

МП и МПС

Лекция 3.

Подключение кнопок. Порты ввода-вывода

Потенциал - к земле.

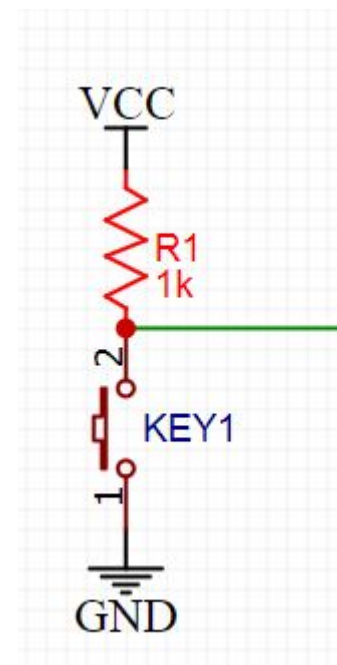
Подтягивающий резистор - для удержания высокого потенциала, пока кнопка не нажата

Подтягивающий резистор можно активировать записью соответствующего $PORTx.b$ при $DDRx.b = 0$

Считывать значение кнопки можно из регистра $PINx$

Корректная обработка нажатий - задача программиста

Определение фронта срабатывания, одиночное срабатывание, борьба с дребезгом, длительное нажатие, комбинация нажатий.



Матрица кнопок

Матрица состоит из строк и столбцов

Нажатие (i,j)-кнопки замыкает
i-строку с j-столбцом

Все контакты подключены к портам ввода-вывода
Строки - формируются, столбцы - сканируются
(или наоборот)



```
| ph0 | ph1 | ph2 | ph3
| 0111 | 1011 | 1101 | 1110
in1| ----1|----1|----1|----1
in2| -x--1|-x--0|-x--1|-x--1
in3| ----1|----1|----1|----1
in4| ----1|----1|----1|----1
```

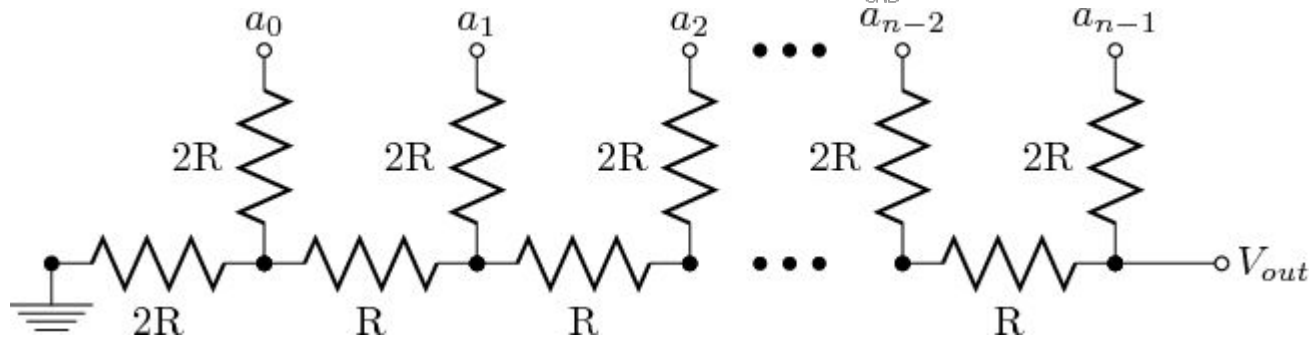
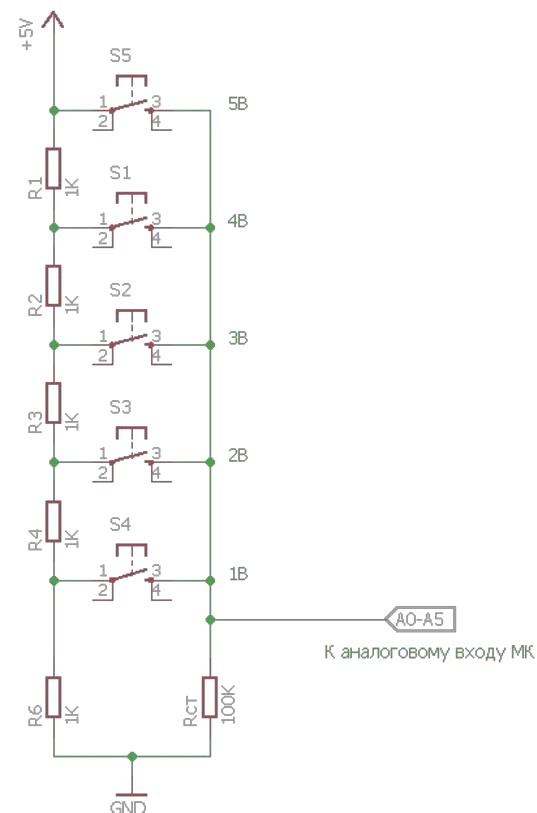
Варианты (сканирующая единица; перевод в pull-up, или выставление 1)

Работа с кнопками через АЦП

Несколько вариантов схем, главная идея - кнопками формируется то, или иное соотношение плеч в делителе напряжения.

Или - формируется цифровой код, поступающий на вход R2R-ЦАП.

другие варианты подключения кнопок



Работа с кнопками через прерывания

MCUCR – MCU Control Register - для INT0 и INT1

MCUCSR – MCU Control and Status Register - для INT2

В статусном регистре (SREG) разрешение на глобальные прерывания

GICR - General Interrupt Control Register - разрешения на прерывания

GIFR - General Interrupt Flag Register - флаги (очищаются автоматически)

PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
-----	--

PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)

Table 35. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Дребезг контактов

Кратковременные замыкания и размыкания электрической цепи, приводящие к многократному срабатыванию кнопки.

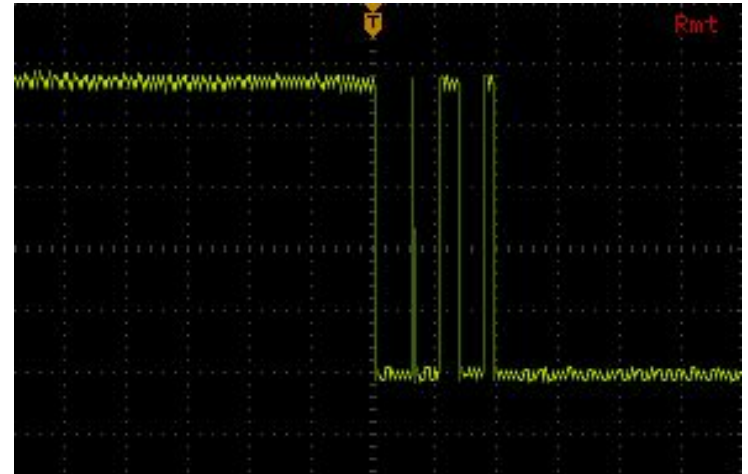
Возникают, как правило, в момент нажатия или отжатия.

Борьба с дребезгом: аппаратные способы

- фильтр низких частот + триггер Шмитта

Борьба с дребезгом: программные способы

- Задержка, повторная проверка сигнала
- Несколько срабатываний подряд
- Цифровое интегрирование



7-сегментный индикатор

Устройство для отображения цифровой информации

Общий катод / Общий анод

Динамическая индикация - переключение между разрядами

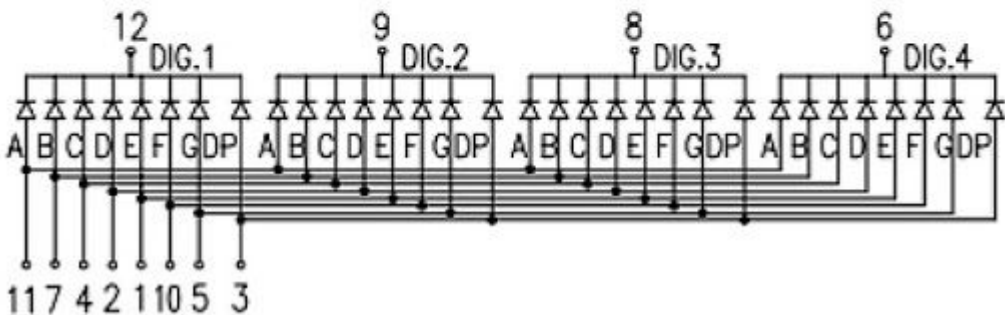
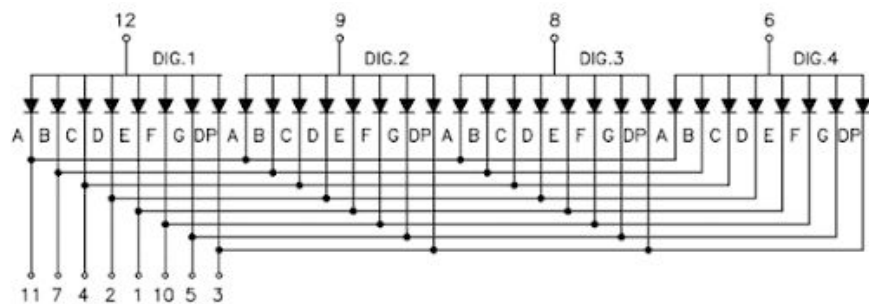
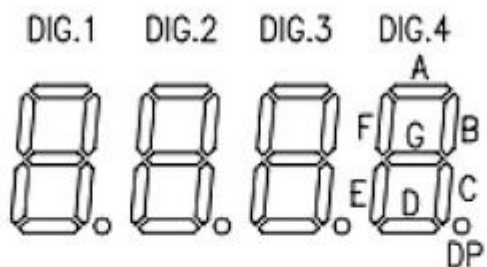
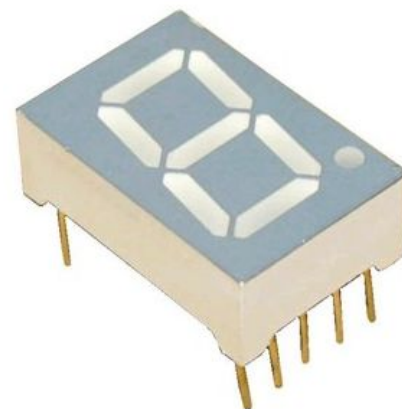
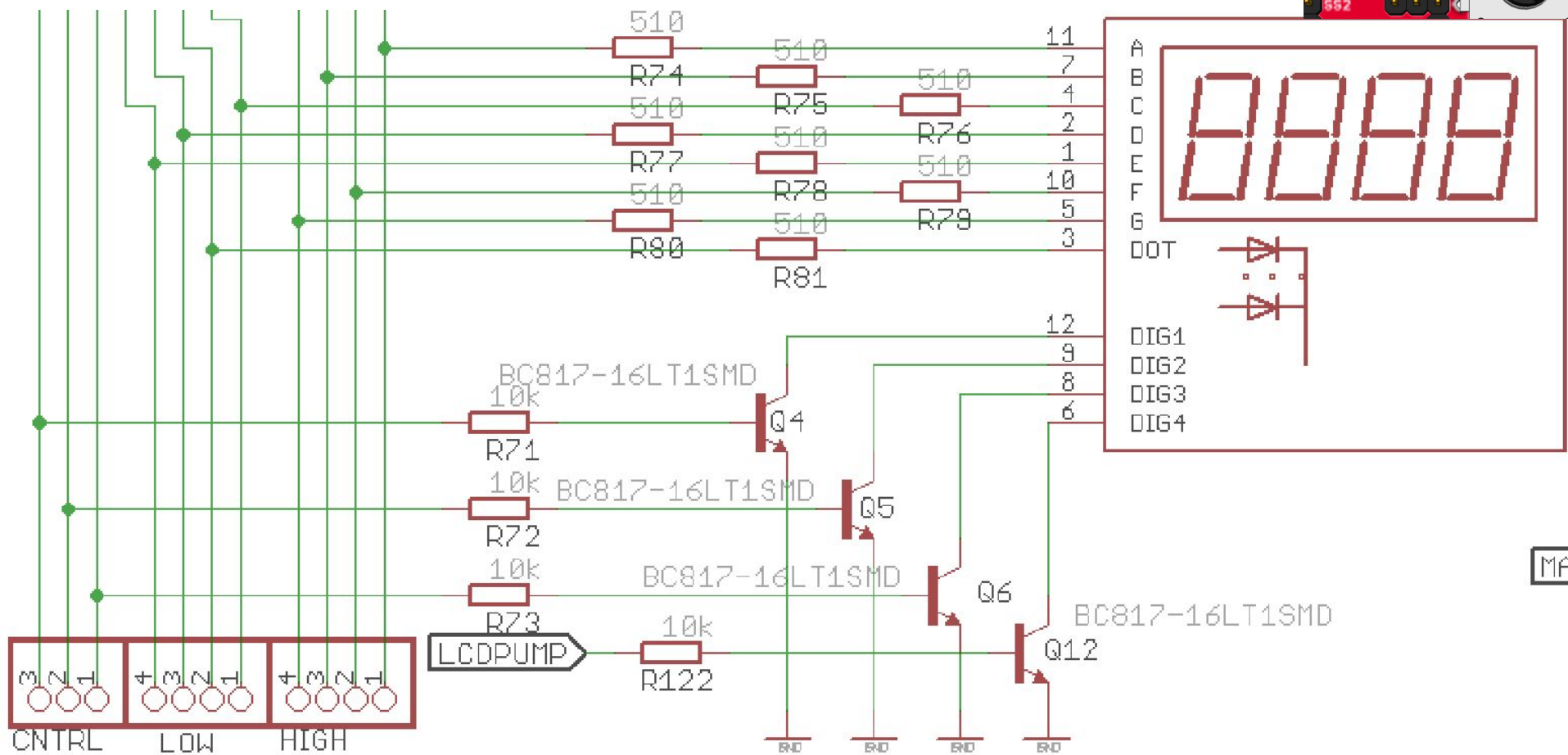


Схема подключения на плате pinBoard

Для сегментов - один порт,
для разрядов - ещё 4 бита с другого порта.



Задания

Задание 2.

Подключение одной кнопки. Переключение/изменение внутреннего состояния, программная борьба сдребезгом.

Подключение матрицы кнопок. Какое-нибудь отображение нажатой кнопки (на светодиодах, или на 7-сегментный индикатор)

Задание 3.

Статическая индикация. Без переключения между разрядами.
Дополнительно: изменяющаяся во времени.

Динамическая индикация. На разных разрядах одновременно разные числа.
Дополнительно: изменение во времени; взаимодействие с другим оборудованием (кнопки, АЦП).

Код ассемблера - полезные команды

Директива .def - имена регистров

```
.def tmp = r16  
.def counter = r20
```

Директива .macro - макросы

```
.macro outi  
    ldi tmp, @1  
    out @0, tmp  
.endm
```

```
outi SPL, Low(RAMEND)
```

Директива .db - массивы данных

```
data: .db 0x60, 0xda, 0xF2, 0x66
```

Для чтения с программной памяти:

```
ldi ZH, High(data*2)  
ldi ZL, Low(data*2)  
lpm
```

Команда sbic/sbis - skip, if set/clear

```
sbic PIND, 0  
jmp button_label
```

Команда clr - очистка регистра

```
clr r17  
clr tmp
```

Очистка оперативной памяти

```
LDI    ZL,Low(SRAM_START)  
LDI    ZH,High(SRAM_START)  
CLR    R16  
lp:    ST     Z+,R16  
CPI    ZH,High(RAMEND+1)  
BRNE   lp
```

```
CPI    ZL,Low(RAMEND+1)  
BRNE   lp
```

```
CLR    ZL  
CLR    ZH
```