CC3501 Assignment Task Sheet

Semester 2, 2015

Task Description

You are to design and build an embedded system that acquires sensor data. The emphasis is on producing a useful end product. You are being given considerable flexibility in terms of the features that you choose to implement, in order to allow you to target your product at a particular niche.

This is a group project for teams of 3 – 4 students. You are free to organise your own team.

There are two suggested projects, described below. It is also possible for your team to propose an alternative project or a variation of these.

Project #1: non-invasive current metering

Using CT (current transformer) sensors, monitor the AC current in external circuits. CT sensors will be supplied that produce an AC voltage in proportion to the current being measured. You will sense this voltage and thereby calculate the current that is flowing.

Your system must be capable of simultaneously monitoring at least 4 circuits per microcontroller.

You must implement **at least one** of the following features:

- A sensor node that operates without access to mains electricity, e.g. using built-in batteries and/or scavenging power. If your device is battery powered then you do not need to implement wireless communication.
- Wireless communication between sensor node(s) and a base station, with near to real-time data upload to the Internet. Your base station will be a Raspberry Pi.

Hints:

- This project favours electronics and circuit design. You will need to implement some analog circuitry to sample the AC voltage, e.g. by DC biasing into the midpoint of the ADC range.

Project #2: distributed data logging

Develop a wireless sensor network that can be deployed for general purpose data logging applications. You will demonstrate the capability of your system by monitoring environmental conditions such as temperature, but your design must be capable of interfacing with arbitrary analog and/or digital sensors with minimal set-up time. Ideally, it should be possible to add new sensors by simply reconfiguring your application; even better is if this reconfiguration can be done without reprogramming the board.

For example, you might provide several ports for analog voltage measurements and several ports that connect to standard digital protocols such as a UART and I2C.

The complexity in this project will be in the software. You must implement highly robust data logging. For example, temporary communication failures must not result in the loss of data. If the base station is temporarily unreachable, your sensor nodes must store the data locally until the base station becomes available again. If there is a temporary loss of power, then you must provide reasonable protection against data loss, e.g. by storing measurements to non-volatile memory.

In addition, you must implement **at least one** of the following features:

- A sensor node that operates without access to mains electricity, e.g. using built-in batteries and/or scavenging power. Note that your data logger still needs a wireless data link to the base station, but this need not be real-time.
- Accurate temperature readings that are calibrated against external references, a realistic estimate of the errors in your data, and documentation justifying how you performed this calibration.

Hints:

- This project favours software design. You will need to carefully consider how to make your design as general as possible to make it easy to add new types of sensor in the future.

Project #3: your own proposal

You are free to propose your own project (or a variation of the above). Your proposal must be accepted by the subject coordinator as being of sufficient complexity for a third year class. Typically you would be expected to demonstrate:

- The acquisition of data from sensor(s),
- The communication and cooperation of at least two independent CPUs, and
- The storage of sensor data using an Internet-based service and/or persistent memory on board your device(s).

These requirements may be relaxed if there is a corresponding increase in complexity elsewhere in your design.

Custom projects must be approved on or before Monday 24 August 2015.

Due Dates

Group allocations	Monday, Week 5 (24 August)
PCB schematics and board layouts due	Friday, Week 8 (18 September)
Approximate date from which PCBs will be available for collection	End of week 9 or start of lecture recess
Software and documentation due; demonstration of final product	Lab session, Week 13

Technical design requirements

You may choose to build either a "shield" that connects on top of a FRDM board, or a standalone microprocessor board that contains all the required elements in a single PCB.

- If you choose to build a shield, then you will be required to show a higher standard of software design to make up for the simplified hardware design.
- If you choose to build a standalone microprocessor board, you are responsible for choosing the microprocessor but a Freescale Kinetis K20 is recommended so that your software can be prototyped on the FRDM board first. The university will supply programmable cables for Kinetis microprocessors only.

Conditions

You are to treat this as a genuine engineering project. You are responsible for the entire design including the choice of components. You must submit a bill of materials. Your materials will be purchased from the subject budget (subject to restrictions; see below). You have access to the School's technical workshop facilities.

Budget

A budget of approximately $80 per group is available for the purchase of electrical consumables. This is in addition to the equipment already supplied as part of the subject, e.g. FRDM boards, Raspberry Pis, XBee radios, etc.

If you require more than this, contact the subject coordinator ASAP. Additional funding may be provided (subject to availability) for reasonable requests that will substantially benefit your project.

Technical report

Your group must submit a report detailed your designs and justifying your choices. **There is a strict limit of 3000 words, but a good report is almost certainly shorter than that.** It is not sufficient to simply describe what you have done; instead, you must critically assess your design choices and make recommendations for a future engineer who might revise your work.

In an Appendix (which is not counted in the word limit), you must include circuit schematic(s) and a reference for your communication protocol(s) and data format(s). The level of detail must be sufficient for someone else to design a system that speaks the same protocol as yours.

User manual

You must also produce a non-technical user manual that clearly describes the operation of your system. This document will be brief (because users are unlikely to read lengthy manuals). It must give guidance on setup/installation and troubleshooting.

Assessment Criteria

Component (weighting)	High standard	Medium standard	Minimum standard	Fail
Design (20%). Primarily judged from the description given in the documentation and from inspection of the CAD files.	The entire design is elegant and very likely to perform to a high standard.	The design is likely to be effective. There may be some minor issues.	The design is likely to perform as required, once several minor errors are corrected.	The design is not likely to perform as required. Major errors are present.
Documentation in two parts: (1) a technical report describing and justifying the design; and (2) a user manual written for a non-technical audience. (30%)	The documentation is clear and easy to read. The technical report justifies the design choices very well. The documents are of a professional appearance and are written in correct Australian English.	The documentation is mostly clear and easy to read. Appropriate justifications are given for the major design choices. The documents have only minor errors in spelling and grammar.	The documentation is sufficient to explain the design and its operation. There are only minor errors in spelling and grammar.	The documentation fails to explain the design, or is unprofessional in appearance or style.
Product (40%)	The overall system is of a highly professional standard. It competently achieves its objectives, works effectively and reliably, and is easy to use.	The overall system is mostly of a professional standard. Its objectives are generally met and the system is functional.	The overall system meets most of its requirements, but there are some issues with reliability and/or ease of use.	The system is not functional or does not satisfy the requirements of the task.
Software code (10%)	The source code is neat, easy to read, and well commented. The software is modular and would be easy to extend.	The source code is mostly neat and easy to read, with good comments in most places. The software is modular and would be easy to extend.	The source code is legible but not always clear. Comments are sparse. Extending the software would not be easy.	The source code is difficult to understand and poorly commented.