

Anleitung zur SEAP Simulationssoftware

23. März 2010

Inhaltsverzeichnis

1	Installation	2
1.1	Target-Rechner	2
1.2	Host-Rechner	2
1.3	Konfigurationsdatei	2
1.4	Bibliotheken	3
2	Oberfläche	3
2.1	Sensoren	3
2.1.1	Digitale Sensoren	3
2.1.2	Analoger Höhenmesser	3
2.2	Aktoren	3
2.2.1	Motor	3
2.2.2	Ampel	4
2.3	Werkstücke	5
3	Schnittstelle zur Anwendung	5
3.1	Belegung der I/O-Ports	5
3.2	Anwendungsbeispiel	6
4	Hinweise	7
4.1	Hinweis zum Timing	7
4.2	Hinweis zur Initialisierung der I/O-Karten	7
4.3	Bugreports	7

1 Installation

1.1 Target-Rechner

Zur Installation des Hauptprogrammes muss zunächst das Archiv `SEAP-Sim_v1.x.tar.gz` per `ftp` auf das Neutrino Zielsystem gebracht werden und danach die `install.sh` aus dem entpackten Archiv mit `root`-Rechten ausgeführt werden.

Die Simulation kann danach durch den Aufruf von `festosim` gestartet werden.

1.2 Host-Rechner

Zur Installation der für die Kommunikation mit der Simulation benötigten Bibliotheken müssen die Dateien `libioaccess.a`, `ioaccess.h` und `wrappers.h` in den QNX Verzeichnisbaum auf dem Host-Rechner integriert werden. Hierzu einfach das Verzeichnis `target` aus dem Archiv `lib.tar.gz` nehmen und auf das originale `target` Verzeichnis in `.../QNX6xx/` draufkopieren.

1.3 Konfigurationsdatei

Über eine Konfigurationsdatei können folgende Parameter der Simulation angepasst werden:

- Hörschwellwert (in mm) ab wann der Sensor "Höhe Bereich 1" reagieren soll (Defaultwert 24mm).
- Streuungsfaktor der analogen Höhenmessung (Defaultwert 0.3%).
- Symbolische Verzögerungszeit für A/D Umsetzung (Defaultwert 10 μ s).
- Reichweite des Hörsensors (Defaultwert 50mm).
- Ermittelter AD-Wert des realen Hörsensors bei leerer Transferstrecke (Defaultwert 816).
- Ermittelter AD-Wert des realen Hörsensors bei zwei Messung eines 50mm hohen Objektes (Defaultwert 4058).

Die Simulation sucht in `/etc/seap-sim.conf` nach Konfigurationsdaten. Wird diese Datei nicht gefunden, werden die Defaultwerte in den Klammern beibehalten. Die ausgegrauten Parameter dienen der Anpassung an Rekonfigurationen der realen Anlagen. Die Defaultwerte entsprechen der aktuellen Konfiguration der Anlagen (Stand 23. März 2010).

1.4 Bibliotheken

Die Benutzeranwendung welche mit der Simulation interagieren soll muss die Headerdatei `ioaccess.h` includieren und bei den Linker-Options entsprechend die Bibliothek hinzufügen. Vor Zugriff auf die Simulation muss die Laufzeitumgebung mit `IOaccess_open` initialisiert werden.

2 Oberfläche

Abb. 1 zeigt die Hauptansicht der Simulationssoftware. Sie stellt die Transferstrecke in 2D-Ansicht aus der Vogelperspektive dar.

2.1 Sensoren

2.1.1 Digitale Sensoren

Die digitalen Sensoren der Anlage (Lichtschranken, Metalldetektor, Höhenschwellwertssensor) werden als kleine Quadrate dargestellt. Die Farbe Rot signalisiert dabei einen nicht ausgelösten Sensor, grün einen ausgelösten Sensor.

Sensoren werden entweder von der Anlage selbst (als Reaktion auf bestimmte Ereignisse) ausgelöst, oder aber durch Klicken mit der Maus:

Linksklick: Togglen des Sensorpegels.

Rechtsklick: *Forcen* des Sensorpegels. Den Pegel eines Sensors zu forcen bedeutet ihn unabhängig von Umwelteinflüssen (die die normale Arbeitsweise des Sensors bestimmen) auf einem bestimmten Pegel zu halten. Zum Beispiel würde ein auf *high* geforcter Lichtschranken-Sensor immer noch *high* anzeigen, auch wenn der Lichtstrahl unterbrochen werden würde.

2.1.2 Analoger Höhenmesser

Der Spannungswert, den der analoge Höhensensor liefert wird in Volt ständig angezeigt. Diesen Wert kann man per Rechtsklick einfrieren (ein Kontextmenü eröffnet weitere Auswahlmöglichkeiten).

2.2 Aktoren

2.2.1 Motor

Der Antriebsmotor des Förderbandes wird über drei Buttons gesteuert. Diese haben eine höhere Priorität als die über die I/O-Ports eingehenden Steuersignale und bieten dadurch einen Override-Mechanismus.

2.2.2 Ampel

Die Ampel kann in der GUI durch einfaches Anklicken manipuliert werden.

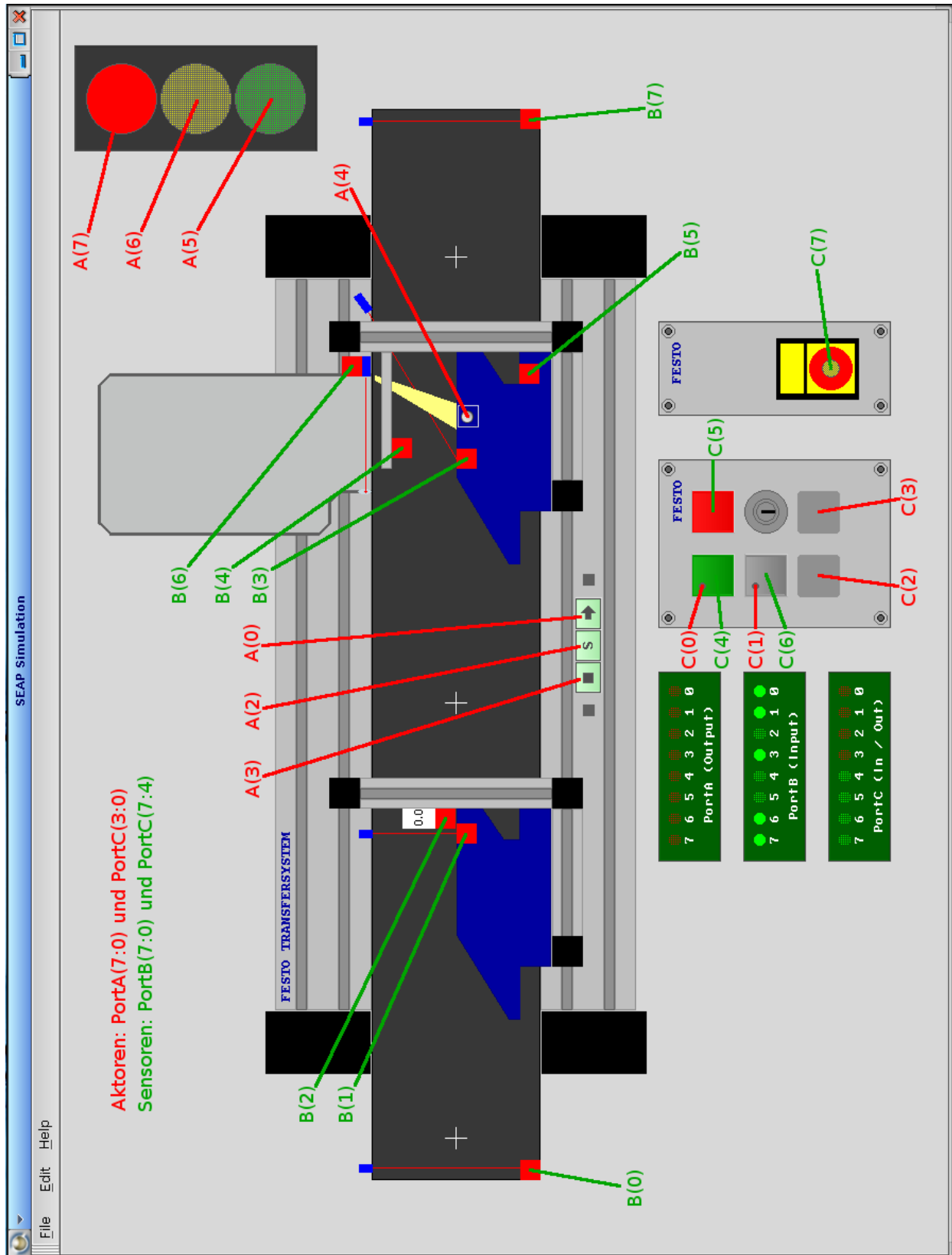


Abbildung 1: Hauptansicht in 2D aus Vogelperspektive

2.3 Werkstücke

Werkstücke können an den drei (durch weiße Kreuze) markierten Stellen auf das Laufband gelegt werden. Ein Rechtsklick auf eines der Kreuze eröffnet ein Kontextmenü in welchem man unter den vier Standardwerkstücken und einem benutzerdefinierten Werkstück wählen kann.

Durch Anklicken können diese wieder aus der Anlage entfernt werden.

3 Schnittstelle zur Anwendung

Nachdem wie im Abschnitt 1: Installation beschrieben, die Bibliotheken eingehangen und die Laufzeitumgebung initialisiert wurden gestaltet sich der Zugriff auf die Simulation wie der Zugriff auf die Tatsächliche Hardware über Interrupts und die QNX I/O-Befehle.

3.1 Belegung der I/O-Ports

Abkürzungen

LS	Lichtschanke
RLT	Reflexlichttaster
IS	Induktiver Sensor
HS	Höhenschwellwertssensor
WS	Werkstück
LU	Lichtschanke Unterbrochen
LNU	Lichtschanke Nicht Unterbrochen

Port C: Sensoren & Aktoren

C(7)	Taste E-Stop	0: on	1: off
C(6)	Taste Reset	0: off	1: on
C(5)	Taste Stop	0: on	1: off
C(4)	Taste Start	0: off	1: on
C(3)	LED Q2	0: off	1: on
C(2)	LED Q1	0: off	1: on
C(1)	LED Reset	0: off	1: on
C(0)	LED Start	0: off	1: on

Port A: Aktoren

A(7)	Ampel Rot	0: off	1: on
A(6)	Ampel Gelb	0: off	1: on
A(5)	Ampel Grün	0: off	1: on
A(4)	Weiche	0: close	1: open
A(3)	Motor Stop	0: off	1: on
A(2)	Motor Langsam	0: off	1: on
A(1)	Motor Linkslauf	nicht belegt	
A(0)	Motor Rechtslauf	0: off	1: on

Port B: Sensoren

B(7)	LS Auslauf	0: LU	1: LNU
B(6)	LS Rutsche voll	0: LU	1: LNU
B(5)	RLT Weiche offen	0: off	1: on
B(4)	IS WS Metall	0: off	1: on
B(3)	LS WS in Weiche	0: LU	1: LNU
B(2)	HS Bereich 1	0: off	1: on
B(1)	LS WS in Höhenmessung	0: LU	1: LNU
B(0)	LS Einlauf	0: LU	1: LNU

3.2 Anwendungsbeispiel

An folgendem Codebeispiel soll gezeigt werden, wie man die Bibliothek `ioaccess` einsetzt um über die `in8`, `out8` und die `Interrupt*`-Befehle auf die Simulation zuzugreifen und wie man den Wechsel zwischen Simulationsumgebung und Laborumgebung vollzieht:

```
#define SIMULATION

#include <stdio.h>
#ifdef SIMULATION
#include <ioaccess.h>
#endif

int main(int argc, char *argv[]) {

#ifdef SIMULATION
    IOaccess_open();
#endif

    printf("Hello World!\n");
    out8(0x300, 0xFF);

#ifdef SIMULATION
    IOaccess_close();
#endif
}
```

Das Umschalten zwischen Simulation und reale Anlage geschieht hier durch Ein- bzw. Auskommentieren des Makros `SIMULATION`.

Der Header `ioaccess.h` muss überall inkludiert werden wo die genannten Befehle durch die der Simulation ersetzt werden sollen.

Bei Anwendung auf die Simulation muss darauf geachtet werden dass die Simulation läuft bevor das Anwenderprogramm gestartet wird.

4 Hinweise

4.1 Hinweis zum Timing

Es wird keine 1:1 detailgetreue Abbildung des Timingverhaltens der Anlage gewährleistet. Gewisse Abweichungen bezüglich Bandlaufgeschwindigkeit, Zeitschranken etc. sind nicht zu vermeiden, daher ein Ratschlag an die Kommilitonen:

Um den Übergang von Simulation auf reale Anlage und zurück so schmerzfrei wie möglich zu gestalten sollten Zeitwerte für Laufzeitüberwachungen sowie auch Spannungswerte zur Erkennung von Werkstücken nicht fest einkodiert sein. Beim Festlegen geeigneter Zeitwerte wäre bspw. auch ein Testlauf eines Werkstückes von Bahnanfang bis Bahnende denkbar, welcher im Initialisierungsteil der Anwendung einmal durchläuft und für das jeweilige zu überwachende System (Simulation oder reale Anlage) die optimalen Zeitwerte bestimmt.

4.2 Hinweis zur Initialisierung der I/O-Karten

Die Simulation der I/O-Karten bilden das Verhalten der realen Karten nach. Also ist auch hier auf fehlerfreie Konfiguration zu achten. Ports die auf *out* eingestellt werden, sollten auch als Ausgang verwendet werden. Und Interrupts sollten enabled sein, wenn die betreffende I/O-Karte IRQs generieren soll. Ausserdem ist zu beachten, dass man über *File* → *Reset Simulation* auch alle Register der I/O-Karten zurücksetzt.

4.3 Bugreports

Bugreports bitte möglichst in reproduzierbar dokumentierter Form an:

bulut_m@informatik.haw-hamburg.de