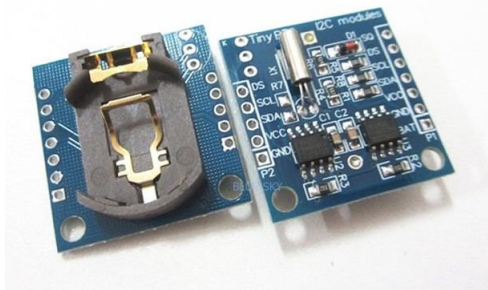


FM радиоприёмник на МК ATmega8 и модуль с RDA5807M.

(с часами на микросхеме DS1307 и термометром на DS18B20)

Выбор микросхемы RTC DS1307 был обусловлен её популярностью и доступностью. Но так как она требует питание в диапазоне 4,5...5,5 В, это обстоятельство ограничивает её применение в приёмниках с низковольтным питанием от аккумуляторов напряжением 3,7 В.



Чтобы не тратить время на поиск компонентов, я приобрёл на Aliexpress вот такой готовый модуль всего за 40 центов. В состав модуля кроме держателя батарейки, цепей её подзарядки, кварца и т.п., входит микросхема памяти, к тому же есть возможность запаять датчик температуры DS18B20.

Так как я не счёл нужным подзаряжать батарейку, я удалил из платы элементы D1, R4, R5 и R6, после чего поставил на место R6 перемычку (см. схему модуля в конце описания на рис.1).

Нельзя забывать, что этот модуль, как и любое другое цифровое устройство, является потенциальным источником радиопомех. Поэтому при разработке печатной платы радиоприёмника нужно строго придерживаться правил обеспечения электромагнитной совместимости (ЭМС), в том числе: располагать блокировочный конденсатор максимально близко к модулю, а сам модуль в максимальном отдалении от антенны радиоприёмника. В отдельных случаях, возможно придётся экранировать весь модуль.

Так как на плате модуля часов предусмотрено место для подключения датчика температуры, то было решено добавить в радиоприёмник ещё и измерение окружающей температуры. Для этого нужно обеспечить контактирование датчика DS18B20 с окружающим воздухом. В случае применения такого модуля, не нужно устанавливать резисторы R3...R5, т.к. эти резисторы уже присутствуют на плате модуля. Нет никаких проблем при использовании такого датчика и в случае, если часы будут собраны из отдельных элементов.

Программа составлена для МК ATMEGA8, работающего на частоте внутреннего генератора 8 МГц. Версия программы: **Radio_M8_N5110_DS1307_v3.**

Основные функции программы:

- Индикация температуры, измеряемой датчиком DS18B20.
- Индикация часов и минут увеличенным шрифтом под частотой.
- Переход в режим установки времени по длительному нажатию на кнопку CH+.
- Установка часов и минут кнопками V+ и V-.
- Переключение между установкой часов и установкой минут кнопкой CH+.
- Запись установленного времени и выход из режима его установки кнопкой CH-.
- Перебор станций, сохранённых в памяти программ вверх и вниз по списку, а также автопоиск станций вверх и вниз по диапазону кнопками CH+ или CH-.
- Переход из режима ручного выбора радиостанции в режим автопоиска и обратно производится длительным нажатием на кнопку CH-.
- Опознавание названия станции по её частоте при автопоиске.
- Вывод на индикатор частоты и названия станции.
- Запись последней частоты настройки и уровня громкости в память EEPROM.
- При включении начинает работу на сохранённой частоте с предыдущим уровнем громкости.
- Непрерывное изменение громкости при удержании одной из кнопок V+ или V-.
- Полное выключение громкости при уровне равном 00 (режим MUTE).
- Индикация уровня громкости на дисплее цифрами от 00 до 16.
- Индикация уровня сигнала на дисплее четырьмя столбиками.
- Индикация "Моно-Стере" как на дисплее, так и светодиодом.
- Индикация уровня заряда батареи как пиктограммой, так и цифрами.

- Подсветка дисплея при включении и при нажатии на любую кнопку в течение 10 сек
- Управление схемой полного отключения питания.

Распределение портов микропроцессора:

PB0 - Св. диод индикатор стерео-сигнала
 PB1 - D/C Выбор режима передачи данных (1) или команды (0) LCD
 PB2 - CE Выбор чипа (0) LCD
 PB3 - DIN Входные данные LCD
 PB4 - RES Линия сброса LCD (0)
 PB5 - CLK Линия тактирования LCD
 PB6 - Выходной сигнал включения питания
 PB7 - Входной сигнал отключения питания

PC0 - Вход АЦП для измерения напряжения батареи
 PC1 - Линия ввода-вывода данных температурного датчика DS18B20
 PC4 - (SDA) - Двухнаправленная шина данных I2C
 PC5 - (SCL) - Двухнаправленная шина тактирования I2C

PD0 - Кнопка V+
 PD1 - Кнопка V-
 PD2 - Вход внешнего прерывания 1 Гц от микросхемы часов
 PD3 - Кнопка CH+
 PD4 - Кнопка CH-
 PD5 - Джампер для отображения напряжения батареи на дисплее
 PD6 - Выходной инверсный сигнал подсветки дисплея
 PD7 - Выходной прямой сигнал подсветки дисплея

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.

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,'','R','a','d','i','o',' ','S','h','a','n','s','o','n',0 ;Радио Шансон 1010

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 14 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.

В некоторых случаях возникает необходимость изменить контрастность дисплея. Настроить контрастность дисплея даже проще, чем записать частоты радиостанций в вашей местности. Для этого в драйвере ЖК-индикатора (**файл *Nokia5110_driver.asm***) нужно найти раздел "Таблица кодов команд инициализации LCD". Там вы увидите строку

.db 0x21, 0xBC, 0x06, 0x13, 0x20, 0x0C

В этой строке замените шестнадцатеричное число *0xBC* на другое, например на *0xC8* и перекомпилируйте проект в *Proteuse* или в другой программе. Возможно придётся подобрать это число, если контрастность будет слишком большой и будут видны тёмные прямоугольники вокруг знаков.

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 команду *Build All*. После этого вы можете, нажимая виртуальные кнопки, понаблюдать на экране как будет выглядеть название станции на дисплее. Если что-то вас не устроит, вы можете оперативно, не прибегая к программированию микроконтроллера, исправить программу.

Если вы работаете в *Proteus*, то в том же в ZIP-архиве, где находятся файлы программы, Вы найдёте файл проекта *Proteus* с расширением *.DSN*. В этом проекте *Proteus* вам ничего менять не нужно. Нужно только в папке на жёстком диске, где у вас лежат все файлы приёмника, отредактировать или подменить мой файл ***User_set.asm*** своим, со своими названиями станций. Затем запустить *Proteus* и подать команду *Build All* из меню *Source*.

Если вы работаете в *AVRStudio*, то компиляцию выполняем в следующем порядке:

1. Создаём отдельную папку, куда копируем все ассемблерные файлы программы. Название папки не должно содержать кириллицы. Итак, в папке лежат файлы:

Driver_DS18B20.asm
Driver_DS1307.asm
Font_6x8a.asm
I2C_hard_driver.asm
Nokia5110_driver.asm
Radio_Clock_Temp.asm
User_set.asm
m8def.inc2.

Запускаем *AVRStudio*, открываем новый проект, в окошко с названием проекта вводим ***Radio_Clock_Temp***, ставим птичку в чекбоксе "*Create initial file*". Указываем путь к нашей папке и жмём кнопку "*Next*". Попадаем в следующее окно, где выбираем платформу *AVR Simulator* и контроллер *ATMega8*, после чего нажимаем кнопку "*Finish*" и попадаем в открытый проект.

3. Подаём команду "Build" и получаем в нашей папке готовый HEX-файл.

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



В верхней строке индикатора выводится уровень принимаемого сигнала, в виде четырёх столбиков разной длины, уровень громкости от 0 до 15, изображение батарейки с условным уровнем заряда и буквами "М" и "S" вид передачи – «моно» или «стерео».

Ниже над разделительной линией выводится частота принимаемой станции, а под линией – температура и время в формате часы и минуты. Значение температуры измеряется и выводится на дисплей каждые 10 секунд. Если установлена перемычка J1, то вместо температуры отображается напряжение батареи питания. В правом верхнем углу отображается пиктограмма элемента питания с условным уровнем заряда. Пороги переключения от одного уровня заряда к другому для разных типов

элементов прописаны в таблице, которая находится в начале файла пользовательских настроек **User_set.asm**. Изначально установлены значения для литий-ионной батареи напряжением 7,4 В, поэтому если в радиоприёмнике будет установлена такая батарея, то ничего менять не надо. Если предполагается питание от батареи из других типов элементов, то значения этих констант нужно взять из соответствующего столбика в комментариях справа и вписать вместо существующих. Возможно после пробной эксплуатации вам захочется подкорректировать эти значения.

Фрагмент файла **User_set.asm**

```

;
; ВНИМАНИЕ! Ввести здесь число местных станций в Вашем регионе
.equ Nst = 15 ;Число местных станций
; Установить значения порогов переключения индикации уровня заряда батареи,
; равные числу десятков милливольт. Пример: для U=7,40 В -> 740
;
; Тип батареи: | Li-Ion | Ni-MH | Alkaline |
.equ Ubat_3 = 740 ;| 740 | 770 | 840 |
.equ Ubat_2 = 720 ;| 720 | 750 | 720 |
.equ Ubat_1 = 680 ;| 680 | 660 | 660 |
.equ Ubat_0 = 640 ;| 640 | 640 | 640 |
;

```

Описание схемы радиоприёмника.

На последней странице представлена полная принципиальная схема радиоприёмника. Радиоприёмник питается от двух литий-ионных элементов с общим напряжением 7,4 В. В схему радиоприёмника введён узел управления питанием, состоящий из ключа на полевом транзисторе VT3 и промежуточного инвертора на биполярном транзисторе VT4. Включение и выключение питания производится кратковременным нажатием на кнопку SB5 «Power». При этом управляющее напряжение для ключа VT3 поступает через кнопку SB5, диод VD2 и резистор R22, после чего ключ VT3 открывается и микроконтроллер получает питание. Микроконтроллер после получения питания своим сигналом логической единицы на выходе порта PB6 открывает транзистор VT4 и, тем самым, поддерживает ключ VT3 в открытом состоянии. Для выключения питания нужно снова кратковременно нажать на кнопку SB5, при этом напряжение на входе порта PB7 через диод VD3 будет близким к логическому нулю. На

это микроконтроллер отреагирует тем, что снимет управляющее напряжение на базе VT4 и, следовательно, ключ VT3 закроется и питание приёмника будет отключено.

Диоды Шоттки VD2 и VD3 служат для развязки цепей включения и выключения питания. При данном напряжении питания (7,4 В) можно применить обычные маломощные диоды, например, 1N4148 и др.

Конденсатор C20 служит для предотвращения самопроизвольного включения питания в результате действия помех.

Непосредственно к напряжению батареи подключен усилитель низкой частоты, и стабилизатор питания +5 В. Из-за разности в напряжениях питания микроконтроллера и радиомодуля пришлось установить стабилизатор на 3,3 В и преобразователи уровня сигнала на транзисторах VT1 и VT2.

Вывод 7 подсветки индикатора соединяется с выводом порта D микроконтроллера напрямую, без промежуточного транзисторного ключа, т. к. ток подсветки по моим измерениям составляет всего 5,2 мА. Для индикатора Nokia5110 на плате синего цвета вывод подсветки нужно соединить с выводом порта PD7, для индикаторов на плате красного цвета этот вывод нужно соединить с выводом порта PD6. На схеме условно показано это соединение через джампер J2.

В целях продления срока службы батареи питания индикатор стереосигнала VD1 и резистор R2 можно не устанавливать.

Чтобы индикатор заряда работал правильно, необходимо как можно более точно измерить напряжение батареи.

Для облегчения настройки измерения в программе предусмотрен вывод на дисплей напряжения батареи в вольтах. Активировать этот вывод можно, если замкнуть вывод порта PD5 на землю, для чего установить перемычку J1. При этом на дисплее вместо температуры будет отображаться напряжение батареи. Для удобства настройки параллельно резистору R12 подключается добавочный резистор R14 большего номинала. При настройке нужно подключить цифровой вольтметр к батарее питания и подбором резистора R14 добиться одинаковых показаний на приборе и на дисплее радиоприёмника.

Напряжение питания батареи измеряется 1 раз в 10 секунд. При снижении напряжения батареи ниже значения, указанного в константе *Ubat_0* (см. файл *User_set.asm*) на дисплее появляется сообщение «Батарея разряжена», после чего питание радиоприёмника отключается.

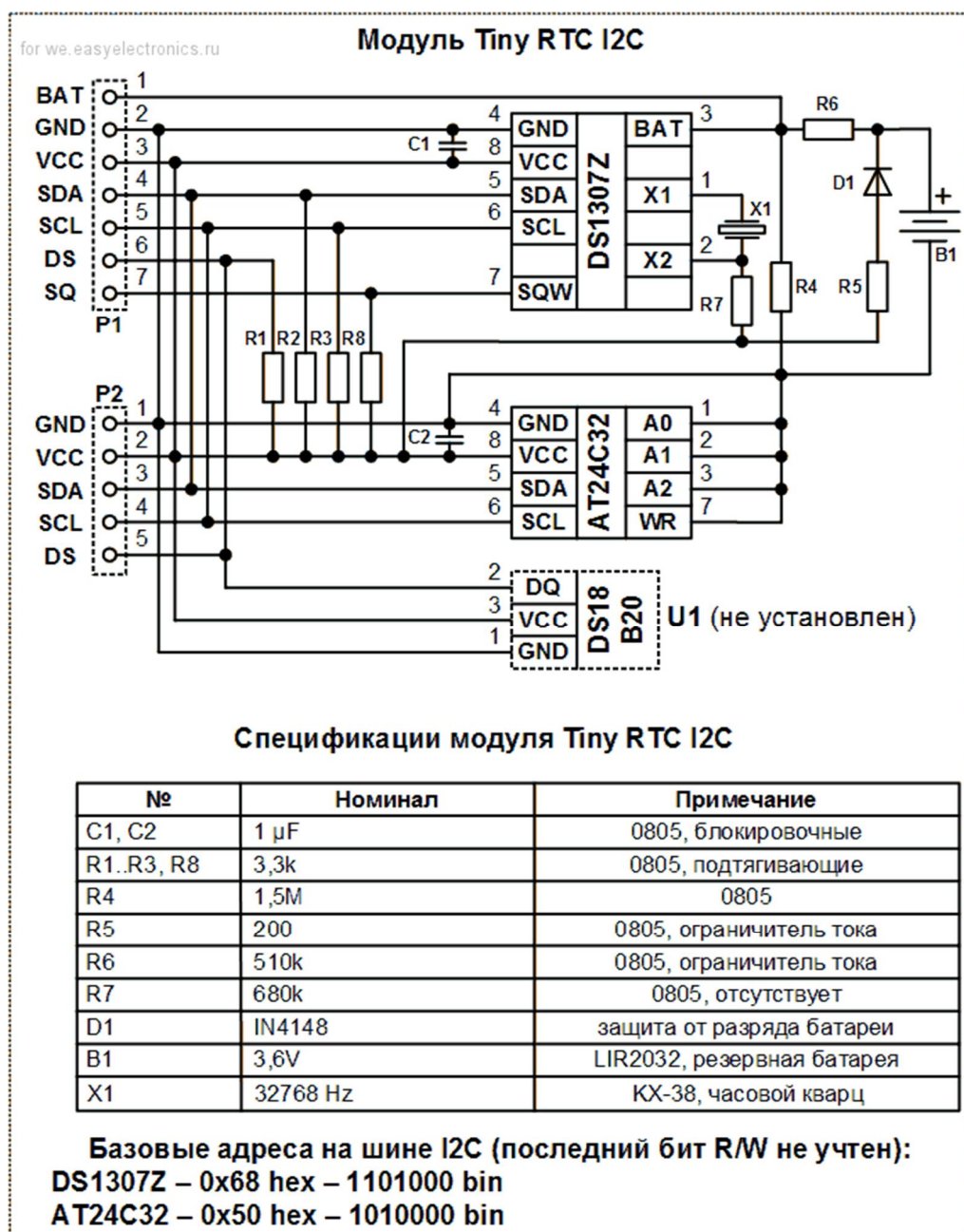


Рис. 1. Схема модуля часов реального времени.

