

# Programmieren für Ingenieure

Sommer 2015

Andreas Zeller, Universität des Saarlandes

## Ein Rechner



- Gerät, das mittels *programmierbarer Rechenvorschriften* *Daten verarbeitet.*

## Rechner sind überall









# Vorlesung

– Programmierthemen –

- Start mit Arduino
- Grundlagen  
(Anweisungen, Steuerung, Hauptschleife)
- Funktionen mit Parametern
- Kontrollstrukturen
- Felder

---

---

---

---

---

---

---

---

---

---

# Vorlesung

- Zeichen
- Ein-/Ausgabe
- Algorithmen
- Graphen
- Datenstrukturen
- Testen + Fehlersuche

---

---

---

---

---

---

---

---

---

---

# Übungen

– Projekte –

- Blinklicht
- Morse-Code
- Fußgängerampel
- Nim-Spiel
- Tic-Tac-Toe

---

---

---

---

---

---

---

---

---

---

# Übungen

- Weitere Sensoren
- Messungen verarbeiten
- Navigation
- Webserver
- Internet der Dinge

---

---

---

---

---

---

---

---

---

---

## Arbeit



---

---

---

---

---

---

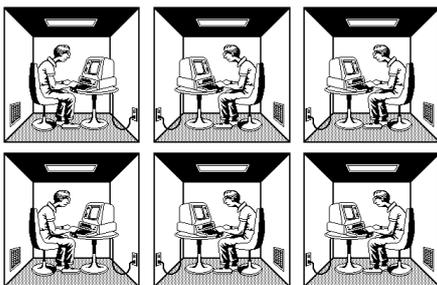
---

---

---

---

## Team



---

---

---

---

---

---

---

---

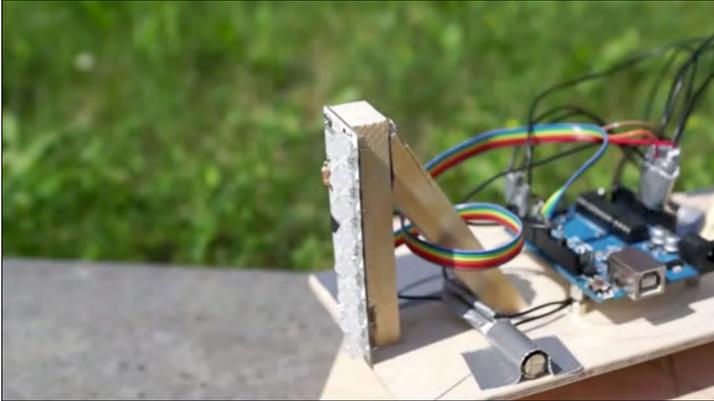
---

---



# Licht-Uhr

Murat Güner, Maximilian Junk, Pierre Kehl und Thomas Kreis



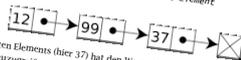
## Klausur

- Prüft Ihre Programmierfähigkeiten
- Am Ende der Vorlesung

### 3 Datenstrukturen [15 Punkte]

Eine verkettete Liste ist eine dynamische Datenstruktur, die eine Speicherung von miteinander Beziehung stehenden Objekten erlaubt. Die Anzahl der Objekte ist im Vorhinein nicht bekannt. Die Liste wird durch Zeiger auf das jeweils folgende Element realisiert. Das folgende Bild zeigt eine Liste, bestehend aus drei Elementen. Jedes Element ist definiert als

```
struct Elem {  
    int value; // Der Wert  
    struct Elem *next; // Zeiger auf das nächste Element  
};
```



Der Zeiger des letzten Elements (hier 37) hat den Wert NULL. Um auf eine Liste zuzugreifen, fängt man beim ersten Element (hier 12) an, und folgt dann dem Wert  $x$  in der Liste  $e$  enthalten ist. Wenn ja, gibt sie einen Zeiger auf das Element zurück, wenn nicht, gibt sie NULL zurück.

```
// Erstes Element der Liste LIST mit Wert X zurückgeben  
(oder NULL, wenn nicht gefunden)  
struct Elem *search(struct Elem *list, int x) {  
    while (e != NULL && e->value != x) {  
        e = e->next;  
    }  
    return e;  
}
```

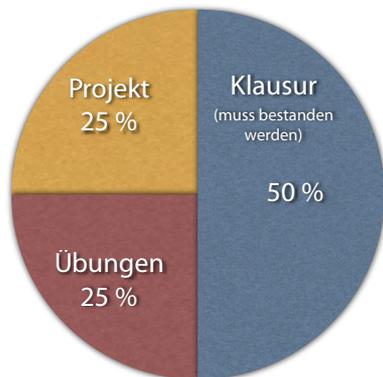
Ihre Aufgabe ist es, eine Funktion zu schreiben, die ein gegebenes Element  $e$  als letztes Element an eine nicht-leere Liste  $list$  anhängt.

a) [3 Punkte] Nehmen wir an, Sie möchten ein Element mit Wert 44 an die Liste anhängen. Zeichnen Sie (ähnlich zu obigem Diagramm), wie die Liste nach dem Anhängen aussieht.

b) [8 Punkte] Im ein Element anhängen, müssen Sie zunächst das letzte Element finden, und das letzte Element liefern. Schreiben Sie eine Funktion `last()`, die (ähnlich wie `search()` oben) durch die Liste geht, // Letztes Element der Liste LIST zurückgeben  
struct Elem \*last(struct Elem \*list) {  
 // Ihr Code hier  
}

c) [4 Punkte] Geben Sie nun ein existierendes Element  $e$ . Nutzen Sie `append()` und `search()` um an das letzte Listenelement das Element  $e$  anzuhängen.

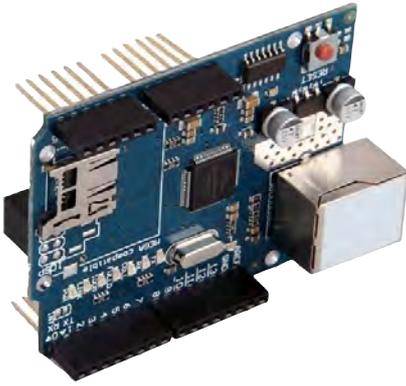
## Gesamtbewertung



- Bei 8 CP



# Was Sie brauchen



W5100 Ethernet Schild für Arduino Mega

---

---

---

---

---

---

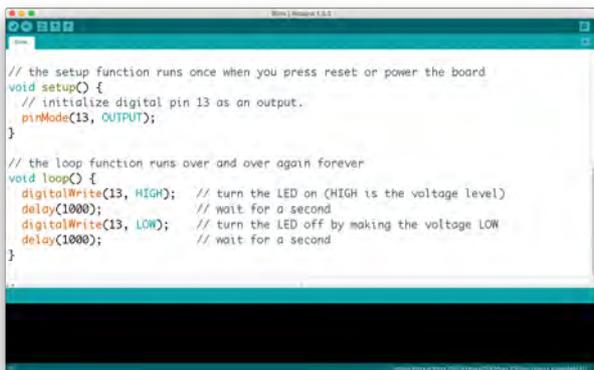
---

---

---

---

# Was Sie brauchen



Entwicklungsumgebung (Windows, Mac OS X, Linux)

---

---

---

---

---

---

---

---

---

---

# Was Sie brauchen



---

---

---

---

---

---

---

---

---

---



<https://www.st.cs.uni-saarland.de/edu/ping/>

## Nächste Schritte

- Anmeldung
- Einteilung in Gruppen
- Beschaffung von Boards und Bauteilen



<https://www.st.cs.uni-saarland.de/edu/ping/>

### Ihr Rechner



### Aufbau

- Vorlesung zur Programmierung
- Übung auf Arduino-Board
- Freies Projekt
- Klausur am Vorlesungsende

<https://www.st.cs.uni-saarland.de/edu/ping/>

### Gesamtbewertung



• Bei 8 CP

### Nächste Schritte

- Anmeldung
- Einteilung in Gruppen
- Beschaffung von Boards und Bauteilen



<https://www.st.cs.uni-saarland.de/edu/ping/>

# Handouts

## Ein Rechner



- Gerät, das mittels programmierbarer *Rechenvorschriften* *Daten verarbeitet.*

## Ihr Rechner

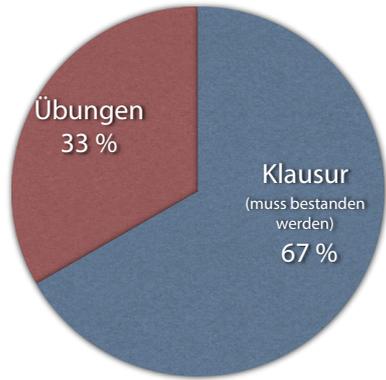


Anfängerset "Funduino MEGA 1"

Funduino



# Gesamtbewertung



• Bei 5 CP

---

---

---

---

---

---

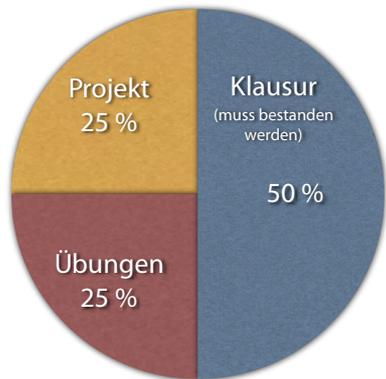
---

---

---

---

# Gesamtbewertung



• Bei 8 CP

---

---

---

---

---

---

---

---

---

---

# Nächste Schritte

- Anmeldung
- Einteilung in Gruppen
- Beschaffung von Boards und Bauteilen



<https://www.st.cs.uni-saarland.de/edu/ping/>

---

---

---

---

---

---

---

---

---

---