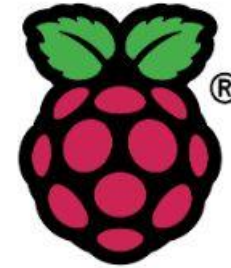
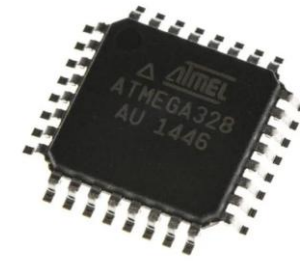




IIT

Industrielle Informationstechnologien

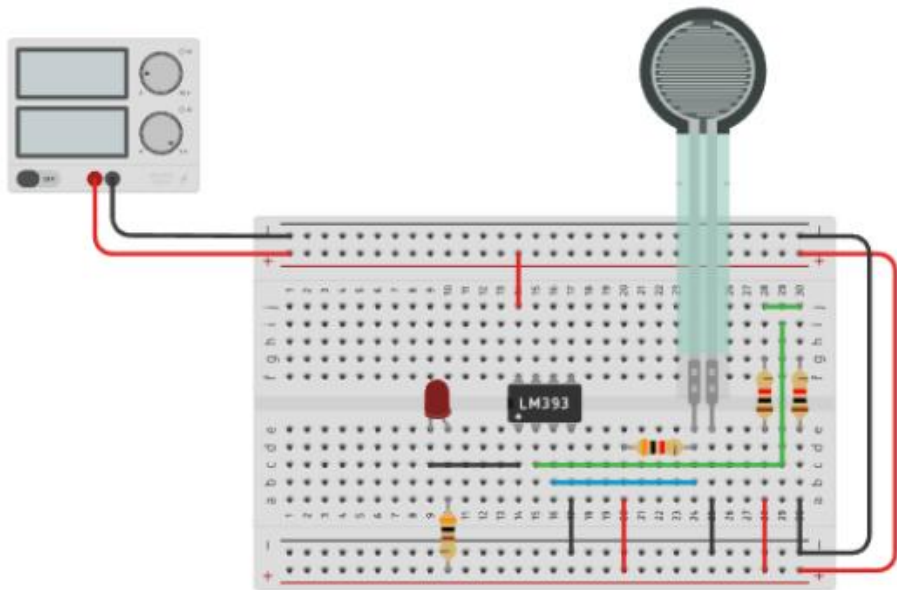
4. / 5. Jahrgang



- » Mikrocontroller- und Single Board Systeme
 - grundlegender Aufbau und Funktionsweise am Beispiel Arduino & Raspberry Pi
- » Prozessdatenverarbeitung & Visualisierung
 - Anbindung von Sensoren, Aktuatoren, Displays, ...
- » Kommunikation und industrielle Bussysteme
 - Anwendung gängiger Schnittstellen (UART, I²C, SPI, ...)
 - Funktionsweise industrieller Bussysteme (CAN-Bus, RS485)
- » Embedded Systems & industrielle Programmiermethoden
 - Theoretische Grundlagen und hardwarenahe Programmierung / Registermanipulation

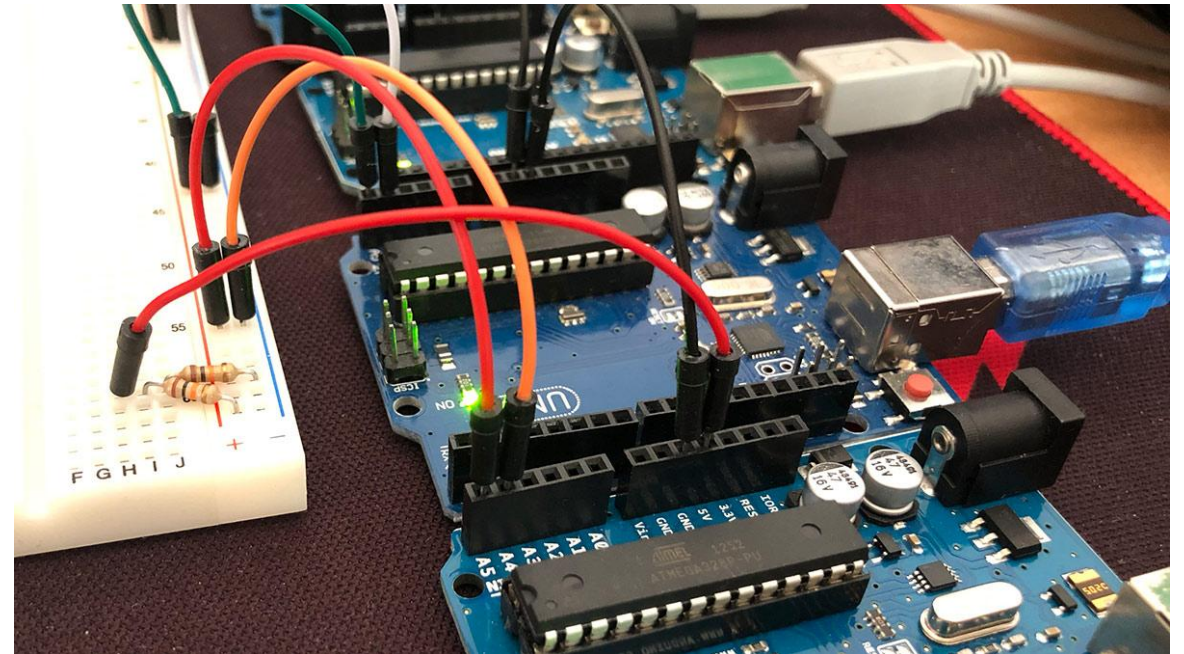
Inhalte

» Umsetzung von Hard- & Software



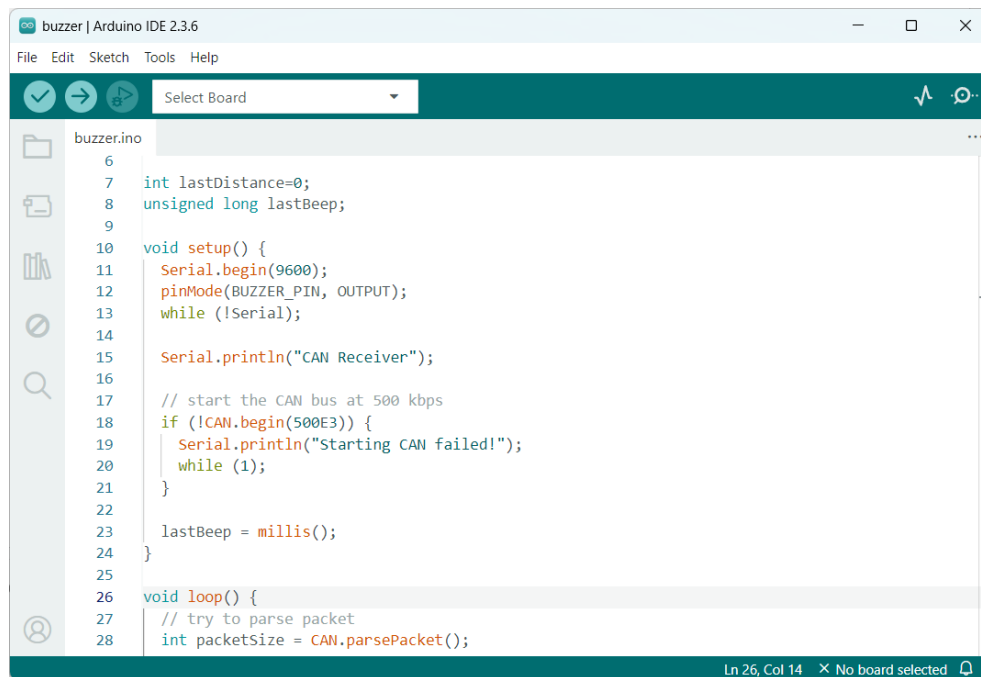
Schaltungsaufbau im Emulator

Umsetzung von Schaltungen im Labor

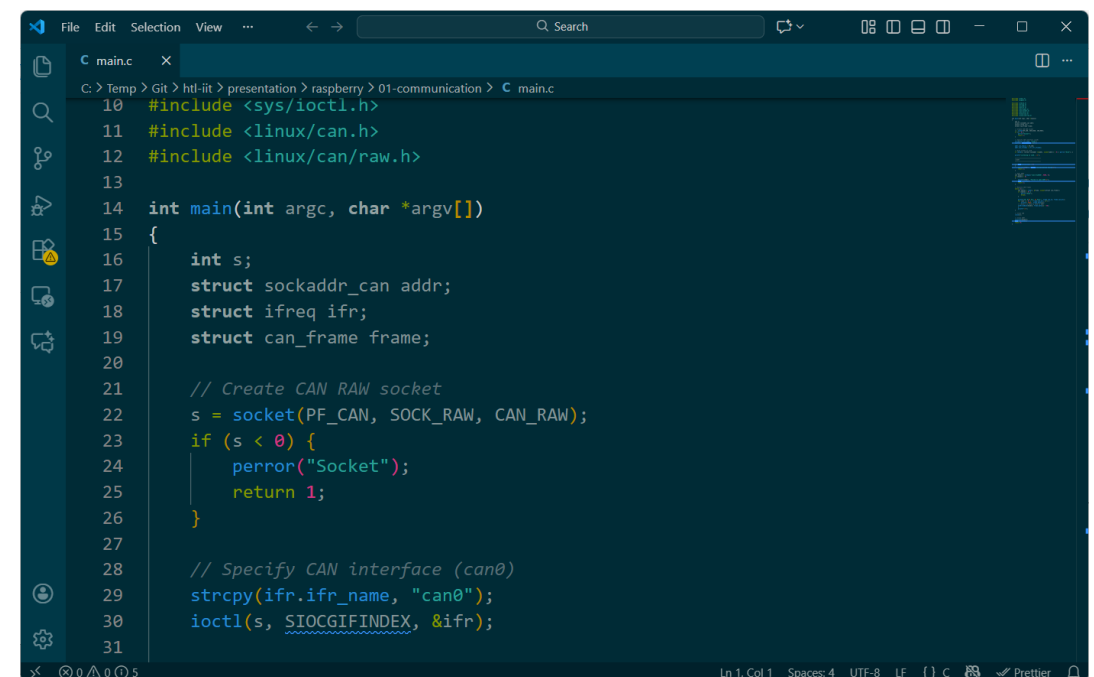


Inhalte

» Umsetzung von Hard- & Software



```
buzzer.ino
6
7 int lastDistance=0;
8 unsigned long lastBeep;
9
10 void setup() {
11   Serial.begin(9600);
12   pinMode(BUZZER_PIN, OUTPUT);
13   while (!Serial);
14
15   Serial.println("CAN Receiver");
16
17   // start the CAN bus at 500 kbps
18   if (!CAN.begin(500E3)) {
19     Serial.println("Starting CAN failed!");
20     while (1);
21   }
22
23   lastBeep = millis();
24 }
25
26 void loop() {
27   // try to parse packet
28   int packetSize = CAN.parsePacket();
```



```
main.c
10 #include <sys/ioctl.h>
11 #include <linux/can.h>
12 #include <linux/can/raw.h>
13
14 int main(int argc, char *argv[])
15 {
16     int s;
17     struct sockaddr_can addr;
18     struct ifreq ifr;
19     struct can_frame frame;
20
21     // Create CAN RAW socket
22     s = socket(PF_CAN, SOCK_RAW, CAN_RAW);
23     if (s < 0) {
24         perror("Socket");
25         return 1;
26     }
27
28     // Specify CAN interface (can0)
29     strcpy(ifr.ifr_name, "can0");
30     ioctl(s, SIOCGIFINDEX, &ifr);
31 }
```

Implementierung in modernen Entwicklungsumgebungen

Versuchsaufbau



Touch UI (Android)

Android Jetpack
Compose Entwicklung
(SEW4./5. JG)

