Главная › I2C › AVR Урок 21. Управление DS1307 кнопками. Часть 2

AVR Урок 21. Управление DS1307 кнопками. Часть 2

Posted on Декабрь 29, 2016 by Narod Stream Опубликовано в I2C, Программирование AVR — 1 комментарий ↓

Урок 21
Часть 2

Управление DS1307 кнопками

Продолжаем работать с кнопками.

В **прошлой части** нашего занятия мы создали и настроили проект, решили вопрос со слишком большой задержкой, добавили макросы и необходимые переменные, так сказать, провели все подготовительные мероприятия.

Давайте сначала посмотрим, как у нас всё подключено к настоящей отладочной плате

Свежие комментарии

- SmNikolay к записи STM Урок 89. LAN. ENC28J60. TCP WEB Server. Подключаем карту SD
- Narod Stream к записи AVR Урок 3. Пишем код на СИ. Зажигаем светодиод
- strannik2039 к записи AVR Урок 3. Пишем код на СИ. Зажигаем светодиод
- Dmitriy к записи AVR Урок 1. Знакомство с семейством AVR
- Narod Stream к записи STM Урок 9. HAL. Шина I2C. Продолжаем работу с DS3231

Форум. Последние ответы

- Narod Stream в Программирование МК STM32
  1 неделя, 2 дн. назад
- Zandy в Программирование МК STM32
  1 неделя, 3 дн. назад
- Narod Stream в Программирование МК STM32
  3 нед. назад
- Narod Stream в Программирование МК STM32
  3 нед. назад
- fireweb в Программирование МК STM32
  3 нед., 3 дн. назад

Январь 2018

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

« Дек

Архивы

- Январь 2018
- Декабрь 2017
- Ноябрь 2017
- Октябрь 2017
- Сентябрь 2017
- Август 2017
- Июль 2017

Мета

- Регистрация
- Войти
- RSS записей
- RSS комментариев
- WordPress.org

искать здесь …
Фильтровать

Заходите на канал Narod Stream

- Июнь 2017
- Май 2017
- Март 2017
- Февраль 2017
- Январь 2017
- Декабрь 2016
- Ноябрь 2016

Всё осталось как и в предыдущем занятии, а также к плате подключена макетная плата с кнопками. Там их целых 5, а задействовано из них только 3.

Теперь начнём писать полезный код.

Сначала мы должны проинициализировать порт, к которому подключены наши кнопки. У нас уже есть функция инициализации портов, туда мы просто добавим код для порта с нашими кнопками

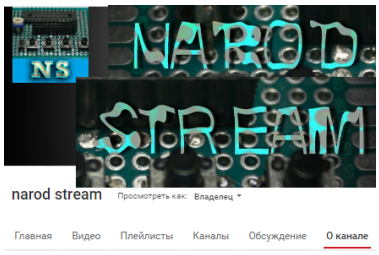
```
void port_ini(void)
{
  PORTD=0x00;
  DDRD=0xFF;
  BUTTONDDR &= ~((1<<BUTTONDDR3)|
(1<<BUTTONDDR2)|(1<<BUTTONDDR1));//
ножки кнопок на вход
  BUTTONPORT |= (1<<BUTTONPORT3)|
(1<<BUTTONPORT2)|(1<<BUTTONPORT1);//
подтянем резисторы к лапкам кнопок
}
```

Ну, код я думаю понятен, благодаря подробным комментариям после каждое его строки.

Мы должны теперь решить, где же именно в бесконечном цикле будем отслеживть состояние кнопок. Я подумал и всё-таки склоняюсь к тому, что это надо делать в том месте, где мы уже считали показания из микросхемы, но ещё их на дисплее не показывали. То есть если мы изменим значение каких-то показаний с помощью кнопки, то они занесутся и в регистр микросхемы и тут же отобразятся и на дисплее, а также нам есть что менять, так как все показания считаны и находятся в переменных.

Вот и давайте проверим, нажата ли у нас кнопка 1 или нет. Для этого потребуется условие. Как отследить нажатие кнопки, именно вторым контактом находящейся на общем проводе, мы уже знаем

```
date = RTC_ConvertFromDec(date); //
Преобразуем в десятичный формат
setpos(0,0); //Ставим курсор на
начало координат
if(!(BUTTONPIN&(1<<BUTTONPIN1))//
Кнопка 1 нажата
   {
   }
```



Рубрики

- 1-WIRE (3)
- ADC (6)
- DAC (4)
- GPIO (26)
- I2C (19)
- SPI (13)
- USART (8)
- Программирование AVR (131)
- Программирование PIC (7)
- Программирование STM32 (213)
- Тесты устройств и аксессуаров (1)

Дальше напишем тело условия, подположив что кнопка данная у нас оказалась нажата. Тогда у нас здесь будет ещё одно условие

```
if(!(BUTTONPIN&(1<<BUTTONPIN1)//
Кнопка 1 нажата
{
  if (clockmode==CLOCKMODE0)
  {
    clockmode=CLOCKMODEDATE;//
перейдем в режим перевода даты
    blinkstate=0;//сбросим счетчик
мигания
    button1state=1;//статус 1 кнопки
  }
}
```

Уже в тело данного условия мы зайдём ещё в том случае, если мы помимо нажатой кнопки ещё находимся в режме нулевом. Ну вернее, не мы, а контроллер, ну, будем говорить мы, так как код пишем именно мы. В данном случае мы переходим в режим перевода даты, сбросим счётчик мигания и статус кнопок установим в единицу, что будет соответствовать именно нажатой первой кнопки.

Дальше мы начнём отображать дату на дисплее в зависимости от режимов. Если мы не находимся не в режиме редактирования даты, то её мы тогда просто отобразим на дисплее

```
    button1state=1;//статус 1 кнопки
  }
}
if(clockmode!=CLOCKMODEDATE)
{
  sendcharlcd(date/10+0x30);//
Преобразуем число в код числа
  sendcharlcd(date%10+0x30);//
Преобразуем число в код числа
}
```

Ну а теперь напишем код, который будет исполняться, если условие не выполнится, то есть в том случае, если в данный момент наоборот как раз и будет режим перевода даты

```
  sendcharlcd(date%10+0x30);//
Преобразуем число в код числа
}
else //если режим перевода даты
{
}
```

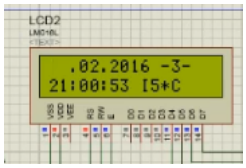
Теперь потихоньку данное тело начнём заполнять кодом. Сначала будет условие

```
else //если режим перевода даты
{
  if (blinkstate==0)
  {
    sendcharlcd(' ');
    sendcharlcd(' ');
    blinkstate=1;
  }
  else
  {
```

```
    sendcharlcd(date/10+0x30);//
Преобразуем число в код числа
    sendcharlcd(date%10+0x30);//
Преобразуем число в код числа
    blinkstate=0;
  }
```

Здесь мы проверяем то, что если у нас статус мигания 0, то мы покажем вместо цифр значения даты пробелы и установим уже статус данный в единицу, чтобы в следующем цикле бесконечного цикла мы уже с другим значением данного статуса попали в else, в котором мы показываем дату, а также обнуляем данный статус. Вот засчёт его постоянного изменения у нас и будет мигать переводимое знакоместо.

Прежде чем писать код дальше, мы попробуем собрать код и поглядеть в протеусе, попадаем ли мы в режимы



При нажатии на соответствующую кнопку у нас начинает мигать дата, значит всё идёт пока в правильном направлении.

Дальше в этом же **else**, в который мы попали в случае режима перевода даты, будем отслеживать опять статус кнопки 1, и если она нажата, будем уже решать вопрос о изменении режима на режим перевода месяца. Нажатие кнопки мы отследим тем же способом, как и отсдеживали на предмет перехода в режим перевода даты. В дальнейшем мы вынесем это отслеживание в отдельную функцию, так как надо ещё и на дребезг исследовать нажатие

```
  blinkstate=0;
}
if(!(BUTTONPIN&(1<<BUTTONPIN1)//
Кнопка 1 нажата
{
  if(button1state==0) //опросим
статус, чтобы сразу не перейти в
режим перевода месяца
  {
    clockmode=CLOCKMODEMONTH; //
перейдем в режим перевода месяца
    button1state=1;
  }
}
```

Таким же образом мы опросим статус кнопки, чтобы он был сброшен, то есть все действия, связанные с нажатием кнопок до этого нажатия, завершены. Статус мы данный в ноль будем устанавливать позже, вернее ниже по коду. Ещё не пришло время. Мы сейчас возможно ещё мигаем датой или переводим её, или ещё что-то делаем.

И вот как раз здесь-то мы это и проделаем, ну конечно при одном

условии, при условии, что мы именно находимся в режиме перевода даты

```
    button1state=1;
  }
}
if (clockmode==CLOCKMODEDATE)
button1state=0;//сбросим статус
```

Дальше должен идти опрос двух кнопок перевода, но мы пока этим заниматься не будем, надо ещё с этой кнопкой закончить.

Теперь пропустим код вывода на экран точки после даты и перед месяцем, точкой мы не мигаем. и дальше обработаем переход в режим перевода месяца аналагично переходу в режим перевода даты. Поэтому скопируем код и кое-что в нем подправим

```
sendcharlcd('.');
if(clockmode!=CLOCKMODEMONTH)
{
  sendcharlcd(month/10+0x30);//
Преобразуем число в код числа
  sendcharlcd(month%10+0x30);//
Преобразуем число в код числа
}
else //если режим перевода месяца
{
  if (blinkstate==0)
  {
    sendcharlcd(' ');
    sendcharlcd(' ');
    blinkstate=1;
  }
  else
  {
    sendcharlcd(month/10+0x30);//
Преобразуем число в код числа
    sendcharlcd(month%10+0x30);//
Преобразуем число в код числа
    blinkstate=0;
  }
  if(!(BUTTONPIN&(1<<BUTTONPIN1)//
Кнопка 1 нажата
  {
    if(button1state==0) //опросим
статус, чтобы сразу не перейти в
режим перевода месяца
    {
      clockmode=CLOCKMODEYEAR; //
перейдем в режим перевода года
      button1state=1;
    }
  }
  if (clockmode==CLOCKMODEMONTH)
button1state=0;//сбросим статус
}
```

Вот так. И, соответственно, пре подобных сложившихся условиях мы уже попадём в режим перевода года.

Также соберём код и проверим в протеусе, перейдём ли мы в режим перевода месяца. Если всё работает, то

продолжаем код дальше. Теперь всё пойдёт намного легче, так как у нас будет только копирование и исправление.

Напишем обработку режима перевода года, также пропустив вывод на экран разделяющей точки

```
sendcharlcd('.');
if(clockmode!=CLOCKMODEYEAR)
{
  sendcharlcd('2');
  sendcharlcd('0');
  sendcharlcd(year/10+0x30);//
Преобразуем число в код числа
  sendcharlcd(year%10+0x30);//
Преобразуем число в код числа
}
else //если режим перевода года
{
  if (blinkstate==0)
  {
    sendcharlcd(' ');
    sendcharlcd(' ');
    sendcharlcd(' ');
    sendcharlcd(' ');
    blinkstate=1;
  }
  else
  {
    sendcharlcd('2');
    sendcharlcd('0');
    sendcharlcd(year/10+0x30);//
Преобразуем число в код числа
    sendcharlcd(year%10+0x30);//
Преобразуем число в код числа
    blinkstate=0;
  }
  if(!(BUTTONPIN&(1<<BUTTONPIN1)//
Кнопка 1 нажата
  {
    if(button1state==0) //опросим
статус, чтобы сразу не перейти в
режим перевода месяца
    {
      clockmode=CLOCKMODEDAY; //
перейдем в режим перевода дня недели
      button1state=1;
    }
  }
  if (clockmode==CLOCKMODEYEAR)
button1state=0;//сбросим статус
}
```

Ну, здесь некоторые изменения ещё связаны с четырехзначностью показаний года, но в этом я думаю ничего сложного нет.

Теперь исследуем на перевод дня недели. Оставим пробел, остальными символами уже мигаем

```
sendcharlcd(' ');
if(clockmode!=CLOCKMODEDAY)
{
  sendcharlcd('-');
  sendcharlcd(day+0x30);//
Преобразуем число в код числа
  sendcharlcd('-');
}
else //если режим перевода года
{
  if (blinkstate==0)
  {
    sendcharlcd(' ');
```

```
    sendcharlcd(' ');
    sendcharlcd(' ');
    blinkstate=1;
  }
  else
  {
    sendcharlcd('-');
    sendcharlcd(day+0x30);//
Преобразуем число в код числа
    sendcharlcd('-');
    blinkstate=0;
  }
  if(!(BUTTONPIN&(1<<BUTTONPIN1))//
Кнопка 1 нажата
  {
    if(button1state==0)
    {
      clockmode=CLOCKMODEHOUR; //
перейдем в режим перевода часов
      button1state=1;
    }
  }
  if (clockmode==CLOCKMODEDAY)
button1state=0;//сбросим статус
}
```

Дальше обрабатываем перевод часов, пропустив позиционирование на вторую строку дисплея

```
setpos(0,1); //Ставим курсор на
начало координат
if(clockmode!=CLOCKMODEHOUR)
{
  sendcharlcd(hour/10+0x30);//
Преобразуем число в код числа
  sendcharlcd(hour%10+0x30);//
Преобразуем число в код числа
}
else //если режим перевода часов
{
  if (blinkstate==0)
  {
    sendcharlcd(' ');
    sendcharlcd(' ');
    blinkstate=1;
  }
  else
  {
    sendcharlcd(hour/10+0x30);//
Преобразуем число в код числа
    sendcharlcd(hour%10+0x30);//
Преобразуем число в код числа
    blinkstate=0;
  }
  if(!(BUTTONPIN&(1<<BUTTONPIN1))//
Кнопка 1 нажата
  {
    if(button1state==0) //опросим
статус, чтобы сразу не перейти в
режим перевода месяца
    {
      clockmode=CLOCKMODEMIN; //
перейдем в режим перевода минут
      button1state=1;
    }
  }
  if (clockmode==CLOCKMODEHOUR)
button1state=0;//сбросим статус
}
```

Обработаем минуты, пропустив двоеточие

```
sendcharlcd(':');
if(clockmode!=CLOCKMODEMIN)
{
  sendcharlcd(min/10+0x30);//
Преобразуем число в код числа
  sendcharlcd(min%10+0x30);//
Преобразуем число в код числа
}
else //если режим перевода минут
{
  if (blinkstate==0)
  {
    sendcharlcd(' ');
    sendcharlcd(' ');
    blinkstate=1;
  }
  else
  {
    sendcharlcd(min/10+0x30);//
Преобразуем число в код числа
    sendcharlcd(min%10+0x30);//
Преобразуем число в код числа
    blinkstate=0;
  }
  if(!(BUTTONPIN&(1<<BUTTONPIN1)//
Кнопка 1 нажата
  {
    if(button1state==0) //опросим
статус, чтобы сразу не перейти в
режим перевода месяца
    {
      clockmode=CLOCKMODESEC; //
перейдем в режим синхронизации
секунд
      button1state=1;
    }
  }
  if (clockmode==CLOCKMODEMIN)
button1state=0;//сбросим статус
}
```

Дальше мы уже не переводим. а синхронизируем или сбрасываем в ноль секунда, переводятся они и так каждую секунду переводятся неплохо. Пропустим двоеточие и напишем код реакции на режим синхронизации секунд

```
sendcharlcd(':');
if(clockmode!=CLOCKMODESEC)
{
  sendcharlcd(sec/10+0x30);//
Преобразуем число в код числа
  sendcharlcd(sec%10+0x30);//
Преобразуем число в код числа
}
else //если режим синхронизации
секунд
{
  if (blinkstate==0)
  {
    sendcharlcd(' ');
    sendcharlcd(' ');
    blinkstate=1;
  }
  else
  {
    sendcharlcd(sec/10+0x30);//
Преобразуем число в код числа
    sendcharlcd(sec%10+0x30);//
Преобразуем число в код числа
    blinkstate=0;
  }
```

```
  if(!(BUTTONPIN&(1<<BUTTONPIN1)//
Кнопка 1 нажата
  {
    if(button1state==0)
    {
      clockmode=CLOCKMODE0; //
перейдем в обычный режим хода
      button1state=1;
    }
  }
  if (clockmode==CLOCKMODESEC)
button1state=0;//сбросим статус
}
```

Особо код ничем от других не отличается синхронизации секунд. Отличается тем, что при дальнейшем нажатии кнопки изменения режимов мы попадаем в обычный режим хода, когда-то мы же должны туда вернуться, вот и настало то время.

Соберём код и проверим работоспособность его в протеусе. В видеоверсии почему-то местами не работало, хотя на живом контроллере всё работает. Но для нас протеус — это не самое главное, а главное чтобы работало на практике. Здесь, в принципе, нет смысла показывать картинку, лучше это всё смотреть именно в видеоверсии.

Пока мы решили вопросы только с переходами в разные режимы, сами режимы начнем обрабатывать в **следующей части** занятия.

Программатор, модуль RTC DS1307 с микросхемой памяти и дисплей можно приобрести здесь:

Программатор (продавец надёжный) **USBASP USBISP 2.0**

Модуль RTC DS1307 с микросхемой памяти

Дисплей LCD 16×2

Смотреть ВИДЕОУРОК
(нажмите на картинку)

‹ AVR Урок 21. Управление DS1307 кнопками. Часть 1

AVR Урок 21. Управление

Один комментарий на “AVR Урок 21. Управление DS1307 кнопками. Часть 2”

Виктор:
Август 5, 2017 в 10:39 дп

DS1307 кнопками. Часть 3 ›

Здравствуйте! Читаю Ваш блок и повторяю уроки, все рассказано доходчиво, то что нужно для новичка! Компилирую примеры в IAR и у меня выдает ошибку в строке
BUTTONDDR &= ~((1<<BUTTONDDR3)|
(1<<BUTTONDDR2)|
(1<<BUTTONDDR1));
Error[Pe020]: identifier "DDRB3" is undefined
Не подскажите как это исправить?

Ответить

Добавить комментарий

Ваш e-mail не будет опубликован.
Обязательные поля помечены *

Комментарий

Имя *

E-mail *

Сайт

× = 54

Отправить комментарий