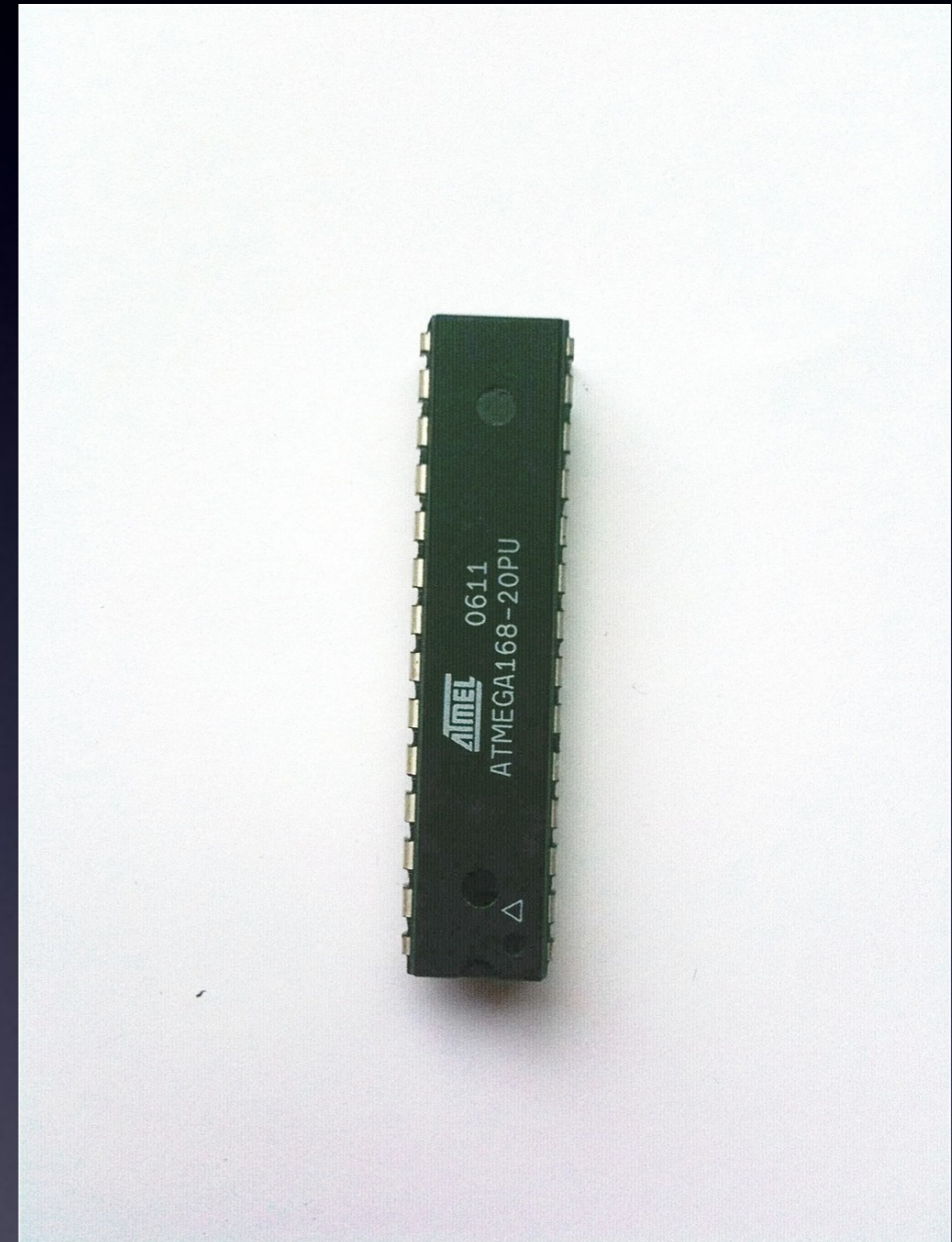


# Mikrocontroller Workshop

Fachschaft Informationssystemtechnik

# Mikrocontroller

- Winziger Computer in einem Chip
- Billig und Strom sparend
- Enthält CPU, RAM, Flash
- Millionenfach in Geräten verbaut



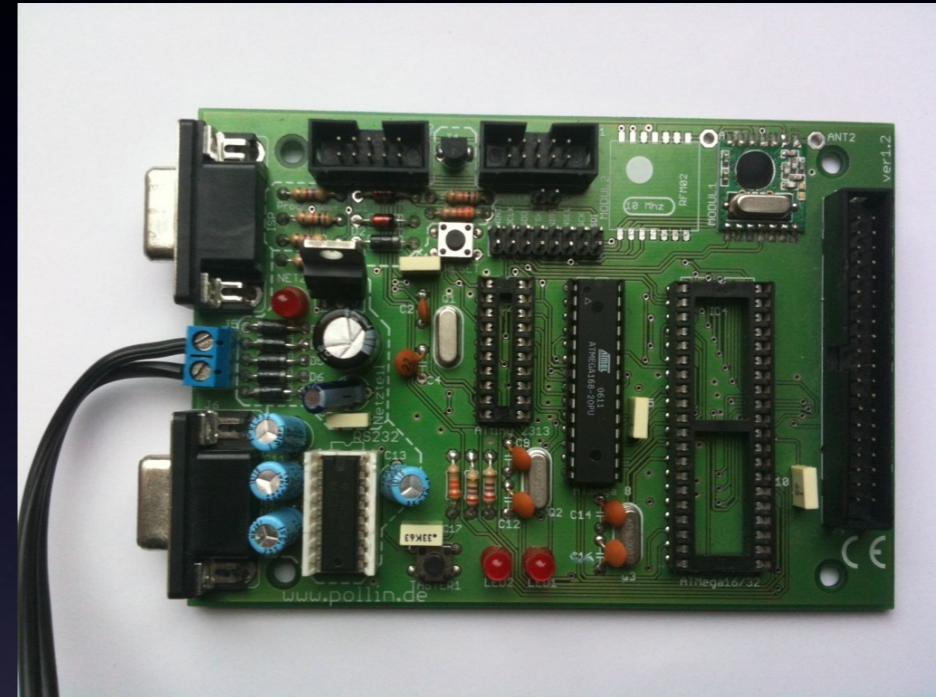
# ATMEGA 168

- Bis zu 20 MHz
- 16 KB  
Programmspeicher
- 1 KB RAM
- 23 umschaltbare Ein-  
und Ausgänge
- Sonderfunktionen auf  
manchen Anschlüssen

(PCINT14/RESET) PC6	1	28	PC8 (ADCS/SCL/PCINT13)
(PCINT16/TXD) PD0	2	27	PC4 (ADCA/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC9 (ADCS/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADCS/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADCO/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/INT0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/CP1) PB0	14	15	PB1 (OC1A/PCINT1)

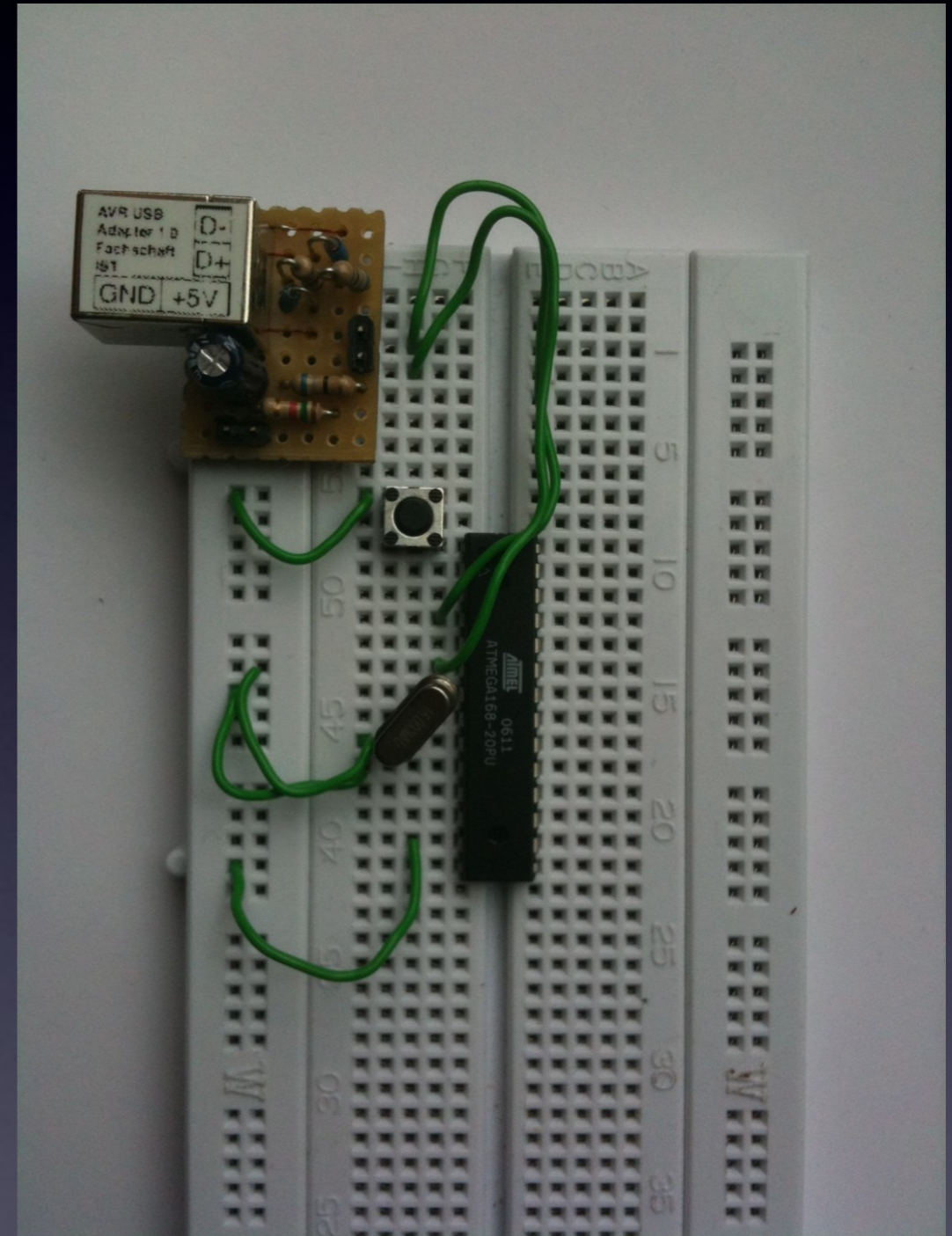
# Wie machen wir das?

- Normalerweise:
  - Platine
  - Programmiergerät
- Relativ teuer für große Gruppe!

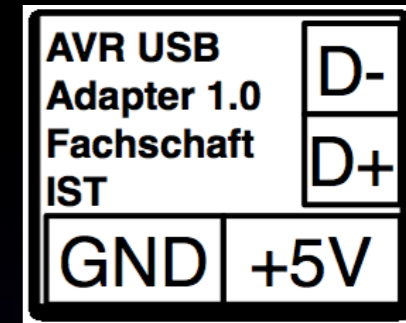


# Low Cost Variante

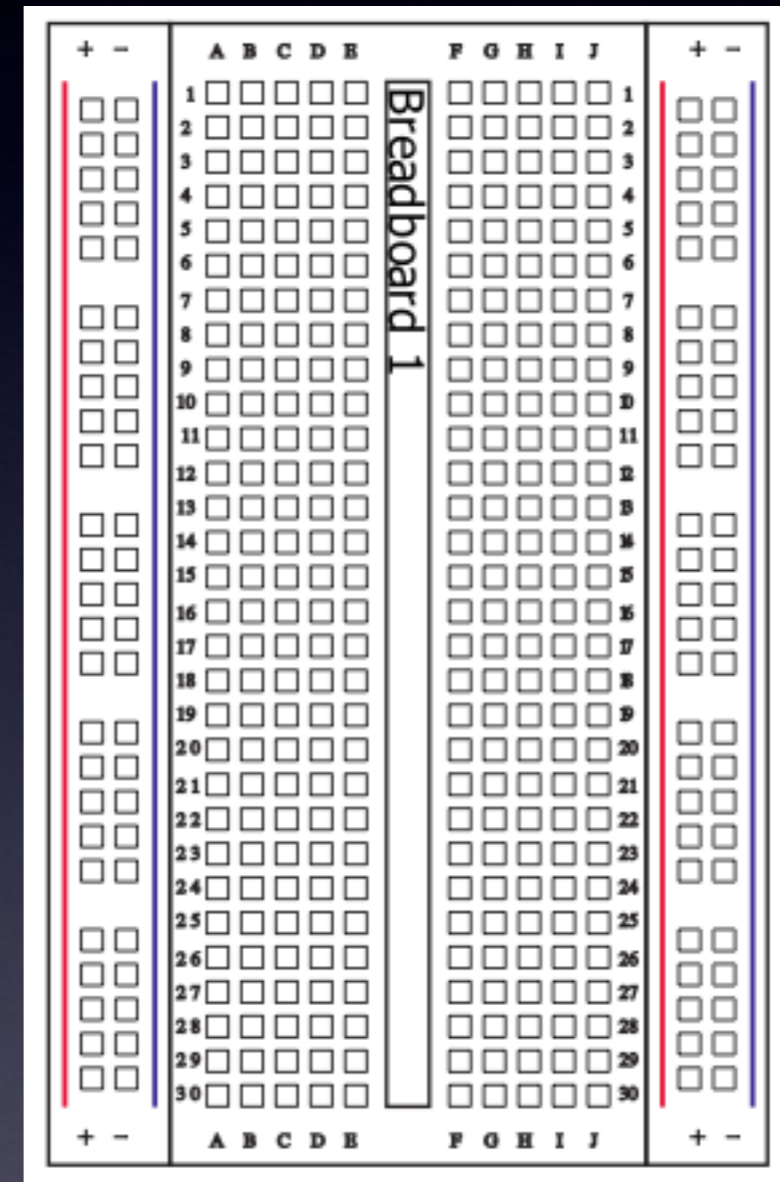
- Steckbrett/Breadboard
- Strom über USB
- USB Programmiermodus (Bootloader)

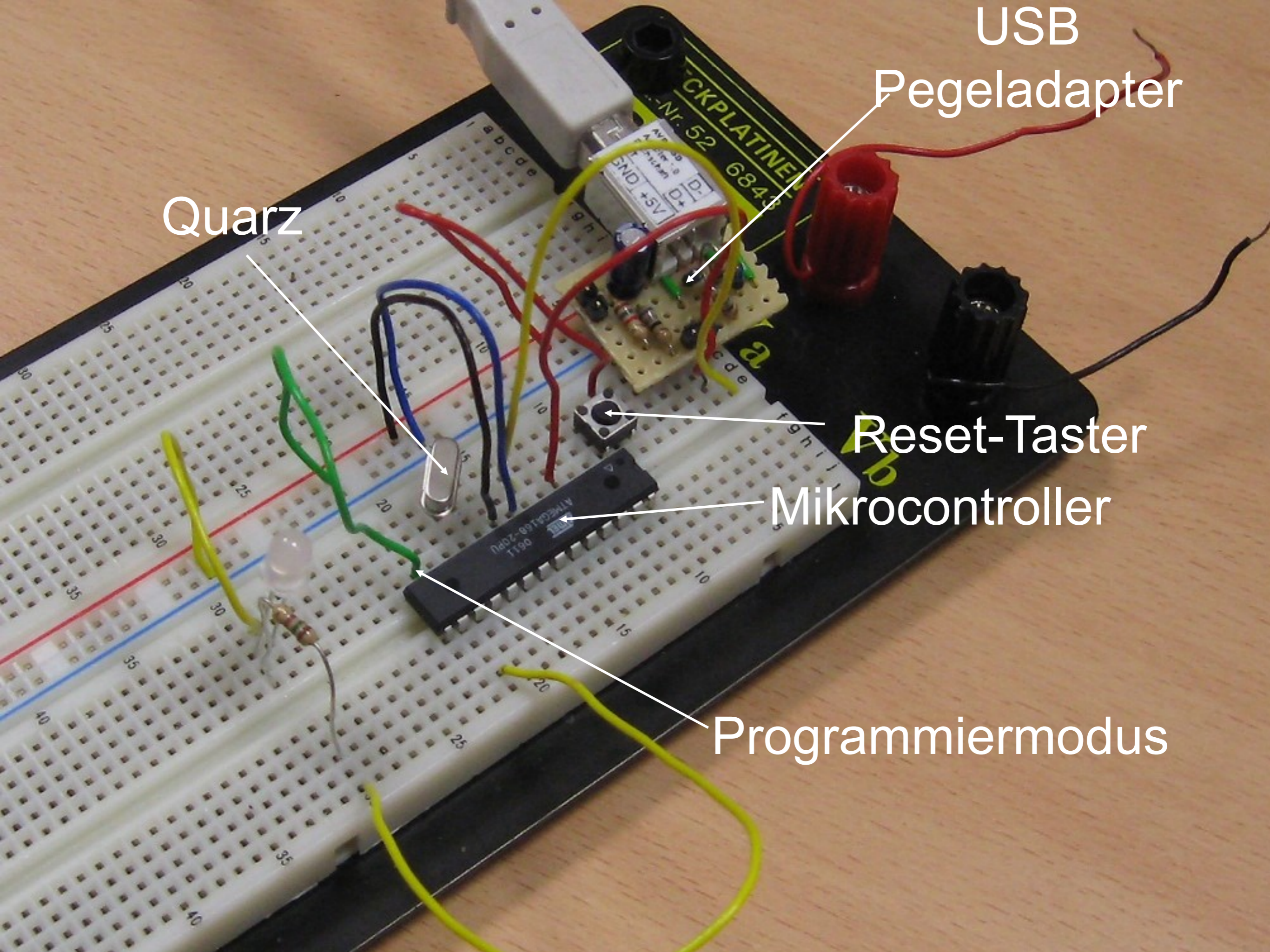


# Steckbrett



- Stromschiene vertikal
- Beschriftung vertauscht!
- Aufbau nach Bild in Anleitung
- Sonstige horizontal





USB

Pegeladapter

Quarz

Reset-Taster

Mikrocontroller

Programmiermodus

# Zusammenbau

- Bitte nichts kaputt machen 😊
- ATMEGA nicht selbst rausziehen!
- Schaltung überprüfen lassen!
- Nach dem Anschließen immer RESET Knopf drücken
- Beim Ändern der Schaltung USB raus!

# Programmier- umgebung

- Von der iST Webseite runterladen
  - `wget http://www.fs-ist.de/public/arduino/arduino.tar.gz`
  - `tar -xzf arduino.tar.gz`
  - `cd arduino`
  - `./ arduino`
- Siehe Anleitung

# Programmier- umgebung



# Vorgehen

- Code eintippen
- Vereinfachtes C/C++
- Play Symbol (Verify) drücken
- Reset-Taster betätigen
- Upload to Board drücken



# Programmaufbau

- `setup()`
  - 1x beim Start
- `loop()`
  - Endlos hintereinander

# Befehle

- `pinMode(pin, mode)`
  - `pinMode(10, OUTPUT)`
  - `pinMode(11, INPUT)`

# Befehle

- `digitalWrite(pin, value)`
  - `digitalWrite(10, HIGH)`
  - `digitalWrite(11, LOW)`
  - Bei INPUTs wird mit `digitalWrite()` der Pull-Up-Widerstand aktiviert.

# Befehle

- `int digitalRead(pin)`
- `void analogWrite(pin, value)`
- `void delay(millis)`

# 2-Farb-LED

- HIGH/LOW = 1. Farbe
- LOW/HIGH = 2. Farbe
- HIGH/HIGH = aus
- LOW/LOW = aus
  
- Widerstand nicht vergessen!

# Aufgabenstellung

- Hello World = LED blinken lassen
- 2-Farb-LED Wechselblinker
- Logikoperationen
- Dimmer
  
- Eigene Ideen? Gerne!

# Was jetzt?

- Fragen?
- Sonst:
  - C-Pool
  - 2er oder 3er Teams
  - Tutoren unterstützen euch
  - Unklarheiten? -> Fragen stellen!