

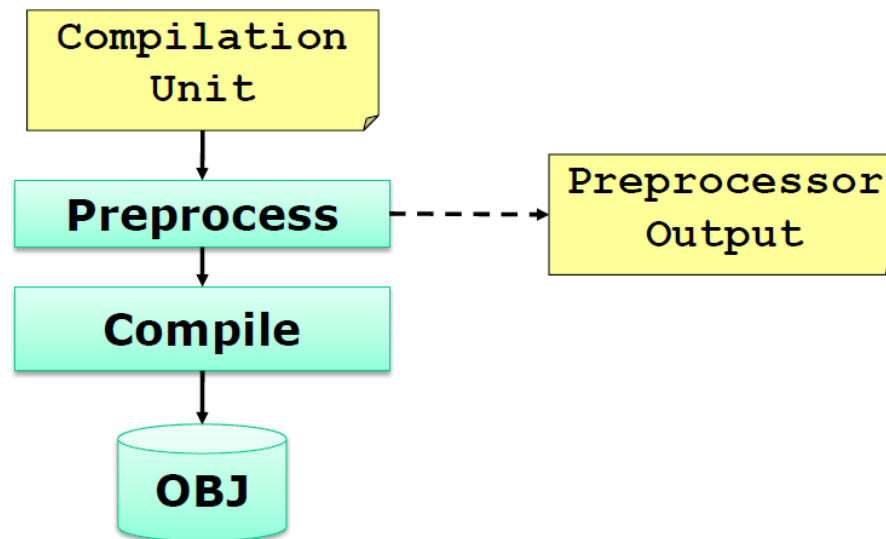
Recap SW 2

Macros, Includes, Doxygen

Team 4: Silvan Murer, Pascal Muff

Macros

- Präprozessor-Instruktionen welche vor dem Kompilieren ersetzt werden
- Unterstützen die Wiederverwendbarkeit von Code



Macros

- Textuelle Ersetzungen
- Zusätzliche Funktionen mit IF-Abfragen können für Konfigurations-Einstellungen genutzt werden

```
#define BLUE 0  
#define RED 1  
#define YELLOW 2
```

```
int ChangeColor(int color) {  
    if (color == BLUE) {  
        return RED;  
    }  
}
```

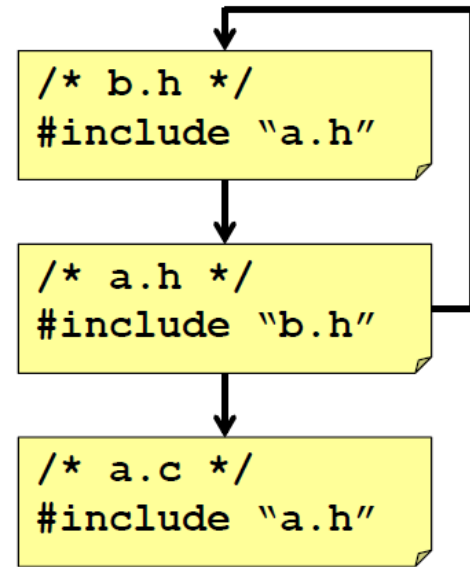
```
int ChangeColor (int color) {  
    if (color == 0) {  
        return 1;  
    }  
}
```

Includes

- 2 verschiedene Files:
 - *.c: Implementation, Definition
 - *.h: Interface/Header File: externe Deklarationen
- Lokale Variablen werden im *.c-File definiert
- Globale Variablen werden im *.h-File mit extern deklariert

Includes

- Mögliches Problem: Rekursive Includes



- Abhilfe:

Präprozessor-Definition: `__<FileName>_H_`

```
/* platform.h */
#ifndef __PLATFORM_H_
#define __PLATFORM_H_
    #define LED PORTA
#endif /* __PLATFORM_H_ */
```

Doxygen

- Tool, welches aus Sourcecode-Kommentaren Dokumentation generiert
- HTML oder LaTeX Output
- Eclox: Plugin mit Doxygen-Compiler für Eclipse

```
/**  
 * text  
 */  
  
/*!  
 * text  
 */  
  
/*!  
 text  
 */
```

```
/*!  
 \brief Switches on a set of LED.  
 \param[in] Leds The set of LED to be switched on.  
 */  
void LED_On(LED_Set Leds);  
  
void LED_On(LED_Set Leds) {  
    /* Turns on all LED's ORed in the  
    /*! \todo implement function */  
}
```

```
void LED_On ( LED_Set Leds )
```

Switches on a set of LED.

Parameters:

[in] Leds The set of LED to be switched on.

Todo:implement function

Definition at line 13 of file LED.c.

Fragen

- Warum sollten bei Präprozessor-Anweisungen mit Berechnungen immer Klammern gesetzt werden?

Beispiele:

```
#define CALC1 (2+5)
```

```
#define CALC2 (5*3)
```

In bestimmten Situationen könnte es sonst zu falschen Berechnungen kommen.

```
int example = 10 * CALC1;
```

Ohne Klammer würde $(10*2) + 5$ gerechnet

Fragen

- Was sind die Vor- und Nachteile von Macros?

Vorteile:

- Schnellerer Code
- Kleinerer Code
- Fixe Einträge im Code können zentral verwaltet werden

Nachteile:

- Interfaces sind Low-Level, es gibt keine Typensicherheit
- Verschachtelung des Codes
- Erschwertes Debugging

Fragen

- Wie sieht die genaue Definition des led.h-Files aus, damit es nicht rekursiv inkludiert werden kann?

```
#ifndef __LED_H_  
#define __LED_H_  
// Inhalt des *.h-Files  
#endif
```

- Wie und wo werden globale Variablen definiert?

Im *.h-File mit dem Zusatz «extern»

Beispiel: extern int LED_global;

Es sollte ein guter Name für die Variable gewählt werden, am besten mit Präfix, da dieser Name anschliessend im ganzen Projekt sichtbar ist.

Fragen

- Was lässt sich zusätzlich zur Sourcecode-Dokumentation noch mit Doxygen generieren?
 - Diagramme für Abhängigkeiten unter den Dateien
 - Message Sequenz – Diagramme
 - Zusätzliche externe Bilder können in die Dokumentation integriert werden