EOS2 - Extension Operating System

Author: Stefan Ehrlich

WHAT IS EOS2?

EOS2 is a fully object oriented real time operation system written in C++, which abstracts the hardware- and underlying OS-resources from the application with a common interface.

EOS2 is designed to be easy portable from small µC systems like Atmel's atmega (e.g. Arduino™ uno, mega, etc.) up to ARM, x86/x64 systems (e.g. Raspberry Pi™, Linux™- or Windows™-computer).

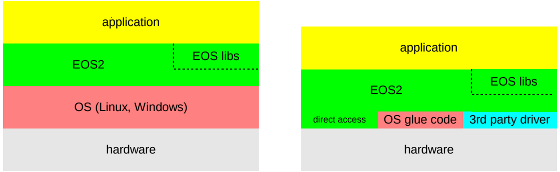

Fig. 1. Layer model of EOS2 abstraction (left with high level OS e.g. Raspberry Pi, right with rudimental OS e.g. Arduino)

HARDWARE ABSTRACTION

Each hardware resource, which might be used by the application is represented by an object. This starts at simple input-/output-pins going up to TCP communication.

```
void CMyTask::onInit()
{
    Pin12.toDigitalInput();
    Pin13.toDigitalOutput();
    Pin3.toAnalogOutput();
    PinA0.toAnalogInput();
    I2C.toI2CMaster();
}

void CMyTask::on10ms()
{
    // Every 10 ms copy the inverted input
    // of pin 12 to pin 13
    Pin13 = !Pin12;
    // invert pin 3 with analog input A0
    Pin3 = PinA0.getAnalogResolution() –
           PinA0;
    // write to the I2C bus
    I2C.write(nAddress,1,0x81);
}
```

SCHEDULER

EOS2 uses its own internal real time scheduler. The specialty of this scheduler is the way how to request a member function to be called.

The scheduler supports calling initialization functions, idle functions and periodic functions.

```
onInit(this,&CMyTask::doMyInit);
onIdle(this,&CMyTask::doWhatEver);
schedule(this,&CMyTask::on10ms,10);
```

TIMER BASED SCHEDULING

The EOS2 scheduler supports the execution of periodic tasks, which are triggered by a common timer. The periodic tasks do not interrupt other running tasks, so there is a possibility of having a jitter depending on the execution time of the currently running task. The time between 2 periodic tasks is used to execute idle tasks, which use the remaining CPU calculation time.

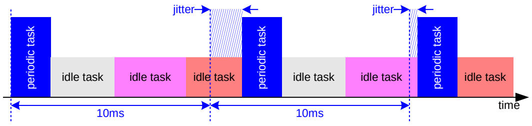

Fig. 2. Non preemptive real time scheduling

REAL TIME SCHEDULER

The EOS2 scheduler provides the execution of real time tasks, which will interrupt currently running tasks with lower priority.

```
schedule(this,&CMyTask::doSomething,6);
```

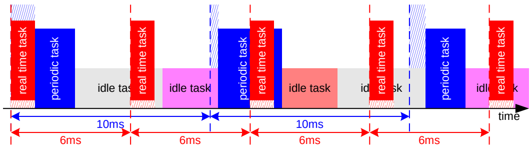

Fig. 3. Interrupt driven resp. preemptive scheduling

EOS2 provides classes to access data being used by real-time- and non-real-time-tasks.

DELAYED TASKS

If a task needs to be executed with a delay, EOS2 supports this type of scheduling.

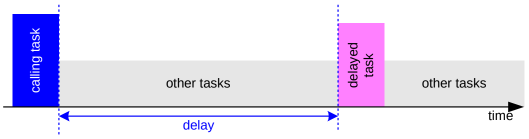

Fig. 4. Delaying the execution of a task (e.g. timeout)

EVENTS

A powerful possibility for task synchronization within EOS2 are events. Each task can generate an event or be a consumer of an event.

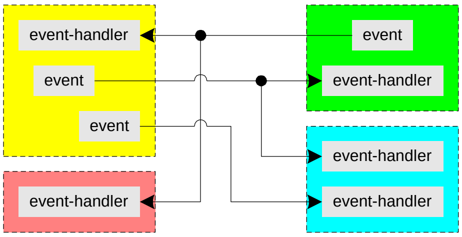

Fig. 5. Triggered events will call the attached event handler functions.

The way how to connect a function to an event is similar to the scheduler functions.

```
EventA.add(this,&CMyClass::onA);
EventA.add(this,&CMyClass::onA1);
[…]
EventA.trigger(); // call onA and onA1
```