```
// C-File ------------------------------------------

#define F_CPU 10000000UL
#include <avr/io.h>
#include <util/delay.h>

void delayms(uint16_t millis) {
  uint16_t loop;
  while ( millis ) {
    _delay_ms(1);
    millis--;
  }
}

int main(void) {
  DDRB |= 1<<PB0; /* set PB0 to output */
  while(1) {
    PORTB &= ~(1<<PB0); /* LED on */
    delayms(100);
    PORTB |= 1<<PB0; /* LED off */
    delayms(900);
  }
  return 0;
}

// Makefile ----------------------------------------

CC=avr-gcc
CFLAGS=-g -Os -Wall -mcall-prologues -mmcu=atmega8
OBJ2HEX=avr-objcopy
UISP=avrdude
TARGET=blink

program: $(TARGET).hex
    $(UISP) -c usbasp -p m8 -U flash:w:$(TARGET).hex:i

%.obj: %.o
    $(CC) $(CFLAGS) $< -o $@

%.hex: %.obj
    $(OBJ2HEX) -R .eeprom -O ihex $< $@

clean:
    rm -f *.hex *.obj *.o

// dmesg ----------------------------------------

[ 8380.296084] usb 3-3.1: USB disconnect, device number 28
[ 8382.865016] usb 3-3.1: new low-speed USB device number 29 using ohci_hcd
[ 8382.975021] usb 3-3.1: New USB device found, idVendor=16c0, idProduct=05dc
[ 8382.975035] usb 3-3.1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 8382.975042] usb 3-3.1: Product: USBasp
[ 8382.975048] usb 3-3.1: Manufacturer: www.fischl.de

// hexfile ----------------------------------------

:1000000012C019C018C017C016C015C014C013C044
:1000100012C011C010C00FC00EC00DC00CC00BC06C
:100020000AC009C008C011241FBECFE5D4E0DEBF5E
:10003000CDBF0DD016C0E4CF07C0E3ECF9E0319797
```

```
:10004000F1F700C0000001970097B9F70895B89A3A
:10005000C09884E690E0F0DFC09A84E893E0ECDF9B
:06006000F7CFF894FFCF7A
:00000001FF

// make aufrufen -----------------------------------

root@hpfun:/home/unknown/Downloads/atmel/projekt1# make
avrdude  -c usbasp -p m8 -U flash:w:blink.hex:i

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.01s

avrdude: Device signature = 0x1e9307
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be perfo
rmed
         To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "blink.hex"
avrdude: writing flash (102 bytes):

Writing | ################################################## | 100% 0.06s

avrdude: 102 bytes of flash written
avrdude: verifying flash memory against blink.hex:
avrdude: load data flash data from input file blink.hex:
avrdude: input file blink.hex contains 102 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.03s

avrdude: verifying ...
avrdude: 102 bytes of flash verified

avrdude: safemode: Fuses OK

avrdude done.  Thank you.

```