Техническая документация на сенсор для звуковой локализации на основе SLLmic

1. Краткий обзор сенсора

Сенсор для звуковой локализации на основе SLLmic (sound localization library with microphones) представляет собой две плоские квадратные микрофонные решётки, соединённые между собой. Модуль обладает следующими функциональными возможностями:

* Измерение угловых координат источника звука в пространстве
* Программирование сенсора для выбора различных режимов измерений
* Определение координат источника звука
* Взаимодействие со сервоприводами для визуализации направления на источник звука
* Взаимодействие с lcd дисплеем для выводы значений координат расположения источника звука
* Взаимодействие с компьютером для передачи измеренных результатов посредством USB

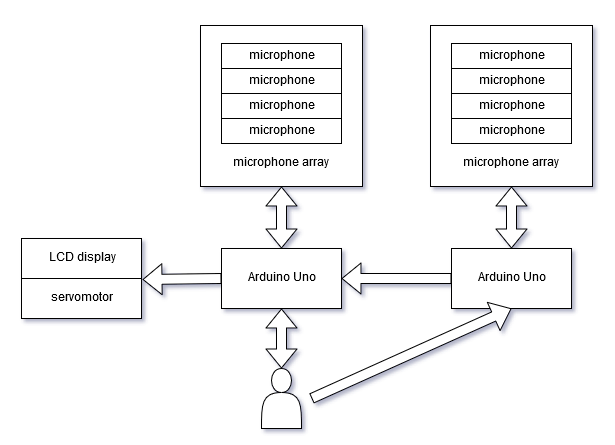


Рис 1. Блок-схема сенсора SLLmic

Основными элементами данного сенсора являются платаArduino Uno и микрофонная решётка, построенная на микрофонах на основе операционного усилителя LM358. Микроконтроллерная плата обладает следующими характеристиками:

* Микроконтроллер: ATmega328
* Тактовая частота: 16 МГц
* Напряжение логических уровней: 5 В
* Портов ввода-вывода общего назначения: 20
* Портов, подключённых к АЦП: 6
* Разрядность АЦП: 10 бит
* Flash-память: 32 КБ
* EEPROM-память: 1 КБ
* Оперативная память: 2 КБ

В качестве микрофонов использовался Arduino analog sound sensor v2, обладающий следующими характеристиками:

* Напряжение питания: 3.3–5 В
* Размер: 22х32 мм
* Схема усилителя на основе операционного усилителя LM358



Рис 2. Arduino analog sound sensor v2

Так же возможно подключение различных средств визуализации, речь о которых пойдёт далее

1. Подключение сенсора

Для включения сенсора необходимо выполнить ряд шагов по подключению его компонентов:

1. Скачать программу Arduino Ide c официального сайта <https://www.arduino.cc/en/main/software>.
2. Скачать библиотеку SLLmic с сайта <https://github.com/roanlane/SLLmic>. Загруженную библиотеку необходимо сохранить в папку …\Arduino\libraries\SLL
3. Установить микроконтроллер и микрофоны на пластиковой платформе. Элементы платформа можно скачать на <https://github.com/roanlane/SLLmic> м распечатать на 3D принтере
4. Подключить микрофоны к микроконтроллерной платформе. При этом можно использовать общее питание и общее заземление для всех 4 микрофонов. Аналоговые выходы микрофонов нужно подключить в соответствующие аналоговые входы микроконтроллера, их порядок задаётся в строке ***int pin\_array[4] = {0 ,1 ,2 ,3}***
5. Подключить микроконтроллеры друг с другом, о чём подробнее рассказывается в пункте 4 данного документа
6. Подключить средства вывода и визуализации полученных данных, о чём подробнее рассказывается в пунктах 5 и 6
7. Загрузить скетч [***SLL\_serv\_and\_lcd.ino***](https://github.com/roanlane/SLLmic/blob/master/SLL_serv_and_lcd.ino) на микроконтроллеры и запустить его.

Пример собранного сенсора показа на рисунке ниже.

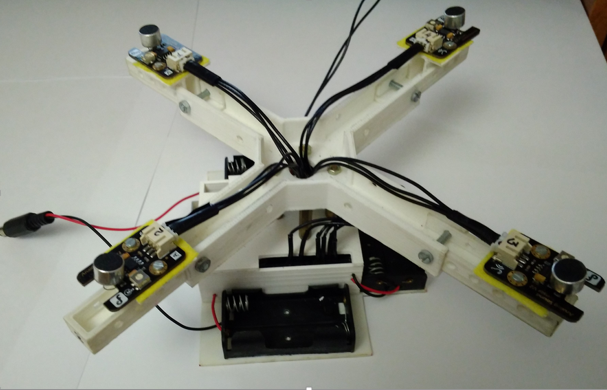


Рис 1. Внешний вид сенсора

1. Режимы работы сенсора

Данный сенсор может функционировать в четырёх основных режимах работы. Для трёх из них достаточно одной микрофонной решётки:

1. Вычисление угловых координат методом кросс-корреляции
2. Вычисление угловых координат методом определения фазовых задержек
3. Аналогично 2., но с учётом ограничения по диапазону принимаемых частот

При использование двух микрофонных решёток возможно:

1. Вычисление координат источника звука

Выбор метода вычисление осуществляется с помощью метода Initialization

**Initialization(f\_division\_factor , f\_distanse\_be\_mic , f\_voltage\_thres , f\_type\_algorithm , f\_min\_filter\_limit , f\_max\_filter\_limit)**

***f\_division\_factor -*** задаёт частоту дискретизации АЦП микроконтроллера Arduino Uno, меньше 4 и больше 16 в данном алгоритме выбирать не рекомендуется – в первом случае точность оцифровки будет недостаточной, во втором мала разрешающая способность микрофонной решётки. Возможные значения: 2 4 8 16 32 64 128

***f\_distanse\_be\_mic*** – минимальное расстояние между микрофонными решётками, возможно изменять данное расстояние посредством регулирования длины плеч микрофонной решётки. В данном алгоритме конфигурация решётки должна оставаться квадратной

***f\_voltage\_thres -*** на сколько отчётов квантовался сдвинут синусоидальный сигнал, если -1, то число считается автоматически. Параметр нужен для сдвижки по напряжению данных, снятых с микрофонов при необходимости

***f\_type\_algorithm*** - тип алгоритма определения углов

***CROSS\_CORRELATION*** или ***0*** - кросс-корреляция

***FFT\_PHASE*** или ***1*** - по фазовым задержкам

***FFT\_FILTER*** или ***2*** - по фазовым задержкам с учёта ограничения по частотам:

***f\_min\_filter\_limit*** - минимальная пропускаемая частота в Гц, если используется алгоритм отличный от FFT\_FILTER, параметр можно не задавать

***f\_max\_filter\_limit*** - максимальная пропускаемая частота в Гц, если используется алгоритм отличный от FFT\_FILTER, параметр можно не задавать

Для вычисления угловых координат источника звука используется метод

**Find\_angle(f\_val\_thres)**

***f\_val\_thres –*** величина порога срабатывания алгоритма в отчётах квантователя, на звук меньшей величины алгоритм не реагирует, выбирается экспериментально в зависимости от характера отслеживаемого сигнала, общего уровня шума и чувствительности микрофонов – в случае отсутствия шума рекомендуется выбрать 40-60, что соответствует 20-30мВ, при наличие шума – 80-100 – 40-50мВ. Так же нужно отметить, что увеличение порога приводит к понижению точности работы алгоритма из-за пропуска части информационного сигнала алгоритмом.

В случае определения расстояния до источника звука - необходимо определить угловые координаты источника звука на обеих микрофонныхрешётках, передать данные на один микроконтроллер и воспользоваться методом

**Find\_distance(angle\_phi\_trans, angle\_qu\_trans, distanse\_be\_mic\_arr)**

***angle\_phi\_trans*** – значение азимутального угла, рассчитанное на дополнительной микрофонной решётке

***angle\_qu\_trans*** - значение угла наклона, рассчитанное на дополнительной микрофонной решётке

***distanse\_be\_mic\_arr*** – расстояние между центрами микрофонных решёток, вычисляемое в см.

1. Коммуникация микроконтроллеров

Для коммуникации между микроконтроллерами используется библиотека SoftwareSerial позволяющая реализовать последовательный интерфейс на любых цифровых выводах Arduino с помощью программных средств, дублирующих функциональность UART. Библиотека позволяет программно создавать несколько последовательных портов, работающих на скорости до 115200 бод. Приведём пример используемой конфигурации

***SoftwareSerial mySerial(7, 8);***

***mySerial.begin(9600);***

В данном случае 7 пин микроконтроллерной платы настроен на приём информации, а 8 - на её передачу. Так же необходимо указать, что из двух микроконтроллеров один был выбран в качестве основного, осуществляющего вычисление координат источника звука, информация об угловых координатах при этом предавалась на данный микроконтроллер. Для выбора микроконтроллера в качестве основного, необходимо записать ***bool is\_master = true***. На втором микроконтроллере при этом должно быть задано ***bool is\_master = false***. Скорость передачи данных между микроконтроллерами составляет 9600 бод/c.

Вектор передаваемых данных от второго микроконтроллере к основному имеет следующий вид

[«,» , //100 , //10 , %10 , «,» , //10 , %10],

− угол наклона, округлённый до целочисленного значения,

− азимутальный угол, округлённый до целочисленного значения.

При этом символы будут переданны в виде соответствующих им ASCII символов – к примеру, для =150° и = 62°, то передаваемый вектор данных будет иметь вид 2C 31 35 30 2C 36 32.

1. Взаимодействие с сервоприводами

Разработанный сенсор может визуализировать направление на источник звука посредством двух сервоприводов. При этом один из сервоприводов задаёт азимутальный угол, а второй – угол наклона. В качестве сервоприводов использовались микросервоприводы FS90



Рис 4. Внешний вид сервопривода FS90

Данный сервопривод обладает следующими характеристиками:

* Диапазон вращения: 180°
* Напряжение питания: 4,8–6 В
* Крутящий момент: 1,3 кг·см при 4,8 В
* Скорость вращения: 60° за 0,12 сек при 4,8 В
* Потребляемый ток: 200 мА (макс. 500 мА)
* Габариты: 23,2×12,5×22 мм
* Вес: 9 г

Для подключения сервоприводов к основному микроконтроллеру используется библиотека Servo. Приведём пример используемой конфигурации:

***Servo myservo; -*** создаётся экземпляр класса для управления сервоприводами. Достаточно создать один экземпляр для двух приводов

***int serv\_az = 9;***

***int serv\_ze = 10;***

В данном случае управление сервоприводом, отвечающим за азимутальный угол включено в 9 пин микроконтроллера, угла наклона – в 10 пин. Для визуализации результатов используется функция ***rotation\_serv(),*** поворачивающий систему из двух сервоприводов в соответствие с вычисленными угловыми координатами.

1. Взаимодействие с lcd-дисплеем

Разработанный сенсор может визуализировать полученные координаты источника звука посредством LCD дисплея. В качестве дисплея используется Grove - LCD RGB Backlight



Рис 5. Внешний вид дисплея Grove - LCD RGB Backlight

Данный дисплей имеет следующие характеристики:

* Количество символов и строк: 16\*2
* Напряжение питания: 5 В
* Подсветка дисплея: RGB
* Коммуникация с микроконтроллером: посредством I2C с использованием двух пинов микроконтроллера
* CGROM: 10880 bit
* CGRAM: 64\*8 bit

Для подключения дисплея используется библиотека rgb\_lcd, которую можно скачать с официального сайта проекта <https://github.com/Seeed-Studio/Grove_LCD_RGB_Backlight>. Библиотеку необходимо будет сохранить в папку …\Arduino\libraries\rgb\_lcd. Приведём пример используемой конфигурации:

***rgb\_lcd lcd;*** - создание экземпляра класса для передачи данных дисплею

***lcd.begin(16, 2);*** - инициация дисплея с 2 строками по 16 символов в каждой

Для вывода информации используется функция **lcd\_print();.** Формат данных, выводящихся на дисплей имеет следующий вид

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X |  | //100 | //10 | %10 |  | Y |  | //100 | //10 | %10 |  | Z |  | //10 | %10 |
| a | //100 | //10 | %10 |  | //100 | //10 | %10 | q | //100 | //10 | %10 |  | //100 | //10 | %10 |

Где Х,Y,Z – декартовы координаты источника звука в пространстве относительно центра главной микрофонной решётки

a,q – азимутальный угол и угол наклона, снятые с микрофонных решёток, при этом сначала выводятся углы снятый с основного микроконтроллера а затем выводятся углы, полученные с дополнительной микрофонной решётки.