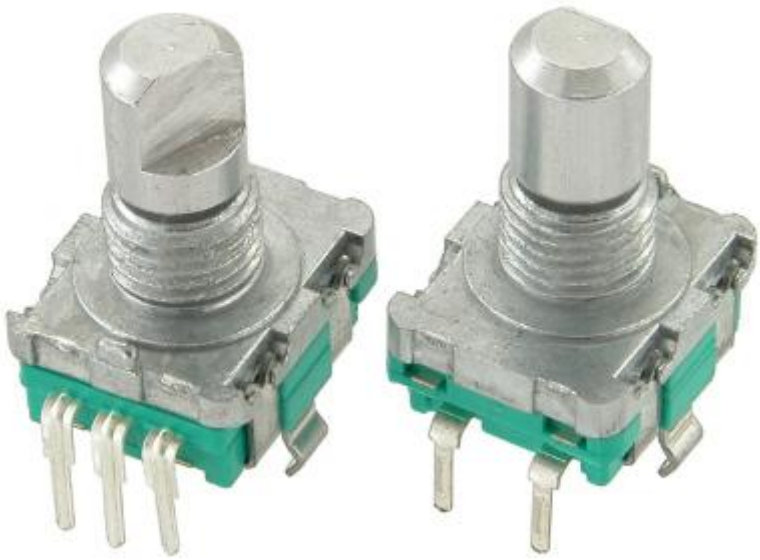
Done at Home
Программирование микроконтроллеров avr

Подключение инкрементальный энкодера к (avr)

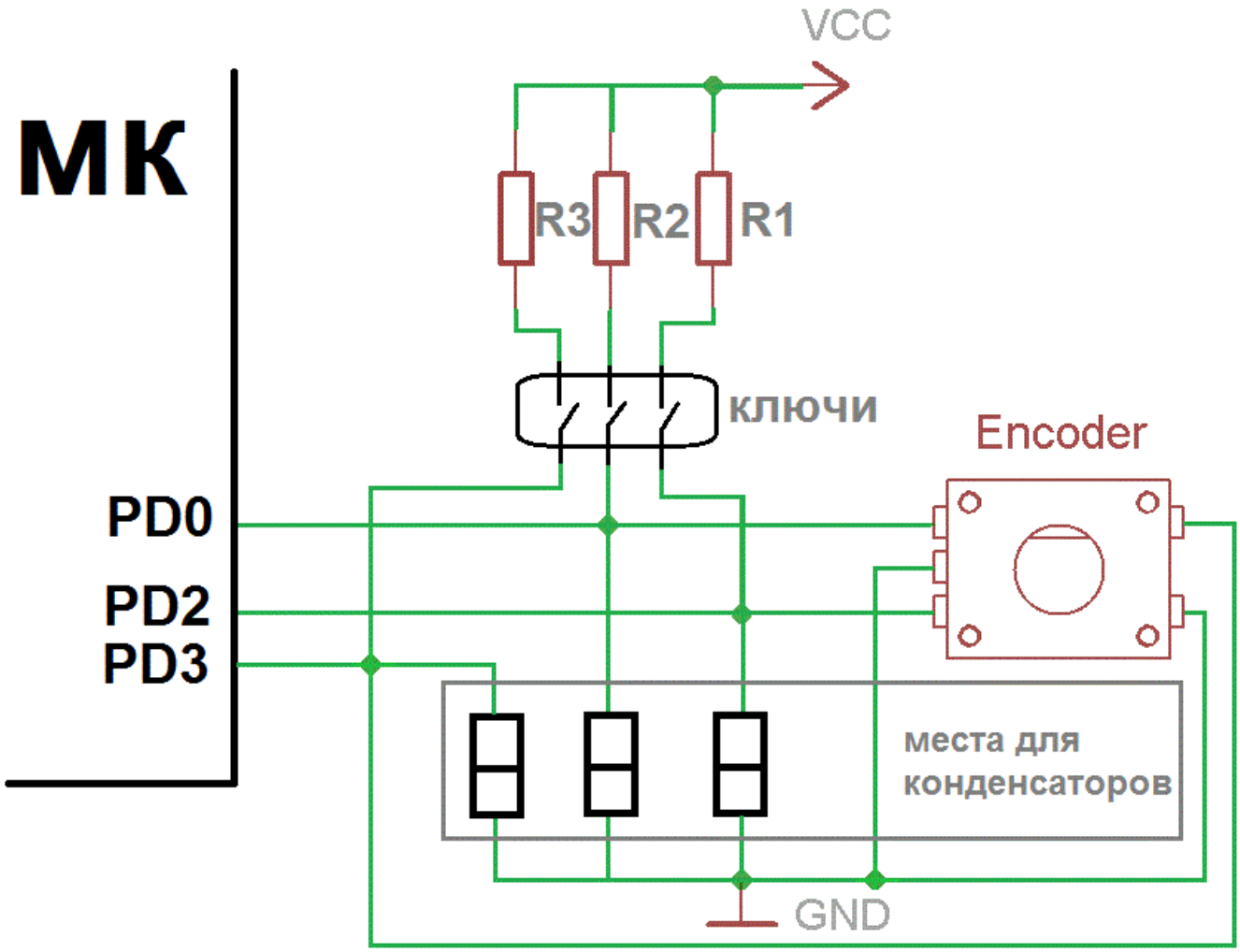
admin | 28.11.2013 0 Comment

Подключение инкрементальный энкодера к (avr)

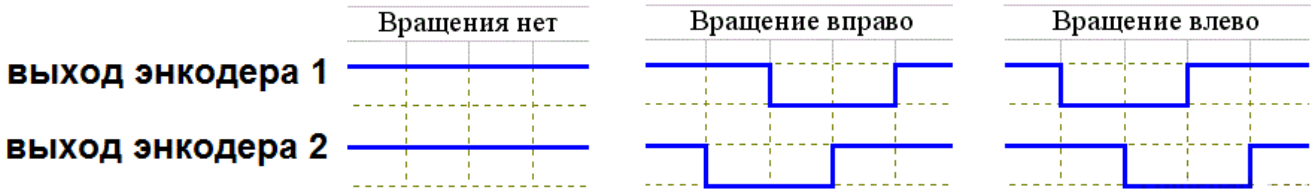
Энкодер – это цифровой датчик угла поворота (ссылка). Внешний вид нашего энкодера весьма похож на потенциометр, но если мы позмем его в руки и покрутим, то окажется что у него нет границ, тоесть его можно бесконечно крутить в одну либо другую сторону. Это устройство прекрасно может заменить несколько кнопок которые в ваших проектах необходимо часто нажимать. Энкодеры часто ставят для управления громкостью .



У нашего энкодера 5 выводов. 2 вывода на выводы кнопки, 3 вывода выводы энкодера. Из них – 2 сигнальных и 1 общий. Он посередине. Схема подключения энкодера ничем не отличается от подключения обычных кнопок. Сигнальные выводы энкодера подключаем к любому порту ввода вывода микроконтроллера. Общий вывод энкодера сажаем на землю. Для защиты от дребезга контактов не лишним будет добавить конденсаторы.



Когда ручка энкодера стоит неподвижно – на входах микроконтроллера присутствуют логические единицы. Когда ручку энкодера поворачивают, на выводах микроконтроллера появляются два прямоугольных сигнала сдвинутых друг относительно друга. От направления вращения вала энкодера зависит, какой из сигналов будет опережать другой. На рисунке ниже представлены возможные варианты сигналов для идеального случая.



Внутри энкодера имеются контакты, которые при вращении то замыкаются, то размыкаются. Этот процесс может сопровождаться дребезгом, поэтому реальные сигналы отличны от показанных выше.

Есть два популярных метода обработки энкодера: по прерыванию (от INT0,INT1,PCINT…) и сравнение предыдущего состояния энкодера и текущего. В этой статье рассмотрим более простой способ с использованием прерываний.

Суть нашего метода опроса весьма проста: мы подключаем один вывод нашего энкодера к ножке, на которой настраиваем прерывание по фронту сигнала. второй выход энкодера к какой либо ножке для опроса в момент возникновения условия прерывания. При опросе мы можем определить “0” на ножке либо “1”, что и послужет нам для определения в какую сторону был повернут энкодер.

Ссылки на комплектующие:

Микроконтроллер: ATmega32 (ссылка)

Энкодер: (ссылка)

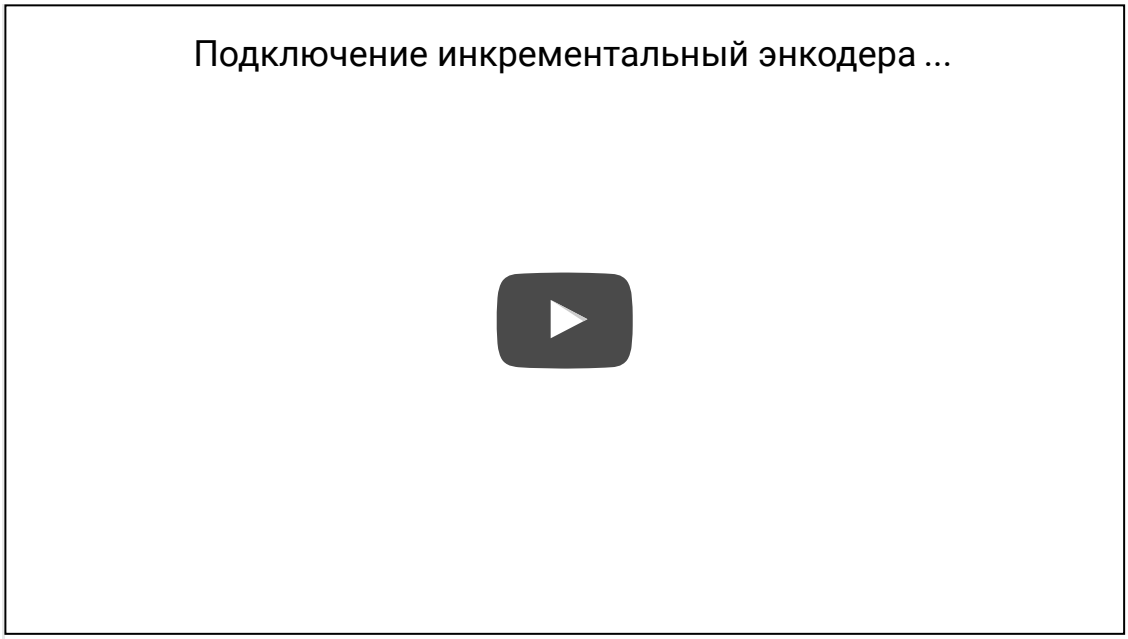
Ключи: (ссылка)

Макетная плата: (ссылка)

Конденсаторы: Рекомендую покупать наборы разных номиналов (ссылка)

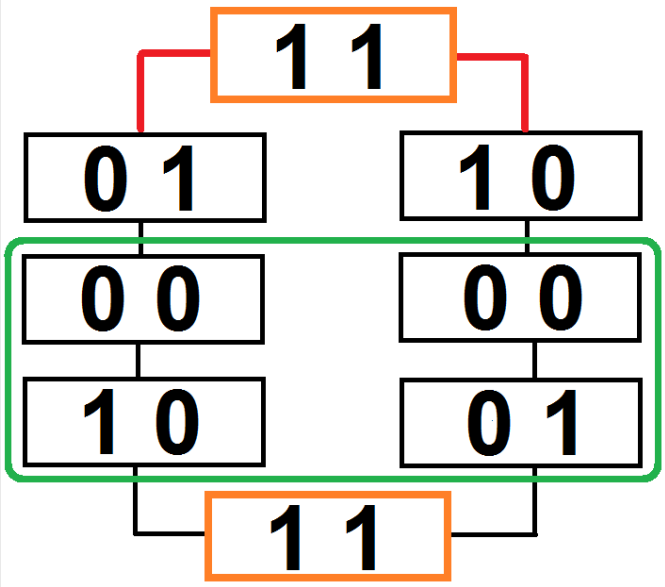
Резисторы: Рекомендую покупать наборы разных номиналов (ссылка)

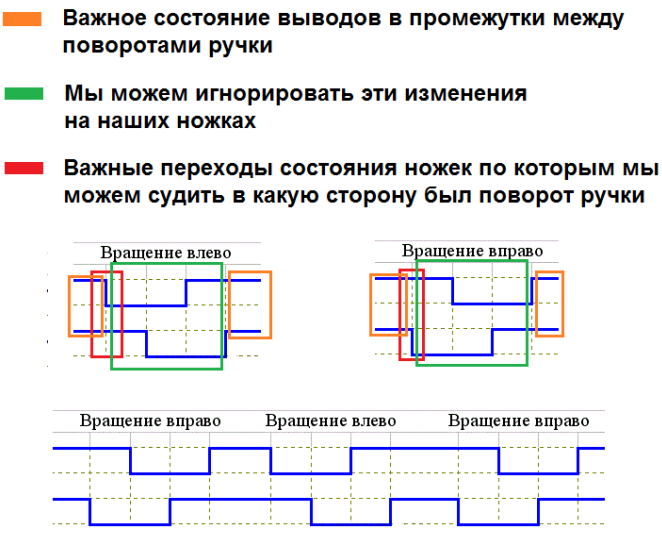
```
#define F_CPU 1000000UL //определяем тактовую частоту как 1Мгц
#include <avr/io.h>
#include <avr/interrupt.h>//заголовочный файл для работы с прерываниями

void port_init () //функция инициализации портов
{
DDRA=0xFF; PORTA=0xFF; //порт на выход светодиоды
DDRB=0xFF; PORTB=0xFF; //порт на выход светодиоды
DDRD=0x00; PORTD=0x00; //порт на вход 3-сост
}

void init_interp(void)
{
MCUCR |=(1<<ISC01)|(1<<ISC11); //регистр настройки прерываний на выводах INT0 и INT1
GICR |= (1<<INT0)|(1<<INT1); //разрешение прерывания на INT0
}

char i=0; //переход на следующий режим при ажатии кнопки
ISR (INT0_vect) //переход на следующий режим при ажатии кнопки
{
if(!(PIND&1))i-=1;
else i+=1;
PORTB=i;
}

ISR(INT1_vect)//переход на следующий режим при ажатии кнопки
{ PORTA=i; }

int main(void)
{
port_init(); //инициализация портов
init_interp(); //инициализация внешних прерываний
sei(); //глобальное разрешение прерываний
while(1);
}
```



Подключение инкрементальный энкодера к (avr)(2-ой метод)

Суть метода заключается в том, чтобы постоянно опрашивать ножки МК подключенные к энкодеру. Можно выделить несколько важных значений и в моменты перехода от одного состояния ножек в другое можно понять в какую сторону была повернута ручка энкодера.





КОД:

```
#define F_CPU 1000000UL //определяем тактовую частоту как 1Мгц
#include <avr/io.h>
#include <avr/interrupt.h>//заголовочный файл для работы с прерываниями

void port_init () //функция инициализации портов
{
DDRA=0xFF; PORTA=0xFF; //порт на выход светодиоды
```

```
DDRC=0×00; PORTC=0×00; //порт на вход 3-сост
}

void timer_init (void)
{
TCCR0=(1<<WGM01); // устанавливаем режим СТС (сброс по совпадению)
TIMSK |= (1<<OCIE0); // устанавливаем бит разрешения прерывания  счетчика по совпадению
OCR0 = 0b11111111; // определяем число сравнения
TCCR0|=(1<<CS01); //запуск таймера с предделителем
}

char i=0;  //глобальн переменная управляемая энкодером
#define ENCOD 0b00000011 //макрос в помощ

void Encoder (void)
{
static char E=ENCOD;
char buf=PINC&ENCOD;//спрашиваем и сохраняем сост ножек

if (E==ENCOD&&buf!=E) //если предыдущь сост единицы + сост ножек не равно предыдущему состоянию
{
if(buf==0b00000001)//если новое сост 01
{i+=1; E=0; } //увеличиваем переменную на 1 + преходим в режим ожидания единиц

if(buf==0b00000010)//если новое сост 10
{i-=1; E=0; } //уменьшаем переменную на 1 + преходим в режим ожидания единиц
}
if (buf==ENCOD) E=ENCOD;//если сейчас на ножках единицы то запомнить это
}

ISR (TIMER0_COMP_vect)
{ Encoder(); }

int main(void)
{
port_init(); //инициализация портов
timer_init(); //инициализация таймера
sei();

while(1)
PORTA=i;
}
```

Рубрика: Все посты AVR Подключаем к AVR

Сайт