

---

# Fachprojekt for Embedded System: Design and Implement Your Own Embedded Systems (4)

Junjie Shi   Niklas Ueter

**LS 12, TU Dortmund**

09, April, 2018

- Grenzen des Arduino Frameworks
- Derzeitige Lösungen
  - ARTe
  - ERIKA Enterprise
- Einfache Beispiele

# Grenzen des Arduino Frameworks

---

- Keine *parallelen* Tasks unterstützt
- Nur eine einzige Schleife, in der Instruktionen ausgeführt werden
- Keine periodischen Ausführungen

- **Arduino Real-Time extension (ARTE):**

ARTE ist eine Erweiterung des Arduino Frameworks, das multitasking und real-time preemptive Scheduling unterstützt.  
⇒ Die Benutzer können mehrere Schleifen (Tasks) mit verschiedenen Perioden verwenden.

- **Arduino Real-Time extension (ARTE):**

ARTE ist eine Erweiterung des Arduino Frameworks, das multitasking und real-time preemptive Scheduling unterstützt.  
⇒ Die Benutzer können mehrere Schleifen (Tasks) mit verschiedenen Perioden verwenden.

- **ERIKA Enterprise:**

Ein Real-Time Operating System für kleine Microcontroller, das neuste Multi- und Many-core Chips unterstützt.

**Aber:** ERIKA unterstützt nur Arduino UNO.

# ARTe Beispiel I

---

```
//normal Arduino      //ARTe structure
void loop(){          | void loop1(p1){
                      |     <activity A>
                      | }
<activity A>         | void loop1(p2){
                      |     <activity B>
<activity B>         | }
                      | void loop1(p3){
<activity C>         |     <activity C>
                      | }
}                    | }
                      | //p1, p2, and p3 are the periods
```

---

# ARTE Beispiel

---

Drei verschiedenfarbige LEDs, die mit unterschiedlichen Frequenzen blinken.

- 1  $LED_1$ : 3s
- 2  $LED_2$ : 7s
- 3  $LED_3$ : 11s

# Normale Implementierung

---

```
int led_1 = 13; int led_2 = 14; int led_3 = 15;
int count = 0;
void loop(){
    if (count%3 == 0)
        digitalWrite(led_1);
    if (count%7 == 0)
        digitalWrite(led_2);
    if (count%11 == 0)
        digitalWrite(led_3);
    if (count == 3*7*11)
        count = 0;
    count ++;
    delay(1000);
}
```



# Implementierung in ARTe

---

```
int led_1 = 13;
int led_2 = 14;
int led_3 = 15;

void loop1(3000){
    digitalToggle(led_1);
}

void loop2(7000){
    digitalToggle(led_2);
}

void loop3(11000){
    digitalToggle(led_3);
}
```

---

# Arduino Real-Time extension: Woher?

---

- ARTe kann hier heruntergeladen werden:  
<http://retis.sssup.it/?q=arte>
- **Hinweis:**
  - ARTe nur für Windows
  - Java 8 (32bit)
- ARTe ist nicht *maintained*  $\Rightarrow$  Fehler / Probleme nicht ausgeschlossen!

- API ist *portable* – Die ERIKA API für verschiedene Microcontroller vorhanden
- Verbesserte Performance – Hochkonfigurierbarer *ROM footprint*, hoch effizienter Kernel und *Stack sharing*
- Multi-core Unterstützung – Multi-core Probleme werden ohne weiteres Zutun des Benutzers behandelt  $\Rightarrow$  Einfaches Applikationsdesign und Verifikation
- Quellcode ist vollständig verfügbar!

- ERIKA Enterprise kann hier heruntergeladen werden:

[http://www.erika-enterprise.com/wiki/index.php?title=Quick\\_start\\_guide](http://www.erika-enterprise.com/wiki/index.php?title=Quick_start_guide)

- Die Anleitung zur Installation, dem Betrieb und Beispiel-Applikationen für Arduino UNO können hier heruntergeladen werden:

[http://www.erika-enterprise.com/wiki/index.php?title=AVR8\\_-\\_Arduino\\_application\\_build\\_on\\_Windows](http://www.erika-enterprise.com/wiki/index.php?title=AVR8_-_Arduino_application_build_on_Windows)

- Die Installation und der Betrieb sind auf Linux ähnlich

# Fragen

---

Weitere Fragen?

