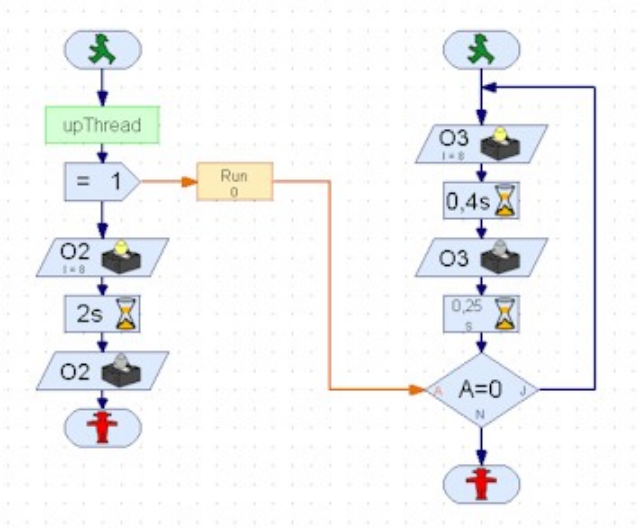
Threads

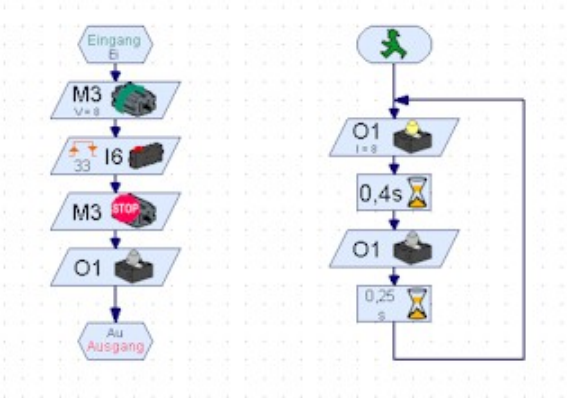
Arbeiten mit mehreren Prozessen (Threads)

Aufgabe : Ein RobMotor (M3) soll exact 33 Impulse fahren. Die Impulse werden über eine Kombination Impulsrad / Taster (I6) am Getriebe des Motors gezählt. Während des Motorlaufs soll die Lampe an O3 blinken. Nach Erreichen der 33 Impulse leuchtet die Lampe an O2 für 2 Sekunden.



Gelöst wird das durch ein Hautprogramm mit zwei Prozessen : Steuerung des Motors (in upThread) und dem zweiten Prozess mit dem Blinker. Beide Prozesse starten automatisch bei Programmstart (grüner Button). Das Blinken geschieht in einer Schleife, die beendet wird, wenn die Variable Run auf 1 gesetzt wird (zu Programmstart ist sie 0). Das geschieht im ersten Thread nach der Rückkehr aus upThread.

Alternative



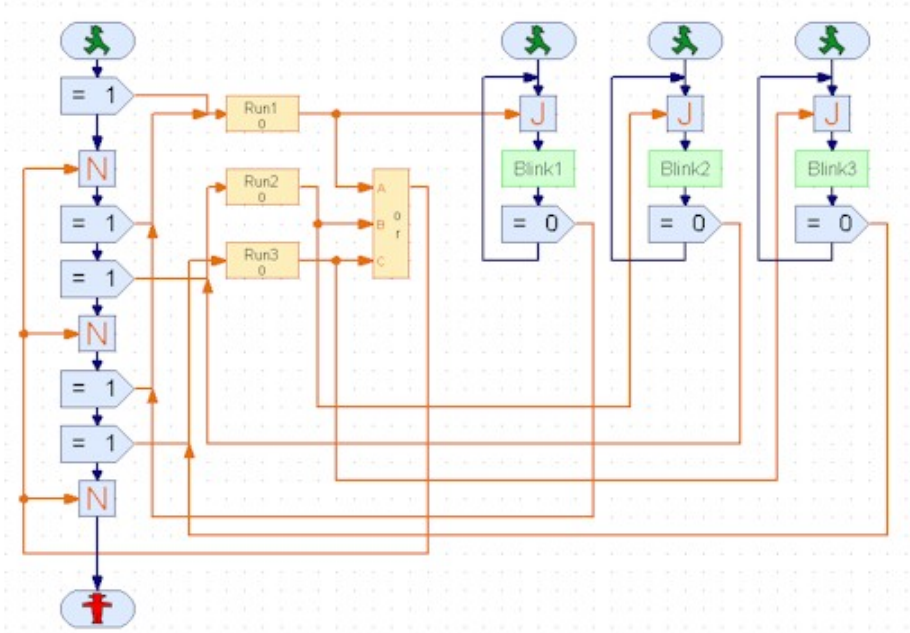
Man kann das Blinken auch in einem zweiten Thread auf der Seite des upThread unterbringen. In diesem Fall erübrigt sich eine Endabfrage, da dieser Thread parallel zu upThread gestartet **und** beendet wird. Da das Beenden recht zufällig passieren kann (die Lampe kann noch an sein), wird die Lampe an O1 vorsichtshalber am Ende von upThread noch abgeschaltet.

Jetzt blinken in dem Programm gleich zwei Lampen, wenn der Motor läuft, auf eine

kann man natürlich verzichten.

Achtung : Wenn man auf den Geschmack gekommen ist und gleich noch ein paar Prozesse kreiert, sollte man an die Defaultvorgabe 5 denken und sie ggf. erhöhen (Beim jeweiligen Programm unter Tab Eigenschaften).

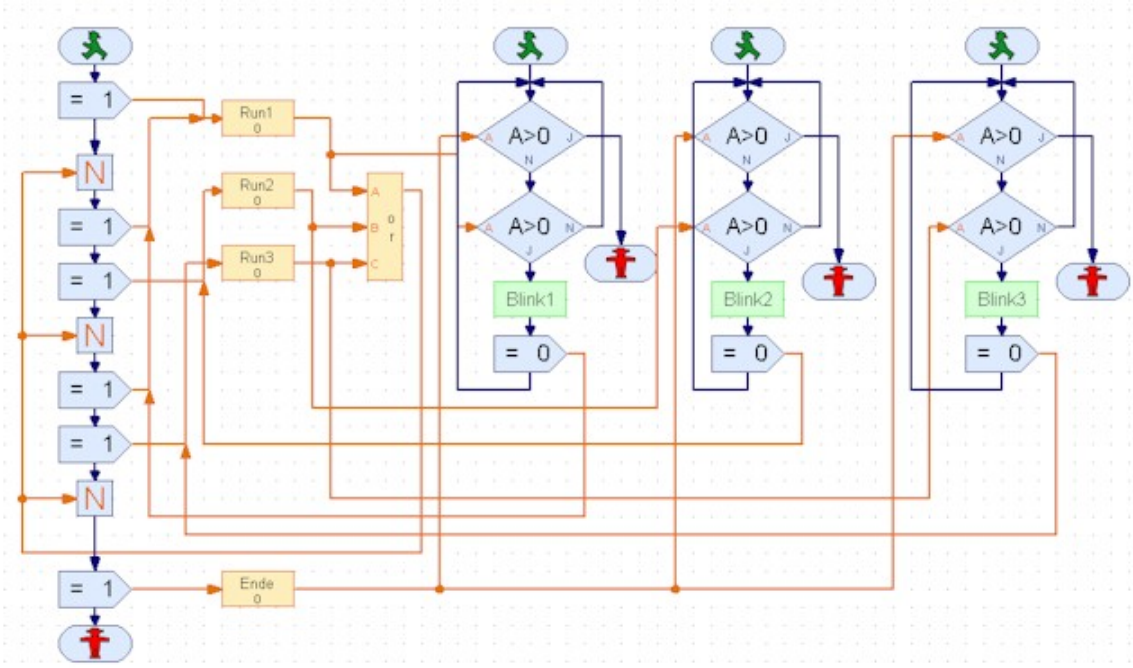
Noch mehr Prozesse



Beim vorhergehenden Beispiel startet der Thread des Hauptprogrammes zusammen mit dem Hauptprogramn, nur dessen Beendigung wurde über den ersten Thread gesteuert. Wenn man denn mehrere Threads hat und die dann gezielt anstoßen will, muß man das auch steuern.

Hier wird im ersten Thread zu Demo-Zwecken zuerst der Thread mit Blink1, dann die Threads mit Blink1 und Blink2 und zum Schluß noch Blink1 und Blink3 angestoßen. Die Blink-Routinen enthalten das Nutzprogramm (hier ein schlichtes Blinken an einem O-Ausgang). Wenn die jeweilige Blink-Routine beendet ist, wird die entsprechende Run-Variable auf 0 gesetzt. Im ersten Thread wird solange gewartet bis Blink-Routinen ihre Arbeit getan haben. Das ist der Fall, wenn alle Run-Variable den Wert 0 haben (Zusammenführung durch den OR-Befehl).

Noch schöner



Wenn man sich die Angelegenheit genauer ansieht, bleibt das vorherige Programm nach getaner Arbeit auf dem Ampelmännchen bzw. den J's stehen, es wird also nicht beendet (Dialog-Box Ende). Bei "endlos" laufenden Programmen ist es zu vertreten, einfach einen "roten Button" zu bemühen um dem Programm ein Ende zu bereiten. Kunstgerechter jedoch ist ein programmiertes Ende. Und das erfordert nochmals Aufwand.

Die Ende Variable, die vor dem roten Ampelmännchen gesetzt wird und eine regelrechte Abfrageschleife zu Beginn eines jeden Threads (Feierabend, Arbeiten, Warten). Bei Feierabend wird auch noch ein Ampelmännchen bemüht um dem jeweiligen Thread ein Ende zu machen.

Source in rbThread.ZIP

Noch mehr Prozesse findet man in der Taktstrasse

Stand : 14.01.2007