Semester 1 - 6 points

Aims
Course outcomes:
- Students will gain in-depth technical competence in the following areas:
  * Selection of appropriate sensors, actuators, and interfaces for solving a particular technical problem
  * Designing an embedded system hardware with specific sensors and actuators
  * Programming an embedded system with a combination of C and assembly language
- Students will be aware of typical embedded system application areas and will be able to design an embedded system to solve problems in these application areas
- Students will be able to use high-level system function units for designing embedded computer systems and be able to verify their performance and use it to modify the design
- The course will involve lab sessions where students work in groups, where they have the opportunity to work as team leaders or effective team members, which are being assigned specific sub-tasks. Students will rotate their roles as team leaders and effective team members.
- Students will learn about special demands of life-critical embedded systems in application areas like medical systems, transportation systems and control of manufacturing machinery.
- Students are provided with a foundation in the embedded systems area. This is a rapidly expanding area, which they are encouraged to explore further and keep up with technology trends.

Unit Description
This unit includes the following topics:

Embedded microcontrollers:
- CPU functionality, function units, structured CPU design, embedded system design, interfacing and integration, reconfigurable computing.

Instrumentation:
- Sensors, actuators, digital and analog I/O, motor drivers, shaft encoders, interface standards, standard protocols.

Programming for embedded systems:
- Assembly languages, C/C++ and interfacing with assembly languages, interrupt handling, timer interrupts, real-time systems, multi-threading, device drivers, data structures, programming techniques.

Contact Hours

<table>
<thead>
<tr>
<th>Type</th>
<th>Hours</th>
<th>Start</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lectures</td>
<td>26 hrs</td>
<td>week 1</td>
<td></td>
</tr>
<tr>
<td>Tutorials</td>
<td>10 hrs</td>
<td>week 4</td>
<td></td>
</tr>
<tr>
<td>Labs</td>
<td>16 hrs</td>
<td>week 5</td>
<td>weekly labs, 3h each, 2h supervised</td>
</tr>
</tbody>
</table>

For days, time and venues, see: [www.timetable.uwa.edu.au](http://www.timetable.uwa.edu.au)

Tutorial and Lab assignments are available on the web.
Students should work in **groups of two**, so please find a partner from the same lab group.
Unit Co-ordinator:  Associate Professor Thomas Bräunl
Tutors / Lab Supervisors: Stephen Whitely, Teoh Soo, James Porteous

Textbooks:  

Recommended Reading:  
Kernighan, Ritchie:  *The C Programming Language*,  
  alternatively:  <any C programming book>

Course Notes:  see link (pdf format)

Lectopia Recording: see link
OLCR Registration: see link

Tutorials:  see link
Tutorial Solutions: see link
Lab Assignments: see link
Previous Exams: see link
Supplem. Material: see link

Marks (Cont. Asses.): see link

Assessment

<table>
<thead>
<tr>
<th>Type</th>
<th>% of final mark</th>
<th>Comments</th>
</tr>
</thead>
</table>
| Lab 1-8      | 30%             | labs are conducted in groups of two students  
  each lab is due at the end of the scheduled session in the **same week** |
| Final examination | 70%         | open book exam                                                           |

Penalties
Assignments and labs will receive a 20% penalty for each day late.

Plagiarism
All work submitted must be the student's (or group's, resp.) own work.  
Citations must be clearly marked as such.  
See the faculty policy on plagiarism.

Scaling
See the faculty policy for scaling marks.

Appeals
See the faculty policy for appeals.