

МБОУ «Гимназия №11 г. Ельца»

Научно-исследовательская работа

Создание умной системы пешеходного перехода

Автор:

Смолин Кирилл

Владимирович

МБОУ «Гимназия № 11 г.

Ельца», кл. 11А

Научные руководители:

Австриевских Наталья

Михайловна,

учитель физики МБОУ

«Гимназия № 11 г. Ельца»

Злобин М.С.

Елец, 2018-2019

Содержание

1. Введение	3
2. Теоретический материал.....	3
3. Практическая часть.....	6
4. Заключение.....	6
5. Список литературы.....	7
6. Приложение.....	7

1. Введение

В настоящее время нередко возникают ситуации, когда необходимо перейти дорогу в темное время суток. В этот момент водитель может не заметить пешехода, что может привести к несчастному случаю. Для того чтобы снизить вероятность возникновения ДТП с участием пешеходов, был разработан проект умной системы пешеходного перехода.

Он направлен как на предупреждение пешеходов о высокой скорости приближающегося автомобиля, так и на предупреждение водителей о том, что на пешеходном переходе находится пешеход.

Новизна проекта заключается в том, что его, в отличие от многих других вариантов, можно будет реализовать на практике. Для его работы необходимы только лампочки, провода, источник тока и ультразвуковые датчики. Этот набор компонентов будет дешевле, чем набор компонентов для других систем. Кроме того, его можно использовать на любом участке дороги, где есть стоб.

Я поставил перед собой цель: создать работающую систему «умного» пешеходного перехода.

Для этого необходимо выполнить следующие задачи:

1. Разработать схему сборки модели и написать программу для неё.
2. Собрать работающую модель.
3. Провести испытания модели.

Также была поставлена гипотеза: в современных условиях можно реализовать систему умного пешеходного перехода.

2. Теоретический этап

Для того чтобы создать рабочий прототип, нужно разобраться, как это работает. Во-первых, необходимо измерить расстояние до объекта. Это можно сделать при помощи ультразвуковых датчиков.

Ультразвуковые датчики расстояния Ардуино очень востребованы в робототехнических проектах из-за своей относительной простоты, достаточной точности и доступности. Они могут быть использованы как приборы, помогающие объезжать препятствия, получать размеры предметов, моделировать карту помещения и сигнализировать о приближении или удалении объектов. Одним из распространенных вариантов такого устройства является датчик расстояния, в конструкцию которого входит ультразвуковой дальномер HC SR04. Он работает по принципу действия ультразвука. Направленные узкие пучки ультразвука применяются для измерения расстояния до объекта. Для этой цели в датчике находятся излучатель и приёмник ультразвука. Излучатель даёт короткие сигналы, которые доходят до тела и, отражаясь от него, достигают приёмника. Моменты излучения и приёма сигнала регистрируются. Таким образом, за время t , которое проходит с момента отправления сигнала до момента его приёма, сигнал, распространяющийся со скоростью v , проходит путь, равный удвоенному расстоянию до тела, т. е. $2h$:

$$2h = vt$$

Отсюда легко вычислить расстояние h :

$$h = \frac{vt}{2}$$

Описанный метод определения расстояния до объекта называется эхолокацией. Если это расстояние меньше определенных значений, то загораются лампочки. Это свидетельствует о том, что водитель должен сбросить скорость, так как рядом с дорогой находится пешеход. Пешеход, увидев, что лампочки загорелись, должен подождать, пока проедет транспортное средство.

Исходя из этого, соберем схему (рис.1 см. Приложение). Для этого потребуются плата Arduino, провода, плата, ультразвуковые датчики, макетная плата, лампочки. Нужно заметить, что сборка происходит, когда плата не подключена к источнику тока. Чтобы подключить систему к источнику тока, достаточно подсоединить плату Arduino к компьютеру с помощью USB кабеля.

Чтобы все работало, нужно написать программу (см. Приложение). Она написана на языке с простым названием C. Этот язык существует с самых первых дней развития вычислительной техники. А вот что действительно привносит Arduino — это набор простых в использовании команд, написанных на C, которые вы можете использовать в своих программах.

Пуристы могут заметить, что в Arduino используется C++, объектно-ориентированное расширение языка C. Строго говоря, они правы, однако наличие всего 1–2 Кбайт памяти обычно означает, что использование объектно-ориентированных приемов при программировании для Arduino не самая лучшая идея, за исключением особых ситуаций, и фактически программы пишутся на C.

Приведем фрагменты программы (с комментариями):

```
delayMicroseconds(2);  
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);  
duration = pulseIn(echoPin, HIGH);  
distance = (duration/2) / 29.1;
```

Данная часть программы нужна для того, чтобы измерить расстояние до объекта (пешехода или транспортное средство (для

транспортного средства используется переменная distance1, а для пешехода – distance)).

```
if (distance <= 30 && duration1 <=200) {  
digitalWrite(led, HIGH);  
digitalWrite(led1, HIGH); }  
else {  
digitalWrite(led, LOW);  
digitalWrite(led1, LOW);  
delay(1000); }
```

Эта часть программы необходима для того, чтобы узнать, нужно ли в данный момент лампочкам загораться. В ней сравниваются значения, полученные с датчиков, с данными, которые необходимы для того, чтобы лампочки вовремя загорелись, тем самым предупредив как водителя, так и пешехода. Оператор && нужен, чтобы показать условие И, то есть условие distance <= 30 выполняются одновременно.

Но недостаточно просто написать программу. Нужно загрузить её в плату Arduino. Для этого достаточно подсоединить плату Arduino к компьютеру с помощью USB кабеля.

3.Практическая часть

После того, как была собрана модель и написана программа, необходимо провести испытания. Для этого были использованы два объекта. Их ставили на разные расстояния от датчиков и, исходя из этого, видели, загораются лампочки или нет. Сначала это симулировалось на сайте Tinkercad, а затем в реальной жизни. В ходе проведения испытаний были получены и занесены в таблицу следующие результаты:

Расстояние до объекта на дороге (в метрах)	Расстояние до пешехода (в метрах)		
	28	33	30
201	-	-	-
198	+	-	+
200	+	-	+

+ лампочки горят

- лампочки не горят

4. Заключение

В данной работе я рассмотрел устройство системы умного пешеходного перехода, а также её возможности и принцип работы. Я смог собрать схему и написать программу для того, чтобы система смогла успешно работать. После этого были проведены испытания, в ходе которых система показала свою эффективность. Была подтверждена гипотеза о работоспособности модели. Из этого можно сделать вывод, что система умного пешеходного перехода может быть воплощена в жизнь.

Считаю, что знания и умения, которые я получил в ходе работы с устройством, пригодятся мне в жизни.

5. Список литературы

1. А. В. Перышкин, Е. М. Гутник. — Физика. 9 кл.: учебник Дрофа, 2014.
2. Сайт <https://arduinomaster.ru/datchiki-arduino/ultrazvukovoj-dalnomer-hc-sr04/>
3. С. Монк. Програмуем Arduino. Профессиональная работа со скетчами. — СПб.: Питер, 2017.

Приложение

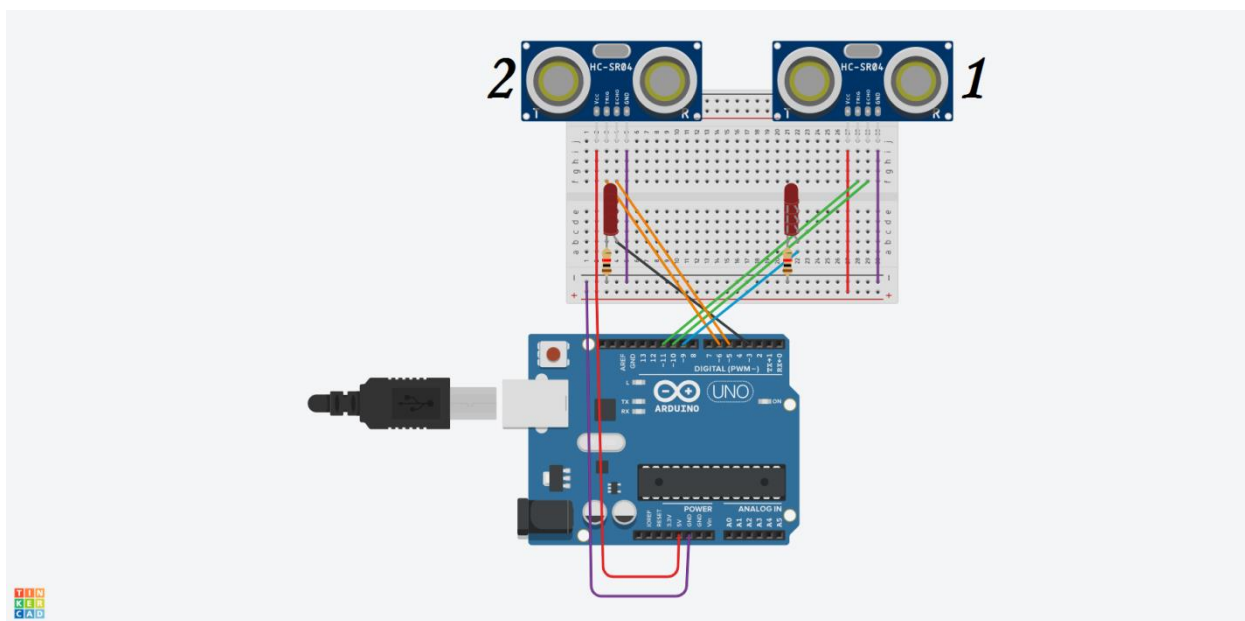


Рис. 1. Схема умной системы пешеходного перехода

1 – датчик измеряет расстояние до пешехода

2 – датчик измеряет расстояние до объекта на дороге

Полный текст программы

```
#define trigPin 11
#define echoPin 10
#define trigPin1 6
#define echoPin1 5
#define led 3
#define led1 9
void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led, OUTPUT);
  pinMode(led1, OUTPUT);
}
```



```
}  
void loop() {  
  long duration, distance, duration1, distance1 ;  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  duration = pulseIn(echoPin, HIGH);  
  distance = (duration/2) / 29.1;  
  digitalWrite(trigPin1, LOW);  
  delayMicroseconds(2);  
  digitalWrite(trigPin1, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin1, LOW);  
  duration1 = pulseIn(echoPin1, HIGH);  
  distance1 = (duration1/2) / 29.1;  
  if (distance <= 30 && duration1 <=200) {  
    digitalWrite(led, HIGH);  
    digitalWrite(led1, HIGH);  
  }  
  else {  
    digitalWrite(led, LOW);  
    digitalWrite(led1, LOW);  
  }  
  delay(1000);  
}
```