

УДК 004.051

Попов Д. А., Костенко В. Л., Ядрова М. В., Кондратьев С. Б.  
Одесский национальный политехнический университет

## **РАЗРАБОТКА МУЛЬТИСЕНСОРНОГО СОНАРА ДЛЯ СЛЕПЫХ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА STM8L**

**Попов Д. А., Костенко В. Л., Ядрова М. В., Кондратьев С. Б. Разработка мультисенсорного сонара для слепых на основе микроконтроллера STM8L.** В данной статье приведены результаты разработки мультисенсорного сонара для слепых. Показана возможность расширения угла обзора, снижения энергопотребления за счет применения современной элементной базы, в том числе микроконтроллеров с минимальным энергопотреблением. Разработан алгоритм работы микроконтроллера и код программы на языке C++. Представлен макет устройства.

**Ключевые слова:** сонар для слепых; микроконтроллер; звуковой излучатель; программные коды; энергопотребление.

**Попов Д. О., Костенко В. Л., Ядрова М. В., Кондратьев С. Б. Розробка мультисенсорного сонара для сліпих на основі мікроконтролера STM8L.** В даній статті наведені результати розробки мультисенсорного сонара для сліпих. Показана можливість розширення кута огляду, зниження енергоспоживання за рахунок застосування сучасної елементної бази, в тому числі мікроконтролерів з мінімальним енергоспоживанням. Розроблено алгоритм роботи мікроконтролера і код програми на мові C++. Представлений макет пристрою.

**Ключові слова:** сонар для сліпих; мікроконтроллер; звуковий випромінювач; програмні коди; енергоспоживання.

**Popov D., Kostenko V., Yadrova M., Kondratyev S. Development of a multisensory sonar for the blind based on the STM8L microcontroller.** This article presents the results of developing a multisensory sonar for the blind. The possibility of expanding the viewing angle, reducing power consumption through the use of a modern element base, including microcontrollers with minimal power consumption, is shown. The algorithm of the microcontroller and the code of the program in C++ are developed. The device layout is presented.

**Keywords:** sonar for the blind; microcontroller; voice emitter; codes of programs; energy consumption.

Для решения важной социальной проблемы – реабилитации слабовидящих и незрячих людей требуются разработки новых и совершенствования известных средств обнаружения травматически потенциальных препятствий, находящихся в пространстве перед ними. Среди таких средств в настоящее время большое внимание отводится мобильным ультразвуковым устройствам с микроконтроллерами, интегрированными на одной плате с миниатюрными датчиками и устройствами индикации [1, 2, 3].

В настоящее время наибольшую популярность получили ультразвуковые сонары на базе микроконтроллера Arduino [4, 5]. При эффективном угле наблюдения до 15° и дальности до 1,5 м, устройство потребляет ток до 140 мА, при этом расстояние до препятствия не определяется, а среднее время автономной работы на одном заряде элемента питания составляет 8 часов. Для повышения локальности обнаружения, в работе [6] предлагаются ультразвуковые перчатки для слепых, пригодные для обнаружения объектов на расстоянии до 3,5 м с эффективным сектором наблюдения не менее 30 градусов и с интегрированным процессором Arduino Pro Mini 328. При энергопотреблении от 33 мВт до 100 мВт, время автономной работы устройства составляет 4–8 часов в зависимости от используемой платы. Устройство удобно для использования внутри помещений, однако не оптимально на улице из-за необходимости постоянно направлять руки с перчатками в сторону предполагаемых препятствий. Для обнаружения препятствий на уровне подстилающей поверхности и расстояния до них в [7] предлагается ультразвуковая «умная трость» для незрячих на базе микроконтроллера семейства Arduino. Обладая расширенными функциональными возможностями данное устройство, вместе с тем, обеспечивает невысокий эффективный угол обзора – порядка 14–30 градусов и обладает недостаточно большим сроком работы в автономном режиме без подзарядки элемента питания. С целью повышения информативности обнаружения, в работе [8] предлагается использование системы обнаружения и позиционирования с применением трех ультразвуковых сенсоров для определения препятствия на пути слепых в трех различных направлениях, GPS и система сообщения СМС, микроконтроллер Arduino, зуммер и генератор вибраций. Недостатком систем обнаружения препятствий для слепых на базе этих плат является их сравнительно высокое энергопотребление и, соответственно, малое время работы на одном заряде элемента питания, необходимость частого заряда устройства.

Проведенные нами исследования показали, что одним из основных факторов, ограничивающих применение известных сонаров для слепых, является большое потребление мощности микроконтроллером Arduino, которое, в зависимости от применяемой платы, меняется

от 33 мВт (плата Arduino LilyPad USB, питание от аккумулятора 3,7 В) до 105 мВт (плата Beetle Leonardo, при питании от аккумулятора 5 В) [9].

Поэтому исследования по энергосбережению сонаров для слепых являются важной научно-прикладной **проблемой**, которая может быть решена на основе применения современной элементной базы, в том числе микроконтроллеров с минимальным энергопотреблением [11, 12], и разработки соответствующего программного обеспечения.

**Изложение основного материала.** С учетом проведенных исследований были разработаны программное обеспечение и макет сонара для слепых, который включает два ультразвуковых сенсора HC-SR04, микроконтроллер STM8L с низким энергопотреблением, разработанный специально для изделий с батарейным питанием, и стандартный малогабаритный элемент питания на 3 В. Конструктивно предусмотрена возможность заряда элемента питания через USB интерфейс. В качестве индикатора направления на препятствие и расстояния до него, с учетом минимизации энергозатрат, был выбран звуковой излучатель Piezo Buzzer 17855 с током потребления не более 2 мА.

Программным путем обеспечивается последовательный опрос сенсоров и возможность перехода неактивного датчика в режим тишины до следующего акта измерений с целью снижения энергопотребления и экономии заряда батареи.

Схема алгоритма работы микроконтроллера с двумя сенсорами представлена на рис. 1.



Рисунок 1 – Алгоритм работы микроконтроллера с двумя сенсорами

В процессе работы сонара микроконтроллер формирует короткий запускающий импульс, который подается поочередно на вход каждого ультразвукового датчика. На выходе датчиков получаем импульс, длительность которого прямо пропорциональна расстоянию до объекта. Микроконтроллер преобразует полученный импульс в серию импульсов, длительность которых постоянна, а частота следования обратно пропорциональна расстоянию до объекта. Эти импульсы поступают на один из звуковых излучателей, смонтированных на плате совместно с соответствующим датчиком. По частоте следования звукового сигнала можно ориентировочно определить удаленность до объекта на слух, а по номеру звукового излучателя – направление на препятствие.

Код программы написан на языке C++. Основной фрагмент кода программы представлен на рис. 2. Звуковые излучатели расположены в блоке с датчиками и индицируют расстояние до препятствия паузами между звуковыми сигналами. Индикация заканчивается в тот момент, когда индикация самого длинного расстояния будет произведена, после этого программа производит переход на начало измерений.

```
void UltrasonicSensor::BeepersActivation()
{
    if(echoLengthFirstSensor <= echoBackFront && echoLengthSecondSensor <= echoBackFront)
    {
        distanceFirstObstacle = echoLengthFirstSensor / soundSpeed;
        distanceSecondObstacle = echoLengthSecondSensor / soundSpeed;
        if(distanceFirstObstacle <= distanceOneMeter && distanceSecondObstacle <=
            distanceOneMeter)
        {
            if(distanceFirstObstacle < distanceSecondObstacle)
            {
                /* Determine the number of beeps */
                uint32_t temp = (echoLengthSecondSensor * delayAfterBeep + delayBetweenBeep)
                /(echoLengthFirstSensor * delayAfterBeep + delayBetweenBeep);
                /* Activating the beeper */
                GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);
                Delay(delayBetweenBeep);
                GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_RESET);
                for(uint32_t i = 0; i < temp+1; ++i)
                {
                    /* Activating the beeper */
                    GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_SET);
                    Delay(delayBetweenBeep);
                    GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_RESET);
                    Delay(echoLengthFirstSensor * delayAfterBeep);
                }
            }
            else if(distanceSecondObstacle <= distanceFirstObstacle)
            {
                /* Determine the number of beeps */
                uint32_t temp = (echoLengthFirstSensor * delayAfterBeep + delayBetweenBeep)
                /
                (echoLengthSecondSensor * delayAfterBeep + delayBetweenBeep);
                /* Activating the beeper */
                GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_SET);
                Delay(delayBetweenBeep);
                GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_RESET);
                for(uint32_t i = 0; i < temp+1; ++i)
                {
                    /* Activating the beeper */
                    GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);
                    Delay(delayBetweenBeep);
                    GPIO_WriteBit(GPIOC, GPIO_Pin_11,
                    Bit_RESET);
                    Delay(echoLengthSecondSensor * delayAfterBeep);
                }
            }
        }
    }
}

else if(echoLengthFirstSensor <= echoBackFront && echoLengthSecondSensor > echoBackFront)
{
    distanceFirstObstacle = echoLengthFirstSensor / soundSpeed;
    if(distanceFirstObstacle <= distanceOneMeter)
    {
        /* Activating the beeper */
        GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_SET);
        Delay(delayBetweenBeep);
        GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_RESET);
        Delay(echoLengthFirstSensor * delayAfterBeep);
    }
}

else if(echoLengthFirstSensor > echoBackFront && echoLengthSecondSensor <= echoBackFront)
```

```
{  
    distanceSecondObstacle = echoLengthSecondSensor / soundSpeed;  
    if(distanceSecondObstacle <= distanceOneMeter)  
    {  
        /* Activating the beeper */  
        GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);  
        Delay(delayBetweenBeep);  
        GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_RESET);  
        Delay(echoLengthSecondSensor * delayAfterBeep);  
    }  
}
```

Рисунок 2 – Основной фрагмент кода программы

Внешний вид макета сенсора с двумя датчиками представлен на рис. 3



Рисунок 3 – Макет сенсора с двумя ультразвуковыми датчиками

Ультразвуковые датчики расположены под углом друг к другу. Такое расположение датчиков по сравнению с аналогами [4], обеспечивает увеличение эффективного угла обзора в горизонтальной плоскости.

Предложенное программное обеспечение позволяет наращивать количество датчиков и звуковых излучателей, расположенных на одной плате с микроконтроллером, без увеличения потребляемого тока, с целью расширения угла обзора сенсора. Разработаны алгоритм работы, программа и макет сенсора с тремя сенсорами и тремя звуковыми излучателями. Алгоритм работы микроконтроллера для устройства с тремя сенсорами представлена на рис. 4.



Рисунок 4 – Алгоритм работы микроконтроллера с тремя сенсорами

Програма базується на основному фрагменті (рис. 5)

```
void UltrasonicSensor: BeepersActivation()
{
    if(echoLengthFirstSensor <= echoBackFront && echoLengthSecondSensor <= echoBackFront &&
        echoLengthThirdSensor <= echoBackFront)
    {
        distanceFirstObstacle = echoLengthFirstSensor / soundSpeed;
        distanceSecondObstacle = echoLengthSecondSensor / soundSpeed;
        distanceThirdObstacle = echoLengthThirdSensor / soundSpeed;
        if(distanceFirstObstacle <= distanceOneMeter &&
            distanceSecondObstacle <= distanceOneMeter &&
            distanceThirdObstacle <= distanceOneMeter)
        {
            if(distanceFirstObstacle < distanceSecondObstacle &&
                distanceSecondObstacle < distanceThirdObstacle)
            {
                /* Determine the number of beeps */
                uint32_t tempFirst = (echoLengthThirdSensor * delayAfterBeep +
                    delayBetweenBeep) /
                    (echoLengthFirstSensor * delayAfterBeep + delayBetweenBeep);
                uint32_t tempSecond = (echoLengthThirdSensor * delayAfterBeep +
                    delayBetweenBeep) /
                    (echoLengthSecondSensor * delayAfterBeep + delayBetweenBeep);
                /* Activating the beeper */
                GPIO_WriteBit(GPIOC, GPIO_Pin_13, Bit_SET);
                Delay(delayBetweenBeep);
                GPIO_WriteBit(GPIOC, GPIO_Pin_13, Bit_RESET);
                for(uint32_t i = 0; i < tempSecond+1; ++i)
                {
                    /* Activating the beeper */
                    GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);
                    Delay(delayBetweenBeep);
                    GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_RESET);
                    Delay(echoLengthSecondSensor * delayAfterBeep);
                }
                for(uint32_t i = 0; i < tempFirst+1; ++i)
                {
                    /* Activating the beeper */
                    GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_SET);
                    Delay(delayBetweenBeep);
                    GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_RESET);
                    Delay(echoLengthFirstSensor * delayAfterBeep);
                }
            }
        }
    }
}
```

Рисунок 5 – Основной фрагмент программной реализации работы микроконтроллера с тремя сенсорами

Проведен анализ зависимости потребляемого тока от тактовой частоты микроконтроллера и выбрана оптимальная тактовая частота 1 МГц, при этом ток потребления снижается с 3,54 мА на частоте 16 МГц до 0,54 мА на частоте 1 МГц.

В результате испытаний установлено, что суммарный ток потребления устройства составляет 17,7 мА. При использовании литий-полимерного аккумулятора емкостью 2800 мАч, время работы в автономном режиме без подзарядки элемента питания составляет 80 часов, т.е. не менее 10 дней, при этом обеспечивается уверенная фиксация препятствий в диапазоне расстояний от 0,02 до 3 м в пределах диаграммы направленности ультразвуковых датчиков.

Испытания подтвердили, что энергопотребление сонара по сравнению с аналогами [4–10] снижено, а время работы без подзарядки элемента питания увеличено в 4–5 раз.

Наличие двух и трех ультразвуковых датчиков позволяет расширить угол обзора и обеспечить панорамность «зрения», т.е. возможность определять наличие препятствий прямо перед человеком, а также слева и справа от него.

**Выводы.** На основе проведенных исследований можно сделать следующие выводы:

1. Проведен анализ устройств ультразвукового обнаружения препятствий для слепых.
2. На основании сравнительного анализа, был выбран микроконтроллер с низким энергопотреблением STM8L и оптимальная тактовая частота микроконтроллера.
3. Разработан алгоритм работы микроконтроллера STM8L и программа работы на языке C++. Приведены результаты разработки программной модели и кода программы.
4. Собраны и испытаны макеты сонаров с двумя и тремя сенсорами.
5. Показана возможность наращивания числа сенсоров.
6. Расширен угол обзора сонара за счет наличия нескольких сенсоров.
7. Увеличено время работы устройства в автономном режиме без подзарядки элемента питания за счет применения современной элементной базы, в том числе микроконтроллеров с минимальным энергопотреблением, и разработки соответствующего программного обеспечения.

1. Ершов, С. О. Системы технического зрения с невизуальным представлением пространственной информации / С. О. Ершов и П. Майер // *Известия вузов. Приборостроение*, Т. 51, № 1. С. 42–47, 2008.
2. Ершов, С.О. Локальные и глобальные программно-аппаратные средства позиционирования и навигации для слепых / С.О. Ершов, П. Мейер, Е. Л. Баки-Бородов // *Информатизация и связь*, № 1, с. 58–62, 2011.
3. Jayant, Sakhardande. Arduino Based Mobility Canel / Jayant Sakhardande, Pratik Pattanayak, Mita Bhowmick // *International Journal of Scientific & Engineering Research*, Vol. 4, Issue 4, pp 1163–1166, (April 2013).
4. Бюджетное устройство на базе Arduino для незрячих [Электронный ресурс] / Режим доступа: <https://future.d3.ru/biudzhethnoe-ustroistvo-na-baze-arduino-dlia-nezriachikh-open-hardware-750859>.
5. Ультразвуковые фонарики «Сонар-5УФ» [Электронный ресурс] / Режим доступа: <https://www.sites.google.com/site/tehniceskiesredstvadalaslepyh/1-ultrazvukovoj-fonar-sonar-5uf>.
6. Сонар для слепых на базе Arduino Mini Pro [Электронный ресурс] / Режим доступа: <http://meandr.org/archives/21701>.
7. Shubham, Adhe. Ultrasonic Smart Stick for Visually Impaired People / Shubham Adhe, Sachin Kunthewad, Preetam Shinde, Mrs.V.S.Kulkarni // *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)* e-ISSN: 2278-2834,p- ISSN: 2278-8735.PP 11–15 [www.iosrjournals.org](http://www.iosrjournals.org).
8. Osama, Bader. 3D Ultrasonic Stick for Blind" / Osama Bader, AL-Barrm, Jeen Vinouth // *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, Vol.3, No.3, pp. 108–114, January 2014.
9. Энергопотребление Arduino [Электронный ресурс] / Режим доступа: [https://www.arduino.md/arduino\\_power\\_consumption](https://www.arduino.md/arduino_power_consumption).
10. Sylvain, Cardin. Wearable Obstacle Detection System for visually impaired peopl / Sylvain Cardin, Daniel Thalmann and Frederic Vexo // *Virtual Reality Laboratory (VRlab)Ecole Polytechnique Fédérale de Lausanne (EPFL)CH-1015 Lausanne, Switzerland{sylvain.cardin, daniel.thalmann, frederic.vexo} @epfl.ch*.
11. Костенко, В. Л. Информационно-измерительная система контроля нормированных параметров производственных факторов / Костенко В. Л., Николенко А. А., Поперека Е. Д., Ядрова М. В., Тьманюк К. С. // *Східно-європейський журнал передових технологій. Серія «Інформаційно-управляючі системи»* Т. 3, № 9 (69), С. 51–56, 2014.
12. Костенко, В. Л. Информационная система измерения и контроля параметров воздушной среды / Костенко В. Л., Кондратьев С. Б., Ядрова М. В., Попов Д. А. // *Вісник НТУ «ХПІ». Серія: «Нові рішення в сучасних технологіях»*. – Харків НТУ ХПІ, 2017 – № 23 (1245). – С. 117–122.