**"Устройство для ориентации слепых в пространстве в условиях городской местности"**

**Автор:**

**Белоусов Евгений Александрович**

**МГТУ им Н.Э.Баумана**

**Научный руководитель:**

**Лазарев Сергей Викторович**

**Устройство для ориентации слепых в пространстве в условиях городской местности**

**Цель проекта**

Создать прототип устройства, позволяющий слепым определять местоположение препятствий.

**Актуальность проекта**

Большую часть информации человек получает с помощью зрения. Когда в результате врожденной или приобретенной патологии человек не имеет возможности видеть мир, он теряет способность обнаруживать, а следовательно и преодолевать препятствия. Устройство, которое будет создано в ходе данного проекта, призвано помочь слепым вовремя обнаружить препятствие и, как следствие, частично вернуть возможность ориентироваться в пространстве.

**Требования к устройству**

Система должна автономно в режиме реального времени обнаруживать препятствия и сообщать об этом пользователю.

**Анализ**

Для начала введем следующие допущения:

1. Городская местность - местность, на которой находится большое количество фактурных препятствий.
2. ‎Определить местоположение препятствия — определить в каком направлении оно находится.

Начнем анализ с рассмотрения аппаратурой части системы. Любая система реального времени, состоит из сенсоров, контроллера (в нашем случае микроконтроллера) и актуаторов, частей конвейера, слаженная работа которых обеспечивает правильное функционирование системы.

Сенсор — это датчик, который получает сведения от реального мира и переводит их в понятный для программы вид.

Микроконтроллер — главный вычисляющий модуль, содержащий программу, определяющую поведение системы.

Актуатор — устройство, посредством которого программа воздействует на реальный мир (в нашем случае передает сигнал пользователю).

Как видно из требований, система должна обнаруживать препятствия, то есть она должна обладать сенсорами, с помощью которых можно определить местоположение препятствий. Существуют несколько подходов к решению данной проблемы:

1. Установка видеокамеры и последующее детектирование препятствий посредством анализа видеосигнала
2. Составление виртуальной карты местности и создание системы позиционирования в ней
3. Установка дальномеров

Все варианты имеют как свои преимущества, так и недостатки.

Например, первый вариант позволяет детектировать практически любые виды препятствий в режиме реального времени, однако для обработки видеосигнала нужен микроконтроллер с большой вычислительной мощностью, также данный метод будет работать только при условии достаточной освещенности местности.

Второй вариант способен обнаруживать любые статические виды препятствий, однако для него необходим большой объем памяти, в котором можно будет хранить карту местности. Помимо этого карта будет постоянно устаревать, что вызовет необходимость ее постоянного обновления. Позиционирование в пространстве тоже представляет проблему, так как ошибка, появляющаяся при определении местоположения со временем будет накапливаться и впоследствии станет критичной.

Третий вариант предоставляет минимальные требования к вычислительной мощности микроконтроллера. Он позволяет получать сведения об препятствиях в режиме реального времени, а значит способен к детектированию движущихся объектов и не подвержен накоплению ошибки. Однако он сильно подвержен воздействию помех.

Учитывая все плюсы и минусы, остановимся на последнем варианте. Однако, для уменьшения помех, будем использовать не один дальномер, а несколько.

После того, как система обнаружит препятствие, она должна сообщить об этом пользователю. Так как устройство ориентированно на людей с полной или частичной потерей зрения, то сообщение можно передать с помощью:

* звукового сигнала
* тактильного сигнала

Оба способа имеют право на существование. Выбор того или иного способа передачи сигнала скорее зависит от конкретного пользователя. Однако на данный момент остановимся на звуковом сигнале. Впоследствии, при необходимости, мы сможем заменить актуатор с минимальными затратами.

В качестве микроконтроллера будем использовать платформу Arduino, ввиду ее простоты, надежности и соответствия требованиям сенсоров и актуаторов, выбранных нами.

Итак, после проведения аппаратурного анализа системы, мы остановились на следующем:

* Используем платформу Arduino
* Для детектирования препятствий используем несколько дальномеров
* Для передачи сигнала об обнаружении препятствия пользователю используем звуковой сигнал.

**Проектирование**

Разобьем проектирование системы на ряд этапов:

* Проектирование подсистемы детектирования препятствий
* Проектирование подсистемы передачи сигнала пользователю
* Соединение всех подсистем

**Проектирование подсистемы детектирования препятствий**

На стадии анализа было решено использовать для детектирования препятствий несколько дальномеров. Широко применяются два типа дальномеров: ультразвуковые и инфракрасные. Инфракрасные проще в использовании и работают быстрее, однако на ультразвуковой дальномер не влияют такие факторы, как засветка от солнца или цвет объекта, и даже прозрачная поверхность будет для него препятствием. Таким образом было принято решение остановиться на ультразвуковом дальномере HC-SR04.

Ультразвуковой дальномер определяет расстояние до объектов точно так же, как это делают дельфины или летучие мыши. Он генерирует звуковые импульсы на частоте 40 кГц и слушает эхо. По времени распространения звуковой волны туда и обратно можно однозначно определить расстояние до объекта.

Дабы устранить нежелательные помехи, будем использовать показания не с одного, а с двух дальномеров и усреднять их значения. Тем самым мы максимально снизим сбои в работе нашей системы.

**Проектирование подсистемы передачи сигнала пользователю**

Дабы не усложнять нашу систему будем использовать пьезодинамик. При этом, будем включать динамик, если препятствие находится ближе некого (определенного экспериментально) порога. Данный подход позволит определять местонахождение препятствия с точностью, достаточной для того, чтобы успеть обойти его.

**Соединение всех подсистем**

Корпус устройства зависит непосредственно от предпочтений пользователя. Готовое устройство может быть размещено:

* на очках
* на руке
* на груди

Так как нашей задачей является создание прототипа, остановимся на наиболее простом втором варианте.

**Проектирование ПО**

**Схема алгоритма**

**Код**

**#include** "Arduino.h"

//прототипы функций, нужны, чтобы компилятор зннал, что эти функции есть

**int** **readSonar**(**int**, **int**);

**int** **average**(**int**, **int**);

//объявляем пины ультразвуковых дальномеров

**int** leftSonarEchoPin = 1;

**int** leftSonarTrigPin = 0;

**int** rightSonarEchoPin = 3;

**int** rightSonarTrigPin = 2;

//объявляем пин пьезодинамика

**int** piezoelectricPin = 10;

//The setup function is called once at startup of the sketch

**void** **setup**()

{

//устанавливаем Echo-пины сонаров в write режим

pinMode(leftSonarTrigPin, OUTPUT);

pinMode(rightSonarTrigPin, OUTPUT);

//устанавливаем Trig-пины сонаров в режим чтения

pinMode(leftSonarEchoPin, INPUT);

pinMode(rightSonarEchoPin, INPUT);

}

// The loop function is called in an endless loop

**void** **loop**()

{

//считываем показания дальномеров

**int** leftDistance = readSonar(leftSonarTrigPin, leftSonarEchoPin);

**int** rightDistance = readSonar(rightSonarTrigPin, rightSonarEchoPin);

//усредняем их показания дальномеров, дабы устранить возможные помехи, будем считать среднее значение

**int** distance = average(leftDistance, rightDistance);

//в зависимости от значения distance подать сигнал на динамик

**if**(distance < 40)

{

analogWrite(piezoelectricPin, 160); //динамик работает от 3 вольт, 255 - 5 вольт

}**else**

{

analogWrite(piezoelectricPin, 0); // 0 вольт: динамик не работает

}

}

**int** **readSonar**(**int** trigPin, **int** echoPin)

{

**int** duration, cm;

digitalWrite(trigPin, LOW); //посылаем 0 на trigPin

delayMicroseconds(2);

digitalWrite(trigPin, HIGH); //посылаем 1 на trigPin

delayMicroseconds(10);

digitalWrite(trigPin, LOW); //посылаем 0 на trigPin

duration = pulseIn(echoPin, HIGH); // считываем продолжительность импульса на echoPin в микросекундах

/\*

Нам потребуется умножить время на скорость звука:

s = duration \* v = duration \* 340 м/с

Переводим скорость звука из м/с в см/мкс:

s = duration \* 0.034 м/мкс

Для удобства преобразуем десятичную дробь в обыкновенную:

s = duration \* 1/29 = duration / 29

А теперь вспомним, что звук прошел два искомых расстояния: до цели и обратно.

Поделим всё на 2:

s = duration / 58

Теперь мы знаем откуда взялось число 58 в программе!

\*/

cm = duration / 58;

delay(10);

**return** cm;

}

**int** **average**(**int** left, **int** right)

{

//возвращаем среднее значение

**return** (left+right)/2;

}

**Сопровождение**

Полная реализация рассматриваемой системы является не слишком большой. Однако не стоит забывать, что это всего лишь прототип. Тем не менее, для любой работающей системы, процесс последующей модернизации неизбежен. Рассмотрим, что придется сделать, чтобы реализовать еще одно дополнительное требование для нашей системы.

Допустим, мы хотим чтобы помимо детектирования препятствий, наша система определяла местоположение пользователя и отправляла его на некий сервер (например родственникам пользователя). Для этого будет необходимо внести следующие изменения:

* установить на Arduino GPS модуль
* установить на Arduino Internet модуль
* дописать часть программы, ответственную за получение данных с GPS и отправку этих данных на сервер

При этом ни сама архитектура системы, ни основные ее механизмы не потерпят серьезных изменений.

В дальнейшем система может быть улучшена за счет установления на нее большего числа дальномеров (возможно не только ультразвуковых). Также, при усложнении задач, нам потребуется большая вычислительная мощность процессора, что вынудит нас перейти на другую платформу (например Ruspberry Pi).

**Результаты проекта**

Создан прототип устройства, которое позволяет слепым людям определять местоположений препятствий в пространстве.