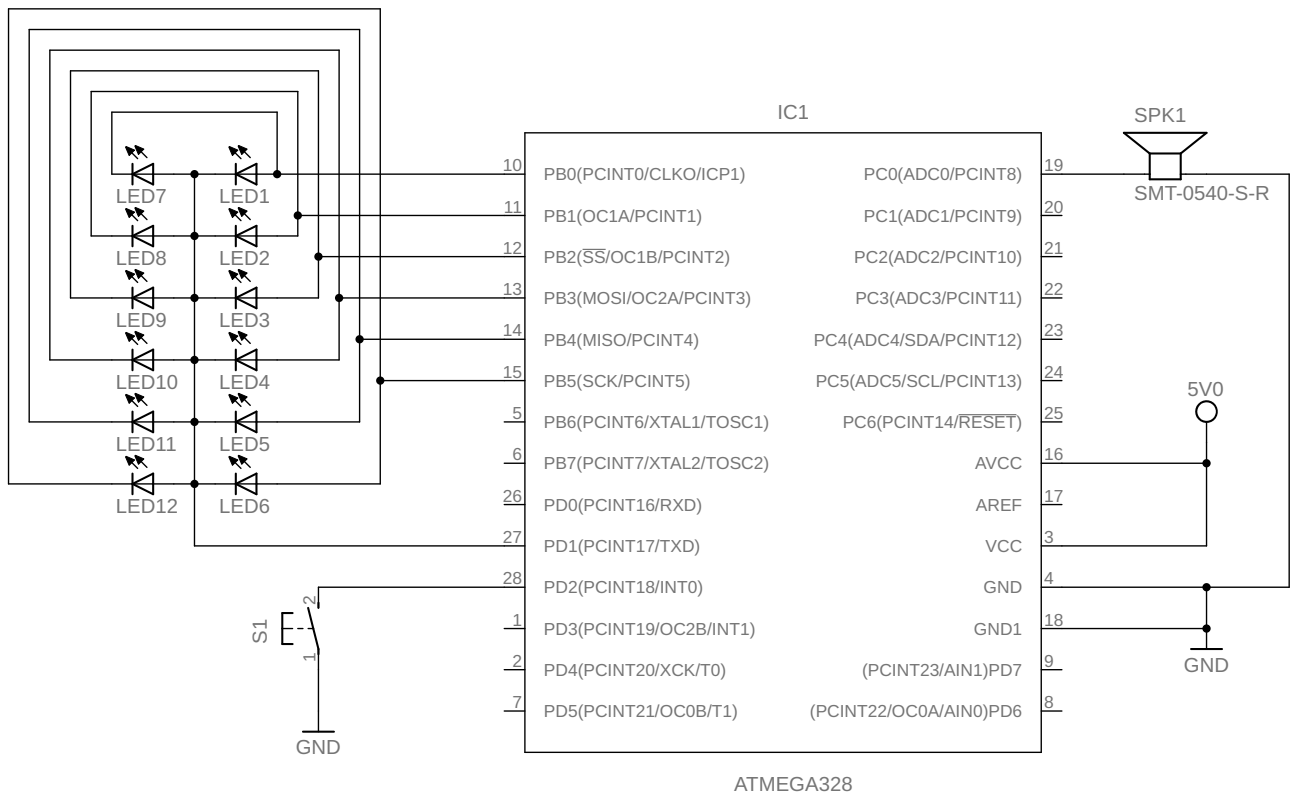


Für die folgenden Aufgaben gilt das folgende Schaltbild. Es werden dabei nur die für die Aufgaben relevanten Teile der Schaltung abgebildet.

Diese Schaltung bildet eine Binäruhr ab. Die LEDs 1 bis 6 dienen der Anzeige der Sekunden bzw. Minuten und die LEDs 7 bis 11 zeigen die Stunden an. Die Taster dienen zur Einstellung der Uhr.



## 1. Portprogrammierung (5)

Implementieren Sie die Initialisierung der Ports so, dass alle Taster, Lautsprecher und LEDs störungsfrei nutzbar sind! Die Konfiguration der für diesen Zweck nicht benötigten Pins darf hierbei nicht verändert werden! Vervollständigen Sie zudem die Funktion **schreibeSekunden**!

```
1
2 void initPorts()
3 {
4     DDRB|=0b00111111;
5     DDRD|=0b00000010;
6     DDRC|=0b00000001;
7     PORTD|=(1<<PD2);
8 }
9
10 void schreibeSekunden(uint8_t out)
11 {
12
13 PORTD &= ~(1<<PD1);
14
15 PORTB= (PORTB&0b11000000)|(out&0b00111111);
16
17 }
```

## 2. Timer-Konfiguration (10)

Implementieren Sie die Initialisierung des Timer 1 im CTC-Modus für eine Zeitbasis von 1 Sekunde! Die Taktfrequenz des Microcontrollers beträgt 2 MHz. Vervollständigen Sie zudem die für den Betrieb des Timers notwendige ISR! (Üben Sie auch mit einem 8-Bit-Timer (Timer0), denken Sie daran dass es CTC und Normal (Overflow) Modi gibt. )

```
1 volatile int sekunde;
2 void initTimer1()
3 {
4     TCCR1B|=(1<<WGM12)|(1<<CS10)|(1<<CS11);
5     TIMSK1|=(1<<OCIE1A);
6     OCR1A=31250-1;
7     sei();
8 }
9
10
11 ISR(TIMER1_COMPA_vect){
12
13     sekunde++;
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34 }
```

### 3. Externer Interrupt (5)

Implementieren Sie die externen Interrupts so, dass bei fallender Flanke am Taster S1 der Lautsprecher eingeschaltet wird und bei erneutem Tastendruck wieder ausgeschaltet wird!

```
1 volatile int sekunde;
2 void initTasterS1()
3 {
4
5     EICRA|=(1<<ISC01);
6     EIMSK|=(1<<INT0);
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 }
26
27 ISR(INT0_VECT){
28
29     PORTC^=0b00000001;
30
31
32
33
34 }
```