PoelControl - Application Logic

1.Status

- 2006-04-16: First draft with images, description etc.

2.Intro

PoelControl is a project aimed at the development of a flexible and powerful microcontroller based solutions for the control of an installation for vegetable fuel (poel) as installed here in my vehicle.

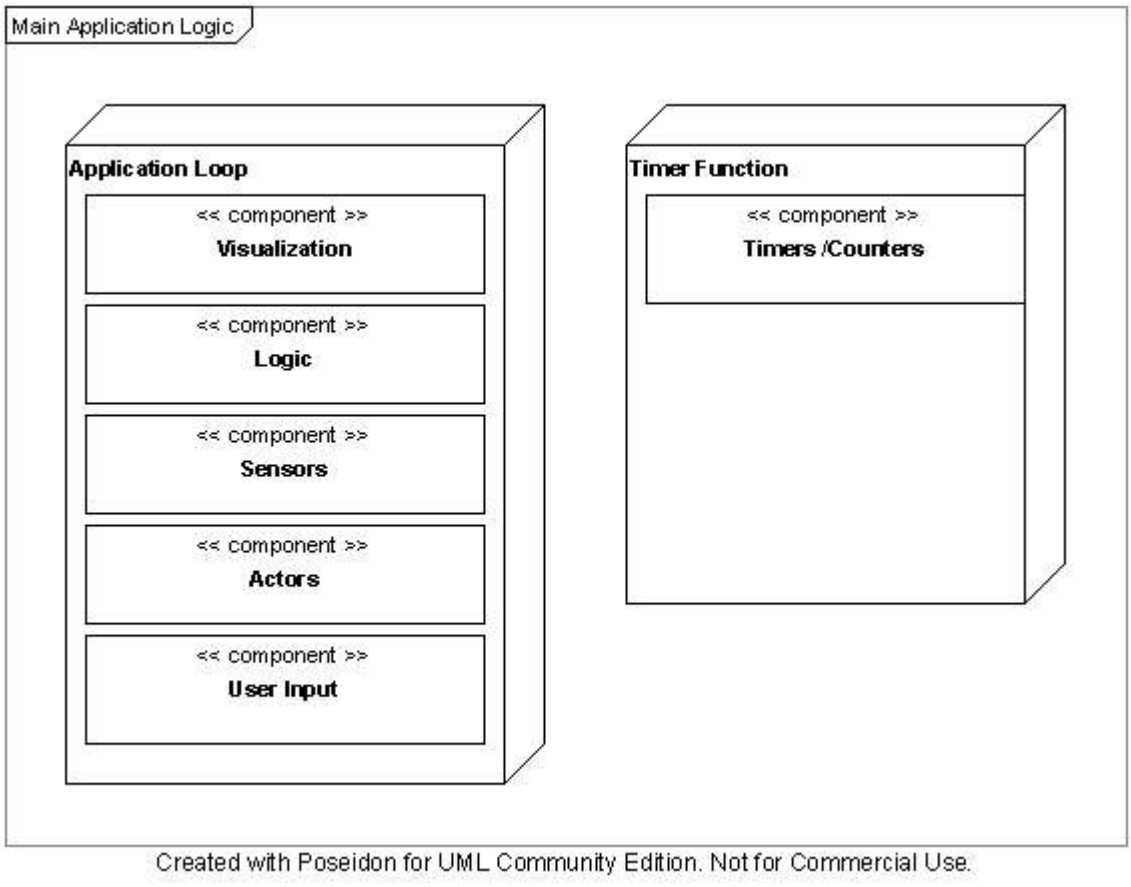
The objective of this document is to give an overview over the main application logic implemented.

3.Logic

The application can be devided in the following blocks of functionality or modules:

- **visualization:** Displays status information for the user
- **logic:** The main control logic for the poel conversion
- **sensor:** Samples data from the input sensors and provides the data for the other modules
- **user input:** Checks the keys for user input and reacts on it
- **actors:** Acts on the results of the *logic* module

In UML notation this can be described as follows:



PoelControl - Application Logic

Mostly, the application uses global variables to provide status information, data values etc. for all modules. Often global flags are set in one module to notify other modules about important conditions.

The main application logic consists of two independent parts:

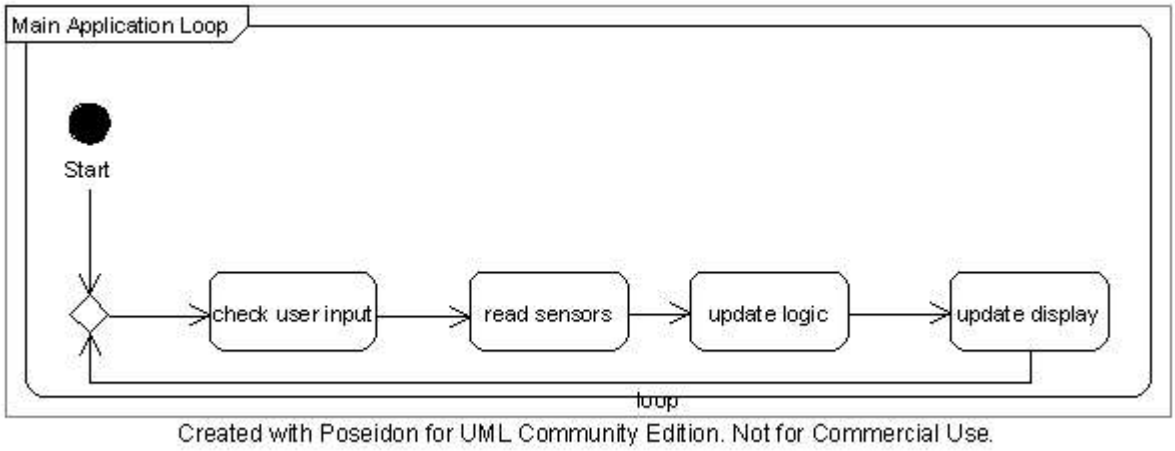
- the main application loop
- a timer interrupt based method

Timer

Upon *timer0* timer overflow – see documentation *Timers* – the corresponding method in *timer.c* is called. This method is used to keep track of the time elapsed since application startup. 10ms and 100ms counters are updated. Also, for the blocks of functionality mentioned above separate 100ms counters <*moduleName*>*Ticks* are kept.

Application Loop

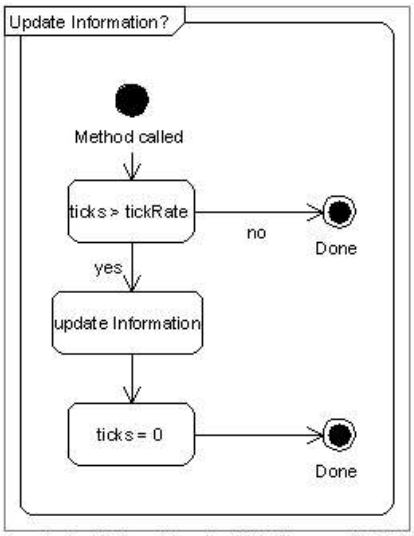
The main application loop continously calls the controller methods for all modules except *actors*[1].



To avoid flicker on the display etc. the <*moduleName*>*TickRate* counters are evaluated. Except for the *user input* module an update or check is performed only every 100ms * <n> intervall. For each module <n> is defined in the modules header file by *pc*<*modulename*>*TickRate.*

1 Actor action are automatically called from the *logic* module.

PoelControl - Application Logic



Created with Poseidon for UML Community Edition. Not for Commercial Use.