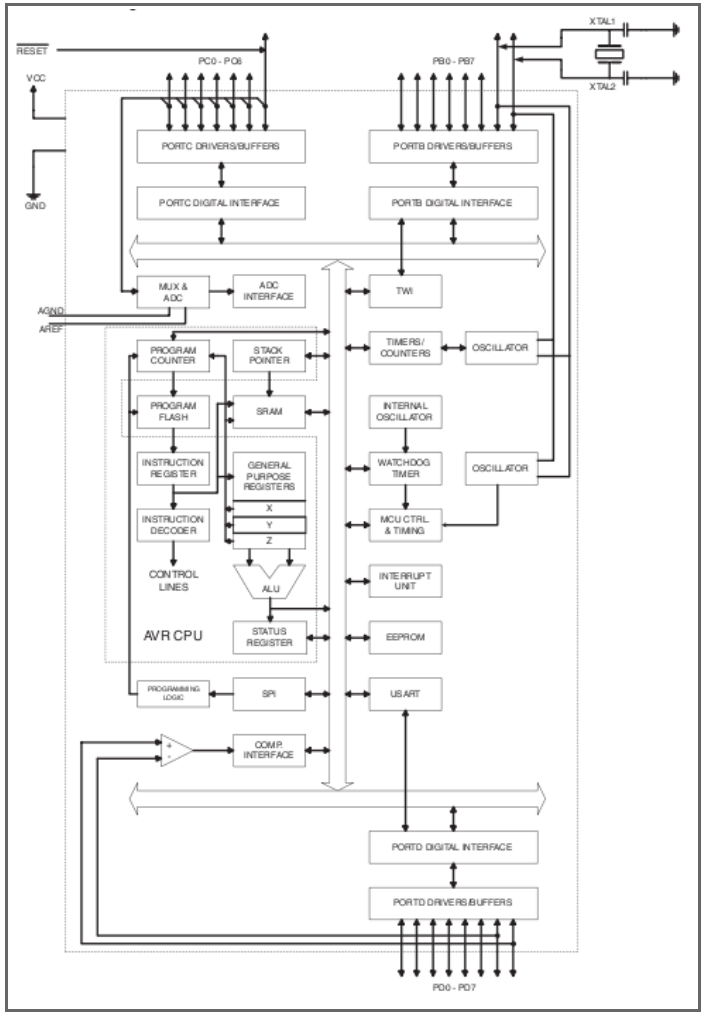
```
#include <avr/io.h>
void main (void) {
  DDRB = 0b00000001; //1==Ausgang
  PORTB= 0b00000000; //1== HIGH, 0 == LOW
  while(1); //endlosschleife
}
```



AVR-Mikrocontroller

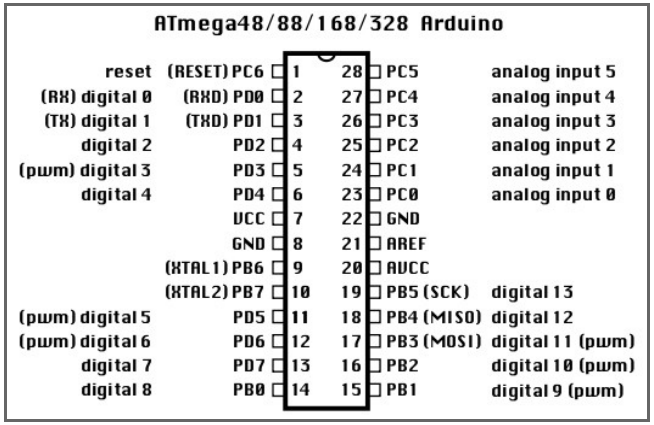
Thaddäus Krönert
kroenert@cs.tu-berlin.de

PROJEKT LABOR

```
#include <avr/io.h>
void main (void) {
  DDRB = 0b00000000; //Eingang ; 1==Ausgang
  PORTB= _BV(PB1); //Pullup ; 0==ohne Pullup

  int  i; // int ist 8-bit gross auf einem Atmega

  do {
    i = PINB; //Lesen des gesammten PORTs B
  while(!(i & (_BV(PB1)))); //<- Alternative!
  while(1); //endlosschleife
}
```




```
#include <avr/interrupt.h>

//Timer0 overflow vector-ISR:
ISR (TIMER0_OVF_vect)

{
  // insert interrupt-handler here
}

int main (void)
{
  // io-clk  /  1024:
  TCCR0 |= ( _BV(CS00) | _BV(CS02) );
  // Interrupt bei overflow aktivieren:
  TIMSK |= _BV(TOIE0);
   sei();
   while (1);
}
```

Software:
- AVR-Studio (Windows, GUI)
- KontrollerLAB (Linux, GUI)
- avr-gcc (compiler, commandline)
- Arduino (Komplette IDE fuer Arduinos)

Programmer:
- USBasp (~5Euro zum selberloeten)
- Dragon (~60Euro , aber mit debugWire)

Evaluationboards:
- Steckbrett (Breadini)
- STK500 (teuer, aber von Atmel)
- stk500 aehnliche Nachbauten (noname)
- Arduino (~20Euro, viele Beispiele)