

# Projektaufgabe

1.62 ELEKTRISCHE ARBEIT / LEISTUNG

Gliederung

Aufgabenstellung

Programmierungsumgebung

Erarbeiten der Lösungswege

ASM Code

Performance vergleich mit C

Fazit

“ In ui.txt sind Spannungs- / Stromwerte abgespeichert. Berechnen Sie jeweils die elektrische Arbeit  $W = I \times t \times U$ , sowie die elektrische Leistung  $P = I \times U$ .

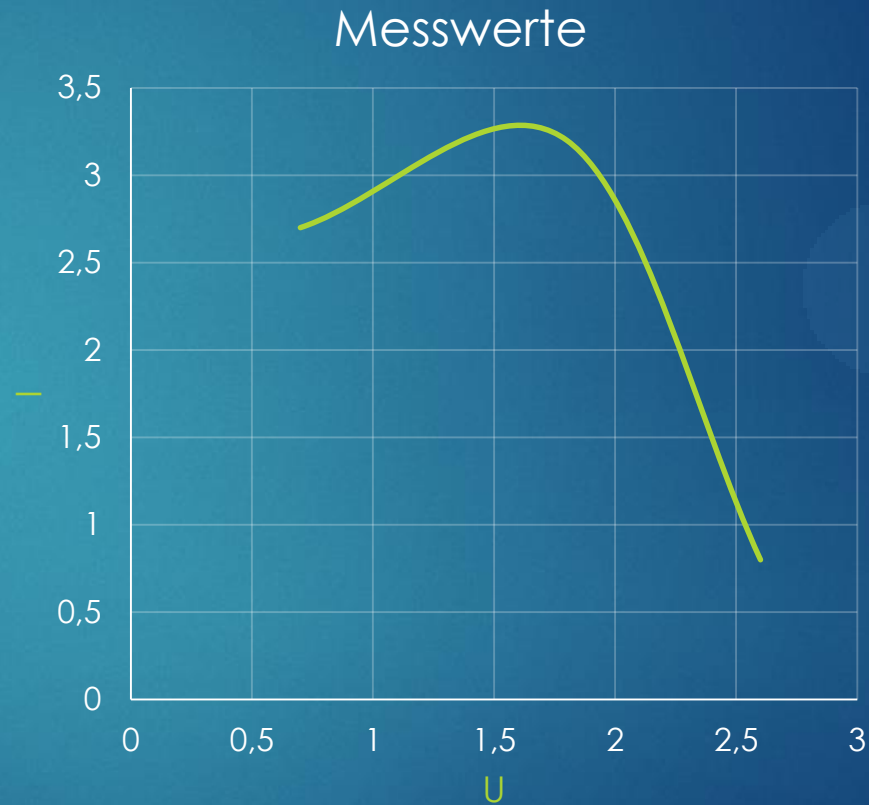
Sie benötigen den zusätzlichen Parameter  $t$ . ”  
Gehen. Sie. Damit. Sinnvoll. Um.

Aufgabenstellung

# Vorbereitung

4

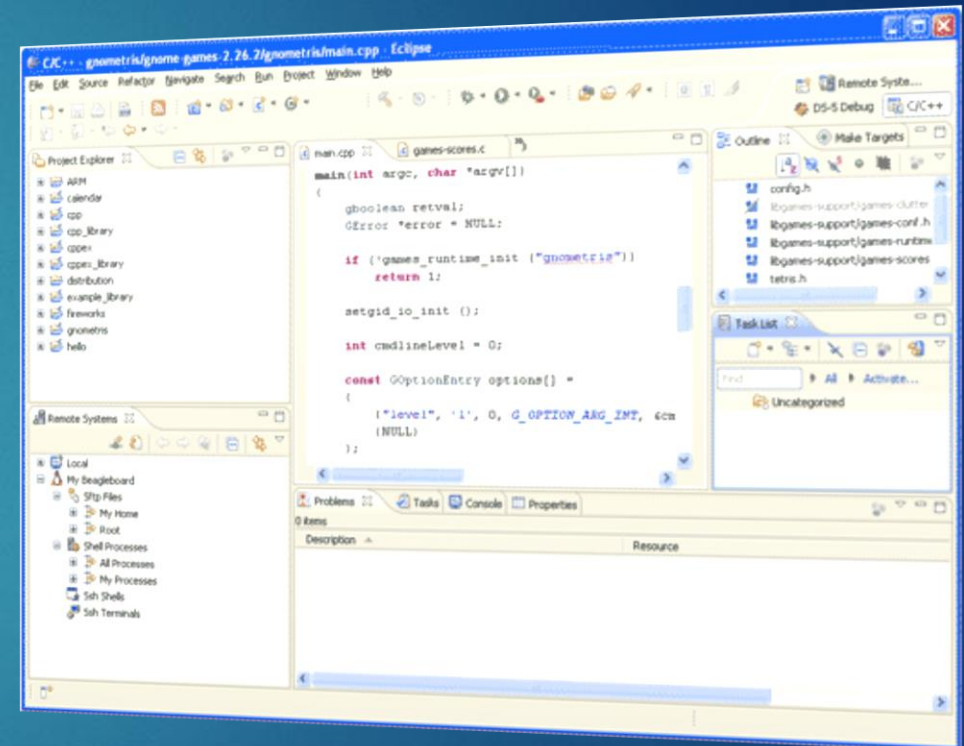
- ▶ Erschließung der Aufgabe
- ▶ Planung
- ▶ Messwerte



# Entwicklungsumgebung: DS-5

5

- ▶ Eingebunden in Eclipse
- ▶ ARM-Compiler
- ▶ C/C++ Editor
- ▶ Einfach Handhabung

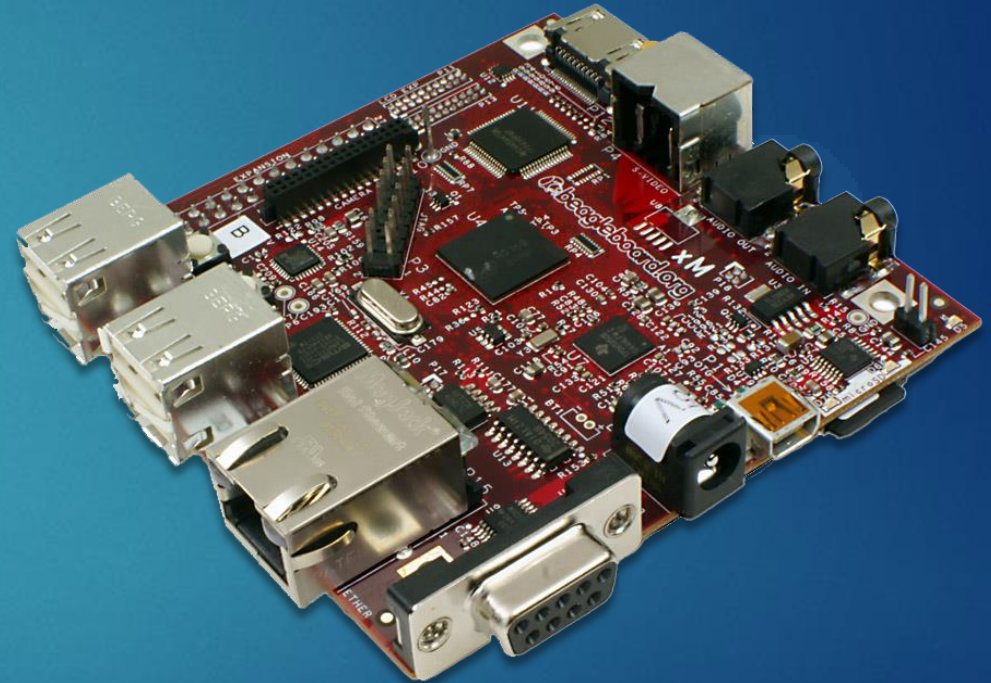




# BeagleBoard-xM

Der kleine Hobby-Computer für 150,- Euro:

- ARM® Cortex™ A8 @1 GHz
- 512MB DDR RAM
- ... und eine ganze Menge Anschlüsse!



# Lösungsmöglichkeiten

7

- ▶ Übergabe der Parameter
- ▶ Ausgabe
- ▶ Berechnung

# Erstellung des Codes

```
void calc_c(unsigned int num, float *current, float *voltage,  
float *time, float *result) {  
    int i;  
    float *res2 = result + num;  
  
    for(i = 0; i < num; i++) {  
        result[i] = current[i] * voltage[i];  
        res2[i] = result[i] * time[i];  
    }  
}
```



# Vorteile der ASM-Implementierung:

9

- ▶ kein Stack Frame
- ▶ keine temporären Variablen
- ▶ bessere Ausnutzung der Speicheranbindung
- ▶ bessere Ausnutzung der NEON-Recheneinheit

# Performance Vergleich

100 Durchläufe mit je 1.000.000 Messwerten

10

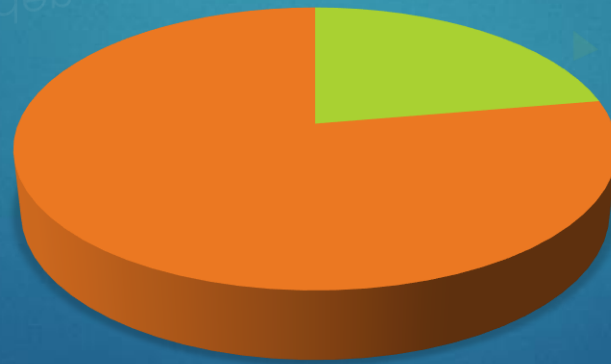
C-Code

▶ 14,71 Sekunden

Unserer ASM Code

▶ 4,22 Sekunden

▶ 249% Schneller



# Fazit

Auch bei weniger komplexen Rechnungen lässt sich mit ASM viel Zeit sparen.



Vielen Dank für Eure  
Aufmerksamkeit!

© JAKOB LEITMEIR && PIERRE SIEBENTRITT