

## FM радиоприёмник на МК ATmega8 и RDA5807M с OLED дисплеем SSD1306.

Программа для МК ATMEGA8, который работает на частоте внутреннего генератора 8 МГц. Версия программы: **FM\_RDA5807\_SSD1306\_v3.2.**

Основные функции программы:

- Задание режима настройки по умолчанию в файле User\_set.asm.
- Поиск станций из списка при вращении энкодера и настройка на станцию после остановки вращения.
- Автопоиск станций вверх и вниз по диапазону поворотом энкодера по часовой или против часовой стрелки.
- Отображение процесса поиска станции путём вывода на дисплей номера канала или начальной частоты настройки.
- Переключение из режима перебора сохранённых станций в автопоиск и обратно производится по нажатию кнопки энкодера.
- Опознавание названия станции по её частоте при автопоиске.
- Вывод на индикатор частоты и названия станции.
- Запись последней частоты настройки в память EEPROM.
- При включении начинает работу на сохранённой частоте.
- Индикация уровня сигнала на дисплее четырьмя столбиками.
- Индикация "Моно-Сtereo" светодиодом.
- Гашение дисплея через 16 сек. после нажатия на любую из кнопок.
- Регулировки громкости нет.

Распределение портов микропроцессора:

PC0 - Входной сигнал от энкодера

PC1 - Входной сигнал от энкодера

PC2 - Кнопка энкодера

PC3 -

PC4 - (SDA) - Двухнаправленная шина данных I2C

PC5 - (SCL) - Двухнаправленная шина тактирования I2C

PD7 - Св. диод индикатор стерео-сигнала



Особенностью данной версии программы является поиск нужной станции при вращении ручки энкодера. При этом в нижней строке дисплея отображается увеличивающийся или уменьшающийся номер выбираемого канала в том порядке, в котором он записан пользователем в перечне радиостанций. При появлении на дисплее нужного номера канала, вращение энкодера нужно прекратить, после чего производится настройка на выбранную станцию. Эта функция особенно будет удобна в городах с большим количеством вещающих станций. Так же для удобства поиска, можно при составлении списка поместить перед названием станции её номер в списке.

Другой особенностью программы является вывод на дисплей начальной частоты, с которой начинается автопоиск станции в режиме

автопоиска. Какого-то особого преимущества это не даёт, но позволяет визуально контролировать направление будущего поиска и несколько «оживляет» экран дисплея.

Кроме того, введена возможность по желанию радиолюбителя перед прошивкой выбрать режим настройки на станцию по умолчанию, т. е. при включении радиоприёмника будет установлен или режим перебора станций или режим автопоиска, в зависимости от значения константы **Seek\_Mode**, которую нужно установить в файле **User\_set.asm**.

В целях продления срока службы батареи питания светодиодный индикатор стереосигнала можно не устанавливать.

В памяти МК записаны частоты и названия радиостанций, вещающих в моём городе. Их легко можно изменить на действующие в вашей местности, не вдаваясь в детали работы программы и затем перекомпилировать программу.

Размеры дисплея невелики, и шрифт размером 6x8 пикселей читается на нём с трудом. Поэтому для улучшения читаемости был применён шрифт удвоенной ширины, из-за чего в строке помещается только 10 символов. Так, что приходится, используя сокращения, вписать название станции в эти 10 символов.

Для того, чтобы прописать частоту и название станций в вашем городе, нужно открыть ассемблерный файл программы **User\_set.asm** в любом текстовом редакторе, можно даже в блокноте, но удобнее это будет сделать в привычной вам среде программирования. Здесь вы увидите строку вида:

```
.equ Nst = 15 ;Число местных станций
```

Вместо числа 15 поставьте своё число радиостанций, вещающих в вашей местности. Далее идут строки данных с перечнем радиостанций и их частот, например, такая строка:

```
.DB 140, ',', 'Ш', 'а', 'н', 'с', 'о', 'н', ',', ' ', 0 ;Радио Шансон
```

После директивы "DB" нужно ввести число шагов настройки частоты (один шаг равен 100 КГц). Это число определяется как разность между частотой радиостанции представленной в виде числа шагов и нижней границей диапазона равной 870 шагов (87 МГц). Например, для частоты 101,2 МГц получаем:

$$1012 - 870 = 142$$

Далее через запятую в одинарных кавычках идут 10 символов названия станции, в том числе пробелы, при помощи которых вы можете отцентрировать название на экране. Заканчивается строка нулём, который является признаком окончания вывода сообщения на дисплей. После точки с запятой идёт ваш комментарий, который при компиляции игнорируется, поэтому можно писать там, что угодно.

Чтобы получить HEX-файл из ассемблерной программы, нужно создать проект в удобной для вас среде программирования, например, в программе *AVRStudio*, загрузить в него все исходники и скомпилировать проект, нажав на кнопку *Build*. Существуют и другие программы, например, *Proteus*. Для этой программы я уже создал соответствующий проект и Вам остаётся только отредактировать файл **User\_set.asm**, в котором прописать свои радиостанции, запустить *Proteus* и выполнить на команду *Build All*.

При программировании МК необходимо запрограммировать следующие фьюзы: CKSEL0, CKSEL1, CKSEL3 и SUT0. (см. скриншот в статье).

Ниже приведена схема радиоприёмника, содержащая минимальное количество деталей. Схема радиоприёмника рассчитана на питание от одного литий-ионного элемента напряжением 3,7 В.

Напряжение аккумулятора через выключатель питания S1 подаётся на стабилизатор напряжения на 3,3 В. Я применил для этой цели недорогой стабилизатор типа XC6206P332MR, он имеет очень малые размеры (корпус SOT-23), но способен отдавать ток в

нагрузку до 150 мА. При снижении входного напряжения ниже напряжения стабилизации (3,3 В) падение напряжение на стабилизаторе в радиоприёмнике составляет всего 11 мВ.



На корпусе стабилизатора нанесена маркировка: 662K. Необходимо отметить, что при снижении напряжения питания ниже 3,1 В на некоторых каналах приёма появляются шумы, напоминающие мощные промышленные помехи. Из-за недостатка времени я не стал выяснять причину этого явления. Но с другой стороны этот шум является признаком того, что пора ставить аккумулятор на подзарядку, чтобы не допустить его чрезмерного разряда.

**asvetp**

