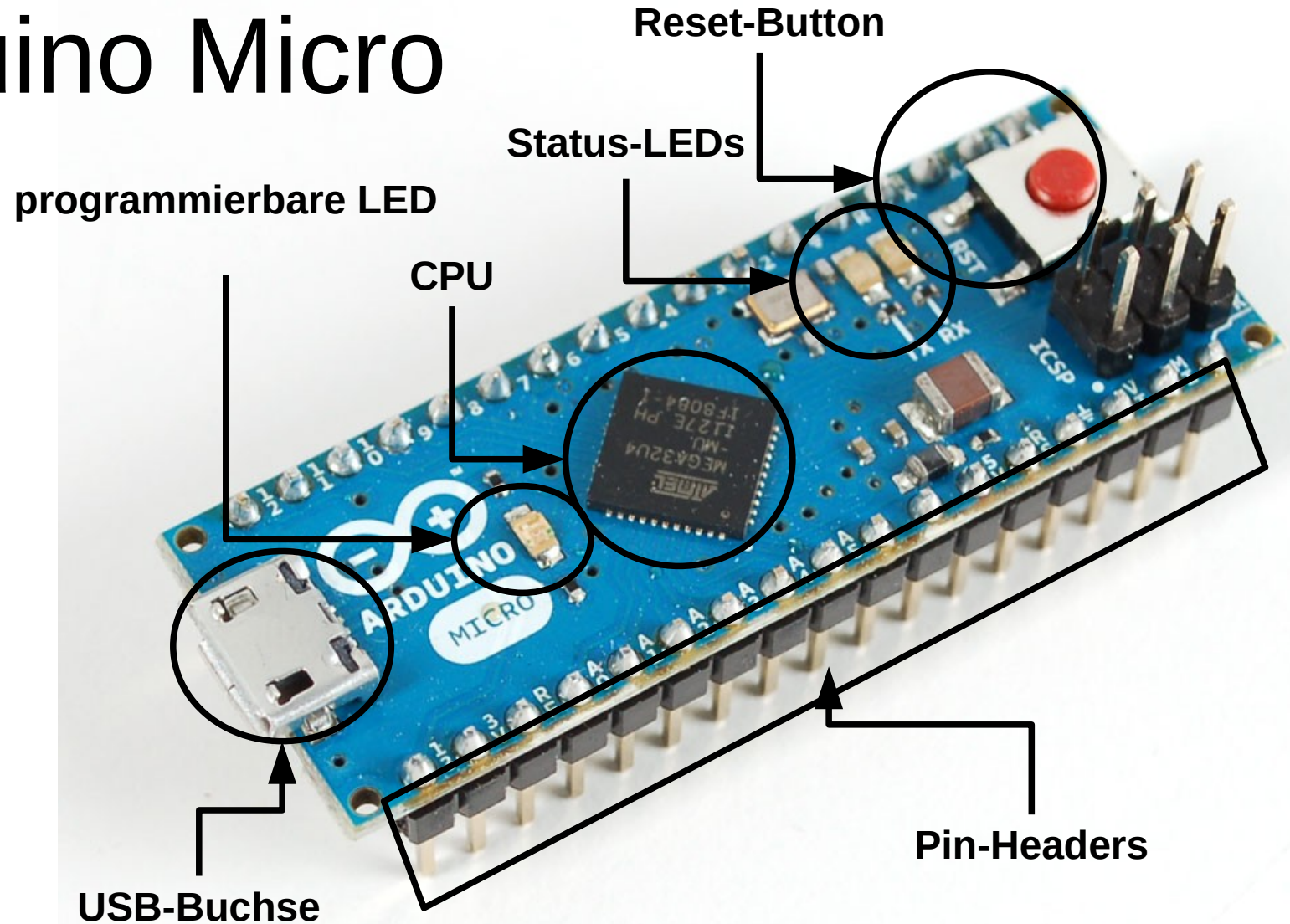


Sketching with Hardware

05: Arduino

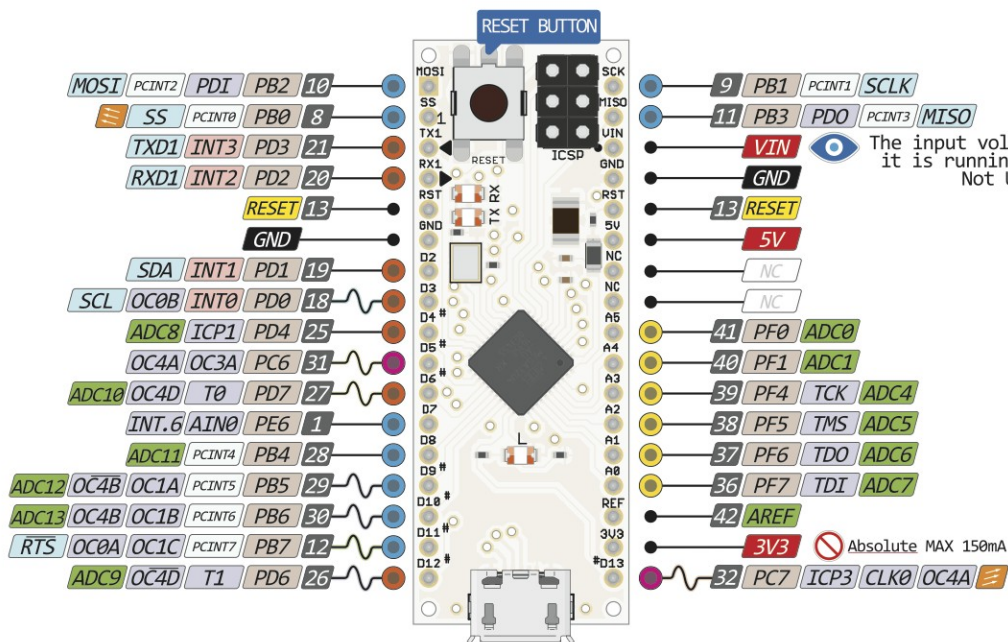
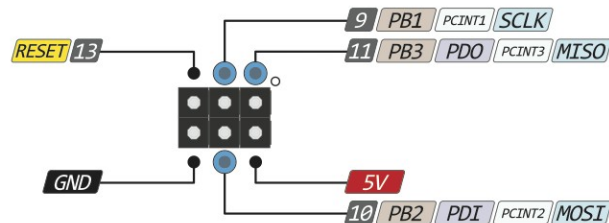
Arduino Micro



MICRO PINOUT

PWM TYPE

- 10bit
- 8/16bit
- HS
- 16bit
- 8bit



The input voltage to the board when it is running from external power. Not USB bus power.

- Power
- GND
- Serial Pin
- Analog Pin
- Control
- INT
- Physical Pin
- Port Pin
- Pin function
- Interrupt Pin
- PWM Pin
- Port Power

The power sum for each pin's group should not exceed 100mA

Absolute MAX per pin 20mA recommended 10mA

Absolute MAX 200mA for entire package

Den Arduino programmieren

- Microcontroller sind in Sachen Rechenleistung und Speicher eingeschränkt
- Programme werden in C oder C++ geschrieben
- Einfache Anwendungsfälle → Simple Programme
- Andere Patterns und Paradigmen als z.B. in Java

Grundlegende Programmstruktur

```
// declare variables here
```

```
// the setup routine runs once on startup
```

```
void setup() {
```

```
}
```

```
// the loop routine runs over and over again forever:
```

```
void loop() {
```

```
}
```

Hello World: Blink

```
// Pin 13 has an LED connected on most Arduino boards.  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)  
    delay(1000);                // wait for a second  
    digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW  
    delay(1000);                // wait for a second  
}
```

Arduinospezifische Funktionen

| Funktion | Beschreibung |
|---|--|
| <code>delay(int millis)</code> | Hält das Programm für <i>millis</i> Millisekunden an |
| <code>pinMode(int pin, int direction)</code> | Definiert einen Pin als <i>INPUT</i> oder <i>OUTPUT</i> . Sollte während <i>setup()</i> aufgerufen werden. |
| <code>digitalRead(int pin)</code> | Liest den Wert eines Pins. Gibt <i>HIGH</i> (1) oder <i>LOW</i> (0) zurück. |
| <code>analogRead(int pin)</code> | Liest den Wert eines Analogpins (zwischen 0 und 255). |
| <code>digitalWrite(int pin, int value)</code> | Setzt einen Pin auf <i>HIGH</i> (5 V) oder <i>LOW</i> (0 V). |
| <code>analogWrite(int pin, int value)</code> | Setzt die Spannung an einem Pin auf einen Wert zwischen 0 (0 V) und 255 (5 V). Funktioniert nur mit Pins, die <i>PWM</i> unterstützen. |
| <code>Serial.println("Hello")</code> | Ausgabe über den <i>Serial Monitor</i> . Kann zum Debuggen verwendet werden. <i>Serial.begin(9600)</i> muss vorher aufgerufen worden sein. |

Codestruktur

- Arduino-Code ist oft simpel und sequentiell
- Wird schnell unübersichtlich
- Fehlersuche in Code und Hardware ist umständlich
- Gut strukturierter Code ist leichter zu lesen und einfacher zu debuggen

Einige Tipps

- Arduino ist gut dokumentiert – nutzt die Doku!
- Code und Hardware ordentlich halten
- Code erweiterbar halten

Warum braucht man diskrete ICs?


- Einfache Schaltkreise können einfacher und **billiger** gebaut werden
 - Arduino: 5 – 30€
 - Simpler IC: < 1€
- Erweitern die Möglichkeiten des Arduino
- Vielzahl an spezialisierten Chips, z.B. für Audiowiedergabe, Sensoren, FPGA, etc.

```
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```



PlatformIO

```
arduino_blink/  
├── lib  
├── platformio.ini  
└── src  
    ├── arduino_blink.ino  
    └── main.cpp
```