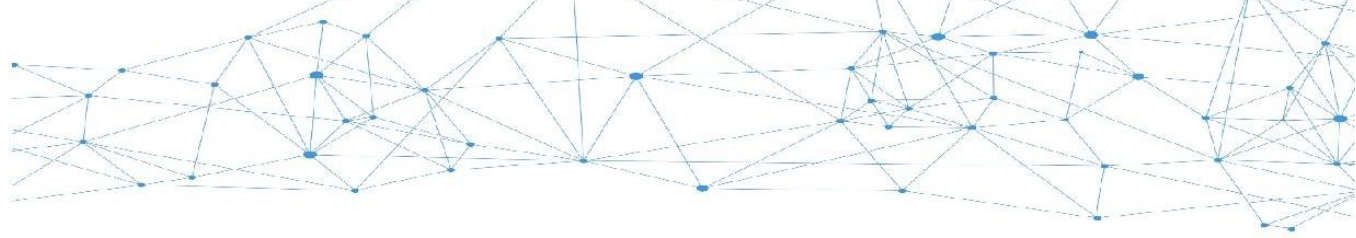


# Sitzung EEROS

## [Sequencer]

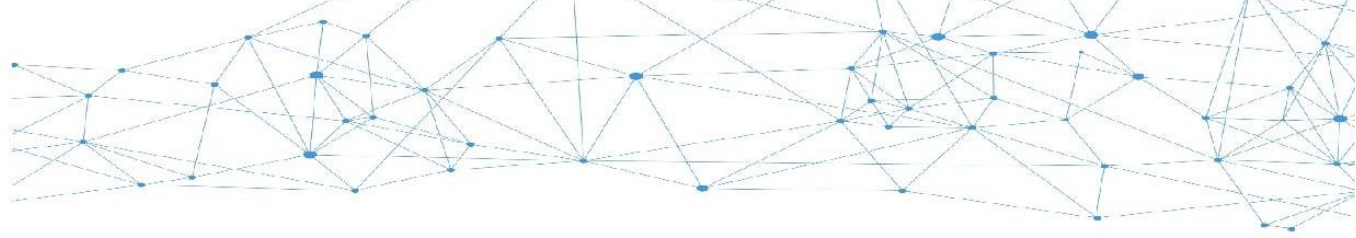
### Kalenderwoche 44

02.11.2016



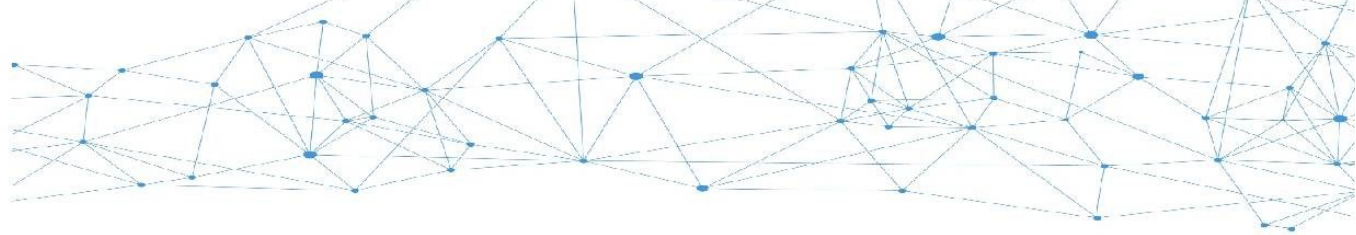
# Übersicht

- Definitionen Bezeichnungen
- Generelle Richtlinien
- Konzept: Condition / Action
- Weitere Arbeiten



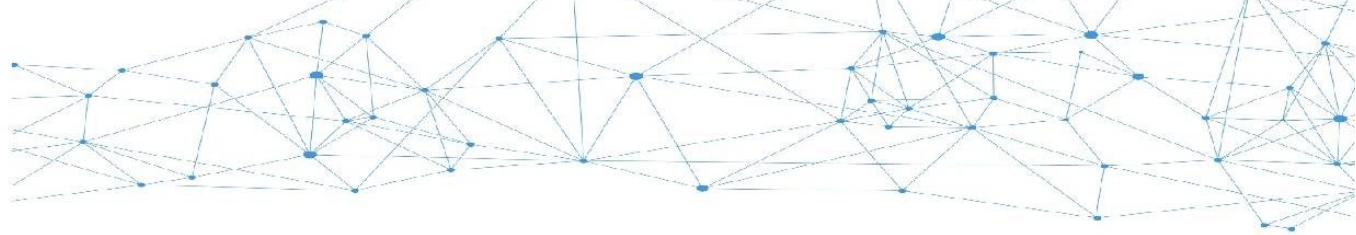
# Definitionen Bezeichnungen

- Applikations-Entwickler:
  - Wenig Programmierkenntnisse notwendig
  - Kann bestehende Sequenzen erstellen und abändern
- Roboter- / Steuerungsentwickler
- EEROS-Entwickler



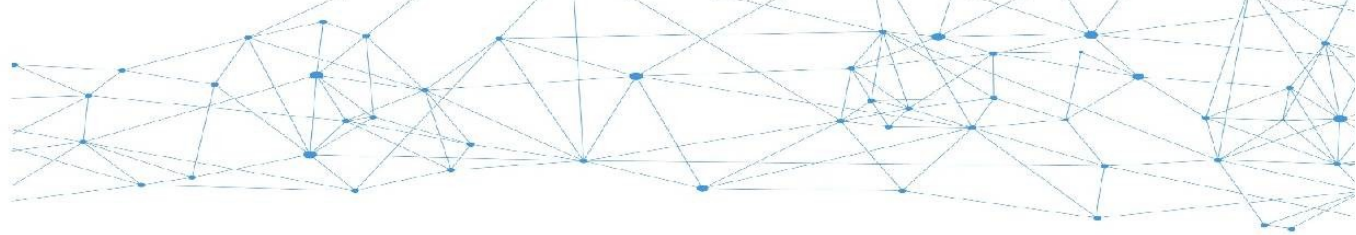
# Generelle Richtlinien

- Safety System soll entlastet werden
- Control System enthält keine Funktionen. Die Funktionen sind im Sequencer
- Step = Sequenz



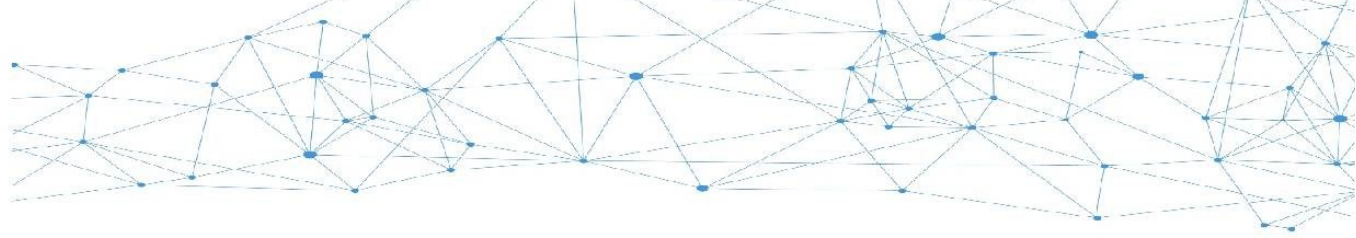
# Konzept **Condition** / Action 1

- Eine Condition ist ein Objekt, dass eine boolsche Variable enthält
- Eine Condition kann als Pre- oder Postcondition verwendet werden
- Eine Condition kann eine schlafende Sequenz wecken



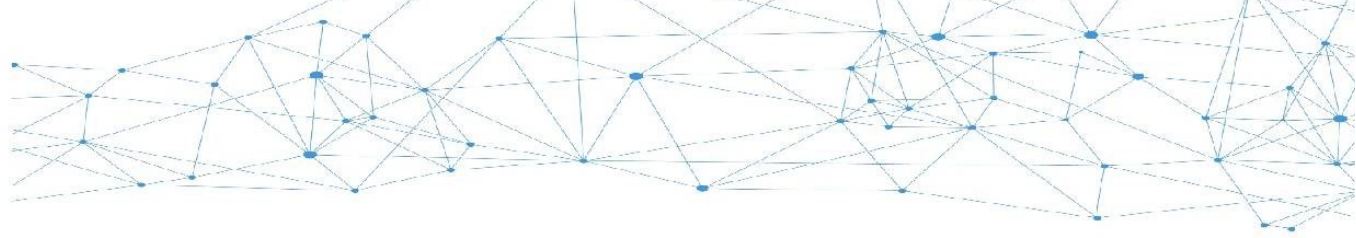
# Konzept **Condition** / Action 2

- Eine Condition wird möglicherweise sehr oft abgefragt → Beim Check einer Condition soll nur eine Boolean-Variable Abgefragt werden
- Ein Condition-Handler übernimmt, wenn notwendig, die Überprüfung der Condition
  - Je nach Natur der Condition:
    - Polling eines Wertes, z.B. ein Schalter
    - Periodisches Überprüfen von komplexeren Zusammenhängen aus SS, CS, Sequencer und HAL



# Konzept **Condition** / Action 3

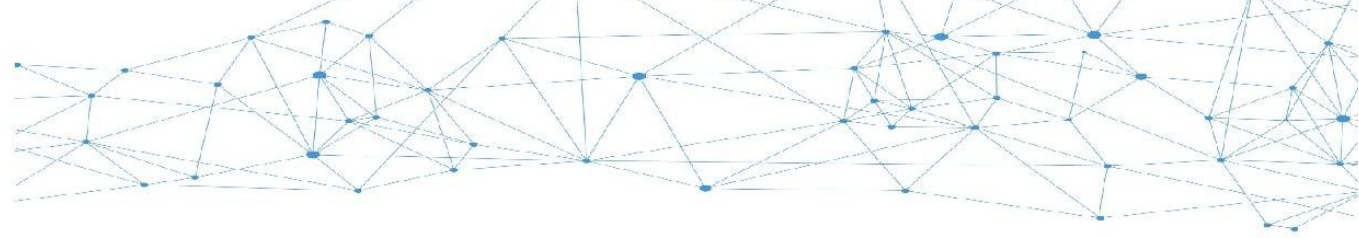
- Beispiele für Conditions:
  - Timeout
  - Safety Level Bereich
  - Signal aus CS
  - Roboter in Bewegung
  - Maus in den letzten 5 Sekunden bewegt



# Konzept Condition / **Action** 4

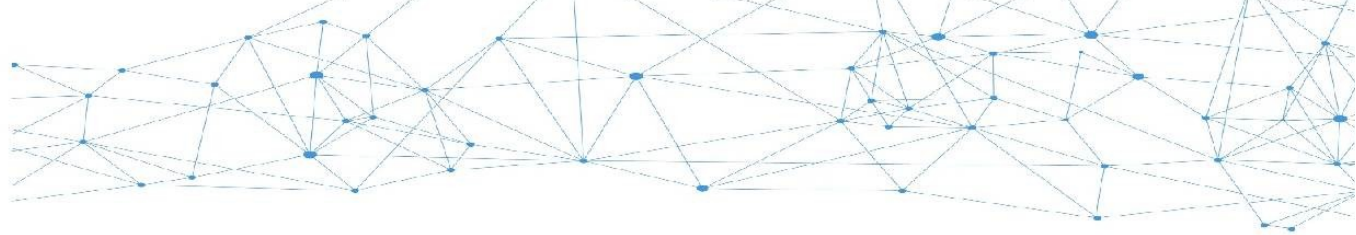
- Eine Action kann:
  - Eine blockierende Sequenz aufrufen
  - Eine nicht blockierende Sequenz aufrufen
  - Die aktuelle Sequenz pausieren (schlafen legen) oder aufwecken





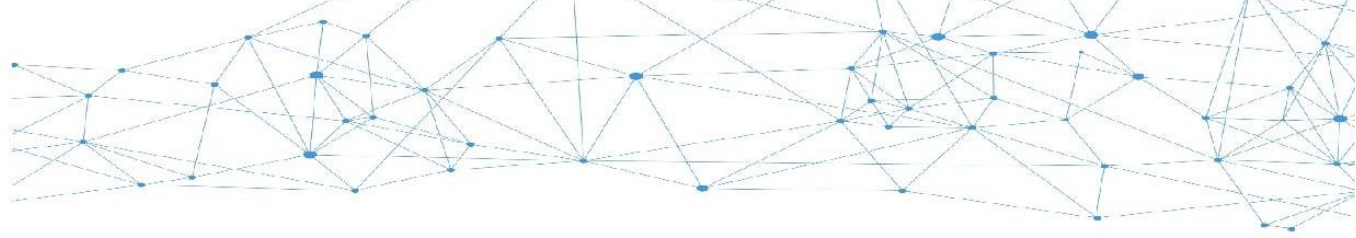
# Konzept **Condition / Action** 4

- Ein Condition-Action-Paar wird einer Sequenz per Referenz übergeben → Eine Action gehört zu einer bestimmten Condition von einer bestimmten Sequenz
- Ein Condition-Action-Paar kann anderen Sequenzen vererbt werden



# Konzept Condition / Action

```
class ConditionMouseNotMovedFor : public eeros::sequencer::Condition {  
  
    public:  
    ConditionMouseNotMovedFor(int timeout=5) {}  
  
    int period=100;>>    //in miliseconds  
  
    bool getConditionState() {  
        > return conditionState;  
    }  
  
    void checkCondition() {  
        > actPosition = HAL.getPosition();  
        > if (actPosition != lastPosition) {> //mouse moved  
        > > timer.reset();  
        > > }  
        > if (timer.getTime() > timeout) conditionState = true;  
        > else conditionState = false;  
    }  
};
```



# Weitere Arbeiten

- Konzept “Condition / Action” weiter ausarbeiten.  
Pseudocode Beispiel schreiben um Plausibilität zu zeigen