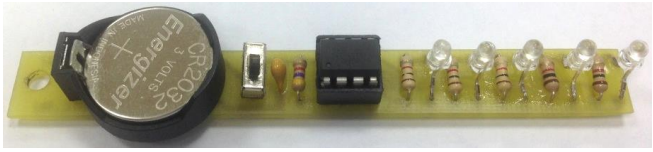


Рис. 9. Прибор № 4 с POV-дисплеем на печатной плате
на основе микроконтроллера ATtiny85

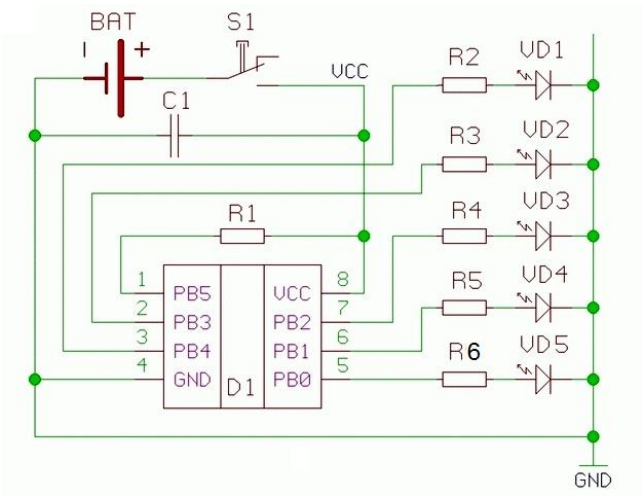


Рис. 10. Принципиальная схема прибора № 4 с POV-дисплеем
на основе микроконтроллера ATtiny85

Поскольку я уменьшил количество светодиодов с 8 до 5, а также захотел, помимо латиницы, отображать на POV-дисплее и кириллицу, пришлось

для написания скетча разобраться с кодировкой символов. Для этого я переделал найденный в интернете генератор символов 15x8 [8] в 5x5.

Например, буква «Ш» (перевернутая для правильного отображения на POV-дисплее) согласно этому генератору имеет следующие кодировки: {31, 1, 31, 1, 31} – в шестнадцатеричной системе исчисления и {11111, 1, 11111, 1, 11111} – в двоичной системе (рис. 11). Генератор автоматически высчитывает кодировку символа после его рисования щелчками по соответствующим ячейкам.

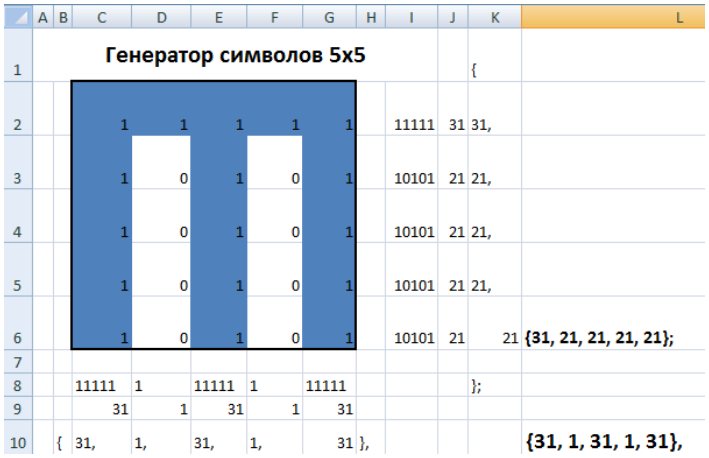


Рис. 11. Кодировка перевернутой буквы «Ш» в генераторе символов 5x5

Для проведения испытаний был написан скетч (см. Приложение 6), отображающий надпись «ШКОЛА 1538» с сердечком – символом любви к нашей школе. В скетче была использована шестнадцатеричная кодировка.

Для проведения испытаний прибора № 4, на основе робототехнического набора Lego Mindstorms Education EV3 был запроектирован и изготовлен испытательный стенд (рис. 12, 13), управляемый кнопками блока EV3 (Приложение 7), планшетом или сотовым телефоном по Bluetooth с помощью приложения Commander, позволяющие плавно менять скорость вращения прибора № 4. Для увеличения количества оборотов на стенд была установлена повышающая зубчатая передача 40:8=5:1.

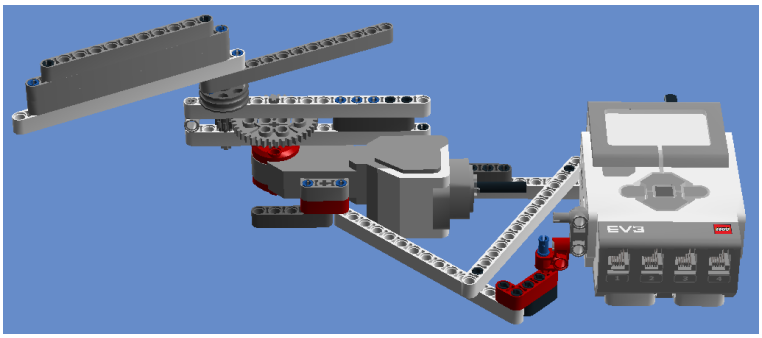


Рис. 12. 3-D модель стенда для испытания светового прибора № 4 с POV-дисплеем в горизонтальном положении, выполненная в приложении Lego Digital Designer

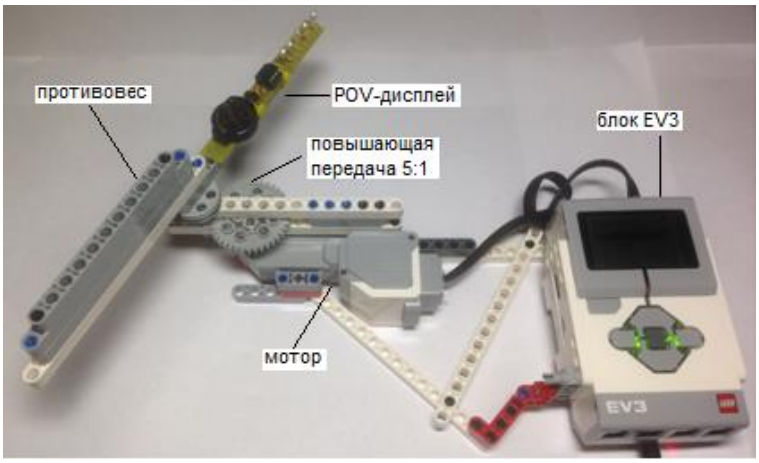


Рис. 13. Испытательный стенд в горизонтальном положении с прикрепленным к несущей балке прибором № 4 с POV-дисплеем и противовесом для уменьшения вибрации

Первые испытания сопровождались значительной вибрацией, для уменьшения которой на несущей балке был установлен противовес (рис. 13). Результаты испытаний в темноте представлены на рис. 14.

Данный стенд может быть установлен как в горизонтальное положение (рис. 14), так и в вертикальное (рис. 15).

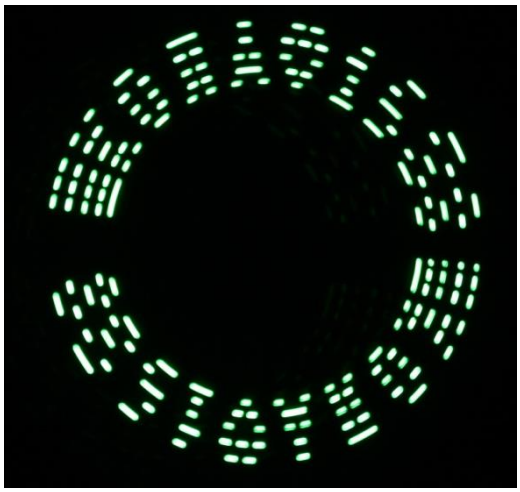


Рис. 14. Результат испытания при вращении прибора № 4 с POV-дисплеем в темноте в горизонтальном положении – изображение надписи «ШКОЛА ❤ 1538»

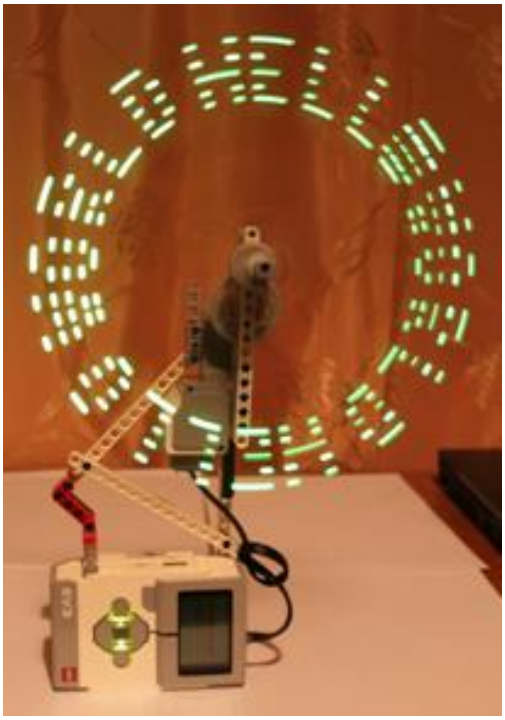


Рис. 15. Результат испытания при вращении прибора № 4 с POV-дисплеем в вертикальном положении при освещении – изображение надписи «HELLO WORLD»

Проведенные испытания показали, что при плавном увеличении скорости вращения световые символы сначала вращаются в одну сторону, затем «замирают», а потом вращаются в другую. Такая «неподвижность» символов возникает за счет стробоскопического эффекта. Было установлено, что этот эффект возникает при мощности вращения мотора EV3, равной 63,5. Поскольку, согласно таблице [9], такая мощность достигается при 1,8 об/сек, тогда, с учетом повышающей передачи 5:1, имеем скорость вращения прибора № 4 с POV-дисплеем – 9 об/сек.

Отверстие в плате (рис. 9) позволяет продеть в него шнурок и вращать световой прибор № 4 с POV-дисплеем вручную, получая отчетливые изображения надписей, особенно в темное время суток.

Полевые испытания прибора № 4 с POV-дисплеем, проведенные на автомобильной дороге в темное время суток показали его надежность. При этом световая надпись «СТОП» не ослепляет водителя и видна с расстояния в 20 м (что соответствует тормозному пути автомобиля, движущегося со скоростью V=60 км/час), а свет, излучаемый прибором № 4, виден с расстояния 35 м (V=80 км/час) [17].

3. Результаты работы

1. Запроектированы и разработаны бюджетные приборы с POV-дисплеем для обеспечения безопасности детей в темное время суток при переходе неосвещенных участков автомобильной дороги на основе плат Arduino Uno и Arduino Nano, а также микроконтроллеров ATmega 328P-PU и ATTiny 85-20PU.

2. Разработанные приборы с POV-дисплеем дают отчетливые изображения надписей и символов в темное время суток, экономичны, не ослепляют водителя и могут быть также использованы для образовательных целей – изучения стробоскопического эффекта на уроках физики, программирования в среде Arduino IDE различных световых надписей и изображений на уроках информатики и робототехники, а также для развлечений (например, для

установки на колеса велосипеда, самоката, скутера и др. средств передвижения).

3. Прибор № 4 на печатной плате с микроконтроллером ATTiny 85-20PU и POV-дисплеем компактен (умещается в кармане), надежен, дешев (около 100 руб.), привлекателен для детей и пригоден для серийного производства.

4. Проведенные лабораторные и полевые испытания прибора № 4 с POV-дисплеем на неосвещенной трассе в темное время суток показали его надежность. При этом световая надпись «СТОП», при вращении прибора вручную, видна с расстояния в 20 м (что соответствует тормозному пути автомобиля, движущегося со скоростью V=60 км/час), а свет, излучаемый прибором № 4, виден с расстояния 35 м (V=80 км/час).

5. Потенциальными заказчиками и потребителями результатов этого проекта могут быть дети и их родители, а также образовательные учреждения: детские сады, школы, колледжи и вузы.

Приложение 1. Список источников информации

1. Количество ДТП в России. [Электронный ресурс]. – Режим доступа: https://advokatycom.com/%D0%BA%D0%BE%D0%BB%D0%B8%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE-%D0%B4%D1%82%D0%BF-%D0%B2-%D1%80%D0%BE%D1%81%D1%81%D0%B8%D0%B8/#i – Заглавие с экрана. – (Дата обращения 10.11.2018).

2. Введение обязательных светоотражателей для пешеходов с 1 июля 2015 года. [Электронный ресурс]. – Режим доступа: https://pddmaster.ru/documentsnews/vvedenie-obyazatelnyh-svetootrazhatelei-dlya-peshehodov-s-1-iyulya-2015-goda.html – Заглавие с экрана. – (Дата обращения 04.11.2018).

3. Механический дисплей из лего и Arduino. [Электронный ресурс]. – Режим доступа: https://geektimes.ru/post/255718/ – Заглавие с экрана. – (Дата обращения 04.05.2018).

4. Механический дисплей на Arduino – работа над ошибками. – Режим доступа: https://geektimes.ru/post/255734/ – Заглавие с экрана. – (Дата обращения 05.05.2018).

5. M. Gasperi, P. Hurbain. Extreme NXT: Extending the LEGO MINDSTORMS NXT to the Next Level. Second Edition, 2009. – 350 p.

6. POV-бегущая строка из 8 светодиодов. [Электронный ресурс]. – Режим доступа: http://wiki.amperka.ru/projects:povdisplay – Заглавие с экрана. – (Дата обращения 08.09.2018).

7. 4-х канальный осциллограф на ATtiny13. [Электронный ресурс]. – Режим доступа: http://sotvorimvmeste.ru/viewtopic.php?f=37&t=202 – Заглавие с экрана. – (Дата обращения 08.10.2018).

8. Голограмма своими руками. Рисуем светом. [Электронный ресурс]. – Режим доступа: https://usamodelkina.ru/11132-gologramma-svoimi-rukami-risuem-svetom.html – Заглавие с экрана. – (Дата обращения 05.11.2018).

9. EV3 Motor Power vs. Rotation Speed – Affinity Laws and a Cool Experiment to Try! [Электронный ресурс]. – Режим доступа: https://builderdude35.com/2017/11/09/ev3-motor-power-vs-rotation-speed-affinity-laws-and-a-cool-experiment-to-try/ – Заглавие с экрана. – (Дата обращения 15.11.2018).

10. LED POV Display with Arduino Uno. [Электронный ресурс]. – Режим доступа: https://www.instructables.com/ID/LED-POV-DISPLAY-WITH-ARDUINO-UNO/ – Заглавие с экрана. – (Дата обращения 05.05.2018).

11. Arduino + Android POV Display. – Режим доступа: https://www.instructables.com/ID/ARDUINO-SIMPLE-POV-DISPLAY/ – Заглавие с экрана. – (Дата обращения 15.05.2018).

12. POV Display with Accelerometer. – Режим доступа: https://forums.adafruit.com/viewtopic.php?t=6446#p31328 – Заглавие с экрана. – (Дата обращения 30.05.2018).

13. Make a simple persistence-of-vision (POV) display with an Arduino for about $6. – Режим доступа: https://maker.pro/projects/arduino/arduino-pov-display/ – Заглавие с экрана. – (Дата обращения 25.05.2018).

14. Как делать проекты на чипе Atmega328? – Режим доступа: https://diy-life.ru/arduino-za-2-kak-delat-proekty-na-chipe-atmega328.html – Заглавие с экрана. – (Дата обращения 05.09.2018).

15. ATTINY85 POV DISPLAY. – Режим доступа: http://www.instructables.com/id/ATtiny8545-POV-Display/ – Заглавие с экрана. – (Дата обращения 15.09.2018).

16. Трехосный акселерометр MMA8452 GY-45. – Режим доступа: https://amperkot.ru/products/trehosnyiy_akselerometr_mma8452_gy45/23871294.html – Заглавие с экрана. – (Дата обращения 25.05.2018).

17. От чего зависит длина тормозного пути и как его рассчитать. – Режим доступа: http://znanieavto.ru/stop/tormoznoj-put-avtomobilya.html – Заглавие с экрана. – (Дата обращения 15.09.2018).

Приложение 2. Дорожная карта проекта

Направление работы, кючевые задачи \ Сроки	2018 год				
	май	сентябрь	октябрь	ноябрь	декабрь
Введение в тематику проекта. Теоретические основы микроконтроллеров	Лекции – 3 Мастер-класс - 1	Лабораторные занятия – 4			
Подготовка концепции решения: - формулирование цели проекта; - анализ рынка светоотражателей для пешеходов; - составление дорожной карты проекта; - составление списка необходимых для реализации ресурсов	Сформулирована цель проекта Подготовлен анализ рынка светоотражателей для пешеходов	Составлена дорожная карта проекта Составлен список необходимых для реализации проекта ресурсов			
Составление программы (скетча) в среде Arduino IDE	Освоена среда программирования	Составлена программа и получены первые результаты на мониторе последовательного порта			
Сборка и испытание прибора № 1 с POV-	Собран и испытан				

дисплеем на основе конструктора Lego, беспаечной макетной платы и платы Arduino Uno	прибор №1 с POV-дисплеем на основе Lego и Arduino Uno				
Сборка и испытание прибора № 2 с POV-дисплеем на основе беспаечной макетной платы и платы Arduino Nano		Собран и испытан прибор №2 с POV-дисплеем на основе Arduino Nano			
Сборка и испытание прибора № 3 с POV-дисплеем на основе беспаечной макетной платы и микроконтроллере ATmega 328P-PU		Собран и испытан прибор №3 с POV-дисплеем на основе ATmega 328P-PU			
Сборка прибора №4 с POV-дисплеем на основе микроконтроллера ATTiny 85-20PU на печатной плате. Сборка испытательного стенда из Lego. Испытание прибора №4			Собран и испытан прибор №4 с POV-дисплеем на основе ATTiny 85-20PU		
Анализ полученных				Прове-	

результатов				ден анализ полученных результатов	
Подготовка доклада и презентации к школьной конференции, участие в конференции					Подготовлен доклад и презентация к школьной конференции. Принял участие в школьной конференции
Оформление разработанного решения в форме схем, таблиц, приборов для размещения в личном кабинете на сайте Московского городского конкурса ***mgk.olimpiada.ru***. Размещение на сайте					Оформлено и размещено на сайте разработанное решение

Приложение 3. Список использованного оборудования

1. Ноутбук HP ProBook 4720s – 2 шт.
2. Робототехнический набор Lego WeDo Education 1.0 – 1 компл.
3. Набор Lego Power Functions – 1 компл.
4. Набор Lego Education «Возобновляемые источники энергии» – 1 компл.
5. Робототехнический набор Lego Mindstorms EV3 Education – 1 компл.
6. Набор электроники «Матрешка Z» – 3 компл.
7. Плата: Arduino Nano – 1 шт.
8. Микроконтроллеры: ATmega 328P-PU – 1 шт., ATTiny 85-20PU – 1 шт.
9. Акселерометр MMA8452 – 1 шт.
10. Светодиод (3мм) – 25 шт.
11. Выключатель – 5 шт.
12. LiPo-аккумулятор, 3,7 В – 1 шт.
13. Батарейка CR2032– 1 шт.
14. Паяльная станция AOYUE-968A – 1 шт.

Приложение 4. Скетч отображения на POV-дисплее прибора № 1 слова «АТОМ»

```
// Программа отображения на POV-дисплее прибора №1 слова «АТОМ»
// задержка между столбцами одного символа
// ширина символа
#define TIMER_COLUMS 3

// задержка между символами
// ширина между символами
#define TIMER_FRAMES 7
// количество светодиодов, высота символа
#define ROW  8
// количество столбцов в символе
#define COL  5
// массив пинов, к которым подключены светодиоды
int pins[] = {2, 3, 4, 5, 6, 7, 8, 9};

int dir;

// двумерный массив шестнадцатеричных символов
const unsigned char image[][COL] = {
  {0x00, 0x00, 0x00, 0x00, 0x00},  // пробел
  {0x00, 0xF6, 0xF6, 0x00, 0x00},  // !
  {0x00, 0xE0, 0x00, 0xE0, 0x00},  // "
  {0x28, 0xFE, 0x28, 0xFE, 0x28},  // #
  {0x00, 0x64, 0xD6, 0x54, 0x08},  // $
  {0xC2, 0xCC, 0x10, 0x26, 0xC6},  // %
  {0x4C, 0xB2, 0x92, 0x6C, 0x0A},  // &
  {0x00, 0x00, 0xE0, 0x00, 0x00},  // '
  {0x00, 0x38, 0x44, 0x38, 0x00},  // )
  {0x00, 0x82, 0x44, 0x38, 0x00},  // )
  {0x88, 0x50, 0xF8, 0x50, 0x88},  // *
  {0x08, 0x08, 0x3E, 0x08, 0x08},  // +
  {0x00, 0x00, 0x05, 0x06, 0x00},  // ,
  {0x08, 0x08, 0x08, 0x08, 0x08},  // -
  {0x00, 0x00, 0x06, 0x06, 0x00},  // .
  {0x02, 0x0C, 0x10, 0x60, 0x80},  // /
  {0x7C, 0x8A, 0x92, 0xA2, 0x7C},  // 0
  {0x00, 0x42, 0xFE, 0x02, 0x00},  // 1
  {0x42, 0x86, 0x8A, 0x92, 0x62},  // 2
  {0x44, 0x82, 0x92, 0x92, 0x6C},  // 3
  {0x10, 0x30, 0x50, 0xFE, 0x10},  // 4
  {0xE4, 0xA2, 0xA2, 0xA2, 0x9C},  // 5
  {0x3C, 0x52, 0x92, 0x92, 0x0C},  // 6
  {0x80, 0x86, 0x98, 0xE0, 0x80},  // 7
  {0x6C, 0x92, 0x92, 0x92, 0x6C},  // 8
  {0x60, 0x92, 0x92, 0x94, 0x78},  // 9
  {0x00, 0x00, 0x36, 0x36, 0x00},  // :
  {0x00, 0x00, 0x35, 0x36, 0x00},  // ;
  {0x10, 0x28, 0x44, 0x82, 0x00},  // <
  {0x28, 0x28, 0x28, 0x28, 0x28},  // =
  {0x00, 0x82, 0x44, 0x28, 0x10},  // >
  {0x40, 0x80, 0x8A, 0x90, 0x60},  // ?
  {0x7C, 0x82, 0xBA, 0xBA, 0x62},  // @
  {0x3E, 0x48, 0x88, 0x48, 0x3E},  // A
  {0xFE, 0x92, 0x92, 0x92, 0x6C},  // B
  {0x7C, 0x82, 0x82, 0x82, 0x44},  // C
  {0xFE, 0x82, 0x82, 0x82, 0x7C},  // D
  {0xFE, 0x92, 0x92, 0x92, 0x82},  // E
  {0xFE, 0x90, 0x90, 0x90, 0x80},  // F
  {0x7C, 0x82, 0x82, 0x8A, 0x4E},  // G
  {0xFE, 0x10, 0x10, 0x10, 0xFE},  // H
```

```
  {0x82, 0x82, 0xFE, 0x82, 0x82},  // I
  {0x84, 0x82, 0xFC, 0x80, 0x80},  // J
  {0xFE, 0x10, 0x28, 0x44, 0x82},  // K
  {0xFE, 0x02, 0x02, 0x02, 0x02},  // L
  {0xFE, 0x40, 0x20, 0x40, 0xFE},  // M
  {0xFE, 0x60, 0x10, 0x0C, 0xFE},  // N
  {0x7C, 0x82, 0x82, 0x82, 0x7C},  // O
  {0xFE, 0x90, 0x90, 0x90, 0x60},  // P
  {0x7C, 0x82, 0x82, 0x86, 0x7E},  // Q
  {0xFE, 0x90, 0x98, 0x94, 0x62},  // R
  {0x64, 0x92, 0x92, 0x92, 0x4C},  // S
  {0x80, 0x80, 0xFE, 0x80, 0x80},  // T
  {0xFC, 0x02, 0x02, 0x02, 0xFC},  // U
  {0xF8, 0x04, 0x02, 0x04, 0xF8},  // V
  {0xFC, 0x02, 0x0C, 0x02, 0xFC},  // W
  {0xC6, 0x28, 0x10, 0x28, 0xC6},  // X
  {0xC0, 0x20, 0x1E, 0x20, 0xC0},  // Y
  {0x86, 0x8A, 0x92, 0xA2, 0xC2},  // Z
  {0x00, 0x00, 0xFE, 0x82, 0x00},  // [
  {0x00, 0x00, 0x00, 0x00, 0x00},  // \
  {0x80, 0x60, 0x10, 0x0C, 0x02},  // ]
  {0x20, 0x40, 0x80, 0x40, 0x20},  // ^
  {0x01, 0x01, 0x01, 0x01, 0x01},  // _
  {0x80, 0x40, 0x20, 0x00, 0x00},  // `
  {0x04, 0x2A, 0x2A, 0x2A, 0x1E},  // a
  {0xFE, 0x12, 0x22, 0x22, 0x1C},  // b
  {0x1C, 0x22, 0x22, 0x22, 0x14},  // c
  {0x1C, 0x22, 0x22, 0x12, 0xFE},  // d
  {0x1C, 0x2A, 0x2A, 0x2A, 0x18},  // e
  {0x10, 0x7E, 0x90, 0x80, 0x40},  // f
  {0x18, 0x25, 0x25, 0x25, 0x1E},  // g
  {0xFE, 0x10, 0x10, 0x10, 0x0E},  // h
  {0x00, 0x12, 0x5E, 0x02, 0x00},  // i
  {0x02, 0x01, 0x01, 0x11, 0x5E},  // j
  {0xFE, 0x08, 0x08, 0x14, 0x22},  // k
  {0x00, 0x82, 0xFE, 0x02, 0x00},  // l
  {0x3E, 0x20, 0x1C, 0x20, 0x1E},  // m
  {0x3E, 0x20, 0x20, 0x20, 0x1E},  // n
  {0x1C, 0x22, 0x22, 0x22, 0x1C},  // o
  {0x3F, 0x24, 0x24, 0x24, 0x18},  // p
  {0x18, 0x24, 0x24, 0x3F, 0x01},  // q
  {0x3E, 0x10, 0x20, 0x20, 0x10},  // r
  {0x12, 0x2A, 0x2A, 0x2A, 0x04},  // s
  {0x00, 0x10, 0x3C, 0x12, 0x04},  // t
  {0x3C, 0x02, 0x02, 0x02, 0x3E},  // u
  {0x30, 0x0C, 0x02, 0x0C, 0x30},  // v
  {0x38, 0x06, 0x18, 0x06, 0x38},  // w
  {0x22, 0x14, 0x08, 0x14, 0x22},  // x
  {0x38, 0x05, 0x05, 0x05, 0x3E},  // y
  {0x22, 0x26, 0x2A, 0x32, 0x22},  // z
  {0x00, 0x10, 0x6C, 0x82, 0x82},  // {
  {0x00, 0x00, 0xFF, 0x00, 0x00},  // |
  {0x82, 0x82, 0x6C, 0x10, 0x00},  // }
  {0x08, 0x10, 0x18, 0x08, 0x10},  // ~
  {0x38, 0x7c, 0x3e, 0x7c, 0x38},  // Сердечко 127
};

void setup()
{
  // устанавливаем пины из массива pins[] в режим выхода
  for (int i = 0; i < ROW; i++) {
    pinMode(pins[i], OUTPUT);
```

```
  }
  // гасим все светодиоды
  ledsAllOff();
}

void loop()
{
  // определяем направление качения планки со светодиодами
  // с помощью оптопар
  while (analogRead(1)>700){
      showLR('A');
      showLR('T');
      showLR('O');
      showLR('M');
      delay(10);
  }
  while (analogRead(0)>500){
      showRL('M');
      showRL('O');
      showRL('T');
      showRL('A');
      delay(10);
  }
}

// функция вывода символа слева направо
void showLR(unsigned char c)
{
  // перебор столбцов матрицы
    // перебор строк матрицы (светодиодов)
    ledsSet(c, 1);
    delay(3);
    ledsSet(c, 2);
    delay(2);
    ledsSet(c, 3);
    delay(2);
    ledsSet(c, 4);
    delay(3);
  // гасим все светодиоды
  ledsAllOff();
  // задержка задающая ширину между символами
  delay(TIMER_FRAMES);
}

// функция вывода символа справа налево
void showRL(unsigned char c)
{
  // перебор столбцов матрицы
  // перебор строк матрицы (светодиодов)
    ledsSet(c, 4);
    delay(3);
    ledsSet(c, 3);
    delay(2);
    ledsSet(c, 2);
    delay(2);
    ledsSet(c, 1);
    delay(3);
  // гасим все светодиоды
  ledsAllOff();
  // задержка задающая ширину между символами
  delay(TIMER_FRAMES);
}
```

```
// функция гасящая все светодиоды
void ledsAllOff()
{
  for (int i = 0; i < ROW; i++) {
    digitalWrite(pins[i], LOW);
  }
}

// функция включения светодиодов в текущем столбце
void ledsSet(unsigned char c, int i)
{
  for (int j = 0; j < ROW; j++) {
    digitalWrite(pins[j], bitRead(image[c-32][i], j));
  }
}
```

Приложение 5. Скетч отображения на POV-дисплее приборов № 2 и № 3 слова «STOP»

```
// Программа работы ручных приборов № 2 и № 3 с POV-дисплеем для обеспечения безопасности
// пешеходов в темное время суток на основе Arduino Nano,
// светодиодной шкалы и акселерометра MMA8452

// Библиотека обеспечения работы шины I2C для работы акселерометра
#include <Wire.h>

// Библиотека для работы акселерометра MMA8452Q
#include <SFE_MMA8452Q.h>
// Создаём объект для работы с акселерометром
MMA8452Q accel;
// Задаем:
// задержку между столбцами одного символа - ширину символа
#define TIMER_COLUMS 3
// задержку между символами - пробел
#define TIMER_FRAMES 8
// количество светодиодов - высоту символа
#define ROW  8
// количество столбцов в символе
#define COL  5
// массив портов, к которым подключены светодиоды
int pins[] = {2, 3, 4, 5, 6, 7, 8, 9};
// переменную для хранения направления и величины
// ускорения по оси Y
float y = 0;
// двумерный массив символов
const unsigned char image[][COL] = {
  {0x00, 0x00, 0x00, 0x00, 0x00},  // пробел
  {0x00, 0xF6, 0xF6, 0x00, 0x00},  // !
  {0x00, 0xE0, 0x00, 0xE0, 0x00},  // "
  {0x28, 0xFE, 0x28, 0xFE, 0x28},  // #
  {0x00, 0x64, 0xD6, 0x54, 0x08},  // $
  {0xC2, 0xCC, 0x10, 0x26, 0xC6},  // %
  {0x4C, 0xB2, 0x92, 0x6C, 0x0A},  // &
  {0x00, 0x00, 0xE0, 0x00, 0x00},  // '
  {0x00, 0x38, 0x44, 0x38, 0x00},  // )
  {0x00, 0x82, 0x44, 0x38, 0x00},  // )
  {0x88, 0x50, 0xF8, 0x50, 0x88},  // *
  {0x08, 0x08, 0x3E, 0x08, 0x08},  // +
  {0x00, 0x00, 0x05, 0x06, 0x00},  // ,
  {0x08, 0x08, 0x08, 0x08, 0x08},  // -
  {0x00, 0x00, 0x06, 0x06, 0x00},  // .
  {0x02, 0x0C, 0x10, 0x60, 0x80},  // /
  {0x7C, 0x8A, 0x92, 0xA2, 0x7C},  // 0
  {0x00, 0x42, 0xFE, 0x02, 0x00},  // 1
  {0x42, 0x86, 0x8A, 0x92, 0x62},  // 2
  {0x44, 0x82, 0x92, 0x92, 0x6C},  // 3
  {0x10, 0x30, 0x50, 0xFE, 0x10},  // 4
  {0xE4, 0xA2, 0xA2, 0xA2, 0x9C},  // 5
  {0x3C, 0x52, 0x92, 0x92, 0x0C},  // 6
  {0x80, 0x86, 0x98, 0xE0, 0x80},  // 7
  {0x6C, 0x92, 0x92, 0x92, 0x6C},  // 8
  {0x60, 0x92, 0x92, 0x94, 0x78},  // 9
  {0x00, 0x00, 0x36, 0x36, 0x00},  // :
  {0x00, 0x00, 0x35, 0x36, 0x00},  // ;
  {0x10, 0x28, 0x44, 0x82, 0x00},  // <
  {0x28, 0x28, 0x28, 0x28, 0x28},  // =
  {0x00, 0x82, 0x44, 0x28, 0x10},  // >
  {0x40, 0x80, 0x8A, 0x90, 0x60},  // ?
  {0x7C, 0x82, 0xBA, 0xBA, 0x62},  // @
```

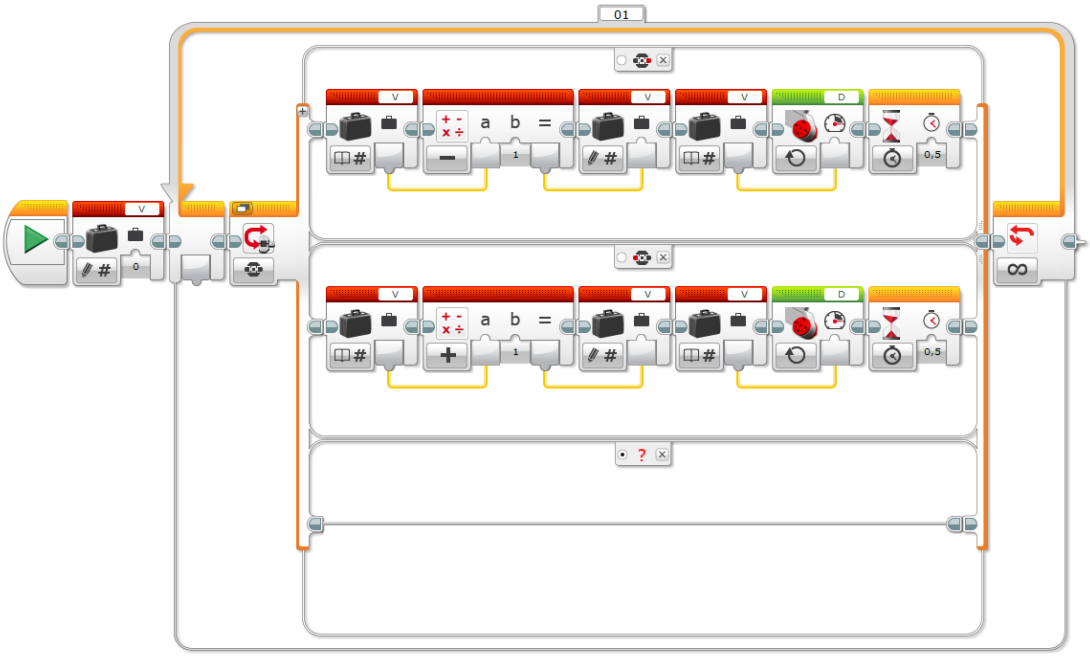
```
{0x3E, 0x48, 0x88, 0x48, 0x3E},  // A
{0xFE, 0x92, 0x92, 0x92, 0x6C},  // B
{0x7C, 0x82, 0x82, 0x82, 0x44},  // C
{0xFE, 0x82, 0x82, 0x82, 0x7C},  // D
{0xFE, 0x92, 0x92, 0x92, 0x82},  // E
{0xFE, 0x90, 0x90, 0x90, 0x80},  // F
{0x7C, 0x82, 0x82, 0x8A, 0x4E},  // G
{0xFE, 0x10, 0x10, 0x10, 0xFE},  // H
{0x82, 0x82, 0xFE, 0x82, 0x82},  // I
{0x84, 0x82, 0xFC, 0x80, 0x80},  // J
{0xFE, 0x10, 0x28, 0x44, 0x82},  // K
{0xFE, 0x02, 0x02, 0x02, 0x02},  // L
{0xFE, 0x40, 0x20, 0x40, 0xFE},  // M
{0xFE, 0x60, 0x10, 0x0C, 0xFE},  // N
{0x7C, 0x82, 0x82, 0x82, 0x7C},  // O
{0xFE, 0x90, 0x90, 0x90, 0x60},  // P
{0x7C, 0x82, 0x82, 0x86, 0x7E},  // Q
{0xFE, 0x90, 0x98, 0x94, 0x62},  // R
{0x64, 0x92, 0x92, 0x92, 0x4C},  // S
{0x80, 0x80, 0xFE, 0x80, 0x80},  // T
{0xFC, 0x02, 0x02, 0x02, 0xFC},  // U
{0xF8, 0x04, 0x02, 0x04, 0xF8},  // V
{0xFC, 0x02, 0x0C, 0x02, 0xFC},  // W
{0xC6, 0x28, 0x10, 0x28, 0xC6},  // X
{0xC0, 0x20, 0x1E, 0x20, 0xC0},  // Y
{0x86, 0x8A, 0x92, 0xA2, 0xC2},  // Z
{0x00, 0x00, 0xFE, 0x82, 0x00},  // [
{0x00, 0x00, 0x00, 0x00, 0x00},  // \
{0x80, 0x60, 0x10, 0x0C, 0x02},  // ]
{0x20, 0x40, 0x80, 0x40, 0x20},  // ^
{0x01, 0x01, 0x01, 0x01, 0x01},  // _
{0x80, 0x40, 0x20, 0x00, 0x00},  // `
{0x04, 0x2A, 0x2A, 0x2A, 0x1E},  // a
{0xFE, 0x12, 0x22, 0x22, 0x1C},  // b
{0x1C, 0x22, 0x22, 0x22, 0x14},  // c
{0x1C, 0x22, 0x22, 0x12, 0xFE},  // d
{0x1C, 0x2A, 0x2A, 0x2A, 0x18},  // e
{0x10, 0x7E, 0x90, 0x80, 0x40},  // f
{0x18, 0x25, 0x25, 0x25, 0x1E},  // g
{0xFE, 0x10, 0x10, 0x10, 0x0E},  // h
{0x00, 0x12, 0x5E, 0x02, 0x00},  // i
{0x02, 0x01, 0x01, 0x11, 0x5E},  // j
{0xFE, 0x08, 0x08, 0x14, 0x22},  // k
{0x00, 0x82, 0xFE, 0x02, 0x00},  // l
{0x3E, 0x20, 0x1C, 0x20, 0x1E},  // m
{0x3E, 0x20, 0x20, 0x20, 0x1E},  // n
{0x1C, 0x22, 0x22, 0x22, 0x1C},  // o
{0x3F, 0x24, 0x24, 0x24, 0x18},  // p
{0x18, 0x24, 0x24, 0x3F, 0x01},  // q
{0x3E, 0x10, 0x20, 0x20, 0x10},  // r
{0x12, 0x2A, 0x2A, 0x2A, 0x04},  // s
{0x00, 0x10, 0x3C, 0x12, 0x04},  // t
{0x3C, 0x02, 0x02, 0x02, 0x3E},  // u
{0x30, 0x0C, 0x02, 0x0C, 0x30},  // v
{0x38, 0x06, 0x18, 0x06, 0x38},  // w
{0x22, 0x14, 0x08, 0x14, 0x22},  // x
{0x38, 0x05, 0x05, 0x05, 0x3E},  // y
{0x22, 0x26, 0x2A, 0x32, 0x22},  // z
{0x00, 0x10, 0x6C, 0x82, 0x82},  // {
{0x00, 0x00, 0xFF, 0x00, 0x00},  // |
{0x82, 0x82, 0x6C, 0x10, 0x00},  // }
{0x08, 0x10, 0x18, 0x08, 0x10},  // ~
```

```
  {0x38, 0x7c, 0x3e, 0x7c, 0x38},  // Сердечко 127
};
void setup()
{
  // Инициализируем акселерометр
  accel.init();
  // Устанавливаем дипазон работы акселерометра: +/-2g
  accel.init(SCALE_2G);
  // Задаем пины из массива pins[] как порты вывода
  for (int i = 0; i < ROW; i++) {
    pinMode(pins[i], OUTPUT);
  }
  // Гасим все светодиоды
  ledsAllOff();
}

void loop()
{
  // Считываем направление и величину ускорения по оси Y
  accel.read();
  // Даем задержку на считывание
  delay(10);
  y = accel.cy;
  // Если ускорение положительно, т.е. направлено справа налево
  if (y > 0.3) {
      // выводим символы зеркально отражаенными справа налево
      showRL('P');
      showRL('O');
      showRL('T');
      showRL('S');
      delay(10);
    }
   // иначе, если ускорение отрицательно, т.е. направлено слева
   // направо
   else if (y < -0.3) {
      // выводим символы в обычном порядке - слева направо
      showLR('S');
      showLR('T');
      showLR('O');
      showLR('P');
      delay(10);
    }
}

// Функция вывода символов слева направо
void showLR(unsigned char c)
{
  // перебор столбцов матрицы
  for (int i = 0; i < COL; i++) {
    // перебор строк матрицы - светодиодов
    ledsSet(c, i);
    // задержка, задающая ширину символа
    delay(TIMER_COLUMS);
  }
  // гасим все светодиоды
  ledsAllOff();
  // задержка, задающая пробел между символами
  delay(TIMER_FRAMES);
}

// Функция вывода зеркально отраженных символов справа налево
void showRL(unsigned char c)
```

```
{
  // перебор столбцов матрицы
  for (int i = COL - 1; i >= 0; i--) {
    // перебор строк матрицы - светодиодов
    ledsSet(c, i);
    // задержка, задающая ширину символа
    delay(TIMER_COLUMS);
  }
  // гасим все светодиоды
  ledsAllOff();
  // задержка, задающая пробел между символами
  delay(TIMER_FRAMES);
}

// Функция, гасящая все светодиоды
void ledsAllOff()
{
  for (int i = 0; i < ROW; i++) {
    digitalWrite(pins[i], LOW);
  }
}

// Функция включения светодиодов в текущем столбце
void ledsSet(unsigned char c, int i)
{
  for (int j = 0; j < ROW; j++) {
    digitalWrite(pins[j], bitRead(image[c-32][i], j));
  }
}
```

Приложение 6. Скетч отображения на POV-дисплее прибора № 4 надписи «ШКОЛА 1538» с сердечком

```
// Программа отображения на POV-дисплее прибора № 4 надписи «ШКОЛА 1538» с сердечком
int delayTime = 1;
int charBreak = 2.1;

int LED1 = 0;
int LED2 = 1;
int LED3 = 2;
int LED4 = 3;
int LED5 = 4;

void setup()
{
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
}

int a[] = {1, 6, 26, 6, 1};
int k[] = {31, 4, 10, 17, 0};
int o[] = {14, 17, 17, 14, 0};
int a1[] ={0, 17, 31, 1, 0}; // 1
int a3[] ={10, 17, 21, 21, 10}; // 3
int a5[] ={13, 21, 21, 2, 0}; // 5
int a8[] ={10, 21, 21, 10, 0}; // 8
int LR[] ={1, 30, 16, 31, 0}; // Л
int SCH[] ={31, 1, 31, 1, 31}; // Ш
int heart[] ={12, 18, 9, 18, 12}; // сердечко

void displayLine(int line)
{
  int myline;
  myline = line;
  if (myline>=16) {digitalWrite(LED1, HIGH); myline-=16;} else {digitalWrite(LED1, LOW);}
  if (myline>=8)  {digitalWrite(LED2, HIGH); myline-=8;}  else {digitalWrite(LED2, LOW);}
  if (myline>=4)  {digitalWrite(LED3, HIGH); myline-=4;}  else {digitalWrite(LED3, LOW);}
  if (myline>=2)  {digitalWrite(LED4, HIGH); myline-=2;}  else {digitalWrite(LED4, LOW);}
  if (myline>=1)  {digitalWrite(LED5, HIGH); myline-=1;}  else {digitalWrite(LED5, LOW);}
}

void displayChar(char c)
{
  if (c == 'a'){for (int i = 0; i <5; i++){displayLine(a[i]);delay(delayTime);}displayLine(0);}
  if (c == 'k'){for (int i = 0; i <5; i++){displayLine(k[i]);delay(delayTime);}displayLine(0);}
  if (c == 'o'){for (int i = 0; i <5; i++){displayLine(o[i]);delay(delayTime);}displayLine(0);}
  if (c == '1'){for (int i = 0; i <5; i++){displayLine(a1[i]);delay(delayTime);}displayLine(0);}
  if (c == '3'){for (int i = 0; i <5; i++){displayLine(a3[i]);delay(delayTime);}displayLine(0);}
  if (c == '5'){for (int i = 0; i <5; i++){displayLine(a5[i]);delay(delayTime);}displayLine(0);}
  if (c == '8'){for (int i = 0; i <5; i++){displayLine(a8[i]);delay(delayTime);}displayLine(0);}
  if (c == 'L'){for (int i = 0; i <5; i++){displayLine(LR[i]);delay(delayTime);}displayLine(0);}
  if (c == 'S'){for (int i = 0; i <5; i++){displayLine(SCH[i]);delay(delayTime);}displayLine(0);}
  if (c == 'H'){for (int i = 0; i <5; i++){displayLine(heart[i]);delay(delayTime);}displayLine(0);}
  delay(charBreak);
}

void displayString(char* s)
{
  for (int i = 0; i<=strlen(s); i++)
```

```
  {
    displayChar(s[i]);
  }
}
void loop()
{
  displayString(" SkoLaH1538 ");
}
```

Приложение 7. Программа плавного управления мотором EV3 левой и правой кнопками контроллера